

**СПРАВОЧНИК
ПО ПЕРСОНАЛНИ
КОМПЮТРИ**



КИРИЛ Л. БОЯНОВ
НИКОЛА Г. ВЕЦЕВ
СТЕФАН С. МАЧЕВ
ВЛАДИМИР П. ПАНТЕЛЕЕВ
АЛЕКСАНДЪР Г. АЛЕКСАНДРОВ
ВЛАДИМИР Р. ЧИЛОВ
ХРИСТО Г. ХРИСТОВ

СПРАВОЧНИК

ПО ПЕРСОНАЛНИ КОМПЮТРИ

Под общата редакция
на ст. н. с. I ст., д. т. н. Кирил Л. Боянов

ДЪРЖАВНО ИЗДАТЕЛСТВО „ТЕХНИКА“ — СОФИЯ, 1988

В справочника са дадени сведения за широко разпространените у нас персонални компютри от фамилията Правец-8 и Правец-16, а също за компютрите, използващи операционната система CP/M. Разгледани са основните принципи на работа, архитектурата на компютрите и програмното им осигуряване. Особено внимание е обърнато на фамилията Правец-16 и съвместимите с нея персонални компютри, като са включени данни за използваните микропроцесорни комплекти и приетите схемни решения. Дадени са и указания, необходими на потребителите при самостоятелна работа с персоналните компютри. Разгледано е базовото програмно осигуряване за отделните компютри, а за фамилията Правец-16 са дадени основни сведения за най-широко разпространените пакети приложни програми.

Справочникът е предназначен за инженери, разработващи микрокомпютърни системи, и за широк кръг специалисти с различни професии, които усвояват приложението на персоналните компютри в своята работа.

Кирил Любенов Боянов
Никола Георгиев Вещев
Стефан Спасов Мачев
Владимир Петров Пантелеев
Александър Георгиев Александров
Владимир Русланов Чинлов
Христо Георгиев Христов, 1988
с о Jusatort, Sofia

ВЪВЕДЕНИЕ

Основната разлика в работата с широко разпространените днес персонални компютри (ПК) и с други цифрови електронноизчислителни машини (ЦЕИМ) е в това, че персоналните компютри се използват индивидуално от потребителите, което позволява да се постигне по-голяма гъвкавост и ефективност. Развитието на комуникациите позволи персоналните компютри да се включват към различни типове големи или малки ЦЕИМ. Това означава, че потребители, притежаващи ПК, могат да станат абонати на мрежи от ЦЕИМ и да ползват общите ресурси на мрежата.

Създаването на персоналните компютри се дължи преди всичко на появата и развитието на интегралните схеми с голяма степен на интеграция и на микропроцесорите. Отначало микропроцесорите са се използвали за решаване на специфични задачи. Intel 8008, който е един от първите микропроцесорни чипове, изпълнява елементарни функции, затова фирмата го обявява като микрокомпютър. Първият ПК с търговско значение е Altair 8800, създаден през 1975 г. След 1977 г. няколко фирми започват да предлагат персонални компютри: Apple II на Apple Computer Inc., PET на Commodore Business Machines, TRS80 на Tandy-Radio Shack. Дължината на думата на тези компютри е 8 разряда, а по това време не е имало програми, които да работят на компютър с такава дължина на думата. В следващите години разпространението на персоналните компютри предизвиква силно развитие на програмните продукти за тях, като се създава цяла промишленост. Междувременно се появяват 16-разредните ПК, които намират добър прием сред потребителите. По параметри съвременните персонални компютри приближават или превишават възможностите на малките или даже на някои класове големи ЦЕИМ. По различни признаци, като размер, апаратна реализация и предназначение, ПК могат да се отнесат към различни групи, например: преносими, едноплаткови, домашни, универсални и т.н. Особен интерес представляват универсалните персонални компютри, като понякога те се разделят на персонални и професионални. В книгата е прието обобщено наименование персонален компютър.

Производителите на персонални компютри са изключително много и постоянно се увеличават. При това на потребителя е трудно да направи избор на съответен тип ПК, тъй като съществува голямо разнообразие от модели в различни конфигурации.

За да се избере един ПК, от особено значение е да се дефинира точно какво се иска и какво се очаква от техническите и от програмните средства. Най-често ПК се използват за: извършване на инженерни пресмятания, контрол на апаратура за управление на производствени процеси, финансово-счетоводни операции, текстообработка, електронна поща, планиране, изучаване на алгоритмични езици, електронни игри, комуникации с други компютри, управление на складово стопанство и следене на наличността, информационни услуги и др.

За да се направи сполучлив избор както на техническите средства, така и на пакетите с приложни програми, необходимо е да се познаят

добре отделните устройства и програмното осигуряване за ПК. При това обикновено се използва модулният подход, т.е. отначало се купува версия на продукт с по-малки възможности, като след нарастване на изискванията или потребностите се купуват допълнителни компоненти, които разширяват функциите. Необходимо е да се познават възможностите на отделните устройства, за да се прецени доколко те отговарят на предявените изисквания и какви евентуални допълнителни средства биха били необходими.

Съвременните ПК работят под управлението на операционни системи, някои от основните компоненти на които се съхраняват в постоянната памет на компютъра заедно с транслятора на съответния алгоритмичен език. В редица случаи това е интерпретатор на БЕЙСИК.

Обикновено приложните програми се продават отделно като пакети или се разработват от потребителите като програми с определено предназначение.

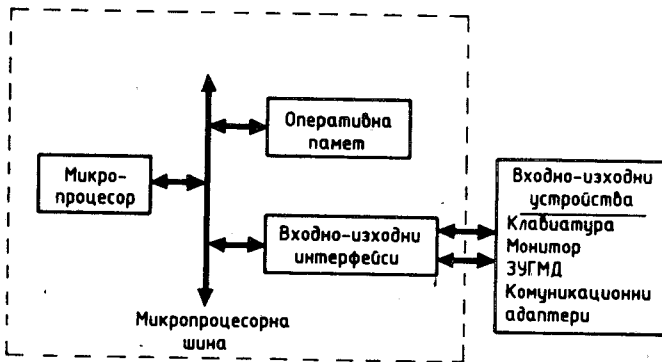
У нас са разпространени най-вече 8- и 16-битови персонални компютри от фамилията Правец, както и ПК от Единната система ЕИМ. Съществуват и ограничен брой ПК от фамилията Apple и IBM. Изложението в справочника е ориентирано предимно към ПК от фамилията Правец.

Справочникът има за цел да запознае потребителите с най-същественото за техническите средства и програмното осигуряване, за да могат да изберат подходяща конфигурация. Освен това специалисти и потребители могат да намерят в него сведения, необходими при работата с тези фамилии персонални компютри. Авторският колектив изказва благодарност на Лиляна Христова, Николай Пъчев, Нася Пенчева и Владимир Гетов за съветите при оформяне на ръкописа, както и на рецензентите доц. Павел Мартинов и инж. Алекси Боюклиев за направените бележки и препоръки. Отделно трябва да се отбележи активната редакторска намеса на инж. Румяна Ковачева.

ОБЩИ СВЕДЕНИЯ ЗА ПЕРСОНАЛНИТЕ КОМПЮТРИ

1.1. МИКРОПРОЦЕСОРНА СИСТЕМА

Блоквата схема на една микропроцесорна система е аналогична на тази на една универсална цифрова електронноизчислителна машина. На фиг.1.1 са показани схематично основните блокове на микропроцесорната система.



Фиг.1.1. Основни блокове на микропроцесорната система

Основна структурна единица е микропроцесорът (МП), който обработва информацията в съответствие с дадена програма. Инструкциите, които я съставят, се подават на неговия вход за изпълнение. Той може да бъде реализиран на една или няколко големи интегрални схеми (ГИС). Освен това за неговото функциониране са необходими и допълнителни схеми, които изпълняват функции по синхронизиране, съгласуване между функционалните блокове и т.н. Съвкупността от тези интегрални схеми, които са съвместими помежду си по конструктивно и технологично изпълнение и са предназначени за съвместно използване при изграждане на микропроцесорни системи, представлява *микропроцесорен комплект* (или *микропроцесорна фамилия*). Микропроцесорите изпълняват аритметични и логически операции, анализират и взимат решения, които изменят процеса на пресмятане, управляват входно-изходните процеси, т.е. реализират функции, които обикновено се възлагат на централния процесор на една ЦЕИМ.

Появата на микропроцесорите е свързана преди всичко с успехите на интегралната технология. Стремешът на производителите на ГИС да удовлетворят търсенето на устройства с различно предназначение доведе до създаването на програмно управлявани универсални кристали. За целта към централния процесор бяха включени памет и схеми за управление

на обмена с външните устройства и фактически бе реализирана ЦЕИМ на един кристал. Първите МП бяха предназначени за използване съответно в електронните калкулатори и видеотерминали, но благодарение на тяхната универсалност намериха широко приложение и в други устройства на цифровата техника.

Основен блок на всеки микропроцесор е аритметично-логическото устройство. Освен него във всеки микропроцесор има бързодействаща памет за съхраняване на операндите и междинните резултати. Това са регистрите с общо предназначение, като между тях често се отделя акумулатор — регистър, в който се съхранява операндът и където се изпраща резултатът от операцията. Има и регистри със специално предназначение: брояч на инструкциите, регистър на инструкцията, флагове и указатели на стека. Към допълнителните блокове се отнасят дешифраторът на инструкциите, схемите за организация на машинните цикли, схемите за синхронизация, схемите за формиране на сигналите, необходими за управление на обмена с паметта и външните устройства. Помощно предназначение имат и различните формиратели — схеми за увеличаване на товароспособността, и съгласуващите схеми при свързване на МП с различни типове технологии. Освен блоковете, които влизат в състава на МП, като ГИС се произвеждат оперативни паметни, постоянни паметни за съхраняване на данни, таймери, портове за вход-изход, аналогово-цифрови преобразуватели, цифрово-аналогови преобразуватели и др.

Микропроцесорните ГИС се оценяват въз основа на функционалното предназначение, техническите характеристики и архитектурните им особености. По функционално предназначение микропроцесорните ГИС могат да бъдат разделени на два класа: универсални и специализирани.

Към универсалните микропроцесори отнасяме българските микропроцесори СМ601 и СМ650, съветските фамилии: КР580ИК80А, К1810ВМ86, К1801ВМ1А(В), К1801ВМ3, К586ИК1. От западните фирми такива предлагат: Intel — I4004, I8008, I8080, I8085, I8086, I8088, I80186, I80286, iAPX432(01;02), I80188, I80386; Motorola — МС6800, МС6809, МС68000, МС68010, МС68020; Zilog — Z80 (Z80А, Z80В, Z8400Н), Z8000; Z80000 и др.

От специализираните микропроцесорни ГИС могат да се отделят няколко групи: процесорни елементи (АЛУ и група от регистри), блокове за управление, схеми за формиране на адресите на микрокомандите, блокове и регистри със специално предназначение, блокове и регистри с общо предназначение, схеми за синхронизация, генератори на тактови импулси, схеми за управление на обмена с паметта, паралелни периферни адаптери, буферни регистри, контролери за директен достъп до паметта, програмируеми таймери, контролери за протоколи, шинни формиратели, контролери за вход-изход. Трябва да се има предвид, че точна граница между тези подгрупи няма, тъй като съществуват микропроцесорни ГИС, които могат да изпълняват няколко от горните функции едновременно.

Сред процесорните елементи, произвеждани в СССР, трябва да посочим: К1800ВС1, К1802ВС1, К1804ВС1, К1804ВС2, К581ИК1, КР582ИК1, КР583ВС1 (универсална микропроцесорна секция), КР583ВМ1, КР584ИК1, КР587ИК2, КР588ВС2; КР589ИК02.

От продукцията на западните фирми ще отбележим: МС10800, Am2901, Am2903 (AMD), I3002.

По-долу са приведени означенията на най-използваните ГИС у нас, производство на СССР, НРБ и на основните западни производители:

—блокове за управление и схеми за формиране на адреса на микрокомандата: К581ИК2, КР583ИК1, КР584ВУ1, КР587РП1, КР588ВУ2 (програмируема логическа матрица с памет), К589ИК1, К1800ВУ1;

—схеми за управление на адреса на микрокомандата: К1804ВУ1, К1804ВУ2, К1804ВУ3;

—блокове регистри с общо предназначение: КР1802ИР1, КР1804ИР1;

—схеми за синхронизация и генератори на тактови импулси — КР580ГФ24, К1800ВБ2, КР580ГФ84, I8224, I8284, I82284, МС10802, МС6870, МС6871, МС6875;

—схеми за управление на обмена с паметта: КР588ВГ2, К1800ВТ3, К1801ВП1-30 и др.;

—периферни адаптери и комутатори: КР580ВВ55, КР583ХЛ1 (универсален магистрален комутатор), КР583КП1 (магистрален комутатор), КР583КП2 (магистрален приемопредавател), КР584КП1 (магистрален приемопредавател), К586ИК2, КР587ИК1, КР588ВА1, К1801ВП1-34, I8255, I8291 (интерфейс с шина по стандарт IEEE488), МС6820, МС6822, МС68488, Z80P10, Z8036, Z8038, Z8060, СМ602 (НРБ);

—схеми за управление на последователния обмен: КР580ИК51, КР581ВА1, К1801ВП1-35, К1809ВВ2, I8251, МС6850, МС6852, МС6854, Z80SIO, Z8030 и произвежданите у нас СМ603, СМ604, СМ674, СМ621, СМ622;

—буферни регистри — КР588ИР1, К589ИР12, I8212, I3212;

—контролери за директен достъп до паметта: КР580ВТ57, I8257, I8237, Am2940, МС6844, Z80DMA, СМ608 (НРБ);

—контролери за управление на прекъсванията: КР580ВН59, К589ИК14, I8259, I3214, I8214, МС6828;

—програмируеми таймери. КР580ВИ453, I8253, МС6840, Z80СТС, СМ606 (НРБ);

—контролери за управление на клавиатура: КР580ВВ79, I8279;

—контролери за управление на видеомонитори: КР580ВГ75, К1809ВГ3, СМ607 (НРБ), I8275, I82720, МС6845;

—контролери за управление на ЗУГМД: КР1818ВГ93, СМ609 (НРБ), I8271, МС6843, МС6849, Am9580, Am9581;

—контролери за управление на ЗУМД: WD1010, WD1014, I82062, СМ610 (НРБ);

—ускорители на изчисленията: КР587ИК3, КР588ВГ3 (умножител 16×16), К1800ВР8 (изместващ регистър), КР1802ВР2 (схема за последователно умножение), МС10808 (изместващ регистър), I8087, I80287, I8032 (процесор за плаваща запетая), I8231 (аритметичен процесор);

—контролери за шини, системни контролери, протоколни контролери: КР580ВГ18, КР588ВГ1, КР580ВК28, КР580К38, I8218, I8288, I8228, I8238 (системни контролери), iAPX43203, iAPX43204 (интерфейсни контролери), I8273 (протоколен контролер за HDLC/SDLC), I8292 (контролер за шина по стандарта IEEE488);

—шинни формируватели: КР580ВА86, КР580ВА87, К531АП2, К589АП16, К589АП26, К1800ВА4 (двупосочен преобразувател на нива), К1800ВА7 (двупосочен магистрален приемопредавател), I8286, I8287;

—входно-изходни контролери: К1801ВП1-33 (контролер за интерфейс ИРПР), I8089 (входно-изходен процесор), I8041/I8741 (универсален интерфейс), Z8034 (универсален периферен контролер).

1.2. ТЕХНИЧЕСКИ ХАРАКТЕРИСТИКИ НА МИКРОПРОЦЕСОРНИТЕ ИНТЕГРАЛНИ СХЕМИ

Технологията на микропроцесорните интегрални схеми е един от важните фактори, които определят основните технически характеристики на схемата. Най-разпространени за микропроцесорните системи са MOS и биполярната технология: PMOS, NMOS, CMOS, DMOS, VMOS, I²L, I²L, TTLS, ECL.

От степента на развитие на технологията зависи броят на ГИС, необходими за реализация на микропроцесорния модул, който изпълнява функциите на централния процесор. Към характеристиките на силициевия кристал на микропроцесорните ГИС се отнасят и размерът на кристала, количеството активни елементи в един кристал, количеството изводи, типът на корпуса. Бързодействието на ГИС може да се оцени със следните характеристики: брой на операциите в секунда, тактова честота, време за изпълнение на операция над данни, които се съхраняват в акумулатора. Най-важните електрически параметри на микропроцесорите са: брой на захранващите източници и тяхното напрежение, консумиран ток, консумирана мощност, брой и амплитуда на синхронизиращите импулси, входен и изходен капацитет, товарно съпротивление, време за закъснение на сигналите.

Изходният ток и изходният капацитет определят товарната способност по изход, т.е. броя на входовете, които могат да бъдат включени към един изход. Тези параметри са важни за ГИС, когато се търси съвместимост между различни типове логика.

В зависимост от условията на експлоатацията на микропроцесорните ГИС към тях могат да бъдат предявени и специални изисквания: температурен диапазон на функциониране и съхранение, маса, устойчивост за работа в радиационни и електромагнитни полета, устойчивост към вибрации в определен честотен диапазон, надеждност, в която се включват коефициент на готовност и интензивност на отказите. В табл.1.1 са дадени основните технически характеристики на някои ГИС, производство на СССР.

Таблица 1.1

Серия	Тип на микропроцесора	Технология	Брой операции в секунда	Захранващо напрежение, V	Консумирана мощност, W	Температурен обхват, °C
KP580	KP580ИК80А	NMOS	0.5.10 ⁶	-5; +5; +12	1.5	-10 - +70
KP582	KP582ИК1	I ² L	0.7.10 ⁶	1,2-5	0.29	-10 - +85
KP583	KP583BC1	I ² L	0.1.10 ⁶	1,2-5	0.6	-10 - +70
KP584	KP584ИК1А(В)	I ² L	0.5.10 ⁶	5	0.25	-10 - +70
K586	K586ИК1	NMOS	0.5.10 ⁶	5; 12	0.9	-10 - +70
K587	K587ИК2	CMOS	0.5.10 ⁶	9	0.005	-60 - +85
KP588	KP588BC1А(Б,В)	CMOS	0.5.10 ⁵	5	0.005	-60 - +85
K589	K589ИК02	TTLS	1.10 ⁶	5	1	-10 - +70
K1800	K1800BC1	ECL	3.10 ⁶	5,2; -2	1,7	-10 - +75
K1801	K1801BM1	NMOS	0.5.10 ⁶	5	1,4	-10 - +70
K1802	K1802BC1	TTLS	0.8.10 ⁶	5	1,4	-10 - +70
K1804	K1804BC1	TTLS	0.8.10 ⁶	5	1,5	-10 - +70
K1808	K1808BM1	I ² L	0.3.10 ⁴	5	0.05	-25 - +55
K1810	K1810BM86	VMOS	0.5.10 ⁶	5	1,75	-10 - +70

1.3. ОРГАНИЗАЦИЯ НА УПРАВЛЕНИЕТО И ОБМЕНА В МИКРОПРОЦЕСОРНИТЕ СИСТЕМИ

Съществено влияние на организацията на микропроцесорните системи оказва начинът на реализация на устройството за управление. Различаваме *вътрешно (вградено)* и *външно управление*, всяко от които може да бъде реализирано чрез ГИС или на микропрограмен принцип.

Микропроцесорните системи с външно микропрограмно управление имат възможност за увеличаване броя на инструкциите чрез разширение на микропрограманата памет. Едно външно управляващо устройство може да управлява паралелната работа на няколко специално изпълнени процесорни елемента (микропроцесорни секции), което позволява да се създадат процесори с произволна разредност. За секционните МП с външно управление се използват по-бързи технологии от тези за МП с вътрешно управление.

Усъвършенстването на MOS-технологиите, които се използват за изграждане на еднокристални МП, позволява да се получат 32-и и 64-разредни микропроцесорни варианти с бързодействие няколко милиона операции в секунда. Броят на ГИС, необходими за създаването на секционен процесор, е значително по-голям отколкото за еднокристалния вариант на микропроцесор с вградено управление. Пример на МП с вградено управление е микропроцесорът I8086. Процесорните елементи от серията K1802BC1, K1804BC1 позволяват включването на външни микропрограшни блокове за управление, а освен това в тях са реализирани средства за секционирането им. Засега разпространените ПК са на базата на микропроцесори с постоянна разредност и вътрешно управление.

Особено важна роля при изграждане на микрокомпютрите с ГИС са предвидените средства за поддържане на обмена, реализиращ различни видове взаимодействие между структурните единици. Известни са следните *видове обмен*: *синхронен*, *асинхронен* и *асинхронно-синхронен*. В двата последни случая може да се отделят три фази — *инициализация*, *изпълнение* и *потвърждение*.

При синхронния обмен сигналите за управление, сигналите за адреса и за данните се подават и приемат в строго определени интервали от време (тактове), които се формират от синхронизатора за обмен. Към предимствата на синхронния обмен можем да отнесем: високата скорост, определена от допустимата честота на синхронизатора и бързодействието на източника и приемника, както и относително по-простата реализация, която осигурява по-висока достоверност на обмена. Недостатък на синхронния обмен е това, че той е невъзможен между модули с различна скорост на приемане (предаване) на информацията. Обикновено такъв вид обмен се осъществява между модули с еднакво бързодействие, които винаги са готови за обмен, например микропроцесора и регистровата памет.

При асинхронния обмен във фазата на инициализация се изпраща заявка за обмен и се проверява готовността на партньора за обмен. При получаване на сигнал за потвърждение или установяване на готовност настъпва следващата фаза — изпълнение, при която източникът предава информацията към приемника. В завършващата фаза — потвърждение, приемникът потвърждава факта на приемане на информацията. Последната фаза на обмена често се пропуска. Асинхронният и асинхронно-

синхронният обмен могат да се извършват между модули с различно бързодействие, но скоростта на обмен се определя от възможностите на модула с най-малко бързодействие. Трябва да се отбележи, че всеки от горните видове обмен може да се реализира или по апаратен път, или чрез програмно-апаратни средства, които обикновено са разпределени между модулите на системата. При това в зависимост от сложността на вградените в програмируемите ГИС средства обменът може активно да се управлява или пасивно да се поддържа.

Програмируемите ГИС, ориентирани за управление на обмена с някой определен модул на системата, с шината или с външно устройство, се наричат *контролери* или *специализирани процесори*. Микропроцесорите и специализираните процесори самостоятелно извличат от паметта инструкции, съдържащи управляваща информация (код на операцията), и ако местоположението на операндите не е определено в самия код, извличат се и адресите или самите операнди. Съществуват ГИС, известни като *адаптери*, които изпълняват ролята на буфери, пасивно поддържащи обмена между модулите на системата или между системата и външната среда. Те зареждат управляващите думи за начало на инициализацията, които осигуряват преминаване от един режим в друг. Подобно зареждане се извършва или автоматично с помощта на МП, който активно управлява обмена, или ръчно с помощта на специален интерфейс.

Към класа на МП се отнасят например K1810BM86 и I8086, а към класа на специализираните процесори — процесорът за входно-изходен обмен I8089. Трябва да отбележим, че към класа на контролерите можем да отнесем и програмируемите ГИС, които управляват само фазата на инициализацията на обмена, например контролерът за прекъсване KP580BH59.

В многомодулните системи, при които право на инициализация на обмена има само един модул, например централният процесор, времето на чакане за обмен между другите модули е много голямо. При това ефективността е ниска, тъй като МП трябва постоянно да „опипва“ другите модули, за да открие заявките за обмен. *Последователно опипване* може да се реализира с помощта на вградени средства в ГИС, което позволява по-ефективно използване на МП. В микропроцесорния комплект KP580 такава роля изпълнява схемата KP580BH59. Един от режимите ѝ на работа позволява лесно реализиране на последователно опипване.

Средствата, реализиращи обмен по прекъсване, често се наричат подсистема за прекъсване и обикновено се намират в няколко модула на системата. Различават се средства за *вътрешни, програмни и външни прекъсвания*. Прекъсвания, които се получават в резултат на събития, локализирани вътре в модула на системата, например в МП при деление на нула, се обработват с помощта на средствата за вътрешни прекъсвания. Обикновено те разпознават събитието, съхраняват информацията за текущия процес и зареждат брояча на инструкциите на МП с начален адрес на подпрограма, която обработва дадената ситуация.

Програмните прекъсвания се управляват от специални инструкции, при изпълнение на които в стека се запомня информация за състоянието на текущия процес (програма), съдържанието на регистрите с общо предназначение и на специалните регистри, в това число и на флагите. След това се осигурява зареждането на брояча на инструкциите с ново съдър-

жание, което обикновено се формира от кода, който се съдържа във формата на инструкцията за програмно прекъсване. Възстановяването на прекъснатата програма може да стане както с помощта на специализирани инструкции, така и със служебна подпрограма.

При обслужване на външните прекъсвания се разпознават постъпните заявки за прекъсване, определя се източникът, който подлежи на обслужване, и се анализира текущият процес (програма) с оглед на неговото прекъсване, като се запомня информация за състоянието му. След това управлението се предава на подпрограма, която обслужва дадената заявка, възстановява прекъснатия процес (програма) и връща управлението на прекъснатия процес (програма). В зависимост от начина на реализация на всяка от изброените функции подсистемите за прекъсване могат да се класифицират по следния начин: с маскиране и без маскиране на входните заявки, с приоритетни и с безприоритетни заявки, с възможност за прекъсване на едно или на няколко нива и т.н.

В изчислителните системи често възникват задачи за предаване на големи блокове информация между външните устройства и оперативната памет, например между запомнящите устройства с магнитни дискове и оперативната памет. Скоростта на обмена, която може да осигури МП, често се оказва недостатъчна. Освен това предаването на блокове информация с голям обем в режим на последователно опипване или прекъсване е свързано с големи загуби на време за МП. Ето защо функциите за *директен достъп до паметта* (ДДП) — предаване на информация от външен източник в паметта на системата или обратно, се възлагат на специализирани модули, които се наричат *контролери за директен достъп*. В мултипроцесорните системи с обща шина, където обменът може да се осигури само между един източник и в общия случай между няколко приемника за обмен в режим на ДДП, модулите, които не участват, трябва да се изключат от шината. Подобна възможност е свързана с наличие в по-голямата част от микропроцесорната ГИС на специален режим за захранване. В някои съвременни МП е възможен скоростен обмен в режим на ДДП, например в I80186.

Организирането на обмен между различни физически несъвместими модули е невъзможно без допълнително съгласуване. Средствата, осигуряващи съвместимостта на модулите, се наричат *интерфейси*. Под *външен интерфейс* разбираме съвкупност от средства, осигуряващи взаимодействие между две различни йерархични нива в архитектурата на системата. Под *вътрешен интерфейс* разбираме съвкупност от средства, които осигуряват взаимодействие между структурните единици в дадено ниво на архитектурата. Например интерфейсът за обмен между модулите на системата е вътрешен, а между системата и външната среда — външен. Често пъти външният интерфейс е известен и като *системен интерфейс*.

В понятието интерфейс в по-тесен смисъл на думата се включват технически средства, чрез които се извършва обмен между модулите на системата и външните модули или шини. При това входовете и изходите имат определени стойности за дадената съвкупност от сигналите за управление, данните, адресите, установените нива и т.н. Следователно в понятието интерфейс се включват и процедурите на обмен, съвместимостта на входовете с изходите, което се определя от технологията и възможностите за натоварване.

Разработени са определени стандарти за интерфейси, като най-голя-

мо разпространение за реализация на обмена в микропроцесорните системи са получили МЭК-625-1, който е предназначен за включване на измерителни модули, ОШ и И41 — шинни магистрала за включване на различни модули, ИРПС и СТЫК-2 (RS232) за включване на модули с последователен обмен на информация, ИРПР за включване на модули с паралелен обмен на информацията и модификацията му ИРПР-М (Centronics).

Тъй като съществува различие в бързодействието и интерфейса между модулите памети и външните устройства, предвидени са средства за обмен между микропроцесора и паметта (подсистема за управление на паметта) и средства за обмен с външните устройства (подсистема за вход-изход), които обикновено са разпределени между микропроцесора и специализираните ГИС — интерфейси и контролери. Съществуват два основни подхода при организацията на управлението на обмена в микропроцесорните системи: *управление на паметта и външните устройства с разделено адресно пространство или тяхното разделено управление.*

Първият подход предполага обръщение към външните устройства както към клетки от паметта. По такъв начин адресното пространство, определено от разредността на адресната шина, се разпределя между паметта и външните устройства. Съответно с увеличаването на броя на външните устройства се намалява обемът на непосредствено адресираната памет. Ето защо даденият подход се използва при построяване на системи, които не изискват памет с голям обем. Едно от неговите предимства е това, че за обмен с външните устройства могат да се използват всички видове инструкции за предаване, при които има много начини за адресиране.

При втория подход микропроцесорът или контролерът изработва две непресичащи се множества управляващи сигнали, едното от които е предназначено за паметта, а другото — за външните устройства. При такъв подход цялото адресно пространство може да се използва както за обръщение на модулите към паметта, така и за обръщение към външните устройства.

1.4. НАЧИНИ ЗА АДРЕСИРАНЕ В МИКРОПРОЦЕСОРНИТЕ СИСТЕМИ

В персоналните компютри се използват инструкции, които работят с различни формати от данни. Най-често се използват двоични числа без знак — байт, дума, двойна дума; двоични числа със знак (формат с фиксирана запетая) — байт, дума, двойна дума; двоични числа във формат с плаваща запетая; десетични числа (в зонов и в пакетирания формат). При зоновия формат в един байт се помества едно десетично число, а в пакетирания — две десетични числа. Освен това се работи със символи и низове от символи.

Трябва да се подчертае, че в микропроцесорите съществуват инструкции, които работят с различни формати на данните. Например, въпреки че микропроцесорът КР580ИК80А е 8-разреден и по-голямата част от неговите инструкции работят с байтове, в него са реализирани също инструкции от двубайтовата аритметика. Обикновено 8-разредните

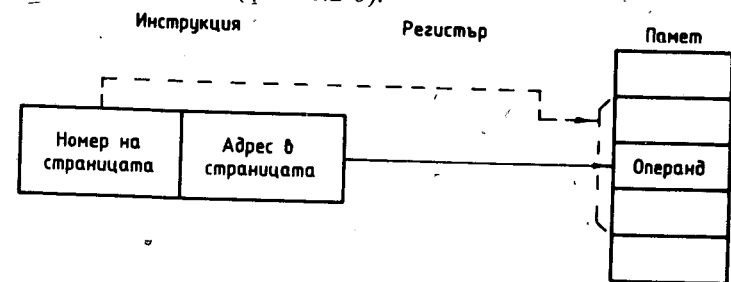
микропроцесори с вградено управление работят с едно-и двубайтови числа без знак. За реализация на операции над числа със знак, десетична аритметика и операции над числа с плаваща запетая се изискват специални програми. В 16- и 32-разредните микропроцесори съществуват операции за десетична аритметика, операции за работа със символи, низове от символи, операции с плаваща запетая и т.н.

Начините за адресиране в общия случай са еднакви както за 8-, така и за 16-разредните ПК. Най-често използваните видове адресация са: странична, сегментна, виртуална, неявна, непосредствена, пряка, пряка регистрова, косвена, индексна, базова, относителна, стекова, автоувеличаваща, автомаляваща и т.н.

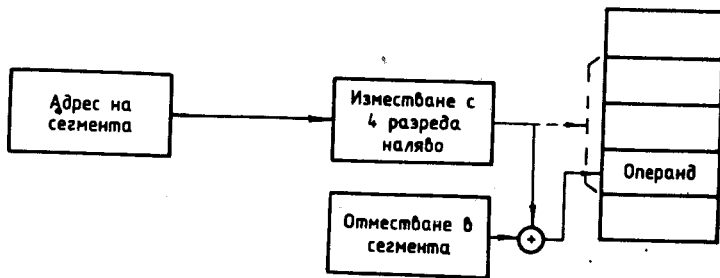
Странична и сегментна адресации. Ограниченото число разреди, което се отделя във формата на инструкцията за адрес на операнда, не позволява директно адресиране на памет с голям обем. Затова паметта се разделя на страници или сегменти. За да се формира физическият адрес, чиято разредност е по-голяма от адресната част, в инструкцията се използват два начина:

— получаване на физическия адрес от адреса на страницата (сегмента) и от адресната част на инструкцията (фиг. 1.2 а);

— изчисляване на физическия адрес, като към адресната част на инструкцията се прибавя съдържанието на сегментния регистър, изместено с 4 разряда наляво (фиг. 1.2 б).



а



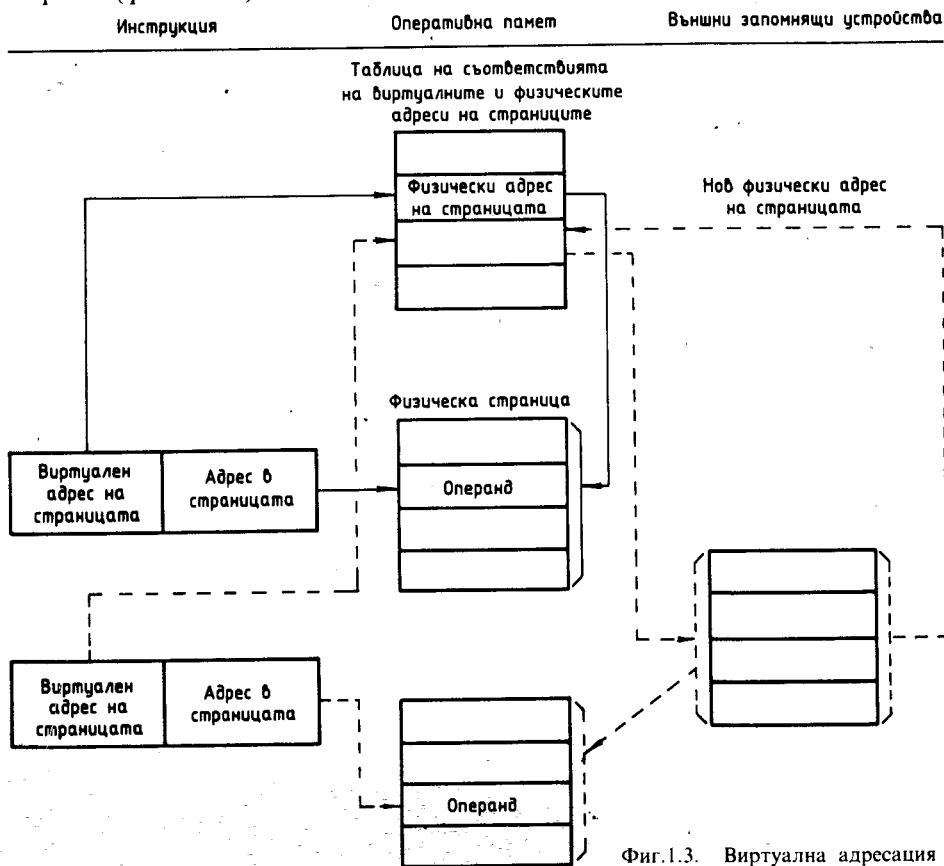
б

Фиг.1.2. Начини за формиране на физическия адрес:

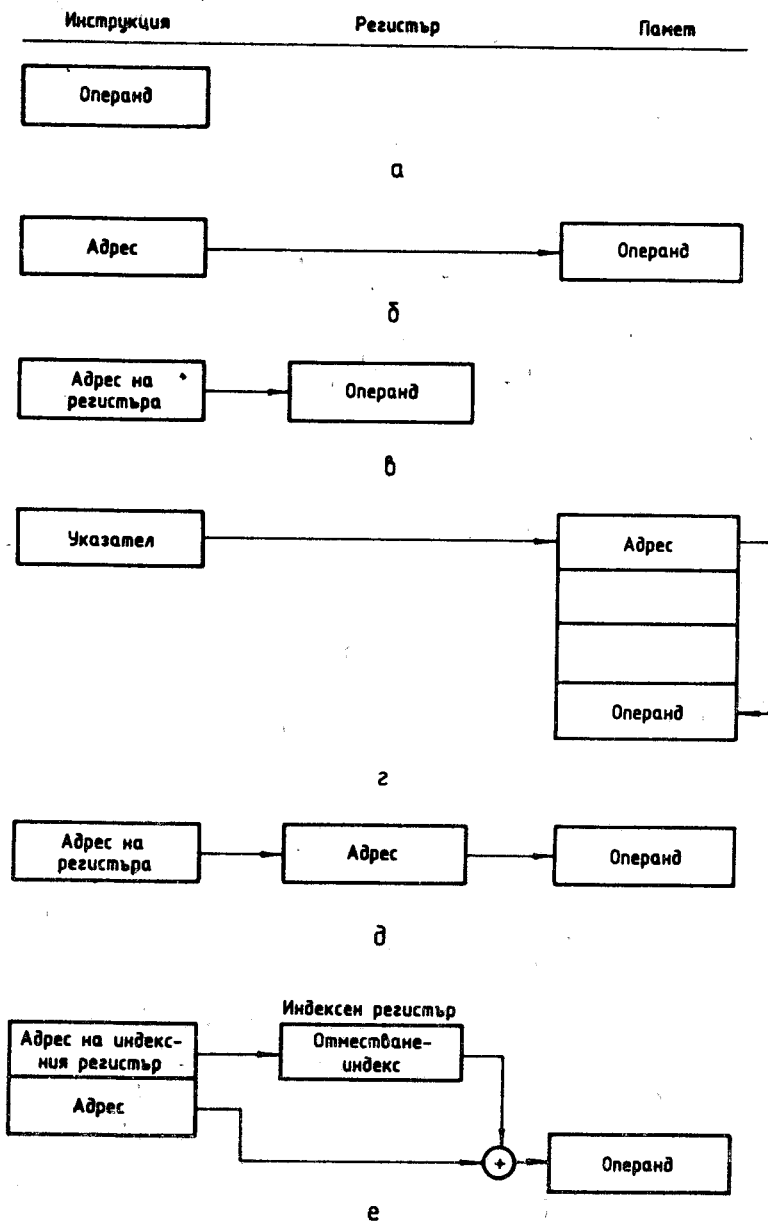
а -- с указване на страницата б с указване на сегмента

Виртуална адресация. Използват се виртуални адреси, като се създава представа, че паметта има неограничен обем. Това става чрез механизма на виртуалната адресация, в основата на който е заложен принципът на динамичното преразпределение между страниците на оперативната памет и външната памет на системата. За всеки потребител операционната система създава таблици на съответствие на виртуалните и физическите страници. Ако има обръщение към физическата страница, която не е в основната памет, тя се извлича от външната памет и се зарежда в основната, а ненужната страница се записва във външната памет (фиг. 1.3). Страничната организация е реализирана в микропроцесора K1810BM86, а виртуалната адресация — в микропроцесорите iAPX286 и MC68010.

Непосредствена адресация. Ако операндът се намира в кода на инструкцията, адресацията се нарича непосредствена. Когато в инструкцията е фиксирано мястото на операндите и резултата, адресацията се нарича **неявна (подразбираща се)**. Подобни са инструкциите SPHL и PCHL в МП 18080. Инструкциите с непосредствена и неявна адресация се изпълняват много бързо, но липсва гъвкавост. Непосредствената адресация се използва при работа с константи, които остават неизменни за дадена програма (фиг. 1.4 а).



Фиг.1.3. Виртуална адресация



Фиг. 1.4. Видове адресация:

- а — непосредствена адресация;
- б — пряка абсолютна адресация;
- в — пряка регистрова адресация;
- г — косвена адресация;
- д — косвена адресация с едно ниво;
- е — индексна адресация

Пряка адресация. Операндите се извличат от паметта по адрес, който се съхранява във формата на инструкцията. Ако адресът в инструкцията не е даден чрез символи, а с числов запис, адресацията се нарича абсолютна (фиг. 1.4 б). Вместо адрес в кода на инструкцията може да се съхранява име на регистър, в който се намира операндът. Тогава адресацията се нарича пряка регистрова (фиг. 1.4 в). Ясно е, че гъвкавостта на пряката адресация също е недостатъчна, тъй като тя не позволява да се изпълняват процедури по модификация на адреса, които са необходими при преместване на програмата в паметта, при работа с масиви и т.н.

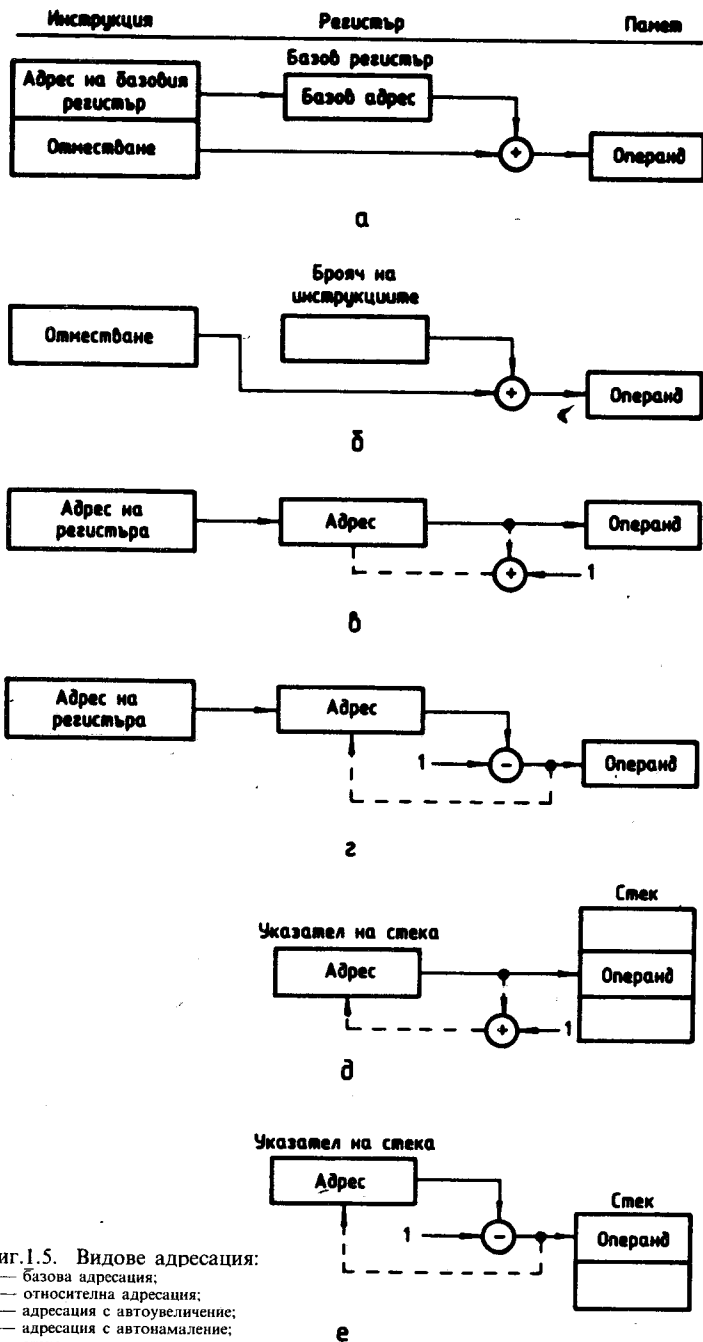
Косвена адресация. Операндите от паметта се извличат по адрес, посочен в клетка от паметта (фиг. 1.4 г). В кода на инструкцията се намира указател на адреса на операнда, който е разположен в паметта. При изпълнение на инструкция с такава адресация два пъти се извършва обръщение към паметта — отначало се извлича адресът, а след това операндът. Така, без да се изменя кодът на инструкцията, може да се измени адресът на операнда, който се съдържа в клетка от паметта, адресирана от полето на указателя в кода на инструкцията. На фиг. 1.4 д е показан механизъм на косвена адресация с едно ниво. Вместо указател на адреса в кода на инструкцията може да се съхранява име на регистър, в който се намира адресът на операнда. В този случай имаме **косвена регистрова адресация**.

Индексна адресация. При работа с масиви се използва индексна адресация, при която адресът на операнда се изчислява чрез сумиране на адреса на първия елемент на масива, който се извлича от инструкцията, и положителното отместване (индекс) — цяло число, съхранявано в специален индексен регистър (фиг. 1.4 е). В инструкциите с индексна адресация адресирането на операнда е свързано със загуба на време, но с такива инструкции може да се адресира произволен елемент от масива без изменение на съдържанието на инструкцията. Този тип адресация е много удобен при организиране на цикли. Програми, в които се намират инструкции с абсолютни адреси, не могат да се преместват в паметта без модификация на адресите.

Базова адресация. За да се осигури преместваемост на програмите, се използва базова адресация, при която адресът на операнда се изчислява, като се сумира съдържанието на базовия регистър и адреса, разположен в адресната част на инструкцията (фиг. 1.5 а). Съдържанието на базовия регистър може да е положително или отрицателно цяло число.

Относителна адресация. Преместваемостта на програмите, които съдържат инструкции за преход, извикване и връщане към стандартни подпрограми, се осигурява чрез относителна адресация. При нея адресът на операнда се извлича чрез събиране на съдържанието на брояча на инструкциите с отместването, разположено в адресната част на инструкцията (фиг. 1.5 б). Отместването може да е положително или отрицателно цяло число. Тази адресация е подобна на базовата, като тук в качеството на базов регистър се използва броячът на инструкции.

Адресация с автоувеличение и автономаление. При адресирането с автоувеличение и автономаление преди или след изпълнение на инструкцията съдържанието на регистъра, в който се намира адресът на операнда, се увеличава или намалява с единица. Този тип адресация се използва при организация на цикли (фиг. 1.5 в и 1.5 г).



Фиг.1.5. Видове адресация:
 а — базова адресация;
 б — относителна адресация;
 в — адресация с автоувеличение;
 г — адресация с автонамаление;
 д — стекова адресация с увеличение;
 е — стекова адресация с намаление

Стекова адресация. Широко разпространение е намерила стековата адресация. При нея в специален регистър, наречен указател на стека, се задава адресът на върха на стека. Самият стек се използва за съхраняване на операндите или клетките от оперативната памет, или общите регистри. Със специална инструкция данните се изпращат в стека, като указателят на стека автоматично се увеличава с единица при въвеждане на нов операнд и се намалява с единица при извличане на операнд (фиг. 1.5 *d* и 1.5 *e*).

1.5. ОСНОВНИ МОДУЛИ НА ПЕРСОНАЛНИТЕ КОМПЮТРИ

Персоналните компютри се изграждат на модулен принцип, като се спазва определена йерархия. Това позволява чрез комбинация от различни модули гъвкаво да се създават компютри с различна мощност и предназначение. Самите модули най-общо могат да се разделят на две групи — *апаратни и програмни*. При изграждането на конкретна конфигурация модулите се подреждат на четири йерархични нива в зависимост от тяхната функция и предназначение. Тази четиристепенна структура (фиг. 1.6) заедно с интерфейсите връзки между отделните слоеве представлява архитектурен модел на персоналните компютри. На най-ниското ниво са разположени апаратните модули на системата, управлявани от вътрешното програмно осигуряване (ВПО), което е вторият слой. То обикновено се реализира чрез постоянна памет (памет от тип ROM), включена в състава на някой от апаратните модули. Най-често това е процесорът или системната платка. Поради естеството на тяхната реализация тези два слоя са недостъпни за крайния потребител.

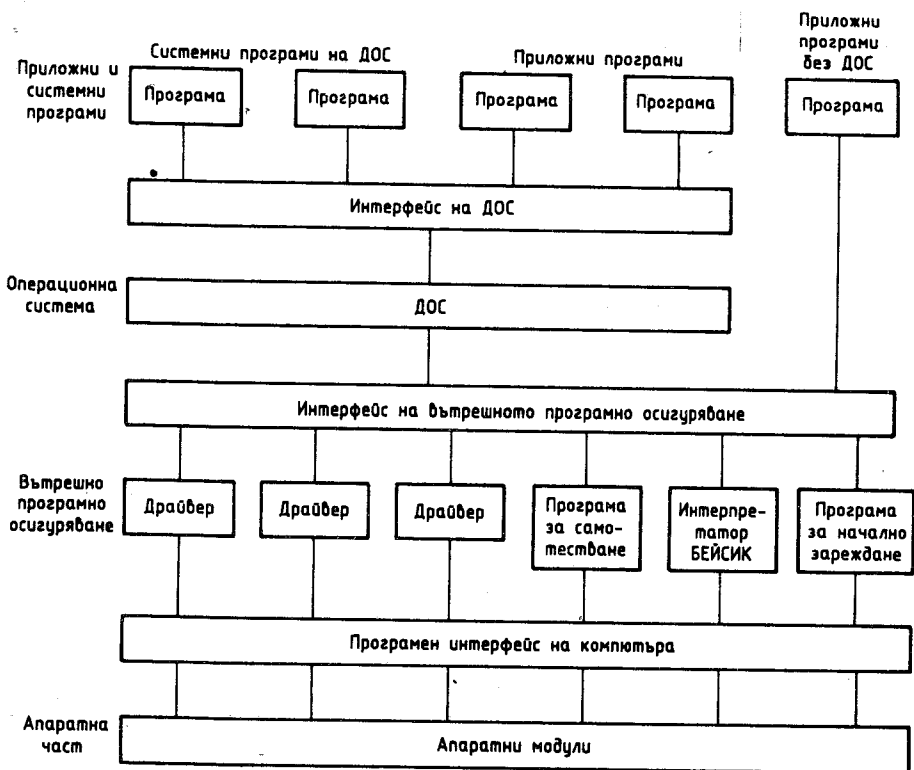
Програмите на вътрешното програмно осигуряване работят директно с апаратните модули на компютъра, поради което те са функционално свързани с тях и при замяна на който и да е апаратен модул трябва да се замени и програмата от ВПО за работа с него. Тези програми се наричат драйверни програми (или накратко драйвери). Вътрешното програмно осигуряване предоставя програмен интерфейс за работа с компютъра на всички останали програми. Достъпът до програмите на ВПО става чрез системата за програмни прекъсвания.

Горните две нива на обобщения архитектурен модел включват операционната система и приложното и системно програмно осигуряване. Когато модулите, от които те се състоят, са разположени в оперативната памет, те са активни и тогава са достъпни за потребителя. Той може да избира набора от активни модули, да ги подменя с по-нови версии или да ги модифицира в зависимост от изискванията на някои специални приложения.

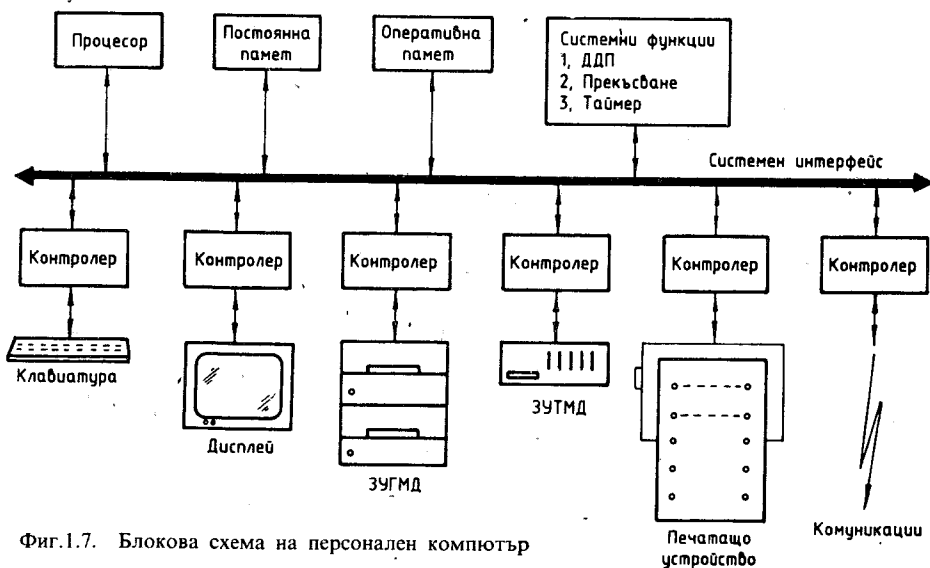
Операционната система управлява ресурсите на компютърната система чрез интерфейса за ВПО. Към тези ресурси се отнасят оперативната и външната памет, входно-изходните устройства и програмите на потребителя.

Интерфейсът на ВПО дава възможност ПК да работят с една и съща операционна система въпреки различията в апаратната им реализация.

Най-горното йерархично ниво се заема от приложните и системните програми. Последните служат за създаване и съпровождане на прилож-



Фиг.1.6. Архитектура на персоналния компютър с интерфейсни връзки между отделните нива

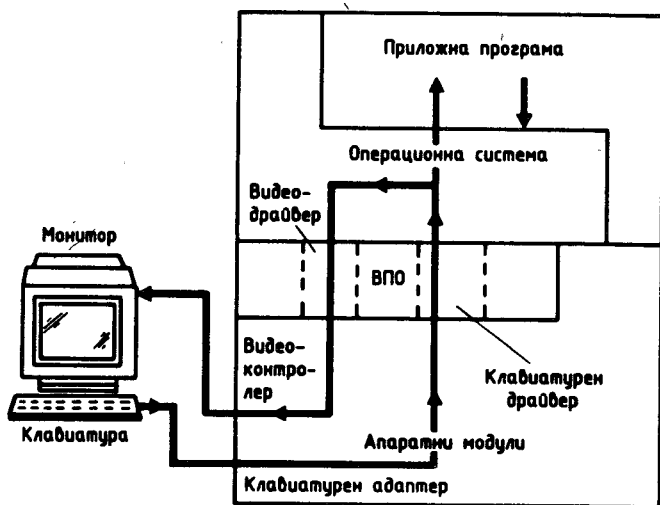


Фиг.1.7. Блокова схема на персонален компютър

ните програми. Системни програми са текстовите и свързващите редактори, програмите за настройка, асемблерите, транслаторите и интерпретаторите за различни програмни езици и др. Приложните програми се създават за решаване на определен проблем. Съществуват някои характерни класове приложни програми: за текстообработка, за финансово-счетоводна дейност, за обслужване на бази от данни, за автоматизация на инженерния труд, за решаване на математически задачи и др. Системните програми заедно с операционната система образуват т.нар. *системно програмно осигуряване*.

Обща блокова схема на персонален компютър е дадена на фиг. 1.7, където са показани основните апаратни модули: процесор, постоянна и оперативната памет, модул за директен достъп до паметта, система за прекъсване, системен таймер, контролери, периферни устройства и т.н. Чрез системния интерфейс апаратните модули имат достъп до общото адресно пространство на паметта и до входно-изходните регистри на системата.

Най-обща представа за функционирането на един персонален компютър може да се получи, ако се проследи последователността от събития в режим на въвеждане на текст (фиг. 1.8). При натискането на клавиш от клавиатурата се изпраща съответен код към клавиатурния адаптер. Същевременно системата за прекъсване активира клавиатурния драйвер от ВПО, който приема кода, преобразува го във вътрешния код на компютъра и изпраща потвърждение към клавиатурата. След това ВПО подава сигнал към операционната система, която приема прекодирания символ и го изпраща както към приложната програма за текс-



Фиг.1.8. Работа на персонален компютър при въвеждане на текст

тообработка, така и към видеомонитора. Символът се появява на екрана след активиране на видеодрайвера, който изпраща символа за визуализация чрез видеоконтролера. Същевременно приложната програма инструктира операционната система да следи за получаването на следващия символ от клавиатурата, с което цикълът се затвря.

1.6. БАЗОВО ПРОГРАМНО ОСИГУРЯВАНЕ

Базовото програмно осигуряване на персоналните компютри включва вътрешно програмно осигуряване, операционна система, компилатори и интерпретатори на алгоритмични езици, средства за настройка на програмите, програмни продукти за свързване на компютъра към големи или малки изчислителни машини и др.

Програмите на вътрешното програмно осигуряване (firmware) се създават при проектирането на компютъра и се съхраняват в постоянната му памет. Основните функции на ВПО са:

— управление на широк набор от периферни устройства, като се осигурява програмен интерфейс към тях за различни операционни системи;

— бърза проверка на компютъра при включване на захранването му или при начално установяване (самотестване);

— инициализиране на апаратните модули на компютъра;

— зареждане на операционната система (ако има нужда от нея).

Както се вижда от фиг. 1.6, основните елементи на ВПО са входно-изходните драйвери, програмата за самотестване, програмата за начално зареждане и интерпретаторът на БЕЙСИК. При взаимодействието си с функционалните модули на компютъра ВПО използва картата на паметта на компютъра, картата на входно-изходните регистри, разпределението на каналите за ДДП, както и разпределението на програмните и апаратните прекъсвания.

Входно-изходните драйверни програми (драйвери) работят директно със съответните контролери, като по този начин позволяват на програмиста да не се интересува от физическата организация на периферното устройство, а единствено от инструкциите, които реализира обслужващият го драйвер.

Драйверите имат:

— отворена структура — добавянето на нов драйвер не влияе на работата на останалите;

— гъвкава структура — организацията на достъпа до драйверите чрез програмни прекъсвания позволява те да не изискват фиксирани области в паметта, което дава възможност бързо и лесно да бъдат заменени;

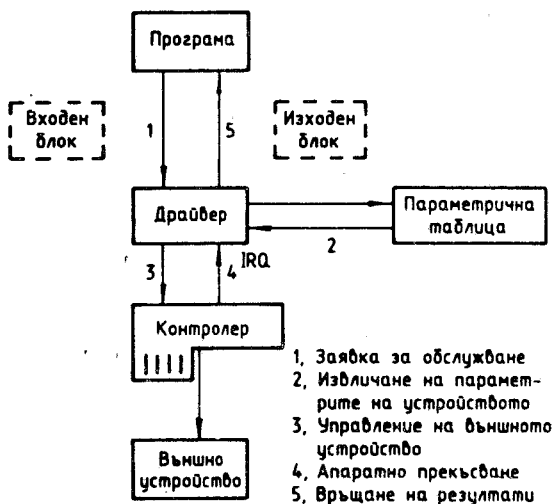
— настройваема структура — драйверните програми са ориентирани към клас периферни устройства, чиито характерни параметри се намират в специални таблици (достъпни чрез вектори на програмни прекъсвания); драйверите се настройват към конкретните периферни устройства чрез промяна на стойностите в тези параметрични таблици;

— асинхронна структура — режимът на работа на драйверите е асинхронен, като във всеки момент те са достъпни за другите програми.

Драйверите се активират от програмно прекъсване, а самите те изпращат необходимите команди към периферните устройства, които отговарят асинхронно чрез апаратни прекъсвания. Всеки драйвер използва едно програмно прекъсване за комуникация с другите програми и едно апаратно прекъсване за работа с периферните устройства.

Механизмът на работа на драйвера е илюстриран на фиг. 1.9. Чрез програмно прекъсване съответната програма подава заявка за обслужване към драйвера и му предава специален входен блок с информация за командата, която той трябва да изпълни. Въз основа на тази информация се определя и взаимодействието на драйвера с периферното ус-

ройство. Параметрите на това устройство се извличат от параметричната таблица на драйвера. След това се изпълнява командата за управление на периферното устройство, което информира драйвера за изпълнението ѝ чрез апаратно прекъсване. След завършване на операцията драйверът формира специален изходен блок за връщане към програмата, от която е заявката за обслужване. Той се състои от данни, получени от периферните устройства, и информация за състоянието при завършване на операцията. Най-често за съхраняване на входния и изходния блок се използват регистрите на процесора, което увеличава бързината на реакцията в драйвера. Когато тези блокове са с голяма дължина, процесорните регистри съдържат само указатели към тях, а самите блокове се предават чрез системата за директен достъп до паметта на компютъра.



Фиг.1.9. Действие на драйвер на периферно устройство

върът формира специален изходен блок за връщане към програмата, от която е заявката за обслужване. Той се състои от данни, получени от периферните устройства, и информация за състоянието при завършване на операцията. Най-често за съхраняване на входния и изходния блок се използват регистрите на процесора, което увеличава бързината на реакцията в драйвера. Когато тези блокове са с голяма дължина, процесорните регистри съдържат само указатели към тях, а самите блокове се предават чрез системата за директен достъп до паметта на компютъра.

Операционната система разпределя ресурсите на компютърната система и облекчава работата на програмиста при съставяне на приложни програми, като при това му позволява да се абстрахира от физическата организация на периферните устройства. Когато е записана на магнитен диск, тя се нарича *дисква операционна система (ДОС)*.

Основен компонент на ДОС е файловата система, която дава възможност програмите и данните за даден проблем да се представят като логически свързани последователности, наречени *файлове*. Всеки файл има свое име. Файловата система разполага формираните файлове върху свободните сектори на магнитните носители в запомнящите устройства.

Достъпът до файловете става чрез задаване на техните имена и името на устройството, върху което те са разположени. Тъй като физическата организация на устройството остава скрита за програмиста, обръщението към устройствата става с логически имена, които им се присвояват от операционната система. За програмиста периферните устройства се идентифицират като логически устройства.

ДОС се състои от две части. Едната от тях се въвежда от програмата за начално зареждане в оперативната памет на персоналния компютър и през цялото време на работата му се съхранява там. Тя се нарича *резидентна част* и осигурява достъп на приложните програми до логическите устройства, грижи се за разпределянето на паметта, обслужва различните типове прекъсвания и т.н. Другата част на ДОС е *нерезидент-*

на — тя включва *външни програми*, които изпълняват някои по-сложни функции, като форматиране и копиране на дискети, сравняване на файлове и дискети, възстановяване на повредени носители и др. Тези програми се съхраняват върху външни носители и при необходимост се въвеждат в оперативната памет със средствата на резидентната част.

Трябва да се има предвид, че фирмите производители на програмни продукти постоянно развиват и усъвършенстват операционните системи за микрокомпютри, като разпространяват при изгодни за потребителя условия новите версии. При стартиране на всяка операционна система върху екрана се извежда номерът на нейната версия. Особеностите за работа със съответната версия трябва да се търсят във фирмената литература.

Програмните продукти, предназначени за съставяне и проверка на програми — асемблери и транслатори на алгоритмични езици, също се причисляват и към базовото програмно осигуряване. Асемблерът е език, който е най-близък до машинния. Основното му предназначение е за изграждане на системни програми. Алгоритмичните езици (БЕЙСИК, ПАСКАЛ, ФОРТРАН и др.) са насочени към съставяне на приложни програмни продукти, като използват интерпретатори и компилатори. Интерпретаторите изпълняват веднага след изчитането му всеки оператор от програмата, написана на алгоритмичния език. Компилаторите превеждат цялата програма от алгоритмичен език в изпълним код, след което преведената програма може да се зареди в паметта на компютъра и да се изпълни. Интерпретаторите позволяват бърза настройка на програмите, но изпълнението на програмите е сравнително бавно, докато получената от компилаторите изпълнима програма е много по-бързодействаща.

Програмните продукти за връзка на персоналния компютър с други ЕИМ се включват в базовото програмно осигуряване. Използването на персонални компютри в големи предприятия и учреждения с вече изградени електронноизчислителни центрове води до необходимостта от свързването им към голяма ЕИМ. За създаването на тази връзка компютърът преминава в *режим на терминална емуляция* (имитиране на терминал). Този режим се получава при взаимодействието на два специфични компонента — *комуникационен адаптер* и *програмен емулатор*. За целта в персоналния компютър се вгражда платка — комуникационен адаптер, а самият режим на терминална емуляция се постига при стартирането на емуляционна програма в компютъра. Тогава персоналният компютър, който е свързан чрез модеми и съобщителни линии към голямата ЕИМ, имитира определен тип терминал, познат за комуникационните средства на голямата ЕИМ. В резултат потребителят работи така, както би работил с терминала, който компютърът имитира.

От гледна точка на потребителя връзката между персоналния компютър и голямата ЕИМ трябва да осигурява:

- изпращане на данни към голямата ЕИМ;
- предаване на данни от голямата ЕИМ към персоналния компютър;
- преобразуване на данните във формат, подходящ за персоналния компютър;
- достъп до данните на голямата ЕИМ от страна на приложните програми, изпълнявани на персоналния компютър.

Зависимост от реализираните потребителски функции се различа-

ват две нива на терминална емуляция. В първия случай от гледна точка на голямата ЕИМ микрокомпютърът по нищо не се отличава от емуляцията от него терминал, което дава възможност да се заменят скъпи терминали (например от типа IBM 3270) със значително по-евтини персонални компютри.

Второто, по-съвършено ниво на комуникации дава възможност за прехвърляне на информационни масиви от големи ЕИМ към персоналния компютър и обратно. Възможността за двустранен обмен на данни между персоналния компютър и централната база от данни на голямата ЕИМ създава предпоставки за по-пълно използване на персоналните компютри в големи предприятия и учреждения. Там, където те са свързани в локална мрежа, комуникациите с голямата ЕИМ се осъществяват от всички компютри, като се използва комуникационна линия.

Тъй като микрокомпютрите не са специално проектирани за ползване в мрежа с голяма ЕИМ, интегрирането им във вече съществуващите системи е сложна задача. Макар че на пазара се предлагат множество програмни продукти, осигуряващи възможност за комуникации между персонални компютри и големи ЕИМ, повечето от тях решават само частни проблеми. Същинското интегриране на персоналните компютри към съществуващата голяма ЕИМ изисква сериозно проучване за избор на подходящи програмни и технически средства и продължителен процес на внедряване за всеки конкретен случай.

1.7. ПРИЛОЖНИ ПРОГРАМИ

От гледна точка на необходимите технически средства различните области на приложение на микрокомпютрите се разделят на няколко основни групи. Първата група обединява домашните приложения, за които е достатъчна минимална конфигурация с използването на битова електронна апаратура — системно устройство с клавиатура, битов телевизор и битов касетофон. Такива микрокомпютърни системи се използват най-често за забавни игри, образователни програми, елементарна текстообработка и др.

Втората група включва области на приложение, които изискват стандартна конфигурация — системно устройство, видеомонитор, клавиатура и печатащо устройство. Това са най-често дейности от административно-управленската сфера — текстообработка, планиране, финанси и счетоводство, архивиране и др.

Включването на графични периферни устройства към основната конфигурация определя третата група приложения — за автоматизация на инженерното проектиране. Най-често използваните за тази цел графични периферни устройства са плотерът, дигитайзерът и мишката. При това към видеомонитора и видеоконтролера се предявяват повишени изисквания за разделителната им способност в графичен режим.

На основата на микрокомпютърните системи се изграждат т.нар. *автоматизирани работни места*, които представляват комплекс от технически и програмни средства, предназначени за автоматизация на рутинните дейности в определени области на стопанството. Те включват съответните приложни програмни продукти за конкретната област на приложение. Така се обособяват автоматизирани работни места от типа „секретар“, „плановик“, „ръководител“, „личен състав“, „конструктор“, „технолог“, „редактор“ и др.

В сферата на административното обслужване могат да се обособят четири основни функционални групи от програмни пакети:

- за текстообработка;
- за създаване и работа с таблици;
- за управление на бази от данни;
- за обработка на икономическа графична информация.

Те се наричат още *специализирани пакети*, тъй като са предназначени за изпълнение само на една от посочените по-горе функции. Например програмният продукт ТЕКСТ-16 оговаря на всички съвременни изисквания за текстообработка. Този пакет, съставен от няколко програмни модула, удовлетворява с възможностите си и най-претенциозния редактор. Все повече са обаче потребителите на персонални компютри, които се нуждаят от пакет, съчетаващ в себе си възможности за текстообработка, за въвеждане на таблици и графично изобразяване на данните в тях, за управление на бази от данни и т.н. Удобството тези често използвани функции да се намират в един пакет гласна програмната промишленост към разработката на т.нар. *интегрирани пакети*, които включват голяма част от възможностите на посочените по-горе функционални групи. Трябва обаче да се има предвид, че в случаите, когато се извършват строго определени дейности (например само текстообработка), се предпочитат пакети, специално предназначени за съответната дейност.

За масовите потребители остава само задачата правилно да подбере най-подходящия пакет за разрешаването на проблемите си. Днес правилният избор на пакет осигурява бързината и ефективността при решаването на дадена задача.

Повечето от интегрираните и специализираните пакети за персонални компютри осигуряват достъп до функциите си чрез т.нар. *меню* (списък на възможности и параметри за избор). Това е един от най-разпространените начини за улесняване на диалога между човека и компютъра. Менюто е изградено йерархично — на две, по-рядко на три и повече нива, които позволяват достъп до нови, по-конкретни подфункции и параметри. Във всеки момент от работата си потребителят може да се обърне към която и да е от функциите на пакета. Това става под формата на въпроси и отговори на естествен език — съответно извеждани на екрана на видеомонитора и въведени чрез натискане на един или два клавиша от клавиатурата. По този начин потребители, които не са специалисти по програмиране, използват без затруднения компютрите в работата си.

Основният обем от работата в една проектантска организация включва дейности, като проектиране, чертане, експериментиране, внасяне на корекции и др. При това около половината от времето за разработка на дадено изделие се използва за подготовка на документацията — съставянето на скици, чертежи, тяхното редактиране и коригиране. Въобще процесът на проектиране е свързан с обработката на голямо количество графична информация. Не случайно сред специалистите е разпространен изразът, че чертането е „езикът на инженера“. Създаването и преработката на графичната информация, която е свързана с даден инженерен проект, се разпределя на два основни етапа — системно и логическо проектиране, от една страна, и конструктивно проектиране и чертане, от друга. И двата етапа включват редица рутинни дейности, които могат да се автоматизират, като естествено обемът на тези рутинни дейности

във втория етап е значително по-голям. Възможността за използването на *системи за автоматизация на проектирането* зависи до голяма степен и от съответната област на инженерното проектиране — машиностроене, изчислителна техника, строителство, електронни елементи, вътрешно обзавеждане и др.

Ползата от системите за автоматизация на проектирането е доказана отдавна, в резултат на което съществуват редица реализации и търговски продукти. Основните предимства на автоматизираното проектиране са свързани със същественото съкращаване на времето за разработка, но особено значение има и рязкото повишаване на качеството и културата на проектантския труд. Известни са редица системи за автоматизация на проектирането, изградени на основата на голяма ЕИМ или микроминикомпютър, към който се свързват специализирани работни станции. В сравнение с тях системите за проектиране на базата на професионални компютри имат значително по-ниска цена, което доведе до масовото им приложение в по-малки проектантски бюра и организации.

Пакетите приложни програми, които днес се предлагат за проектантите, са много ефективни, тъй като осигуряват възможности за поставяне, преместване и преразпределение на компоненти и връзки според изискванията на конкретен проект. Те предоставят по-голяма свобода на проектанта и съкращават много времето за достигане на оптималния вариант. Тяхното най-важно предимство пред ръчния метод на проектиране обаче е възможността за тестване на избраното решение още в началните етапи от цикъла на проектирането, като се използват методи за моделиране, при което непрекъснато се проверяват получаваните резултати.

Изследванията показват, че обемът на информацията, която подлежи на предаване или разпространение, е най-голям в рамките на едно предприятие. Именно това доведе през последните години до развитието на *локалните мрежи от ЕИМ*, в които микрокомпютрите играят ролята на работни станции. Локалните мрежи осигуряват една сравнително висока скорост на обмен при ниска цена, което позволява да се осигури предаването на голямо количество информация между десетки и стотици потребители.

Локалните мрежи от персонални компютри осигуряват възможност за електронизация на учрежденията. Една от основните функции, които те реализират, е *електронната поща*. С нейна помощ може да се автоматизира цялата вътрешна кореспонденция на едно предприятие. Освен това потребителите могат да осъществяват разпределена обработка на информацията, като използват общи ресурси от външни запомнящи устройства с много голям обем, висококачествени печатащи устройства и др. Това е голямо предимство, тъй като високата цена на тези технически средства не позволява инсталирането им на всяко работно място. Същевременно се осигурява възможност работещите в локалната мрежа да използват общи файлове от данни и общи програми. Така се съкращава значително разходът на външна памет, като в същото време се дава възможност на потребителите да разработват общи задачи, без да си обменят непрекъснато последните версии на файловете или програмите.

1.8. ИЗБОР НА ПЕРСОНАЛЕН КОМПЮТЪР

Първата и може би най-важна стъпка при избора на персонален компютър е да се определи какво точно е неговото *предназначение* — за електронни игри, за текстообработка, за обучение, за електронен архив, за автоматизация на проектирането, за електронна поща, за управление на процеси и т.н.

Следващата стъпка е определянето на подходящо *приложно програмно осигуряване*. Съществуват два коренно различни подхода при решаването на тази задача. Първият от тях е осигуряването на *готови пакети* с приложни програми. Обикновено те се купуват отделно като самостоятелни продукти и са предназначени за работа в определена област на приложение. Съществуват доста организации, които предлагат приложни пакети с широк спектър на функционалните им възможности и цени. Не трябва да се забравя и качеството на документацията, от което до голяма степен зависи дали потребителят ще усвои бързо и ще използва ефективно даден програмен продукт.

Вторият подход е да се осигурят *приложни програми, написани специално за конкретното приложение*. Той се прилага в случаите, когато готовите приложни пакети са неподходящи за решаване на поставената задача. Този вариант е далеч по-неефективен в сравнение с първия, тъй като за създаването на приложната програма са необходими много повече ресурси — кадрови и финансови. Освен това от потребителя се изисква много точно дефиниране на задачата, защото в противен случай усилията за нейното решаване могат да се окажат напразни. При качествено съставяне на новия програмен продукт е възможно да се получи съществено увеличение на ефективността на ПК.

Днес съществуват фамилии от съвместими помежду си персонални компютри. *Съвместимостта* осигурява разпространението на програмното осигуряване върху различни модели и марки компютри в рамките на дадена фамилия. При избора на ПК е необходимо да се обърне специално внимание на този въпрос.

В персоналните компютри различаваме няколко вида съвместимост. Най-важна е *съвместимостта на ниво машинен език*. Тя е свързана с използвания микропроцесор и възприетото архитектурно решение на системата и до голяма степен определя принадлежността на ПК към дадена фамилия. Не по-малко важна е *съвместимостта на информационните носители* — най-често дискети. От нея зависи възможността за прехвърляне на програми и данни между различни ПК. *Съвместимостта на ниво кодове* за представяне на информацията позволява безпрепятствено разпространение на файлове от данни и програмни продукти. *Съвместимостта на клавиатурите* е свързана не само с подреждането на клавишите върху клавиатурата, но и с кодовете, които се генерират при натискането на всеки клавиш. Трябва да се отбележи и *конструктивната съвместимост*, която осигурява възможност за функционирането на апаратни модули в различни ПК. Редът, в който са дадени видовете съвместимост, определя и тяхната важност при определяне на глобалната съвместимост на микрокомпютърните системи. В практиката често се говори за частична съвместимост. Обикновено това става, когато съществува разлика в кодовете за представяне на информацията или в

подредането на клавишите на клавиатурите. В повечето от тези случаи се изискват допълнителни усилия от страна на потребителя за работа с частично съвместимите персонални компютри.

Друга основна характеристика на ПК, която винаги трябва да се има предвид при техния избор, е *производителността*. Тя определя броя на решаваните задачи за даден интервал от време. С нея е свързано бързодействието на ПК, което е важно в случаите, когато се решават задачи в реално време. Особено внимание трябва да се обърне на оперативната памет и на видовете външни запомнящи устройства. Важно е да има *възможност за модулно разширение* както на ОП, така и на включваните периферни устройства. Естествено при избор на ПК потребителят трябва да отчита и редица други показатели: цена, удобство за работа, осигуряване на сервиз, обучение от фирмата доставчик, възможност за докомплектуване и т.н.

ЛИТЕРАТУРА КЪМ ГЛ. 1

1. Боровски, Б. и др. Справочник по изчислителна техника. С., Техника, 1985.
2. Боянов, К., В. Кисимов. ЕИМ и мрежи от ЕИМ. С., Техника, 1985.
3. Самофалов, К. Г., О. В. Викторов, А. К. Кузник. Микропроцесоры. Киев, Техника, 1986.

ОСЕМРАЗРЕДНИ ПЕРСОНАЛНИ КОМПЮТРИ

Може да се твърди, че перспективите на персоналните компютри се очертаха и доказаха с осемразредните персонални компютри.

Осембитовите микропроцесори I8080 и 6502 на фирмите Intel и MOS Technology станаха базата, която осигури развитието на 8-разредните персонални компютри в периода 1978—1980 год. Архитектурата на тези микропроцесори, наборът им от инструкции и историческият приоритет на фирмите производители на персонални компютри определиха оформянето на две основни групи 8-разредни персонални компютри на световния пазар.

Едната група е свързана със създаването на първия наложил се масово ПК на фирмата Apple (САЩ). Компютърът е изграден на базата на микропроцесора 6502 и включва типични архитектурни решения. За него е създадена дисковата операционна система ДОС 3.3, която е предназначена за работа с микропроцесора 6502 и се използва при персоналните компютри, съвместими с APPLE II. Между тях са българските Правец-82, Правец-8М и Правец-8А, които са изградени на базата на микропроцесора СМ630 (аналог на 6502).

Втората група осемразредни персонални компютри отразява един стандартизиран подход (фирмено независим) при създаването на голям брой различни типове персонални компютри. Свързващото звено между тези компютри е дисковата операционна система СР/М, създадена от фирмата Digital Research (САЩ). Тази операционна система е създадена за микропроцесора I8080. В персоналните компютри тя се е наложила с версията СР/М 2.2. Като основна операционна система тя може да се изпълнява на три микропроцесора — I8080, I8085 и Z80.

В началото на 80-те години в света съществуваша вече над 400 микропроцесорни изделия, реализирани на базата на един от тези микропроцесори, върху които се използваше операционната система СР/М и за които беше създадено приложно програмно осигуряване. Така СР/М навлезе в персоналните компютри със значително по обем програмно осигуряване: езици от високо ниво (БЕЙСИК, ФОРТРАН, ПАСКАЛ), мощни програми, реасемблери и специализирани приложни програми. По времето на появата на първите 16-разредни ПК на фирмата IBM в света се разпространяваха над 30 хиляди приложни програми, предназначени за изпълнение върху СР/М съвместими персонални компютри. Усвояването в редовно производство в социалистическите страни на функционални аналози на микропроцесорите I8080 (K580IK80) и Z80 (U880) доведе до развитието и навлизането в експлоатация на този тип персонални компютри във всички социалистически страни. Основни представители на тези ПК у нас са ИЗОТ1031 и Правец-8М.

При реализацията на първата група осемразредни персонални компютри водещ фактор е пълната съвместимост със схемните решения на ПК APPLE II. Дисковата операционна система ДОС 3.3 трябва да се разглежда като надстройка на апаратната реализация.

За втората група ПК водещ фактор е ДОС СР/М 2.2. Самата организация на тази операционна система позволява реализирането на голямо разнообразие от персонални компютри, които използват различни 8-разредни микропроцесори.

2.1. ОСЕМРАЗРЕДНИ ПЕРСОНАЛНИ КОМПЮТРИ ОТ ФАМИЛИЯТА ПРАВЕЦ

У нас се произвеждат няколко модела 8-разредни персонални компютъра от фамилията Правец: Правец-82, Правец-8М и Правец-8А. Те са разработени на базата на общи архитектурни принципи, като по-новите модели включват по-съвременни технически и конструктивни решения. Основната конфигурация на персоналния компютър се състои от компютърен блок и видеомонитор. Компютърният блок включва клавиатура, захранване и системна платка с 8 куплунга (слота) за включване на контролерите на периферни устройства.

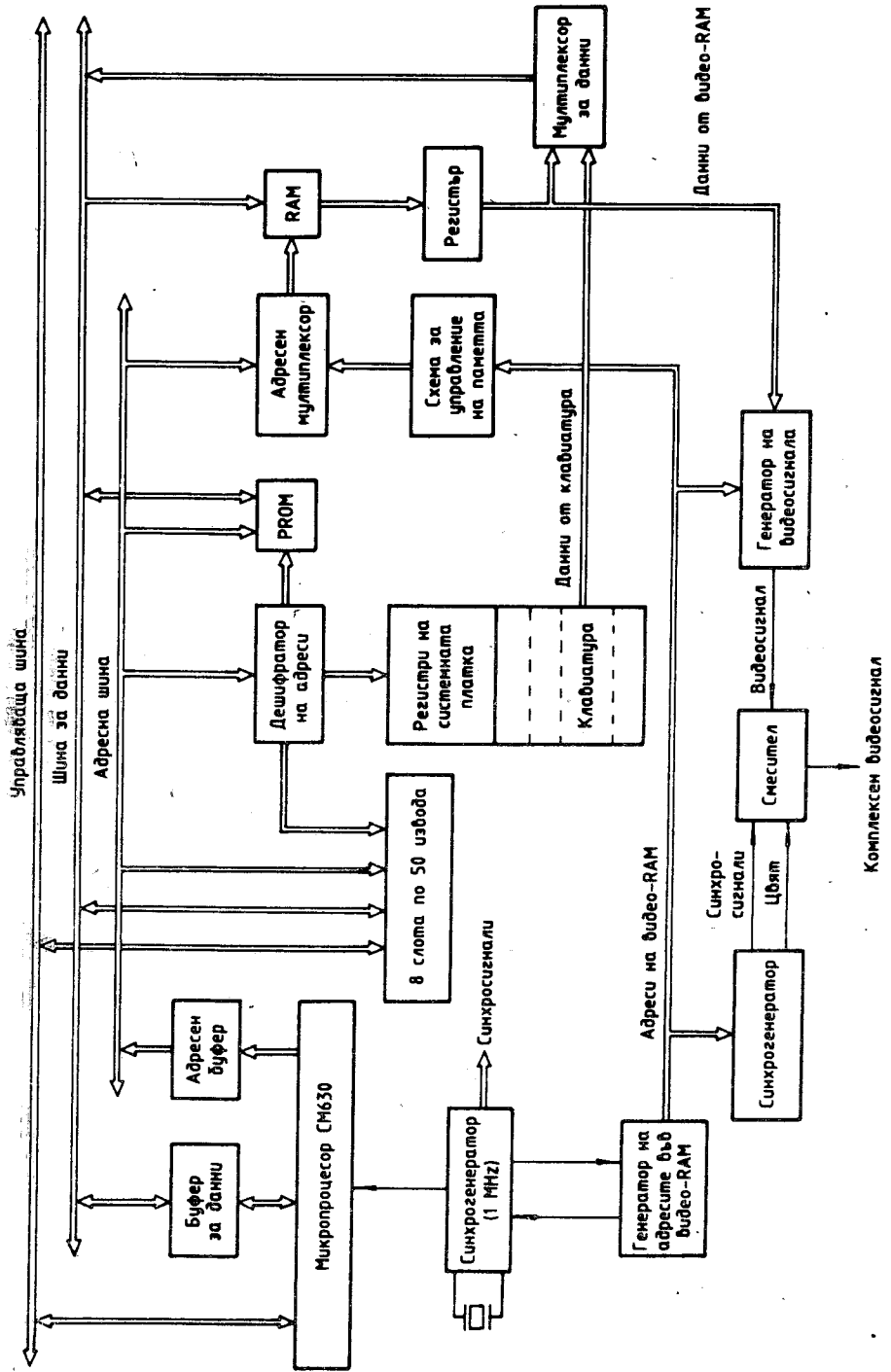
Обобщената блокова схема на 8-разредния персонален компютър, съвместим с APPLE II, е показана на фиг.2.1. Основен блок е микропроцесорът СМ630. Той е 8-разреден с възможност за директна адресация на 64 Кбайта памет. Работи с тактова честота 1 МHz, която се създава от кварцово стабилизиран генератор на синхроимпулси. Генераторът осигурява и синхросигналите за останалите блокове на персоналния компютър.

Адресната шина е 16-разредна, а двупосочната шина за данни е 8-разредна. Организацията на персоналния компютър осигурява директен достъп до паметта и маскируемо прекъсване с приоритет, зависещ от местоположението на периферната платка. Системната шина се извежда на 8 куплунга с 50 извода.

Персоналният компютър има 12 Кбайта постоянна памет (ROM), разположени в старшата част на адресното пространство. Останалите адреси се разпределят между оперативната памет и буферите на периферните устройства. Карта на паметта на персоналния компютър е дадена на фиг. 2.2.

Достъп до оперативната памет могат да имат микропроцесорът, генераторът на адресите на видеопаметта и периферните контролери. Те работят в режим на времеделене, като паметта се адресира чрез адресен мултиплексор. Преди да постъпят на шината за данни на микропроцесора, данните от оперативната памет се мултиплексират с данните от клавиатурата.

На системната платка са разположени управляващите схеми за високоговорителя, игровия адаптер, касетофона и клавиатурата. Генераторът на адресите на видеопаметта управлява схемите за формиране на комплексния видеосигнал, осигурява получаването на данните за видеосигнала и преснява оперативната памет.



Фиг.2.1. Обобщена блокова схема на 8-разреден персонален компютър, съвместим с APPLE II

2.1.1. ПЕРСОНАЛЕН КОМПЮТЪР ПРАВЕЦ-82

Структурата на персоналният компютър Правец-82 съответства изцяло на показаната на фиг.2.1. Микропроцесорът CM630 е интегрална схема с 40-изводен корпус. Разположението на изводите и тяхното предназначение е дадено на фиг.2.3.

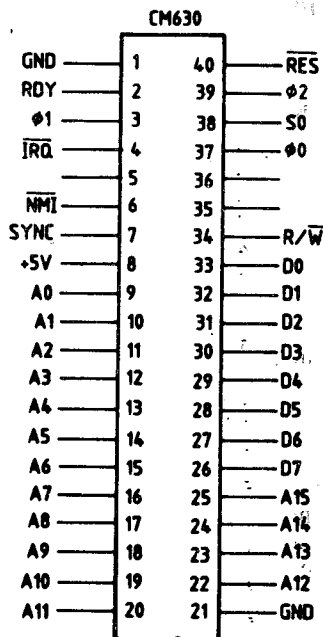
Изводите D0—D7 са двупосочни и се свързват към шината за данни на компютъра. Изводите A0—A15 са изходни и по тях се предават адресните сигнали при работата на микропроцесора. Прекъсванията се възприемат по входните линии IRQ и NMI. Входните вериги на линията IRQ се задействат по ниво (логическа 0), а тези на линията NMI — по падащия фронт на сигнала.

Чрез входната линия RES микропроцесорът се установява в начално състояние. Входният сигнал на извода RDY може да се използва за удължаване на цикъла за четене от периферните контролери. При изпълнение на цикъл за запис този сигнал не се използва. Сигналът на извода S0 установява флага за препълване и не се използва в персоналният компютър.

На базата на входния синхросигнал Ф0 се изработват двата изходни сигнала Ф1 и Ф2, които са необходими за синхронната работа на схемите

0000	Област на оперативната памет: - работна област на операционната среда - стек - клавиатурен буфер - междинни вектори на монитора - текстова видеопамет - работна област на БЕЙСИК - графична видеопамет - потребителска работна област - резидентна част на ДОС
BFFF	
C000 C07F	Област на вътрешните регистри на системната платка
C080 C0FF	Област на вътрешните регистри на периферните контролери
C100 C7FF	Област на постоянната памет на периферните контролери
CB00 CFFF	Не се използва
D000	Област на постоянната памет: - интерпретатор на БЕЙСИК - монитор
FFFF	

Фиг.2.2. Карта на паметта на 8-разреден персонален компютър от фамилията Правец



Фиг.2.3. Разположение на изводите и сигналите на микропроцесора CM630

за управление на паметта на персоналния компютър. Изходният сигнал R/W определя посоката на обмена при извършване на операции с паметта. Сигналят на изходната линия SYNC показва циклите, в които микропроцесорът извлича код на инструкцията и не се използва в Правец-82.

Микропроцесорът SM630 има пет 8-битови вътрешни регистра:

— акумулатор А — регистър с универсално предназначение;

— индексен регистър X;

— индексен регистър Y;

— указател на стека S;

— регистър на флаговете (регистър за състоянието на процесора P). Предназначението на битовете на регистър P е показано на фиг.2.4.

Броячът на инструкциите PC е 16-битов регистър, който задава адреса на следващата инструкция.

Микропроцесорът SM630 изпълнява 56 инструкции. Кратки сведения за инструкциите,

мнемоничните им кодове и съдържанието на регистъра на състоянието P след изпълнението им са дадени в приложение 1.

Начини на адресиране в SM630

Инструкциите могат да се използват с 13 вида адресация.

Непосредствена адресация. Инструкциите са двубайтови, като първият байт съдържа кода на инструкцията, а вторият — операнда. При изпълнението се използват акумулаторът и индексните регистри X и Y на SM630. Прилага се при 11 инструкции на микропроцесора.

Абсолютна адресация. Инструкциите са трибайтови. Първият байт съдържа кода на инструкцията, вторият байт — младшите 8 бита на действителния адрес на операнда, а третият — старшите 8 бита на същия адрес. Използва се от 23 инструкции на SM630.

Адресация в нулевата страница. Инструкциите са двубайтови. Първият байт съдържа кода на инструкцията, а вторият — младшият байт на действителния адрес на операнда (старшият байт на действителния адрес е 00*). Прилага се при 21 инструкции на микропроцесора.

Подразбираща се адресация. Инструкциите се състоят само от един байт, който освен кода на инструкцията, определя кой от вътрешните регистри или флагове на микропроцесора е операнд на инструкцията. Използва се от 25 инструкции на SM630.

Bit		
0	C	- пренос (Carry)
1	Z	- нула (Zero)
2	I	- забрана за прекъсване (Interrupt disable)
3	D	- десетична аритметика (Decimal mode)
4	B	- програмно прекъсване BRK (Break command)
5		- не се използва
6	V	- препълване (Overflow)
7	N	- отрицателно число (Negative)

Фиг.2.4. Съдържание на регистъра на флаговете в SM630

*Навсякъде в справочника съдържанието на регистрите, данните и адресната информация се представят като шестнадесетични числа без знак.

Акумулаторна адресация. Инструкциите се състоят само от един байт, който съдържа кода на инструкцията. Подразбира се, че операндът е съдържанието на акумулатора А. Използва се от 4 инструкции на микропроцесора.

Индексна абсолютна адресация с регистъра X. Инструкциите са трибайтови. Първият байт съдържа кода на инструкцията, вторият — младшият байт на базовия адрес и третият — старшият байт на базовия адрес. Действителният адрес на операнда се получава, като към базовия адрес се прибави съдържанието на индексния регистър X (индекс или отместване). Използва се от 15 инструкции на CM630.

Индексна абсолютна адресация с регистъра Y. Напълно идентичен с предишния метод на адресация, като вместо съдържанието на индексния регистър X се използва съдържанието на индексния регистър Y. Прилага се при 9 инструкции на микропроцесора.

Индексна адресация в нулевата страница с регистъра X. Инструкциите са двубайтови, като първият байт съдържа кода на инструкцията, а вторият — младшият байт на базовия адрес от нулевата страница (старшият байт на базовия адрес е 00). Действителният адрес на операнда се получава, като към базовия адрес се прибави съдържанието на индексния регистър X. Преносът при събирането се пренебрегва. Използва се от 16 инструкции на CM630.

Индексна адресация в нулевата страница с регистъра Y. Напълно идентичен с предишния метод на адресация, като вместо съдържанието на индексния регистър X се използва съдържанието на индексния регистър Y. Прилага се само при инструкциите LDX и STX.

Относителна адресация. Инструкциите са двубайтови, като първият байт съдържа кода на инструкцията, а вторият байт — отместване в допълнителен код. Отместването е число, което се прибавя към съдържанието на програмния брояч IP и е в обхвата от -128 до +127 байта спрямо първия байт на следващата инструкция. Използва се от 8 инструкции за условен преход.

Абсолютна косвена адресация. Използва се от инструкцията JMP, която е трибайтова. Първият байт съдържа кода на инструкцията, вторият — младшият байт на адрес, а третият — старшият байт на адрес. В две последователни клетки от паметта, адресирани от втория и третия байт на инструкцията, се намира действителният адрес на прехода.

Косвена адресация с предварително индексирание по X. Инструкциите са двубайтови. Първият байт съдържа кода на инструкцията, а вторият — младшият байт на базовия адрес (по подразбиране старшият байт е 00). Младшият байт на базовия адрес се сумира със съдържанието на индексния регистър X, като преносът се пренебрегва. На получения адрес в нулевата страница се намира клетка, съдържаща младшият байт на действителния адрес на операнда, а в следващата клетка е записан старшият байт на действителния адрес на операнда. Използва се от 8 инструкции на микропроцесора.

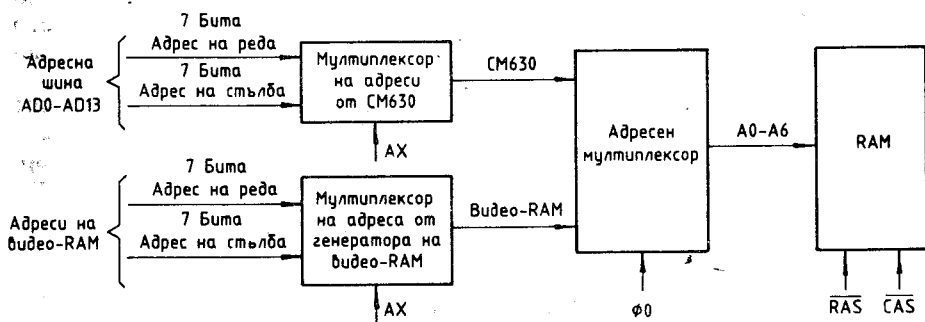
Косвена адресация с последващо индексирание по Y. Инструкциите са двубайтови, като първият байт съдържа кода на инструкцията, а вторият — младшият байт на адрес в нулевата страница (старшият байт е 00). В две последователни клетки от нулевата страница, адресирани от втория байт на инструкцията, се намира базовият адрес. Съдържанието на индексния регистър Y се прибавя към младшият байт на базовия адрес,

а възможният пренос се прибавя към старшия байт на базовия адрес. Резултатът от сумирането е действителният адрес на операнда. Прилага се при 8 инструкции на CM630.

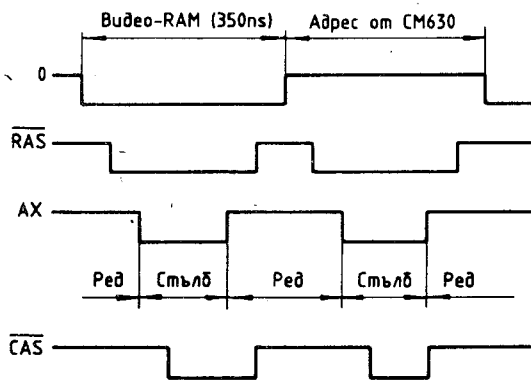
Организация на паметта на ПК Правец-82

Оперативната памет на Правец-82 е с обем 48 Кбайта с възможност за разширение до 64 Кбайта. Реализирана е с динамични RAM памети I4116. Тя е разположена в адресното пространство от 0000 до C000. Осембитовите индексни регистри на микропроцесора CM630 — X и Y дават възможност за адресиране на 256 страници по 256 байта всяка, което определя и логическото разделяне на цялото адресно пространство на персоналния компютър на страници с размер 256 байта. За адресиране на първите 256 байта от оперативната памет (адреси от 0000 до 00FF) в микропроцесора CM630 са предвидени отделни инструкции за адресация в нулева страница. Това определя и по-различното използване на тази страница от адресното пространство на персоналния компютър.

Оперативната памет на персоналния компютър Правец-82 се използва съвместно от микропроцесора, периферните устройства и от схемите за формиране на видеосигнала. На фиг.2.5 е дадена опростена схема на



а



б

Фиг.2.5. Схема на достъпа до паметта в Правец-82
 а - блокова схема на адресните мултиплексори на паметта;
 б - времедиаграма на оперативната памет

адресния мултиплексор на паметта и времедиаграма на един цикъл на обмен. По време на първия полупериод на сигнала Φ_0 достъп до паметта имат видеоконтролерните схеми, а по време на втория — микропроцесорът и периферните контролери в режим на ДДП. Сигналят АХ осигурява мултиплексването на адресите на реда и на стълба на клетките от оперативната памет, като ниското му ниво съответства на избор на адрес на стълба.

Разпределението на адресното пространство на оперативната памет е показано в табл. 2.1.

Таблица 2.1

Разпределение на адресното пространство на оперативната памет в Правец-82

Номер на страницата	Адреси	Предназначение
0	0000 — 00FF	Работна област на монитора, интерпретатора на БЕЙСИК и ДОС
1	0100 — 01FF	Област на стека
2	0200 — 02FF	Буфер на символите, въведени от клавиатурата
3	0300 — 03FF	Работна област на ДОС, монитора и потребителя
4 — 7	0400 — 07FF	Буфер на текстовата страница и на графиката с малка разделителна способност
8 — 11	0800 — 0BFF	Втори буфер за текст и графика с малка разделителна способност
	0800 —	Начало на областта за програми на интерпретатора на БЕЙСИК
32 — 63	2000 — 3FFF	Първа страница за режим на графика с голяма разделителна способност
64 — 95	4000 — 5FFF	Втора страница за режим на графика с голяма разделителна способност
96 — 149	6000 — 95FF	Работна памет за данни
150 — 191	9600 — BFFF	Резидентна област на ДОС

Постоянната памет е с обем 12 Кбайта. Реализирана е с 6 чипа памети EPROM от типа I2716 (всяка с обем 2 Кбайта \times 8). Паметта е разпределена между вградения интерпретатор на програмния език БЕЙСИК (от адрес D000 до F7FF) и монитора на персоналния компютър (от адрес F800 до FFFF). Свързващите вектори на монитора са разположени в старшите адреси на страница 3 (от адрес 03D0 до 03FF). Така например на адрес 03D0 се намира команда JMP за преход към интерпретатора на БЕЙСИК.

Организация на регистрите на входно-изходните устройства

Регистрите на входно-изходните устройства на персоналния компютър са разделени на две основни групи: регистри, разположени на системната платка и регистри на контролерите (периферните адаптери).

Разположените на системната платка регистри са 22. Девет от тях са входни, останалите тринадесет регистъра са изходни. Адресите на регистрите и посоката на обмен с тях са дадени в табл. 2.2.

Разположение на регистрите на входно-изходните устройства
на системната платка в адресното пространство на персоналния компютър

Адрес	Предназначение
C000	Вход за данни от клавиатурата
C010	Строб за нулиране на клавиатурата (изход)
C020	Изход към касетофон
C030	Високоговорител (изход)
C040 до C04F	Строб към съединителя за игрови адаптери (изходи)
C050	Ключ за графичен режим (GR или HGR) — изход
C051	Ключ за текстови режим (TEXT) — изход
C052	Ключ само текст или само графика (изход)
C053	Ключ смесен текст и графика (изход)
C054	Ключ за първа страница (PAGE 1) — изход
C055	Ключ за втора страница (PAGE 2) — изход
C056	Ключ за малка разделителна способност (GR) — изход
C057	Ключ за голяма разделителна способност (HGR) — изход
C058	Цифров изход AN0 изключен
C059	Цифров изход AN0 включен
C05A	Цифров изход AN1 изключен
C05B	Цифров изход AN1 включен
C05C	Цифров изход AN2 изключен
C05D	Цифров изход AN2 включен
C05E	Цифров изход AN3 изключен
C05F	Цифров изход AN3 включен
C060 и C068	Вход от касетофон
C061	Цифров вход PB0
C062	Цифров вход PB1
C063	Цифров вход PB1
C064	Стойност на аналогов вход GC0
C065	Стойност на аналогов вход GC1
C066	Стойност на аналогов вход GC2
C067	Стойност на аналогов вход GC3
C06C до C06F	Управление на таймерите на аналоговите входове (изходи)
C070	Строб за нулиране на аналоговите входове (изход)

За разпределяне на входно-изходните регистри в периферните адаптери на персоналния компютър Правец-82 е предвидена специална схема за дешифриране на адресите на слотовете за разширение. Всеки слот има собствени линии за избор. Те предоставят на всяко периферно устройство по два адресни блока: 16 адреса за входно-изходните регистри на контролера и 256 адреса в адресното пространство, предназначено за постоянна памет — за програмите, които са записани в PROM, разположен върху контролера.

Разпределението на адресното пространство, запазено за входно-изходни регистри на периферните адаптери с обем по 16 последователни адреса за всеки слот на персоналния компютър, е дадено в табл. 2.3, а разпределението на адресното пространство за PROM на периферните адаптери с обем по 256 последователни адреса за всеки слот — в табл. 2.4.

Таблица 2.3

**Разпределение на адресното пространство, запазено за регистри
на периферните адаптери, между слотовете на персоналния компютър**

Адрес	Предназначение
C080 — C08F	16 адреса за слот 0
C090 — C09F	16 адреса за слот 1
C0A0 — C0AF	16 адреса за слот 2
C0B0 — C0BF	16 адреса за слот 3
C0C0 — C0CF	16 адреса за слот 4
C0D0 — C0DF	16 адреса за слот 5
C0E0 — C0EF	16 адреса за слот 6
C0F0 — C0FF	16 адреса за слот 7

Таблица 2.4

**Разпределение на адресното пространство за PROM
на периферните адаптери между слотовете на персоналния компютър**

Адрес	Предназначение
C100 — C1FF	256 адреса за слот 0
C200 — C2FF	256 адреса за слот 1
C300 — C3FF	256 адреса за слот 2
C400 — C4FF	256 адреса за слот 3
C500 — C5FF	256 адреса за слот 4
C600 — C6FF	256 адреса за слот 5
C700 — C7FF	256 адреса за слот 6

Характеристики на видеомонитора

Видеомониторът на персоналния компютър Правец-82 и схемите за неговото управление работят в три основни режима: монохроматичен текстов, графика с малка разделителна способност (позволява изобразяването на 16 цвята) и графика с голяма разделителна способност (позволява изобразяването на 6 цвята).

В текстов режим форматът на екрана е 25 реда по 40 символа, които могат да се съхраняват в две страници. Матрицата на символа е 7×8 точки, като 5×7 от тях се използват за изобразяване на самия символ.

В режима на графика с малка разделителна способност на екрана на видеомонитора се изобразяват 1920 точки — 48 реда по 40 точки. Отделната точка може да бъде оцветена в един от 16 цвята. Всеки байт от паметта на изображението съдържа информация за две съседни по вертикала точки: старшият полубайт е определен за долната, а младшият — за горната точка. В четирите бита на полубайта се съдържа кодът на цвета на точката.

Графиката с голяма разделителна способност позволява изобразяването на 192×280 точки. На практика в режим на черно-бяла графика броят на точките по хоризонтала достига 560. Когато графиката е цветна, за сметка на информацията за цвят (шест цвята) броят на точките по хоризонтала намалява до 140. Съществуват извесни ограничения на

този графичен режим: две съседни точки не могат да бъдат изобразявани в цвят и цветът на всяка точка може да бъде избран от следните две палитри:

- виолетов, зелен, бял и черен;
- син, оранжев, бял и черен.

Един байт от паметта съдържа информация за 7 съседни по хоризонтала точки. Така за една страница в този графичен режим са необходими 8 Кбайта памет.

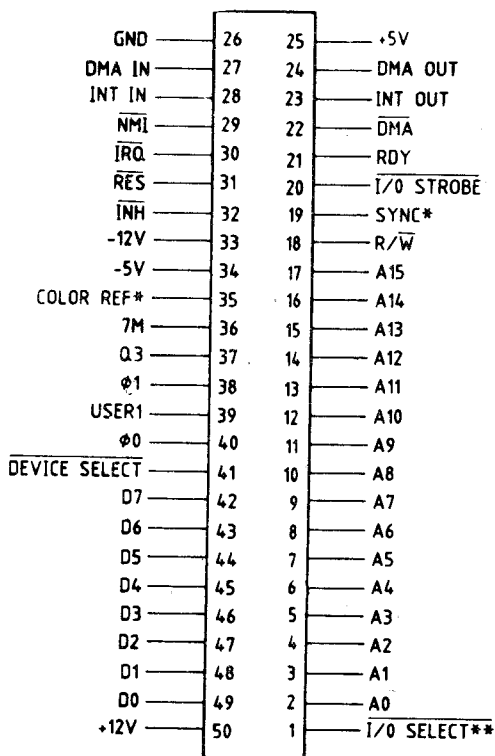
Системен интерфейс

Персоналният компютър Правец-82 има осем 50-изводни слота за периферни адаптери (контролери) на входно-изходни устройства, разширяващи базовата му конфигурация. Разположението на сигналите и тяхното наименование е показано на фиг.2.6. Сигналите, отбелязани с една звездичка, са свързани само към слот 7, а сигналът I/O SELECT (отбелязан с две звездички) не е свързан към слот 0. Slot 7 е определен за поставяне на текстов видеоконтролер с формат на екрана 25 реда по 80 символа, което обяснява особеностите му по отношение на системния интерфейс.

Характерни за интерфейса са веригите за определяне на приоритета при изпълнение на обмен с директен достъп до паметта (ДДП) и при обслужване на заявка за прекъсване.

Възможността за обмен чрез ДДП може да се използва само от един периферен адаптер в даден момент. Веригата за определяне на приоритета започва от извод 27 на слот 0 и продължава до извод 24 на слот 7, който не е свързан никъде и не се използва. За да се реализира обмен чрез ДДП, е необходимо инсталираните платки да създават непрекъснатата верига за определяне на приоритета, като платката, която е разположена най-близо до слот 0, има най-висок приоритет.

Веригата за определяне на приоритета при обслужване на заявка за прекъсване има същите характеристики като разгледаната верига за приоритета на ДДП. Тя започва от извод 28 на слот 0 и продължава до извод 23 на слот 7.



Фиг.2.6. Разположение на сигналите в слота на персоналния компютър Правец-82

2.1.2. ПЕРСОНАЛЕН КОМПЮТЪР ПРАВЕЦ-8М

Правец-8М е модернизиран вариант на базовия персонален компютър Правец-82.

Оперативната памет в Правец-8М е с обем 64 Кбайта, а постоянната е с обем 12 Кбайта. Паметта RAM се реализира с 8 чипа от типа 4864 (64К × 1), а постоянната памет — с два чипа I2764 или един I27128. Върху основната платка на Правец-8М е предвидена възможност за разширение на паметта. За работа с операционна система CP/M е добавен втори процесор Z80 (U880).

За да се запази програмната съвместимост с Правец-82, е създадена същата вътрешна организация на оперативната памет. Първите 48 Кбайта на паметта са аналогични на съществуващите при Правец-82, като в случая е отпаднала необходимостта от формиране на три отделни сигнала CAS за различните банки. Вместо тези сигнали е използван директен сигналът, изработван от тактов генератор.

По-големият обем на използваните схеми изисква добавяне на още един мултиплексирания адрес RA7. Той се формира от допълнителен мултиплексор 74LS153, на който се подават старшият адрес A14, а също така и сигнал, определящ коя от графичните страници е активирана в момента, за да се осигури правилното извеждане на информацията на екрана.

Последните 16 Кбайта от паметта имат аналогична организация на стандартното разширение на паметта за Правец-82. На част от освободеното място на основната платка е монтирана схема за управление на тези 16 Кбайта. Тази схема функционално е подобна на схемата за управление на RAM за Правец-82 със следните по-съществени различия:

- премахната е частта, формираща самостоятелен сигнал CAS за паметите от разширението;

- въведено е допълнително управление на общия мултиплексирания адрес RA6, за да могат да се получат три отделни блока памет в рамките на последните 16 Кбайта, съответстващи на организацията в разширението за Правец-82;

- въведена е възможност за изключване на допълнителните 16 Кбайта разширение, за да се избегнат конфликтите при включване на по-големи разширения (128 Кбайта). Изключването на последните 16 Кбайта става с помощта на монтирано на основната платка ключе от тип ПИС-8.

Монтираният втори процесор Z80 позволява работа с широко разпространената операционна система CP/M. Функционално това разширение е аналогично на модул CP/M за Правец-82, като са въведени някои апаратни подобрения.

Така например адресното отместване между 6502 и Z80 е осъществено с помощта на специално програмиран биполярен PROM от типа 82S129 вместо популярната платка на фирмата Microsoft.

Модулът CP/M заема адресното пространство на четвърти слот. Поради това на него не трябва да се монтира периферни платки, които използват сигналите DEV.SEL. и I/O SEL., когато е активирана операционна система CP/M. При работа с ДОС 3.3 тези ограничения отпадат.

Управлението на модул CP/M става също с помощта на ключета, като функциите на целия превключвател са дадени в табл. 2.5.

Функции на превключвателя за използване на CP/M

Ключе	Предназначение	Положение	
		0	1
1	Оперативна памет	64K	48K
2	Z80 — ДДП	OFF	ON
3	Z80 — Немаскируемо прекъсване	OFF	ON
4	Z80 — Маскируемо прекъсване	OFF	ON
5	Z80 — Изместване на адресното пространство	ON	OFF
6	Знаков генератор	NORM	ALT
7	Звук	OFF	ON
8	Не се използва		

В знаковия генератор на Правец-8М е предвидено място за четири знакови таблици. Две от тях се определят от ключе 6 на превключвателя, при което на екрана се изобразяват съответно кирилица и латиница — само главни, или латиница — главни и малки букви. Останалите две таблици могат да се превключат от клавиатурата и да съдържат символи, зададени от потребителя.

Клавиатурата на Правец-8М има някои подобрения:

- въведено е автоматично повтаряне на натиснатия клавиш;
- добавени са клавиши със стрелки за движение на курсора;
- включени са кодовете на всички ASCII-символи;
- таблицата на генерираните символи е записана в EPROM, което позволява произволно пренареждане на клавиатурата при специална заявка;
- коригирани са някои несъответствия в кодовата таблица на Правец-82;

— променено е действието на клавишите SHIFT и LOCK, което създава допълнителни удобства на потребителя.

Вграденото на допълнителните 16 Кбайта в компютъра се получава донякъде автоматично с използването на новата организация на паметта (64Кx1). При Правец-82 това се постига с включването на допълнителна платка в нулевия слот. Същият слот се използва и за по-голямо разширение на паметта чрез модул от 128 Кбайта. Именно тази възможност не може пряко да се използва в известните модели с 64 Кбайта организация поради това, че в адресното поле на процесора се получават две области с физически налична памет с еднакви адреси, което е конфликтна ситуация. При Правец-8М този проблем е решен с превключване на паметта за работа с 48 Кбайта адресно пространство както при Правец-82.

Процесорът Z80 изпълнява всички инструкции на I8080 плюс някои допълнителни. Могат да се изпълняват програми, написани за I8080 или за Z80.

CP/M като операционна система може да се приложи в два варианта. Първият вариант се използва при работа с оперативна памет 48 Кбайта, от които 4 Кбайта са необходими за поддържане на екрана и програмите на CP/M за четене/запис. При втория вариант оперативната памет е 56 Кбайта, като отново 4 Кбайта са отделени за екрана.

Разпределение на паметта при работа със СР/М

6502(СМ630) Адреси	Z80(U880) Адреси	Предназначение
0B00 — 0FFF	FF80 — FFFF	Дискови драйвери и буфери
0400 — 07FF	F400 — F7FF	Екранна памет
0200 — 03FF	F200 — F3FF	В/И конфигурация
0000 — 01FF	0F00 — 0F1F	За стек и нулева страница на резервираната памет в 6502
C000 — CFFF	E000 — EFFF	В/И област от паметта
FFFA — FFFF	DFFA — DFFF	Вектори за начално установяване и немаскируемо прекъсване в 6502
D400 — FFF9	C400 — DFF9	56K за СР/М при работа с 64K
D000 — D3FF	C000 — C3FF	Старши адреси на свободната памет RAM при система 56K
A400 — BFFF	9400 — 0AFF	44K свободна памет за СР/М
1000 — A3FF	0000 — 93FF	Младши адреси на свободната памет RAM

Всички прекъсвания в Правец-8М се поддържат от СМ630, който приема заявките за прекъсване. Извършва се анализ към кой от двата микропроцесора е издадена заявката и се стартира съответстващата подпрограма за обслужване на прекъсването.

Към 8-разредните компютри Правец се произвеждат голям брой контролери, осигуряващи включването на различни по вид и предназначение периферни устройства.

Някои от периферните модули работят само в точно определени слотове, но повечето от тях могат да бъдат инсталирани на произволен слот.

Контролер за флопидисково устройство

Това е основният контролер, който осигурява управлението на дисковата памет за нуждите на ДОС. Към един контролер могат да се включат до две ЗУГМД.

Периферните слотове са номерирани от 0 до 7, като се брои отляво надясно. Контролерът се монтира стандартно в слот номер 6, като слот 0 задължително е резервиран за модул от типа разширение на паметта — 16 Кбайта или 128 Кбайта.

Асинхронен последователен интерфейс

Предназначен е за работа с модем със скорост на предаване 300 бода. Чрез него Правец-82/8М може да се свързва с други четири компютъра по телефонна линия, да обменя файлове и програми.

Често този модул се използва за включване на печатащо устройство с последователен вход, като стандартно се монтира към слот 1 и порядко към слот 2.

Паралелен интерфейс

Той се използва за включване на печатащо устройство към компютъра. Монтира се към слот 1 или слот 2.

Допълнително се произвеждат следните модули:

— видеоконтролер (VIDEOCARD 80/40), разширяващ формата на екрана;

— RGB-контролер за работа с цветен монитор тип RGB;

— модул АЦП — аналогово-цифров преобразувател;

— модул ЦАП — цифрово-аналогов преобразувател, както и цял набор от инструментални измерителни модули.

2.1.3. ДИСКОВА ОПЕРАЦИОННА СИСТЕМА ДОС 3.3

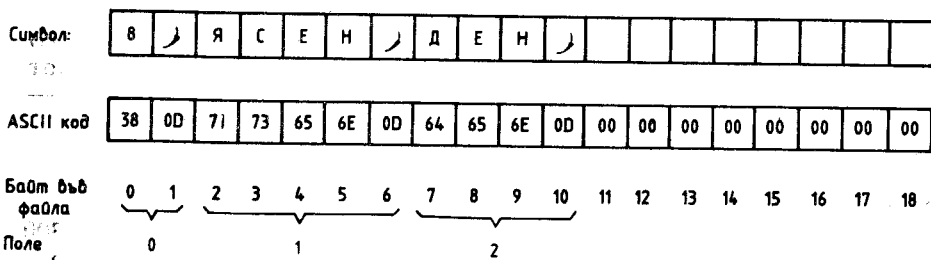
Възможностите на компютъра стават особено осезаеми, когато той работи под управлението на операционна система. При 8-разредните персонални компютри отначало като външно запомнящо устройство се използваше обикновен битов касетофон. Сега като носител на информация се използва гъвкав или твърд магнитен диск. Записването, съхраняването и четенето на информация от касета се управлява с оператори от вградения в Правец-82 БЕЙСИК. Използването на касетофон към персоналния компютър има само едно предимство — касетофонът е широко разпространен битов уред.

Операционната система за Правец-82 и за Правец-8М е ДОС 3.3.

В Правец-82 се използват два типа текстови файлове: с последователен и с директен достъп. И двата съхраняват низове в код ASCII, но в различни формати. Символът (↵) замества RETURN и се изпраща автоматично в края на повечето оператори PRINT.

На фиг.2.7 е показан последователен текстов файл, а на фиг.2.8 — текстов файл с директен достъп.

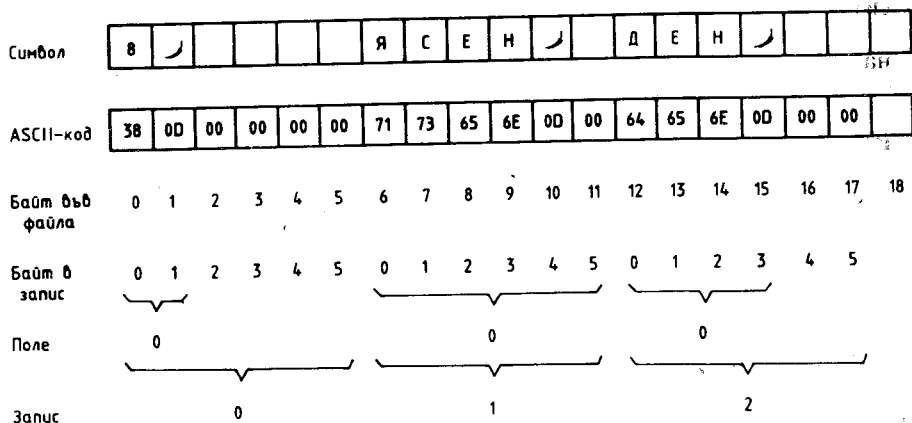
Последователните текстови файлове се създават по-лесно, но те са по-бавни за четене и при някои случаи по-неудобни за използване отколкото текстовите файлове с директен достъп.



Фиг.2.7. Структура на последователен текстов файл в персоналния компютър Правец-82

Зареждането на ДОС 3.3 става автоматично от системна дискета, като е възможно да се извърши и от програмиста с помощта на оператор PR #6, ако контролерът е включен в слот 6. След зареждането ДОС заема областта от паметта от адрес 9D00 до адрес FFFF. Областта от

адрес 9600 до адрес 9CF8 се заема от трите файлови буфера за вход и изход, всеки с размери по 263 байта. След извършеното зареждане, е възможно да се инициализира всяка нова дискета. Този процес включва разделяне на дискетата на 16 сектора и 35 пътечки, както и записване на ДОС върху първите (най-външните) две пътечки.



Фиг. 2.8. Структура на текстов файл с директен достъп в персоналия компютър Правец-82

Основните команди в ДОС 3.3 са: CATALOG, INIT, LOAD, RUN, SAVE, RENAME, DELETE.

Командата INIT се използва за инициализиране на нова дискета.

] INIT име на програма

При това програмата трябва да бъде подготвена от програмиста. Това е описано подробно в инструкцията на ДОС 3.3, приложена към компютъра [9].

В процеса на инициализиране се унищожава цялата информация върху дискетата.

CATALOG е команда, с чиято помощ се проверява съдържанието на дискетата, какво е точното име на файловете, на какъв език са, размерът и състоянието им — заключени или отключени. При задаване на командата CATALOG за системната дискета на Правец-82 се получава следната картина на екрана:

```

] CATALOG
DISK VOLUME 254
* A 003 ИМЕ НА ПРОГРАМА
* A 028 CBASIC
  В 011 МЕНЮ
* В 020 ОНФ
* А 004 ПРАВИ ТЕКСТ
  
```

DISK VOLUME има значението на „том номер“ и се установява още при инициализиране на дискетата, като номерът може да бъде от 1 до 254 (например INIT име на програма, V 128 ще инициализира дискетата с том номер 128). Буквата А показва, че програмата е написана на разширен БЕЙСИК.

Трицифреното число показва броя на заетите сектори, а звездичката преди буквата — заключените програми.

С различни букви се означават: целочислен БЕЙСИК (I), за файлове на машинен език (B) и споменатите вече по-горе текстови файлове (T).

Ако командата LOAD се напише, без да е последвана от име на файл, компютърът ще се обърне към касетния интерфейс и ако програмистът не се е погрижил там да има включен касетофон, излизането от ситуацията може да стане само с RESET или изключване на компютъра.

Дисковата команда LOAD трябва винаги да бъде следвана от името на файла:

] LOAD *име на файл*

Зареждането на двоичните файлове е аналогично, но командата е BLOAD.

RUN се използва за зареждане и автоматично стартиране изпълнението на програмата:

] RUN *име на програма*

Двоичните файлове се зареждат и изпълняват по същия алгоритъм, но с командата BRUN.

SAVE предизвиква запис на файл върху външен носител. След SAVE задължително следва име на файл, който се намира в оперативната памет на компютъра. В противен случай компютърът търси касетния интерфейс.

За двоични файлове командата е RSAVE.

DELETE изтрива от дискетата файла с име, указано след нея.

RENAME е за преименуване на файлове.

] RENAME *име на файл 1, име на файл 2* — ще замени името на файл 1 с името на файл 2.

Всяка от описаните команди има допълнителни параметри, които определят номера на слота, номера на дисковото устройство, номера на тома. За по-задълбочено изучаване може да се използва фирмената документация към компютъра [9].

Примерна програма, създаваща последователен текстов файл:

```
10 REM ПРИМЕР — СЪЗДАВА ФАЙЛ
```

```
20 DS = CHRS (4): REM CTRL-D
```

```
30 PRINT DS: "OPEN ПРИМЕР"
```

```
40 PRINT DS: "WRITE ПРИМЕР"
```

```
50 PRINT "МИКРОПРОЦЕСОРНИ"
```

```
60 PRINT "СИСТЕМИ"
```

```
70 PRINT "ПРАВЕЦ"
```

```
80 PRINT "МИКРОКОМПЮТЪРНА"
```

```
90 PRINT "ТЕХНИКА"
```

```
100 PRINT "АВАНГАРДНА"
```

```
110 PRINT "ТЕХНОЛОГИЯ"
```

```
120 PRINT DS: "CLOSE ПРИМЕР"
```

```
130 END
```

Ред 20 от програмата присвоява значението на променливата DS — комбинацията CTRL-D. Това е информация за компютъра, че следващите команди след DS са дискови команди.

Когато изпълни ред 30, компютърът отваря (OPEN) файл с име ПРИМЕР и в ред 40 записва последователен файл с данните от редовете 50 до 110. Следва затваряне на файла в ред 120 (CLOSE).

Ако се стартира програмата с RUN на екрана, няма да се покаже никаква информация. В ДОС 3.3 може да се наблюдава изпълнението на команди на ДОС върху екрана. Командата е MON C, I, O (monitor),

където С, I, O са параметри, задавани по избор.

С е параметър за извеждане на екрана на командите в последователността на тяхното изпълнение;

I е параметър за извеждане на информацията, постъпила в компютъра от ЗУГМД;

O е параметър за извеждане на информацията от компютъра към ЗУГМД.

При наличие на описаната по-горе програма в оперативната памет нека се въведат в компютъра следните команди:

```
] MON I,O
] RUN
На екрана ще се изведе:
OPEN ПРИМЕР
WRITE ПРИМЕР
МИКРОПРОЦЕСОРНИ
СИСТЕМИ
ПРАВЕЦ
МИКРОКОМПЮТЪРНА
ТЕХНИКА
АВАНГАРДНА
ТЕХНОЛОГИЯ
CLOSE ПРИМЕР
```

В каталога файлът е отбелязан с буквата Т. Командата MON се отменя само с NOMON.

Създаденият текстов файл може да се прочете със следната програма:

```
10 REM ЧЕТЕНЕ НА ПРИМЕР
20 DS = CHR$(4): REM CTRL-D
30 PRINT DS; "OPEN ПРИМЕР"
40 PRINT DS; "READ ПРИМЕР"
50 FOR I=1 TO 7
60 INPUT AS(I):PRINT AS(I)
70 NEXT I
80 PRINT DS; "CLOSE ПРИМЕР"
90 END
```

Нова команда се появява в ред 40 — READ.

Тази команда съобщава на ДОС, че следващите оператори INPUT AS (I) се отнасят за данни, които ще постъпят от дискетата, а не от клавиатурата. С помощта на RUN се проверява действието на програмата:

```
] RUN
МИКРОПРОЦЕСОРНИ
СИСТЕМИ
ПРАВЕЦ
МИКРОКОМПЮТЪРНА
ТЕХНИКА
АВАНГАРДНА
ТЕХНОЛОГИЯ
```

Създаването и използването на последователни текстови файлове с програми на БЕЙСИК дава в ръцете на програмиста мощно средство за обработка на голямо количество данни.

Както е показано по-горе, текстовите файлове с директен достъп се отличават от последователните по това, че при тях предварително се

резервира клетка с определена дължина, наречена *запис*. Когато се записва в тези клетки, техният обем не винаги се използва ефективно, но лесно и бързо се извлича или видоизменя информация от коя да е част на файла.

Такива файлове са подходящи при решаване на задачи, свързани с бърз достъп до различни части на файла, или когато трябва често да се променя определена информация.

Файловете с директен достъп се създават по подобен начин, но някои команди имат допълнителни параметри:

OPEN изисква L (дължина на запис), а READ и WRITE използват R (номер на записа). Ето една двойка програми, която създава и чете текстов файл с директен достъп:

```
10 REM СЪЗДАВА ФАЙЛ РЕГИСТРАЦИЯ
20 DS=CHRS(4):REM CTRL-D
30 INPUT "НОМЕР НА ПИСМОТО:"; NS
40 INPUT "ВХОДЯЩ НОМЕР:"; PS
50 INPUT "АДРЕСАНТ:"; ZS
60 PRINT DS; "OPEN РЕГИСТЪР, L200"
70 PRINT DS; "WRITE РЕГИСТЪР, R1"
80 PRINT NS: PRINT PS:PRINT ZS
90 PRINT DS; "CLOSE РЕГИСТЪР"
```

В ред 60 се отваря файл РЕГИСТЪР с дължина на записа 200 байта. Ред 70 подготвя за запис информацията в запис 1, а ред 80 изпраща NS и ZS върху дискетата и тази информация се записва в запис 1.

```
] MON C, O
] RUN
НОМЕР НА ПИСМОТО: 11-20-403/10.01.88
ВХОДЯЩ НОМЕР : 2845/20.01.88
АДРЕСАНТ : БРВ
OPEN РЕГИСТЪР, L200
WRITE РЕГИСТЪР, R1
11-20-403/10.01.88
2845/20.01.88
БРВ
CLOSE РЕГИСТЪР
```

Записът може да се прочете със следната програма:

```
10 REM ЧЕТЕ ФАЙЛ РЕГИСТЪР
20 DS=CHRS(4): REM CTRL-D
30 PRINT DS; "OPEN РЕГИСТЪР, L200"
40 PRINT DS; "READ РЕГИСТЪР, R1"
50 INPUT N1$, P1$, Z1$:PRINT N1$, P1$, Z1$
60 PRINT DS; "CLOSE РЕГИСТЪР"
```

При желание да се правят повече записи вместо 1 след 40 е необходимо да се укаже име на променливата, като преди това се определи границата, в която ще се променя тази променлива.

Командите на ДОС 3.3 LOCK и VERIFY служат за защита на файловете от случайно изтриване и за проверка на записите.

LOCK се задава с името на файла, който трябва да бъде заключен (защитен).

```
]LOCK име на файл
```

В резултат се появява звездичка в каталога и не може да се запише друг файл със същото име, нито да се изтрие този. Действието на тази команда се отменя с командата UNLOCK.

VERIFY проверява файл, който може да е повреден или записан неправилно.

При откриване на грешка, свързана с използването на дисковите устройства, ДОС издава съобщение, описващо грешката и спира изпълнението на програмата. Съобщенията са подобни на познатите съобщения за грешки в разширения и целочислен БЕЙСИК, но докато в разширения БЕЙСИК съобщението има отпред въпросителен знак (?), а в целочисления БЕЙСИК има три звездички (***), в ДОС няма никакви знаци преди съобщението.

ДОС изпраща 15 съобщения за грешки, които са описани във фирмената литература [9].

2.2. ОСЕМРАЗРЕДНИ ПЕРСОНАЛНИ КОМПЮТРИ С ДИСКОВА ОПЕРАЦИОННА СИСТЕМА CP/M

2.2.1. СТРУКТУРА И ОРГАНИЗАЦИЯ НА CP/M 2.2

Дисковата операционна система CP/M (Control Program for Microprocessors) е създадена за микропроцесорни изделия, реализирани с микропроцесора I8080. Тя се изпълнява и с микропроцесорите I8085 и Z80, както и с всички функционални аналози на тези три микропроцесора, произведени от голям брой фирми в света.

На фиг.2.9 е дадена блоковата схема на микропроцесора I8080. Той съдържа шест 8-разредни регистъра с общо предназначение и един акумулатор. Регистрите с общо предназначение могат да се адресират поотделно или по двойки. Аритметичните и логическите операции установяват или нулират четири флага. Още един флаг се използва при аритметични операции с десетични числа. Процесорът може да съхранява във външен стек съдържанието на акумулатора, флаговете, програмния брояч или регистрите с общо предназначение. Адресирането на стека става чрез 16-разреден указател на стека. В приложение 2 е даден наборът инструкции на I8080.

Микропроцесорът Z80 е напълно съвместим програмно с I8080, но е развит в сравнение с него както в схемно, така и в програмно отношение. Блокова схема на Z80 е показана на фиг.2.10.

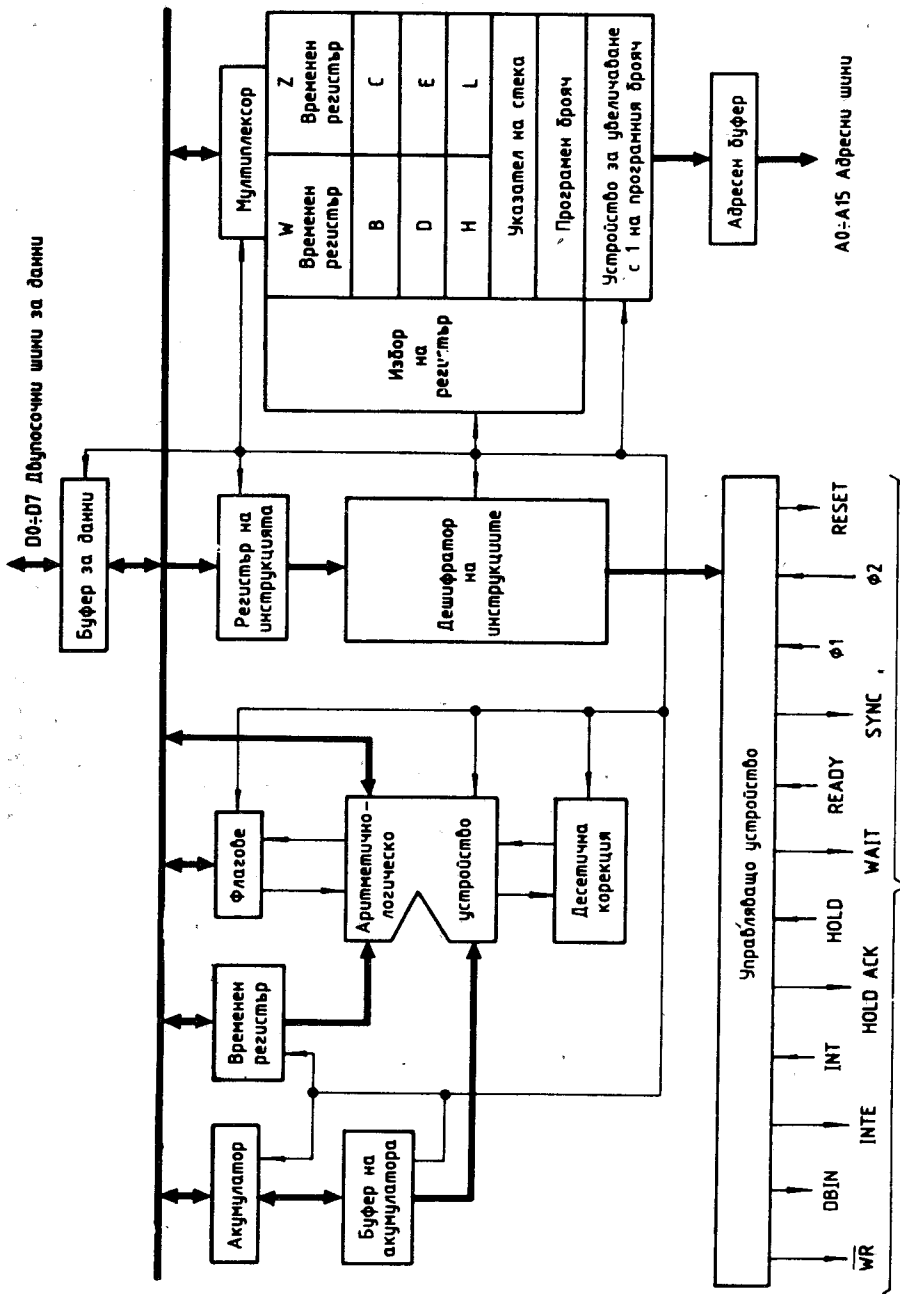
Z80 съдържа:

- две банки, включващи по шест регистъра с общо предназначение, акумулатор и флагове;
- два 16-разредни индексни регистъра;
- 16-разреден указател на стека;
- 16-разреден програмен брояч;
- регистър за адрес на страницата от паметта, в която се разполагат адресите на програмите, обслужващи прекъсванията;
- регистър за адрес на опресняване на паметта.

В приложение 3 е даден наборът инструкции на Z80.

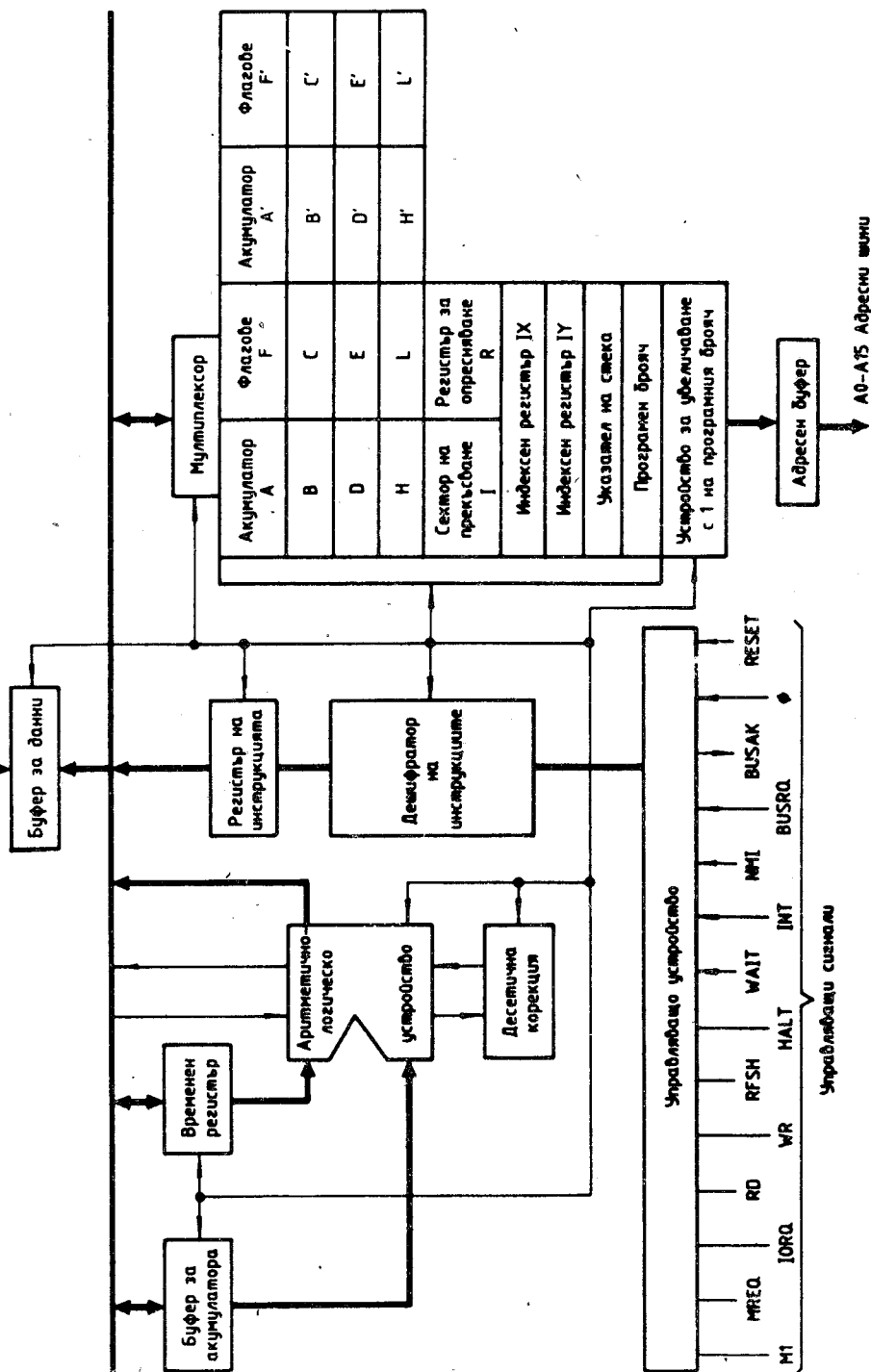
Подробно описание на микропроцесорите и указания за програмиране могат да се намерят в [3,6,7,10,13,15].

CP/M включва следните структурно оформени основни модули:



Управляващи сигнали

Фиг. 2.9. Блокова схема на микропроцесора I8080



Фиг. 2.10. Блокова схема на микропроцесора Z80

- BIOS (Basic Input/Output System) — базова система за вход/изход;
 BDOS (Basic Disk Operating System) — базова дискова операционна система;
 CCP (Control Command Processor) — програма за обработване на командите от конзолата;
 TPA (Transient Program Area) — област за зареждане на транзитни програми — външни команди и приложни програми.

В персоналните компютри, произведени след 1980 г., операционната система CP/M се разполага обикновено в адресното пространство на 64-Кбайтова оперативна памет. Клетките на паметта с адреси от 0000 до 00FF са резервирани за параметри на операционната система. От адрес 0100 започва областта на TPA, в която се разполагат програмите. Стартовият адрес на тези програми е 0100. Резервираната за TPA област обикновено надхвърля 48 Кбайта. Това осигурява изпълнението на програми, които изискват в произволен момент от време оперативна памет с обем не повече от 48 Кбайта. Програмите с по-голям обем трябва да използват техниката на припокриване.

След областта, резервирана за TPA, са разположени съответно модулите CCP, BDOS и BIOS.

BIOS се състои от управляващи подпрограми за връзка на операционната система с периферните устройства на ПК. Обръщението към тези подпрограми се извършва през дефинирани входни точки, разположени на последователни адреси веднага след края на BDOS. Всяка от тези подпрограми завършва с инструкция RET за правилно връщане към изпълнението на програмата, предизвикала активирането на съответната управляваща подпрограма. В модула BIOS са разположени управляващи подпрограми, които осигуряват връзката със следните стандартни периферни устройства на типичния ПК:

- две ЗУГМД (възможно е до четири ЗУГМД);
- клавиатура;
- текстов видеомонитор;
- печатащо устройство;
- изход по последователен интерфейс;
- вход от последователен интерфейс.

Освен запомнящите устройства на магнитен диск операционната система CP/M различава четири логически периферни устройства:

- CON: (Console) — клавиатурата и текстовия видеомонитор;
- LST: (List) — текстово печатащо устройство;
- PUN: (Punch) — изход по последователен интерфейс;
- RDR: (Reader) — вход от последователен интерфейс.

BDOS осигурява динамичното разпределение на дисковото пространство, организацията му във файлове и достъпа до тях. Всеки файл се състои от блокове обикновено с обем 2 Кбайта. Блоковете се състоят от логически сектори от по 128 байта. Най-често в персоналните компютри от тази група в един физически сектор с размер 512 байта се разполагат четири логически сектора.

CCP осъществява връзката между потребителя, от една страна, и ПК и приложните програми, от друга страна. Този модул на CP/M следи въвежданите от клавиатурата команди, грижи се за тяхната визуализация, разпознава ги и осигурява тяхното изпълнение. Това изпълне-

ние се свежда до извикването на разпознатата в командата вградена команда или транзитна програма и до нейното стартиране.

Върху системната дискета CP/M е разположена на пътечките с най-малки номера (0, 1 и т.н.) и броят им зависи от обема на информацията, записвана на една пътечка. Обикновено пътечката е разделена на 512-байтови физически сектори, като броят на секторите за конкретните ПК варира от 8 до 10.

Освен описаните модули на операционната система системната дискета съдържа и помощни програми, изпълнявани след зареждане в ТРА. По своята организация и разположение върху дискетата тези програми не се различават по нищо от разполаганите върху дискетите приложни програми. Наборът от тези програми, обобщавани под общото наименование външни команди на CP/M, се различава за отделните типове ПК.

Потребителят на ПК ползва операционната система по три начина:

— като извиква посредством клавиатурата команди и приложни програми за изпълнение;

— като се обръща в създадените от него програми към системни функции;

— като се обръща в своите програми към входни точки на BIOS.

Операционната система съобщава на потребителя, че очаква следващата команда, чрез визуализиране на името на текущо избраното дисково устройство и подсещания символ „>“. Извикването на командите става чрез въвеждането на команден ред от клавиатурата.

Команди на CP/M

Вградени команди на CP/M 2.2. Вградените (вътрешни) команди са част от CCP и се изпълняват незабавно:

ERA — изтрива зададените файлове от дискетата;

DIR — извежда на дисплея имената на файловете, намиращи се на дискетата;

REN — преименува зададен файл;

SAVE — запазва съдържанието на част от паметта като файл;

TYPE — извежда на дисплея съдържанието на файл;

USER — дава възможност за разделяне на дискетата на области за ползване от различни потребители.

Външни команди на CP/M 2.2. Външните команди са програми, намиращи се на дискетата като файлове с име, съответстващо на командата, и със суфикс COM. Под управлението на CCP те се зареждат в ТРА и се изпълняват.

Основните външни команди на CP/M 2.2 са:

FORMAT — форматира дискета;

DOSGEN — записва операционната система върху дискета;

PIP — (Peripheral Interchange Program) обменя файлове между периферни устройства;

STAT — информира за разпределението на дисковото пространство;

LOAD — създава изпълними файлове със суфикс COM, които стават транзитни програми за CCP;

DUMP — извежда на видеомонитора съдържанието на даден файл в шестнадесетичен код.

Обръщението към всяка потребителска програма в CP/M се извършва аналогично на обръщението към външна команда.

Системни функции в CP/M 2.2

При създаване на приложни програми програмистите ползват системните функции на CP/M за управление на периферните устройства и обмен с тях. Системните функции са 37 на брой. Те се стартират чрез изпълнение на инструкция CALL 0005 (обръщение към подпрограма, записана на адрес 0005) при зададен в регистър С номер на функцията. Преди извикване на системната функция нейните входни параметри трябва да бъдат установени в регистрите D и E. След изпълнението на функцията изходните ѝ параметри се получават в регистри A, H и L.

На адрес 0005 (в областта със системните параметри на CP/M) е записана инструкцията JP [адрес] за предаване на управлението на най-малкия адрес, от който е разположен модулът BDOS. Този адрес е записан в клетки от паметта 0006 и 0007 и определя началото на BDOS, както и максималният адрес, до който може да се разполага една транзитна програма. При това е допустимо програмата да заеме TPA и да припокрие CCP. Една такава максимална по обем програма преди приключването на изпълнението си трябва да възстанови CCP.

Изпълнението на инструкцията CALL 0005 предава управлението на BDOS. Изпълнението на самата системна функция се извършва от BDOS.

По-долу са изброени системните функции на CP/M 2.2 по реда на техните номера, като е дадено и състоянието на входните и изходните им параметри. Някои от тях използват блок за управление на файла (file control block) — FCB, който съдържа и името на файла. Подробности за тяхното използване се намират в експлоатационната документация, придружаваща персоналния компютър, или в описанията на операционната система [14].

- Функция 0 — начално установяване на системата
- Функция 1 — въвеждане на символ от конзолата
Изход: Регистър A — код на символа
- Функция 2 — извеждане на символ на конзолата
Вход: Регистър E — код на символа
- Функция 3 — въвеждане на символ от серийния интерфейс
Изход: Регистър A — код на символа
- Функция 4 — извеждане на символ по серийния интерфейс
Вход: Регистър E — код на символа
- Функция 5 — извеждане на символ на печат
Вход: Регистър E — код на символа
- Функция 6 — директен вход-изход на конзолата
Вход: Регистър E — FF (вход) или код на символа (изход)
Изход: Регистър A — код на символа или статус
- Функция 7 — четене на входно-изходния байт
Изход: Регистър A — входно-изходен байт
- Функция 8 — установяване на входно-изходния байт
Вход: Регистър E — входно-изходен байт
- Функция 9 — извеждане на низ
Вход: Регистри DE — адрес на низ
- Функция 10 — четене на ред в потребителски буфер
Вход: Регистри DE — адрес на буфер
Изход: Въведени символи в буфера
- Функция 11 — проверка на статуса на конзолата
Изход: Регистър A — статус на конзолата

- Функция 12 — четене на версията на CP/M
Изход: Регистри HL — номер на версията
- Функция 13 — начално установяване на дисковата система
- Функция 14 — избор на текущ диск
Вход: Регистър E — номер на диск
- Функция 15 — отваряне на файл
Вход: Регистри DE — адрес на FCB на файла
Изход: Регистър A — код от каталога на диска или FF, ако файлът не е намерен на диска
- Функция 16 — затваряне на файл
Вход: Регистри DE — адрес на FCB на файла
Изход: Регистър A — като за функция 15
- Функция 17 — търсене на файл от началото на каталога
Вход: Регистри DE — адрес на FCB на файла
Изход: Регистър A — като за функция 15
- Функция 18 — търсене на файл от достигнатото място в каталога
Вход: Регистри DE — адрес на FCB на файла
Изход: Регистър A — като за функция 15
- Функция 19 — изтриване на файл
Вход: Регистри DE — адрес на FCB на файла
Изход: Регистър A — като за функция 15
- Функция 20 — последователно четене на сектор от файл
Вход: Регистри DE — адрес на FCB на файла
Изход: Регистър A — 00, ако четенето е успешно, или стойност, различна от 00, ако последният сектор на файла е вече прочетен
- Функция 21 — последователен запис на сектор във файл
Вход: Регистри DE — адрес на FCB на файла
Изход: Регистър A — 00, ако четенето е успешно, или стойност, различна от 00, при запълване на диска
- Функция 22 — създаване на файл
Вход: Регистри DE — адрес на FCB на файла
Изход: Регистър A — код от каталога на диска или FF при запълване на каталога
- Функция 23 — преименуване на файл
Вход: Регистри DE — адрес на FCB на файла
Изход: Регистър A — като за функция 15
- Функция 24 — четене на информация за активни дискове
Изход: Регистри HL — статус на дисковете
- Функция 25 — четене на номера на текущо избрания диск
Вход: Регистър E — номер на диск
- Функция 26 — установяване на адрес на буфер в паметта за обмен с дисковете
Вход: Регистри DE — адрес на буфера за обмен
- Функция 27 — четене на адреса на таблицата за разпределение на дисковото пространство
Изход: Регистри HL — адрес на таблицата
- Функция 28 — установяване на защита от запис върху текущото избрано устройство
- Функция 29 — четене на информация за състоянието само четене на дисковете
Изход: Регистри HL — статус на дисковете
- Функция 30 — установяване на атрибути на файл по FCB
Вход: Регистри DE — адрес на FCB на файла
Изход: Регистър A — като за функция 15
- Функция 31 — четене на адреса на таблица, съдържаща параметрите на дисковете
Изход: Регистри HL — адрес на таблицата
- Функция 32 — четене/установяване на номера на активния потребител
Вход: Регистър E — FF за четене или номер на потребител
Изход: Регистър A — номер на активния потребител

- Функция 33 — четене на сектор от файл с произволен достъп до сектора
 Вход: Регистри DE — адрес на FCB на файла
 Изход: Регистър A — 00, ако четенето е успешно, или стойност, различна от 00, ако секторът не е прочетен
- Функция 34 — запис на сектор от файл с произволен достъп до сектора
 Вход: Регистри DE — адрес на FCB на файла
 Изход: Регистър A — 00, ако четенето е успешно, или стойност, различна от 00, ако секторът не е записан
- Функция 35 — изчисляване на размера на файла
 Вход: Регистри DE — адрес на FCB на файла
 Изход: адрес на сектора, следващ последния сектор на файла (установен в FCB)
- Функция 36 — четене на адреса на сектора, достигнат с последователен достъп
 Вход: Регистри DE — адрес на FCB на файла
 Изход: адрес на сектора (установен в FCB)

Входни точки в CP/M 2.2

Изпълнението на подпрограмите за управление на периферните устройства и обмен с тях, записани в BIOS, се извършва чрез обръщение към входни точки на BIOS. Те представляват последователност от 17 инструкции (JP) за предаване на управлението към първата инструкция от съответната подпрограма. Седемнадесетте инструкции са разположени в последователни клетки от паметта веднага след BDOS. Адресът на втората инструкция (представляваща входна точка към подпрограмата WBOOT) се съдържа в клетки 0001 и 0002. Тези клетки са част от параметрите на CP/M, разположени в областта от 0000 до 00FF. На адрес 0000 в тази област е записана инструкцията JP WBOOT за предаване на управлението към входната точка на подпрограмата WBOOT.

При работата на една приложна програма се извършва обръщение към входните точки в два случая:

— при обръщение към системните функции (последните се изпълняват от BDOS, но в повечето случаи чрез последователни обръщения към входните точки на BIOS);

— при директни обръщения към входните точки от самата приложна програма.

По-долу накратко е описано действието на подпрограмите, включени в BIOS на CP/M 2.2. Редът им съответства на възходящия ред на адресите на входните точки за тези подпрограми.

BOOT Програма BOOT (*start*) получава управлението след зареждането на операционната система от дискетата. Програмата предава управлението на операционната система на ниво CCP.

WBOOT Обръщение към WBOOT (*top of start*) се изпълнява, когато в потребителската програма се срещне инструкция JP 0000 или когато от клавиатурата се въведе CTRL-C. CCP и BDOS се зареждат отново от дискетата в оперативната памет. WBOOT не променя подразбиращото се дисково устройство. След изпълнението управлението се предава на CCP.

CONST Проверява дали има готов символ за четене от клавиатурата.

CONIN Чете в регистър A въвеждания от клавиатурата символ.

CONOUT Извежда символа, записан в регистър C, на видеомонитора.

LIST Извежда символа, записан в регистър C, на печат.

PUNCH Извежда символа, записан в регистър C, по последователния интерфейс.

READER Чете в регистър A постъпилия по последователния интерфейс символ.

HOME	Установява главата на подразбиращото се дисково устройство на пътека 00.
SELDSK	Избира за подразбиращо се дисковото устройство, указано в регистър С.
SETTRK	Установява номера на пътека на подразбиращото се дисково устройство за следваща дискова операция. Номерът се задава в регистри ВС.
SETSEC	Установява номера на сектор на подразбиращото се дисково устройство за следваща дискова операция. Номерът се задава в регистри ВС.
SETDMA	Установява адреса на буфера за обмен за следваща дискова операция. Адресът се задава в регистри ВС. Буферът е с големина 128 байта.
READ	След като дисковото устройство е избрано и са определени номерът на пътека, номерът на сектор и адресът на буфера за обмен с дисковете, READ чете един логически сектор и установява в регистър А стойност 00 при успешно четене или стойност 01 при възникнала грешка. Съдържанието на сектора постъпва в буфера за обмен.
WRITE	След като дисковото устройство е избрано и са определени номерът на пътека, номерът на сектор и адресът на буфера за обмен с дисковете, WRITE записва един логически сектор и установява в регистър А стойност 00 при успешен запис или стойност 01 при възникнала грешка. Съдържанието на буфера за обмен се записва в сектора.
LISTST	Проверява за готовност печатащото устройство.
SECTRAN	Преобразува логически номер на сектор във физически номер на сектор по зададена таблица. Подпрограмата е създадена за увеличаване на скоростта на обмен с диска чрез въвеждане на приплъзващ фактор. Това преобразуване е реализирано в BIOS и входната точка не трябва да се използва от програмиста.

Изпълнение на приложна програма от дискета в информационната среда на CP/M 2.2

Изпълнимата приложна програма е записана върху дискета като файл с име, съответстващо на името на програмата, и суфикс COM. Ако на екрана е изведен подсещащият символ „>“, операторът може да въведе от клавиатурата команден ред.

Командният ред съдържа само името на програмата или името, допълнено с един или два спецификатора на файлове, с които работи програмата. Спецификаторът на файл се състои от името на файла, точка (.) и суфикс.

ССР зарежда файла от дискетата в областта на ТРА с начален адрес 0100. Ако командният ред съдържа един или два спецификатора на файлове, ССР подготвя един или два блока за управление на файл в областта на системните параметри (от 0000 до 00FF). Първият блок се подготвя в област с начален адрес 005С, а вторият в област с начален адрес 006С. Приложната програма получава управлението от ССР (стартира се изпълнението и от адрес 0100).

Програмата може да завърши по два начина:

- с инструкция RET — управлението се предава на ССР;
- с инструкция JP 0000 — управлението се предава за изпълнение на подпрограма WBOOT (топъл старт) и оттам на ССР.

В първия случай програмата може да ползва паметта до края на областта ТРА (модулет ССР остава непроменен). Във втория случай програмата може да ползва паметта до началото на областта за BDOS. Приложната програма може да ползва стека на ССР или да формира

собствен стек. В първия случай при завършване на програмата преди инструкцията RET тя трябва да възстанови стека на ССР.

2.2.2. ПЕРИФЕРНИ УСТРОЙСТВА И КОНТРОЛЕРИ ЗА ТЯХ

Периферните устройства и контролерите за тях, вградени в 8-разредните ПК с ДОС СР/М 2.2, носят отпечатъка на особеностите на дисковата операционна система и на времето, през което са разработвани. Конструктивната реализация на СР/М съвместимите компютри е модулна. Като отделни модули в тях традиционно съществуват видеоконтролер и контролер за ЗУГМД. Останалите контролери за един стандартен компютър от този клас (за клавиатура, за печатащо устройство, за комуникация по последователен интерфейс) са разположени на системната платка на компютъра или последните два са конструктивно оформени в отделен модул. С оглед постигането на минимални схемни решения за контролерите и за целия ПК управляващите (драйверните) програми, обслужващи тези контролери, са разположени в схеми от тип ROM на системната платка на ПК. Върху нея се намира и микропроцесорът, на който е възложено управлението на контролерите. В тази група ПК не се използват решения, при които в контролерите за периферните устройства са вградени самостоятелни 8-разредни микропроцесори с цел осигуряване на паралелна асинхронна работа на контролерите и разтоварване на централния микропроцесор на ПК. Управляващите програми на тези контролери трябва да се разглеждат като неразделна част от тях. Те са записани в постоянната памет и от архитектурна гледна точка са част от BIOS на СР/М. За някои ПК в ROM е записана само програмата, необходима за начално зареждане на СР/М (програмата BOOT или част от нея). Управляващите програми за контролерите се зареждат в паметта на ПК заедно със самата операционна система. В този случай те се разполагат в област от паметта RAM, а не в ROM. Такова решение е нетипично и по същество то с нищо не променя организацията на обслужването на контролерите и периферните устройства.

Особеностите на СР/М, които влияят на реализацията на контролерите, са следните:

— обменът с логическите устройства, различавани от СР/М (клавиатура, видеомонитор, печатащо устройство, последователен вход и последователен изход), е посимволен, т.е. той включва по един символ при сеанс за обмен;

— обменът с дисковите устройства се извършва на ниво логически сектори от по 128 байта.

В редица ПК от тази група са вградени възможности, надхвърлящи тези ограничения (като графика, цвят и т.н.). Ползването на тези възможности се постига с програмно осигуряване, разпространявано за тези ПК. То не може да се нарече СР/М съвместимо и да бъде ползвано на други ПК.

Видеотракт

Стандартният видеотракт за ПК от разглежданата група включва монохроматичен дисплей (видеомонитор) и контролер, осигуряващи ин-

дицирането в текстов формат на 24 (25) реда от по 80 знака. При различните реализации мониторът е вграден в ПК или е изпълнен като отделен модул. Видеоконтролерът, разглеждан като част от видеотракта, включва минимум 2 Кбайта памет за поддържане на изображението върху екрана и знаков генератор, осигуряващ изобразяването на кодирания в един байт символ. Схемната реализация на контролера използва най-често интегралната схема MC6845 или неин функционален аналог (на пример SM607). В ПК от тази група съществува голямо разнообразие от принципно неразличаващи се схемни решения.

Видеотрактът е програмно достъпен за CP/M и за приложното програмно осигуряване през входната точка CONOUT на BIOS. В резултат на предаване на управлението на тази входна точка видеоконтролерът получава байт, който той трябва да обработи. Изпращаните към видеоконтролера байтове пренасят изобразими символи или команди, управляващи работата на видеомонитора. Изобразимите символи се извеждат върху екрана на място, определено от моментното разположение на курсора върху него. За правилната работа на приложната програма е необходимо видеоконтролерът с принадлежащата му драйверна програма да осигурява изпълнението на предвидените в нея управляващи команди. Последните достигат до него под формата на последователно предавани байтове с управляващи, а не изобразими символи.

В CP/M не е дефиниран набор от управляващи символи и команди, предавани чрез тях, а само начинът за обмен на символите. Използваните в разпространяваното приложно програмно осигуряване управляващи команди и символи могат да се разглеждат като развитие на кодовата таблица на кода ASCII. Създателите на програмно осигуряване се стремят да използват минимума на този набор, за да могат създаваните от тях програми да работят на различни типове компютри. Обратен, със същата цел производителите на ПК се стремят да разширяват до максимум този набор.

За илюстрация на типичните възможности на разглежданата група ПК по-долу е описан видеотрактът на ИЗОТ1031С. Този тракт осигурява индицирането на информация в текстов формат 24 реда по 80 знака. В нормален режим изобразимите символи са с кодове от 20 до 7E и от A0 до FE. Кодовете от 7F до 9F в такъв режим не се възприемат. Тяхното предаване не води до промяна на индицираната на монитора информация. Част от кодовете от 00 до 1F представляват команди или първи байт на команди.

Видеотрактът на ИЗОТ1031С изпълнява 13 еднобайтови и 20 двубайтови команди. Осигурена е възможност за работа в режим, при който еднобайтовите команди не се изпълняват и кодовете от 00 до 1F и от 80 до 9F могат да се използват за индициране на псевдографични символи.

Описание на еднобайтовите команди

- 03 — изтрива всички знаци от текущото положение на курсора до края на реда, включително и знака, сочен от курсора.
- 07 — извежда звуков сигнал чрез програмно управляемия говорител.
- 08 — установява курсора с една позиция наляво без изтриване на знака. Ако курсорът е бил в началото на реда, отива на края на горния ред. Ако курсорът е бил в началото на първия ред, командата не се изпълнява.

- 09 — хоризонтална табулация на 8 символа. Ако курсорът е на края на реда, отива на следващия ред. Ако е на края на екрана, не се мести.
- 0A — установява курсора на долния ред. Ако той е бил на последния ред на екрана, всички редове се повдигат с един ред нагоре, най-горният ред се губи, а отдолу се появява празен ред, на който е позициониран курсорът.
- 0B — установява курсора с един ред нагоре. Ако е бил на първия ред, командата не се изпълнява.
- 0C — изтрива целия екран и позиционира курсора в началото на екрана.
- 0D — установява курсора в началото на реда и изтрива всички знаци от позицията на курсора до края на реда. Ако курсорът е в началото на реда, командата не се изпълнява.
- 10 — установява курсора с една позиция надясно. Ако курсорът е бил на края на реда, отива в началото на следващия ред. Ако е бил на края на екрана, командата не се изпълнява.
- 11 — установява курсора с една позиция наляво и изтрива знака, сочен от курсора след преместването. Ако курсорът е бил в началото на реда, отива на края на горния ред. Ако е бил в началото на първия ред, командата не се изпълнява.
- 15 — изтрива всички знаци от текущото положение на курсора до края на екрана, включително и знака, сочен от курсора.
- 1C — установява курсора в началото на екрана.
- 1D — установява курсора в началото на реда.

Описание на двубайтовите команди

Кодът на първия байт на двубайтовите команди е 1B. Вторият байт съдържа кода на командата.

- 07 — изменя формата на екрана.
- 08 — вмъква шпация на позицията, сочена от курсора. Символите, включително соченият от курсора, се изместват с една позиция надясно. Изместването става до края на екрана. Символите, които са били на последна позиция от реда, отиват на първа позиция от следващия ред. Символът, който е бил на последна позиция на екрана, се губи.
- 09 — изтрива знака, сочен от курсора, и измества всички символи надясно от курсора с една позиция наляво. Изместването става до края на екрана, като първият символ от реда отива на последна позиция на горния ред. В последната позиция на екрана се записва шпация.
- 0B — установява курсора на долния ред. Ако е бил на последния ред, отива на първия.
- 0C — установява курсора на горния ред. Ако е бил на първия ред, отива на последния.
- 0D — изтрива реда, на който е курсорът, без да измести курсора.
- 0E — изтрива знаците от текущата позиция на курсора до началото на реда. Знакът, сочен от курсора, не се изтрива.
- 0F — изтрива всички знаци от текущата позиция на курсора до края на реда, включително и знака, сочен от курсора.
- 11 — позиционира курсора по зададени координати. След разпознаване на командата програмата очаква още 2 байта за координати на курсора. Първият байт е номер на колона, а вторият — номер на ред. Броенето на редовете и колоните започва от 0. Ако някоя от координатите е невалидна, командата не се изпълнява.
- 12 — мести курсора в началото на екрана.
- 13 — изтрива ред. Всички редове, които се намират под курсора, се изместват с един ред нагоре. Курсорът не се мести.
- 14 — записва знак по зададени координати. Описаното за команда „позициониране на курсора по зададени координати“ (1B,11) важи и за тази команда с допълнението, че след координатите трябва да се изпрати и кодът на знака.
- 15 — вмъква шпация на позицията, сочена от курсора, в рамките на един ред. Всички знаци надясно от курсора се изместват с една позиция надясно. Последният знак от реда се губи. Останалите редове остават непроменени.
- 16 — изтрива знак в рамките на един ред. Всички символи надясно от курсора се изместват с една позиция наляво, като в последната позиция на реда се записва шпация. Останалите редове остават непроменени.

- 17 — изменя вида на курсора.
- 18 — изтрива всички символи от текущата позиция на курсора до краъ на екрана, включително и знака, сочен от курсора.
- 1A — вмъква празен ред. Всички редове от текущата позиция на курсора се изместват с един ред надолу, като последният ред се губи. Редът, на който се намира курсорът, се запълва с шпации и курсорът се премества в началото на реда.
- 1C — изтрива целия екран и позиционира курсора в началото на екрана.
- 1B — в режима, в който не се изпълняват еднобайтовите команди (установен с командата 1B,1D), се извежда псевдографичният символ с код 1B.
- 1D — въвежда режим, при който еднобайтовите команди не се изпълняват. Този режим се използва за индициране на псевдографичните символи.
- 1E — отменя режима, при който не се изпълняват еднобайтовите команди.

В приложение 10 са показани символите на знаковия генератор на ПК ИЗОТ1031С.

Дисков тракт

Стандартният дисков тракт за ПК от разглежданата група включва два броя ЗУГМД и контролер за управление на четири дискови устройства. Обикновено целият тракт е вграден в ПК, като е изведен куплунг, към който могат да се включват допълнително още две дискови устройства. С изключение на първите модели на ПК, в тракта се използват 133-милиметрови (5,25-инчови) ЗУГМД с двустранен запис върху дискетата. ЗУГМД се използват в режим на запис и четене с двойна плътност, реализиран по метода на модифицираната честотна модулация (метод MFM). Дисковите устройства са свързани с контролера по т. нар. OEM или Shugart интерфейс. В табл.2.7 е дадено кратко описание на този интерфейс.

Таблица 2.7

Описание на сигналите на интерфейс OEM (Shugart)
за връзка със ЗУГМД — 133 мм (5 1/4 инча)

Означение	Наименование на сигнала	Направление на сигнала
<u>IND</u>	индекс/сектор	от ЗУГМД
<u>DSA</u>	избор на устройство 1	към ЗУГМД
<u>DSB</u>	избор на устройство 2	към ЗУГМД
<u>DSC</u>	избор на устройство 3	към ЗУГМД
<u>DSD</u>	избор на устройство 4	към ЗУГМД
<u>MD</u>	старт	към ЗУГМД
<u>DIR</u>	посока	към ЗУГМД
<u>STEP</u>	стъпка	към ЗУГМД
<u>WD</u>	данни за запис	към ЗУГМД
<u>WG</u>	запис	към ЗУГМД
<u>TROO</u>	пътечка 00	от ЗУГМД
<u>WPRT</u>	защита от запис	от ЗУГМД
<u>FDD</u>	данни за четене	от ЗУГМД

Масово разпространените ЗУГМД в тази група ПК са с плътност на пътечките 48 пътечки на инч при 40 пътечки на 133-милиметрова дискета. Пътечките носят номера от 0 до 39, като нулевата пътечка е с най-голям радиус. В последните модели ПК се срещат и 80-пътечкови устройства. Те работят с плътност на пътечките 96 пътечки на инч. Местоположението на нулева пътечка за двата типа устройства е еднакво.

При форматирането на дискетите на двустранни ЗУГМД четните пътечки се разполагат на едната страна (страна 0), а нечетните — на другата страна (страна 1) на дискетата. Независимо че СР/М обменя с дисковия тракт 128-байтови логически сектори, повечето ПК работят с 512-байтови физически сектори. В драйверните програми на дисковия контролер се извършва опаковане и разопаковане на логическите сектори при операциите запис и четене на сектор. Този алгоритъм на работа е оптимизиран за работа на програми с последователен достъп до секторите на файл, но води до значително забавяне на изпълнението на някои класове приложни програми, ползващи произволен достъп (например програми за бази от данни). Така се постига максимално използване на обема на ЗУГМД за сметка на бързодействието. В зависимост от броя на секторите, създавани върху една пътечка от форматиращата програма на конкретния ПК, обемът на полезната информация върху една дискета е различен. За двустранни устройства с 40 пътечки, работещи с двойна плътност и сектори от 512 байта, обемът на дискетата в зависимост от броя сектори на пътечка е, както следва: за 8 сектора — 320 Кбайта, за 9 сектора — 360 Кбайта и за 10 сектора — 400 Кбайта.

Схемните решения на контролерите използват най-често интегралните схеми от серията FD1790 на фирмата Western Digital (САЩ) или 18272 (както и техни аналози).

На фиг.2.11 е показана блоковата схема на интегралната схема КР1818ВГ93 (СССР) — контролер за ЗУГМД, функционален аналог на FD1793. Контролерът съдържа следните блокове:

— изместващ регистър — в него се формират в байтове данните, четени от диска, и се преобразуват в сериен вид данните за запис върху диска;

— регистър за данни — през този регистър процесорът на компютъра чете и записва данните, обменяни със ЗУГМД;

— регистър за пътечка — съдържа номера на пътеката, върху която е позиционирана главата за запис—четене на ЗУГМД;

— регистър за сектор — съдържа номера на търсения сектор;

— команден регистър — съдържа изпълняваната в момента коман-

да;

— регистър на състоянието — съдържа информация за резултата от изпълнението на командата;

— логика за CRC — предназначена е за генериране и за проверка на 16-битовия цикличен контролен код;

— аритметично-логическо устройство — използва се за модифициране и сравняване на съдържанието на регистрите;

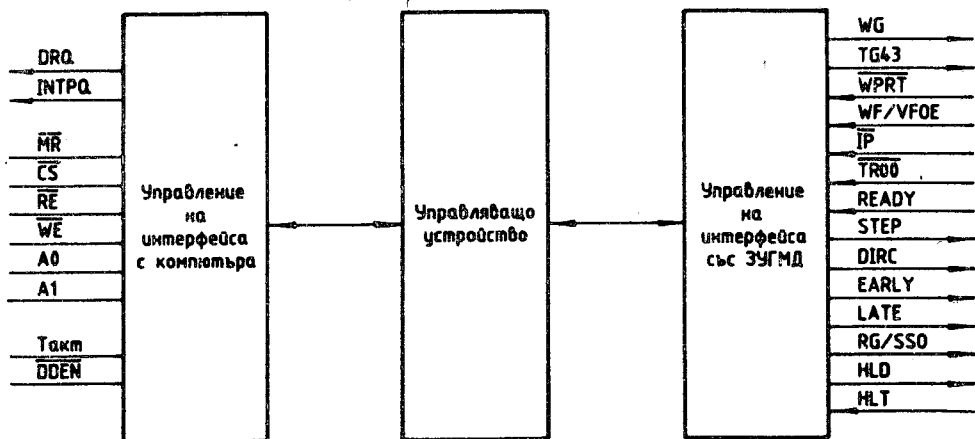
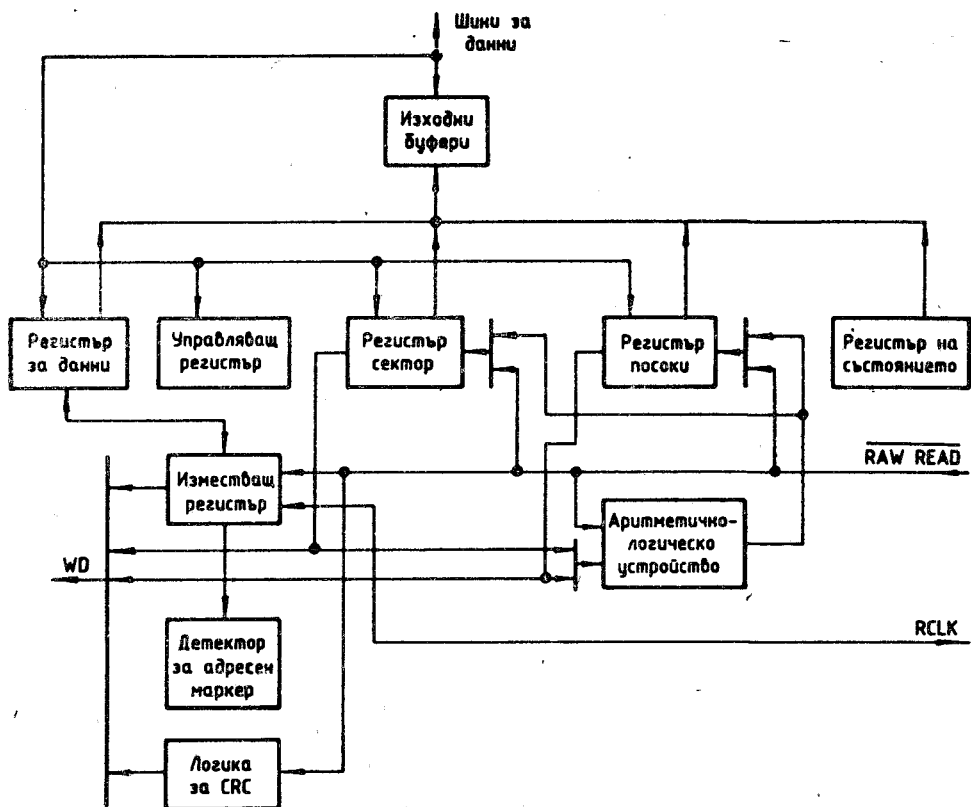
— управляващо устройство — изработва всички сигнали на интерфейса с процесора и на интерфейса със ЗУГМД;

— детектор за адресен маркер — блок за разпознаване на адресни маркери.

Връзката с процесора се осъществява чрез 8-битовата шина за данни и чрез управляващи сигнали.

Контролерът разпознава и изпълнява следните команди: възстановяване, търсене, стъпка, стъпка навътре, стъпка навън, четене на сектор, запис на сектор, четене на адрес, четене на номер на пътечка, форматиране на пътечка, предизвикване на прекъсване.

Командата се зарежда от процесора в командния регистър. По време на изпълнението на командата се поддържа вдигнат флаг в регистъра



Фиг. 2.11. Блокова схема на контролер на ЗУГМД КР1818ВГ93 (FD1790)

на състоянието. След завършване на изпълнението флагът се сваля и се генерира прекъсване към процесора.

Скоростта на обмен между контролера и дисковото устройство при двойна плътност на записа е 250 Кбайта/s или 32 μ s на байт. Това време е достатъчно за обслужване на контролера по програмен път от микропроцесора на ПК. Поради това в повечето ПК се използват схемни решения на контролери, извършващи обмена между диска и паметта на ПК по прекъсване и чрез програма, обслужваща прекъсването. Твърде рядко се срещат контролери, използващи за този обмен директен достъп до паметта на ПК.

Независимо че CP/M 2.2 осигурява поддържането на дискови устройства с обем 8 Мбайта, включването на твърд диск в дисковите трактове на тази група ПК е рядкост. Това се обяснява с периода на активна разработка на тези ПК (1980—1982 г.) и със следващото стесняване на кръга на тяхното приложение, предизвикано от появата на 16-разредните ПК.

Клавиатура

Стандартната клавиатура за разглежданите ПК включва стандартизиран буквено-цифров блок и допълнителен цифров блок. Клавиатурата най-често е оформена в отделен подвижен модул. Често се срещат и конструктивни реализации, при които тя е вградена в основната конструкция на ПК. В модула клавиатура освен клавишното поле почти винаги е включено схемно решение, осигуряващо управлението на клавишното поле. Реализират се схемни решения, изискващи матрично сканиране на клавишите. В схемните решения се включват интегрални схеми с голяма степен на интеграция, предназначени за управление на клавиатури от този тип. По-рядко се използват схемни решения, в които сканирането на матрицата на клавиатурата е възложено на процесора на ПК. Тези решения са по-прости, но са свързани с подпрограми, забавящи работата на ПК при изпълнение на приложните програми.

Поради това, че за работата на приложните програми в CP/M среда не са дефинирани кодове за специални клавиши с определено функционално предназначение или свободно програмируеми, върху разглежданите клавиатури рядко се разполагат такива клавиши.

Клавиатурата е достъпна за програмиста чрез системните функции на CP/M и през входните точки на BIOS — CONST и CONIN.

Въвеждането на възможността за работа на клавиатурата (и на целия ПК) на латиница и кирилица с големи и малки букви е свързано с модификация на оригиналната версия на CP/M 2.2 и със създаването на нейни функционални аналози. В произвеждания у нас ИЗОТ1031С с ДОС UMCO е осигурена работа по избор на оператора в няколко кодови таблици:

- КОИ-8 — осигурява кирилица и латиница главни и малки букви;
- КОИ-7 Н2 — осигурява кирилица и латиница главни букви;
- КОИ-7 Н0 — осигурява латиница главни и малки букви (съответства на ASCII);
- КОИ-7 Н1 — осигурява кирилица главни и малки букви.

Най-голямо приложение намира режимът на работа в КОИ-7 Н2. Този режим осигурява работата на всички оригинални програми, раз-

пространявани за СР/М 2.2 при ползването на главни букви от латиницата и кирилицата. За удобната експлоатация на оригиналните програми е необходима тяхната преработка, свеждаща се до промяна на текста на съобщенията. Този текст трябва да съдържа само главни латински букви.

В табл.2.8 са дадени изобразимите символи от кода КОИ-7 Н2. Тази кодова таблица се ползва и от единната серия малки машини СМ.

Таблица 2.8

Изобразими символи от кода КОИ-7 Н2

	0	1	2	3	4	5	6	7
0				0	@	P	Ю	П
1			!	1	A	Q	А	Я
2			..	2	B	R	Б	Р
3			#	3	C	S	Ц	С
4			\$	4	D	T	Д	Т
5			%	5	E	U	Е	У
6			&	6	F	V	Ф	Ж
7			'	7	G	W	Г	В
8			(8	H	X	Х	Ь
9)	9	I	Y	И	Ы
A			*	:	J	Z	Й	З
B			+	;	K	[К	Э
C			.	<	L	\	Л	Ш
D			—	=	M]	М	Щ
E			.	>	N	^	Н	Ч
F			/	?	O	—	О	

Печатащо устройство

Печатащото устройство на разглежданата група ПК работи по правило само в текстов режим. Използват се както знаковсинтезиращи (мозайкови) печатащи устройства, така и знакоизписващи с печатащ механизъм от тип *маргаритка*. Скоростта на печат на последните за отделните типове се движи между 30 и 45 знака в секунда. Преобладаващият тип интерфейс за връзка между ПК и печатащото устройство е тип Centronics (известен още като паралелен OEM интерфейс за печатащи устройства или ИРПР-М).

В табл.2.9 е дадено описание на сигналите на този интерфейс. Стандартното им изпълнение е с 36-изводен куплунг, известен като тип AmphenoI. В социалистическите страни този интерфейс най-често се изпълнява с 37-изводен куплунг, известен като тип Canon.

Схемното решение на контролера за печатащо устройство обикновено включва стандартна интегрална схема за паралелен интерфейс. Най-разпространено е използването на схемата I8255 или нейни функционални аналози. Контролерът, управляван от микропроцесора на ПК, се разполага върху системната платка или се изпълнява в отделен модул заедно с контролера за сериен интерфейс. Драйверните програми на контролера са предвидени за работа с неинтелигентно печатащо устройство (несъдържащо микропроцесор) за съвсем ограничен брой стан-

дартни управляващи кодове, като LF (line feed) — нов ред, CR (carriage return) — връщане на каретката. Поради това тези програми реализират сами изпълнението на управляващите кодове от рода на FF (form feed) — нова страница, HT (horizontal tabulation) — хоризонтална табулация и т.н. Така реализираните драйверни програми осигуряват връзка на ПК с разнообразни печатащи устройства.

Печатащото устройство е достъпно за програмиста чрез системните функции на CP/M и през входните точки на BIOS — LIST и LISTST.

Таблица 2.9

Описание на сигналите на паралелен интерфейс от тип Centronics за връзка с печатащо устройство (ПУ)

Извод	Наименование на сигнала	Направление на сигнала
1	строб за данните	STROBE към ПУ
2	данни 1	DATA 1 към ПУ
3	данни 2	DATA 2 към ПУ
4	данни 3	DATA 3 към ПУ
5	данни 4	DATA 4 към ПУ
6	данни 5	DATA 5 към ПУ
7	данни 6	DATA 6 към ПУ
8	данни 7	DATA 7 към ПУ
9	данни 8	DATA 8 към ПУ
13	ПУ неготово	SLCT от ПУ
14	автоматичен преход на нов ред	AUTO FEED към ПУ
15	—	
16		GND
17		GND
18	—	
19		GND
20		GND
21		GND
22		GND
23		GND
24		GND
25		GND
26		GND
27		GND
28		GND
29		GND
30		GND
31	начално установяване	INIT към ПУ
32	грешка	ERROR от ПУ
33	земя	GND
34	—	
35	—	
36	избор на ПУ	SLCTIN към ПУ
37	—	

Комуникационен тракт

За разглежданата група ПК комуникационният тракт се свежда до наличие на контролер за асинхронна старт-стопна комуникация по интерфейс RS232C (Стък-2). За някои ПК този интерфейс е разширен с възможност за връзка по токов кръг (20 или 40 mA), осигуряваща галва-

нично развързване между комуникаращите устройства. В табл.2.10 е дадено описанието на едно типично изпълнение на интерфейса за серийна комуникация, използвано в ИЗОТ1031С. Върху стандартния 25-изведен куплунг за интерфейс RS232C са изведени и сигналите за връзка по токов кръг (интерфейс ИРПС).

Таблица 2.10

Описание на сигналите на последователния интерфейс на ПК ИЗОТ1031С

Извод	Верига N	Означение	RS232C (Стък-2)		Токов кръг (ИРПС)	
			Наименование на сигнала	Означе- ние	Наименова- ние на сиг- нала	
1	101		Защитна земя (Protective Ground)	—	—	
2	103	TxD	Предавани данни (Transmitted Data)	—	—	
3	104	RxD	Приемани данни (Received Data)	—	—	
4	105	RTS	Заявка за предаване от ПК към модема (Request to Send)	—	—	
5	106	CTS	Сигнал към ПК, че приемникът е готов (Clear to Send)	—	—	
6	107	DSR	Модемът е готов за предаване (Data Set Ready)	—	—	
7	102		Сигнална земя (Signal Ground)	—	—	
8	109	DCD	Наличие на носещата честота (Receiver Line Signal Detector)	—	—	
9	—	—	—	—	—	
10	—	—	—	—	—	
11	126	STF	Избор на честотата на предавателя (Select Transmit Frequency)	—	—	
12	—	—	—	+T	—	предавател
13	—	—	—	-T	—	предавател
14	—	—	—	—	—	
15	114	TxC	Такт за предаване (Transmitter Signal Element Timing)	—	—	
16	—	—	—	-R	—	приемник
17	115	RxC	Такт за приемане (Receiver Signal Element Timing)	—	—	
18	141	LLB	Управление на обратната връзка (Local Loopback)	—	—	
19	—	—	—	+R	—	приемник
20	108.2	DTR	Сигнал към модема за свързване (Data Terminal Ready)	—	—	
21	—	—	—	—	—	
22	125	RIR	Повикване (Ring Indicator)	—	—	
23	111	DSRS	Избор на скорост (Data Signal Rate Selector)	—	—	
24	—	—	—	—	—	
25	142	TIR	Сигнал за тестване (Test Indicator)	—	—	

Стандартно решение за този тракт в разглежданата група ПК е възможността за програмиране на скоростта за обмен. С помощни установяващи програми тази скорост обикновено може да се изменя в обхвата от 300 до 4800 бода. Подразбиращата се стойност за отделните ПК е 300 или 1200 бода.

Схемното решение на контролера включва стандартна интегрална схема за асинхронна старт-стопна комуникация в режим на пълен дуплекс. Най-разпространено е използването на схема 18251, CM603 или тех-

ни функционални аналози. Комуникационният контролер обикновено се разполага на отделен модул заедно с контролера за връзка с печатащо устройство.

Достъпът на програмиста до комуникационния интерфейс е чрез системните функции на CP/M и през входните точки на BIOS — **READER** и **PUNCH**. Връзката на ПК с друго устройство през модем за асинхронна комуникация изисква използването на специализирана програма. Реализацията на тази програма налага директното управление на комуникационния контролер без използване на програмите на BIOS. Това изисква детайлното познаване на архитектурата на компютъра. Програмата е ориентирана към съответния компютър и не може да бъде ползвана на друг ПК.

2.2.3. ПРОГРАМНО ОСИГУРЯВАНЕ

Програмното осигуряване на ПК, използващи CP/M, може да се раздели на няколко групи:

Асемблери, настройващи програми и реасемблери. Повечето са за микропроцесорите 18080, 18085 и Z80. Съществуват и кросасемблери за други процесори.

Езици от високо ниво (интерпретатори и компилатори) — БЕЙСИК, ПАСКАЛ, ФОРТРАН и др.

Текстообработващи програми. Най-широко разпространение е получила Wordstar и нейните функционални аналози;

База от данни. Най-широко разпространение с тези компютри са получили пакетите програми Infostar и dBASE II;

Специализирани пакети за математически изчисления (Mumath), за статистически изчисления (Statpac) и други;

Пакети за специализирани работни места, като счетоводство, кадри и др.

Електронни таблици (Supercalc, Multiplan) и др.

В някои от пакетите програми за специализирани работни места се използват инструкции, съществуващи само за микропроцесора Z80. За тях изрично е отбелязано, че могат да се използват само на ПК с този микропроцесор.

При доставката на пакетите програми трябва да се отчита формата на дискетите на ПК. Фирмите обикновено отбелязват в каталозите си форматите за най-популярните компютри от тази група. В тези формати по избор те доставят желаните пакети.

ЛИТЕРАТУРА КЪМ ГЛ. 2

1. Аладжем, М., П.Аладжем. Паскал за персонални компютри. С., Техника, 1986.
2. Ангелов, А., П.Петров. Микропроцесорът — сърцето на микрокомпютъра. С., Техника, 1986.
3. Балашов, Е.П., Д.В.Пузанков. Микропроцесоры и микропроцессорные системы. Москва, Радио и связь, 1987.
4. Вачков, П., Х.Христов. Как работи Правец-82. С., Техника, 1985.

5. Евтимов, Т., А.Обайски, Г.Тошков. 50 програми за персонален компютър. С., Техника, 1986.
6. ИЗОТ 1031С — Базова система за програмиране. Описание на езика. ЦИИТТ—София, 1.А0А1.00798-01 35.
7. Погорелый, С.Д., Т.Ф.Слободняк. Программное обеспечение микропроцессорных систем. Киев, Техника, 1985.
8. ПРАВЕЦ 8М. Руководство по работе с персональным компьютером. КМПТ—Правец.
9. Ръководство за работа с персонален компютър ПРАВЕЦ-8. БЕЙСИК/ДОС. КМПТ—Правец.
10. Соучек, Б. Микропроцессоры и микро-ЭВМ, Москва, Советское радио, 1979.
11. Шишков, А., И.Марангозов. Работа с персонален компютър. С., Техника, 1985.
12. Шишков, А., Т.Бояджиева. БЕЙСИК — език на персоналните компютри. С., Техника, 1986.
13. Assembler Language Programming Manual. Zilog Inc., 1977.
14. CP/M 2.2 — Interface Guide. Digital Research, 1979.
15. Z80 Microcomputer Devices. Technical Manual. Mostek, 1980.

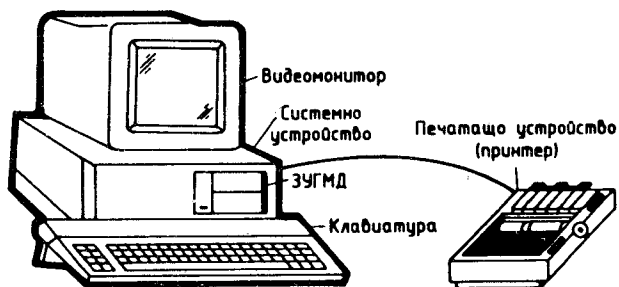
ШЕСТНАДЕСЕТРАЗРЕДНИ ПЕРСОНАЛНИ КОМПЮТРИ

3.1. АРХИТЕКТУРА НА ШЕСТНАДЕСЕТРАЗРЕДНИТЕ ПЕРСОНАЛНИ КОМПЮТРИ

Истинското навлизане на шестнадесетразредните ПК в практиката на програмистите и потребителите стана с появата на персоналните компютри от фамилията IBM PC, IBM PC/XT на фирмата IBM. Те са разработени на базата на 16-разредния микропроцесор 18088 на Intel. Тези компютри се налагат бързо на пазара и редица фирми започват производството на ПК, съвместими с IBM PC/XT. В момента над 80% от персоналните компютри в света са от този клас. За тях са разработени голямо количество пакети приложни програми за всички области на човешкото познание и ежедневната практическа дейност.

У нас са разработени няколко компютъра, съвместими с фамилията персонални компютри на фирмата IBM. От тях най-широко разпространен е Правец-16, който е функционално и конструктивно съвместим с IBM PC/XT. Конфигурацията на Правец-16 (ЕС1839) е показана на фиг.3.1. Тя включва системно устройство, клавиатура, видеомонитор и печатащо устройство (принтер). Към тази конфигурация могат да се включат и други видове периферни устройства.

Основната част на системата — системното устройство, се състои от централен процесор, памет, запомнящи устройства с магнитен диск и няколко функционални модула, като всички тези компоненти са разположени в един конструктивен блок.



Фиг.3.1. Базова конфигурация на персоналния компютър Правец-16

На фиг.3.2 е показано примерно разположение на отделните съставни части в системното устройство. Процесорният модул е реализиран върху една основна платка-майка, наречена *системна платка*, към която чрез куплунги се монтират платките на функционалните модули. На фиг.3.3 е показана максимално пълна конфигурация на професионален компютър, която демонстрира основните модули и възможностите за

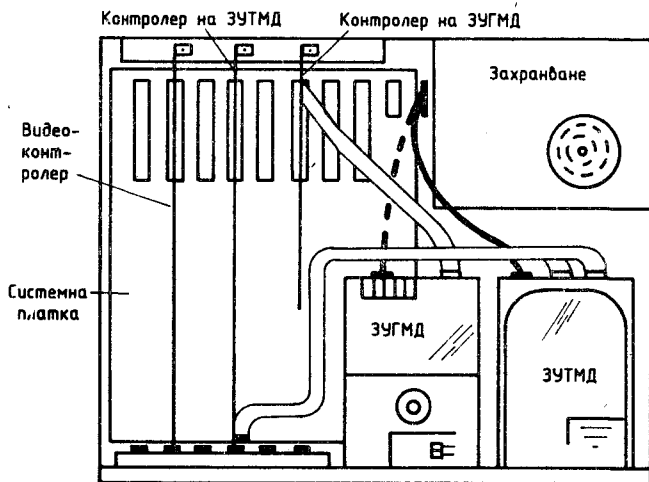
разширение на функциите на компютъра.

Върху системната платка са разположени 8 гнездови куплунга за включване на 8 допълнителни печатни платки. Това са платките на функционалните модули за управление на запомнящите устройства с гъвкав магнитен диск (ЗУГМД), запомнящите устройства с твърд магнитен диск (ЗУТМД), печатащото устройство, видеомонитора и др. Освен това с тези платки се разширяват някои от функционалните възможности на компютъра: увеличаване на оперативната памет, включване на допълнителни периферни устройства, като плотери, дигитайзери и т.н.

Захранващият блок преобразува променливото напрежение на мрежата в няколко постоянни напрежения, необходими за функционирането на отделните модули на системното устройство: +5 V, -5 V, +12 V и -12 V. Обикновено общата консумирана мощност не надвишава 130 W.

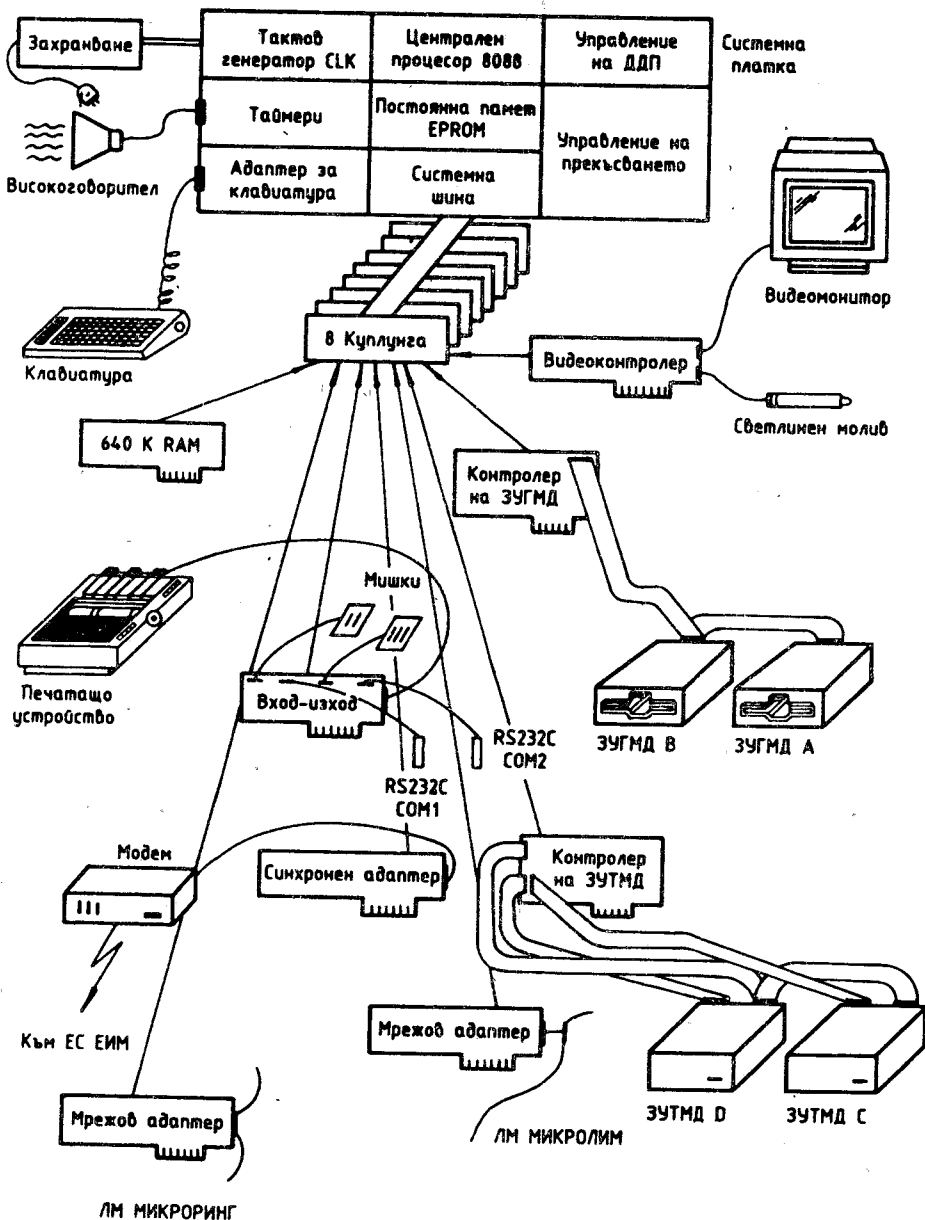
Като външна памет на персоналните компютри се използват два вида магнитни дискове: сменяеми гъвкави магнитни дискове (дискети) в ЗУГМД и херметично затворени несменяеми твърди магнитни дискове в ЗУТМД. В едно системно устройство могат да бъдат разположени няколко ЗУГМД и ЗУТМД. В някои типове персонални компютри като външна памет се използват и магнитни ленти. Те са удобни при създаване на архиви, когато е нужно да се запази за по-дълго време информацията, записана върху ЗУТМД, или при прехвърлянето ѝ върху ЗУТМД на друг компютър.

В системното устройство е вграден малък високоговорител. Той произвежда звуков сигнал, който привлича вниманието на оператора към някои извънредни ситуации, настъпили в програмата или в отделните устройства на компютъра.



Фиг.3.2. Системно устройство на Правец-16

Видеомониторите за шестнадесетразредните ПК са два основни вида — за черно-бяло изображение (монохроматични) и за цветно изображение (цветни). Повечето от тях предлагат графични възможности, които се определят от съответния контролер. Затова е правилно да се говори



Фиг.3.3. Максимална конфигурация на персоналния компютър Павец-16

за тракт на видеомонитора. Когато контролерът работи с графика с по-голяма разделителна способност (например 640×400 точки — т.нар. професионална графика), към него се включват видеомонитори с подобрени възможности и нови функционални характеристики. Видеомо-

ниторът има самостоятелно захранване и се свързва към контролера с гъвкав кабел.

Клавиатурите трябва да предоставят на потребителя максимално удобство при работа. Ето защо буквено-цифровото им поле е гравирано със знаците на латиница и на кирилица. Разположението им съответства на общоприетите стандарти за пишещи машини. Определени клавиши се дефинират програмно за лесно превключване от един към друг стандарт. Клавиатурата се свързва към централното устройство с гъвкав кабел, по който получава захранващо напрежение.

Шестнадесетразредните ПК използват голямо разнообразие от печатащи устройства — от най-простите с 80 знака в ред с ниска скорост на печат до високоскоростните със 132 знака в ред, снабдени с шрифтове с високо качество. В последно време се предлагат лазерни печатащи устройства, които по характеристики се доближават до типографско качество на печата. Печатащото устройство има самостоятелно захранване и се свързва към контролера през стандартизиран интерфейс чрез гъвкав кабел.

В наши дни се разработват все по-нови и съвършени контролери, адаптери и периферни устройства: плотери, дигитайзери, външни за помнящи устройства с магнитни дискове и магнитни ленти и т.н.

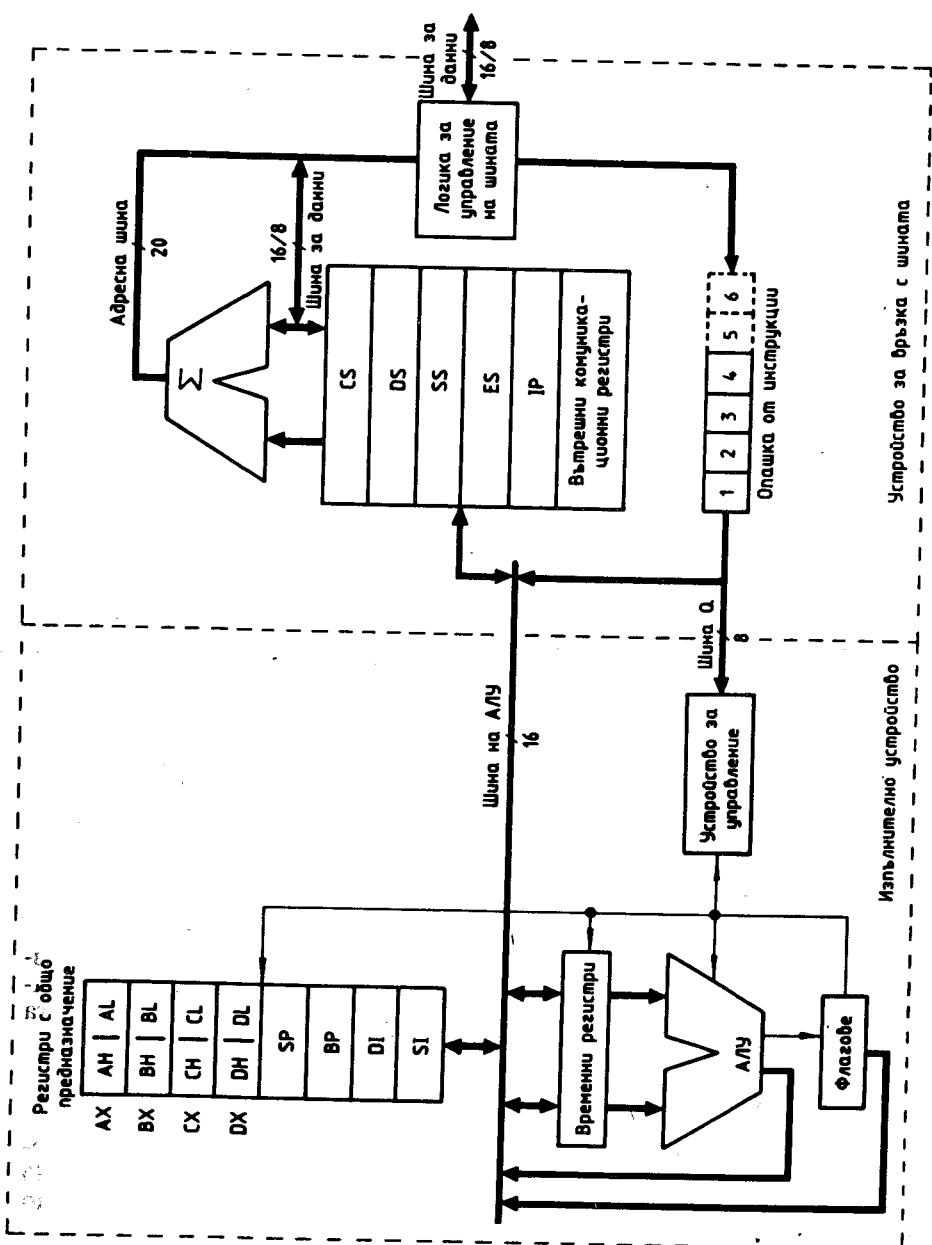
3.2. ПРОЦЕСОРЕН МОДУЛ

3.2.1. МИКРОПРОЦЕСОРИ I8086 И I8088

В системната платка на компютъра е използвана микропроцесорната фамилия I8086. Освен микропроцесорите с общо предназначение I8086 и I8088, фамилията включва група специализирани елементи, проектирани за съвместна работа, между които са разпределени системните функции: генериране на синхронизиращи и времезадаващи сигнали, управление на операциите по системната шина, управление на директния достъп до паметта, управление на прекъсванията и т.н.

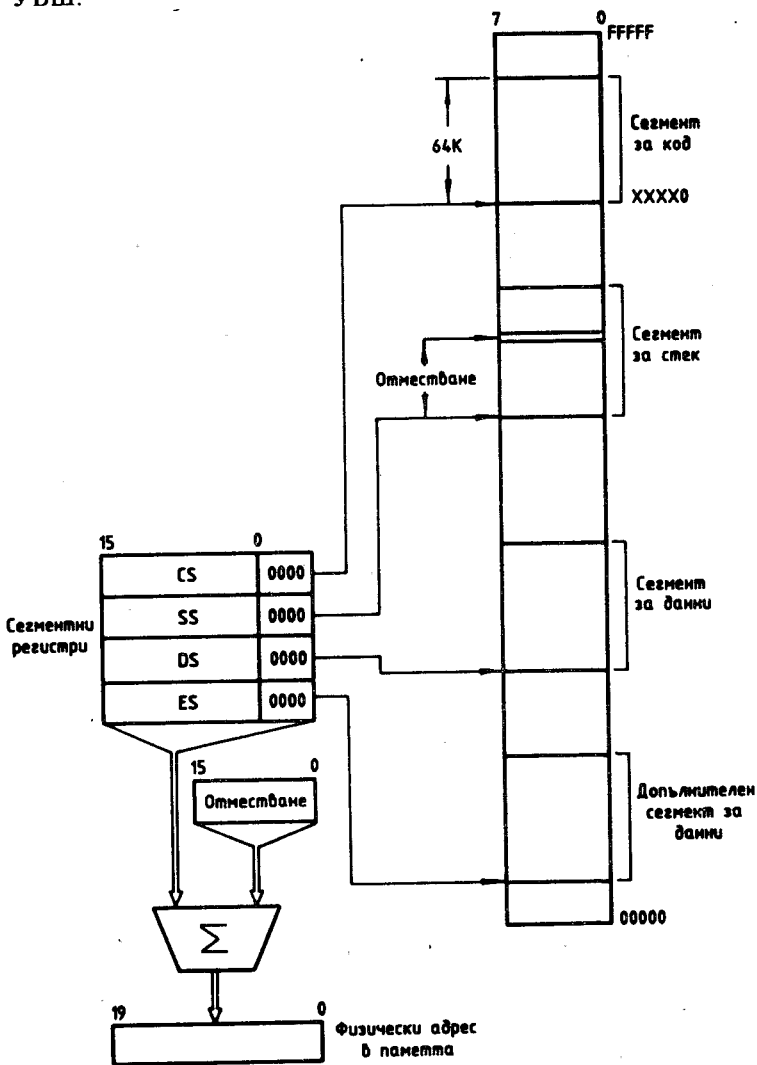
Микропроцесорите I8086 и I8088 са програмно съвместими и се различават главно по броя на външните си линии за обмен на данни. Микропроцесорът I8086 може да обменя в един цикъл по 16 или по 8 бита данни с останалите устройства в системата, а I8088 — само по 8, което определя и по-малкото му бързодействие. Вътрешната структура и на двата микропроцесора е с 16 линии за данни и е почти идентична. I8086 и I8088 могат да обработват както 8-разредни, така и 16-разредни операнди. Броят на адресните линии на микропроцесорите е 20, което позволява използването на адресно пространство за памет до 1 Мбайт. I8086 и I8088 разполагат с отделно входно-изходно адресно пространство до 64 Кбайта. Обръщанията към адресните пространства за памет и за вход-изход се определят от изпълняваните инструкции.

Микропроцесорите I8086 и I8088 се класифицират като микропроцесори от трето поколение, при които извличането на инструкциите и изпълнението им се припокриват по време. Това става чрез едновременно действие на две отделни вътрешни устройства (фиг.3.4): *изпълнително*



Фиг.3.4. Вършна структура на микропроцесора 18086/8088

устройство (ИУ) и устройство за връзка с шината (УВШ). ИУ изпълнява инструкциите, поддържа управляващи флагове и флагове за състояние на микропроцесора и предава данни и адреси към УВШ. Структурата на ИУ, включваща регистри с общо предназначение, регистри за временно съхраняване на информация и аритметично-логическо устройство (АЛУ), е 16-разредна. С изключение на някои управляващи линии ИУ няма връзка с външната шина на микропроцесора. Извличането на инструкциите, четенето на операндите и записът на резултатите се осъществява от УВШ.



Фиг.3.5. Формиране на физическия адрес на паметта

Всички адресни компоненти, които се обработват от изпълнително устройство, са 16-разредни. Те определят отместването на даден адрес спрямо началото на определен сегмент (с максимален обем 64 Кбайта) от адресното пространство на микропроцесора. Чрез сегментните регистри на УВШ — CS, DS, SS и ES, микропроцесорът поддържа във всеки момент четири сегмента на паметта. Началните адреси на тези сегменти са кратни на 16, т.е. в младшата част на двоичния им запис се съдържат четири нули. Старшата част на адреса на сегментите се задава от съдържанието на сегментните регистри, които също са 16-разредни.

Устройството за връзка с шината има собствено АЛУ, чрез което комбинира началния адрес на даден сегмент и отместването и получава 20-разреден адрес, осигуряващ достъп до пълното адресно пространство памет на микропроцесора. Формирането на адреса (фиг.3.5) се извършва, като към изместеното с четири разряда наляво съдържание на даден сегментен регистър се прибавя отместването.

Изпълнителното устройство и устройството за връзка с шината работят независимо едно от друго. Когато ИУ е заето с изпълнение на инструкции, УВШ извлича предварително следващи инструкции от паметта и ги записва във вътрешен буфер — *опашка за инструкции*. От там инструкциите се предават за изпълнение в ИУ. Когато дадена инструкция изисква достъп до системната или входно-изходната памет, ИУ издава заявка до УВШ за четене или за запис на съответните данни. Буферът за опашката от инструкции на микропроцесора I8086 е с обем 6 байта, а на I8088 — 4 байта. Друга съществена разлика между двата микропроцесора е разредността на външната им шина за данни: 16 линии за данни при микропроцесора I8086 и 8 — при I8088.

Вътрешни регистри на микропроцесора I8086/8088

Регистри с общо предназначение. Микропроцесорът I8086/8088 съдържа осем 16-разредни регистъра с общо предназначение: AX, BX, CX, DX, SP, BP, SI и DI. Първите четири от тях се наричат регистри за данни, а вторите — указателни и индексни регистри.

Всеки от регистрите за данни може да се използва като 16-разреден регистър или като два отделни 8-разредни регистъра (фиг.3.4), т.е. старшите и младшите части на регистрите AX, BX, CX и DX могат да се адресират поотделно. Указателните и индексните регистри могат да се използват само като 16-разредни регистри.

Регистрите с общо предназначение намират приложение при повечето аритметични и логически операции. Освен това някои инструкции използват строго определени регистри, без те да са специфицирани явно (табл.3.1), с което се постига по-компактно кодиране.

Сегментни регистри. Във всеки момент микропроцесорът има възможност за пряк достъп до четири сегмента на паметта. Техните базови (начални) адреси се съхраняват в 16-разредните регистри CS, SS, DS и ES.

Регистърът CS съдържа началния адрес на сегмента за код, където е разположена програмата за изпълнение. Регистърът SS посочва началото на сегмент за изграждане на стек, регистърът DS — началото на сегмент за данни, регистърът ES — началото на допълнителен сегмент за данни.

Регистри, специфицирани неявно от инструкциите

Регистър	Операции
AX	Умножение на думи, деление на думи, входно-изходни операции с думи
AL	Умножение на байтове, деление на байтове, входно-изходни операции с байтове
AH	Умножение на байтове, деление на байтове
BX	Преобразуване при работа с таблици
CX	Операции с низове, цикли
CL	Изместване, ротация
DX	Умножение на думи, деление на думи, входно-изходни операции — косвена адресация
SP	Операции със стек
SI	Операции с низове
DI	Операции с низове

Брояч на инструкции. Регистърът брояч на инструкции IP е 16-разреден. Той се обновява от устройството за връзка с шината (УВШ) и съдържа отместването на следващата инструкция спрямо началото на текущия кодов сегмент. При нормално изпълнение на дадена програма IP съдържа отместването на следващата инструкция, която УВШ трябва да извлече от паметта. Преди записване на съдържанието на IP в стека то се коригира автоматично за посочване на следващата инструкция, която трябва да се изпълни. Директният достъп до съдържанието на IP по програмен път е невъзможен.

Регистър за флагове. В регистъра за флаговете (регистър за състояние) са групирани флагове за състояние и флагове за управление на микропроцесора по следния начин:

	Битове на регистъра за флаговете															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Флагове	—	—	—	—	OF	DF	IF	TF	SF	ZF	—	AF	—	PF	—	CF

Флаговете за състояние са CF, PF, AF, ZF, SF и OF. Те се установяват от изпълнителното устройство и отразяват определени признаци на резултата от аритметичните и логическите операции. Наборът от инструкции на микропроцесора включва група инструкции, чрез които може да се променя изпълнението на дадена програма в зависимост от текущото състояние на тези флагове.

Флаговете за състояние се установяват или нулират от различните инструкции по различен начин, но най-общо те отразяват посочените по-долу условия.

Флагът за пренос CF отразява наличието или липсата на пренос от старшия бит на резултата (байт или дума).

Флагът за четност PF се установява в 1, ако броят на единиците в резултата е четно число.

Флагът за допълнителен пренос AF отразява наличието или липсата на пренос от младшата тетрада на осембитов резултат или от младшия байт на шестнадесетбитов резултат.

Флагът за нулев резултат ZF се установява в 1, ако резултатът от операцията е нула.

Флагът за знак SF отразява знака на резултата, т.е. най-старшият бит на резултата (байт или дума).

Флагът за препълване OF отразява наличието или липсата на препълване при изпълнение на аритметичните операции.

Флаговете за управление TF , IF и DF се установяват или нулират програмно и се използват, за да се измени изпълнението на операциите на микропроцесора.

Чрез *флага за трасиране* TF микропроцесорът се установява в режим на постъпково изпълнение на програмата. В този режим след изпълнението на всяка инструкция микропроцесорът извършва автоматично вътрешно прекъсване, което позволява проследяване и настройка на програмата.

Чрез установяване на *флага за разрешаване на прекъсванията* IF се дава възможност на микропроцесора да приема заявки за външни (апаратни) прекъсвания по линията $INTR$. Този флаг не влияе на немаскируемите и на вътрешните прекъсвания на микропроцесора.

Флагът за посока DF определя посоката при обработване на низове: при $DF=0$ посоката е „положителна“, т.е. от младшите към старшите адреси на операндите; при $DF=1$ обработката се извършва в обратна посока.

Сигнали на микропроцесора I8086/8088

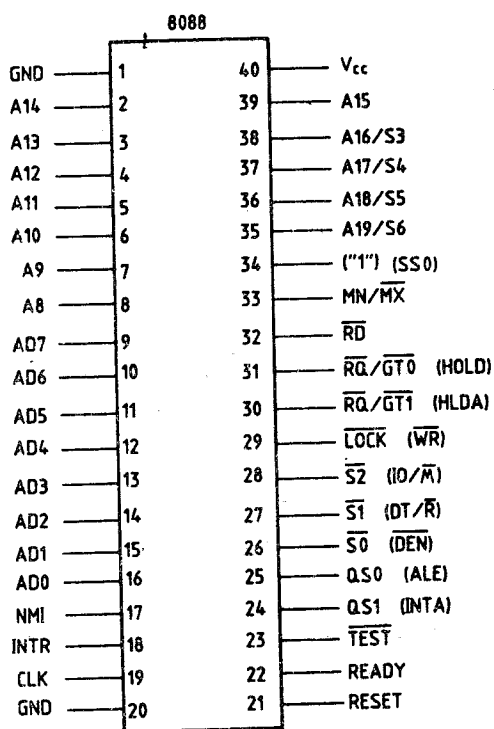
С цел да се намали броят на изводите на микропроцесорите I8086 и I8088, адресните линии и линиите за данни се мултиплексират по време в самите микропроцесори, т.е. най-напред се предава адресът, след което по същите физически линии се пренасят данните.

Микропроцесорите I8086 и I8088 могат да работят в два режима: *режим на максимална и режим на минимална конфигурация*. Режимът се определя от приложения потенциал на извода MN/\overline{MX} на микропроцесора: в минимален режим изводът MN/\overline{MX} се свързва към $+5\text{ V}$, а в максимален — към маса. В режим на минимална конфигурация (еднопроцесорна система) микропроцесорът осигурява самостоятелно всички необходими управляващи сигнали за обмен на информация с паметта и устройствата, разположени във входно-изходното адресно пространство. На фиг. 3.6 означенията на тези сигнали са дадени в скоби. В режим на максимална конфигурация същите сигнали се изработват от контролера на шината I8288, а по освободените от тази функция линии на микропроцесора се осигурява управлението на I8288 и възможност за съгласуване работата на няколко микропроцесора. В персоналните микрокомпютри се използва аритметичен копроцесор I8087, позволяващ разширение на набора от инструкции на главния микропроцесор и увеличаване на бързодействието.

По-долу са описани сигналите по линиите (изводите) на микропроцесора I8088 в режим на максимална конфигурация, тъй като този режим е намерил приложение в разглежданите компютри. След означението на даден сигнал в скоби е показано дали сигналът е входен (В), изходен (И) или входно-изходен (В/И) за микропроцесора.

Линиите за адреси и данни $AD7-AD0$ (В/И) на микропроцесора са мултиплексирани по време, като първоначално се предава най-младшата част на адреса за паметта или входно-изходните устройства, а по-късно

се пренасят данните. Сигналите по тези линии са с активно високо ниво. По време на възприемане на прекъсване и в неактивно състояние на микропроцесора изходите AD7-AD0 са в състояние на висок импеданс.



Фиг. 3.6. Разположение на изходите на микропроцесора 18088

памятта и използването на съответните сегментни регистри, както следва:

S4	S3	Сегмент	Регистър
0	0	данни (допълнителен)	ES
0	1	стек	SS
1	0	код или липса на обръщение	CS
1	1	данни	DS

Битът S5 отразява състоянието на флага-маска за разрешение на прекъсванията. Сигналят по линията S6 е винаги с ниско ниво при работа на микропроцесора. При неактивно състояние на микропроцесора изходите A19/S6-A16/S3 са в състояние на висок импеданс.

Сигналят по линията за четене \overline{RD} (И) показва, че микропроцесорът изпълнява цикъл за четене от паметта или от входно-изходните устройства, в зависимост от състоянието на сигнала по линията S2, респ. линията IO/M в минимален режим. Сигналят \overline{RD} е с активно ниско ниво и управлява четенето от устройства, включени към вътрешната шина на микропроцесора AD7-AD0. В неактивно състояние на микропроцесора този изход е в състояние на висок импеданс.

По линиите за адреси A15-A8 (И) на микропроцесора се осигуряват адресните битове A15-A8 през целия цикъл на обръщението към паметта или входно-изходните устройства. Сигналите по тези линии са с активно високо ниво. По време на възприемане на прекъсване и в неактивно състояние на микропроцесора изходите A15-A8 са в състояние на висок импеданс.

Линиите за адреси и състояния A19/S6-A16/S3 (И) на микропроцесора са мултиплексирани по време, като първоначално се предава най-старшата част на адреса за паметта (при обръщение към входно-изходните устройства тази част от адреса е нула), а по-късно по тези линии се предава информация за състоянието на микропроцесора. Битовете за състояние S4 и S3 отразяват обръщението към отделните области на

По линията за готовиост **READY (B)** микропроцесорът получава сигнал потвърждение от адресираната памет или входно-изходните устройства, че пренасянето на данни ще завърши в рамките на определения цикъл. Сигналят **READY** е с активно високо ниво и се получава от аналогичния сигнал **RDY** на паметта или входно-изходните устройства след синхронизиране в интегралната схема I8284A.

Линията за прекъсване INTR (B) на микропроцесора осигурява приемането на сигнал с активно високо ниво като заявка за прекъсване. Микропроцесорът анализира този сигнал в последния такт на всяка инструкция и определя необходимостта от въвеждането на цикъл за разпознаване на прекъсване. Ако това се налага, изпълнението на текущата програма се прекъсва и управлението се предава на съответна обслужваща програма чрез таблица на векторите на прекъсванията, разположена в системната памет. Прекъсванията по линията **INTR** могат да бъдат маскирани (забранени) по програмен път чрез нулиране на флага **IF** за разрешение на прекъсванията в микропроцесора.

Линията за немаскируемо прекъсване NMI (B) на микропроцесора осигурява приемането на сигнал — заявка за прекъсване, което се интерпретира като прекъсване от тип 2. В този случай управлението се предава на обслужваща програма чрез таблица на векторите на прекъсванията, разположена в системната памет. Прекъсванията по линията **NMI** не могат да се маскират (забраняват) по програмен път. Всеки сигнал с нарастващ фронт по тази линия се синхронизира вътре в микропроцесора и предизвиква прекъсване в края на изпълнението на текущата инструкция.

Сигналят по линията за проверка **TEST (B)** се анализира програмно при изпълнение на инструкцията **WAIT**. Ако сигналят **TEST** е активен, изпълнението на инструкциите продължава. В противен случай изпълнението на програмата се прекратява и микропроцесорът влиза в т.нар. *пасивно състояние*, като повтаря изпълнението на инструкцията **WAIT**. Сигналят **TEST** трябва да бъде активен (с ниско ниво) най-малко 5 такта на микропроцесора. Той се синхронизира вътре в микропроцесора по нарастващия фронт на тактовия сигнал **CLK**.

Сигналят по линията за начално установяване **RESET (B)** предизвиква незабавно прекратяване на всякакви действия в микропроцесора. Този сигнал трябва да бъде с активно високо ниво и продължителност не по-малка от 4 машинни такта. Сигналят се синхронизира в микропроцесора и при прехода му в неактивно състояние предизвиква вътрешна за микропроцесора последователност от действия с продължителност 7 машинни такта. След изпълнението ѝ регистрите, флаговете и опашката от инструкции на микропроцесора преминават в следното състояние:

Регистри на микропроцесора	Състояние след RESET
Флагове	нулирани
Опашка от инструкции	празна
Брояч на инструкции IP	0000
Кодов сегментен регистър CS	FFFF
Регистри DS,ES,SS	0000

При показаното съдържание на кодовия сегментен регистър **CS** и на брояча на инструкциите **IP** следва, че първата инструкция, която микропроцесорът ще изпълни след действието на сигнала **RESET**, ще бъде извлечена от клетка на паметта с абсолютен адрес **FFFF0**. Тази инструкция при разглежданите компютри е инструкция за междусегментен преход

към програма за начално установяване и проверка на работоспособността на цялата система.

По линията на основния синхросигнал CLK (В) в микропроцесора постъпват основните синхронизиращи сигнали на системата. Те са несиметрични, с коефициент на запълване 33%, при който се осигурява оптимален работен режим.

По захранващата линия V_{CC} на микропроцесора се подава постоянно напрежение +5 V.

Микропроцесорът има две линии за връзка към маса GND.

По линиите за състояние $\overline{S2}$, $\overline{S1}$, $\overline{S0}$ (И) микропроцесорът предава код на състоянието си, съответстващ на изпълнявания обмен по шината за данни (табл.3.2). Сигналите за състояние $\overline{S2}$, $\overline{S1}$, $\overline{S0}$ са валидни в началото на цикъла на микропроцесора (обмена по шината), а в края на цикъла съответстват на пасивно състояние — липса на обмен по шината. Кодът на състоянието се предава към контролера на шината 18288, който го дешифрира и изработва всички управляващи сигнали за обръщение към входно-изходната или системната памет. При неактивно състояние на микропроцесора тези изходи са в състояние на висок импеданс.

Таблица 3.2

Код на състоянието на микропроцесора

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	Състояние
0	0	0	Разпознаване на прекъсване
0	0	1	Четене от входно-изходна памет
0	1	0	Запис във входно-изходна памет
0	1	1	Спряно състояние
1	0	0	Извличане на инструкция
1	0	1	Четене от системната памет
1	1	0	Запис в системната памет
1	1	1	Пасивно състояние

В спряно състояние на микропроцесора всички действия в него се прекратяват до получаване на сигнал за външно прекъсване по линиите INTR или NMI или до активиране на сигнал по линията RESET. Преминването на микропроцесора в спряно състояние се извършва при изпълнение на инструкцията HLT.

По линиите за заявка/разрешение $\overline{RQ/GT0}$, $\overline{RQ/GT1}$ (В/И) се предават сигнали, които осигуряват заемането на вътрешната шина на микропроцесора от други устройства, включени към нея. В края на текущия му цикъл. Всяка от тези линии е двуосочна, при което и сигналът *заявка* \overline{RQ} (В) и сигналът *разрешение* \overline{GT} (И) са с активно ниско ниво. Заявката по линията $\overline{RQ/GT0}$ има по-висок приоритет от тази по линията $\overline{RQ/GT1}$.

По линията за блокиране \overline{LOCK} (И) микропроцесорът издава сигнал, по време на който други устройства, включени към системната шина, не трябва да извършват обмен по нея. Сигналят \overline{LOCK} се изработва при изпълнение на инструкцията LOCK и остава с активно ниско ниво до завършване на следващата инструкция. При неактивно състояние на микропроцесора този изход е в състояние на висок импеданс.

По линиите за състояние на опашката $\overline{QS1}$, $\overline{QS0}$ (И) се предава информация за състоянието на вътрешната опашка от инструкции на микропроцесора, в съответствие с табл.3.3.

Код на състоянието на опашката от инструкции

Q51	Q50	Състояние на опашката от инструкции
0	0	Липса на операция с опашката
0	1	Извлечен е първият байт на инструкцията
1	0	Опашката е празна
1	1	Извлечен е следващ байт на инструкцията

Посочените в таблицата състояния отразяват каква операция е била извършена с опашката от инструкции в предишния такт на микропроцесора.

На извод 34 (И) в максимален режим на микропроцесора се получава винаги логическа единица.

Наборът от инструкции на микропроцесора 18086/8088 е даден в приложение 4.

3.2.2. СИНХРОНИЗАЦИЯ В ПРОЦЕСОРА

За оптимална работа на микропроцесора 18088 са необходими тактови синхросигнали с време на нарастване и време на спадане, не по-голямо от 10 ns. При това ниското ниво се изменя в границите от $-0,5$ до $+0,6$ V, а високото — в границите от 3,9 до 5 V, като коефициентът на запълване е 33%, а максималната честота — 5 MHz. Посочените изисквания се удовлетворяват с помощта на синхрогенератор, изграден с 18284A. Тази интегрална схема осигурява 3 вида синхронизиращи сигнали, използвани в компютъра, а също така входния сигнал READY за микропроцесора и сигнала за начално установяване RESET на цялата система.

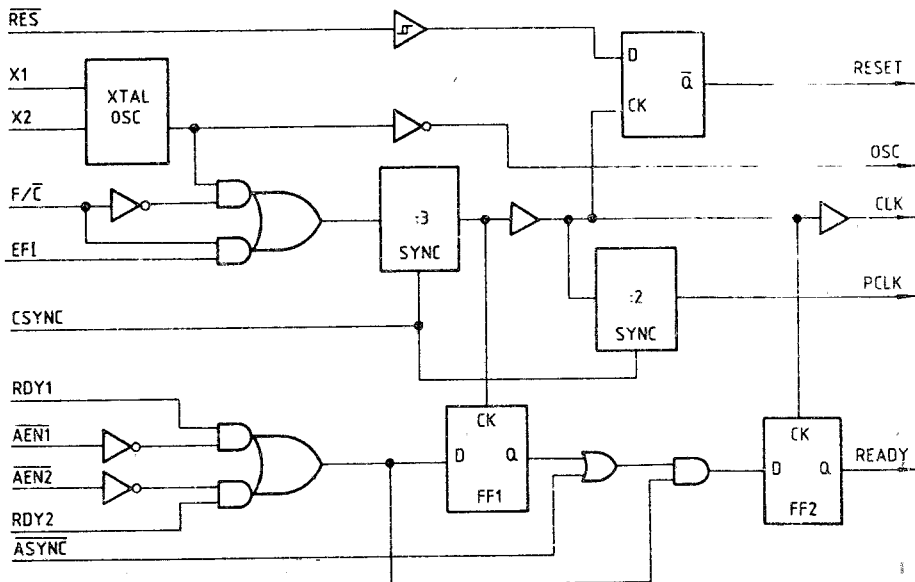
Блоковата схема на 18284A е показана на фиг.3.7, а разположението на изводите — на фиг.3.8.

Към извода V_{CC} се подава постоянно захранващо напрежение + 5 V. Изводът GND се свързва към маса.

Към изводи X1 и X2 се включва външен кварцов резонатор с честота 14,31818 MHz, която трябва да бъде 3 пъти по-висока от желаната за основния синхросигнал на микропроцесора. Необходимо е кварцовият резонатор да бъде с минимално съпротивление при серийния си резонанс, тъй като се включва във веригата за обратна връзка на осцилатора XTAL OSC и трябва да внася минимално затихване. Евентуалното изместване на работната честота от резонансната, водещо понякога до изчезване на осцилациите, се компенсира чрез последователно (серийно) включване към кварцовия резонатор на кондензатор с подходящ капацитет. Изходният сигнал от осцилатора след буферизиране постъпва към изход OSC. Този сигнал, наречен сигнал на осцилатора OSC, е с честота 14,31818 MHz и коефициент на запълване 50%. Разпространява се до всички кундунги за разширение на системната конфигурация.

В интегралната схема 18284A се формират още два синхросигнала: CLK и PCLK. Те се получават при последователното делене на 3 и на 2 (фиг.3.7) на честотата на сигнала от вътрешния осцилатор или на честотата на външен сигнал. Външният сигнал трябва да бъде с коефициент

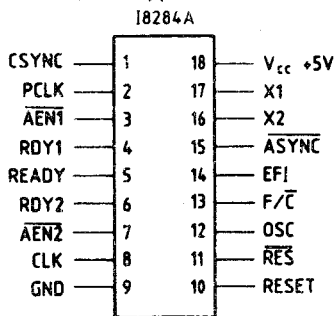
на запълване 50% и с честота също така 3 пъти по-висока от желаната за основния синхросигнал на микропроцесора. Той се подава към вход **EFI**. Използването на **външен синхросигнал EFI** или на сигнала от вътрешния осцилатор се определя от потенциала на **извода за избор на източник на синхросигнали F/C**. За работа със сигнала на вътрешния осцилатор изводът **F/C** се свързва към маса, а за избиране на външен сигнал — към +5 V



Фиг. 3.7. Блокова схема на I8284A

При работа на I8284A в персоналните компютри **изводът за синхронизиране на тактовите сигнали CSYNC** не се използва и трябва да бъде свързан към маса.

На извод **CLK** на I8284A се получава синхросигнал с честота, 3 пъти



Фиг. 3.8. Разположение на изводите на I8284A

по-ниска от честотата на външния синхросигнал **EFI** или от честотата на сигнала на вградения осцилатор. В разглежданите компютри този сигнал се означава с **CLK88** и има честота 4,77 MHz (честотата на осцилатора, разделена на 3), коефициент на запълване 33% и максимална амплитуда +4,5 V. Сигналят **CLK88** е основен за микропроцесора I8088 и устройствата, включени към вътрешната мушина (аритметичния копроцесор I8087 и контролера на системната шина I8288). След буферизиране от него се получава основният синхросигнал **CLK**, постъпващи

към всички куплунги за разширение на конфигурацията на компютъра. От синхросигнала CLK88 в системната платка се формира и синхросигналят DCLK за контролера за директен достъп до паметта I8237A-5, който е със същата честота, но с коефициент на запълване приблизително 50%.

На извод PCLK на I8284A се получава синхросигнал с честота 2,38 MHz (честотата на синхросигнала CLK, разделена на 2) и коефициент на запълване 50%. Той се използва в управляващата логика на интерфейса за клавиатурата. След допълнително разделяне на честотата му на 2 от него се формира синхросигналят CTCLK (с честота 1,19 MHz и коефициент на запълване 50%), подаван към тактовите входове на програмируемите интервални таймери на интегралната схема I8253-5.

При включване на захранването на компютъра на входа за начално установяване RES на I8284A се подава сигнал с активно ниско ниво, който след вътрешно синхронизиране се използва за генериране на системния сигнал за начално установяване RESET. Входният сигнал постъпва от захранващия блок на компютъра при изработване на стабилно захранващо напрежение и има означението POWER GOOD. Входът RES на I8284A е вход на тригер на Шмит, така че постъпващият сигнал може да се формира с проста външна RC-верига. Времоконстантата на тази верига се определя от изискванията за параметрите на изходния сигнал RESET, подаван към микропроцесора I8088. За нормална работа на микропроцесора е необходимо сигнала RESET да бъде с продължителност не по-малка от 4 машинни такта (840 ns), а спаданият му фронт да се формира не по-рано от 50 μ s след включване на захранването. Сигналят RESET е с активно високо ниво. Той постъпва в микропроцесора I8088, в аритметичния копроцесор I8087, в големите интегрални схеми I8287A и I8255A и в редица други схеми на системната платка. След буферизиране се разпространява по линията за начално установяване RESET DRV на системната шина.

Изходният сигнал за готовност READY на I8284A потвърждава, че обменът на данни ще завърши в рамките на определения от микропроцесора цикъл. Той постъпва в микропроцесора и в нормален режим е с активно високо ниво. При необходимост от удължаване на циклите за запис или четене адресираните от микропроцесора устройства трябва в определен момент да забранят сигнала READY. Това става чрез входните линии за готовност RDY1 и RDY2 и съответните им линии за избиране AEN1 и AEN2 (фиг. 3.7). Двойките сигнали, постъпващи по тях, се обединяват чрез функция ИЛИ, резултантният сигнал се анализира във всеки машинен такт и се изработва сигнала READY към микропроцесора. От фигурата се вижда, че в зависимост от входния сигнал ASYNC при формирането на сигнала READY се включват две (при ASYNC = 0) или едно стъпало за синхронизиране (при ASYNC = 1). В разглежданите компютри за формирането на сигнала за готовност READY се използва само едната от двойките входни линии — RDY1 и AEN1. По линията AEN1 постъпва сигнала RDY/WAIT с активно високо ниво, който предизвиква удължаване на циклите за запис или четене на микропроцесора. Този сигнал се изработва в два случая, при това — с различна продължителност: при входно-изходни операции и по асинхронен сигнал от устройствата, включени към системната шина. При изпълнение на входно-изходни операции сигнала RDY/WAIT въвежда в цикъла на микропроцесора един допълнителен машинен такт — такт на изчакване Tw. Вто-

рият случай е свързан с действието на асинхронен сигнал с определена продължителност по линията за готовност на системната шина I/O CHRDY. За времето на *неготовност*, т.е. при ниско ниво на сигнала I/O CHRDY, постъпващ в системната платка, сигналът RDY/WAIT предизвиква удължаване на цикъла на микропроцесора чрез вмъкване на съответен брой тактове на изчакване T_w .

Другата входна линия — RDY1, използвана при формирането на сигнала за готовност READY, има по-особено предназначение. Постъпващият по нея сигнал DMAWAIT с ниско ниво предизвиква принудително задържане на микропроцесора в пасивно състояние в края на текущия му цикъл. Този сигнал се получава при предоставяне на системната шина на контролера за директен достъп до паметта I8237A-5.

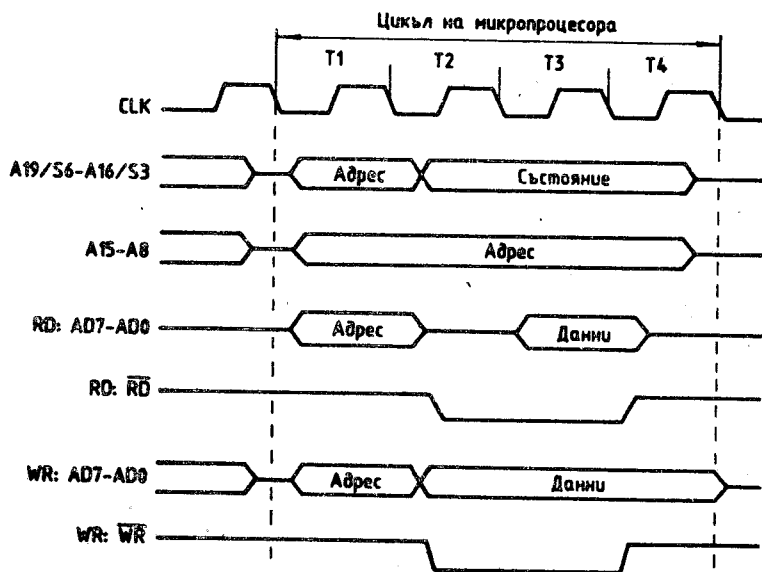
3.2.3. СИСТЕМНА ШИНА

Управление на системната шина от микропроцесора

За осъществяване на обмен с устройствата, включени в адресното му пространство, I8088 изпълнява определени действия в строго определена последователност, съставлящи т.нар. *цикъл на микропроцесора*. В този цикъл той предава адрес за избиране (селектиране) на дадено устройство, следван от сигнал за четене или от данни и сигнал за запис. При цикъл четене адресираното устройство предава необходимите данни, а при цикъл запис — приема данните от микропроцесора. Всеки цикъл на микропроцесора има продължителност най-малко четири периода на синхросигнала CLK, наречени тактове на микропроцесора. Те се означават с T1, T2, T3 и T4. Единствено възможната намеса на устройството в цикъла на микропроцесора е въвеждането на тактове на изчакване T_w чрез активиране на сигнал за неготовност по линията READY на микропроцесора. На фиг.3.9 е показана времедиagramата на циклите на микропроцесора.

При описанието на сигналите на микропроцесора е посочено, че линиите му за адреси, състояния и данни са мултиплексирани по време и образуват неговата вътрешна шина. Така в такта T1 микропроцесорът предава 20-разреден адрес по линиите си A19/S6-A16/S3, A15-A8 и AD7-AD0. В такта T2 той прекратява предаването на адреса по шината си (с изключение на битове A15-A8) и извършва различни действия в зависимост от изпълнявания цикъл. При *цикъл четене* микропроцесорът привежда изходите си AD7-AD0 в състояние на висок импеданс и ги подготвя за приемане на данни. При *цикъл запис* по линиите AD7-AD0 предава данните към адресираното устройство. В такта T2 микропроцесорът издава и сигнали за състояние по линиите си A19/S6-A16/S3, отразяващи обръщението към различни области на адресното пространство. В режим на максимална конфигурация микропроцесорът изработва код за типа на изпълнявания цикъл и го предава по линиите си S2, S1, S0. Сигналите S2, S1 и S0 постъпват в контролера на шината I8288, който ги дешифрира и издава необходимите управляващи сигнали за четене или запис към избраното устройство. В такта T3 микропроцесорът поддържа сигналите си за състояние и данните за запис. При цикъл четене приема данните от устройството по линиите си AD7-AD0. При сигнал за неготовност от

устройството микропроцесорът въвежда след такта T3 определен брой тактове на изчакване T_w , в които изпълнява действия, аналогични на тези в T3. В такта T4 изходите за адреси и данни се привеждат в състояние на висок импеданс, а по линиите S2, S1 и S0 се предава кодът за пасивно състояние. Това сигнализира завършване на цикъла на микропроцесора.



Фиг.3.9. Времени диаграма на циклите на микропроцесора 18086/8088

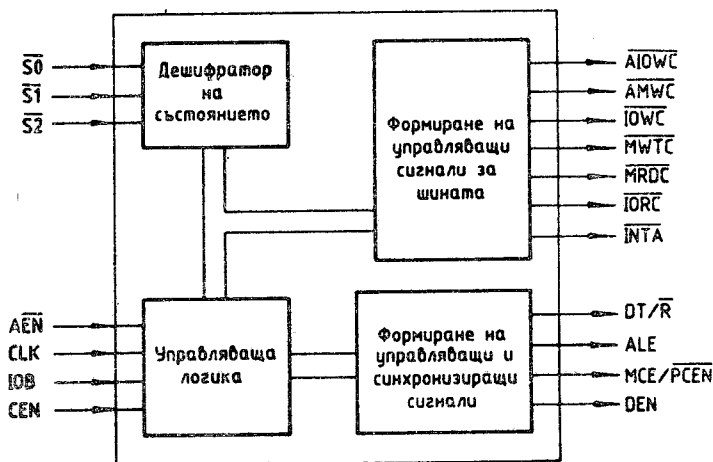
Устройствата, които обменят информация с микропроцесора, имат отделни линии за адреси и за данни и изискват непроменящи се адресни сигнали през целия цикъл на четене или запис. Затова е необходимо в такта T1 на всеки цикъл на микропроцесора адресът от мултиплексирания му шина да се записва във външни регистри. В персоналните компютри за запис на адресните битове A19-A16 и A7-A0 се използват регистри с три устойчиви състояния. Обикновено това са ИС от тип LS373. Адресните линии A15-A8 на микропроцесора не са мултиплексирани и се свързват към входовете на еднопосочни буфери с три устойчиви състояния (ИС от тип LS244). За по-лесно управление на обмена на данни по мултиплексирания шина на I8088 и за повишаване на товароспособността на изходите му линиите за данни AD7-AD0 са свързани към двупосочни буфери с три устойчиви състояния (ИС от тип LS245). Така чрез външни регистри и буфери се демултиплексира вътрешната шина на микропроцесора и се получават отделни линии за адреси, състояния и данни. Тези линии са част от т.нар. *системна шина*. Тъй като по нея се извършва обменът на данни между микропроцесора и системната или входно-изходната памет, тя се нарича още *входно-изходен канал*. За намаляване на общия брой на устройствата, свързани директно към системната шина, в системната платка на компютъра се извършва и второ буфериране на

адресите, данните и управляващите сигнали.

В режим на максимална конфигурация на микропроцесора сигналът за запис на адреса във външните му адресни регистри, сигналите за управление на двупосочните буфери за данни и управляващите сигнали за запис и четене от системната или входно-изходната памет се изработват от контролера на шината I8288.

Контролер на шината I8288

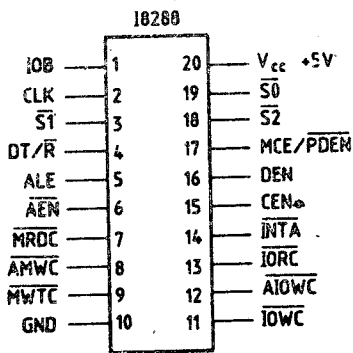
Блоковата схема на контролера I8288 е показана на фиг.3.10, разположението на изводите — на фиг.3.11, а времедиagramата на сигналите му — на фиг.3.12.



Фиг.3.10. Блокова схема на I8288

Към извода V_{cc} се подава постоянно захранващо напрежение +5 V. Изводът GND се свързва към маса.

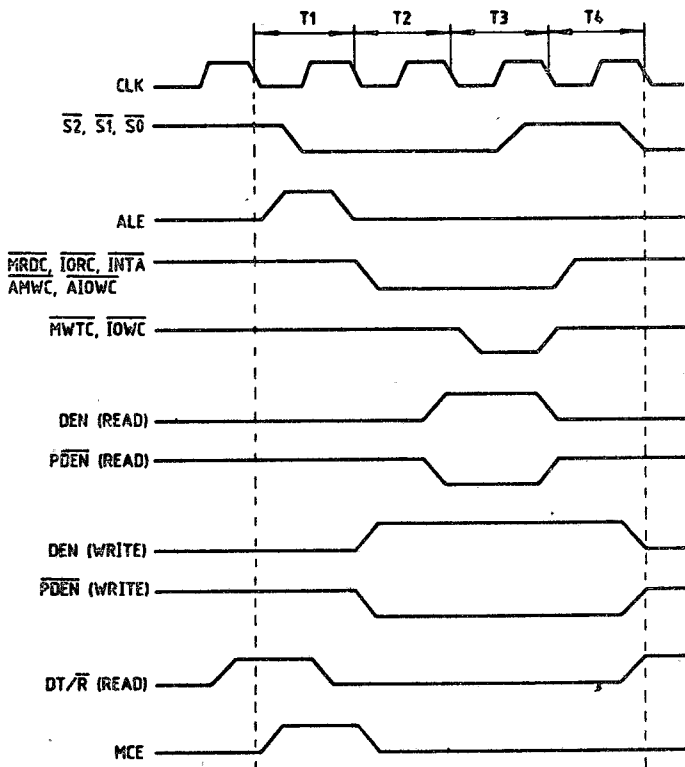
Към изводите за състояние S_2, S_1, S_0 (B) се предават съответните сигнали на микропроцесора I8088, кодиращи състоянието му. Тези сигнали указват необходимостта от започване на обмен на данни по системната шина, наречен цикъл на канала, и определят типа, началото и завършването му. Контролерът на шината анализира входните сигнали S_2, S_1 и S_0 в началото на всеки машинен такт. За да иницира цикъл на канала, микропроцесорът променя кода за пасивно състояние по линиите S_2, S_1 и S_0 в код, идентифициращ определен цикъл (табл.3.4). Това се извършва в последния такт на предишния цикъл T4 или в тактове на пасивно състояние T1. Контролерът



Фиг.3.11. Разположение на изводите на I8288

18288 дешифрира получения код и генерира в строго определена последователност необходимите управляващи и синхронизиращи сигнали за изпълнението на активирания цикъл.

Към извода CLK (B) се подават тактовите сигнали CLK, изработвани от 18284A. Те се използват при вътрешната синхронизация в изработването на управляващите сигнали на контролера на шината.



Фиг.3.12. Времениаграма на сигналите на 18288

Таблица 3.4

Управляващи сигнали на контролера на шината 18288

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	Цикъл на канала	Сигнали на 18288
0	0	0	Разпознаване на прекъсване	\overline{INTA}
0	0	1	Четене от входно-изходна памет	\overline{IORC}
0	1	0	Запис в входно-изходна памет	\overline{IOWC} , $\overline{A10WC}$
0	1	1	Спряно състояние на микропроцесора	—
1	0	0	Извличане на инструкция	\overline{MRDC}
1	0	1	Четене от системната памет	\overline{MRDC}
1	1	0	Запис във системната памет	\overline{MWTC} , \overline{AMWC}
1	1	1	Пасивно състояние на микропроцесора	—

На извод \overline{IORC} (И) се получава управляващ сигнал за четене от входно-изходната памет. Сигналят е с активно ниско ниво.

На извод \overline{IOWC} (И) се получава управляващ сигнал за запис във входно-изходната памет. Сигналят е с активно ниско ниво.

На извод \overline{AIOWC} (И) се получава т.нар. изпреварващ управляващ сигнал за запис във входно-изходната памет. Спадащият фронт на този сигнал се формира с един такт по-рано в сравнение със спадащия фронт на сигнала \overline{IOWC} . Сигналят \overline{AIOWC} е с активно ниско ниво. Управляващите сигнали \overline{IOWC} и \overline{AIOWC} се използват в зависимост от изискванията към времедиаграмите на работа на конкретните входно-изходни устройства.

На извод \overline{MRDC} (И) се получава управляващ сигнал за четене от системната памет. Сигналят е с активно ниско ниво.

На извод \overline{MWTC} (И) се получава управляващ сигнал за запис в системната памет. Сигналят е с активно ниско ниво.

На извод \overline{AMWTC} (И) се получава изпреварващ управляващ сигнал за запис в системната памет. Спадащият фронт на този сигнал се формира с един такт по-рано в сравнение със спадащия фронт на сигнала \overline{MWTC} . Сигналят \overline{AMWTC} е с активно ниско ниво.

На извод \overline{INTA} (И) се получава управляващ сигнал с активно ниско ниво, въвеждащ цикъл за разпознаване на прекъсване. При получаване на сигнал заявка за маскируемо прекъсване по линията \overline{INTR} микропроцесорът започва изпълнението на два последователни такива цикъла. Във втория от тях контролерът на прекъсванията I8259A трябва да предаде към микропроцесора номера на прекъсването.

На извод \overline{ALE} (И) се получава сигнал за запис на адреса от мултиплексиранията вътрешна шина на микропроцесора във външните му регистри. Сигналят е с активно високо ниво и записът се извършва по спадащия му фронт.

На извод \overline{DEN} (И) се получава разрешаващ сигнал за буферите за данни на микропроцесора. Сигналят е с активно високо ниво.

На извод $\overline{DT/R}$ (И) се получава сигнал, управляващ посоката на пропускане на двупосочните буфери за данни на микропроцесора. При циклите на канала за запис във входно-изходната или системната памет сигналят $\overline{DT/R}$ е с високо ниво, а при циклите на канала за четене — с ниско ниво.

При подаване на сигнал с ниско ниво към извод \overline{AEN} (В) се разрешава изработването на изходните управляващи сигнали за запис и четене от входно-изходната или системната памет, както и на сигнала за разпознаване на прекъсване \overline{INTA} . При сигнал с високо ниво на този вход изходите на контролера на шината преминават в състояние на висок импеданс.

Когато изводът \overline{IOB} е свързан към +5 V, сигналят \overline{AEN} не влияе на изработването на управляващите сигнали \overline{IORC} , \overline{IOWC} , \overline{AIOWC} и \overline{INTA} . Сигналят \overline{AEN} се изработва от управляващата логика за предоставяне на системната шина на микропроцесора ($\overline{AEN}=0$) или на контролера за директен достъп до паметта I8237A ($\overline{AEN}=1$).

При подаване на сигнал с ниско ниво към извод \overline{CEN} (В) всички изходи на контролера на шината, включително изходите \overline{DEN} и \overline{PDEN} , преминават в неактивно състояние. При високо ниво на сигнала \overline{CEN} всички изходи на контролера на шината са разрешени.

Режимът на работа на контролера на шината I8288 се определя от приложения потенциал на извода IOB (B). При свързване на този извод към +5 V контролерът се привежда в режим, позволяващ управление на системната шина и на отделна входно-изходна шина, предназначена единствено за микропроцесора. Тази възможност не се използва при разглежданите персонални компютри.

При свързване на извода IOB към маса контролерът се привежда в режим на управление на системната шина. В този режим всички управляващи сигнали се изработват след получаване на входния сигнал с активно ниско ниво AEN.

Сигналът на извод MCE/PDEN (И) се дефинира по различен начин, в зависимост от режима на работа на контролера I8288. При свързване на извод IOB към +5 V изходният сигнал PDEN се използва за управление (разрешаване) на буферите за данни на вътрешната входно-изходна шина на микропроцесора. Този сигнал е аналогичен на управляващия сигнал за буферите на системната шина DEN, но е с активно ниско ниво. При свързване на извод IOB към маса изходният сигнал MCE се изработва в циклите за разпознаване на прекъсване. Той е с активно високо ниво и с него се управлява четенето на адреса на главния контролер на прекъсванията I8259A, работещ в каскаден режим. При системи с един контролер на прекъсванията сигналът MCE не се използва.

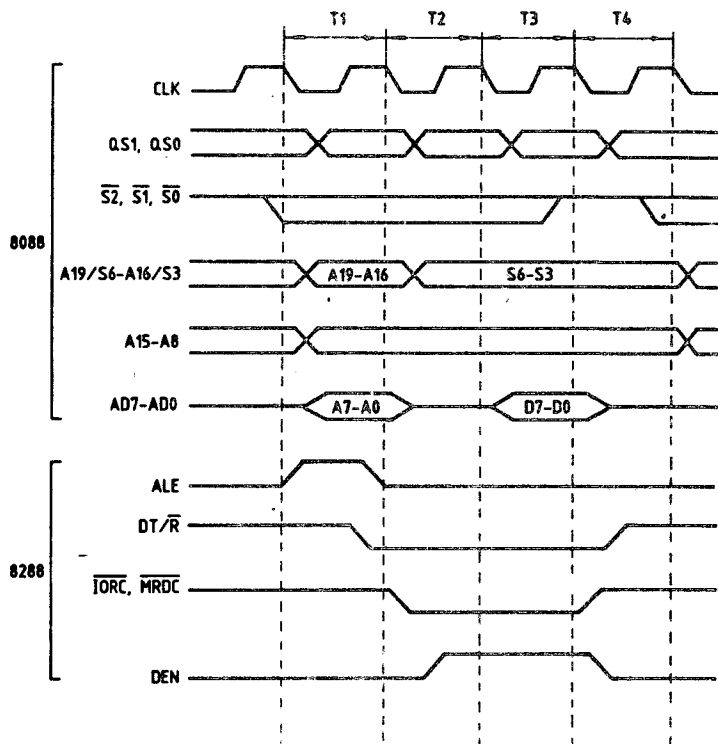
Времедиаграми на циклите на канала

Контролерът на шината I8288 се използва в режим на максимална конфигурация на микропроцесора за управление на обмена с устройствата, свързани към системната шина (входно-изходния канал). Времедиаграмите на циклите на канала са показани на фиг.3.13 и 3.14. Типът на цикъла — разпознаване на прекъсване, четене или запис в системната или във входно-изходната памет, се задава чрез сигналите за състояние на микропроцесора S2, S1 и S0. Въвеждането на нов цикъл става чрез промяна на състоянието в края на текущия цикъл (такта T4) или в текущия такт на пасивно състояние T1.

Контролерът на шината дешифрира състоянието S2, S1, S0 и в следващия такт на микропроцесора изработва сигнала ALE. Този такт е първият от въвеждания цикъл и се означава с T1. По спадания фронт на сигнала ALE адресната информация, установена по това време върху мултиплексираните линии A19/S6-A16/S3 и AD7-AD0, се записва във външните адресни регистри. Линиите на микропроцесора A15-A8 не са мултиплексирани и по тях през целия цикъл се предават адресните битове A15-A8.

В такта T2 микропроцесорът премахва адреса от линиите си AD7-AD0; в цикъл за четене ги подготвя за приемане на данни, а в цикъл за запис по тях предава незабавно данните към адресираното устройство. Управлението на двупосочните буфери за данни се осъществява от сигналите на контролера на шината DEN и DT/R. В зависимост от типа на изпълнявания цикъл контролерът изработва и съответния управляващ сигнал за четене или за запис в паметта. Осигуряват се два типа управляващи сигнали за запис — нормални и изпреварващи.

Нормалните сигнали за запис — \overline{MWTC} и \overline{IOWC} се активират в такта T3, т.е. след установяването на данните, предавани от микропроцесора. Тези сигнали се използват при устройства, в които записът се извършва по спадания фронт на управляващия сигнал. Изпреварващите сигнали за запис — \overline{AMWC} и \overline{AIOWC} се активират един такт по-рано, т.е. в началото на такта T2. Данните, предавани от микропроцесора, се установяват по-късно в такта. Те се записват в адресираното устройство по нарастващия фронт на управляващия сигнал.



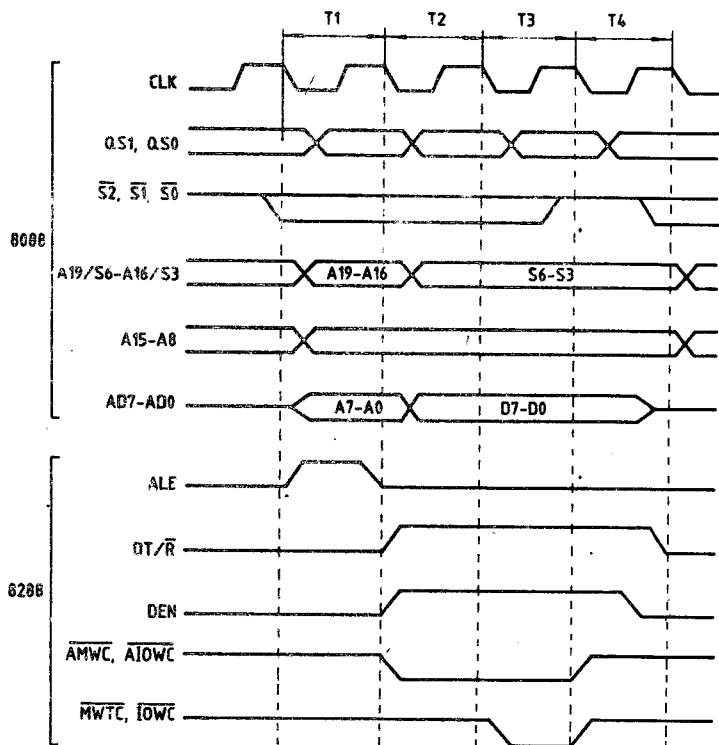
Фиг. 3.13. Времедиаграма на цикъла на канала за четене от системната памет

Сигнал за готовност по линията \overline{READY} се приема най-късно в края на такта T2. В зависимост от продължителността на този сигнал микропроцесорът въвежда след такта T3 определен брой тактове на изчакване T_w . В последния такт T_w или в края на такта T3 (при неудължен цикъл) микропроцесорът предизвиква преход към високо ниво на сигналите си $\overline{S2}, \overline{S1}$ и $\overline{S0}$. Това съответства на преминаването му в пасивно състояние и указва предстоящото завършване на цикъла. В следващия, последен такт T4 контролерът на шината прекратява действието на сигналите си и следи за нова промяна в състоянието $\overline{S2}, \overline{S1}, \overline{S0}$ на микропроцесора, т.е. за евентуално въвеждане на нов цикъл.

Времедиаграмата на цикъла за разпознаване на прекъсване е аналогична на тази на цикъл четене. Основната разлика се състои в изработването на сигнал \overline{INTA} вместо сигнал за четене. Микропроцесорът не изда-

ва адрес, тъй като сигналът \overline{INTA} постъпва единствено в контролера на прекъсванията I8259A. При втория от двата последователно изпълнявани цикъла за разпознаване на прекъсване контролерът I8259A предава по шината за данни номера на прекъсването.

Минималната продължителност на цикъла на канала е 4 такта (840 ns). Той се удължава с един такт T_w (210 ns) при изпълнение на входно-изходните инструкции и с определен брой тактове T_w в зависимост от продължителността на сигнала за неготовност на устройствата, включени към системната шина.



Фиг.3.14. Времениаграма на цикъла на канала за запис в системната памет

Линии на системната шина

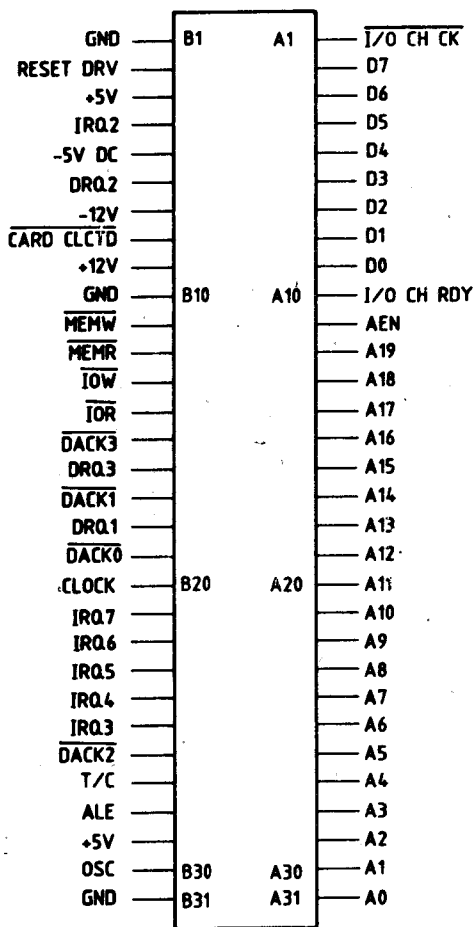
Към всеки от куплунгите за разширение на системната конфигурация на персоналните компютри са свързани 62 линии, образуващи системната шина. Разположението на изводите на куплунгите е показано на фиг.3.15.

По линията на сигнала на осцилатора OSC (И) се разпространява синхронизиращ сигнал с честота 14,31818 MHz (период на повторение 70 ns) и коефициент на запълване 50%.

По линията на основния синхросигнал на системата CLK (И) се разпространява синхронизиращ сигнал с честота, 3 пъти по-ниска от честота-

та на осцилатора — 4,77 MHz (период на повторение 210 ns) и с коефициент на запълване 33%.

! По линията за начално установяване **RESET DRV (И)** се разпространява сигнал, който привежда системата в начално състояние след включване на захранващото напрежение на компютъра. Сигналят е с активно високо ниво.



Фиг.3.15. Разположение на изводите на куплунгите за разширение на системната конфигурация

Адресните линии **A19-A0 (И)** се използват при адресирането на системната и входно-изходната памет. Най-младшият бит на адреса се предава по линията **A0**, а най-старшият — по линията **A19**. Адресните сигнали се генерират от микропроцесора или от контролера за директен достъп до паметта (ДДП).

По линиите за данни **D7-D0 (В/И)** се осъществява обменът на данни между микропроцесора, паметта и входно-изходните устройства. Сигналите по тези линии са с активно високо ниво. Най-младшият бит на данните е **D0**, а най-старшият — **D7**.

По линията за разрешаване на запис на адреса **ALE (И)** се разпространява сигнал, който се изработва от контролера на шината **I8288**. Този сигнал се използва в системната платка за запис на адреса, изработен от микропроцесора, във външните му адресни регистри. Записът се извършва по спадания фронт на сигнала. За периферните устройства този сигнал служи като индикация за установен адрес от микропроцесора.

По линията за разрешаване на адреса при ДДП **AEN (И)** се разпространява сигнал, който показва, че системната шина се намира под управлението на контролера за ДДП **I8237A-5**.

По линиите за заявки за ДДП **DRQ1-DRQ3 (В)** постъпват асинхронни заявки от периферните устройства за обмен чрез ДДП. Най-висок приоритет има сигналът заявка по линията **DRQ1**, а най-нисък — сигналът по линията **DRQ3**. Сигналят заявка трябва да се поддържа с активно високо ниво до получаването на съответния сигнал потвърждение **DACK** от контролера за ДДП.

По линиите за потвърждаване на заявката за ДДП **ДАСК0-ДАСК3 (И)** към периферните устройства се предават сигналите за потвърждение на обмена чрез ДДП. Тези сигнали са с активно ниско ниво. Сигналят **ДАСК0** служи за опресняване на динамичната памет.

По линията за край на обмена чрез ДДП **Т/С (И)** се предава сигнал с активно високо ниво, който показва, че броячът на активния канал на контролера за ДДП е достигнал гранично състояние **Т/С**.

По линията за четене от паметта **MEMR (И)** се разпространява управляващ сигнал с активно ниско ниво. Сигналят се изработва от контролера на шината **I8288** или от контролера за ДДП и служи за четене от паметта.

По линията за запис в паметта **MEMW (И)** се разпространява управляващ сигнал с активно ниско ниво. Сигналят се изработва от контролера на шината **I8288** или от контролера за ДДП и служи за запис на информация в паметта.

По линията за четене от входно-изходната памет **IOR (И)** се разпространява управляващ сигнал с активно ниско ниво. Сигналят се изработва от контролера на шината **I8288** или от контролера за ДДП и служи за четене от входно-изходната памет.

По линията за запис във входно-изходната памет **IOW (И)** се разпространява управляващ сигнал с активно ниско ниво. Сигналят се изработва от контролера на шината **I8288** или от контролера за ДДП и служи за запис на информация във входно-изходната памет.

По линията за готовност на системната шина **I/O CH RDY (И)** се разпространява сигнал за потвърждение от периферните устройства, че обменът на данни ще завърши в рамките на определения цикъл на канала. Устройства с по-малко бързодействие могат да удължат цикъла на канала чрез задържане на този сигнал с ниско ниво в продължение на определен брой машинни тактове (не повече от 10).

По линията за грешка в системната шина **I/O CH CK (В)** се разпространява сигнал с активно ниско ниво при откриване на грешка по четност в устройствата, включени към входно-изходния канал.

По линиите за заявки за прекъсване от входно-изходните устройства. Най-висок приоритет има сигнала за заявка по линията **IRQ2**, а най-нисък — сигнала по линията **IRQ7**. Заявка за прекъсване по дадена линия се генерира по нарастващия фронт на съответния сигнал.

Линиите за захранващите напрежения се използват за подаване на захранващи напрежения към устройствата, включени към куплуните за разширение на системната конфигурация. Осигурени са линии за подаване на захранващи напрежения **+5 V**, **-5 V**, **+12 V** и **-12 V**, както и три линии за свързване към маса **GND**.

3.2.4. СИСТЕМНА ПАМЕТ

Микропроцесорът **I8086/8088** може да използва адресно пространство за памет до **1 Мбайт** и отделно входно-изходно адресно пространство до **64 Кбайта**. Обръщенията към адресните пространства за памет и вход-изход се определят от изпълняваните инструкции. Примерно разпределение на паметта в персоналните компютри е посочено в табл.3.5.

Разпределение на паметта в персоналните компютри

Адреси	Обем	Тип	Предназначение
00000—9FFFF	640К	RAM	Памет за вектори на прекъсвания (1К) — адреси 00000-003FF Памет, използвана от вътрешното програмно осигуряване Работна област на ДОС Резидентна област на ДОС Памет за програмите на потребителя Памет за командния интерпретатор на ДОС
A0000—AFFFF	64К		Неизползвано адресно пространство
B0000—B7FFF	32К	RAM	Буферна памет на текстовия видеоконтролер
B8000—BFFFF	32К	RAM	Буферна памет на графичния видеоконтролер
C0000—EFFFF	192К	ROM	Допълнително ПЗУ — управляващи програми на контролера на ЗУТМД и др.
F0000—F5FFF	24К		Неизползвано адресно пространство
F6000—FDFFF	32К	ROM	ПЗУ за вътрешното програмно осигуряване — интерпретатор на БЕЙСИК
FE000—FFFFF	8К	ROM	ПЗУ за вътрешното програмно осигуряване — базова система за вход-изход

Оперативната памет (ОП) на компютъра е изградена с ИС от тип 4164 — 64К × 1 с време на достъп 200 ns и време на цикъл 345 ns. Елементите на паметта са динамични и изискват опресняване. За тази цел се използва канал 0 на контролера за ДДП I8237A-5. Върху системната платка са разположени цокли за 36 ИС памет, което осигурява максимален обем 256 Кбайта с контрол по нечетност. Увеличаването на обема става чрез добавяне на нови платки върху куплингите за разширение на системната конфигурация.

Установеният обем памет се задава чрез микропревключватели върху процесорната платка, като състоянието им се отчита при дешифриране на адреса за обръщението към паметта. В дешифрирането участват старшите адресни битове A19-A16, които определят използваната област от паметта. Младшата част на адреса A15-A0 се предава в елементите на паметта на два пъти — като адрес на ред и адрес на колона по сигналите RAS и CAS, получени в резултат на дешифрирането. Адресите за обръщението към паметта се издават от микропроцесора или от контролера за ДДП. При запис схемите за проверка изработват контролен разред въз основа на постъпващата информация. При четене откритите грешки по нечетност предизвикват формирането на сигнала PCK и на заявка за немаскируемо прекъсване NMI.

Постоянното запомнящо устройство (ПЗУ) има обем 40 Кбайта и заема адреси от F6000 до FFFFF. То съхранява интерпретатор на езика БЕЙСИК и програмите на базовата система за вход-изход. Тези програми осигуряват следните системни функции:

- установяване на системата в начално състояние;
- вътрешно тестване при включване на захранването или при начално установяване;
- управление на обмена с периферните устройства, включени в основната конфигурация на системата;

- зареждане на операционната система от дисково устройство;
- знаков генератор за 128 символа.

При някои персонални компютри ПЗУ заема и адреси от F4000 до F5FFF, където се съхраняват управляващи програми и прекодиращи таблици за работа с различни клавиатури.

Постоянното запомнящо устройство е изградено с ИС от тип 2764 — 8К × 8 с време на достъп, не по-голямо от 250 ns, и време на цикъл 350 ns. Върху системната платка са разположени цокли за 8 ИС, осигуряващи възможност за изграждане на ПЗУ до 64 Кбайта.

Разпределението на входно-изходното адресно пространство е посочено в табл.3.6.

Таблица 3.6

Разпределение на входно-изходното адресно пространство

Адрес	Предназначение	използват се от:
000-00F	Контролер за ДДП 8237А-5	
020-021	Контролер на прекъсванията 8259А	
040-043	Програмируем интервален таймер 8253-5	
060-063	Програмируем паралелен интерфейс 8255А-5	
080-083	Регистри на страниците при ДДП	
0АХ	Регистър за разрешение на немаскируемите прекъсвания	
0С0-0СF	Не се използват	
0Е0-0ЕF	Не се използват	
200-20F	Адаптер за игрови пулт	
210-217	Устройство за разширение на системната конфигурация	
278-27F	Втори адаптер за паралелен интерфейс	
2F8-2FF	Втори адаптер за асинхронен последователен интерфейс	
320-32F	Контролер на ЗУТМД	
378-37F	Адаптер за паралелен интерфейс	
380-38F	Контролер SDLC	
3В0-3ВF	Контролер за дисплей за черно-бяло изображение и адаптер за паралелен интерфейс	
3D0-3DF	Контролер за дисплей за цветна графика	
3F0-3F7	Контролер на ЗУГМД	
3F8-3FF	Адаптер за асинхронен последователен интерфейс	

3.2.5. ДИРЕКТЕН ДОСТЪП ДО ПАМЕТТА

Цикли на контролера за ДДП

За периферните устройства на персоналните компютри е предвидена възможност да извършват обмен на данни със системната памет директно, без непосредственото управление на микропроцесора. Това се определя от включения програмируем контролер за директен достъп до паметта I8237А-5. Контролерът осигурява четири независими канала за обмен чрез ДДП при висока скорост на предаване на данни. Всеки от каналите може да бъде програмиран отделно за работа в различни режими на обмен чрез ДДП. Обменът се инициира програмно или по сигнал-заявка от периферното устройство, включено към дадения канал. Специална управляваща логика в компютъра предоставя системната шина на контролера за ДДП, който поема управлението ѝ и осигурява прехвърлянето на данни между системната памет и периферното устройство. В perso-

налните компютри предназначението на отделните канали на контролера за ДДП е следното:

Канал 0 — поддържа опресняването на оперативната памет (с най-висок приоритет)

Канал 1 — обслужва комуникационните адаптери

Канал 2 — поддържа обмена на контролера на ЗУГМД

Канал 3 — поддържа обмена на контролера на ЗУТМД (с най-нисък приоритет)

Контролерът 18237А-5 изпълнява два основни цикъла — пасивен цикъл и цикъл за обмен чрез ДДП (активен цикъл). Във всеки от тях се различават определен брой вътрешни състояния S_i , всяко с продължителност един период на входния синхросигнал DCLK (210 ns). *Пасивен цикъл* се въвежда, когато няма заявка за обмен чрез ДДП в нито един от каналите. В тактовете на *пасивно състояние* S_i се следи за получаването на заявка за обмен DREQ от периферните устройства, както и за получаването на сигнал CS за обръщение на микропроцесора към контролера. При липса на обмен чрез ДДП по сигнала CS се въвежда *състояние на програмиране на контролера*. В това състояние се осъществява обмен с микропроцесора по двупосочните линии за данни DB7-DB0.

Вътрешните регистри на контролера заемат шестнадесет адреса от входно-изходното адресно пространство на компютъра. За запис в тях или за прочитане на съдържанието им микропроцесорът изпълнява входно-изходни инструкции. Отделните регистри се избират чрез адресните линии A3-A0, а записът и четенето се управляват съответно по сигналите IOW и IOR.

Поради големия брой на вътрешните регистри и различната им редност за генериране на допълнителен адресен бит се използва вътрешен **тригер указател на байтове**. При състояние 0 на този тригер се адресират младшите байтове на адресните регистри и регистрите броячи, а при състояние 1 — старшите им байтове. При всяко обръщение за четене или запис в тези регистри състоянието на тригера се променя алтернативно. Тригерът се нулира при получаване на апаратен сигнал за начално установяване RESET или при изпълнение на специалните програмни команди *Начално установяване* и *Нулиране на тригера указател на байтове*. Посочените команди, както и специалната програмна команда *Нулиране на регистъра за маските* се задават от микропроцесора само чрез определени адресни сигнали и чрез управляващия сигнал за запис във входно-изходната памет IOW. Информацията по шината за данни в този случай е без значение. Сигналите на микропроцесора за задаване на специалните програмни команди към контролера за ДДП, както и за обръщения за четене или запис във вътрешните му регистри са дадени в табл. 3.7 и 3.8.

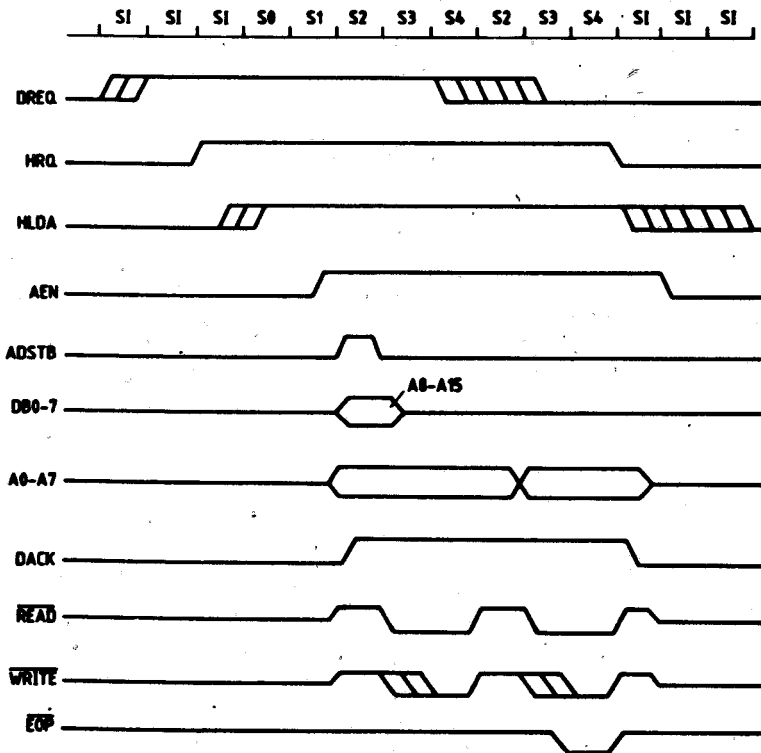
След установяване на начално състояние от микропроцесора контролерът е в състояние да приема заявки и да управлява извършването на обмен чрез ДДП. Заявките за обмен от периферните устройства постъпват по линиите DREQ0-DREQ3. Те се приемат в тактовете на пасивно състояние на контролера S_i при условие, че не са маскирани, и в съответствие с установения приоритет на каналите. Когато в някой от каналите се приеме заявка за обмен чрез ДДП, контролерът въвежда *активен цикъл*. Той издава сигнал за заемане на системната шина HRQ и преминава в състояние, означавано с S_0 .

Времедиagramата на активния цикъл на контролера е показана на фиг. 3.16. Сигналят HRQ постъпва в *управляващата логика за предоставяне на системната шина* на микропроцесора или на контролера за ДДП.

Регистри и програмни команди на I8237A-5

A3-A0	Сигнали			Операция
	CS	IOR	IOW	
8	0	0	1	Четене на регистъра за състояние
8	0	1	0	Запис в управляващия регистър
9	0	0	1	—
9	0	1	0	Запис в регистъра за заявки
A	0	0	1	—
A	0	1	0	Запис в определен разред на регистъра за маските
B	0	0	1	—
B	0	1	0	Запис в регистъра за режима
C	0	0	1	—
C	0	1	0	Команда Нулиране на тригера указател на байтове
D	0	0	1	Четене на буферния регистър за данни
D	0	1	0	Команда Начало установяване
E	0	0	1	—
E	0	1	0	Команда Нулиране на регистъра за маските
F	0	0	1	—
F	0	1	0	Запис във всички разреди на регистъра за маските

По този сигнал при пасивно състояние на микропроцесора или в края на текущия му цикъл управлението на системната шина се предава на контролера I8237A-5. Той получава сигнала потвърждение HLDA и пре-



Фиг. 3.16. Времедиаграма на цикъла на контролера за ДДП I8237A-5

Таблица 3.8

Адресни и броячни регистри на I8237A-5

Канал	Регистър	Операция	Сигнали			Тригер-указател на байт	Шина за данни DB7-DB0	
			CS	IOR	IOW			
0	Начален и текущ адрес	запис	0	1	0	0	0	A7—A0
		текущ адрес	0	1	0	0	1	A15—A8
	Текущ адрес	четене	0	0	1	0	0	A7—A0
		текущ адрес	0	0	1	0	1	A15—A8
	Начален и текущ брояч	запис	0	1	0	1	0	W7—W0
		текущ брояч	0	1	0	1	1	W15—W8
Текущ брояч	четене	0	0	1	1	0	W7—W0	
	текущ брояч	0	0	1	1	1	W15—W8	
1	Начален и текущ адрес	запис	0	1	0	2	0	A7—A0
		текущ адрес	0	1	0	2	1	A15—A8
	Текущ адрес	четене	0	0	1	2	0	A7—A0
		текущ адрес	0	0	1	2	1	A15—A8
	Начален и текущ брояч	запис	0	1	0	3	0	W7—W0
		текущ брояч	0	1	0	3	1	W15—W8
Текущ брояч	четене	0	0	1	3	0	W7—W0	
	текущ брояч	0	0	1	3	1	W15—W8	
2	Начален и текущ адрес	запис	0	1	0	4	0	A7—A0
		текущ адрес	0	1	0	4	1	A15—A8
	Текущ адрес	четене	0	0	1	4	0	A7—A0
		текущ адрес	0	0	1	4	1	A15—A8
	Начален и текущ брояч	запис	0	1	0	5	0	W7—W0
		текущ брояч	0	1	0	5	1	W15—W8
Текущ брояч	четене	0	0	1	5	0	W7—W0	
	текущ брояч	0	0	1	5	1	W15—W8	
3	Начален и текущ адрес	запис	0	1	0	6	0	A7—A0
		текущ адрес	0	1	0	6	1	A15—A8
	Текущ адрес	четене	0	0	1	6	0	A7—A0
		текущ адрес	0	0	1	6	1	A15—A8
	Начален и текущ брояч	запис	0	1	0	7	0	W7—W0
		текущ брояч	0	1	0	7	1	W15—W8
Текущ брояч	четене	0	0	1	7	0	W7—W0	
	текущ брояч	0	0	1	7	1	W15—W8	

минава в първото работно състояние S1. Едновременно с това управляващата логика на системната шина изпраща към микропроцесора сигнала DMAWAIT, с което той се задържа принудително в състояние на изчакване. Изработват се и сигнали за отделяне на буферите за адреси, данни и управляващи сигнали на микропроцесора от системната шина и за разрешаване на буферите на контролера за ДДП.

При адресирането на системната памет каналите на контролера използват 16-разредни вътрешни адресни регистри и регистри броячи на

байтове, които се модифицират автоматично след всеки обмен чрез ДДП. Това определя възможността за пълна адресация само в рамките на т.нар. *страници на паметта* с максимален обем 64 Кбайта. Адресът на страницата, т.е. старшата част на пълния 20-разреден адрес за системната памет, се задава чрез 4-разредни външни **регистри на страниците**. Всеки от каналите на контролера за ДДП, с изключение на канал 0, използва по един такъв регистър. Регистрите на страниците засмат следните адреси от входно-изходното адресно пространство на компютъра:

Адрес на регистъра	Предназначение
80	
81	Регистър на страница за канал 2
82	Регистър на страница за канал 3
83	Регистър на страница за канал 1

Зареждането им се извършва от микропроцесора при подготовката на обмен чрез ДДП. При започване на обмен в даден канал съдържанието на съответния регистър на страницата определя битове A19-A16 от адреса за системната памет. Останалата част от адреса — битове A15-A0 се задава от контролера за ДДП. При това с цел намаляване броя на изходите му линиите за старшия байт на адреса са мултиплексирани с линиите за данни DB7-DB0. По тях в началото на цикъла за обмен чрез ДДП адресните битове A15-A8 се предават за запис във **външен адресен регистър**. Записът се осъществява по сигнала ADRSTB, изработен от контролера в състояние S1. Тъй като при обмен чрез ДДП адресите за паметта са последователни, външният регистър се зарежда само при възникване на пренос от младшия адресен байт, т.е. по веднъж на всеки 256 цикъла. Затова в показаната времедиаграма на фиг. 3.16 във втория цикъл на контролера се различават само работните състояния S2, S3 и S4. По такъв начин обменът на данни се ускорява значително. Младшата част на адреса за паметта се предава от контролера директно по линиите му A7-A0. В циклите за обмен чрез ДДП изходите на външния адресен регистър и на съответния регистър на страницата, както и на адресните буфери се разрешават по сигнала DMA AEN. В разглежданите компютри този сигнал се изработва от управляващата логика за предоставяне на системната шина на контролера за ДДП.

Едновременно с предаването на адреса за системната памет в работното състояние S2 контролерът изпраща сигнал към периферното устройство, с което ще бъде изпълнен обмен чрез ДДП. Използва се съответната линия за потвърждаване на заявката за обмен DACK. В зависимост от типа на обмена, зададен в регистъра за режима на дадения канал, данните се предават директно от периферното устройство към системната памет или обратно. За тази цел в работните състояния S3 и S4 контролерът изработва управляващите сигнали за четене IOR и за запис MEMW (или MEMR и IOW), които са активни едновременно. В контролера за ДДП данни постъпват единствено при обмен от типа памет-памет. При този случай в два последователни цикъла на контролера се изработват управляващите сигнали за четене MEMR и за запис MEMW.

Броят на байтовете, които трябва да се прехвърлят до завършване на обмена с ДДП, се съхранява в текущия брояч на байтове на канала. Този брой се намалява с единица автоматично след всяко предаване. Когато съдържанието на текущия брояч премине от 0000 в FFFF, се

генерира сигналът за достигане на гранично състояние на брояча TC. Този сигнал се предава към периферното устройство по линията за край на обмена с ДДП TC на системната шина. В същото време контролерът I8237A-5 премахва сигнала HRQ към управляващата логика на шината. Това предизвиква прекратяване на сигнала HLDA, с което управлението на системната шина се предава на микропроцесора.

В режим на единичен обмен контролерът премахва сигнала HRQ и освобождава системната шина в края на всеки цикъл за обмен чрез ДДП. За въвеждане на нов цикъл сигналът HRQ се формира повторно и се изчаква получаването на сигнала потвърждение HLDA. Това осигурява между циклите на контролера поне един цикъл на микропроцесора за управление на системната шина.

Нормалната продължителност на цикъла за обмен чрез ДДП на контролера I8237A-5 е 5 такта ($1.05 \mu\text{s}$), отговарящи на работните му състояния S0-S4. При необходимост от повече време за извършване на обмена, между тактовете S3 и S4 се въвеждат тактове на изчакване Sw, всеки с продължителност 210 ns. За тази цел се използва входната линия за готовност READY на контролера. В персоналните компютри се извършва удължаване с 1 такт (210 ns) на всички цикли на обмен чрез ДДП, с изключение на цикъла за опресняване на оперативната памет.

Регистри и режими на работа на контролера за ДДП

Всеки канал на контролера има 16-разреден регистър за текущ адрес, който се използва за адресирането на паметта. След всеки цикъл адресът се увеличава или намалява автоматично в зависимост от бит 5 на регистъра за режима. Микропроцесорът извършва запис или четене от този регистър чрез последователни обръщения към младшия и към старшия му байт. В режим на автоинициализация при завършване на обмена с ДДП или при прекратяването му по външен сигнал $\overline{\text{EOP}}$, в регистъра за текущ адрес се възобновява началното му съдържание.

Текущият брояч на байтовете на всеки канал е 16-разреден. Той съхранява броя на байтовете, които трябва да се предадат до края на обмена с ДДП. След всяко предаване броят на байтовете се намалява с единица автоматично. Когато съдържанието на текущия брояч премине от 0000 в FFFF, генерира се сигнал за достигане на гранично състояние на брояча TC. Микропроцесорът извършва запис или четене от този брояч чрез последователни обръщения към младшия и към старшия му байт. В режим на автоинициализация при завършване на обмена с ДДП или при прекратяването му по външен сигнал $\overline{\text{EOP}}$, в текущия брояч на байтове се възобновява началното му съдържание. Без автоинициализиране след завършване на обмена с ДДП съдържанието на текущия брояч е FFFF.

Всеки канал има регистър за начален адрес и регистър за общ брой на байтовете. Тези 16-разредни регистри съхраняват началното съдържание на съответните „текущи“ регистри и се използват при автоинициализацията. Записът в тях се извършва от микропроцесора едновременно със записа в текущите регистри, но не е предвидена възможност за прочитане на съдържанието им.

Управляващият регистър на контролера е 8-разреден и се зарежда от микропроцесора. По апаратния сигнал за начално установяване RESET

или при изпълнение на специалната програмна команда *Начално установяване* регистърът се нулира. Управляващата функция на отделните битове на регистъра е посочена в табл 3.9.

При обмен от тип *памет-памет* се осигурява прехвърляне на блокове от данни от една област на системната памет в друга за минимално време. Този тип обмен се задава чрез установяване на бит 0 на управляващия регистър. Четенето от областта източник се извършва по адреси, изработени от канал 0, а записът в областта назначение — по адреси, изработени от канал 1. За изпълнението на тези операции във всеки цикъл контролерът формира управляващите сигнали *MEMR* и *MEMW*, а прехвърляният байт данни се съхранява временно във вътрешен буферен регистър. Обменът започва чрез програмно установяване на заявка *DREQ* за канал 0 в регистъра за заявките. Предаването на данни протича в режим на обмен по блокове и продължава до възникване на състояние *ТС* в първи канал или до получаване на външен сигнал *ЕОР*. *Модифицирането на адреса за областта източник* може да бъде забранено чрез установяване в 1 на бит 1 на управляващия регистър. В този случай във всички клетки на областта назначение ще бъде записана една и съща информация. Обменът от тип *памет-памет* може да се разпознае по липсата на сигнали по линиите *DACK* в циклите за обмен чрез *ДДП*, включващи изработването на сигнала *AEN*.

Таблица 3.9

Управляващ регистър на 18237А-5

Бит	Състояние	Функция
0	0	Забранява обмен от тип <i>памет-памет</i>
	1	Разрешава обмен от тип <i>памет-памет</i>
1	0	Разрешава модифицирането на адреса в канал 0
	1	Забранява модифицирането на адреса в канал 0
	X	Бит 1 е без значение, ако бит 0 = 0
2	0	Разрешава обмен чрез <i>ДДП</i>
	1	Забранява обмен чрез <i>ДДП</i>
3	0	Сигнали за четене с нормална продължителност
	1	Сигнали за четене с намалена продължителност
	X	Бит 3 е без значение, ако бит 0 = 1
4	0	Фиксиран приоритет на каналите
	1	Ротация на приоритета на каналите
5	0	Сигнал за запис с нормална продължителност
	1	Сигнал за запис с увеличена продължителност
	X	Бит 5 е без значение, ако бит 3 = 1
6	0	Сигналят <i>DREQ</i> е с активно високо ниво
	1	Сигналят <i>DREQ</i> е с активно ниско ниво
7	0	Сигналят <i>DACK</i> е с активно ниско ниво
	1	Сигналят <i>DACK</i> е с активно високо ниво

Приоритетът на каналите се определя в зависимост от бит 4 на управляващия регистър на контролера. Когато този бит е нулиран, приоритетът на каналите е фиксиран: най-висок приоритет има канал 0, а най-нисък — канал 3. Независимо от установения приоритет, когато в даден канал започне обмен чрез *ДДП*, останалите канали изчакват завършването му. Ако разред 4 на управляващия регистър е установен в състояние 1, след всеки обмен се извършва ротация на приоритета на каналите. В персоналните компютри приоритетът на каналите е фиксиран.

Всеки канал има по един 6-разреден регистър за режима. Микропроцесорът програмира режимите на каналите поотделно чрез запис на определена информация, постъпваща по шината за данни DB7-DB0. При това битове DB1, DB0 от данните за запис определят избора на един от регистрите за режима. Управляващата функция на отделните битове на регистрите за режима е посочена в табл.3.10.

При обмен чрез ДДП от тип запис прехвърляните данни се прочитат от регистър на входно-изходната памет и се записват в системната памет по адрес, определен от регистрите на канала. В този случай контролерът изработва управляващите сигнали IOR и MEMW.

При обмен чрез ДДП от тип четене прехвърляните данни се прочитат от системната памет по адрес, определен от регистрите на канала, и се записват в регистър на входно-изходната памет. Контролерът изработва управляващите сигнали MEMR и IOR.

Таблица 3.10

Регистър за режима на I8237A-5

Бит	Състояние	Функция
1,0	0 0	Избор на канал 0
	0 1	Избор на канал 1
	1 0	Избор на канал 2
	1 1	Избор на канал 3
3,2	0 0	Режим на проверка
	0 1	Обмен от тип запис
	1 0	Обмен от тип четене
	1 1	—
X X	X X	Тези битове са без значение, ако битове 7 и 6 съдържат 11
4	0	Забрана на автоинициализацията
	1	Разрешаване на автоинициализацията
5	0	Модификация на адреса с +1
	1	Модификация на адреса с -1
7,6	0 0	Режим на обмен при поискване
	0 1	Режим на единичен обмен
	1 0	Режим на обмен по блокове
	1 1	Каскаден режим

В режим на проверка обмен на данни не се извършва. Осигурява се възможност за проследяване на правилното установяване и модифициране на адресните регистри и на регистрите-бройчи, проверява се и изработването на сигнала за завършване на обмена ТС. Този режим не е възможен при обмен чрез ДДП от тип памет-памет.

Посочените три типа на обмен могат да бъдат задавани във всеки от трите описани по-долу режима на обмен — единичен обмен, обмен по блокове и обмен при поискване.

В режим на единичен обмен при получаване на заявка за обмен чрез ДДП се изпълнява само едно обръщение за четене или запис в паметта. Съдържанието на адресния регистър и на регистъра брояч на байтовете се модифицира с единица след всеки обмен. При преминаване на съдържанието на брояча на байтовете от 0000 в FFFF се генерира сигналът за достигане на граничното му състояние ТС. По този сигнал се извършва автоинициализиране на регистрите на канала, ако това е било зададено

предварително в регистъра за режима.

За да се приеме от контролера, заявката за обмен DREQ трябва да се поддържа активна до получаването на сигнала-потвърждение DACK. В случай че сигналът DREQ се поддържа активен през целия цикъл, в края на цикъла контролерът ще премахне сигнала HRQ и системната шина ще бъде освободена. След това контролерът отново ще издаде сигнал-заявка за заемане на системната шина HRQ и при получаване на сигнала HLDA ще изпълни нов единичен цикъл на обмен. По такъв начин между циклите на контролера за ДДП ще се осигури поне един цикъл на микропроцесора за управление на системната шина.

В персоналните компютри се използва разгледаният режим на единичен обмен чрез ДДП.

В *режим на обмен по блокове* при получаването на заявка за обмен в канала се извършват последователни цикли на обръщение към паметта до възникване на състояние TC или до получаване на външен сигнал за прекратяване на обмена EOP. В края на обмена се извършва автоинициализиране на регистрите на канала, ако това е било зададено предварително в регистъра за режима.

Заявката за обмен DREQ трябва да се поддържа активна, само докато се получи сигналът потвърждение DACK.

В *режим на обмен при поискване* в канала се извършват последователни цикли на обмен чрез ДДП, докато се получи сигналът TC или сигналът EOP или докато сигналът заявка DREQ стане неактивен. Ако обменът с ДДП е прекратен поради премахване на заявката DREQ, той може да бъде възобновен по нова заявка DREQ — т.нар. *при поискване*. При това междинните стойности на адреса за паметта и на брояча на байтове се съхраняват в регистрите на канала. Ако е зададена предварително, автоинициализацията се извършва само по сигналите за край на обмена TC или EOP.

Каскадният режим се прилага при свързване на два и повече контролера за ДДП в многогостъпална система. При него заявката за обмен от даден контролер преминава през схемите за приоритет на предходните стъпала. Наличието на няколко контролера осигурява по-голям брой независими канали за ДДП.

Всеки канал може да бъде програмиран за работа в *режим на автоинициализиране* чрез установяване на бит 4 на регистъра за режима. Автоинициализирането се извършва при генериране на сигнал TC за извършване на обмена с ДДП или при получаване на външен сигнал EOP за прекратяването му. И в двата случая в регистъра за текущ адрес и в текущия брояч на байтове автоматично се възобновява началното им съдържание, съхранявано в регистъра за начален адрес и в регистъра за общ брой на байтовете. При програмирането на контролера всяка от двойките регистри се зарежда от микропроцесора едновременно. По време на обмен чрез ДДП съдържанието на основните (базовите) регистри не се променя. Това позволява след автоинициализирането каналът да бъде готов за нов обмен чрез ДДП без допълнително програмиране от микропроцесора. Обменът се възобновява незабавно след получаването на входен сигнал-заявка DREQ, тъй като в режим на автоинициализация маската на канала остава нулирана.

Контролерът I8237A управлява обмена с ДДП, който се инициира програмно или чрез входен сигнал по една от линиите DREQ. Обменът

по определен канал се задава програмно чрез установяване на съответен бит в 4-разредния регистър за заявките. Тези заявки са немаскируеми. Всеки бит на регистъра се установява или нулира от микропроцесора поотделно чрез запис на информация, достъпваща по шината за данни DB7-DB0. Както е посочено в табл.3.11, чрез битове DB1 и DB0 се избира един от разредите на регистъра на заявките, а установяването или нулирането се определя от бита DB2.

Програмно установяваните заявки се обслужват само ако съответният канал е в режим на обмен по блокове.

Таблица 3.11

Регистър на заявките на I8237A-5

Бит	Състояние	Функция
1,0	0 0	Избор на канал 0
	0 1	Избор на канал 1
	1 0	Избор на канал 2
	1 1	Избор на канал 3
2	0	Нулиране на бита за заявка
	1	Установяване на бита за заявка
7-3	X X X X X	Битове DB7—DB3 са без значение

За задаване на обмен от тип памет-памет трябва да се установи програмно заявка за канал 0.

Отделните битове на регистъра за заявките се нулират при получаване на сигнал TC за завършване на обмена в съответния канал или при външен сигнал EOP за прекратяването му. По сигнала за начално установяване RESET или при изпълнението на специалната програмна команда *Начално установяване* се нулира целият регистър.

Апаратните сигнали заявки за обмен чрез ДДП DREQ могат да се маскират (забраняват) поотделно чрез установяване на съответни битове в 4-разредния регистър за маските.

Микропроцесорът записва информация в регистъра за маските побитово или едновременно във всички разреди. Побитовото установяване или нулиране (табл.3.12) е подобно на запис в регистъра за заявките. То се извършва чрез обръщение по посочения в табл.3.7 адрес 00A от входно-изходното адресно пространство.

Записът във всички разреди на регистъра за маските чрез една инструкция на микропроцесора става чрез обръщение към адрес 00F от входно-изходното адресно пространство. Функцията на отделните битове на предаваните данни по шината DB7-DB0 е посочена в табл.3.13.

По сигнала за начално установяване RESET или при изпълнение на специалната програмна команда *Начално установяване* всички разреди на регистъра се установяват в състояние 1. Чрез друга специална програмна команда — *Нулиране на регистъра за маските* — маските на всички канали се нулират, с което се разрешава приемането на заявки за обмен чрез ДДП DREQ.

Всеки бит за маска се установява в 1, когато в съответния му канал се получи сигналът TC (EOP) и при условие, че каналът не е програмиран за автоинициализиране.

Побитово установяване на регистъра на маските

Бит	Състояние	Функция
1-0	0 0	Избор на канал 0
	0 1	Избор на канал 1
	1 0	Избор на канал 2
	1 1	Избор на канал 3
2	0	Нулиране на бита за маска
	1	Установяване на бита за маска
7-3	X X X X X	Битове DB7—DB3 са без значение

Регистър на маските на I8237A-5

Бит	Състояние	Функция
0	0	Нулиране на бита за маска на канал 0
	1	Установяване на бита за маска на канал 0
1	0	Нулиране на бита за маска на канал 1
	1	Установяване на бита за маска на канал 1
2	0	Нулиране на бита за маска на канал 2
	1	Установяване на бита за маска на канал 2
3	0	Нулиране на бита за маска на канал 3
	1	Установяване на бита за маска на канал 3
7-4	X X X X	Битове DB7—DB4 са без значение

Съдържанието на регистъра за състояние отразява във всеки момент кои канали са завършили обмена с ДДП и в кои канали има т.нар. *висящи заявки за обмен* (табл.3.14). Тази информация може да бъде прочетена от микропроцесора.

Разредите 0-3 на регистъра се установяват в състояние 1, когато в съответните канали се изработи сигналът TC за завършване на обмена с ДДП или се получи външен сигнал EOP за прекратяването му. При всяко прочитане на съдържанието на регистъра от микропроцесора разредите 0-3 се нулират. Разредите 4-7 се установяват в състояние 1, когато в съответните им канали има заявка за обмен чрез ДДП. Целият регистър за състояние се нулира по апаратния сигнал за начално установяване RESET или при изпълнението на специалната програмна команда *Начално установяване*.

Регистър за състояние

Бит	Състояние	Състояние на каналите
0	1	TC — край на обмена в канал 0
1	1	TC — край на обмена в канал 1
2	1	TC — край на обмена в канал 2
3	1	TC — край на обмена в канал 3
4	1	Заявка за обмен в канал 0
5	1	Заявка за обмен в канал 1
6	1	Заявка за обмен в канал 2
7	1	Заявка за обмен в канал 3

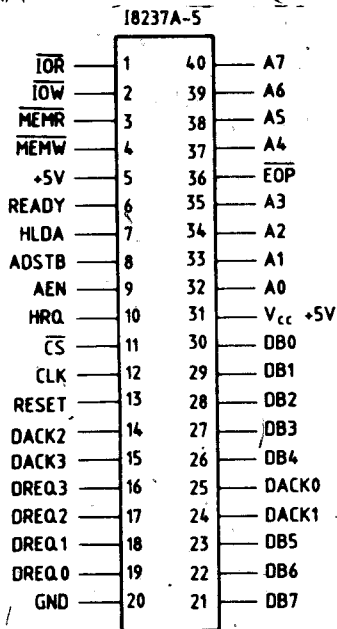
При изпълнение на обмен от тип памет-памет се използва 8-разреден буферен регистър за данни. Във всеки цикъл в него се съхраняват пренасяните данни между двете области на системната памет. След завършване на обмена в регистъра се запазва байтът, прехвърлен в последния цикъл. Тази информация може да бъде прочетена от микропроцесора. Буферният регистър за данни се нулира по апаратния сигнал за начално установяване RESET или при изпълнението на специалната програмна команда *Начално установяване*.

Специалните програмни команди се изпълняват от контролера за ДДП в състояние на програмиране. Те се задават от микропроцесора само чрез определена адресна информация и чрез управляващия сигнал за запис IOW. Информацията по шината за данни в случая е без значение.

Командата *Нулиране на тригера указател на байтове* се използва преди изпълнение на запис или четене от адресните регистри или от регистрите-бройчи. Тази команда нулира тригера, с което се осигурява правилното избиране на младшите и на старшите байтове на посочените регистри.

При командата *Начално установяване* в контролера за ДДП се изпълняват същите действия както при получаването на сигнала за начално установяване RESET. Управляващият регистър, регистърът за състояние, регистърът за заявките, буферният регистър за данни и тригерът-указател на байтове се нулират. Всички тригери на регистъра за маските се установяват в състояние I и контролерът преминава в пасивно състояние.

Командата *Нулиране на регистъра за маските* нулира маските на всички канали, с което се разрешава приемането на заявки за обмен чрез ДДП по линиите DREQ.



Фиг. 3.17. Разположение на изводите на 18237A-5

Сигнали на контролера за ДДП

Разположението на изводите на контролера 18237A-5 е показано на фиг.3.17.

Към извода V_{cc} се подава постоянно захранващо напрежение +5 V.

Изводът GND се свързва към маса.

Сигналят по линията CLK (B) е за синхронизация на вътрешните действия на контролера и влияе върху скоростта на обмена на данни. В персоналните компютри този сигнал се означава с DCLK. Получава се от основния синхросигнал CLK88 за микропроцесора и е също с честота 4,77 MHz, но с коефициент на запълване приблизително 50%.

Сигналят по линията за начално установяване RESET (B) на контролера е с активно високо ниво. Той води до асинхронно нулиране на управляващия регистър, на регистъра за състояние, на

регистъра за заявките, на буферния регистър за данни и на тригера указател на байтове. Всички тригери на регистъра за маските се установяват в състояние 1 и контролерът преминава в пасивно състояние.

Сигналят по линията за готовност **READY (B)** е за удължаване на циклите на контролера. Това осигурява необходимото изчакване при работата с относително по-бавнодействащи устройства. При ниско ниво на входния сигнал **READY** се предизвиква съответно увеличаване на продължителността на импулсите на контролера за четене и запис в паметта. В персоналните компютри чрез сигнала **READY** се извършва удължаване с 1 такт (210 ns) на всички цикли на контролера за ДДП, с изключение на цикъла за опресняване на паметта.

Сигналят по линията за избор на чип **\overline{CS} (B)** е за обръщение към контролера (в пасивното му състояние) като към входно-изходно устройство. Това позволява обмен с микропроцесора по двупосочната шина за данни **DB7-DB0**. Сигналят **CS** е с активно ниско ниво.

Адресните линии **A3-A0 (B/I)** са двупосочни. В пасивно състояние на контролера чрез входните сигнали **A3-A0** се адресират вътрешните му регистри. При обмен чрез ДДП контролерът предава по линиите **A3-A0** най-младшите 4 бита на адреса за паметта. При липса на обмен чрез ДДП изходите **A3-A0** са в състояние на висок импеданс.

Сигналите по адресните линии **A7-A4 (I)** се предават от контролера при формиране на адреса за обръщението към паметта. При липса на обмен чрез ДДП изходите **A7-A4** са в състояние на висок импеданс.

Двупосочните линии за данни **DB7-DB0 (B/I)** са свързани към системната шина. По тях в състояние на програмиране на контролера се осъществява обменът с микропроцесора. Чрез изпълнение на входно-изходни инструкции микропроцесорът прочита съдържанието на вътрешните регистри или програмира действието на контролера. По време на обмен чрез ДДП по тези линии се предават адресните битове **A15-A8**, които се записват във външен регистър по сигнала **ADRSTB**. При обмен от типа памет-памет прочетените данни от паметта първоначално постъпват в контролера по линиите **DB7-DB0**, след което по същите линии се предават за запис в друга клетка на паметта. При липса на обръщение към контролера изходите му **DB7-DB0** са в състояние на висок импеданс.

Сигналите по двупосочната линия за четене от входно-изходната памет **\overline{IOR} (B/I)** са с активно ниско ниво. В състояние на програмиране на контролера вътрешните му регистри се прочитат по входния сигнал **\overline{IOR}** , изработен от микропроцесора. При цикъл за запис в системната памет чрез ДДП информацията за запис се извлича от входно-изходната памет по сигнала **\overline{IOR}** , изработен от контролера. Във всички други случаи изходът **\overline{IOR}** е в състояние на висок импеданс.

Сигналите по двупосочната линия за запис във входно-изходната памет **\overline{IOW} (B/I)** са с активно ниско ниво. В състояние на програмиране на контролера по входния сигнал **\overline{IOW}** , изработен от микропроцесора, се осъществява запис във вътрешните регистри. При цикъл за четене от системната памет чрез ДДП прочетената информация се записва във входно-изходната памет по сигнала **\overline{IOW}** , изработен от контролера. Във всички други случаи изходът **\overline{IOW}** е в състояние на висок импеданс.

Сигналът по линията за четене от паметта $\overline{\text{MEMR}}$ (И) е с активно ниско ниво. Той се използва за четене от избраната клетка на системната памет по време на обмен чрез ДДП от тип четене или от тип памет-памет. Във всички други случаи изходът $\overline{\text{MEMR}}$ е в състояние на висок импеданс.

Сигналът по линията за запис в паметта $\overline{\text{MEMW}}$ (И) е с активно ниско ниво. Той се използва за запис в избраната клетка на системната памет по време на обмен чрез ДДП от тип запис или от тип памет-памет. Във всички други случаи изходът $\overline{\text{MEMW}}$ е в състояние на висок импеданс.

Линиите за заявки за ДДП DREQ0-DREQ3 (В) са входни линии към отделните канали на контролера. По тях периферните устройства изпращат асинхронно сигнали заявки за обмен чрез ДДП. При фиксиран приоритет заявката по линията DREQ0 има най-висок приоритет, а тази по линията DREQ3 — най-нисък. След получаване на сигнала за начално установяване RESET сигналите DREQ0-DREQ3 са с активно високо ниво. Сигналът заявка за ДДП по дадена линия трябва да се поддържа активен до получаването на съответния сигнал за потвърждаване на заявката DACK .

Сигналите по линиите за потвърждаване на заявките за ДДП DACK0-DACK3 (И) се предават към съответните периферни устройства, с които ще бъде изпълнен цикъл за обмен чрез ДДП. След получаване на сигнала за начално установяване RESET сигналите DACK0-DACK3 са с активно ниско ниво.

Сигналът по линията за заемане на системната шина HRQ (И) е заявка от контролера I8237A за поемане на управлението на системната шина и за изпълнение на обмен чрез ДДП. Сигналът HRQ се изработва при заявка за обмен, установена програмно или чрез сигнал по някоя от линиите DREQ . В последния случай съответната маска в контролера трябва да бъде нулирана. Сигналът потвърждение HLDA към контролера трябва да се получи най-рано в следващия такт.

По линията HLDA (В) се получава сигнал потвърждение, че управлението на системната шина се предава на контролера за ДДП. Сигналят HLDA е с активно високо ниво.

При обмен чрез ДДП по линията ADRSTB (И) се разпространява сигнал за запис на адресните битове A15-A8 във външен регистър. Тази адресна информация се предава от контролера по линиите за данни DB7-DB0 . Сигналят ADRSTB е с активно високо ниво.

Изходният сигнал по линията за разрешение на адреса при ДДП AEN (И) показва, че е започнал цикъл с ДДП и системната шина се управлява от контролера I8237A-5 . Сигналят AEN е с активно високо ниво. В персоналните компютри тази линия на контролера не се използва. При тях аналогичният сигнал AEN се изработва от съответна управляваща логика при предаване на управлението на системната шина на контролера за ДДП.

Линията за край на обмена $\overline{\text{EOP}}$ (В/И) е двупосочна. По нея се разпространяват входни и изходни сигнали с активно ниско ниво за завършване на обмена с ДДП. При получаване на външен сигнал на входа $\overline{\text{EOP}}$ контролерът прекратява текущия обмен чрез ДДП. От своя страна

контролерът издава импулс по линията \overline{EOP} , когато в някой от каналите му се достигне гранично състояние ТС на брояча на обменяните байтове. В този случай на нормално завършване на обмена, както и при прекратяването му по външен сигнал, контролерът изпълнява еднотипни действия. Той нулира заявката за обмен в дадения канал и ако е зададена автоинициализация, преписва съдържанието на регистъра за начален адрес и на регистъра за общ брой на байтовете в съответните текущи регистри. Битът ТС за дадения канал в регистъра за състояние и маската на канала се установяват в 1. Ако каналът е програмиран за автоинициализиране, маската остава нулирана. По време на обмен от тип памет-памет контролерът издава импулс за край на обмена при достигане на състояние ТС в първи канал. За предотвратяване на действието на паразитни входни сигнали линията \overline{EOP} се свързва през резистор към +5 V.

3.2.6. СИСТЕМА ЗА ПРЕКЪСВАНЕ

Организация на прекъсванията

Прекъсвания в изпълнението на текущата програма на микропроцесора I8086/8088 могат да се инициират от външни за него устройства, от инструкции за прекъсване и от самия микропроцесор. Персоналните компютри имат проста и универсална система за прекъсване. Микропроцесорът идентифицира отделните типове прекъсвания по присвоените им номера (кодове). Максималният брой на типовете прекъсвания, които той може да обработва, е 256. Обработката на дадено прекъсване се състои в изпълнение на определена служебна (обслужваща) програма.

Външни (апаратни) прекъсвания. Микропроцесорът I8086/8088 има две линии (INTR и NMI), по които външни устройства могат да изпращат сигнали заявки за прекъсвания. В персоналните компютри по линията за маскируеми прекъсвания INTR постъпват сигнали от контролера на прекъсванията I8259A, към който се свързват до 8 устройства, изискващи обслужване. I8259A се управлява програмно от микропроцесора и разширява значително възможностите на системата за прекъсване. Основната му функция е да приема заявки за прекъсвания от свързаните към него устройства, да определя приоритета на постъпилите заявки и при необходимост да изпраща сигнал за прекъсване по линията INTR към микропроцесора. Заявките за прекъсване се приемат по нарастващ фронт на постъпващите към контролера сигнали IR0-IR7 (табл.3.15).

Таблица 3.15

Предназначение на входните линии на контролера I8259A

Линия	Сигнал за прекъсване
IR0	Прекъсване от таймер 0 на I8253-5 за обслужване на часовника за реално време
IR1	Прекъсване от управляващата логика на клавиатурата
IR2	Не се използва
IR3	Прекъсване от адаптера за синхронен последователен интерфейс
IR4	Прекъсване от адаптера за асинхронен последователен интерфейс
IR5	Прекъсване от контролера на ЗУТМД
IR6	Прекъсване от контролера на ЗУГМД
IR7	Прекъсване от адаптера за паралелен интерфейс

Приоритетът им е фиксиран: най-висок приоритет (ниво на прекъсване) има заявката по линията IR0, а най-нисък — тази по линията IR7. Две от линиите се използват в процесорната платка, а останалите шест са свързани към системната шина.

При получаване на сигнал INTR с активно високо ниво микропроцесорът извършва различни действия в зависимост от състоянието на вътрешния си флаг за разрешаване на прекъсванията IF. Когато този флаг е в състояние 0, апаратните прекъсвания, сигнализирани по линията INTR, са маскирани (забранени), микропроцесорът игнорира получената заявка за прекъсване и продължава нормалното изпълнение на програмата си. Сигналят заявка INTR не се записва в микропроцесора, така че трябва да се поддържа активен до приемането му или в противен случай той се отхвърля. Ако флагът IF е в състояние 1, прекъсванията по линията INTR са разрешени. В този случай след изпълнението на текущата инструкция микропроцесорът приема получената заявка за прекъсване и я обработва. Флагът за разрешаване на прекъсванията IF може да се управлява програмно чрез инструкциите STI — установяване на IF в 1, и CLI — нулиране на IF.

Микропроцесорът получава кода на прекъсването от контролера I8259A чрез изпълнение на два последователни цикъла за разпознаване на прекъсването. При първия от тях чрез сигнала INTA, изработен от контролера на шината I8288, се съобщава на контролера на прекъсванията, че заявката INTR е приета. По време на втория цикъл INTA контролерът I8259A изпраща към микропроцесора кода на прекъсването по шината за данни. Микропроцесорът използва този код за извикване на съответната обслужваща програма.

Заявка за външно прекъсване към микропроцесора може да постъпи и по линията за немаскируемо прекъсване NMI. Заявките по тази линия не могат да се маскират (забранят) по програмен път и имат по-висок приоритет от тези по линията INTR. Всеки сигнал с нарастващ фронт по линията NMI се записва в микропроцесора и предизвиква прекъсване в края на изпълнението на текущата инструкция. Немаскируемите апаратни прекъсвания са дефинирани предварително като прекъсвания от тип 2. Затова при възникването им микропроцесорът въвежда съответната обслужваща програма, без да изпълнява цикли за разпознаване на прекъсването.

Програмни прекъсвания. Възникват при изпълнението на инструкциите, разгледани по-долу. При тях кодът на прекъсването се съдържа в самата инструкция или е дефиниран предварително. Затова при възникването им микропроцесорът не изпълнява цикли за разпознаване на прекъсването. Програмните прекъсвания не могат да бъдат маскирани и имат по-висок приоритет от апаратните.

Инструкцията INT (Прекъсване) предизвиква прекъсване незабавно след завършване на изпълнението ѝ. Кодът на типа на прекъсването се съдържа в самата инструкция и се използва от микропроцесора за въвеждане на съответната обслужваща програма. В инструкцията INT може да бъде специфициран всеки от дефинираните апаратни и програмни типове прекъсвания. Тази инструкция се използва за въвеждане на обслужващите програми на базовата система за вход—изход, както и на дефинираните от потребителя прекъсвания.

Прекъсването от тип 3 е дефинирано предварително като точка на прекъсване в изпълняваната програма. То се въвежда чрез инструкцията INT 3 и се използва като средство за настройка на програмите.

Инструкцията INTO (Прекъсване при препълване) предизвиква прекъсване от тип 4 незабавно след завършване на изпълнението ѝ при условие, че флагът за препълване OF е установен в 1.

Вътрешни прекъсвания. Вътрешните прекъсвания възникват в процеса на работата на микропроцесора, генерират се от вътрешната му логика и са немаскируеми.

Микропроцесорът генерира прекъсване от тип 0 незабавно след изпълнението на инструкцията DIV (Деление без знак) или на инструкцията IDIV (Деление на цели числа без знак) при условие, че пресметнатото частно е по-голямо от максималното число, което може да се запише в специфицирания регистър назначение (например при опит за деление на нула).

При условие, че флагът за постъпково изпълнение TF е установен, микропроцесорът генерира прекъсване от тип 1 след изпълнението на всяка инструкция. Този режим се използва като средство за настройка на програмите.

Таблица на векторите на прекъсванията. Връзката между кода (номера) на даден тип прекъсване и съответната обслужваща програма се осъществява чрез таблицата на векторите на прекъсванията. Таблицата заема до 1024 байта от най-младшата област на паметта и съдържа до 256 елемента, по един за всеки тип прекъсване в системата. Всеки елемент на таблицата е с дължина двойна дума (4 байта) и съхранява началния адрес (вектора) на обслужващата програма за даден тип прекъсване: старшата дума задава базовия адрес на сегмента от паметта, съдържащ програмата, а младшата дума — отместването спрямо началото на този сегмент (табл.3.16). Тъй като елементите на таблицата са с дължина 4 байта, микропроцесорът може да пресметне началния адрес на даден елемент чрез умножение на номера на типа на прекъсването по 4. Чрез последователно прочитане на младшата и на старшата дума от елемента на таблицата и чрез зареждането им съответно в брояча на инструкциите IP и в кодovия сегментен регистър CS управлението на микропроцесора се насочва към програмата за обслужване на прекъсването.

Таблица 3.16

Таблица на векторите на прекъсванията

Адреси на паметта	Елемент на таблицата (4 байта)	Вектор
000H, 001H	Отместване на програмата, обслужваща прекъсване 0	Вектор 0 (Прекъсване 0)
002H, 003H	Базов адрес на програмата, обслужваща прекъсване 0	
004H, 005H	Отместване на програмата, обслужваща прекъсване 1	Вектор 1 (Прекъсване 1)
006H, 007H	Базов адрес на програмата, обслужваща прекъсване 1	
...
3FCH, 3FDH	Отместване на програмата, обслужваща прекъсване 255	Вектор 255 (Прекъсване 255)
3FEH, 3FFH	Базов адрес на програмата, обслужваща прекъсване 255	

Описаната структура позволява началният адрес на всяка обслужваща програма в паметта да се специфицира чрез поставяне на адрес (база и отместване) в клетка от таблицата, осигурена за съответния тип прекъсване.

Преди да въведе програма за обработка на прекъсване от даден тип, микропроцесорът влага текущото съдържание на флаговия регистър и на регистрите CS и IP в стека. Автоматично се нулират флагът TF и флагът за разрешение на апаратните прекъсвания IF. Обслужващата програма може да разреши наново маскируемите апаратни прекъсвания чрез инструкцията STI. С това се осигурява възможност самата тя да бъде прекъсната при заявка по линията INTR на микропроцесора. Обслужващата програма може да бъде прекъсната и по програмен път или при заявка по линията за немаскируемо прекъсване NMI на микропроцесора.

Когато е необходимо, обслужващата програма трябва да осигурява съхраняване в стека на текущото в момента на прекъсването съдържание на всички регистри на микропроцесора и възстановяването им преди края на прекъсването. Всяка програма за обработка на прекъсване завършва с инструкцията IRET. При изпълнение на тази инструкция съдържанието на брояча на инструкциите IP, на кодовия сегментния регистър CS и на флаговете се възстановява от стека и по такъв начин управлението се връща към прекъснатата програма.

Програмируем контролер на прекъсванията I8259A

Блоквата схема на контролера на прекъсванията I8259A е показана на фиг.3.18, а разположението на изводите му — на фиг.3.19.

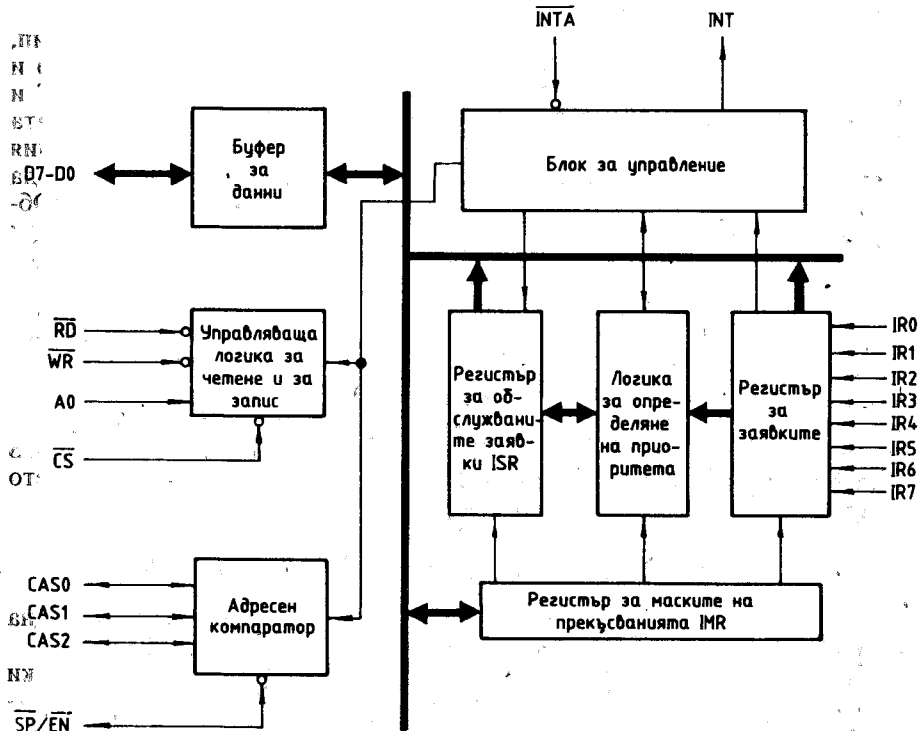
В регистъра за заявките за прекъсванията IRR се съхраняват всички заявки за прекъсвания, постъпващи по линиите IR0-IR7. Заявките се приемат асинхронно. В зависимост от установения режим в контролера разредите на регистъра IRR се установяват по нарастващ фронт или при високо ниво на съответните входни сигнали.

В регистъра на обслужваните заявки ISR се съхраняват всички заявки за прекъсвания, чието обслужване е започнало.

Регистърът за маските на прекъсванията IMR се зарежда програмно и съхранява индивидуалните маски на заявките за прекъсвания, постъпващи по линиите IR0-IR7.

Логиката за определяне на приоритета на прекъсванията определя необходимостта от изработване на сигнал за прекъсване INT към микропроцесора въз основа на съдържанието на регистрите IRR, ISR и IMR. Всеки от разредите на регистъра за заявките за прекъсвания IRR е свързан с определен приоритет, който, в зависимост от установения режим в контролера, може да бъде фиксиран или да се изменя след всяко обрботено прекъсване. Сигналят INT се формира с активно високо ниво след избиране на най-приоритетната немаскирана заявка за прекъсване и постъпва директно на входа INTR на микропроцесора. Ако този сигнал заявка се приеме, към I8259A се изпращат два последователни импулса за разпознаване на прекъсването INTA с активно ниско ниво. По втория от тях контролерът предава към микропроцесора по шината за данни кода на приетото прекъсване. Едновременно с това се установява съответен разред на регистъра за обслужваните заявки ISR. Нова заявка за

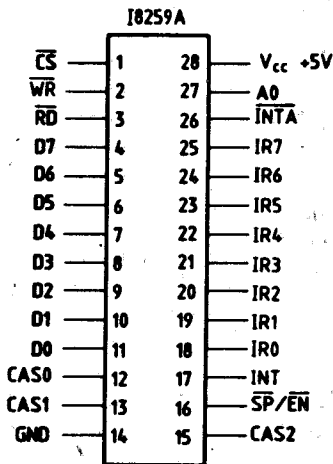
прекъсване към контролера I8259A се приема, в случай че не е маскирана и има по-висок приоритет от обслужваните в момента заявки.



Фиг.3.18. Блокова схема на I8259A

Контролерът на прекъсванията се свързва към системната шина или към шината на микропроцесора чрез вътрешния 8-разреден двупосочен буфер за данни. По линиите D7-D0 се предава управляваща информация към контролера I8259A или информация за състояние и кодове на прекъсвания към микропроцесора. Сигналят CS определя програмно управляваните обръщания за четене или за запис към контролера, докато циклите за разпознаване на прекъсване INTA се изпълняват независимо от този сигнал.

Обръщанията на микропроцесора към I8259A се осъществяват от управляващата логика за четене и за запис. По сигнала за запис WR при активен сигнал CS се извършва зареждане на



Фиг.3.19. Разположение на изводите на I8259A

инициализиращи командни думи ICW и операционни командни думи OCW в съответни регистри на контролера. По сигнала за четене RD при активен сигнал CS към микропроцесора се предава съдържанието на регистрите IRR, ISR и IMR. За избирането на отделните вътрешни регистри на контролера I8259A се използва само една адресна линия A0; като обръщението към тях се извършват в строго определена последователност.

Адресният компаратор се използва при каскадно свързване на няколко контролера I8259A, осигуряващи по-голям брой входни линии за заявки за прекъсвания. При това един от контролерите се дефинира като главен, а останалите — като подчинени. В цикъла за разпознаване на прекъсване главният контролер изпраща по адресните линии CAS2-CAS0 идентификатора на контролера, чиято заявка за прекъсване е приета. Чрез адресните компаратори този идентифициращ адрес се разпознава само в един от контролерите, който предава към микропроцесора кода на прекъсването.

Линията SP/EN на контролера на прекъсванията има двойно предназначение. В режим на използване на външни буфери към шината за данни, към която са свързани каскадно няколко контролера, по тази линия избраният контролер издава сигнал с активно ниско ниво за разрешаване на съответния буфер. В нормален режим на работа при свързване на извода SP/EN към +5 V контролерът се дефинира като главен, а при свързването на извода към маса — като подчинен.

Управляващи думи и режими на работа на контролера I8259A ГО

Действието на контролера на прекъсванията I8259A се програмира чрез два типа управляваща информация, предавана от микропроцесора: инициализиращи командни думи ICW и операционни командни думи OCW. Първите от тях се използват за начално установяване на контролера в определено състояние, а вторите — за управление на операциите в него и за изменение на работните режими. Вътрешните регистри на контролера заемат два адреса от входно-изходното адресно пространство на компютъра. За извършване на обръщението към тях микропроцесорът изпълнява входно-изходни инструкции. При активни сигнали CS и WR командните думи ICW и OCW се предават за запис в съответни вътрешни регистри на I8259A по шината за данни D7-D0. Контролерът различава отделните видове командни думи по състоянието на адресния сигнал A0, по последователността, в която те се предават, и по някои техни предварително специфицирани битове. В дадените по-долу формати на ICW и OCW и в табл.3.17 тези битове са посочени с фиксираните си стойности — 0 или 1.

Инициализиращи командни думи. Установяването на необходимото начално състояние и на режимите на работа на контролера на прекъсванията се извършва чрез запис на последователност от два до четири бита, наречени инициализиращи командни думи ICW. При задаване от микропроцесора на командната дума ICW1 (табл.3.17) в контролера I8259A се стартира последователност от действия, свързани с установяването, по време на което се изпълнява автоматично следното:

- установява се специална логика за управление на последователността на запис в останалите командни думи ICW (фиг.3.20);
- регистрите ISR и IMR се нулират;

Основни операции в I8259A

ОТ	D4	D3	RD	WR	CS	Операция
ICW0	0	0	1	0	0	Запис в OCW2
ICW1	0	1	1	0	0	Запис в OCW3
ICW2	1	X	1	0	0	Запис в ICW1
ICW3	X	X	1	0	0	Запис в OCW1, ICW2, ICW3 и ICW4 ¹
ICW4			0	1	0	Четене от IRR или от ISR ²
ICW5			0	1	0	Четене от IMR
ICW6	X	X	1	1	0	— (Вътрешният буфер е в състояние
ICW7	X	X	X	X	1	— на висок импеданс)

¹ Последователността за запис в тези регистри се определя от специална вътрешна логика.

² Избирането на IRR или на ISR се определя в зависимост от предварително установено съдържание на OCW3.

— премахват се евентуално установените преди това режими за специално маскиране на прекъсванията и за ротация на приоритетите;

— за бъдещо четене по общия адрес на двойката регистри IRR/ISR се избира регистърът IRR;

— при ICW1(0)=0 всички функции, задавани от командната дума ICW4, се нулират;

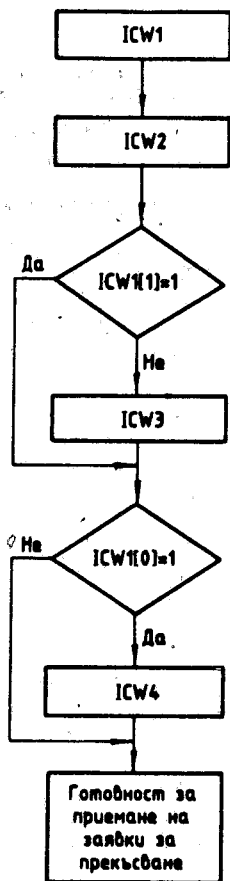
— въвежда се режим на влагане на прекъсванията едно в друго, т.е. възможност за ново прекъсване по време на процедура на обработване на прекъсване, при което първоначално най-висок приоритет има заявката за прекъсване по линията IR0, а най-нисък — тази по линията IR7;

— установява се режим на приемане на заявките за прекъсвания по нарастващ фронт на съответните сигнали;

— адресът на контролера, дефиниран като подчинен, се установява равен на 7.

Както е посочено на фиг.3.20, командите ICW1 и ICW2 се задават от микропроцесора винаги, докато ICW3 и ICW4 са необходими само ако това е указано в ICW1. След инициализиране на контролера изменения в командните думи ICW са възможни само при изпълнение на цялата последователност по програмирането им.

Форматите на инициализиращите командни думи са посочени в табл.3.18—3.21.



Фиг.3.20. Последователност на инициализиране на I8259A

Таблица 3.18

Командна дума ICW1 (A0=0; D4=1)

Бит	Състояние	Функция
7,6,5		Използват се само в режим I8080/85 (вж. ICW4)
4	1	Дефиниране на ICW1
3	0	Заявките за прекъсвания се приемат по нарастващ фронт на съответните сигнали IR0-IR7
	1	Заявките за прекъсвания се приемат при високо ниво на съответните сигнали IR0-IR7
2		Използва се само в режим I8080/85
1	0	Режим на каскадно свързани контролери I8259A. Необходима е и командна дума ICW3
	1	Режим на единичен контролер I8259A
0	0	Не е необходима командна дума ICW4
	1	Необходима е командна дума ICW4

Командна дума ICW2 (A0=1)

Таблица 3.19

Бит	Състояние	Функция
7-3		Определят старшата част T7-T3 на номера на прекъсването. (Младшата част съответства на номера i на входната линия IRi на контролера)
2-0		Използват се само в режим I8080/85

Командна дума ICW3 (A0=1)

Таблица 3.20

Контролер	Бит	Състояние	Функция
Главен	i	0	Към съответната входна линия IRi не е свързан подчинен контролер I8259A
		1	Към съответната входна линия IRi е свързан подчинен контролер I8259A
Подчинен	7-3	00000	Дефинират номера на подчинения контролер
	2-0	XXX	

Командна дума ICW4 (A0=1; D7-D5=000)

Таблица 3.21

Бит	Състояние	Функция
7-5	000	Дефиниране на ICW4
4	0	Режим на влагане на прекъсванията
	1	Режим на специално влагане на прекъсванията (не се използва персоналните компютри)
3,2	0X	Работа на контролера без външен буфер
	10	Използване на външен буфер за подчинен контролер
1	11	Използване на външен буфер за главен контролер
	0	Нормално завършване на прекъсване — чрез команда EOI за контролера (вж. OCW2)
0	1	Автоматично завършване на прекъсване — без команда EOI за контролера
	0	Режим 8080/85 (свързване към микропроцесор I8080/85)
	1	Режим 8086/88 (свързване към микропроцесор I8086/88)

81 **Операционни командни думи.** Операционните командни думи OCW са за управление на работата и за изменение на режимите на контролера на прекъсванията. Те не изискват определена последователност при програмирането. Форматите им са посочени в табл.3.22—3.24.

Таблица 3.22

Командна дума OCW1 (A0=1)

Бит	Състояние	Функция
i	0	Нулиране на маската на прекъсването по линията IRi
-D _i	1	Установяване на маска на прекъсването по линията IRi

Таблица 3.23

Командна дума OCW2 (A0=0; D4,D3=00)

Бит	Състояние	Функция
7-5	001	Команда EOI — край на прекъсване (нулиране на бита ISRi за съответното обработено прекъсване)
-D _i	0	← (липса на операция) Останалите команди (режими) не се използват в персоналните компютри
-ЭГ	010	

Таблица 3.24

Командна дума OCW3 (A0=0; D4,D3=01)

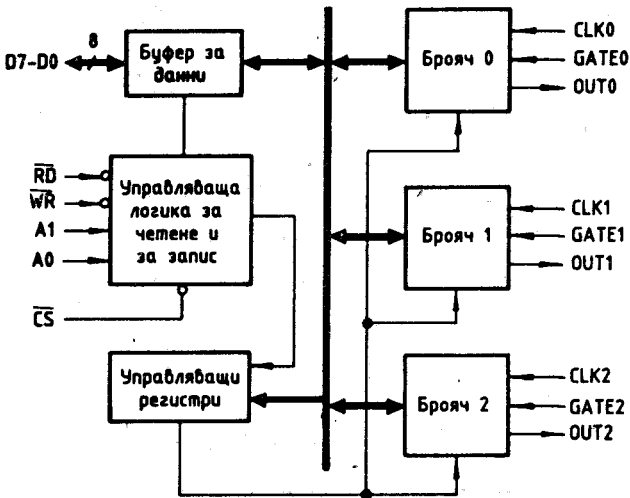
Бит	Състояние	Функция
7	X	Този бит е без значение
6,5,2		Не се използват в персоналните компютри
4,3	01	Дефиниране на OCW3
1,0	10	Избиране на регистър IRR за следващо четене
	11	Избиране на регистър ISR за следващо четене
	0X	— (липса на операция)

3.2.7. ТАЙМЕРИ НА СИСТЕМНАТА ПЛАТКА

За измерване на различни интервали от време чрез програмно управление и за осигуряване на нормалната работа на компютъра към системната шина е включена интегралната схема I8253-5. Тя съдържа три независими интервални таймера, всеки от които може да бъде програмиран от микропроцесора индивидуално за работа в различни режими. Блоквата схема на I8253-5 е показана на фиг.3.21, а разположението на изходите ѝ — на фиг.3.22. Свързването към системната шина за данни D7-D0 остава чрез 8-разреден вътрешен двупосочен буфер.

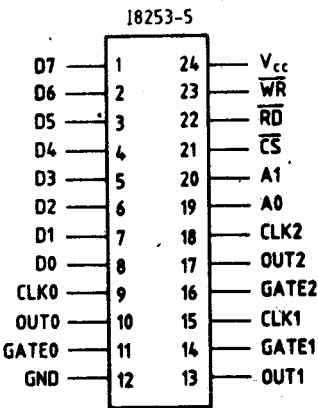
Всеки от таймерите е изграден от 16-разреден брояч и от 6-разреден управляващ регистър. Осигурени са независими входове за тактови сигнали CLK (с максимална честота 2 MHz), входове за управляващи сигнали GATE и изходи OUT. Вътрешните регистри на интегралната схема

заемат четири адреса от входно-изходното адресно пространство на компютъра. За обръщения към тях микропроцесорът изпълнява входно-изходни инструкции.



Фиг.3.21. Блокова схема на I8253-5

Отделните броячи се избират чрез адресните сигнали A1, A0 (табл.3.25), а записът или четенето се управляват съответно от сигнала IOW или IOR. Сигналят CS определя обръщението на микропроцесора към I8253-5, но не влияе на работата на броячите. В управляващите регистри е възможен само запис. Те се избират при адресните битове A1, A0 = 11 и в зависимост от старшите два бита на данните за запис (табл.3.26). При неактивен сигнал CS или липса на входно-изходна операция (WR = 1, RD = 1) вътрешният буфер на I8253-5 е в състояние на висок импеданс.



Фиг.3.22. Разположение на изводите на I8253-5

Таймерите се програмират индивидуално чрез запис на определена информация в съответните броячи и управляващи регистри. Функцията на отделните битове на управляващите регистри е посочена в табл.3.26. Всеки от броячите може да бъде настроен като двоичен или като двоично-десетичен брояч в един от шест възможни работни режима. Чрез битове 5 и 4 на управляващия регистър може да бъде зададено функционирането само на младшия байт, само на старшия байт или и на двата байта на даден брояч. В последния случай за правилната работа на брояча са необходими винаги две обръщени към него за запис или за четене, при посочената в табл.3.26 последователност. В съответствие с това инициализирането на всеки брояч се извършва чрез зареждане на един или на два байта. Всички броя-

чи са реверсивни. Зареждането на нули задава максималната стойност за броеве 2^{16} при двоичен брояч и 10^4 при двоично-десетичен брояч.

Основни операции с I8253-5

Таблица 3.25

IOR	IOW	CS	A1,A0	Операция
1	0	0	0 0	Зареждане на брояч 0
1	0	0	0 1	Зареждане на брояч 1
1	0	0	1 0	Зареждане на брояч 2
1	0	0	1 1	Запис в управляващия регистър
0	1	0	0 0	Четене на брояч 0
0	1	0	0 1	Четене на брояч 1
0	1	0	1 0	Четене на брояч 2
0	1	0	1 1	—
X	X	1	X X	—
1	1	0	X X	—

Управляващ регистър на I8253-5

Таблица 3.26

Бит	Състояние	Функция
0A	7,6	0 0 Избор на брояч 0
вп.		0 1 Избор на брояч 1
вс		1 0 Избор на брояч 2
-вс		1 1 —
9B	5,4	0 0 Запис на текущата стойност на брояча
с		0 1 Зареждане/четене само на младшия байт на брояча
		1 0 Зареждане/четене само на старшия байт на брояча
		1 1 Зареждане/четене на младшия байт, след това на старшия байт на брояча
	3,2,1	0 0 0 Установяване на режим 0
		0 0 1 Установяване на режим 1
		X 1 0 Установяване на режим 2
		X 1 1 Установяване на режим 3
		1 0 0 Установяване на режим 4
		1 0 1 Установяване на режим 5
вп.	0	0 Конфигуриране в двоичен брояч (16 разреда)
вс		1 Конфигуриране в двоично-десетичен брояч (4 тетради)

вп. За четене на текущата стойност на броячите са възможни два начина. При първия от тях се използва обикновена инструкция за четене от входно-изходен регистър, като изборът на даден брояч се определя от адресните битове A1 и A0. За да се осигурят стабилни данни при четенето, работата на избрания брояч трябва да се забрани чрез управляващия сигнал на входа GATE или чрез прекратяване на тактовия сигнал CLK, което не винаги е удобно.

При втория начин текущата стойност на даден брояч се прочита, без да се нарушава действието му. За тази цел в I8253-5 е предвиден специален вътрешен регистър. Той приема съдържанието на брояча, след което осигурява стабилни данни за четене по системната шина. Записът на те-

кущата стойност на брояча се извършва чрез установяване на определена информация в управляващия регистър. Както е показано в табл.3.26, битове 5,4 трябва да се нулират, а чрез битове 7,6 — да се избере желаният брояч. В случая битове 3—0 са без значение. Съдържанието на вътрешния регистър се прочита чрез инструкция за четене на избрания брояч. При работа с двубайтов брояч чрез две последователни инструкции за четене най-напред се прочита младшият байт, след това — старшият байт. За осигуряване на по-нататъшната правилна работа на брояча тези две обръщения са задължителни.

В персоналните компютри се използват два от работните режими на таймерите, означавани, съгласно табл.3.26 и фирмените описания на I8253-5, с режим 2 и режим 3.

След установяване на режим 2 таймерът работи като брояч делител на честота. Времедиagramата на сигналите му в този режим е показана на фиг.3.23. Изходният сигнал OUT е с ниско ниво в продължение на един период на тактовия сигнал CLK. Периодът на повторение на изходните импулси се определя от предварително зареденото съдържание на брояча — коефициента на делене n . Ново зареждане на брояча в интервала от време между два изходни импулса не изменя текущия цикъл на броене, а се отразява в следващите цикли.

При ниско ниво на управляващия сигнал, подаван на входа GATE, изходният сигнал се поддържа с високо ниво. Когато управляващият сигнал стане активен (с високо ниво), броенето се инициира от първоначалната стойност. По такъв начин, управляващият сигнал може да се използва за синхронизиране на работата на брояча.

В процеса на установяването на режим 2 изходният сигнал се поддържа с високо ниво до зареждането на брояча. Следователно действието на брояча може да се синхронизира и програмно.

В режим 3 таймерът работи като генератор на правоъгълни импулси. Времедиagramата на сигналите му в този режим е показана на фиг.3.24. Изходният сигнал е с високо ниво до отброяване на половината от първоначално заредената стойност (четно число) в брояча, след което до завършване на броенето — се поддържа с ниско ниво. При достигане на граничната стойност на брояча нивото на изходния сигнал се променя, броячът се зарежда наново с първоначалната стойност и цикълът се повтаря. Ако първоначално заредената стойност в брояча е нечетно число n , във всеки цикъл изходният сигнал ще бъде с високо ниво в продължение на $(n+1):2$ такта, а с ниско ниво — $(n-1):2$ такта.

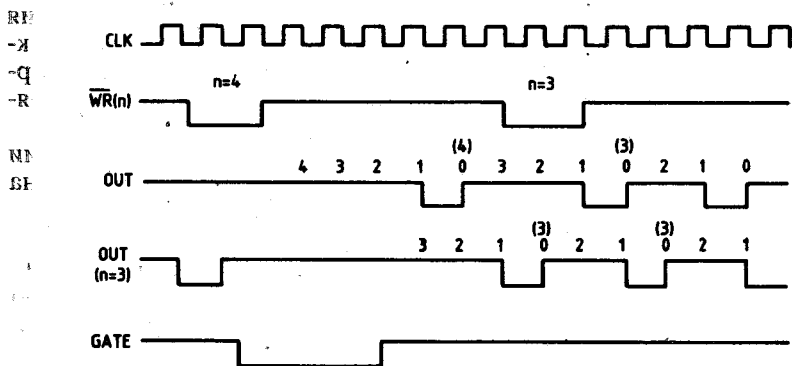
В разгледания режим възможностите за синхронизиране на работата на брояча са подобни на тези в режим 2.

В табл.3.27 е обобщено действието на управляващия сигнал на входа GATE върху таймера, установен в режим 2 и, в режим 3.

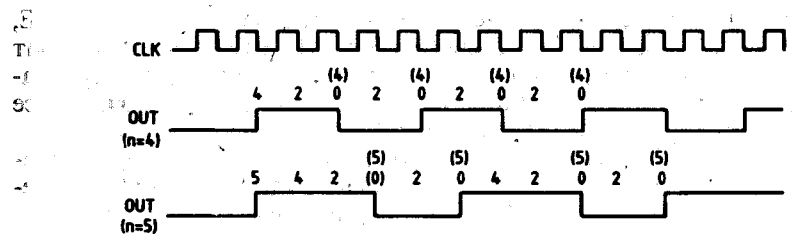
В персоналните компютри за трите таймера на I8253-5 — таймер 0, таймер 1 и таймер 2 — се използва тактов сигнал с честота 1,19318 MHz. Той се получава от синхросигнала PCLK чрез разделяне на честотата му на две. Таймер 0 и таймер 1 се програмират за работа в режим 2. Действието им е разрешено винаги, тъй като управляващите входове GATE са свързани към +5 V.

Таймер 0 е системен таймер с общо предназначение. Той осигурява изграждането на часовник за реално време и се използва от системните програми. Изходът му е свързан към линията IRQ0 на контролера на

прекъсванията I8259A. Таймерът се настройва за генериране на сигнал заявка за прекъсване всеки 54,925 ms. Програмата за обслужване на прекъсването обновява часовника за реално време.



Фиг.3.23. Времениаграма на сигналите на таймера в режим 2



Фиг.3.24. Времениаграма на сигналите на таймера в режим 3

Таблица 3.27
Действие на управляващия сигнал на входа GATE на I8253-5

Режим	Управляващ сигнал		
	ниско ниво	нарастащ фронт	високо ниво
Режим 2	Забранява броенето Установява изходен сигнал с високо ниво	Установява първоначална стойност в брояча Инициира броенето	Разрешава броенето
Режим 3	Забранява броенето Установява изходен сигнал с високо ниво	Инициира броенето	Разрешава броенето

Таймер 1 е част от апаратурата, извършваща опресняването на оперативната памет. Изходът му е свързан към линията DREQ0 на контролера за директен достъп до паметта I8237A-5. Таймерът се настройва за генериране на сигнал заявка за обмен с ДДП всеки 15,12 μ s. Извършване-

то на обмена от канал 0 на контролера I8237A-5 осигурява опресняването на динамичната памет.

Таймер 2 се използва за управление на вградения в системното устройство високоговорител. За тази цел той се програмира (в работен режим 3) за генериране на изходен сигнал с определена звукова честота. Едновременно с това сигналът към високоговорителя се управлява и чрез зареждане на подходяща информация в изходния регистър — порт В на I8255A-5. Разред 0 на този регистър е свързан към управляващия вход на таймер 2, а разред 1 определя предаването или блокирането на изходния сигнал към високоговорителя.

3.2.8. ПРОГРАМИРУЕМ ПАРАЛЕЛЕН ИНТЕРФЕЙС

Към системната шина на персоналните компютри е включена интегралната схема I8255A-5. Тя е проектирана за осигуряване на програмируем паралелен интерфейс за различни периферни устройства чрез 24 входно-изходни линии при три основни режима на работа. В разглежданите компютри интегралната схема се използва за свързване на управляващата логика на клавиатурата към системната шина и за задаване на различни управляващи функции в процесорната платка. Блоквата схема на I8255A-5 е показана на фиг.3.25, а разположението на изходите ѝ — на фиг.3.26. Свързването на интегралната схема към системната шина за данни D7-D0 се осъществява чрез вътрешен двупосочен буфер. I8255A-5 съдържа три 8-разредни порта (входно-изходни регистъра) — PA, PB и PC, всеки от които може да бъде програмиран от микропроцесора отделно за работа в различни режими и в различни конфигурации. Възможностите за конфигуриране на портовете на I8255-A са следните:

- Порт А — един 8-разреден изходен регистър с буфер за данни или един 8-разреден входен регистър.
Порт В — един 8-разреден входно-изходен регистър с буфер за данни или един 8-разреден входен буфер.
Порт С — един 8-разреден изходен регистър с буфер за данни или един 8-разреден входен буфер. Порт С може да се раздели на два 4-разредни порта, програмирани независимо.

Логиката за управление на конфигурирането и на режимите на работа на портовете е разделена в два блока (фиг.3.25):

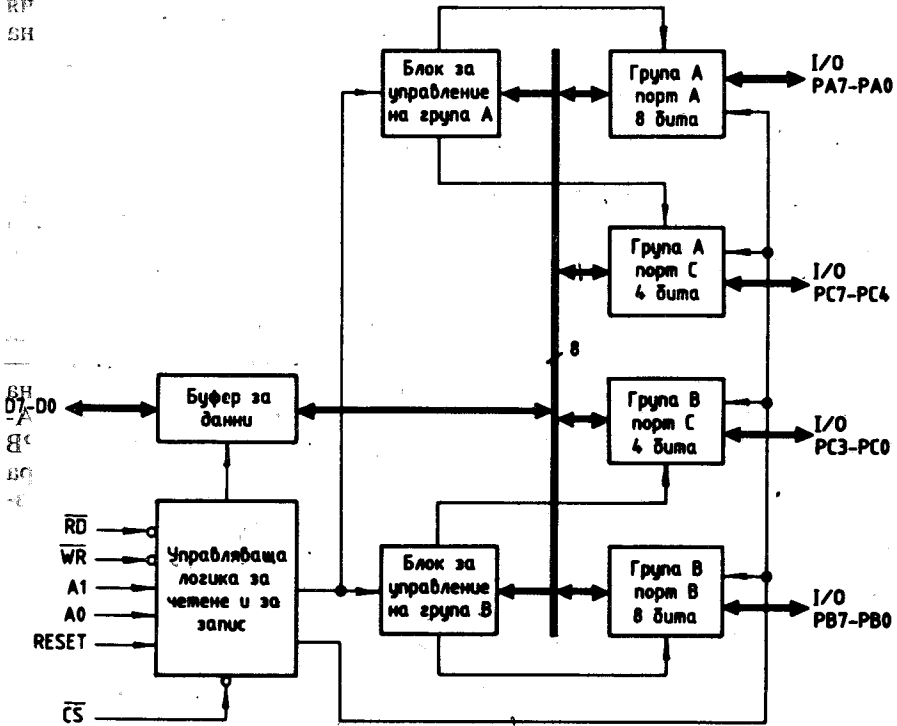
- блок за управление на портовете в група А — PA и PC7-PC4;
- блок за управление на портовете в група В — PB и PC3-PC0.

Функционирането на портовете се програмира чрез запис на определена информация във вътрешен управляващ регистър, свързан с тези блокове. Портовете и управляващият регистър заемат четири адреса от входно-изходното адресно пространство на компютъра. За обръщения към тях микропроцесорът изпълнява входно-изходни инструкции. Отделните регистри на I8255A-5 се избират чрез адресните сигнали A1, A0 (табл.3.28), а записът или четенето се управлява съответно от сигнала \overline{IOW} или \overline{IOR} . В управляващия регистър е възможен само запис. Сигналът \overline{CS} определя обръщението на микропроцесора към I8255A-5. При неактивен сигнал \overline{CS} или липса на входно-изходна операция ($\overline{WR}=1$ и $\overline{RD}=1$) вътрешният буфер на интегралната схема е в състояние на висок

импеданс. Сигналят RESET нулира управляващия регистър и конфигурира всички портове като входове.

Функцията на отделните битове на управляващия регистър е посочена в табл.3.29. Както се вижда от таблицата и от фиг.3.25, режимите за RA и за PB се дефинират независимо един от друг, но фиксират и работата на всяка от частите на PC: за PC7-PC4 се задава винаги режимът на RA, а за PC3-PC0 — режимът на PB.

RF
BH



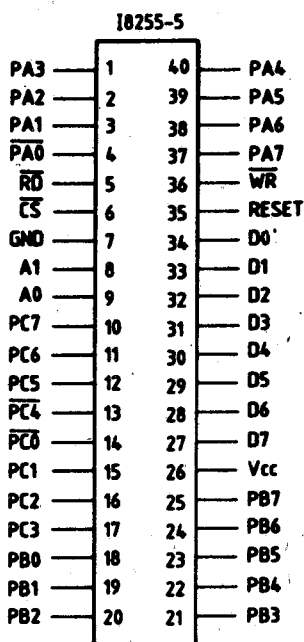
Фиг.3.25. Блокова схема на I8255A-5

Таблица 3.28

Основни операции с I8255A-5

A1	A0	\overline{WR}	\overline{RD}	\overline{CS}	Операция
0	0	0	1	0	Запис в RA
0	1	0	1	0	Запис в PB
1	0	0	1	0	Запис в PC
1	1	0	1	0	Запис в управляващия регистър
0	0	1	0	0	Четене от RA
0	1	1	0	0	Четене от PB
1	0	1	0	0	Четене от PC
1	1	1	0	0	—
X	X	X	X	1	—
X	X	1	1	0	—

В системната платка на персоналните компютри и трите порта на I8255A-5 се програмират за работа в т.нар. *основен режим* — *режим 0*. В



Фиг.3.26. Разположение на изходите на I8255A-5

този режим интегралната схема осигурява два 8-разредни и два 4-разредни порта с просто входно или изходно действие (работа без определен протокол). Всеки от портовете може да бъде дефиниран отделно като изход или като вход, при което изходната информация се записва в съответен регистър, а за постъпващите данни е осигурен само буфер.

По-долу е описано предназначението на портовете на I8255A-5 в системната платка на персоналните компютри.

След установяване на системата в начално състояние порт А се конфигурира като вход, към който постъпват преобразуваните в паралелен код данни от клавиатурата. При изпълнение на инструкция за четене от входно-изходен регистър по адреса на порт А тези данни се предават по системната шина към микропроцесора за по-нататъшна обработка.

По време на изпълнението на т.нар. вътрешен тест на системата порт А се конфигурира като изход, в който последователно се записват различни константи, идентифициращи отделните секции на този тест.

Таблица 3.29

Управляващ регистър на I8255A-5

Бит	Състояние	Функция
1	2	3
7	1	Установяване на режимите на портовете <i>Портове от група В</i>
0	0	Конфигуриране на PC3-0 като изход
	1	Конфигуриране на PC3-0 като вход
1	0	Конфигуриране на PB като изход
	1	Конфигуриране на PB като вход
2	0	Установяване на режим 0 за PB и PC3-0
	1	Установяване на режим 1 за PB и PC3-0 <i>Портове от група А</i>
3	0	Конфигуриране на PC7-4 като изход
	1	Конфигуриране на PC7-4 като вход
4	0	Конфигуриране на PA като изход
	1	Конфигуриране на PA като вход
6,5	00	Установяване на режим 0 за PA и PC7-4
	01	Установяване на режим 1 за PA и PC7-4
	1X	Установяване на режим 2 за PA и PC7-4

БИТ	1	2	3
PC	7	0	Установяване/нулиране на отделните разреди на PC (само за режими 2 и 3)
TO	6,5,4	XXX	(В случая тези битове са без значение)
-TO	3,2,1	000	Избиране на разред 0 на PC
-SO		001	Избиране на разред 1 на PC
-PC		:	:
-IB		111	Избиране на разред 7 на PC
	0	0	Нулиране на избрания разред на PC
		1	Установяване на избрания разред на PC

Порт В се конфигурира като изходен регистър. Чрез запис на определена информация в него микропроцесорът задава различни управляващи функции в системната платка. Предназначението на отделните разреди на РВ е посочено в табл.3.30.

Таблица 3.30

Порт В на I8255A-5

БИТ	Състояние	Функция	
PC	0	Забранява действието на таймер 2 на I8253-5	
SO	1	Разрешава действието на таймер 2 на I8253-5	
-O	1	0	Забранява предаването на сигнал към високоговорителя
NT	1	1	Разрешава предаването на сигнал към високоговорителя
-O*	2	X	
	3*	0	Разрешава прочитане на състоянието на микропревключватели 1-4 на системната платка чрез PC0-PC3
		1	Разрешава прочитане на състоянието на микропревключватели 5-8 на системната платка чрез PC0-PC3
	4*	0	Разрешава действието на схемите за контрол на паметта в системната платка
		1	Забранява действието на схемите за контрол на паметта в системната платка
	5	0	Разрешава действието на схемите за контрол на паметта в модулите за разширение
		1	Забранява действието на схемите за контрол на паметта в модулите за разширение
	6	0	Забранява тактовия сигнал от клавиатурата
		1	Разрешава тактовия сигнал от клавиатурата
	7	0	Разрешава действието на управляващата логика на клавиатурата
		1	Забранява действието на управляващата логика на клавиатурата

* В някои от компютрите тези разреди не се използват.

Порт С се конфигурира като вход, към отделните разреди на който постъпва различна информация, в съответствие с посоченото в табл.3.31. При изпълнение на входно-изходна инструкция за четене по адреса на порт С тази информация се предава по системната шина към микропроцесора.

Порт С на I8255A-5

Разред	Постъпваща информация.
0-3*	Състоянието на микропревключватели 1—4 (при $PB3=0$) или на микропревключватели 5—8 (при $PB3=1$), определящи системната конфигурация
4*	Сигналят за управление на високоговорителя
5	Сигналят от изхода на таймер 2 на I8253-45
6	Сигналят за грешка в паметта в модулите за разширение
7*	Сигналят за грешка в паметта в системната платка

* В някои от компютрите тези разреди не се използват.

3.3. БАЗОВА СИСТЕМА ЗА ВХОД-ИЗХОД

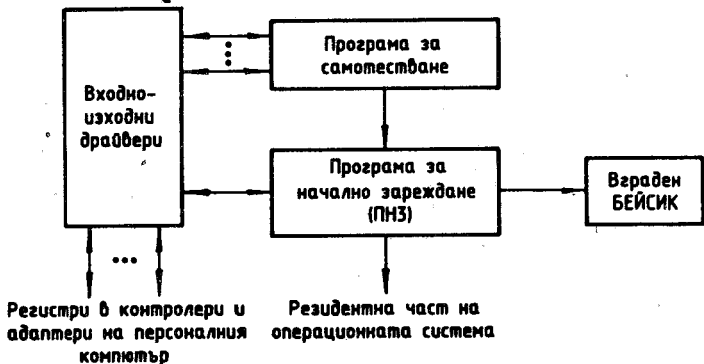
3.3.1. ОСНОВНИ ХАРАКТЕРИСТИКИ

Основните функции на базовата система за вход-изход¹ (БСВИ) са:
 — да управлява основния набор от периферни устройства, като осигурява програмния интерфейс към тях за операционните системи на персоналния компютър;

— да осъществява бърза проверка на работоспособността на компютъра при включване на захранването му или при повторно стартиране на операционната система;

— да задава началните параметри на апаратните модули на компютъра.

Както се вижда от фиг.3.27, основните елементи на БСВИ са входно-изходните драйвери, програмата за самотестване, програмата за начално зареждане (ПНЗ) и вграденият БЕЙСИК.



Фиг.3.27. Обобщена блокова схема на базовата система за вход-изход

¹ Basic Input Output System (BIOS).

Базовата система за вход-изход осъществява, от една страна, програмното управление на апаратните средства (функционалните модули) на системата, а от друга — реализира програмния интерфейс за операционната система. За тази цел БСВИ създава и използва картата на паметта, картата на входно-изходните портове, разпределението на каналите за директен достъп, разпределението на програмните и апаратните прекъсвания. Списък на прекъсванията, установявани от една стандартна версия на БСВИ за компютър, съвместим с IBM PC/XT, е даден в табл. 3.32. Някои персонални компютри имат други адреси на драйверите в БСВИ.

Таблица 3.32

Списък на прекъсванията, установявани от БСВИ на компютър, съвместим с IBM PC/XT

Номер (или) на прекъсването	Наименование (предназначение) на прекъсването	Идентификатор и адрес на драйвера в БСВИ
1	2	3
0	Деление на нула	няма
1	Единична стъпка	няма
2	Немаскируемо прекъсване	NMI_INT (F000:E2C3)
3	Контролна точка	няма
4	Препъване	няма
5	Разпечатване на екрана	PRINT_SCREEN (F000:FF54)
6	Не се използва	—
7	Не се използва	—
Вектори за прекъсване на 8259A		
8	Календарен часовник	TIMER_INT (F000:FEA5)
9	Клавиатурен тракт	KB_INT (F000:E987)
A	Не се използва	—
B	Последователен интерфейс	не се задава от БСВИ
C	Последователен интерфейс	RS232_INT
D	Тракт на ЗУТМД	FDISK_INT (C800:0000)
E	Тракт на ЗУГМД	DISK_INT (F000:EF57)
F	Паралелен интерфейс	PRN_INT
Входни точки на БСВИ		
10	Тракт на видеомонитора	VIDEO_IO (F000:F065)
11	Конфигурация на системата	EQUIPMENT (F000:F84D)
12	Обем на оперативната памет	MEMORY_SIZE (F000:F841)
13	Дисков тракт (ЗУГМД или ЗУТМД)	DISK_IO (F000:EC59) (C800:0057)
14	Последователни интерфейси	RS232_IO (F000:E739)
15	Касетофон или нови функции на ПК	зависи от версията
16	Клавиатурен тракт	KEYBOARD_IO (F000:E82E)
17	Тракт на печатащото устройство	PRINTER_IO (F000:EF02)
18	Вграден интерпретатор на БЕЙСИК	(F600:0000)
19	Начално зареждане	BOOT_STRAP (F000:E6F2)
1A	Календарен часовник	TIME_OF_DAY (F000:FE6E)
Прекъсвания, обслужвани от потребителя		
1B	Прекъсване (Break) от клавиатурата	DUMMY_RETURN (F000:FF53)
1C	Интервал на календарния часовник	DUMMY_RETURN (F000:FF53)
Параметри на БСВИ		
1D	Таблица на тракта на видеомонитора	VIDEO_PARAMS (F000:F0A4)
1E	Таблица на тракта на дисковите устройства	DISK_BASE (F000:EF67) (C800:008E)
1F	Знаков генератор за графични символи	зависи от версията

Потребителят може да провери конкретното разпределение с помощта на системната програма **DEBUG**, като разгледа таблицата на векторите на прекъсванията.

Зареждането на таблицата на векторите на прекъсванията се извършва от програмата за самотестване. В 16-разредните персонални компютри тази таблица съдържа и адресите на области от паметта, в които се съхраняват параметрите за начално установяване на апаратните модули за управление на основните периферни устройства (видеомонитора и запомнящите устройства с магнитен диск).

Операционната система и някои други системни програми участват в разпределението на векторите на прекъсванията, като за тях са запазени векторите с номера от 20 до FF.

3.3.2. ПРОГРАМА ЗА САМОТЕСТВАНЕ И ПРОГРАМА ЗА НАЧАЛНО ЗАРЕЖДАНЕ

За постигане на висока надеждност на персоналните компютри БСВИ съдържа програма за самотестване. Тя се стартира с включване на захранването и след нейното изпълнение се преминава към програмата за начално зареждане на операционната система. Основните функции на програмата осигуряват: проверка на функционалните модули в системата, установяване схемите на компютъра в начално състояние, зареждане на програмните им регистри с необходимата информация, начално зареждане на паметта с векторите на прекъсванията и др. В зависимост от конфигурацията на системата, която се задава чрез микропревключватели на процесорната платка, програмата за самотестване може да конфигурира апаратните модули по различен начин (табл. 3.33). Например, ако потребителят използва битов телевизор, преди включването на компютъра се задава режим, при който контролерът на видеомонитора изобразява по 40 символа на ред, докато при използване на специален монитор се преминава в режим на изобразяване по 80 символа на ред.

При проверка на отделните функционални модули на персоналния компютър е възможно откриването на неизправност в тях. Програмата за самотестване съобщава това на потребителя чрез подходящ текст на екрана или чрез звуков сигнал.

При грешка може да се продължи проверката на компютъра чрез диагностични програми, които се зареждат от дискета. Ако грешката не нарушава работоспособността на компютъра, по желание на потребителя тя може да се пренебрегне и да се продължи работата.

При включването на нов функционален модул е възможно добавяне на неговата програма за самотестване, която се записва в съответната област на паметта тип **PROM**. Основната програма за самотестване винаги проверява за наличието на допълнителни програми за самотестване. Тя ги открива чрез последователна проверка за наличието на кода **AA55** в началото на блокове с размер от по 2 Кбайта в областта от паметта с адреси от **C0000** до **EFFFF**. Откриването на този код е указание, че там се намира допълнителна програма за самотестване, на която се предава управлението в съответния момент. Следователно при запис на допълнителната програма трябва да се спазва правилото за въвеждане на кода **AA55** в началото на всеки блок с размер 2 Кбайта. Функциите

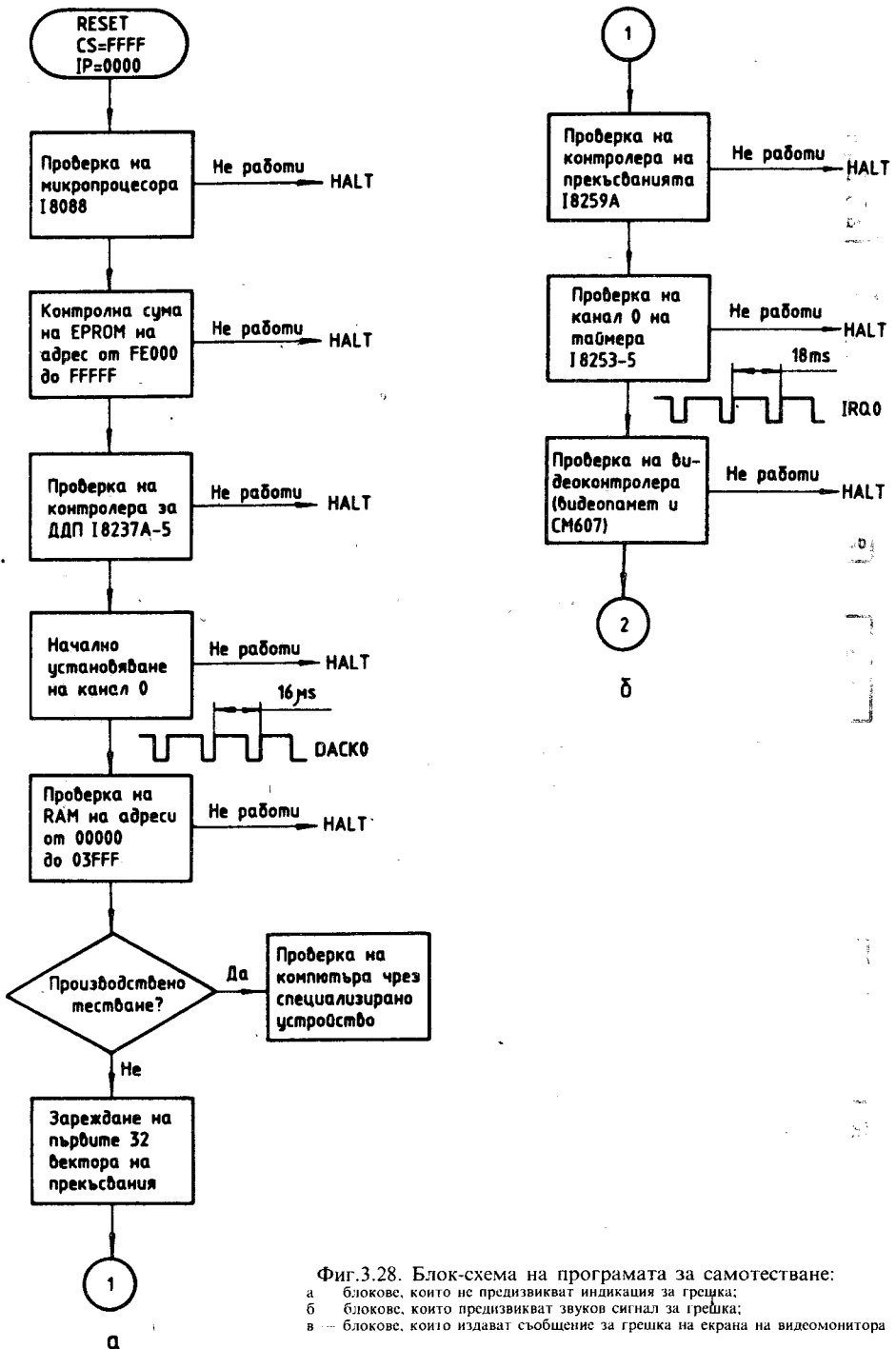
**Функции на микропревк.пчовчателите за определяне
на системната конфигурация**

Микропревк.пчовч. тел. номер	Функция	Значение	
1	Наличие на ЗУ за начално зареждане на операционната система	0	конфигурацията вк.пчовча ЗУ за начално зареждане на операционната система (стандартно значение)
		1	конфигурацията не вк.пчовча ЗУ за начално зареждане
2	Наличие на аритметичен копроцесор 8087	0	аритметичният копроцесор е монтиран и активиран
		1	аритметичният копроцесор не е монтиран или е неактивен (стандартно значение)
3 и 4	Не се използват (определят размера на памята в по-стари модели)	00	задължително състояние на микропревк.пчовчателите
5 и 6	Режим на работа на тракта на видеомонитора	00	конфигурацията вк.пчовча монохроматичен монитор или повече от един монитор
		01	конфигурацията вк.пчовча графичен видеомонитор в режим на изображение 40×25
		10	конфигурацията вк.пчовча графичен видеомонитор в режим на изображение 80×25 (стандартно значение)
		11	към персоналния компютър не е включен видеомонитор
7 и 8	Брой на ЗУГМД в конфигурацията	00	конфигурацията вк.пчовча 4 ЗУГМД (не се използва)
		10	конфигурацията вк.пчовча 3 ЗУГМД (не се използва)
		01	конфигурацията вк.пчовча 2 ЗУГМД (стандартно значение)
		11	конфигурацията вк.пчовча 1 ЗУГМД

Възтази програма осигуряват самотестването на новия функционален модул, зареждането на векторите на прекъсванията, които го обслужват, установяването на регистрите му в начално състояние и връщането на управлението към основната програма за самотестване.

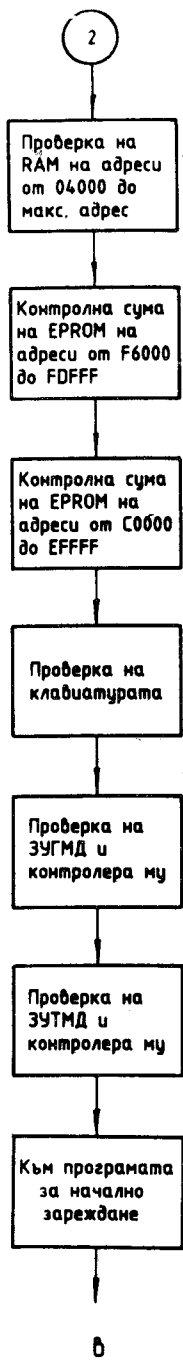
Опростена блок-схема на програмата за самотестване е показана на фиг.3.28. Всеки потребител може самостоятелно да добие представа за съдържанието на тази програма, като използва системната програма DEBUG.

При включване на захранването на компютъра се изпълнява начално установяване – RESET на микропроцесора. В резултат на това в регистрите CS:IP се зарежда адресът FFFF:0000 и се стартира изпълнението на записания там обектен код. В персоналните компютри това е инструкция за безусловен преход към адрес, който може да се различава в зависимост от варианта на БСВИ. От този адрес започва програмата за самотестване.



Фиг.3.28. Блок-схема на програмата за самотестване:

- а — блокове, които не предизвикват индикация за грешка;
- б — блокове, които предизвикват звуков сигнал за грешка;
- в — блокове, които издават съобщение за грешка на екрана на видеомонитора



Първият блок на програмата проверява работоспособността на микропроцесора чрез последователен запис и четене на подходящо подобрани данни в регистрите на микропроцесора. Прави се проверка на флаговия регистър и на правилното изпълнение на инструкциите за условен преход. При откриване на неизправност микропроцесорът се установява в състояние HALT.

В следващия блок се изчислява *контролната сума* на съдържанието на EPROM от адрес FE000 до FFFFF. При получаване на контролна сума, различна от 00, микропроцесорът се установява в състояние HALT. Тази проверка гарантира правилната работа на БСВИ.

Блокът за начална проверка и установяване параметрите на канал 0 на контролера за ДДП 8237А-5 включва:

— проверка за правилната работа на канал 1 на таймера 8253-5 (при откриване на неизправност микропроцесорът се установява в състояние HALT);

— проверка на работоспособността на контролера за ДДП 8237А-5 чрез запис и четене на подходящо подобрани данни във вътрешните регистри на контролера;

— установяване на параметрите и стартиране на канал 0 на контролера за ДДП 8237А-5 (при неуспешно завършване на горните операции микропроцесорът се установява в състояние HALT).

На практика след успешното изпълнение на тази част от програмата за самотестване се стартира една основна функция на централния процесор — *опресняване на паметта* по канал 0 на контролера за ДДП 8237А-5. На линията DACK0 се появява последователност от отрицателни тактови импулси с период около 16 μ s и ширина на импулса — около 1 μ s.

Следващият блок проверява работоспособността на оперативната памет от адрес 00000 до 03FFF с обем 16 Кбайта, разположена в началото на адресното пространство на персоналния компютър. Проверката се извършва по *метода на бягащата нула*. За целта последователно се записват на всички адреси от блока подходящо подобрани данни, след което се извършва четене и проверка на коректността на данните. Най-напред се прави запис на FF, след това — на AA, 55 и 00. Накрая се симулира грешка по нечетност, като предварително се забранява възприемането на прекъсването по линията NMI на микропроцесора. При неуспешно завършване на теста микропроцесорът се установява в състояние HALT.

В началото на този блок се проверява флагът,

указващ дали тестът се изпълнява след начално включване на захранването или след повторно стартиране на компютъра. В последния случай се прескачат всички блокове от програмата за самотестване, проверяващи работоспособността на оперативната памет.

Следващият блок проверява дали към куплунга за клавиатурата е включено специализирано устройство за производствено тестване. За целта се подава последователност за начално установяване на клавиатурата. Ако се получи отговор АА, програмата разбира, че към компютъра е включена клавиатура. При отговор 65 следва, че към този куплунг е присъединено специализирано устройство за производствено тестване. В някои версии на БСВИ получаването на други отговори е указание за неизправност на клавиатурата.

След това се зареждат началните стойности на първите 32 вектора на прекъсвания в горната част на таблицата на векторите (табл. 3.32). Проверява се работоспособността на контролера за приоритетни прекъсвания 8259А чрез запис и четене на подходящо подобрени данни в управляващите му регистри. Последователно се проверява състоянието на линиите за заявка за прекъсване, които в този момент трябва да бъдат неактивни. При неуспешно завършване на теста се издава един продължителен и един кратък звуков сигнал и микропроцесорът се установява в състояние HALT.

Следващият блок проверява работоспособността на канал 0 на таймера 8253-5. При правилна работа канал 0 се програмира като генератор на поредица от импулси с период около 18,2 ms, които осигуряват чрез прекъсване номер 8 обновяването на системния *календарен часовник*. При откриване на неизправност се издава един продължителен и един кратък звуков сигнал и микропроцесорът се установява в състояние HALT.

Блокът за проверка на работоспособността на видеоконтролера изпълнява няколко основни функции:

— проверка на функционирането на видеопаметта по метода на бягащата нула;

— проверка на работоспособността на интегралната схема СМ607 и на общите регистри на видеоконтролера (при откриване на неизправност във видеопаметта или във видеоконтролера се издава един продължителен и два кратки звукови сигнала, като в зависимост от характера на неизправността самотестването може да продължи по-нататък);

— зареждане на началните параметри на интегралната схема СМ607 в зависимост от състоянието на микропревключвателите за определяне на системната конфигурация.

От този момент нататък на екрана на видеомонитора могат да се изведат съобщения за хода на програмата за самотестване и за възникнали неизправности. Систематизирана информация за съобщенията за грешки, изобразявани на екрана по време на самотестването, е дадена в приложение 5.

Следващият блок проверява работоспособността на оперативната памет от адрес 04000 до максималния адрес в зависимост от конкретната конфигурация на персоналният компютър. Проверката се осъществява по метода на бягащата нула на блокове от по 16 Кбайта. По време на теста на екрана на видеомонитора се появява индикация за текущия проверяван блок като число, указващо обема на успешно проверената памет.

При откриване на неизправност на екрана на видеомонитора се появява съобщение за грешка или проверката се прекратява.

След изпълнението на тази тестова програма се фиксира обемът на оперативната памет в конфигурацията на персоналния компютър. Той е равен на стойността, получена до възникването на първата грешка. Така например, ако последният успешно проверен блок е бил с адреси от 6C000 до 6FFFF и е регистрирана неизправност при проверката на блока с адреси от 70000 до 73FFF, обемът на оперативната памет на компютъра не бъде определен като 448 Кбайта. При това грешка възниква както при неизправност в интегралните схеми на паметта, така и при липсата им - когато не е инсталирана поредната банка памет.

Програмата за проверка на контролната сума на съдържанието на EPROM прочита записа и изчислява контролната му сума за адреси от F6000 до FDFFF в блокове от по 8 Кбайта. При получаване на контролна сума, различна от 00, се издава съобщение на екрана на видеомонитора. След това се проверява за наличие на други паметни от тип EPROM, съдържащи допълнителни модули от БСВИ в адресното пространство от C0000 до EFFFF. За целта през 2 Кбайта се четат по два байта. Ако съдържанието е AA55, това е признак за наличието на допълнителни EPROM. В този случай се изчислява контролната сума на открития EPROM (в рамките на блок от 2 Кбайта) и при несъвпадение се издава съобщение за грешка.

Следващата програма проверява работоспособността на включената към персоналния компютър клавиатура. Към нея се подава последователност за начално установяване и клавиатурата трябва да отговори с кода AA. Ако се получи друг код, на екрана на видеомонитора се издава съобщение за грешка, съдържащо получения код.

Последният блок от програмата за самотестване проверява работоспособността на трактовете на запомнящите устройства с гъвкав магнитен диск (ЗУГМД) и на запомнящите устройства с твърд магнитен диск (ЗУТМД). Проверката на ЗУГМД се извършва, когато те са обявени в конфигурацията на системата чрез микропревключватели (табл. 3.33). ЗУТМД се проверяват, когато са монтирани в персоналния компютър.

Изпълнява се тест на регистрите на специализираните интегрални схеми, външните регистри и логиката за управление на контролера на запомнящото устройство (ЗУ). При отсъствие на грешка се зареждат началните параметри и се установява режимът на работа на контролера със ЗУ. Проверява се връзката със запомнящото устройство: състояние на интерфейсия кабел, коректност на интерфейсните сигнали и функционална годност на ЗУ. При възникване на грешка във всеки от посочените случаи, на екрана на видеомонитора се издава съобщение за грешка.

След правилното преминаване на самотестването компютърът е готов за работа и неговото управление се предава на програмата за начално зареждане (ПНЗ). Това става чрез програмно прекъсване.

Програмата за начално зареждане е предназначена за определена конфигурация на персоналния компютър. Всяко включване на нов функционален модул предполага промяна на ПНЗ.

След активиране програмата се опитва да прочете от ЗУГМД А първия сектор на нулевата му пътека и да запише съдържанието му в паметта на компютъра. При успешно завършване на операцията управлението се предава на прочетената програма, която има за задача да зареди опе-

рационната система от дискетата. Когато в ЗУГМД не е поставена дискета или възникне апаратна грешка на системата контролер — ЗУГМД, при опит за начално зареждане се получава неуспешна дискова операция и автоматично се стартира интерпретаторът на БЕЙСИК, който се съхранява в постоянната памет на персоналния компютър.

Когато записът на първия сектор от нулевата пътека е повреден, на екрана на видеомонитора се появява съобщение за грешка:

Disk boot failure

За продължаване на работата е необходимо да се смени дискетата с операционната система и да се рестартира компютърът.

При опит за начално зареждане от дискета, която не е системна, на екрана на видеомонитора се появява съобщението:

Non-System disk or disk error

Replace and strike any key when ready

За продължаване на работата е необходимо да се смени дискетата с друга, която има операционна система и с натискането на произволен клавиш да се направи опит за ново зареждане на системата.

При наличие на контролер и ЗУГМД след неуспешно четене от дискета се прави опит за зареждане от ЗУГМД С. В случай на нова неуспешна операция автоматично се стартира интерпретаторът на вградения БЕЙСИК.

3.3.3. ВХОДНО-ИЗХОДНИ ДРАЙВЕРНИ ПРОГРАМИ

Драйверните програми служат за обслужване на външните устройства на персоналния компютър. Тези програми работят директно с контролерите и позволяват на програмиста да не се интересува от физическата организация на външното устройство, а единствено от командите, които реализира обслужващата го драйверна програма (драйвер). В приложение 6 е дадена по-подробна информация за функциите на регистрите, които се използват за предаване на входна и изходна информация към драйверите (команди, данни, състояние на периферното устройство и др.)

Принципите на организация на програмното осигуряване на 16-редните персонални компютри позволяват на потребителя относително леко да модифицира начина на функционирането му. На практика програмистът може да подмени всеки първоначално зареден драйвер (независимо от разположението му в постоянната или оперативната памет) с нов, който отговаря на нови системни изисквания. За целта е необходимо новият драйвер да бъде зареден в оперативната памет на компютъра и да се подмени адресът, записан в съответното поле на таблицата на векторите на прекъсванията с новия адрес на драйвера. Потребителят може да се информира за състоянието на активираните прекъсвания и за адресите на обслужващите ги програми с помощта на системната програма DEBUG.

Някои системни функции се обслужват от програми с организация, подобна на драйверите, които не са свързани с управлението на периферни устройства. Такива са функциите за обслужване на системния часовник, определянето на конфигурацията на компютъра, определяне на личната му памет и др.

Драйвер на ЗУГМД

Тази програма е предназначена да управлява до 4 ЗУГМД при едностранен или двустранен запис с двойна плътност на записваната информация. Информацията се разполага върху 40 пътечки с по 8 или 9 сектора и 512 байта информация във всеки от тях. Драйверът може да обслужва ЗУГМД от най-различен тип, като параметрите на използваните устройства се намират в *параметричната таблица* (входна точка на БСВИ). Тя съдържа времето за позициониране между две пътечки, времето за изчакване преди спиране на мотора и след неговото пускане, броят на секторите на пътечка, броят на байтовете в сектор, информацията за запълване при форматиране и др.

Основните операции, поддържани от драйвера на ЗУГМД, са:

- установяване на ЗУГМД в начално състояние;
- проверка на текущото състояние на ЗУГМД;
- четене на сектори;
- запис на сектори;
- проверка на сектори;
- форматиране на пътечка.

При операцията проверка на сектори за разлика от четенето не се прехвърля информация в паметта, а само се проверява правилността на записа върху дискетата (чрез контролната сума в края на всеки сектор).

Драйвер на видеомонитора

Тази програма е предназначена да управлява монохроматичен или цветен видеомонитор от текстов или графичен тип. Освен видеомонитор драйверът може да управлява и светлинен молив.

И тук както при дискетния драйвер, основните параметри са записани в параметрична таблица (входна точка на БСВИ). Това дава възможност за работа с различни честоти на редовата и кадровата развивка, различен формат на изобразявания екран и др.

Основните операции, поддържани от този драйвер, са:

- установяване на схемите на видеоконтролера в начално състояние и смяна на текстовите и графични режими;
- определяне вида на курсора и управление на движението му по екрана;
- четене и запис на текстова информация;
- проверка на състоянието и преместване на информацията по екрана.

Драйверът на видеомонитора задава основните атрибути, присъщи на всеки символ: цвят (ако мониторът е графичен), инверсно представяне, мигане, подчертаване (само за монохроматичен текстов монитор). В графичен режим за всяка точка се определя цветът (тъмен, светъл или градиран по яркост за монохроматичен монитор), като наборът от всички цветове образува *палитра*.

В текстов режим драйверът управлява разполагането на информацията в рамките на екрана, поддържа функциите: вмъкване на символ, изтриване на символ, вмъкване на ред, придвижване на редовете отдолу и нагоре (scroll) и др.

Клавиатурен драйвер

Програмата за обслужване на клавиатурата използва програмни таблици на символите, което позволява чрез промяна на съдържанието им да се установяват различни подреждания на клавиатурата. Самата клавиатура генерира само кода при натискане или отпускане на клавиш. Този код, наречен *позиционен*, се интерпретира от драйвера, който изработва съответстващия му програмен код (използва се разширена ASCII таблица с 256 символа). Това позволява чрез клавиатурния драйвер да се формират върху една и съща клавиатура две подреждания (например латиница и кирилица), като превключването им става с един клавиш. Символите върху клавиатурата се подреждат по съответния стандарт, като се възприемат всички големи и малки букви. Клавиатурният драйвер буферира до 20 символа. В клавиатурата също има буфер, който позволява да се възприемат до 16 едновременно натиснати клавиша, без да се нарушава нейната работа.

Наред с клавишите за смяна на регистъра (Shift и Lock) има още два клавиша с подобно действие (Ctrl и Alt). В комбинация с кой да е от останалите клавиши те формират специални *управляващи кодове*. В този случай драйверът предава към горните нива на програмното осигуряване т. нар. *разширени кодове*. Това увеличава възможностите на клавиатурата.

Характерна особеност на клавиатурния драйвер е поддържането на таблица на клавиши със задържане. Това позволява еднократното натискане на определен управляващ клавиш да осигурява изпълнението на присъщата му функция до повторното му натискане.

Този драйвер поддържа следните основни операции:

- проверка за въведен символ от клавиатурата;
- четене на поредния символ;
- проверка на състоянието на клавишите за смяна на регистъра (Shift, Lock, Ctrl, Alt и др.);
- изработване на разширени кодове.

Драйвер на печатащото устройство

Чрез този драйвер се управляват до три печатащи устройства, свързани чрез паралелен интерфейс. Специална таблица съхранява характерни параметри за печатащите устройства: време на изчакване, адрес на контролера, който ги обслужва, и др.

Основните операции, изпълнявани от драйвера, са:

- установяване схемите на контролера на печатащото устройство в начално състояние;
- проверка на състоянието на печатащото устройство;
- управление на логическия интерфейс към печатащото устройство;
- отпечатване на символ.

Системни драйвери

Тези програми, независимо че не обслужват периферни устройства работят като драйвери и изпълняват следните функции:

- четене на регистрите на календарния часовник;

- установяване на календарния часовник;
- проверка и промяна на конфигурацията на компютъра;
- определяне на оперативната памет на компютъра;
- разпечатване съдържанието на екрана на видеомонитора.

Допълнителни драйвери

Тези програми обслужват периферни устройства, които не са в основната конфигурация на системата. Те се инсталират допълнително и затова драйверите могат да не се намират в базовата система за вход-изход на компютъра. В повечето случаи допълнителните драйвери са записани в EPROM, разположен в контролера на добавяното периферно устройство. Тук се включват и драйверите на устройства, които имат специализирано предназначение или спомагателни функции в конфигурацията на компютъра.

Някои от характерните допълнителни драйвери са:

— драйвер за последователна асинхронна комуникация (интерфейс RS232C);

— драйвер за запомнящо устройство с твърд магнитен диск;

— драйвер за запомнящо устройство с магнитна лента и др.

Всяка от тези програми изпълнява операции, свързани с четене или запис на информация, установяване на режима, управление на логическия интерфейс и проверка на състоянието на периферното устройство.

Драйверът на ЗУТМД управлява до 2 дискови устройства, като в таблица се съхраняват техните основни параметри: брой на повърхностите за запис и четене, брой на цилиндрите, време за позициониране между два цилиндъра, брой на секторите на една пътечка, брой на байтовете в сектора и др. Всяко от запомнящите устройства може да е с капацитет 10, 20 или повече Мбайта. Като допълнителна информация в таблицата се задават стойности на времето за избор, време за готовност на дисковото устройство, данни за форматиране и др.

Драйверът на ЗУТМД използва същото програмно прекъсване, както и дискетният драйвер. Той определя дали заявката за обслужване е предназначена за ЗУТМД. В противен случай управлението се предава на драйвера на ЗУГМД.

3.4. ВЪНШНИ ЗАПОМНЯЩИ УСТРОЙСТВА С ДИСКОВ НОСИТЕЛ

Външните запомнящи устройства с дисков магнитен носител се разделят на две основни групи според вида на магнитния диск — носител на информацията. Първата група обхваща запомнящите устройства с гъвкав магнитен диск — ЗУГМД, които са получили гражданско название флопидискови устройства. Основната им особеност е използването на сменяем, гъвкав магнитен диск (дискета) като носител на информацията. Това определя техните експлоатационни, технически и икономически характеристики.

Втората група обхваща запомнящите устройства с твърд магнитен диск — ЗУТМД, които най-често се наричат уинчестър дискове. Те се характеризират с използването на твърд, несменяем магнитен диск като

носител на информацията и с техниката на позициониране чрез фабрично записана сервоповърхност.

Доскоро водеща тенденция в разработването на персоналните компютри беше задължителното използване на ЗУГМД като основно външно запомнящо устройство и включването на ЗУТМД като допълнителна възможност. Тази тенденция се определяше от относително по-ниската цена на ЗУГМД, гъвкавостта при работа със сменяем информационен носител, простотата на експлоатация и поддържане на устройствата и не на последно място — от психологическата предразположеност за използване на дискетата като индивидуален, личен програмен и информационен носител.

Развитието на технологията и техниката в последните години значително снижи цената на ЗУТМД. Освен това е налице рязко нарастване на обема на информацията, която може да се съхрани върху ЗУТМД. Тези причини обуславят въвеждането на ЗУТМД с относително голям капацитет като основно външно запомнящо устройство в съвременните фамилии персонални компютри. ЗУГМД все повече се използва като спомагателно устройство — за архивиране, пренасяне, първоначално зареждане, обмен и проверка на неголеми масиви от информация, програми и данни.

3.4.1. ЗАПОМНЯЩИ УСТРОЙСТВА С ГЪВКАВ МАГНИТЕН ДИСК

В персоналните компютри най-широко разпространение са намерили ЗУГМД с диаметър на носителя 133 mm (5,25 инча) и 89 mm (3,5 инча). В съответствие с диаметъра на дискетата се различават значително и габаритните размери на самите ЗУГМД. Така например устройствата със 133-милиметрови дискети обикновено имат дължина 203 mm, ширина 146 mm и височина 82 mm. В съвременните фамилии персонални компютри се вграждат преди всичко т.нар. *устройства с 1/2 височина*. Те имат два пъти по-малка височина от нормалните, като запазват всички останали характеристики на ЗУГМД. Основното им предимство е възможността да бъдат монтирани в персоналния компютър по две такива ЗУГМД на мястото на едно с нормална височина.

При работа с дискови носители за съхранение на информацията се използват едната или двете кръгови повърхности на диска. Според броя на информационните повърхности магнитните дискове могат да бъдат едностранни или двустранни, а запомнящите устройства — съответно с една или с две магнитни глави за запис и четене. В персоналните компютри се използват предимно двустранни дискети. Възможността за съхраняване на информацията върху едната или върху двете повърхности е характеристика на дискетата, която се гарантира от завода производител.

Информацията се записва върху отделни пътечки. Те се номерират от 0 до N в посока от най-външната към вътрешните. Числото N задава максималния брой на пътечките върху една дискета. Номерацията на пътечките е еднаква и за двете повърхности на двустранните дискети, а самите повърхности се означават като нулева и първа в съответствие с номерата на двете глави за запис-четене на ЗУГМД.

Характерна особеност на ЗУГМД е **плътността на записа** върху дискетата. Определят се напречна и линейна плътност на записа. *Напречна-*

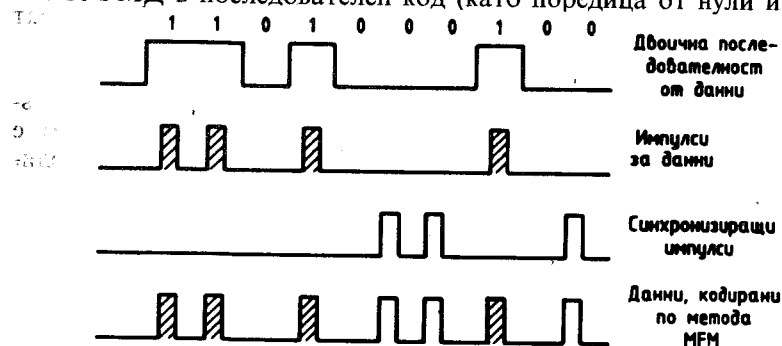
та плътност се измерва с броя пътечки на единица дължина по радиуса на дискетата. *Линейната плътност* се отчита с броя битове информация на единица дължина по окръжността на дадена пътечка. Плътността на записа се определя предимно от качествата на магнитното покритие на диска и от параметрите на главата за запис-четене.

Напречната плътност на записа при 133-милиметровите ЗУГМД най-често е 1,89 или 3,78 пътечки/mm.л. Устройствата с по-ниската напречна плътност разполагат информацията на една дискова повърхност върху 40 пътечки. При устройствата с по-висока напречна плътност се работи с 80 или повече пътечки. Сега в персоналните компютри масово се използват ЗУГМД с 40 пътечки на всяка повърхност.

Линейната плътност на записа при 133-милиметровите ЗУГМД е 218 бита/mm. Съществуват и устройства със значително по-голяма линейна плътност на записа — т.нар. висока плътност. При същия метод на кодиране в тях се постига неколкостратно увеличение на обема на информацията, записана върху дискетата.

Броят на пътечките върху дискетата се определя в зависимост от параметъра напречна плътност на записа. Някои ЗУГМД могат да работят с различен брой пътечки. Напречната плътност на записа се задава чрез микропревключватели или мостчета. Те трябва да бъдат инсталирани преди монтиране на ЗУГМД в компютъра. Най-често в персоналните компютри се използват ЗУГМД с 40 пътечки. Последните версии на дисковата операционна система (например PC DOS версия 3.30) допускат работа и със ЗУГМД с 80 пътечки.

Друга характерна особеност на дисковите запомнящи устройства е **методът на запис** на информацията върху носителя. Този метод определя плътността на разполагане на данните върху магнитния диск и поради това оказва съществено влияние на обема на съхраняваната информация. Освен това методът на запис е свързан и с достоверността на съхраняваните данни, със скоростта на обмен между контролера и запомнящото устройство, със сложността на контролера и т.н. В 16-разредните персонални компютри се използва метод на запис върху носителя с *модифицирана честотна модулация* (modified frequency modulation — MFM). На фиг.3.29 е показан пример за MFM-кодиране на една поредица данни. В контролера данните се обработват в двоичен вид и се предават към ЗУГМД в последователен код (като поредица от нули и единици).



Фиг. 3.29. Времедиаграма на кодовите последователности при кодиране по метода MFM

За да се намали броят на синхроимпулсите, за синхронизация при метода MFM се използват самите импулси за данни. Включването на допълнителни синхроимпулси се налага само в случаите на няколко последователни нули, когато импулсите за данни отсъстват. Общото правило за кодиране по метода MFM е следното:

а) предаване на импулс за данни за всяка единица от двоичната поредица;

б) предаване на синхроимпулс за всяка втора и за всяка следваща нула от група последователни нули в двоичната поредица.

Получената като резултат поредица обединява импулсите за данни и синхроимпулсите.

Според вида на дискетата и на самото ЗУГМД обемът на съхраняваната информация е различен. В табл. 3.34 са представени основните параметри на произвежданите у нас 133-милиметрови дискети за персонални компютри.

Таблица 3.34

Дискети, произвеждани в НРБ

Именовање на дискетата	Описание за параметрите на записа	Брой на работните повърхности	Напечена плътност на записа (пътечки/mm)	Капацитет на дискетата (Кбайта)
EC 5288*	SS/SD (1S/1D)	1	единична 1,89	180
EC 5289	SS/DD (1S/2D)	1	двойна 1,89	180
EC 5287	DS/DD (2S/2D)	2	двойна 1,89	360
EC 5259**	DS/QD (2S/4D)	2	четворна 3,78	720

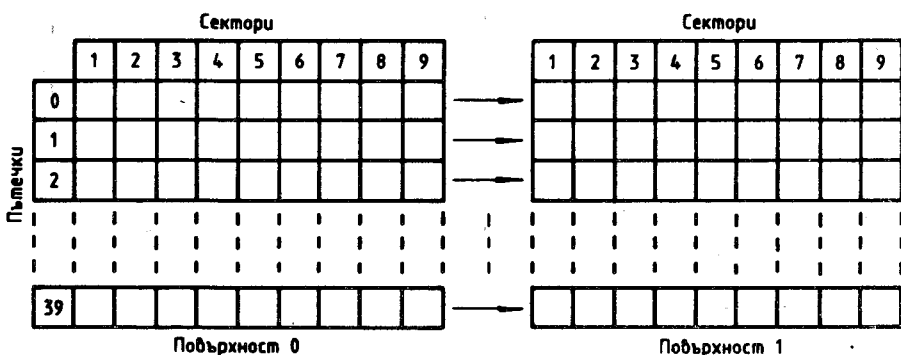
Забелжки: * Не се препоръчва използването им в 16-разредните персонални компютри.
 ** Пълният капацитет на записа може да се постигне с помощта на PC DOS версия 3.30 на ЗУГМД с 80 пътечки (препоръчват се за персонални компютри, съвместими с IBM AT).

За описание на начина на разполагане на информацията върху дискетата се въвежда понятието **формат**. По определение това е набор от правила за разполагане на информацията — служебна и данни на потребителя върху носителя. Форматът определя мястото и размера на записите за служебна информация, необходими за номериране на отделните области, за разграничаването им една от друга, за контрол на информацията и др. Съществен параметър на формата е размерът на областта за данни на потребителя. В ЗУГМД, както и във всички видове дискови запомнящи устройства се прилагат стандартни формати на информацията.

Информацията върху дискетата се разделя на порции, наречени *сектори*. Секторът е част от пътечката с определен размер. Всяка пътечка се състои от фиксиран брой сектори. Размерът на сектора е онова най-малко количество данни, което може да бъде прочетено или записано върху дискетата с една входно-изходна операция. С размера на сектора (в брой байтове или думи) са свързани устройството и действието на самото ЗУГМД и на неговия контролер.

В персоналните компютри се използва различна организация на данните, записани върху ЗУГМД. Различните варианти се отличават както по броя на секторите върху една пътечка (най-често 8, 9 или 16), така и по обема на данните в един сектор (обикновено 128, 256 или 512 байта).

Секторите се номерират последователно от 1 до М, като се започва от първия сектор след началото на пътечката. Числото М задава максималния брой на секторите на една пътечка. Схемата на фиг. 3. 30 илюстрира реда на записване на секторите при организация с 40 пътечки на



Фиг.3.30. Номериране на физическите сектори върху двустранна дискета

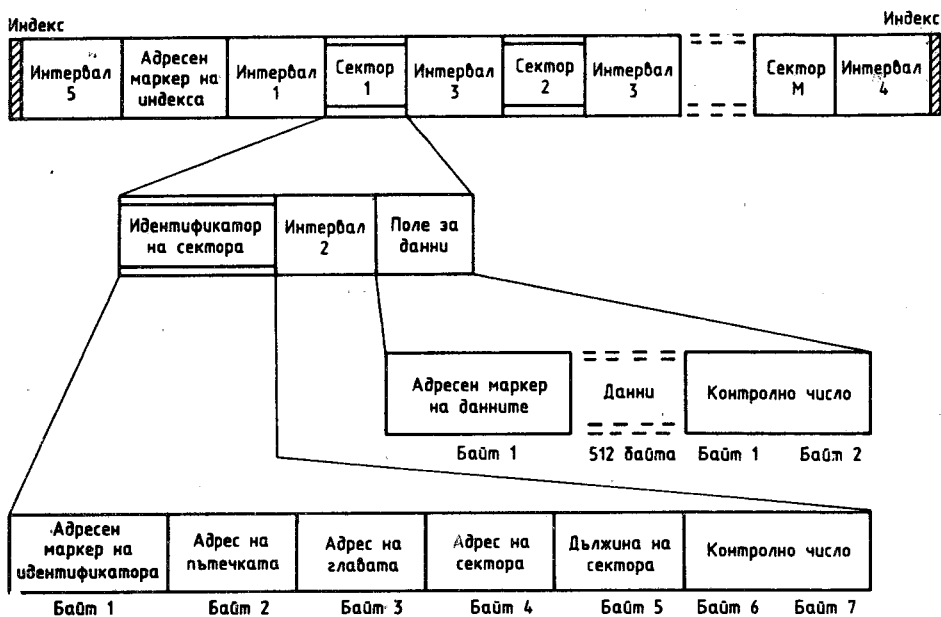
дискетата и 9 сектора на пътечка. Начален сектор за цялата дискета е този с номер 1 от пътечката с номер 0. Броят на секторите на една пътечка е фиксиран и се определя от операционната система на компютъра. Секторите се отделят един от друг чрез интервали, в които не се записва никаква информация. Разполагането на секторите се определя от сигнала ИНДЕКС, указващ физическото начало на пътечката.

Форматът на записа на данните в ЗУГМД за персоналните компютри ЕС1831, ЕС1832, Правец-16 (ЕС1839), Правец-16Н и други е показан на фиг.3.31. Като служебна информация се записват пет типа интервали, адресният маркер на индекса и полетата за идентификатори на секторите, а като потребителска — полетата за данни в рамките на всеки сектор.

Интервалите са с различен размер и служат за синхронизиране на схемите на контролера на ЗУГМД. Най-голям е интервал 5, който следва индекса (физическото начало на дискетата), а най-малък е интервал 2, който разделя полето на идентификатора на сектора от полето за данни. В някои случаи интервал 4 може да превиши по размер интервал 5, тъй като той се получава като резултативна величина в зависимост от запълването на пътечката с информация. *Адресният маркер на индекса* представлява логическото начало на пътечката.

Полето на **идентификатора на сектора** се създава по време на форматиране на дискетата и може да съдържа различна информация в зависимост от версията на операционната система. Във всички случаи той съдържа адреса на пътечката, адреса (номера) на главата и адреса на сектора. Дължината на сектора е характерен параметър за персоналния компютър и неговата операционна система. За осигуряване достоверността на записа в полето на идентификатора на сектора в края на полето се записва 1 байт за *контролното число* (код за цикличен контрол) на идентификатора.

Полето за данни съдържа потребителската информация, която се записва при работата със ЗУГМД. Размерът на полето данни трябва да съответства на числото, записано в идентификатора. За контролното



Фиг.3.31. Формат на записа върху дискетата на 16-разреден персонален компютър

число на данните са предвидени 2 байта, с помощта на които могат да се откриват единични грешки (в 1 бит от записа) и грешки в поредици до 5 бита.

Програмистът може да записва или да чете от дискетата само информацията, разположена в полето за данни. Служебната информация е недостатъчна за програмните средства на компютъра. Тя намалява незначително полезния обем на дискетата, но осигурява ефективен достъп до всеки отделен сектор и възможности за четене или запис на данните с по-голяма достоверност.

Записът на служебната информация върху дискетата се извършва от системната програма FORMAT. С нейна помощ се създават полетата на интервалите и идентификаторите на секторите в зависимост от логическата организация на дисковото пространство на конкретната версия на операционната система. Така за дисковата операционна система ДОС-16 версия 3.10 размерът на сектора е 512 байта, а на всяка пътека има по 9 сектора.

Системната програма FORMAT създава служебен запис и в полетата за данни, като ги запълва с информация F7. Освен това тя записва в първия сектор на нулевата пътека таблица за разпределение на дисковото пространство (ТРДП)¹, която съдържа информация за логическата организация на дискетата — размери и брой на секторите, едностранна или двустранна, определя дали дискетата е системна или не, име

¹ File allocation table (FAT).

на тома и други данни, необходими на програмата за начално зареждане, както и за файловата система на компютъра.

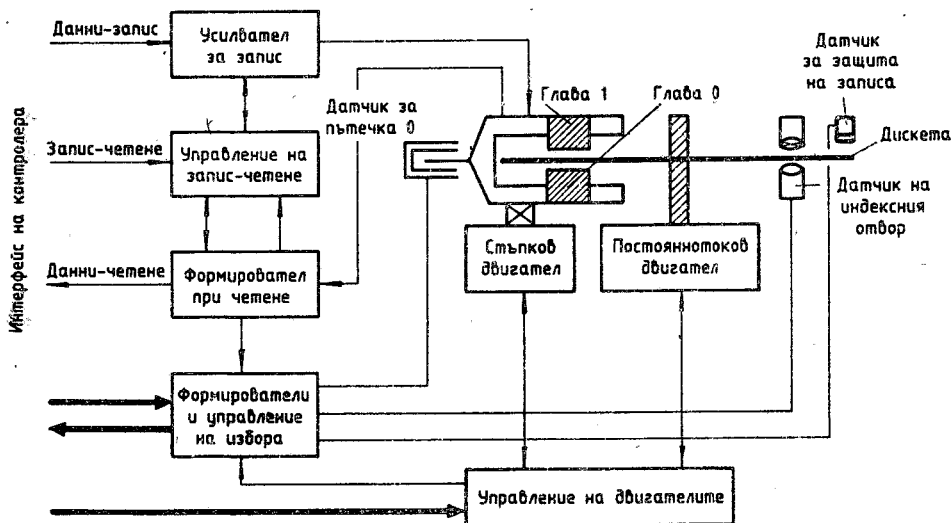
Преди да се започне работа с нова дискета, необходимо е върху нея да се запише служебната информация, която я подготвя като логически носител чрез системната програма FORMAT, т.е. да се *форматира*. Естествено следва, че дискети, създадени (форматирани) с друга операционна система, не могат да бъдат прочетени от потребителя поради несъответствието на логическата им организация (друг формат или параметри на записа).

Опростена функционална схема на ЗУГМД е показана на фиг.3.32. Постояннотоковият двигател задвижва шпиндела, към който е свързан директно или чрез ремъчна предавка. Специална регулираща схема в блока за управление поддържа с точност 1% оборотите на въртене на двигателя. Стъпковият двигател управлява позициониращата система в съответствие с импулсите по интерфейсната линия СТЪПКА. Сигналът за пътечка 0 се изработва от електромеханичен датчик, регистриращ положението на главата. Двигателите на ЗУГМД се захранват с постоянен ток с напрежение +12 V.

Към един контролер в персоналният компютър обикновено се включват няколко ЗУГМД. Поради това в самото ЗУГМД има средства за установяване на неговия номер (1, 2, 3 и т.н., респективно А, В, С и т.н.). Те се изпълняват като микропревключватели или мостчета, които се инсталират при монтиране на устройството в компютъра. Начинът за установяването им се дава при описанието на интерфейса в т.3.4.2.

Датчикът за защита е от механичен или фотоелектронен тип и осигурява формирането на сигнала ЗАБРАНА НА ЗАПИСА по интерфейса към контролера.

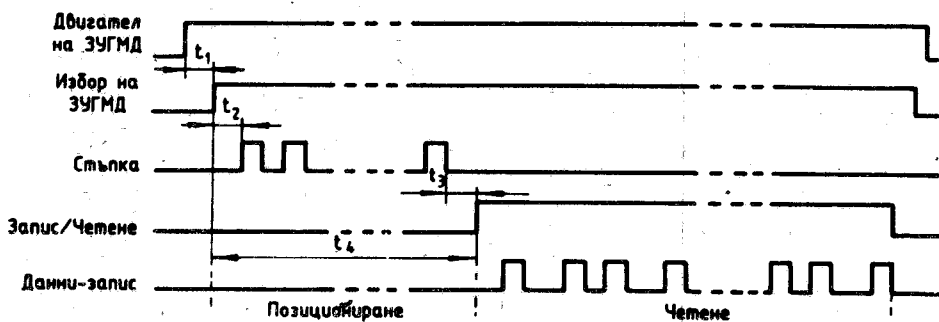
Главите за запис-четене се обединяват в пакет, който се задвижва от стъпковия двигател. Всяка глава включва в себе си отделна четяща,



Фиг.3.32. Обща функционална схема на ЗУГМД

записваща и изтриваща намотка. Главата, която работи с долната повърхност на дискетата, е нулева, а горната глава е първа. В повечето съвременни ЗУГМД са предвидени мерки за защита на записа при аварийно изключване на захранващото напрежение на компютъра. Така се гарантира целостта на записа при аварийни ситуации без специални действия на потребителя.

При изпълнение на операциите със ЗУГМД отначало е необходимо да се включи постояннотоковият двигател и да се избере устройството чрез съответните сигнали по интерфейса. На фиг.3.33 операцията четене е илюстрирана с времедиаграма на интерфейсите сигнали. Операцията се изпълнява в следната последователност:



Фиг.3.33. Примерна времедиаграма на операция четене от ЗУГМД

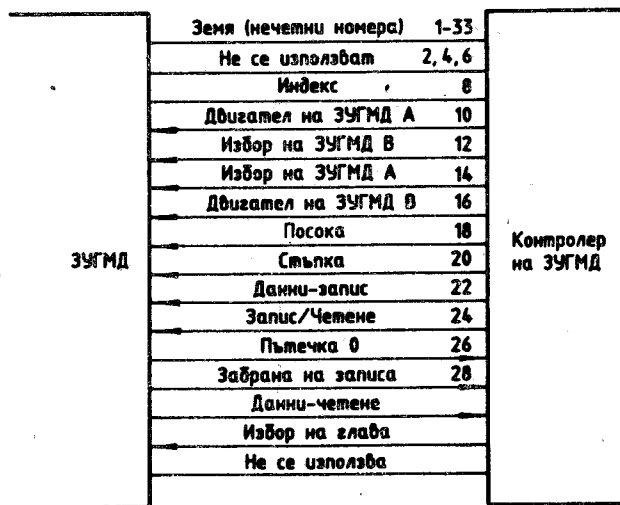
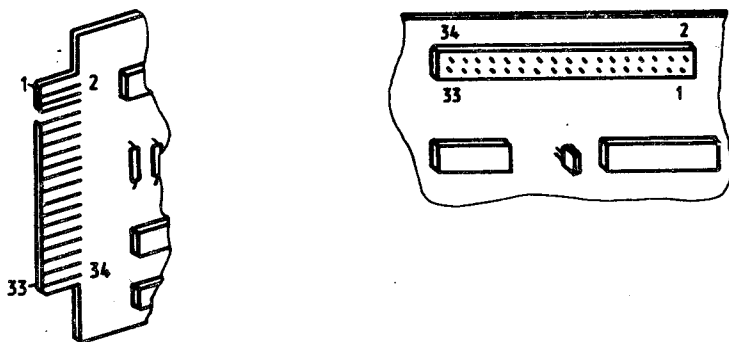
- подава се сигнал за включване на двигателя;
- избира се съответното ЗУГМД;
- позиционира се главата на необходимата пътечка;
- установява се сигнал за четене по линията ЗАПИС/ЧЕТЕНЕ.

Прочетените от диска данни се предават към контролера в интервала t_n до t_k , зададен от сигнала ЧЕТЕНЕ. Времената t_1 , t_2 , t_3 и t_4 имат определени минимални стойности съобразно конкретния тип ЗУГМД. Контролерът трябва да съблюдава тези времеви съотношения между сигналите по интерфейса. Времевите характеристики на ЗУГМД се дават от завода производител и спазването на изискванията към тях гарантира правилно функциониране и максимално бързодействие на устройството. Параметрите на ЗУГМД се записват в таблицата на адрес FEFC7 в БСВИ и са достъпни за програмиста чрез програмното прекъсване с номер 1Е.

3.4.2. ИНТЕРФЕЙС МЕЖДУ КОНТРОЛЕРА И ЗУГМД

Свързването на ЗУГМД към неговия контролер става с интерфейсите линии. Интерфейсите към външните запомнящи устройства са стандартизирани, т.е. предназначението на всяка от линиите е строго определено. Тази унификация позволява включването на произволно ЗУГМД, отговарящо на изискванията на интерфейса, към стандартен контролер. В персоналните компютри се използва модифициран вариант на интерфейса Shugart ST-412 (фиг.3.34), който е стандартизиран в рамките на СИВ в документите: РТМ „ЕС ЭВМ. Персональные профес-

сиональные ЭВМ — Ряд 3. Принципы работы“ и ММ СМ ЭВМ 010—77 „Накопители на гибких магнитных дисках, однодисковые. Интерфейс. Структура и состав. Требования к функциональным характеристикам“. Характерно за този интерфейс е използването на линиите за избор на устройство. Устройството, включено към първия куплунг на интерфейсия кабел, се определя като ЗУГМД В, а устройството, включено към втория куплунг (след завъртането на интерфейските линии за избор) — като ЗУГМД А (фиг.3.35). За да се осигури правилната работа на системата контролер — ЗУГМД, микропревключвателите или мостчетата за избор в ЗУ трябва да бъдат установени като за избор на устройство В (1), което се илюстрира на същата фигура.



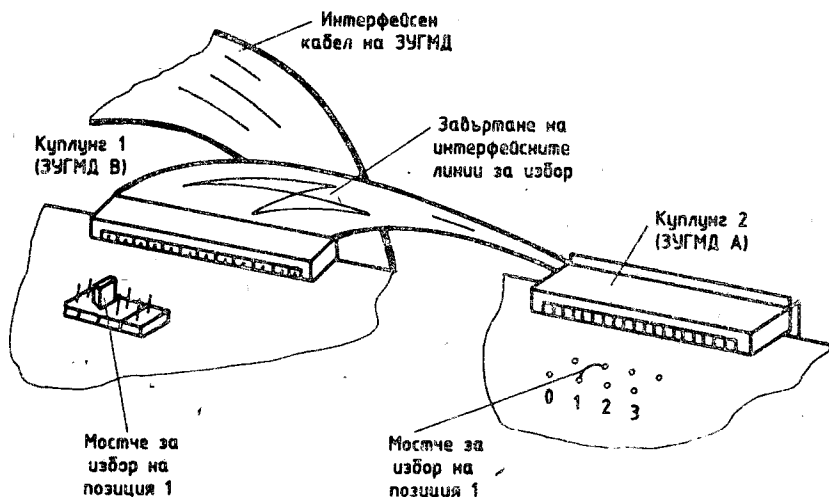
Фиг.3.34. Интерфейс между ЗУГМД и контролера в 16-разреден ПК

Основните режими на работа по интерфейса между ЗУГМД и неговия контролер са: избор на ЗУГМД, позициониране на зададена пътечка, четене на един или няколко сектора, запис на един или няколко сектора. За всеки режим е зададена определена последователност на сигналите, предавани по интерфейса (пример за изпълнение на операция четене е

даден на фиг.3.33). Нарушаването на тази последователност означава неправилно функциониране на ЗУГМД и в такъв случай контролерът предприема мерки за възстановяване на нормалното му действие.

Интерфейсът на ЗУГМД се състои от две групи линии — входни и изходни. Входните линии служат за предаване на данни и управляващи сигнали към ЗУГМД, а изходните — за предаване на данни и информация за състоянието от ЗУГМД към неговия контролер.

Входните линии са предназначени за избиране на дадено ЗУГМД, за стартиране на неговия двигател, за управление на позиционирането и за указване на вида на операцията: четене или запис.



Фиг.3.35. Схема на включване на интерфейсният кабел към ЗУГМД

За избиране на ЗУГМД А или В, с което ще се обменят данни в дадения случай, се използват линиите ИЗБОР НА ЗУГМД А или ИЗБОР НА ЗУГМД В. Контролерът подава сигналите по тези две линии алтернативно, т.е. в даден момент работи с устройство А или с устройство В.

Стартирането на постояннотоковите двигатели, които задвижват дискетата, се извършва чрез подаването на сигнали по линиите ДВИГАТЕЛ НА ЗУГМД А или ДВИГАТЕЛ НА ЗУГМД В. Активното ниво на сигналите осигурява включването на двигателите на устройство А или на устройство В. Сигналят по тези линии се подава около 1 s преди началото на обмена и се прекратява 2 s след завършването му, ако в това време не постъпи нова команда към ЗУГМД. С това се гарантира минимално износване на главата за запис-четене на ЗУГМД и на самия магнитен диск, тъй като той се намира в покой през цялото време, когато няма обмен на данни. Сигналите ДВИГАТЕЛ са валидни само когато е активен съответният сигнал за избор на ЗУГМД.

При изпълнение на операцията четене и запис е необходимо предварително да се позиционира магнитната глава върху зададената пътечка. За управление на позиционирането служат линиите СТЬПКА и НАПРАВЛЕНИЕ. По линията СТЬПКА се предават импулси, всеки от които указва придвижване на главата на следващата пътечка. Придвиж-

ване на повече от една пътечка се извършва чрез поредица от импулсни сигнали по тази линия. За указване на посоката на придвижване на главата служи линията НАПРАВЛЕНИЕ. В случай че по тази линия е подаден активен сигнал, при всеки импулс по линията СТЬПКА главата се премества с една пътечка назад (към периферията на диска). В противен случай при всеки импулс СТЬПКА главата се премества с една пътечка напред (към центъра на диска).

За задаване на вида на основните операции със ЗУГМД служи линията ЗАПИС/ЧЕТЕНЕ. При наличие на сигнал по нея (логическа 1) устройството изпълнява операция запис, при която върху диска се записват данните, постъпващи по линията ДАННИ-ЗАПИС. Отсъствието на сигнал по линията (логическа 0) указва операция четене, при която четящо-записващата система на ЗУГМД прочита определена порция информация от носителя (един или няколко сектора) и я предава към контролера по линията ДАННИ-ЧЕТЕНЕ.

По линията ДАННИ-ЗАПИС в ЗУГМД постъпва информацията за запис върху диска. Данните се предават кодирани по метода MFM във вид на поредица от импулси. Подготовката на данните в такъв последователен код се извършва в контролера на ЗУГМД.

За определяне на една от двете глави за запис-четене на ЗУГМД се използва линията ИЗБОР НА ГЛАВА. При наличие на логическа 1 по нея се избира глава 1 на устройството и операциите четене или запис се изпълняват върху повърхност 1 на диска. В противен случай (логическа 0 по линията) операциите се изпълняват за повърхност 0.

Исходните линии на информационния интерфейс служат за предаване на данни и сигнали за състоянието от ЗУГМД към неговия контролер.

Линията ИНДЕКС е предназначена за указване на началото на пътечката на диска. По тази линия ЗУГМД предава импулс към контролера при всяко преминаване на индексния отвор на диска през фотоелектронния датчик. Така контролерът синхронизира своята работа с момента, в който главата за запис-четене се намира в началото на дадената пътечка. Линията ИНДЕКС се използва и за проверка за наличието на дискета в ЗУГМД. Когато дискетата не е поставена в устройството, периодичните импулси ИНДЕКС не се предават и сигналът по линията е постоянен. За контролера липсата на дискета също означава неизправност в даденото дисково устройство.

За определяне на началното положение на главата се използва линията ПЪТЕЧКА 0. Активирането на сигнала по тази линия показва на контролера, че позициониращата система на ЗУГМД е установила пакет с главите за запис-четене върху пътечката с номер 0, т.е. на най-външната пътечка на дискетата.

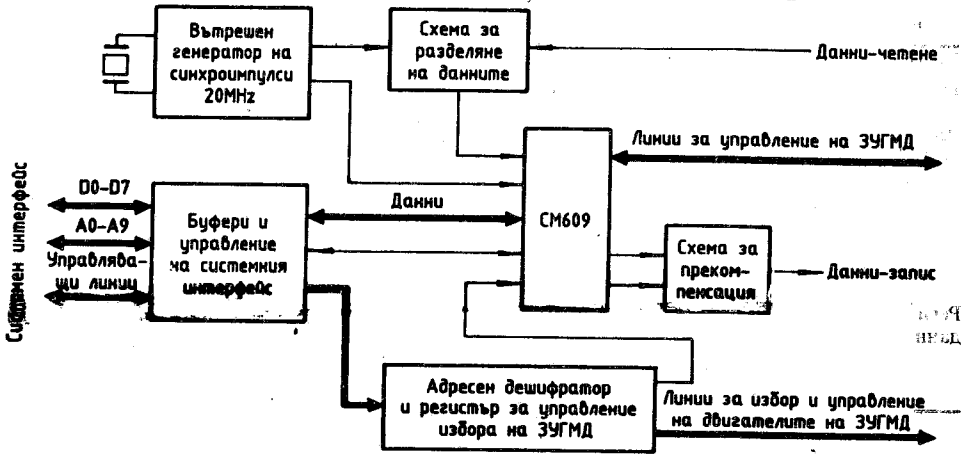
Чрез линията ЗАБРАНА НА ЗАПИСА се указва на контролера, че не трябва да записва информация върху носителя. Обикновено сигналът по тази линия е неактивен, т.е. записът е разрешен. Сигналът се активира, в случай че липсва или е затворен отворът за защита на записа върху дискетата. Тогава механичният или фотоелектронен датчик на ЗУГМД, който следи този отвор, издава сигнал по линията ЗАБРАНА НА ЗАПИСА.

Линията за предаване на информация от ЗУГМД към контролера е ДАННИ-ЧЕТЕНЕ. По нея се предават информационните импулси в последователен код така, както се четат от носителя. В контролера тази поредица се декодира и се възпроизвеждат прочетените от диска данни.

3.4.3. КОНТРОЛЕР НА ЗУГМД

Контролерът на ЗУГМД се свързва към централния процесор чрез системната шина на компютъра, а към ЗУГМД — чрез описания по-горе интерфейс.

Общата структура на контролера, прилаган в персоналните компютри ЕС1831, ЕС1832, Правец-16, Правец-16Н и други е показана на фиг.3.36. Основните блокове на контролера са тактов генератор, регистър за управление, схема за прекомпенсация на записа, схема за разделяне на данните, дешифратор на адреса и ИС СМ609 (18272А).



Фиг.3.36. Обобщена блокова схема на контролера на ЗУГМД на 16-разреден ПК

Тактовият генератор изработва тактови импулси с честота 16 МНз, необходими за управление на логическите схеми на контролера. Той е изпълнен с кварцов резонатор, който поддържа честотата на генератора с висока точност.

Дешифраторът на адреса е предназначен да опознава адресите на регистрите на контролера на ЗУГМД, когато централният процесор или контролерът за директен достъп 8237А-5 се обръща към него за обмен на информация. Контролерът на ЗУГМД разполага с три програмно достъпни регистъра, чиито основни характеристики са посочени в табл. 3.35.

Схемата за разделяне на данните служи за отделяне на импулсите за данни от синхронизиращите импулси при четене от ЗУГМД. Действието на схемата е непосредствено свързано с метода на кодиране на данните при запис върху дискетата. На фиг.3.37 е илюстрирано разделянето на данните от синхроимпулсите в случай на кодиране по метода на модифицираната честотна модулация (МФМ). D_m представлява импулсната поредица в код МФМ, която е била записана върху носителя. При операция четене същата поредица D_m се извежда от ЗУГМД по линията ДАННИ—ЧЕТЕНЕ на интерфейса.

За да се декодират данните, необходимо е те да бъдат отделени от синхроимпулсите, тъй като в D_m те са записани заедно. Отделянето на

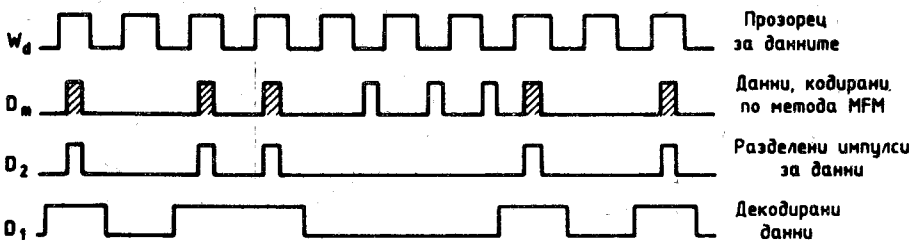
импулсите за данни от поредицата се извършва в схемата за разделяне. В нея се формира импулсна поредица W_d , наречена *прозорец за данните*. С помощта на прозореца от D_m се отделят само тези импулси, които попадат в неговото високо ниво (когато прозорецът е 1). Така на изхода на схемата се получава поредицата D_2 , в която логическа 1 се представя с импулс, а логическа 0 — с липса на импулс.

Таблица 3.35

Регистри на контролера на ЗУГМД

Наименование на регистъра	Адрес във В/И памет	Местоположение	Функции при обмена с централния процесор	Функции при обмена с контролера за ДДП
Управляващ регистър	3F2	Интегрална схема 74LS273 (74LS175)	Входен — съдържа информация за управление на ЗУГМД	Няма
Регистър на състоянието	3F4	Интегрална схема СМ609	Изходен — съдържа информация за състоянието на контролера след завършване на операцията със ЗУГМД. Работи с прекъсване по линия IRQ5	Изходен — съдържа прочетените от ЗУГМД данни, които се записват в ОП
Регистър за данни	3F5	Интегрална схема СМ609	Входен — приема командите за следващата операция със ЗУГМД	Входен — приема от ОП данните за запис в ЗУГМД

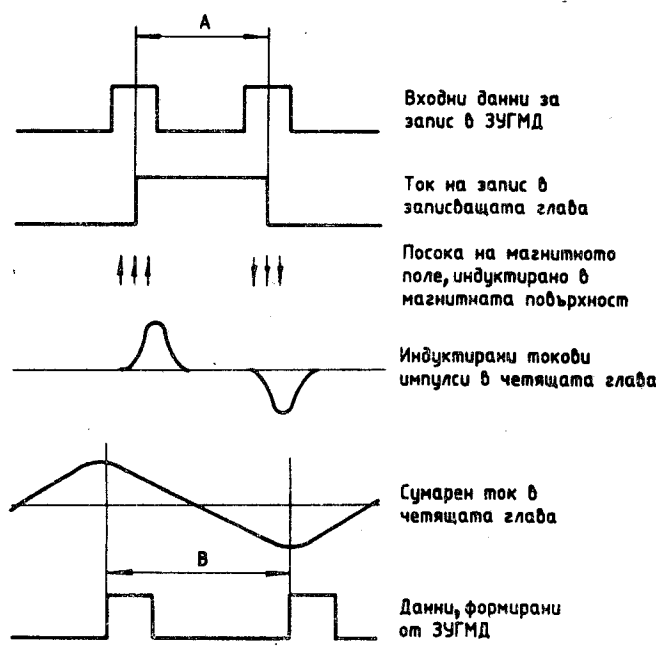
В зависимост от конкретното схемно решение най-широко разпространени са контролерите с т.нар. *цифрово и аналогово разделяне*. При аналоговото разделяне се реализира схема на диференциален усилвател с активна обратна връзка, свързан към генератор на импулси, управляван от ниво. В зависимост от параметрите на елементите чрез тази схема могат да се постигнат много високи характеристики на разделяне. Нейни недостатъци са относително трудната настройка, зависимостта от параметрите на елементите и относително по-високата цена. В контролерите с цифрово разделяне ефектът на обратната връзка се постига с помощта на последователности от импулси, записани в ПЗУ, които се избират в зависимост от входната (за схемата) последователност. Тези схеми имат относително по-лоши работни характеристики, но по-лесно се изработват и настройват, което ги прави и по-евтини.



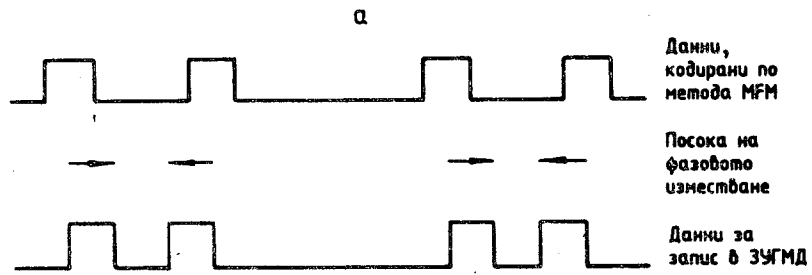
Фиг.3.37. Времедиаграма на схемата за разделяне на данните

Схемата за прекомпенсация е предназначена за фазово изместване на импулсите от записваната върху диска поредица с цел осигуряване на достоверно четене на данните в случаите на последователност от импулси, отстоящи на много кратки интервали от време.

Записът на данни върху магнитния носител става чрез пропускане на ток за запис през четящо-записващата глава на ЗУГМД. Както е показано на фиг.3.38а, токът за запис променя посоката си при всеки импулс от входната поредица.



A - начално разстояние (при запис)
 B - увеличено разстояние (при четене)



б

Фиг.3.38. Времедиаграма на импулсната поредица при запис и четене от ЗУГМД:
 а — без прекомпенсация
 б — с прекомпенсация

При четене е необходимо да се възстанови точно входната импулсна поредица, от която да се декодират съхраняваните данни. Получените в четящата намотка на главата токови импулси имат алтернативно променяща се полярност — положителна или отрицателна. Тук е важно да се отбележи, че сумирането на разнопосочните токови импулси в магнитната глава (фиг.3.38а) води до отдалечаване на техните максимуми. Извежданите от ЗУГМД импулси се формират според моментите на максимум на тока. Поради това, при четене импулсите се появяват на по-голямо разстояние един от друг в сравнение с първоначално записаните. Това е нежелателен ефект, който се проявява особено силно при отстоящи на малък интервал съседни импулси. Той може да доведе до погрешно възпроизвеждане на данните при декодиране. Затова в контролера е включена схема за прекомпенсация при запис. Тя извършва *фазово изместване* на входните импулси в посока, обратна на нежелателното изместване при четене (фиг.3.38б). Схемата анализира предаваната към ЗУГМД импулсна поредица и в зависимост от последователността на импулсите във времето (вида на кодовата дума) измества някои от тях в необходимата посока спрямо нормалната им позиция. Така предварително по време на записа се компенсира изместването на импулсите, предизвиквано от особеностите на процеса четене. Величината на изместването на даден импулс зависи както от кодовата дума, в която той участва, така и от конкретните физически характеристики на ЗУГМД.

Схемата за прекомпенсация се реализира с помощта на брояч, изпълнен на интегрална схема 74LS153, и регистър от типа 74LS175, свързан в броячна схема. Импулсите за управление на прекомпенсацията се подават от интегралната схема СМ609.

Регистърът за управление (адрес във входно-изходната памет 3F2) е 8-битов. Функционалното предназначение на отделните битове на регистъра е следното:

— битове 0 и 1 — задават избор на един от четирите ЗУГМД, свързани към контролера (на практика комбинациите 10 и 11 не се използват, тъй като в максималната конфигурация на 16-разредните персонални компютри има две ЗУГМД):

00	— избор на ЗУГМД А,
01	— избор на ЗУГМД В,
10	— избор на ЗУГМД С,
11	— избор на ЗУГМД D;

— бит 2 — задава начално установяване на логическите схеми на контролера;

— бит 3 — забранява заявките за директен достъп и за прекъсване към централния процесор от страна на контролера;

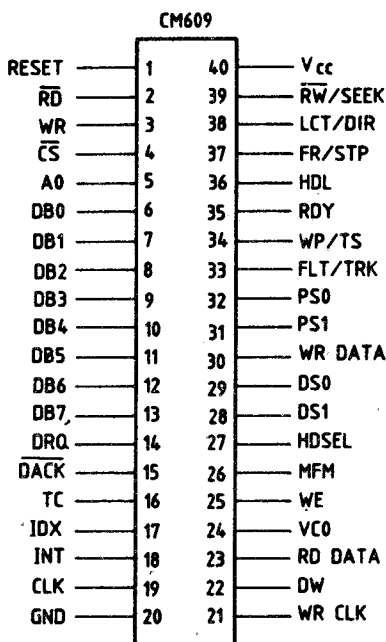
— бит 4 — записът на 1 в него предизвиква включване на двигателя на ЗУГМД А;

— битове 5, 6 и 7 — аналогично на бит 4 управляват включването на двигателите на ЗУГМД В, С и D съответно.

Към управляващия регистър е свързан дешифратор за изработване на управляващите сигнали към ЗУГМД. Към един контролер могат да се включват няколко ЗУГМД, поради което и дешифраторът има изходни линии за избор и за включване на двигателите на няколко ЗУГМД.

Основният функционален блок на контролера на ЗУГМД е интег-

ралната схема CM609 (аналог на I8272A), реализирана в 40-изводен корпус. Разпределение на изводите и сигналите на CM609 е дадено на фиг.3.39.



Фиг.3.39. Разположение на изводите и сигналите на ИС CM609

CM609 изпълнява набор от 15 команди, между които основните са: позициониране, форматиране на дискетата, четене, запис, проверка на състоянието и др. Изпълнението на всяка команда се състои от три фази: подготвителна, изпълнителна и заключителна. Интегралната схема CM609 преминава автоматично в **подготвителната фаза** веднага след подаване на сигнала за начално установяване (бит 2 в регистъра за управление). В подготвителната фаза централният процесор предава към ИС (регистъра за данни) управляващите байтове, които включват кода на операцията и останалите параметри, необходими за нейното изпълнение. На базата на тази информация CM609 извършва в **изпълнителната фаза** действията, зададени в командата. В **заключителната фаза** на командата в регистъра на състоянието се записва последователност от байтове, характеризиращи резултата от изпълнението на операцията. Неговото съдържание информира централния процесор за условията, при които завършва операцията. Подробна информация за структурата на командите, действията на интегралната схема CM609 и съдържанието на байтовете за състояние при завършване на операцията е дадена в приложение 7.

При операция запис регистърът на данните на CM609 се използва като буфер, в който се подават побайтово данните от страна на процесора.

ра. ИС приема данните от регистъра, преобразува ги в последователна форма като поредица от битове и след това ги кодира по метода MFM. В този вид данните преминават през схемата за прекомпенсация и се предават по интерфейса към ЗУГМД.

При операция четене данните от ЗУГМД постъпват в схемата за разделяне, където се отделят и игнорират синхроимпулсите. След това данните се приемат от интегралната схема СМ609, която ги декодира и преобразува в паралелен вид (по байтове). Отделните байтове се буферират временно в регистъра на данните, откъдето се прехвърлят в оперативната памет на компютъра.

Предаването на информацията за състоянието на СМ609 става с помощта на прекъсване. За целта се използва линията за приоритени прекъсвания IRQ5, която е свързана към интегралната схема.

Обменът на данни между контролера на ЗУГМД (респективно ИС СМ609) и централния процесор (неговата памет) се извършва чрез директен достъп до паметта. За целта се използва каналът за ДДП с номер 2. В контролера са реализирани съответните схеми, осигуряващи обмена. Самата интегрална схема СМ609 има възможност за обмен на данни чрез ДДП. От особено важно значение е сигналът ТС (terminal count), който уведомява контролера за ДДП 8237А-5 в централния процесор за завършена операция на обмен. В контролера на ЗУГМД е предвидена възможност за забрана на обмена с ДДП чрез бит 3 от регистъра за управление.

3.4.4. ЗАПОМНЯЩИ УСТРОЙСТВА С ТЪВЪРД МАГНИТЕН ДИСК

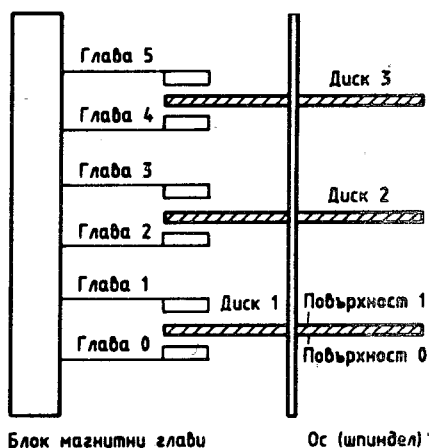
Запомнящите устройства с твърд магнитен диск (ЗУТМД), които се използват в 16-разредните персонални компютри, са известни преди всичко като дискове от тип уинчестър. Броят на магнитните дискове в едно ЗУТМД е от 2 до 6, затова се въвежда понятието *дисков пакет*. Дисковият пакет заедно с главите за запис-четене са изолирани от въздействията на околната среда в херметично затворена кутия. Габаритните размери на твърдите магнитни дискове са стандартните за запомнящите устройства, вграждани в персоналните компютри — 133 и 89 mm. Същото важи и за присъединителните (монтажните) размери на ЗУТМД. Използват се устройства с нормална и 1/2 височина.

След включване на захранването дисковият пакет се развърта до номинални обороти и се върти непрекъснато по време на работа на персоналния компютър. Тъй като обемът на съхраняваната информация в едно устройство от този вид е значителен (най-често от порядъка на 10, 20 и повече Мбайта), ЗУТМД се използват съвместно от всички потребители на компютъра.

Магнитните глави на ЗУТМД четат и записват информацията без допир с повърхността на носителя. Дисковият пакет се върти със стандартна скорост 3600 об/мин. Обменът на информация по дисковия интерфейс се извършва със скорост 5 и повече Мбита/с, а времето за достъп е от порядъка 20—40 ms.

В сравнение с устройствата с гъвкав магнитен диск при ЗУТМД плътността на записа е значително по-висока. Линейната плътност най-често е 357 бита/mm, а напречната е в интервала от 10 до 20 пътечки/mm. При тези параметри броят на пътечките върху една повърхност обикно-

вено е от 153 до 640 и така се постига големият обем на съхраняваната информация.



Блок магнитни глави Ос (шпиндел)
Фиг.3.40. Схематичен разрез на дисков пакет в ЗУТМД

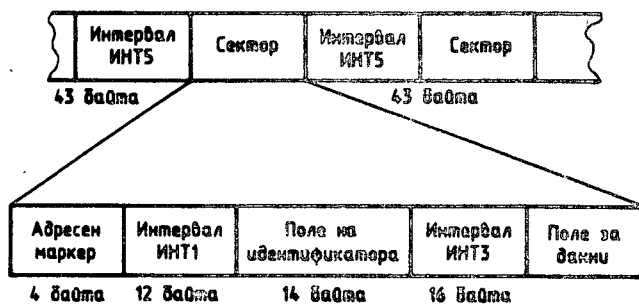
Конструкцията на дисков пакет е показана схематично на фиг.3.40. Той се състои от няколко диска, разположени на обща ос и прикрепени неподвижно към нея. Използват се едновременно всички дискови повърхности. Магнитните глави са обединени в блок и се придвижват заедно в радиална посока спрямо дисковете. Най-често се предлагат пакети от 2, 3 или 4 магнитни диска съответно с 4, 6 или 8 работни повърхности (съответно ЗУТМД има 4, 6 или 8 четящо-записващи глави).

В ЗУТМД се прилагат формати на данните с фиксиран брой сектори в една пътечка. Най-често срещани са форматите със 17 или 32 сектора на пътечка, като обемът на информацията в един сектор е 256, 512 или 1024 байта.

Дадено ЗУТМД има възможности за работа с различни формати в зависимост от компютъра, в състава на който е включено. Конкретният формат на информацията се определя както от вътрешното програмно осигуряване на персоналния компютър, така и от техническите характеристики на контролера на ЗУТМД.

На фиг.3.41 е представен форматът на записа в ЗУТМД, използван в персоналните компютри ЕС1831, ЕС1832, Правец-16, Правец-16Н и др. Структурата на формата е аналогична на тази при ЗУГМД. Всеки сектор съдържа поле на идентификатора, което съхранява служебната информация за сектора, и поле за данни, записвани от потребителя.

Началото на всеки сектор се означава чрез адресен маркер. Аналогично на адресния маркер при ЗУГМД той се записва като кодова комбинация, която не съществува в множеството кодови комбинации при избрания метод на кодиране на информацията върху диска.



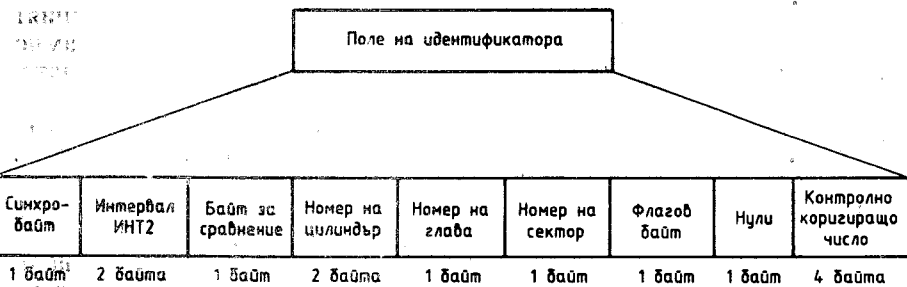
Фиг.3.41. Формат на записа в ЗУТМД

В началото на идентификатора, а също и в началото на полето за данни се записват т.нар. *синхробайтове*. Те са предназначени за синхронизиране на логиката на контролера с честотата на следване на записаната върху диска информация при операциите четене и запис.

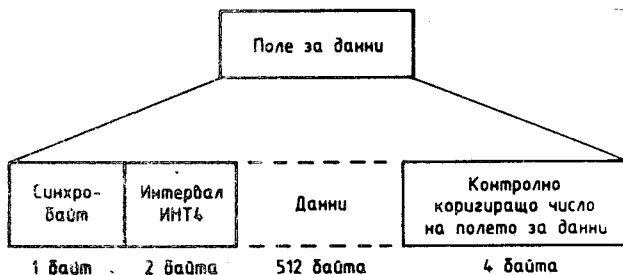
Идентификаторът на сектора (фиг.3.42) съдържа неговия адрес в дисковия пакет, представен като номер на цилиндъра, номер на главата и номер на сектора. Освен тях в идентификатора са поместени байт за сравнение и флагов байт. Байтът за сравнение е еднакво за всички сектори число, чрез което се проверява дали идентификаторът е прочетен вярно. Ако при търсене на даден сектор бъде прочетено друго съдържание на този байт, тогава се сигнализира за грешка при четене на идентификатора. Флаговият байт съдържа флагове указатели за състоянието на сектора (например лош сектор, върху който не може да се записва и чете, и др.).

Съдържанието на полето за данни е показано на фиг.3.43. Размерът на записа в полето *данни* определя основната характеристика на всеки сектор — неговия размер.

Между отделните полета вътре в сектора се разполагат **интервали**, в които не се записва полезна информация. Тяхното предназначение е да осигурят време на логиката на ЗУТМД за преминаване от една към друга операция, извършвани върху две съседни информационни полета. Това е времето за преминаване на главата над дадения интервал. Така например интервалът ИНТ1 е предназначен да осигури време за подготовка за четене или запис на даден сектор след откриване на адресния маркер, Интервалите ИНТ2 (в полето на идентификатора) и ИНТ4 (в полето



Фиг. 3.42. Структура на идентификатора на сектора на ЗУТМД



Фиг. 3.43. Структура на полето за данни в сектора на ЗУТМД

за данни) осигуряват време за подготовка на операциите четене на идентификатора и четене или запис на полето за данни след синхронизиране на логиката на контролера по синхробайтовете.

Интервалът ИНТЗ осигурява време за регистриране на правилно прочетен идентификатор на търсения сектор и подготовка за четене или запис на полето за данни на този сектор. ИНТ5 е междусекторен интервал, който служи за подготовка на логиката за откриване на адресния маркер на следващия сектор.

Контролното число на идентификатора се записва еднократно при запис на идентификатора на сектора, а контролното число на полето за данни се записва при всеки нов запис на данните. Характерно за контролните числа е, че те са предназначени както за откриване на грешки при четене на информацията от сектора (контролиращ код), така и за коригиране на грешки (коригиращ код). В ЗУТМД най-често се използват т.нар. *полиномни коригиращи кодове*. Видът на използвания код и дали той е контролиращ или коригиращ обикновено се определя от възможностите на контролера на ЗУТМД.

При форматиране на ЗУТМД се записва служебна информация и се проверяват полетата за данни. Проверката се извършва чрез пробно записване и четене на полето за данни на всеки сектор. В резултат на това се попълва флаговият байт, указващ състоянието на сектора — доден или не за съхраняване на данни.

За по-ефективно четене и запис се използва такова разполагане на секторите върху пътеката, при което секторите с последователни номера се намират на разстояние N физически сектора един от друг (броят N се нарича кратност). *Кратността на разполагането на секторите* е параметър, който се задава при форматиране на диска. С промяната на кратността може да се постигне оптимална производителност при обмен на дисковата памет. Пример за разполагане на секторите при кратност 6 и 17 сектора на пътека е показан на фиг.3.44.

За да се подготви за работа, ЗУТМД трябва да бъде формиран. Тази процедура е по-продължителна и по-сложна за изпълнение, отколкото при ЗУМД.

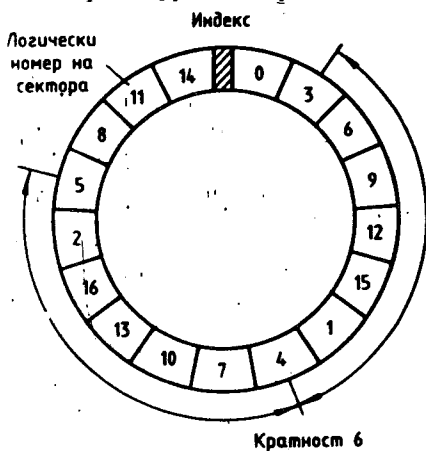
Най-напред се извършва форматиране на физическо ниво по един от следните два начина:

1. Форматиране с програма, записана в EPROM с базовата система за вход-изход (БСВИ) на контролера на ЗУТМД.

Не всички версии на БСВИ разполагат с такава програма. За да бъде стартирана, е необходимо да се зареди системната програма DEBUG. В командния режим на DEBUG се въвежда:

$g = c800:адрес\ на\ отместването$

Адресът на отместването е характерен за всяка версия на БСВИ и се дава в описанието на контролера на ЗУТМД.



Фиг.3.44. Разполагане на секторите в ЗУТМД

В резултат от изпълнението на командата на екрана на видеомонитора се появяват съобщенията на програмата за форматиране на физическо ниво. С тяхна помощ потребителят може да зададе различни параметри на ЗУТМД, които определят характеристиките на системата контролер — диск.

2. Форматиране с програмата за тестване на ЗУТМД.

По този начин се работи, когато версията на БСВИ не разполага с програма за форматиране на физическо ниво. За целта се зарежда тестовата дискета на персоналния компютър. Избира се режим на проверка и след определяне на системната конфигурация се стартира еднократна проверка на ЗУТМД и неговия контролер. Една от функциите на теста е физическото форматиране на ЗУТМД. След изпълнението на функцията се излиза от режимите на тестовата дискета и се зарежда дискета с ДОС.

След завършване на форматирането на физическо ниво по един от описаните по-горе два начина, се преминава към изпълнение на системната програма FDISK. Тя служи за предварително определяне на дяловете, в които работи операционната система, и е необходима, за да се подготви информацията за логическото форматиране на ЗУТМД.

Последната операция е самото логическо форматиране с помощта на системната програма FORMAT. Тя записва всички незапълнени полета за служебна информация, задава размера на сектора (стандартно — 512 байта), определя лошите сектори, записва FAT и всичко необходимо за работата на ЗУТМД в конфигурацията на персоналния компютър.

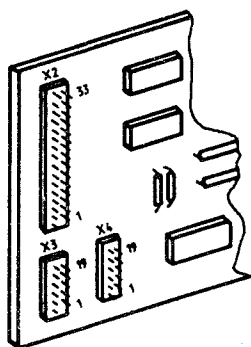
3.4.5. ИНТЕРФЕЙС МЕЖДУ КОНТРОЛЕРА И ЗУТМД

Както при ЗУГМД свързването на ЗУ с твърд магнитен диск към неговия контролер става посредством интерфейсни линии, които са стандартизирани. В 16-разредните персонални компютри се използва интерфейсът Shugart ST-506 (фиг.3.45), който е стандартизиран за страничните членки на СИБ, в документа РТМ „ЕС ЭВМ. Персональные профессиональные ЭВМ — Ряд 3. Принципы работы“. Характерно за него е, че сигналите за управление и състояние на ЗУТМД са изведени на отделен куплунг. Те се предават чрез общ и за двете ЗУ интерфейсен кабел. Изборът на едно от двете устройства се осъществява чрез завъртане на кабела (фиг.3.46). Данните за запис-четене се предават отделно за двете ЗУТМД. За целта са предвидени два куплунга на контролера. Всяко ЗУ има собствен кабел, който се включва към съответния куплунг на контролера.

Повечето интерфейсни линии и сигнали, предавани между ЗУТМД и неговия контролер, имат същото предназначение като разгледаните в т.3.4.2.

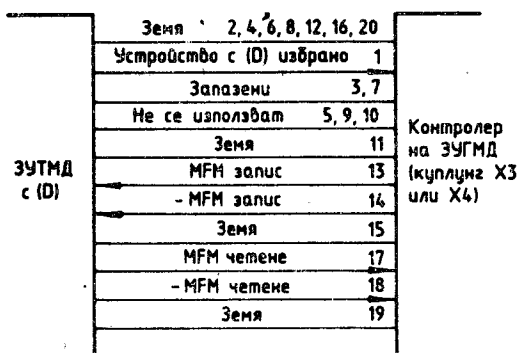
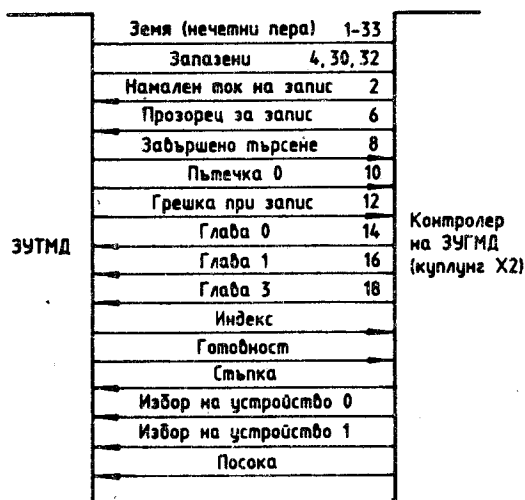
Сигналите по линии ГЛАВА 2, ГЛАВА 1 и ГЛАВА 0 са изходни за контролера. Те формират код за избор на една от 8 глави за запис-четене в ЗУТМД, с което се изпълнява текущата операция.

Изходният сигнал по линията ПРОЗОРЕЦ ЗА ЗАПИС се издава от контролера, когато се изпълнява операция за запис в ЗУТМД и данните са подадени на съответната линия към запомнящото устройство. На практика това е сигнал, който определя наличието на данни за запис в



ЗУТМД и участва в управлението на логиката за дешифриране в запомнящото устройство.

Сигналите по линии ИЗБОР НА УСТРОЙСТВО 0 и ИЗБОР НА УСТРОЙСТВО 1 са също изходни. Те не могат да бъдат едновременно в активно състояние и осигуряват избора на едно от двете устройства, включени към контролера. За разлика от ЗУГМД тук не е необходимо подаването на сигнал за развъртане на двигателя, тъй като дисковият пакет се върти непрекъснато, докато е включен персоналният компютър. Поради завъртането на интерфейския кабел има значение мястото на включване на ЗУТМД. Устройството, свързано към първия куплунг, се избира като ЗУТМД D, а устройството, свързано към втория куплунг — като ЗУТМД С.



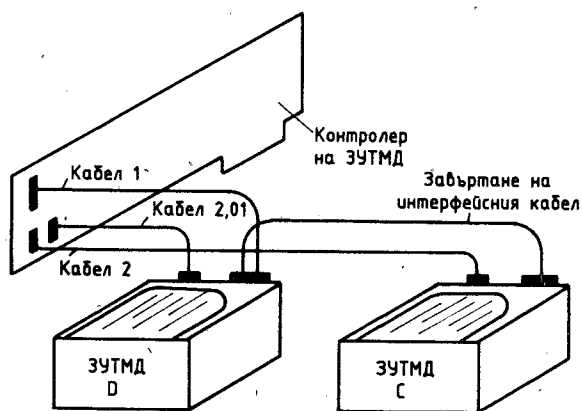
Фиг.3.45. Интерфейс между ЗУТМД и контролера в 16-разредния ПК

По входната за контролера линия ЗАВЪРШЕНО ТЪРСЕНЕ се подава сигнал за успешно завършена операция търсене от избраното ЗУТМД. Сигналът е необходим на контролера, за да преустанови изпълнението на текущата операция.

Сигналът по линията ГРЕШКА ПРИ ЗАПИС е входен и уведомява контролера за възникнала неизправност в тракта за запис на ЗУТМД или за неуспешно завършила поради други причини операция за запис.

Последната линия от кабела за управление и предаване на състоянието на ЗУТМД — ГОТОВНОСТ, е входна за контролера. Сиг-

налтът за готовност се издава след завършване на операциите за начално установяване на ЗУТМД (развъртане на двигателя, проверка на позициониращата система и установяване на главите на цилиндър 0) и след изпълнението на операциите за възстановяване.



Фиг.3.46. Схема на включване на интерфейсни кабели към ЗУТМД

Интерфейсните кабели за предаване и приемане на данни от ЗУТМД имат еднакво разпределение на сигналите, но всеки от тях е свързан към съответното устройство — С или D. Поради тази причина наименованията на сигналите на фиг.3.45 се различават по последната си буква.

Сигналт УСТРОЙСТВО X ИЗБРАНО е входен за контролера и се издава по съответната линия на интерфейсния кабел като отговор на управляващата последователност за избор на ЗУТМД. Този сигнал е потвърждение, че схемите за управление на устройството са изправни и че то е готово за изпълнение на текущата операция.

Двойката линии MFM ЗАПИС осигурява предаването на данните за запис от контролера към ЗУТМД. Аналогично по двойката MFM ЧЕ-ТЕНЕ се получават прочетените от устройството данни. Използването на две линии с различна полярност се определя от високата скорост на предаване на информацията по интерфейса между ЗУТМД и неговия контролер — 5 и повече Мбита/с. За осигуряване на добра шумоустойчивост на информационните сигнали приемните страни (в ЗУТМД и в контролера) са реализирани като диференциални усилватели. В предавателната страна се използват предаватели с диференциален изход.

Направените изследвания показват, че при така изпълнения интерфейс между ЗУТМД и контролера се постига висока достоверност на обмена при дължина на интерфейсните кабели до 50—60 cm. Самите кабели са лентови и е осигурено редуването на сигнални линии с линии, които са заземени. Такава конструкция при определени изисквания към параметрите на лентовия кабел има свойствата на усукана двойка от проводници.

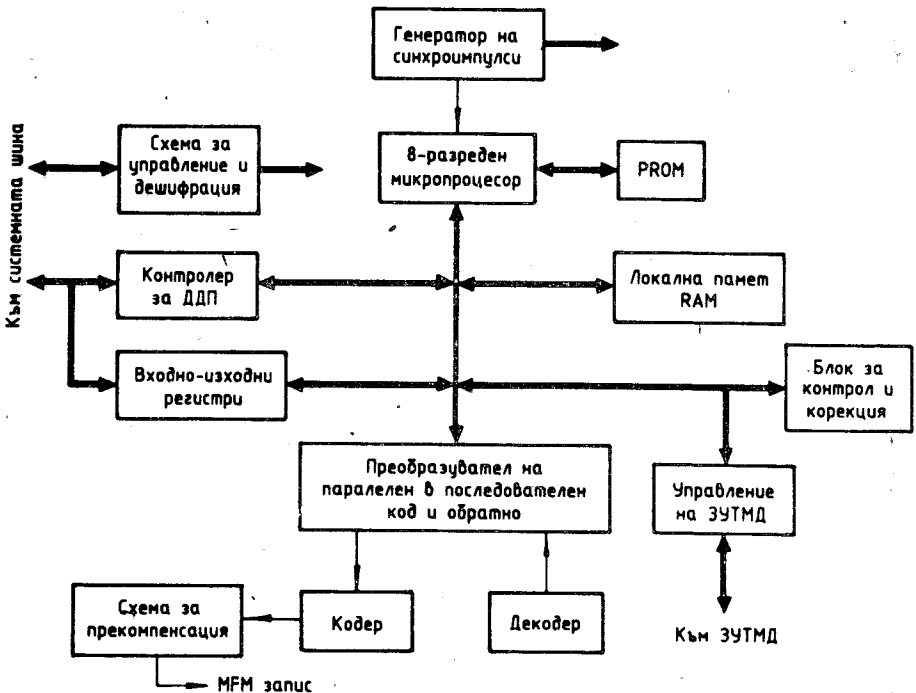
3.4.6. КОНТРОЛЕР НА ЗУТМД

Дисковите устройства с твърд носител поставят значително по-големи изисквания към организацията на обмена с компютъра, тъй като съхраняват голям обем информация и имат висока скорост на обмен на данни.

Поради това контролерите на ЗУТМД са по-сложни. Обикновено се прилага микропроцесорно управление на основните операции — позициониране, запис, четене, търсене, форматиране и др. Така се създава възможност за входно-изходен обмен паралелно с работата на централния процесор и се повишава производителността на персоналния компютър. За целта се използва директен достъп до паметта. Секторите в ЗУТМД имат фиксирана дължина и в контролера има буфер за данни с обем един сектор. За да се повиши достоверността на записаната в ЗУТМД информация, в повечето контролери се прилагат коригиращи кодове. Процедурата за корекция на грешки се вгражда в алгоритъма на управляващата програма на контролерния микропроцесор. Често се използват специализирани интегрални схеми, които генерират полинома на коригиращия код, проверяват правилността на данните и при необходимост коригират грешките.

Контролерът се свързва от едната страна към системната шина на персоналния компютър, а от другата — към интерфейса на ЗУТМД.

Блоквата схема на един типичен контролер на ЗУТМД е дадена на фиг.3.47. Устройството е изградено като автономен микропроцесорен модул на основата на микропроцесора Z-80A.



В постоянната памет са записани програмите за управление на микропроцесора. Те съставят вътрешното програмно осигуряване на контролера. Отделните програми са предназначени за управление на микропроцесорната система при различните операции по обмена на информация със ЗУТМД и за диагностика на възникналите неизправности и грешки.

Интегралната схема за управление на директния достъп до паметта (ДДП) управлява обмена на данните между контролера (буфера за данни) и ЗУТМД, а също между контролера и оперативната памет на компютъра.

Контролерът разполага със собствена памет от тип RAM, наречена *локална памет*. Тя се използва като работна област на микропроцесора и като буфер за данни (секторен буфер). В буфера могат да се съхраняват данни с обем, равен на един сектор.

Данните се преобразуват от последователен в паралелен код в **схемите за преобразуване**. Записът върху дисковата повърхност е кодиран по метода на модифицираната честотна модулация (МФМ). Физическите особености на магнитния запис налагат необходимостта от прекомпенсация. Принципът на действие на съответната схема е както при устройствата с гъвкави магнитни дискове, но за разлика от тях в ЗУТМД се прилага т.нар. *увеличена прекомпенсация*. Чрез нея се постига по-голямо фазово изместване на импулсите при запис на вътрешните цилиндри на дисковия пакет. Номерът на цилиндъра, след който се извършва увеличената прекомпенсация, се задава като параметър на контролера в зависимост от типа на използваното ЗУТМД.

Контролерът използва **схема за разделяне на данните**, чието предназначение е аналогично на схемата за разделяне в контролера на ЗУГМД. Съществената разлика е свързана със значително по-високата скорост на обмен със ЗУТМД — 5 и повече Мбита/с. Това налага изпълнението на по-прецизна схема за разделяне на данните от синхроимпулсите.

В **схемата за дешифриция** на входно-изходните адреси се изработват сигналите за избор на регистрите в контролера на ЗУТМД. Контролерът разполага с четири входно-изходни регистъра, чието разпределение и функции е дадено в табл. 3.36.

Таблица 3.36

Регистри на контролера на ЗУТМД

Наименование на регистъра	Адрес във В/И памет	Функция при команда запис по системната шина	Функция при команда четене от системната шина
Регистър за данни	320	Съхранява данните, получени от системната шина	Съдържа данните за четене от системната шина
Регистър на състоянието	321	Предизвиква начално установяване на схемите и регистрите на контролера	Съдържа информация за състоянието на апаратната част на контролера
	322/	Изработва импулса ИЗБОР в контролера	Запазена
	323	Съхранява границите при обмена с ДДП и маската на прекъсването	Няма

Осембитовият регистър за данни се състои на практика от няколко регистъра (*регистров файл*), подредени в стек, като в даден момент е достъпен само един от тях. Регистърът съдържа данните за запис или четене, командите, техните параметри и информация за състоянието на контролера след изпълнение на командата.

Байтът за състояние издаван след изпълнението на всяка команда има следния формат:

Бит	7	6	5	4	3	2	1	0
	0	0	д	0	0	0	г	0

Битове 0, 2, 3, 4, 6 и 7 — винаги са равни на нула.

Бит 1 (г) — когато е установен (единица) показва, че е възникнала грешка по време на изпълнение на командата.

Бит 5 (д) — показва логическия номер на устройството (0 — ЗУТМД С или 1 — ЗУТМД D).

Ако прекъсванията са разрешени (чрез маската във входно-изходен регистър 323), контролерът подава заявка за прекъсване и предава байта за състояние към системната шина.

Ако бит 1 в байта за състояние е установен в единица, контролерът предава към системната шина чрез заявки за прекъсване още 4 байта, които уточняват информацията за възникналата неизправност във формата, показан в табл. 3.37.

Съдържанието на полетата тип на грешката и код на грешката, които са най-съществени за определяне на вида на неизправността, е дадено в приложение 8. Грешките се разделят на четири основни типа: в управлението на ЗУТМД, в тракта запис-четене на ЗУТМД, във формата на командите, в апаратната част на контролера.

Таблица 3.37

Формат на байтовете за уточняване на състоянието след възникване на грешка в системата контролер — ЗУТМД

Байт	Битове							
	7	6	5	4	3	2	1	0
0	а	0	Тип на грешката	Код на грешката				
1	0	0						
2	Номер на цилиндър — старша част			Номер на глава				
3	Номер на цилиндър — младша част			Номер на сектор				

а — бит за валиден адрес, който се установява само когато предишната команда изисква подаването на адреса на диска;

д — бит, показващ логическия номер на диска (със значението в байта за състояние).

За определяне на операцията със ЗУТМД към контролера се изпраща блок от 6 байта, който се нарича *блок за управление на устройството (БУУ)*. Форматът на този блок е показан в табл. 3.38.

Управляващото поле съдържа информация за физическите параметри на ЗУТМД. Форматът на полето е следният:

Бит	7	6	5	4	3	2	1	0
	в	с	0	0	0	с	с	с

Битове 5, 4 и 3 са винаги равни на нула.

Бит 7 (в) — установен в нула, забранява четирите опита за завършване на операцията, които се правят от контролера при неуспешно изпълнение на всички команди за достъп. Установява се в единица само по време на проверката на работоспособността (качеството на носителя) на ЗУТМД.

Таблица 3.38

Формат на блока за управление на устройството

Байт	Битове							
	7	6	5	4	3	2	1	0
0	Клас на команда			Код на операция				
1	0	0	д	Номер на глава				
2	Номер на цилиндър			Номер на сектор				
	— старша част							
3	Номер на цилиндър — младша част							
4	Кратност или брояч на блока							
5	Управляващо поле							

д — бит, показващ логическия номер на диска (със значението в байта за състояние).

Бит 6 (е) — установен в нула, по време на команди за четене разрешава повторното четене при възникване на грешка в контролния коригиращ код. Ако повторното четене се изпълни успешно, командата завършва без индикация за грешка. Когато бит 6 е установен в единица, не се разрешава повторно четене.

Битове 2, 1 и 0 (с) — определят типа на ЗУТМД и избират параметрите на сигнала СТЪПКА. Значението им се определя според табл. 3.39.

Според предназначението си командите, подавани към контролера на ЗУТМД, са няколко типа. Основните команди са четене, запис, форматиране, търсене и позициониране. Диагностичните команди служат за проверка на състоянието на запомнящото устройство и на неговия контролер. Предвидени са още и допълнителни команди за възстановяване след грешки и др. Пълен списък и описание на командите е даден в приложение 9.

Полученият от системната шина команден блок се записва в локалната памет на контролера. Изпълнението на операцията започва с активиране на съответната програма за управление на контролера на ЗУТМД, която е записана в постоянната памет на вътрешния микропроцесор. В зависимост от операцията микропроцесорът програмира предварително интегралната схема за управление на ДДП. Тя се използва за извършване на конкретен тип обмен на данните: между ОП на компютъра и локалната памет на контролера или между локалната памет и ЗУТМД.

При запис в ЗУТМД данните от системния интерфейс постъпват във входно-изходните регистри на контролера по байтове. Пристигащите

байтове се прехвърлят един по един в секторния буфер, където се подреждат във вида, в който ще бъдат записани в полето за данни на сектора. По-нататък чрез ДДП данните преминават през преобразувателя от паралелен в последователен код. Получената поредица от битове постъпва в кодера, където се кодира по метода MFM и се предава по интерфейса към ЗУТМД. Едновременно с преобразуването данните преминават и през блока за контрол и корекция, който изчислява тяхното контролно коригиращо число. То се предава след данните и се записва в определено то за него поле на сектора.

Таблица 3.39

Параметри на позициониращата система на ЗУТМД, определени в управляващото поле на БУУ

Битове			Значение на комбинацията
2	1	0	
0	0	0	Това ЗУТМД не е определено и се избира по подразбиране стъпка с период 3 ms
0	0	1	Не се използва
0	1	0	Не се използва
0	1	1	Не се използва
1	0	0	Период на стъпката 200 μ s
1	0	1	Период на стъпката 70 μ s (определя се от БСВИ)
1	1	0	Период на стъпката 3 ms
1	1	1	Период на стъпката 3 ms

По време на операция четене данните от ЗУТМД постъпват най-напред в схемата за разделяне на данните, която отделя от тях синхронизиращите импулси. След това данните се декодират и постъпват на входа на преобразувателя от последователен в паралелен код. Така получените байтове от данни се прехвърлят чрез ДДП в секторния буфер на локалната памет. По-нататък данните се предават през входно-изходните регистри към системната шина на персоналния компютър.

След като се изпълни дадена команда, контролерът предава обратно към системата байт за състоянието, който информира за успешно или неуспешно завършена операция и за евентуалните причини за неизправността.

Контролерът проверява достоверността на данните и има възможност за коригиране на грешки при четене. Блокът за контрол и корекция изчислява контролното коригиращо число за всеки сектор от данни по време на неговото четене от ЗУТМД. Контролното коригиращо число се сравнява с това, което е записано при записа на сектора върху диска. Когато двете числа са равни, се счита, че данните са прочетени правилно. В противен случай се сигнализира за грешка в прочетените данни. Обикновено се прави опит за отстраняване на грешката чрез повторно прочи-

тане на сектора (ако бит 6 в управляващото поле е нула). Ако отново има грешка, се задейства процедурата за корекция. Тя открива неправилно прочетените битове информация чрез контролното коригиращо число. Тъй като информацията е двоична, коригирането се извършва чрез инвертиране на тези битове. Използваният коригиращ код позволява да се отстраняват както единични, така и пакети от грешки с дължина до 11 бита. Контролното коригиращо число се получава като цикличен код с полином от 32-ра степен.

3.5. ВИДЕОМОНИТОРИ И ВИДЕОКОНТРОЛЕРИ

Възможностите за изобразяване и наборът от функции, които се предлагат на потребителя, зависят до голяма степен от характеристиките на видеомонитора и видеоконтролера.

Съществува голямо разнообразие от видеомонитори и видеоконтролери. Стандарт за персоналните компютри, съвместими с IBM PC/XT, е цветният графичен монитор, управляван от съответния контролер (color graphics adapter — CGA). Някои фирми вместо цветен предлагат монохроматичен графичен видеомонитор. В този случай цветовете се интерпретират от монитора като градации на основния цвят на екрана.

Пример за относително евтин подход за реализиране на системата видеомонитор-контролер е използването на *монохроматичен текстов монитор*. Контролерът (monochrome adapter — MA) изобразява само текстова информация и не позволява да се работи с графика. Предвидена е възможност за използване на псевдографични символи, включени в кодовата таблица на персоналния компютър.

По-нататъшно развитие на системата цветен графичен видеомонитор-контролер осигурява т.нар. *разширен графичен контролер* (enhanced graphics adapter — EGA). Като използва нов видеомонитор, той предлага по-добри характеристики на графичния режим с разделителна способност 640×350 точки и по-богата цветова гама.

За персоналните компютри, съвместими с IBM PC/XT, е разработен и т.нар. *професионален графичен контролер* (professional graphics adapter — PGA). Този контролер работи с видеомонитор с подобрени функционални характеристики. Той осигурява изобразяването на графична информация с разделителна способност 640×480 точки и цветова гама от 256 цвята. Тези възможности се постигат в резултат на микропроцесорното управление на самия контролер, големия обем видеопамет и видеомонитора, който може да работи с различни честоти на редовата и кадрова развивка. Поради това системата видеоконтролер-монитор е по-скъпа и подходяща за действително „професионални“ приложения.

В табл. 3.40 е дадено сравнение на основните параметри на най-широко разпространените системи видеомонитор-контролер за персонални компютри, съвместими с IBM PC/XT.

Видеомониторите могат да бъдат векторни и растерни, черно-бели (монохроматични) и цветни. В персоналните компютри са намерили най-широко приложение растерните монохроматични и цветни видеомонитори.

Основни параметри на най-разпространените системи видеомонитор-контролер

Тип на системата	Графика разделителна способност	Цветова гама в цветова палитра	Интерфейс към видеомонитора	Честота на кадровата развивка, Hz	Честота на редовата развивка, KHz
MA	Няма	Няма	TTL — b/w	50	18
CGA	320 × 200	4 цвята	TTL — RGB	50	15
EGA	640 × 200 640 × 350 (емулация на CGA)	16 цвята от 64	TTL — RGB	60	21
PGA	640 × 480 (емулация на CGA)	256 цвята от 1096	аналогов	61	31

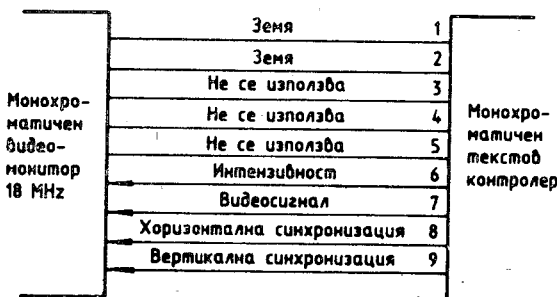
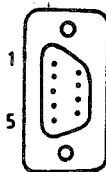
Съкращенията b/w и RGB са обяснени в т. 3.5.1 и 3.5.2.

3.5.1. МОНОХРОМАТИЧЕН ТЕКСТОВ ВИДЕОМОНИТОР И КОНТРОЛЕР

Монохроматичният текстов видеомонитор е предназначен за обработка на текстова информация в 16-разредните персонални компютри. Той е особено удобен за терминалните приложения на персоналните компютри, защото се отличава с по-добро изображение на буквено-цифровите символи върху на екрана.

Видеомониторът работи с честота на кадровата развивка 50 Hz и честота на редовата развивка 18,432 kHz. Броят на точките е 720 × 350. Екранът е изработен с луминофор фосфор P39, който се характеризира с относително голямо послесветене. Максималната честотна лента на видеосигнала е 16,27 MHz.

Матрицата на символа е 9 × 14 точки, което осигурява по-високо

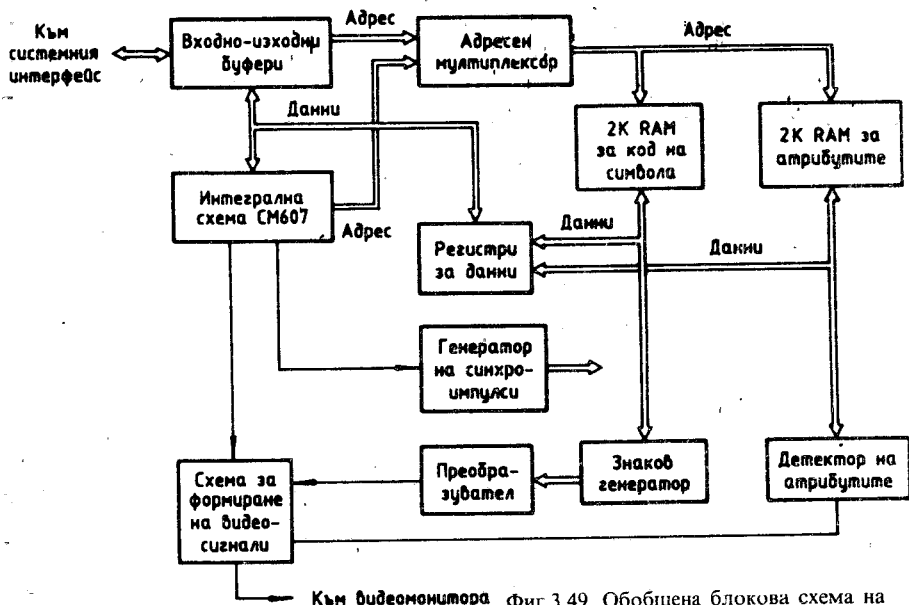


Фиг. 3.48. Интерфейс между контролера и монохроматичния текстов видеомонитор (b/w интерфейс)

качество на изображението. Форматът на екрана е 25 реда по 80 символа. Интерфейсът към видеоконтролера (MA) е TTL с положителен поляритет с изключение на сигнала за редовата развивка и се означава със съкращението b/w (black and white) — фиг.3.48. Видеомониторът се включва към персоналния компютър чрез гъвкав кабел с 9-цифтов куплунг.

Обобщена блокова схема на монохроматичния текстов видеоконтролер е дадена на фиг. 3.49. Той е изграден на базата на интегралната

схема CM607 (аналог на MC6845). Разположението на изводите и тяхното наименование е дадено на фиг.3.50. Вътрешните регистри на CM607



Фиг.3.49. Обобщена блокова схема на монохроматичен текстов видеоконтролер CM607

са достъпни за запис и за четене от централния процесор чрез регистъра за данни и индексния регистър на видеоконтролера. Стойностите за начално установяване на регистрите на интегралната схема CM 607 в монохроматичния видеоконтролер са дадени в табл. 3.41.

Видеоконтролерът разполага с четири регистъра, чиито основни характеристики са показани в табл. 3.42.

Достъпът до регистрите на CM607 се осъществява по следния начин:

- в индексния регистър на видеоконтролера се записва номерът на регистъра (от 0 до 15);

- записът или четенето на избрания регистър става чрез регистъра за данни на видеоконтролера.

За регистрите от R0 до R13 е възможна само операция запис, а регистрите R14 и R15 са достъпни за запис и за четене.

Регистърът за задаване на режима е 8-битов.

CM607			
GND	1	40	VSYNC
RESET	2	39	HSYNC
LPSTB	3	38	RA0
MA0	4	37	RA1
MA1	5	36	RA3
MA2	6	35	RA4
MA3	7	34	RA5
MA4	8	33	D0
MA5	9	32	D1
MA6	10	31	D2
MA7	11	30	D3
MA8	12	29	D4
MA9	13	28	D5
MA10	14	27	D6
MA11	15	26	D7
MA12	16	25	\bar{CS}
MA13	17	24	RS
AV	18	23	\bar{E}
CURSOR	19	22	R/W
V _{cc}	20	21	CLK

Фиг.3.50. Разположение на изводите и сигналите на ИС CM607

Битове 1, 2, 4, 6 и 7 не се използват.

Бит 0 определя формата на екрана (разделителната способност):

0 — 25 реда по 40 символа;

1 — 25 реда по 80 символа (висока разделителна способност).

Бит 3 издава разрешение за видеосигнала:

0 — екранът на видеомонитора е тъмен;

1 — нормален режим на работа.

Бит 5 задава режим на мигане (на целия екран):

0 — нормален режим на работа;

1 — изобразяваните символи мигат (последователно светване и из-
тасване с период около 1 s).

Таблица 3.41

Използване на вътрешните регистри на CM607 в монохроматичния видеоконтролер

Номер на ре- гистъра	Наименование (функция) на регистъра	Стойност за начално уста- новяване
R0	Период на редовата развивка	61
R1	Брой на символите в реда	50 (28)
R2	Позиция на импулса VSYNC	52
R3	Продължителност на импулса VSYNC	0F
R4	Период на кадровата развивка	19
R5	Точна настройка на периода на кадровата развивка	06
R6	Брой на изобразяваните редове	19
R7	Позиция на импулса HSYNC	19
R8	Задаване на режим на презредова развивка	02
R9	Брой на линиите в един символ	0D
R10	Начална линия на курсора	0B
R11	Крайна линия на курсора	0C
R12	Адрес на изобразяваната страница във видеопаметта (младша част)	00
R13	Адрес на изобразяваната страница във видеопаметта (старша част)	00
R14	Адрес на курсора спрямо началото на изобразяваната страница (младша част)	00
R15	Адрес на курсора спрямо началото на изобразяваната страница (стар- ша част)	00

Таблица 3.42

Регистри на монохроматичния видеоконтролер

Наименование на регистъра	Адрес във В И памет	Функция при команда запис по системната шина	Функция при команда четене от системната шина
Индексен регистър	384	Регистър на ИС CM607 — осигу- ржава избор на един от вътрешни- те ѝ регистри	Няма
Регистър за данни	3B5	Съхранява данните, получени от системната шина Като регистър на CM607 — осигурява записа в един от вътрешните регистри	Съдържа данните за четене от системната шина Като регистър на CM607 — осигурява четене на един от вътрешните регистри
Регистър за задаване на режима	3B8	Задава режимите на работа на видеоконтролера	Няма
Регистър на състоянието	3BA	Няма	Съдържа информация за съ- стоянието на видеоконтролера

Регистърът на състоянието също е 8-битов. Значението на битовете му при четене е следното:

Битове 1, 2, 4, 5, 6 и 7 не се използват.

Бит 0 съдържа информация за синхронизацията по редове. Той задава интервала от време, през който е възможен достъпът на централния процесор до видеопаметта:

0 — достъпът е забранен;

1 — достъпът е разрешен.

Бит 3 съдържа информация за синхронизацията по кадри. Използва се от тестовите програми на видеоконтролера за определяне на интервала от време, през който те имат достъп до видеопаметта:

0 — достъпът е забранен;

1 — достъпът е разрешен.

Видеопаметта е основен блок в структурата на видеоконтролера. Достъпът до нея е реализиран както от страна на интегралната схема СМ607, така и от системната шина на персоналния компютър. Централният процесор записва информацията, която трябва да бъде изобразена на екрана, а СМ607 и схемите на видеоконтролера изпълняват функциите по формиране на видеосигнала.

Принципите на работа на 16-разредните персонални компютри предвиждат наличието на два байта памет за всеки символ, изобразяван на екрана на видеомонитора — един байт за кода на символ и един байт за неговите атрибути. Това определя максималния размер на видеопаметта за всеки видеоконтролер, който работи в текстов режим.

Обемът на видеопаметта в монохроматичния текстов видеоконтролер е 4 Кбайта (за изобразяването на 25×80 символа са необходими 2000 байта). Паметта е разделена условно на две части — памет за кода на символа и памет за атрибутите. На практика в адресното пространство на видеопаметта за всеки символ са отделени по два последователни байта.

Съдържанието на байта за атрибутите на символа е следното:

Бит	7	6	5	4	3	2	1	0
	м	ч	з	с	и	ч	з	с

Битове 0, 1 и 2 определят „цвета“ на символа (с — син, з — зелен, ч — червен).

Бит 3 определя интензивността на символа (и — интензивност).

Битове 4, 5 и 6 определят „цвета“ на фона.

Бит 7 определя режим на мигане на символа (м — мигане).

Думата *цвет* е поставена в кавички, защото функцията на тези атрибути за разглеждания контролер има друг смисъл. Валидните комбинации на атрибутите за определяне на цвят на символа и цвят на фона при монохроматичния текстов видеоконтролер са дадени в табл. 3.43.

Информацията, записана във видеопаметта като код на символа постъпва на входа на **знакогенератора** (фиг.3.50). Той е реализиран като EPROM и в него са записани изображенията на всеки символ в матрица 9×14 точки. Кодовата таблица на 16-разредните персонални компютри се състои от 256 символа и в знакогенератора се съхраняват 256 изображения (приложение 11). На практика знакогенераторът може да се раз-

глежда като преобразувател на кода в матрица на изображението.

Предназначението на блока за изработване на видеосигнала е да подготви поредицата от данни, управляващи видеомонитора. Като входна информация той получава преобразуваните в последователен вид изображения на символите, декодираните атрибути и сигналите за управление на синхронизацията на монитора. На изхода му се формират сигналите, които се подават към интерфейса на видеомонитора. Схемите на блока работят под управлението на няколко основни синхросигнала, които осигуряват качеството на изображението.

Таблица 3.43

**Валидни комбинации на атрибутите за цвят
в монохроматичния текстов видеоконтролер**

Атрибути за цвят на символа			Атрибути за цвят на фона на символа			Функция на комбинацията
ч	з	с	ч	з	с	
0	0	0	0	0	0	Няма изображение на екрана
0	0	0	0	0	1	Подчертаване на символ
0	0	0	1	1	1	Светли символи/тъмен фон
1	1	1	0	0	0	Инверсно изображение (тъмни символи/светъл фон)

Интегралната схема CM607 и централният процесор на персоналния компютър осъществяват достъпа си до видеопаметта в режим на времеделене. Синхронизирането на тяхната работа се осигурява от програмите в драйвера на видеоконтролера. Когато изпълняваната от централния процесор операция за запис или четене от видеопаметта не може да завърши в установените граници, видеоконтролерът въвежда тактове за неготовност Tw посредством сигнала от системния интерфейс I/O CH READY.

Организацията на монохроматичния текстов видеоконтролер позволява четенето едновременно на два байта от видеопаметта през период от 553 ns. По този начин се постига много висока скорост на обмен — 1,8 Мбита/s. Поради това интегралните схеми във видеопаметта са с повишени изисквания по отношение на времето за достъп. Началният адрес на видеопаметта в адресното пространство на централния процесор е B0000.

3.5.2. ЦВЕТЕН ГРАФИЧЕН ВИДЕОМОНИТОР И КОНТРОЛЕР

Цветният графичен видеомонитор е предназначен за изобразяване на цветна текстова и графична информация. Количеството на цветовете се определя от възможностите на включения към него видеоконтролер. Това е най-широко разпространеният видеомонитор за 16-разредните персонални компютри.

Видеомониторът работи с честота на редовата развивка 15,75 kHz и честота на кадровата развивка 60 Hz. Броят на точките, изобразявани в режима на графика с висока разделителна способност, е 640 × 200. Мо-

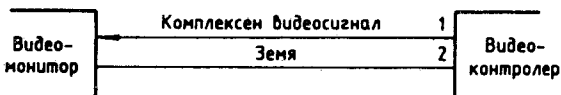
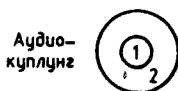
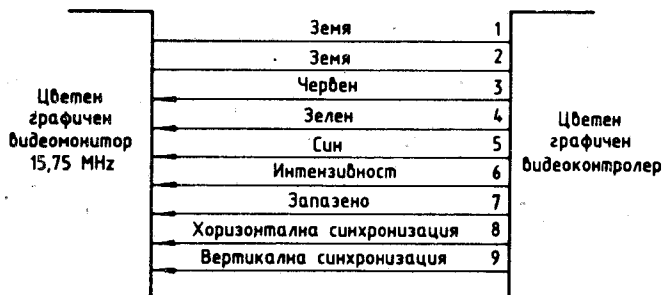
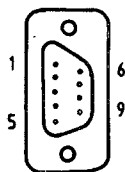
мониторът позволява работа с 16 цвята, получени като комбинация от основните три — червен, зелен и син. Тъй като английското название на основните цветове е съответно red, green и blue, видеомониторите от този клас са известни и като *RGB-монитори*. Честотната лента на видеосигнала в зависимост от режима на видеоконтролера е 7 или 14 MHz.

Матрицата на символите в текстов и графичен режим е 8×8 точки. Форматът на екрана при изобразяване на символи е 25 реда по 80 символа или 25 реда по 40 символа. Според разделителната способност в режим на графика се определят две основни разновидности:

— графика със средна разделителна способност (цветна графика) — полето на екрана се покрива от 320×200 точки, оцветени в 4 цвята;

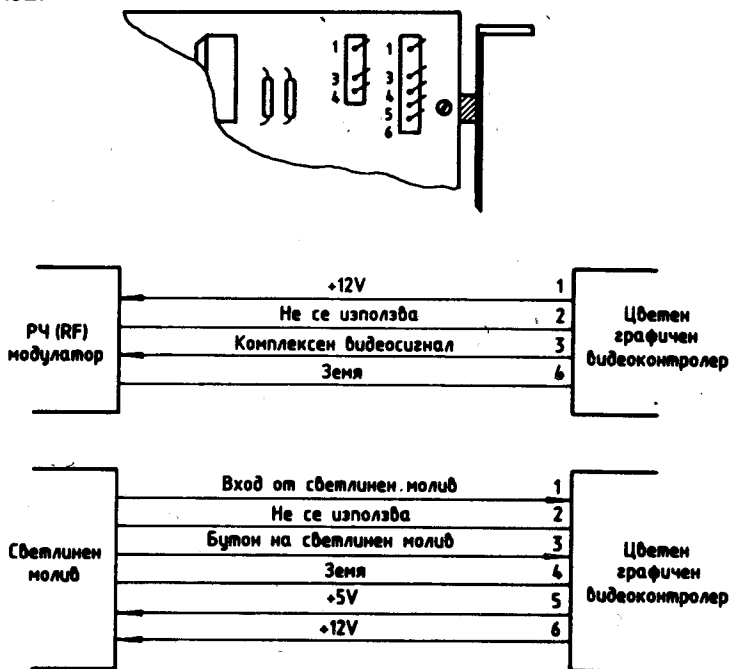
— графика с висока разделителна способност (черно-бяла графика) — полето на екрана се покрива от 640×200 точки, които могат да бъдат само черни или бели.

Интерфейсът към видеоконтролера е TTL и е известен още като *RGB-интерфейс* (фиг.3.51). Видеомониторът се включва към персоналния компютър чрез гъвкав кабел с 9-щифтов куплунг.



Фиг.3.51. Интерфейс между контролера и цветния графичен видеомонитор (RGB-интерфейс)

Освен с видеомонитор, контролерът има възможност за работа и с цветен телевизор (включва се непосредствено или чрез *радиочестотен модулатор*—RF-модулатор). Към него може да се свързва светлинен молив. Интерфейсите към тези допълнителни устройства са дадени на фиг.3.52.

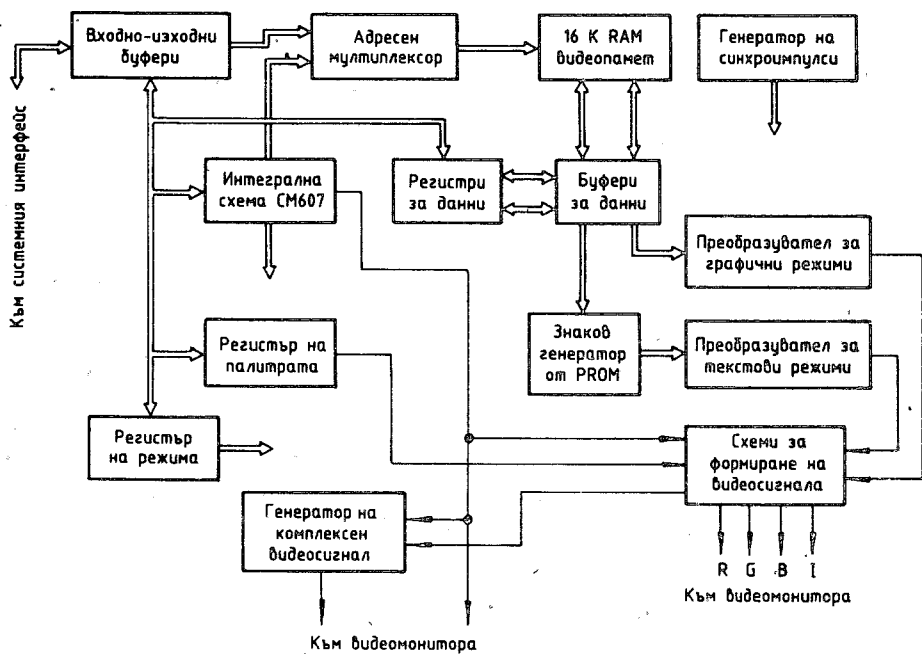


Фиг.3.52. Интерфейс на цветния графичен видеоконтролер към допълнителните устройства

Цветният графичен видеоконтролер може да управлява и монохроматичен графичен монитор; който има RGB-интерфейс. В този случай изображението на екрана няма да бъде цветно, а различните цветове ще се интерпретират като различни полутонове на основния му цвят. Този вариант се прилага широко от някои фирми поради значително по-ниската цена на монохроматичните видеомонитори. Трябва да се отбележи, че някои приложни програми за проектиране работят интензивно с изображения с различен цвят и работата им с монохроматичния вариант на графичната система видеомонитор — контролер е практически невъзможна.

Обобщена блокова схема на цветния графичен видеоконтролер е дадена на фиг.3.53. Той е изграден на базата на специализираната интегрална схема СМ607 (аналог на МС6845). Тъй като в т.3.5.1 са отбелязани някои особености на използването ѝ в 16-разредните персонални компютри и регистровата структура на цветния графичен видеоконтролер е много близка до тази на монохроматичния текстов видеоконтролер, по-нататък се разглеждат само различията и допълнителните възможности на новия контролер.

Стойностите за начално установяване на регистрите на интегралната схема SM607 в цветния графичен видеоконтролер са дадени в табл. 3.44. От нея се вижда, че графичният видеоконтролер в зависимост от режима установява по различен начин регистрите си в начално състояние. Символите XX означават, че съответните регистри не получават начална стойност от драйверната програма. В сравнение с табл. 3.41 са добавени още два регистра — R16 и R17. Информацията в тях е достъпна само за четене.



Фиг. 3.53. Обобщена блок схема на цветен графичен видеоконтролер

Цветният графичен видеоконтролер разполага със седем регистра, чиито основни характеристики са показани в табл. 3.45. Функциите на първите четири съвпадат с показаните в табл. 3.42. Добавени са още три регистра, които управляват новите режими, характерни за графичния видеоконтролер.

Регистърът за задаване на режима е 8-битов.

Битове 6 и 7 не се използват.

Бит 0 определя формата на екрана в текстов режим:

0 — 25 реда по 40 символа;

1 — 25 реда по 80 символа.

Бит 1 определя режима на цветния графичен видеоконтролер:

0 — текстов режим;

1 — графичен режим.

Бит 2 определя цветността на изображението:

0 — цветен режим;

1 — черно-бял (монохроматичен) режим.

Бит 3 издава разрешение за видеосигнала:

0 — екранът на видеомонитора е тъмен;

1 — нормален режим на работа.

Бит 4 задава разделителната способност в графичен режим:

0 — цветна графика (320 × 200);

1 — черно-бяла графика (640 × 200).

Бит 5 задава мигане на екрана в текстов режим или определя цветността на фона в графичен режим 320 × 200:

0 — нормален режим на работа или избор измежду 16 цвята за фона на символа (включва се бит 3 в регистъра за избор на цвета);

1 — изобразяваните символи мигат или избор измежду 8 цвята за фона на символа (не се включва бит 3 в регистъра за избор на цвета).

Таблица 3.44

Използване на вътрешните регистри на СМ607
в цветния графичен видеоконтролер

Номер на регистъра	Наименование (функция) на регистъра	Стойности за начално установяване в режим		
		текст 25 × 40	текст 80 × 40	графика
R0	Период на редовата развивка	38	71	38
R1	Брой на символите в реда	28	50	28
R2	Позиция на импулса В.СИНХ	2D	5A	2D
R3	Продължителност на импулса В.СИНХ	0A	0A	0A
R4	Период на кадровата развивка	1F	1F	7F
R5	Точна настройка на периода на кадровата развивка	06	06	06
R6	Брой на изобразяваните редове	19	19	64
R7	Позиция на импулса Х.СИНХ	1C	1C	70
R8	Задаване на режим на презредова развивка	02	02	02
R9	Брой на линиите в един символ	07	07	01
R10	Начална линия на курсора	06	06	06
R11	Крайна линия на курсора	07	07	07
R12	Адрес на изобразяваната страница във видеопаметта (младша част)	00	00	00
R13	Адрес на изобразяваната страница във видеопаметта (старша част)	00	00	00
R14	Адрес на курсора спрямо началото на изобразяваната страница (младша част)	00	00	00
R15	Адрес на курсора спрямо началото на изобразяваната страница (старша част)	00	00	00
R16	Адрес на светлинния молив спрямо началото на изобразяваната страница (младша част)	XX	XX	XX
R17	Адрес на светлинния молив спрямо началото на изобразяваната страница (старша част)	XX	XX	XX

В табл. 3.46 са показани валидните комбинации от битовете на регистъра и съответните им режими, поддържани от цветния графичен видеоконтролер.

Регистърът за избор на цвета е 8-битов.

Битове 0 (с — син), 1 (з — зелен), 2 (ч — червен) и 3 (и — интензивност) определят цвета на фона в графичен режим 320 × 200, или опреде-

лят втория цвят в графичен режим 640 × 200, или задават цвета на външната рамка на екрана във всички текстови режими.

Бит 4 определя интензивността на фона в текстовите режими:

0 — нормални цветове на фона;

1 — цветовете на фона са с повишена интензивност.

Бит 5 задава номера на палитрата в графичен режим 320 × 200:

0 — втора палитра;

1 — първа палитра.

Битове 6 и 7 не се използват

Таблица 3.45

Регистри на цветния графичен видеоконтролер

Наименование на регистъра	Адрес във В И памет	Функция при команда запис по системната шина	Функция при команда четене от системната шина
Индексен регистър	3D0	Регистър на ИС CM607 осигурява избор на един от вътрешните ѝ регистри	Няма
Регистър за данни	3D1	Съхранява данните, получени от системната шина Като регистър на CM607 — осигурява записа в един от вътрешните регистри	Съдържа данните за четене от системната шина Като регистър на CM607 — осигурява четене на един от вътрешните регистри
Регистър за задаване на режима	3D8	Задава режимите на работа на видеоконтролера	Няма
Регистър на състоянието	3DA	Няма	Съдържа информация за състоянието на видеоконтролера
Регистър за избор на цвета	3D9	Задава базовите характеристики на цвятната гама в графичните режими	Няма
Регистър за нулиране на триера на светлинния молив	3DB	Нулира триера на светлинния молив	Няма
Регистър за възстановяване на триера на светлинния молив	3DC	Възстановява съдържанието на триера на светлинния молив	Няма

Валидни комбинации на битовете за задаване на режима
в цветния графичен видеоконтролер

Таблица 3.46

Битове						Режим на видеоконтролера
0	1	2	3	4	5	
0	0	1	1	0	1	Текстов монохроматичен 25 × 40
0	0	0	1	0	1	Текстов цветен 25 × 40
1	0	1	1	0	1	Текстов монохроматичен 25 × 80
1	0	0	1	0	1	Текстов цветен 25 × 80
0	1	1	1	0	X	Графичен монохроматичен 320 × 200
0	1	0	1	0	X	Графичен цветен 320 × 200
0	1	1	1	0	X	Графичен монохроматичен 640 × 200

Видеопаметта в цветния графичен видеоконтролер също осигурява достъпа от страна на специализираната интегрална схема CM607 и централния процесор. От друга страна, паметта се различава по обем и организация в различните режими на видеоконтролера.

Максималният обем на видеопаметта се определя от изискванията на графичния режим 640 × 200. При него е необходимо съхраняването на информация за 128000 точки, поради което обемът на паметта е 16 Кбайта.

В текстовите режими информацията се записва по описания в т.3.5.1 начин. За всеки символ се отделят по два байта, като първият от тях съхранява кода на символа, а вторият — неговите атрибути. Съдържанието на байта за атрибутите е същото, както и в монохроматичния текстов контролер. Съществената разлика е, че при цветния графичен контролер битовете за цвят имат своя действителен смисъл и определят набор от 8 цвята (с участието на бита за интензивност цветовете стават 16). В резултат на това броят на валидните комбинации нараства значително. Комбинацията 001 за цвят на фона на символа не се тълкува като подчертаване, а предизвиква изобразяването на символа на син фон. Пълен списък на валидните комбинации за цвят на символа заедно с интензивност е даден в табл. 3.47.

Валидни комбинации за цвят на символа заедно
с интензивност при цветния графичен видеоконтролер

Таблица 3.47

и	Битове за цвят на символа			с	Цвят
	ч	з	с		
0	0	0	0	0	Черен
0	0	0	0	1	Син
0	0	0	1	0	Зелен
0	0	0	1	1	Кобалтово син
0	1	0	0	0	Червен
0	1	0	0	1	Виолетов
0	1	0	1	0	Кафяв
0	1	0	1	1	Светлосив
1	0	0	0	0	Тъмносив
1	0	0	0	1	Светлосин
1	0	0	1	0	Светлозелен
1	0	0	1	1	Светъл кобалтово син
1	1	0	0	0	Светлочервен
1	1	0	0	1	Светловиолетов
1	1	0	1	0	Жълт
1	1	0	1	1	Бял

При тази организация на видеопаметта в текстовите режими се получава една характерна особеност на цветния графичен видеоконтролер. В режим 25 реда по 40 символа, за записа им във видеопаметта са необходими 2 Кбайта. Следователно в пълния обем от 16 Кбайта могат да бъдат записани изходните данни за изобразяване на 8 екрана. Един блок от видеопаметта с размер 2 Кбайта се нарича страница. В даден момент е активна само една страница (т.е. SM607 работи в рамките на 2 Кбайта). Паралелно с това програмата може да подготвя изходната информация в останалата част от видеопаметта (в другите страници). Превключването на активната страница става с няколко команди в драйвера на видеоконтролера. Номерирането на страниците е от 0 до 7.

В текстов режим 25 реда по 80 символа един екран се записва в 4 Кбайта. Това е размерът на страницата във видеопаметта. В този режим може да се работи с 4 страници. Номерирането на страниците е от 0 до 3.

В графичните режими информационната единица, изобразявана на екрана, е точката (pixel). В режима на цветна графика — 320×200 , за всяка точка могат да се отделят по два бита от видеопаметта:

Бит	7	6	5	4	3	2	1	0
	ц1	ц0	ц1	ц0	ц1	ц0	ц1	ц0

Така в един байт се записва информацията за четири точки, които ще бъдат изобразени на екрана на видеомонитора. Комбинацията ц1 ц0 определя избора на 4 цвята от 16-те възможни. Поради тази причина се въвежда понятието *набор (палитра)*. Персоналните компютри разполагат с две основни палитри:

Палитра 1	Палитра 2
цвят 1 — кобалтово синьо	цвят 1 — зелено
цвят 2 — виолетово	цвят 2 — червено
цвят 3 — бяло	цвят 3 — кафяво

Значението на различните варианти на комбинацията ц1 и ц0 е дадено в табл. 3.48.

Таблица 3.48

Значение на битовете от видеопаметта за кодиране на една точка в режим на цветна графика — 320×200 точки

Битове		Значение на комбинацията
ц1	ц0	
0	0	Избор на един от 16 (или 8) цвята за фона на графичното изображение — определя се от регистъра за избор на цвят
0	1	Определя цвят 1 от палитра 1 или 2
1	0	Определя цвят 2 от палитра 1 или 2
1	1	Определя цвят 3 от палитра 1 или 2

В режима на черно-бяла графика — 640×200 , има възможност да се отдели само по един бит от видеопаметта за всяка точка на графичното изображение:

Бит	7	6	5	4	3	2	1	0
	г	г	г	г	г	г	г	г

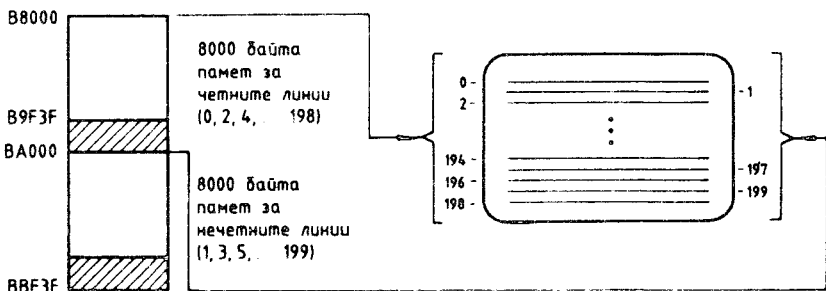
Така в един байт се записва информация за осем точки. В един бит може да се запише само 0 или 1, поради което в този режим графиката може да бъде само черно-бяла.

Последната особеност в организацията на видеопаметта на цветния графичен видеоконтролер е свързана с разположението на байтовете в графичните режими (фиг.3.54). Една линия от изображението на екрана на видеомонитора се състои от 320 или 640 точки в зависимост от разделителната способност. И в двата случая за изписване на линията е необходимо прочитане на 80 байта. Адресното пространство на видеопаметта е разделено на две области от по 8000 байта. В първата област са записани байтовете за четните линии, а във втората — за нечетните линии на изображението.

Видеопаметта на цветния графичен видеоконтролер е разположена на адрес В8000 в адресното пространство на централния процесор. Това, заедно с различieto на адресите на регистрите във входно-изходната памет, позволява едновременно включване на два видеоконтролера в конфигурацията на един персонален компютър — монохроматичен и цветен графичен.

Последната особеност на цветния графичен видеоконтролер е свързана с изобразяването на символите в различните режими. В контролера е разположен знакогенератор, който осигурява преобразуването на кода на символа в изображение в текстовите режими по начина, описан в т.3.5.1.

Подходът при изобразяването на символите в графичните режими се различава от този в текстовия режим. В областта на базовата система за вход-изход (БСВИ) на персоналния компютър е записан знакогенератор за първите 128 символа от кодната му таблица в графичен режим.



Фиг.3.54. Организация на видеопаметта в цветния графичен видеоконтролер

Когато се формират данните за изображението, драйверната програма на видеоконтролера използва информацията от знакогенератора в БСВИ, за да запише съответната буквено-цифрова информация във видеопаметта. Символът не се подава едновременно като матрица от точки, а се формира ред по ред, като съответните битове се разполагат на мястото им в рамките на изобразяваната линия от екрана на видеомонитора.

Знаковият генератор за останалите 128 символа от кодовата таблица на персоналния компютър може да се разположи в постоянната или в оперативната памет. Адресът на знакогенератора на допълнителните 128 символа е една от входните точки на БСВИ. Някои фирми производители на 16-разредни персонални компютри разполагат таблицата с графичните изображения на символите от втората половина на знакогенератора в EPROM на адрес F4000. Такъв е подходът, реализиран в Правец-16. Други фирми, включително и IBM, записват втората половина на знакогенератора в оперативната памет посредством програма на дисков носител. В различните версии на ДОС това е системната програма GRAFTABL.

Основният синхросигнал на цветния графичен видеомонитор е подбран така, че да осигурява непосредствено изобразяването на видеоинформацията върху цветен телевизор от системата NTSC. Когато се използва телевизор от друга система (PAL или SECAM), включването му към видеоконтролера става чрез RF-модулятор.



3.6. КЛАВИАТУРА

Операторът въвежда команди и данни в персоналния компютър от клавиатурата. Тя е свързана с процесора чрез адаптер, реализиран върху системната платка. От клавиатурата към компютъра се подават сигнали за натискане или отпускане на клавишите. Адаптерът предоставя постъпилата от клавиатурата информация на управляващите програми за по-нататъшна обработка. Параметрите на връзката между клавиатурата и адаптерната част са точно определени и дефинират интерфейса между тях.

Информацията се въвежда от клавиатурата по следния начин: на всеки клавиш от клавиатурата в зависимост от мястото му е присвоен т.нар. *позиционен код*. Той е седембитов. При всяко натискане или отпускане на даден клавиш клавиатурата издава неговия позиционен код заедно с признак за това, дали клавишът е натиснат или отпуснат. Този признак се кодира с един допълнителен бит. По такъв начин се формира осембитов код, който се предава последователно (бит след бит) от клавиатурата към персоналния компютър заедно със синхронизиращата го тактова поредица. Тази информация постъпва в адаптера, където се преобразува обратно в паралелен код. След това се издава заявка за прекъсване, при удовлетворяване на която управлението се предава на клавиатурния драйвер. Той обработва постъпилите позиционен код, преобразува го в съответния вътрешен код на ПК и го предоставя за по-нататъшна интерпретация от изпълняваната програма.

3.6.1. КЛАВИАТУРЕН ИНТЕРФЕЙС

Свързването на клавиатурата към персоналния компютър става по четирипроводна линия. По два от проводниците се подава захранване 0

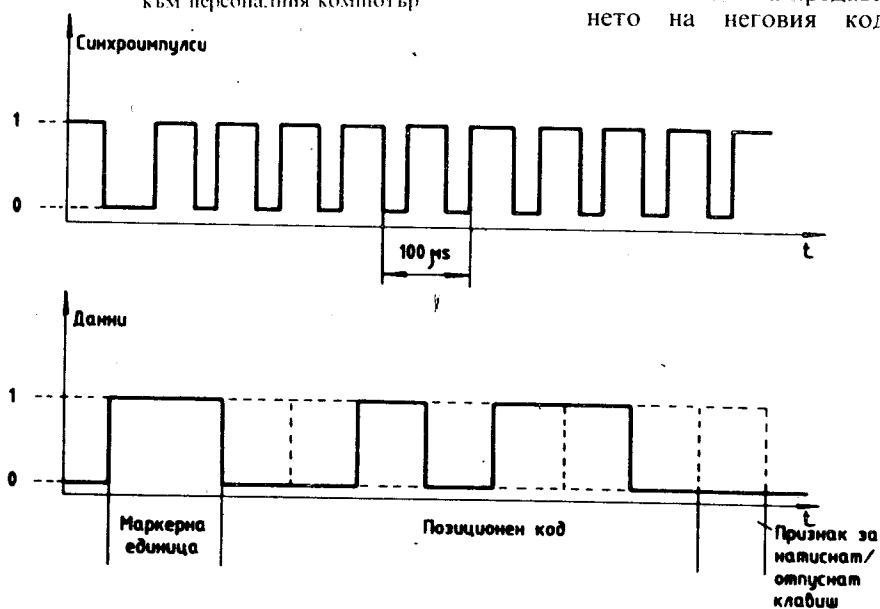
Сигнал	Извод	
		
Синхроимпулси	1	1
Данни	2	2
-	3	3
0V	4	4
5V	5	5
0V	-	6
0V	-	7
-	-	8
-	-	9

Фиг. 3.55. Типове куплунги за свързване на клавиатурата към персоналния компютър

и 5 V, а останалите два служат за издаване на данни и синхроимпулси към персоналния компютър. На фиг.3.55 са показани най-разпространените типове куплунги за свързване на клавиатурата към компютъра, както и значението на отделните пера на куплунга.

На фиг.3.56 е показана времедиаграма на сигналите за данни и синхронизация.

При отсъствие на предаване сигналът синхроимпулси е състояние логическа 1 (високо ниво), а сигналът данни — в логическа 0 (ниско ниво). При натискане на клавиш започва предаването на неговия код.



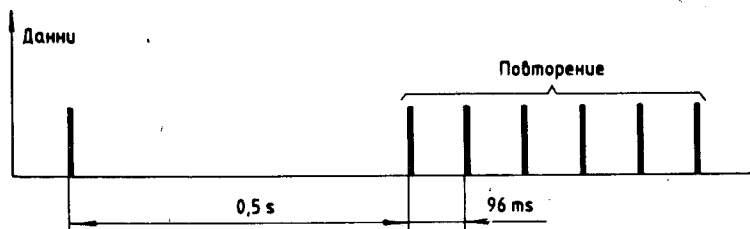
Фиг.3.56. Времедиаграма на сигналите за данни и синхронизация на клавиатурата

Преди данните винаги се предава *маркерна единица*, която е служебен символ и задава началото на информационния байт. Следва последователното предаване на позиционния код, като се започне от най-младшият бит. Последният, осмият от битовете, показва дали позиционният код се издава при натискане на клавиша (0 в допълнителния бит), или при отпускане на клавиша (1 в допълнителния бит). По такъв начин позиционният код на клавиша при отпускане е с 80 по-голям от този при натискане.

В адаптера данните се стробират по отрицателния фронт на съпроваждащите ги синхроимпулси. Първият преход на такта от 1 към 0 е подготвителен и гарантира, че предаването ще започне с маркерна единица. След завършване на предаването синхросигналят отново преминава в състояние 1, а сигнала данни — в състояние 0.

На фиг.3.56 са показани също така и основните времеви съотношения. Стробирането на данните се извършва през 100 μ s, а целият символ се предава за малко повече от 1 ms.

Клавиатурата на ПК има т.нар. функция повторение. Тази функция осигурява повторното издаване на кода на даден клавиш при задържането му в натиснато състояние за повече от 0,5 s (фиг.3.57). Кодът се повтаря през 96 ms. Това време се определя от максималната възможна скорост на въвеждане от оператора. Счита се, че ако през това време операторът не е натиснал следващия клавиш, кодът на вече натиснатия трябва да бъде повторно предаден.



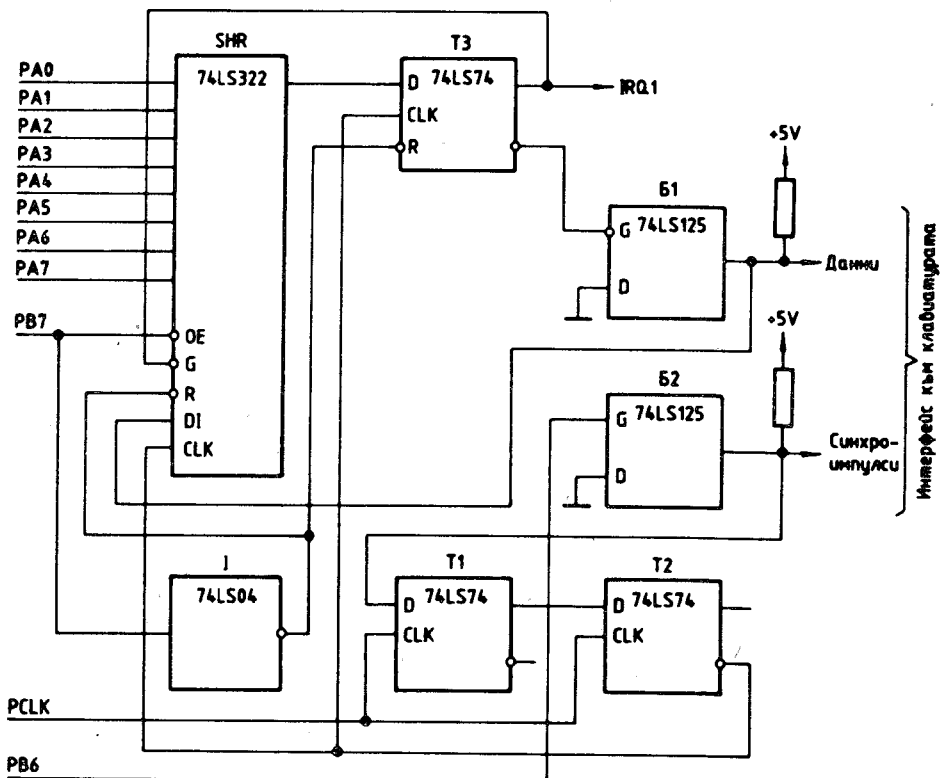
Фиг.3.57. Повторно издаване на кода

3.6.2. КЛАВИАТУРЕН АДАПТЕР

На фиг.3.58 е показана схемата на адаптера, реализираща описания интерфейс от страна на ПК. В основата ѝ лежи изместващият регистър (ИС от типа 74LS322), който има последователен вход и паралелен изход. В него се преобразува кодът, постъпващ от клавиатурата. Сигналят данни от клавиатурата се подава директно на последователния вход на изместващия регистър. Синхроимпулсите от клавиатурата постъпват на входа на единия от двата тригера T1 и T2, на тактовия вход на който се подава тактовата поредица от ПК. По този начин синхроимпулсите се синхронизират с вътрешния тактов генератор на процесора на ПК. Така преработени те се подават на входа CLK на изместващия регистър, като с всеки такт изместват съдържанието му с един разред.

Така след осем такта изместващият регистър ще се запълни и на деветия такт (по отрицателния фронт на синхроимпулса) маркерната

единица, която е била приета първа, ще се появи като пренос на измествания регистър и ще се запише в тригера Т3, към който сигналът за пренос е свързан. В момента, когато маркерната единица е установила Т3 в състояние 1, в измествания регистър се намират всички битове на позиционния код — седем информационни плюс признаът за натиснат/отпуснат клавиш.



Фиг. 3.58. Схема на адаптера на клавиатурата

Тъй като изходът на Т3 е свързан към линията за прекъсване IRQ1, то със запис на маркерната единица в него се вдига заявка за прекъсване IRQ1 и се забранява по-нататъшното изместване на информацията в измествания регистър. Едновременно с това, установяването на Т3 в 1 предизвиква чрез буфера Б1 удържане на линията данни от клавиатурата в състояние 0, като прави невъзможно издаването от нея на нови данни до момента, докато заявката за прекъсване не бъде обслужена.

При удовлетворяване на заявката за прекъсване управлението се предава на клавиатурния драйвер, който прочита позиционния код в измествания регистър, обработва и сваля заявката за прекъсване. Постъпилите позиционен код се чете чрез порт А (PA0 ÷ PA7) на паралелния входно-изходен контролер на процесора I8255A-5 (входно-изходни адреси 0060—0063), към който са свързани шините на паралелния изход на

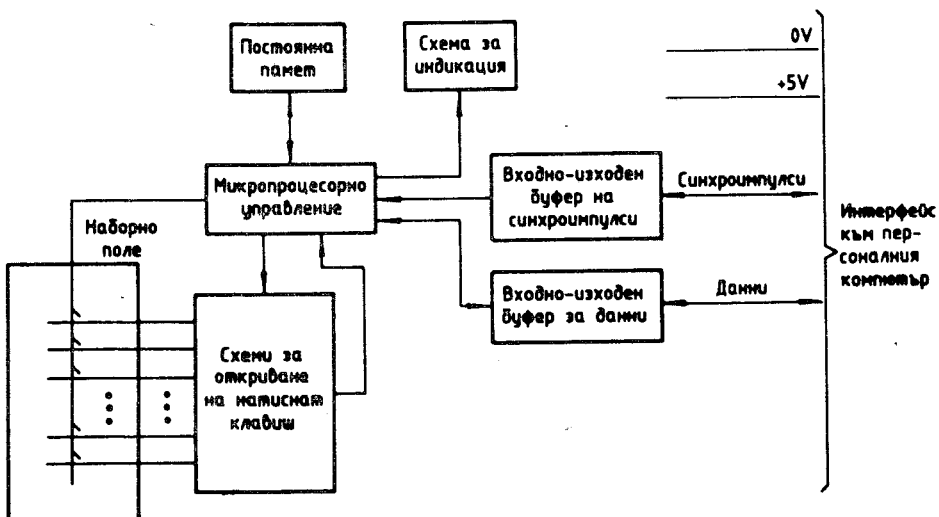
изместващия регистър. Снемането на заявката за прекъсване се извършва чрез бит 7 на порт В — PB7 на същия контролер. Еднократното установяване в 0 на този бит нулира изместващия регистър, а също така и тригера ТЗ. Това от своя страна води до деблокиране на линията данни от клавиатурата, което прави възможно приемането на следващия позиционен код.

Чрез порт В се извършва също така инициализация на клавиатурата. Запис на 0 в PB6 за повече от 20 ms предизвиква нулиране и удържане в 0 на линията за синхроимпулси от клавиатурата чрез буфера Б1. Тази ситуация се разпознава от схемите на клавиатурата и се възприема като сигнал за установяване в изходно състояние. Резултатът от нормалното завършване на процеса на инициализация е издаването от клавиатурата на код АА. Всеки друг код се възприема от процесора като признак за неизправна или липсваща клавиатура.

3.6.3. ДЕЙСТВИЕ НА КЛАВИАТУРАТА

Основната функция на клавиатурата е издаване на позиционния код на даден клавиш както при натискане, така и при отпускане, в съответствие с описания вече интерфейс. Тя се установява в начално състояние до сигнал от ПК. Клавиатурата осигурява издаването на позиционните кодове на клавишите в същата последователност, в която те са натискани от оператора. Когато се въвежда с по-голяма скорост от тази, с която централният процесор може да възприема кодовете, интерфейсът се блокира, а кодовете на междувременно натисканите клавиши се съхраняват в буфер до момента, когато издаването им стане възможно. Клавиатурата на ПК има възможност за буферизиране на 20 позиционни кода.

На фиг.3.59 е показана структурна схема на клавиатурата на ПК.



Фиг.3.59. Структурна схема на клавиатурата на ПК

Въпреки различните схемни реализации тази структура остава неизменна. Управлението на клавиатурата обикновено се извършва на базата на микрокомпютър (18048, МС6805 и др.) или от микропроцесор (например СМ601).

В първите ПК се използват предимно кондензаторни клавиши. Те са прости и надеждни, но изискват сравнително по-сложна схема за управление. В последно време все повече фирми използват контактни клавиши, с които управляващата схема се опростява извънредно много. За да се избегне възможността за поява на неправилно задействане (т.нар. „фантомни“ вериги), се прилагат различни начини. Например в Правец-16 (ЕС1839) наборното поле е реализирано във вид на диодна матрица. В ЕС1831/ЕС1832 наборното поле е реализирано във вид на матрица с размерност 96 × 1.

При включване на захранването схемата на клавиатурата осигурява своя вътрешна диагностика и подготовка за работа. Част от диагностичната програма на ПК е инициализацията на клавиатурата. Тя се стартира чрез нулиране от процесора на линията за синхроимпулси. Клавиатурата възприема този сигнал и отговаря с код АА, в случай че работи изправно. По-нататък работата на клавиатурата протича при непрекъснатото ѝ преминаване през два основни цикъла: цикъл на сканиране (обхождане) и цикъл на предаване на кода.

По време на сканирането схемите за откриване на натиснат клавиш постоянно следят състоянието на клавишите, докато микропроцесорното управление издава сигнали за постоянното им обхождане. Тъй като в паметта се съхранява информация за състоянието на всеки клавиш, по време на сканирането се уточнява кои от клавишите са натиснати и кои са отпуснати за времето от предишната проверка. При откриване на натиснат/отпуснат клавиш, от постоянната памет се извлича съответният му позиционен код, който заедно с признака за натиснат или отпуснат се записва в буфера за предаване. Обемът на този буфер е 20 кода и той е от типа FIFO.

След като сканирането е завършено, клавиатурата преминава към издаване на позиционните кодове, от буфера към ПК. Кодовете се издават заедно със съпровождащата ги синхронизираща последователност (вж. фиг.3.56). След предаването на един символ управлението на клавиатурата отново преминава към сканиране.

По време на изпълнението на тези два цикъла постоянно се следи дали някой от клавишите не е задържан за по-дълго време натиснат. Ако се открие, че даден клавиш е в състояние натиснат за повече от 0,5 s, започва периодичното издаване на позиционния му код с период 96 ms. Това продължава до натискането на следващия клавиш или до отпускането на дадения.

59	60	1	2	3	4	5	6	7	8	9	10	11	12	13	14	69	70	71	72	73	74	75	76	77	78	79	80
F1	F2	Esc	1	2	3	4	5	6	7	8	9	0	№	-	+	Back Space	Num Lock	Scroll Lock	Break								
F3	F4	Tab	Q	W	E	R	T	Y	U	I	O	P	[]	§	~	Home	B	8	9							
F5	F6	Ctrl	A	S	D	F	G	H	J	K	L	;	'	~	Home	5	6										
F7	F8	:	W	Shift	Z	X	C	V	B	N	M	,	<	>	?	Shift	PrtSc	1	2	3							
F9	F10	Alt														Caps Lock	0	Ins		Del							

Фиг.3.60. Общ вид на клавиатурата на Правец-16

На фиг.3.60 е показано разположението на клавишите върху клавиатурата на Правец-16 (ЕС1839). В горния ляв ъгъл на всеки клавиш е означен позиционният му код.

Тъй като интерфейсът на клавиатурата към ПК е строго дефиниран, независимо от различията в изпълнението на различните клавиатури те са съвместими помежду си.

3.7. ПЕЧАТАЩИ УСТРОЙСТВА

Печатащото устройство (принтерът) е съставна част от основната конфигурация на съвременните 16-разредни персонални компютри. То се свързва към базовия блок на персоналния компютър с гъвкав информационен кабел и има собствено захранване.

Използваните в ПК печатащи устройства трябва да възпроизвеждат както символна, така и графична информация. Те се включват към базовия блок на персоналния компютър най-често чрез стандартизирания паралелен интерфейс ИРПРМ (тип Centronics). Режимите на работа на печатащото устройство от компютъра се управляват чрез подаването на управляващи последователности, които започват с ASCII-символа ESC (в КОИ-8 това е символът AP2). Видът и броят на тези последователности са важни характеристики както за печатащото устройство, така и за персоналния компютър. Стандартната драйверна програма за принтера издава набор от последователности, който не може да се възпроизведе от всички печатащи устройства, работещи с интерфейс тип Centronics. Ето защо при избора на принтер трябва да се спазва условието за съвместимост с управляващите последователности, характерни за персоналните компютри от фамилията IBM PC/XT, които са описани в приложение 12.

Печатащите устройства могат да се включат към базовия блок на персоналния компютър и чрез последователен интерфейс с асинхронен старт-стопен протокол на обмен. Интерфейсът трябва да отговаря на стандарта C2 (RS232). И в този случай са валидни изискванията за съвместимост с управляващите последователности, характерни за 16-разредните персонални компютри.

Важна особеност на печатащото устройство и на неговия контролер е възможността за възпроизвеждане на текстова информация с кирилица. За целта се използва т. нар. втора, *алтернативна таблица* на знаковия генератор на печатащото устройство (приложение 13). Тя трябва да съвпада с втората половина на знаковия генератор на видеоконтролера и с втората половина на кодовата таблица на персоналния компютър. Това съответствие се постига по два начина:

— подменя се записаният в PROM знаков генератор на печатащото устройство с такъв, който в алтернативната си таблица съдържа кириллицата, подредена съгласно осембитов код, набор 2 в приложение 13;

— за печатащи устройства, които имат зареждаем (разположен в RAM) знаков генератор, се подготвя драйверна програма, разполагаща кириллицата във втората таблица съгласно осембитов код, набор 2.

В редица случаи разположението на кириллицата в знаковия генератор на видеоконтролера и в знаковия генератор на печатащото устрой-

во не съвпада. Тогава се налага прекодиране на подаваната към принтера информация съобразно с неговия знаков генератор. За тази цел в Правец-16 се използва драйверната програма P16 — M88.SYS, която се извиква от системния файл на ДОС — CONFIG.SYS.

3.7.1. ПЕЧАТАЩИ УСТРОЙСТВА ЗА 16-РАЗРЕДНИ ПЕРСОНАЛНИ КОМПЮТРИ

Съществува голямо многообразие на печатащи устройства, които могат да бъдат включени в конфигурацията на 16-разредния персонален компютър. Според принципа на създаване на изображението върху листа са обособени два типа печатащи устройства:

— *знакосинтезиращи* — символът се създава като съвкупност от множество точки, които съставят неговото изображение и разположението им се определя в знаковия генератор на принтера в т.нар. матрица на символа;

— *с фиксиран шрифтов набор* — принтерът разполага с фиксиран набор изображения на символите, които обикновено са гравирани върху печатащия механизъм на главата (тип маргаритка, сферична ротационна и др.).

Промяната на шрифта в знаковсинтезиращите печатащи устройства се извършва чрез програмна или ръчна (чрез органите за управление) подмяна на знаковия генератор, който е записан в PROM или RAM на принтера. Същата операция при печатащите устройства с фиксиран шрифтов набор е свързана с механична подмяна на елемента от печатащия механизъм, върху който са гравирани символите (маргаритката, сферичната ротационна глава и др.).

Съществено предимство на знаковсинтезиращите принтери е, че те имат възможност за изобразяване на графика с различна разделителна способност. В същото време качеството на печат при фиксиран шрифтов набор е значително по-добро, и то не отстъпва на това при пишещите машини.

Според принципа на изобразяване на символите знаковсинтезиращите печатащи устройства са няколко вида:

— *мозайкови* — символът се отпечатва посредством печатаща глава от иглен тип, която въздейства върху едноцветна или многоцветна маслилена лента;

— *термични* — символът се отпечатва посредством печатаща глава от термичен тип върху специална термочувствителна хартия;

— *струйни* — символът се изобразява посредством специална глава, която изстрелва върху хартията микроскопични капчици от печатарско мастило;

— *лазерни* — символът се изобразява чрез електростатично нанасяне на оцветяващо вещество посредством светлочувствителен барабан, върху който въздейства лазерен лъч.

Най-високо качество на изображението осигуряват лазерните принтери, които работят с най-голяма матрица на символа, а най-високо бързодействие се постига при струйните. Тези предимства обаче определят значително по-високата цена на съответните устройства. Относително най-евтини, прости за експлоатация и даващи задоволително качество на печата са мозайковите печатащи устройства. Поради това те се

включват най-често в конфигурацията на 16-разредния персонален компютър.

Мозайковите печатащи устройства се произвеждат в различни модификации: със 7-, 8- или 9-иглена печатаща глава и матрица на символа — съответно 7×7 , 8×8 , 9×9 , 9×14 , 9×18 и др. Скоростта на печат варира от 60 до 400 зн./с. Може да се използва хартия на отделни листа и безконечна перфорирана хартия. В зависимост от това механизъмът за придвижване може да бъде от фриксионен тип (като на стандартните пишещи машини) и/или от т.нар. тракторен тип (чрез водещи колела, които захващат перфорираната част на хартиената лента).

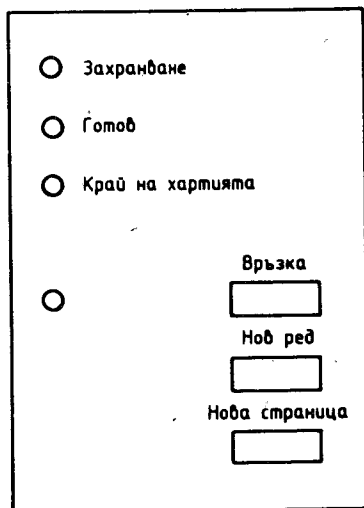
Типичен представител на мозайковите знаковсинтезиращи печатащи устройства е принтерът М88, с който стандартно се конфигурира персоналният компютър Правец-16. Печатащата глава е 8-иглена. Печатното поле е с ширина 203,3 mm (8 инча), което позволява изобразяването на 80 знака в ред с блоков нормален шрифт *пика*, 96 знака в ред с шрифт тип *елит* и 136 знака в ред с кондензиран шрифт. В трите посочени режима скоростта на печат е съответно — 80 зн./с, 96 зн./с и 68 зн./с. Разстоянието между редовете се избира 4,23 mm ($1/6$ инча) или 3,18 mm ($1/8$ инча). В първия случай един стандартен лист хартия съдържа 66 реда, а във втория — 88.

Стандартният **знаков генератор** на М88 се състои от три 7-битови таблици: КОИ-7 Н0 (ASCII), КОИ-7 Н1, КОИ-7 Н2 и от три 8-битови таблици: КОИ-8, Набор 1 и Набор 2 (съдържанието им е показано в Приложение 13). При разпечатване на текстове с *кирилица* драйвърната програма превключва М88 в 8-битовата таблица Набор 2.

Печатащото устройство М88 има **графичен режим** с две разделителни способности: 480 и 960 точки/линия. Едновременно могат да се отпечатват по 8 линии. Методът на печат е с директна кодова адресация на всяка игла. Програмируемо се задава стъпка между линиите, равна на: 0,35 mm ($1/72$ инча) и 0,118 mm ($1/216$ инча).

Хартията за печат може да бъде *форматна* (отделни листове) и *безкрайна* с перфорирани ивици. М88 разполага с двата стандартни механизма за придвижване на хартията — фриксионен и тракторен. Монтиран е датчик за край на хартията, който извежда автоматично принтера в състояние на неготовност при липса на хартия. Предвидено е лостче за регулиране на разстоянието между иглите на печатащата глава и хартията със 7 степени на регулиране. Мастилената лента е безконечна, полиестерна, черна, с ширина 13 mm и се разполага в специализирана касета с придвижващ механизъм.

Разположението на органите за управление и индикация на печатащото устройство М88 е показано на фиг.3.61. При включване на принтера чрез мре-



Фиг.3.61. Органи за управление и индикация на печатащото устройство М88

жовия ключ светва индикаторът ЗАХРАНВАНЕ. По-нататък след изпълнение на вътрешната програма за проверка печатащата глава се установява в началното състояние и светват индикаторите ГОТОВ и ВРЪЗКА. Когато хартията не е поставена добре или изобщо не е поставена, светва червеният индикатор ХАРТИЯ, чува се звук сигнал и не светят индикаторите ГОТОВ и ВРЪЗКА. След правилно зареждане на хартията и угасване на индикатора ХАРТИЯ с еднократно натискане на бутона Връзка се предизвиква светване на индикаторите ГОТОВ и ВРЪЗКА.

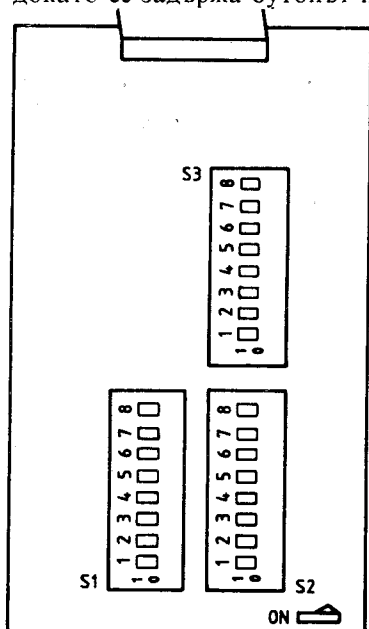
При светещ индикатор ВРЪЗКА натискането на бутоните Ред и Страница не трябва да предизвиква реакция на принтера. Еднократното натискане на бутона Връзка води до изгасване на съответния индикатор. В това състояние еднократното натискане на бутона Ред трябва да предизвика придвижване на хартията с един ред. Хартията се придвижва на нов ред през около 0,5 s, ако бутонът се държи постоянно натиснат. В същото състояние натискането на бутона Страница предизвиква придвижване на хартията до края на страницата.

Печатащото устройство М88 има възможности за проверка на работата на печатащата глава, нейната правилна позиция, качеството на печата и действието на механичния блок.

Автотест 1 предизвиква разпечатване на избраната знакова таблица от знаковия генератор на принтера. Тестът се стартира, като при изключено състояние на принтера се натисне бутонът Ред и се задържи до включване на захранващото напрежение. Разпечатването продължава, докато се задържа бутонът натиснат или докато се изключи принтерът.

Автотест 2 предизвиква разпечатване на част от знаковата таблица (96 символа от първата ѝ половина), като във всеки следващ ред знаците се изместват с един наляво. Тестът се стартира, като при изключен принтер се натисне бутонът Страница и се задържи до включване на захранващото напрежение. Разпечатването продължава около 4 min и след отпускане на бутона до свършване на хартията или до изключване на принтера.

Печатащото устройство М88 е програмируема система, изградена на базата на микропроцесора I8080A. Голяма част от функционалните му параметри и основните режими на работа се задават от микропревключватели. Те се намират под управляващия панел на устройството и са достъпни за потребителя след изваждане и обръщане на панела. Видът и разположението на функционалните микропревключватели е даден на фиг.3.62. Предназначението на ключетата на трите микропревключвателя е описано в табл. 3.49.



Фиг.3.62. Разположение на функционалните микропревключватели на печатащото устройство М88

Микропревключвателят S1 управлява формата на печата и режимите на механизма за придвижване на хартията. Микропревключвателят S2 е свързан с избора на параметрите на асинхронния последователен интерфейс към компютъра (когато е монтирана съответната интерфейсна платка). Микропревключвателят S3 управлява избора на таблица в знаковия генератор на принтера, определя шрифта и задава вида на кирилицата при работа с подходяща таблица.

Таблица 3.49

**Предназначение на функционалните микропревключватели
в печатащото устройство M88**

Функционален микропревк.почвател	Функция	Включено	Изключено	Състояние при по- купка
S1 — 1	Стъпка на редовете	1/8 инча	1/6 инча	Изключен
— 2	Дължина на страница	12 инча	11 инча	Изключен
— 3	Прескачане на перфорация 1 инч	да	не	Изключен
— 4	Изписване на знак Нула	0	ø	Включен
— 5	Датчик за хартията	валиден	невалиден	Включен
— 6	Автоматично подаване на хартията	да	не	Изключен
— 7	Не се използва	—	—	Изключен
— 8	Не се използва	—	—	Изключен
S2 — 1	Скорост на информационен об- мен по последователния асин- хронен интерфейс	вж. табл. 3.50		2400
— 2				
— 3				
— 4				
— 5	Битове Стоп	2	1	Изключен
6	Проверка по четност	да	не	Изключен
7	Бит за четност	четност	нечетност	Включен
8	Дължина на думата	8 бита	7 бита	Изключен
S3 1	Избор на таблица от знаковия генератор	вж. табл. 3.51		Набор 2
— 2				
— 3				
— 4	Не се използва	—	—	Изключен
5	Начален шрифт	вж. табл. 3.52		тика
6				
7	Вид на знаците в кирилицата	български	руски	Изключен
8	Не се използва	—	—	Изключен

Печатащото устройство M88 се предлага със стандартно включена интерфейсната платка за паралелен интерфейс тип ИРПРМ (Centronics). То има възможност да работи и с интерфейсната платка за последователен асинхронен интерфейс тип C2 (RS232). Тя поддържа стандартно установените скорости на обмен, като изборът на конкретна стойност се определя от положението на ключета 1, 2, 3 и 4 на микропревключвателя S2 — табл. 3.50. В таблицата на 1 съответства състояние Включено на ключето, а на 0 — състояние Изключено.

Изборът на една от шестте таблици в знаковия генератор на печатащото устройство се осигурява от ключета 1, 2 и 3 на микропревключвателя S3 — табл. 3.51. В таблицата на 1 съответства състояние Включено на ключето, на 0 — състояние Изключено, а X означава, че положението на ключето е без значение.

Печатащото устройство М88 може да работи с три стандартни шрифтови фонта (стандарт за графическото изображение на печатния символ) — *тика*, *италик* и *елит*. Изборът на един от шрифтовете се осигурява от ключета 5 и 6 на микропревключвателя S3 — табл. 3.52. В таблицата на 1 съответства състояние Включено на ключето, на 0 — състояние Изключено, а X означава, че положението на ключето е без значение.

Таблица 3.50
Избор на скорост на обмен по асинхронния последователен интерфейс с помощта на ключета 1, 2, 3 и 4 на микропревключвателя S2

Скорост на обмена, бодове	Ключета на микропревключвателя S2			
	1	2	3	4
19 200	1	1	1	1
9 600	0	1	1	1
4 800	1	0	1	1
2 400	0	0	1	1
1 200	1	1	0	1
600	0	1	0	1
300	1	0	0	1
200	0	0	0	1
150	1	1	1	0
110	0	1	1	0
100	1	0	1	0
75	0	0	1	0
50	1	1	0	0

Таблица 3.51
Избор на таблица от знаковия генератор на М88 с помощта на ключета 1, 2 и 3 на микропревключвателя S3

Ключе от S3	Таблица от знаковия генератор на М88					Набор1	Набор2
	КОИ7-Н0	КОИ7-Н1	КОИ7-Н2	КОИ8			
S3 — 1	0	1	X	0	1	X	
— 2	0	0	1	0	0	1	
— 3	0	0	0	1	1	1	

Таблица 3.52
Избор на стандартен шрифтов фонт с помощта на ключета 5 и 6 на микропревключвателя S3

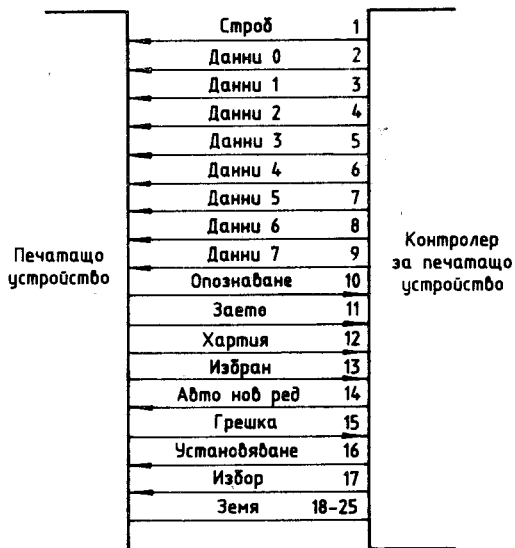
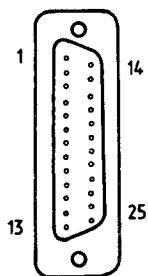
Ключе от микропревключвателя S3	Начален стандартен шрифтов фонт		
	тика	италик	елит
S3 — 5	X	0	1
— 6	0	1	1

Предвидена е възможност пълната информация за състоянието на ключетата на функционалните микропревключватели да се извежда на хартия по време на работа на печатащото устройство М88. За целта е необходимо чрез бутон Връзка да се прекрати обменът с ПК (индикаторът ВРЪЗКА изгасва). След това отново се натиска бутонът Връзка и като се задържа в това положение, се натиска и бутонът Ред. Бутоните се отпускат в същата последователност — най-напред Връзка, а след това Ред. Възприемането на командната последователност се потвърждава от звук сигнал. Следва изпълнението на програма за вътрешна диагнос-

тика (за по-малко от 1 s), която завършва с издаването на втори звуков сигнал и разпечатването на информация за състоянието на ключетата на функционалните микропревключватели. След завършване на печата, принтерът остава изключен от режим ВРЪЗКА и за възстановяване на обмена е необходимо да се натисне еднократно съответният бутон.

3.7.2. ИНТЕРФЕЙСИ И КОНТРОЛЕРИ ЗА ПЕЧАТАЩИ УСТРОЙСТВА

Основният интерфейс за свързване на печатащите устройства към 16-разредния персонален компютър е паралелният интерфейс тип ИРПРМ (Centronics). Данните се предават в паралелен код побайтово със скорост около 1000 символа/s. Структурата на интерфейса е показана на фиг.3.63. Сигналът по линията СТРОБ е нормално в логическа 1. Установяването му в 0 за не по-малко от 0,5 μ s задава момента, в който байтът, намиращ се на линиите ДАННИ0—ДАННИ7, може да бъде получен от печатащото устройство за разпечатване. Подаването на данните към принтера е възможно, когато нивото на сигнала ИЗБОР е 0. В повечето печатащи устройства е предвидено вътрешно фиксиране на сигнала, което се изпълнява по желание на потребителя. Сигналят УСТАНОВЯВАНЕ е нормално в 1. Когато той се установи в 0 за не по-малко от 50 μ s (за M88 — не по-малко от 0,5 ms), печатащото устройство незабавно прекратява текущата операция, буферът му се нулира и се извършва начално установяване на вътрешните му схеми и вериги. Сигналят АВТО НОВ РЕД с установяването си в 0 предизвиква автоматично преминаване на нов ред при завършване разпечатката на текущия ред. В повечето печатащи устройства е предвидено вътрешно фиксиране на сигнала към 0 посредством мостчета или микропревключватели.



Фиг.3.63. Интерфейс между печатащото устройство и контролера в 16-разредния ПК

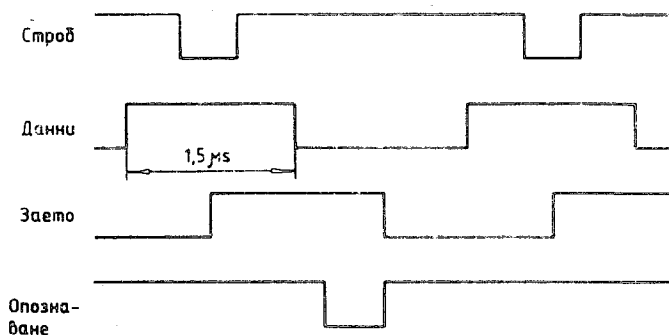
Сигналът ОПОЗНАВАНЕ е нормално в състояние 1. Установяването му в 0 за около $5 \mu s$ означава, че печатащото устройство е получило данните и е готово за възприемане на нови данни. Сигналът ЗАЕТО със своето ниво 1 указва, че принтерът не може да приема данни. Такова ниво на сигнала се получава в следните случаи:

- по време на приемане на данни;
- по време на разпечатването;
- когато печатащото устройство е в автономен режим (за М88 индикаторът ВРЪЗКА е изгаснал);
- когато печатащото устройство е в състояние на грешка.

Нивото 1 на сигнала ХАРТИЯ показва, че в печатащото устройство не е заредена или е заредена неправилно хартията за печат. Сигналът ИЗБРАН указва, че печатащото устройство е избрано, когато се намира в състояние 1. В някои печатащи устройства, например в М88, този сигнал е постоянно фиксиран към логическа 1. Сигналът ГРЕШКА преминава в 0 и указва за възникнала неизправност в печатащото устройство в следните случаи:

- когато е свършила хартията;
- когато печатащото устройство е в автономен режим (за М88 индикаторът ВРЪЗКА е изгаснал);
- когато схемите и веригите на печатащото устройство са неизправни.

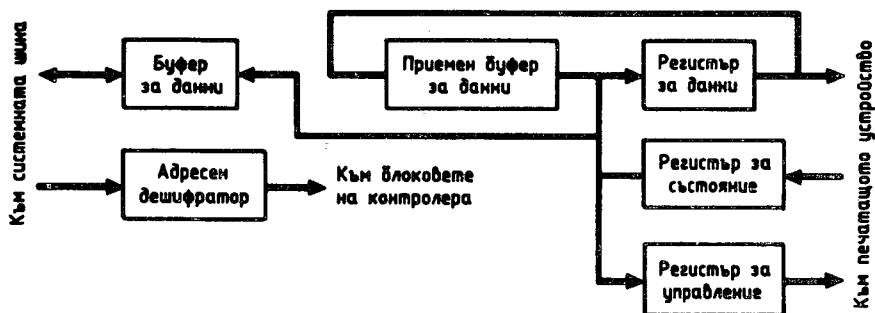
Типична последователност за предаване на данни по интерфейса тип ИРПРМ е показана на фиг.3.64. Времето, за което са валидни данните, е минимум $1,5 \mu s$, като сигналът СТРОБ се появява в средата на този интервал. Сигналът ОПОЗНАВАНЕ предизвиква появата на данните върху линиите ДАНИИ0—ДАНИИ7 и задължително е следван от установяване в 0 на сигнала ЗАЕТО. Установяването в нула на сигнала СТРОБ предизвиква активирането на сигнала ЗАЕТО.



Фиг.3.64. Времедиаграма за предаване на данни по интерфейса ИРПРМ

Обобщена блокова схема на контролера за паралелен интерфейс тип ИРПРМ (Centronics) е дадена на фиг.3.65. Той се състои от два регистъра за данни, един регистър за управление на интерфейса (съответно — печатащото устройство) и два регистъра за състояние — на интерфейса и на печатащото устройство. За управлението на тези регистри е изпълнена

схема за дешифриция, която отговаря на три адреса (порта) от входно-изходната памет на персоналния компютър: два за запис и четене и един само за четене.



Фиг.3.65. Обобщена блокова схема на контролера на печатащото устройство

На адреса във входно-изходната памет 378 (278 или 3BC) е разположен регистърът за данни на контролера за паралелен интерфейс. Той е достъпен за запис и за четене от системната шина на персоналния компютър. Предназначението на отделните битове на регистъра и тяхното съответствие на изводите на куплунга за паралелен интерфейс е следното:

Бит на регистъра	7	6	5	4	3	2	1	0
Съдържание	D7	D6	D5	D4	D3	D2	D1	D0
Извод №	9	8	7	6	5	4	3	2

При операция запис от системната шина в този регистър се записват данните, които трябва да се подадат към печатащото устройство за отпечатване или за управление на неговите режими. След записа им в регистъра, данните се установяват на съответните изводи на интерфейския куплунг.

При операция четене от системната шина от регистъра могат да бъдат прочетени даните, които са записани на същия адрес. Ако периферното устройство, свързано към интерфейския куплунг, също има възможност да предава данни, от този регистър може да се прочете резултатът от операцията жично ИЛИ между данните на контролерния регистър и данните на периферното устройство.

На адреса във входно-изходната памет 37A (27A или 3BE) при операция запис от системната шина на персоналния компютър се разполага регистърът за управление на интерфейса (печатащото устройство). Съответствието между отделните битове на регистъра и изводите на интерфейския куплунг, и тяхното предназначение е следното:

Бит на регистъра	7	6	5	4	3	2	1	0
Предназначение (съответствие на интерфейсен сигнал)	—	—	—	IRQ разрешение	ИЗБОР	УСТА-НОВА-ВАНЕ	АВТО-НОВ РЕД	СТРОБ
Извод №					17	16	14	1

На същия адрес във входно-изходната памет при операция четене от системната шина се получава съдържанието на регистъра за състоянието на паралелния интерфейс. Ако периферното устройство не влияе върху състоянието на съответните интерфейсни линии, съдържанието на този регистър съответства точно на съдържанието на регистъра за управление на интерфейса. В противен случай от регистъра за състоянието може да се прочете резултатът от операцията жично ИЛИ между съдържанието на регистъра за управление и получените от периферното устройство сигнали.

Съответствието между отделните битове на регистъра и състоянието на изводите на интерфейския куплунг, тяхното предназначение и стойностите им след начално установяване на контролера е следното:

Бит на регистъра	7	6	5	4	3	2	1	0
Предназначение (съответствие на интерфейсния сигнал)	—	—	—	IRQ разрешение	ИЗБОР	УСТА- НОВЯ- ВАНЕ	АВТО- НОВ- РЕД	СТРОБ
Извод №					*17	16	*14	*1
Съдържание след начално установяване				0	1	0	1	1

Звездчките пред номерата на изводи 17, 14 и 1 показват, че в регистъра за състоянието на интерфейса се записват инверсните стойности на състоянието на съответните изводи.

Регистърът за състояние на принтера е достъпен за операция четене от системната шина на адрес във входно-изходната памет 379 (279 или 3BD). В този регистър се записва моментното състояние на интерфейсните сигнали, получавани от печатащото устройство. Съответствието между отделните битове на регистъра и изводите на интерфейския куплунг и тяхното предназначение е следното:

Бит на регистъра	7	6	5	4	3	2	1	0
Предназначение (съответствие на интерфейсния сигнал)	ЗАЕТО	ОПОЗНА- ВАНЕ	ХАРТИЯ	ИЗБРАН	ГРЕШКА	—	—	—
Извод №	11	10	12	13	15			

Контролерът за паралелен интерфейс осигурява издаването на всички изходни сигнали за интерфейса тип ИРПРМ чрез изпълнението на драйверна програма. Съдържанието на регистрите за състояние отразява реалното състояние на съответните интерфейсни сигнали на входа на контролера, а не запомнени в предишен момент техни стойности.

В персоналния компютър могат да бъдат инсталирани едновременно три контролера за паралелен интерфейс с адреси във входно-изходната памет съответно:

- 378, 379 и 37A;
- 278, 279 и 27A;
- 3BC, 3BD и 3BE.

Тези контролери получават от БСВИ и ДОС имена на физическо устройство съответно PRN1, PRN2 и PRN3.

Подаването на информация към печатащото устройство чрез стандартния интерфейс от типа С2 (RS232) се извършва чрез асинхронния комуникационен адаптер на персоналния компютър. Подробно описание на интерфейса и адаптера е дадено в т.3.8. Имената на физически устройства, присвоявани от ДОС на комуникационните адаптери, са COM1 и COM2.

Някои програми работят с логическото име на печатащото устройство PRN. За да се осигури правилна работа на операционната система и на приложните програми, е необходимо да се направи съответствие между физическите и логически имена, присвоявани на печатащите устройства в конфигурацията на персоналния компютър.

3.8. КОМУНИКАЦИОННИ АДАПТЕРИ ЗА 16-БИТОВИ ПЕРСОНАЛНИ КОМПЮТРИ

С развитието на персоналните компютри нарасна необходимостта от все по-съвършени комуникационни връзки между тях. Изчислителните ресурси вече се предоставят на работното място на крайния потребител и това доведе до тяхното обособяване и индивидуализиране. Отделните потребители започнаха да организират микро-ЕИЦ на базата на ПК, като започнаха да натрупват свои библиотеки от програми, да строят собствени бази от данни. Това доведе до редица неудобства поради нарастващата необходимост от квалифицирана поддръжка на тези микро-ЕИЦ. На практика се премахваше главното предимство на ПК — възможността за непосредствена връзка на потребителя с него. От друга страна, появиха се редица скъпи периферни устройства за ПК — главно графична периферия. Естествен бе стремежът за колективно ползване на тези устройства чрез осигуряване на комуникационни връзки между отделните ПК.

Едновременно с това изнасянето на ПК на работните места на потребителите силно увеличи разстоянията между тях. Стана необходимо да се създават отдалечени връзки между ПК.

Наличието на бази от данни върху големи и мини-ЕИМ наложи разработването на средства за свързване на персонални компютри към тях и осигуряване на достъпа им до тези данни. ПК се оказа изключително удобен като интелигентен терминал към големи и мини-ЕИМ. При това емулирането на широко разпространени типове терминали става върху универсални ПК на базата на специализирани комуникационни адаптери и съответното програмно осигуряване. Така не се губи възможността за използването на ПК за други цели в случаите, когато това е необходимо.

Подобно на развитието на комуникациите между другите видове изчислителни средства и при ПК се разграничиха рязко два типа комуникации — местни и отдалечени. Между местното и отдалечено свързване съществува качествена разлика, която се определя не само от разликата във физическите разстояния.

Местното свързване на персоналните компютри се осъществява чрез т.нар. *локални мрежи*. При него разстоянията между отделните компют-

ри са от няколко метра до няколко километра. Обикновено в локална мрежа се свързват ПК в една сграда или в няколко съседни сгради. Връзката става чрез специално предвидени за целта кабели (симетрични линии, коаксиални кабели и др.), като малките разстояния предполагат наличието на ширококестотна лента или обмен на данни с високи скорости. Това позволява директно предаване в двоичен вид без допълнителна модулация. Високата скорост на обмен позволява прилагането на протоколи за връзка, използващи допълнителна управляваща информация за опростяване на управлението на съответните комуникационни адаптери.

Отдалеченото свързване на персонални компютри е нужно за разстояние от десетки и повече километри. За съобщителна среда се използват съществуващите съобщителни телефонни или телексни линии или пък специализираните глобални обществени мрежи за предаване на данни. Връзките се осъществяват обикновено чрез модеми. Спецификата на физическата среда ограничава скоростта на обмен от порядъка на няколко десетки Кбита/s. Това налага и избора на по-икономични протоколи за връзка. Обикновено отдалеченото свързване на персонални компютри, а също така и връзката им към големи и мини-ЕИМ става чрез широко разпространени традиционни протоколи за връзка, които могат да бъдат обслужвани и от устройството, към което се свързва ПК. Прилагането на специфични протоколи за връзка е възможно само в еднородни мрежи от ПК.

3.8.1. КОМУНИКАЦИОННИ АДАПТЕРИ ЗА МЕСТНО СВЪРЗВАНЕ

Комуникационните адаптери за свързване на персоналните компютри към локални мрежи обикновено са изпълнени във вид на печатни платки, които се монтират директно в касетата на ПК. От една страна, платката е свързана към системната шина на ПК, а от друга — директно към съобщителната среда. Върху адаптера са разположени схемите за обмен на данни с паметта на ПК, схемите за управление на достъпа до средата и приемопредавателите. Управлението е изпълнено с микропроцесор, а отделните функции на адаптера се изпълняват от програмируеми интегрални схеми с голяма степен на интеграция. Обменът на данни с паметта най-често се извършва чрез директен достъп, като върху адаптера е предвидена буферна памет с капацитет, не по-малък от необходимия за буфериране на един пакет.

В последно време има тенденция за поемане на все повече функции от комуникационния адаптер чрез реализацията им на апаратна и програмно-апаратна (фърмуерна) основа. По такъв начин значително нараства реалната производителност на локалната връзка между отделните персонални компютри.

3.8.2. КОМУНИКАЦИОНЕН АДАПТЕР ЗА СВЪРЗВАНЕ НА ПК КЪМ ЛОКАЛНА МРЕЖА МИКРОРИНГ

Локалната мрежа МИКРОРИНГ е предназначена за връзка както между персонални компютри, така и с други устройства — принтери, плотери, терминали.

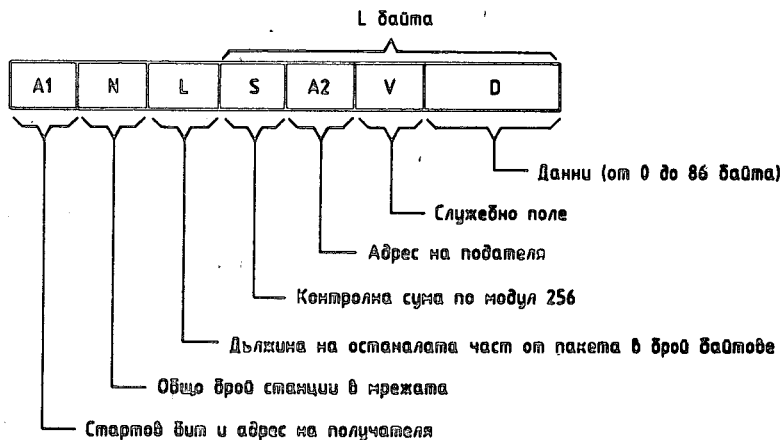
Локалната мрежа се състои от комуникационни възли, свързани помежду си в кръг. Броят на възлите може да бъде от 2 до 125, а максималното разстояние между два възела е до 6 km. Скоростта на обмен между отделните възли в мрежата може да бъде 125 или 250 Кбита/s. Обменът на информация в МИКРОРИНГ се осъществява на пакети, а достъпът до съобщителната среда се извършва по метода на вмъкване на регистри. При него всеки възел приема постъпващите на входа му пакети и ги буферира. Ако даден пакет е предназначен за определен възел, той го обработва. В противен случай пакетът се препредава към следващия възел. При предаване се изчаква преминаването на текущия пакет и след определено време започва предаване на собствения пакет. Ако по време на предаване на входа на възела постъпят други пакети, те се записват в буферна памет. След завършване на предаването те се обработват и тези, които са предназначени за други възли, се предават към следващия възел.

В зависимост от предназначението си пакетите за информационен обмен могат да бъдат три типа: информационни, командни и служебни. Чрез служебните се осигурява началното включване на даден възел в мрежата, потвърждават се приетите пакети, проверява се целостта на кръга и др.

На фиг.3.66 е показан форматът на пакета в МИКРОРИНГ. В него може да се различи заглавна част (първите шест еднобайтови полета и информационна част (данни, с дължина на полето от 0 до 86 байта).

Първото поле съдържа адреса на получателя на пакета. Тъй като първият бит се използва като стартов (приемането на пакет започва при преминаване на линията от състояние 1 в 0), то за кодиране на адреса остават 128 възможни значения. Три от тях са отделени за служебни цели, следователно в мрежата може да има 125 възела с различни адреси. Програмното осигуряване на мрежата работи с логически имена, които се задават от потребителя, докато адресът на всеки възел се формира автоматично при включването му в мрежата.

Второто поле съдържа число, показващо общия брой възли в мрежата в даден момент.



Фиг.3.66. Формат на пакета в локална мрежа МИКРОРИНГ

В третия байт е указана дължината в байтове на цялата останала част от пакета.

Следващият байт съдържа контролен код. Това е контролна сума по модул 255. Следва полето с адреса на подателя на пакета.

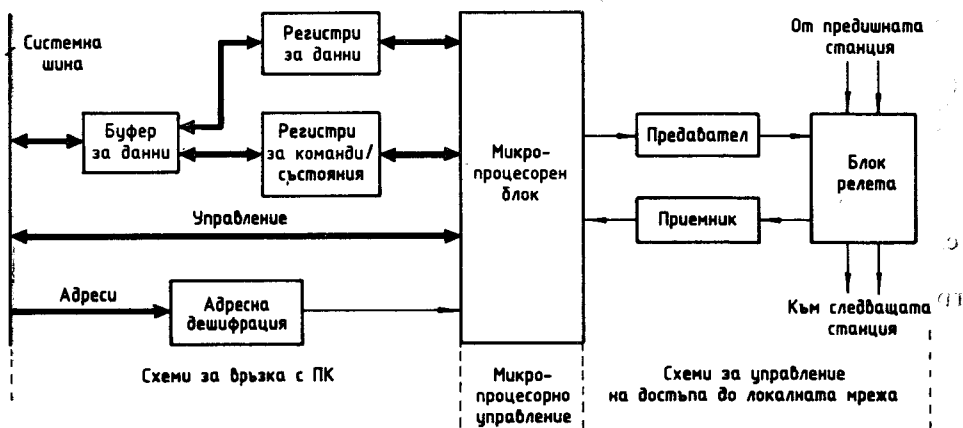
Шестият байт е отделен за служебна информация, след която започва полето за данни. То може да има дължина от 0 до 86 байта. Пакетите са разделени помежду си с интервали, не по-малки от един байт.

Всеки пакет при приемане се проверява за грешки чрез проверка на контролната му сума и проверка за коректна дължина. При откриване на грешка пакетите се отхвърлят и приемането им не се потвърждава. Ако изпращащият възел не получи потвърждение за приет пакет след неколкратни опити за предаване, той прави проверка дали мрежата е цяла чрез изпращане на служебен пакет, адресиран до себе си. Ако в течение на определено време този пакет (или пакет, генериран от друг възел) не се получи, даденият възел приема, че кръгът е разкъсан точно преди него. В този случай той започва да генерира отново служебни пакети, съдържащи информация за мястото на прекъсването. Всеки следващ възел предава тези пакети, като увеличава с 1 брояча, намиращ се в тях. Така всеки възел разполага с информация за мястото на повредата.

Персоналните компютри се свързват към локалната мрежа МИКРОРИНГ чрез комуникационен адаптер. Съществуват две конструктивни реализации на комуникационния адаптер. В едната от тях той е изработен във вид на платка, която се монтира непосредствено в ПК. През него обменът на управляваща информация и данни между ПК и комуникационния адаптер става с висока скорост по системната шина на компютъра.

Във втория случай адаптерът е изпълнен във вид на самостоятелен конструктивен блок със собствено захранване. Връзката на ПК с него е чрез интерфейс С2 и е асинхронна.

Независимо от конструктивните си различия двата адаптера реализират едни и същи функции и с изключение на частта за връзка с ПК имат еднаква структура.



Фиг.3.67. Структурна схема на комуникационния адаптер за свързване на ПК към локалната мрежа МИКРОРИНГ

На фиг.3.67 е показана структурната схема на комуникационния адаптер за свързване на ПК към локалната мрежа МИКРОРИНГ. На нея могат функционално да се обособят три групи логически схеми: схеми за връзка с ПК, схеми на микропроцесорното управление и схеми за управление на достъпа до локалната мрежа.

Комуникационният адаптер се свързва с ПК чрез буфери за данни и се управлява от два програмно достъпни регистъра за управление — регистър за команди и регистър за състояние. И двата регистъра се намират на адрес 2EF. Буферите за данни се намират на адрес 2EE. На фиг.3.68 са показани програмнодостъпните регистри на комуникационния адаптер.

Предназначение на битовете в регистъра за команди:

- Бит 0 — при стойност 1 в този бит се разрешава работата на предавателя.
- Бит 1 — не се използва.
- Бит 2 — при стойност 1 в този бит се разрешава работата на приемника.
- Бит 3 — задава състояние „break“.
- Битове 4,5,6,7 — не се използват.

Предназначение на битовете в регистъра за състояние:

- Бит 0 — показва, че буферният регистър на предавателя е празен и може да приеме поредния байт данни. Този бит винаги се установява след записа на поредния байт данни от буферния регистър на предавателя в неговия изместващ регистър. Със зареждането на следващия символ за предаване бит 0 автоматично се нулира.
- Бит 1 — явява се индикатор за приети данни. Установява се в 1, когато в приемния буферен регистър е приет поредният байт данни. Нулирането на този бит става при четене на приемния буферен регистър.
- Бит 2 — показва, че съдържанието на изместващия регистър на предавателя е предадено. Той се нулира при записа на поредния символ от буферния регистър на предавателя в неговия изместващ регистър.
- Битове 3,4,5 — не се използват. При четене състоянието им винаги е 0.
- Бит 6 — служи като индикатор за състояние „break“ по линията при приемане.
- Бит 7 — не се използва. При четене състоянието му винаги е 1.

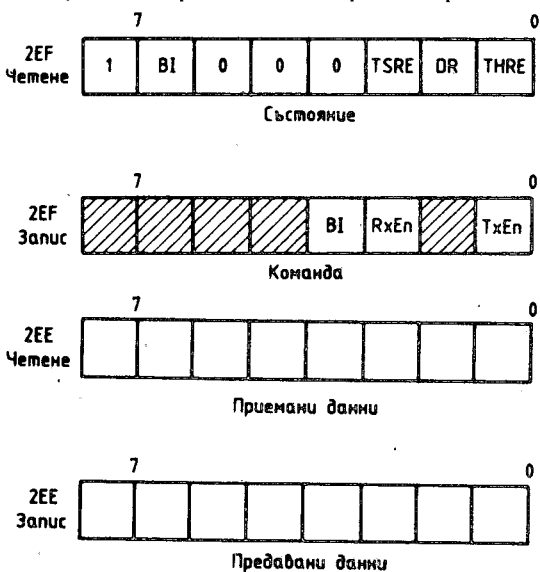
Обменът на данни между ПК и комуникационния адаптер може да се извършва в един от следните два режима: чрез *полиране* и чрез *програмно прекъсване*.

При обмен чрез полиране управляващата драйверна програма периодично проверява съдържанието на регистъра за състояние. Когато бит 0 е установен в 1, записът на поредния байт данни от ПК в регистъра за данни е възможен. Записът на следващия байт е забранен, докато поредният байт не бъде прехвърлен към микропроцесорната схема на комуникационния адаптер. През цялото това време бит 0 е установен в 0. Установяването му в 1 означава, че адаптерът може да приеме следващия байт.

Когато данните постъпват от адаптера към ПК, драйверните програми следят състоянието на бит 1. Единицата в този бит означава, че поредният байт данни може да бъде прочетен от регистъра за данни.

При обмен чрез програмно прекъсване появата на байт за предаване към ПК или прочитането на регистъра за приемане на данни от ПК води до възникване на заявка за прекъсване. В този случай програмата за обработка на прекъсването проверява каква е причината за неговото възникване чрез прочитане на регистъра за състояние. По-нататък обменът се извършва както при полирането.

Схемите за управление чрез вътрешното програмно осигуряване осъществяват както обмен с ПК, така и достъп до ресурсите на локалната мрежа. Управлението е реализирано с микропроцесор от типа Z80.



Фиг. 3.68. Програмнодостъпни регистри на комуникационния адаптер

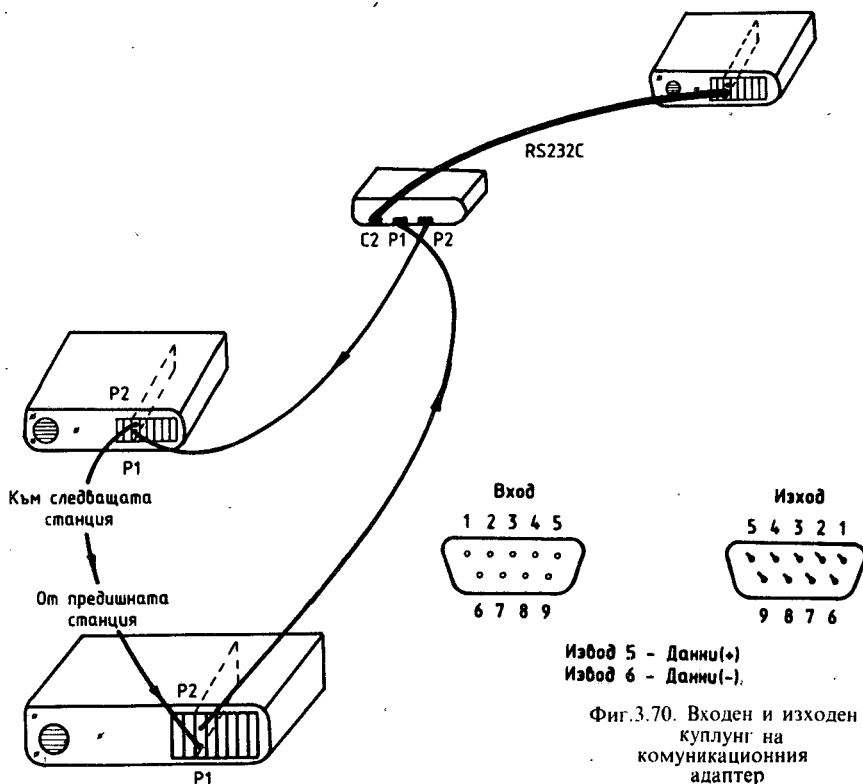
Включването към съседните два възела става чрез двойка превключващи релета. Когато комуникационният адаптер работи, те осигуряват свързване на двойката проводници от предишната станция към входовете на приемниците, а изходите на предавателите — към двойката проводници на следващия възел. В случай на пропадане на захранващото напрежение нормално затворените контакти на релетата осигуряват непосредствено свързване на входните линии към изходните, като по този начин кръгът остава цял, въпреки че даденият възел е изключен.

На фиг. 3.69 е показан начинът на свързване на отделните комуникационни адаптери в затворен кръг. Използва се радиочестотен кабел с вълново съпротивление 240 Ω.

На фиг. 3.70 са показани входният и изходният куплунг на комуникационния адаптер. И в двата случая са използвани куплунги от тип Сапон, 9 цифрови. Входният куплунг е тип „женски“, а изходният — тип „мъжки“. Това прави невъзможно погрешното свързване на съседните станции при конфигурирането на локалната мрежа. Освен това произволен кабел може да бъде използван за осигуряване на обратна връзка при проверка на комуникационния адаптер. Допълнително предимство е възможността за удобно преконфигуриране на мрежата, без да се променя кабелната инсталация. Това става, като на мястото на премахнат адаптер двата кабела се свързват накъсо. Има възможност за удължаване на връзката между отделни възли чрез „наставяне“ на отделни сегменти помежду им.

Предвидена е възможност за работа със Z80A, което позволява два пъти по-висока скорост на обмен между отделните възли на мрежата. Вътрешното програмно осигуряване заема памет с обем 4 Кбайта, а обемът на оперативната памет е 2 Кбайта.

Схемите за осигуряване на достъп до локалната мрежа преобразуват данните от последователен в паралелен код при приемане и от паралелен в последователен код при предаване. С помощта на тези схеми под управлението на вътрешното програмно осигуряване се реализира методът за достъп до съобщителната среда.



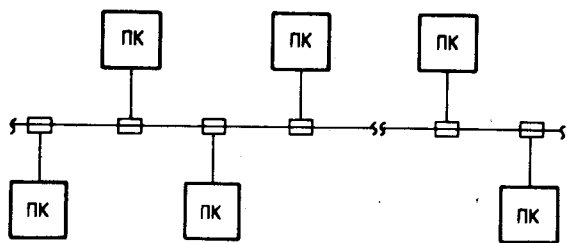
Фиг. 3.69. Начин на свързване на отделните комуникационни адаптери в затворен кръг

3.8.3. КОМУНИКАЦИОННИ АДАПТЕРИ ЗА СВЪРЗВАНЕ НА ПК КЪМ ЛОКАЛНА МРЕЖА МИКРОЛИМ

Локалната мрежа МИКРОЛИМ е с шинна топология (фиг. 3.71). Тя е разработена в пълно съответствие със стандартите на ISO 8802.2 и 8802.3, т.е. тя е от тип Ethernet. Комуникационният адаптер е платка, която се монтира в един от куплунгите на 16-битовите персонални компютри Правец-16 (ЕС1839), ЕС1838, IBM PC/XT/AT и съвместимите с тях. Едно от основните му предимства е наличието на вграден приемопредавател, който позволява директното включване на тънък коаксиален кабел към него с помощта на съединители от тип BNC.

Физическата среда за предаване на данните е коаксиален кабел с импеданс 50 Ω. Съществуват два типа кабели — тънък и дебел. За-

дължително изискване при изграждането на мрежата е кабелът във всеки сегмент да бъде съгласуван от двата края с 50-омови съгласуващи уст-



Фиг.3.71. Локална мрежа МИКРОЛИМ

ройства. Тънкия кабел се свързва към контролера с помощта на стандартни 50-омови BNC-съединители. Дебелият кабел не може да се свърже директно към платката. За свързването му се използват външни приемопредаватели. Кабелът се свързва към приемопредавателя с помощта на N-съединители. Връзката между дистанционния приемопреда-

давател и платката на контролера се осъществява чрез специален интерфейсен кабел, завършващ от двете страни с 15-щифтови съединители от тип Cannon.

Максималната дължина на един сегмент е 300 или 500 m. Тя зависи от вида на кабела. С тънкия кабел максималната дължина на един сегмент е 300 m, но той позволява директното му включване чрез T-съединител към BNC-съединителя на платката. Дебелият кабел позволява максималната дължина на един сегмент да е до 500 m. Кабелът, с който дистанционният приемопредавател се свързва към платката на контролера, може да бъде с максимална дължина 15 m, като превключването от вграден към дистанционен приемопредавател става със 7 мостчета. В една локална мрежа могат да се включат по няколко сегмента с помощта на повторители. Информацията, предавана между производни станции, може да преминава най-много през два повторителя, т.е. повторителите трябва да се включат само към един сегмент на мрежата. Това позволява максималната дължина в мрежата да бъде до три пъти дължината на един сегмент — 900 или 1500 m в зависимост от използвания кабел. Съществува възможност за съвместно използване на тънък и дебел кабел в една и съща мрежа.

Максималният брой станции в един сегмент е 100. Този брой се определя от минималното разстояние от станция до станция, което е 3 m. В една локална мрежа максималният брой станции е 1024.

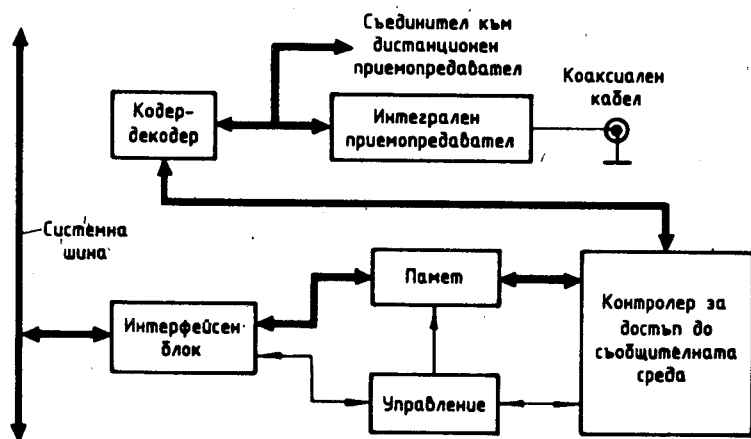
Блоквата схема на платката е показана на фиг.3.72. В нея могат да се разграничат 3 главни възела: интерфейсен възел между компютъра и контролера за достъп до съобщителната среда, мрежов контролер, вграден приемопредавател.

Интерфейсният възел има следните основни блокове:

Интерфейс към компютъра. Съдържа буфер за запис и четене, дешифратор на базовия адрес, дешифратор на началния адрес на постоянната памет, ключета за избор на канал за директен достъп до паметта и ниво на прекъсване.

За потребителя са интересни двата дешифратора и изборът на канал за ДДП и прекъсване. Те позволяват в зависимост от конфигурацията

на компютъра потребителят сам да избере базовия адрес на 16-те програмно достъпни регистъра на платката, началния адрес на постоянната памет за вграден драйвер на контролера и каналите за ДДП и прекъсване. Постоянната памет, която може да се сложи в предвидения за целта



Фиг.3.72. Блокова схема на платката на МИКРОЛИМ

цокъл, съдържа програма (драйвер), която при включване на захранването на компютъра позволява на потребителя да се свърже към обслужващото устройство, без да използва собствените дискетни устройства на компютъра. Програмата емулира наличието на един твърд диск (уинчестър) в компютъра. Тази памет може чрез съответно ключе да бъде разрешавана или забранявана в зависимост от желанието на потребителя.

С микропревключватели могат да се избират също базовият входно-изходен адрес, каналът за директен достъп до паметта, нивото на прекъсване, както и началният адрес на постоянната памет. Това дава възможност на потребителя да изгражда различни конфигурации в зависимост от желани състав на периферните устройства, включени към компютъра му. Най-често базовият адрес е 300, за ДДП каналът е 1, за ниво на прекъсване каналът е 3 и началният адрес на постоянната памет е EC000.

Блок за управление. Включва команден регистър и регистър за състоянието, броячи за предаване и приемане, управляваща логика.

В командния регистър и регистъра за състояние всеки бит има определен конкретен смисъл. Командният регистър има съответно битове за: начално установяване, разрешаване на предаването, разрешаване на приемането, маска на прекъсването, бит за начало на обмена с ДДП. Регистърът за състояние на контролера във всеки момент показва в какъв режим се намира той: предаване, приемане, завършило ли е предаването, има ли приет пакет и т.н. Управляващата логика позволява използването на една от двете интегрални схеми върху платката — контролера за достъп до съобщителната среда на фирмата SEEQ DQ 8001 или DQ 8003.

Буферна памет за пакети. Интензивността на обмена на пакети и избраният режим на работа със заявки и отговори позволяват тази памет

да бъде с минимален обем — 2 Кбайта. Приеманите пакети се записват в началните ѝ адреси, а тези които се предават — в горната част на паметта. От гледна точка на програмиста паметта е само един регистър, а адресът на клетката, до която има достъп, се задава чрез брояча за предаване.

В буферната памет на компютъра може да се записва или чете по два начина:

— първият е чрез последователен запис на данните в регистъра за достъп до паметта, като първоначалният адрес се записва в брояча за предаване и при всеки запис или четене на данни в паметта неговата стойност се увеличава автоматично с 1;

— вторият начин на запис или четене в паметта е чрез съответния канал за ДДП на компютъра, като преди началото на обмена е необходимо началният адрес да се запише отново в съответните регистри на броячите за предаване.

Памет за адреса. Съдържа 6-байтовия адрес на станцията. Контролерът позволява работа с всички видове пакети, които могат да се обменят между станциите, включени в локална мрежа от тип Ethernet — с адрес на станцията, с групов адрес и с общодостъпен адрес.

Блокът на мрежовия контролер се състои от две части, реализирани чрез две специализирани интегрални схеми: контролер за достъп до съобщителната среда и кодер-декодер.

Контролерът за достъп до съобщителната среда има следните функции:

— реализира функциите на каналния слой от мрежовата архитектура, т.е. оформя пакет със съответните служебни полета за осигуряване на синхронизацията в началото на пакета и за проверка на верността на данните в неговия край (тези полета се генерират при предаване на кадър и се отстраняват след използването им по съответния начин при приемане);

— реагира при наличие на конфликт при предаване — чрез изчакване и предаване на информацията;

— изработва сигнали при предаване и приемане, необходими за работата на останалата част от схемата;

— преобразува от паралелен в последователен код и обратно.

Кодер-декодерът е частта от схемата, която реализира връзката на контролера с предавателя. Неговите функции са следните: изработва необходимата за работа на останалата схема тактова поредица от 10 МHz, кодира данните при предаване на Манчестерски код и при приемане ги декодира, изработва необходимите сигнали за приемане и предаване към приемопредавателя и контролера за достъп до съобщителната среда.

Изходите на кодер-декодера към приемопредавателя могат да се превключват чрез мостчета както към вградените на платката приемопредавател, така и към съединителя за включване на дистанционен приемопредавател.

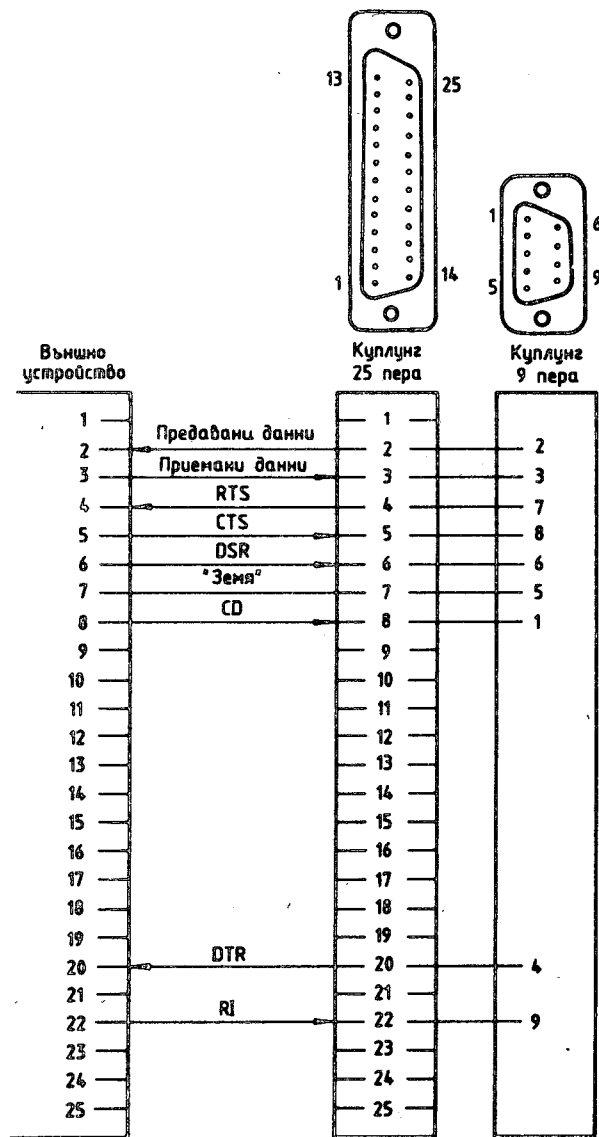
Приемопредавателят представлява монтираната върху платката схема за приемане и предаване. Неговата функция е: да развърже галванично първите два блока от физическата среда (кабели) и да преобразува сигналите от кодер-декодера в необходимите за предаване в кабела и обратно.

3.8.4. КОМУНИКАЦИОННИ АДАПТЕРИ ЗА ОТДАЛЕЧЕНО СВЪРЗВАНЕ

Физически интерфейси за отдалечена връзка между ПК

Комуникационните адаптери за отдалечено свързване могат да служат както за връзка между отдалечени ПК, така и за свързване на ПК към други типове ЕИМ или устройства. Ето защо такива връзки се осъществяват винаги по стандартни интерфейси. Засега най-широко разпространение е намерила връзката чрез интерфейс С2. Международен аналог на тази връзка е препоръката на ССІТТ V.24 или американския стандарт на EIA RS-232C. В СССР С2 се стандартизира от ГОСТ 18145—81.

Интерфейсът С2 се е появил като спецификация на връзките между крайното оборудване (терминали, а в случая ПК) и апаратурата за предаване на данни (модеми). Предмет на указаните стандарти са общите характеристики на връзката (скорост на предаване, форма на представяне на информацията и др.), механичните данни (тип на куплунга, номера на перата и др.), електрическите параметри (нива на сигналите), видовете



Фиг. 3.73. Основни сигнални вериги на С2

връзки (данни, управление, синхронизация и заземяване) и някои процедурни условия (установяване на връзка, избиране и др.).

В общи линии ПК се придържат към стандарта RS232C, като в някои случаи правят изключение само при използване на допълнителни линии, неспецифицирани в стандарта.

На фиг.3.73 са показани основните сигнални вериги, използвани от ПК, а също така номерата на перата на съответния куплунг. В повечето случаи интерфейсът на комуникационните адаптери е изведен върху „мъжки“ куплунг тип Cannon с 25 крачета. В някои случаи обаче се използва куплунг тип Cannon с 9 крачета. На фиг.3.73 е показано и съответствието между изводите на двата типа куплунги.

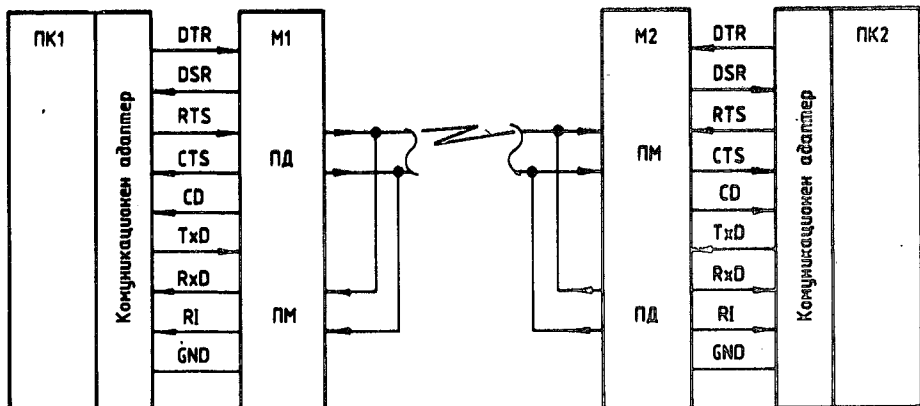
Основните сигнални вериги, използвани в ПК, са следните:

TxD	(Transmitted Data) Предавани данни;
RxD	(Received Data) Приемани данни;
RTS	(Request to Send) Заявка за предаване от ПК към модема;
CTS	(Clear to Send) Сигнал към ПК, че приемникът е готов;
DSR	(Data Set Ready) Модемът е готов за предаване;
GND	(Signal Ground) Сигнална земя;
CD	(Carrier Detected) Означава откриване на носещата честота от модема-детектор;
DTR	(Data Terminal Ready) ПК издава сигнал към модема за установяване на връзка;
RI	(Ring Indicator) Използва се при работа по комутируема линия и е признак за повикване.

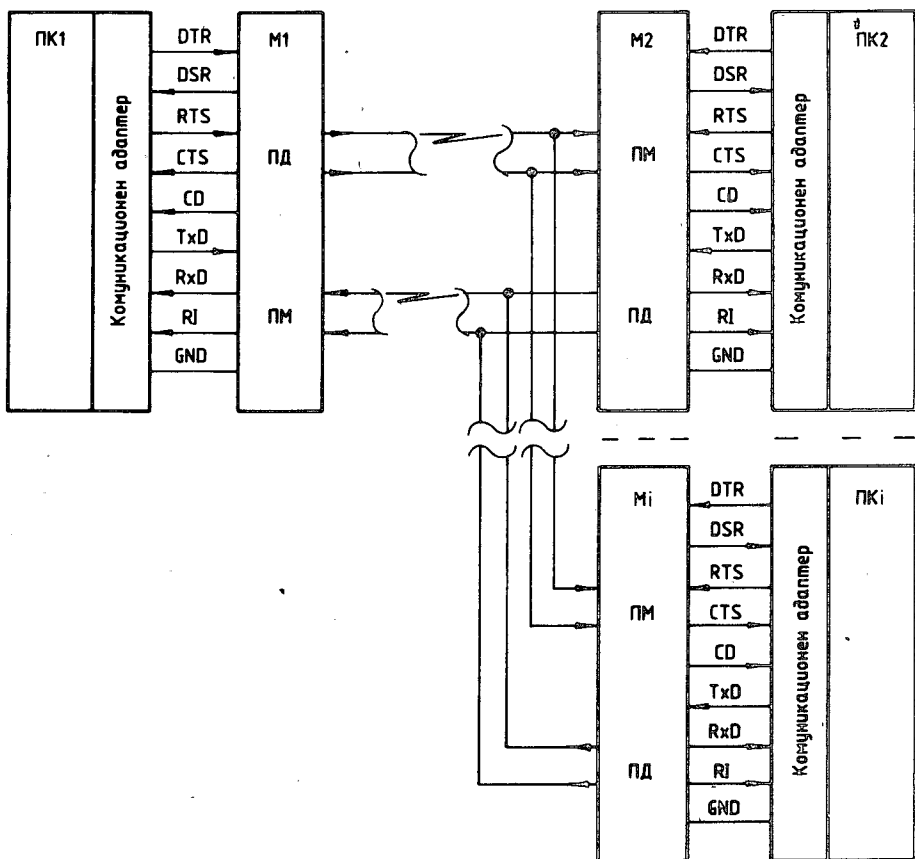
По-долу е приведена последователността от действия при три предимно срещани случая за установяване на отдалечена връзка между ПК.

Установяване на връзка от типа точка—точка между ПК по некомутируема телефонна линия (фиг.3.74):

1. ПК от двата края на връзката установяват сигнал DTR.
 2. Свързаните към ПК модеми трябва да отговорят със сигнал DSR.
 3. M1 издава сигнал RTS. В резултат на това M1 издава носещата честота.
 4. M2 детектира носещата честота и генерира сигнал CD към ПК2.
 5. След известно време M1 изпраща към ПК1 сигнал CTS, че предаването може да започне.
 6. ПК1 издава в последователен код данните по линията TxD към M1.
 7. В M1 се извършва модулиране на носещата в съответствие с постъпващите данни.
 8. В M2 се извършва демодулация на данните, които се изпращат в последователен код към ПК2 по линията RxD.
 9. След завършване на предаването ПК1 дезактивира линията RTS, което принуждава M1 да прекрати издаването на носещата честота и да дезактивира сигнала CTS.
 10. M2 дезактивира сигнала CD.
- В случай че следва предаване от ПК2 към ПК1, то протича по аналогичен начин, а именно:
11. ПК2 установява RTS, което предизвиква издаване на носещата от M2.
 12. M2 активира линията CTS.
 13. M1 детектира носещата и изпраща към ПК1 сигнала CD.
 14. ПК1 е готов за приемане на данни от ПК2.



Фиг.3.74. Установяване на връзка между ПК по некомутируема телефонна линия при съединения от типа точка-точка



Фиг.3.75. Установяване на връзка при многоточково свързване по некомутируема линия

Установяване на връзка при многоточково свързване по некомутируема линия (фиг.3.75):

1. Предварително се установява връзката между ПК1 и M1, което води до издаване на носещата от M1 към всички подчинени ПК. (Това е последователността DTR, DSR, RTS, CTS.)

2. Главният ПК (в случая ПК1) издава по линията TxD адреса на подчинения ПК. M1 модулира носещата с този адрес, който се разпространява до всички подчинени ПК.

3. Всички модеми на подчинения ПК демодулират адреса и го изпращат към съответните ПК по линиите RxD.

4. ПК_i, за когото е предназначено съобщението, разпознава адреса и активира линията RTS, което кара M1 да започне издаване на носещата. Останалите ПК продължават да следят за адрес по линията RxD.

5. Управляващият ПК1 получава от M1 сигнала CD при откриване на носещата.

6. M1 активира към ПК2 сигнала CTS, с което последният може да предаде съобщение към ПК1.

7. Следва дезактивиране на RTS от ПК2, което кара M2 да престане да издава носеща честота.

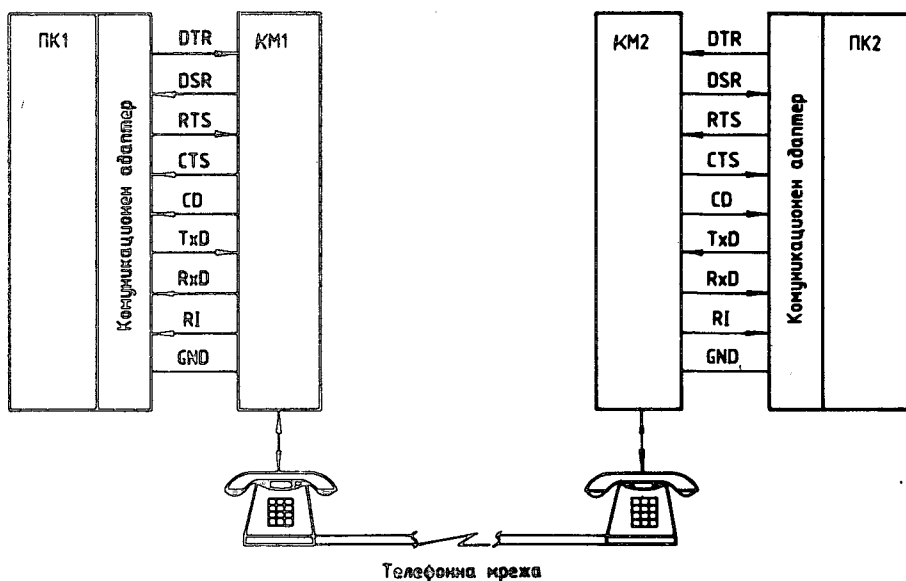
8. M1 престава да приема носещата и сигнала CD.

9. Подчиненият ПК отново е в режим на приемане, като чака да бъде отново избран от главния ПК.

Установяване на връзка от типа точка—точка между ПК по комутируема линия (фиг.3.76):

1. Отначало ПК1 е с установен сигнал DTR, готов за комуникация.

2. Операторът от ПК1 вдига телефонната слушалка, която генерира съответен сигнал към комутируемия модем KM1.



Фиг.3.76. Установяване на връзка от типа точка-точка между ПК по комутируема линия

3. КМ1 детектира този сигнал, след което превключва линията към телефонната връзка.

4. Операторът от ПК1 превключва данни/разговор в положение „разговор“, за да свърже слушалката към комуникационната линия, след което набира телефонния номер на оператора при ПК2.

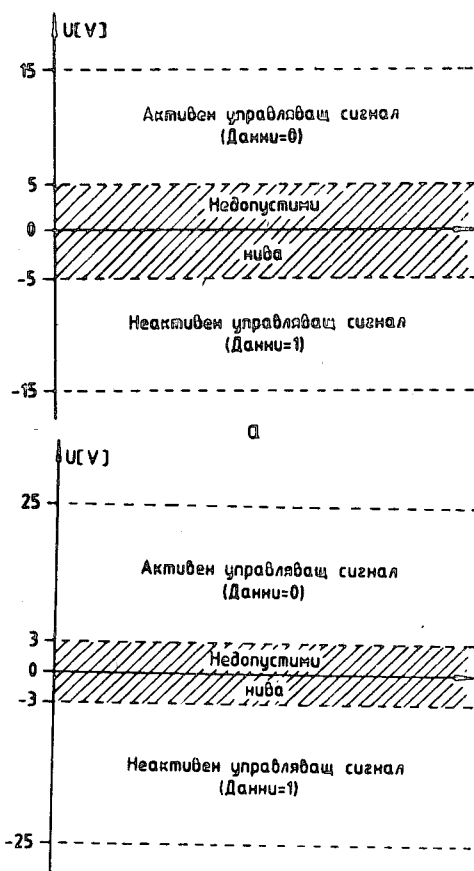
5. При повикване КМ2 генерира към ПК2 сигнал R1, след което ПК2 активира DTR към КМ2. Това предизвиква задействане на схемите за автоматичен отговор и изпращане на съответен сигнал към ПК1.

6. Операторът на ПК1 чува съответния сигнал и превключва ключа данни/разговор в позиция „данни“, което позволява свързването на КМ1 към комуникационната линия. КМ1 активира сигнала DSR, с което извества ПК1, че връзката е установена и предаването може да започне.

7. Последователността за установяване на връзката по-нататък е същата, както и в случая на използване на некомутируема линия от типа точка-точка.

8. При прекратяване на връзката и двата ПК дезактивират сигналите си DTR, за да предизвикат изключване на съответните комутируеми модеми от телефонната мрежа.

Стандартът специфицира освен сигналните вериги и електрическите нива на сигналите, които се намират в пределите на ± 12 V. Съществуват специални приемници и предаватели, които преобразуват TTL-сигналите по C2 и обратно. На фиг.3.77а са показани допустимите от стандарта нива на управляващите сигнали и данните на изходите на предавателите, а на фиг.3.77б — на входовете на приемниците. Трябва да се обърне внимание на факта, че управляващите сигнали и данните се кодират по различен начин. Така например нивото, означаващо активно състояние на управляващ сигнал, има значение логическа 0 по линиите за данни, а нивото, означаващо отсъствие на управляващ сигнал — логическа 1 по линиите за данни.



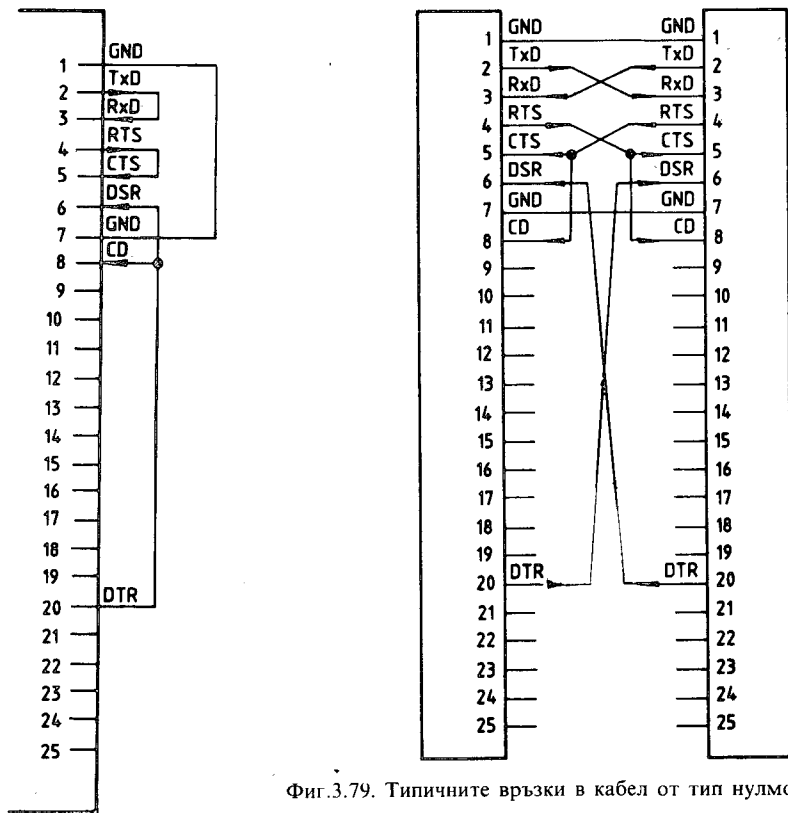
Фиг. 3.77 Допустими нива на управляващите сигнали и данните по C2.

а — на изходите на предавателите
 б — на входовете на приемниците

В комуникационните адаптери са предвидени тестове за проверка и диагностика. За да бъде възможна проверката без наличието на втори възел, обикновено се правят съответните обратни връзки, така че да бъдат проверени основните информационни пътища и управляващи сигнали. Тези обратни връзки са предвидени на различни нива, като последното от тях включва проверка на приемопредавателите, на куплунгите, а понякога и на съединителните кабели. В този случай обратната връзка се осъществява чрез специални устройства (често наричани *заглушки*), които осигуряват както обратно връщане на предаваните данни на входа на приемника, така и проверката за установяване на управляващите сигнали.

На фиг.3.78 е показана схемата на обратните връзки за тестване на C2. Те са реализирани върху куплунг от типа Canon с 25 пера.

Въпреки че комуникационните адаптери за отдалечено свързване са предназначени преди всичко за връзка чрез модеми, често на тяхна основа се строят и местни връзки. Това дава възможност за използване на готови програмни продукти при параметри на връзката, които задоволяват потребителя в конкретни случаи. Местното свързване става чрез специален кабел, наричан обикновено *нулев модем* или *нулмодем*. На фиг.3.79 са показани типичните връзки в такъв кабел.

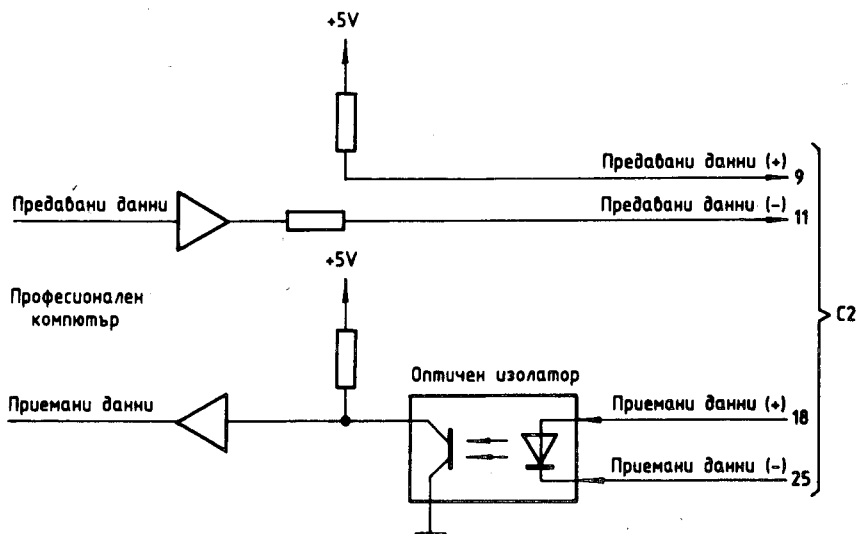


Фиг.3.79. Типичните връзки в кабел от тип нулмодем

▲ Фиг.3.78. Схемата на обратните връзки за тестване на C2

Управляващите модемни вериги са свързани по такъв начин, че не осигуряват нормално установяване на връзката, както това би станало, ако между двата ПК има модеми. Предаваните данни от ПК1 са свързани към приеманите от ПК2 и обратно, предаваните от ПК2 — към приеманите от ПК1. Сигналната земя е обща. Такова съединение е от типа точка-точка по некомутируеми линии, като по такъв начин двата ПК могат да се свързват на разстояние до няколко десетки метри непосредствено.

Друг начин на местно свързване на ПК при използването на комуникационни адаптери за отдалечена връзка е чрез т.нар. токов кръг. Тази връзка е разпространена най-вече в асинхронните адаптери на ПК. Тъй като стандартът за C2 не специфицира токов кръг, то ПК използват за тази цел част от недефинираните изводи на 25-краковия куплунг. На фиг.3.80 са показани веригите за предаване и приемане на данни с използване на токов кръг, а също така и използваните пера на куплунга. В този случай модемните управляващи сигнали не участват във връзката, по линията се работи само с данни. При използването на токов кръг е възможно както свързване от типа точка-точка, така и многоточково свързване с един главен и няколко подчинени ПК. При такава връзка е възможна работа със скорости до 9,6 Кбита/s на разстояние до няколко километра.



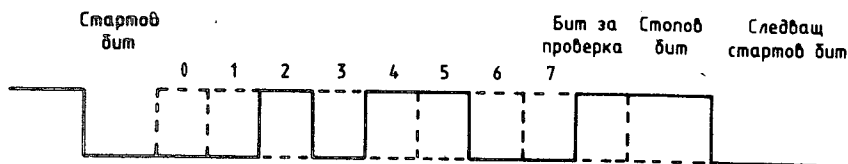
Фиг.3.80. Вериги за предаване и приемане на данни по токов кръг

Отдалечени комуникации. Протоколи за връзка

При използването на описания физически интерфейс C2 са възможни различни начини на предаване с различни комуникационни протоколи.

Според начина на синхронизация на приемника и предавателя комуникациите се делят на две основни групи: асинхронни и синхронни.

При асинхронната комуникация информацията се предава байт по байт, като синхронизацията се извършва в рамките на всеки определен байт.



Фиг. 3.81. Формат на данните при асинхронно (старт-стопно) предаване

На фиг. 3.81 е показан форматът на данните при *асинхронното* или както го наричат *старт-стопно предаване*. Нормално при отсъствие на информация линията се намира в състояние „data-marking“, което съответства на логическа 1. Сигнал за начало на байта е т.нар. *старт-бит*, който представлява установяване на линията в състояние 0 за определено време. Приемникът следи за наличието на стартовия бит и неговите параметри, като определя дали това е начало на байт или случайно сработване. Следват информационните битове, които могат да бъдат от 5 до 7 и се издават, като се започне от най-младшия бит. След това се намират т.нар. *бит за контрол по четност* и също така *стоп-битовете*. С това предаването на дадения байт завършва и може да бъде издаден старт-битът на следващия байт. По време на приемане приемникът стробира постъпващите битове, като започва от стартовия, с честота, която превишава многократно честотата на постъпващата информация (обикновено 16 или 64 пъти). Отчитането на състоянието на даден информационен бит става по средата на неговото предполагаемо място. По този начин приемането е възможно дори и при големи разлики в честотите на предавателя и приемника, тъй като вярно предаване се гарантира при изместване в рамките на половин интервал от времето за последния, осми бит. Ако изместването е по-голямо, приемникът сигнализира за грешка при приемане на кадъра (framing error).

Обикновено комуникационните адаптери за асинхронно свързване на ПК имат възможност за задаване на голямо разнообразие от параметри на връзката (скорост на предаване, брой на битовете в сигнала, наличие на бит за четност, контрол по четност или нечетност, брой на стоп-битовете). Това е предвидено, за да стане възможно свързването на ПК към различни типове линии, през различни модеми, а също така ПК да може да бъде свързан към други изчислителни устройства с подобни параметри на връзката.

При синхронната комуникация информацията се предава във вид на блокове от данни — *кадри*, като синхронизацията е обща за целия кадър. Това се постига чрез т.нар. *синхросимволи*, които предхождат всеки кадър. Видът на синхросимволите е такъв, че те не могат да бъдат срещнати като байтове вътре в полето за данни на кадъра. В зависимост от използвания протокол за връзка това се постига по различен начин.

В ПК са реализирани двата най-широко разпространени протокола, представители на двата основни класа синхронни протоколи: битово ориентираният протокол SDLC (Synchronous Data Link Communication) и байтово ориентираният протокол BSC (Binary Synchronous Communication).

На фиг.3.82 е показан форматът на кадъра при работа с протокол SDLC. Началото на кадъра се задава от синхросимвол, който в случая представлява т.нар. *флаг* и винаги има вида 01111110. Следват адресно-

Флаг 01111110	Адрес 8 бита	Управление 8 бита	Данни произволна дължина	Контрол 16 бита	Флаг 01111110
------------------	-----------------	----------------------	-----------------------------	--------------------	------------------

Фиг.3.82. Формат на кадъра при работа с протокол SDLC

то поле, полето за управление и полето за данни. Кадърът завършва с поле за контрол на грешките с помощта на цикличен код и отново с флаг 01111110. Флагът е уникална комбинация от нули и единици, която не може да бъде срещната вътре в кадъра. За тази цел е предвидена специална техника на вмъкване и изтриване на нули. Тя се състои в следното: специална логика на предавателя следи за вида на предаваната информация. В случай че сред предаваните данни се срещне поредица от пет последователни единици, задължително след тях се вмъква нула. Приемната част също следи за вида на постъпващата информация и при приемане на поредица от пет последователни единици изтрива следващата ги нула, като по този начин възстановява първоначалния вид на информацията. По този начин кадърът съдържа произволен брой битове, който не винаги е цяло число байтове. Синхронизацията на кадъра се извършва по флагите, а побитовата синхронизация — от самите модеми.

На фиг.3.83 е показан форматът на кадъра при използване на **байтово ориентирания протокол BSC**. Той има полета с аналогично предназначение, както и при битово ориентираните протоколи. Разликата тук се състои главно в начина на синхронизиране на кадъра. Всеки кадър в BSC започва с последователност от два еднакви синхросимвола. Видът на синхросимволите е произволен и може да бъде задаван от погребителя. Полето за данни съдържа цяло число байтове. Ако възникне необходимост от предаване на такъв байт данни, чийто вид съвпада с този на синхросимвола, пред този байт се вмъква допълнително един управляващ байт. В приемната страна допълнителният байт се отстранява и информацията се възстановява. И тук синхронизацията на кадрите се извършва в комуникационния адаптер, а побитовата синхронизация — в модема.

При реализацията на комуникационните адаптери за отдалечено свързване на ПК са използвани различни подходи в различните случаи. Това се дължи главно на различните изисквания към параметрите на връзката при асинхронните, SDLC и BSC комуникации. Съществуват редица универсални синхронно/асинхронни приемопредаватели (т.нар. USART), които са програмируеми. На тяхната база би могъл да бъде изграден универсален комуникационен адаптер, който с помощта на различни драйвери да обслужва който и да е от приведените протоколи. В

SYN 00110010	SYN 00110010	Начало на заглавна част	Заглавна част	Начало на данни	Данни	Край на данни	Контрол
-----------------	-----------------	-------------------------------	------------------	--------------------	-------	------------------	---------

Фиг.3.83. Формат на кадъра при работа с протокол BSC

ПК този подход не е приложен. Тук основният стремеж е ПК да бъде разтоварен в максимална степен от функциите по обслужване на комуникациите, като ресурсите му бъдат предоставени преди всичко на потребителския процес. Ето защо при реализациите на комуникационните адаптери в различните случаи са използвани различни типове интелигентни големи ИС, обикновено специализирани за обслужване на дадения тип свързване. Това е довело до създаване във вид на отделни платки на комуникационни адаптери за асинхронна връзка, за протокол SDLC и за протокол BSC.

3.8.5. АСИНХРОННИ АДАПТЕРИ ЗА ПК

Асинхронните комуникационни адаптери за ПК са изпълнени или във вид на самостоятелни печатни платки, или в комбинация с други адаптери. Във всички случаи те са реализирани почти по един и същ начин, като поддържат стандартната асинхронна (старт-стопна) връзка.

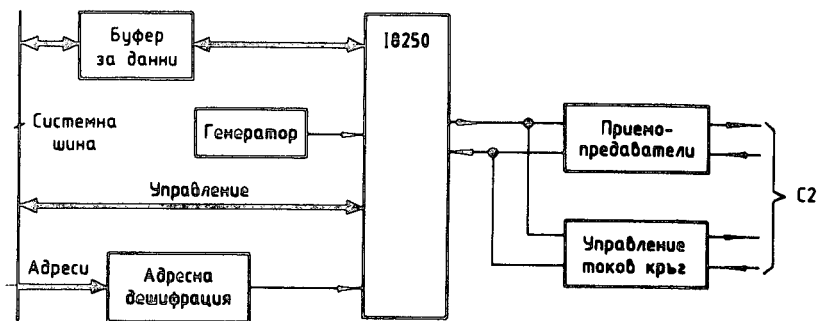
Асинхронните адаптери са напълно програмируеми. Към основните функции, които те изпълняват, могат да се отнесат следните:

- автоматично вмъкване и премахване на старт-бит, стоп-битове и бит за контрол по четност;
- скорости на предаване от 50 до 9600 бит/s;
- задаване на брой на битовете в един символ — 5, 6, 7 или 8;
- задаване на брой на стоп-битовете — 1, 1,5 или 2;
- задаване на контрол по четност/нечетност или отсъствие на контрол;

— развита система за прекъсвания, обслужваща предаването и приемането на данни, управлението на модемните вериги и ситуацията, възникнали при различни типове грешки;

— диагностични възможности за проверка чрез обратна връзка както на коректността на издаваните сигнали, така и на работата на приемниците и предавателите и др.

На фиг. 3.84 е показана структурната схема на асинхронния комуникационен адаптер. Връзката към системната шина на ПК се осъществява чрез буфери, като към адаптера са свързани шините за данни, управляващите шини и част от адресните шини. От другата страна адаптерът е



Фиг. 3.84. Структурна схема на асинхронен комуникационен адаптер

свързан към устройството за предаване на данни (модема) чрез приемо-предавателите и съответния куплунг съгласно интерфейс С2. Адаптерът има свой генератор за тактови импулси с честота 1,8432 MHz. В основата на асинхронния комуникационен адаптер лежи голямата интегрална схема от типа INS8250, която е специализиран чип за обслужване на асинхронни комуникации. Той е напълно програмируем. Ето защо преди започване на работа е необходимо да му бъдат зададени желаните параметри, като по време на работа възниква необходимост от постоянното му обслужване. Тези функции са възложени на специализирано програмно осигуряване, което представлява т.нар. драйверна програма, или просто драйвер. В ПК драйверът за обслужване на асинхронните комуникации обикновено е част от базовата входно-изходна система.

В зависимост от функциите, които изпълняват, сигналите върху изводите на INS8250 могат да се обособят в следните групи: входни сигнали, изходни сигнали, входно-изходни сигнали и захранване. Ето накратко тяхното предназначение.

Входни сигнали

Избоди

- 12,13,14 — Chip Select (CS0, CS1, CS2). Избор на чипа. Когато CS0 и CS1 са в състояние логическа 1, а CS2 — в 0, чипът е избран. Комуникация с процесорната система обаче е възможна само когато едновременно с избора на чипа е активиран сигналят за стробиране на адреса ADS.
- 22, 21 — Data Input Strobe (DISTR, $\overline{\text{DISTR}}$). Сигнал за четене от INS8250. Когато чипът е избран, а DISTR е в 1 или $\overline{\text{DISTR}}$ — в 0, възможно е прочитането на данни или състоянието от INS8250. При това е необходимо наличието само на един от сигналите. Ако някой от тези сигнали не се използва, той трябва да бъде свързан така, че да е в неактивно състояние.
- 19, 18 — Data Output Strobe (DOSTR, $\overline{\text{DOSTR}}$). Сигнал за запис в INS8250. Когато чипът е избран, а DOSTR е в 1 или $\overline{\text{DOSTR}}$ — в 0, възможен е записът на данни или команда в INS8250. При това е необходимо наличието само на един от сигналите. Ако някой от тези сигнали не се използва, той трябва да бъде свързан така, че да е в неактивно състояние.
- 25 — Address Strobe (ADS). Стробиране на адреса. Използва се за стробиране на адреса на регистъра, задаван от младшите адресни шини A2, A1 и A0. В случай, че такова стробиране не е необходимо, достатъчно е просто да бъде занулен.
- 26,27,28 — Register Select (A0,A1,A2). Шини за избор на регистър в INS8250. С тези три шини се избира необходимият регистър в INS8250. Това става в съответствие с табл. 3.53. Както се вижда, тези три шини в комбинация със сигналите за запис четене и значението на бита DLAV позволяват обръщение към 10 програмно достъпни регистъра. DLAV е старшият байт в регистъра за управление на линията и трябва да бъде предварително установен, ако е необходимо обръщение към последните два регистъра.
- 35 — Master Reset (MR). Общо нулиране. Единица на този извод води до нулиране на всички регистри на INS8250, с изключение на приемния и предавателния буфер за данни и регистрите, съдържащи фактор на делене на честотата. Както ще бъде показано по-долу, това води до установяване в начално състояние на някои изходни сигнали.
- 9 — Receive Clock (RCLK). Входна тактова честота. Върху този извод се задава честота, 16 пъти превъзпаваша зададената скорост на приемане. По този сигнал се извършва сканиране на постъпващите данни.
- 10 — Serial Input (SIN). Вход за последователно постъпващите данни, които се приемат от други ПК или от модема.
- 36 — Clear to Send (CTS). CTS е управляващ сигнал, издаван от модема. Състоянието върху този извод се отразява в бит 4 (CTS) на регистъра за състояние на модема. В същия регистър (бит 0) се намира информация за промяна в състоянието на сигнала CTS, т.е. DCTS. Ако такава промяна настъпи, това веднага

- води до генериране на заявка за прекъсване.
- 37 — Data Set Ready (DSR). Когато е в активно състояние (ниско ниво), информира INS8250, че връзката е установена и модемът е готов за предаване на данни. Състоянието на извода DSR може да бъде прочетено от процесора като бит 5 (DSR) на регистъра за състояние на модема. Бит 1 (DDSR) на същия регистър съдържа информация за промяна в състоянието на DSR. При настъпване на такава промяна се генерира заявка за прекъсване.
- 38 — Carrier Detected (CD). Откриване на носещата. Когато е в активно състояние (ниско ниво), този сигнал информира INS8250, че е открита носещата честота от модема. Състоянието на този сигнал е достъпно за процесора като бит 7 (CD) на регистъра за състояние на модема. В бит 3 (DCD) на същия регистър се съдържа информация за промяна на състоянието на този сигнал. При настъпване на такава промяна се генерира заявка за прекъсване.
- 39 — Ring Indicator (RI). Индикатор за повикване. Активният сигнал по тази линия (ниско ниво) е индикация за това, че модемът е получил телефонен сигнал за повикване. Информация за състоянието на тази линия процесорът може да получи от съдържанието на бит 6 (RI) на регистъра за състояние на модема. В бит 2 (TERI) на същия регистър се съдържа информация за това, дали състоянието на линията RI не се е променило от ниско във високо ниво за времето от последното прочитане на регистъра за състояние на модема до дадения момент. Ако такава промяна е настъпила, това води до генериране на заявка за прекъсване.

Исходни сигнали

Изводи

- 33 — Data Terminal Ready (DTR). Активното състояние на този сигнал (ниско ниво) информира модема, че INS8250 е готов за комуникация. Този сигнал може да се установи в активно състояние чрез запис на 1 в бит 0 (DTR) на регистъра за управление на модема. По време на общото нулиране DTR се установява в неактивно състояние (високо ниво).
- 32 — Request to Send (RTS). Активното състояние на този сигнал (ниско ниво) информира модема, че INS8250 е готов за предаване на данни. Този сигнал може да се установи в активно състояние чрез запис на 1 в бит 1 (RTS) на регистъра за управление на модема. По време на общото нулиране RTS се установява в неактивно състояние (високо ниво).
- 34 — Output 1 (OUT1). Това е изход, предоставен за използване по усмотрение на потребителя. Сигналят може да бъде установен в активното си състояние (ниско ниво) чрез запис на 1 в бит 2 (OUT1) на регистъра за управление на модема. По време на общото нулиране OUT1 се установява в неактивно състояние (високо ниво).
- 31 — Output 2 (OUT2). Това е изход, предоставен за използване по усмотрение на потребителя. Сигналят може да бъде установен в активното си състояние (ниско ниво) чрез запис на 1 в бит 3 (OUT2) на регистъра за управление на модема. По време на общото нулиране OUT2 се установява в неактивно състояние (високо ниво).
- 24 — Chip Select Out (CSOUT). Активното състояние на този сигнал (високо ниво) означава, че е извършен избор на INS8250 (сигналите CS0, CS1 и CS2 са активирани). Обменът на информацията между процесорната система и INS8250 е възможен само ако CSOUT е в състояние 1.
- 25 — Driver Disable (DDIS). Този сигнал се установява в 0 по време на четене на данни от INS8250 от страна на процесора.
- 15 — Baud Out (BAUDOUT). Върху този извод присъства тактова поредица с честота, която е 16 пъти по-голяма от зададената скорост на предаване. Честотата на предаване се задава чрез запис в регистрите на фактора на делене на честотата. Това е число, показващо на колко се дели честотата на генератора 1,8432 MHz, за да се получи честота, шестнадесеткратно превишаваща желаната скорост на предаване. Обикновено този извод се свързва към RCLK (извод 9) и задава скорост на приемане, еднаква с тази на предаване.
- 30 — Interrupt (INTRPT). Прекъсване. Сигналят по тази линия се установява в ак-

тивно състояние (високо ниво) във всички случаи, когато настъпи едно или няколко от следните условия за прекъсване: флаг за грешка при приемане, приет байт данни, регистъра за предаване на данни е празен, промяна в състоянието на модемните линии. Сигналят за прекъсване INTRPT се нулира или при обслужване на съответното прекъсване, или при общото нулиране.

11 — Serial Output (SOUT). Изход за последователно предаваните данни, които се изпращат към другия ПК или модема. Общото нулиране установява този изход в състояние 1 (това е неактивното състояние на линията при отсъствие на предаване).

Входно-изходни сигнали

Изводи

- 1—8 — Data Bus (D7—D0). Шини данни. Върху тези шини се съдържат осемте бита данни. Линиите са с три състояния. По тези шини се извършва двупосочното предаване между INS8250 и процесора на данни, команди и състояния.
- 16, 17 — External Clock Input/Output (XTAL1, XTAL2). Линии за тактова честота. През тези изводи се задава външната тактова честота. Възможно е свързването както на външен тактов генератор, така и задаването на честотата единствено с помощта на кварцов резонатор.

Захранване

- 40 — VCC. Захранване +5 V.
 20 — GND. Захранване 0 V (земя).
 Извод 29 на INS8250 не се използва.

Таблица 3.53

Адресиране на регистри в INS8250

DLAB	A2	A1	A0	Регистър	Мнемоника
0	0	0	0	Приеман буфер за данни (четене)	RBR
0	0	0	0	Предавателен буфер за данни (запис)	THR
0	0	0	1	Разрешение на прекъсванията	IER
X	0	1	0	Идентифициране на прекъсванията (само четене)	IR
X	0	1	1	Управление на линията	LCR
X	1	0	0	Управление на модема	MCR
X	1	0	1	Състояние на линията	LSR
X	1	1	0	Състояние на модема	MSR
X	1	1	1	(Не се използва)	
1	0	0	0	Фактор за делене на честотата (младши байт)	DLL
1	0	0	1	Фактор за делене на честотата (старши байт)	DLM

x — стойността на DLAB не влияе на избора на регистър.

В табл.3.54 са показани условията за нулиране, както и състоянието, в което се установяват при нулиране програмно достъпните регистри и някои от изходните сигнали на INS8250.

Цялото управление на асинхронната комуникация, както и обменът на данни, се извършва чрез програмно достъпните регистри на INS8250.

Следва по-подробна информация за предназначението на регистрите и отделните битове в тях.

Условия за нулиране на INS8250

Регистър/сигнал	Тип на нулирането	Състояние след нулирането
Регистър за разрешение на прекъсванията	Общо нулиране	Всички битове в 0 (от 0 до 3 се нулират, а от 4 до 7 са винаги 0)
Регистър за идентифициране на прекъсванията	Общо нулиране	Бит 0 се установява в 1, битове 1 и 2 — в 0, битове от 3 до 7 са винаги 0
Регистър за управление на линията	Общо нулиране	Всички битове се нулират
Регистър за управление на модема	Общо нулиране	Всички битове се нулират
Регистър за състояние на линията	Общо нулиране	Битове 5 и 6 се установяват в 1, останалите — в 0
Регистър за състояние на модема	Общо нулиране	Битове от 0 до 3 се нулират, битовете от 4 до 7 отразяват състоянието на входните сигнали.
SOUT	Общо нулиране	1
INTRPT (RCVR ERROR)	Четене на LSR Общо нулиране	0
INTRPT (RCVR Data Ready)	Четене на RDP Общо нулиране	0
INTRPT (TRSMT Data Ready)	Четене на IIR Запис в THR Общо нулиране	0
INTRPT (Modem Status Changes)	Четене на MSR Общо нулиране	0
OUT2	Общо нулиране	1
RTS	Общо нулиране	1
DTR	Общо нулиране	1
OUT1	Общо нулиране	1

Регистър за управление на линията (LCR)

В този регистър се задават параметрите на асинхронната връзка. Това става чрез запис на управляващата информация в LCR в съответствие с желаните параметри и назначението на отделните битове. Регистърът е достъпен за четене, благодарение на което зададените параметри могат лесно да бъдат контролирани. По този начин отпада необходимостта от запомняне на тези параметри в оперативната памет.

Предназначение на отделните битове:

Битове 0,1 — с тези битове се задава дължината на предаваните и приеманите символи

Бит 1	Бит 0	Дължина на символа
0	0	5 бита
0	1	6 бита
1	0	7 бита
1	1	8 бита

- Бит 2** — задава броя на стоп-битовете в символа. Ако в бит 2 е записана 0, генерира се или се проверява един стоп-бит. Ако в бит 2 е записана 1, а чрез битове 0 и 1 е зададен символ с дължина 5 бита, тогава се генерират или проверяват 1,5 бита. Ако обаче в бит 2 е записана 1, но е зададен символ с дължина 6, 7 или 8 бита, генерират се или се проверяват два стоп-бита.
- Бит 3** — управлява контрола по четност. Ако този бит е в 1, генерира се (при предаване) или се проверява (при приемане) битът за контрол по четност или нечетност.
- Бит 4** — задава типа на контрола по четност или нечетност. Значението на този бит има смисъл само в случаите, когато е зададен контрол, т.е. когато бит 3 е в 1. Ако в бит 4 е записана 0, общият брой единици (включително бита за контрол) при предаване или при приемане трябва да да бъде нечетен. Единица в бит 4 означава общ четен брой единици в предавания или приеман символ.
- Бит 5** — това е т.нар. „долепен“ бит. Ако този бит има стойност 1 и е зададен контрол (бит 3 = 1), когато бит 4 = 0, битът за четност ще бъде 1. В случай че бит 4 = 0, битът за четност ще бъде 0.
- Бит 6** — чрез този бит линията се установява в състояние „break“. Когато този бит се установи в състояние 1, на изхода за последователно предаване SOUT се установява 0 независимо от другите условия за предаване на данни. Това състояние продължава догавата, докато в бит 6 не бъде записана отново 0. По този начин става възможно програмното генериране на сигнал „break“ по линията.
- Бит 7** — използва се допълнително при адресацията на регистрите. Ако той е в 1, осигурява се достъп до регистрите, съдържащи фактора на делене на честотата (вж.табл.3.53). Когато този бит е в състояние 0, по същите адреси могат да бъдат избрани регистрите на приемния или предавателния буфер или пък на регистъра за разрешение на прекъсванията.

Регистри за фактора на делене на честотата (DLL и DLM)

Регистрите за фактора на делене на честотата са два: единият от тях (DLL) съдържа младшия байт, а другият (DLM) — старшия байт на фактора на делене на честотата. По такъв начин в двата регистъра може да бъде записана 16-разредна дума, или число от 1 до $2^{16}-1$. Това число показва на колко се дели тактовата честота на чипа INS8250 (1,8432 MHz), за да се получи честотата на стробирание. Тя от своя страна е 16 пъти по-висока от честотата, с която данните се предават (или приемат). По такъв начин за всяка от желаните скорости на предаване (СП) факторът на делене (ФД) може да бъде изчислен по формулата

$$\text{ФД} = \frac{1,84332 \cdot 10^6}{16 \times \text{СП}}$$

Стойността на фактора на делене на честотата трябва предварително да бъде заредена в DLL и DLM съответно. Тези стойности се прехвърлят в брояча автоматично и незабавно след зареждането на регистрите.

В табл.3.55 са показани стойностите на фактора на делене за поредица от най-широко разпространените скорости на предаване при асинхронни комуникации. От таблицата се вижда, че за дадени скорости на предаване точната честота не се осигурява, но погрешността е достатъч-

но малка, за да не предизвика неизправно приемане в рамките на един символ дори когато дължината му е максимална. Предаване със скорост, превишаващи 9600 бит/s, не се препоръчва поради прогресивно нарастване на погрешността.

Таблица 3.55

Задаване на скоростта на предаване при асинхронни комуникации			
Желана скорост на предаване, бит/s	Фактор на делене на 10-ична стойност	Честотата 16-ична стойност	Процентна грешка
50	2304	900	—
75	1536	600	—
110	1047	417	0,026
134,5	857	369	0,058
150	768	300	—
300	384	180	—
600	192	0C0	—
1200	96	060	—
1800	64	040	—
2000	58	03A	0,69
2400	48	030	—
3600	32	020	—
4800	24	018	—
7200	16	010	—
9600	12	00C	—

Регистър за състояние на линията (LSR)

Регистърът за състояние на линията предоставя на процесора всички необходими сведения за състоянието на комуникационната линия в даден момент.

Предназначение на отделните битове:

- Бит 0 — индикатор за приети данни. Той се установява в 1, когато в приемния буферен регистър е приет поредният байт данни. Този бит се нулира или при четене на приемния буферен регистър, или чрез директен запис на 0 в бит 0.
- Бит 1 — индикатор за препълване. В този бит стойност 1 показва, че поредният байт данни е приет в приемния буферен регистър, преди процесорът да е прочел от него предишния байт данни. При това значението на първия от приетите байтове данни се загубва. Този бит се нулира при прочитане на състоянието от LSR.
- Бит 2 — индикатор за грешка при контрол по четност (или нечетност). Той се установява в 1, когато приетият байт не съдържа четен (или нечетен) брой единици. Контролът по четност или нечетност се задава чрез битовете за контрол и четност LCR. Този бит се нулира при прочитане на състоянието от LSR.
- Бит 3 — индикатор за грешка при приемане. Показва, че не е получен коректен стоп-бит на символа. Той се установява в 1, когато след информационните битове и бита за четност не е открита единицата на стоп-бита, а е открита нула.
- Бит 4 — индикатор за състояние „break“ по линията при приемане. Той се установява в 1, когато линията на входа се задържа в състояние 0 за време, превишаващо пълното време за приемане на един символ, т.е. времето за приемане на стар-бита, информационните битове, бита за контрол и стоп-битовете.
- Бит 5 — показва, че предавателният буферен регистър е празен и може да приеме поредния байт данни. Този бит предизвиква прекъсване към процесора (когато то е разрешено). Бит 5 винаги се установява след запис на поредния байт данни от предавателния буферен регистър в изместващия предавателен регистър за издаването на последователния му код по линията. Със зареждането на следващия символ за предаване бит 5 автоматично се нулира.

Бит 6 — показва, че съдържанието на измествания предавателен регистър е предадено. Той се нулира при записа на поредния символ от предавателния буферен регистър в премествания предавателен регистър. Бит 6 е програмно достъпен само за четене.

Бит 7 — Не се използва. Стойността му винаги е 0.

Битовите на LSR от 1 до 4 включително се активира г при възникване на грешки при приемане. Активирането на който и да е от тях води до издаване на заявка за прекъсване

Регистър за идентификация на прекъсването (IRR)

Чипът INS8250 има възможност за издаване на една единствена заявка за прекъсване, а причините, които я предизвикват, могат да бъдат различни. Затова в него са предвидени схеми за определяне на приоритетите на прекъсванията, като на процесора се предоставя допълнителна информация за източника на прекъсване чрез IIR. Това свежда до минимум необходимостта от допълнителна програмна намеса за обслужване на прекъсванията. Прекъсванията в INS8250 са групирани в следните четири нива: най-висок приоритет (приоритет 1) имат прекъсванията от състоянието на линията, следва прекъсването при приети данни (приоритет 2), прекъсването при празен буферен регистър за предаване (приоритет 3) и прекъсвания от промяна на състоянието на модема (приоритет 4). Регистърът за идентифициране на прекъсването съдържа информация за това, че е възникнала заявка за прекъсване (в съответствие с установените приоритети), а също индикация за източника на заявката за прекъсване. При това заявките за прекъсване с по-нисък приоритет не се обслужват, докато не бъде обслужена заявката с най-висок приоритет.

Типове прекъсвания в INS8250

Таблица 3.56

Стойност на битовите 0,1,2 в IIR			Источник на прекъсването и условия за снемане на заявката за прекъсване	Источник на прекъсването	Условие за снемане на заявката за прекъсване	
бит 2	бит 1	бит 0				
0	0	1	—	Няма прекъсване	—	
1	1	0	1	Състояние на линията при приемане	Прекъсване по препълване (бит 1 на LSR) Грешка при контрола на четност нечетност (бит 2 на LSR) Грешка при приемане (бит 3 на LSR) Състояние „break“ (бит 4 на LSR)	Четене на състоянието от LSR
1	0	0	2	Наличие на приет байт данни	Приет байт данни (бит 1 на LSR)	Четене от приемния буферен регистър
0	1	0	3	Празен буферен регистър за предаване	Готовност за предаване (бит 5 на LSR)	Запис в буферния регистър за предаване или четене на IIR
0	0	0	4	Състояние на модема	Промяна в състоянията: CTS (бит 0 на MSR), DSR (бит 1 на MSR), RI (бит 2 на MSR), CD (бит 3 на MSR)	Четене на състоянието

Предназначение на отделните битове:

Бит 0 — активира се при наличие на заявка за прекъсване. Може да се използва, когато не се прилага системата на прекъсване, а се работи в режим на полиране.

Битове 1,2 — използват се за идентифициране на типа на прекъсването. Чрез тях се кодира прекъсването, имащо в дадения момент най-висок приоритет съгласно табл.3.56.

Битове 3,4,5,6,7 — не се използват и винаги съдържат 0.

Регистър за разрешение на прекъсванията (IER)

Този регистър позволява някои или всички типове прекъсвания да бъдат маскирани. За всеки от четирите типа прекъсвания е отделен по един бит. Единица в този бит означава, че съответният тип прекъсване е разрешен, а нула — че съответният тип прекъсване е забранен. Когато всички прекъсвания са забранени, забранява се изходът на INS8250 INTRPT, а също и регистърът за идентифициране на прекъсванията. В същото време INS8250 продължава да работи нормално, като информацията в LSR и MSR е напълно валидна.

Предназначение на отделните битове:

Бит 0 — когато е в 1 разрешава прекъсването при наличие на приети данни.

Бит 1 — когато е в 1 разрешава прекъсването при празен буферен регистър за предаване.

Бит 2 — когато е в 1 разрешава прекъсването при възникване на грешки по линията.

Бит 3 — когато е в 1 разрешава прекъсването при промяна в състоянието на модема.

Битове 4,5,6,7 — не се използват. Състоянието им винаги е 0.

Регистър за управление на модема (MCR)

Чрез този регистър се осъществява управлението на модемните вериги към модема или пък към съответното периферно устройство или друг ПК.

Предназначение на отделните битове:

Бит 0 — управлява веригата DTR. При стойност 1 предизвиква активиране на съответния извод DTR на INS8250 (ниско ниво).

Бит 1 — управлява веригата RTS. При стойност 1 предизвиква активиране на съответния извод RTS на INS8250 (ниско ниво).

Бит 2 — управлява състоянието на извода OUT1 на INS8250. При стойност 1 предизвиква активиране на съответния извод OUT1 на INS8250 (ниско ниво).

Бит 3 — управлява състоянието на извода OUT2 на INS8250. При стойност 1 предизвиква активиране на съответния извод OUT2 на INS8250 (ниско ниво).

Бит 4 — осигурява обратната връзка за тестване на INS8250. Когато бит 4 е в 1, последователният изход за предаване на данни (SOUT) се установява в състояние 1, последователният вход за приемане (SIN) се изключва от приемника, като изходът на преместващия регистър за предаване директно се свързва към приемния преместващ регистър вътре в чипа. Четирите входни сигнала откъм модема (CTS, DSR, CD и RI) се изключват от изводите на INS8250, като четирите изходни сигнала на MCR (DTR, RTS, OUT1 и OUT2) вътрешно се превключват към четирите входа за сигналите откъм модема. В режима на вътрешна диагностика предаваните данни се приемат непосредствено, като това позволява тестване на основните информационни пътища и управляващи вериги на INS8250. В този режим също така се проверява функционирането на системата за прекъсване. С повторното установяване на бит 4 в 0 нормалният режим на работа се възстановява.

Битове 5,6,7 — не се използват. Състоянието им винаги е 0.

Регистър за състоянието на модема (MSR)

Чрез този регистър на процесора се предоставя информация за текущото състояние на модемните линии. Освен това в този регистър се съдържа информация за промяна в състоянието на модемните линии. При настъпване на такава промяна съответните битове се установяват в 1 и възниква заявка за прекъсване при промяна в състоянието на модема (приоритет 4). Тези битове възстановяват предишното си състояние след прочитане на информацията от MSR.

Предназначение на отделните битове:

- Бит 0 — отразява промяната на състоянието на входа CTS на INS8250 от последното му прочитане от процесора до дадения момент.
- Бит 1 — отразява промяната на състоянието на входа DSR на INS8250 от последното му прочитане от процесора до дадения момент.
- Бит 2 — отразява промяната на състоянието на входа RI на INS8250 от последното му прочитане от процесора до дадения момент.
- Бит 3 — отразява промяна на състоянието на входа CD на INS8250 от последното му прочитане от процесора до дадения момент.
- Бит 4 — отразява състоянието на входа CTS на INS8250 (активното му състояние е високо ниво). При наличието на обратна връзка (бит 4 на MCR в 1) този бит се свързва към бит 1 на регистър MCR.
- Бит 5 — отразява състоянието на входа DSR на INS8250 (активното му състояние е високо ниво). При наличието на обратна връзка (бит 4 на MCR в 1) този бит се свързва към бит 0 на регистър MCR.
- Бит 6 — отразява състоянието на входа RI на INS8250 (активното му състояние е високо ниво). При наличието на обратна връзка (бит 4 на MCR в 1) този бит се свързва към бит 2 на регистър MCR.
- Бит 7 — отразява състоянието на входа CD на INS8250 (активното му състояние е високо ниво). При наличието на обратна връзка (бит 4 на MCR в 1) този бит се свързва към бит 3 на регистър MCR.

Приеман буферен регистър (RBR)

В този регистър се съхранява постъпващият символ (само информационните му битове). Символът постъпва на входа на регистъра с най-младшия бит напред.

Буферен регистър за предаване на данни (THR)

В този регистър се записва символът (само информационните му битове), подлежащ на предаване. Следва презапис на този символ в предавателния изместващ регистър, откъдето символът започва да се предава по линията последователно, като се започне от най-младшия бит.

В ПК е предвидена възможност за работа с два комуникационни асинхронни адаптера — основен и допълнителен. При това са фиксирани определени адреси от входно-изходното адресно пространство за обръщане към програмно-достъпните регистри на всеки от тях. В табл.3.57 са дадени тези адреси заедно с допълнителните условия за адресиране (DLAB, запис, четене), а в табл.3.58 са показани в двоичен вид и състоянията на съответните адресни шини. С × са означени случаите, когато стойността на DLAB не влияе на избора на регистър.

Регистри на основния и допълнителния адаптер

Входно-изходни адреси		Регистър	Състояние на DLAB
Основен адаптер	Допълнителен адаптер		
3F8	2F8	Буфер за предаване	0 (запис)
3F8	2F8	Буфер за приемане	0 (четене)
3F8	2F8	Фактор за делене на честотата (млад-ши байт)	1
3F9	2F9	Фактор за делене на честотата (стар-ши байт)	1
3F9	2F9	Разрешение на прекъсванията	×
3FA	2FA	Идентифициране на прекъсването	×
3FB	2FB	Управление на линията	×
3FC	2FC	Управление на модема	×
3FD	2FD	Състояние на линията	×
3FE	2FE	Състояние на модема	×

Таблица 3.58

Адресиране на основния и допълнителния адаптер

Адрес											Регистър
A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	DLAB	
1	10	1	1	1	1	1	0	0	0	0	RBR(четене) THR(запис)
1	10	1	1	1	1	1	0	0	1	0	IER
1	10	1	1	1	1	1	0	1	0	×	IR
1	10	1	1	1	1	1	0	1	1	×	LCR
1	10	1	1	1	1	1	1	0	0	×	MCR
1	10	1	1	1	1	1	1	1	0	×	MSR
1	10	1	1	1	1	1	1	1	1	×	Не се използва
1	10	1	1	1	1	1	0	0	0	1	DLL
1	10	1	1	1	1	1	0	0	1	1	DLM

↑ ↑
основен допълнителен
адаптер адаптер

Тъй като старшите разреди (A9 ÷ A3) са еднакви за всички адреси, само те постъпват на входа на дешифриращата логика, по която се изработва сигнал за избор на INS8250. Това определя т.нар. *базов адрес*. Останалите три адресни шини A2, A1 и A0 постъпват направо в INS8250 и служат за избор на регистър вътре в него.

От табл.3.58 се вижда, че базовият адрес на основния комуникационен асинхронен адаптер е 3F8, а на допълнителния — 2F8. Тъй като разликата в базовите адреси се определя само от A8, на платките винаги се предвижда възможност за избор на основен или допълнителен адаптер чрез съответен превключвател.

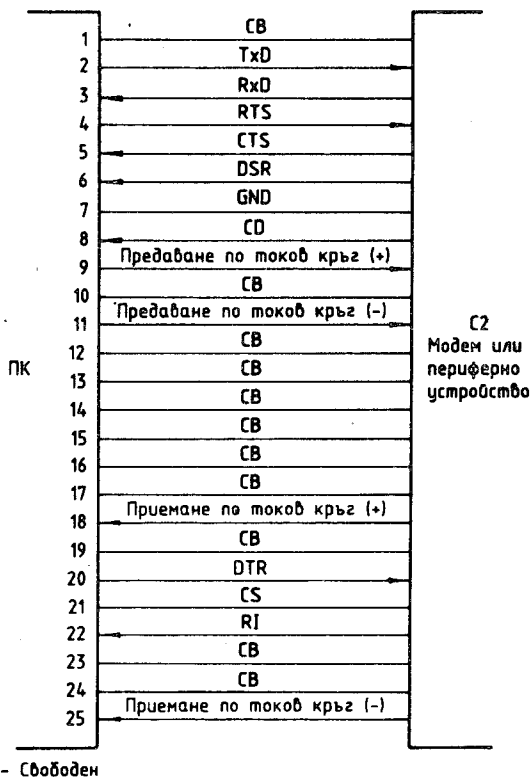
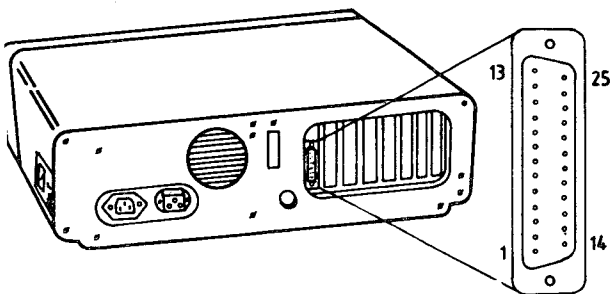
Друга разлика между двата адаптера се състои в това, че за тях има отделени точно определени линии за прекъсване. Основният адаптер винаги трябва да бъде свързан към IRQ4, а допълнителният — към IRQ3. Върху всяка платка има възможност за избор на линия за прекъсване.

Съществуват най-различни реализации на комуникационните асинхронни адаптери. В първите ПК те са реализирани на отделни платки, всяка от които може да бъде основна или допълнителна. Така е в компютрите IBM PC и IBM XT. В следващите модели на ПК все по-силно започна да се чувства недостигът на куплунги за допълнителни модули. Асинхронните адаптери започнаха да се комбинират заедно с други модули върху една и съща платка.

Появиха се т.нар. *многофункционални платки*. Най-често върху такава многофункционална платка е разположена допълнителна памет, таймер и един или два асинхронни адаптера. Така например върху многофункционалната платка на първите модели на Правец-16 се намира 256 Кбайта памет и един асинхронен адаптер. В момента Правец-16 се комплектува с т.нар. *комуникационна платка*, върху която се намира един паралелен порт и два асинхронни адаптера. В компютрите от серията Intelle XT най-разпространена е многофункционалната платка на фирмата Sigma Designs, върху която са разположени два асинхронни адаптера.

На фиг.3.85 са показани видът на куплунга и сигналите на интерфейса на комуникационния асинхронен адаптер.

В ПК може да има до два асинхронни адаптера.



Фиг.3.85. Вид на куплунга и сигналите на интерфейса на комуникационния асинхронен адаптер

3.8.6. СИНХРОННИ АДАПТЕРИ ЗА БИТОВО ОРИЕНТИРАН ПРОТОКОЛ SDLC

Синхронните комуникационни адаптери за обслужване на битово ориентиран протокол SDLC (или накратко SDLC-адаптерите) са изпълнени във вид на отделни печатни платки за ПК. Те са напълно програмируеми и заедно с програмното си осигуряване (драйверните програми) осигуряват синхронен обмен на данни в съответствие с протокола SDLC.

Към основните функции на SDLC-адаптерите могат да се отнесат следните:

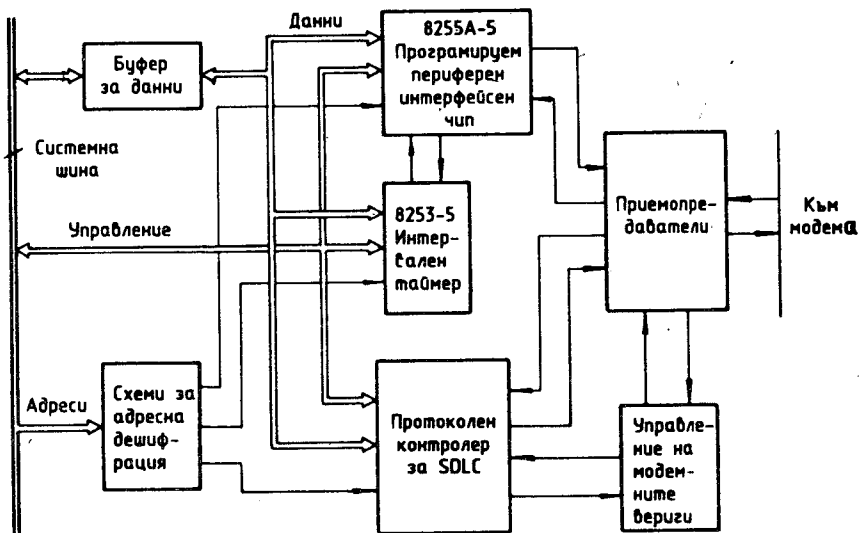
- работа в полудуплексен синхронен режим;
- скорост на предаване до 9600 бит/s;
- автоматично генериране и проверка на контролен код;
- автоматично вмъкване и премахване на нули;
- възможност за обмен чрез директен достъп до паметта;
- осигуряване на обратна връзка за диагностични цели.

На фиг.3.86 е показана структурната схема на SDLC-адаптера. Връзката му към системната шина на ПК се осъществява чрез шините за данни, управляващите шини и част от адресните шини. От другата страна SDLC-адаптерът е свързан към устройството за предаване на данни (модема) чрез приемопредавателите и съответния кудлунг. Основните функции на адаптера се изпълняват от програмируемия протоколен контролер за SDLC I8273. В SDLC-адаптера също така се използва програмируем периферен интерфейс чип I8255A-5, чрез който се осъществява както управлението на част от модемните вериги, така и управлението на другите схеми в адаптера. За генериране на необходимите времеви съотношения в адаптера се използва програмируем интервален таймер I8253-5.

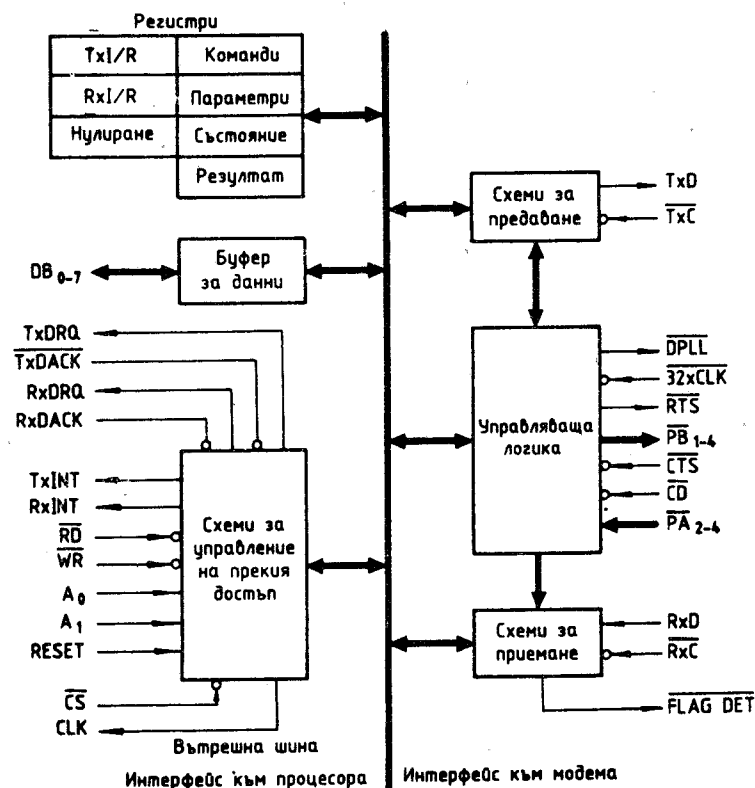
На фиг.3.87 е показана обобщена структурна схема на протоколния контролер SDLC.

Както при предаване или приемане на данни, така и при изпълнение на управляващи функции при работата си контролерът преминава през следните три основни фази: *командна, изпълнителна и фаза на издаване на резултата* (заклучителна фаза). Когато се намира в командната си фаза, контролерът получава от процесора поредица от команди и параметри (ако са необходими). В изпълнителската фаза зададените команди се изпълняват. Това включва установяване на връзката, предаване/приемане на данни, прекратяване на връзката. Тъй като обменът на данни се извършва чрез механизма за пряк достъп, по време на изпълнителната фаза процесорът е напълно свободен за работа по други програми. В заключителната фаза контролерът издава към процесора т.нар. *резултат* — съвкупност от параметри, характеризиращи изпълнението на командата. Работата на контролера във всяка от тези фази се осъществява чрез програмно достъпните регистри и управляващата логика на I8273.

На фиг.3.87 цялата логика на контролера може да бъде обособена в две части: логика, обслужваща интерфейса към процесора, и логика, обслужваща интерфейса към модема. Логиката на процесорния интерфейс се състои от програмно достъпните регистри, буферите за данни и схемите за управление на достъпа до системната шина. Логиката за модемния интерфейс включва схемите за предаване/приемане на данни и управляващи схеми за обслужване на модемните вериги. Схемите за



Фиг.3.86. Структурна схема на SDLC-адаптера



Фиг.3.87. Структурна схема на протоколния контролер SDLC

управление на достъпа до системната шина осигуряват връзката на процесора със седем програмно достъпни регистъра. Адресирането е възможно само при избор на чипа и се извършва чрез сигналите за запис и четене от системната шина IOR и IOW и двата младши бита на адреса A0 и A1. В табл.3.59 е показан начинът на адресиране на регистрите и техните наименования.

Адресиране на регистрите на I8273

Таблица 3.59

CS	IOR	IOW	A1	A0	Регистър
0	1	0	0	0	Команден
0	0	1	0	0	Състояние
0	1	0	1	0	Параметри
0	0	1	1	0	Резултат
0	1	0	0	1	Нулиране
0	0	1	0	1	Прекъсване при предаване (TxI/R)
0	1	0	1	1	(Не се използва)
0	0	1	1	1	Прекъсване при приемане (RxI/R)

Командният регистър служи за задаване на командите за I8273. Това става чрез запис на кода на командата в този регистър.

Регистърът за състояние осигурява издаването на необходимата информация за текущото състояние на I8273 към процесора.

Регистърът за параметри служи за задаване на допълнителни параметри към дадена команда.

Регистърът за резултат съдържа байт, в който е отразен резултатът от непосредствено изпълняваните команди.

Регистърът за прекъсване при предаване съдържа резултата от изпълнение на операцията при предаване. Винаги при възникване на такъв резултат се генерира прекъсване.

Регистърът за прекъсване при приемане съдържа резултата от изпълнение на операциите по приемане. Винаги при възникване на такъв резултат се генерира прекъсване.

Регистърът за нулиране осигурява възможността за програмно нулиране на I8273.

Чрез линиите за прекъсване TxINT и RxINT се издават заявки за прекъсване към процесора. Тези заявки възникват в случаите, когато предаването или приемането на данни се нуждае от обслужване, или когато възниква необходимост от обмен на данни. В първия случай причините за възникване на прекъсването могат да бъдат прочетени от съответните регистри за резултати.

Тъй като състоянието на всяка от тези линии се отразява в байта за състояние, възможна е работа без използване на физическите линии за прекъсване (чрез полиране).

Обменът на данни между I8273 и процесора може да стане или чрез прекъсване, или с директен достъп до паметта (ДДП). В SDLC-адаптера е използван вторият начин. В този случай предаването на данни в даден момент е възможно само в едната посока. За тази цел се използва ДДП-контролерът, който осъществява адресацията и управлението, като той предварително трябва да бъде програмиран (да му бъде зададен началният адрес на блока и неговата големина). SDLC-адаптерът използва ниво 1 на директния достъп.

Заявките за директен достъп до паметта се издават от I8273 по линиите TxDRQ (при предаване) и RxDRQ (при приемане). Чрез схеми ИЛИ тези две заявки са свързани към линията DRQ1 на системната шина. Съответните сигнали за потвърждение са TxDACK и RxDACK. Тези сигнали се формират от сигнала за потвърждение по системната шина DACR1 и условията, при които е възникнала заявката.

Линиите IOR и IOW служат за определяне на посоката на обмена на данни с процесора. При активиране на IOR данните се четат от I8273, а при активиране на IOW данните се записват в I8273. При обмен с директен достъп сигналите по тези линии се издават от ДДП-контролера.

Обменът чрез пряк достъп се извършва в следната последователност. I8273 издава заявка за директен достъп (DRQ1). Контролерът за ДДП получава заявката и при удовлетворяването ѝ изпраща потвърждение DACK1. Следва самият обмен, който се ръководи от контролера за ДДП посредством сигналите IOR или IOW. Обменът на данни протича независимо от състоянието на CS, като по този начин отпада необходимостта от използване на адресируеми входно-изходни регистри за данни.

Схемите за обслужване на модемните вериги съдържат както сигнални вериги с твърдо зададено предназначение, така и вериги, позволяващи дефинирането им от потребителя. Модемните вериги се обслужват от порт А (дефиниран като входен) и порт В (дефиниран като изходен) на I8273.

Предназначение на отделните битове на порт А:

- Бит PA0 — отразява състоянието на сигнала върху извода CTS на I8273. Предаването може да започне едва след активиране на CTS. Ако по време на предаване CTS се установи в неактивно състояние, целият кадър се отхвърля, а процесорът се уведомява чрез прекъсване за това събитие. Типът на прекъсване се отразява в съответния регистър за прекъсване. Бит PA0 е с твърдо зададено предназначение.
- Бит PA1 — отразява състоянието на сигнала върху извода CD на I8273. По време на приемането CD трябва да е в активно състояние. Ако при приемане CD премине в неактивно състояние, възниква прекъсване, като типът му се отразява в съответния регистър за прекъсване. Бит PA1 е с твърдо зададено предназначение.
- Бит PA2 — отразява състоянието на линията DSR.
- Бит PA3 — отразява промяната в състоянието на линията CTS.
- Бит PA4 — отразява промяната в състоянието на линията DSR.
- Битове — не се използват. При четене на порт А състоянието им PA5,PA6,PA7 винаги е 1.

Предназначение на отделните битове на порт В:

- Бит PB0 — задава състоянието на линията RTS. Установява се автоматично от I8273. Бит PB0 е с твърдо зададено предназначение.
- Бит PB1 — не се използва.
- Бит PB2 — задава състоянието на линията DTR.
- Бит PB3 — не се използва.
- Бит PB4 — не се използва.
- Бит PB5 — има същото значение, както и сигналът върху извода за открит флаг (FD) на I8273. Този бит се активира при откриване на флаг от активирания приемник. Бит PB5 е с твърдо зададено предназначение.
- Бит PB6 — не се използва.
- Бит PB7 — не се използва.

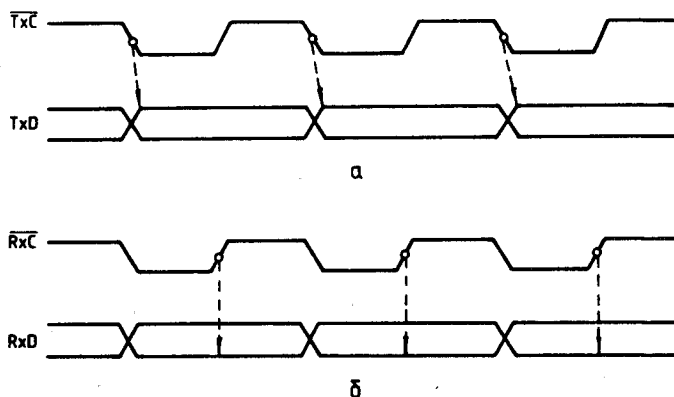
Схемите за последователното предаване и приемане на данни осигуряват правилния обмен на данните с модема. На фиг.3.88а е показан начинът на предаване на данни чрез извода TxD с помощта на тактовата поредица TxC. Всяко ново значение на поредния бит се установява по водещия фронт на тактовата поредица TxC. На фиг.3.88б е показан начинът на стробиране на последователно приеманите данни. Стойността се отчита по време на падащия фронт на тактовата поредица TxC, което е приблизително по средата на времевия интервал за приемане на един бит.

Комуникацията между I8273 и процесора се извършва чрез издаване на команди към I8273 и получаване на резултат от изпълнението на тези команди. При това обменът на управлящата информация става чрез програмно достъпните регистри.

В табл. 3.60 са показани основните команди, използвани при управлението на I8273. Дадени са необходимите параметри за всяка от командите, които се записват в регистъра за параметри, а също и мястото, където се намира резултатът от изпълнението на всяка команда.

От таблицата се вижда, че някои команди не изискват допълнително задаване на параметри. За изпълнението на други е необходим само един параметър, а трети се нуждаят от няколко параметъра, които последователно се записват в регистъра за параметри.

По отношение на резултата командите се делят на незабавно изпълнявани и други, които се изпълняват в течение на някакъв интервал от време (обикновено той зависи от връзката). Незабавно изпълняваните команди завършват или без издаване на резултат, или с издаване на резултат в регистъра за резултат. (Единствените две команди от този тип са командите за четене от порт А (код 22) или порт В (код 23), при което в регистъра на резултата се появява значението на порт А или порт В съответно. В този случай процесорът може непосредствено по адрес да прочете резултата веднага след изпълнение на командите.)



Фиг.3.88. Начин за обмен на данни чрез SDLC-адаптера:

а — предаване; б — приемане

Във всички останали случаи командата започва да се изпълнява, като се формира съответен резултат в един от регистрите TxI/R или RxI/R.

Процесорът може да узнае тези резултати, като прочете съдържанието на тези регистри. Сигнал за завършване на тези операции се издава чрез прекъсването.

Таблица 3.60

Команди за управление на I8273

Наименование на командата	Код на командата	Регистър за резултата	Допълнителни параметри в регистъра за параметри
Установяване на еднобитово закъснение	A4	—	Установяваща маска
Нулиране на еднобитово закъснение	64	—	Нулираща маска
Установяване на режим за предаване	97	—	Установяваща маска
Нулиране на режим за предаване	57	—	Нулираща маска
Установяване на режима за обработка	91	—	Установяваща маска
Нулиране на режима за обработка	51	—	Нулираща маска
Установяване на серийен В/И режим	A0	—	Установяваща маска
Нулиране на серийния В И режим	60	—	Нулираща маска
Общо приемане	C0	RxI, R	Младши и старши байт на дължината на буфера за приемане
Избирателно приемане	C1	RxI, R	Младши и старши байт на дължината на буфера за приемане, адрес (A1 и A0)
Забрана на приемането	C5	—	Младши и старши байт на дължината на предавания кадър, адресното поле, управляващото поле
	C8	TxI, R	
Програмно предаване	C9	TxI, R	Младши и старши байт на дължината на предавания кадър
Отхвърляне на предаван кадър	CC	TxI, R	
Отхвърляне на програмно предаване	CD	TxI, R	
Четене на порт А	22	Резултат	—
Четене на порт В	23	Резултат	—
Установяване на битове в порт В	A3	—	Установяваща маска
Нулиране на битове в порт В	63	—	Нулираща маска

При включване на адаптера е необходимо той да се установи в начално състояние. Това се извършва чрез командите за задаване на режима на работа на I8273. Тези команди предизвикват запис на определени стойности в т.нар. *регистри за задаване на режима*. Такива регистри има общо четири: регистър за режим на операциите, регистър за серийен входно-изходен режим, регистър за режима на предаване и регистър за задаване на режим с един бит закъснение. Тези регистри не са програмно достъпни. Чрез кода на командата се осъществява достъпът до тях, като е възможно установяването в 1, или нулирането на отделни техни битове.

което също така се задава от командата. Всички команди за задаване на режима се нуждаят от един допълнителен параметър. Това е т.нар. *маска*, с която в зависимост от командата могат да се установяват определени битове или да се нулират.

При командите за установяване се извършва логическо ИЛИ между съдържанието на маската и вътрешния регистър за режима. Установяващата маска трябва да съдържа единици във всички разрези, които е необходимо да бъдат установени в регистъра за режима. При 0 в даден разред на маската съответстващият бит в регистъра за режима остава без промяна.

При командите за нулиране се извършва логическо И между съдържанието на маската и регистъра за режима. Там, където маската съдържа 0, съответният бит на регистъра за режима се нулира.

По-долу са описани регистрите за задаване на режима.

Регистър за режим на операциите:

Бит 0 — определя режима на изпращане на флаговете. Ако този бит е в 1, при свободен предавател по линията започват да се издават флагове. Ако в момента на установяване на този бит в 1 предавателят е бил зает, веднага след приключване на предаването започва издаване на флагове. Ако бит 0 е установен в 0, по линията се установява веднага (или след изтичане на предаването, ако такова има) състояние „празно“.

Бит 1 — ако този бит е установен в 1, I8273 ще изпраща два допълнителни символа преди първия флаг на кадъра. Тези символи са 00 в случаите, когато се работи с код NRZI, или 55 в случаите, когато код NRZI не се използва.

Бит 2 — когато този бит е в 1, първите два символа от приетия кадър се буферират от I8273 и не се изпращат в паметта. Ако този бит е в 0, първите два байта не се обработват.

Бит 3 — управлява издаването на прекъсване от I8273 при край на кадъра. Ако бит 3 е в 1, в момента на преминаване на последния байт данни през I8273 се генерира заявка за прекъсване. Ако процесорът в резултат на това прекъсване отговори с нова команда за предаване, и това стане преди постъпването на флага в края на кадъра, то няма да настъпи прекъсване поради откриването на флага за край на кадъра. При това предаването на следващия кадър ще започне непосредствено след приключване на предаването на текущия. По такъв начин става възможно предаването на няколко поредни кадъра, разделени само от по един флаг. При нулиране на бит 3 прекъсване ще се генерира винаги след приемане на флага за край на кадъра.

Бит 4 — не се използва в SDLC-адаптера и винаги трябва да бъде установен в 0.

Бит 5 — При използване на I8273 в SDLC-адаптера в този бит винаги се записва 0. Това позволява адаптерът да разпознава комбинацията от осем последователни единици (01111111) като сигнал за отхвърляне.

Битове 6,7 — не се използват.

Регистър за последователен входно-изходен режим

Бит 0 — ако в този бит има записана 1, това означава работя с код NRZI. Нула в този бит означава, че потокът от данни се разглежда като нормално двоично кодиран.

Бит 1 — когато този бит е установен в 1, тактовата честота за предаване се подава към схемата за приемане. При 0 в този бит схемите за тактови честоти на предавателя и приемника са свързани към съответните изводи на I8273. Бит 1 се използва за диагностични цели.

Бит 2 — когато този бит е в 1, изходът от предаваните данни се свързва вътре в чипа към входа за приемани данни. При 0 в този бит схемите за предаване и приемане се свързват към съответните изводи на I8273. Бит 2 се използва за диагностични цели.

Битове 3,4,5,6,7 — не се използват.

Регистър за режима на предаване

Бит 0 — ако този бит е в 1, то при приемане на поредния байт данни или при готовност за предаване 18273 ще генерира прекъсване. Този случай се идентифицира по факта, че при него в регистъра за състояние отсъства информация за резултата от прекъсването (в TxI/R или RxI/R). Ако този бит е 0, обменът на данни става чрез ДДП без издаване на прекъсване към процесора.

Битове 1,2,3,4,5,6,7 — не се използват.

Регистър за задаване на режим с един бит закъснение

Битове 0,1,2,3,4,5,6 — не се използват.

Бит 7 — ако този бит се установи в 1, това определя такъв режим на работа на 18273, че входният поток данни се препредава с един бит закъснение. Нула в бит 7 забранява този режим.

В табл.3.61 са дадени всички команди за задаване на режима.

Команди за задаване на режима на 18273

Таблица 3.61

Код на командата	Тип на командата	Регистър	Параметър
A4	Установяване	Режим на предаване с един бит закъснение	Установяваща маска
64	Нулиране		Нулираща маска
97	Установяване	Режим на предаване	Установяваща маска
57	Нулиране		Нулираща маска
91	Установяване	Режим на операции	Установяваща маска
51	Нулиране		Нулираща маска
A0	Установяване	Режим на сериен В/И	Установяваща маска
60	Нулиране		Нулираща маска

След задаването на режима работата на програмируемия контролер 18273 може да продължи чрез изпълнение на поредната команда. В зависимост от типа на командата (вж. табл.3.60) резултат или не се очаква, или той се намира в регистъра за резултат (при незабавно изпълнявана команда), или пък командата предизвиква прекъсване, след което процесорът може да прочете резултата в един от регистрите TxI/R и RxI/R. В табл.3.62 са показани възможните кодове на резултата в TxI/R, а в табл.3.63 — възможните кодове на резултата в RxI/R.

Таблица 3.62

Кодове на резултата в регистър TxI/R

Код	Резултат	Състояние след прекъсването
0C	Предварително прекъсване при предаване	Активен предавател
0D	Завършено предаване на кадър	Състояние „празно“ или флаг
0E	Грешка при обмен с ДДП	Отхвърляне
0F	Грешка в линията CTS	Отхвърляне
10	Отхвърлен кадър	Състояние „празно“ или флаг

Знакът × в табл.3.63 е шестнадесетично число и може да приема определени фиксирани стойности. Тъй като при протокола SDLC се допуска произволна дължина на кадъра, която не винаги е кратна на цяло

число байтове, чрез старшите битове на първите два резултата, означени със знака ×, се указва броят на валидните битове в последния приет байт. Съответствието между × и броя на действителните битове е следното:

×	Брой приети битове в последния байт
E	Всичките 8 бита
0	Само бит 0
8	Бит 1 — бит 0
4	Бит 2 — бит 0
C	Бит 3 — бит 0
2	Бит 4 — бит 0
A	Бит 5 — бит 0
6	Бит 6 — бит 0

Таблица 3.63

Кодове на резултата в регистър Tx1/R

Код	Резултат	Състояние след прекъсването
× 0	A1 съвпада, или общо приемане	Активен приемник
× 1	A2 съвпада	Активен приемник
03	CRC-грешка	Активен приемник
04	Наличие на отхвърлен кадър	Активен приемник
05	Открито състояние „празно“	Забрана
06	Открит управляващ символ EOF	Забрана
07	Кадърът е по-къс от 32 бита	Активен приемник
08	Грешка при обмен с ДДП	Забрана
09	Препълнен буфер в паметта	Забрана
0A	Грешка по линията CD	Забрана
0B	Повторен запис след прекъсването при приемане	Забрана

Освен чрез регистрите за резултата процесорът може в произволен момент от време да получи информация за състоянието на I8273, като прочете регистъра за състояние.

Предназначение на битовете в регистъра за състояние:

- Бит 0 — показва, че в Tx1/R се намира актуална информация. Установява в 1, когато I8273 записва код на резултата в Tx1/R, а при четенето на този регистър от процесора се нулира.
- Бит 1 — показва, че в Rx1/R се намира актуална информация. Установява се в 1, когато I8273 записва код на резултата в Rx1/R, а при четенето на този регистър от процесора се нулира.
- Бит 2 — отразява състоянието на извода Tx1/R на I8273. Установява се в 1 при издаване на заявка за прекъсване от преподавателя на I8273 и се нулира или при прочитане на резултата от процесора, или при запис на поредния подлежащ на предаване байт данни.
- Бит 3 — отразява състоянието на извода Rx1/R на I8273. Установява се в 1 при издаване на заявка за прекъсване от приемника на I8273 и се нулира или при прочитане на резултата от процесора, или при прочитане на поредния постъпил в приемника байт данни.
- Бит 4 — показва дали има резултат в регистъра за резултати при изпълнение на незабавно изпълнявани операции. Той се установява в 1 при появата от такъв резултат в регистъра за резултати и се нулира при прочитане на резултата от процесора или при започване на поредния обмен на данни.
- Бит 5 — показва, дали регистърът за параметри е запълнен или не. Той се установява

в 1 при запис на параметър от страна на процесора и се нулира при възприемане на този параметър от I8273.

Бит 6 — показва състоянието на регистъра за команди. Ако този бит е в 1, там се намира команда и следваща команда не може да бъде приета. При стойност 0 следващата команда може да бъде приета.

Бит 7 — показва дали I8273 се намира в командна или друга фаза. Установява се в 1 при запис на команда от страна на процесора, който по този начин стартира изпълнението на командната фаза. Нулира се, когато и последният от необходимите параметри за дадената команда е записан и приет от I8273, с което завършва командната фаза.

Описаната структура на програмируемия контролер I8273 позволява да се разгледа накратко неговата работа при преминаването му през различните фази.

Командна фаза

Тъй като в I8273 има единствен команден регистър, в даден момент е възможно изпълнението на една-единствена команда. Командната фаза протича в една последователност. Записва се кодът на командата в командния регистър, след което в регистъра на параметрите се записват параметри, ако това е необходимо за дадената команда (вж. табл.3.60). В действителност задаването на командата е по-сложно поради необходимостта от проверка дали I8273 е възприел поредния запис. С отчитането на този флаг командната фаза протича по следния начин: чете се регистърът за състояние и се проверява дали е свободен командният регистър (бит 6 трябва да бъде 0). Ако е зает, изчаква се докато се освободи, тъй като запис на команда върху съществуваща в командния регистър би я разрушило. След задаването на командата започва запис на параметрите (ако има такива към нея). При това преди всеки пореден запис се проверява дали I8273 е възприел предишния параметър (бит 5 на регистъра за състояние трябва да бъде 0). Ако предишният параметър не е приет, изчаква се докато регистърът за параметри не се освободи. Записът на нов параметър върху предишния води до разрушаването му. Този процес продължава, докато не бъдат записани всички необходими за дадена команда параметри, с което командната фаза приключва.

Изпълнителна фаза

По време на изпълнителната фаза I8273 извършва операциите, предписани от въведената команда. Ако има обмен на данни чрез директен достъп, процесорът не участва в изпълнението на командата. Ако е зададена команда за обмен и се работи с прекъсвания, при възникване на необходимите условия тези прекъсвания се генерират (прекъсване при предаване или при приемане). При възникване на прекъсване процесорът трябва да определи дали то е породено от необходимост за обмен или означава готов резултат в някой от регистрите за резултати. Това става чрез прочитане на бит 0 (или бит 1) на регистъра за състояние. Ако съответният бит е в 1, това означава, че поредната команда е изпълнена и резултатът от изпълнението ѝ е готов. Ако този бит е в състояние 0, прекъсването е породено от необходимост за обмен на информация от страна на I8273.

Фаза на издаване на резултата (заклучителна фаза)

В тази фаза се навлиза или при възникване на грешка, или при успешно завършване на операцията. В заключителната фаза резултатът от изпълнението на зададената команда трябва да бъде предоставен на процесора.

При незабавно изпълняването команди резултатът веднага се появява в регистъра за резултати и това се индицира с наличието на 1 в бит 4 на регистъра за състоянието.

При другите команди (напр. команди за предаване и приемане) резултатът се появява след известно време в един от регистрите Txl/R и RxI/R. Освен прекъсването индикация за наличен резултат може да бъде единица в бит 0 или бит 1 на регистъра за състоянието. Резултатът представлява един или няколко байта, съдържащи код на резултата, в съответствие с показаното в табл.3.62 и 3.63.

Операциите по предаване и приемане на кадър от програмируемия контролер при използване на ДДП протичат по следния начин.

Предаване. Предаването на кадъра започва с установяване на контролера за ДДП (задава се началният адрес на кадъра). Следва запис на команда за предаване в I8273. При предаване на кадър командата изисква допълнителното задаване на следните параметри (вж. табл.3.68): адресно поле на кадъра, управляващо поле на кадъра и два байта, които съдържат младшата и старшата част на числото, показващо дължината на полето за данни (информационното поле) в кадъра. След зареждане на тези параметри I8273 издава към модема сигнал RTS и очаква постъпването на ответния сигнал CTS (към този момент вече връзката с модема е установена, т.е. в отговор на DTR е получен сигнал DSR). След получаване на сигнала CTS I8273 започва предаване на кадъра. Отначало се издава начален флаг, байтовете на адресното поле, байтовете на управляващото поле. След това започват да се издават заявки за пряк достъп за предаване на съдържанието на информационното поле. Байт по байт данните се предават до момента на изчерпване на брояча, в който е била зададена дължината на информационното поле. След това се прекратява издаването на заявки за директен достъп, предава се контролният код и крайният флаг и се генерира заявка за прекъсване при предаване, което уведомява процесора, че предаването е завършило и резултатът от изпълнението на командата е налице.

Общо приемане. При приемане работата протича по аналогичен начин. Отначало се установява контролерът за ДДП, като му се задава началният адрес от паметта, където ще бъде зареден приетият кадър. Следва издаване на команда към I8273. За разлика от предаването при приемане на кадъра са възможни два случая (две различни команди): общо приемане, при което всички постъпили на входа кадри се приемат и записват в паметта, и избирателно приемане, при което I8273 приема само тези съобщения, адресното поле на които съвпада с едно или две адресни полета, предварително зададени като параметри на I8273.

Командата за общо приемане изисква допълнителен параметър: *дължината на приемния буфер*. Тя се съдържа в два последователни байта. След задаването на дължината, приемникът се активира и следи постъпващите кадри. При появата на кадър започва издаването на заявки за директен достъп за всеки байт, което продължава до изчерпване на ин-

формационното поле на приемания кадър. След приемане на крайния флейт се проверява контролният код и се генерира заявка за прекъсване към процесора, която го информира, че изпълнението на командата е завършило. Както при предаването и тук всички действия по приемането на кадъра се извършват без намеса от страна на процесора.

Избирателно приемане. При избирателното приемане командата изисква допълнително задаване на два параметъра A1 и A2. Това са два байта, като при съвпадение на който и да е от тях с адресното поле на кадъра на входа на приемника става неговото приемане и записване в паметта. Избирателното приемане се използва обикновено от подчинените станции, за които A1 е адресът на подчинената станция, а A2 е адресът за общодостъпно предаване. Ако е необходимо да бъдат приемани само кадрите, адресното поле на които има едно-единствено значение, командата се задава при еднакви параметри A1 и A2. Изпълнението на командата в останалата си част протича и завършва както в предишния случай.

SDLC-адаптерът изпълнява своите функции в тясно взаимодействие с останалите програмируеми големи интегрални схеми I8255A-5 и I8253-5.

Програмируемата периферна интерфейсна схема I8255A-5 служи както за буфериране на част от модемните сигнали в двете посоки, така и за издаване на управляващи сигнали към останалата логика. I8255A-5 съдържа три порта A, B и C. При инициализацията те се програмират по следния начин: порт A — входен порт; порт B — изходен порт; порт C — четири входни и четири изходни линии.

Предназначение на отделните битове на порт A:

- Бит A0 — нула означава, че сигналът RI по линията е в активно състояние.
- Бит A1 — нула означава, че сигналът CD по линията е в активно състояние.
- Бит A2 — поява на сигнал означава наличие на тактова честота за предаване.
- Бит A3 — нула означава, че CTS по линията е в активно състояние.
- Бит A4 — поява на сигнал означава наличие на тактова честота за приемане.
- Бит A5 — единица означава, че е настъпила промяна в състоянието на модема.
- Бит A6 — единица означава, че изходът на таймер 2 е в активно състояние.
- Бит A7 — единица означава, че изходът на таймер 1 е в активно състояние.

Предназначение на отделните битове на порт B:

- Бит B0 — нула предизвиква включване на сигнал за избор на скоростта на предаване към модема.
- Бит B1 — нула предизвиква включване на сигнал за спомагателен избор към модема (нестандартен сигнал).
- Бит B2 — нула предизвиква свързване на вериите за тестване.
- Бит B3 — единица предизвиква нулиране на схемата, следяща за изменение в състоянието на модема.
- Бит B4 — единица предизвиква нулиране на I8273.
- Бит B5 — единица предизвиква стартиране на таймер 2.
- Бит B6 — единица предизвиква стартиране на таймер 1.
- Бит B7 — единица разрешава прекъсването на ниво 4.

Битове от 0 до 3 на порт C са програмирани като изходни, от 4 до 7 — като входни.

Предназначение на отделните битове на порт C:

- Бит C0 — единица разрешава вътрешна синхронизация.
- Бит C1 — единица разрешава външна синхронизация.
- Бит C2 — единица предизвиква затваряне на обратната връзка при диагностика.

- Бит С3 — нула разрешава прекъсвания 3 и 4.
- Бит С4 — наличието на променлив сигнал означава приемане на данни.
- Бит С5 — отразява състоянието на изхода на таймер 0.
- Бит С6 — нула означава активирани схеми за тестване.
- Бит С7 — не се използва.

Програмируемият интервален таймер I8253-5 се използва както за генериране на тактови поредици, така и за задаване на времеви интервали. Работата на I8253-5 се задава от тактова честота, два пъти по-ниска от тази на процесора. Отделните **изходи на таймера** са програмирани по следния начин:

- таймер 0 — програмиран е да генерира честота, която е входна за таймер 2. Изходът му се следи чрез бит С5 на порт С.
- таймер 1 — програмиран е да генерира времеви интервали. Изходът му е свързан към линията за прекъсване на 4-то ниво. Моментът на отчитане на времевия интервал се управлява чрез бит А7 на порт А. Входната му честота е същата както на таймер 0.
- таймер 2 — програмиран е да генерира времеви интервали. Изходът му е свързан към линията за прекъсване на 4-то ниво. Моментът на отчитане на времевия интервал се управлява чрез бит А6 на порт А.

Механизмът на работа с таймерите е следният. За изпълнението на всяка команда е предвидено определено време. В съответния таймер (1 или 2) се задава времеви интервал, който е по-голям от очаквания. В случай че операцията не завърши своевременно, изходният сигнал от таймера ще генерира прекъсване. Ако операцията завърши нормално, тя ще предизвика прекъсване преди таймера, а обслужващата прекъсването програма ще го нулира. Таймер-0 служи единствено като делител на честотата.

Преди започване на работа е необходимо интервалният таймер да се програмира, т.е да бъде зададен режимът му на работа. Това става чрез т.нар. *управляваща дума*.

Предназначение на отделните битове на **управляващата дума**:

Бит 0 — задава типа на брояча. При 0 броячът е 16-битов, двоичен, а при 1 броячът е двоично-десетичен (четири декади).

Битове 1,2,3 — задават режима по следния начин:

Бит 3	Бит 2	Бит 1	Режим
0	0	0	режим 0
0	0	1	режим 1
×	1	0	режим 2
×	1	1	режим 3
1	0	0	режим 4
1	0	1	режим 5

Битове 4,5 — показват начина на четене/запис на младшия и старшия байт на числото, задавашо времеви интервал:

Бит 4	Бит 5	Режим
0	0	Работа в режим на броене
1	0	Четене/запис на старшия байт
0	1	Четене/запис на младшия байт
1	1	Четене/запис на младшия байт след което и на старшия байт

Битове 6,7 — битове за избор на таймера:

Бит 6	Бит 7	Таймер
0	0	Избор на таймер 0
0	1	Избор на таймер 1
1	0	Избор на таймер 2
1	1	Невалидна комбинация

В табл. 3.64 е показана обобщена информация за всички входно-изходни адреси на програмно достъпните регистри в SDLC-адаптера.

В табл. 3.65 са показани типовете прекъсвания, които възникват в SDLC-адаптера и линиите, по които те се предават, а също така случаите при заявки за директен достъп.

Тук е разгледана типична последователност при инициализация на SDLC-адаптера.

Инициализацията на SDLC-адаптера започва с установяване на програмируемите чипове. Програмируемият интерфейсен чип I8255A-5 се конфигурира, като на адрес 383 се записва кодът 98. Той дефинира порт А като входен, порт В — като изходен, битове 0 ÷ 3 на порт С — като изходни, и битове 4 ÷ 7 на порт С — като входни.

Таблица 3.64

Входно-изходни регистри на SDLC-адаптера

Входно-изходен адрес	Чип	Регистър	Функция
380	I8255A-5	Порт А	Следене на модемните линии
381	I8255A-5	Порт В	Управление на модемните линии
382	I8255A-5	Порт С	Управление на SDLC-адаптера
383	I8255A-5	Установяване на режима	Инициализация
384	I8253-5	Таймер 0 (младши байт)	Генератор на тактова честота
384	I8253-5	Таймер 0 (старши байт)	Генератор на тактова честота
385	I8253-5	Таймер 1 (младши байт)	Задаване на таймаут
385	I8253-5	Таймер 1 (старши байт)	Задаване на таймаут
386	I8253-5	Таймер 2 (младши байт)	Задаване на таймаут
386	I8253-5	Таймер 2 (старши байт)	Задаване на таймаут
387	I8253-5	Установяване на режима	Инициализация
388	I8273	Команда/състояние	Команда (запис)
389	I8273	Параметър/резултат	Параметър (запис)
38А	I8273	Резултат при прекъсване при предаване/нулиране	Нулиране (запис)
38В	I8273	Резултат при прекъсване при приемане	TxI/R (четене)
38С	I8273	Данни	RxI/R (четене)
			Само при ДДП

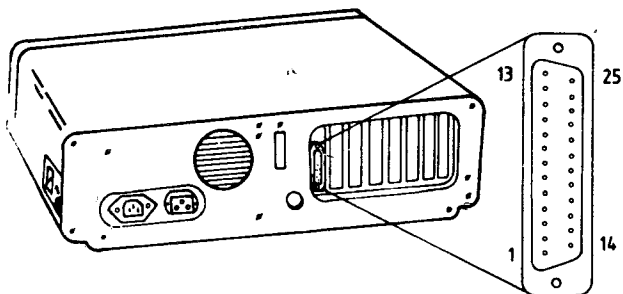
Типове прекъсвания в SDLC-адаптера

Таблица 3.65

Заявка за прекъсване или прък достъп	Причина
Прекъсване на ниво 3	Прекъсване при приемане/предаване на данни
Прекъсване на ниво 4	Прекъсване от таймер 1
	Прекъсване от таймер 2
	Прекъсване при промяна на CTS
	Прекъсване при промяна на DSR
Заявка за ДДП на ниво 1	Приемане/предаване на данни

След това в порт С се записва необходимата информация, забраняваща прекъсванията, като се установява режим на тестване и се разрешава вътрешната синхронизация. Това се постига чрез запис по адрес 382 на кода 0D. По такъв начин SDLC-адаптерът се изолира от външните линии.

Следва нулиране на I8273, което се постига чрез запис на 1 и последваща 0 в бит 4 на порт В, т.е. по адрес 381. Режимът на работа на таймера се задава, като се избира адресът на управляващата го дума (383) и за всеки от броячите последователно се определя желаният режим. Таймер 0 се програмира като брояч, а таймери 1 и 2 — като генератори на времеви интервали. В броячите на таймери 1 и 2 (по адреси 385 и 386) се зареждат необходимите стойности, които определят



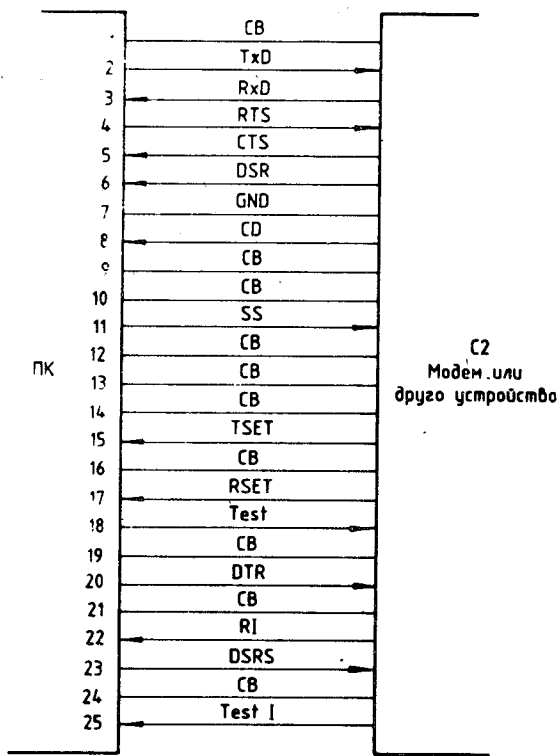
желания таймаут при генериране на заявка по линия за прекъсване по 4-то ниво. Тези прекъсвания възникват, когато предварително определеното време е изтекло, без да настъпи очакваният резултат. Стойностите на времевите интервали могат да бъдат заредени в произволен момент, тъй като таймерите се стартират програмно чрез битове 6 и 5 на порт В (адрес 381).

След като спомагателните чипове са програмирани и I8273 е нулиран, може да започне неговата работа чрез задаване на команди за определяне на режимите му, както бе описано по-горе.

При реализацията на SDLC-адаптерите разновидностите са малко. Обикновено адаптерът е изпълнен във вид на самостоятелна печатна платка (IBM PC/XT, Правец-16).

На фиг.3.89 е показан видът на куплунга и сигналите на интерфейса на SDLC-адаптера.

В ПК може да има само един SDLC-адаптер.



CB - Свободен

Фиг.3.89. Разположение на сигналите на интерфейса на SDLC-адаптера

3.8.7. СИНХРОННИ АДАПТЕРИ ЗА БАЙТОВО ОРИЕНТИРАН ПРОТОКОЛ BSC

Синхронните комуникационни адаптери за обслужване на байтово-ориентиран протокол BSC (или накратко BSC-адаптерите) обикновено са разположени върху отделни печатни платки за ПК. Както и другите комуникационни адаптери, те са напълно програмируеми, като функционирането им се обслужва от специализирано програмно осигуряване (драйверни програми), с което се реализира синхронен байтово ориентиран протокол BSC.

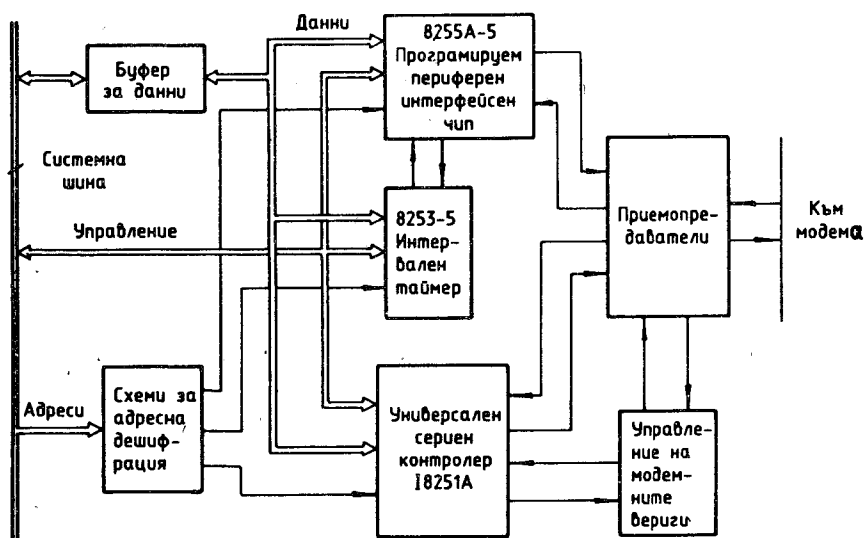
Към основните функции на BSC-адаптера могат да се отнесат:

- скорост на предаване до 9600 бит/s;
- автоматично издаване на синхросимволи;
- автоматично генериране и проверка на контролен код;
- автоматично вмъкване на управляващи символи (DLE);
- осигуряване на обратна връзка за диагностични цели.

На фиг.3.90 е показана структурната схема на BSC-адаптера. Връзката му към системната шина на ПК се осъществява чрез шините за данни, управляващите шини и част от адресните шини. От другата страна BSC-адаптерът е свързан към модема чрез приемопредаватели и стандартен куплунг за C2.

Основните функции на BSC-адаптера се изпълняват от универсалния синхронно/асинхронен приемопредавател I8251A. Управлението на част от модемните вериги, а също така на логиката на адаптера се извършва от процесора посредством програмируемия периферен интерфейс чип I8255A-5, а за задаване на необходимите времеви съотношения се използва програмируемият интервален таймер I8253-5.

За разлика от другите комуникационни адаптери BSC-адаптерът има изключително опростена структура. За реализиране на протокола



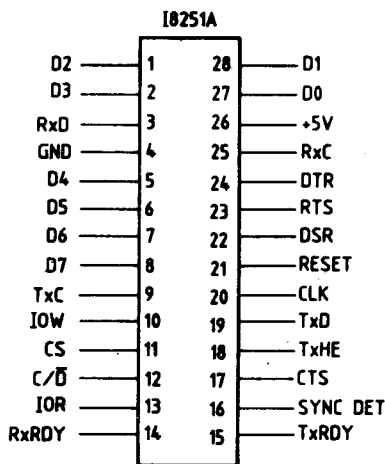
Фиг.3.90. Структурна схема на BSC-адаптера

е използван универсален чип, което е довело до прехвърлянето на голяма част от функциите на адаптера върху драйверните програми. Обменът на данни се извършва чрез системата за прекъсване, без да се използва директен достъп. Това опростява схемата на адаптера, но натоварва допълнително процесора с функции по обмена на данни.

На фиг.3.91 е показан I8251A.

Предназначение на отделните изводи:

- 27,28,1,2,5,6,7,8 — Шини за данни (D0÷D7). Двупосочни шини за данни с три състояния. Използват се за предаване и присмане на данни, команди и състояние.
- 11 — CS. Избор на чипа.
- 13 — IOR. Четене от I8251A в паметта.
- 10 — IOW. Запис от паметта в I8251A.
- 12 — C/D. Сигнал, показващ дали се извършва запис/четене на данни или на управляваща информация.
- 20 — CLK. Вход за задаване на тактовата честота на I8251A. В BSC-адаптера на този вход постъпва тактовата честота на процесора.
- 9 — TxC. Вход за тактова честота за стробиране на последователно предаваните данни. В BSC-адаптера тази честота постъпва от изхода на таймера.
- 25 — RxC. Вход за тактовата честота за стробиране на последователно приеманите данни. Обикновено тази честота е еднаква с TxC.
- 17 — CTS. Вход, на който постъпва сигнал от модема CTS.
- 22 — DSR. Вход, на който постъпва сигнал от модема DSR.
- 3 — RxD. Вход за приемане на данни.
- 21 — RESET. Вход за нулиране на I8251A.
- 19 — TxD. Изход за предаване на данни.
- 24 — DTR. Изход за издаване на сигнал DTR към модема.
- 23 — RTS. Изход за издаване на сигнал RTS към модема.
- 14 — RxDY. Сигнал, показващ, че е приет поредният байт данни. В BSC-адаптера при появата на този сигнал се генерира заявка за прекъсване.
- 15 — TxDY. Сигнал, показващ, че буферът за приемане на байта данни от процесора е празен и поредният байт данни може да бъде приет. В BSC-адаптера при появата на този сигнал се генерира заявка за прекъсване.
- 4 — GND. Захранване 0 V (земля).
- 26 — +5 V. Захранване +5 V.



Фиг.3.91. Универсален асинхронно-синхронен чип I8251A

За програмното управление на I8251A в него са предвидени четири програмно достъпни регистъра. Изборът на регистър става чрез сигналите CS, IOR, IOW и C/D.

В табл.3.66 е показан начинът на адресиране на тези регистри.

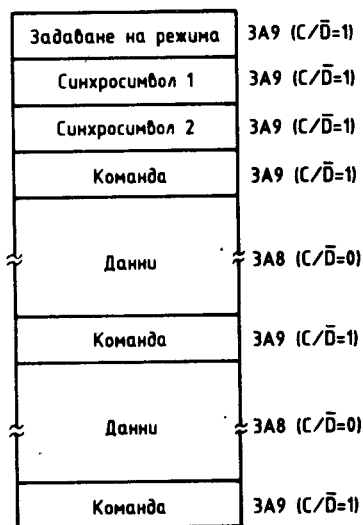
В BSC-адаптера към извода C/D на I8251A е свързана шината A0 (A0 — младшия бит на адреса). По този начин винаги по четния адрес се извършва обмен на данни (запис, четене), а по нечетния — обмен на управляваща информация (команда, състояние).

Първоначалното програмиране на I8251A (инициализацията му) се извършва чрез задаване на поредица от команди, които определят режима на работа на I8251A, а също така предписват по-нататъшните му действия. След нулиране на чипа новата команда се възприема като инструкция за задава-

не режима на работа на I8251A. След като е зададен режимът на обслужване на BSC-протокол, необходимо е да се заредят двата синхросимвола. Всички управляващи думи, които се записват след това ($C/\bar{D}=1$), се възприемат от I8251A като команди. След задаване на командите може да започне обменът на данни ($C/\bar{D}=0$). Команди могат също така да бъдат записвани в произволен момент по време на обмен на данни. На фиг.3.92 е показана схематично типична последователност при работа с I8251A.

Ако възникне необходимост от смяна на режима, това се извършва чрез нулирането на I8251A (чрез издаване на команда за нулиране), след което първата команда се интерпретира като команда за задаване на режима.

Управлението на BSC-адаптера се извършва, като се използва програмируемият периферен интерфейс чип I8255A-5. При инициализация на BSC-адаптера I8255A-5 се програмира така, че порт А да бъде входен порт, порт В — изходен порт, битове $0 \div 3$ на порт С — изходни, а битове $4 \div 7$ на порт С — входни.



Фиг.3.92. Типична последователност при работа с I8251A

Таблица 3.66

Адресиране на регистрите на I8251A

CS	IOR	IOW	C/D	Регистър
0	1	0	0	Предавани данни
0	0	1	0	Приемани данни
0	1	0	1	Команда
0	0	1	1	Състояние

Предназначение на отделните битове на порт А:

- Бит PA0 — нула означава, че RI по линията е в активно състояние.
- Бит PA1 — нула означава, че CD по линията е в активно състояние.
- Бит PA2 — появата на сигнал означава наличие на тактовата честота за предаване.
- Бит PA3 — нула означава, че CTS по линията е в активно състояние.
- Бит PA4 — появата на сигнал означава наличие на тактовата честота за приемане.
- Бит PA5 — единица означава, че сигналът TxRDY на I8251A е активиран.
- Бит PA6 — единица означава, че изходът на таймер 2 е в активно състояние.
- Бит PA7 — единица означава, че изходът на таймер 1 е в активно състояние.

Предназначение на отделните битове на порт В:

- Бит BV0 — нула предизвиква включване на сигнал за избор на скоростта на предаване към модема.
- Бит BV1 — нула предизвиква включване на сигнал за спомагателен избор към модема (нестандартизиран сигнал).
- Бит BV2 — нула предизвиква свързване на веригата за тестване.
- Бит BV3 — не се използва.

- Бит PB4 — единица предизвиква нулиране на I8251A.
- Бит PB5 — единица предизвиква стартиране на таймер 2.
- Бит PB6 — единица предизвиква стартиране на таймер 1.
- Бит PB7 — единица разрешава свързването на изходите на таймер 1 и таймер 2 към линията за прекъсване на ниво 4.

Битовете от 0 до 3 на порт С са програмирани като изходни, а битовете от 4 до 7 — като входни.

Предназначение на отделните битове на порт С:

- Бит PC0 — единица разрешава вътрешна синхронизация.
- Бит PC1 — единица разрешава външна синхронизация.
- Бит PC2 — единица предизвиква затваряне на обратната връзка при диагностика.
- Бит PC3 — нула разрешава прекъсванията на нива 3 и 4.
- Бит PC4 — наличието на променлив сигнал означава приемане на данни.
- Бит PC5 — отразява състоянието на изхода на таймер 0.
- Бит PC6 — не се използва.
- Бит PC7 — не се използва.

Програмируемият интервален таймер I8253-5 се използва както за генериране на тактови поредици, така и за задаване на времеви интервали. Отделните изходи на таймера са програмирани по следния начин:

- таймер 0 — програмиран е да генерира честота, която може да се използва за стробирание на предаването или приемани данни (постъпва на входовете TxC и RxC на I8251A).
- таймер 1 — програмиран е да генерира времеви интервали. Изходът му е свързан към линията за прекъсване на 4-то ниво. Моментът на отчитане на времевия интервал се управлява чрез бит A7 на порт А.
- таймер 2 — програмиран е да генерира времеви интервали. Изходът му е свързан към линията за прекъсване на 4-то ниво. Моментът на отчитане на времевия интервал се управлява чрез бит A6 на порт А.

На входовете и на трите таймера се подава тактовата честота на процесора.

На практика таймер 0 се използва само като делител на честота. Механизмът на работа с таймер 1 и таймер 2 е следният: при стартиране на дадена команда се пуска съответният таймер, в който предварително е зададен желаният времеви интервал. Ако изпълняваната команда не завърши в рамките на този интервал, генерира се заявка за прекъсване (ниво 4). Ако операцията завърши навреме, то в резултат от нейното изпълнение ще възникне прекъсване, при обслужването на което таймерът се дезактивира.

Начинът за избиране на брояч вътре в таймера и задаването на режима на всеки от изходите са описани в т. 3.8.6.

В табл.3.67 е показана обобщена информация за всички входно-изходни регистри на BSC-адаптера.

В табл. 3.68 е показана обобщена информация за типовете прекъсвания, които възникват в BSC-адаптера и линиите, по които те се издават.

Следва пример за типична последователност при инициализация на BSC-адаптера.

Инициализацията на BSC-адаптера започва с установяването на програмируемите чипове. Най-напред се задава режимът на работа на програмируемия интерфейс чип I8255A-5. Това става, като по адрес 3A3 се записва код 78, който дефинира порт А като входен, порт В — като изходен, битове 0 ÷ 3 на порт С — като изходни, и битове 4 ÷ 7 на порт С — като входни. След това в порт С се записва необходимата

информация, забраняваща прекъсванията, като се установява режим на тестване и се разрешава вътрешната синхронизация. Това се постига чрез запис по адрес 3A2 на кода 0D. По такъв начин BSC-адаптерът се изолира от външните комуникационни линии.

Таблица 3.67

Входно-изходни регистри на BSC-адаптера

Входно-изходен адрес	Чип	Регистър	Функция
3A0	I8255A-5	Порт А	Следене на модемните линии
3A1	I8255A-5	Порт В	Управление на модемните линии
3A2	I8255A-5	Порт С	Управление на BSC-адаптера
3A3	I8255A-5	Установяване на режима	Инициализация
3A4	I8253-5	Таймер 0 (младши байт)	Генератор на тактова честота
3A4	I8253-5	Таймер 0 (старши байт)	Генератор на тактова честота
3A5	I8253-5	Таймер 1 (младши байт)	Задаване на таймаут
3A5	I8253-5	Таймер 1 (старши байт)	Задаване на таймаут
3A6	I8253-5	Таймер 2 (младши байт)	Задаване на таймаут
3A6	I8253-5	Таймер 2 (старши байт)	Задаване на таймаут
3A7	I8253-5	Установяване на режима	Инициализация
3A8	I8251A	Предавани данни	Предавани данни (запис)
3A8	I8251A	Приемани данни	Приемани данни (четене)
3A9	I8251A	Команда	Задаване на команда (запис)
3A9	I8251A	Състояние	Прочитане на състояние (четене)

Таблица 3.68

Типове прекъсвания в BSC-адаптера

Заявка за прекъсване	Причина
Прекъсване на ниво 3	Прекъсване при приемане на данни
Прекъсване на ниво 4	Прекъсване при предаване на данни
	Прекъсване от таймер 1
	Прекъсване от таймер 2

След това се нулира I8251A чрез запис на 1 и последваща 0 в бит 4 на порт В, т.е. по адрес 3A1. Режимът на работа на таймера се задава чрез адреса на управляващата го дума (3A3), като за всеки от броячите последователно се определя желаният режим. Таймер 0 се програмира като брояч, а таймери 1 и 2 — като генератори на времеви интервали. В броячите на таймери 1 и 2 (по адреси 3A5 и 3A6) се зареждат необходимите стойности, които ще определят нужните времеви интервали при генериране на заявка по линия на прекъсване на ниво 4. Тези прекъсвания ще възникват, когато предварително определеното време е изтекло, без да настъпи очакваният резултат. Стойностите на времевите интервали могат да бъдат заредени в произволен момент, тъй като стартирането на таймерите се извършва програмно чрез битове 6 и 5 на порт В (адрес 3A1).

След установяване на спомагателните чипове се задава режимът на работа на I8251A. След това се записват синхробайтовете (чрез команд-

ния регистър) и се задава поредицата от команди за изпълнение на желаните операции.

BSC-адаптерите обикновено се реализират върху една самостоятелна печатна платка (например ПК IBM PC/XT и Правец-16).

Сигналите на интерфейса на BSC-адаптера и куплунгът имат вида, показан на фиг. 3.85.

В един персонален компютър може да има само един BSC-адаптер.

3.9. ОСОБЕНОСТИ НА ВИСОКОПРОИЗВОДИТЕЛНИТЕ ПЕРСОНАЛНИ КОМПЮТРИ ОТ ФАМИЛИЯТА ПРАВЕЦ

В последно време намират все по-голямо разпространение и високопроизводителните персонални компютри (ВПК), съвместими със серията IBM PC/AT. Такива са ВПК от фамилията Правец с шифър ЕС1838. Те имат определени конструктивни разлики от Правец-16 и трябва да се има предвид, че не всички програми, които работят на високопроизводителните персонални компютри, могат да бъдат изпълнявани на Правец-16. Тъй като общата архитектурна концепция е една и съща, програмното осигуряване е изградено по такъв начин, че да има определена съвместимост и приемственост отдолу нагоре, като почти всички програми за фамилията Правец-16 работят и върху ВПК. Потребителят трябва да знае разликите между двете фамилии.

Високопроизводителните персонални компютри от серията АТ са изградени на базата на микропроцесор I80286, който е съвместим с I8088 на ниво инструкции. Системната платка за ВПК освен основния куплунг с 64 пера на Правец-16 има допълнителен куплунг с 36 пера. Адаптерите, които използват допълнителния куплунг, не могат да се вграждат в персоналните компютри Правец-16.

Честотата от основния тактов генератор служи само за синхронизиране на процесора, но не се използва от контролери и адаптери, които изискват фиксирана честота. При това трябва да се има предвид, че основният тактов генератор, задаващ честотата на процесора, не е синхронизиран с генератора, чиято честота се разпространява по системната шина.

В някои от сигналите също има определени разлики. Сигналят АLE се активира по време на циклите за ДДП, а сигналят IQW не е активен по време на циклите за опресняване на паметта. Върху перо В04 на основния куплунг е фиксиран сигналят за прекъсване IRQ9.

Във ВПК се включват ЗУТМД с обем 20 Мбайта. Записът и четенето се инициализират по същия начин, както и за ПК от фамилията Правец. Адаптерите за дисковете могат да се адресират от различни области на БСВИ. ЗУГМД във ВПК позволява да се формират и използват дискети с обем 160/180 Кбайта, 320/380 Кбайта и 1,2 Мбайта. Когато дискетата се форматира за 160/180 Кбайта или за 320/360 Кбайта, тя може да се чете само от собственото ЗУГМД, т.е. дискети, форматирани на други персонални компютри Правец-16, не могат да се четат на това устройство. Дискети с обем 1,2 Мбайта не могат да се използват в

ЗУГМД за работа със 160/180 Кбайта или с 320/360 Кбайта. Адаптерите за разширение на паметта със 128 Кбайта, с 512 Кбайта и контролерът за ЗУГМД използват допълнителния куплунг и не са съвместими с останалите компютри на фамилията Правец. Клавиатурата на ВПК има 84 клавиша, които могат да изпълняват всичките функции на клавиатурите на другите ПК, но не могат да се включват директно към тях, тъй като имат различен интерфейс. Съществуват обаче клавиатури, които разпознават съответните компютри и могат да работят и с ВПК, и с ПК.

Към ВПК не могат да се включват директно следните адаптери на ПК от фамилията Правец: устройства за разширение, асинхронен комуникационен адаптер, адаптер за разширение на паметта 64/256 Кбайта, адаптер за печатащо устройство. Видеоконтролерът за ПК може да се включва директно към ВПК.

Програми, съвместими и за двата типа компютри, могат да се пишат на езиците С, БЕЙСИК, ФОРТРАН, ПАСКАЛ, APL. Когато програмите са свързани с особеностите на апаратната част на ВПК, те могат да не се изпълняват правилно на ПК. Това е валидно за асемблерските програми, използващи специфични команди, като In, Out, Pack, Poke.

Тъй като микропроцесорите на ВПК и ПК работят с различни честоти, всяка програма, която изисква определени времеви интервали, трябва да ги уточни със средствата на ДОС или със съответния език от високо ниво, например TIME в БЕЙСИК. Съществено за входно-изходната система БСВИ на ВПК е, че тя съдържа средства за поддържане на многозадачна работа.

Високопроизводителните персонални компютри дават възможност редица програмни продукти да се използват по-ефективно, тъй като притежават по-висока производителност и разширено адресно пространство. Такива са: програмни системи на базата на езиците ЛИСП и ПРОЛОГ, а също и някои приложни програми за автоматизация на инженерната дейност (програмен пакет за автоматизация в областта на машиностроенето — т. 4.2.7).

Освен това ВПК позволяват използването на операционни системи, работещи в многопрограмен режим с виртуална памет. Новата серия персонални компютри на IBM — PS2 работи с операционна система OS2, която може да се използва и от високопроизводителните персонални компютри.

В асемблерския език има редица кодове на операции, които работят различно за ВПК и за останалите ПК, съвместими с фамилията Правец. Съществуват и особености при изпълнение на част от командите.

ЛИТЕРАТУРА КЪМ ГЛ. 3

1. Дисплей цветной ВМЦ 3701. Паспорт. НПП „Аналитик“ — Михайловград. Д22.049.005 ПС.
2. Дисплей черно-белый ВММ 31-02. Паспорт. НПП „Аналитик“ — Михайловград. Д22.049.003 ПС.
3. Клавиатура ИЗОТ-1036С.С001. Техническое описание и инструкция по эксплуатации. ЦИИТТ—София. Ц16.065.104 ТО.
4. МПУ М88. Руководство по эксплуатации. ИМПТ-София. П13.043.010 РЭ.
5. НГМД ЕС 5326. Паспорт. СО ОЗЗУ — Стара Загора. К53.060.008 ПС.

6. НМД на твердом носителе СМ 5508. Паспорт. СО ОЗЗУ — Стара Загора. В33.060.101 ПС.
7. ПП ЭВМ ЕС 1832. Инструкция по эксплуатации. ЦИИТТ—София. Ц13.055.056 ИЭ.
8. ПП ЭВМ ЕС 1832. Паспорт. ЦИИТТ—София. Ц13.055.056 ПС.
9. ПП ЭВМ ЕС 1832. Техническое описание. ЦИИТТ—София. Ц13.055.056 ТО.
10. ПП ЭВМ ЕС 1839. Инструкция по эксплуатации. ИМПТ—София. Е30.305.001 ИЭ.
11. ПРАВЕЦ-16. ПАСПОРТ. ИТКР-БАН. КР3.021.003 ПС.
12. ПП ЭВМ ЕС 1839. Техническое описание. ИМПТ—София. Е30.305.001 ТО.
13. ТЭЗ ВПО и управление коммуникации ЕС 1832.1003. Техническое описание и инструкция по эксплуатации. ЦИИТТ—София. Ц13.092.497 ТО.
14. ТЭЗ Кодер ЕС 1832.1101. Техническое описание и инструкция по эксплуатации. ЦИИТТ—София. Ц13.092.503 ТО.
15. ТЭЗ Процессор и 256К ОП ЕС 1832.1001. Техническое описание и инструкция по эксплуатации. ЦИИТТ—София. Ц13.092.495 ТО.
16. ТЭЗ Синхронные коммуникации — BSC ЕС 1839.0002. Техническое описание и инструкция по эксплуатации. ИМПТ—София. 662451.3.090.497 ТО.
17. ТЭЗ Синхронные коммуникации — SDLC ЕС 1839.0001. Техническое описание и инструкция по эксплуатации. ИМПТ—София. 662451.3.090.496 ТО.
18. ТЭЗ Управление коммуникации ИЗОТ 1036С.0007. Техническое описание и инструкция по эксплуатации. ЦИИТТ—София. Ц13.092.544 ТО.
19. ТЭЗ Управление НГМД ИЗОТ 1036С.0006. Техническое описание и инструкция по эксплуатации. ЦИИТТ—София. Ц13.092.543 ТО.
20. ТЭЗ УУ дисплея ЕС 1832.1004. Техническое описание и инструкция по эксплуатации. ЦИИТТ—София. Ц13.092.498 ТО.
21. УУ НМД на твердом носителе ЕС 1839.2001. Техническое описание и инструкция по эксплуатации. ИМПТ—София. 662451.3.090.504 ТО.
22. Центральное устройство ЕС 1839.A001. Техническое описание. ИМПТ—София. КР3.021.005 ТО.
23. iAPX-88 Architecture And Instructions. Intel, 1983.
24. Personal Computer XT Hardware Reference Library. Technical Reference. IBM, 1983.

ГЛАВА 4

ПРОГРАМНО ОСИГУРЯВАНЕ НА 16-РАЗРЕДНИ ПЕРСОНАЛНИ КОМПЮТРИ

4.1. БАЗОВО ПРОГРАМНО ОСИГУРЯВАНЕ

4.1.1. ДИСКОВА ОПЕРАЦИОННА СИСТЕМА ДОС-16

Основната дискова операционна система за ПК Правец-16 (ЕС1839) е ДОС-16 [5].

ДОС-16 е еднозадачна еднопотребителска операционна система, която изпълнява следните основни функции:

- създаване, поддържане и управление на файловата система;
- поддържане на дървовидна файлова структура;
- управление на стандартния вход-изход;
- управление на паметта на микрокомпютъра;
- управление на изпълнението на програмите.

ДОС-16 се състои от няколко функционално обособени части. На най-високото ниво и най-близо до потребителя е диалоговата система, а на най-ниското ниво — компонентите, които управляват работата на компютъра с периферните устройства (драйверните програми). Базовата система за вход—изход (BIOS) представлява пакет програми за обмен на данни между компютъра и периферните устройства на ниво байт. Тези програми са записани в постоянна памет и физически са част от компютъра.

Драйверите са отделни програми за управление на периферни устройства, които не са включени в BIOS. Те се съхраняват като файлове на системната дискета. Една част от драйверите са стандартни и се доставят с операционната система, а друга част се осигуряват от производителите на периферни устройства. Възможно е и създаването на трети вид драйвери, които се разработват от потребителя за обмен на информация с устройство, което не влиза в основната конфигурация на компютъра.

P16BIO е интерфейс между BIOS и същинската част на операционната система — P16DOS, като P16BIO е записан на системна дискета и се зарежда в паметта при стартиране на ДОС-16.

P16DOS е ядрото на операционната система и съдържа програми за работа с файлове, управление на паметта и управление на изпълнението на програмите. Тези програми реализират конкретни функции и се изпълняват по заявка от горното ниво (команда и оператор в програма). При работата си повечето функции формират заявки към P16BIO, които от своя страна се обръщат към BIOS. Програмите, съставляващи пакета P16DOS, са записани в дисков файл с име P16DOS.COM, който се зарежда в паметта при стартиране на системата.

На най-високото ниво и най-близо до програмиста се намира командният интерпретатор на ДОС-16. Той интерпретира командите и

обработка грешките при изпълнението на програмите или другите части на ДОС-16. Командният интерпретатор е разделен на две части— резидентна и нерезидентна (транзитна).

Транзитната част на командния интерпретатор се зарежда в най-старшите адреси на паметта.

Резидентната част обработка възникнали особени състояния в системата (обработка на грешки, завършване на програми и др.). Резидентната част на командния интерпретатор съдържа програма за зареждане на операционната система, както и програма за зареждане на транзитната част на командния интерпретатор.

Резидентната и транзитната част на командния интерпретатор се съдържат във файл с име COMMAND.COM и се зареждат в паметта при стартиране на ДОС-16.

Част от командите на операционната система са включени в командния интерпретатор и затова се наричат *вградени*. Останалата част от командите са обособени като отделни файлове, записани на системната дискета и всеки път при извикване се зареждат в паметта. Тези команди се наричат *външни*.

Стартиране на системата

При включване на захранването на компютъра управлението се предава на BIOS, който изпълнява серия тестови процедури за инициализация, наречени програма за самотестване или автоматичен тест. При рестартиране на системата с помощта на клавиши Alt-Ctrl-Del автоматичният тест не се изпълнява.

След приключване на автоматичния тест се изпълнява програма за начално зареждане на ДОС-16, в която се реализира следната последователност:

1. BIOS зарежда в паметта програмата за начално зареждане, която се намира на страна 0, писта 0, сектор 1 от дискетата в първо дисково устройство и ѝ предава управлението. Стартираната програма проверява наличността на файловете P16BIO.COM и P16DOS.COM. Ако е намерен файлът P16BIO.COM, той се зарежда в паметта и му се предава управлението.

2. Изпълняват се програми за начална инициализация на периферни устройства и се конфигурира системата в зависимост от съдържанието на файла CONFIG.SYS. Установяват се векторите на прекъсванията в определената за това област от паметта. Управлението се предава на P16DOS.

Програмата CONFIG.SYS съдържа някои параметри, необходими за работата на системата, които се задават от потребителя във вид на команди. Тези команди се интерпретират от инициализиращата програма на ДОС-16. По този начин се определят броят на буферите за вход-изход, броят на едновременно отворените файлове, имената на драйверите, които се зареждат при инсталиране на ДОС-16. Ако липсва файлът CONFIG.SYS, тези параметри се зареждат автоматично с подразбиращи се стойности.

3. Инициализиращата програма в P16DOS задава някои вектори на прекъсване и предава управлението на P16BIO, който зарежда в паметта резидентната част на командния интерпретатор и му предава управлението.

4. Подпрограмата за инициализация в резидентната част зарежда транзитната част на командния интерпретатор, инициализира някои вектори на прекъсване и изпълнява командите, зададени във файла за автоматично стартиране на системата AUTOEXEC.BAT (ако той съществува върху дискетата). След това се определя първият адрес от паметта, свободен за зареждане на програми, и управлението се предава на транзитната част на командния интерпретатор.

След зареждане на операционната система при наличие в компютъра на работна памет 512 Кбайта се получава разпределение, илюстрирано на табл. 4.1.

Таблица 4.1

**Разпределение на адресното пространство
в компютри, работещи под управлението на ДОС-16**

Адреси	Размер, Кбайта	Предназначение
0000—9FFF	640	Работна памет (RAM)
A000—F000	128	Памет за обслужване на видеоконтролери (RAM)
C000—C7FFF	32	Резервирана
C8000—F3FFF	176	Памет за контролери на други периферни устройства (ROM)
F4000—FFFFF	48	Постоянен ROM на основната платка, BIOS, вграден БЕЙСИК, разширения на BIOS

Физическа организация на дисковите устройства

За персоналният компютър Правец-16 (ЕС1839) се използват ЗУГМД с диаметър 133 mm и ЗУТМД.

Физическото форматиране на дискетите се осъществява с командата FORMAT на ДОС-16. Операционната система поддържа сектори с размер 512 байта. Броят на секторите в една пътечка е постоянно число и не може да бъде изменен.

С персоналният компютър Правец-16 могат да се използват следните типове дискети:

- едностранно форматирани дискети;
- двустранно форматирани дискети;
- дискети с голям капацитет.

Едностранните дискети след форматиране съдържат 40 пътечки с 8 или 9 сектора на всяка от тях, като всеки сектор съдържа по 512 байта. Обемът на дискетата е съответно 160 или 180 Кбайта.

Двустранните дискети съдържат 40 цилиндъра. Всеки цилиндър съдържа пътечки с едни и същи номера от двете страни на дискетата. По този начин капацитетът на дискетата се удвоява (320 или 360 Кбайта).

Дискетите с голям капацитет съдържат 80 пътечки с по 15 сектора на всяка от тях и имат капацитет 1,2 Мбайта.

ЗУГМД с голям капацитет се поддържа от BIOS на компютрите IBM PC/AT и програмно съвместимите с него. Използването му с компютър Правец-16 е възможно при наличието на специално инсталиран драйвер.

На табл. 4.2 е дадена информация за параметрите на дисковете, поддържани от ДОС-16.

Таблица 4.2

Магнитни носители, поддържани от ДОС-16

Вид диск	Брой повърхности	Брой пътечки на повърхност	Брой сектори в пътечка
Дискета 160 Кбайта	1	40	8
Дискета 180 Кбайта	1	40	9
Дискета 320 Кбайта	2	40	8
Дискета 360 Кбайта	2	40	9
Дискета 1.2 Мбайта	2	80	15
Твърд диск 10 Мбайта	4	305	17
Твърд диск 20 Мбайта	4	614	17

Логическа организация на дисковата памет

1. Организация на данните върху дискета

Физическото форматиране на дискета с команда **FORMAT** е съпроводено с логическо форматиране, при което се изгражда нейната логическа структура. Логическото форматиране се изразява в подходящо разпределение на физическото пространство върху дискетата, което се използва за запис на блок за начално зареждане на системата, таблица за разположение на файловете, главен каталог и област за данни. При форматирането се записва служебна информация, която задава логическата структура на дискетата.

Най-малката физическа единица, която участва в дисковите операции, е *секторът*. За адресирането на който и да е сектор на дискетата е достатъчен неговият абсолютен адрес, който се образува от номера на пътечката и номера на сектора върху пътечката. Освен абсолютен адрес всеки сектор има и относителен адрес (логически номер), който определя номера на сектора в рамките на общия брой сектори на дискетата. При едностранна дискета логически номер 0 се присвоява на сектор с физически номер 1 на пътечка 0. Логическите номера се присвояват в нарастващ ред до края на пътечка 0 и продължават със сектор 1 на пътечка 1 и т.н. При двустранните дискети логическите номера се присвояват по цилиндри: отначало се номерират всички сектори на повърхност 0 пътечка 0, след това секторите от повърхност 1 пътечка 0, после повърхност 0 пътечка 1, повърхност 1 пътечка 1 и т.н.

ДОС-16 разпределя дисковото пространство върху дискетата с 8 сектора на пътечка по следния начин:

- пътечка 0, сектор 1 — програма за начално зареждане (ПНЗ);
- пътечка 0, сектори 2 и 3 — две копия на таблицата за разпределение на дисковото пространство (ТРДП), всяко копие в по един сектор;
- пътечка 0, сектори 4—7 — каталог на файловете върху диска;
- от пътечка 0, сектор 8 до пътечка 39, сектор 8 — област за данни при едностранните дискети.

За дискети с 9 сектора на пътечка разпределението е следното:

- пътечка 0, сектор 1 — ПНЗ;
- пътечка 0, сектори 2, 3, 4 и 5 — две копия на ТРДП, всяко копие в два сектора;

— от пътека 0, сектор 6 — каталог на файловете.

Файловете се записват в областта за данни. За всеки файл се заделва необходимият обем дисково пространство. Най-малката единица разпределено дисково пространство се нарича *сегмент*. Всеки сегмент се състои най-малко от два физически последователно разположени сектора. Сегментите, заделени за определен файл, не е задължително да бъдат физически последователни. При определяне на поредния сегмент за файла операционната система търси първия незаает независимо от неговото физическо разположение и ако намери такъв, го заема за файла. Това води до рационално използване на дисковото пространство, тъй като всички свободни сегменти след изтриване на файлове могат да бъдат заети без ограничения от нови файлове.

Файловете на операционната система P16DOS.COM и P16IOS.COM са разположени на фиксирани адреси върху системната дискета, което позволява на зареждащата програма успешно да ги зареди в оперативната памет при стартиране на системата. При едностранните дискети файлът P16IOS.COM е разположен от пътека 0 сектор 8 до пътека 1 сектор 3, а файлът P16DOS.COM — от пътека 1 сектор 4 до пътека 2 сектор 8. При двустранните дискети файлът P16IOS.COM е разположен на първа страна на дискетата, пътека 0, сектори 7 и 8, и продължава на нулева страна на дискетата, пътека 1, сектори 1—8 и страна 1, пътека 1, сектори 1—4.

2. Организация на данните върху твърд диск

С цел общо използване на твърдия диск от няколко операционни системи той може да бъде разпределен на части — до четири. В отделните части на диска паметта е организирана последователно и може да бъде назначена за определена операционна система (в даден момент работи само една операционна система). Всяка операционна система може да заема само една част от твърдия диск. Броят на частите, на които се разделя твърдият диск, се определя от потребителя по време на работата на командата FDISK, която подготвя твърдия диск за работа. Информацията за всяка част от твърдия диск се съхранява в таблица, която е разположена на фиксиран дисков адрес (в първия сектор на диска).

Всяка операционна система разполага със своята част от диска като с отделно устройство и нейните функции и обслужващи програми не трябва да използват другите части.

Всяка част може да съдържа в първия си сектор програма за начално зареждане, а също програми и данни по желание на потребителя.

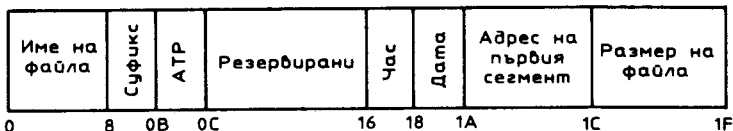
Всеки твърд диск трябва да има в първи сектор нулева пътека, цилиндър 0, таблица, описваща частите на твърдия диск. Всеки елемент на таблицата е с дължина 16 байта — по 4 байта за максимално възможните 4 части за разделянето на диска, за броя на секторите пред съответната част и за броя на заетите от нея сектори. Съществува и отделен байт — „индикатор за зареждане“, който определя коя от операционните системи може да бъде заредена в даден момент.

3. Каталог на файловете

Всеки диск, предназначен за работа с операционна система, има каталог на файловете (КФ). Всеки елемент на каталога описва един файл. Създаването и инициализацията на каталога се изпълнява от командата FORMAT, а всички останали команди на операционната система поддържат този каталог.

За едностранни дискети КФ заема 4 сектора и може да съдържа до 64 елемента, всеки от които има размер 32 байта.

КФ за двустранните дискети заема 7 сектора и максималният брой елементи в него е 112. Структурата на всеки елемент от КФ е дадена на фиг. 4.1.



Фиг.4.1. Структура на елемент от каталога на файловете

Шестнадесетичните числа под съответните полета означават относителното отместване спрямо началото на елемента.

ДОС-16 позволява създаването и поддържането на дървовидна структура на каталозите. При форматирането на диска се създава един главен каталог, който може да съдържа 64 елемента за едностранна и 112 елемента за двустранна дискета.

Освен имена на файлове основният каталог може да съдържа и други каталози, които от своя страна могат да съдържат имена на файлове и/или каталози и т.н. За разлика от главния каталог тези каталози се наричат подкаталози и сами по себе си представляват файлове, поради което те са неограничени по размер. Подкаталозите могат да съдържат неограничен брой файлове и подкаталози, които се ограничават единствено от наличното дисково пространство.

Имената на подкаталозите имат същия формат като имената на файловете. Всеки подкаталог може да съдържа имена на файлове и имена на подкаталози, които съществуват и в други каталози.

Освен текущото подразбиращо се устройство ДОС-16 съхранява информация и за текущия каталог във всяко друго устройство. В текущия каталог се търси името на заредим файл или каталог, ако не е указано явно устройството или каталогът.

4. Таблица за разпределение на дисковото пространство

Таблицата за разпределение на дисковото пространство (ТРДП) съдържа информация за разпределението на сегментите в областта за данни на диска. Всяка форматирана дискета съдържа две кония на ТРДП. В оперативната памет също се поддържа ТРДП за последната използвана дискета от всяко устройство. ТРДП е съвкупност от 12-битови елементи, на всеки от които съответства по един сегмент от дискетата.

Първите два елемента на ТРДП съдържат информация за типа на диска и могат да съдържат стойности във вида XXFFFF, където XX се изменя и може да заема следните стойности:

- FF — двустранна дискета с 8 сектора в пътечка;
- FE — едностранна дискета с 8 сектора в пътечка;
- FD — двустранна дискета с 9 сектора в пътечка;
- FC — едностранна дискета с 9 сектора в пътечка;
- F8 — твърд диск.

От третия елемент на ТРДП започват указателите на сегментите, в които са занесени данните. За всеки файл се поддържа последователност от елементи в таблицата. Всеки от тях е указател към област от дисковото пространство с фиксирана дължина, заемана от файла върху диска (сегмент). В каталога, към който принадлежи файла, има указател на първия му сегмент. При изтриване на файла елементите се освобождават и могат да бъдат използвани от друг файл.

Команди на ДОС-16

Командите на ДОС-16 се делят на две големи групи - вътрешни и външни. Вътрешните команди са въведени в командния интерпретатор и се отнасят до:

- копиране на файлове;

- преименуване на файлове;

- работа с каталози и др.

Всеки файл върху дискетата с разширение .COM, .EXE или .BAT се счита за външна команда.

Командите на ДОС-16 служат за:

- поддържане на файловата система - сравняване, изтриване, копиране, преименуване и преглеждане съдържанието на файлове — COMP, DEL (ERASE), COPY, RENAME, TYPE;

- поддържане на носителите - форматиране, копиране, сравняване на дискети и твърди дискове — FORMAT, FDISK, DISKCOPY, DISKCOMP;

- изпълнение на системни (EDLIN, DEBUG, LINK) и потребителски програми;

- определяне на режима на екрана и на принтера — MODE;

- заявка за пауза — PAUSE;

- прехвърляне на ДОС-16 на дискета със запазване на информацията върху нея - - SYS;

- замяна на стандартната конзола (клавиатура и дисплей) с друго устройство за вход-изход — CTTY;

- възстановяване на един или повече файлове от повреден диск и на дискета - RECOVER;

- отпечатване на съдържанието на файл, докато системата изпълнява друга задача — PRINT;

- създаване на резервно копие върху дискети (архивиране) на един или повече файлове от твърдия диск и възстановяване на файловете от архивни дискети --- BACKUP, RESTORE;

- сортиране на текстови данни — SORT;

- прекъсване на файла за запис, съдържащи определен символен низ — FIND;

- определяне на нов системен знак — PROMPT;

- определяне на параметрите на операционната среда — SET;

- превръщане на файлове със суфикс .EXE във файлове със суфикс .COM — EXE2BIN;

- инсталиране на системни програми, управляващи съвместното използване на файловете — SHARE;

- установяване на атрибути на файла — команда ATTRIB;

- присвояване или промяна на етикет на том — LABEL;

- заявяване на разширена проверка за команда **BREAK CTRL-BREAK**;
- създаване и отстраняване на каталози, смяна на текущия каталог — **MKDIR, RMDIR, CHDIR**;
- извеждане на списък с всички каталози на диска — **TREE**;
- анализ на коректността на каталози, файлове и таблицата за разпределение на дисковото пространство — **CHKDSK**;
- задаване или извеждане на датата и часа, с които **DOC-16** работи — **DATE, TIME**;
- извеждане на информация за файловете, съхранявани на диска — **DIR**;
- присъединяване на дисково устройство към каталог на друго дисково устройство, за да стане достъпно чрез този каталог — **JOIN**;
- заместване на спецификатор на дисково устройство или каталог със спецификатор, определящ друго дисково устройство за осъществяването на достъп чрез заместващия спецификатор — **SUBST**;
- възможност за бърз достъп до файлове чрез съхранение в паметта на адресите на каталога за текущо отворените файлове — **APPEND**.

Текстов редактор **EDLIN**

EDLIN е текстов редактор, ориентиран към поредова обработка на текста. **EDLIN** се използва за създаване, изменение и преглеждане на текстови файлове и файлове, съдържащи програми в изходен код. **EDLIN** може да извършва следното:

- създаване и съхранение на нови програмни и текстови файлове;
- изменение на съществуващи файлове и съхраняване както на файла с измененията, така и на оригинала;
- изтриване, редактиране, вмъкване, дублиране, преместване и преглеждане на редове;
- прегърване на текста в рамките на един или повече реда с цел отстраняване или замяна.

Текстът в създадените или редактирани файлове с помощта на **EDLIN** се състои от редове с променлива дължина. За всеки ред от първичния текст редакторът генерира пореден номер и го извежда на монитора. Номерата на редовете не се записват върху диска при съхраняване на файла. При вмъкване (или изтриване) на редове номерата на редовете, разположени след вмъкнатия (или изтрития) ред, автоматично нарастват (или намаляват) с броя на добавените (изтритите) редове.

Ако при извикването на програмата **EDLIN** е специфициран файл, който съществува върху дискетата на указаното или текущото дисково устройство, тя зарежда файла (ако той не превишава 75% от свободната памет), след което може да се започне редактирането на заредения файл.

Ако файлът е много голям, може да се редактира само част от него и след запис на тази част върху диска може да се добавят от диска нередитирани редове.

При редактиране на съществуващ файл и завършване на работата оригиналният файл се преименува със суфикс **.BAK**. Файловете с такъв суфикс не могат да бъдат редактирани с **EDLIN**, тъй като те се считат за резервно копие.

Ако при извикването на EDLIN е специфицирано име на файл, което не съществува върху диска на указаното или текущото устройство, EDLIN създава нов файл с указано име.

Програма за свързващо редактиране LINK

Програмата LINK е предназначена за свързване на един или няколко отделно създадени обектни модула в един изпълним модул. За да се удовлетворят външните обръщения, програмата LINK претърсва указаните библиотечни файлове и установява връзката между външните имена и външните обръщения. Като резултат от работата на програмата LINK се създава изпълним преместваем файл, съдържащ удовлетворени външни връзки и листингов файл със съобщения за грешки (ако има такива).

Програмата LINK използва цялата достъпна памет за обработка на данните, изграждащи изпълнимия модул. Ако модулът е много голям и не е възможно той да бъде обработен в достъпната памет, програмата LINK създава временен файл върху текущото дисково устройство. След завършване на работата на програмата LINK временният файл се изтрива.

Файловете спецификации на обектните модули, които ще бъдат свързани в изпълним модул, на изпълнимия и листинговия файл, както и тези на библиотечните модули се залават на програмата LINK в диалогов режим или се приемат подразбиращите се.

Програма за настройка DEBUG

Програмата за настройка DEBUG е един от основните компоненти на дисковата операционна система ДОС-16. Програмата дава следните възможности:

- зареждане на избрани сектори от диска в паметта;
- зареждане и стартиране на изпълними програми в оперативната памет;
- управляемо тестване на програми, при което програмистът наблюдава и контролира изпълнението на програмата;
- прехвърляне на съдържанието на една област от паметта в друга с указан начален адрес;
- модификация на области от паметта и извеждане съдържанието им на дисплея;
- запис върху диск на настройваната програма;
- запис и четене на байт в или от зададен порт;
- реасемблиране на съдържанието на области от оперативната памет;
- визуализиране на съдържанието на всички регистри и флагове с възможност за изменението им;
- претърсване на област от паметта.

Програмата DEBUG позволява да се осъществи проверка на изпълнимата програма с възможност за внасяне на корекции и изпълнение на модифицираната програма без междинно транслиране.

Версии на интерпретатора на БЕЙСИК

Съществуват три версии на интерпретатора на БЕЙСИК микрокомпютър Правец-16 [1]:

- касетен БЕЙСИК (не се използва в Правец-16);
- дисков БЕЙСИК;
- разширен БЕЙСИК.

Трите версии са изцяло съвместими отдолу нагоре, т.е. всяка програма, написана на касетен БЕЙСИК, може да бъде изпълнена от интерпретатора на дисковия и разширения БЕЙСИК. Обратното не винаги е възможно. Версиите на БЕЙСИК включват следните възможности:

- използване на разширен набор от символи (до 256);
- използване на два графични режима (графика със средна разделителна способност и графика с висока разделителна способност);
- използване на външни устройства (като джойстик, светлинна писалка, високоговорител и др.).

Дисков БЕЙСИК. Това е файл с име BASIC, който се зарежда в оперативната памет на микрокомпютъра и има следните възможности:

- работа с дискови файлове;
- поддържане на системен часовник;
- комуникация по асинхронен интерфейс от тип RS232;
- поддържане на две печатани устройства.

Разширен БЕЙСИК. Притежава най-големи възможности в сравнение с другите две версии:

— възможност за обработка на прекъсвания от определени източници (активност на комуникация, натискане на функционален клавиш, прекъсване от таймера или светлинната писалка и т.н.). При настъпване на определено събитие може да се осъществи преход към даден клон на програмата:

— допълнителни оператори за работа с графика в режим на висока разделителна способност;

— допълнителни оператори, разширяващи възможностите за възпроизвеждане на тонове.

Разширеният БЕЙСИК е записан върху дискетата като файл с име BASICA.

Режим на работа

Връзката с интерпретатора на БЕЙСИК се осъществява в два режима:

1. Директен, при който на всеки въведен ред интерпретаторът отговаря незабавно. Инструкциите след изпълнението им не се запомнят. Този режим се използва за настройка на програми или за бързо изпълнение на някои изчисления.

2. Индиректен, при който въвежданите редове са части от програма. След въвеждане на такъв ред той се запомня в паметта като част от програма. Програмата, която се намира в паметта, може да бъде изпълнена с команда RUN.

Интерпретаторът на БЕЙСИК разполага със собствен екранен ре-

дактор по редове. Това означава, че в даден момент може да бъде редактиран кой да е ред, намиращ се на екрана, но само той. Описание на възможностите на езика и отделните команди са дадени в [11].

4.1.3. ТЕРМИНАЛЕН ЕМУЛАТОР МИКРОТЕРМ

МИКРОТЕРМ [15] е комуникационна система, която дава следните възможности:

- терминална емуляция;
- пренасяне на файлове;
- обща асинхронна комуникация.

Програмата МИКРОТЕРМ дава възможност персоналният компютър (Правец-16, ЕС1839, ЕС1838 или програмно съвместими с тях) да работи като терминал на машина CM4, VAX и др. МИКРОТЕРМ емулира терминали CM1604, CM1604M1, CM1604M3 и др., аналогични на VT52 и VT100. По всяко време на работата на персоналния компютър в режим на терминална емуляция може да се премине в режим на работа с операционната система ДОС-16. Това позволява компютърът да изпълнява програми, без да се прекъсва сесията с главния компютър. След като локалната програма завърши работата си, може да се премине в режим на терминална емуляция.

МИКРОТЕРМ позволява да бъдат обменяни файлове с данни между главния компютър и терминала по следните видове протоколи:

- XMODEM с автоматична корекция на грешките чрез добавяне на определени „протоколни“ символи към данните;
- MTRANS7 и MTRANS8 за обмен на файлове със собствено коригиране на грешките;
- безпротоколен обмен, при който обменът на файлове е освободен от всякакви протоколи.

Програмата поддържа асинхронна комуникация чрез модем или чрез директно свързан кабел (нулев модем). Максималната скорост на предаване е 9600 бит/с. Реализирана е възможност за автоматично избиране на телефонен номер и включване към дадена система.

4.1.4. ЕКРАНЕН РЕДАКТОР

Екранният редактор PE2 [6] предоставя възможности за създаване и модифициране на файлове, съдържащи информация в първичен код. Файловете могат да съдържат текст на програми, документи или данни. Редакторът осигурява възможности за търсене на определена последователност от символи, дублиране, преместване или изтриване на текст.

Тези възможности се реализират чрез маркиране на части от текста (дума, изречение, ред или множество от редове) и манипулиране с маркирания участък.

Екранният редактор позволява разделяне екрана на монитора на 4 прозореца, във всеки от които може да бъде визуализиран текст от различни файлове. Това значително улеснява работата при комбиниране на редове от различни файлове. Възможна е едновременна работа с няколко (до 20) едновременно отворени файла в паметта на компютъра.

наречени *активни*. Превключването от един активен файл на друг става чрез функционални клавиши. Програмистът разполага с богат набор от функции, комбинации от които могат да бъдат присвоени на един клавиш.

Екранният редактор реализира функциите си чрез два режима на работа:

- текстов;
- команден.

Когато редакторът се намира в команден режим, могат да бъдат въвеждани само команди. Преминването от един режим в друг става чрез функционален клавиш.

Командите на екранния редактор дават следните възможности:

- поддържане на файлове — създаване, съхраняване;
- установяване и изменение на лявата и дясна граница на текст;
- установяване и изменение на табулаторите;
- изменение на режима на монитора — цветен, чернобял и монохроматичен;
- търсене и изменение на текст;
- извеждане на съдържанието на текущия активен файл;
- дефиниране и предефиниране на клавиши.

В текстов режим редакторът позволява създаване, коригиране и изтриване на текст с помощта на множество функции. Всяка функция е присвоена на един клавиш или комбинация от клавиши. Присвояването става при стартирането на редактора или по време на работата чрез предефиниране на началните присвоявания.

Функциите на редактора реализират различни действия.

1. Придвижване на курсора до:
 - предходна позиция;
 - първия или последния знак на текущата дума;
 - първия или последния знак на текущия ред;
 - първия или последния знак на текущия маркиран текст;
 - първия или последния ред във файла и др.
2. Изпълнение на команди, записани в команден ред.
3. Преобразуване на всички малки букви в големи или обратно в маркирана област.
4. Преформатиране на текст в маркирана област и др.

Екранният редактор има възможност за работа в многопотребителски режим под управлението на операционната среда ПДОС-16.

4.1.5. ТЕРМИНАЛНИ ЕМУЛАТОРИ ЗА ПЕРСОНАЛНИ КОМПЮТРИ

Емулаторите за персоналните компютри позволяват тези компютри да бъдат включвани в качеството на терминали към главна ЕИМ в отворена система за мрежова телеобработка, като се използва протокол за синхронно побитово предаване на данни (SDLC).

Емулаторът [26] функционира като терминална станция ЕС8566 или програмируем терминал ЕС8531М1 с програма за отдалечено въвеждане на задания в главната ЕИМ [9].

Емулаторът ЕС8566 представлява множество от програми и данни.

които работят върху персоналните компютъри, снабдени със синхронен адаптер за пробитово предаване на данни. Комуникацията между персоналния компютър и главната ЕИМ се осъществява по комутируеми и некомутируеми линии. В главната ЕИМ се стартира програма, поддържаща комуникациите.

Емулаторът работи в два режима:

— избор на параметрите на терминала;

— комуникация.

При първия режим се установява наборът от параметри за системата. Тези параметри са задължителни за стартиране на емулатора и за установяване на връзка с главната ЕИМ. След като връзката е установена, въвежданите от клавиатурата данни се изпращат към ЕИМ, а съобщенията, изпращани към терминала, се извеждат на екрана на монитора.

Емулаторът на функции за връзка с отдалечено въвеждане на задания емулира възможностите на терминала ЕС8531, който осигурява връзка с програмата за отдалечено въвеждане на задания, работеща в главната ЕИМ. Емулаторът прехвърля предварително подготвени за работа задания за ОС/351 като множество данни от персоналния компютър към главната ЕИМ. Заданията се изпълняват в главната ЕИМ под управлението на ОС/351 и резултатите от изпълнението могат да бъдат върнати обратно към изходния персонален компютър или към друг терминал. Връзката на персоналния компютър с главната ЕИМ може да бъде от тип *точка-точка* или *многоточкова* и се установява по комутируема и некомутируема линия.

Емулаторът за комуникация с програмата за отдалечено въвеждане на задания също има два режима на работа: избор на параметри и комуникация.

В първия режим се установяват параметрите за конкретната комуникационна среда, например скорост на обмен, тип контрол и т.н.

По време на комуникацията се установява връзка с главната ЕИМ, след което могат да се изпращат задания към програмата за отдалечено въвеждане на задания и да бъдат получавани резултати. Информацията от главната ЕИМ по желание може да бъде извеждана на екрана на монитора, печатащото устройство или като файл върху гъвкав магнитен диск.

При комуникация с главната ЕИМ се използва код ДКОИ. Средата е прозрачна за предавания текст, т.е. могат да се изпращат и получават всичките 256 знака.

Емулаторът на програмата за отдалечено въвеждане на задания дава статистика в края на всяка сесия, която съдържа броя на изпратените към главната ЕИМ записи, броя получени записи, броя записи за обмен и др.

4.1.6. СИСТЕМА ЗА ЛОГИЧЕСКО ПРОГРАМИРАНЕ ПРОЛОГ-16

ПРОЛОГ е мощен език, развит като практически инструмент за решаване на задачи за логическо програмиране. ПРОЛОГ е най-популярният език в областта на изкуствения интелект. Неговото основно предимство е декларативното описание на проблема — не как да бъде намерено решението, а какви са неговите характеристики.

Езикът ПРОЛОГ е подходящ за решаване на задачи в различни области — от образованието и научния експеримент до създаването на ефективни програмни пакети в организацията и управлението.

Програмният пакет ПРОЛОГ-16 е оригинална българска реализация на езика Пролог. Той включва интерпретатор на езика, библиотека мощни програми и набор демонстрационни програми. Налице е съвместимост с дефакто-стандарта на ПРОЛОГ, с което е осигурена висока степен на преносимост на програмите. В ПРОЛОГ-16 са интегрирани над 110 вградени предикати, реализирани са аритметични операции с цели и реални числа, графика, работа с прозорци. Развити са възможности за настройка на програмите. Съществува възможност за запазване имената на променливите след въвеждане на програмата, запазване историята на връзките между променливите, деструктивно задаване стойности на списъци, както и оригинални възможности за намиране на отговори на въпроси, съдържащи неуточнени релации.

4.1.7. ТРАНСЛАТОРИ, РАБОТЕЩИ ПОД УПРАВЛЕНИЕТО НА ДОС-16

Под управлението на ДОС-16 работят следните транслятори:

Макроасемблер [1]. Транслира (асемблира) програми, написани на асемблерски език. Има възможност за дефиниране на макроинструкции, сегментиране на програмите и дефиниране на модули. Позволява определяне на входни точки и външни имена за програмите.

Компилатор ФОРТРАН [14]. Транслира (компилира) програми, написани на езика ФОРТРАН, предназначени за решаване на широк кръг научно-технически и други задачи. Възможностите му са близки до тези на ФОРТРАН-77.

Компилатор С. Компилира програми, написани на езика С. Намира приложение при съставяне на системни програми. Удобен е за изготвяне на програми, преносими от една операционна система към друга или от една машина на друга. Езикът С позволява да се съставят програми с максимална ефективност, които по обем памет и бързодействие са близки до асемблерските. Използва се и като език от високо ниво, с което се повишава ефективността на труда на програмиста, който може да се абстрахира от подробностите, свързани с операционната система или машината.

Компилатор ПАСКАЛ [4]. Предназначен е за компилиране на програми на езика ПАСКАЛ, който е удобен за структурно програмиране. Към компилатора е включен собствен екранен редактор, свързващ редактор, система за зареждане и изпълнение на програмите, система за динамична настройка по време на компилирането и трасиране по време на изпълнението на програмата. В разширенията на езика се съдържат възможности за работа с променливи с абсолютни адреси, операции с битове и байтове данни, вмъкване на машинен код в програмата и др.

4.1.8. МНОГОПОТРЕБИТЕЛСКА ОПЕРАЦИОННА СРЕДА ПДОС-16

ПДОС-16 [22] е надстройка на операционната система ДОС-16, като я превръща в многозадачна и многопотребителска среда. Това означава, че е осигурена възможност за общ достъп до микрокомпютъра на няколко потребителя, като едновременно могат да се изпълняват няколко задачи. ПДОС-16 принадлежи към класа на операционните системи с времеделение.

ПДОС-16 позволява към персоналния компютър да се свързват до 8 терминала от типа на VT52, CM1604 и др. Освен стандартните терминали, които се поддържат, потребителят може да използва терминал, чиито параметри определя сам. Като терминали могат да се включват и персонални компютри. ПДОС-16 позволява разпределяне на паметта на микрокомпютъра на дялове. Във всеки дял се разполага програмата на една от задачите, изпълнявани в режим на времеделение. Всяка задача получава определен интервал от време (квант), в което работи със системните ресурси. Допустимо е някои задачи да получат няколко кванта време.

Системата позволява и дефиниране на приоритети и може да се използва за решаване на задачи, изискващи бърза реакция на външни събития. Връзката между задачите, изпълнявани едновременно в средата на ПДОС-16, се осъществява чрез общи области на паметта, които могат да бъдат защитени с примитиви, наречени *ресурси*. Те могат да служат за синхронизация на задачите.

В средата на ПДОС-16 е допустимо да работят програми, които записват данни директно във видеоконтролера на компютъра. Предвидена е системна програма за преработване на изпълнимите модули на приложните програми, за да се заменят адресите от видеоконтролера с тези на страницата, която се поддържа за всеки потребител.

В средата на ПДОС-16 могат да се изпълняват без преработка програмните продукти: COBOL, MULTIMATE, PFS, SUPERCALC II и III, PE, SYMPHONY, WORDSTAR, MULTIPLAN, LOTOS 1-2-3 и др.

Към системата е предвиден и терминален емулатор, който позволява включване на персонален компютър Правец-16 като терминал на главния компютър.

4.2. ПРИЛОЖНО ПРОГРАМНО ОСИГУРЯВАНЕ

4.2.1. МИКРОФАЙЛ/16

Програмната система МИКРОФАЙЛ/16 [17] съдържа средства за организация на релационна база от данни и език от високо ниво, насочен към съставяне на приложни програми. Системата притежава средства за генериране на отчети и възможност за връзка с други програмни системи. Системата МИКРОФАЙЛ/16 поддържа няколко типа файлове:

- файлове с данни, за определени области на приложение;
- индексни файлове, в които записите са подредени според стойностите на ключово поле;

- вторични текстови файлове, които съдържат текстова информация, свързана с определено поле на файла с данни;
- програмни файлове, които съдържат програми, написани на езика на МИКРОФАЙЛ/16;
- форматни файлове за управление на разположението на информацията върху екрана или печатащото устройство;
- файлове за потребителски отчети, които съхраняват данни за формата на отчета;
- етикетни файлове, които служат за съставяне на справки;
- изходни текстови файлове, които служат за връзка с други програмни продукти или за документирание на диалога с потребителя.

Език за програмиране

МИКРОФАЙЛ/16 използва символни, числови и логически променливи и променливи за съхраняване по датите.

Операторите върху данните във файловете могат да действат върху всички записи във файла или върху определени записи. Изразите са съставени от константи, променливи, аритметични и логически операции, операции за сравнение, операции за обработка на символни низове.

В езика се използват и множество функции, необходими за манипулация с данните извън базовите операции.

Операции с данните във файловете

Системата МИКРОФАЙЛ/16 притежава възможности за дефиниране и изменение на структурата на файл с данни. В диалогов режим или по програмен път може да се изменя дължината, типът и наименованието на отделните полета, да се добавят, вмъкват и изтриват определени полета в структурата на файла.

Манипулациите със записите във файл с данни са: добавяне, изменение и изтриване (временно или физическо). Файловете могат да се сортират по определен ключ или да се индексират така, че да бъдат достъпни по ключове, различни от този, по който са подредени. Достъп до определен запис може да се осъществи по номер, по определено условие в неиндексирания файл или по индекс в индексирания файл.

МИКРОФАЙЛ/16 позволява сумиране и осредняване на информацията от цифрови полета във файла с данни.

Особено важни за практиката са възможностите за свързване на данните в два файла чрез общо за тях ключово поле, като свързаният файл може да се индексира по това поле. От такъв характер е и функцията за смесване на два файла при подреждане на записите по определено условие. Файловете могат да се преименуват, изтриват и копират.

Създаване на потребителски отчети и справки

МИКРОФАЙЛ/16 притежава средства за дефиниране на формата на отчета. За целта се използват форматни файлове. В този файл се указват данни за заглавието на отчета, размер на страницата, структурата на файла, по който се съставя отчетът, полета за обобщаване. Файлът за формат на отчета може да се променя.

За съставяне на справки се използват етикетни файлове. В етикетния файл се задава информация за формата на етикета (текстова информация, която се извежда за заглавие на справка или за съдържанието ѝ). Създадените справки могат да се разпечатват или извеждат на монитора.

Съставяне на програмите в МИКРОФАЙЛ/16

МИКРОФАЙЛ/16 позволява работа в диалогов или програмен режим. При работа в диалогов режим с МИКРОФАЙЛ/16 операторът въвежда от клавиатурата команда, която се изпълнява от командния интерпретатор, след което операторът може да въведе следващата команда. За работа в програмния режим се използват програмните файлове. Програмните файлове се създават и редактират чрез вградения текстов редактор. С този редактор могат да се създават и редактират и форматните, и изходните файлове.

Програмите на МИКРОФАЙЛ/16 използват средства на структурното програмиране чрез процедури и елементи на езика за организация на цикли, обмен на информация с оператора и т.н.

В помощ на потребителя, който не е запознат с възможностите на МИКРОФАЙЛ/16, е предоставена диалогова система от менюта в програмата АСИСТЕНТ. Тя позволява анализ за наличие на грешки, документиране, съставяне на списъци на променливите, справочник на модулите, разпечатване и др.

Програмата АСИСТЕНТ се разпространява като самостоятелен програмен продукт [2].

4.2.2. СИСТЕМА ЗА ТЕКСТООБРАБОТКА ТЕКСТ-16

Програмната система ТЕКСТ-16 [28] е предназначена за създаване, редактиране, записване и отпечатване на текстови документи, отчети, делова кореспонденция, брошури, финансово-счетоводни форми и др.

Създаване на документ

Функциите на ТЕКСТ-16 за създаване на нов документ включват форматиране на екрана и операции по въвеждане на данните. С форматирането на екрана се задава форматът на реда — броят на символите в ред, разстоянието между два реда и табулация, както и форматът на страницата — брой редове в страница и т.н.

При въвеждане на текста на документите ТЕКСТ-16 осигурява автоматично пренасяне на нов ред, корекции с помощта на курсора върху екрана, бързо преместване в текста по реда, преместване върху екрана в рамките на страницата, изтриване на текст след засветяване, за да се избегне грешка при изтриването и др.

Редактиране на документ

За редактиране на документа ТЕКСТ-16 осигурява възможност за преместване на текст от едно място в документа на друго, копиране на даден текст на определени места в документа, копиране на друг доку-

мент, търсене на определена последователност от символи в текста, заместване на фрагмент от текста с друг, търсене на определен форматен ред, заместване на форматен ред, центриране на текст, отместване на текста в абзац, сливане на две съседни страници, поставяне и редакция на надпис над и под текст на страница, преформатиране на документ, въвеждане на индекси в текста, маркиране на текста, пренасяне по срички на нов ред (не автоматично).

Отпечатване на документ

При отпечатване на документ ТЕКСТ-16 позволява да се изберат параметрите на печата — начална страница, място на текста в страницата, изравняване отдясно, разстояние между редовете, избор на мащаб на символите. Възможно е по време на отпечатване на един документ да се редактира друг.

Управлението на печатащото устройство включва: еднократен печат, двукратен печат с отместване във височина, получен печат (двукратно без отместване), отместен печат (с отместване надясно), подчертаване на текст и др.

Работа с текстове и таблици

ТЕКСТ-16 дава възможност за създаване на документи с общ основен текст и различаващи се по част от текста (например по адресанта на писмото). При съставяне на документа се налага често да се повтарят думи и изрази. ТЕКСТ-16 позволява създаване и работа с библиотеки, в които могат да се използват готови модели на такива текстове.

Допълнителните операции на ТЕКСТ-16 с цели документи са: изтриване на документ, копиране или прехвърляне от една на друга дискета, отпечатване на резюме на документ, автоматично търсене на документ по ключови думи, промяна на името на документ.

Друга възможност на ТЕКСТ-16 е наборът операции за обобщаване и манипулиране с колони от числа, оформени в таблица, събиране по редове и колони, вмъкване, изтриване, преместване и копиране на колони.

За да се опрости работата при многократно повтарящи се действия на оператора, в ТЕКСТ-16 се използват процедурни файлове.

4.2.3. СУПЕРПЛАН

Пакетът СУПЕРПЛАН [8] е предназначен за решаване на широк кръг приложни задачи, свързани с обработка на таблици, в клетките на които могат да се записват числа, текст или формули.

Таблицата се помещава в паметта на персоналния компютър, а на монитора се показват *прозорци* от нея. С управляващи клавиши прозорецът може да се премества по таблицата.

Таблицата е правоъгълник, съставен от редове и колони. Ако в дадена позиция на таблицата е записан текст или число, те се извеждат на екрана, а ако там се намира формула, тя се изчислява и получената стойност се индицира на екрана. СУПЕРПЛАН дава възможност за въвежда-

не и редакция на данните в таблицата. Обръщението към дадена позиция в таблицата се състои от задаване номера на съответния ред или колона (абсолютно обръщение), задаване на отместването спрямо текущото положение на курсора (относително обръщение) или чрез обръщение по име.

Допустими са и сложни обръщения към други клетки -- пресичане на две обръщения, обединение на две обръщения или формиране на таблица от обръщения на клетки в двата срещуположни края на таблицата.

Таблиците могат да се извеждат или въвеждат от дисков файл. Предвидени са средства за преобразуване на таблиците -- изтичане, вмъкване, преместване и подреждане. Таблиците могат да се извеждат на печат.

4.2.4. ИНТЕГРИРАН ПАКЕТ МУЛТИПАК

МУЛТИПАК [19] е мощен многофункционален интегриран пакет за информационно обслужване. Пакетът съчетава възможностите за текстообработка с използване на база от данни, електронни таблици, графика и комуникация с други компютри или програмни системи.

Текстообработка

Текстообработката в МУЛТИПАК дава възможност за: въвеждане на текст с автоматично подреждане, форматиране, прилагане на различни шрифтове и формати на текста (подчертан, удебелен, курсив), автоматично изравняване вляво и вдясно, центриране и разпределение на части от текста, нанасяне на корекции, автоматично включване на готови части от текста, печат на няколко копия на документа, получаване на екрана на точния вид на отпечатвания документ. Описаните функции на пакета се активират чрез менюта или функционални клавиши.

С МУЛТИПАК могат да се използват: макроси за автоматично въвеждане на кратки текстове с натискане на съответен клавиш, автоматично въвеждане на заглавни и завършващи страницата надписи, автоматично номериране на страници, симетричен печат на 2 страници на лист, обработка на писма и формуляри, буфериране на изхода за печат.

Електронни таблици

МУЛТИПАК предоставя широки възможности за съставяне и изчисления на данни в таблици: построяване на големи по обем таблици, ограничени само от оперативната памет на компютъра; използване на различни формати на числа -- цели, с фиксирана точка, с порядък, представяне в търговски и валутен формат; наименуване на колони, редове и полета. Могат да се използват формули за връзка между колони, редове, полета в дадена таблица или между различни таблици, както и набор от аритметични и логически операции, специализирани функции за статистически изчисления или обработка на финансово-статистически данни. Възможно е също изчисляване стойността на поле по данни в други полета и автоматично преизчисляване на таблицата при изменение на данните.

Допълнително пакетът предлага някои възможности за манипулация с таблици, достъп до полета от таблицата по координати и логически имена, избор на части от таблицата и пренасяне в друга таблица или документи, работа с полетата от няколко таблици, генериране на графика с данните от таблиците, пренасяне на данни от база от данни в таблица; връзка с таблици, подготвени от LOTUS 1-2-3 или VISICALC.

База от данни

МУЛТИПАК предлага мощна по функционални възможности база от данни, но физически ограничена от оперативната памет на компютъра. Тя не е проектирана да замести големите бази от данни, а по-скоро да допълни възможностите им за бърз достъп при малък обем записи. Базата от данни на МУЛТИПАК позволява сортиране и търсене, избор, въвеждане и корекция на данни, отпечатване, прилагане на формули, аритметични операции за формиране съдържанието на всяко поле в записа.

Осигурена е връзка на базата от данни на МУЛТИПАК с dBASE-III или МИКРОФАЙЛ/16, с което тя може да се разглежда като надстройка на голямата база от данни. Ако файлът в dBASE-III има голям брой записи, той може да се обработва на части от МУЛТИПАК.

Графика

МУЛТИПАК дава възможност за графично изобразяване на данни във вид на стълбовидни, наложени, точкови, линейни, процентни (кръгови) или аналитични графични таблици. Запълващите знаци се избират автоматично, възможно е въвеждане на надписи на графиките, оразмеряването им, добавяне на една графична таблица към друга.

Стълбовидните графични таблици изобразяват данните от една или повече колони или редове на таблицата като вертикални стълбове (например обем на продукцията по години). Наложената графика прилича на стълбовидната, но в нея всеки стълб се състои от две или повече наложени части, отговарящи на съдържанието на данните в определен ред или колона от таблицата.

Кръговата графична таблица изобразява кръг, в който данните са представени като сектори, пропорционални на съдържанието на полето в таблицата.

Линейната графика се изобразява чрез начупена линия, точковата графика — чрез точки (без свързващи линии), а аналитичната (x-y графика) дава връзката между стойностите на две колони в таблицата и служи за визуална оценка на съотношението между тези колони.

Телекомуникация

Пакетът МУЛТИПАК дава възможност за синхронна телекомуникационна връзка с други компютри. Възможно е използването на комутируеми или некомутируеми линии, с управление на модеми или връзка без модеми, автоматично избиране, използване на пароли за връзка, пренасяне на файлове с различни стандартни протоколи.

4.2.5. ПОМОШНИ ПРОГРАМИ

Възможностите на ДОС се разширяват с някои помощни програми, които работят паралелно с основната програма и модифицират някои от функциите на ДОС.

Програма за смяна на кодовата таблица

Програмата КОИ8 служи за смяна на кодовата таблица от основната кодова таблица (МИК) на Правец-16 в код КОИ-8. Извиква се преди работата с някоя от системите за текстообработка за изменение на стандарта на кирилицата.

Програма за резервиране на информацията

Програмата ИНДИГО [10] служи за резервиране на информацията при работа с компютъра в случаите на: внезапно изключване на захранването, погрешна манипулация на оператора, препълване на диска (дискетата), ненормално прекратяване работата на активната програма.

След стартирането ѝ програмата ИНДИГО остава в паметта и след работата на оператора с компютъра. Тя позволява:

— автоматично съхраняване на въведените данни от оператора, което създава определени удобства при работа с програми за текстообработка, електронни таблици и други приложни програми.

— запомняне на всяко натискане върху клавиатурата и последващото му възпроизвеждане в реда, в който е работено. Възпроизвеждането може да бъде извършено бързо, бавно или знак по знак, което дава възможност за коригиране на действията;

— следене работата на максимум три програми, избрани от оператора (които най-често се използват от него) — в този случай самата приложна програма стартира и прекратява действието на програмата ИНДИГО;

— заключване с парола на екрана на монитора или клавиатурата на компютъра чрез съответната команда.

Програми за ускоряване на обмена с дискови устройства

Програмата ТУРБОДИСК [4] е за ускоряване на обмена със запомнящи устройства с магнитен диск чрез намаляване на обръщенията към външните устройства. Тя се извиква преди изпълнение на потребителската програма. ТУРБОДИСК използва буфер в паметта на компютъра, в който се поддържат копия на секторите на магнитния диск. При заявка за четене от диска програмата търси копие на необходимите сектори в буфера. Ако ги открие, данните се четат от буфера, т.е. отпада необходимостта от работа със значително по-бавното външно устройство. Ако приложната програма чете неколккратно едни и същи данни и те са налични в буфера, постига се намаляване на обръщенията към дисковите устройства.

Програмата е полезна и за анализ на входно-изходните операции, тъй като води статистика за броя на операциите четене и запис върху запомнящите устройства.

4.2.6. ПРОГРАМИ ЗА ОПТИМИЗАЦИЯ

Програмните продукти МИКРООПТИМА-16 [8] и ЛИНА-16 [9] предлагат няколко програми за оптимизация: безусловна минимизация, намиране на най-късо разстояние в мрежа, пренасяне на максимален поток с минималната стойност на извършените разходи и линейно програмиране.

Програма за безусловна минимизация

Програмата за безусловна минимизация намира екстремум на функцията от n независими променливи, върху които не са наложени никакви ограничения, т.е.

$$\min_{x \in R^n} f(x_1, \dots, x_n),$$

където R^n е n -мерно Евклидово пространство.

За решаване на тази задача се използва градиентният метод на Флетчер—Рийвс, известен като метод на спрегнатия градиент.

Градиентните методи обикновено притежават по-голяма скорост на сходимост в сравнение с директните. Предимството на метода на Флетчер и Рийвс е малкият обем използвана памет поради факта, че при изчисленията не се налага обръщане на матрицата. Ако целевата функция е многоекстремална, по този метод в общия случай се намира локален екстремум.

Програма за намиране на най-късо разстояние в мрежа

Голям брой реални проблеми могат да бъдат моделирани и решавани с математическия апарат на графи и потоци върху мрежи. Такива са например транспортните процеси, разпределителни и съобщителни системи, компютърни мрежи и т.н.

Едни от най-често срещаните фундаментални задачи в тази област са задачите за определяне на най-късите разстояния в мрежа (ориентиран или неориентиран граф, зададен с множество върхове и множество дъги). За множеството дъги е зададена функция на разстоянията, присвояваща реално число за всяка дъга.

Програмата за намиране на най-късите разстояния в мрежа използва три от най-ефективните известни алгоритми: на Дийкстра, на Моор, Белман, Д'Езопо, Пейп и на Флойд.

Алгоритъмът на Дийкстра се използва за определяне на най-късия път между зададен източник S и приемник T в мрежата — т.е. последователността на върховете по най-късото разстояние, както и числовата му стойност. В алгоритъма на Дийкстра времето за изпълнение на програмата расте квадратично с увеличаване броя на върховете в мрежата.

По метода на Моор, Белман, Д'Езопо и Пейп се определят най-късите пътища между източника S и всички останали върхове на мрежата. Този метод е подходящ за разпределени мрежи с голям брой върхове (в които не са дефинирани всички дъги). Алгоритъмът работи правилно дори когато някои от дъгите имат отрицателни стойности. Не може да се даде точна аналитична оценка на времето за изпълнение на алгори-

тъма, но голям брой практически експерименти показват, че този метод е един от най-бързите, ако огношението на броя на дъгите към броя на върховете е малко.

По метода на Флойд се определят най-късите разстояния между всички двойки върхове на мрежата. Времето на решението расте с квадрата на броя на върховете независимо от гъстотата на мрежата.

Програма за определяне на поток с минимална стойност

Съществуват практически задачи, в които е необходимо да се пренесе през зададена мрежа определен поток. При това всяка дъга на мрежата се характеризира с определена пропускателна способност относно максималното количество на потока, който тя може да пренесе. Заедно с това пренасянето на потока е свързано с определени за всяка дъга разходи. Задачата за пренасяне на поток с минимална стойност на разходите може да се формулира математически по следния начин:

$$\min \sum d_{ij} f_{ij}$$

където d_{ij} е цената за пренасяне на единица поток по дъгата (i, j) ,
 f_{ij} е величината на потока през дъгата (i, j) .

При това важат ограниченията

$$\sum f_{ij} - \sum f_{ji} = \begin{cases} q & \text{за } j = S \\ 0 & \text{за } j \neq i \\ -q & \text{за } j = T \end{cases}$$

$$\begin{aligned} 0 &\leq f_{ij} \leq c_{ij}; \\ 0 &\leq d_{ij} \end{aligned}$$

където S е източникът на потока, T е приемникът, q е величината на потока, c е пропускателната способност на дъгата (i, j) .

Така формулираната задача се нарича задача за определяне на поток с минимална стойност и може да бъде решена с алгоритъма на Бузакер и Тоуен.

Линейна интерактивна оптимизираща система ЛИНА-16

Интерактивната програмна система ЛИНА-16 [16] е предназначена за решаване и анализ на решението на общата задача на линейното програмиране — намиране минимум или максимум на линейна функция при линейни ограничения. В зависимост от стойностите, допустими за променливите на задачата, се решават следните подкласове задачи: задачи на линейното програмиране за реални непрекъснати стойности на променливите, задачи за двоично линейно програмиране за двоични стойности за променливите, задачи за смесено двоично линейно програмиране — с непрекъснати и двоични променливи, и задачи за параметрично програмиране. Предвидено е стъпково изпълнение на симплекс-алгоритъма, редактиране на данните, форматиране на екрана и др.

Този тип оптимизационни задачи най-често се срещат в приложни области, като планиране, проектиране, организация, управление, обучение и други.

Интерактивната система ЛИНА-16 има следните възможности:

- използване на файл HELP и режими HELP;
- използване на интерактивни режими за решаване на задачата или анализ на решението;

- решаване на правата и обратната задача на линейното програмиране;
- анализ на симплексната таблица и чувствителността на решението и др.

4.2.7. ПАКЕТ ПРИЛОЖНИ ПРОГРАМИ ЗА МАШИНОСТРОЕНЕТО

Този пакет [23] е предназначен за автоматизация на проектирането в машиностроенето и за решаване на редица задачи, свързани с въвеждане и обработка на графични фигури — архитектурни чертежи, планове за вътрешно оформяне, работни диаграми, художествена графика и др.

Основните функции на пакета са: въвеждане и корекция на чертежа, изобразяването му на графичния монитор, извеждане на плотер или графично печатащо устройство.

Пакетът работи с множество графични обекти, от които се съставя чертежът. Такива обекти са отсечки, окръжности, текст и др. Определянето на мястото на обектите върху чертежа става с командите на пакета.

Те се въвеждат от клавиатурата, от бутон върху таблета или дигитайзера, със светлинен молив или с мишка. След това, ако е необходимо, се определят стойностите на други параметри за избрания обект — например големина или ъгъл на завъртане, и обектът се извежда на монитора.

Дигитайзерът може да се приложи и за копиране на чертежи, начертани на хартия. Когато се използва мишка и тя се движи по таблета, курсорът проследява движението ѝ на екрана. За да се избере точката, на която е спрял курсорът, натиска се бутонът на мишката.

С писалката може да се посочи директно площ от екрана на монитора за въвеждане на точки. Курсорът се появява на екрана и следва писалката, докато се избере точката от екрана с натискане на бутона.

Пакетът може да работи с един или два монитора. При работа с два монитора единият служи за изобразяване на чертежа, а другият — за индициране на командите. При работа с един монитор командите се изобразяват в долната част на екрана. На екрана на графичния монитор се появява част от чертежа.

Изображението може да се мащабира, да се увеличи или намали видимата част от чертежа. Изображението на екрана може да се измества в произволна посока, така че да се работи с различни части на чертежа.

За въвеждане или редактиране на чертеж е предназначен графичният редактор на пакета. Той извежда на екрана чертежа и осигурява команди за създаване, модифициране, преобразуване и извеждане на копия върху плотера. Информацията за чертежа се запазва във файл на дисковото устройство на компютъра.

Различните части на чертежа могат да се разположат в отделни слоеве. Разделянето на слоеве позволява да се наблюдават различни части на чертежа поотделно или в комбинация. Например, ако единият слой на чертежа съдържа план на етаж на сграда, а електрическата и водопроводната инсталация са върху други слоеве, могат да се извадят копия върху хартия без или с инсталациите.

Пакетът предлага използването на прототипен чертеж, който съдържа например рамка и форма на заглавието, имена на слоевете, набор шрифтове за надписи и други. Достъпна е помощна информация за формата на отделните команди.

Разширенията на пакета предлагат полуавтоматично оразмеряване на чертежи, зашриховане на обекти, скициране на ръка чрез посочващо устройство.

Може да се създаде тримерна видимост, като някои обекти „изпъкнат“ спрямо останалите и чертежът се наблюдава от различни ъгли като триразмерен. Възможно е също така да се „скрият“ отсечките, които не трябва да се виждат.

4.2.8. СИСТЕМА ЗА ПРОЕКТИРАНЕ НА ПРИНЦИПНИ СХЕМИ

Системата за проектиране на принципни схеми [24] осигурява средства за въвеждане на данните за схемите и тяхното проектиране. Проектирането на схемите включва няколко етапа.

1. Създаване и съхраняване на символите, които се използват в чертежите на принципните схеми. Тези елементи на пакета се съхраняват в библиотеката на символите. Библиотеката съдържа най-често използваните стандартни символи — графическите означения на елементите — И, ИЛИ, тригери и др. Потребителят може да допълва библиотеката с нови символи или да изменя съществуващите. Символът може да бъде съставен от текст, прави линии, дъги, полуокръжности, окръжности и изводи, с които се манипулира за изчертаване на необходимия вид на символа. Символите се създават чрез функционални клавиши и функции, които се избират в менюто за управление на пакета и менюта за добавки.

2. Проектиране на принципната схема. След създаване и записване в библиотеката на необходимите символи се пристъпва към разполагане на символите в полето за проектиране на принципната схема. В този етап символите се разполагат на чертежа, който се извежда на екрана на монитора, и се въвеждат всички останали елементи — изводи, съединителни линии, текстове и други. След създаване на чертежа той може да се запише в дисков файл. След това този дисков файл може да се използва отново за редактиране или допълване с нови елементи.

В пакета е предвидена възможност за проектиране на принципни схеми на йерархични нива, т.е. допуска се разклоняване на един чертеж от чертежа на горно ниво в дървовидна структура. Така могат да се съставят големи чертежи, без да е необходимо да се изобразяват всички детайли на най-високото ниво.

3. Документиране на принципните схеми. След проектирането на принципна схема тя може да се изведе на плотер или да бъде генерирана на дисков файл информация за управление на плотер.

Към пакета е включена програма, която извежда списък на логическите елементи и връзките между тях. Списъкът осигурява данни, необходими за други системи за автоматизация на инженерната дейност. Тази програма позволява да се откриват грешки в чертежите.

4.2.9. ПАКЕТ ПРОГРАМИ ЗА ПРОЕКТИРАНЕ НА ПЕЧАТНИ ПЛАТКИ

Пакетът е предназначен за проектиране и документиране на топологията на печатните платки [25]. Съдържа няколко основни програмни системи: организация на библиотеки на компонентите, организация на библиотеки на елементите, система за проектиране на печатни платки, система за документиране на платката.

Библиотеката на компонентите съхранява данни за размерностите на елементите, които се разполагат на платката — номер на елемента, размери по x и y на елемента, брой на контактните площадки, координати на контактните площадки и др. Допустимо е добавяне на нови компоненти в библиотеката, корекции или изтриване на съществуващи компоненти.

Библиотеката на елементите на пакета съдържа данни за име на елемента, текстово описание, номер от библиотеката на компонентите, имена на сигналите за различните изводи, разположение на различните изводи, тип на контакта и др. Пакетът позволява поддържане на актуални библиотеки от потребителя с корекции и въвеждане на нови елементи в нея.

Проектирането на печатни платки включва следните фази:

1. Начална подготовка на данните. В тази фаза се задават данни за външния размер на платката. Компютърът се обръща към началните данни за печатните платки и към библиотеката на компонентите и елементите, за да създаде масив на елементите и съединенията.

2. Разполагане на компонентите на платката. В тази фаза проектантът разполага ръчно някои от елементите на определените места на чертежа на платката, а също задава местоположението на куплунгите и др. След това компонентите автоматично се разполагат на чертежа. Потребителят може да измени интерактивно схемата на съединяване на компонентите и да повтори автоматичното разположение на компонентите, докато се получи окончателната схема на платката.

3. Трасиране на съединенията. В тази фаза се извършва автоматично трасиране на съединенията на различните слоеве на печатната платка. След това е възможно да се проведе ръчно трасиране и корекция на неудачните съединения или да се фиксира местоположението на някои компоненти на места, избрани от проектанта, след което да се повторят фазите за автоматично разполагане и трасиране.

След приключване на проектирането на печатната платка, пакетът дава възможност за извеждане на чертежа на платката върху плотер, създаване на файл за управление на фотоплотер, създаване на файл за системата за пробиване на отворите върху изработената печатна платка или за системата за автоматично подреждане на елементите върху платките.

Пакетът осигурява връзка със системата за проектиране на принципни схеми.

4.2.10. КОМПИЛАТОР ЗА РЕЛАЦИОННА БАЗА ОТ ДАННИ

Компиляторът [13] дава възможност за компилиране на програми, разработени на езика на МИКРОФАЙЛ/16. Той позволява многократно

ускоряване на изпълнението на приложните програми, както и намаляване на обема на заетата оперативна памет. Има всички възможности за самостоятелна работа без използване на интерпретатора и другите мощни средства на МИКРОФАЙЛ/16. В компилатора са включени и средства за изпълнение на програми, изискващи достъп до мрежови ресурси в локална мрежа с шинна топология.

Компилаторът осигурява и редица функционални предимства за потребителя в сравнение с интерпретатора: в структурата на релационната таблица могат да се включват по-голям брой полета, съществува възможност за свързване по ключ на една таблица с няколко други. Практически е неограничен броят на активните независими променливи.

Възможно е дефиниране на потребителски функции, обръщение към външни модули, написани на Асемблер или С, както и предоставяне на помощна информация за потребителя посредством функционални клавиши. Съществуват удобни команди за очертаване на прозорци върху екрана, както и допълнителни възможности за форматиране.

4.2.11. УНИВЕРСАЛНА ГРАФИЧНА СИСТЕМА УНИГРАФ

Пакетът програми УНИГРАФ [38] е предназначен за създаване на делова графика, съхраняване на графични приложения като метафайл в персонален компютър, мултипликативни ефекти и др.

Програмният продукт дава възможност за работа с библиотечни подпрограми, които се извикват от главната програма и са написани на езика С или на БЕЙСИК.

УНИГРАФ съдържа 4 различни системи — ядро на графичната система, графична файлова система, система за построяване на диаграми и система за разработка на графични програми. Програмният продукт е изпълнен на базата на стандарта GKS на ISO (ANSI) и гарантира преносимост на програмите.

4.2.12. ПАКЕТИ ПРИЛОЖНИ ПРОГРАМИ ЗА КОНКРЕТНИ ПРИЛОЖЕНИЯ

Създадени са и се предлагат на потребителя различни пакети приложни програми (ППП) с практическа насоченост [39,40].

ППП Експерт

ППП ЕКСПЕРТ [39] е предназначен да освободи счетоводителя от рутинните операции, свързани с изчисления и проверката им, като му даде възможност да насочи вниманието си към анализа на резултатите от своята работа. Основните функции на пакета са: въвеждане на първичните документи за счетоводна дейност в персоналния компютър, въвеждане на синтетична и аналитична информация, извеждане на справки, поддържане на номенклатури.

При въвеждане на първичните документи счетоводителят дава информация за месец, пачка, номер на документа, дата на мемориалния ордер, вид на документа, номер на документа и дата на създаване.

Въвеждането на синтетичната информация включва въвеждане на номерата на сметките в дебита и кредита и стойностите, с които се заверяват или задължават. При това данните се контролират за: въвеждане на сметка само от определения специфичен сметкоплан, равняване на дебита с кредита, червени (отрицателни) записвания, балансови или зад-балансови кореспонденции.

След въвеждане на синтетичната информация ЕКСПЕРТ автоматично преминава към въвеждане на аналитична информация за сметките с аналитични партиди. Дадена е възможност по конкретна сметка да се въведат произволен брой партиди до изчерпване на сумата, записана по време на синтетичното въвеждане.

При въвеждането се контролира валидността на цифровата информация, съпадението с номенклатурните кодове, валидността на датите в документите, показват се задължителните полета.

ЕКСПЕРТ дава възможност да се изведат справки от синтетичната и аналитична информация, съхранявани в системата. Справките са два вида: с постоянни и с променливи анетки.

С постоянна анетка са справки, като оборотна ведомост, баланс и др. С променлива анетка са справките, които се получават за сметки по аналитични партиди. При извеждане на информация за конкретни сметки могат да се зададат един, няколко или всички показатели, по които се води сметката, както и редът на подреждане (групиране, междинни сборове), в който е необходимо да се изведе справката.

ЕКСПЕРТ позволява съхраняване и актуализация на справки по номенклатурите, участващи в описанието на всяка сметка.

Подобни функции при автоматизацията на финансово-счетоводната дейност има и пакетът СТРАНД.

ППП Норма—справка

Програмният продукт НОРМА-СПРАВКА е предназначен да обслужва информационната система НОРМА за действащото законодателство. Пакетът дава възможност да се визуализират каталозите с нормативни актове по раздели и токове и да се избере конкретен раздел или том. След избор на нормативния акт чрез функционалните клавиши потребителят може да проследява текста. Възможно е позициониране на курсора на определен текст — член, линия, точка, буква, определена дума или фраза от нормативния документ.

ППП Омега

Програмният продукт ОМЕГА е предназначен за обработка на заплати и тяхното осчетоводяване.

ОМЕГА дава възможност за: счетоводно отчитане на работната заплата в предприятия с повременно заплащане на труда; извеждане на необходимите документи, свързани с отчитане на работната заплата и справките за БНБ, ДСК и др.; извеждане на кадрови справки по различни признаци; пълен контрол на данните.

Програмният продукт включва: поддържане на база от данни с кадрова и счетоводна информация за служителите, поддържане на архив за

произволен период от време, възможност за бърза и лека настройка на продукта към конкретните особености на предприятието.

ППП Контрол на решенията

ППП КОНТРОЛ НА РЕШЕНИЯТА е предназначен за автоматизиране на дейността на контролните служби в предприятия, ведомства и др.

Този пакет осигурява: регистрация на решенията от оперативки, документи и др., следене на процеса на изпълнение; получаване на справки по срокове и превантивен контрол; възможност за търсене по различни критерии; средства за защита и съхранение на информацията за определен период.

Пакетът съдържа модул за регистрация и актуализация, модул за селекция на данни и модул за създаване и поддържане на номенклатурна база.

ППП Ръководител

Пакетът **РЪКОВОДИТЕЛ** е предназначен за организация на автоматизирано работно място на ръководителя или на неговия секретар.

Пакетът осигурява средства за поддържане и получаване на информация с нормативно-справочен, планов, отчетен, ретроспективен и прогнозен характер; средства за разработка на разпоредителна документация и формиране на задания по заложените в тях решения, издаване на справки и напомнания, отчет за изпълнението на заданията, средства за водене на личен електронен бележник—календар за планиране на мероприятията и разпределяне на личното време, средства за водене на кореспонденция, включващи текстообработка, поддържане на архив на документите, поддържане на адресни списъци.

ППП Технокад

ППП ТЕХНОКАД е универсална система за автоматизация на проектирането в областта на машиностроенето, електрониката, архитектурата и строителството, както и за автоматизация на изготвяне на различни шрифтове и надписи.

Пакетът поддържа библиотека от стандартни символи, както и символи и изображения, създадени от потребителя.

Функционалните възможности на пакета са следните: проектиране на различни нива (до 255), шриховане на области от чертежи, като се използват до 40 вида линии, автоматично нанасяне на размери, построяване на координатна мрежа, преместване и въртене на изображението на детайли и др.

На базата на пакета **ТЕХНОКАД** са разработени програмите **КОНСИС**, **ВАЛ** и **ГЪРБИЦА**.

Системата **КОНСИС** осигурява разширение на информационните средства на програмния продукт **ТЕХНОКАД** в насока за създаване на **САПР**.

КОНСИС дава възможност за изграждане на главно меню и подменю за възли и детайли. Всеки тип детайли или възел може да бъде опре-

делен с набор параметри. Предвидена е възможност за търсене на файлове на чертежи по зададен набор признаци, извеждане на информация за чертежите по типове детайли или възли, поддържане на пароли и др.

Системата ВАЛ е универсална програма за проектиране на прави валове за машиностроенето.

Системата дава възможност за изчисляване на реакциите на опорите, изчисляване на идеалните форми, изчисляване на лагерите, определяне на конструктивната форма на ВАЛ, изчисляване за якост на вала, определяне на размерите и др.

Програмата ГЪРБИЦА е предназначена за автоматизирано проектиране на равнинни гърбични механизми в областта на машиностроенето. С нея могат да бъдат определени профилът на гърбицата, еквидистантната крива, диаметрите на ролката и други геометрични размери.

Информационна система ИНФОС

ИНФОС е универсална информационно-търсеща система, създадена за използване от потребители, които не притежават специални познания в областта на електронната обработка на данните.

ИНФОС дава възможност за създаване на 32 информационни бази от данни, съдържащи голям обем документална и текстова информация — отчети, извадки от статии, договори, технически проекти, каталози и т.н. Системата може да поддържа архив на създадените информационни бази.

Каталогът на ИНФОС съдържа описания на логическата структура на документите в съществуващите информационни бази от данни. Системата за управление на базите от данни на ИНФОС дава възможност за работа на много потребители. Системата допуска дефиниране на структурата на базата от данни, създаване на базата и актуализация на данните. Помощните програми на ИНФОС позволяват сливане на бази от данни, реорганизация на базите и др.

Системата за избор на ИНФОС предлага възможности за намиране на документи по зададени имена, данни, стойности, кодове и други чрез използване на релационни оператори.

Системата за търсене използва четири нива: документ, параграф, изречение и фраза, като търсенето на всяко ниво е в пълния обем на текста.

Системата ИНФОС намира приложение в библиотеки, архиви, информационни центрове, патентни бюра, проектантски организации, поликлиники, болници и други учреждения, в които се оперира с голям брой документи.

ППП Интересексперт

Системата ИНТЕРЕКСПЕРТ е предназначена да даде възможност на потребителя да разработва експертни системи за широк кръг приложения — информационно-търсещи системи, системи за автоматизация на управлението и др.

Пакетът притежава набор от правила за описване на експертни системи и база от данни с таблична структура, съчетани с език от високо

ниво, за опериране на данните в базата, електронни таблици и системи за графично изображение на цифрова информация.

Системата поддържа псевдоестествен език, включващ около 500 български, руски или английски думи.

Всички програмни средства на ИНТЕРЕКСПЕРТ са достъпни за потребителя чрез меню, съчетано със системата за справочна информация (HELP).

4.2.13. МРЕЖОВО ПРОГРАМНО ОСИГУРЯВАНЕ

Като надстройка над ДОС-16 за персоналните компютри Правец-16 е разработено мрежово програмно осигуряване на локалните мрежи МИКРОЛИМ и МИКРОРИНГ.

Във всяка локална мрежа се включват няколко персонални компютъра, които могат да се разделят по функции на две групи — работни станции и компютри за обслужване на достъпа до файловете (файлови процесори или сървери). Обикновено станциите имат само ЗУГМД, а файловете процесори имат дискове с голям капацитет в ЗУТМД.

Програмното осигуряване на локална мрежа МИКРОЛИМ [3,33,34] включва многозадачна система, която позволява едновременното изпълнение на заявки за вход—изход към диска, за комуникация с мрежата и работа на потребителя с приложна програма.

Предаването и приемането на пакети от съобщения по мрежата се извършва от набор програми по протоколи XNS, съгласно 7-слоиния модел на ISO/OSI.

Мрежовото програмно осигуряване съдържа файлова система, която позволява едновременно обръщение на няколко процеса към диска. МИКРОЛИМ предлага следните програмни продукти, които често се наричат *услуги* на мрежата:

Файл и Печат. Позволяват на потребителя да използва общите поддиректории на мрежовия процесор чрез стандартните функции на ДОС-16, отваряне на файлове, пренасочване на устройства и спублиране на данните при печат върху мрежовите печатащи устройства.

Име. Дава възможност за централизирано обслужване на каталог от имена, общи за работните станции и мрежовите процесори. С тези услуги администраторът на мрежата добавя и изважда потребители от мрежата, определя техните права за достъп до общите ресурси.

Маршрут. Определя възможностите за свързване на няколко мрежи.

Връзка. Дава възможност за свързване по асинхронна комутируема линия на отдалечена работна станция към мрежата или за мост към други типове мрежи.

Поща. Допуска обмен на съобщения и файлове с данни между потребителите в мрежата.

Старт. Използва се за отдалечено стартиране от мрежовия процесор на работата на станция без зареждане на програмите от дискета или от твърд диск.

Глобална мрежа. Служи за връзка на локалната мрежа с мрежа от големи ЕИМ тип SNA.

ЛИТЕРАТУРА КЪМ ГЛ. 4

1. Асемблер для ДОСПК. Описание языка. ЦИИТТ—София. 1.А0А2.00602-01.
2. АСИСТЕНТ. Ръководство за потребителя. Микросистеми—София. 1.А0А2.01753-01.
3. Базово програмно осигуряване на локални мрежи с шинна топология. Ръководство за програмиста. ИМПТ—София. 1.А0А3.02044-01.
4. Быстродействующий компилятор Паскаль для ДОСПК. Описание языка. ЦИИТТ—София. 1.А0А2.00820-01.
5. ДОС-16. Ръководство за потребителя. КМПТ—Правец, Софтуерна къща „Програма“. 1.А0А3.01311-01.
6. Екранен редактор за ПДОС-16. Ръководство за оператора. ИМПТ—София. 1.А0А3.01931-01.
7. Електронна поща за локална мрежа с шинна топология. Ръководство за програмиста. ИМПТ—София.
8. Електронна таблица СУПЕРПЛАН. Софтуерна къща „Програма“—Правец, ОАПС—Благоевград.
9. Эмуляторы для ЕС1831/ЕС1832 в ОССС. Руководство оператора. ЦИИТТ—София. 1.А0А2.01366-01.
10. ИНДИГО. Ръководство за потребителя. Микросистеми — София. 1.А0А2.01424-01.
11. Интерпретатор БЕЙСИК. Ръководство за потребителя. Софтуерна къща „Програма“—Правец, КМПТ—Правец. 1.А0А3.01310-01.
12. Каталог на програмни продукти. Корпорация ППС СК НППФ.
13. Компилятор за релационна база данни. Ръководство за програмиста. ИМПТ—София. 1.А0А3.02105-01.
14. Компилятор Фортран под управлением ДОСПК. Описание языка. ЦИИТТ—София. 1.А0А2.00707-01.
15. Комуникационна система за файлов трансфер и терминална емуляция. Описание на приложението. Ръководство за програмиста. ИМПТ—София. 1.А0А3.01973-01.
16. ЛИНА-16. Линейна интерактивна оптимизираща система. Ръководство за програмиста. ИТКР—БАН, София, КМПТ—Правец. 1.А0А2.01134-01.
17. МИКРОФАЙЛ/16. Команди и функции. Микросистеми—София. 1.А0А3.01370-01.
18. МИКРОФАЙЛ/16. Ръководство за потребителя. Микросистеми—София. 1.А0А3.01372-01.
19. МУЛТИПАК. Ръководство за потребителя. ИТКР—БАН, София. 1.А0А3.00891-01.
20. Описание на ППП „Експерт“. ГИИЦ на Министерство на търговията. 1.А0А3.01370-01.
21. Операционна система ДОС-16. Ръководство за потребителя. Софтуерна къща „Програма“—Правец, КМПТ—Правец. 1.А0А3.01311-01.
22. ПДОС-16. Ръководство за програмиста. ИМПТ—София. 1.А0А3.01311-02.
23. Пакет приложни програми за машиностроенето на база ЕС 1832. Ръководство за оператора. ЦИИТТ—София. 1.А0А3.01375-01.
24. Пакет приложни програми за проектиране на принципни схеми. Ръководство за оператора. ИМПТ—София. 1.А0А3.1944-01.
25. Пакет программ для синхронной коммуникации для ДОСПК. Руководство оператора. ЦИИТТ—София. 1.А0А3.00917-01.
26. Програмна система за текстообработка. ТЕКСТ-16. Ръководство за програмиста. ИТКР—БАН, София, КМПТ—Правец.
27. Програми за оптимизация. Ръководство за програмиста. ИТКР—БАН—София, КМПТ—Правец. 1.А0А2.01134-01.
28. Пакет приложни програми за проектиране на печатни платки. Ръководство за оператора. ИМПТ—София. 1.А0А3.1945-01.
29. Програмна система за отдалечена междумрежова комуникация за локални мрежи. Ръководство за програмиста. ИМПТ—София.
30. Програмна система за специализиран терминален процесор за връзка на локални мрежи с шинна топология с ЕС ЕИМ. Ръководство за програмиста. ИМПТ—София.
31. Програмно осигуряване на нискоскоростна локална мрежа с кръгова топология за Правец-8М. Ръководство за програмиста. ИМПТ—София. 1.А097.02018-01.

32. Програмно осигуряване на нискоскоростна локална мрежа с кръгова топология за Правец-16. Ръководство за програмиста. ИМПТ—София. 1.А0А2.02019-01
33. Програмно осигуряване за мрежов процесор — персонален компютър за управление на достъпа до мрежови ресурси на локални мрежи с шинна топология. Ръководство за програмиста. ИМПТ—София. 1.А0А3.02031-01.
34. Програмно осигуряване на работна станция за управление на достъпа до мрежови ресурси за локални мрежи с шинна топология. Ръководство за програмиста. ИМПТ—София. 1.А0А3.02030-01.
35. Програмно осигуряване на мрежов процесор за управление на локална мрежа с шинна топология. Ръководство за програмиста. ИМПТ—София.
36. Релационна СУБД с мрежови функции. Описание на езика. ИМПТ—София.
37. ТУРБОДИСК. Ръководство за потребителя. Микросистеми—София. 1.А0А2.01423-01.
38. Универсальная графическая система для ЕС1832—УНИГРАФ. Руководство программиста. ЦИИТТ—София. 1.А0А3.01894-01.

Приложение 1


Набор инструкции на микропроцесора CM630

Мнемоничен код	Действие	Видове адресация	Запис на асемблерски език	МЦ	Код	Флагове			
						Б	75543210 NV BDIZC		
1	2	3	4	5	6	7	8		
ADC	A + M + C → A, C (4), (1)	IMD	ADC # oper	2	69	2	NV...ZC		
		ZP	ADC oper	3	65	2			
		ZP, X	ADC oper, X	4	75	2			
		ABS	ADC oper	4	6D	3			
		ABS, X	ADC oper, X	4(1)	7D	3			
		ABS, Y	ADC oper, Y	4(1)	79	3			
		(IND, X)	ADC (oper, X)	6	61	2			
		(IND), Y	ADC (oper), Y	5(1)	71	2			
AND	A ∧ M → A	IMD	AND # oper	2	29	2	N...Z		
		ZP	AND oper	3	25	2			
		ZP, X	AND oper, X	4	35	2			
		ABS	AND oper	4	2D	3			
		ABS, X	AND oper, X	4(1)	3D	3			
		ABS, Y	AND oper, Y	4(1)	39	3			
		(IND, X)	AND (oper, X)	6	21	2			
		(IND), Y	AND (oper), Y	5(1)	31	2			
ASL		ACC	ASL A	2	0A	1	N...ZC		
		ZP	ASL oper	5	06	2			
		ZP, X	ASL oper, X	6	16	2			
		ABS	ASL oper	6	0E	3			
		ABS, X	ASL oper, X	7	1E	3			
BCC	преход при C = 0 (2)	REL	BCC oper	2 2+1 2+2	90	2		
		BCS	преход при C = 1 (2)	REL	BCS oper	2 2+1 2+2	B0	2
BIT	A ∧ M M7 → N M6 → V	ZP	BIT oper	3	24	2	M6M7...Z		
		ABS	BIT oper	4	2C	3			
BMI	преход при N ≠ 1 (2)	REL	BMI oper	2 2+1 2+2	30	2		
		BNE	преход при Z = 0 (2)	REL	BNE oper	2 2+1 2+2	D0	2

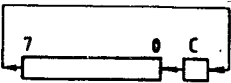
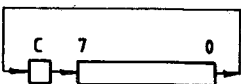
Продължение на приложение 1

1	2	3	4	5	6	7	8
BRK	програмно прекъсване	IMPL	BRK	7	00	1	...1....
BVC	преход при V=0	REL	BVC <i>oper</i>	2 2+1 2+2	50	2
BVS	преход при V=1	REL	BVS <i>oper</i>	2 2+1 2+2	70	2
CLC	0→C	IMPL	CLC	2	18	10
CLD	0→D	IMPL	CLD	2	D8	10..
CLI	0→I	IMPL	CLI	2	58	10..
CLV	0→V	IMPL	CLV	2	B8	1	.0.....
CMP	A→M	IMD	CMP # <i>oper</i>	2	C9	2	N.....ZC
		ZP	CMP <i>oper</i>	3	C5	2	
		ZP,X	CMP <i>oper</i> ,X	4	D5	2	
		ABS	CMP <i>oper</i>	4	CD	3	
		ABS,X	CMP <i>oper</i> ,X	4(1)	DD	3	
		ABS,Y	CMP <i>oper</i> ,Y	4(1)	D9	3	
		(IND,X)	CMP (<i>oper</i> ,X)	6	C1	2	
		(IND),Y	CMP (<i>oper</i>),Y	5(1)	D1	2	
CPX	X↔M	IMD	CPX # <i>oper</i>	2	E0	2	N.....ZC
		ZP	CPX <i>oper</i>	3	E4	2	
		ABS	CPX <i>oper</i>	4	EC	3	
CPY	Y↔M	IMD	CPY # <i>oper</i>	2	C0	2	N.....ZC
		ZP	CPY <i>oper</i>	3	C4	2	
		ABS	CPY <i>oper</i>	4	CC	3	
DEC	M-1→M	ZP	DEC <i>oper</i>	5	C6	2	N.....Z.
		ZP,X	DEC <i>oper</i> ,X	6	C5	2	
		ABS	DEC <i>oper</i>	6	CE	3	
		ABS,X	DEC <i>oper</i> ,X	7	DE	3	
DEX	X-1→X	IMPL	DEX	2	CA	1	N.....Z.
DEY	Y-1→Y	IMPL	DEY	2	88	1	N.....Z.
EOR	A.XOR.M→A	IMD	EOR # <i>oper</i>	2	49	2	N.....Z.
		ZP	EOR <i>oper</i>	3	45	2	
		ZP,X	EOR <i>oper</i> ,X	4	55	2	
		ABS	EOR <i>oper</i>	4	4D	3	
		ABS,X	EOR <i>oper</i> ,X	4(1)	5D	3	
		ABS,Y	EOR <i>oper</i> ,Y	4(1)	59	3	
		(IND,X)	EOR (<i>oper</i> ,X)	6	41	2	
		(IND),Y	EOR (<i>oper</i>),Y	5(1)	51	2	
INC	M+1→M	ZP	INC <i>oper</i>	5	E6	2	N.....Z.
		ZP,X	NC <i>oper</i> ,X	6	F6	2	
		ABS	INC <i>oper</i>	6	EE	3	
		ABS,X	INX <i>oper</i> ,X	7	FE	3	

Продължение на приложение 1

1	2	3	4	5	6	7	8
INX	$X+1 \rightarrow X$	IMPL	INX	2	E8	1	N.....Z.
INY	$Y+1 \rightarrow Y$	IMPL	INY	2	C8	1	N.....Z.
JMP?	$(PC+1) \rightarrow PCL$	ABS	JMP <i>oper</i>	3	4C	3
	$(PC+2) \rightarrow PCH$	IND	JMP (<i>oper</i>)	5	6C	3	
JSR	$PC \leftarrow (PC)+2$	ABS	JSR <i>oper</i>	6	20	3
	$MS \leftarrow (PCH)$						
	$S \leftarrow S - 1$						
	$MS \leftarrow (PCL)$						
	$S \leftarrow S - 1$						
	$PCL \leftarrow ADL$						
	$PCH \leftarrow ADH$						
LDA	M → A	IMD	LDA # <i>oper</i>	2	A9	2	N.....Z.
		ZP	LDA <i>oper</i>	3	A5	2	
		ZP.X	LDA <i>oper.X</i>	4	B5	2	
		ABS	LDA <i>oper</i>	4	AD	3	
		ABS.X	LDA <i>oper.X</i>	4(1)	BD	3	
		ABS.Y	LDA <i>oper.Y</i>	4(1)	B9	3	
		(IND.X)	LDA (<i>oper.X</i>)	6	A1	2	
		(IND).Y	LDA (<i>oper</i>).Y	5(1)	B1	2	
LDX	M → X	IMD	LDX # <i>oper</i>	2	A2	2	N.....Z.
		ZP	LDX <i>oper</i>	3	A6	2	
		ZP.Y	LDX <i>oper.Y</i>	4	B6	2	
		ABS	LDX <i>oper</i>	4	AE	3	
		ABS.Y	LDX <i>oper.Y</i>	4(1)	BE	3	
LDY	M → Y	IMD	LDY # <i>oper</i>	2	A0	2	N.....Z.
		ZP	LDY <i>oper</i>	3	A4	2	
		ZP.Y	LDY <i>oper.Y</i>	4	B4	2	
		ABS	LDY <i>oper</i>	4	AC	3	
		ABS.Y	LDY <i>oper.Y</i>	4(1)	BC	3	
LSR		ACC	LSR A	2	4A	1	0.....ZC
		ZP	LSR <i>oper</i>	5	46	2	
		ZP.X	LSR <i>oper.X</i>	6	56	2	
		ABS	LSR <i>oper</i>	6	4E	3	
		ABS.X	LSR <i>oper.X</i>	7	5E	3	
NOP	$PC = PC + 1$	IMPL	NOP	2	EA	1
ORA	A.OR.M → A	IMD	ORA # <i>oper</i>	2	09	2	N.....Z.
		ZP	ORA <i>oper</i>	3	05	2	
		ZP.X	ORA <i>oper.X</i>	4	15	2	
		ABS	ORA <i>oper</i>	4	0D	3	
		ABS.X	ORA <i>oper.X</i>	4(1)	1D	3	
		ABS.Y	ORA <i>oper.Y</i>	4(1)	19	3	
		(IND.X)	ORA (<i>oper.X</i>)	6	01	2	
		(IND).Y	ORA (<i>oper</i>).Y	5(1)	11	2	
PHA	$A \rightarrow MS$ $S-1 \rightarrow S$	IMPL	PHA	3	48	1
PHP	$P \rightarrow MS$ $S-1 \rightarrow S$	IMPL	PHP	3	08	1
PLA	$S+1 \rightarrow S$ $MS \rightarrow A$	IMPL	PLA	4	68	1

Продължение на приложение 1

1	2	3	4	5	6	7	8
PLP	S+1→S MS→P	IMPL	PLP	4	28	1	от стека
ROL		ACC ZP ZP,X ABS ABS,X	ROL A ROL <i>oper</i> ROL <i>oper,X</i> ROL <i>oper</i> ROL <i>oper,X</i>	2 5 6 6 7	2A 26 36 2E 3E	1 2 2 3 3	N.....ZC
							
ROR		ACC ZP ZP,X ABS ABS,X	ROR A ROR <i>oper</i> ROR <i>oper,X</i> ROR <i>oper</i> ROR <i>oper,X</i>	2 5 6 6 7	6A 66 76 6E 7E	1 2 2 3 3	N.....ZC
							
RTI	S+1→S MS→P S+1→S MS→PCL S+1→S MS→PCH	IMPL	RTI	6	40	1	от стека
RTS	S+1→S MS→PCL S+1→S MS→PCH PC+1→PC	IMPL	RTS	6	60	1
SBC	A-M-C→A,C	IMD ZP ZP,X ABS ABS,X ABS,Y (IND,X) (IND),Y	SBC # <i>oper</i> SBC <i>oper</i> SBC <i>oper,X</i> SBC <i>oper</i> SBC <i>oper,X</i> SBC <i>oper,Y</i> SBC (<i>oper,X</i>) SBC (<i>oper</i>),Y	2 3 4 4 4(1) 4(1) 6 5(1)	E9 E5 F5 ED FD F9 E1 F1	2 2 2 3 3 3 2 2	N.....Z(3)
SEC	1→C	IMPL	SEC	2	38	11
SED	1→D	IMPL	SED	2	F8	11...
SEI	1→I	IMPL	SEI	2	78	11..
STA	A→M	ZP ZP,X ABS ABS,X ABS,Y (IND,X) (IND),Y	STA <i>oper</i> STA <i>oper,X</i> STA <i>oper</i> STA <i>oper,X</i> STA <i>oper,Y</i> STA (<i>oper,X</i>) STA (<i>oper</i>),Y	2 4 4 5 5 6 6	85 95 8D 9D 99 81 91	2 2 3 3 3 2 2
STX	X→M	ZP ZP,Y ABS	STX <i>oper</i> STX <i>oper,Y</i> STX <i>oper</i>	3 4 4	86 96 8E	2 2 3
STY	Y→M	ZP ZP,X ABS	STY <i>oper</i> STY <i>oper,X</i> STY <i>oper</i>	2 4 4	84 94 8C	2 2 3

1	2	3	4	5	6	7	8
TAX	A→X	IMPL	TAX	2	AA	1	N.....Z.
TAY	A→Y	IMPL	TAY	2	A8	1	N.....Z.
TSX	S→X	IMPL	TSX	2	BA	1	N.....Z.
TXA	X→A	IMPL	TXA	2	8A	1	N.....Z.
TXS	X→S	IMPL	TXS	2	9A	1	N.....Z.
TYA	Y→A	IMPL	TYA	2	98	1	N.....Z.

Пояснения към символите и съкращенията

A — акумулатор

X — индексен регистър X

Y — индексен регистър Y

P — регистър на състоянието на микропроцесора

S — указател на стека

PC — програмния брояч

PCH — старши байт на програмния брояч

PCL — младши байт на програмния брояч

M — клетка от паметта, определена от действителния адрес

MS — клетка от паметта, чиито адрес се намира в указателя на стека

M6 — бит 6 от клетка на паметта

M7 — бит 7 от клетка на паметта

oper — операция

. — този флаг от регистъра P не се променя

+ — събиране

- — изваждане

.AND. или \wedge — логическо И.OR. или \vee — логическо ИЛИ

.XOR. — изключващо ИЛИ

→ — прехвърляне от

← — прехвърляне в

↔ — сравнение (равнозначност)

Б — брой байтове

МЦ — брой машинни цикли

IMD — непосредствена адресация

ABS — абсолютна адресация

ZP — адресация в нулевата страница

IMPL — по разбираща се адресация

ACC — акумулаторна адресация

ABS.X — индексна абсолютна адресация с регистъра X

ABS.Y — индексна абсолютна адресация с регистъра Y

ZP.X — индексна адресация в нулевата страница с регистъра X

ZP.Y — индексна адресация в нулевата страница с регистъра Y

REL — относителна адресация

IND — абсолютна косвена адресация

(IND.X) — косвена адресация с предварително индексирание по регистъра X

(IND).Y — косвена адресация с последователно индексирание по регистъра Y

(1) — към МЦ се прибавя единица, ако се излезе извън границата на страницата

(2) — за инструкциите, използващи относителния начин за адресация, са валидни следните стойности на МЦ:

2 ако не се извърши преход;

2 + 1 ако преходът се извърши в същата страница;

2 + 2 ако преходът се извърши в друга страница.

(3) — флагът C е инверсия е заем.

(4) — ако флагът Z не е валиден в десетичен режим, акумулаторът трябва да бъде проверен за нулев резултат.

Набор инструкции на микропроцесора I8080

Мисмоничен код	Описание	Код	Брой цикли
1	2	3	4
MOV d,s	Преместване от регистър в регистър	01DDSSSS	5
MOV M,r	Преместване от регистър в паметта	01110SSS	7
MOV r,M	Преместване от паметта в регистър	01DDD110	7
MVI r	Директно зареждане в регистър	00DDD110	7
MVI M	Директно зареждане в паметта	00110110	10
LXI B	Директно зареждане в двойка BC	00000001 01	10
LXI D	Директно зареждане в двойка DE	00010001 11	10
LXI H	Директно зареждане в двойка HL	00100001 21	10
STAX B	Записване на акумулатора в паметта по адрес в двойка BC	00000010 02	7
STAX D	Записване на акумулатора в паметта по адрес в двойка DE	00010010 12	7
LDAX B	Зареждане на акумулатора от паметта по адрес в двойка BC	00001010 0A	7
LDAX D	Зареждане на акумулатора от паметта по адрес в двойка DE	00011010 1A	7
STA	Записване на акумулатора в паметта директно	00110010 32	13
LDA	Зареждане на акумулатора от паметта директно	00111010 3A	13
SHLD	Записване на двойка HL в паметта директно	00100010 22	16
LHLD	Зареждане на двойка HL от паметта директно	00101010 2A	16
XCHG	Разменяне на двойки HL и DE	11101011 EB	4
PUSH B	Записване в стека на двойка BC	11000101 C5	11
PUSH D	Записване в стека на двойка DE	11010101 D5	11
PUSH H	Записване в стека на двойка HL	11100101 E5	11
PUSH PSW	Записване в стека на акумулатора и флаговете	11110101 F5	11
POP B	Извличане от стека на двойка BC	11000001 C1	10
POP D	Извличане от стека на двойка DE	11010001 D1	10
POP H	Извличане от стека на двойка HL	11100001 E1	10
POP PSW	Извличане от стека на акумулатора и флаговете	11110001 F1	10
XTHL	Разменяне на върха на стека и двойка HL	11100011 E3	18
SPHL	Преместване на двойка HL в указателя на стека	11111001 F9	5
LXI SP	Директно зареждане в указателя на стека	00110001 31	10
INX SP	Увеличаване на указателя на стека с 1	00110011 33	5
DCX SP	Намаляване на указателя на стека с 1	00111011 3B	5
JMP	Безусловен преход	11000011 C3	10
JC	Преход, когато има пренос	11011010 DA	10
JNC	Преход, когато няма пренос	11010010 D2	10
JZ	Преход при резултат, равен на 0	11001010 CA	10
JNZ	Преход при резултат, различен от 0	11000010 C2	10
JP	Преход при положителен резултат	11110010 F2	10
JM	Преход при отрицателен резултат	11111010 FA	10
JPE	Преход при четен резултат	11101010 EA	10
JPO	Преход при нечетен резултат	11100010 E2	10
PCHL	Зареждане на програмния брояч от двойка HL	11101001 E9	5
CALL	Извикване на подпрограма	11001101 CD	17
CC	Извикване на подпрограма, когато има пренос	11011100 DC	11/17
CNC	Извикване на подпрограма, когато няма пренос	11010100 D4	11/17

Продължение на приложение 2

1	2	3	4
CZ	Извикване на подпрограма при резултат, равен на 0	11001100 CC	11/17
CNZ	Извикване на подпрограма при резултат, различен от 0	11000100 C4	11/17
CP	Извикване на подпрограма при положителен резултат	11110100 F4	11/17
CM	Извикване на подпрограма при отрицателен резултат	11111100 FC	11/17
CPE	Извикване на подпрограма при четен резултат	11101100 EC	11/17
CPO	Извикване на подпрограма при нечетен резултат	11100100 E4	11/17
RET	Възврат	11001001 C9	10
RC	Възврат, когато има пренос	11011000 D8	5/11
RNC	Възврат, когато няма пренос	11010000 D0	5/11
RZ	Възврат при резултат, равен на 0	11001000 C8	5/11
RNZ	Възврат при резултат, различен от 0	11000000 C0	5/11
RP	Възврат при положителен резултат	11110000 F0	5/11
RM	Възврат при отрицателен резултат	11111000 F8	5/11
RPE	Възврат при четен резултат	11101000 E8	5/11
RPO	Възврат при нечетен резултат	11100000 E0	5/11
RST	Рестартиране — преход към адрес 0,8,16,24,32,40,48 или 56	11AAA111	11
INR	Увеличаване на регистър с 1	00DDD100	5
DCRr	Намаляваляване на регистър с 1	00DDD101	5
INR M	Увеличаване на клетка от паметта с 1	00110100 34	10
DCR M	Намаляване на клетка от паметта с 1	00110101 35	10
INX B	Увеличаване на двойка BC с 1	00000011 03	5
INX D	Увеличаване на двойка DE с 1	00010011 13	5
INX H	Увеличаване на двойка HL с 1	00100011 23	5
DCX B	Намаляване на двойка BC с 1	00001011 0B	5
DCX D	Намаляване на двойка DE с 1	00011011 1B	5
DCX H	Намаляване на двойка HL с 1	00101011 2B	5
ADD r	Прибавяне на регистър към акумулатора	10000SSS	4
ADC r	Прибавяне на регистър към акумулатора с пренос	10001SSS	4
ADD M	Прибавяне на клетка от паметта към акумулатора	10000110 86	7
ADC M	Прибавяне на клетка от паметта към акумулатора с пренос	10001110 8E	7
ADI	Директно прибавяне към акумулатора	11000110 C6	7
ACI	Директно прибавяне към акумулатора с пренос	11001110 CE	7
DAD B	Прибавяне на двойка BC към двойка HL	00001001 09	10
DAD D	Прибавяне на двойка DE към двойка HL	00011001 19	10
DAD H	Прибавяне на двойка HL към двойка HL	00101001 29	10
DAD SP	Прибавяне на указателя на стека към двойка HL	00111001 39	10
SUB r	Изваждане на регистър от акумулатора	10010SSS	4
SBC r	Изваждане на регистър от акумулатора със заем	10011SSS	4
SUB M	Изваждане на клетка от паметта от акумулатора	10010110 96	7
SBB M	Изваждане на клетка от паметта от акумулатора със заем	10011110 9E	7
SUI	Директно изваждане от акумулатора	11010110 D6	7
SBI	Директно изваждане от акумулатора със заем	11011110 DE	7

Продължение на приложение 2

1	2	3	4
ANA r	Логическо умножение на регистър и акумулатора	10100SSS	4
XRA r	Сума по модул 2 на регистър и акумулатора	10101SSS	4
ORA r	Логическо събиране на регистър и акумулатора	10110SSS	4
CMP r	Сравнение на регистър и акумулатор	10111SSS	4
ANA M	Логическо умножение на клетка от паметта и акумулатора	10100110 A6	7
XRA M	Сума по модул 2 на клетка от паметта и акумулатора	10101110 AE	7
ORA M	Логическо събиране на клетка от паметта и акумулатора	10110110 B6	7
CMP M	Сравнение на клетка от паметта и акумулатора	10111110 BE	7
ANI	Логическо умножение на акумулатора с число	11100110 E6	7
XRI	Сума по модул 2 на акумулатора и число	11101110 EE	7
ORI	Логическо събиране на акумулатора с число	11110110 F6	7
CPI	Сравнение на акумулатора с число	11111110 FE	7
RLC	Ротация наляво на акумулатора	00000111 07	4
RRC	Ротация надясно на акумулатора	00001111 0F	4
RAL	Ротация наляво на акумулатора през флага за пренос	00010111 17	4
RAR	Ротация надясно на акумулатора през флага за пренос	00011111 1F	4
CMA	Инвертиране на акумулатора	00101111 2F	4
STC	Установяване на флага за пренос в 1	00110111 37	4
CMC	Инвертиране на флага за пренос	00111111 3F	4
DAA	Десетична корекция на акумулатора	00100111 27	4
IN	Четене на входен порт в акумулатора	11011011 DB	10
OUT	Запис на акумулатора в изходен порт	11010011 D3	10
EI	Разрешение за прекъсване	11111011 FB	4
DI	Забрана на прекъсване	11110011 F3	4
NOP	Празен оператор	00000000 00	4
HLT	Стоп на процесора	01110110 76	7

където

DDD или SSS	Регистър	AAA	Адрес
000	B	000	00H
001	C	001	08H
010	D	010	10H
011	E	011	18H
100	H	100	20H
101	L	101	28H
110	памет	110	30H
111	A	111	38H

Набор инструкции на микропроцесора Z80

Използвани означения

- n — 8-битова стойност от 0 до 255
- nn — 16-битова стойност от 0 до 65535
- r — регистър на процесора: A, B, C, D, E, H, L
- s — 8-битов операнд: r,n,(HL),(IX+d),(IY+d)
- dd,ss,pp,qq,rr — 16-битов операнд: BC,DE,HL,IX,IY,SP,AF
- IFF — тригер за разрешаване на прекъсване

Флагове

- C или CY(Carry) — C=1, ако при операцията възниква пренос или заем в старшият бит на резултата
- Z (Zero) — Z=1, ако резултатът от операцията е 0
- S (Sign) — S=1, ако старшият бит на резултата е 1
- P (Parity или Overflow) — логическите операции обръщат този флаг в зависимост от четността на резултата — P=1 при четен резултат. Аритметичните операции обръщат този флаг в 1 при препълване.
- H (Half-carry) — използва се за десетична корекция (DAA) — H=1, ако има пренос или заем в бит 4 на акумулатора.
- N (Add/subtract) — използва се за десетична корекция (DAA) — N=1, ако предишната операция е изваждане

Състояние на флаговете след операцията

- . — флагът не се изменя от операцията
- 0 — флагът се установява в 0 от операцията
- 1 — флагът се установява в 1 от операцията
- ↑ — флагът се установява в зависимост от резултата от операцията
- × — флагът не се контролира от операцията — може да е 0 или 1
- P — флаг P се установява в зависимост от четността на резултата
- V — флаг P се установява при препълване на резултата

1. Инструкции за преместване на 8-битови данни

Мнемоничен код	Операция	Флагове SZ-H-PNC	Код 76 543 210	Брой байтове		Брой цикли	
				5	6	М	Т
1	2	3	4	5	6	7	
LD r,s	r←s	..×.×...	01 r s	1	1	4	
LD r,n	r←n	..×.×...	00 r 110	2	2	7	
LD r,(HL)	r←(HL)	..×.×...	01 r 110	1	2	7	
LD r,(IX+d)	r←(IX+d)	..×.×...	11 011 101 DD	3	5	19	
			01 r 110				
LD r,(IY+d)	r←(IY+d)	..×.×...	11 111 101 FD	3	5	19	
			01 r 110				
			d				
LD (HL),r	(HL)←r	..×.×...	01 110 r	1	2	7	
LD (IX+d),r	(IX+d)←r	..×.×...	11 011 101 DD	3	5	19	
			01 110 r				
			d				
LD (IY+d),r	(IY+d)←r	..×.×...	11 111 101 FD	3	5	19	
			01 110 r				
			d				
LD (HL),n	(HL)←n	..×.×...	00 110 110 36	2	3	10	
			n				

Продължение на приложение

1	2	3	4	5	6	7
LD (IX+d),n	(IX+d)←n	.. × .. × ...	11 011 101 DD 00 110 110 36 d n	4	5	19
LD (IY+d),n	(IY+d)←n	.. × .. × ...	11 111 101 FD 00 110 110 36 d n	4	5	19
LD A,(BC)	A←(BC)	.. × .. × ...	00 001 010 0A	1	2	7
LD A,(DE)	A←(DE)	.. × .. × ...	00 011 010 1A	1	2	7
LD A,(nn)	A←(nn)	.. × .. × ...	00 111 010 3A n	3	4	13
LD (BC),A	(BC)←A	.. × .. × ...	00 000 010 02 n	1	2	7
LD (DE),A	(DE)←A	.. × .. × ...	00 010 010 12 n	1	2	7
LD (nn),A	(nn)←A	.. × .. × ...	00 110 010 32 n	3	4	13
LD A,I	A←I	↑↑ × 0 × *0,	11 101 101 ED 01 010 111 57	2	2	9
LD A,R	A←R	↑↑ × 0 × *0,	11 101 101 ED 01 011 111 5F	2	2	9
LD I,A	I←A	.. × .. × ...	11 101 101 ED 01 000 111 47	2	2	9
LD R,A	R←A	.. × .. × ...	11 101 101 ED 01 001 111 4F	2	2	9

КЪДЕТО

r,s	регистър	* флаг P = IFF (тригер за разрешаване на прекъсване)
000	B	
001	C	
010	D	
011	E	
100	H	
101	L	
111	A	

2. Инструкции за преместване на 16-битови данни

Мнемоничен код	Операция	Флаговے SZ-H-PNC	Код 76 543 210	Брой байтове	Брой пикли M	Брой пикли I
1	2	3	4	5	6	7
LD dd,nn	dd←nn	.. × .. × ...	00 dd0 001 n	3	3	10
LD IX,nn	IX←nn	.. × .. × ...	11 011 101 DD 00 100 001 21 n	4	4	14
LD IY,nn	IY←nn	.. × .. × ...	11 111 101 FD 00 100 001 21 n	4	4	14

Продължение на приложение 3

1	2	3	4	5	6	7
LD HL,(nn)	H←(nn+1) L←(nn)	..×.×...	00 101 010 2A n n	3	5	16
LD dd,(nn)	dd _H ←(nn+1) dd _L ←(nn)	..×.×...	11 101 101 ED 01 dd1 011 n n	4	6	20
LD IX,(nn)	IX _H ←(nn+1) IX _L ←(nn)	..×.×...	11 011 101 DD 00 101 010 2A n n	4	6	20
LD IY,(nn)	IY _H ←(nn+1) IY _L ←(nn)	..×.×...	11 111 101 FD 00 101 010 2A n n	4	6	20
LD (nn),HL	(nn+1)←H (nn)←L	..×.×...	00 100 010 22 n n	3	5	16
LD (nn),dd	(nn+1)←dd _H (nn)←dd _L	..×.×...	11 101 101 ED 01 dd0 011 n n	4	6	20
LD (nn),IX	(nn+1)←IX _H (nn)←IX _L	..×.×...	11 011 101 DD 00 100 010 22 n n	4	6	20
LD (nn),IY	(nn+1)←IY _H (nn)←IY _L	..×.×...	11 111 101 FD 00 100 010 22 n n	4	6	20
LD SP,HL	SP←HL	..×.×...	11 111 001 F9	1	1	6
LD SP,IX	SP←IX	..×.×...	11 011 101 DD 11 111 001 F9	2	2	10
LD SP,IY	SP←IY	..×.×...	11 111 101 FD 11 111 001 F9	2	2	10
PUSH qq	(SP-2)←qq _L (SP-1)←qq _H	..×.×...	11 qq0 101	1	3	11
PUSH IX	(SP-2)←IX _L (SP-1)←IX _H	..×.×...	11 011 101 DD 11 100 101 E5	2	4	15
PUSH IY	(SP-2)←IY _L (SP-1)←IY _H	..×.×...	11 111 101 FD 11 100 101 E5	2	4	15
POP qq	qq _H ←(SP+1) qq _L ←(SP)	..×.×...	11 qq0 001	1	3	10
POP IX	IX _H ←(SP+1) IX _L ←(SP)	..×.×...	11 011 101 DD 11 100 001 E1	2	4	14
POP IY	IY _H ←(SP+1) IY _L ←(SP)	..×.×...	11 111 101 FD 11 100 001 E1	2	4	14

където

dd	двойка	qq	двойка
00	BC	00	BC
01	DE	01	DE
10	HL	10	HL
11	SP	11	AF

3. Инструкции за разменяне на данни и за преместване на блокове от данни

Мнемоничен код	Операция	Флагове SZ-H-PNC	Код 76 543 210	Брой байтове		Брой шестки	
				5	6	7	8
1	2	3	4	5	6	7	8
EX DE,HL	DE \leftrightarrow HL	..x.x...	11 101 011 EB	1	1	4	
EX AF,AF'	AF \leftrightarrow AF'	..x.x...	00 001 000 08	1	1	4	
EXX ¹	BC \leftrightarrow BC'	..x.x...	11 011 001 D9	1	1	4	
	DE \leftrightarrow DE'						
	HL \leftrightarrow HL'						
EX (SP), HL	H \leftrightarrow (SP+1) L \leftrightarrow (S)	..x.x...	11 100 011 E3	1	5	19	
EX (SP),IX	IXH \leftrightarrow (SP+1)	..x.x...	11 011 101 DD	2	6	23	
	IXL \leftrightarrow (SP)		11 100 011 E3				
EX (SP),IY	IYH \leftrightarrow (SP+1)	..x.x...	11 111 101 FD	2	6	23	
	IYL \leftrightarrow (SP)		11 100 011 E3				
LDI	(DE) \leftarrow (HL)	..x0x*0.	11 101 101 ED	2	4	16	
	DE \leftarrow DE+1		10 100 000 A0				
	HL \leftarrow HL+1						
	BC \leftarrow BC-1						
LDIR	(DE) \leftarrow (HL)	..x0x00.	11 101 101 ED	2	5	21	
	DE \leftarrow DE+1			(за BC \neq 0)	2	4	16
	HL \leftarrow HL+1		10 110 000 B0				
	BC \leftarrow BC-1			(за BC=0)			
LDD	изпълнява се до BC=0 (DE) \leftarrow (HL)	..x0x*0.	11 101 101 ED	2	4	16	
	DE \leftarrow DE-1		10 101 000 A8				
	HL \leftarrow HL-1						
	BC \leftarrow BC-1						
LDDR	изпълнява се до BC=0 (DE) \leftarrow (HL)	..x0x00.	11 101 101 ED	2	5	21	
	DE \leftarrow DE-1			(за BC \neq 0)	2	4	16
	HL \leftarrow HL-1		10 111 000 B8				
	BC \leftarrow BC-1			(за BC=0)			
CPI	изпълнява се до BC=0 A=(HL)	$\uparrow^* \times \uparrow^* \times^* 1.$	11 101 101 ED	2	4	16	
	HL \leftarrow HL+1		10 100 001 A1				
	BC \leftarrow BC-1						
CPIR	изпълнява се до BC=0 A=(HL)	$\uparrow^* \times \uparrow^* \times^* 1.$	11 101 101 FD	2	5	21	
	HL \leftarrow HL+1			BC \neq 0 \wedge A \neq (HL)	2	4	16
	BC \leftarrow BC-1		10 110 001 B1				
CPD	изпълнява се до BC=0 или A=(HL) A=(HL)	$\uparrow^* \times \uparrow^* \times^* 1.$	11 101 101 ED	2	4	16	
	HL \leftarrow HL-1		10 101 001 A9				
	BC \leftarrow BC-1						
CPDR	изпълнява се до BC=0 или A=(HL) A=(HL)	$\uparrow^* \times \uparrow^* \times^* 1.$	11 101 101 ED	2	5	21	
	HL \leftarrow HL-1			BC \neq 0 \wedge A \neq (HL)	2	4	16
	BC \leftarrow BC-1		10 111 001 B9				
	изпълнява се до BC=0 или A=(HL)			BC=0 \vee A=(HL)			

където

* флаг P=0 при BC=1=0, иначе P=1

флаг Z=1 при A=(HL), иначе Z=0

¹ Разменяне на двете банки регистри.

4. Инструкции за аритметични и логически операции с 8-битови данни

Мнемоничен код	Операция	Флагове SZ-H-PNC	Код 76 543 210	Брой байтове	Брой цикли	
					М	Т
ADD A,r	$A \leftarrow A + r$	$\uparrow\uparrow \times \uparrow \times V0\uparrow$	10 000 r	1	1	4
ADD A,n	$A \leftarrow A + n$	$\uparrow\uparrow \times \uparrow \times V0\uparrow$	11 000 110 n	2	2	7
ADD A,(HL)	$A \leftarrow A + (HL)$	$\uparrow\uparrow \times \uparrow \times V0\uparrow$	10 000 110	1	2	7
ADD A,(IX+d)	$A \leftarrow A + (IX + d)$	$\uparrow\uparrow \times \uparrow \times V0\uparrow$	11 011 101 DD 10 000 110 d	3	5	19
ADD A,(IY+d)	$A \leftarrow A + (IY + d)$	$\uparrow\uparrow \times \uparrow \times V0\uparrow$	11 111 101 FD 10 000 110 d	3	5	19
ADC A,s ¹	$A \leftarrow A + s + CY$	$\uparrow\uparrow \times \uparrow \times V0\uparrow$	001			
SUB s	$A \leftarrow A - s$	$\uparrow\uparrow \times \uparrow \times V0\uparrow$	010			
SBC A,s	$A \leftarrow A - s - CY$	$\uparrow\uparrow \times \uparrow \times V0\uparrow$	011			
AND s	$A \leftarrow A \wedge s$	$\uparrow\uparrow \times 1 \times P00$	100			
OR s	$A \leftarrow A \vee s$	$\uparrow\uparrow \times 0 \times P00$	110			
XOR s	$A \leftarrow A \oplus s$	$\uparrow\uparrow \times 0 \times P00$	101			
CP s	$A \leftarrow A - s$	$\uparrow\uparrow \times \uparrow \times V1\uparrow$	111			
INC r	$r \leftarrow r + 1$	$\uparrow\uparrow \times \uparrow \times V0.$	00 r 100	1	1	4
INC (HL)	$(HL) \leftarrow (HL) + 1$	$\uparrow\uparrow \times \uparrow \times V0.$	00 110 100	1	3	11
INC (IX+d)	$(IX + d) \leftarrow (IX + d) + 1$	$\uparrow\uparrow \times \uparrow \times V0.$	11 011 101 DD 00 110 100 d	3	6	23
INC (IY+d)	$(IY + d) \leftarrow (IY + d) + 1$	$\uparrow\uparrow \times \uparrow \times V0.$	11 111 101 FD 00 110 100 d	3	6	23
DEC s ²		$\uparrow\uparrow \times \uparrow \times V1.$	101			

където

г	регистър
000	B
001	C
010	D
011	E
100	H
101	L
111	A

¹ Кодирането на инструкциите и броят на циклите са както за инструкция ADD. Трите бита се поставят на мястото на означените. s е 8-битов операнд: r,n,(HL), (IX+d), (IY+d).

² Кодирането на инструкцията и броят на циклите са както за инструкция INC. Трите бита се поставят на мястото на означените. s е 8-битов операнд: r, (HL), (IX+d), (IY+d).

5. Инструкции за специфични операции и за управление

Мнемоничен код	Операция	Флагове SZ-H-PNC	Код 76 543 210	Брой байтове	Брой цикли	
					М	Т
DAA	десетична корекция на акумулатора	$\uparrow\uparrow \times \uparrow \times P.\uparrow$	00 100 111 27	1	1	4
CPL ¹	$A \leftarrow \bar{A}$	$.. \times 1 \times .1.$	00 101 111 2F	1	1	4
NEG ²	$A \leftarrow \bar{A} + 1$	$\uparrow\uparrow \times \uparrow \times VI\uparrow$	11 101 101 ED 01 000 100 44	2	2	8
CCF ³	$CY \leftarrow \bar{CY}$	$.. \times \times \times .0\uparrow$	00 111 111 3F	1	1	4
SCF ⁴	$CY \leftarrow 1$	$.. \times 0 \times .01$	00 110 111 37	1	1	4
NOP	празен оператор	$.. \times . \times \dots$	00 000 000 00	1	1	4
HALT	стоп	$.. \times . \times \dots$	01 110 110 76	1	1	4
DI	IFF ← 0	$.. \times . \times \dots$	11 110 011 F3	1	1	4
EI	IFF ← 1	$.. \times . \times \dots$	11 111 011 FB	1	1	4
IM 0		$.. \times . \times \dots$	11 101 101 ED 01 000 110 46	2	2	8
IM 1		$.. \times . \times \dots$	11 101 101 ED 01 010 110 56	2	2	8
IM 2		$.. \times . \times \dots$	11 101 101 ED 01 011 110 5E	2	2	8

¹ Обратен код на акумулатора.

² Допълнителен код на акумулатора.

³ Инвертиране на флаг C.

⁴ Установяване на флаг C.

6. Инструкции за аритметични операции с 16-битови данни

Мнемоничен код	Операция	Флагове SZ-H-PNC	Код 76 543 210	Брой байтове	Брой цикли	
					М	Т
1	2	3	4	5	6	7
ADD HL,ss	$HL \leftarrow HL + ss$	$.. \times \times \times .0\uparrow$	00 ss1 101	1	3	11
ADC HL,ss	$HL \leftarrow HL + ss + CY$	$\uparrow\uparrow \times \times \times V0\uparrow$	11 101 101 ED 01 ss1 010	2	4	15
SBC HL,ss	$HL \leftarrow HL - ss - CY$	$\uparrow\uparrow \times \times \times V1\uparrow$	11 101 101 ED 01 ss0 010	2	4	15
ADD IX,pp	$IX \leftarrow IX + pp$	$.. \times \times \times .0\uparrow$	11 011 101 DD 00 ppl 001	2	4	15
ADD IY,rr	$IY \leftarrow IY + rr$	$.. \times \times \times .0\uparrow$	11 111 101 FD 00 rr1 001	2	4	15

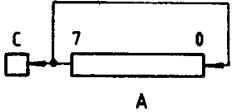
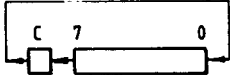
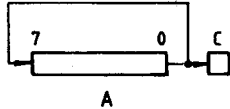
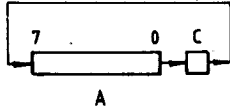
1	2	3	4	5	6	7
INC ss	ss ← ss + 1	.. × . × ..	00 ss0 011	1	1	6
INC IX	IX ← IX + 1	.. × . × ..	11 011 101 DD 00 100 011 23	2	2	10
INC IY	IY ← IY + 1	.. × . × ..	11 111 101 FD 00 100 011 23	2	2	10
DEC ss	ss ← ss - 1	.. × . × ..	00 ss1 011	1	1	6
DEC IX	IX ← IX - 1	.. × . × ..	11 011 101 DD 00 101 011 2B	2	2	10
DEC IY	IY ← IY - 1	.. × . × ..	11 111 101 FD 00 101 011 2B	2	2	10

КЪДЕТО

ss	двойка	pp	двойка	rr	двойка
00	BC	00	BC	00	BC
01	DE	01	DE	01	DE
10	HL	10	IX	10	IY
11	SP	11	SP	11	SP

7. Инструкции за изместване и ротация

Мнемоничен код	Операция	Флаговe SZ-H-PNC	Код 76 543 210	Брой байтове		Брой пикли	
				М	Т		
1	2	3	4	5	6	7	

RLCA		.. × 0 × . 0 ↓	00 000 111 07	1	1	4
RLA		.. × 0 × . 0 ↓	00 010 111 17	1	1	4
RRCA		.. × 0 × . 0 ↓	00 001 111 0F	1	1	4
RRA		.. × 0 × . 0 ↓	00 011 111 1F	1	1	4

Продължение на приложение 3

1	2	3	4	5	6	7
RLC r		$\updownarrow \times 0 \times P0 \updownarrow$	11 001 011 CB 00 000 r	2	2	8
RLC (HL)		$\updownarrow \times 0 \times P0 \updownarrow$	11 001 011 CB 00 000 110	2	2	8
RLC (IX+d)		$\updownarrow \times 0 \times P0 \updownarrow$	11 011 101 DD 11 001 011 CB d 00 000 110	4	6	23
RLC (IY+d)		$\updownarrow \times 0 \times P0 \updownarrow$	11 111 101 FD 11 001 011 CB d 00 000 110	4	6	23
RL s ^l		$\updownarrow \times 0 \times P0 \updownarrow$	010			
RRC s		$\updownarrow \times 0 \times P0 \updownarrow$	011			
RR s		$\updownarrow \times 0 \times P0 \updownarrow$	001			
SLA s		$\updownarrow \times 0 \times P0 \updownarrow$	100			
SRA s		$\updownarrow \times 0 \times P0 \updownarrow$	101			
SRL s		$\updownarrow \times 0 \times P0 \updownarrow$	111			

1	2	3	4	5	6	7	
RLD ²			$\updownarrow \times 0 \times P0.$	11 101 101 ED 01 101 111 6F	2	5	18
RRD			$\updownarrow \times 0 \times P0.$	11 101 101 ED 01 100 111 67	2	5	18

където

r	регистър
000	B
001	C
010	D
011	E
100	H
101	L
111	A

¹ Кодиратето на инструкциите и броят на циклите са както за инструкцията RLC. Трите бита се поставят на мястото на означените, s е 8-битов операнд: r, (HL), (IX+d), (IY+d).

² Ротация на полубайт между акумулатора и клетката от паметта, адресирана от регистри HL.

8. Инструкции за тестване, нулиране и установяване на битове

Мнемоничен код	Операция	Флагове SZ-IF-PC	Код 76 543 210	Брой байтове		Брой цикли	
				М	И	М	И
1	2	3	4	5	6	7	8
BIT b,r	Z←b(r)	$\times \updownarrow \times 1 \times \times 0.$	11 001 011 CB 01 b r	2	2	2	8
BIT b,(HL)	Z←b(HL)	$\times \updownarrow \times 1 \times \times 0.$	11 001 011 CB 01 b 110	2	3	3	12
BIT b,(IX+d)	Z←b(IX+d)	$\times \updownarrow \times 1 \times \times 0.$	11 011 101 DD 11 001 011 CB d	4	5	5	20
BIT b,(IY+d)	Z←b(IY+d)	$\times \updownarrow \times 1 \times \times 0.$	01 b 110 11 111 101 FD 11 001 011 CB d	4	5	5	20
SET b,r	b(r)←1	...x...x...	11 001 011 CB 11 b r	2	2	2	8
SET b,(HL)	b(HL)←1	...x...x...	11 001 011 CB 11 b 110	2	4	4	15
SET b,(IX+d)	b(IX+d)←1	...x...x...	11 011 101 DD 11 001 011 CB d	4	6	6	23
			11 b 110				

Продължение на приложение

1	2	3	4	5	6	7
SET b,(Y+d)	b(Y+d)←1	.. × . × ...	11 111 101 FD	4	6	23
			11 001 011 CB			
			d			
RES b,s ¹	b(s)←0	.. × . × ...	11 b 110			
			10			

където

г	регистър	b	бит
000	B	000	0
001	C	001	1
010	D	010	2
011	E	011	3
100	H	100	4
101	L	101	5
111	A	110	6
		111	7

¹ Кодирването на инструкцията и броят на циклите са както за инструкцията SET. Двата бита се поставят на мястото на означените: s е 8-битов оператор: г.(HL),(IX+d), (Y+d).

9. Инструкции за предаване на управлението

Мнемоничен код	Операция	Флагове SZ-H-PNC	Код 76 543 210	Брой битове	Брой цикли	
					M	I
JP nn	PC←nn	.. × . × ...	11 000 011 C3	3	3	10
			n			
JP cc,nn	изпълнено условие с: PC←nn	.. × . × ...	11 cc 011 n	3	3	10
JR e ¹	PC←PC+e	.. × . × ...	00 011 000 18	2	3	12
			e-2			
JR c,e	изпълнено условие с: PC←PC+e	.. × . × ...	00 1 c 011 e-2	2	2	7
			e-2	2	3	12
JP (HL)	PC←HL	.. × . × ...	11 101 001 E9	1	1	4
JP (IX)	PC←IX	.. × . × ...	11 011 101 DD	2	2	8
			11 101 001 E9			
JP (IY)	PC←IY	.. × . × ...	11 111 101 FD	2	2	8
			11 101 001 E9			
DJNZ e	B←B-1 докато B стане 0: PC←PC+e	.. × . × ...	00 010 000 10	2	2	8
			e-2		B=0	
				2	3	13
					B≠0	

където

cc	условие
000	NZ - резултат, различен от нула
001	Z - резултат, равен на нула
010	NC - невъзникване на пренос или заем
011	C - възникване на пренос или заем
100	PO - печетен резултат от операцията
101	PE - четен резултат от операцията
110	P - положителен резултат от операция
111	M - отрицателен резултат от операция

с	условие
00	NZ — резултат, различен от нула
01	Z — резултат, равен на нула
10	NC — невъзникване на пренос или заем
11	C — възникване на пренос или заем
¹ е	— число със знак между -126 и +129

10. Инструкции за обръщение към подпрограма и за връщане от подпрограма

Мнемоничен код	Операция	Ф.дан.ове SZ- H-PNC	Код 76 543 210	Брой байтове	Брой цикли	
					М	Т
CALL nn	(SP-1)←PC _H	.. × . × ...	11 001 101 CD	3	5	17
	(SP-2)←PC _L		n			
	PC←nn		n			
CALL cc,nn	изпълнено	.. × . × ...	11 cc 100	3	3	10
	условие cc:		n	3	5	16
	като CALL nn		n			
RET	PC _L ←(SP)	.. × . × ...	11 001 001 C9	1	3	10
	PC _H ←(SP+1)					
RET cc	изпълнено	.. × . × ...	11 cc 000	1	1	5
	условие cc:		n	1	3	11
	като RET		n			
RETI	връщане от прекъсване	.. × . × ...	11 101 101 ED 01 001 101 4D	2	4	14
RETN	връщане от немаскируемо прекъсване	.. × . × ...	11 101 101 ED 01 000 101 45	2	4	14
RST p ¹	(SP-1)←PC _H (SP-2)←PC _L PC _H ←0 PC _L ←p	.. × . × ...	11 t 111	1	3	11

където

cc	условие
000	NZ — резултат, различен от нула
001	Z — резултат, равен на нула
010	NC — невъзникване на пренос или заем
011	C — възникване на пренос или заем
100	PO — нечетен резултат от операцията
101	PE — четен резултат от операцията
110	P — положителен резултат от операция
111	M — отрицателен резултат от операция

¹ Управлението се предава на инструкция с адрес p

t	p
000	00H
001	08H
010	10H
011	18H
100	20H
101	28H
110	30H

11. Инструкции за входно-изходни операции

Мнемоничен код	Операция	Флагове SZ-H-PNC	Код 76 543 210	Брой		
				байтове	цикли	
				M	T	
IN A,(n)	A ← (n)	..x.x...	11 011 011 DB n	2	3	11
IN r,(C)	r ← (C) при r=110 се изменят фла- говете	↑↑×↑×P0.	11 101 101 ED 01 r 000	2	3	12
INI	(HL) ← (C) B ← B - 1 HL ← HL + 1	×↓ ¹ ××××1×	11 101 101 ED 10 100 010 A2	2	4	16
INIR	(HL) ← (C) B ← B - 1 HL ← HL + 1 докато B стане 0	×↑××××1×	11 101 101 ED 10 110 010 B2	2	5	21
				2	4	16
IND	(HL) ← (C) B ← B - 1 HL ← HL - 1	×↓ ¹ ××××1×	11 101 101 ED 10 101 010 AA	2	4	16
INDR	(HL) ← (C) B ← B - 1 HL ← HL - 1 докато B стане 0	×↓××××1×	11 101 101 ED 10 111 010 BA	2	5	21
				2	4	16
OUT (n),A	(n) ← A	..x.x...	11 010 011 D3 n	2	3	11
OUT (C),r	(C) ← r	..x.x...	11 101 101 ED 01 r 001	2	3	12
OUTI	(C) ← (HL) B ← B - 1 HL ← HL + 1	×↓ ¹ ××××1×	11 101 101 ED 10 100 011 A3	2	4	16
OTIR	(C) ← (HL) B ← B - 1 HL ← HL + 1 докато B стане 0	×↑××××1×	11 101 101 ED 10 110 011 B3	2	5	21
				2	4	16
OTD	(C) ← (HL) B ← B - 1 HL ← HL - 1	×↓ ¹ ××××1×	11 101 101 ED 10 101 011 AB	2	4	16
OTDR	(C) ← (HL) B ← B - 1 HL ← HL - 1 докато B стане 0	×↑××××1×	11 101 101 ED 10 111 011 BB	2	5	21
				2	4	16

където ¹ — флаг Z е 1 при B-1=0, иначе Z=0

Набор инструкции на микропроцесора I8086/8088

Мнемоника	Наименование	Общ вид	Операнди	Бт	Флагове ODIFSZAPC
AAA	Корекция при събиране на ASCII-символи	AAA		1	U...UUXUX
AAD	Корекция при деление на ASCII-символи	AAD		2	U...XXUXU
AAM	Корекция при умножение на ASCII-символи	AAM		1	U...XXUXU
AAS	Корекция при изваждане на ASCII-символи	AAS		1	U...UUXUX
ADC	Събиране с пренос	ADC d,s	r,r r,m m,r r,i m,i a,i	2 2-4 2-4 3-4 3-6 2-3	X...XXXXX
ADD	Събиране	ADD d,s	r,r r,m m,r r,i m,i a,i	2 2-4 2-4 3-4 3-6 2-3	X...XXXXX
AND	Логическо И	AND d,s	r,r r,m m,r r,i m,i a,i	2 2-4 2-4 3-4 3-6 2-3	0...XXUX0
CALL	Преход към подпрограма	CALL t	p _n p _r mp16 rp16 mp32	3 5 2-4 2 2-4	
CBW	Преобразуване на байт в дума	CBW		1	
CLC	Нулиране на флаг C	CLC		10
CLD	Нулиране на флаг D	CLD		1	.0.....
CLI	Нулиране на флаг I	CLI		1	.0.....
CMC	Двоично допълнение на флаг C	CMC		1X
CMP	Сравнение	CMP d,s	r,r r,m m,r r,i m,i a,i	2 2-4 2-4 3-4 3-6 2-3	X...XXXXX
CMPS	Сравнение на низове	CMPS d _s ,s _s (rep)d _s ,s _s	d _s ,s _s	1 1	X...XXXXX
CWD	Преобразуване на дума в двойна дума	CWD		1	
DAA	Корекция при десетично събиране	DAA		1	X...XXXXX

Продължение на приложение 4

Мнемоника	Наименование	Общ вид	Операнди	Бт	Флагове ODITZAPC
DAS	Корекция при десетично изваждане	DAS		1	U...XXXXX
DEC	Намаляване с 1	DEC d	r16 r8 m	1 2 2-4	X...XXXXX
DIV	Деление без знак	DIV s	r8 r16 m8 m16	2 2 2-4 2-4	U...UUUUU
ESC	Освобождаване	ESC extr.s	i,m i,r	2-4 2 1	
HLT	Установяване на 18086/8088 в спряно състояние	HLT		1	
IDIV	Деление на цели числа със знак	IDIV s	r8 r16 m8 m16	2 2 2-4 2-4	U...UUUUU
IMUL	Умножение на цели числа със знак	IMUL s	r8 r16 m8 m16	2 2 2-4 2-4	X...UUUUX
IN	Четене от порт	IN a,p	a,i8 a,DX	2 1	
INC	Увеличаване с 1	INC d	r16 r8 m	1 2 2-4	X...XXXXX
INT	Прекъсване	INT n	i8(n=3) i8	1 2	..00....
INTO	Прекъсване при препълване	INTO		1	..00....
IRET	Връщане от прекъсване	IRET		1	RRRRRRRRR
JA/JNBE	Преход при по-голямо — флагове $CF \vee ZF = 0$	JA/JNBE l_s	l_s	2	
JAE/JNB	Преход при по-голямо или равно — $CF = 0$	JAE/JNB l_s	l_s	2	
JB/JNAE	Преход при по-малко — $CF = 1$	JB/JNAE l_s	l_s	2	
JBE/JNA	Преход при по-малко или равно — $CF \vee ZF = 1$	JBE/JNA l_s	l_s	2	
JC	Преход при наличие на пренос — $CF = 1$	JC l_s	l_s	2	
JCXZ	Преход при $CX = 0$	JCXZ l_s	l_s	2	
JE/JZ	Преход при равно /Преход при нула — $ZF = 1$	JE/JZ l_s	l_s	2	
JG/JNLE	Преход при по-голямо —	JG/JNLE l_s	l_s	2	
JGE/JNL	Преход при по-голямо или равно — $SF \oplus OF = 0$	JGE/JNL l_s	l_s	2	

Продължение на приложение 4

Мнемоника	Наименование	Общ вид	Операнди	Бг	Флагове	ODITSZAPC
JL/JNGE	Преход при по-малко — SF ⊕ OF = 1	JL/JNGE I _s	I _s	2		
JLE/JNG	Преход при по-малко или равно — SF ⊕ OF ∨ ZF = 1	JLE/JNG I _s	I _s	2		
JMP	Безусловен преход	JMP t	I _s I _n I _r mp16 rp16 mp32	2 3 5 2-4 2 2-4		
JNC	Преход при липса на пренос — CF = 0	JNC I _s	I _s	2		
JNE/JNZ	Преход при неравно /Пре- ход при не нула — ZF = 0	JNE/JNZ I _s	I _s	2		
JNO	Преход при липса на препълване — OF = 0	JNO I _s	I _s	2		
JNP/JPO	Преход при нечетен резул- тат — PF = 0	JNP/JPO I _s	I _s	2		
JNS	Преход при положителен знак — SF = 0	JNS I _s	I _s	2		
JO	Преход при препълване — OF = 1	JO I _s	I _s	2		
JP/JPE	Преход при четен резул- тат — PF = 1	JP/JPE I _s	I _s	2		
JS	Преход при отрицателен знак — SF = 1	JS I _s	I _s	2		
LAHF	Зареждане на регистър AH с флаговете	LAHF		1		
LDS	Зареждане на общ регистър и на регистър DS	LDS d,s	r16,m32	2-4		
LEA	Зареждане на ефективен ад- рес	LEA d,s	r16,m16	2-4		
LES	Зареждане на общ регистър и на регистър ES	LES d,s	r16,m32	2-4		
LOCK	Блокиране на шината	LOCK		1		
LODS	Зареждане на низ	LODS s _s	s _s (rep) s _s	1 1		
LOOP	Цикъл	LOOP I _s	I _s	2		
LOOPE/ LOOPZ	Цикъл при равно /Цикъл при нула	LOOPE I _s /LOOPZ I _s	I _s	2		
LOOPNE/Цикъл при неравно /Цикъл LOOPNZ при не нула		LOOPNE I _s /LOOPNZ I _s	I _s	2		
MOV	Прехвърляне	MOV d,s	m,a a,m r,r r,m	3 3 2 2-4		

Продължение на приложение 4

Мнемоника	Наименование	Общ вид	Операнди	Бт	Флагове ODITZAPC
			m, r	2-4	
			r, i	2-3	
			m, i	3-6	
			sr, r16	2	
			sr, m16	2-4	
			r16, sr	2	
			m16, sr	2-4	
MOVS	Прехвърляне на низове	MOVS d _s ,s _s	d _s ,s _s	1	
			(rep)d _s ,s _s	1	
MUL	Умножение без знак	MUL s	r8	2	X...UUUUX
			r16	2	
			m8	2-4	
			m16	2-4	
NEG	Умножение с -1	NEG d	r	2	X...XXXX1
			m	2-4	
NOP	Празна операция	NOP		1	
NOT	Логическо отрицание	NOT d	r	2	
			m	2-4	
OR	Логическо ИЛИ	OR d,s	r,r	2	0...XXUX0
			r,m	2-4	
			m,r	2-4	
			a,i	2-3	
			r,i	3-4	
			m,i	3-6	
OUT	Запис в порт	OUT p,a	i8,a	2	
			DX,a	1	
POP	Извличане на дума от стека	POP d	r	1	
			sr	1	
			m	2-4	
POPF	Зареждане на флаговете от стека	POPF		1	RRRRRRRRR
PUSH	Запис на дума в стека	PUSH s	r	1	
			sr	1	
			m	2-4	
PUSHF	Запис на флаговете в стека	PUSHF		1	
RCL	Ротация наляво през флага за пренос	RCL d,c	r,l	2	X.....X
			r,CL	2	
			m,l	2-4	
			m,CL	2-4	
RCR	Ротация надясно през флага за пренос	RCR d,c	r,l	2	X.....X
			r,CL	2	
			m,l	2-4	
			m,CL	2-4	
REP	Повторение на инструкция с низове	REP		1	
REPE/REPZ	Повторение при равенство Повторение при нула	REPE REPZ		1	
REPNE/REPNZ	Повторение при неравенство. Повторение при не нула	REPNE REPZ		1	

Продължение на приложение 4

Мнемоника	Нименование	Общ вид	Операнди	Бт	Флагове ODISZAPC
RET	Връщане от подпрограма	RET [v]	(s _n) (s _n ,v) (s _p) (s _p ,v)	1 3 1 3	
ROL	Ротация наляво	ROL d,c	r,l r,CL m,l m,CL	2 2 2-4 2-4	X.....X
ROR	Ротация надясно	ROR d,c	r,l r,CL m,l m,CL	2 2 2-4 2-4	X.....X
SAHF	Зареждане на флаговете от регистър AH	SAHF		1RRRRR
SAL/SHL	Аритметично изместване наляво /Логическо изместване наляво	SAL/SHL d,c	r,l r,CL m,l m,CL	2 2 2-4 2-4	X.....X
SAR	Аритметично изместване надясно	SAR d,s	r,l r,CL m,l m,CL	2 2 2-4 2-4	X...XXUXX
SBB	Изваждане с пренос (със заемане)	SBB d,s	r,r r,m m,r a,i r,i m,i	2 2-4 2-4 2-3 3-4 3-6	X...XXXXX
SCAS	Сканиране на низ	SCAS d _s	d _s (rep)d _s	1 1	X...XXXXX
SHR	Логическо изместване надясно	SHR d,c	r,l r,CL m,l m,CL	2 2 2-4 2-4	X.....X
STC	Установяване на флаг C в I	STC		11
STD	Установяване на флаг D в I	STD		1	.1.....
STI	Установяване на флаг I в I	STI		1	..1.....
STOS	Запис на низ (в паметта)	STOS d _s	d _s (rep)d _s	1 1	
SUB	Изваждане	SUB d,s	r,r r,m m,r a,i r,i m,i	2 2-4 2-4 2-3 3-4 3-6	X...XXXXX
TEST	Тестване за установяване на флаговете	TEST d,s	r,r r,m a,i r,i m,i	2 2-4 2-3 3-4 3-6	0...XXUX0
WAIT	Установяване на I8086/8088 в чакащо състояние	WAIT		1	

Продължение на приложение 4

Мнемоника	Наименование	Общ вид	Операнди	Бт	Флагове ODITZAPC
XCHG	Размяна на съдържанието	XCHG d,s	a,r16	1	
			m,r	2-4	
			r,r	2	
XLAT	Преобразуване при работа с таблици	XLAT tabl	tabl	1	
XOR	Изключващо ИЛИ	XOR d,s	r,i	2	0...XXUX0
			r,m	2-4	
			m,r	2-4	
			a,i	2-3	
			r,i	3-4	
			m,i	3-6	

Пояснения към символите и съкращенията:

- Бт — брой байтове
- d — назначение (destination) — регистър или клетка от паметта
- s — източник (source) — регистър, клетка от паметта или непосредствен операнд
- t — адрес за преход (target)
- d_s — назначение от тип низ (string)
- s_s — източник от тип низ
- extr — непосредствен операнд, използван от копроцесора
- p — адрес на порт
- n — тип (номер) на прекъсване
- l_s — адрес от паметта (label), разположен в областта от —128 до 127 байта спрямо първия байт на следващата инструкция (short)
- l_n — адрес в текущия кодов сегмент (near)
- l_f — адрес в друг кодов сегмент (far)
- c — брояч
- tabl — таблица за преобразуване, не по-голяма от 256 байта
- a — акумулатор — регистър AX или AL
- r — регистър
- m — клетка от паметта
- i — непосредствен операнд
- p_n — процедура в текущия кодов сегмент (near)
- p_f — процедура в друг кодов сегмент (far)
- m16 — 16-разредна клетка от паметта
- m8 — 8-разредна клетка от паметта
- m32 — 32-разредна клетка от паметта
- r8 — 8-разреден регистър
- r16 — 16-разреден регистър
- mp16 — 16-разредна клетка от паметта, съдържаща отместването на адреса на прехода в текущия кодов сегмент
- mp32 — 32-разредна клетка от паметта, съдържаща сегментиран адрес за преход
- rp16 — 16-разреден регистър, съдържащ отместването на адреса за прехода
- (rep) — инструкция за повторение
- i8 — 8-разреден непосредствен операнд
- sr — сегментен регистър
- [v] — незадължително зададен брой байтове за игнориране от стека
- (s_n) — текущ кодов сегмент (near)
- (s_f) — друг кодов сегмент (far)

Въздействие върху флаговете:

- 0 — нулиране
- 1 — установяване в 1
- X — установяване или нулиране в зависимост от резултата от операцията
- U — непредсказуемо въздействие
- R — възстановяване на предишно състояние

Съобщения за грешки на БСВИ

I. Контролни точки, записвани на адрес 0060 във входно-изходната памет (порт А на I8255A)

Състояние на изходните линии на порт А на I8255A	Състояние на програмата за самотестване, съответстващо на контролната точка
0 0 0 0 0 0 0 1	Микропроцесорът I8088 работи нормално. Следва проверка на контролната сума на 8 Кбайта PROM от адрес FE000 до FFFFF
0 0 0 0 0 0 1 0	Контролната сума на 8 Кбайта PROM е правилна. Следва проверка на таймер 1 от I8253-5
0 0 0 0 0 0 1 1	Таймер 1 от I8253-5 работи нормално. Следва проверка на контролера за ДДП I8237A-5, установяване на параметрите на канал 0 и стартиране на цикъла за опресняване на паметта
0 0 0 0 0 1 0 0	Контролерът за ДДП I8237A-5 работи нормално, стартиран е цикълът за опресняване на паметта и на линия DACK0 може да се наблюдава импулсна поредица с период около 15 µs. Следва проверка на 16 Кбайта RAM от адрес 00000 до 04000

Забележка. При неизправност в RAM в порт А на I8255A се записват алтернативно през около 1 s кодът на контролна точка 4 (00000100) и позиционен код на грешката в RAM. В позиционния код на грешката се установява 1 в битовете, за които е открита грешка при теста.

II. Съобщения за грешки с издаване на звуков сигнал

1. Един продължителен и два кратки звукови сигнала:
 - при неизправност във видеоконтролера (ИС CM607, вътрешния синхрогенератор или други схеми);
 - при неизправност във видеопаметта.
2. Един продължителен и един кратък звуков сигнал:
 - при неизправност на приоритетния контролер за прекъсвания I8259A;
 - при неизправност в таймер 0 на ИС I8253-5.

III. Съобщения за грешки, изобразявани на екрана на видеомонитора

- 201 XXXXX YY — неизправност в интегралните схеми на оперативната памет RAM, където:
 XXXXX — абсолютен начален адрес на блок от паметта тип RAM с размер 16 Кбайта, в който е възникнала неизправността;
 YY — позиционен код на грешката в RAM, като в позиционния код на грешката се установява 1 в битовете, за които е открита грешка при теста.
- 301 — неизправност в клавиатурата или в клавиатурния адаптер при подаване от компютъра на последователност за начално установяване на клавиатурата.
- 301 XX — неизправност в клавиатурата при проверката за постоянно натиснат клавиш, където:
 XX — позиционен код на постоянно натиснатия клавиш.
- 601 — неизправност в системата ЗУГМД—контролер: не е възможно зареждане на DOS от дискета.
- 1701 — неизправност в системата ЗУТМД—контролер: не е възможно зареждане на DOS от ЗУТМД.
- 1801 — неизправност в устройството за разширение (ако е включено в конфигурацията на персоналияния компютър или е поставена платка предавател към устройството за разширение).

Основни параметри на драйверите на БСВИ

I. Драйвер за асинхронна комуникация — INT 14

RS232_IO — Този драйвер осигурява обмен на последователност от данни през входно-изходния регистър на асинхронния адаптер в съответствие със следните входни параметри:

АН=0 Установява в начално състояние асинхронния адаптер. AL съдържа параметрите за начално установяване:

7	6	5	4	3	2	1	0
скорост, бит/с			контрол по четност		стоп бит	дължина на думата	
000 — 110			X0 — без		0 — 1	10 — 7 бита	
001 — 150			01 — нечетен		1 — 2	11 — 8 бита	
010 — 300			11 — четен				
011 — 600							
100 — 1200							
101 — 2400							
110 — 4800							
111 — 9600							

АН=1 Изпраща към асинхронния адаптер COM0 байта, който се намира в AL.

При връщане от драйвера:

— ако той не е в състояние да предаде байта, бит 7 в АН е установен в 1;

— ако драйверът е предал байта, бит 7 в АН е установен в 0, като останалите битове на АН съдържат информация за текущото състояние на комуникационните линии на асинхронния адаптер (както при АН=3).

АН=2 Приема байт от асинхронния адаптер COM0, като го записва в AL. При връщане от драйвера, съдържанието на АН е същото, както при входен параметър АН=1.

АН=3 Като резултат, записва в АХ информация за текущото състояние на асинхронния адаптер COM0. АН съдържа със състоянието на комуникационните линии на асинхронния адаптер COM0:

бит 7 — таймаут;

бит 6 — предавателният изместващ регистър е празен;

бит 5 — предавателният буферен регистър е празен;

бит 4 — открито е състояние „break“;

бит 3 — грешка при приемане;

бит 2 — грешка по четност;

бит 1 — грешка по преплъване;

бит 0 — индикатор за приети данни.

AL съдържа състоянието на модемните линии на асинхронния адаптер

COM0:

бит 7 — състояние на входа CD;

бит 6 — състояние на входа RI;

бит 5 — състояние на входа DSR;

бит 4 — състояние на входа CTS;

бит 3 — промяна на входа CD;

бит 2 — промяна на входа RI;

бит 1 — промяна на входа DSR;

бит 0 — промяна на входа CTS.

DX — параметър, който определя кой асинхронен адаптер (0 или 1) се използва за обмен.

II. Драйвер на клавиатурата — INT 16

KEYBOARD_IO — Този драйвер обслужва обмена с клавиатурата в съответствие със следните входни параметри:

- AH=0** Чете информация за поредния натиснат клавиш на клавиатурата.
Като резултат, в **AL** се записва ASCII-кодът на натиснатия клавиш, а в **AH** — позиционният код на клавиша.
- AH=1** Установява флаг **Z (ZF)**, който показва, че може да бъде прочетен поредният байт от клавиатурата:
- ZF=1** — забранява четенето;
 - ZF=0** — разрешава четенето.
- Когато **ZF=0**, поредният байт се намира в **AX**, а нововъведените байтове остават в буфера.
- AX=2** Връща в **AL** текущото състояние на регистъра за управляващите клавиши. Състоянието на бутоните, което съответства на този код, може да се прочете в полето **KB_FLAG**.

III. Драйвер на ЗУГМД — INT 13

DISKETTE_IO — Този драйвер управлява обмена със ЗУГМД в съответствие със следните входни параметри:

Регистри, използвани от командите ЧЕТЕНЕ/ЗАПИС/ПРОВЕРКА/ФОРМАТ:

- DL** — номер на ЗУГМД (стойност 0–3, стойността се проверява);
- DH** — номер на глава (стойност 0–1, не се проверява стойността);
- CH** — номер на пътечка (стойност 0–39, не се проверява стойността);
- CL** — номер на сектор (стойност 1–8, не се проверява стойността, не се използва при форматиране);
- AL** — брой на секторите (най-много 8, не се проверява стойността, не се използва при форматиране);
- ES:BX** — адрес на буфера за данни (не се изисква при проверка).
- AH=0** Установява в начално състояние системата контролер—ЗУГМД (начално установяване на ИС **CM609**, подготовка за възприемане на нова команда, възстановяване на всички ЗУГМД).
- AH=1** Чете информацията за състоянието на системата в **AL**.
- AH=2** Чете определените сектори в паметта.
- AH=3** Записва определените сектори от паметта.
- AH=4** Проверява определените сектори.
- AH=5** Форматира определената пътечка.

При командата **ФОРМАТ** указателят на буфера **ES:BX** трябва да сочи набор от адресни полета, определящи пътечката. Всяко поле се състои от 4 байта — **C**, **H**, **R** и **N**, където **C** е номер на пътечката, **H** е номер на главата, **R** е номер на сектора и **N** е брой на байтовете в сектора (**00=128**, **01=256**, **02=512** и **03=1024**). Необходимо е да има по едно поле за всеки сектор на пътечката.

DISK_POINTER Двойна дума, указател на таблицата с набора от параметри на ЗУГМД.

Регистри, които носят информация след излизане от драйвера:

- AH** — състояние след изпълнение на командата;
- CU** — установен в 0, показва успешно изпълнение на командата (**AH=0**);
- CU** — установен в 1, показва неуспешно изпълнение на командата (**AH** съдържа информация за грешката);
- AL** — брой на секторите, които са прочетени.

IV. Драйвер на ЗУТМД — INT 13

FDISK__IO — Този драйвер управлява обмена със ЗУТМД в съответствие със следните входни параметри:

Регистри, използвани от командите на ЗУТМД:

DL — номер на ЗУТМД (шестнадесетично число 80 — 87, стойността се проверява);

DH — номер на глава (стойност 0 — 7, не се проверява стойността);

CH — номер на пътека (стойност 0 — 1023, не се проверява стойността);

CL — номер на сектор (стойност 1 — 17, не се проверява стойността, не се използва при форматиране);

AL — брой на секторите (най-много 128, не се проверява стойността, не се използва при форматиране);

ES:BX — адрес на буфера за данни (не се изисква при проверка).

Забележки: Старшите два бита от номера на цилиндъра са старшите два бита на регистъра CL. Така за номер на цилиндъра са отделени 10 бита. При DL > 128 се избира ЗУТМД, а при DL < 128 — ЗУГМД.

АН=00 Установява в начално състояние ЗУГМД (DL=128,129) или ЗУТМД (DL < 128).

АН=01 Чете в AL информацията за състоянието на системата ЗУ — контролер след изпълнение на текущата команда.

АН=02 Чете определените сектори в паметта.

АН=03 Записва определените сектори от паметта.

АН=04 Проверява определените сектори.

АН=05 Форматира определената пътека.

АН=06 Форматира определената пътека и записва флагите за лош сектор.

АН=07 Форматира ЗУ, като започва от определената пътека.

АН=08 Връща в AX текущите параметри на ЗУ.

АН=09 Установява началните стойности на параметрите на ЗУ. INT 41 указва началото на блока от данни.

АН=0A Дълго четене (пропуска контролното коригиращо число за 512+4 байта).

АН=0B Дълъг запис (пропуска контролното коригиращо число за 512+4 байта).

АН=0C Търсене.

АН=0D Установява в начално състояние второто ЗУ.

АН=0E Чете от секторния буфер (локалната памет).

АН=0F Записва в секторния буфер (локалната памет).

АН=10 Проверява готовността на ЗУ.

АН=11 Възстановява параметрите на ЗУ.

АН=12 Проверява работоспособността на локалната памет.

АН=13 Проверява работоспособността на ЗУ.

АН=14 Проверява работоспособността на схемите на контролера.

Регистри, които носят информация след излизане от драйвера:

АН — състояние след изпълнение на командата;

CY — установен в 0, показва успешно изпълнение на командата (АН=0);

SU — установен в 1, показва неуспешно изпълнение на командата (АН съдържа информация за грешката);

AL — брой на секторите, които са прочетени;

DL — брой на последователно разпознатите ЗУ (0—2);

DH — използвана максимална стойност за номер на глава;

CH — използвана максимална стойност за номер на цилиндър;

CL — използвана максимална стойност за номер на сектор и старши битове на номера на цилиндър.

V. Драйвер на печатащо устройство — INT 17

PRINTER_IO — Този драйвер управлява обмена с печатащото устройство в съответствие със следните входни параметри:

- АН=0 Отпечатва записания в AL символ. Ако символът не може да бъде отпечатан след изтичане на контролното време, драйверът връща АН=1.
- АН=1 Установява в начално състояние входно-изходния регистър на контролера на принтера. След изпълнение драйверът връща в АН информация за състоянието на принтера.
- АН=2 Прочита информация за състоянието на принтера в АН.
- DX Съдържа информация за използваното в текущия момент печатащо устройство (0, 1 или 2).

VI. Драйвер на видеомонитора — INT 10

VIDEO_IO — Този драйвер управлява интерфейса към видеомонитора в съответствие със следните входни параметри:

- АН=0 Установява режима в съответствие със съдържанието на AL:
- | | |
|------|---|
| AL=0 | 40 × 25 монохроматичен (по подразбиране); |
| AL=1 | 40 × 25 цветен; |
| AL=2 | 80 × 25 монохроматичен; |
| AL=3 | 80 × 25 цветен. |

Графични режими:

- | | |
|------|---------------------------|
| AL=4 | 320 × 200 цветен; |
| AL=5 | 320 × 200 монохроматичен; |
| AL=6 | 640 × 200 монохроматичен; |

Вид на видеоконтролера:

- | | |
|------|---|
| AL=7 | 80 × 25 монохроматичен текстов контролер. |
|------|---|

АН=1 Установява вида на курсора в съответствие със съдържанието на:

CH* — битове 4–0 задават началната линия на курсора

Забележка. Установяването на битове 5 или 6 от CH може да предизвика неправилно изобразяване на курсора или изгубването му.

CL — битове 4–0 задават крайната линия на курсора.

АН=2 Установява положението на курсора в съответствие със съдържанието на:

DH, DL — ред, стълб (0,0 е горе, ляво);

BH — номер на страницата (трябва да бъде 0 за графичните режими).

АН=3 Чете позицията на курсора:

BH — номер на страница (трябва да бъде 0 за графичните режими).

Драйверът връща след изпълнението:

DH, DL — ред, стълб на текущото положение на курсора;

CH, CL — текущо установения вид на курсора.

АН=4 Чете положението на светлинния молив, като драйверът връща след изпълнението: АН=0 бутонът на светлинния молив не е натиснат;

АН=1 получена е стойност за положението на светлинния молив, като:

DH, DL определят ред и стълб на символа, отбелязан със светлинния молив;

CH определя растерната линия (0–199);

BX определя стълба на точката (0–319,639).

АН=5 Избира активна страница във видеопаметта (валидно само при текстовите режими):

AL — стойност (номер) на новата страница (0–7 за режими 0 и 1 или 0–3 за режими 2 и 3).

- АН=6 Избира активна страница при движението на изображението нагоре:
 AL — брой на линиите (входните линии се изчистват в горната част на прозореца);
 AL=0 — изчиства се целият екран;
 CH,CL — ред, стълб на горния ляв ъгъл на прозореца;
 DH,DL — ред, стълб на долния десен ъгъл на прозореца;
 BH — атрибути на линиите, които се изчистват.
- АН=7 Избира активна страница при движението на изображението надолу:
 AL — брой на линиите (входните линии се изчистват в долната част на прозореца);
 AL=0 — изчиства се целият екран;
 CH,CL — ред, стълб на горния ляв ъгъл на прозореца;
 DH,DL — ред, стълб на долния десен ъгъл на прозореца;
 BH — атрибути на линиите, които се изчистват.
- АН=8 Чете атрибутите и символа на текущата позиция на курсора:
 BH — номер на страница (валиден само в текстовите режими).
 Драйверът връща след изпълнението:
 AL — прочетения символ;
 AH — атрибутите на символа (валидни само в текстовите режими).
- АН=9 Записва атрибутите и символа на текущата позиция на курсора:
 BH — номер на страница (валиден само в текстовите режими);
 CX — брояч на символите за запис;
 AL — символ за запис;
 AH — атрибутите на символа (в текстов режим) или цвета на символа (в графичен режим).
- АН=10 Записва само символ на текущата позиция на курсора:
 BH — номер на страница (валиден само в текстовите режими);
 CX — брояч на символите за запис;
 AL — символ за запис.
- АН=11 Установява цветовата палитра (само за режима 320×200):
 BH — идентификатор на цвета на палитрата (0 — 127);
 VL — стойност на цвета, който ще се използва с идентификатора в BH.
- АН=12 Записва точка в съответствие със стойностите на:
 DX — номер на реда;
 CX — номер на стълба;
 AL — стойност на цвета.
- АН=13 Чете точка в съответствие със стойностите на:
 DX — номер на реда;
 CX — номер на стълба.
 Драйверът връща след изпълнението:
 AL — номер на цвета.
- АН=15 Получава информация за текущото състояние на видеоконтролера:
 AL — текущо установен режим;
 AH — брой на стълбовете, изобразявани на екрана;
 BH — номер на текущата активна страница.

Набор команди на ИС СМ630

Фаза	3/4	Шина за данни								Забележки
		D7	D6	D5	D4	D3	D2	D1	D0	
Четене на данни										
Командна	3	MT	MF	SK	0	0	1	1	0	Код на операцията
	3	X	X	X	X	X	HD	US1	US0	
	3						C			Информация за идентифициране на сектора преди изпълнението на командата
	3						H			
	3						R			
	3						N			
	3						EOT			
3						GPL				
3						DTL				
Изпълнителна										Прехвърляне на данни от ЗУГМД в оперативната памет
Заклучителна	4						ST0			Информация за състоянието след изпълнението на командата
	4						ST1			
	4						ST2			
	4						C			
	4						H			Информация за идентифициране на сектора след изпълнението на командата
	4						R			
	4						N			
	4									
Четене на изтрити данни										
Командна	3	MT	MF	SK	0	1	1	0	0	Код на операцията
	3	X	X	X	X	X	HD	US1	US0	
	3						C			Информация за идентифициране на сектора преди изпълнението на командата
	3						H			
	3						R			
	3						N			
	3						EOT			
3						GPL				
3						DTL				
Изпълнителна										Прехвърляне на данни от ЗУГМД в оперативната памет
Заклучителна	4						ST0			Информация за състоянието след изпълнението на командата
	4						ST1			
	4						ST2			
	4						C			
	4						H			Информация за идентифициране на сектора след изпълнението на командата
	4						R			
	4						N			
	4									
Запис на данни										
Командна	3	MT	MF	0	0	0	1	0	1	Код на операцията
	3	X	X	X	X	X	HD	US1	US0	

Продължение на приложение 7

Фаза	З Ч	Шина за данни								Забележки	
		D7	D6	D5	D4	D3	D2	D1	D0		
	3									C	Информация за идентифициране на сектора преди изпълнението на командата
	3									H	
	3									R	
	3									N	
	3									EOT	
	3									GPL	
	3									DTL	
Изпълнителна											Прехвърляне на данни от оперативната памет в ЗУГМД
Заклучителна	4									ST0	Информация за състоянието след изпълнението на командата
	4									ST1	
	4									ST2	
	4									C	Информация за идентифициране на сектора след изпълнението на командата
	4									H	
	4									R	
	4									N	
		Запис на изгрити данни									
Командна	3	MT	MF	0	0	1	0	0	1		Код на операцията
	3	X	X	X	X	X	HD	US1	US0		
	3									C	Информация за идентифициране на сектора преди изпълнението на командата
	3									H	
	3									R	
	3									N	
	3									EOT	
	3									GPL	
	3									DTL	
Изпълнителна											Прехвърляне на данни от оперативната памет в ЗУГМД
Заклучителна	4									ST0	Информация за състоянието след изпълнението на командата
	4									ST1	
	4									ST2	
	4									C	Информация за идентифициране на сектора след изпълнението на командата
	4									H	
	4									R	
	4									N	
		Четене на пътечка									
Командна	3	0	MF	SK	0	0	0	1	0		Код на операцията
	3	X	X	X	X	X	HD	US1	US0		
	3									C	Информация за идентифициране на сектора преди изпълнението на командата
	3									H	
	3									R	
	3									N	
	3									EOT	
	3									GPL	
	3									DTL	

Продължение на приложение 7

Фаза	3/4	Шина за данни								Забелжки
		D7	D6	D5	D4	D3	D2	D1	D0	
Изпълнителна										Прехвърляне на данни от ЗУГМД в оперативната памет. Контролерът чете цялото съдържание на цилиндъра от индекса до ЕОТ
Заключителна	Ч								ST0	Информация за състоянието след изпълнението на командата
	Ч								ST1	
	Ч								ST2	
	Ч								С	
	Ч								Н	Информация за идентифицирането на сектора след изпълнението на командата
	Ч								Н	
	Ч								Н	
	Ч								Н	
Четене на идентификатор										
Командна	3	0	MF	0	0	1	0	1	0	Код на операцията
	3	X	X	X	X	X	HD	US1	US0	
Изпълнителна										Първата правилна идентифицираща информация от цилиндъра се записва в регистъра за данни
Заключителна	Ч								ST0	Информация за състоянието след изпълнението на командата
	Ч								ST1	
	Ч								ST2	
	Ч								С	
	Ч								Н	Информация за идентифициране на сектора по време на изпълнителната фаза
	Ч								Н	
	Ч								Н	
	Ч								Н	
Форматиране на пътека										
Командна	3	0	MF	0	0	1	1	0	0	Код на операцията
	3	X	X	X	X	X	HD	US1	US0	
	3								Н	Байтове/Сектор Сектор/Пътека Промеждутък 3 Запълващ байт
	3								SC	
	3								GPL	
	3								D	
Изпълнителна										Контролерът форматира цял цилиндър
Заключителна	Ч								ST0	Информация за състоянието след изпълнението на командата
	Ч								ST1	
	Ч								ST2	
	Ч								С	
	Ч								Н	Информацията за идентифициране на сектора в този случай няма смисъл
	Ч								Н	
	Ч								Н	
	Ч								Н	

Продължение на приложение 7

Фаза	З/ч	Шина за данни								Забележки
		D7	D6	D5	D4	D3	D2	D1	D0	

Сканиране за равенство

Командна	3	MT	MF	SK	1	0	0	0	1	Код на операцията
	3	X	X	X	X	X	HD	US1	US0	
Изпълнителна	3						C			Информация за идентифициране на сектора преди изпълнението на командата
	3						H			
	3							R		
	3							N		
	3							EOT		
	3							GPL		
	3							DTL		
Заклучителна	ч						ST0			Сравняване на данни в оперативната памет и ЗУГМД
	ч						ST1			
Заклучителна	ч						ST2			Информация за състоянието след изпълнението на командата
	ч						C			
	ч						H			
	ч						R			
	ч						N			

Сканиране за по-малко или равно

Командна	3	MT	MF	SK	1	1	0	0	1	Код на операцията
	3	X	X	X	X	X	HD	US1	US0	
Изпълнителна	3						C			Информация за идентифициране на сектора преди изпълнението на командата
	3						H			
	3							R		
	3							N		
	3							EOT		
	3							GPL		
	3							DTL		
Заклучителна	ч						ST0			Сравняване на данни в оперативната памет и ЗУГМД
	ч						ST1			
Заклучителна	ч						ST2			Информация за състоянието след изпълнението на командата
	ч						C			
	ч						H			
	ч						R			
	ч						N			

Сканиране за по-голямо или равно

Командна	3	MT	MF	SK	1	1	1	0	1	Код на операцията
	3	X	X	X	X	X	HD	US1	US0	

Продължение на приложение 7

Фаза	З/Ч	Шина за данни								Забележки	
		D7	D6	D5	D4	D3	D2	D1	D0		
	3										Информация за идентифициране на сектора преди изпълнението на командата
	3										
	3										
	3										
	3										
	3										
Изпълнителна											Сравняване на данни в оперативната памет и ЗУГМД
Заключителна	Ч										Информация за състоянието след изпълнението на командата
	Ч										
	Ч										
	Ч										
											Информация за идентифициране на сектора след изпълнението на командата
	Ч										
	Ч										
	Ч										
Възстановяване — установяване											
Командна	3	0	0	0	0	0	1	1	1		Код на операцията
	3	X	X	X	X	X	0	US1	US0		
Изпълнителна											Изтегляне на главата на нулевата пътечка
Разпознаване на прекъсването											
Командна	3										Код на операцията
Заклучителна	Ч										Информация за състоянието на контролера в края на операцията
	Ч										
Специфициране											
Командна	3	0	0	0	0	0	0	1	1		Код на операцията
	3	---	SRT	---	---	---	---	HUT	---		
	3	---	HLT	---	---	---	---	---	ND		
Разпознаване на състоянието на ЗУГМД											
Командна	3	0	0	0	0	0	1	0	0		Код на операцията
	3	X	X	X	X	X	HD	US1	US0		
Заклучителна	Ч										Състояние на ЗУГМД
Позициониране											
Командна	3	0	0	0	0	1	1	1	1		Код на операцията
	3	X	X	X	X	X	HD	US1	US0		
	3										
Изпълнителна											Позициониране на главата над указания цилиндър на дискетата

Продължение на приложение 7

Фаза	З/Ч	Шина за данни D7 D6 D5 D4 D3 D2 D1 D0	Забележки
Командна	3	Невалидно Невалидни кодове	Невалидни кодове на операции (контролерът преминава в състояние на празен ход поради липса на операция)
Заклучителна	4	ST0	ST0 = 80

където: 0 — логическа нула, 1 — логическа единица, X — не се използва

Описание на символите

Символ	Наименование	Описание
A0	Адресна шина 0	A0 управлява избора на главния регистър на състоянието (A0=0) или на регистъра за данни (A0=1).
C	Номер на цилиндър	C означава текущия/избрания номер на цилиндър (пътечка) на носителя.
D	Данни	D означава шаблона на данните, които ще се записват в даден сектор.
D7—D0	Шина за данни	Осемразредна шина за данни, където D7 означава старшия, а D0 — младшия бит.
DTL	Дължина на данните	Когато N е определен като 00, DTL е дължината на данните, които потребителите ще четат от или ще записват в сектора.
EOT	Край на пътечка	EOT означава последния номер на сектор на един цилиндър.
GPL	Дължина на промеждутъка	GPL означава дължината на <i>Промеждутък 3</i> (интервала между секторите, с изключение на полето за синхронизация VCO).
H	Адрес на глава	H означава номер на глава 0 или 1, както е указано в полето ID.
HD	Глава	HD означава избрана глава номер 0 или 1 (H=HD във всички управляващи думи).
HLT	Време за натоварване на глава	HLT означава времето за натоварване на глава в ЗУГМД (от 4 до 512 ms през 4 ms).
HUT	Време за разтоварване на глава	HUT означава времето за разтоварване на глава след изпълнението на операция <i>Четене</i> или <i>Запис</i> (от 0 до 480 ms през 32 ms).
MF	Режим FM или MFM	Ако MF е с ниско ниво, избира се режим FM, а ако е с високо ниво, избира се режим MFM — само ако MFM е реализиран.
MT	Многопътечкова операция	Ако MT е с високо ниво, ще се изпълни многопътечкова операция (ще бъде прочетен или записан цилиндърът под двете глави HD0 и HD1)
N	Брой	N означава броя на байтовете данни, записани в сектор.
NCN	Номер на нов цилиндър	NCN означава нов номер на цилиндър, който ще бъде достигнат в резултат на операцията <i>Позициониране</i> , т.е. желаната позиция на главата.
ND	Режим без ДДП	ND означава операция в режим без ДДП.
PCN	Номер на текущ цилиндър	PCN означава номер на цилиндър при завършването на командата <i>Разпознаване на състоянието на прекъсване</i> , като указва положението на главата в текущия момент.
R	Запис	R означава номера на сектора, който ще бъде прочетен или записан.

Продължение на приложение 7

Символ	Наименование	Описание
Z, Ч	Запис/Четене	Z/Ч означава сигнал <i>Четене</i> (Ч) или <i>Запис</i> (З).
SC	Сектор	SC показва броя на секторите на цилиндър.
SK	Пропускане	SK означава пропускане на адресния маркер на изтрити данни.
SRT	Период на стъпката	SRT означава периода на стъпката за ЗУГМД (от 2 до 32 ms през 2 ms).
ST0	Състояние 0	ST0—ST3 са четирите регистра, които съхраняват информацията за състоянието след изпълнението на командата. Тази информация е достъпна през заключителната фаза след изпълнението на командата. Тези регистри не трябва да се объркват с главния регистър на състоянието (избирая чрез A0=0). ST0—ST3 могат да бъдат четени само след като е била изпълнена команда и съдържат информация, която е значеща за въпросната команда.
ST1	Състояние 1	
ST2	Състояние 2	
ST3	Състояние 3	
STP	Тест сканиране	По време на операция <i>Сканиране</i> , ако STP=1, данните в последователно разположени сектори се сравняват байт по байт с тези, изпращани от процесора (или ДДП), а ако STP=2, редуващи се сектори се четат и сравняват
US0, US1	Избор на устройство	US означава номера на избраното устройство.

Регистър на състоянието 0

Бит			Описание
№	Име	Символ	
D7	Код на прекъсване	IC	D7=0 и D6=0 : Нормално завършване на командата (NT). Командата е завършена и коректно изпълнена.
D6			D7=0 и D6=1 : Ненормално завършване на командата (AT). Изпълнението на командата е било стартирано, но не е завършило успешно. D7=1 и D6=0 : Невалидно подаване на команда (IC). Подадената команда не е стартирана. D7=1 и D6=1 : Ненормално завършване поради това, че по време на изпълнението на командата сигналът за готовност на ЗУГМД е променил състоянието си.
D5	Завършено позициониране	SE	Този флаг се установява в 1 (високо ниво), когато контролерът завърши изпълнението на командата <i>Позициониране</i> .
D4	Апаратна грешка	EC	Ако от ЗУГМД се получи сигнал за грешка или ако сигналът <i>Пътека 0</i> не се получи след 77 стъпкови импулса (команда <i>Възстановяване</i> — <i>Установяване</i>), този флаг се вдига.
D3	Неготов	NR	Този флаг се установява, когато ЗУГМД не е в състояние на готовност и е подадена команда за четене или запис. Същото се получава, когато е подадена команда за четене или запис към страна 1 на едностранно ЗУГМД.

1	2	3	4
D2	Адрес на глава	HD	Този флаг се използва за указване на състоянието на главата при прекъсване.
D1	Избор 1	US 1	Тези флагове се използват за показване на номера на ЗУГМД при прекъсване.
D0	Избор 0	US 0	

Регистър на състоянието 1

Бит			Описание
№	Име	Символ	
D7	Край на цилиндър	EN	Неизползван. Този бит е винаги 0.
D6	—	—	
D5	Грешка в данните	DE	Когато контролерът открие грешен CRC в полето за идентификация или за данни, този флаг се установява.
D4	—	OR	Този флаг се вдига, ако контролерът не е обслужен от системата по време на прехвърлянето на данни в рамките на известен времеви интервал.
D3	—	—	Неизползван. Този бит е винаги 0.
D2	Липса на данни	ND	Този флаг се установява, ако: <ul style="list-style-type: none"> — контролерът не може да намери сектора, специфициран в регистъра ID по време на изпълнението на команда <i>Четене на данни</i>, <i>Запис на изтрити данни</i> или <i>Сканиране</i>; — контролерът не може да прочете без грешка полето за идентификация по време на изпълнението на командата <i>Четене на идентификатор</i>; — началният сектор не може да бъде намерен по време на изпълнението на командата <i>Четене на цилиндър</i>.
D1	Защитен от запис	NW	Този флаг се вдига, ако контролерът открие сигнал за защита от запис на ЗУГМД по време на изпълнението на команда <i>Запис на данни</i> , <i>Запис на изтрити данни</i> или <i>Форматиране на цилиндър</i> .
D0	Липсващ адресен маркер	MA	Ако контролерът не може да открие адресния маркер в ID, този флаг се установява. Същевременно се установява и битът MD (липсващ адресен маркер в полето за данни) в регистъра на състоянието 2.

Регистър на състоянието 2

Бит			Описание
№	Име	Символ	
D7	—	—	Неизползван. Този бит е винаги 0.
D6	Контролен маркер	CM	Ако контролерът срещне по време на изпълнението на команда <i>Четене на данни</i> сектор, съдържащ адресен маркер на изтрити данни, този флаг се вдига.
D5	Грешка в данните в полето за данни	DD	Флагът се вдига, ако контролерът открие грешка в CRC на данните.
D4	Грешен цилиндър	WC	Този бит е свързан с бита ND и флагът се установява, когато съдържанието на C на носителя се различава от това, което е съхранено в регистъра ID.

Продължение на приложение 7

1	2	3	4
D3	Успешно сканиране за равенство	SH	Този флаг се установява, ако условието за равенство е удовлетворено по време на изпълнението на командата <i>Сканиране</i> .
D2	Неуспешно сканиране	SN	Този флаг се установява, ако контролерът не може да открие сектор в цилиндъра, който да удовлетворява условието при изпълнението на командата <i>Сканиране</i> .
D1	Лош цилиндър	BC	Този бит е свързан с бита ND и флагът се установява, когато съдържанието на C на носителя се различава от това, което е съхранено в регистъри ID и съдържанието на C е FF.
D0	Липсващ адресен маркер в полето за данни	MD	Когато се четат данни от носителя и контролерът не може да намери адресен маркер на данни или адресен маркер на изтрети данни, този флаг се установява.

Регистър на състоянието 3

Бит			Описание
№	Име	Символ	
D7	Грешка	FT	Този бит се използва за показване на състоянието на сигнала <i>Грешка</i> от ЗУГМД.
D6	Защитен от запис.	WP	Този бит се използва за показване на състоянието на сигнала <i>Защита от запис</i> от ЗУГМД.
D5	Готов	RY	Този бит се използва за показване на състоянието на сигнала <i>Готовиост</i> от ЗУГМД.
D4	Пътечка 0	T0	Този бит се използва за показване на състоянието на сигнала <i>Пътечка 0</i> от ЗУГМД.
D3	Двустранен	TS	Този бит се използва за показване на състоянието на сигнала <i>Двустранен</i> от ЗУГМД.
D2	Адрес на глава	HD	Този бит се използва за показване на състоянието на сигнала <i>Избор на страна</i> към ЗУГМД.
D1	Избор на устройство 1	US 1	Този бит се използва за показване на състоянието на сигнала <i>Избор на устройство 1</i> към ЗУГМД.
D0	Избор на устройство 0	US 0	Този бит се използва за показване на състоянието на сигнала <i>Избор на устройство 0</i> към ЗУГМД.

Типове и номера на грешки, издавани от контролера на ЗУТМД

Тип на грешката (битове)	Код на грешката (битове)	Описание на грешката
5 4	3 2 1 0	
0 0	0 0 0 0	Контролерът не е открил грешка по време на изпълнението на предишната команда
0 0	0 0 0 1	Контролерът не е получил сигнал ИНДЕКС от устройството
0 0	0 0 1 0	Контролерът не е получил сигнал ЗАВЪРШЕНО ТЪРСЕНЕ след завършване на операция търсене (за всички небуферирани стъпки на търсене)
0 0	0 0 1 1	Контролерът е получил сигнал ГРЕШКА ПРИ ЗАПИС по време на изпълнение на операция запис
0 0	0 1 0 0	След като контролерът е избрал устройството, то не е отговорило със сигнал ГОТОВНОСТ
0 0	0 1 0 1	Не се използва
0 0	0 1 1 0	След подаване на сигнали СТЬПКА, които са повече от максималния номер на цилиндър, контролерът не е получил сигнал ПЪТЕКА 0 от устройството
0 0	0 1 1 1	Не се използва
0 0	1 0 0 0	ЗУТМД е в състояние на търсене: получава се при опит за изпълнение на команда търсене преди завършване на предишната команда за търсене
0 1	0 0 0 0	Грешка при четене на идентификатора: контролерът открива грешка в контролното число на полето на идентификатора
0 1	0 0 0 1	Грешка при четене на полето за данни: контролерът открива невъзстановима грешка по контролното число на полето за данни
0 1	0 0 1 0	Грешка при четене на адресния маркер: контролерът не може да открие правилен адресен маркер върху диска
0 1	0 0 1 1	Не се използва
0 1	0 1 0 0	Не е намерен сектор: контролерът открива правилен цилиндър и глава, но не намира правилен сектор
0 1	0 1 0 1	Грешка при търсене: адресите на цилиндъра или на главата (поотделно или и двата) не съпадат с очакваните при изпълнение на операция търсене
0 1	0 1 1 0	Не се използва
0 1	0 1 1 1	Не се използва
0 1	1 0 0 0	Възстановима грешка при четене на полето за данни: контролерът открива възстановима грешка по контролното число на полето за данни
0 1	1 0 0 1	Лоша пътечка: контролерът е открил флаг за лоша пътечка при изпълнение на предишната команда
1 0	0 0 0 0	Невалидна команда: контролерът е получил невалидна команда от системната шина
1 0	0 0 0 0	Неправилен дисков адрес: контролерът открива адрес, който надхвърля максималния адрес
1 1	0 0 0 0	Грешка в RAM: контролерът открива грешка по време на диагностичната проверка на локалната памет
1 1	0 0 0 1	Грешка в контролната сума на PROM: по време на диагностичната проверка на PROM контролерът открива грешка в контролната му сума
1 1	0 0 1 0	Грешка в генератора на контролното коригиращо число: по време на диагностичната проверка, се открива неизправност в схемите за изчисляване и запис на контролното коригиращо число

Набор команди на контролера на ЗУТМД

Команда	Блок за управление на устройството									Забелжки	
1	2									3	
Проверка за готовност на ЗУТМД (клас 0, КОП 00)	Битове	-	7	6	5	4	3	2	1	0	д — ЗУ (0 или 1) × — не се използва
	Байт 0	0	0	0	0	0	0	0	0	0	Байтове 2, 3, 4 и 5 не се използват
	Байт 1	0	0	д	×	×	×	×	×	×	
Установяване — възстановяване (клас 0, КОП 01)	Битове	7	6	5	4	3	2	1	0	д — ЗУ (0 или 1) × — не се използва	
	Байт 0	0	0	0	0	0	0	0	0	1	п — повторения с — избор на стъпка Байтове 2, 3 и 4 не се използват
	Байт 1	0	0	д	×	×	×	×	×	×	
Байт 5	п	0	0	0	0	с	с	с	с		
Запазена (клас 0, КОП 02)											Тази команда не се използва
Получаване на информация за състоянието (клас 0, КОП 03)	Битове	7	6	5	4	3	2	1	0	д — ЗУ (0 или 1) × — не се използва	
	Байт 0	0	0	0	0	0	0	0	1	1	Байтове 2, 3, 4 и 5 не се използват
	Байт 1	0	0	д	×	×	×	×	×	×	
Битове	7	6	5	4	3	2	1	0	д — ЗУ (0 или 1) не — старша част на номера на цилиндъра кратност — 1 до 16 за 512-байтови сектори п — повторения с — избор на стъпка		
Форматиране на ЗУТМД (клас 0, КОП 04)	Байт 0	0	0	0	0	0	1	0	0		
	Байт 1	0	0	д	номер на глава						
	Байт 2	не	0	0	0	0	0	0	0		
	Байт 3	номер	на	цилиндър (мл.)							
	Байт 4	0	0	0	кратност						
	Байт 5	п	0	0	0	0	с	с	с		
Проверка на ЗУТМД (клас 0, КОП 05)	Битове	7	6	5	4	3	2	1	0	д — ЗУ (0 или 1) не — старша част на номера на цилиндъра п — повторения с — избор на стъпка к — възможност за повторение при грешка по ККЧ ¹ в полето за данни	
	Байт 0	0	0	0	0	0	1	0	1		
	Байт 1	0	0	д	номер на глава						
	Байт 2	не	номер	на	сектор						
	Байт 3	номер	на	цилиндър (мл.)							
	Байт 4		брой	на	блока						
Байт 5	п	к	0	0	0	с	с	с			

¹ ККЧ — контролно копиращо число.

Продължение на приложение 9

1		2	3
Форматиране на пътечка (клас 0, КОП 06)	Битове	7 6 5 4 3 2 1 0	д — ЗУ (0 или 1) nc — старша част на номера на цилиндъра кратност — 1 до 16 за 512-байтови сектори п — повторения с — избор на стъпка
	Байт 0	0 0 0 0 0 1 1 0	
	Байт 1	0 0 д номер на глава	
	Байт 2	nc 0 0 0 0 0 0	
	Байт 3	номер на цилиндър (м.л.)	
	Байт 4	0 0 0 кратност	
Форматиране на лоша пътечка (клас 0, КОП 07)	Битове	7 6 5 4 3 2 1 0	д — ЗУ (0 или 1) nc — старша част на номера на цилиндъра кратност — 1 до 16 за 512-байтови сектори п — повторения с — избор на стъпка
	Байт 0	0 0 0 0 0 1 1 1	
	Байт 1	0 0 д номер на глава	
	Байт 2	nc 0 0 0 0 0 0	
	Байт 3	номер на цилиндър (м.л.)	
	Байт 4	0 0 0 кратност	
Четене (клас 0, КОП 08)	Битове	7 6 5 4 3 2 1 0	д — ЗУ (0 или 1) nc — старша част на номера на цилиндъра п — повторения с — избор на стъпка к — възможност за повторение при грешка по ККЧ в полето за данни Байт 4 не се използва
	Байт 0	0 0 0 0 1 0 0 0	
	Байт 1	0 0 д номер на глава	
	Байт 2	nc номер на сектор	
	Байт 3	номер на цилиндър (м.л.)	
	Байт 5	п к 0 0 0 с с с	
Запазена (клас 0, КОП 09)			Тази команда не се използва
Запис (клас 0, КОП 0A)	Битове	7 6 5 4 3 2 1 0	д — ЗУ (0 или 1) nc — старша част на номера на цилиндъра п — повторения с — избор на стъпка
	Байт 0	0 0 0 0 1 0 1 0	
	Байт 1	0 0 д номер на глава	
	Байт 2	nc номер на сектор	
	Байт 3	номер на цилиндър (м.л.)	
	Байт 4	брояч на блока	
Байт 5	п 0 0 0 0 с с с		

Продължение на приложение 9

1	2	3	
Търсене (клас 0, КОП 0B)	Битове	7 6 5 4 3 2 1 0	д — ЗУ (0 или 1) це — старша част на номера на цилиндъра п — повторения × — не се използва с — избор на стъпка
	Байт 0	0 0 0 0 1 0 1 1	
	Байт 1	0 0 д номер на глава	
	Байт 2	це номер на сектор	
	Байт 3	номер на цилиндър (м.л.)	
	Байт 4	× × × × × × × ×	
	Байт 5	п 0 0 0 0 с с с	
Начално установяване на параметрите на ЗУТМД ¹ (клас 0, КОП 0C)	Битове	7 6 5 4 3 2 1 0	Байтове 1, 2, 3, 4 и 5 не се използват
	Байт 0	0 0 0 0 1 1 0 0	
Прочитане на количеството на възстановимите грешки след бърза проверка чрез ККЧ (клас 0, КОП 0D)	Битове	7 6 5 4 3 2 1 0	Байтове 1, 2, 3, 4 и 5 не се използват
	Байт 0	0 0 0 0 1 1 0 1	
Четене на данни от секторния буфер (клас 0, КОП 0E)	Битове	7 6 5 4 3 2 1 0	Байтове 1, 2, 3, 4 и 5 не се използват
	Байт 0	0 0 0 0 1 1 1 0	
Запис на данни в секторния буфер (клас 0, КОП 0F)	Битове	7 6 5 4 3 2 1 0	Байтове 1, 2, 3, 4 и 5 не се използват
	Байт 0	0 0 0 0 1 1 1 1	
Диагностика на локалната памет (клас 7, КОП 00)	Битове	7 6 5 4 3 2 1 0	Байтове 1, 2, 3, 4 и 5 не се използват
	Байт 0	1 1 1 0 0 0 0 0	
Запазена (клас 7, КОП 01)			Този команда не се използва
Запазена (клас 7, КОП 02)			Този команда не се използва

¹ Блокът с параметрите на ЗУТМД трябва да се състои от 8 байта:

Максимален брой на цилиндрите	2 байта
Максимален брой на главите	1 байта
Начален цилиндър за намаляване тока на запис	2 байта
Начален цилиндър за прекомпенсацията при запис	2 байта
Максимално количество на възстановимите грешки след бърза проверка чрез ККЧ	1 байт

Продължение на приложение 9

1		2								3	
Диагностика на ЗУТМД (клас 7, КОП 03)	Битове	7	6	5	4	3	2	1	0	д — ЗУ (0 или 1) п — повторения х ₁ — не се използва с — избор на стъпка	
	Байт 0	1	1	1	0	0	0	1	1		
	Байт 1	0	0	д	х	х	х	х	х		
	Байт 2	х	х	х	х	х	х	х	х		
	Байт 3	х	х	х	х	х	х	х	х		
	Байт 4	х	х	х	х	х	х	х	х		
	Байт 5	п	0	0	0	0	с	с	с		
Вътрешна диагностика на контролера (клас 7, КОП 04)	Битове	7	6	5	4	3	2	1	0	Байтове 1, 2, 3, 4 и 5 не се използват	
	Байт 0	1	1	1	0	0	1	0	0		
Чегне ¹ без проверка на ККЧ (клас 7, КОП 05)	Битове	7	6	5	4	3	2	1	0	д — ЗУ (0 или 1) пс — старша част на номера на цилиндъра п — повторения с — избор на стъпка	
	Байт 0	1	1	1	0	0	1	0	1		
	Байт 1	0	0	д	номер на глава						
	Байт 2	пс номер на сектор									
	Байт 3	номер на цилиндър (мл.)									
	Байт 4	брояч на блока									
	Байт 5	п	0	0	0	0	с	с	с		
Запис ² без проверка на ККЧ (клас 7, КОП 06)	Битове	7	6	5	4	3	2	1	0	д — ЗУ (0 или 1) пс — старша част на номера на цилиндъра п — повторения с — избор на стъпка	
	Байт 0	1	1	1	0	0	1	1	0		
	Байт 1	0	0	д	номер на глава						
	Байт 2	пс номер на сектор									
	Байт 3	номер на цилиндър (мл.)									
	Байт 4	брояч на блока									
	Байт 5	п	0	0	0	0	с	с	с		

¹ Прочитат се 512+4 байта за всеки сектор, като четирите байта са за полето на контролното коригиращо число (ККЧ).

² Записват се 512+4 байта за всеки сектор, като четирите байта са за полето на контролното коригиращо число (ККЧ)

Управляващи последователности за печатащото устройство M88

Код	Функция
NUL	<p>Празен Използва се с ESC B и с ESC D като краен ограничител на списък. NUL се използва също така и с други управляващи кодове на принтера за избор на режим (напр. ESC S).</p> <p>Пример: LPRINT CHR\$(0);</p>
BEL	<p>Звънец Предизвиква издаване на звук от зумера на принтера в продължение на 1 s.</p> <p>Пример: LPRINT CHR\$(7);</p>
HT	<p>Хоризонтална табулация Табулира до хоризонталната позиция за спиране. Когато се включи захранването или принтерът се установи в начално състояние, позициите за спиране се установяват през всеки осем колонки. Позициите за спиране могат да бъдат променени чрез командата ESC D.</p> <p>Пример: LPRINT CHR\$(9);</p>
LF	<p>Нов ред Отпечатва данните, останали в буфера на принтера, и премества хартията с един ред нагоре. Интервалите между редовете са 1/6 инча (4,23 mm), освен ако параметърът не е отменен чрез ESC A, ESC 0, ESC 1, ESC 2 или ESC 3.</p> <p>Пример: LPRINT CHR\$(10);</p>
VT	<p>Вертикална табулация Кодът VT се обработва като LF.</p> <p>Пример: LPRINT CHR\$(11);</p>
FF	<p>Нова страница Хартията се подава до началото на следващата страница. Началото на страницата се определя от разположението на хартията по време на включването на захранването или при начално установяване на принтера. Следващото начало на страница е на 11 инча (279,4 mm) от това положение. За промяна на дължината на страницата може да бъде използван ESC C.</p> <p>Пример: LPRINT CHR\$(12);</p>
CR	<p>Връщане на каретката Завършва реда, на който се е установил принтерът и отпечатва данните, останали в неговия буфер (принтерът не извършва операция за преместване на нов ред).</p> <p>Пример: LPRINT CHR\$(13);</p> <p>Езикът БЕЙСИК добавя команда LF към всяка CR, освен ако не се избере Символен набор 1, не се добави 128 към CR и не се осъществи връщане към Символен набор 2 (вж. ESC 6 и ESC 7).</p>

Код	Функция
	Пример: LPRINT "0000"; CHR\$(27);CHR\$(55);CHR\$(13+128); CHR\$(27); CHR\$(54); "////"
SO	Разтеглине (двойна ширина) Променя режима на печат на <i>Двойна ширина</i> . Командите CR, LF, DC4 и CAN отменят въведения чрез SO режим на печат <i>Двойна ширина</i> . За непрекъснатото печатане с двойна ширина се използва ESC W. Пример: LPRINT CHR\$(14);
SI	Свиване (сгъстяване) Променя режима на печат на <i>Сгъстен шрифт</i> . Пример: LPRINT CHR\$(15);
DC2	Управление на устройството 2 (анулиране на сгъстяването) Анулира печатането в режим <i>Сгъстен шрифт</i> , въведен от SI. Пример: LPRINT CHR\$(18);
DC4	Управление на устройството 4 (премахване на двойната ширина) Анулира печатането в режим <i>Двойна ширина</i> , въведен от SO. Пример: LPRINT CHR\$(20);
CAN	Анулиране Изчиства буфера на принтера, без да отпечата данни. Всички управляващи кодове (с изключение на SO) остават в сила. Пример: LPRINT CHR\$(24);
ESC	Преход Указва на принтера, че следващите изпратени данни представляват команда за принтера (вж. следващия списък от команди). Пример: LPRINT CHR\$(27);
ESC -	Подчертаване Формат: ESC - n; ESC -, следван от 1, отпечата всички следващи данни с подчертаване. ESC -, следван от 0 (нула), отменя режима на печат с подчертаване. Пример: LPRINT CHR\$(27);CHR(45);CHR\$(1);
ESC 0	Стъпка на подаване 1/8 инча Променя стъпката на подаване на хартията на 1/8 инча (3,175 mm). Пример: LPRINT CHR\$(27);CHR\$(48);
ESC 1	Стъпка на подаване 7/72 инча Променя стъпката на подаване на хартията на 7/72 инча (2,47 mm). Пример: LPRINT CHR\$(27);CHR\$(49);
ESC 2	Стартиране на режим <i>Променлива стъпка на подаване</i> ESC 2 е изпълнителна команда за ESC A. Ако не е зададена команда ESC A, подаването на редовете се връща на 1/6 инча (4,23 mm).

Код	Функция
ESC 3	<p>Пример: LPRINT CHR\$(27);CHR\$(50);</p> <p>Променлива стъпка на подаване (само за графичен принтер) Формат: ESC 3;<i>n</i>; Променя стъпката на подаване на хартията на <i>n</i>/216 инча (1/216 инча е 0,1176 mm). На <i>n</i> трябва да бъде присвоена стойност от 1 до 255. Примерът установява стъпката на подаване на хартията на 54/216 (1/4) инча (6.35 mm).</p> <p>Пример: LPRINT CHR\$(27);CHR\$(51);CHR\$(54);</p>
ESC 6	<p>Избор на Символен набор 2 Избира Символен набор 2 (вж. Символен набор 2 на принтера в приложение 13).</p> <p>Пример: LPRINT CHR\$(27);CHR\$(54);</p>
ESC 7	<p>Избор на Символен набор 1 Избира Символен набор 1 (вж. Символен набор 1 на принтера в приложение 13).</p> <p>Пример: LPRINT CHR\$(27);CHR\$(55);</p>
ESC 8	<p>Игнориране на <i>Край на хартията</i> Позволява на принтера да печата до края на хартията. Принтерът игнорира сигнала от датчика за край на хартията. Тази команда трябва да бъде изпратена преди свършване на хартията.</p> <p>Пример: LPRINT CHR\$(27);CHR\$(56);</p>
ESC 9	<p>Отменяне на игнорирането на <i>Край на хартията</i> Отменя командата за игнориране на края на хартията. ESC 9 се избира, когато се включи захранването или принтерът се установи в начално състояние.</p> <p>Пример: LPRINT CHR\$(27);CHR\$(57);</p>
ESC <	<p>Глава в изходно положение Печатащата глава ще се върне до лявото поле, за да отпечата реда, следващ ESC <. Това става само за един ред.</p> <p>Пример: LPRINT CHR\$(27);CHR\$(60);</p>
ESC A	<p>Установяване на променливата стъпка на подаване Формат: ESC A;<i>n</i>; ESC A установява подаването на нов ред на <i>n</i>/72 инча (1/72 инча е 0,3528 mm). Примерът установява стъпката на подаване на хартията на 24/72 инча (8,47 mm). На принтера трябва да се подаде ESC 2, преди да се промени подаването по редове. Текстът, който следва ESC A;<i>n</i>;, се разпечатва на предварително установения междуредов интервал. Текстът, който следва ESC 2, се разпечатва с новите междуредови разстояния от 24/72 инча (8,47 mm).</p> <p>Може да се използва всякава стъпка между 1/72 и 85/72. Подразбиращата се стойност за тази команда при включване на захранването или след начално установяване е 6/72 инча (4,23 mm).</p>

Код	Функция
ESC C	<p>Пример: LPRINT CHR\$(27);CHR\$(65);CHR\$(24);CHR\$(27);CHR\$(50);</p> <p>Установяване броя редове на страница</p> <p>Формат: ESC C;<i>n</i>;</p> <p>Установява дължината на страницата в брой редове. Командата ESC C трябва да бъде следвана от стойност (от 1 до 127), която указва желаните брой редове на страница. Примерът установява дължината на страницата на 55 реда. При включване на захранването или при начално установяване принтерът възприема по подразбиране 66 реда на страница.</p> <p>Пример: LPRINT CHR\$(27);CHR\$(67);CHR\$(55);</p> <p>Установява инчове на страница</p> <p>Формат: ESC C;<i>n</i>;<i>m</i>;</p> <p>Установява дължината на страницата в инчове (1 инч е равен на 2,54 mm). Тази команда изисква стойност 0 (нула) за <i>n</i> и стойност между 1 и 22 за <i>m</i>.</p> <p>Пример: LPRINT CHR\$(27);CHR\$(67);CHR\$(0);CHR\$(12);</p>
ESC D	<p>Установяване на хоризонталните позиции за табулация</p> <p>Формат: ESC D;<i>n1</i>;<i>n2</i>;...<i>nk</i>;NUL;</p> <p>Примерът показва установяването на хоризонталните позиции за спирање при табулация в 10, 20 и 40 стълбове на принтера. Последователността от номера на стълбове, подредени в нарастващ ред, е следвана от CHR\$(0) — кода NUL. В нормален режим на печат позициите за спиране могат да бъдат между 1 и 80. В състения режим на печат те могат да бъдат установявани до 132. Символите с двойна ширина заемат по две хоризонтални позиции. Максималният брой табулации, които могат да бъдат установени, е 28. За изпълнение на операцията табулация се използва HT — CHR\$(9).</p> <p>Пример: LPRINT CHR\$(27);CHR\$(68);CHR\$(10);CHR\$(20);CHR\$(40);CHR\$(0);</p>
ESC E	<p>Удебеляване</p> <p>Сменя режима на принтера на <i>Удебелен шрифт</i>. Скоростта на печат намалява наполовина при режима на Удебеляване.</p> <p>Пример: LPRINT CHR\$(27);CHR\$(69);</p>
ESC F	<p>Изключване на удебеляването</p> <p>Спира печатането в режима <i>Удебелен шрифт</i>.</p> <p>Пример: LPRINT CHR\$(27);CHR\$(70);</p>
ESC G	<p>Повторен печат</p> <p>Сменя режима на принтера на <i>Повторен печат</i>. Хартията се отстранява на 1/216 инча (0,1176 mm) преди втория ход на печатащата глава.</p> <p>Пример: LPRINT CHR\$(27);CHR\$(71);</p>
ESC H	<p>Изключване на повторния печат</p> <p>Отменя работата в режим на <i>Повторен печат</i>.</p> <p>Пример: LPRINT CHR\$(27);CHR\$(72);</p>

Код	Функция
ESC J	Подаване на хартията на $n/216$ инча Формат: ESC J; <i>n</i> ; Когато към принтера бъде изпратен ESC J, хартията ще бъде подадена с $n/216$ инча ($1/216$ инча е 0.1176 mm). Стойността на n трябва да бъде между 1 и 255. Примерът предизвиква подаване на хартията през 50,216 инча (5.88 mm). ESC J се анулира, след като се извърши подаването на хартията. Пример: LPRINT CHR\$(27);CHR\$(74);CHR\$(50);
ESC K	Графичен режим с 480 позиции на линия Формат: ESC K; <i>n1:n2:v1:v2:...vk</i> ; Сменя режима от Текстов на Графичен с линейно точково изображение. Числата $n1$ и $n2$ са с размер по 1 байт и задават пълния брой байтове за данни, необходими за една линия на графичното изображение. Пълният брой k на данните $v1$ до vk не може да превишава 480 и се изчислява по формулата $n1 + 256 * n2$. На всеки хоризонтален ред могат да се отпечатват до 8 вертикални точки. Данните за графичното изображение могат да бъдат смесвани на същия ред с данни за текстов режим. Пример: LPRINT CHR\$(27);CHR\$(49);
ESC L	Графичен режим с 960 позиции на линия Формат: ESC L; <i>n1:n2:v1:v2:...vk</i> ; Сменя режима от Текстов на Графичен с линейно точково изображение. Въвеждането е както при ESC K. В този режим скоростта на печат е равна на половината от тази на режима с 480 позиции на линия, но може да възпроизвежда графичния образ с по-голяма плътност. Броят на байтовете за данни на графичното изображение k се изчислява по формулата $n1 + 256 * n2$, но не може да бъде по-голям от 960. Числото $n1$ е в интервала от 0 до 255.
ESC N	Установяване на прескачане на перфорацията Формат: ESC N; <i>n</i> ; Установява функцията за прескачане на перфорацията между страниците. Числото след ESC N указва броя на редовете, за които се прескача перфорацията, като стойността му е от 1 до 127. Примерът показва прескачане на перфорацията с 12 реда. При дължина на страницата 66 реда това ще предизвика отпечатването на 54 реда и подаване на хартията на 12 реда. ESC N се отменя всеки път, когато се променя дължината на страницата (ESC C). Пример: LPRINT CHR\$(27);CHR\$(78);CHR\$(12);
ESC O	Отменяне на прескачането на перфорацията Отменя функцията за прескачане на перфорацията. Пример: LPRINT CHR\$(27); CHR\$(79);
ESC S	Долно и горно индексирание Формат: ESC S; <i>n</i> ; Установява режим на принтера <i>Долно индексирание</i> , когато ESC S е следван от единица. Когато ESC S е следван от нула, принтерът установява режим <i>Горно индексирание</i> .

Код	Функция
ESC T	<p>Пример: LPRINT CHR\$(27);CHR\$(83);CHR\$(1);</p> <p>Премахване на индексирането</p> <p>Принтерът прекратява печата в режим <i>Индексиране</i>.</p>
ESC U	<p>Пример: LPRINT CHR\$(27);CHR\$(84);</p> <p>Еднопосочен печат</p> <p>Формат: ESC U:n;</p> <p>След въвеждането на ESC U, принтерът ще печата само отляво надясно. Когато ESC U е следван от 0 (нула), операцията печат отляво надясно се отменя. Еднопосочният печат (ESC U) осигурява по-прецизно установяване на началната позиция за печат и по-добро качество.</p>
ESC W	<p>Пример: LPRINT CHR\$(27);CHR\$(85);CHR\$(1);</p> <p>Двойна ширина</p> <p>Формат: ESC W:n;</p> <p>Когато е следвана от 1, последователността променя режима на <i>Двойна ширина</i>. Този режим не се отменя от операциите преминаване на нов ред или DC4. Отменя се с ESC W, следвана от 0 (нула).</p>
ESC Y	<p>Пример: LPRINT CHR\$(27);CHR\$(87);CHR\$(1);</p> <p>Графичен режим с разделителна способност 960 бита</p> <p>Формат: ESC Y:n1:n2;v1: v2:...vk;</p> <p>Сменя режима от Текстов на Графичен с разделителна способност 960 бита. По време на тази операция, принтерът печата с нормална скорост и не може да печата точки в последователни позиции от реда. Въвеждането на данни е както при ESC L.</p>
ESC Z	<p>Формат: ESC Z:n1:n2;v1: v2:...vk;</p> <p>Сменя режима от Текстов на Графичен с разделителна способност 1920 бита. Въвеждането на входните данни за тази операция е подобно на това при другите графични режими с ниска и повишена разделителна способност. ESC Z печата с нормална скорост, но може да печата само на всяка трета позиция на точка.</p>

Знаков генератор на печатащото устройство M88

								0	0	0	0	1	1	1	1								
								0	0	1	1	0	0	1	1								
								0	1	0	1	0	1	0	1								
87	86	85	84	83	82	81		0	1	2	3	4	5	6	7								
			0	0	0	0	0	0	ПУС NUL			32	PP SP	48	0	64	⊙	80	P	96	'	112	p
			0	0	0	1	1					33	!	49	1	65	A	81	Q	97	q	113	q
			0	0	1	0	2			18	СУ2 DC2	34	"	50	2	66	B	82	R	98	б	114	г
			0	0	1	1	3					35	#	51	3	67	C	83	S	99	с	115	е
			0	1	0	0	4			20	СУ4 DC4	36	\$	52	4	68	D	84	T	100	d	116	т
			0	1	0	1	5					37	%	53	5	69	E	85	U	101	e	117	u
			0	1	1	0	6					38	&	54	6	70	F	86	V	102	f	118	v
			0	1	1	1	7			7	3B BEL			55	7	71	G	87	W	103	g	119	w
			1	0	0	0	8					40	(56	8	72	H	88	X	104	h	120	x
			1	0	0	1	9					41)	57	9	73	I	89	Y	105	i	121	y
			1	0	1	0	10			10	ПС LF			42	*	58	:	74	J	90	Z	106	z
			1	0	1	1	11			27	AP2 ESC			43	+	59	;	75	K	91	[107	{
			1	1	0	0	12			12	ПФ FF			44	,	60	<	76	L	92	/	108	
			1	1	0	1	13			13	ВК CR			45	-	61	=	77	M	93]	109	~
			1	1	1	0	14			14	ВМХ SO			46	.	62	>	78	N	94	^	110	я
			1	1	1	1	15			15	ВХ SI			47	/	63	?	79	O	95	_	111	о

Седмбитов код, набор 0 (КОИ-7 Н0)

87	86	85	84	83	82	81
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

0	1	2	3	4	5	6	7
0	0	ПУС NUL					
1	1						
2	2		18	CY2 DC2	"		
3	3				#		
4	4		20	CY4 DC4	+		
5	5				%		
6	6						
7	7	3B BEL					
8	8						
9	9						
10	A	10 LF					
11	B		27	AP2 ESC	+		
12	C	12 FF					
13	D	13 BK CR					
14	E	14 ВЫХ SO					
15	F	15 ВХ SI					

0	1	2	3	4	5	6	7
0	0	0	0	1	1	1	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71
72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87
88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103
104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	

Седмбитов код, набор 1 (КОИ-7 Н1)

87	86	85	84	83	82	81	0	0	0	0	1	1	1	1
							0	0	1	1	0	0	1	1
							0	1	0	1	0	1	0	1

	0	1	2	3	4	5	6	7
0	0	ПУС NUL						
1	1							
2	2		18	СУ2 DC2				
3	3							
4	4		20	СУ4 DC4				
5	5							
6	6							
7	7	7	3В BEL					
8	8							
9	9							
10	А	10	ПС LF					
11	В		27	AP2 ESC				
12	С	12	ПФ FF					
13	Д	13	ВК CR					
14	Е	14	ВМХ SO					
15	Ф	15	ВХ SI					

32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
NP SP	!	"	#	↓	%	&	'	()	*	+	,	-	.	/
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
⊗	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	
П	Я	Р	С	Т	Ч	Ш	Ъ	Ь	Э	Ю	Я	Щ	Ъ	Ч	

Седембитов код, набор 2 (КОИ-7 Н2)

КРАТЪК РЕЧНИК НА ОБЩОПРИЕТИ СЪКРАЩЕНИЯ И ОЗНАЧЕНИЯ НА АНГЛИЙСКИ ЕЗИК

- ACK** (acknowledge) — символ за потвърждение на приет сигнал
- AI** (artificial intelligence) — изкуствен интелект
- ASCII** (American standard code for information interchange) — стандартен код на САЩ за обмен на информация. Състои се от 7-битови символи (8-битови — заедно с бита за четност), като освен букви в кода се съдържат управляващи и графични символи
- BASIC** (beginner's all-purpose symbolic instruction code) — език за програмиране от високо ниво със сравнително малък брой команди и прост синтаксис (БЕЙСИК)
- bps** (bit per second) — единица за измерване на скорост на обмен на информация, изразяваща брой предадени битовете за секунда
- BS** (back space) — клавиш за връщане с една позиция назад на курсора на екрана
- BSC** (binary synchronous communications) — байтово ориентиран синхронен протокол за предаване на данни
- CAD** (computer-aided design) — системи за автоматизирано проектиране с помощта на компютри
- CAM** (computer-aided manufacturing) — системи за автоматизирано производство с помощта на компютри
- CCITT** (Comite Consultatif International Telegraphique et Telephonique) — Европейски комитет за стандарти в областта на телекомуникациите
- CMOS** (complementary metal oxide semiconductor) — технология за подготвяне на полупроводникови прибори с много ниска консумация
- CRC** (cyclic redundancy check) — контролна сума за проверка на предадени или обработени данни
- Ctrl** (control) — клавиш, с помощта на който се извършват управляващи процедури
- DBMS** (database management system) — система за управление на база от данни (СУБД)
- DIP** (dual-in-line package) — корпус за интегрални схеми с двуредно разположение на изводите
- DMA** (direct memory access) — директен достъп до паметта за обмен на данни между оперативната памет и други устройства без участието на процесора (ДМДП)
- DOS** (disk operating system) — дискова операционна система (ДОС)
- dot matrix** — матрица от точки, с която се формират символите, извеждани на печат или на екран
- DTE** (data terminal equipment) — терминална апаратура за обработване на данни
- duplex** (дуплекс) — начин за предаване на информация, при който обменът става независимо в двете посоки
- EBCDIC** (extended binary-coded decimal interchange code) — разширен 8-битов двоично-десетичен код за обмен на информация
- ECL** (emitter-coupled logic) — технология за изготвяне на логически схеми на транзистори с емитерни връзки, при която се постига много високо бързодействие; емитерно свързана логика (ЕСЛ)
- EEPROM** (electronically erasable programmable read-only memory) — електронно изтриваема програмируема постоянна памет, която е енергонезависима
- EPROM** (erasable programmable read-only memory) — изтриваема, програмируема постоянна памет, която е енергонезависима и записаната информация може да се променя след изтриване на данните с ултравиолетова светлина
- ESC** (escape) — символ от кодовата таблица, който обикновено показва, че следващият го символ или група от символи трябва да се интерпретират по особен начин
- FIFO** (first-in, first-out) — дисциплина на обслужване, при която първата пристигнала заявка се обслужва първа (тип опашка)
- FLOPS** (floating-point operations per second) — единица за измерване на броя операции с плаваща запетая за една секунда

- half-duplex** — полудуплекс, т.е. предаване на информация в две посоки, като в даден момент предаването е само в едната посока
- IC (integrated circuit)** — интегрална схема (ИС)
- IFIP (International Federation of Information Processing)** — Международна федерация по обработка на информацията
- instruction set** — съвкупност от инструкции за даден процесор или на език за програмиране
- interface (интерфейс)** — съвкупност от средства за свързване на отделни устройства, програми или системи
- INR (interrupt)** — прекъсване; събитие, което прекъсва дадена последователност от действия, като активира друга последователност, предвидена за конкретния вид прекъсване
- I/O (input/output)** — вход — изход
- IPL (initial program loader)** — начално зареждане на системата
- ISO (International Standards Organisation)** — Международна организация за стандартизация
- LAN (local area network)** — локална мрежа (ЛМ)
- LCD (liquid-crystal display)** — индикатор или екран с течни кристали
- LCN (local computer network)** — локална изчислителна мрежа (ЛИМ)
- LIFO (last-in, first-out)** — дисциплина на обслужване, при която последната пристигнала заявка се обслужва първа (тип стек)
- LIPS (logical inferences per second)** — единица за измерване на производителността на ЕИМ от пето поколение, изразяваща броя логически изводи за 1 секунда
- LSI (large-scale integration)** — голяма интегрална схема (ГИС)
- machine code (машинен код)** — код, който се изпълнява директно от процесора
- matrix printer (матричен принтер)** — печатачо устройство, на което всеки символ се отпечатва като комбинация от точки
- microcode (микрокод)** — една или повече микроинструкции
- microinstruction (микроинструкция)** — елементарна машинна инструкция, която е съставна част от инструкция на процесор
- modem (modulator-demodulator)** — устройство, което преобразува последователни цифрови сигнали в аналогови, подходящи за предаване по телефонна мрежа и обратно (модем)
- modulation (модулация)** — процес на промяна на някоя от характеристиките на един сигнал в съответствие с промяната на друг сигнал, който се нарича модулиращ
- monitor (монитор)** — програмни и апаратни средства, с които се управлява и следи работата на система за обработка на данни
- mouse (мишка)** — устройство за управление на движението на курсора по екрана
- multiprogramming (мултипрограмиране)** — едновременно изпълнение на две или повече програми от компютъра
- NRZ (nonreturn-to-zero)** — метод за кодиране, при който сигналът променя нивото си, когато се кодира логическа единица, и остава въщото ниво, когато се кодира логическа нула
- odd-even check** — проверка за четност или нечетност
- OEM (original equipment manufacturer)** — основен производител на дадено оборудване
- off-line** — автономен, независим от компютъра режим на работа на дадено устройство
- one-shot** — схема, образуваща единичен импулс с необходима продължителност на изхода, съответстващ на всеки входен импулс
- on-line** — неавтономен режим на работа на дадено устройство, свързано към компютъра, непрекъснато обменящо с него информация
- OS (operating system)** — съвкупност от програми за управление на паметта, процесите, вход-изхода, файловата система, ресурсите и цялостната работа на ЕИМ (ОС)
- OSI (open systems interconnection)** — модел на ISO за взаимодействие на информационни системи
- overflow (препълване)** резултатът от аритметична операция превишава разредността на регистъра, в който тя се извършва
- overrun** — загуба на данни, когато приемното устройство не е в състояние да приема данни при висока скорост на предаване

- PAL** (programmable array logic) — програмируема матрична логика
- parity bit** (бит за четност) — контролен бит, който се добавя към група двоични цифри и образува четна или нечетна сума в зависимост от това какъв е контролът
- parity check** (проверка по четност)—образуване на сума от битовете на дадена поредица и сравняването ѝ с бита за четност, от което се получава или не сигнал за грешка
- polling** (полиране) — обхождане на абонатите (устройства или процеси) в определен ред с проверка на тяхна готовност за предаване или приемане на данни
- port** (порт) — многоразредна област за въвеждане или за извеждане на данни от устройство
- PROM** (programmable read-only memory) — еднократно програмируема енергонезависима памет
- RAM** (random-access memory) — памет с произволен достъп. Времето за четене/запис до всички клетки е едно и също. Статичната RAM памет е енергозависима памет на тригери без необходимост от опресняване. Динамичната RAM памет използва транзистори и кондензатори и е необходим цикъл за опресняване на всеки няколко ms
- recoverable error** — възстановима грешка, която може да се коригира, след което изпълнението на програмата продължава
- reverse video** — начин за отделяне на символ, поле или курсор чрез реверсиране (размяна) на цвята с този на фона
- RF-modulator** — устройство за конвертиране на комплексния видеосигнал до чивото на антенния вход за битов телевизионен приемник
- RGB** (red-green-blue) — интерфейс за цветни монитори, използващ като основни цветове червен, зелен и син
- RISC** (reduced instruction set computer) — архитектура на компютри със съкратен набор инструкции
- ROM** (read only memory) — памет, чието съдържание може само да се чете, а не може да се променя (информацията не се губи при изключване на захранването)
- RS-232C** — асинхронен комуникационен интерфейс V.24 или Стък-2 (C2)
- SDLC** (synchronous data link control) — битово ориентиран синхронен протокол за предаване на данни
- start bit** (стартов бит) — бит, указващ на приемника началото на данните (символ или блок)
- timeout** (тайм-аут) — интервал от време, в който се изчаква появата на някакво събитие, например отговор при полиране или адресиране
- VLSI** (very large-scale integration) — свръхвисока степен на интеграция в интегралните схеми, свръхголеми интегрални схеми (СГИС)
- workstation** — работна станция

СЪДЪРЖАНИЕ

	<i>Въведение</i>	5
Глава 1	Общи сведения за персоналните компютри	7
1.1.	Микропроцесорна система	7
1.2.	Технически характеристики на микропроцесорните интегрални схеми	10
1.3.	Организация на управлението и обmena в микропроцесорните системи	11
1.4.	Начини за адресиране в микропроцесорните системи	14
1.5.	Основни модули на персоналните компютри	20
1.6.	Базово програмно осигуряване	23
1.7.	Приложни програми	26
1.8.	Избор на персонален компютър	29
	Литература към гл.1	30
Глава 2	Осемразредни персонални компютри	31
2.1.	Осемразредни персонални компютри от фамилията Правец	32
2.1.1.	Персонален компютър Правец-82	34
2.1.2.	Персонален компютър Правец-8М	42
2.1.3.	Дискова операционна система ДОС 3.3	45
2.2.	Осемразредни персонални компютри с дискова операционна система CP/M	50
2.2.1.	Структура и организация на CP/M 2.2	50
2.2.2.	Периферни устройства и контролери за тях	59
2.2.3.	Програмно осигуряване	69
	Литература към гл.2	69
Глава 3	Шестнадесетразредни персонални компютри	71
3.1.	Архитектура на шестнадесетразредните персонални компютри	71
3.2.	Процесорен модул	74
3.2.1.	Микропроцесори I8086 и I8088	74
3.2.2.	Синхронизация в процесора	83
3.2.3.	Системна шина	86
3.2.4.	Системна памет	95
3.2.5.	Директен достъп до паметта	97
3.2.6.	Система за прекъсване	111
3.2.7.	Таймери на системната платка	119
3.2.8.	Програмируем паралелен интерфейс	124
3.3.	Базова система за вход—изход	128
3.3.1.	Основни характеристики	128
3.3.2.	Програма за самотестване и програма за начално зареждане	130
3.3.3.	Входно-изходни драйверни програми	136
3.4.	Външни запомнящи устройства с дисков носител	139
3.4.1.	Запомнящи устройства с гъвкав магнитен диск	140
3.4.2.	Интерфейс между контролера и ЗУГМД	146
3.4.3.	Контролер на ЗУГМД	150
3.4.4.	Запомнящи устройства с твърд магнитен диск	155
3.4.5.	Интерфейс между контролера и ЗУТМД	159
3.4.6.	Контролер на ЗУТМД	162
3.5.	Видеомонитори и видеоконтролери	167
3.5.1.	Монохроматичен текстов видеомонитор и контролер	168
3.5.2.	Цветен графичен видеомонитор и контролер	172
3.6.	Клавиатура	181

3.6.1.	Клавиатурен интерфейс	182
3.6.2.	Клавиатурен адаптер	183
3.6.3.	Действие на клавиатурата	185
3.7.	Печатащи устройства	187
3.7.1.	Печатащи устройства за 16-разредни персонални компютри	188
3.7.2.	Интерфейси и контролери за печатащи устройства	193
3.8.	Комуникационни адаптери за 16-разредни персонални компютри	197
3.8.1.	Комуникационни адаптери за местно свързване	198
3.8.2.	Комуникационни адаптери за свързване на ПК към локална мрежа МИКРОРИНГ	198
3.8.3.	Комуникационни адаптери за свързване на ПК към локална мрежа МИКРОЛИМ	203
3.8.4.	Комуникационни адаптери за отдалечено свързване	207
3.8.5.	Асинхронни адаптери за ПК	216
3.8.6.	Синхронни адаптери за битово ориентиран протокол SDLC	228
3.8.7.	Синхронни адаптери за байтово ориентиран протокол BSC	243
3.9.	Особености на високопроизводителните персонални компютри от фамилията Правец	248
	Литература към гл.3	249
Глава 4.	Програмно осигуряване на 16-разредни персонални компютри	251
4.1.	Базово програмно осигуряване	251
4.1.1.	Дискова операционна система ДОС-16	251
4.1.2.	БЕЙСИК за Правец-16	260
4.1.3.	Терминален емулатор МИКРОТЕРМ	261
4.1.4.	Екранен редактор	261
4.1.5.	Терминални емулатори за персонални компютри	262
4.1.6.	Система за логическо програмиране ПРОЛОГ-16	263
4.1.7.	Транслатори, работещи под управлението на ДОС-16	264
4.1.8.	Многопотребителска операционна среда ПДОС-16	265
4.2.	Приложно програмно осигуряване	265
4.2.1.	МИКРОФАЙЛ/16	265
4.2.2.	Система за текстообработка ТЕКСТ-16	267
4.2.3.	СУПЕРПЛАН	268
4.2.4.	Интегриран пакет МУЛТИПАК	269
4.2.5.	Помощни програми	271
4.2.6.	Програми за оптимизация	272
4.2.7.	Пакет приложни програми за машиностроенето	274
4.2.8.	Система за проектиране на принципини схеми	275
4.2.9.	Пакет програми за проектиране на печатни платки	276
4.2.10.	Компилатор за релационна база от данни	276
4.2.11.	Универсална графична система УНИГРАФ	277
4.2.12.	Пакети приложни програми за конкретни приложения	277
4.2.13.	Мрежово програмно осигуряване	281
	Литература към гл.4	282
<i>Приложение 1.</i>	Набор инструкции на микропроцесора СМ630	284
<i>Приложение 2.</i>	Набор инструкции на микропроцесора 18080	289
<i>Приложение 3.</i>	Набор инструкции на микропроцесора Z80	292
<i>Приложение 4.</i>	Набор инструкции на микропроцесора 18086	304
<i>Приложение 5.</i>	Съобщения за грешки на БСВИ	310
<i>Приложение 6.</i>	Основни параметри на драйверите на БСВИ	311
<i>Приложение 7.</i>	Набор команди на ИС СМ609	316
<i>Приложение 8.</i>	Типове и номера на грешки, издавани от контролера на ЗУТМД	325
<i>Приложение 9.</i>	Набор команди на контролера на ЗУТМД	326
<i>Приложение 10.</i>	Символи на знаковия генератор на ПК ИЗОТ 1031С	330
<i>Приложение 11.</i>	Кодови таблици на ПК Правец-16	331
<i>Приложение 12.</i>	Управляващи последователности за печатащото устройство * М88	335
<i>Приложение 13.</i>	Знаков генератор на печатащото устройство М88	341
	Кратък речник на общоприетите съкращения и означения на английски език	347

СПРАВОЧНИК ПО ПЕРСОНАЛНИ КОМПЮТРИ

Под общата редакция на ст.н.с.І ст.д.т.н. инж. Кирил Любенов Боянов

Автори: ст.н.с.І ст.д.т.в. инж. *Кирил Любенов Боянов*

к.т.н.инж. *Никола Георгиев Вецев*

инж. *Стефан Списов Мачев*

инж. *Владимир Петров Пантелеев*

ст.н.с. к.т.н. инж. *Александър Георгиев Александров*

ст.н.с.к.т.н. инж. *Владимир Русланов Чилов*

инж. *Христо Георгиев Христов*

Рецензенти: доц.к.т.н.инж. *Павел Мартинов Мартинов*

инж. *Алекси Стефанов Бояклиев*

Българска

Първо издание

Код 03 $\frac{9533115531}{3205-4-88}$

Изд. № 15428

Научен редактор инж. *Румяна Ковачева*

Художник *Трайко Попов*

Художествен редактор *Досю Досев*

Технически редактор *Дарина Асенова*

Коректор *Лиляна Кондева*

Дадена за набор м. януари 1988 г.

Подписана за печат м. май 1988 г.

Излязла от печат м. май 1988 г.

Формат 60 × 90/16

Печ. коли 22

Изд. коли 22

УИК 28.46

Тираж 54 010 + 111

Цена 3,90 лв.

Държавно издателство „Техника“, бул. Руски 6, София

Държавна печатница „Г. Димитров“, София