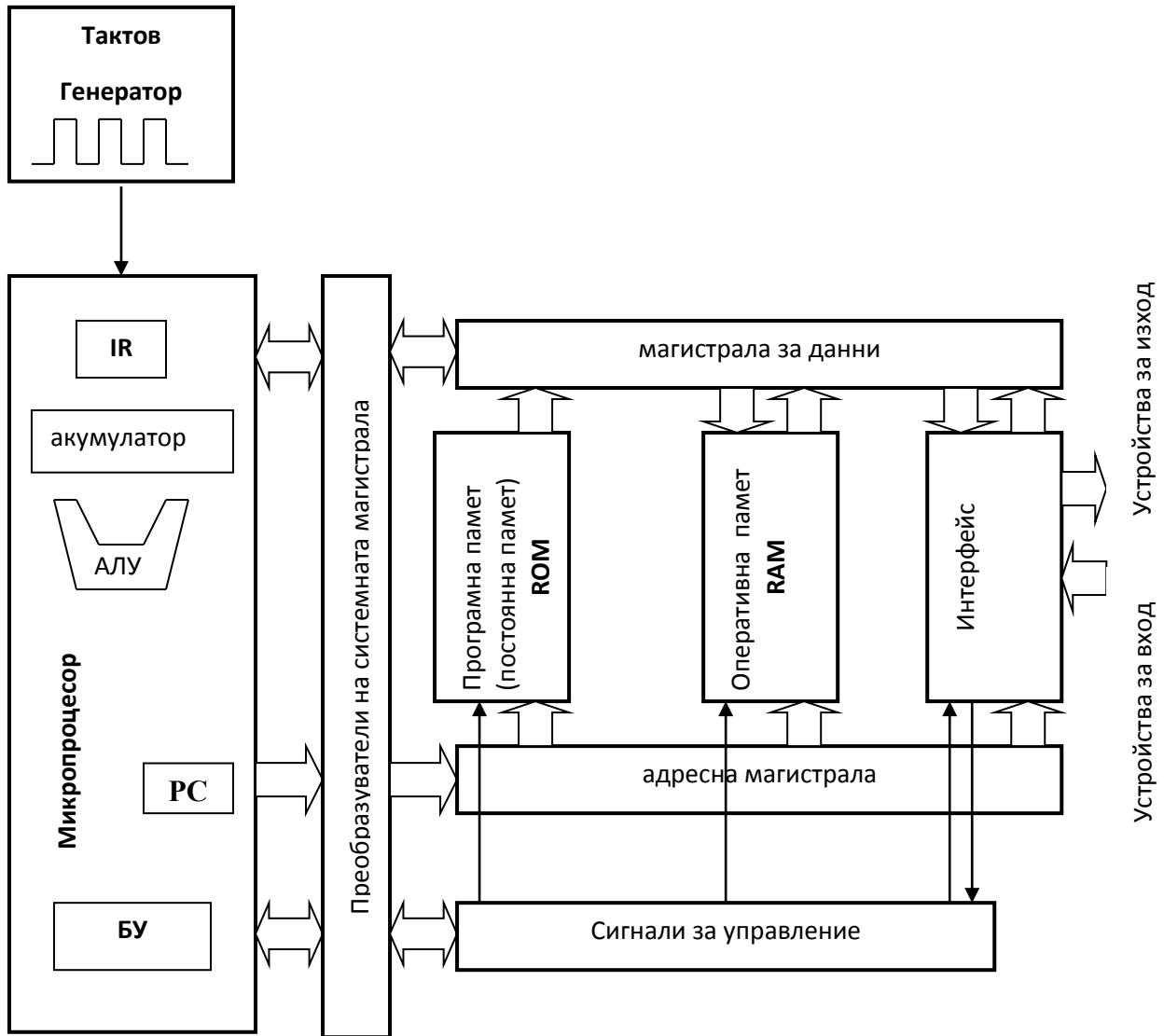
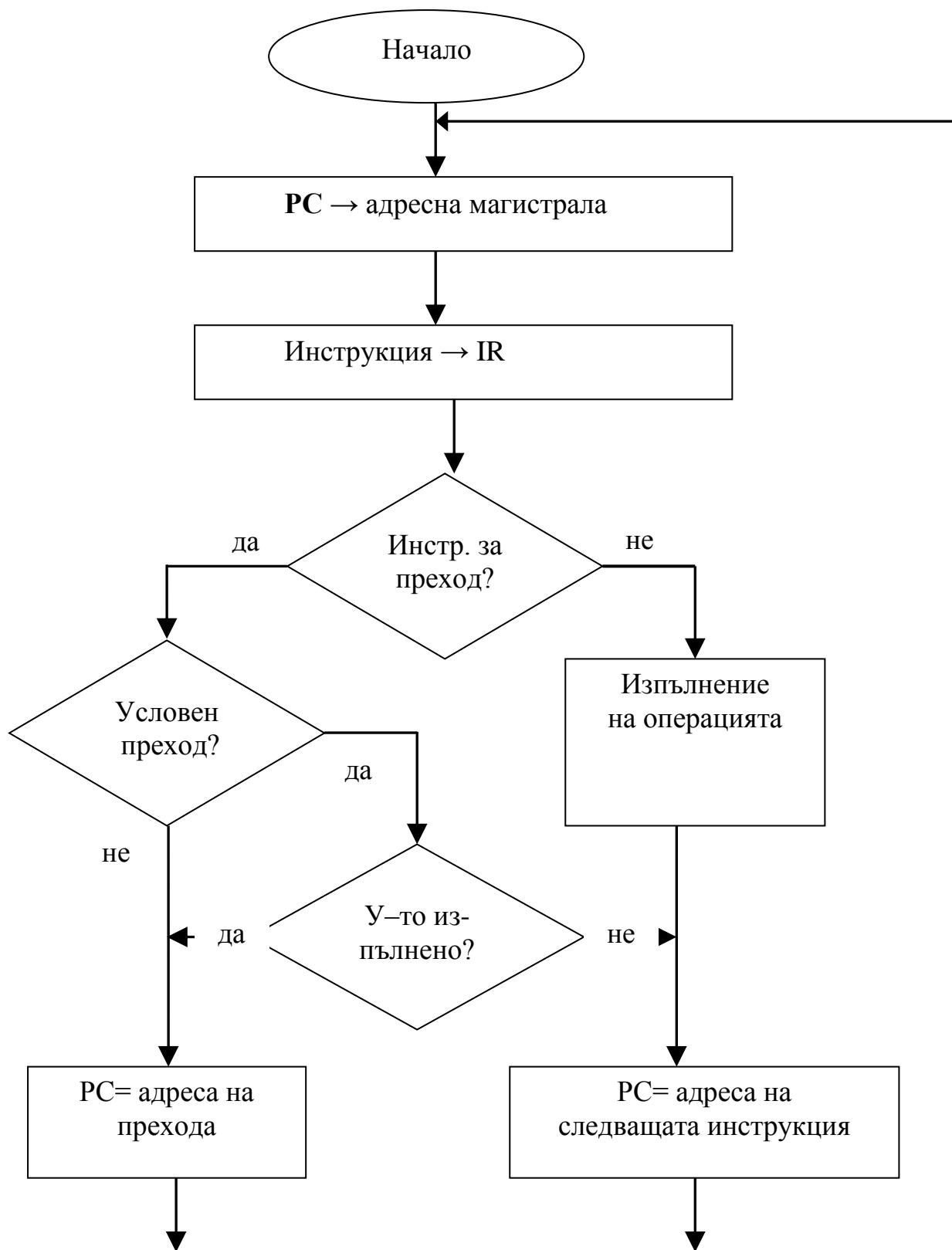


# **Методически материали по микропроцесорна техника**

**2020**



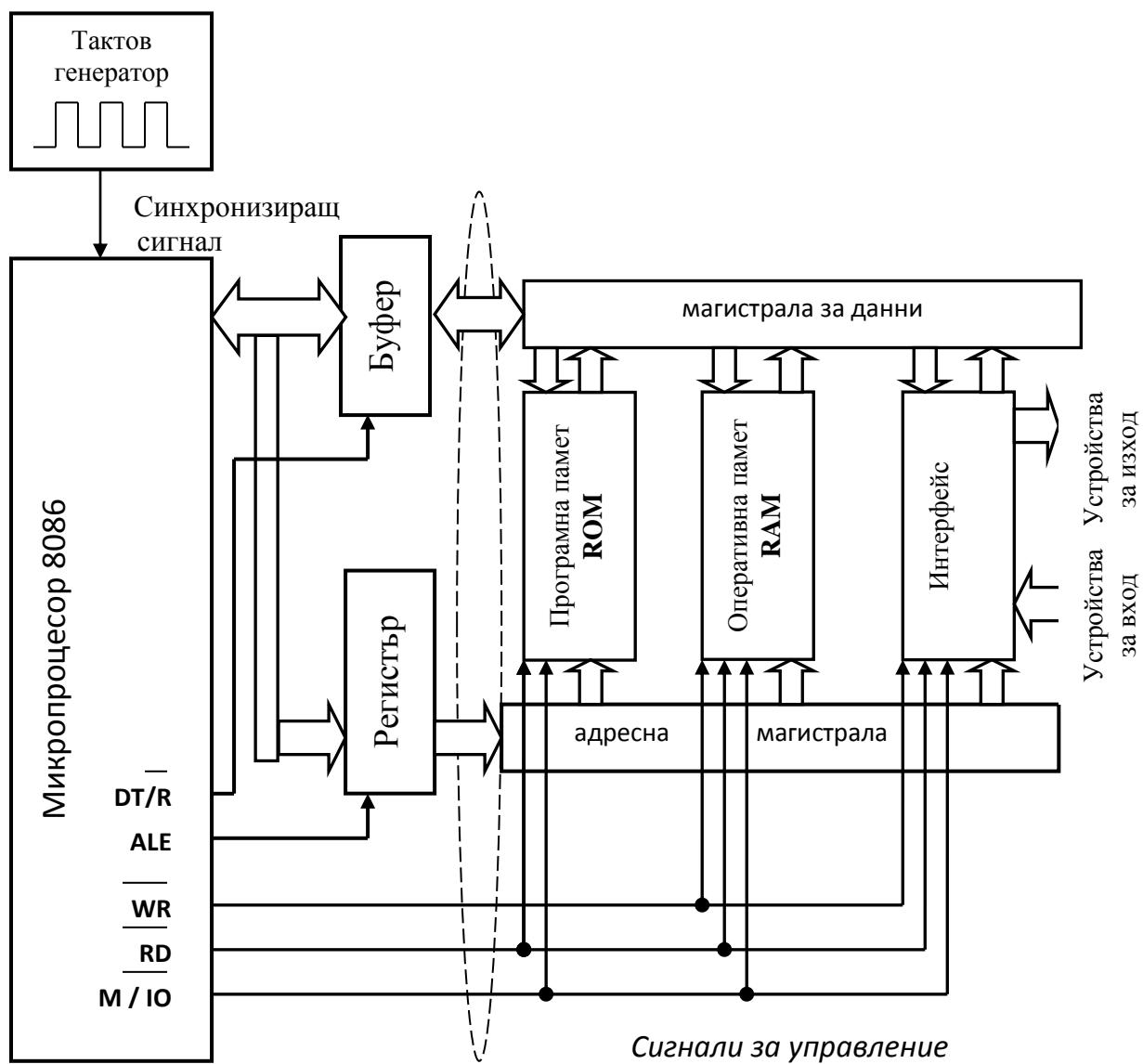
Фиг.1.1. Блокова схема на микропроцесорната система



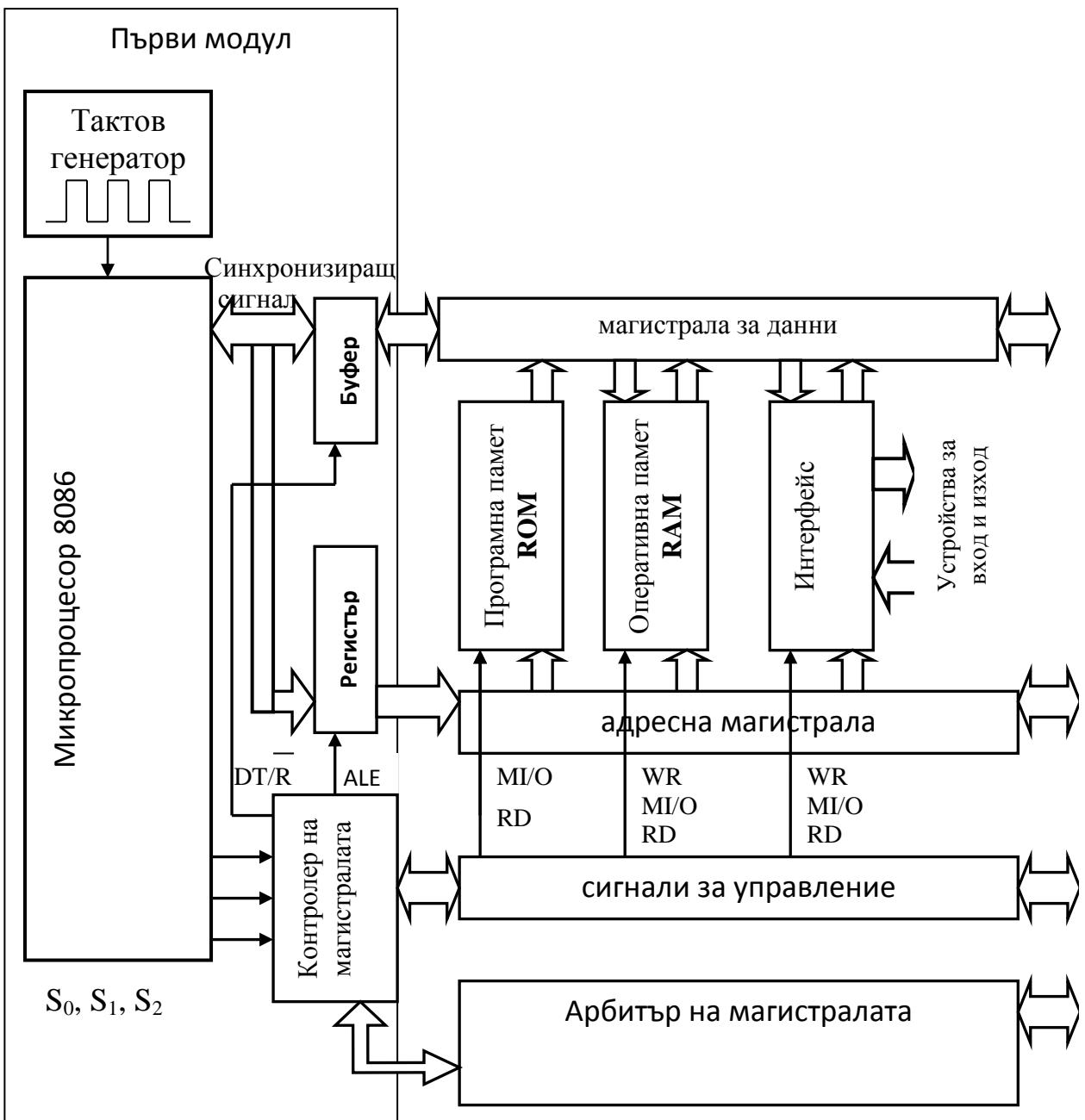
Фиг.1.2. Алгоритъм на изпълнение на инструкциите в микропроцесорна система

Таблица 1.1

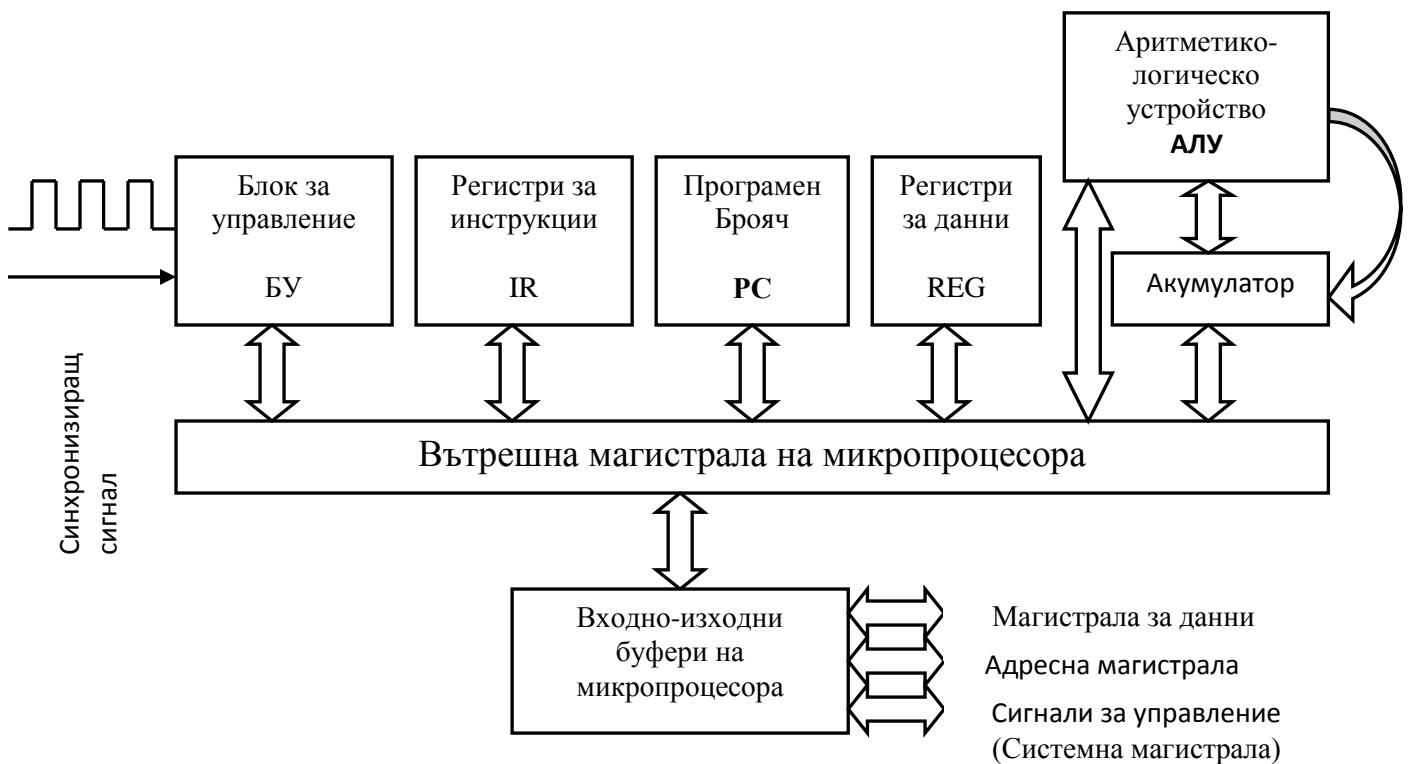
M/IO	RD	WR	Обмен по магистралата за данни
0	0	1	МП чете от интерфейса (МП← интерфейс)
0	1	0	МП записва в интерфейса (МП→интерфейс)
1	0	1	МП чете от паметта (МП←ROM,RAM)
1	1	0	МП записва в паметта (МП→ RAM)
X	1	1	Обмен не се извършва



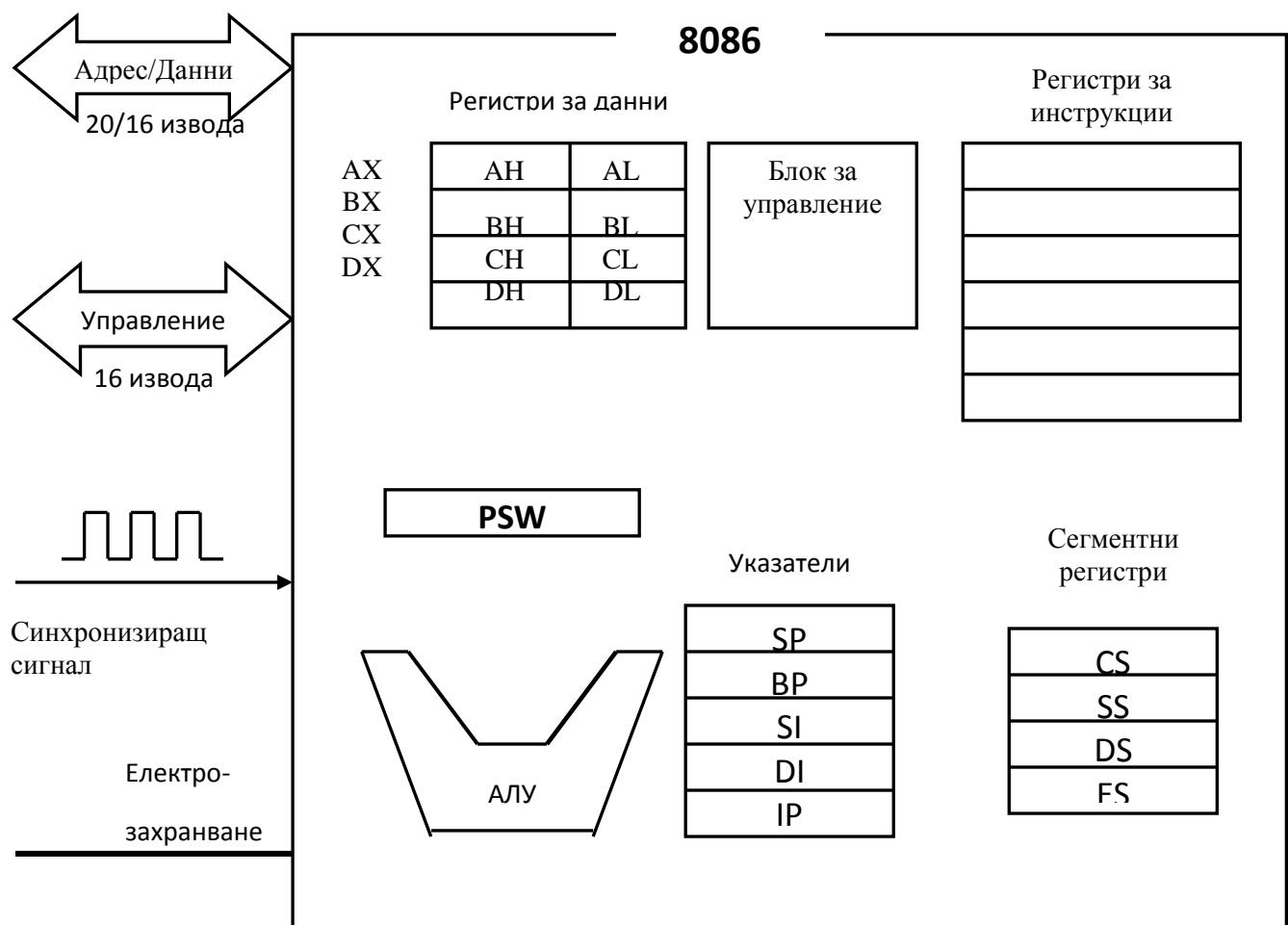
Фиг.1.3. Блокова схема при минимална конфигурация



Фиг.1.4. Блокова схема при максимална конфигурация



Фиг.2.1. Блокова схема на микропроцесор



Фиг.2.2. Архитектурата на микропроцесор 8086

## Позиция в PSW

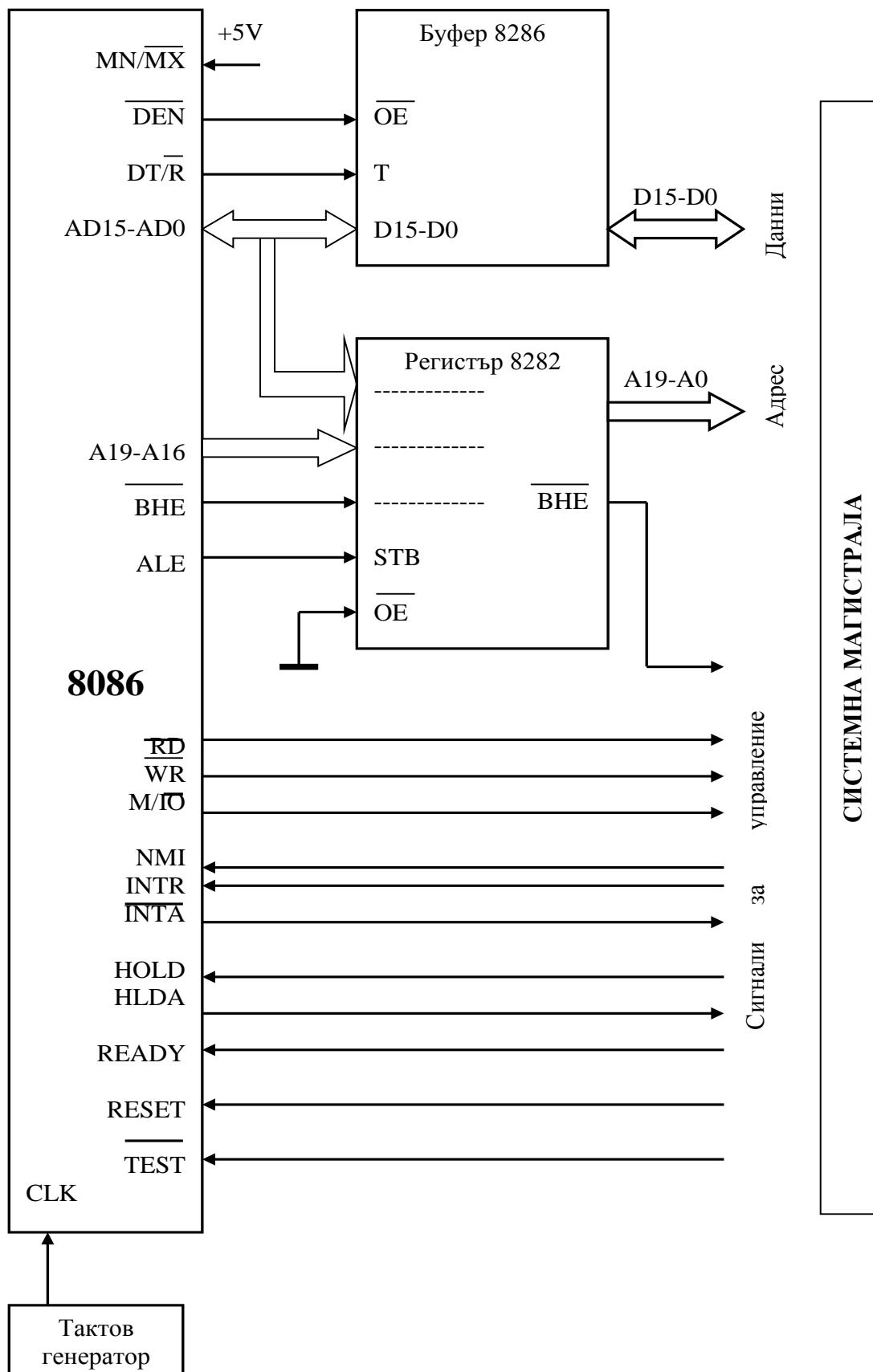
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZP		AF		PF		CF

Обозначение

Фиг.2.3. Регистър на състоянията PSW на микропроцесор 8086

Предназначението им е следното:

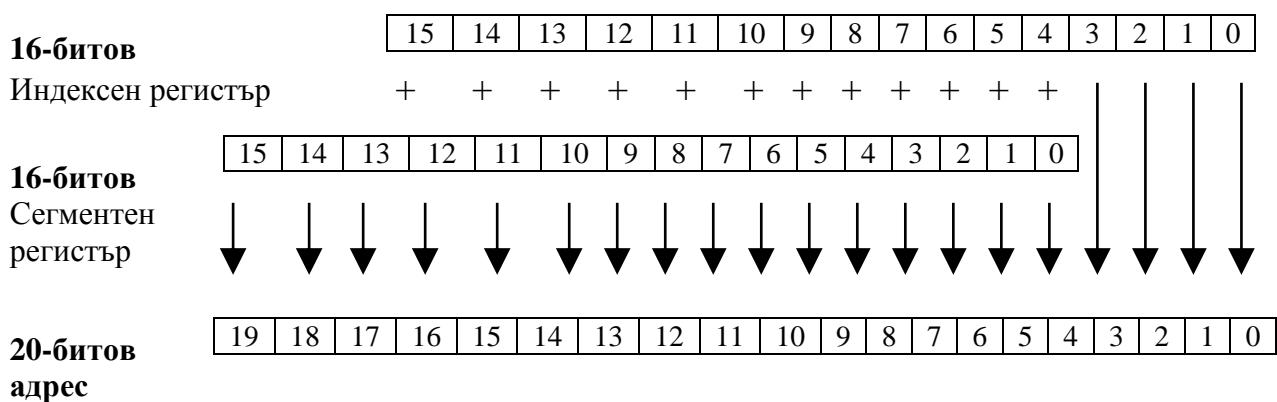
- OF – флаг за препълване. Установява се в 1, ако резултатът е извън допустимия диапазон.
- DF – флаг за посока. Използва се при командите за работа с последователно записани данни в паметта на СНИЗ. Ако е равен на 0, данните се обработват от клетките с най-малък адрес нагоре. При 1 обработката започва от клетката с най-голям адрес надолу.
- IF – флаг за разрешаване на прекъсването. Ако е равен на 1, работата на микропроцесора може да бъде прекъсната от зададеното външно събитие. При IF=0 прекъсването се игнорира.
- TF – флаг за трасиране. Ако е установлен в 1, след изпълнението на всяка инструкция се генерира вътрешно прекъсване. Този флаг се използва най-често за постъпково изпълнение на инструкциите от програмата в процесора на разработването ѝ.
- SF – флаг на знака. Равен е на най-старшия бит от резултата, получен при изпълнението на последната инструкция. При представянето на числата в допълнителен код най-старшия бит определя знака на числото. SF = 1 при отрицателно число и SF = 0 - при положително;
- ZF – флаг за нула. Установява се в 1, когато резултатът от изпълнението на последната инструкция е нула;
- AF – флаг на допълнителния пренос. Използва се при двоично – десетични изчисления и става равен на 1 при пренос от 3-тия бит;
- PF – флаг на четност ( паритет). Установява се в 1, ако младшите 8 бита от резултата съдържат четно число единици. При нечетно число PF = 0;
- CF – флаг за пренос. Равен е на 1, ако при събиране (или изваждане) възникне пренос ( заем) от най-старшия бит.



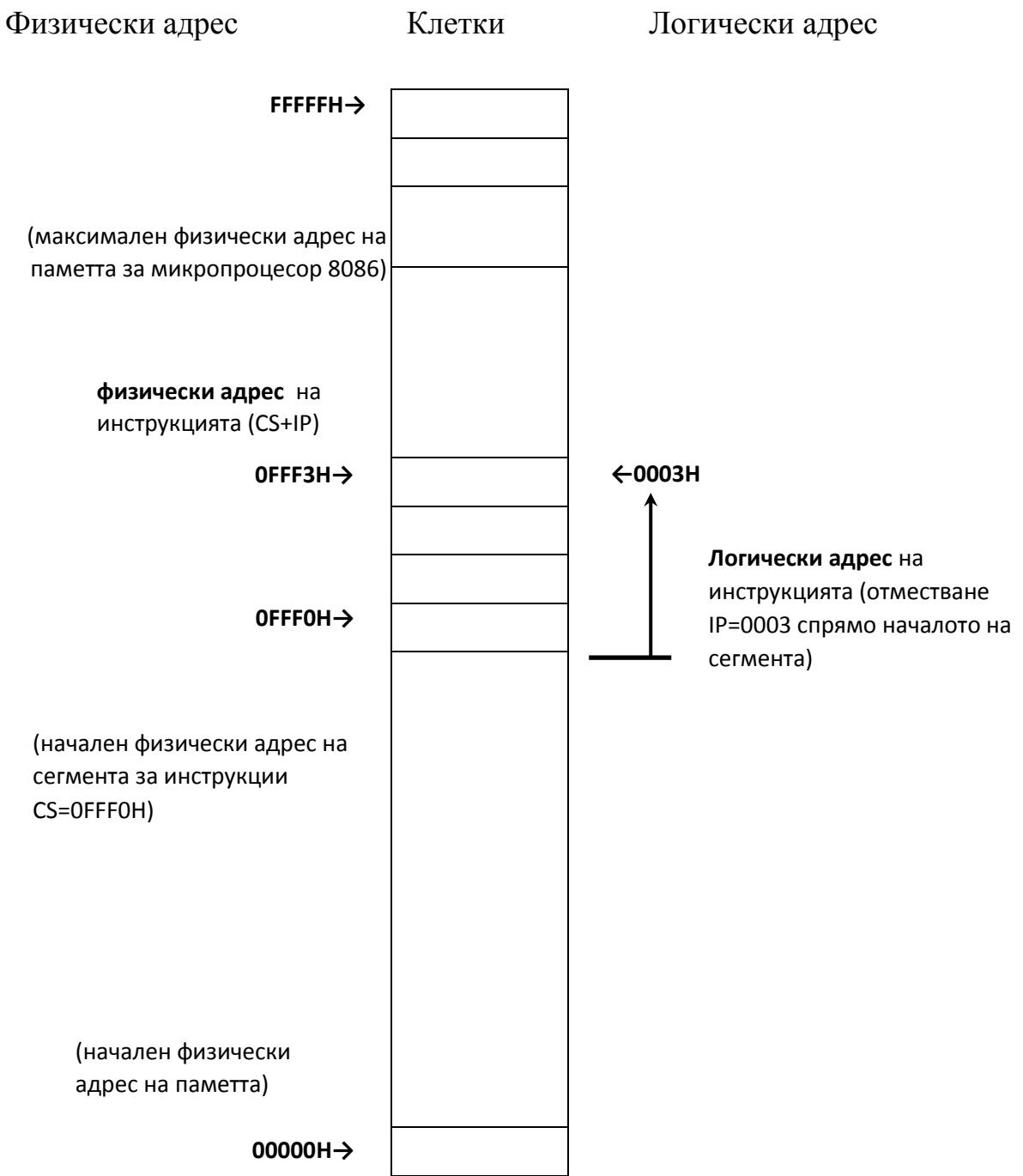
Фиг.2.4. Свързване на 8086 към системната магистрала

Вътрешната архитектура на микропроцесор 8086 е 16-битова и регистрите също са 16-битови. За да може микропроцесорът да работи с 20 – битов адрес, разработчикът е разделил блока с адресни регистри на две групи – сегментни и индексни.

Сегментните регистри CS, SS, DS и ES определят сегмента (частта) от паметта, в която е разположена необходимата клетка. Началният адрес на сегмента е равен на съдържанието на сегментния регистър, изместено с 4 бита наляво. Индексните регистри IP, SP, BP, SI и DI определят конкретния адрес на клетката в този сегмент. Физическият адрес на клетката се изчислява като се събере съдържанието на сегментния регистър (изместено с 4 бита) със съдържанието на индексния регистър. Ако CS=OFFFH (00001111111111 в двоичен вид) и IP=003H (00000000000001), физическият адрес на клетката, адресирана от микропроцесора ще бъде OFFF3H (000011111111110011). Това е онагледено на фиг.3.1.

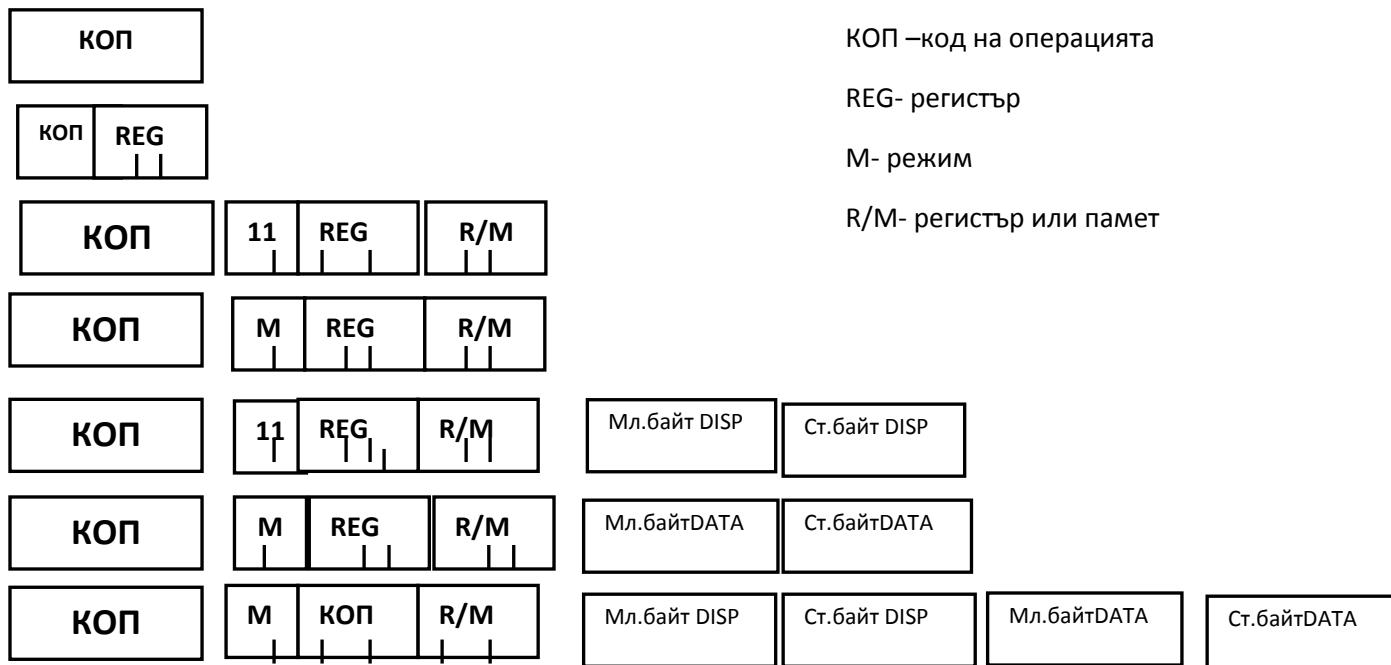


Фиг. 3.1. Изчисляване на 20-битовия адрес в микропроцесор 8086

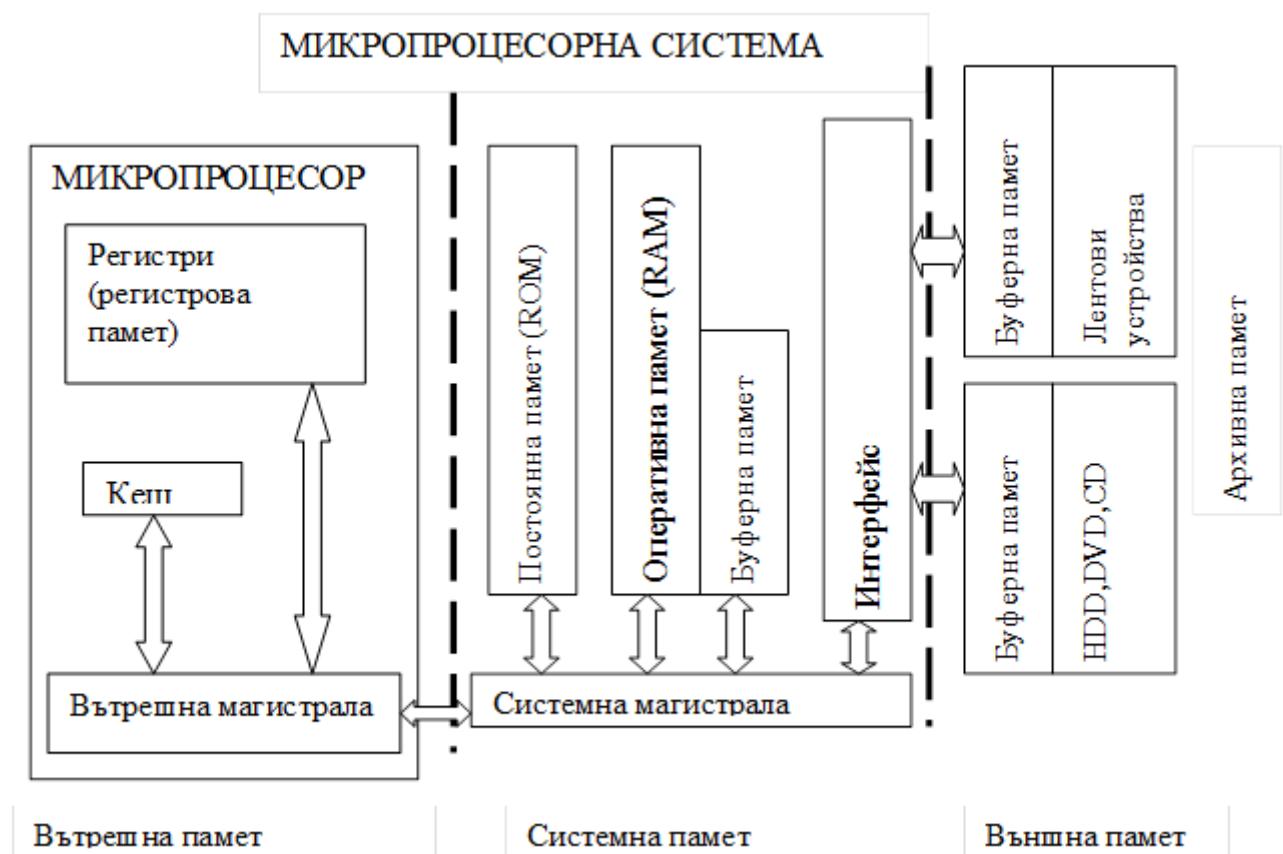


Фиг.3.2. Физически и ефективен адрес на сегментираната памет

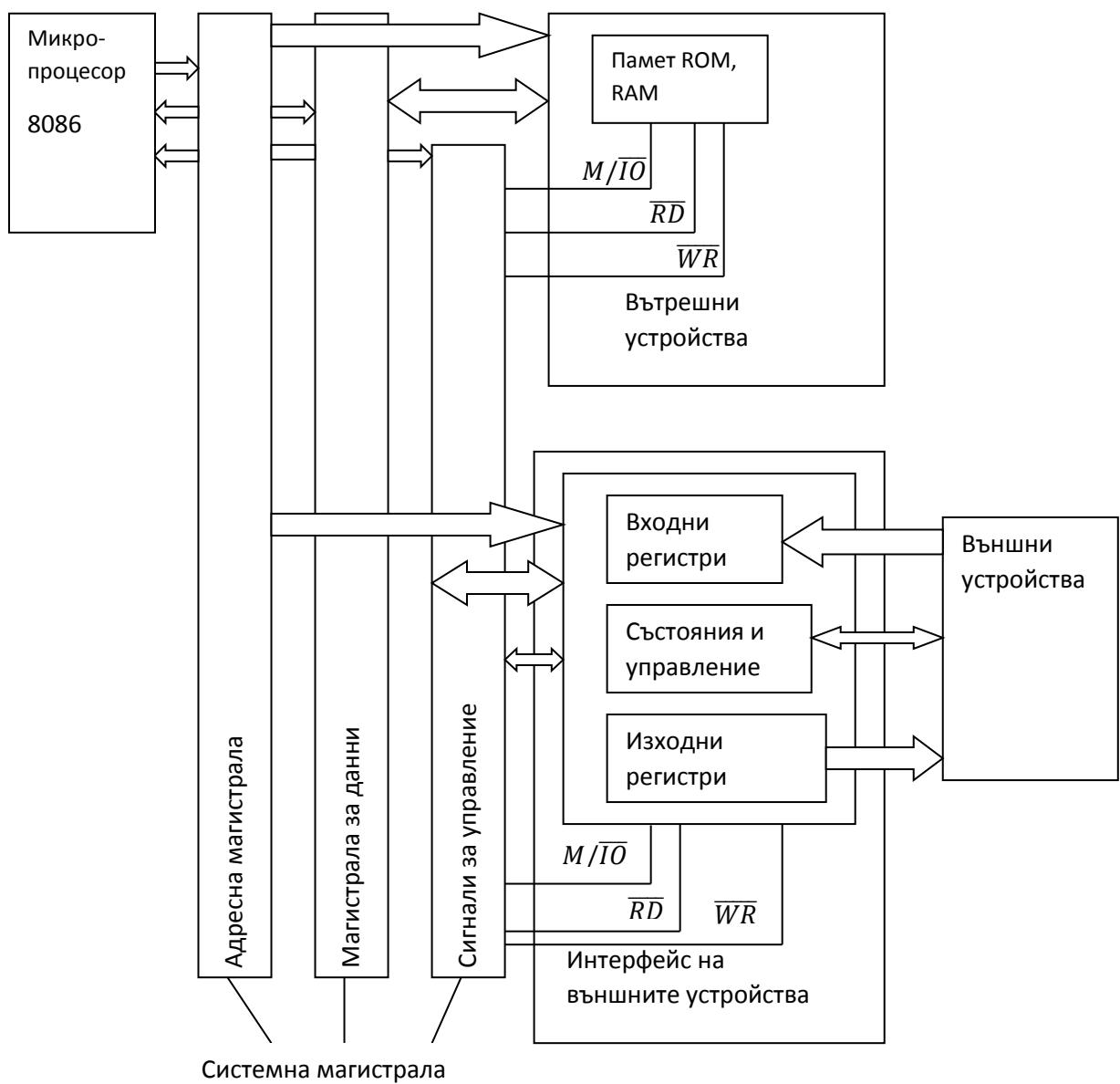
Използването на сегментни регистри по същество разделя адресното пространство на паметта на части. В сегментния регистър е записан началния адрес на сегмента, а индексният регистър съдържа отместването от началото на сегмента, наречено **логически адрес**. Използва се и термина **ефективен адрес**. Нагледно това е показано на фиг. 3.2.



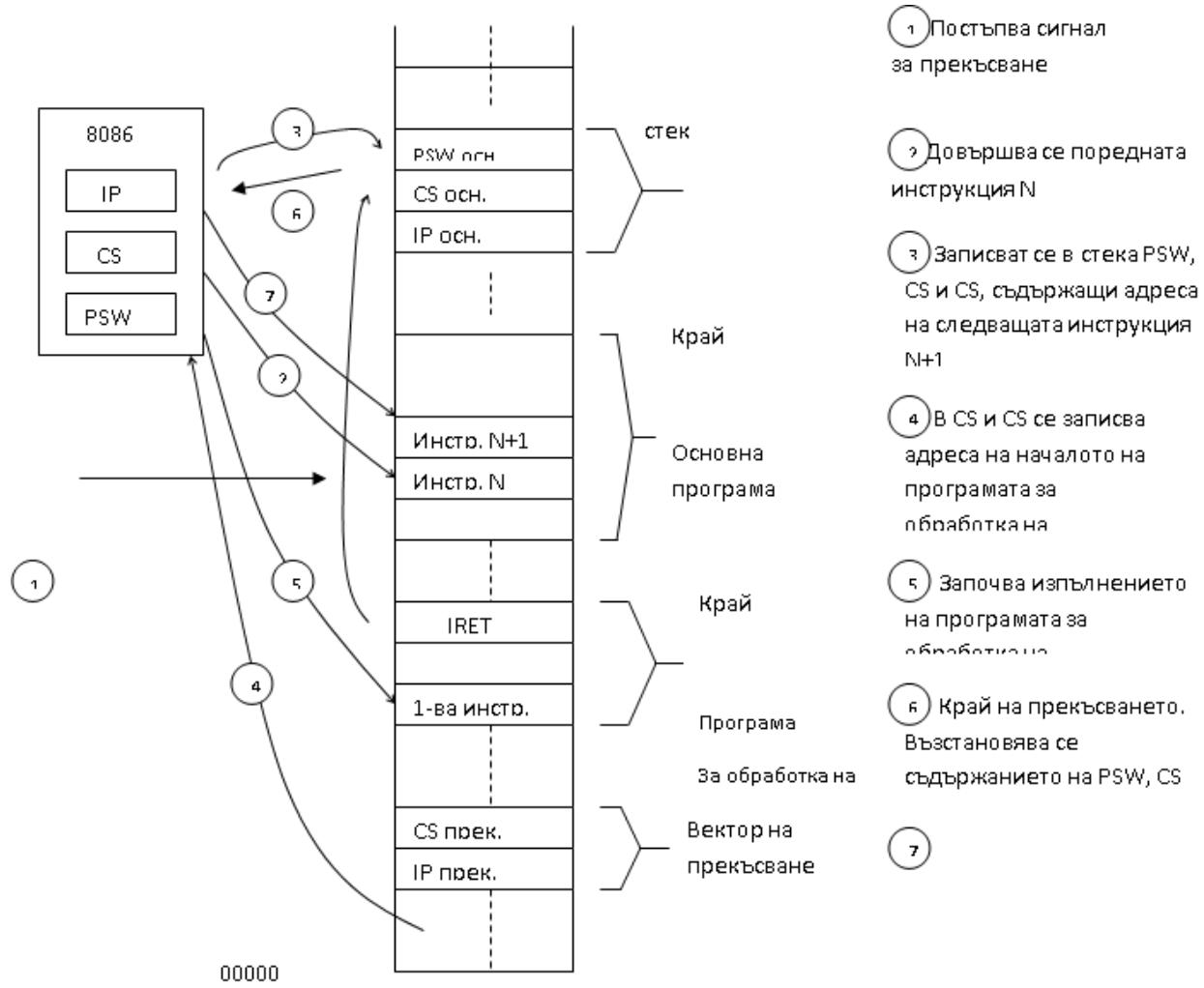
Фиг.4.1. Примерен формат на кода на инструкциите на 8086

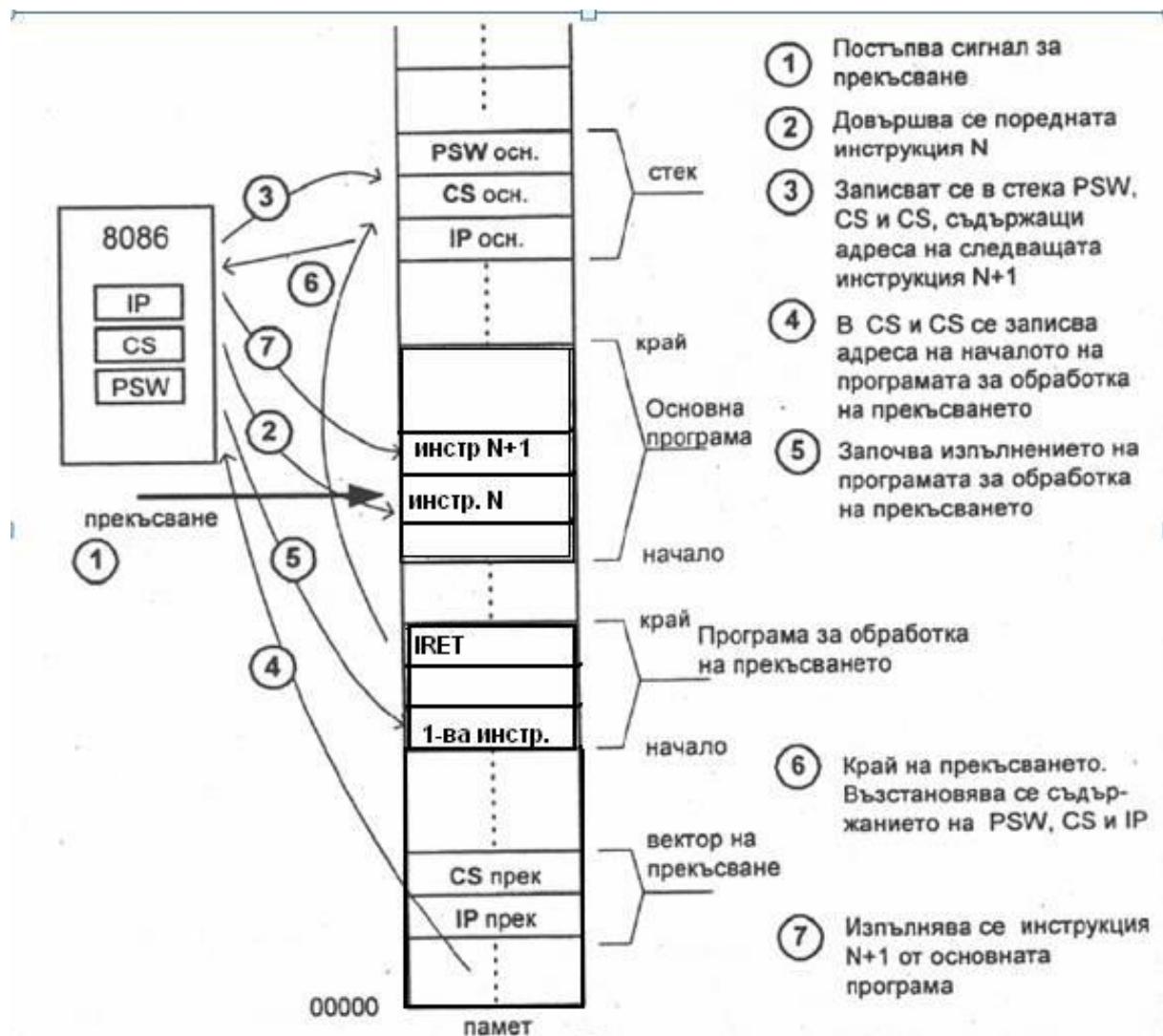


Фиг.4.2. Пример за йерархична структура на паметта

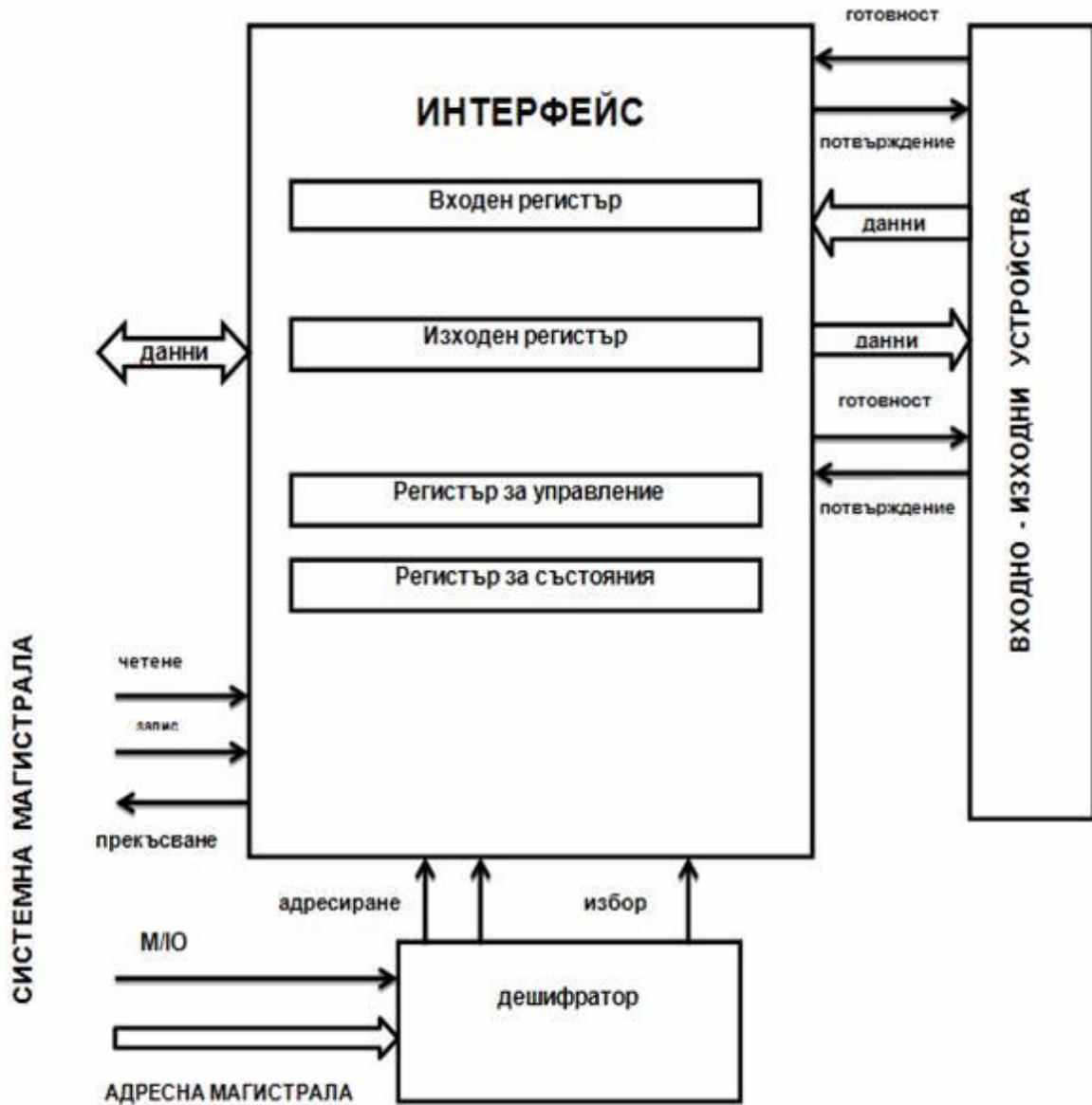


Фиг. 5.1. Вход и изход – блокова схема

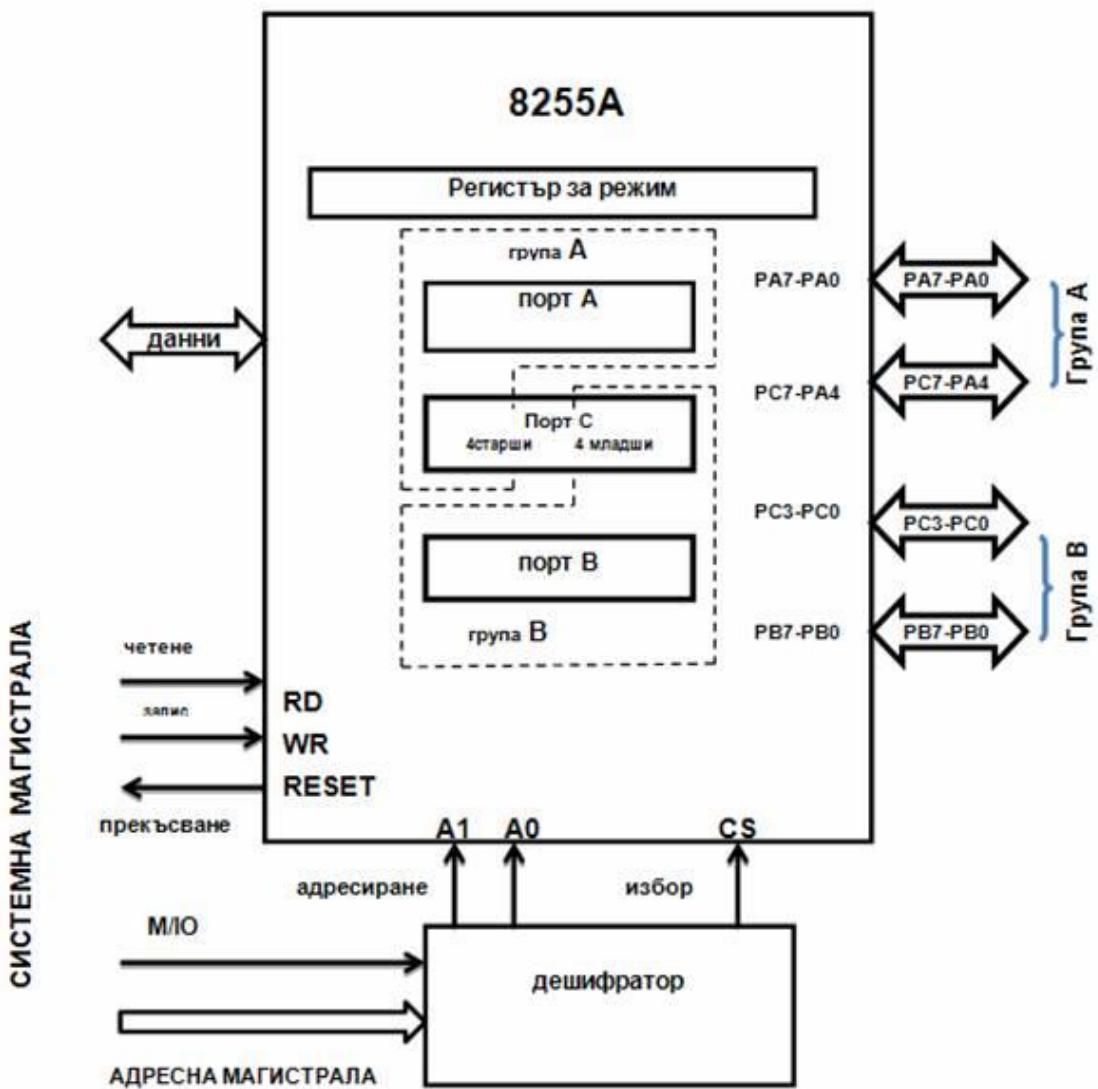




Фиг. 6.1. Обработка на прекъсване



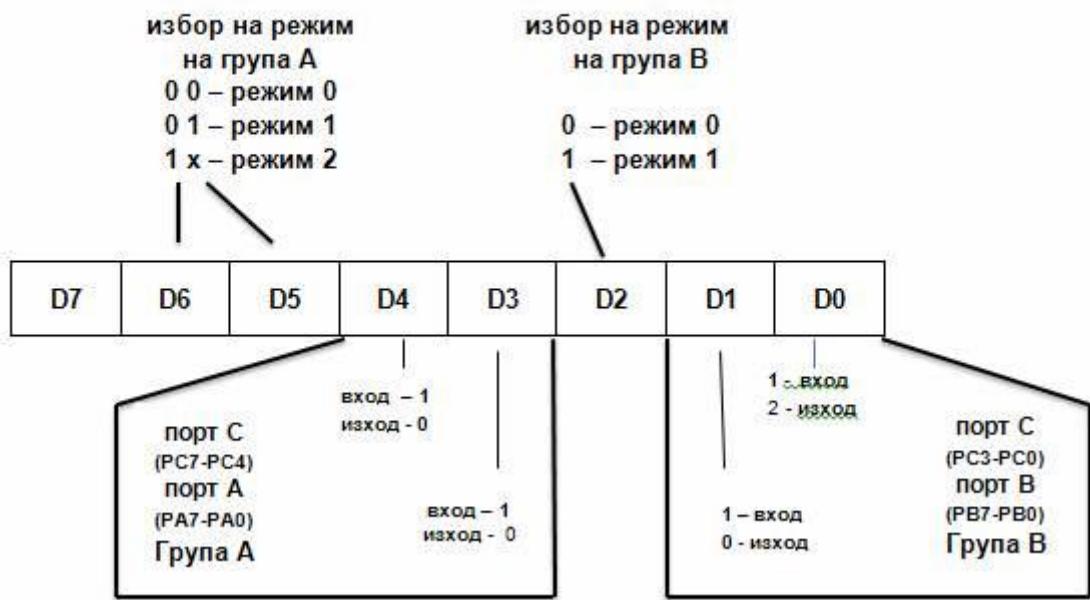
Фиг. 7.1. Двупосочен паралелен интерфейс



Фиг. 7.2. Програмируем интерфейс 8255А

Таблица 7.1.

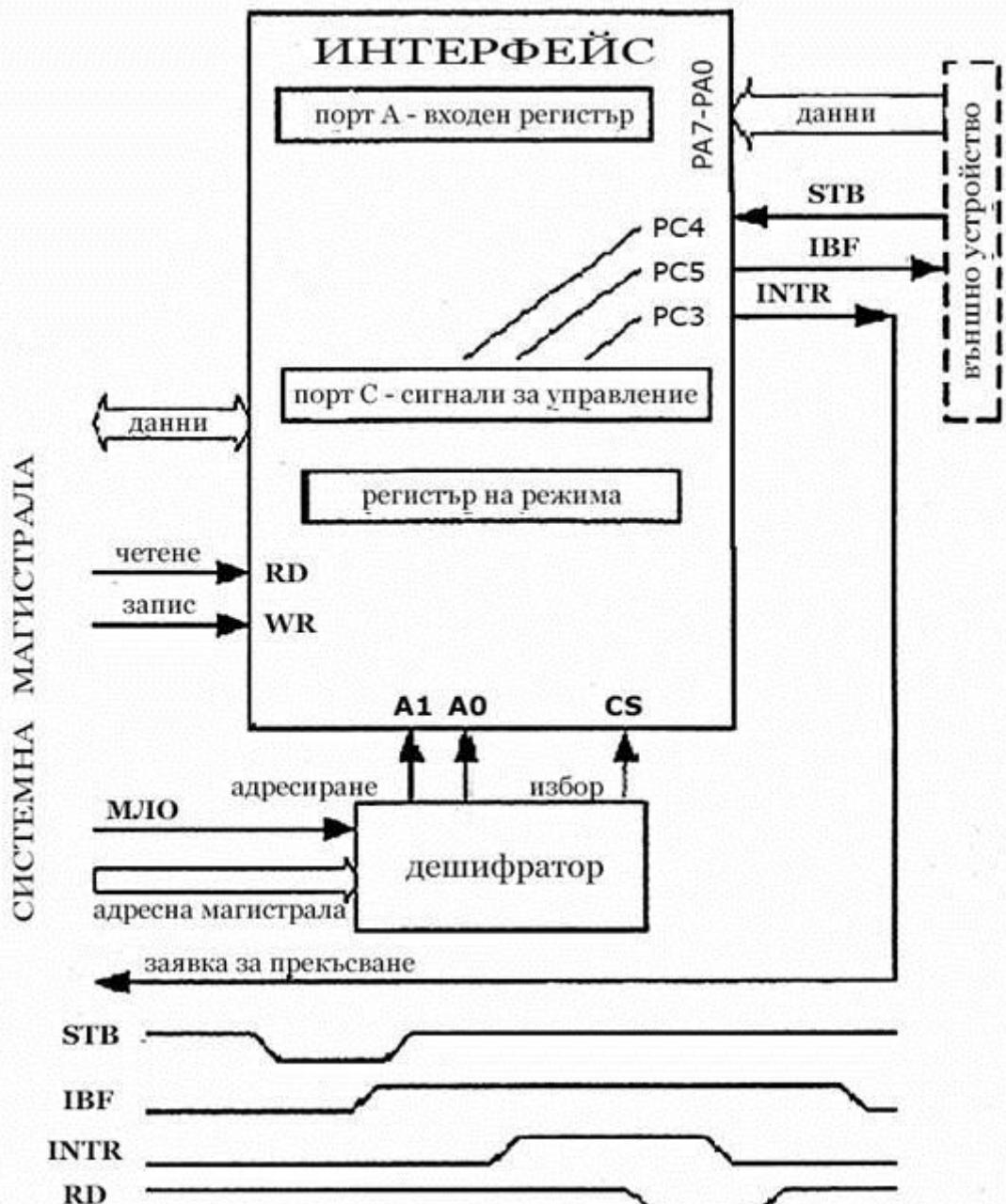
CS	RD	WR	A1	A0	Описание
0	0	1	0	0	Порт А → данни D7-D0
0	0	1	0	1	Порт В → данни D7-D0
0	0	1	1	0	Порт С → данни D7-D0
0	1	0	0	0	данни D7-D0 → Порт А
0	1	0	0	1	данни D7-D0 → Порт В
0	1	0	1	0	данни D7-D0 → Порт С
0	1	0	1	1	данни D7-D0 → рег.управление ( при D7=1)
					Пробитово управление на Порт С ( при D7=0)
1	x	x	x	x	D7-D0 изключени (висок импеданс)
0	0	1	1	1	Недопустима комбинация
0	1	1	x	x	D7-D0 изключени (висок импеданс)



Фиг. 7.3. Предназначение на битовете в регистъра за режим

Таблица 7.3.

		Порт А				Порт В			
C7	C6	C5	C4	C3	C2	C1	C0		
		IBF	STB	INTR	IBF	STB	1NTR		



запис на данните  
в порт А

чтение на данните  
от порт А е 8086

Фиг. 7.4. Режим 1, работа на порт А при вход на данни

Таблица 7.4

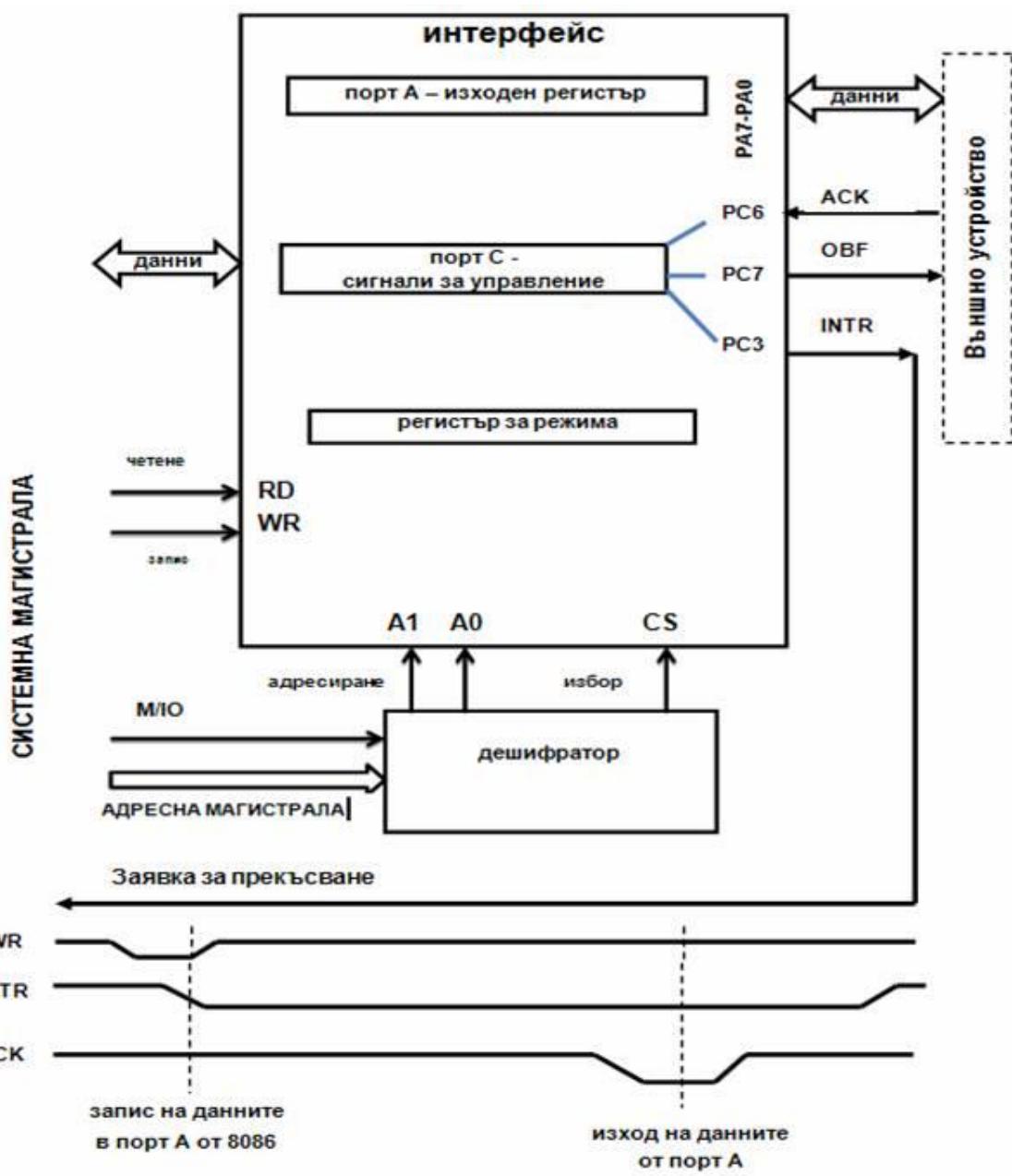
Порт А					Порт В			
C7	C6	C5	C4	C3	C2	C1	C0	
OBF	ACK	-	-	INTR	ACK	OBF	INTR	

**OBF** - използва се за запис на данни от порта във външното устройство;

**ACK** - показва готовността на външното устройство да приеме данни.  
Запис може да се извърши при **ACK = 1**.

**INTR** - използва се като заявка за прекъсване.

Останалите битове **C5** и **C4** от регистър **C** могат да се ползват като отделни линии за вход и изход.



Фиг. 7.5. Режим 1, работа на порт А при изход на данни

**Таблица 7.5.**

Порт А					Порт В		
C7	C6	C5	C4	C3	C2	C1	C0
OBF	ACK	IBF	STB	INTR			

Предназначението на сигналите е следното.

**При вход на данни се ползват:**

STB - сигнал за запис на данни от външното устройство в порт А. Записът се извършва при STB = 0;

IBF - показва готовността на порта да приема данни.

**При изход на данни се ползват:**

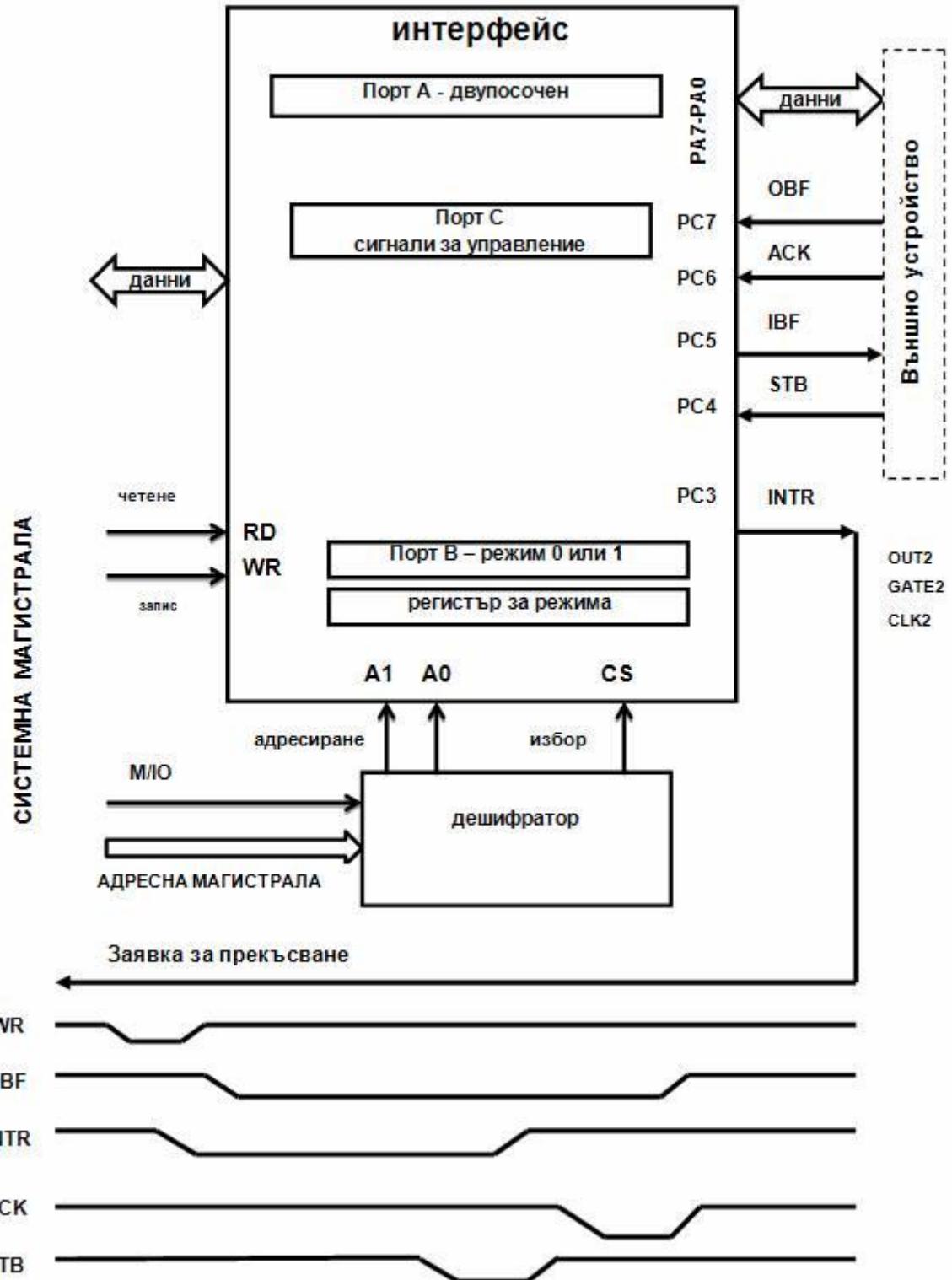
OBF - за запис на данни от порта във външното устройство;

ACK - показва готовността на външното устройство да приеме данни.

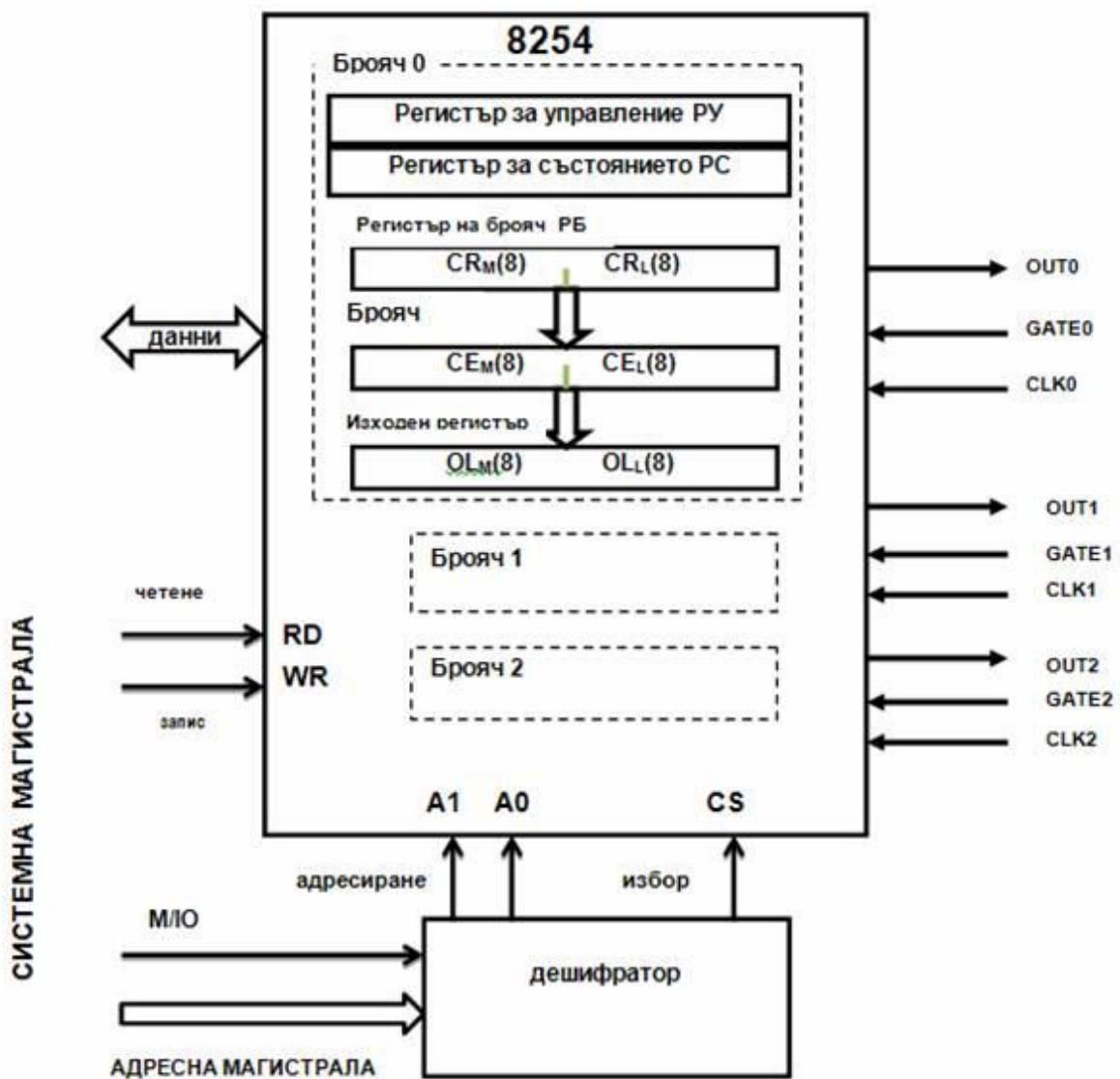
**В двата случая (при вход и изход) се ползват:**

Изводите за данни D7-D0:

INTR - заявка за прекъсване.



Фиг. 7.6. Порт А - режим 2

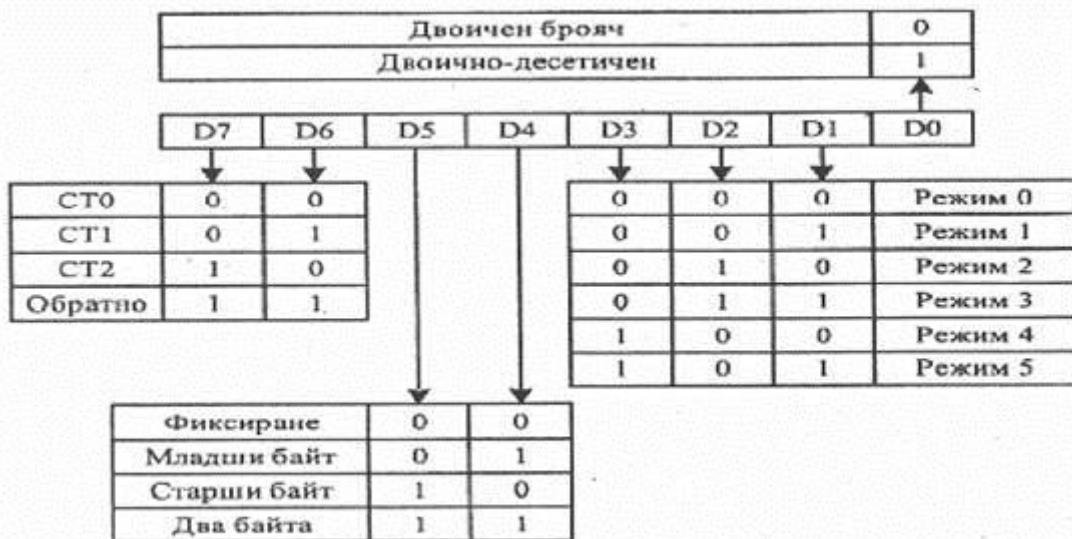


Фиг. 7.7. Програмираме таймер 8254

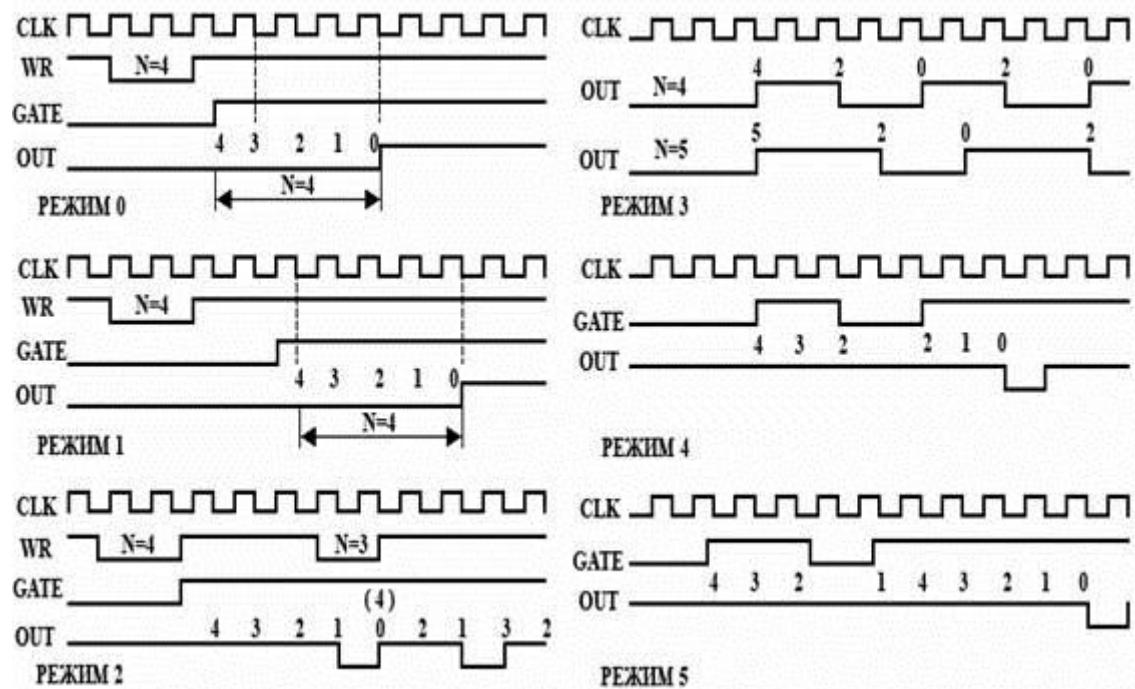
Взаимодействието на таймера със системната магистрала се осъществява чрез двупосочните изводи за данни D7-D0, входовете CS, RD, WR за сигналите за управление и входовете A1, A0 за адресиране. За да се осъществи обмен с микропроцесора, сигнал CS трябва да е равен на лог. 0, посоката на обмен (вход или изход) се определя от сигналите RD и WR, а състоянието на входовете A1, A0 определя частта от таймера, с която ще извърши обмена на данни.

Табл. 7.6.

CS	RD	WR	A <sub>1</sub>	A <sub>0</sub>	Описание
0	0	1	0	0	СТ0 или РС0 → данни D7-D0
0	0	1	0	1	СТ1 или РС1 → данни D7-D0
0	0	1	1	0	СТ2 или РС2 → данни D7-D0
0	1	0	0	0	Данни D7-DO → СТ0
0	1	0	0	1	Данни D7-DO → С1
0	1	0	1	0	Данни D7-DO → СТ2
0	1	0	1	1	Данни D7-DO → рег. за управление РУ



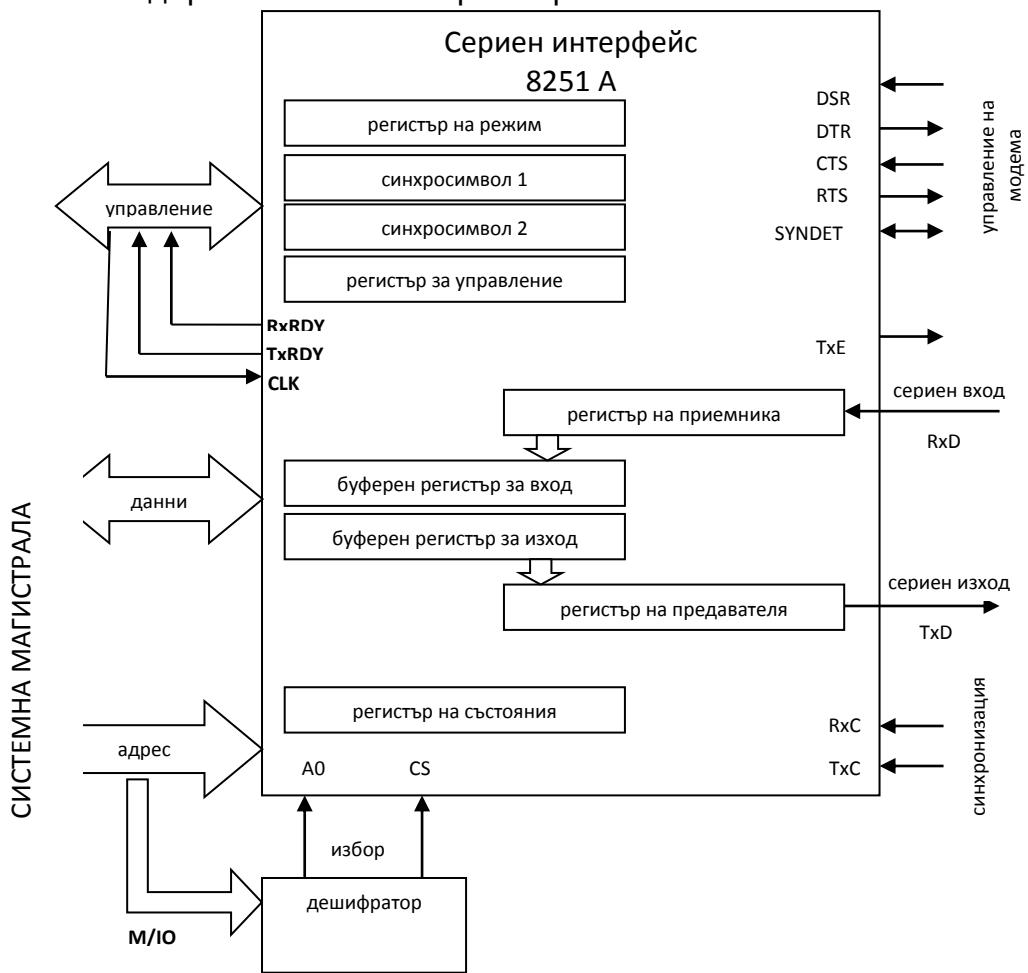
Фиг. 7.8. Предназначение на битовете в регистъра за управление



Фиг. 7.9. Времедиаграма на режимите на работа на таймер 8254

## Асинхронен комуникационен интерфейс

Блоковата схема на програмируемия интерфейс 8251A е представена на фиг. 81. Интерфейсът съдържа следните регистри – за режим, за управление, за състояния и два регистра за два синхросимвола. Функциите на схемата зависят от съдържанието на тези регистри.



Фиг. 8.1. Блокова схема на програмируем интерфейс 8251A

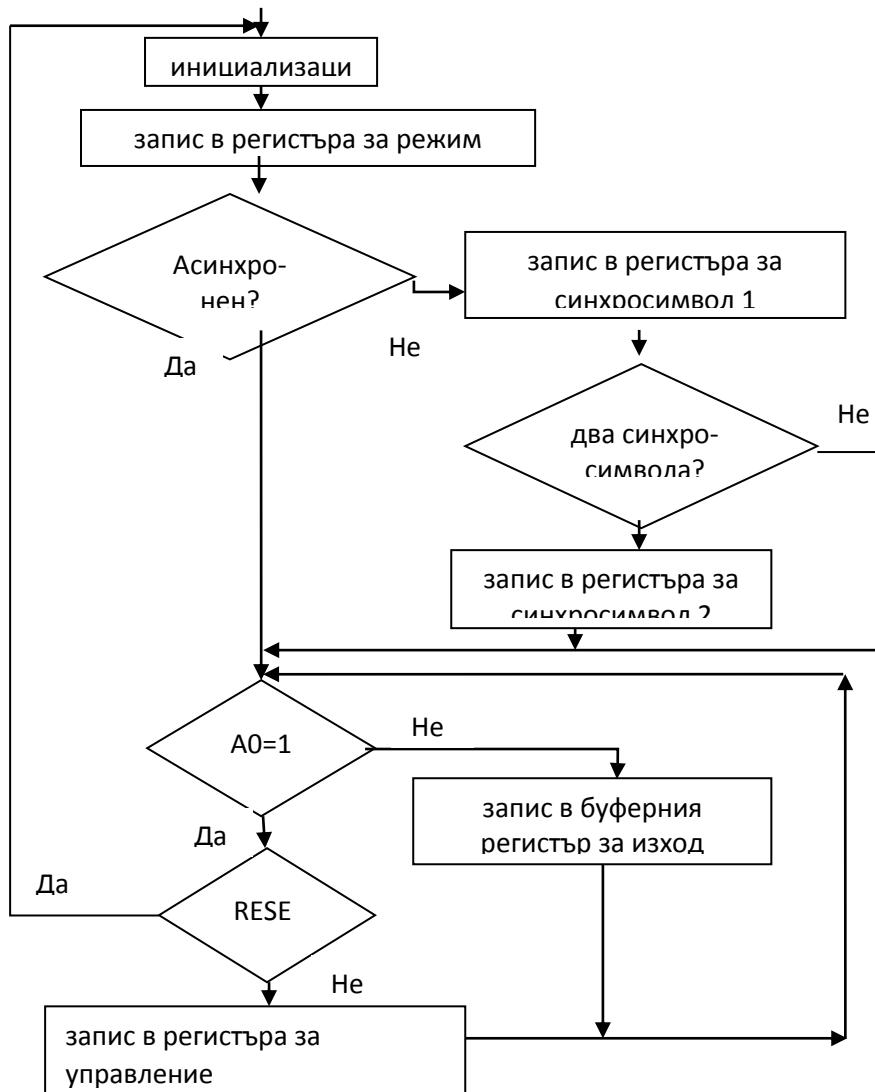
За осъществяване на предаването се използва двойката входен и изходен регистри. Към входния и към изходния регистри са свързани преместващи регистри, чрез които се приемат (и съответно предават) данните към линията.

Взаимодействието на интерфейса със системната магистрала се осъществява чрез двупосочните изводи за данни D7-D0, входовете за сигналите за управление CS, RD, WR и входа за адресиране AO. За да се осъществи обмен с микропроцесора, сигнал CS трябва да е равен на лог.0, посоката на обмен (вход или изход) се определя от сигналите RD и WR, а състоянието на входа AO определя частта от 8251 A, с която ще се извърши обмена на данни. Адресирането на схемата е онагледено в табл. 4.4.1.

Таблица 81.

CS	RD	WR	A <sub>0</sub>	Описание
0	0	1	0	Входен регистър → данни D7-D0
0	1	0	0	Данни D7-D0 → входен регистър
0	0	1	1	Регистър за състоянието → данни D7-D0
0	1	0	1	Данни D7-D0 → рег. за режима, за упр-е и за

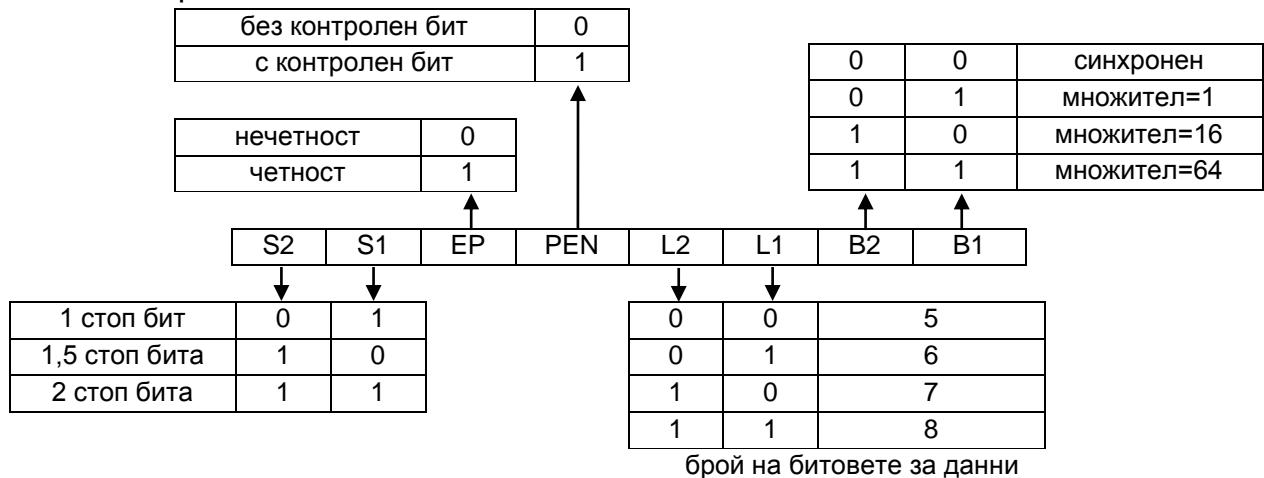
При всички останали комбинации на сигналите на входовете, изводите за данни D7-D0 преминават във висок импеданс.



Фиг. 8.2. Алгоритъм на записа в регистрите на 8251A

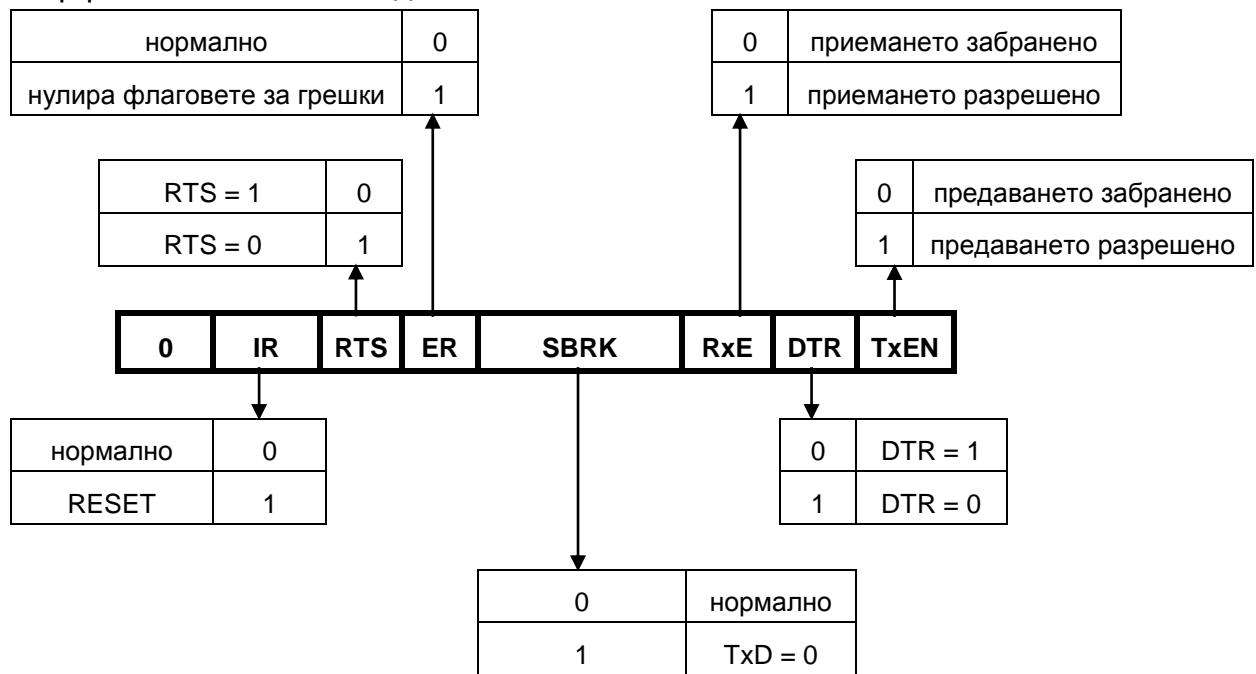
След инициализиране на схемата (например след включване на микропроцесорната система), трябва да се извърши последователно записване на указания в регистрите на схемата за да се дефинират функциите ѝ в конкретната микропроцесорна система. Първият запис е в регистъра за режим. Чрез

двета най-младши бита се задава режимът на работа. Със стойност 00 се задава синхронен режим на работа. Със стойности 01, 10 или 11 се задава асинхронен режим. Предназначението на останалите битове в асинхронен режим е съгласно фиг. 4.4.3.



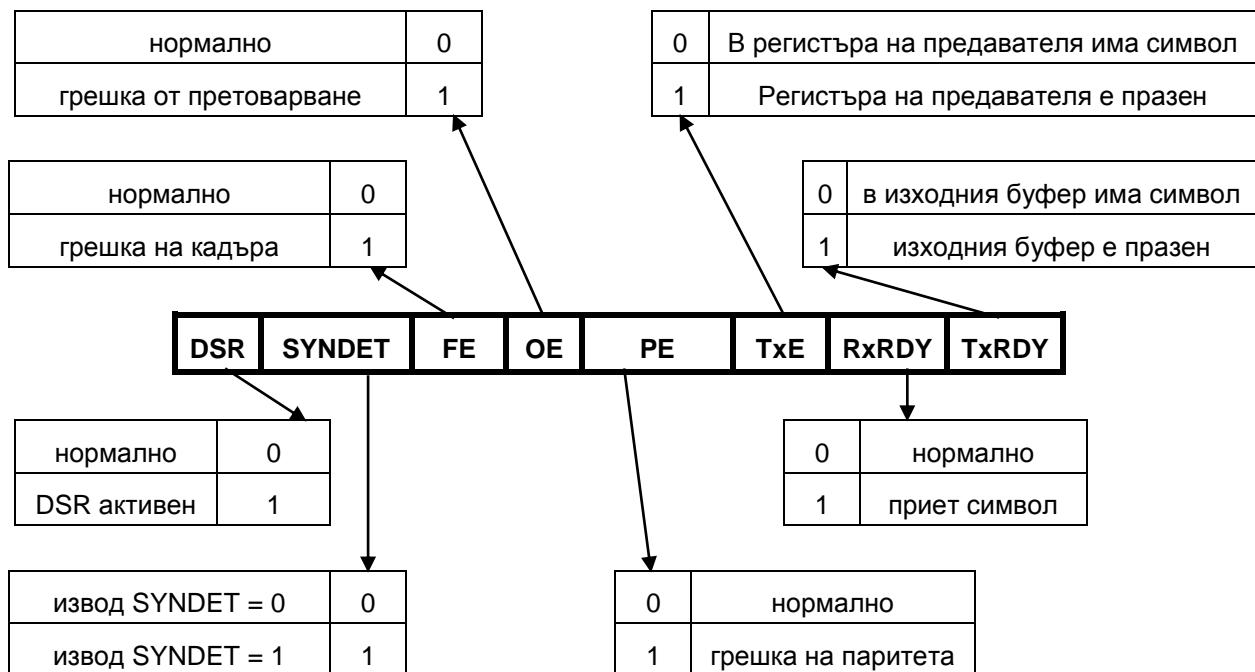
Фиг. 8.3. Предназначение на битовете на регистъра за режим при асинхронна работа

Следващият запис при асинхронен режим е в регистъра за управление. Предназначението на отделните битове е представено на фиг. 4.4.4. Установяването на бит в състояние лог.1 води до активиране на функцията. Част от тези битове (RTS и DTR) са предназначени за управление на връзката на интерфейсната схема с модем.



Фиг. 8.4. Предназначение на битовете на регистъра за управление

Съдържанието на регистъра за състояния може да бъде прочетено от микропроцесора във всеки момент. Предназначението на отделните битове е представено на фиг. 4.4.5.



Фиг. 8.5. Предназначение на битовете на регистъра за състояния

Бит RxRDY = 1 показва, че във входния регистър има приет символ. Този бит може да се използва, за да се организира програмен вход-изход. Стойността на този бит се извежда на извод RxRDY на схемата и може да се използва за организиране на вход-изход по прекъсване. Битът (и съответно изводът RxRDY на схемата) се установява автоматично в лог.0 с операцията по прочитане на данните от входящия регистър.

По аналогичен начин е организирано и предаването (изхода на данни) от интерфейса. За целта се използва бит TxRDY, стойността на който се установява в лог.1 в момента, в който изходният преместващ регистър предаде последния стопов бит на предавания символ. С този бит се индицира, че изходният регистър е готов да приема следващите данни за предаване. Битът може да се използва, за да се организира програмен вход-изход. Стойността на този бит се извежда на извод TxRDY на схемата и може да се използва за организиране на вход-изход по прекъсване. Битът (и съответно извод TxRDY на схемата) се установява автоматично в лог.0 с операцията за запис на данни в изходящия регистър.

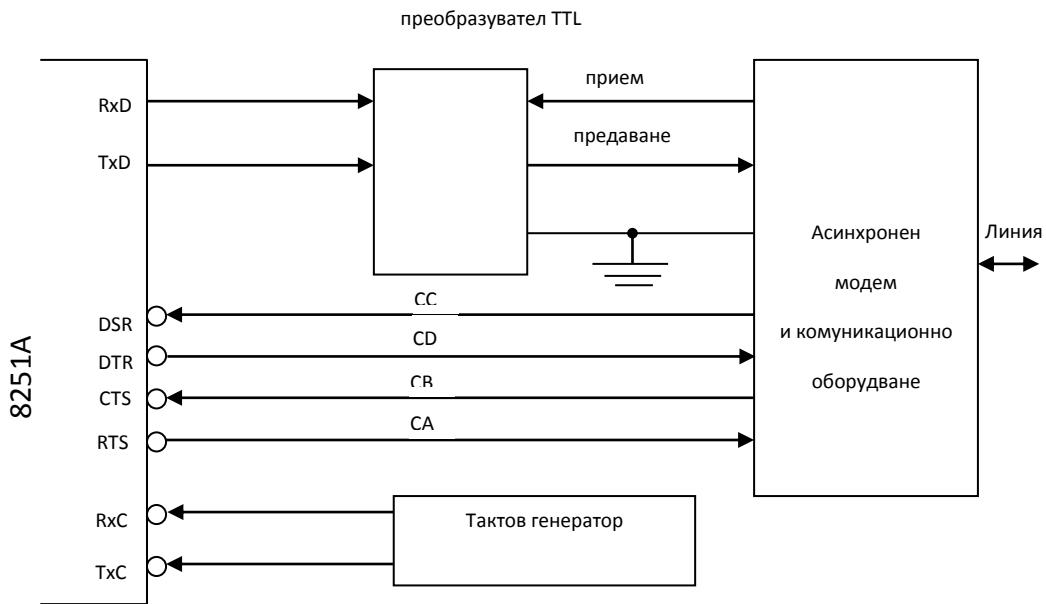
Обменът на данни с микропроцесора винаги е 8-битов. Ако при обмена на данни по линията се използват символи с по-малко от 8 бита – 5, 6 или 7, тогава старшите битове, които се четат от микропроцесора, са равни на лог.0. Контролният бит, ако има такъв, не се предава към микропроцесора, а служи за установяване на бит PE от регистъра за състояние. При наличие на грешка той се установява в лог.1. Стойността на този бит може да се провери по програмен път и ако има грешка при приемане на символа, да се изпълни подпрограма. Обикновено тази подпрограма изтрива грешно приемания символ и отправя към предаващата страна искане за повторно изпращане на символа.

При предаване на символ битът за контрол, ако трябва да има такъв, се

генерира от интерфейсната схема в зависимост от стойността на битовете, записани в изходния регистър.

Сигналът по линията през цялото време между предаваните символи трябва да бъде равен на лог.1. При прекъсване на линията (или изключване на предавателя) ще се приема с постоянен сигнал с лог.0 и извод SYNDET ще се установи в лог.1. С това се индицира, че линията е прекъсната.

За осъществяване на връзка на голямо разстояние интерфейсната схема се свързва към асинхронен modem, така както е показано на фиг. 4.4.6.

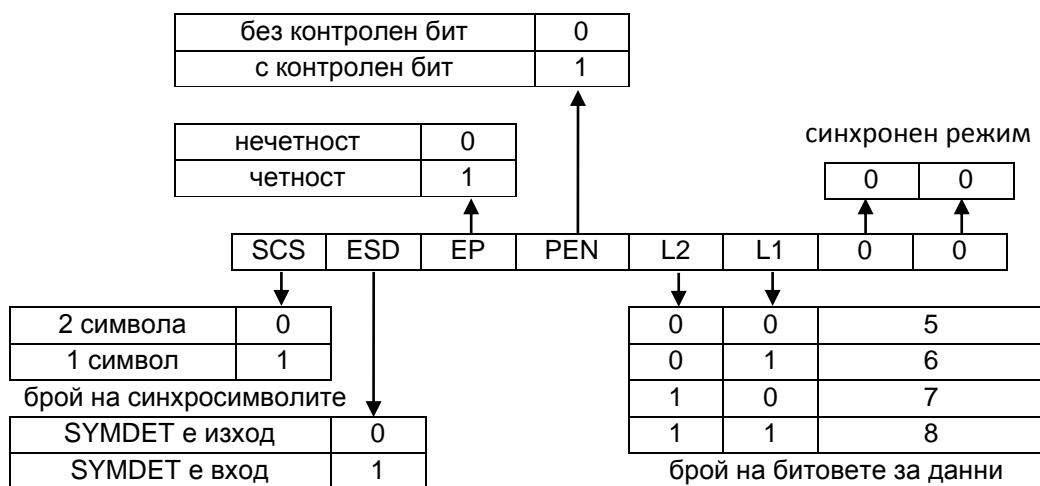


Фиг. 8.6. Включване на 8251A към асинхронен modem

## Синхронен комуникационен интерфейс

За осъществяване на синхронно последователно предаване в микропроцесорна фамилия 8086 се използва универсалния интерфейс 8251 А, програмиран за работа в синхронен режим. Блоковата схема (фиг. 8.1), регистрите и адресирането са същите както е описано в предходния елемент.

За да се зададе режим на синхронна работа, в двата най-младши бита на регистъра за режим се записва стойност 00. Предназначението на останалите битове при работа в синхронен режим е представено на фиг. 8.7.



*Фиг. 8.7. Предназначение на битовете на регистъра за режим при синхронна работа*

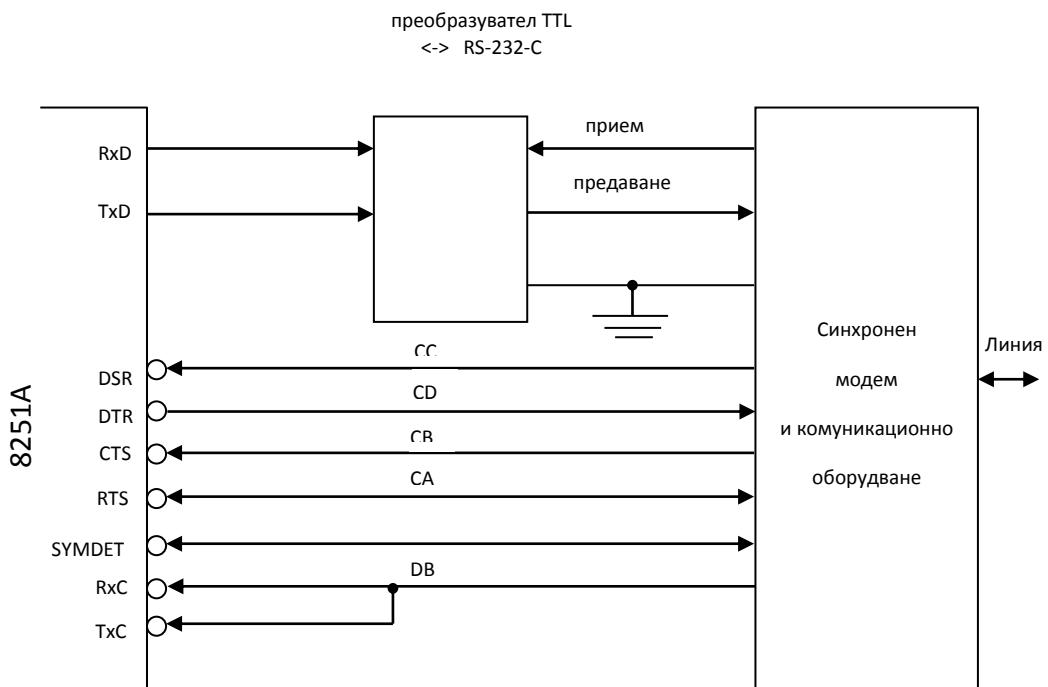
Чрез бит SCS се задава броя на синхросимволите, които се използват. Ако бит SCS е равен на 0, предаването на данни започва, като в линията се извежда първия синхросимвол, веднага след него втория синхросимвол и след това данните. На приемната страна интерфейсната схема проверява приемания от линията цифров поток и ако открие двата синхросимвола, следващи един след друг, започва записване на данните от линията във входния регистър. Другият бит, който е специфичен за синхронния режим на работа е ESD. Чрез този бит се задава типа на извод SYNDET. Изводът може да бъде програмиран като вход или като изход. Когато е програмиран като изход, той става равен на лог.1, когато приемникът регистрира наличие на синхросимволи в приеманите данни. Възможно е търсенето на синхронизация да се осъществява от външно устройство. В този случай извод SYNDET се програмира като вход и при откриване на синхросимволи, външното устройство подава към него сигнал (лог.1).

След записа в регистъра за режим, съгласно алгоритъма от фиг. 8.2, се записва първият синхросимвол. Ако най-старшият бит SCS на регистъра за режима е равен на 0, се записва и втори синхросимвол. След това се записва байт, с който се задават стойностите на битовете на регистъра за управление

(фиг. 8.4).

За синхронизация на предаването и приемането се използва външен генератор. Сигналът на този генератор се подава на управляващите входове на преместващите регистри на предаващата (вход T x C) и на приемащата (вход R x C) страни. Под действието на всеки такт на сигнала за синхронизация един бит от преместващия регистър на предаващата страна се извежда към линията, а на приемната страна преместващият регистър се премества с една стъпка за да приеме един бит. Следователно скоростта на предаване в линията ще бъде равна на честотата на синхронизиращия генератор.

За осъществяване на връзка на голямо разстояние интерфейсната схема се свързва към синхронен modem, така както е показано на фиг. 8.8



Фиг. 8.8. Включване на 8251A към синхронен modem