

Митьо Митев

**МИКРОПРОЦЕСОРНА
СХЕМОТЕХНИКА**

Записки на лекции

**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ
2021**

© Митьо Георгиев Митев
София, 2021 г.

УВОД

Настоящите записки по ”Микропроцесорна схемотехника” са по лекциите, четени пред студентите по специалността „Електронна техника” в Техническият университет – София. Целта на дисциплината е да даде първоначални познания на студентите от бакалавърската степен на обучение за структурата, организацията, областите на приложение и начините за изграждане на микропроцесорни системи. Акцентирано е на характерни особености на различните 8- и 16- разрядни микропроцесорни системи, както и на някои фамилии микроконтролери. Отделено е внимание на разнообразните интерфейсни системи, както и на схемотехничните решения при тяхното свързване с външни устройства, сензори и изпълнителни устройства и системи.

При изготвянето и изучаването на материала се разчита на задълбочени познания по дисциплините, изучавани в предходните курсове – основно по цифрова схемотехника, импулсна схемотехника и аналогова схемотехника, както и програмиране. Знанията, придобити по „Микропроцесорна схемотехника”, са необходими при овладяване на всички останали дисциплини при обучението по „Електронна техника”, както в бакалавърския, така и в магистърския курс.

Независимо от желанието и старанието, с което бе подготвен настоящият материал, съм сигурен, че той е непълен и в него са останали неясноти, неточности и технически пропуски. Надявам се, че ще имам възможност в бъдеще съществено да го преработя и допълня, отстранявайки грешките и допълвайки допуснатите пропуски. Поради това всякакви забележки и препоръки, целящи тяхното отстраняване, ще бъдат приети с благодарност на адрес София, 1797, Технически университет – София, катедра „Електронна техника”.

Януари, 2021 г.

Авторът

ГЛАВА 1.

МИКРОПРОЦЕСОРИ И МИКРОКОНТРОЛЕРИ

1.1. ВЪВЕДЕНИЕ. МИКРОПРОЦЕСОРНИ СИСТЕМИ – СЪЩНОСТ И ОРГАНИЗАЦИЯ

Повсеместното навлизане на микропроцесорните системи в различните сфери на живота едва ли дава възможност за обсъждане на тяхната роля и място, което заемат днес. От друга страна е впечатляващо несъответствието между разбирането и възприятието на масовия потребител на тези системи и тяхната действителна сложност и същност на всички нива – технологично, схемотехнично, системотехнично, алгоритмично, програмно и пр.

Микропроцесорните системи по същество са електронни цифрови устройства, в основата на които е заложен микропроцесор (микроконтролер). Микропроцесорите са цифрови интегрални схеми, които са *програмируеми* – тяхното функциониране се определя от заложената в системата програма, което предопределя много широките и разнообразни области на тяхното приложение.

Първият микропроцесор се появява на пазара през 1971 г. – това е 4-разредният микропроцесор i4004 на фирмата INTEL. През 1972 година се появява първият 8-разряден микропроцесор – i8008. През 1974 г. INTEL представят и първия високопроизводителен 8-разряден микропроцесор i8080, а няколко месеца след това и Motorola пуска на пазара своя първи 8-разряден микропроцесор MC6800. От този момент започва бурното развитие и масово приложение на микропроцесорите в микрокомпютрите, както и микроконтролерите като управляващи системи във всевъзможни промишлени изделия, битови уреди, автомобилостроенето, комуникациите и пр. В табл. 1 са представени някои от най-популярните микропроцесори по реда на тяхното създаване, както и някои основни техни параметри.

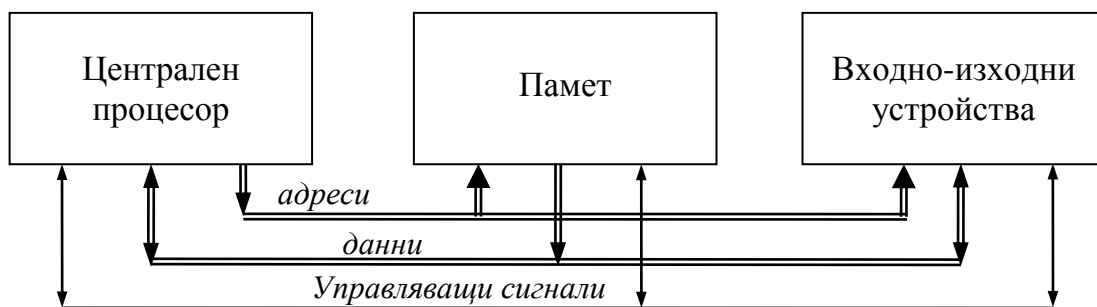
Таблица 1

тип	година	производител	разрядност	Адр/данни	честота	FPU	MMU	MIPS
i4004	1971	INTEL	4		0,108		не	0,06
i8008	1972	INTEL	8	12/8	0,2		не	
i8080	1974	INTEL	8	16/8	2		не	0,64
MC6800	1974	MOTOROLA	8	16/8	1		не	
I8085	1977	INTEL	8	16/8	5		не	
MC6809	1978	MOTOROLA	8	16/8	2		MC6829	
I8086	1978	INTEL	16	20/16	5	I8087	не	
I8088	1979	INTEL	16	20/8	5	I8087	не	0,33
MC68000	1979	MOTOROLA	16	24/16	8		MC68451	
I80286	1982	INTEL	16	24/16	6	I80287	вграден	1
MC68020		MOTOROLA	32	32/32	16	MC68881	MC68851	
I80386	1985	INTEL	32	32/32	16	I80387	вграден	5
MC68030		MOTOROLA	32	32/32	33	MC68882	вграден	
I80486	1989	INTEL	32	32/32	25	вграден	вграден	20
MC68040		MOTOROLA	32	32/32	50	вграден	вграден	
Pentium	1994	INTEL	32/64	32/64	75-233	вграден	вграден	100-600
MC68060		MOTOROLA	32			вграден	вграден	

Огромното разнообразие от технически средства, в които са заложени управляващи и/или измерващи микропроцесорни системи, обуславя и невъобразимото

разнообразие на структурата и начините за изграждането им. От друга страна съществуват основни принципи и правила, които дават възможност да се дефинират няколко обобщени структури, към които да бъде отнесена една или друга микропроцесорна система.

Във всяка микропроцесорна система могат да се обособят три основни блока – централен процесор, памет и периферни (входно-изходни) устройства [3], както е показано на фиг. 1.1. Връзката между тях се осъществява посредством системни шини (системна магистрала), включващи адресни линии, линии за данни и управляващи сигнали.



Фиг. 1.1

Централният процесор на микрокомпютъра е прието да се нарича микропроцесор и съдържа аритметично-логическо устройство (АЛУ), регистри, управляващо устройство (команден интерпретатор) и магистрални буфери. Той изпълнява всички операции по управление на обмена на данни, изпълнение на инструкциите на програмите, аритметичните и логически операции върху данните.

Паметта съхранява инструкциите на програмите и данните. Обикновено съдържа в себе си енергонезависима памет (ROM) и памет с произволен достъп (RAM). Първата има свойството да запазва съдържанието си след изключване на захранването и не позволява съдържанието ѝ да се променя по време на работа. По тази причина в нея се съхраняват машинните кодове на инструкциите и данни, които не се променят (константи). Втората памет позволява провеждане на операции по четене и запис и се използва за съхранение на текущи данни, променливи и пр.

Входно-изходните устройства служат за въвеждане и извеждане на данни в микропроцесорната система. Това могат да бъдат различни серийни или паралелни интерфейсни схеми, аналогово-цифрови и цифрово-аналогови преобразуватели, броячи на импулси и генератори на времеви интервали (наричани най-често таймерни схеми) и др.

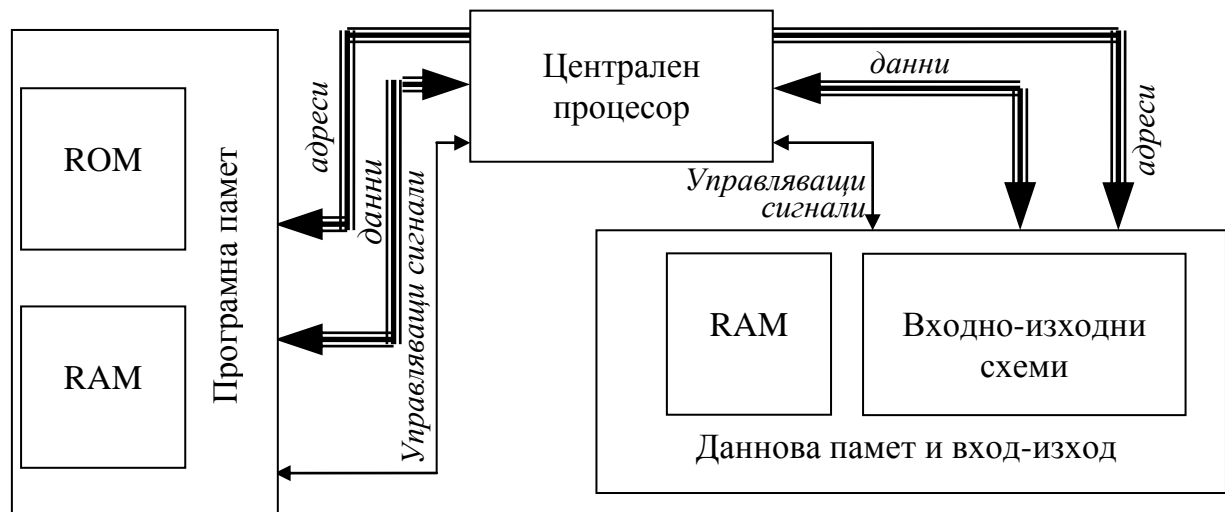
1.2. АРХИТЕКТУРА И СТРУКТУРА НА МИКРОПРОЦЕСОРНИ СИСТЕМИ

1.2.1. Архитектура на микропроцесорните системи

Независимо от огромното разнообразие на микропроцесорните системи, по своята архитектура те могат да бъдат разделени на две големи групи според начина на обособяване на адресното пространство. В случая, когато и машинните кодове на инструкциите, и данните са разположени в общо адресно пространство, се

казва, че имаме класическа, или Фон Нойманова, архитектура. Този случай е илюстриран на фиг. 1.1. Характерно за нея е, че има само една системна магистрала. Това предполага по-лесно изграждане на апаратната част. От друга страна достъпът до системните ресурси се забавя, тъй като програмната и данновата памет, както и входно-изходните схеми, са разположени в едно и също адресно пространство. Достъпът до тях става последователно във времето, което естествено оказва влияние на общата производителност на системата.

Втората възможна структура, показана на фиг.1.2, е добила популярност като Хардвардска архитектура. При нея са обособени две разделни адресни полета – едното за машинните кодове на инструкциите, а другото – за данни. По този начин много добре се разграничава програмната памет от паметта за данни. Нещо



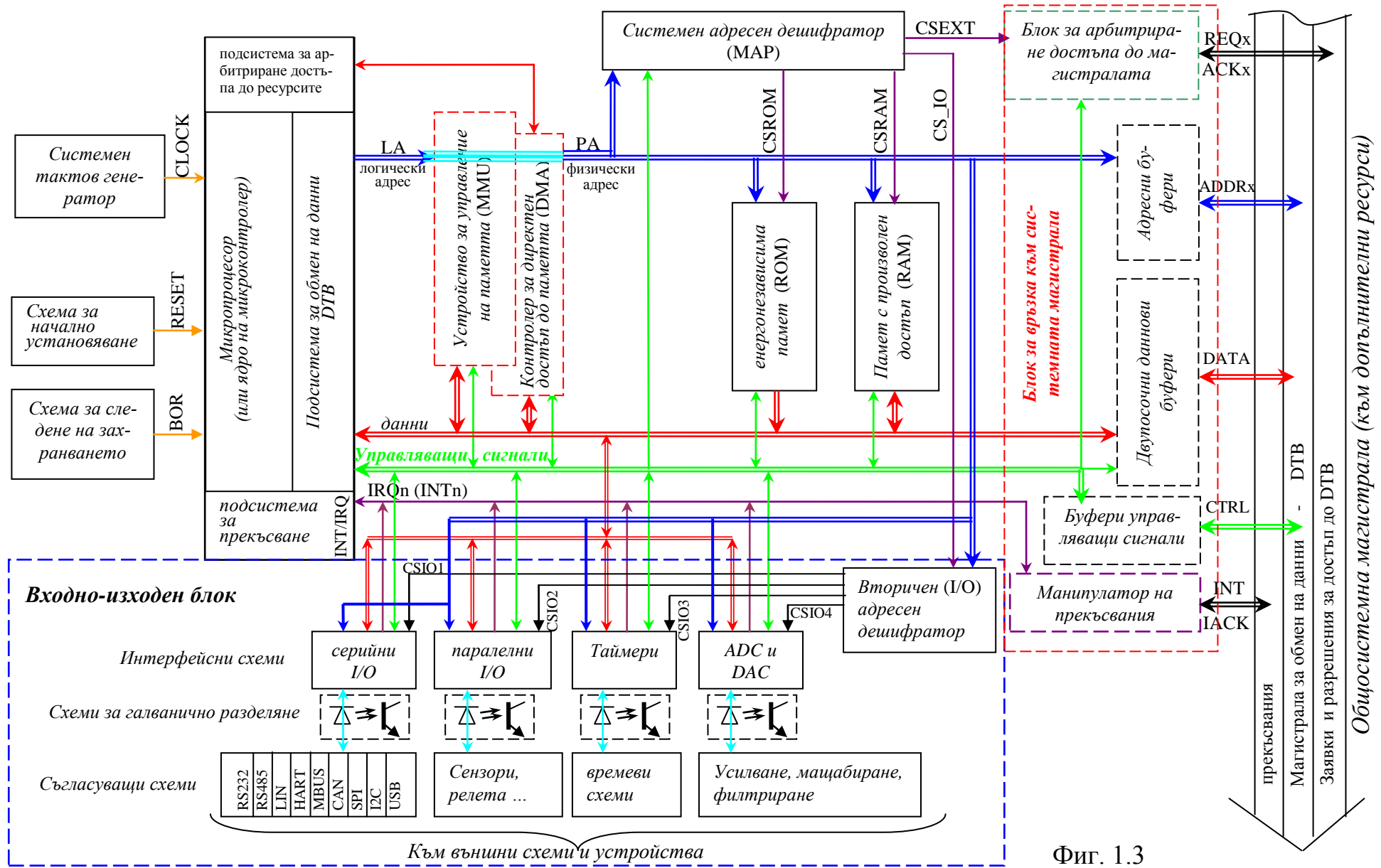
Фиг. 1.2

повече – тази организация дава възможност в един и същи момент от време да се извършва обмен с ресурсите и в двете полета. Не съществува ограничение, което да налага разредността на двете адресни магистрала, респ. на двете даннови магистрала, да бъдат еднакви. Така, за сметка на по-сложната апаратна реализация, може съществено да се повиши производителността и бързодействието на системата.

1.2.2. Структурна схема на микропроцесорна система

Вече беше подчертано огромното разнообразие, което съществува при изграждането на микропроцесорните системи в зависимост от тяхното предназначение. Независимо от това, може да се обобщи една по-подробна функционална схема, показана на фиг.1.3, която в голяма степен отразява повечето съществуващи решения. Разбира се, много често някои функционални модули могат да липсват. Тези, които по-рядко се налага да бъдат включвани в системите, на фигурата са отбелязани с пунктирани линии.

Ядро на всяка микропроцесорна система естествено се явява микропроцесорът. За неговото нормално функциониране е необходимо да присъстват системен тактов генератор, схема за начално установяване и схема за следене на захранващото напрежение. Те много често са вградени в самите микропроцесори (и като правило във всички микроконтролери).



Фиг. 1.3

Задължително във всяка микропроцесорна система присъстват постоянна (енергонезависима) и оперативна (с произволен достъп) памети (ROM - Read Only Memory и RAM – Random Access Memory). Първата от тях съдържа стартовата програма (мониторна програма или системна входно-изходна програма – BIOS), а във втората могат да се съхраняват както данни, така и системни или потребителски програми.

Системата не би могла да функционира и без системен адресен дешифратор (MAP декодер). Неговото предназначение е да декодира логическата комбинация на сигналите, поместени на старшите адресни линии и в съответствие с приетото разпределение на системните ресурси в адресното поле да изработи сигнали за избор на съответните схеми. Много често той се разделя на няколко части. Първичният декодер дефинира областите за ROM, RAM, входно-изходни схеми и адресно поле за разширение на системата. За всяко от тези полета може да се изгради вторичен дешифратор, който изработва сигнали за избор на съответните схеми, изграждащи дадения функционален блок.

Разбира се, от която и да било система не би имало особен смисъл, ако не съществува възможност за въвеждане и извеждане на информация. Затова, под една или друга форма, съществува входно-изходен блок. Той съдържа различни набори от интерфейсни схеми, например:

- серийни входно-изходни схеми (адаптери), които преобразуват данните от паралелен в сериен формат при съблюдаване изискванията за форматиране на някой от широко разпространените формати;
- паралелни интерфейсни адаптери, позволяващи конфигуриране на необходимите за системата цифрови входове, изходи или двупосочни портове;
- таймерни блокове, изградени от броячи или делители на честота, позволяващи генериране или измерване на времеинтервали;
- аналогово-цифрови или цифрово-аналогови преобразуватели, позволяващи въвеждане и извеждане на аналогови данни в микропроцесорните системи;
- други разнообразни входно-изходни схеми в зависимост от конкретната необходимост и предназначение на системата.

След всяка интерфейсна схема е необходимо да се постави съгласуваща схема, която да преобразува сигналите по ниво до изискванията, поставени от външните схеми и устройства, които се свързват към съответната система. Тук имат място както стандартизирани интерфейси, чиито сигнали на физическо ниво са строго регламентирани, така и специфични източници на сигнали и изпълнителни устройства, чиито електрически спецификации (по напрежение, мощност и времеви изисквания) трябва да бъдат удовлетворени. Така например:

- след серийните адаптери могат да бъдат поставени буферни схеми, удовлетворяващи системните спецификации на интерфейсите за връзка – напр. RS232C, RS422/RS485, LIN, HART, Mbus и др. При използване на синхронни серийни интерфейси SPI или I2C може да се наложи да се осъществи връзка между системи с различни логически нива на сигналите – в тези случаи е необходимо да се използват буфери-транслатори на ниво;

- след паралелни интерфейсни схеми могат да бъдат включени сензори, бутони, клавиатури, светодиоди, индикатори, релета и др. изпълнителни механизми. Всеки от конкретните случаи предявява специфични изисквания и се налага използване на подходяща електронна схема за осъществяване на електрическата връзка при гарантиране на необходимата надеждност и електромагнитна съвместимост;
- при измерване на времеви интервали е необходимо преди таймерите да има схеми за привързване към сигналите по време, подбор на входните сигнали по определени признаци, съгласуване по фронтове и нива на сигналите при съхраняване на високо бързодействие и шумоустойчивост. При използване на изходните импулси от таймерите обикновено се поставят широколентови буферни схеми, двойни интегратори (за получаване на аналогово напрежение) или мощни драйвери (при управление на електродвигатели чрез широчинно-импулсна модулация);
- аналогово-цифровите и цифрово-аналоговите преобразуватели дават възможност за дискриминация на аналоговите сигнали по ниво и квантуването им във времето. Това предполага входно/изходните сигнали да бъдат усилвани и мащабирани по ниво, преобразувани (в токови или напрежителни сигнали), да бъде ограничавана тяхната честотна лента (чрез пасивни или активни филтри) или да бъде запомняна моментната им стойност (с помощта на схеми за следене/запомняне или с пикови детектори), определяна ефективната им стойност и пр.;
- други интерфейсни схеми и външни устройства биха предявили различни специфични изисквания към сигналите, което ще наложи разработване на конкретни съгласуващи схеми, които да гарантират устойчива работа на интерфейсната система.

Добре е винаги, когато има възможност, да се осигурява галванично разделяне на микропроцесорната система от външните схеми и устройства. Това гарантира висока шумоустойчивост и дълговременна безотказна работа на системата като цяло. Най-лесно галваничното разделяне се осъществява между интерфейсите и съгласуващите схеми. Понякога (напр. при аналогово-цифрово или цифрово-аналогово преобразуване) може да е по-удачно галваничното разделяне да се осъществи между системата и интерфейсната схема.

1.2.3. Методи за надграждане на системите – системни магистрали

Много често се залага възможност за разширение на микропроцесорната система. Има два конструктивни подхода, които се прилагат при такива системи:

- чрез реализиране ядрото на системата на отделна базова платка (наричана „платка майка” или „дънна платка”) и вграждане в системата на съответните функционални блокове, удовлетворяващи системните спецификации на дадена системна магистрала (например ISA, PCI и др.). Подобен подход намира масово приложение при изграждане на различните модели персонални компютри;
- чрез изграждане на т.нар. „магистрално-модулни системи”. При изграждане на отделните модули (системни платки или интерфейси) задължително се спазват системните спецификации на определената система (напр. VME) на логическо, електрическо и механично ниво. Модулите се поставят в отделен крейт (касета), като връзката между тях се осъществява чрез системна магистрала, разгъната на задната платка, която обикновено е пасивна –

съдържа само куплунги и терминиращи резистори за всички сигнални линии. Съвременните магистрално-модулни системи са многопроцесорни и се използват за управление на сложни обекти и събиране на големи обеми от данни.

Системните спецификации на различните системни магистрали предполагат много различни структури на блока за връзка към тях. Обикновено той включва адресни буфери, двупосочни буфери за данни и буфери за управляващите сигнали, чрез които се обезпечава функцията за обмен на данни с останалите модули.

Почти винаги в блока за връзка със системната магистрала се включва и манипулатор на прекъсванията, чиято основна задача е да гарантира приоритетния ред на възприемане на заявките за прекъсване и да раздели външните от тези, възникващи в локалната система.

В системи, в които е предвидена многопроцесорна работа, е необходимо вграждането на функционален модул, позволяващ генериране на заявка от системната магистрала за използване на подсистемата за обмен на данни .

1.2.4. Механизми за управление на паметта в системата

В съвременните микропроцесорни системи в резултат на изпълнението на програмите се формира логически адрес, на който се намират съответните данни. Действителното разпределение на отделните масиви с данни се определя от операционната система. Прекодирането на логическия адрес до действителен физически адрес се извършва от устройството за управление на паметта (Memory Management Unit – MMU). В някои микропроцесорни фамилии това е външна интегрална схема (като MC68451 за MC68000 и MC68851 за MC68020), а в други случаи е вградена вътре в микропроцесора – напр. MC68030, MC68040, i80286, I80386, i80486, Pentium и др. Подобен механизъм вече се вгражда и в множество съвременни микроконтролери.

Освен функциите по прекодиране на логическите адреси до физически, MMU изпълнява и функции по защита на достъпа до данните в отделни сегменти и страници от паметта от отделни потребителски програми и режими на работа на микропроцесора. Тези механизми са пътят към изграждане на виртуална памет. Тези въпроси излизат извън обема на този курс и няма да бъдат разглеждани подробно тук.

1.2.5. Механизми за директен достъп до паметта

Понякога в системите възниква необходимост от въвеждане (или извеждане) на блокове от данни със значителен обем и висока скорост (напр. четене/запис на дискови устройства, снемане на кадри от видеокамери и др.), при което е невъзможно или нерентабилно управлението на обмена да се извършва от микропроцесора. В такива случаи е по-добре, ако в системата има вграден контролер за директен достъп до паметта (Direct Memory Access – DMA).

При постъпване на заявка от дадено периферно устройство за обмен на данни DMA контролерът изисква от централния микропроцесор управлението на системата и организира обмена без негово участие. При това данните се обменят директно между паметта и периферното устройство, което става само за един такт. Така съществено се повишава бързодействието на системата.

Тенденцията през последните години е в някои 16- и 32- разредни микроконтролери, в които има вградени по-големи масиви RAM, също да бъдат имплементирани контролери за директен достъп до паметта. Обикновено се използват за организиране на обмена по някои високоскоростни интерфейси (напр. USB) или с вградени бързи ADC.

1.3. ВЪТРЕШНА СТРУКТУРА И НАЧИН НА РАБОТА НА МИКРОПРОЦЕСОРИ И МИКРОКОНТРОЛЕРИ

Забележително е повсеместното разпространение, което получиха микропроцесорите и микроконтролерите след появата им през далечната вече 1971 год., навлизайки буквално във всички области на живота. От друга страна е изненадващо колко малко хора могат да дадат смислен (и точен!) отговор на въпросите

„Какво е микропроцесор?“, „Каква е разликата между микропроцесора и микроконтролера?“, „Как работят те?“ ...

1.3.1. Какво е микропроцесор ?

Определението на микропроцесорите като програмируеми цифрови интегрални схеми е точно, но не е пълно от функционална гледна точка.

Представката „микро-“ не омаловажава възможностите и функциите на устройството, а се използва да подчертае, че то е реализирано в една или няколко (не повече от 2-3) интегрални схеми в противовес на централните процесори на големите изчислителни машини, предшественици на днешните микрокомпютри и програмируеми управляващи системи.

„Процесор“ предполага устройство, в което се извършват определени действия над един или няколко входни потоци, които след съответното преобразуване се извеждат като изходен продукт [1]. В случая става дума за информационно устройство, при което [5] процесите касаят обработка на потоци от данни. Действието на процесора се свеждат до:

- Изработване на синхронизиращи сигнали;
- Извличане на инструкции от програмната памет;
- Четене/запис на данни от/към даннова памет или интерфейсни устройства;
- Изпълняване на логически или аритметични операции върху данните;
- Съхраняване на получените резултати в паметта или интерфейсните устройства;
- Анализ на състоянията, получени при изпълнение на операциите, и управление на последователността на командите;
- Възприемане и реагиране на сигнали, отразяващи настъпването на външни за микропроцесора събития и процеси.

Гореописаните действия са присъщи за всички микропроцесори, независимо от тяхната сложност, разредност и пр. Всичко това го позиционира като основно управляващо устройство в дадената микропроцесорна система.

1.3.2. Какво е микроконтролер?

Развитието на технологиите създаде предпоставки в рамките на една интегрална схема освен микропроцесора да бъдат вградени като минимум постоянна и оперативна памет, системен тактов генератор и първичен адресен дешифратор. Вече може да се каже, че цялата микропроцесорна система е интегрирана в рамките на една интегрална схема. В допълнение винаги в системата се имплементират и определени периферни схеми. Тези схеми, които съдържат в себе си:

- Микропроцесорно ядро;
- Системен тактов генератор и схема за начално установяване;
- Системен адресен дешифратор;
- Енергонезависима памет (ROM, EPROM или EEPROM);
- Оперативна памет (RAM);
- Входно-изходни схеми,

е прието да се наричат „микроконтролери” или „вградени микропроцесорни системи”. Понастоящем те намират невъобразимо широко приложение – от електронни системи за управление на чайници, ютии и микровълнови фурни до смартфони и всички управляващи елементи в съвременните автомобили и т.н.

От казаното следва, че всички принципи и особености, които се отнасят за микропроцесорите, са валидни в една или друга степен и спрямо микропроцесорните ядра, вградени в отделните микроконтролери.

1.3.3. Микропроцесорни фамилии и фамилии микроконтролери

Още с появата на първите микропроцесори става ясно, че наличието на даден микропроцесор не е достатъчно, за да се изгради на негова база микропроцесорна система. Осъзнавайки този факт, производителите на микропроцесори разработват и разнотипни периферни схеми, съвместими с техните микропроцесори. Съвкупността от даден микропроцесор и съвместими с него периферни схеми е прието да се нарича ***микропроцесорна фамилия*** – т.е. това понятие касае съвкупност от разнотипни интегрални схеми, чиито интерфейси (подсистеми за данни, за прекъсване и арбитражиране) са съвместими и гарантират по-лесно изграждане на завършени микропроцесорни системи. Една микропроцесорна фамилия освен микропроцесор включва серийни и паралелни интерфейсни адаптери и таймери. Понякога фамилията се допълва от контролер за директен достъп до паметта, приоритетен контролер на прекъсванията, устройство за управление на паметта ...

С появата на първите микроконтролери понятието ***фамилия микроконтролери*** се запазва, но вече неговата същност е малко променена. Това са микроконтролери, които са изградени на базата на идентични микропроцесорни ядра, но вградените ресурси могат много да се различават – обем на енергонезависимата и оперативна памет, както и вградените и различни периферни схеми и таймерни модули. Като правило в съвременните микроконтролери вече се вграждат и многоканални аналогово-цифрови и цифрово-аналогови преобразуватели, компаратори, операционни усилватели и др. аналогови схеми. Това дава възможност на разработчиците да изберат тази схема от фамилията (но само една!), която в максимална степен удовлетворява техните изисквания. Така се намалява силно себестойността на изделията чрез използване на по-евтини микроконтролери от съответното семейство, реализиране на системите на платки с по-малка площ, монтиране на по-малко компоненти, по-лесна настройка и по-ниска консумация. Оптимизирането на системите води и до съществено повишаване и на тяхната надеждност.

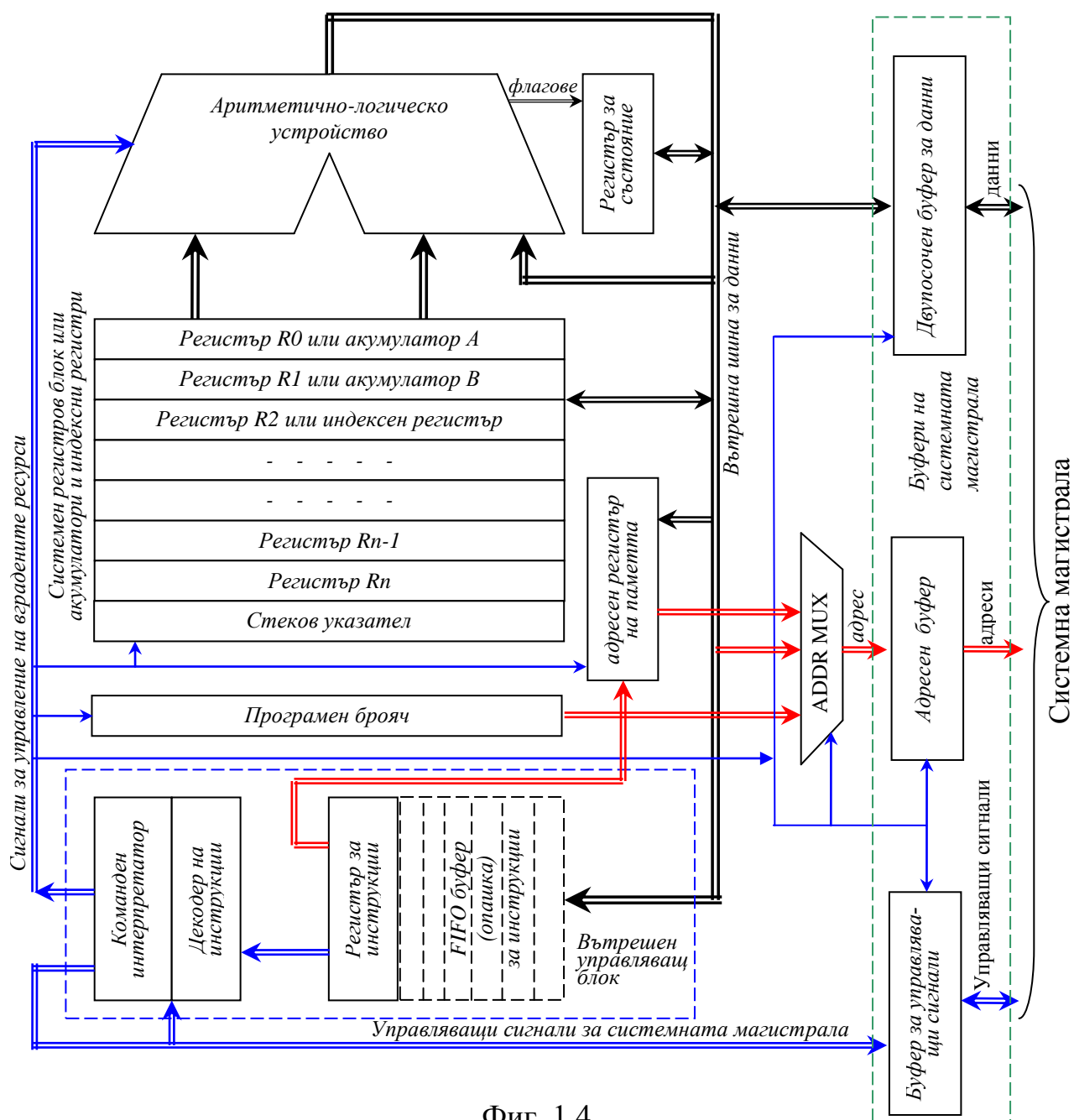
1.3.4. Архитектурни особености и структура на микропроцесорите

Огромно е разнообразието на съществуващите към днешен ден микропроцесори. Архитектурата на системата – Фон Нойманова или Хардварска – естествено оказва влияние на начина на изграждане на самите микропроцесори. От друга страна се счита, че архитектурата на самите микропроцесори се свежда до два основни типа – такива, които съдържат регистри с общо назначение, и микропроцесори от акумулаторен тип.

Акумулаторите са регистри за временно съхранение на данни, които се използват в процеса на изчисленията или логическите операции. В повечето от случаите в акумулатор се пази единия от операндите, върху който се извършва текущата операция. Това са най-често използваните регистри. Обикновено в микропроцесора има един или два акумулатора.

Регистрите с общо предназначение са повече на брой – 4, 6, 8 до 16. Те изпълняват различни функции – за съхранение на данни или адреси, да съхраняват междинни резултати или да се използват като акумулатори.

Независимо от огромното разнообразие на съществуващите микропроцесори, съществуват основни функционални блокове, които са присъщи за всеки един от тях. Обобщената структурна схема на микропроцесор с магистрали от фон Нойманов тип е показана на фиг.1.4. Тя отразява повечето възможни случаи и може да бъде приведена в съответствие със структурата на съществуващите микропроце-



Фиг. 1.4

сори. Не е трудно организацията на микропроцесора така да бъде модифицирана, че данновият поток да бъде отделен от потока машинни кодове.

Основен функционален блок, който задължително присъства във всички микропроцесори, е **аритметично-логическото устройство (АЛУ)**. В него се извършват основните логически и аритметични операции. От разредността му зависи дължината на думата, която може да бъде обработена с една операция. По тази причина **разредността** на даден микропроцесор се определя от и е равна именно на разредността на неговото аритметично-логическо устройство. Често се допуска грешка, като се посочва разредността на данновата магистрала (напр. i8088 е 16 разреден, а данновата магистрала е 8 разредна) или разредността на машинния код (8-разредната фамилия PIC16F8xx използва 14 разреден - !?! – машинен код).

Всеки микропроцесор съдържа и **регистър на състоянията**, в който се съхранява статуса на приключване на операциите в АЛУ – възникнал пренос, заем, препълване, нулев резултат и др.

Програмният брояч (или брояч на инструкциите) също е задължителен елемент за всеки микропроцесор – той съдържа адреса от програмната памет на следващата инструкция, която предстои да бъде изпълнена. След извличане на машинния код, съдържанието на програмния брояч автоматично се инкрементира (увеличава се с 1).

Временно съхранение на данните или адреси на масиви от данни в паметта се извършва във **вътрешен регистров блок** (регистри с общо назначение) или в **аккумулятори** и **индексни регистри**.

В микропроцесорите, които поддържат програмен стек в оперативната памет, има **стеков указател**, с чиято помощ се извършва последователен достъп до клетките от паметта, отделена за стек. След операция „запис“ съдържанието на този регистър автоматично се декрементира (намалява се с 1), а преди четене – автоматично се инкрементира. На практика неговото съдържание винаги сочи първия свободен елемент от стековата памет.

Вътрешният управляващ блок включва регистър за инструкции, декодер на инструкциите и команден интерпретатор. С цел ускоряване на работата понякога се извършва предварително изтегляне на код за следващите инструкции, които се съхраняват в опашка (буфер) за инструкции с организация тип FIFO.

Връзката към системната магистрала става чрез буферен блок, включващ **адресни буфери**, **двупосочни буфери за данни** и **буфери за управляващите сигнали**. Двупосочните буфери за данни осъществяват връзка между вътрешната шина за данни на микропроцесора и шините за данни на микропроцесорната система. Адресните буфери са еднопосочни, чрез тях се подава формираният адрес към паметта или интерфейсите схеми.

Адресният регистър на паметта [6] е регистър за временно съхранение на формирания в резултат на изчислителна операция или получен непосредствено от машинния код на инструкцията адрес от паметта, от където трябва да се прочетат или запишат данните.

Вижда се, че адресите за адресната магистрала на микропроцесора биха могли да се съдържат в програмния брояч, в адресния регистър на паметта или в някой от регистрите на микропроцесора, напр. в стековия указател. Това налага из-

ползване на *адресен мултиплексор*, управляван от командния интерпретатор, превключващ към адресните буфери изходите на програмния брояч, адресния регистър на паметта или съответния регистър чрез вътрешнопроцесорната магистрала за данни.

1.3.5. Как функционира микропроцесорът?

Изпълнението на една инструкция става за няколко такта (фази), които в общия случай могат да се сведат до:

1. **Извличане машинния код на инструкцията.** Чрез *адресния мултиплексор* и *адресния буфер* върху адресните шини от *системната магистрала* се помества съдържанието на *програмния брояч*. От програмната памет (вж. фиг. 1.3) се извлича машинният код на поредната инструкция, който през *двупосочен буфер за данни* и *вътрешната даннова магистрала* постъпва в *регистъра за инструкции*.

2. **Декодиране на машинния код** от *декодера на инструкции* и подаване към *командния интерпретатор*, който изработва необходимите последователности от *управляващи сигнали* към системната магистрала и вградените ресурси (АЛУ, вътрешни регистри, буфери). След приключване на операцията съдържанието на програмния брояч задължително се увеличава с 1;

3. **Извличане на операнд.** В зависимост от типа на адресацията (определя се от машинния код) операндите може да се намират във вътрешните регистри или във външно устройство – памет или интерфейсна схема. Във втория случай е необходимо те да бъдат извлечени, като се инициира цикъл по системната магистрала. Адресът на операнда може да се намира в някой от *вътрешните регистри* или да се определи чрез изчислителна операция, като се запомни в *регистъра за адрес от паметта*. Чрез *адресния мултиплексор* към *системната магистрала* се пропуска съдържанието на регистъра, в който се съдържа адреса според кода на инструкцията. Ако инструкцията предполага използване съдържанието на вътрешен регистър като оператор, извличането на данни чрез системната магистрала се пропуска;

4. **Изпълнителна част** – извършване на аритметични или логически операции в АЛУ. Командният интерпретатор конфигурира АЛУ (чрез подаване на необходимите управляващи сигнали) за операцията, която трябва да се изпълни, и подава на информационните му входове съдържанието на съответните регистри или данните, постъпващи на вътрешната магистрала. Резултатът от изпълнението на съответната функция се намира на изхода на АЛУ, а същевременно се актуализират съответните флагове в регистъра за състояние.

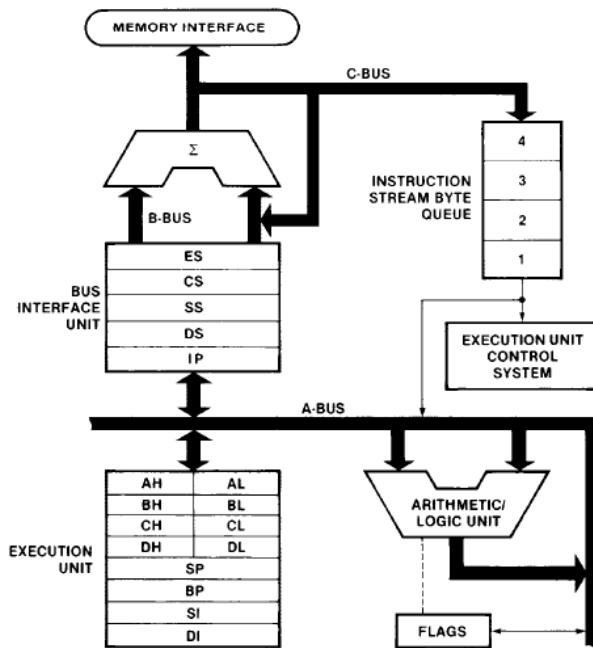
5. **Съхраняване на резултата** – това е заключителната фаза, при която полученият резултат от изхода на АЛУ записва най-често в някой от вътрешните регистри на микропроцесора.

Вижда се, че по време на изпълнение на дадена фаза, част от ресурсите на микропроцесорите стоят неизползвани. Това е дало възможност в почти всички съвременни микропроцесори и в микропроцесорните ядра на множество микроконтролери да се въведат конвейери, позволяващи едновременно изпълнение на отделните фази на последователни инструкции. Това води до еквивалентно повишаване на бързодействието на изчислителното устройство.

1.3.6. Примери за структура на микропроцесори и ядра на микроконтролери

Действителната структура на конкретните микропроцесори или ядра на микроконтролери естествено се различава от разгледаната обобщена структурна схема, запазвайки основните съставни блокове и функционални връзки. Това се илюстрира от приведените структурни схеми на някои популярни микропроцесори и ядра на микроконтролери.

На фиг. 1.5 е представена вътрешната структура на 16-разредния микропроцесор i8088, послужил за основа на първите персонални компютри на IBM. Той е 16 разреден, като обмена с паметта и интерфейсните устройства се извършва посредством 8 разредна даннова магистрала. Адресната магистрала е 20 разредна, като позволява достъп до памет с обем 1 Mbyte.



Фиг. 1.5

Аритметично-логическото устройство извършва операции над 16-разредни данни, които постъпват по вътрешна магистрала, наречена A-BUS. Резултатът от действията е достъпен до регистровия блок чрез същата магистрала. Статусът от приключването на операцията се отразява в регистъра, наречен FLAGS.

Микропроцесорът е изграден с 16-разредни регистри с общо назначение. Първите 4 – **AX**, **BX**, **CX** и **DX** – се използват за временно съхранение на данни и резултати, като всеки от тях може да се използва като два 8-разредни регистъра). Втората група се използва като индексни регистри и включва:

- **SP** (Stack Pointer) – използва се като указател на стека;
- **BP** (Base Pointer) – 16-разреден указател към данни, намиращи се в сегмента на стека;
- **SI** (Source Index) – 16-разреден индексен регистър, използва се за индексване на изходния блок с данни при работа с инструкции за обработка на масиви от данни;
- **DI** (Destination Index) – 16-разреден индексен регистър, използва се за индексване на приемния блок с данни при работа с инструкции за обработка на масиви от данни.

За програмен брояч служи регистърът **IP** (Instruction Pointer), който е 16 разреден. Така неговият капацитет позволява да се работи в адресно поле с големина до 64 Kbytes. За да се разшири адресът до 20 разреда, се използва суматор, в който към съдържанието на IP (или адреса, постъпващ от даден индексен регистър) се прибавя моментната стойност на 16-разреден сегментен регистър, изместена на 4 разреда на ляво. Сегментните регистри са 4, а именно:

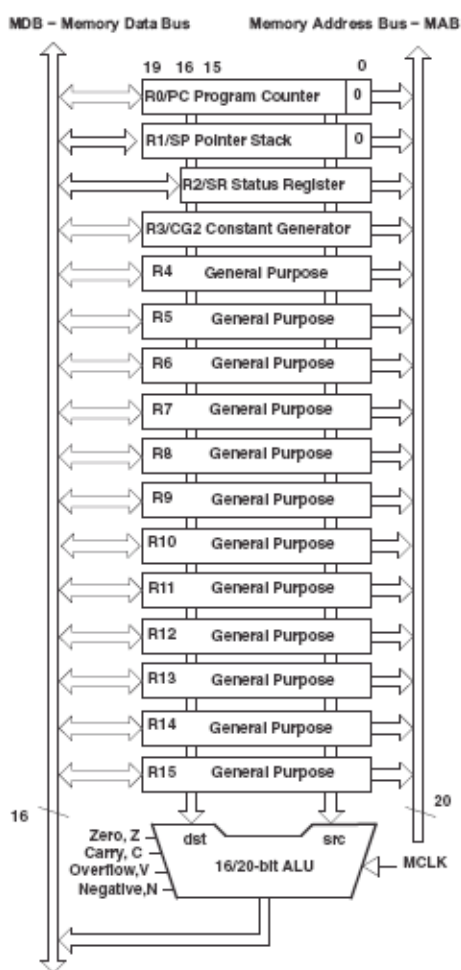
- **CS** (Code Segment) – съдържа адрес на 64 KB сегмент от паметта, в която е разположен изпълним код на програмата. Неговото съдържание се използва за изчисляване на физически адрес в случаите на достъп до инструкции, когато се използва указателя на инструкциите (IP);
- **SS** (Stack Segment) – съдържа адрес на 64 KB сегмент от паметта, заделен за стек на системата. По подразбиране се приема, че при всеки достъп до данни с използване на указателя на стека (SP) или базовия указател (BP), се използва SS, гарантирайки достъп до стековия сегмент от паметта;

- DS (Data segment) – дефинира начален адрес на 64 KB сегмент от данни. По подразбиране се приема, че всички данни, с които работят общите регистри (AX, BX, CX, DX) и индексните регистъри (SI, DI), се намират в сегмента на данните и при определяне на физическия адрес се използва съдържанието на сегментния регистър DS;
- ES (Extra segment) – допълнителен сегмент за данни с големина 64 KB. По подразбиране се приема, че с него се прекодира адреса, съдържащ се в регистъра DI при инструкции, работещи с масиви или стрингове.

Сегментите по подразбиране, използвани от общите и индексни регистри, могат да се променят, ако командният ред в програмата на асемблер съдържа префикс CS, SS, DS или ES.

На фиг. 1.6 е показан регистровият блок и АЛУ на ядрото на фамилията 16-разредни микроконтролери MSP430x5xxx на фирмата Texas Instruments. Това са CISC микроконтролери с Фон Нойманова архитектура. Тази архитектурна особеност не се вижда добре от фигурата, тъй като производителят не е показал връзките с програмната и данновата памет.

Регистровият блок съдържа общо 16 регистъра, като последните 12 от тях (R4 – R15) са



Фиг. 1.6

во-аналогови преобразуватели, генератори на контролни кодове и др. Много добре е развита системата за редуциране на консумацията, която използва възможностите за изключване на определени ресурси и на микропроцесорното ядро. Това я прави особено подходяща за приложения, критични от към консумация на електроенергия (при устройства с батерийно захранване).

Подробна функционална схема на микропроцесорното ядро на фамилията микроконтролери PIC18FxxKxx на фирмата Microchip е показана на фиг. 1.7. От нея много добре се вижда, че предпочетената архитектура е Хардвардска, а организацията на ядрото е с регистри от акумулаторен тип. Ядрото работи с редуцирана система инструкции (RISK).

16-разредни и са с общо назначение – за временно съхранение на данни, операнди, запис на резултата от логически и аритметични операции. Същите могат да се използват и като индексни регистри при операции за последователен достъп до масиви от данни.

Регистърът R0 е 20-разреден и се използва за програмен брояч в системата. Това гарантира достъп до адресно поле с обем 1 Mbyte. Във фамилията има контролери с вградена 256 Kbyte програмна памет.

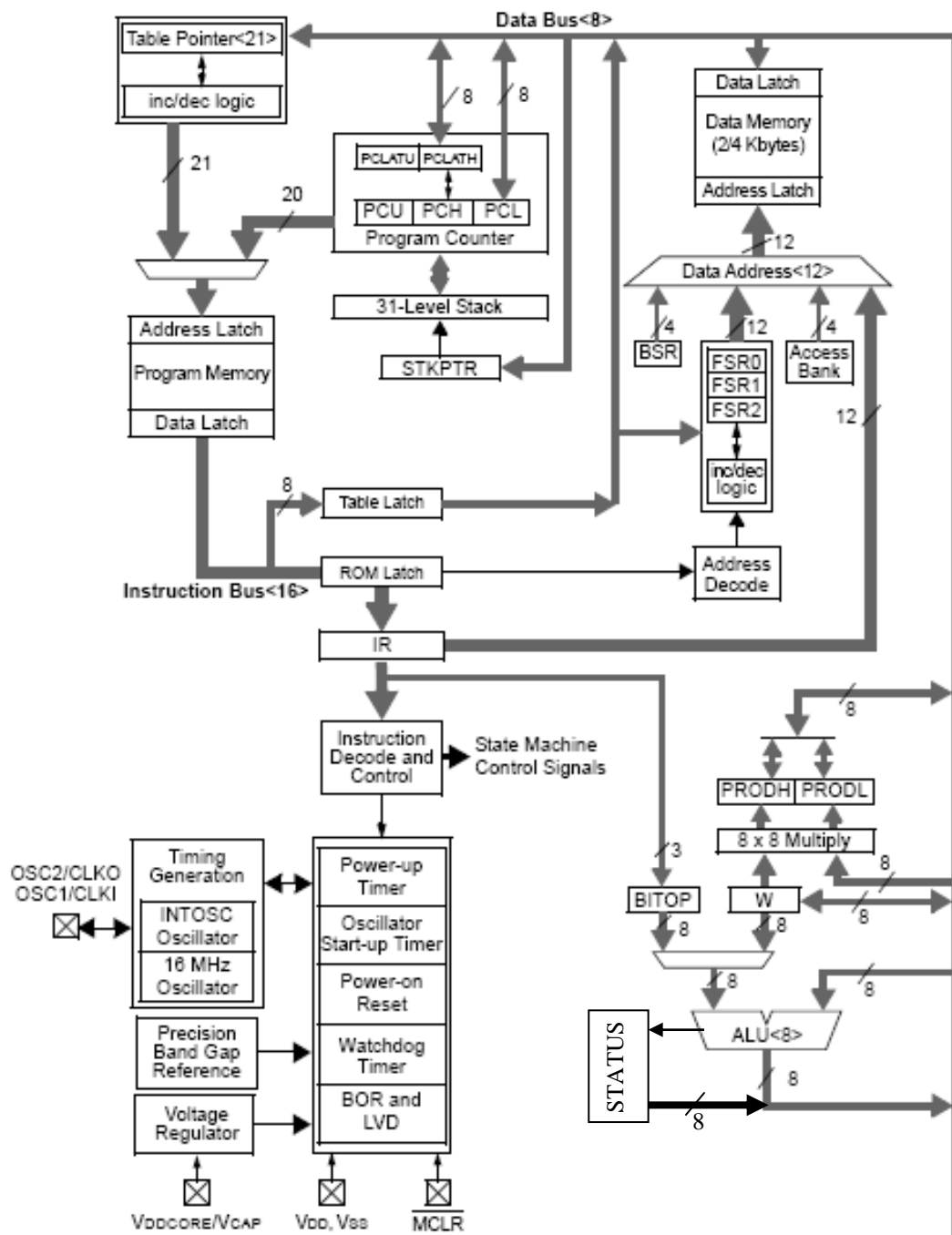
Регистърът R1 също е 20 разреден и се използва като стеков указател, което дава възможност стековата памет да бъде разположена в която и да е 64 KB страница от паметта, достатъчно е в нея да има разположена оперативна памет. Стекът се използва за съхранение на възвратния адрес при извикване на подпрограми и при прекъсвания. Съществува възможност за съхранение на данни в стека по програмен път.

Регистърът R2 се явява регистър на състоянието. Той е 16-разреден, като за флагове се използват само младшите 9 разряда.

Много интересно хрумване е реализирано в регистъра R3. Той е 16 разреден и е генератор на 6 различни константи.

Фамилията микроконтролери се характеризира с добре развита периферия, включваща серийни и паралелни интерфейси, разнообразни таймерни блокове, аналогово-цифрови и цифрови

Аритметично-логическият блок оперира с 8-разредни операнди, което позиционира микроконтролера като 8 битов. Единият оператор винаги постъпва по вътрешната 8-разредна ма-



Фиг. 1.7

гистрала за данни, а вторият – или от акумулатора **W**, или константа, зададена от машинния код. Към изчислителния блок спада и апаратен умножител на 8-разредни целочислени променливи. Състоянието на приключване на логическите или изчислителни операции се отразява в регистър на състоянията, който тук е наречен **STATUS** – 8-разреден регистър, от който се използват младшите 5.

Паметта за данни е с обем 2 или 4 Kbyte (при различните микроконтролери от фамилията). Интерфейсните схеми са разположени в адресното поле на паметта. Достъпът до паметта става чрез 12-разредна адресна магистрала, като е предвидена възможност както за линеен достъп (чрез 3-те 12-разредни индексни регистъра **FSR0** – **FSR1**), или чрез сегментиране с използване

на 4-разреден регистър **BSR** за избор на съответната банка. Така се получават 16 банки по 256 байта всяка.

Програмният брояч (**PCL,PCH,PCU**) е 20-разреден, осигурявайки достъп до 1 MByte програмна памет. В отделните микропроцесори на фамилията има вградена от 32 Kbyte до 256 Kbyte енергонезависима памет flash.

За осигуряване на работа с прекъсвания и обръщение към подпрограми е предвиден хардуерен стек на 31 нива, където се съхраняват възвратните адреси. Съдържанието на указателя на стека (**STKPTR**) е достъпно в регистровото поле от интерфейсни схеми.

Има предвиден механизъм, който позволява таблици, позиционирани в програмната памет, да бъдат четени и интерпретирани като данни – за целта се използват регистрите **Table Pointer** и **Table Latch** при съблюдаване на определен алгоритъм за достъп.

1.3.7. Програмен модел на микропроцесорите

Съвкупността от вътрешни регистри на микропроцесора, които са програмно достъпни, т.е. тяхното съдържание може да бъде прочетено или променено по програмен път или влияят на режимите на работа, се нарича програмен модел на микропроцесора. Това е много важна негова характеристика, тъй като дефинира инструментариума, който е на разположение на програмистите за решаване на тяхната задача.

За хипотетичния микропроцесор от фиг. 1.4, който разгледахме по-горе, програмният модел включва:

- Програмния брояч – **PC**;
- Регистъра за състояние – **CCR**;
- Стековият указател – **SP**;
- Регистъра **R0** (или акумулатор **A**);
- Регистъра **R1** (или акумулатор **B**);
- Регистъра **R2** (или индексен регистър **IX**);
- (всички налични регистри);
- Регистър **Rn**.

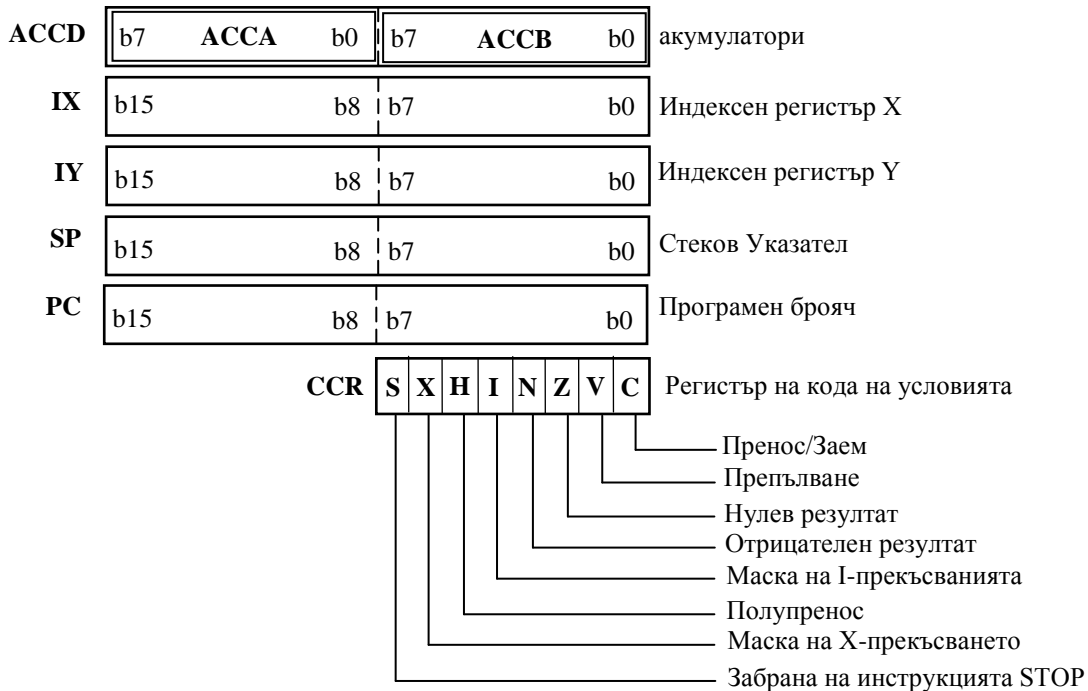
Обърнете внимание, аритметично-логическото устройство, регистърът за адрес от паметта, адресният мултиплексор, елементите на вътрешния управляващ блок и магистралните буфери не се включват в програмния модел, тъй като върху тях не може директно да се влияе по програмен път.

Програмен модел на MC68HC11. Понякога производителите на микропроцесори представят описанията на програмните им модели в графичен вид. На фиг. 1.8 е показан програмният модел на микропроцесорното ядро на микроконтролера MC68HC11. Той съдържа два 8-разредни акумулатора за временно съхранение на данни и операнди, както и два 16-разредни индексни регистъра. Програмният брояч и стековият указател са 16-разредни, осигурявайки работа с 64 Kbyte адресно поле.

Регистъра за кода на условията тук е 8-разреден. Представено е значението на флаговете, които се поддържат. Младшите 4 от тях присъстват на практика във всеки един микропроцесор и отразяват статуса на приключване на аритметичните и логически операции:

- **b0 – C (CARRY/BORROW FROM MSB)** – възникнал е пренос или заем от старшия разряд при операцията;

- b1 – V (OVERFLOW) – установява се в 1, ако в резултат на аритметична операция е възникнало препълване;
- b2 – Z (Zero) – установява се в 1, ако прехвърлените данни или резултатът от последната аритметична или логическа операция е нула;
- b3 – N (Negative) – установява се в 1, ако прехвърлените данни или резултатът от последната аритметична или логическа операция е отрицателен. Приема се, че резултатът е отрицателен, ако старшият бит е 1.



Фиг. 1.8

Тези 4 флага на практика присъстват в регистрите за код на условието на всички микропроцесори. Следващите 4 флага силно се отличават при отделните микропроцесори, те могат да отразяват състоянието на микропроцесорната система или да управляват режима на работа на микропроцесора. В разглеждания пример тяхното назначение е:

- b4 – I (Interrupt Mask) – ако е установен в 1, забранява (възпрепятства) възприемането на прекъсване от всички маскируеми източници на прекъсване – външни и вътрешни. Може да се активира програмно или в резултат на постъпило външно прекъсване;
- b5 – H (Half Carry) – установява се в 1, ако при логическа или аритметична операция възникне пренос от младшата към старшата тетрада в байта (между 3 и 4 бит);
- b6 – X – (X Interrupt Mask) – забранява възприемане на заявка за немаскируемо прекъсване от вход XIRQ. Установява се в 1 само по хардуерен път (след RESET или XIRQ) и се нулира само чрез програмна инструкция (върщане от прекъсване RTI или презапис на ACCA в CCR – TAP);
- b7 – S – (Stop Disable) – при установяване (програмно) в 1 забранява изпълнение на инструкцията STOP (тя се изпълнява като празна операция NOP);

Програмният модел на микропроцесора i8088 (вж. фиг. 1.5), включва:

- регистрите с общо назначение AX, BX, CX и DX;
- указателните (индексни) регистри SP, BP, SI, DI;
- сегментните регистри CS, SS, BS и ES;
- програмният брояч IP;
- регистъра за код на условията FLAGS.

Програмният модел на ядрото на микроконтролера MSP430x5xxx (вж. фиг. 1.6) включва 16-те регистъра R0 – R15. Това означава, че отново към него попадат програмният брояч R0, стековият указател R1 и регистърът за код на условията R2. Списъкът се допълва от генератора на константи R3, както и регистрите с общо назначение R4 – R15.

Програмен модел на ядрото на микроконтролерната фамилия **PIC18FxxKxx** (вж. фиг. 1.7). Тук моделът не е така ясен и видим от пръв поглед, защото всички регистри – и на микропроцесорното ядро, и на интерфейсите схеми и устройства (не са показани на фигурата и са различни за различните представители на фамилията), се намират в общ регистров блок. Разликата е, че докато съдържанието на регистрите на периферните схеми се отнася до работата на самите схеми, то регистрите, участващи в програмния модел на микропроцесорното ядро касаят режима на работа на ядрото, досъпа до ресурсите и изпълнението на самата програма. Програмният модел включва следните регистри:

- програмен брояч **PC**. Той условно е разделен на 3 части – младша част PCL (8-разреда), старша част PCH (8-разреда) и разширение PCU (4-разреда). Към него спадат и два междинни регистъра PCLATH и PCLATU. За промяна на съдържание на PC е необходимо новите стойности за старшата част и разширението да се запишат в междинните регистри, а след това се прехвърлят в PCH и PCL при изпълнение на определени инструкции за преход. Така се получава механизъм за секционирание на програмната памет, което дава възможности за разделяне и защита на отделните програмни модули;
- регистър за код на условията **STATUS**. Той е с размерност 8 разреда, от които се използват младшите 5. Тук отново има флагове за пренос/заем, препълване, нулев и отрицателен резултат;
- акумулатор **W**. Използва се за временно съхранение на данни и операнди;
- индексни регистри (регистри за индиректен достъп) **FSRn**. В ядрото има 3 броя 12-разредни регистъра, чрез които се осъществява индиректен достъп до данновото адресно поле. Всеки регистър е разделен на две – младша част **FSRnL** и старша част **FSRnH**. От старшата част се използват само младшите 4 бита.
- Регистър за избор на банка от паметта **BSR** – 4-разреден регистър, чието съдържание се използва като старша част на адреса към данновата памет при директен достъп към нея.

От направения преглед се вижда, че всеки микроконтролер се характеризира със специфичен набор регистри, дефиниращи програмния му модел. Доброто познаване на програмния модел на използвания микроконтролер гарантира ефектив-

ното му използване и минимизира „изненадите“ при функциониране на създаденото програмно осигуряване.

С нарастване на възможностите на микроконтролерите и сложността на системното и приложно програмно осигуряване асемблерите все повече губят своето място, отстъпвайки го на езици от по-високо ниво, най-вече *C*. При използване на такива езици въпросът за програмния модел на микропроцесора за потребителя минава на по-заден план. Първостепенна роля добива въпросът, доколко добре са обхванати архитектурните особености на използвания микропроцесор (микроконтролер) в използвания език от високо ниво.

1.3.8. Специфични функционални особености на микропроцесорите

Важно е да се разбере спецификата на работа на един микропроцесор (ядро на микроконтролер). Основните механизми, които определят начина му на функциониране към даден момент, водят до обособяване на функционалните подсистеми, обуславящи дадения микропроцесор.

В по-голямата част от времето микропроцесорът е зает да извършва обмен на данни или операнди от/към оперативната/програмната памет за данни или интерфейсни устройства, изпълнявайки алгоритъма на работа, заложен в системното или приложно програмно осигуряване. Начинът, по който се извършва тази комуникация, се определя от апаратните особености, заложен в организацията и управлението на буферния блок за връзка със системната магистрала (вж. фиг. 1.4). Той е специфичен за всеки отделен микропроцесор и е водещ при изграждането на конкретната микропроцесорна система. Подсистемата за обмен на данни (*Data Transfer Subsystem, DTS*) е една от основните подсистеми и характерните ѝ особености проличават при изграждане на връзката с програмната и оперативна памет и интерфейсните схеми. Тези въпроси по-подробно се разглеждат в глава 1.4 – обмен на данни с програмна и оперативна памет.

Втората съществена подсистема, заложена във всички съвременни микропроцесори и микроконтролери (изключенията от правилото могат да бъдат пренебрегнати – виж фамилията PIC10F2xx на Microchip), е подсистемата за премане на прекъсвания (*Interrupts Subsystem, IS*). Настъпването на външни за системата събития в произволен момент спрямо вътрешно-системното време, които се отразяват на работата на системата (*обслужват се* от системата), се възприемат именно благодарение на възможностите, които предлага тази подсистема. Същественото тук е, че микропроцесорът не преустановява своята работа, а „отлага“ изпълнението на текущата програма и се заема с процедура по „*обслужване*“ на постъпилата „*заявка за прекъсване*“. По-подробно функционално тази система е описана в глава 1.5 – подсистеми за прекъсване при микропроцесори и микроконтролери.

Третата подсистема, свързана с начина на функциониране на микропроцесорите, е свързана с възможността за преустановяване на тяхната работа и предоставяне управлението на системните ресурси на друго устройство – контролер за управление на паметта (DMAC) или друг микропроцесор. В този случай е необходим механизъм, който да арбитрира достъпа до управлението чрез въвеждане на определени принципи за арбитриране. В някои микропроцесори (и ядра на микроконтролери, които имат вградени DMAC) има предвидени подсистеми за арбит-

риране на достъпа до DTS (Bus Arbitration Control, ВАС). Този механизъм е в основата на изграждане на многопроцесорни системи и не е обект на разглеждане на този курс.

1.3.9. Режими на работа на микропроцесорите

В процеса на работа на микропроцесорите изискванията към тях и условията на работа може да се променят. Това е наложило вграждането на механизми, които дават възможност да се променя начина на тяхното функциониране в зависимост от определени настройки (инициализации) на режимите.

Една от възможностите е микропроцесорът да работи в **системен** или **потребителски режим**. Обикновено в системен режим (изпълнение на системното програмно осигуряване) ограничения не се налагат, докато в потребителски режим (изпълнение на потребителски програми) се ограничава изпълнението на отделни инструкции. При по-развити системи това може да се разпростре и върху достъпа да определени системни ресурси – памет и интерфейсни схеми – в зависимост от режима на микропроцесора. Обикновено превключване от системен в потребителски режим става по програмен път (изпълнение на определена инструкция), а връщането в системен режим – най-често чрез прекъсване (апаратно или програмно).

В съвременните микроконтролери различните режими на работа най-често се изразяват в управление на включване/изключване на определени ресурси или промяна на работната честота, та дори и изключване на самото ядро на микроконтролера. По този начин в много широки граници може да се променя консумацията на цялата микропроцесорна система. Това има много голямо значение в случаи, когато се налагат ограничения на потреблението на енергия (автономни системи, работещи от батерийни източници в екстремни условия).

1.4. ОБМЕН НА ДАННИ С ПРОГРАМАНА И ОПЕРАТИВНА ПАМЕТ И ИНТЕРФЕЙСНИ СХЕМИ

Подсистемата за обмен на данни включва адресните сигнали, линиите за данни и управляващите сигнали. Опциите (параметрите) на всяка една от тези съставки характеризират конкретните особености на подсистемата.

Адресни линии. При различните микропроцесори броят на адресните линии се променя в широки граници. Той дефинира обема на паметта, която може непосредствено да бъде адресирана. Най-често микропроцесорите поддържат 16, 20, 24 или 32 адресни линии. В последните години се появяват мощни микропроцесори, поддържащи 64 адресни линии.

Даннови линии. Размерността на данновите линии също варира в доста широки граници – от 8 разреда до 64 разреда. Този параметър определя колко байта могат да бъдат обменени по време на един цикъл на магистралата за обмен на данни. Най-главната характерна особеност на шината за данни е, че тя е двупосочна – по нея микропроцесорът може както да записва (предава), така и да чете (приема) данни. Разредността на шината за данни не е пряко свързана с разредността на микропроцесора. Нещо повече, съществуват разновидности на почти идентични микропроцесори, които имат даннови шини с различна разредност

(напр 8086/8088 или MC68000/MC68008, като първите от цитираните двойки микропроцесори имат по 16 даннови линии, а вторите – само по 8).

Управляващи сигнали. Най-съществени различия има при съвкупността от управляващи сигнали. По отношение на принципа на управление на обмена се обособяват две големи групи – микропроцесори със синхронен обмен на данни и такива с асинхронен обмен на данни. В първия случай в системата има синхронизиращ сигнал, по време на който се извършва обменът (напр. сигналът Е при системи MC6800/MC68HC11). Във втория има предвидени сигнали, чрез които устройствата, участващи в обмена, декларират своята готовност за приключването му (напр. DSx/DTASK при системи, базирани на MC68000).

Задължително, в явна или неявна форма, съществува сигнал, който определя посоката на трансфер на данни по данновите линии по време на текущия цикъл за обмен на данни. Това, например, е сигналът R/W при системите MC6800/68HC11, който при високо ниво предопределя четене на данни (посока от паметта към микропроцесора), а при ниско ниво – запис на данни (посока на обмен от микропроцесора към паметта). Този сигнал може логически да бъде обединен с други сигнали, напр. за достъп до паметта или интерфейсни схеми, тогава се получават няколко сигнала за управление на дадена група – MEMR (четене от памет), MEMW (запис в памет), IOR (четене от периферни схеми), IOW (запис в периферни схеми). Подобни сигнали се изработват в системните контролери на INTEL при микропроцесори 8085, 8086, 8088 ...

Мультиплексиране на магистралите. Много често с цел минимизиране броя на изводите на микропроцесора се прилага мультиплексиране, най-често на адресни и даннови линии (напр. MC68HC11, i8086/i8088 и др.). В този случай на определени изводи първо се помества адреса (най-често младшата му част), а след това същите линии се използват за трансфер на данни. Ясно е, че в този случай адресната информация е необходимо да бъде запомнена в буферен адресен регистър. Като такъв най-често се използват интегрални схеми от типа 74HCT374 или 74HCT574, които представляват комбинация от 8-разреден „прозрачен” регистър (едноетажни D-тригери) и буфер с повишена товароспособност. За управление на буферния адресен регистър има предвиден специален управляващ сигнал – напр. AS при MC68HC11 или ALE при 8086/8088.

Блоков обмен на данни. С цел ускоряване работата на системите в по-големите микропроцесори са заложили възможности за блоков трансфер на данни. Идеята е с едно предаване на адрес на данните да се реализира достъп до няколко последователни блока с данни. За пръв път механизмът е реализиран в микропроцесора 80486 и предвижда в един цикъл на четене /запис да се извърши трансфер на 4 групи по 4 байта. Естествено, за да се осъществи този режим, той трябва да се поддържа (да е заложен на хардуерно ниво) и в паметта, с която се извършва обмен.

Контрол на данните. Много е важно данните в програмната и данновата памет да не бъдат разрушавани в процеса на работа на системата. При системи с повишени изисквания за отказоустойчивост се предвиждат механизми за откриване на възникнали грешки.

По отношение на системните програми, записани в постоянни (енергонезависими) памети, ефективен механизъм се явява определянето на контролни циклични кодове на базата на съдържанието на определен блок от паметта и записването му в началото или края на блока. Преди използване на съответното програмно осигуряване се проверява целостта на програмния блок и ако ново определеният код не съответства на записания, модулът се приема за невали-

ден. Така, ако програмният код е разрушен, не се допуска стартиране на системата и се избягва възможността за подаване на грешни управляващи сигнали към изпълнителни устройства. Контролът, който предполага този метод, е за сметка на незначително усложняване на програмното осигуряване. Недостатък е, че той не е перманентен, а се осъществява само в определени моменти от време.

По-трудно се следи за верността на данните в оперативната памет. В подобни системи обикновено се добавя допълнителен 9 бит към всеки байт на оперативната памет. При операция „запис“ се извършва определяне четността на всеки байт и резултатът се записва в този бит. При четене отново се извършва проверка за четност и се сравнява със записаното състояние при предходната операция. Ако има разлика, се генерира сигнал за грешка, който системата трябва да отработи. Видно е, че контролът в този случай става за сметка на вграждане в системата на допълнителни апаратни ресурси, което води до усложняване и оскъпяване на системата.

Системи за контрол верността на данните вече се вграждат и в някои съвременни микроконтролери. Например, в микроконтролера MSP430F6659 [8] е вграден механизъм на базата на контрол на четността, който позволява следене на целостта на данните във вградената енергонезависима памет. Това става с помощта на модул за следене целостта на данните - Memory Integrity Detection (MID).

При системи, изградени на базата на микроконтролери, често се използват паметни, използващи серийни интерфейси, напр. SPI или I²C, които ще бъдат разглеждани по-късно.

1.5. СИСТЕМИ ЗА ПРЕКЪСВАНИЯ ПРИ МИКРОПРОЦЕСОРИ И МИКРОКОНТРОЛЕРИ

Подсистемата за прекъсване дава възможност микропроцесорната система да се синхронизира с външни процеси и събития, които настъпват асинхронно спрямо нейната работа.

Грешно е да се счита, че прекъсването преустановява работата на микропроцесора. То всъщност принуждава микропроцесора да преустанови изпълнението на текущата програма и да пристъпи към изпълнение на друга програмна процедура, включваща действия, свързани с новата ситуация – казва се, че системата „обслужва“ даденото прекъсване. След приключване на задачите по обслужване на прекъсването микропроцесорът се връща към предходната програма и продължава нейното изпълнение.

Владеенето на системата за прекъсване на даден процесор предполага:

- познаване на източниците на прекъсване;
- начина на тяхното разпознаване;
- приоритетен ред за тяхното обслужване;
- механизмите за съхранение на статуса и мястото на прекъсване на текущата програма;
- начините за извличане (определяне) на стартовия адрес на процедурата (програмата) за обслужване на прекъсване;
- обслужване на прекъсването, включително изчистване на флаговете, предизвикващи това събитие;
- възстановяване статуса на микропроцесора и връщане в предходната програма в точката на нейното прекъсване.

Източниците на прекъсване най-общо се делят на два вида – немаскируеми и маскируеми. Както личи от наименованието им, първите не могат да бъдат „мас-

кирани” (забранени) – т.е. да бъде предотвратено тяхното възприемане чрез установяване на съответен флаг по програмен път. Вторият вид прекъсвания могат временно да бъдат игнорирани, като по програмен път се забрани тяхното възприемане. За немаскируемо прекъсване с най-висок приоритет се счита сигналът за начално установяване на системата – **RESET**.

Източниците на прекъсване от един тип (напр. маскируеми) се степенуват по приоритетен ред. При едновременно възникване на две заявки за прекъсване първо се обслужва тази, която е с по-висок приоритет. Втората заявка „изчаква”, докато приключи обслужването на заявката (заявките) с по-висок приоритет.

Най-сериозните различия между подсистемите за прекъсвания касаят начина, по който се определя стартовият адрес на процедурата, обслужваща даденото прекъсване. Могат да бъдат различени три начина, които са заложи в подсистемите за прекъсване, за определяне приоритета на прекъсването и стартовия адрес на процедурата (програмата), която трябва да бъде изпълнена:

1. **Директна система.** При нея стартовият адрес на програмата за обработване на прекъсването е фиксиран за дадения тип микропроцесор – т.е. процедурата за обслужване на прекъсването се намира винаги на едно и също място в програмната памет. Така е например при 8080 на INTEL или при фамилията микроконтролери PIC16Fxxxx на Microchip. Независимо от източника на постъпилото прекъсване започва изпълнение на програмата от този фиксиран адрес. Ако в системата съществуват няколко източника на прекъсване, програмата трябва да ги сканира последователно, за да установи коя от постъпилите заявки е с най-висок приоритет. Самият приоритет не е зададен апаратно, а е заложен в алгоритъма на програмата.

2. **Автовекторна система.** При нея на фиксирани адреси в паметта, характерни за дадения тип микропроцесор, е необходимо да бъдат записани стартовите адреси на програмите, които се изпълняват при постъпване на съответния тип прекъсване. Съдържанието на тези клетки са нарича вектори на прекъсванията, а масивът, в който те са записани – таблица на векторите на прекъсванията. От този тип са системите на прекъсване на MC6800, MC68HC11, i8085 и др.. Системата разделя прекъсванията според техния вид – маскируеми, немаскируеми и пр., но ако постъпват (напр. обединени по „жично или”) прекъсвания от няколко източника от един и същи вид, необходимо е те да бъдат разграничени чрез допълнително сканиране по програмен път.

3. **Векторна система.** При нея при постъпване на прекъсване процесорът изпълнява процедура по потвърждаване на прекъсването. Най-често се изгражда външна разпределена система за определяне прекъсването с най-висок приоритет (при системи, базирани на MC680x0) или в системата се вгражда приоритетен контролер на прекъсванията (при системи, базирани на i80x86). В резултат на това устройството, издало заявка с най-висок приоритет, предава към микропроцесора (чрез ресурсите на подсистемата за трансфер на данни) **векторен номер**. На базата на този векторен номер микропроцесорът определя адреса от векторната таблица, на който се намира векторът (адресът) на обслужващата програма. Така на едно и също приоритетно ниво могат да се издават заявки за прекъсване от множество устройства, като за всяко от тях има отделна програма за обслужване, до стар-

товия адрес на който се достига автоматично без изпълнение на програмна процедура, която да проверява всички възможни източници.

Като се изключат особеностите при определяне стартовите адреси на процедурите, обслужващи отделните заявки за прекъсване, фазите, които се изпълняват от отделните микропроцесори (или ядра на микроконтролери), са сходни и се свеждат до:

- Проверка на **флаговете** дали е разрешено **възприемането на прекъсване** на съответните приоритетни нива (само за маскируемите прекъсвания);
- Проверка на **флаговете** за приоритет на прекъсванията и подготовка за обслужване на това с най-висок приоритет;
- **Съхраняване на текущото състояние и възвратен адрес**. В системния стек се съхранява текущата стойност на програмния брояч. Това гарантира, че след обслужване на прекъсването управлението ще се върне на текущата програма в точката на прекъсване. При повечето микропроцесори заедно с програмния брояч се съхраняват и вътрешните регистри, изграждащи програмния модел на микропроцесора. Ако това не е предвидено (както е при някои микроконтролери), е необходимо първите инструкции на процедурата за прекъсване да гарантират съхранение преди всичко на регистъра с кода на условията и на останалите регистри на микропроцесора (ядрото на микроконтролера);
- Определяне и **извличане на стартовия адрес (вектора)** на процедурата за прекъсване в съответствие с механизмите, разгледани по-горе;
- Установяване на **флаг, забраняващ** възприемането на **прекъсвания** с приоритет, по-нисък и равен на текущия;
- Изпълнение на необходимите операции по **обработване на прекъсването**. Това по същество е програмата, която се стартира за обслужване на даденото прекъсване. Нейният алгоритъм се определя от последователността от действия, които е необходимо да бъдат изпълнени в резултат на настъпване на регистрираното събитие;
- **Изчистване** на заявката (**флага**) за прекъсване, изработен от интерфейското устройство и довел до възникване на прекъсването. Това става най-често чрез прочитане или запис на определен регистър от тази периферия;
- **Възстановяване съдържанието на регистрите**, съхранени към момента на възникване на прекъсване – само в случаите, когато това не е заложено да става автоматично при връщане от прекъсване;
- **Връщане от прекъсване**. Обикновено става чрез изпълнение на една инструкция, специфична за дадения микропроцесор. При нея се извършва възстановяване (прочитане от стековата памет) на съдържанието на вътрешните регистри (ако те са били съхранени там) и възстановяване на програмния брояч – това е моментът на връщане към програмата, изпълнявана в момента на прекъсване.

Използването на възможностите, които предлага подсистемата за прекъсване дава възможност да се повиши ефективността на работа на микропроцесорните системи с външни ресурси. Доброто познаване на механизмите, които предлага конкретния микропроцесор или микроконтролер, за възприемане на заявки за

прекъсване е предпоставка за създаване на ефективни системи за контрол и управление на разнообразни технически устройства.

1.6. КАРТА НА АДРЕСНОТО ПОЛЕ. ДЕКОДЕРИ НА АДРЕСИТЕ.

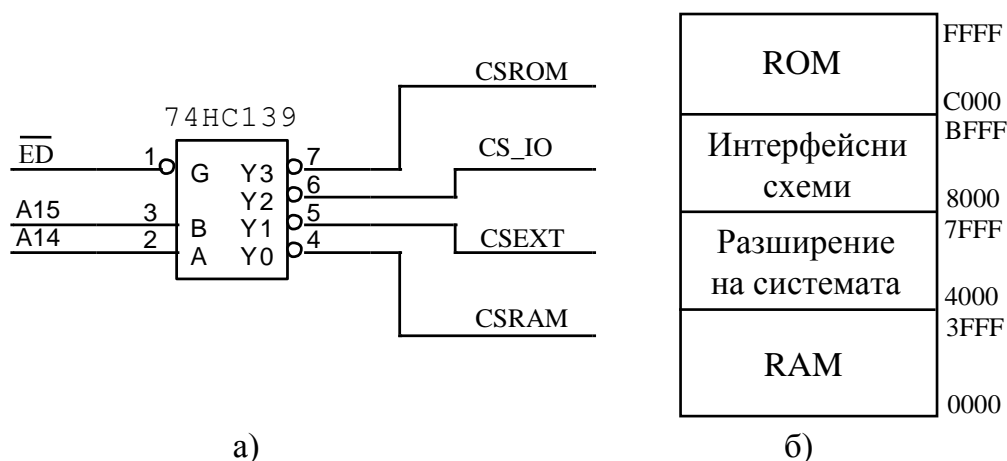
Въпросът с рационалното разпределение на ресурсите в дадена микропроцесорна система е основен за ефективното и функциониране. Разположението на отделните ресурси във физическото адресно поле се определя от структурата и конкретното изпълнение на **системния адресен дешифратор** (вж. фиг. 1.3).

Във всяка микропроцесорна система има поне един първичен системен дешифратор – в случаите, когато архитектурата на систематата е от фон Нойманов тип и входно-изходните схеми споделят едно и също поле с паметта. Ако в системата са предвидени различни полета за памет и входно-изходни схеми (както е при фамилиите 8080, 8085 и 80x86), тогава се налага изграждане на отделен първичен дешифратор за всяко поле. При системи с харвардска архитектура е необходимо да има отделни дешифратори за схемите в полето на програмната памет и за тези в полето на данновата памет.

Декодиране на адресното поле се извършва и в микроконтролерите, като дешифраторите са вградени в системите. Общото в двата случая (микропроцесорна система или микроконтролер) е, че структурата на системния дешифратор обуславя разпределението на ресурсите в адресното поле, което се изразява в **картата на паметта**. Тя представлява графично изобразяване на разположението на ресурсите на системата в рамките на адресното поле.

1.6.1. Симетрични дешифратори на адресното поле

Нека приемем, че системата от фиг.1.3 разполага с 16 адресни линии и единно адресно поле за памет и входно-изходни схеми. Едно възможно решение на системния (първичен) адресен дешифратор е показано на фиг. 1.9.а. Разпределението на ресурсите в адресното поле (картата на паметта), което се обуславя от този адресен дешифратор, е показано на фиг. 1.9.б.



Фиг. 1.9.

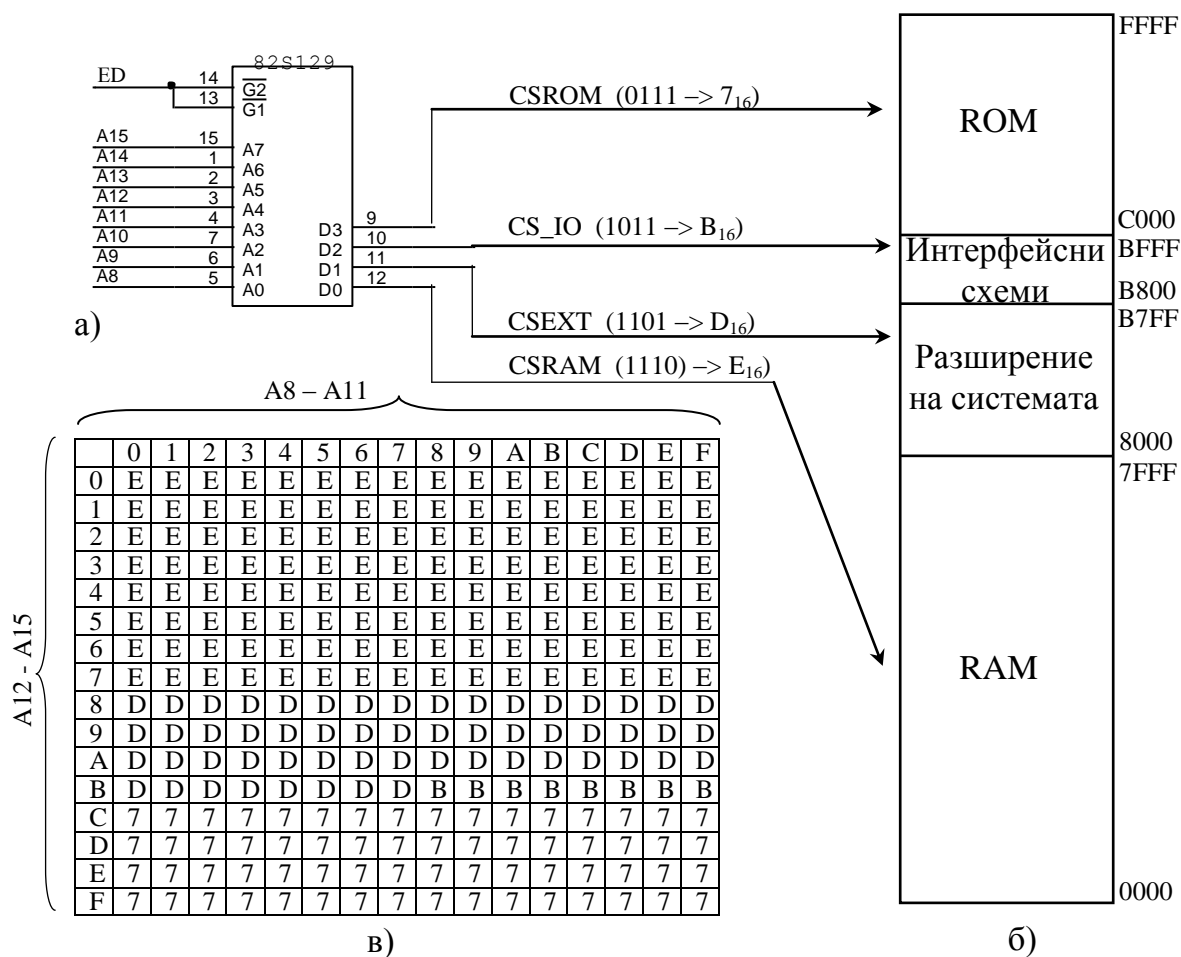
Използването на стандартен декодер 2:4, тип 74HC139, разделя цялото адресно поле от 64 Кbyte на 4 полета с еднаква големина по 16 Кbyte. Реализираният декодер е едностъпален и закъснението при изработване на сигналите за избор

е равно на закъснението на използвания стандартен декодер. Работата на декодера се разрешава от сигнала ED, указващ, че на адресните шини има валиден адрес от паметта. Декодира се състоянието на двете старши адресни линии A14 и A15.

Подобни дешифраторни схеми, при които се получават **еднакви по големина области** в картата на паметта, се наричат **симетрични дешифратори**. Това разпределение на паметта не винаги е удачно – явно е, че зоната за интерфейсни схеми е прекалено голяма и никога няма да се използва напълно и ефективно.

1.6.2. Несиметрични дешифратори на адресното поле

Друго възможно решение е да се използват програмируеми схеми – CPLD или PROM. Те дават възможност да се създаде несиметричен дешифратор, при който да се постигне ефективно **несиметрично разпределение на адресното пространство**. Друго предимство е, че конфигурацията на системата може да бъде променяна чрез прекодиране съдържанието на PROM-а.



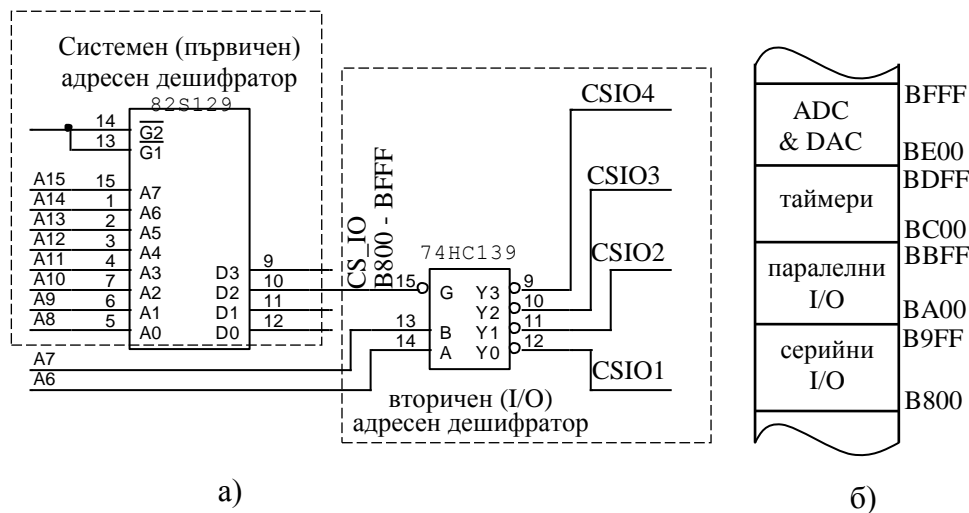
Фиг. 1.10

Едно възможно решение с използване на TTL PROM е показано на фиг. 1.10.а. На фиг.1.10.б е показана картата на паметта, която искаме да бъде реализирана – с 32 Kbyte оперативна памет в началото (адреси 0000 – 7FFF₁₆), с 16 Kbyte EEPROM в края (адреси C000₁₆ – FFFF₁₆) и 2 Kbyte (от адрес B800₁₆ – BFFF₁₆) за интерфейсни схеми. Останалото адресно пространство от 14 Kbyte (от 8000₁₆ – B7FF₁₆) се отделя за разширение на системата чрез общосистемната ма-

гистрала. Използваният TTL PROM с обем 256×4 бита позволява адресиране до 4 зони с минимална стъпка 256 байта (декодиране на старшите 8 адресни линии), което дава възможност желаното разпределение да бъде реализирано. За нормално функциониране на системата в даден момент може да бъде избрана само една зона, което налага да има ниско логическо ниво на не повече от един изход. Съобразено с всички тези изисквания, съдържанието на PROM-а (създадения дешифратор) добива вида, показан на фиг.1.10.в. Ако е необходимо разпределението да се промени, необходимо е да се извърши друга кодировка.

1.6.3. Непълно дешифриране на адресите

Вторичният адресен дешифратор от фиг.1.3 може да бъде изграден по схемата от фиг.1.9, като просто към него се подадат други адресни линии. Нека приемем, че системният адресен дешифратор е изграден по схемата от фиг.1.10 (с TTL PROM). Тогава за вторичния дешифратор можем да използваме схемата, показана



Фиг. 1.11

на фиг.1.11а, а за картата на паметта на входно-изходните схеми (която е част от общата карта на паметта) се получава разпределението, показано на фиг.1.11б.

Периферните схеми съдържат в себе си ограничен брой регистри и използват само няколко адресни линии – напр. A0 – A2. При това положение остават адресни линии, които не участват нито при системните дешифратори, нито във вътрешните дешифратори на входно-изходните схеми. Това означава, че има състояния, които не са декодирани – така една и съща схема може да бъде избрана (активирана) на няколко различни адреса в адресното поле. В такъв случай се казва, че в дадената зона от адресното поле се извършва непълно декодиране. Това не е фатално, ако в дадената зона е разположено само едно входно-изходно устройство.

Входно-изходните схеми имат по няколко извода за избор, при което някои от тях са разрешени при подаване на високо ниво. Нека предположим, че сигналът CSIO1 се подава едновременно към два серийни адаптера, а на допълнителните изводи за избор са подадени съответно адресните линии A4 и A5. Така първата схема ще бъде селектирана (за нашия пример) от адрес $B810_{16}$, а втората – от ад-

рес $B820_{16}$. В подобни ситуации трябва да се подхожда много внимателно, защото при по-щателен анализ лесно се стига до извода, че една операция за четене от адрес $B830_{16}$ ще активира едновременно и двете периферни схеми. При това положение ще възникне колизия по данновите линии и получената информация няма да бъде вярно прочетена.

1.7. СИСТЕМИ ИНСТРУКЦИИ

Под система инструкции се разбира наборът от мнемонични кодове, обозначаващи различните операции, които изпълнява даденият микропроцесор. Това е една съществена част от инструментариума, с който разполага програмиста, за да реализира алгоритъма на програмата в разбираеми за микропроцесора машинни кодове.

1.7.1 Видове системи от инструкции

Съществуват два съвършено различни подхода при изграждането на системата от инструкции на микропроцесорите – **CISC** („Complex Instruction Set Computing“, набор от сложни инструкции за изчисления) и **RISC** („Reduced Instruction Set Computing“; изчисление с ограничен набор от инструкции).

Пълната система от инструкции – **CISC** – се характеризира със значителен брой комплексни инструкции с променлива и голяма дължина. Времето за изпълнение на отделните инструкции е значително и варира в широки граници от инструкция към инструкция. Отделните инструкции имат различни мнемонични кодове. За реализацията на различните математични операции се налага управлението да се реализира на микропрограмно ниво, което забавя изчислителния процес и налага работа с ограничено количество регистри. Днес тази архитектура се използва по-рядко поради големия набор от инструкции и по-голямата дължина на самите инструкции.

Предпочитание, особено при микроконтролерите, се отдава на системите с редуцирана система от инструкции – **RISC**. Тя е по-ефективна по отношение на обем памет, време за изпълнение, оптимизация и организация на изчислителния процес. Типичните особености на RISC микропроцесорите [3] включват:

- всички инструкции имат фиксирани формати, което опростява логическите схеми, осъществяващи тяхното декодиране и изпълнение;
- изпълнението на повечето инструкции отнема само един цикъл;
- инструкциите, включващи логически и аритметични операции, използват регистри, а операциите за зареждане и съхранение на регистрите се използват за достъп до паметта;
- Инструкциите имат прост фиксиран формат и се поддържат ограничен брой адресации;
- Микропроцесорите с **RISC** обикновено имат няколко регистъра с общо предназначение;
- **RISC** допуска лесно изграждане на конвейери при изпълнението на инструкциите, което позволява да обработват едновременно няколко инструкции. Това води до съществено увеличаване на бързодействието;

- Въвеждането на конвейери позволява извличане на следващата инструкция по време на изпълнение на текущата инструкция.

1.7.2 Типове инструкции

Независимо от наличието на два различни подхода при изграждане на системите инструкции (CISC и RISC) и огромните различия между отделните микропроцесори и фамилии микроконтролери, инструкциите могат да се категоризират в няколко функционално обособени групи [5].

Инструкции за обработка на данни

Тези инструкции трансформират данните по някакъв начин, т.е. водят до тяхната промяна. Като правило те използват аритметично-логическото устройство (или хардуерен умножител, ако има вграден такъв в микроконтролера). Към този тип инструкции спадат:

- аритметични инструкции;
- логически инструкции;
- инструкции за преместване (ротации);
- инструкции за сравняване;
- инструкции със специално назначение (напр. смяна на кода).

Инструкции за трансфер (обмен) на данни

Тези инструкции дават възможност за преместване на данните от едно място в друго (регистър-памет, памет-регистър, регистър-регистър), без да ги променят. Те могат да бъдат:

- инструкции за обмен с паметта;
- инструкции за обмен с входно-изходни устройства;
- инструкции за обмен със стековата памет;
- инструкции за вътрешен обмен (между регистрите).

Инструкции за управление на програми

Тези инструкции водят до нарушаване на последователния ход на изпълнение на програмата. Това става, като се променя принудително съдържанието на програмния брояч и управлението се предава на друг клон от програмата. Тази група инструкции включва:

- инструкции за безусловен преход;
- инструкции за условен преход (след анализ флаговете на **CR**);
- извикване на подпрограма;
- връщане от подпрограма или прекъсване;
- спиране или празна операция.

Инструкции за управление на състоянието

Тези инструкции не променят данните или хода на изпълнение на програмата. Имат по-скоро управляващи, отколкото обработващи функции. Те променят състоянието (режима на работа) на самия микропроцесор – напр. разрешаване или забраняване възприемането на прекъсвания, смяна на режима на работа (системен или потребителски) и др.

Инструкции от комбиниран тип

В микропроцесорите и микроконтролерите с редуцирана система инструкции много често се вграждат инструкции от комбиниран тип. Те позволяват в една инструкция да се извърши обработка на данни (например аритметична или логическа операция) и едновременно с това да се осъществи условен преход (проверка на флаг и преход по указан адрес). Като се вземе под внимание, че инструкциите се изпълняват за 1 или 2 такта, става очевидно предимството на такава система по отношение на бързодействието, което се достига.

1.7.3 Формати на данните

Данните, които се обработват в микропроцесорните системи, са логически или числови. Най-често се поддържат целочислени данни, които могат да бъдат без знак или със знак. Отрицателните числа със знак се представят в допълнителен код. Понякога се използват програмни библиотеки с математически функции, които поддържат (на софтуерно ниво) и работа с числа в плаваща запетая. Има и отделен клас микропроцесори, които имат вградено ALU за работа с числа с фиксирана или плаваща запетая. Тази опция е приоритетна за определен клас микропроцесори, оптимизирани за целите на цифровата обработка на сигналите. Прието е да се наричат цифрови сигнални процесори (*Digital Signal Processor – DSP*) и не са обект на разглеждане в този курс.

С появата на микроконтролерите се появиха инструкции, които позволяват операции над отделни битове. Те са особено полезни при манипулиране на отделни разреди от входно-изходните устройства. Обикновено се извършва съставна операция – четене-модифициране-запис. Тъй като тези операции касаят организацията и на интерфейсите схеми, трябва внимателно да се изучават и стриктно да се спазват предписанията на фирмите-производителки на съответните микроконтролери.

1.8. ВИДОВЕ АДРЕСАЦИИ

Методите за адресация касаят методите за формиране на адреса на операндите – определяне на ***изпълнимия адрес IA***. Тук отново съществува разнообразие на решенията, които се прилагат. Докато има микропроцесори, които поддържат само няколко вида адресации (напр. при MC68HC11 те са само 5), при други количеството им става значително (при MC68020 достигат 18). Докато при CISC архитектурите тенденцията е техният брой да расте, при микропроцесорите с RISC система инструкции стремежът е да се поддържа по-малък брой адресации. Насъществува известна нееднозначност и объркване от факта, че различните производители решават проблемите с формиране на адресите по сходни (но не еднакви) начини, но в описанията използват различни термини за видовете адресации. Независимо от това може да се направи опит за обобщение на отделните адресации, като се търси сходство за начина за получаване на данните.

1.8.1. Непосредствена адресация

Често срещан случай е, когато в даден регистър трябва да се зареди ***константа*** или такава да се използва в дадена логическа или аритметична операция.

Характерното в случая е, че тези данни не се променят – еднакви са при всяко изпълнение на програмата. В този случай константата се генерира в процеса на създаване на програмата и се съдържа в машинния код на инструкцията или непосредствено след него, но в програмната памет. Този вид адресация се поддържа от почти всички микропроцесорни системи и се нарича **непосредствена адресация**.

1.8.2. Регистрова адресация

Това също е много често срещан случай, когато операцията се извършва върху **данни**, които вече се намират **в определен регистър** (получени в резултат на логическа или аритметична операция или предварително заредени там). Известна е като **регистрова адресация**. В някои микропроцесорни системи със CISC система инструкции (напр. MC6809, MC68HC11) името на регистъра се съдържа в мнемоничния код на инструкцията и оператора се подразбира, затова там този вид се нарича **адресация по подразбиране**.

1.8.3. Абсолютна или разширена адресация

В този случай данните за операцията (операндът) се намират в данновата памет, а адресът **IA**, на който се намира операндът, е записан в машинния код или непосредствено след него в програмната памет. Това е стандартния начин за достъп до променливи, които се намират в оперативната памет или до съдържанието на регистрите на входно-изходните устройства. Подмножество на този вид е **директната адресация** (поддържана от MC6800, MC6809, MC68HC11 и др.), при които в паметта се поддържа зона по подразбиране с ускорен достъп, при която старшата част от адреса не се задава от инструкцията, а се добавя автоматично от микропроцесора при адресирането на клетката. Така се намалява обемът на програмната памет и се ускорява изпълнението на програмата.

1.8.4. Индексна адресация

Тази адресация дава възможност за последователен достъп до данни, записани в даден масив. Началният адрес на масива се зарежда в определен вътрешен регистър на микропроцесора, който изпълнява ролята на **индексен регистър**. Изпълнимият адрес **IA** се получава, като към съдържанието на индексния регистър се добави **отместването**, съдържащо се в машинния код или непосредствено след него **в програмната памет**. Обхождането на масива става чрез промяна съдържанието на индексния регистър, а не на отместването.

1.8.5. Индексна адресация с пре- декремент или инкремент

Идентична като формиране на изпълним адрес и достъп до данните с разглежданата вече индексна адресация. За разлика от нея, преди изпълнението на описаните операции се извършва автоматично намаляване (декремент) или увеличаване (инкремент) на индексния регистър с 1 или 2, в зависимост дали се обменя байт или два байта.

1.8.6. Индексна адресация с пост- декремент или инкремент

Подобна е на индексната адресация с пре- декремент или инкремент, но модификацията на индексния регистър се извършва не преди, а след изпълнението на операцията с индексна адресация.

Тези две адресации дават възможност за организиране на по-бърз и ефективен достъп и сканиране на масиви с данни. Наличието на тези адресации позволяват лесно да бъдат организирани потребителски стекове.

1.8.7. Индиректна адресация

При нея съдържанието на указан регистър се явява адрес от паметта (**IA**), където се намира операндът (данните) за текущата операция. Подобна е на индексната адресация с тази разлика, че при нея няма отместване и не се налага изчисление на изпълнимия адрес.

1.8.8. Относителна адресация

Отново се извършва индексен достъп с отместване, като в случая като индексен регистър се използва програмният брояч, а отместването е зададено в машинния код на инструкцията или след него, отново в програмната памет. Особеност е, че най-често байтът за отместването се възприема като число със знак. Това дава възможност полученият изпълним адрес да бъде както по-голям, така и по-малък, от текущата стойност на програмния брояч. Най-често тази адресация се използва при инструкциите за условен преход. Разгледаната особеност дава възможност да се правят преходи както напред, така и назад в програмата.

В различните микропроцесори се срещат още разновидности за адресация, при които изпълнимият адрес се получава чрез изчислителни операции на съдържанието на отделни регистри, полета от програмния код и дори съдържание на клетки от паметта за данни. Те са добре разяснени във фирмените описания на съответните микропроцесори и няма да бъдат разглеждани подробно тук.

1.9. РЕЖИМИ НА РАБОТА С ПОНИЖЕНА КОНСУМАЦИЯ

Може би най-голямо развитие от възникването си до днес микропроцесорните системи са претърпели именно по понижаване на консумираната мощност, което се изразява в няколко порядъка (повече, отколкото е при повишаване на тактовата честота и бързодействието). Този въпрос намира сериозно развитие именно с появата и повсеместното използване на микроконтролерите. Днес все повече системи работят в екстремни условия от автономни захранващи източници, като консумацията им е ограничена на десетки или дори пада до единици микроампери.

Няколко са факторите, спомогнали за силното намаление на консумацията. Някои от тях са технологични и конструкторите могат само да го приемат за даденост, като например преминаването към CMOS технология, при което силно се редуцира консумираният ток, особено в статичен режим. Други са комплекс от схемотехнични решения и програмно управление на режимите на работа и условията на експлоатация, които силно зависят от решенията на конструкторския екип. Именно в тази посока съществуват възможности за изявяване на преимуществата на отделни конструкторски решения.

1.9.1 Управление на работната честота

На първо място сериозно влияние върху консумацията оказва работната честота. Тук изискванията са противоречиви – от една страна, с цел повишаване производителността и бързодействието на системите, стремежът е ядрата на микроконтролерите да работят на по-високи тактови честоти. От друга страна, това неминуемо води до повишаване на общата консумация. Могат да се търсят компромисни решения, при които да се търси понижаване на консумацията при запазване на добра производителност.

Съществуват микроконтролери, в които освен вградения RC генератор (с ниска стабилност на честотата), има още по два генератора с кварцова стабилизация, а ядрото може да се превключва да работи със сигналите от единия или другия генератор. Обикновено единият от тях работи на десетки мегахерци, а другият – най-често на 32768 KHz (2^{15} Hz). Така, когато е необходимо високо бързодействие, системата работи на по-високата честота, а през останалото време – на по-ниската, при което консумацията рязко спада. Отговорност на програмното осигуряване е да подбере моментите за превключване на режимите – на високо бързодействие или на редуцирано електропотребление.

В най-новите фамилии микроконтролери има вградени генератори, сфазирани с основния кварцов генератор чрез затворени фазо-честотни вериги. Те позволяват по програмен път честотата им да се променя в много широки граници (най-често повишаване), като запазват дълговременна стабилност, съизмерима с тази на основния генератор. Това позволява алгоритмично да се избира такава работна честота, която да гарантира поддържане на необходимото бързодействие при оптимална консумация, съобразно текущите изисквания към управляващата микропроцесорна система.

1.9.2 Управление на захранващото напрежение

Знае се, че CMOS интегралните схеми могат да работят в сравнително широк диапазон на захранващите напрежения (при стандартните логически схеми този диапазон може да варира от 2 до 18 V). При микроконтролерите този диапазон е по-тесен (обикновено от 3,3V – 5V). Това също е възможност за редуциране на консумацията – чрез понижаване на захранващото напрежение. Този метод трябва да се избягва или да се прилага много внимателно, тъй като редуцирането на консумацията не е голямо, а това може да доведе до недобро съгласуване с външните за микроконтролера интегрални схеми. От друга страна промяната на захранването ще доведе и до промяна на праговете на сработване на цифровите схеми и влошаване на запаса от шумоустойчивост [9].

1.9.3 Управление на набора от активирани ресурси в системата

В повечето микроконтролери има възможност да се управлява включването и изключването на захранването към отделните ресурси и вградени периферни схеми. Естествено че не е необходимо да се държат активирани ресурси, които не се използват (напр. аналогово-цифров преобразувател). От друга страна би било неправилно в името на незначително намаляване на консумацията да се отказваме от някои периферни подсистеми, които имат пряко отношение към стабилната и

надеждна работа на системата – напр. стражеви таймери, подсистеми за следене на хранващото напрежение или системния тактов генератор и др. В конкретните фирмени описания на отделните микроконтролери обикновено има данни кои подсистеми каква енергия консумират в активен режим при номинална работна честота.

1.9.4 Спиране на ядрото на микроконтролера и периферни схеми

Най-често редуциране на консумацията на микроконтролера се постига, като се спре постъпване на тактова честота към ядрото му (микропроцесорната част). Връщането от това състояние става автоматично при постъпване на някое прекъсване. За целта би могло и да се инициализира някой таймер да „събуди” микропроцесора. Трябва да се има предвид, че работата на даден броячен блок води да повишаване на консумацията.

Наред с всички изброени мерки за управление на енергопотреблението в една система, трябва да се държи сметка и за външно присъединени схеми и устройства. Днес микроконтролерите позволяват от цифровите изходи да се консумира значителен ток – до 20 – 25 mA. В много случаи той е напълно достатъчен да се хранят някои сензорни системи – напр. MEMS. В такива случаи трябва да се помни, че след хранване на съответния интерфейс е необходимо да се изчака определено време, за да се установи нормалният му режим на работа.

Често, за да се намали шумът и да се повиши точността на аналогово-цифровото преобразуване, докато работи АЦП се изключва подаването на честота към ядрото на микроконтролера. При приключване на преобразуването АЦП генерира прекъсване и микропроцесорът „се събужда”, за да прочете данните от преобразуването.

Методите за редуциране на консумацията при съвременните микроконтролери са много ефективни, но трябва да се прилагат много внимателно, за да не повлияят на точността и функционалните способности на вградените интерфейсни схеми.

ГЛАВА 2. СЕРИЙНИ ИНТЕРФЕЙСИ В МИКРОПРОЦЕСОРНИТЕ СИСТЕМИ

2.1. АСИНХРОННИ СЕРИЙНИ ИНТЕРФЕЙСИ

В практиката често се налага да се реализира обмен на данни между микрокомпютри и управляващи или измерващи микропроцесорни системи. За целта са разработени и стандартизирани множество интерфейсни подсистеми, което позволява да се свързват разнотипни устройства, произведени от различни производители. За успешното реализиране на тази връзка, се налага интерфейсите устройства да бъдат стандартизирани (уеднаквени) на три нива:

– **физическо ниво** – включва описание на вида, време-последователностите и електрическите нива на сигналите за връзка;

– **механично ниво** – описва механичните характеристики на куплунгите за връзка в точките на присъединяване, както и характеристиките на преносната среда ;

– **логическо ниво** – описва протоколите за обмен и форматите на данните.

Най-общо интерфейсите за връзка могат да се разделят на **паралелни** и **сериенни**. По-широко приложение намират серийните интерфейси поради лесното реализиране на канала за връзка.

При серийните интерфейси данните се преобразуват в сериен (последователен) вид, като отделните разряди се предават в последователни и най-често равни по продължителност моменти от време (това твърдение не винаги е вярно, вж. напр. MFM). Прилагат се различни способности за **кодиране**, **синхронизиране** и **валидиране** (проверки за грешки) при обмена на данни. Това е довело да възникване на голямо разнообразие от стандарти за преобразуване и обмен на данни в сериен формат.

От своя страна серийните интерфейси могат да се разделят на **синхронни** и **асинхронни**. При **асинхронните** интерфейси в предаващата и приемащата страна има отделни тактови генератори, които не са синхронизирани – това предполага, че процесите на предаване и приемане трябва често да **се синхронизират**, обикновено това става на всеки байт. При **синхронните** серийни интерфейси или се използва един тактов генератор за двете страни, или има предвидена схема, която позволява **двата генератора да се синхронизират**.

Всяка система за обмен на данни е необходимо да включва:

– източник на данни (предавател или линеен формирова̀тел), който преобразува сигналите във форма, удобна за предаване по каналите за връзка;

– приемник, който приема сигналите, постъпващи по каналите за връзка и ги преобразува отново във изходния формат;

– канал за връзка, по който се предават тези сигнали. Това може да бъде меден проводник, оптово̀лакна или радиоканал.

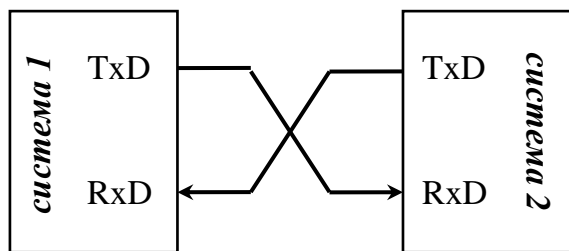
За да бъде обменът на данни успешен, е необходимо приемникът и предавателят да работят с еднотипни данни. Това се постига, като се дефинират следните параметри:

- тип на използваните сигнали;
- определяне на нивата на логическите „единици“ и логическите „нули“;
- кодовете, чрез които се представят предаваните символи;
- осъществяване на синхронизация между приемащата и предаващата част;
- управление на потоците от данни, за да не се претовари приемната част;
- механизъм за откриване и коригиране на грешки, възникващи при обмена на данни.

2.2.1. Асинхронен сериен интерфейс SCI

Един от най-често използваните **асинхронни** серийни интерфейси е **сериенният комуникационен интерфейс**. Той се е наложил поради опростения начин на кодиране на данните, лесна схемотехнична реализация и надеждност на работа. Характеризира се с умерени скорости на обмен и възможност за откриване на единични грешки.

Интерфейсът позволява работа в пълен дуплексен режим (едновременно двупосочно предаване на данни). Разработен е за осигуряване на връзка между две



Фиг. 2.1

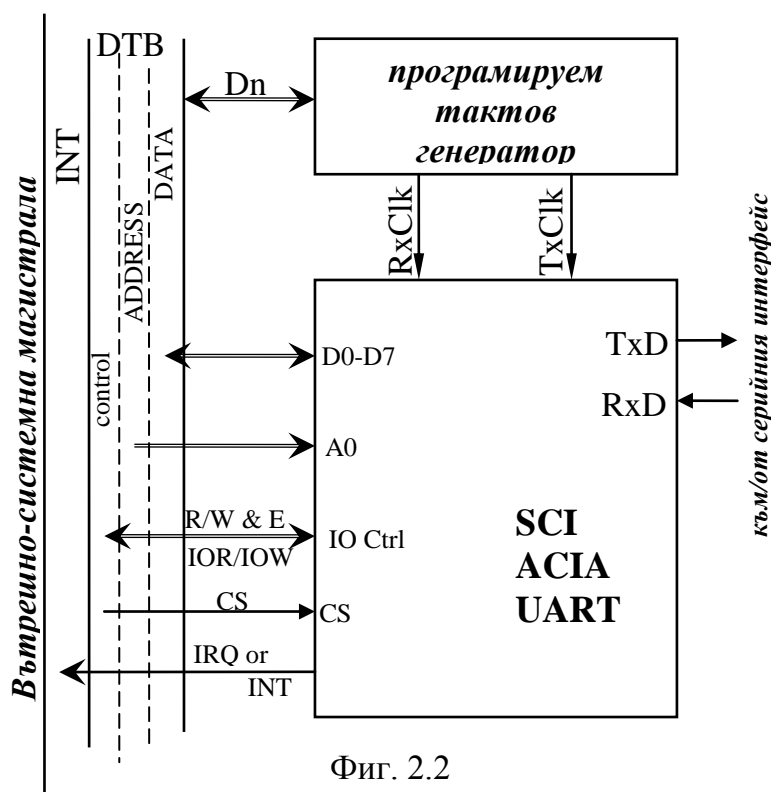
крайни устройства по двупроводна линия (фиг. 2.1), като всяко от тях поддържа предавател и приемник, които са свързани насрещно противоположно.

За преобразуване на данните от паралелен в сериен формат се използват специализирани схеми, наречени контролери или адаптери. В специализираната литература са известни като UART (универсален асинхронен

приемо-предавател) или ACIA (асинхронен комуникационен интерфейсен адаптер). В почти всеки съвременен микроконтролер има вградена периферна схема, изпълняваща тази функция, която обикновено се означава като UART или SCI.

- **Място и връзка към микропроцесорната система, сигнали**

Мястото на UART (ACIA) в една микропроцесорна система се вижда от фиг.1.3, а примерната връзка към вътрешно-системната магистрала се илюстрира от фиг.2.2. Обменът на данни към микропроцесора се осъществява посредством подсистемата за обмен на данни (DTB) на вътрешносистемната магистрала чрез данновите линии под управлението на приетия протокол за управляващите сигнали. Данните са в паралелен формат и най-често са с байтова организация. При завършване на предаване или приемане на байт може да се разреши генериране на прекъсване към микропроцесора.



Фиг. 2.2

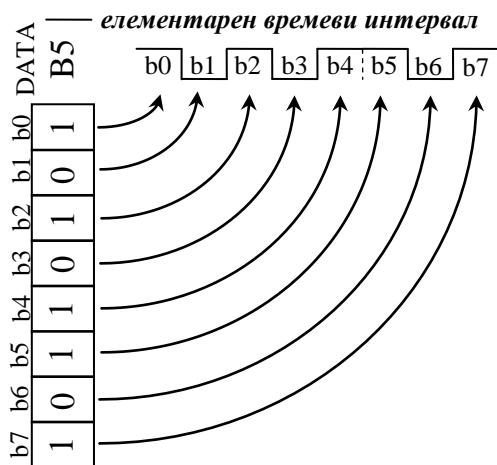
Тактовият генератор обикновено е един и същ при предаване и приемане. Неговата работна честота може да бъде задавана при конфигуриране на системата (чрез установяване на подходяща комбинация) или програмно задавана в процеса на работа (при наличие на управляващ регистър).

Към външния интерфейс има поне два сигнала. Единият е изхода на предаващата част и обикновено е наречен TxD. Входът на приемащата част обикновено се означава като RxD. Някои от

интерфейсните адаптери поддържат и допълнителни протоколни сигнали, за част от които ще стане дума по-късно.

• Скорост на предаване

В интерфейския адаптер става преобразуване на данните от паралелен в сериен формат, като всеки бит се пропуска към изхода на схемата за точно определен интервал от време. Реципрочната стойност на това време има измерение на честота. Прието е да се нарича скорост на предаване на данни (bit rate) и се измерва с битове за секунда, напр. 1200 bit/sec. Понякога се използва и понятието *bod rate*, което в разглеждания от нас случай съвпада с bit rate (това не е вярно в общия случай и зависи от метода за кодиране на сигналите). Наложиха се няколко стандартни скорости на предаване, които са отразени Таблица 2.1.



Фиг. 2.3

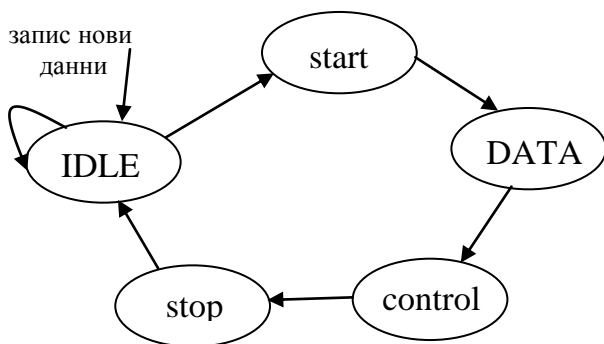
• Кодиране на сигнала

Прието при преобразуване на данните от паралелен в сериен формат първо да се предават младшите значещи разряди, а последен – старшият разряд, както е показано на фиг.2.3. При това лог.“0“ се кодира, като се предава ниско логическо ниво, а лог.“1“ се кодират с високо логическо ниво. На фигурата е изобразено как би изглеждал байт със съдържание B5₍₁₆₎ като се преобразува в сериен формат. Прието е този начин на кодиране да се означава като NRZ (not return to

zero) – предаване без връщане към нулата. Характерно за него е, че изискванията към ширината на честотната лента на канала за връзка са по-големи (сравнете формата на сигнала при данни $00_{(16)}$ или $FF_{(16)}$ спрямо $55_{(16)}$ или $AA_{(16)}$ например). От друга страна, кодирането на сигнала схемотехнично се реализира по-лесно, а възприемането му е интуитивно.

• **Състояния (фази) на интерфейса**

По време на предаване на байт итерфейсът се намира в 5 ясно различими и точно дефинирани състояния (фази) – **състояние на изчакване (idle)**, **начало на предаването (start)**, **предаване на данните (data)**, предаване на **бит за контрол (ctrl)** и **край на предаването (stop)**, както е илюстрирано на фиг.2.4.



Фиг. 2.4

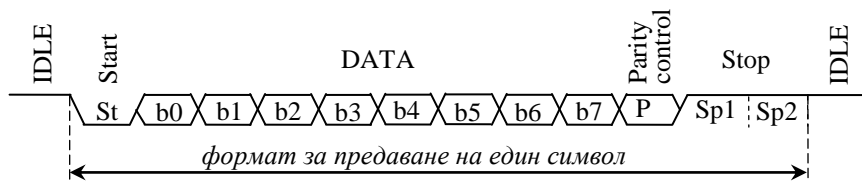
Предназначението и действията в отделните фази се свеждат до:

а) **състояние на изчакване (idle)** - неактивно състояние, при което се изчаква зареждането на байт за предаване. Това е основното състояние, в който се намира интерфейса, докато не се предават данни. Кодира се, като се предава високо логическо ниво. Време-

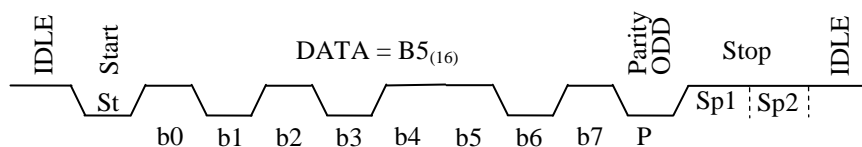
последователността, илюстрираща формата на предаваните данни в съответствия с дефинираните договорености, е показана на фиг.2.5а;

б) **начало на предаването (start)** – това е фазата, в която се извършва синхронизацията на приемащата страна спрямо предаващата. При постъпване на данни за предаване за времето на един елементарен интервал от време t_0 сигналът се установява на ниско логическо ниво. Това е указание за приемащата страна, че от следващия времеинтервал системата ще бъде в състояние на обмен на данни.

в) **предаване на данните (data)** – това е същинската фаза на обмен на данни. Тук се поместват разрядите на предаваните данни, кодирани в съответствие с метода NRZ, както беше изложено по-горе. Трябва да се отбележи, че стандартът



Фиг. 2.5а



Фиг. 2.5б

позволява да се предават 5, 6, 7 или 8 бита, като форматът се договаря преди началото на обмен на данни. Форматите с 5 или 6 бита са използвани в миналото при телеграфните апарати и са останали като архаична възможност, те дори не се поддържат от повечето

интерфейсни адаптери. Днес всички интерфейсни схеми поддържат 7 и 8 битови формати, като по-широко приложение намира 8 битовия формат.

г) **предаване на бит за контрол (ctrl)** – тази възможност е заложена с цел да се откриват навреме грешки, възникнали по време на обмена. Изразява се в добавянето в поредицата на още един времеви интервал, по време на който се предава бит за контрол. Той се формира на базата на предадените (и съответно приети!) информационни битове и допълва техния брой до четен или нечетен – говори се за контрол по **четност** или по **нечетност**. Прилаганата форма за контрол също се договаря преди началото на обмена на данни.

Допустимо е при обмен да не се извършва контрол на верността на предадените битове – казва се, използваният формат е без контрол по четност. Тогава този времеинтервал просто се пропуска, а времето за предаване на думата се съкращава с един времеинтервал (вж. табл.2.1).

Следва да се отбележи, че надеждността на прилагания метод за контрол не трябва да се абсолютизира. Той ще сработи, само ако са възникнали нечетен брой грешки (независимо дали е по четност или по нечетност!).

На фиг. 2.5б е показан конкретен случай, при който се предава като данни $B5_{(16)}$ (примерът, разгледан по-горе при кодирането на данните), при формат 8 бита с използване на контрол по нечетност и два стопови бита.

В серийните интерфейсни контролери, вграждани в микроконтролерите в последно време, често се предвижда възможността на мястото на бита за контрол по четност да се използва предаването на т.н. **адресен бит**. Този механизъм дава възможност за сериозна оптимизация на организацията на програмното осигуряване в системи, обединени в нискоскоростни локално мрежи, както ще бъде разгледано по-късно.

д) **край на предаването (stop)**

Приключването на процеса на предаване на една дума става чрез предаване на „стопов“ бит – задържане на изхода на предавателя на високо логическо ниво за един, един и половина или два времеинтервала. След това интерфейсът отново преминава в състояние на изчакване (idle).

• **Формати на данните**

За да се осъществи успешен обмен е необходимо:

- предаващата и приемащата страна да поддържат една и съща скорост на обмен;
- да се работи с еднаква дължина на данните;
- да се работи с един и същи контрол – по четност, по нечетност или без контрол;
- броят на стоповите битове да е еднакъв.

Тези параметри се уговарят предварително и задават на устройствата, преди те да бъдат стартирани за съвместна работа. Например уговорка от вида 9600/8N2 означава, че ще се работи на скорост 9600 бит/сек, 8 информационни бита, без контрол и два стопови бита. Означението 19200/7E2 говори за скорост 19200 бит/сек, 7 информационни бита, контрол по четност (even) и два стопови бита, респективно 4800/8O1 – скорост 4800 бит/сек, 8 информационни бита, контрол по нечетност и един стопов бит. Ако тези условия за еднаквост не са спазени, в

примната страна много бързо ще възникне форматна грешка или грешка от контрол по четност.

В таблица 2.1 са посочени най-често използваните скорости на обмен, продължителността за предаване на един разряд, както и скоростите за предаване на един символ при използване на различни формати.

Таблица 2.1

№	Скорост за предаване, [bit/sec]	1 такт, [μs]	Време за предаване на 1 символ, [ms]							
			7N2	7O1 7E1	7O2 7E2	8N1	8N2	8O1 8E1	8O2 8E2	
1	150	6666	66,66	66,66	73,326	66,66	73,326	73,326	79,992	
2	300	3333	33,33	33,33	36,663	33,33	36,663	36,663	39,996	
3	600	1666	16,66	16,66	18,326	16,66	18,326	18,326	19,992	
4	1200	833	8,33	8,33	9,163	8,33	9,163	9,163	9,996	
5	2400	416	4,16	4,16	4,576	4,16	4,576	4,576	4,992	
6	4800	208	2,08	2,08	2,288	2,08	2,288	2,288	2,496	
7	9600	104	1,04	1,04	1,144	1,04	1,144	1,144	1,248	
8	19200	52	0,52	0,52	0,572	0,52	0,572	0,572	0,624	
9	38400	26	0,26	0,26	0,286	0,26	0,286	0,286	0,312	
10	57600	17,36	0,174	0,174	0,191	0,174	0,191	0,191	0,208	
11	115200	8,68	0,087	0,087	0,095	0,087	0,095	0,095	0,104	

• **вътрешна структура на примерен серийен интерфейс**

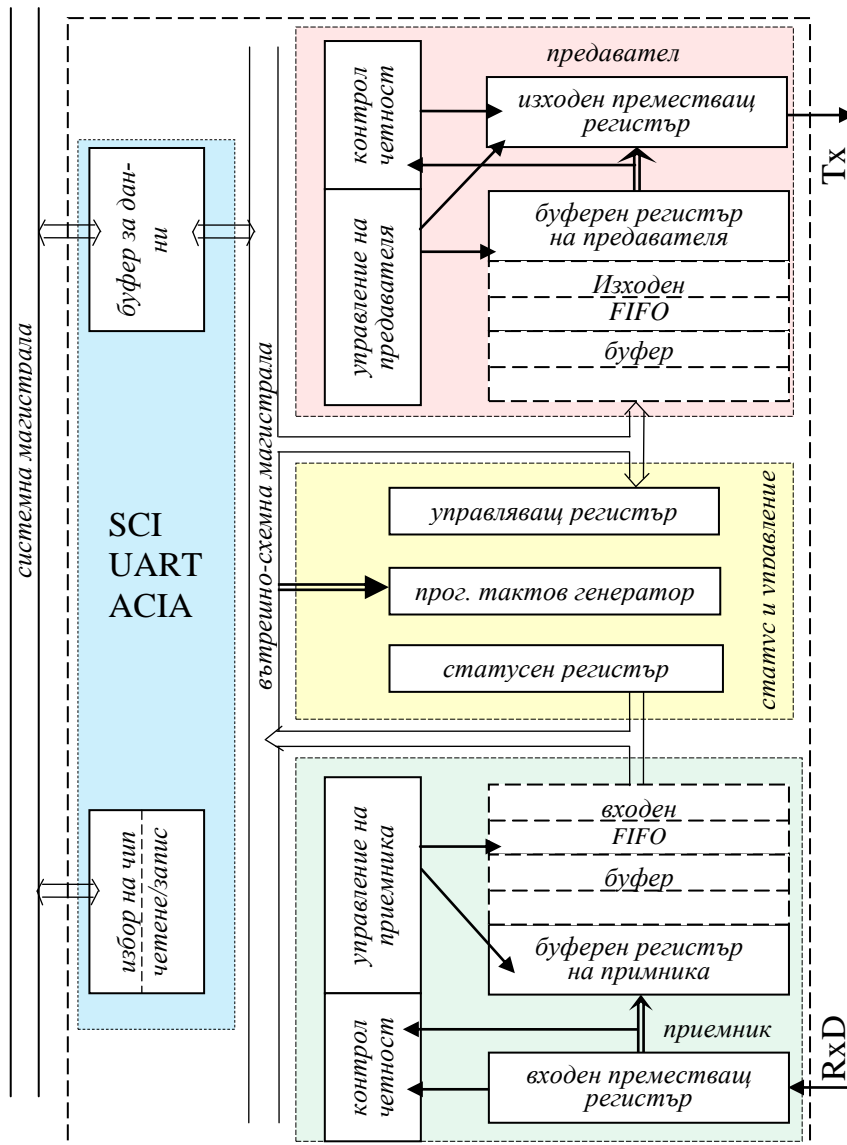
Вътрешната структура както на използваните в миналото серийни интерфейсни адаптери, така и на вгражданите в съвременните микроконтролери асинхронни серийни интерфейси, е много разнообразна. Независимо от това съществуват определени общи белези на структурно ниво, които дават възможност да бъде обобщена една структурна схема, която носи в себе си общите черти на това многообразие.

Една такава структурна схема на примерен асинхронен серийен интерфейсен модул е показана на фиг. 2.6. Вижда се, че е изграден от 4 основни функционални блока – предавател, приемник, контролно-управляваща част и блок за връзка със системата. Последният блок съдържа двупосочен буфер за данни и логическа схема за избор на чип и управление посоката на обмена. Ако се касае за асинхронен серийен интерфейс, вграден в микроконтролер, данновият буфер обикновено липсва, а логиката за избор и управление на обмена е интегрирана в общата управляваща логика на микроконтролера.

Управляващият регистър дава възможност да се преконфигурират схемите на предавателя и приемника така, че да могат за се избират различни формати на данните за обмен. За целта, преди да започне обменът, се извършва инициализиране на режимите на работа чрез запис на подходящи кодови комбинации в него. Това обикновено се свежда до задаване на броя на информационните битове в символа (5, 6, 7 или 8), типа на контрол (по четност, по нечетност или без контрол), брой на стоповите битове (при предаване и при приемане), разрешаване генериране на прекъсване при постъпил символ (в приемника) или празен буферен регистър за данни (в предавателя) и др. Конкретното предназначение на битовете

в управляващия регистър е подробно описано във фирменото описание за конкретния интерфейс адаптер.

Чрез проверка съдържанието на статусния регистър може да бъде проследя-



Фиг. 2.6

вано състоянието на интерфейсния адаптер – приети данни, готовност за предаване на нови данни, възникнала грешка от контрол по четност или форматна грешка и др. Проследявайки това състояние, драйверната програма може да осъществява ефективен обмен на данни, с възможност за откриване на възникнали грешки.

Предавателната част съдържа изходен преместващ регистър и буферен регистър на предавателя. Понякога се изгражда FIFO буфер, което позволява да бъде зареден цял низ от данни, които да бъдат предадени последователно във времето без допълнителна намеса от страна на микропроцесора (ядрото на микроконтролера). В предавателя влиза и генераторът за контрол по четност/нечетност, както и логическата схема за управление на преобразуването в сериен формат в съответствие със заданието в управляващия регистър.

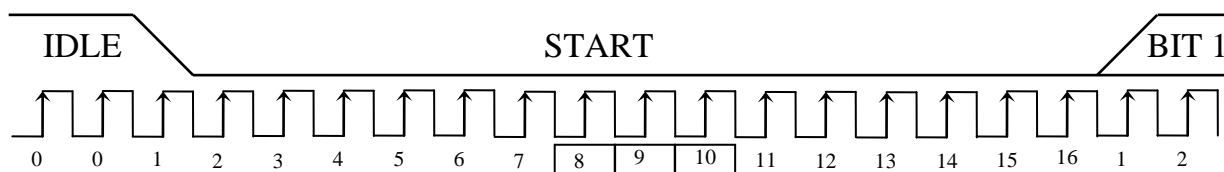
Приемната част отново съдържа преместващ регистър, който тук преобразува постъпващите входни данни от сериен в паралелен вид. Те се записват във входния буферен регистър (или във входния FIFO буфер, ако такъв присъства в адаптера). Преобразуването се извършва под управлението на логическата схема в приемника. Едновременно с това се извършва за контрол по четност/нечетност на постъпващите данни (в схемата за контрол по четност) в съответствие с формата, зададен в управляващия регистър. Резултатът от извършения контрол се отразява в статусния регистър.

Програмируем тактов генератор може да не присъства в интерфейския адаптер, а да е необходим външен такъв. В този случай обикновено съществуват два отделни тактови входа – по един за предавателната и един за приемната част. Това дава възможност скоростите за предаване и приемане да се различават. Обикновено при избора на режим на работа на интерфейсната схема се инициализира и коефициента на кратност на тактовия генератор. Необходимо е честотата, която той изработва, да бъде равна или по-голяма на скоростта на предаване. Най-често се предпочита тази честота да бъде 16 пъти по-голяма, а понякога се използва да бъде 64 по висока от скоростта на предаване. Това няма много голямо значение за работата на предаващата част, но съществено се отразява на стабилността на работата на приемника.

В предавателя квантоването по време (изместване с един разряд на преместващия регистър) става синхронно с подаваната тактова честота – на всеки входен импулс, на всеки 16 или всеки 64. Като се знае желаната скорост на предаване и избраната кратност при конкретния режим, лесно може да се определи каква трябва да бъде честотата на генератора.

При приемника въпросът стои малко по-остро, тъй като постъпващата кодова последователност и сигналът от собствения генератор не са синхронизирани. За осигуряване на надежден обмен на данни е необходимо семплирането (възприемането) на отделните битове да става около средата на елементарния времеви интервал, в който се предава съответният информационен или контролен бит. При коефициент на кратност 1 (когато честотата на генератора е равна на скоростта на обмен) на практика това изискване няма как да бъде гарантирано, затова този режим се прилага само в случаите, когато се работи с един и същи тактов генератор за двете страни – предавателна и приемна. Във всички останали случаи в приемната страна се работи с коефициент на кратност 16 (16x) и по рядко – с 64x.

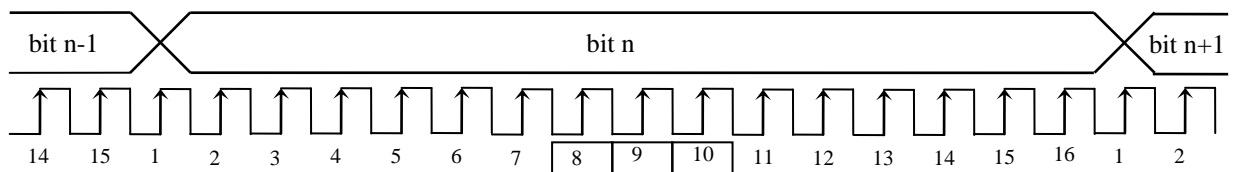
Синхронизирането на приемната част по входния сигнал при режим с делене на 16 е илюстрирано на фиг. 2.7. В режим на изчакване (IDLE) при всеки такт на тактовия генератор се проверява нивото на входния сигнал. Падащият фронт на бита START се използва като точка за синхронизация – начало на поредица от 16



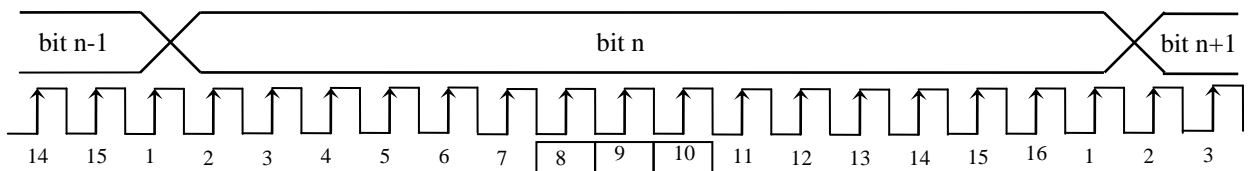
Фиг. 2.7

такта, по време на които се формира стартовия бит. При 8, 9 или 10 такт (варира за различните интерфейсни адаптери) отново се инспектира входната линия. При ниско ниво на сигнала се приема, че е открит действителен стартов бит, в противен случай събитието се интерпретира като шум и се следи за постъпване на действителен старт.

Веднъж определена средата на стартовия бит, може да се семплира (стробира) всеки следващ информационен или контролен бит след 16 периода на тактовата поредица – т.е по време на 8-ия период на съответните 16 такта за дадения разряд (вж. фиг. 2.8.a). Прецизно извършената синхронизация на приемната страна спрямо предаващата дава възможност семплирането на всеки бит в поредицата да



Фиг. 2.8.a



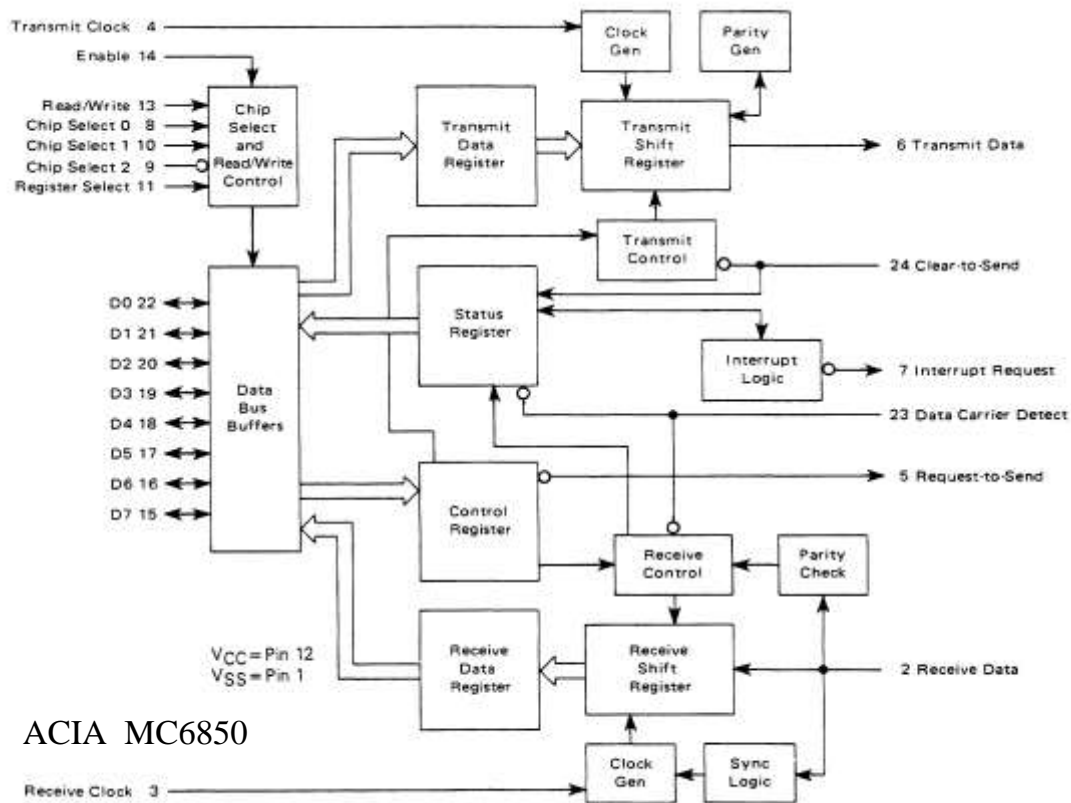
Фиг. 2.8.б

се извършва в средата на времеви интервал, когато той се предава. Така преходните процеси по време на смяна на нивата ще са приключили и вероятността за възникване на грешки силно намалява.

Не бива да се забравя, че асинхронният режим на работа предполага, че предаващата и приемащата страна използват отделни тактови генератори. Естествено е тактовите честоти на тези генератори да се различават в някаква степен. Фиг. 2.8.б илюстрира случая, в който честотата на генератора в приемната страна е по-висока, отколкото на този в предавателя. Веднъж точно определена средата на стартовия бит, с всеки следващ приет информационен бит ще се натрупва едно изместване при определяне момента за стробирането му. Явно е, че това отместване ще бъде най-голямо при приемане на стоповия бит. За правилна комуникация е необходимо позицията на стробиране на последния предаван бит (STOP бит) да бъде във времеинтервала, когато се предава този бит. В противен случай приемникът не може да открие условията за STOP и ще бъде генерирана форматна грешка. Изхождайки от горните съображения, лесно може да се изчисли (как? Направете необходимите изчисления за различните формати!), че за осъществяване на успешен обмен е необходимо разликата в скоростта на комуникация на приемника и предавателя трябва да е по-малка от 4%.

• **Примери на асинхронни адаптери**

На фиг. 2.9 е показана вътрешната структурна схема на асинхронен комуникационен интерфейс адаптер – АСІА МС6850. Както се вижда, той съдържа

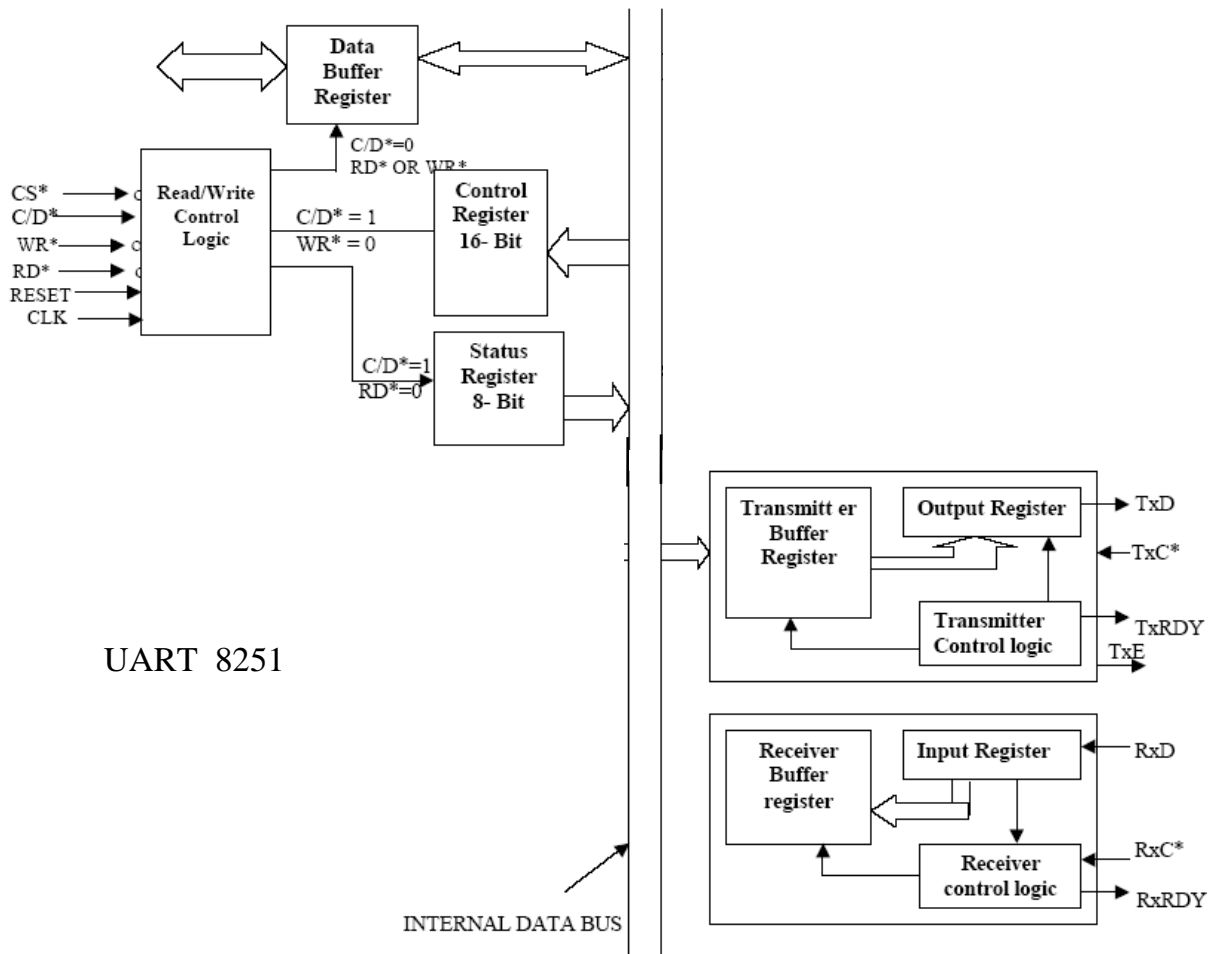


Фиг. 2.9

главните структурно-определящи блокове, които бяха коментирани по-горе:

- **предавател**, включващ *Transmit Data Register*, *Transmit Shift Register*, *Transmit Control* и *Parity Gen*;
- **приемник**, включващ *Receive Data Register*, *Receive Shift Register*, *Receive Control* и *Parity Check*;
- **контролно-управляващата част** включва *Status Register*, *Control Register* и *Interrupt Logic*. Следва да се отбележи, че тази асинхронна интерфейсна система поддържа и три протоколни сигнала – **RTS** (*Request to Send*), **CTS** (*Clear to Send*) и **DCD** (*Data Carrier Detect*). Схемата може да генерира заявка за прекъсване при получаване на символ или празен регистър на предавателя към управляващата микропроцесорна система;
- **блок за връзка със системата** – включва *Data Bus Buffers*, както и *Chip Select* и *Read/Write Control*. Управляващите сигнали са ориентирани за лесна връзка с вътрешно-системни магистрали, изградени на базата на 8-разрядни микропроцесори на фирмата Motorola.

Структурата на друг универсален асинхронен приемо-предавател – UART 8251 е показана на фиг. 2.10. Тук отново се виждат вече разглежданите структу-



Фиг. 2.10

роопределящи елементи. Блокът за връзка към вътрешносистемната магистрала е оптимизиран за съвместна работа с 8 разрядните микроконтролери на Intel.

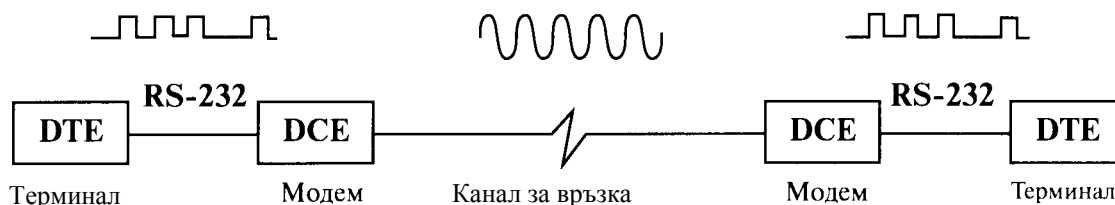
Подобно структура имат и асинхронните серийни комуникационни интерфейсни схеми, които се вграждат в почти всички микроконтролери. Нещо повече, става тенденция в много контролери да има по два такива интерфейса.

Изключение е комуникацията да се осъществи, като се свържат направо сигналите от серийните асинхронни адаптери или микроконтролери. За да се повиши разстоянието между устройствата, както и шумоустойчивостта и надеждността на комуникацията, се използват различни буфериращи схеми. Това е довело до възникване на няколко комуникационни стандарта [10], базирани на използването на разгледания серийен комуникационен интерфейс.

2.2.2. Асинхронен интерфейс RS232

Комуникационният интерфейс RS232 (**R**ecommended **S**tandards 232) е създаден през далечната 1969 г. (две години преди първият микропроцесор!) и намира широко приложение и до днес в разнообразните системи с микропроцесори и микроконтролери. Той има за цел да определи и уеднакви електрическите и механични характеристики на връзката между терминални устройства (data terminal equip-

ment - DTE) и апаратурите за предаване на данни (data communication equipment - DCE), при които се използва последователен двоичен обмен на данни. Фиг. 2.11 илюстрира мястото му в комуникационната система. По-късно този интерфейс е приспособен за работа с най-разнообразни устройства и системи – компютри,



Фиг. 2.11

принтери, програмируеми логически контролери, системи за събиране на данни, измервателни прибори и др.

На скоростта на обмен на данни сериозно ограничение оказват следните фактори, наложени основно от комуникационната линия и обкръжаващата среда:

- ширина на честотната лента;
- наличие на електромагнитни смущения (отношението сигнал/шум);
- пропускателната способност (зависи от съотношението на информационните битове към служебните такива в рамките на един символ);
- честотата на възникване на грешки.

Трябва да се подчертае, че RS-232 дефинира само електрическите и механични параметри на интерфейса, без да ограничава използваните протоколи за връзка.

Първата и основна отличителна характеристика на разглеждания интерфейс е, че е предназначен за осъществяване на двупосочен обмен на данни **само между две устройства**. Досега интерфейсът е претърпял множество изменения и допълнения, като най-широко приложение намира версията RS-232C, основните характерни особености на която е залегнала в изложението по-долу.

• **Режими на предаване на сигналите**

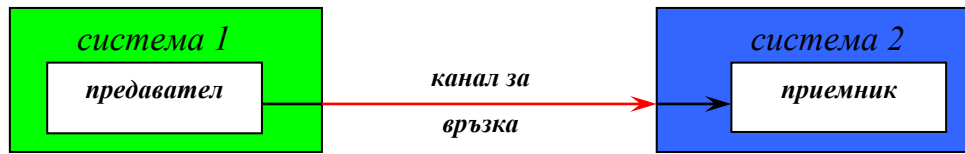
През даден комуникационен канал за връзка между две устройства обменът може да се осъществява при използване на един от следните три режима:

- симплексен режим;
- полудуплексен режим;
- пълен дуплексен режим.

В симплексен режим предаване на данни се осъществява само в една посока (фиг.2.12а). Явно е, че в този случай е необходимо да има един предавател и един приемник, както и само една комуникационна линия.

В полудуплексен режим данните могат да се предават в двете направления, но това не може да става едновременно. В този случай се използва един канал за връзка, който се използва последователно от двойките предавател/приемник на двете устройства, изграждащи комуникацията (фиг.2.12б) – т.е. системата работи в режим на времеделение.

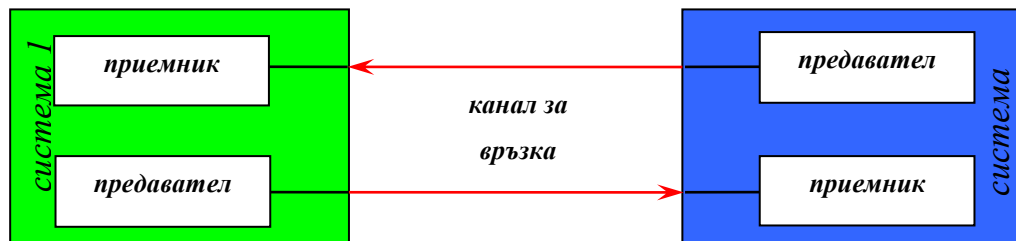
При пълен дуплексен режим предаването на данни може да се осъществява в двете посоки в един и същи момент от време. Явно е, че отново в двете устройства трябва да има и предавател, и приемник (фиг.2.12в). Необходимо е също да има



Фиг. 2.12а Симплексен режим на предаване на данни



Фиг. 2.12б Полудуплексен режим на предаване на данни



Фиг. 2.12в дуплексен режим на предаване на данни

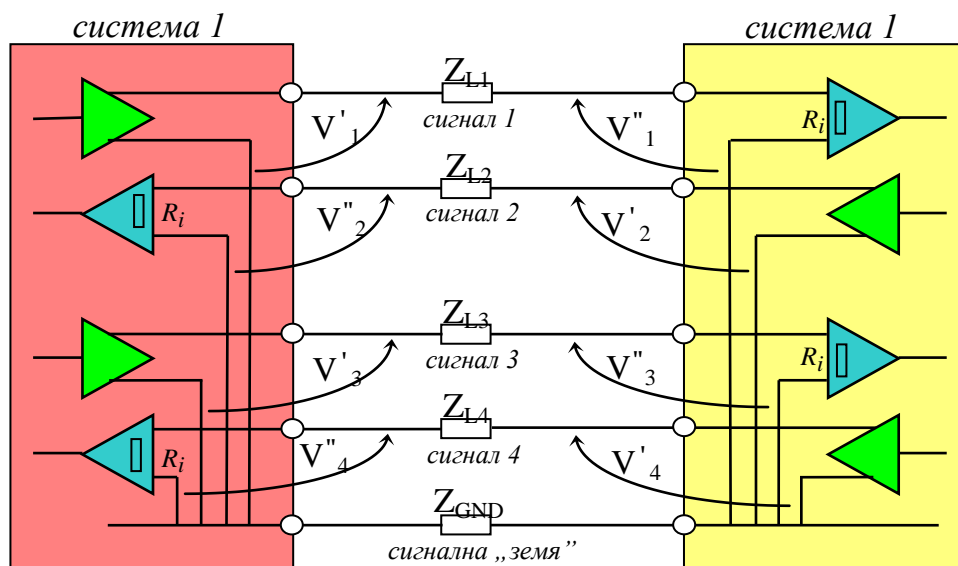
две комуникационни линии (или да се използват различни честотни ленти на един комуникационен канал.

- **Използване на небалансирана (несиметрична) комуникационна линия;**

Стандартът RS232 предвижда обменът на данни да се извършва по небалансирана (несиметрична) комуникационна линия, като се използват двуполярни напрежителни сигнали.

Небалансирана или несиметрична е такава комуникационна линия, при която сигналът се предават по един проводник. Напрежението на този сигнален проводник се определя спрямо друг проводник, който е общ за интерфейсната система и обикновено се нарича сигнална „земя“ (фиг. 2.13).

При анализ на токовете в системата се вижда, че през сигналната земя в различни моменти протичат разнопосочни токове, обусловени от моментните състояния на сигналите на отделните изходи на интерфейса. Общият проводник има характеристики (съпротивление, индуктивност, капацитет), идентични с тези на останалите проводници. Тези токове поражда паразитни напрежения върху отделните малки (но не нулеви!) импеданси по пътя на сигналната земя, които се сумират или изваждат от сигналите, понижавайки много силно шумоустойчивостта на системата.



Фиг. 2.13

Ако токовете в сигналните вериги не са големи (в случаите на високо входно съпротивление на приемниците) и общият проводник се характеризира с нисък импеданс, тогава паразитните напрежения биха имали незначителни стойности и влиянието на възникващите смущения ще бъде незначително. От друга страна сигналните контури, които се получават, могат да обхващат по-съществени потоци от индуцирани външни (смущаващи) електромагнитни полета. В комбинация с високото входно съпротивление на приемниците отново може да се достигне до неприемливи съотношения сигнал/шум.

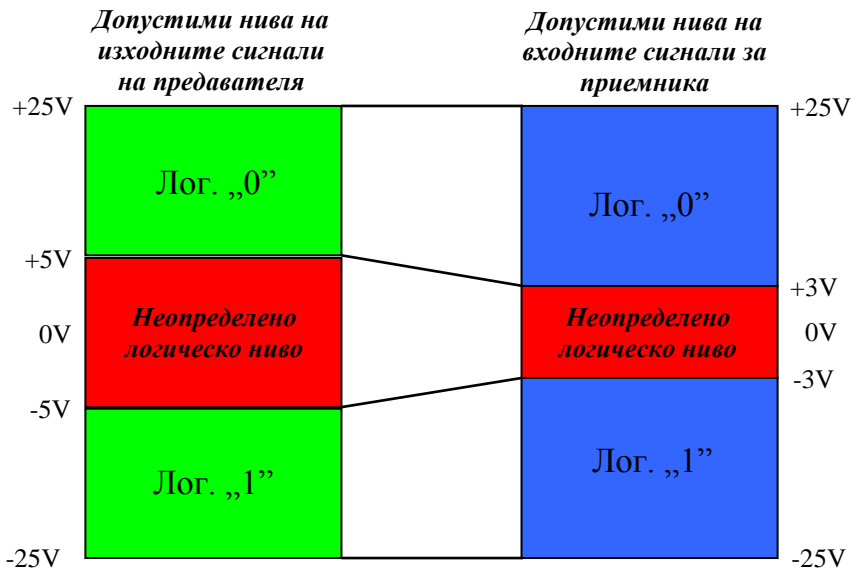
Теоретически обменът на данни може да се осъществи ако сигналните токове са малки и същевременно общият проводник има нисък импеданс. На практика тези изисквания налагат подобни интерфейси да работят само при къси линии за връзка, т.е. на неголеми разстояния.

Екранирането на комуникационните кабели силно намалява индуцираните електромагнитни смущения в системата. Не трябва да се забравя, че заземяването на екрана от двете страни на кабела може да доведе до по-лош резултат (ако се допусне по екрана да текат утечни токове), затова се препоръчва той да се свързва само от едната страна.

Независимо от изложените по-горе съображения, несиметричните комуникационни линии намират широко приложение на къси дистанции поради лесното изпълнение както на самата линия, така и на съответните приемници и предаватели, които се използват за формиране на сигналите.

- **Електрически спецификации на интерфейса RS232**

Електрическите спецификации включват изисквания към нивата и полярността на основните и допълнителните сигнали, използвани в RS232C. Дефинирани са различни по ширина допускови зони за предавателите и приемниците (фиг. 2.14), с което се гарантира възможност за комуникация при възникване на промяна на амплитудата на сигналите от изхода на предавателя до входа на приемника с оглед специфичните особености на комуникационната линия.



Фиг. 2.14

Предавателите, които се използват за буфериране на изходните сигнали за RS232C трябва да осигуряват изходно напрежение в диапазона:

- логическа „1“: от -5 V до -25 V;
- логическа „0“: от +5 V до +25 V;
- неопределено логическо ниво: от +5 V до —5 V.

Зоните, в които приемниците трябва да възприемат вярно входните логически нива, са дефинирани в малко по-широк обхват от напрежения, като се цели да се минимизира влиянието на промяна на сигнала вследствие реалните характеристики на линиите за връзка (вж. фиг. 2.13):

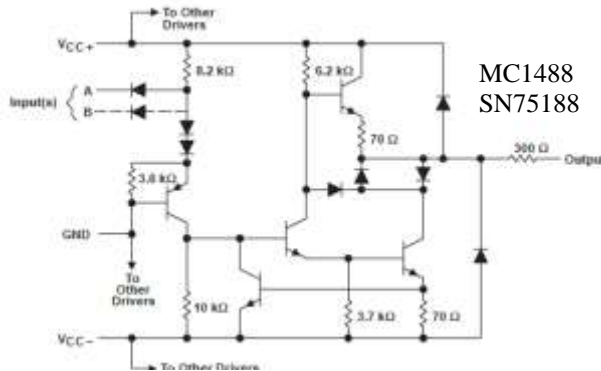
- логическа „1“: от -3 V до -25 V;
- логическа „0“: от +3 V до +25 V;
- неопределено логическо ниво: от +3 V до -3 V.

Вижда се, че дефинираните напрежителни нива са нествойствени за съвременните микропроцесорни системи. За целта са разработени специализирани интегрални схеми, които се използват като приемници и предаватели за работа с интерфейса RS232C.

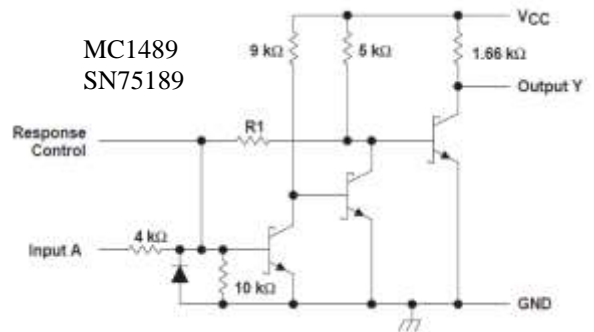
Едни от първите схеми, използвани като предаватели за RS232C, са SN75188 и MC1488. Те съдържат 4 линейни предавателя, които се захранват с $\pm 12V$ до $\pm 15V$, а входовете им са съвместими с TTL и CMOS логически нива. Логическата функция, която реализират, е „отрицание“, което дава възможност директно да бъдат управлявани от изходите TxD на SCI, UART, ACIA и др. Изходните сигнали удовлетворяват изискванията на стандарта RS232C. Не трябва да се забравя, че последователно на изхода има вграден резистор със стойност 300Ω , а изходният им ток е ограничен до 10 mA. Вътрешната схема според Texas Instruments на тези буфери е показана на фиг. 2.15а.

Като входни буфери могат да се използват схемите SN75189 и MC1489. Те също инвертират входния сигнал и ограничават изходните сигнали до стандартни TTL и CMOS нива, което позволява да се свързват към входа RxD на SCI, UART и

АСІА. Захранват се еднополярно с +5V и имат входно съпротивление около 4 КΩ. Принципната вътрешна схема на тези входни буфери е показана на фиг. 2.15б.

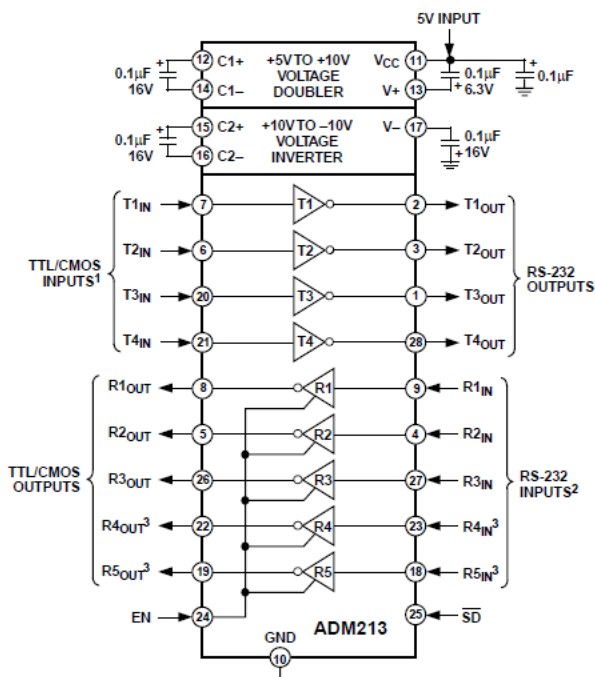


Фиг. 2.15а

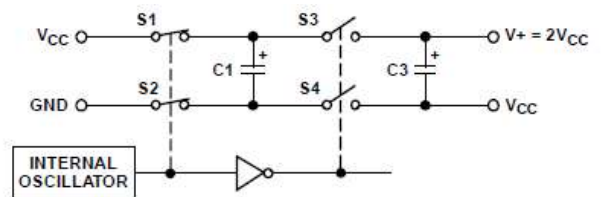


фиг. 2.15б

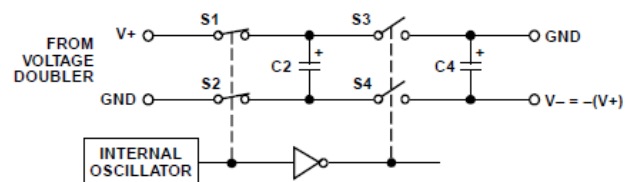
В последствие са разработени и вече широко се използват ново поколение буферни схеми като напр. серията МАХ232-239, АDМ202-213 и още много други. Характерното при тях е, че обединяват по няколко приемника и няколко предавателя, което позволява с една схема да се буферират всички сигнали от интерфейсната система. Те следват обща структурна схема (фиг. 2.16), като разликите се



Фиг. 2.16



Фиг. 2.17а



Фиг. 2.17б

свеждат до броя на включените предаватели и приемници. Друга съществена особеност е, че се захранват само с едно захранващо напрежение 5V. Благодарение на вградените кондензаторни преобразуватели (т.н. „зарядни помпи“) се получава удвояване на захранващото напрежение (фиг. 2.17а), а по аналогичен начин от него се получава и отрицателно напрежение (фиг. 2.17б). При това изходните нива на сигналите от тези буфери имат стойност $\pm 2V_{CC}$ – т.е. около $\pm 10V$, което удовлетворява изискванията на стандарта (вж. фиг. 2.14). С въвеждането на тези схеми много се облекчава използването на интерфейса RS232C в малки измервателни системи, програмируеми логически контролери, системи за управление и натруп-

ване на данни, защото отпада необходимостта от вграждане на захранващи блокове, поддържащи по-голям набор от захранващи напрежения.

- **Сигнали на интерфейса RS232C;**

Сигналите, които са дефинирани от стандарта RS232C условно могат да бъдат разделени на две основни групи – основни, или комуникационни сигнали, и допълнителни, или протоколни сигнали. Първата група винаги присъстват и чрез тях се осъществява комуникацията на данни в съответствие с приетите конвенции. Втората група са спомагателни сигнали, управляват и улесняват протокола на комуникация и не е задължително използването им (или на някои от тях).

Към **основните** комуникационни **сигнали** се отнасят:

- **Предавани данни – TxD, transmitted data, Output;**

По тази комуникационна линия се предават серийните данни от DTE към съответния вход на DCE устройство. По време на паузата (фаза IDLE) на тази линия се поддържа отрицателен потенциал с ниво между -5V и -25V.

- **Приемани данни – RxD, receive data, Input;**

По тази комуникационна линия постъпват за приемане в сериен формат обменяните данни в съответствие с утвърдените спецификации.

- **Общ проводник - сигнална „земя“, Signal Ground;**

Това е общият проводник, спрямо който се предават всички напрежителни сигнали – както комуникационните, така и протоколните. Тя е затваряща за всички контури – по нея протича сумарен ток, съответстващ на токовете във всички сигнални линии (вж. фиг.2.13). Тези изводи от интерфейса е необходимо винаги да бъдат свързани с проводник с нисък импеданс.

Тези три сигнални линии **задължително** присъстват, за да се осъществи обмен на данни по интерфейса.

Протоколни са следните сигнали и тяхното предназначение е:

- **Заявка за предаване на данни – RTS, Request To Send, Output;**

Този сигнал се използва от DTE устройствата за изпращане на запитване за разрешение за предаване на данни. Активното ниво е лог. „0“ (+V).

- **Готовност за изпращане на данни - CTS, Clear To Send, Input;**

Този входен сигнал към DTE се използва, за да укаже, че комуникационното устройство (DCE, модем) е готово да приема данни. Активното му логическо състояние е „0“ - +V.

Тази двойка сигнали управлява хардуерно потока от данни. Обикновено се използва, когато се реализира полу-дуплексен режим на обмен – така DTE и DCE се редуват кое от тях ще предава данни в дадения момент (т.е. в коя посока ще се предават данните по комуникационната линия). Ако е започвал обменът на символ е необходимо той да приключи и чак тогава се минава в състояние на изчакване. При работа в пълен дуплексен режим този протокол няма нужда да се поддържа – тогава и двете устройства във всеки момент могат да предават и приемат данни и сигналите от двете страни трябва да са в активно състояние (лог. „0“). Обикновено се поддържат програмно, което прави тяхното присъствие незадължително.

- **Готовност на терминала – DTR, Data Terminal Ready, Output;**

Активното състояние на сигнала (лог. „0“ - +V) изразява готовност на терминала за обмен на данни. Използва се предимно при работа с модем.

– **Готовност на приемника – DSR, Data Set Ready, *Input***;

Този сигнал е индикация, че устройство тип DCE има готовност за предаване на данни (модемът е свързан към коноло за приемане на данни). Активното му състояние е лог. „0“ (+V).

Тази двойка сигнали също не е задължителна и се поддържа на програмно ниво. Когато се използва, обикновено се поддържа активна за времето, през което се обменя блок от данни.

– **открит носител на данни – DCD, Carrier Detected, *Input***;

Този сигнал се използва изключително при работа с модем или радиомодем. Указва, че е открита носещата модулациона честота, чрез която се осъществява предаването на данни чрез преносната среда. Активен е през цялото време на обмяна. Незадължителен сигнал, поддържа се програмно.

– **Индикатор за позвъняване – RI, Ring Indicator, *Input***;

Сигналът се използва само при работа с модеми, свързани към телефонна линия. Активирането му указва, че е постъпило повикване по линията и е активен, само докато трае повикването.

– **Защитна „земя“ (екран) – Shield Ground**

Това е екраниращата оплетка на свързващия кабел. Не трябва да се забравя, че свързването и към двете устройства може да доведе до възникване на паразитни контури (съвместно със сигналната „земя“), които да доведат до появата на смущения и нестабилна работа на устройствата. В този случай се препоръчва свързването на екрана само към едното устройство.

• **Изисквания към канала за връзка;**

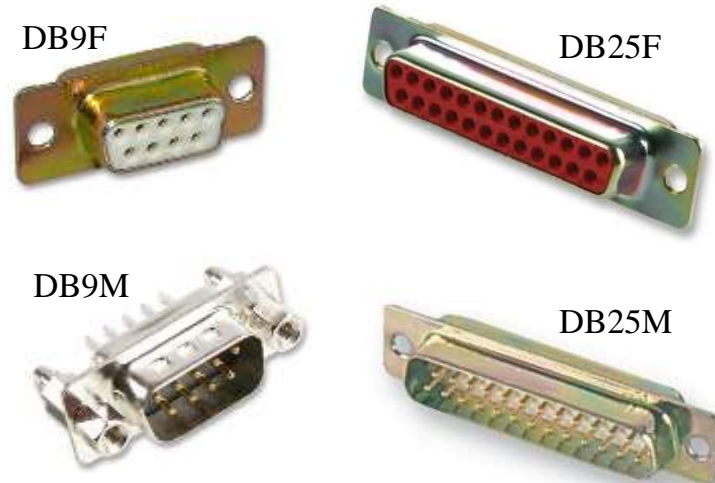
Скорост, бит/сек	Дължина, m
110	850
300	800
600	700
1200	500
2400	200
4800	100
9600	70
19200	50
115 K	20

Допустимата скорост на предаване по RS232C е ограничена от качествата и дължината на свързващия кабел. Формата на сигналите (продължителността на техните фронтове и преходните процеси) в много голяма степен зависи от индуктивността и капацитета на проводниците. Съгласно предписанията на стандарта капацитетът на кабела не трябва да бъде по-голям от 2500 pF. При съвременните кабели специфичният им капацитет е от порядъка на 50 pF/m до 160 pF/m, което ограничава препоръчваната дължина в границите на 15 до 50 m.

Проведени са полеви изпитания за определяне на максималната дължина на кабела, при която се извършва успешен обмен на данни, при различни скорости на предаване. Резултатите [10] са показани в табл. 2.2 и те потвърждават, че данни на по-големи разстояния могат да се предават при по-ниски скорости.

- **Механични характеристики, куплунги за връзка;**

Предписанията на стандарта са за използване на 25 изводни куплунги тип DB25 – щифтови (DB25M) за устройства тип DTE и гнеzdови (DB25F) за устройства тип DCE. В последствие се допуска използване на 9 изводни куплунги тип DB9 и днес те намират много широко приложение. Външният вид на упоменатите съединители е показан на фиг. 2.18.



Фиг. 2.18

Стандартът дава предписания и за местоположението на конкретните сигнали по изводите на съединителите. Така се гарантира безпроблемно свързване за

Разписание сигналите на DTE				
извод № на		име на сигнал	предназначение на сигнала	посока на сигнала
DB25M	DB9M			
1	shield	SG	Shield Ground	
2	3	TxD	Transmitted Data	output
3	2	RxD	Receive Data	input
4	7	RTS	Request To Send	output
5	8	CTS	Clear To Send	input
6	6	DSR	Data Set Ready	input
7	5	GND	Signal Ground	
8	1	DCD	Data Carrier Detect	input
20	4	DTR	Data Terminal Ready	output
22	9	RI	Ring Indicator	input

успешна съвместна работа на устройства, произведени от различни производители. Разписанието на тези сигнали за съединители тип DB25M и DB9M (за устройства от терминален тип – DTE) е показано в таблица 2.3. Аналогично е и разписанието и за устройства тип DCE, но се използват гнеzdови съединители DB25F и DB9F и, естествено при тях вече входовете са изходи и респ. изходите са входо-

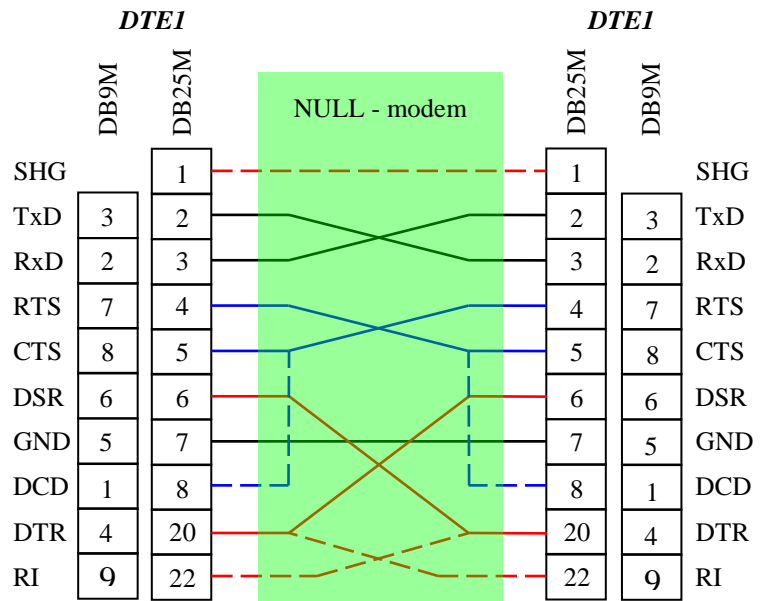
ве. Това дава възможност да се използват кабели, при които едноименните изводи са свързани. Не трябва да се забравя, че свързването на екрана на кабела едновременно към двете страни може да породи проблеми с шумоустойчивостта.

- **NULL-modem;**

Преобладаващо интерфейсът RS232C се използва за осъществяване на обмен на данни между микрокомпютри, измервателни системи, системи за събиране на данни, програмируеми контролери ... Все по-рядко се използват класически модеми за осъществяване на връзка, при които преносната среда е телефонна линия (или друга форма на проводна връзка). Това налага извода, че интерфейсът се е видоизменил и вече в преобладаващия случай се налага да комуникират помежду

си две еднотипни устройства – DTE с DTE, без посредничеството на DCE. Това се постига лесно, като само се спазват ограниченията, наложени от свързващите кабели, а комуникационните сигнали (и протоколните - където е необходимо), се свързват „на кръст“ TxD с RxD, RTS с CTS и DTR с DSR.

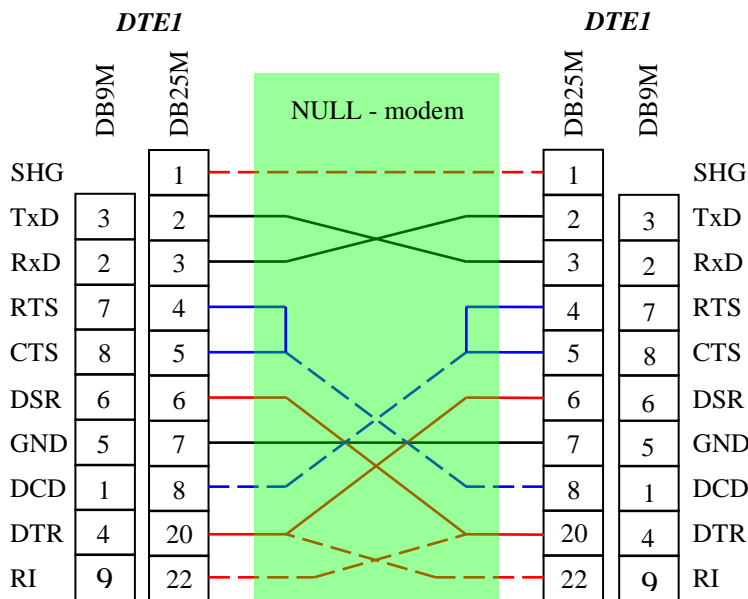
Подобна връзка е показана на фиг. 2.19а. Задължителни сигнали са сигналната „земя“, TxD и RxD (показани със зелен цвят на фигурата). Необходи-



Фиг. 2.19а

мостта от останалите сигнали (връзките са означени със син и червен цвят) се диктува от особеностите на използваните комуникационни програми и вградено програмно осигуряване в микропроцесорните системи. Ако програмното осигуря-

ване изисква поддръжане и на сигналите DCD и RI може да се осъществи връзка към RTS и DTR на насрещната система, както е показано с пунктирани линии.



Фиг. 2.19б

Понякога се използва и алтернативен вариант (фиг 2.19б), при който сигналите RTS и CTS на всяка от системите са свързани накръсо. Тази необходимост може да възникне в случай, че една от драйверните програми не поддържа хардуерния протокол, а другата го изисква – така се избягва ограничението и връз-

ката може да се осъществи.

• **Галванично разделяне на интерфейса**

Много често се налага да се осъществи комуникация между две микропроцесорни системи (микрокомпютри, системи за събиране на данни, измервателни устройства с микрокомпютри и др.), при което е недопустимо осъществяване на галванична връзка между тях. Нещо повече – по-добре е да се реализира галванично разделяне на интерфейса винаги, когато това е възможно. Това е почти задължително, когато се работи в промишлени или полеви условия – така на първо

място се защитава оборудването и при възникване на токови удари щетите могат да се предотвратят или силно да се ограничат. Освен това се редуцират условията за прехвърляне на смущения между отделните системи. Ефекта на галваничното разделяне най-силно се проявява при системи, при които се извършва аналогово-цифрово или цифрово-аналогово преобразуване – в резултат на редуциране на шумовете съотношението сигнал/шум може значително да се повиши.

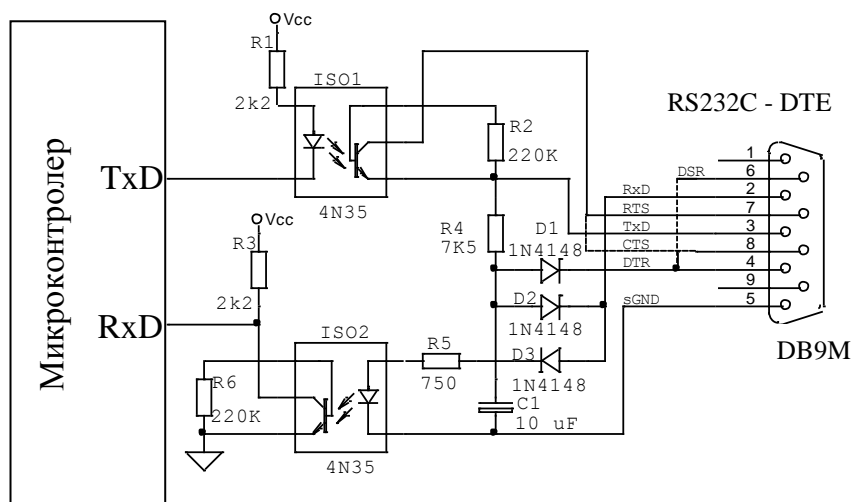
В повечето случаи за реализиране на комуникацията е достатъчно да се обезпечи прехвърлянето само на данните линии (RxD и TxD), без да се поддържат допълнителните протоколни сигнали. По-долу ще бъдат разгледани именно тези редуцирани варианти за осъществяване на серийна връзка с прилагане на галванично разделяне.

На практика могат да възникнат няколко ситуации при реализиране на галванично разделяне на RS232C, при което:

- Връзката и за двете системи се осъществява непосредствено след асинхронните интерфейсни адаптери, където сигналите са на TTL или CMOS ниво. Това е най-простия за реализиране вариант, като същевременно е най-отдалечен от стандарта RS232C, затова няма да бъде подробно разглеждан.

- Вторият вариант предполага да се изгради връзка между стандартен интерфейс RS232C и асинхронен серийен интерфейс, където нивата на сигналите са CMOS или TTL, като се обезпечи галванично разделяне на двете системи. Естествено, може да се приложи галванично разделяне с помощта на оптрони и след тях са се постави буферна схема от типа MAX232-239 или ADM202-213. В този случай ще възникне въпросът за осигуряване на галванично разделен токозахранващ източник за частта след оптроните.

Едно възможно решение, при което се избягва този проблем, е показано на фиг. 2.20. При него не се поставя буферна схема и се разчита оптронът да се зах-



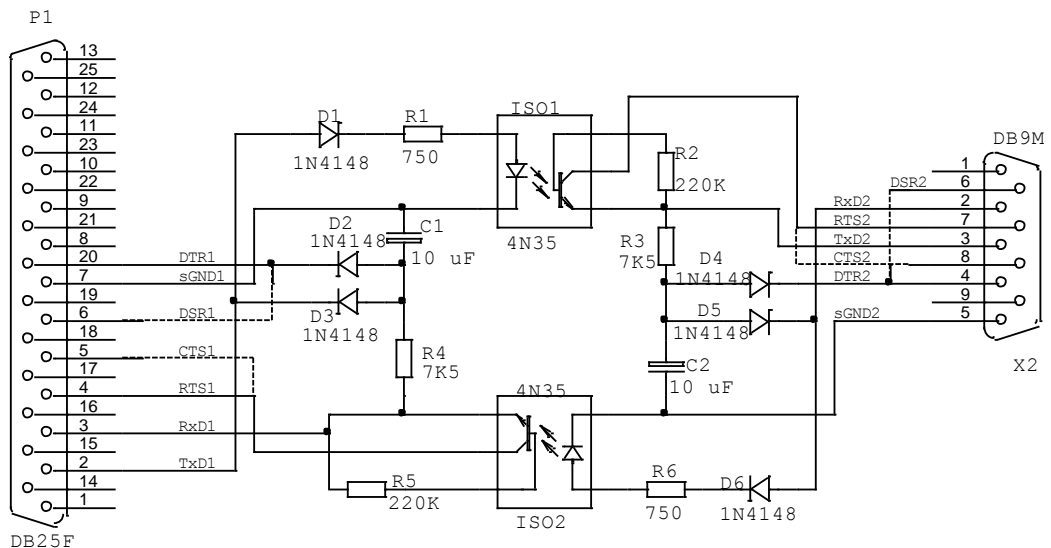
Фиг. 2.20

рани от нивата на допълнителните протоколните сигнали. То е приложимо в случаите, когато връзката се изгражда към интерфейсна система RS232C, която поддържа поне сигналите RTS и CTS. Положително захранващо напрежение за галванично разделената част се получава от сигнала RTS на приемащата страна, кой-

то предварително трябва да се установи на ниско логическо ниво, което се кодира с +V. По сериозен е въпросът с осигуряване на отрицателно захранващо напрежение. Би могъл да се използва сигналът DTR, който трябва да се инициализира на високо логическо ниво . -V. В този случай могат да възникнат проблеми от хардуерен (ако този сигнал не се поддържа) или от софтуерен характер (това състояние може да се възприема като липса на готовност за обмен). Тогава като резервен вариант отрицателното напрежение може да се получи, като се използва нивото на сигнала TxD, което се поддържа по време на състоянието IDLE на данновата линия (лог. „1“, кодирана с -V). За целта е необходимо този сигнал да се раздели с диод и нивото да се фиксира с помощта на неголям електролитен кондензатор, както е показано на фигурата. В приемният канал, реализиран с оптрона ISO2, няма схемотехнични особености, защото приемната страна (от към микроконтролера), която изисква захранване, го получава направо от вградената система.

Схема е подходяща за вграждане в неголеми системи с микроконтролери, които се налага да комуникират с микрокомпютри или други терминални устройства, при което дължината на свързващия кабел не превишава няколко метра. Трябва да се държи сметка и за бързодействието на използваните опторони, т.к. поради увеличеното време за излизане от насищане на транзисторите, се получава удължаване на импулсите, което може да доведе до грешно възстановяване на сигнала. Добре настроена, схемата осигурява скорости на предаване до 9600 бит/сек.

– Третият вариант възниква, когато трябва да се приложи галванично разделяне между два интерфейса RS232C. В този случай може да се използва схемата, показана на фиг. 2.21. Както се вижда, тя е подобна на тази, представена на предната фигура. Изказаните по-горе съображения, които касаят интерфейса RS232C, важат и тук, при това и за двете страни. Към използването на тази схема трябва да се подхожда много внимателно. Предимството е, че двете системи са



Фиг. 2.21

напълно галванично изолирани и защитени от възникване на непредвидени и нежелани аварийни ситуации.

- **Заклучение**

Не трябва да се забравя, че първоначално интерфейсът RS232C е разработен за връзка между терминал и модем. В последствие придобива много широко приложение и днес, поради лесното изграждане, се използва по-често за връзка между микрокомпютри и разнообразни измервателни или управляващи системи на базата на микроконтролери, често в промишлени или полеви условия. Естествено се проявяват някои ограничения, възникващи в следствие на изискванията на новите условия, при които намира приложение. По-важните от тях са:

- Ограничава възможното количество на присъединените устройства до две. С широкото навлизане на „интелигентни“ измервателни устройства това ограничение създава пречки при тяхното присъединяване към системите;
- Изискванията за нива на сигналите трудно се съгласува с възможностите на захранващите устройства на съвременните вградените системи;
- Осигурява връзка на къси разстояния, което представлява сериозно ограничение при много приложения;
- Допустимата скорост на обмен в много случаи не позволява да бъдат предадени потоците от данни, които възникват;
- Постигнатата шумоустойчивост често е недостатъчна при работа в промишлени или полеви условия.

Гореизброените ограничения са довели до множество други стандарти за реализиране на серийна комуникация, в които те са преодолени.

2.2.3. *Асинхронен интерфейс RS422/RS485*

Асинхронният интерфейс RS485, подобно на RS232C, отново стъпва на възможностите, които предлага асинхронният серийен комуникационен интерфейс (SCI, UART, ACIA), като ги разширява с нови функционални възможности – в случая, позволява да се осъществи връзка между повече от две устройства. Така са създадени възможности за изграждане на малка компютърна мрежа. RS485 се явява развитие на стандарта RS422, запазвайки основните принципи и разширявайки възможностите му по отношение броя на абонатите, затова ще бъдат разгледани неговите характеристики и особености.

Стандартът RS485 е ориентиран към двупроводна или четирипроводна „многоточкова“ връзка и гарантира сигурен обмен на данни при следните условия:

- максимално разстояние между крайните точки – 1200 m;
- максимална скорост на предаване на данни – 10 Mbit/s;
- включване до 32 линейни формирателя (предавателя);
- включване до 32 линейни приемника.

При необходимост от присъединяване на повече от 32 устройства е необходимо да се използват повторители, които не са дефинирани от стандарта.

Комуникационният интерфейс RS485 е привлекателен с простотата си и с възможността, която предоставя, за свързване на множество устройства в паралел. Надеждността при предаване на данни и ниската цена го правят предпочитан при изграждане на разпределени микропроцесорни системи и малки промишлени мрежи.

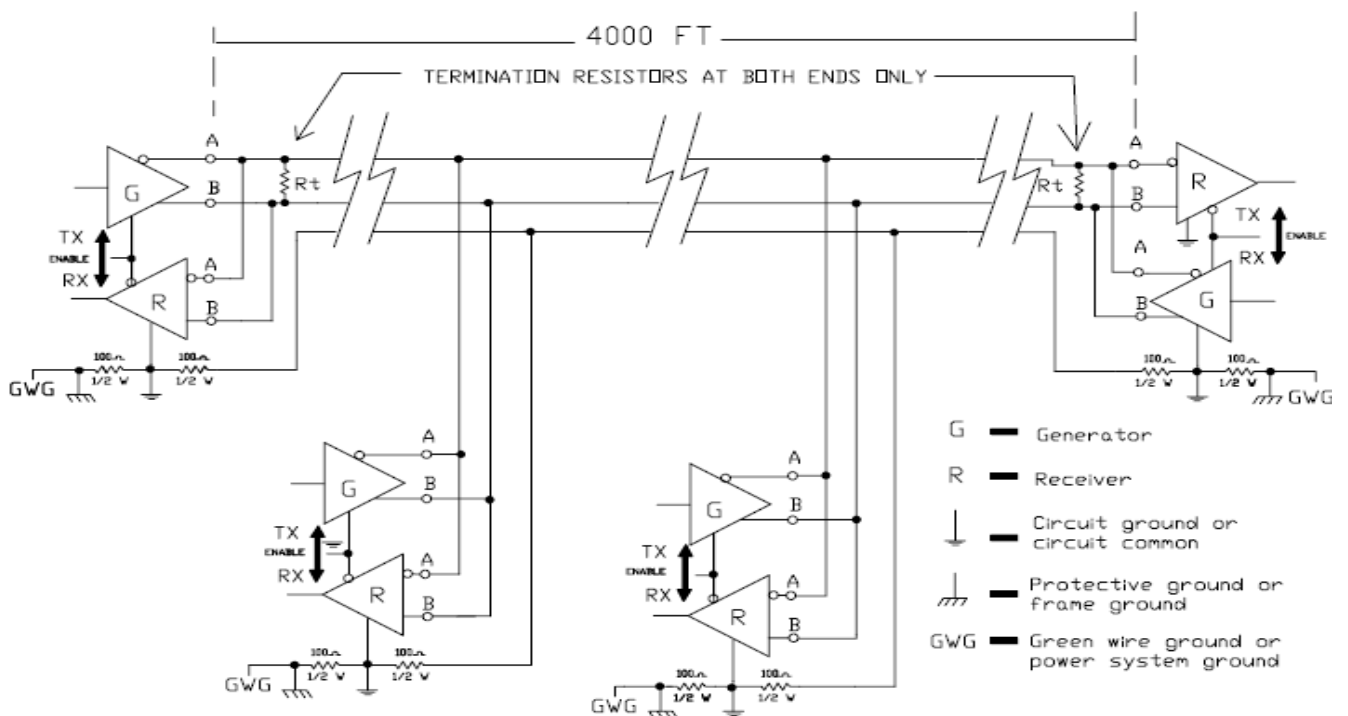
- **Използване на балансирана (симетрична) комуникационна линия;**

Балансирана или **симетрична линия** се използва в случаите, когато изходите на предавателите и входовете на приемниците са диференциални стъпъла. В този случай **токовете** в двата клона на комуникационната линия **са равни** и не зависят от състоянията на други сигнали. Най-често като комуникационна среда се използва **вита (усукана) двойка** с нормирано характеристично съпротивление. Обикновено се прилага съгласуване на линията в двата края с помощта на резистори, равни на характеристичното и съпротивление, като по този начин се премахват отразените сигнали, Това води до повишена шумоустойчивост и дава възможност да се повиши скоростта на обмен на данни. От друга страна, по двойка проводници се предава само един сигнал.

- **Режими на предаване на сигналите**

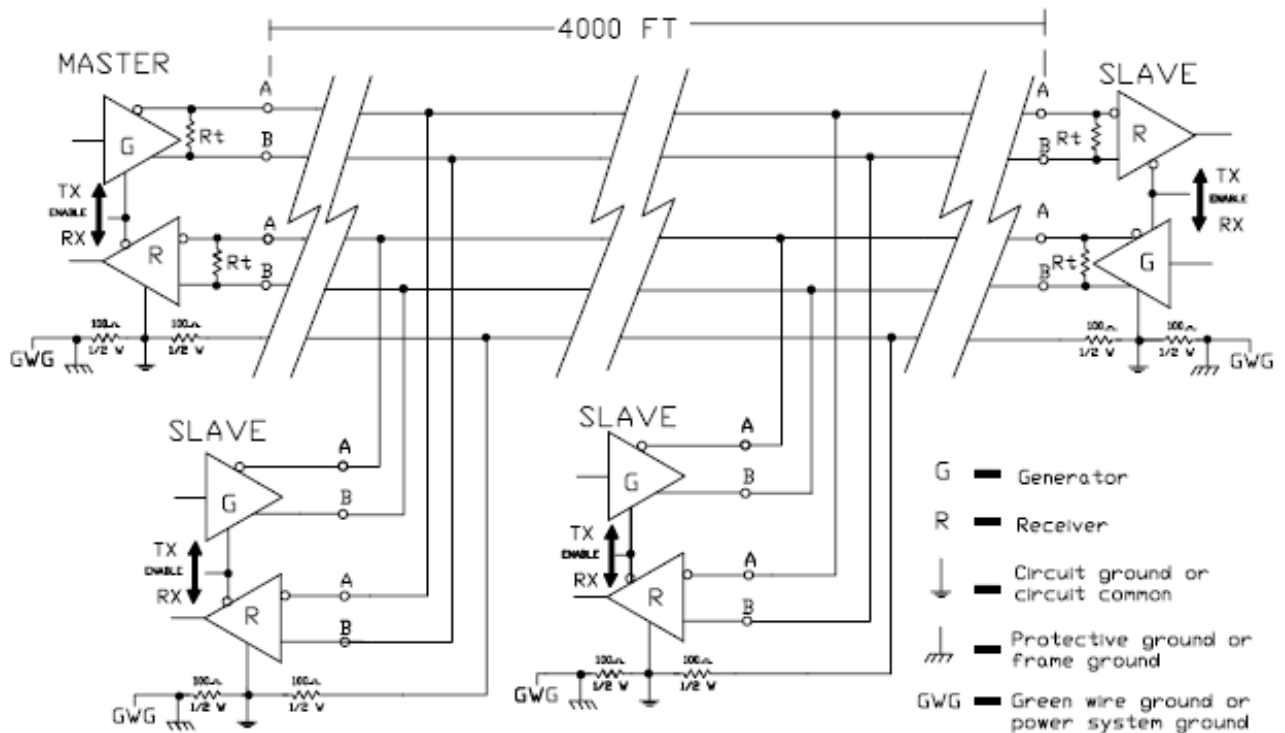
Най-често RS485 се изгражда на базата на **двупроводна комуникационна линия** (фиг. 2.22). В този вариант е приложим **полу-дуплексен режим** на работа, като времеделенето е поверено на драйверната програма, обслужваща интерфейса.

Тази конфигурация намира по-широко приложение поради по-простото изграждане на комуникационната среда. Необходимо е да се съблюдава изискването отклоненията от основната линия до междинните станции да не са много дълги, за да не образуват вторични дълги линии, от които могат да възникнат отразени сигнали.



Фиг. 2.22

Допустимо е изграждането и на четирипроводни линии за връзка, при което конфигурацията добива вида, показан на фиг. 2.23. Тук, както се вижда, има един водещ модул, а всички останали са подчинени. Това означава, че комуникацията от водещия към подчинените модули може да се осъществи във всеки един мо-



Фиг. 2.23

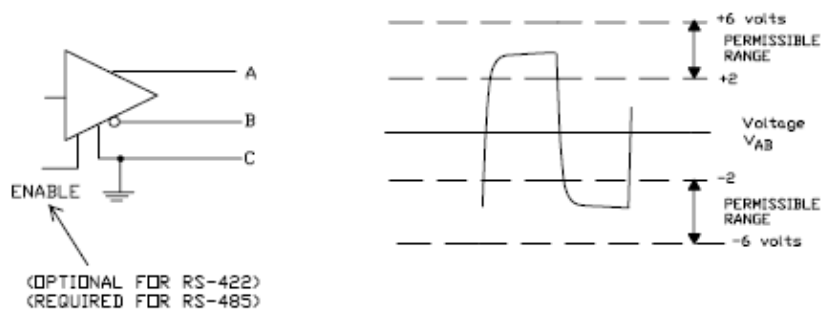
мент от време, докато подчинените модули могат да работят само в режим на времеделене – в даден момент може да предава само един от тях (но едновременно с водещия модул). Тази организация много улеснява алгоритъма на работа на драйверната програма – инициативата за обмен принадлежи винаги на водещия модул, а подчинените са длъжни да връщат отговор само при запитване. От друга страна тук е невъзможно да се осъществи обмен на данни между подчинени модули.

Трябва да се отбележи, че много интерфейсни схеми, които обединяват в себе си приемник и предавател, не допускат използването на този режим, защото при болшинството от тях входовете и изходите вътрешно са обединени и излизат на едни и същи изводи на корпуса.

• *Електрически спецификации на интерфейса RS485*

При предаване на данни в балансирана диференциална система комуникационната линия се състои от два проводника, съответно приемника е с диференциален вход, а предавателят – с диференциален изход. Това означава, че по два проводника се предава само един сигнал. Редно е те да се разглеждат като една неразделна комуникационна линия с нейните специфични параметри - характеристично съпротивление, закъснение, затихване на сигнала на единица дължина и др.

На фиг. 2.24 е показано схематичното означение на предавател (драйвер), предназначен за работа със симетрична линия, както и допустимите нива на изходните сигнали. От тук проличават няколко съществени особености:

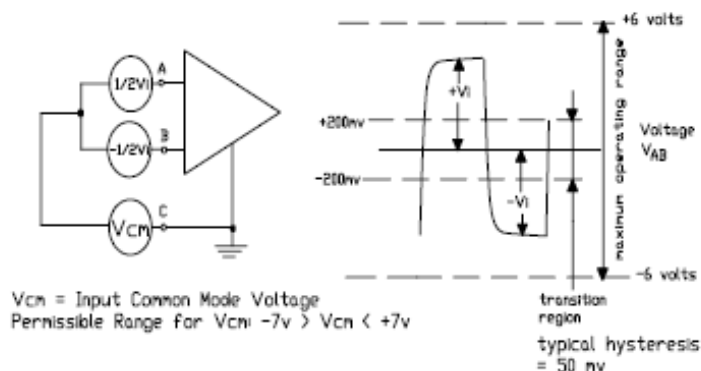


Фиг. 2.24

- схемата съдържа два изхода, чиито сигнали са в противофаза;
- изходите могат да се установяват във високоимпедансно (изключено) състояние с помощта на управляващия сигнал „enable“;
- разликата в изходните нива трябва да бъде по-голяма от 4V, но да не превишава 12V. Синфазното ниво, спрямо което се определят тези сигнали, не е указано;
- Нулевият проводник (означен с извод C) трябва да бъде свързан, но няма изискване линията му да бъде балансирана, а логическите нива на сигналите не се отчитат спрямо него.

Обикновено предавателите за линия се захранват с еднополярно напрежение +5V, а изходните нива са съответно 0V и 5V, които удовлетворяват изискванията на стандарта.

Линейният приемник също е с диференциални входове – фиг. 2.25. Вижда се, че входното диференциално напрежение V_{AB} трябва да е по-голямо от 200 mV, за да се формира правилно изходният сигнал. Отнесено към изискуемите изходни нива на линейния предавател проличава отличната шумоустойчивост на интер-



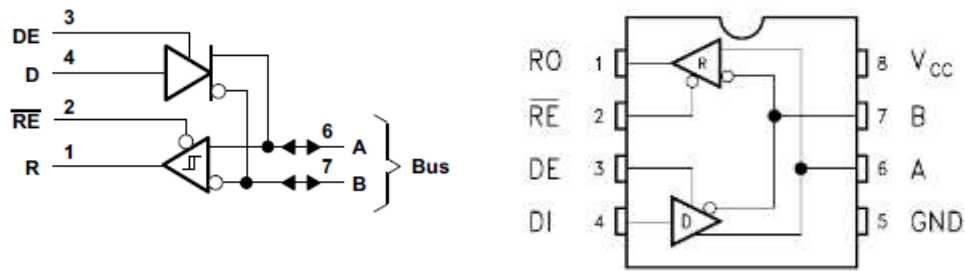
Фиг. 2.25

фейсната система.

На фиг.2.26 е показано условното схематично означение и разположението на изходите на линеен приемо-предавател за RS485, предназначен за работа в полу-дуплексен режим (напр. ST485, MAX3082-MAX3085, MAX485, ADM485 и др.).

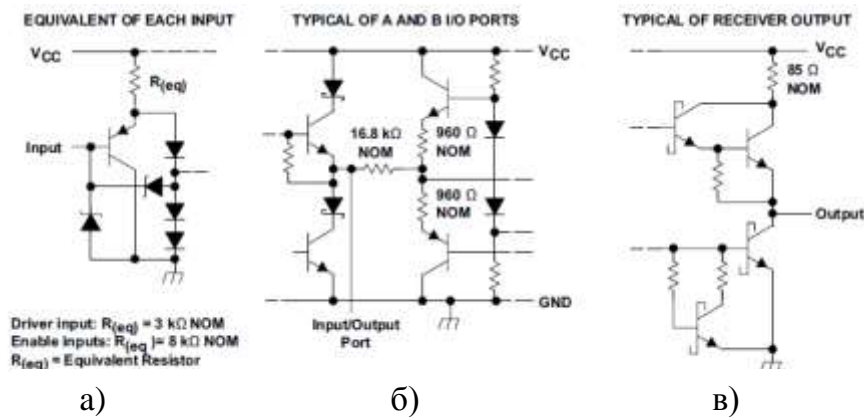
Еквивалентната схема на управляващите входове (**RE**, **DE** и **DI**) са показани на фиг.2.27а, а на изхода **RO** – на фиг.2.27в. Линейните изходи и входове **A** и **B**

съответстват на схемата, показана на фиг. 2.27б. Стойностите на елементите при различните производители варират в известни граници.



Фиг. 2.26

Тези вериги, както и параметрите, дадени в техническите описания, следва да се вземат под внимание при анализ и разчет на съответните вериги.



Фиг. 2.27

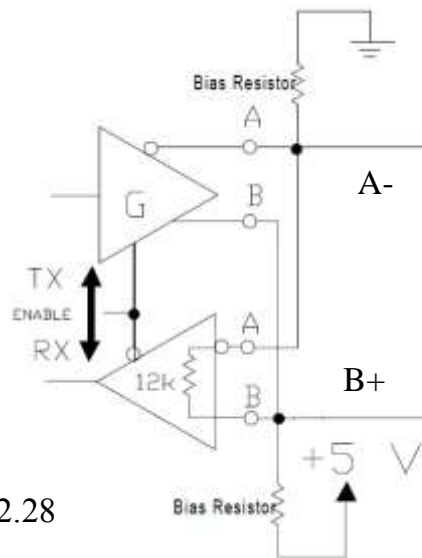
• **Комуникационни и управляващи сигнали;**

Входните диференциални сигнали А- и В+ се преобразуват от линейния диференциален приемник до съответното логическо ниво (5V или 3,3V) на извод RO (receiver output), който се подава към входа на серийния комуникационен интерфейс (SCI, ACIA, UART). Работата на приемника може да бъде забранена чрез сигнала RE (receiver enable), за да се пропускат част от символите (напр. тези, които се предават от този модул). Най-често приемникът се оставя винаги разрешен, а излишните символи се игнорират алгоритмично.

Обикновено се приема, че входното диференциално съпротивление на всеки приемник, присъединен към комуникационната линия, е 12 KΩ. То е необходимо да се вземе предвид (и броя на включените приемници) при провеждане на разчетите за поляризиращите резистори, осигуряващи неактивни нива на линиите, когато няма активиран нито един предавател в системата (вж. фиг.2.28). Тези резистори трябва да гарантират, че разликата на нивата между А- и В+ ще бъде по-голяма от 200 mV.

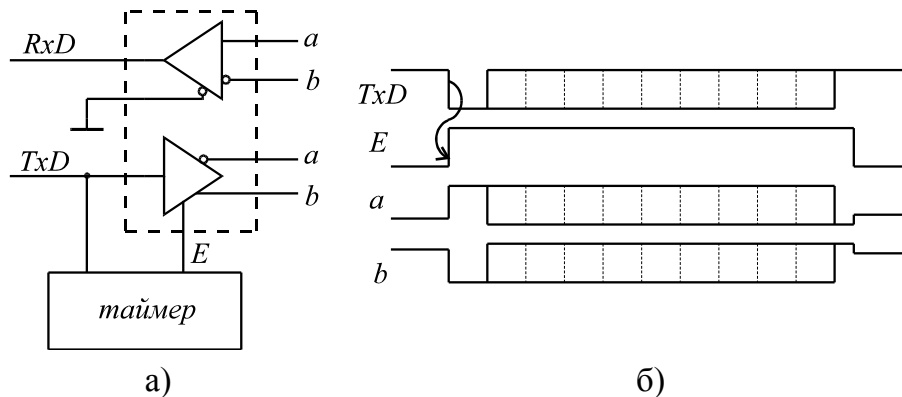
Линейният предавател (line driver) изработва изходните линейни напрежения А- и В+ в зависимост от сигнала, постъпващ от серийния комуникационен интерфейс (SCI, ACIA или UART). Тъй като в даден момент от време към комуникационната линия може да има активиран само един линеен предавател, затова е предвиден управляващ сигнал – DE (driver enable) – който разрешава работата му или

го привежда във високоимпендансно състояние. Този сигнал задължително трябва да се поддържа от системата.



Фиг. 2.28

Възможни са два подхода при реализиране управлението на изходните буфери. Първият от тях предполага използването на отделен сигнал RTS (*Request To Send*) за управление на DE на предавателя (RX enable), който се изработва по програмен път. При втория подход, известен като ASDC (*Automatic Send Data Control*), сигнала за управление на предавателя се изработва от отделна времезадаваща схема (фиг.2.29а), запускана от стартовия бит на предаваните данни. Времедиаграмите, илюстриращи предаването на един символ, са показани



Фиг. 2.29

на фиг.2.29б. Тук запускането на таймера става по падащия фронт на стартовия импулс, предхождащ предаването на първия бит от импулсната поредица. Продължителността на изработвания от него импулс, използван за извеждане на линейния предавател от високоимпендансно състояние, трябва да обхване началото на последния стопов импулс. Предавателят е активиран само за времето, докато си извършва предаването на данни, а през останалото време комуникационната линия е свободна и може да бъде заета от друг предавател. Така се минимизира вероятността ко колизия по линията, при която два предавателя са активирани едновременно.

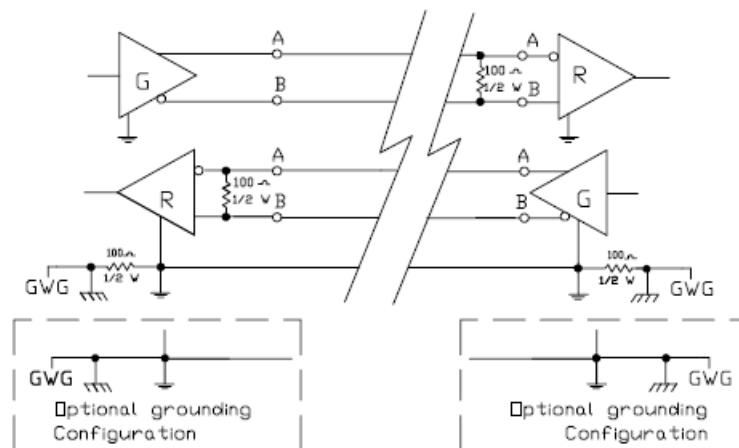
- **Изисквания към канала за връзка. Терминиране;**

Вече беше споменато, че като комуникационна среда се използва симетрична двупроводна линия. Обикновено това е вита двойка, която се характеризира с нормирано характеристично съпротивление, капацитет и затихване на единица дължина. Това означава, че комуникационната линия при определени условия се явява дълга линия, при което се внася определено закъснение в разпространението на сигнала. Освен това могат да съществуват условия за възникване на отразени сигнали, които да доведат до невъзможност да се установи надеждна връзка, без възникване на грешки.

Обикновено се счита, че в такива случаи линията трябва да се терминира, за да се избегнат условията за възникване на отразени сигнали. Терминирането се изразява в изравняване на импеданса на възела, свързан към дадена точка, с импеданса на линията, като при това отраженията се елиминират. Наред с преимуществото от премахване на отразените сигнали, доброто терминиране води след себе си и някои недостатъци – увеличава се натоварването на драйверите (консумираният ток в интерфейлната система), повишава се сложността на инсталацията, променят се изискванията за изграждане на отклонения и се затруднява модификацията на системата. Добрият анализ на всеки конкретен случай дава възможност интерфейлната система да бъде оптимизирана, при което да се получат най-добри резултати. За целта е добре да се имат предвид следните обстоятелства:

1. Да се прецени добре, доколко конкретната симетрична линия се явява дълга линия. Критерий за това може да бъде какво е общото закъснение на сигнала в линията, отнесено към фронта на нарастване или спадане на сигналите (взема се под внимание по-краткият фронт). За фронтовете на нарастване и спадане на сигналите оценка може да се направи, като се вземат под внимание изходното съпротивление на драйверите и еквивалентния капацитет на свързващите проводници. **Ако общото закъснение в линията е по-малко от тези фронтове се счита, че това не е дълга линия и може да не се извършва съгласуване.** При разчетите в първо приближение може да се приеме, че закъснението в 1 метър от линията е 5 ns. Като се знае дължината на линията е лесно да се изчисли цялостното закъснение;

2. При 4-проводна линия е допустимо съгласуването да се извършва са-



Фиг. 2.30

мо при най-отдалечения приемник в двете комуникационни линии (вж. фиг.2.30). При липса на отразени сигнали от към входовете на приемниците (при добро съгласуване) няма да има върнати сигнали към изходите на предавателите и няма да има какво да се отрази (вторична отразена вълна), независимо, че възелът към предавателя не е съгласуван;

3. В случаите на двупроводна линия (при полу-дуплексен режим на работа), ако е необходимо да се извърши съгласуване, това задължително се прави и от двете страни на комуникационната линия;

4. Отклоненията от основната линия да се правят с кабели с такава дължина, че времето за разпространение в тях да е по-малко от фронтовете на сигналите – т.е. те да не се явяват дълги линии. Допълнително условие е входовете на приемниците да бъдат високоомни (входното им съпротивление да е поне на 1-2 порядъка по-високо от характеристичното съпротивление на линията).

5. Ако времето за предаване на 1 бит при най-високата скорост на предаване по интерфейса е по-голямо на порядък от трикратното закъснение на сигнала, дори и комуникационната линия да се явява дълга линия, също е допустимо да не се извършва терминиране. Основание за това ни дава факта, че към момента на семплиране на сигнала приходния процес вече ще е завършил.

Пример. Нека скоростта на обмен бъде 9600 bit/sec , а дължината на линията е 200 м. Тогава времето за предаване на 1 бит ще бъде $104 \mu\text{s}$ (вж. табл. 2.1), а момента на семплиране ще бъде в $52 \mu\text{s}$. Съответно, закъснението на сигнала в линията ще бъде $1 \mu\text{s}$, а преходния процес ще приключи до 3-тата μs . Явно е, че към момента на семплиране нивото на сигнала ще е установено. При същата линия, но скорост на предаване 115 Kbit/s това условие вече няма да е изпълнено и линията трябва да бъде терминирана.

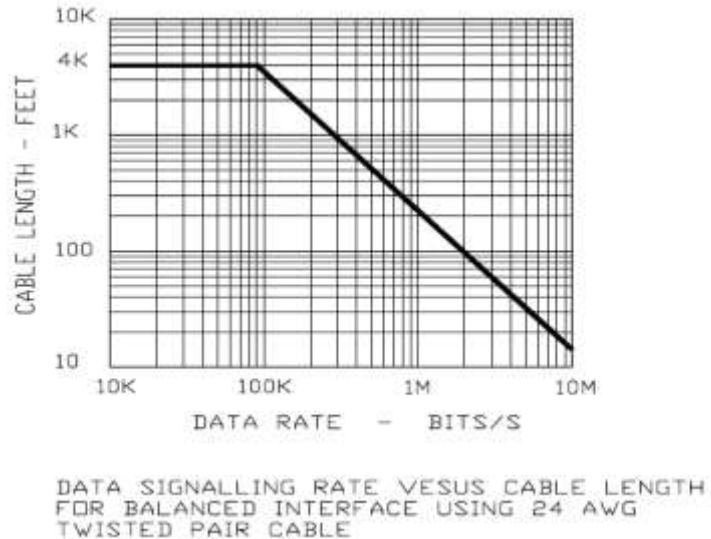
Допустимо е терминирането на комуникационната линия да бъде извършено както по постоянен ток, така и по променлив ток, както е показано на фиг.2.31.



Фиг.2.31

Съществен е въпросът за топологията на преносната среда. Свързващият кабел трябва да има ясно изразени начало и край. Допускат се неголеми отклонения (с дължини, по-къси от тези, които могат да се разглеждат като „дълги“ линии), като входните импеданси на приемниците, свързани към тях, трябва да бъдат много по-големи от характеристичното съпротивление на линията. Ако се допусне „дендритна“ структура на окабеляването, най-вероятно устойчива комуникация няма да може да бъде установена.

Параметрите на комуникационните кабели ограничават възможностите за работа на по-високи скорости на обмен. На фиг. 2.32 е показана в графичен вид зависимостта на допустимата скорост на обмен от дължината на комуникационната линия. Обърнете внимание, че при скорости над 90 Kb/s допустимата дължина на комуникационната линия силно намалява.



Фиг. 2.31

При положение, че съществува неувереност при вземането на решение за терминиране на комуникационната линия, по добре е то да бъде осъществено съгласно предписанията. Това ще гарантира безпроблемна работа при използване на избрания интерфейс.

- **Механични характеристики, куплунги за връзка;**

Стандарта не специфицира куплунгите за връзка. Някои често използвани куплунги са показани на фиг. 2.31.



Фиг. 2.32

Изборът силно зависи от характеристиките на средата, в която ще бъде използвана системата. При лабораторни условия е удачно използване на съединители DB9F/DB9M. Ако се използват комуникационни кабели CAT.5 могат да се използват съединители RJ45F/RJ45M. При работа в промишлени или полеви условия е по-добре да се използват съединители, отговарящи на конкретните условия.

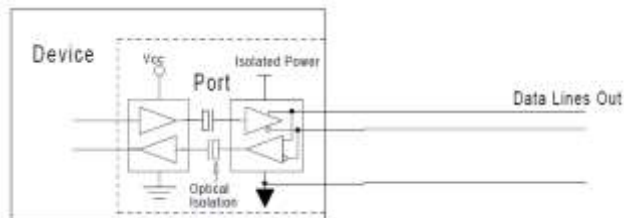
- **Протокол за обмен на данни**

В стандарта няма дефиниран логически протокол за обмен на данни, който да трябва да се спазва от всички потребители. Фактът, че е базиран на SCI позволява да бъдат разработени много мрежови системи, които използват RS485 като физическо ниво за пренос на данни в своите реализации.

- **Галванично разделяне на интерфейса**

Препоръчително е да се реализира галванично разделяне между вградената система и интерфейса. Това се счита за задължително, ако отделните абонати, свързани към интерфейса, се намират в различни помещения и още повече, ако са в различни сгради. При полеви и промишлени приложения също е задължително да се прилага галванична разделяне. Така се повишава общата шумоустойчивост на интегрираната система и се гарантира, че евентуални повреди ще бъдат ограничени само да комуникационното оборудване, непосредствено свързано към преносната среда.

Галваничното разделяне като правило се осъществява между серийния комуникационен интерфейс (SPI, ACIA, UART) и трансмитера към комуникационната



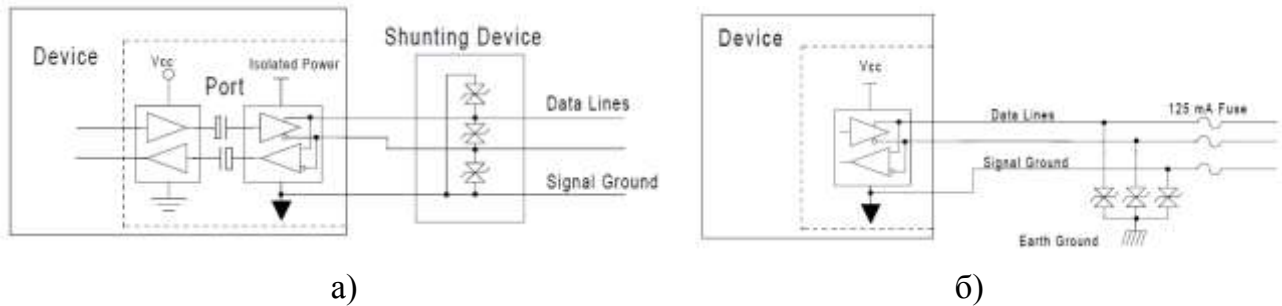
Фиг. 2.33

линия - фиг.2.33. При това трябва да се има предвид, че:

- Трябва да се осигури отделно, галванично разделено и надеждно захранване за интерфейсните схеми. То обикновено се вгражда само в един ат абонатите към мрежата;
- Използваните оптрони трябва да са високоскоростни – 1 или 10 Mbps. На това условие отговарят такива оптрони, при които използваната опто-двойка е от типа светодиод-фотодиод. Подходящи са, например типовете 6N135, 6N136, 6N137, TLP2362 и др.;
- Освен двата сигнални проводника A-/B+, свързани към една вита двойка, в интерфейсната система задължително присъства сигналната „земя“ и захранващото напрежение за интерфейса. Добре е всички проводници да бъдат обхванати от екранираща „земя“.

- **Защита от пренапрежение и електромагнитна съвместимост**

При промишлени и полеви приложения на интерфейса е необходимо да се вземат сериозни мерки срещу проникване на електромагнитни смущения, възникващи в следствие на електростатични разряди в атмосферата (мълнии), неизправности и аварии в силови агрегати и пр. Днес предимно се разчита на използване на различни конфигурации на ценови бариери, в които се използват предимно полупроводникови супресори (понякога се наричат трансил), както е показано на фиг. 2.34. Подходящи супресори са например SMAJ5,0A/SMAJ5,0CA (еднопосочен/двупосочен трансил), SMBJ5,0A/SMBJ5,0CA, SM712 и много други типове.



фиг. 2.34

Понякога се добавят и самовъзстановяеми предпазители. Трябва да се има предвид, че те обикновено са нисковолтови – в изключено състояние обикновено не допускат върху себе си напрежения, по-високи от 60 V.

При възникване на неизправности при подобни интерфейсни системи, редно е те да се търсят най-напред в ценовите бариери. Обикновено при екстремни обстоятелства те успяват да предотвратят възникването на по-сериозни проблеми или ограничават щетите до дефетиране на някой от защитните елементи.

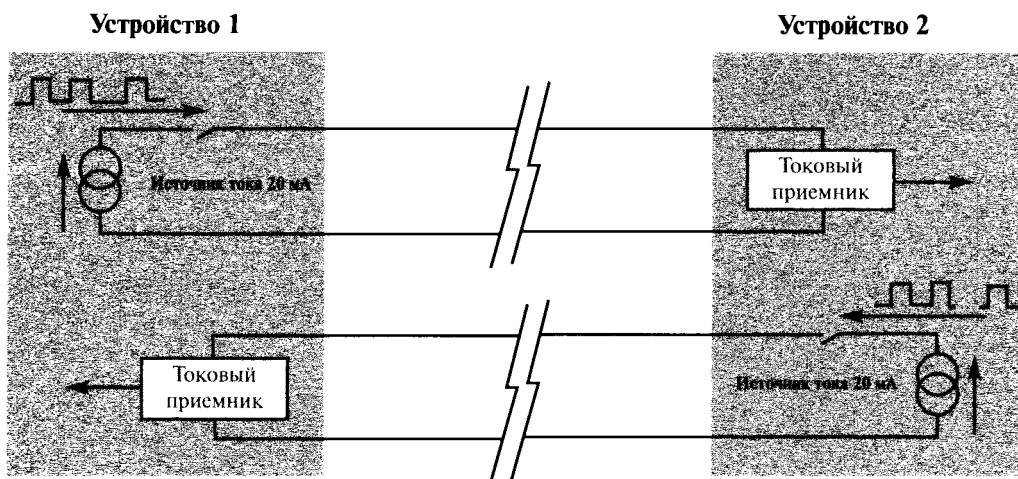
• **Заключение**

При изграждане на интерфейсна система на базата на RS485 могат да бъдат допуснати определени грешки, които да доведат до неустойчива работа. При проектирането ѝ трябва да се намерят еднозначни отговори на следните въпроси, касаещи характеристиките на интерфейса:

- С дву-проводна или четири-проводна интерфейсна система ще работи проектираното устройство?
- Каква ще бъде връзката за сигналната „земя“?
- Предвижда ли се галванично разделяне на устройството от интерфейса?
- Необходима ли е защита от пренапрежения?
- Необходимост от поляризиране и терминиране на комуникационните линии? Предвижда ли се възможност за модификация на интерфейсната система?
- Какво е максимално допустимото закъснение при предаване на данни?
- Какъв е програмируемият диапазон за логически адрес на устройството? Как се задава при конфигуриране логическият номер на устройството?
- Каква скорост на предаване или диапазон на скоростта на предаване ще се поддържа? Как се конфигурира текущия режим на интерфейса?

2.2.4. Токов кръг 4-20 mA

Това е широко разпространен в близкото минало шумоустойчив интерфейс, който се е характеризирал със своята простота и надеждност. Позволява връзка само между две системи (двучточкова връзка) – фиг.2.35. Предавателят генерира определен ток – най-често 20 mA – който се манипулира от състоянието на серийния комуникационен интерфейс (SCI, ACIA, UART). Използва се двупроводна връзка, за да се затвори токовия контур през входа на приемника – в най-простия случай това може да е резистор с подходяща стойност. Същата конфигурация е

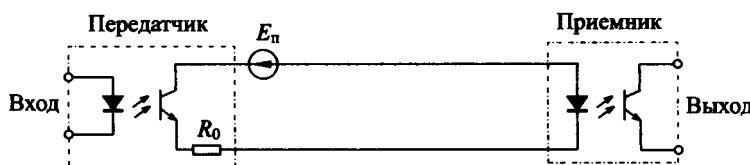


Фиг. 2.35

необходимо да се изгради, за да може да се осъществи комуникация и в обратната посока.

Ако в приемната и/или предавателната част на интерфейса се включат оптрони, много лесно може да се реализира галванично разделяне на интерфейса, както е показано на фиг.2.36. Тук отново трябва да се постави отделен, галванично разделен токоизточник за захранване на веригите на интерфейса.

Понякога интерфейсът се модифицира, при което вместо 0/20 mA се работи с нива 4/20 mA. Това дава възможност да се контролира изправността на комуникационната линия – при прекъсване или окъсяване на двата проводника това събитие лесно може да бъде установено и да се преведат системите в аварийен режим.

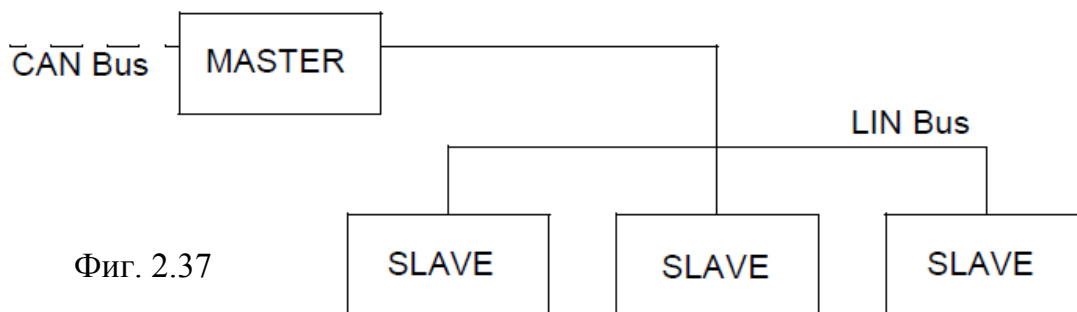


Фиг. 2.36

2.2.5. Асинхронен интерфейс LIN (Local Interconnect Network)

Интерфейсът LIN (Local Interconnect Network) е разработен като евтина и достъпна алтернатива на CAN протокола. През 1998 г. група компании, включително Volvo, Motorola, Audi, BMW, Daimler Chrysler и Volkswagen, сформират консорциум за разработване на LIN. Последната версия на LIN е LIN 2.0 и е издадена през 2003 г..

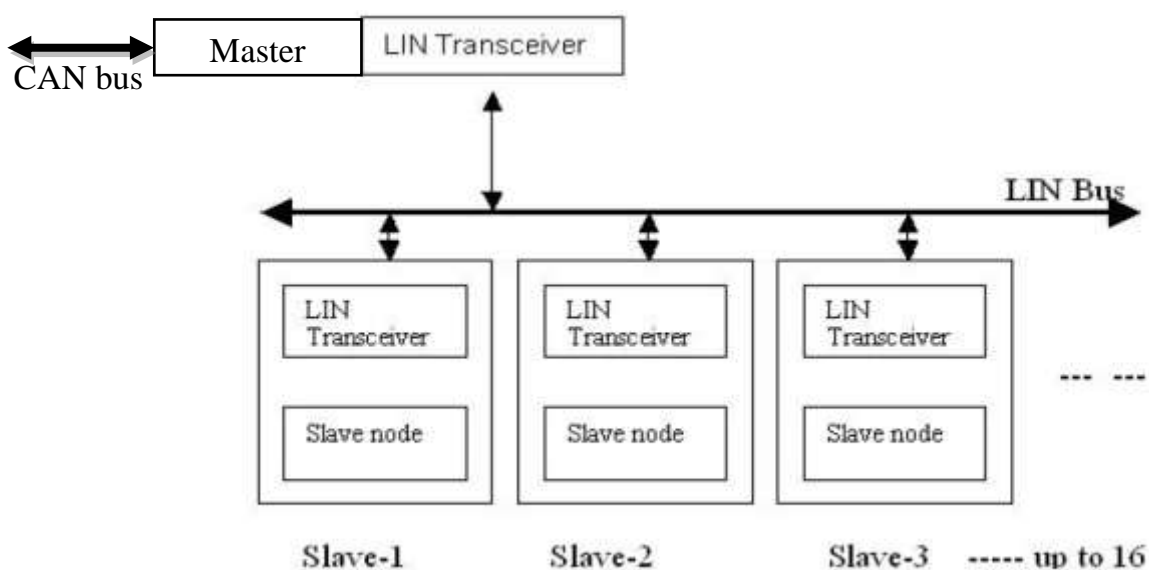
LIN е сериен, байтово ориентиран протокол за комуникация, базиран на SCI/UART интерфейс, предназначен да поддържа автомобилни мрежи като подмножество на мрежовата зона на контролера CAN (Controller area network) - фиг.2.37. Основните предимства на този протокол (в сравнение с CAN) са ниската цена и ниската скорост и се използват за мрежи на къси разстояния. LIN 2.0 предоставя интересни диагностични функции. Намира приложение за изграждане на връзка към измервателни и изпълнителни механизми, които не изискват бързо обслужване при заявка и не генерират интензивни даннови потоци – напр. механиз-



Фиг. 2.37

ми за заключване на врати, управление на стъкла, огледала и чистачки, ниво на горивото в резервоара, измерване на околна температура и пр.

LIN шината има ясно изразена йерархична структура – в нея може да има само един главен модул и до 16 подчинени модули (фиг.2.38). В LIN не е предвиден механизъм за откриване на колизии (едновременно включване на два предавателя), затова всички съобщения се инициират от главния с най-малко един подчинен



Фиг. 2.38

отговор, отговарящ за даден идентификатор на съобщение. Типичният главен модул е микроконтролер от среден клас, докато подчинените могат да бъдат поевтини микроконтролери или специализирани ASIC.

По-съществените характеристики на интерфейса включват:

- **Вид на комуникационната линия**

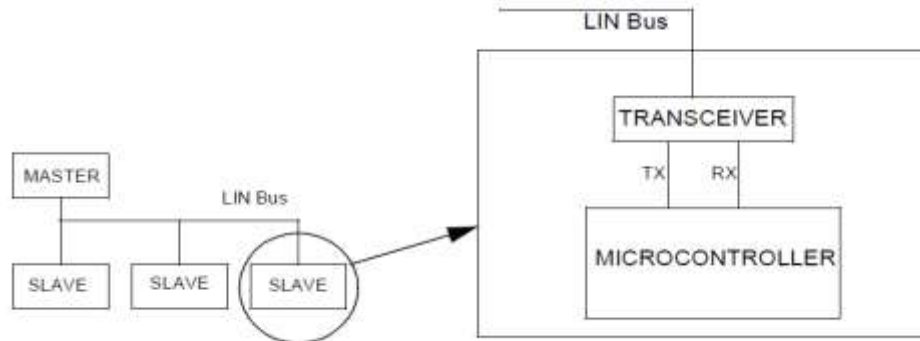
Използва се небалансирана несиметрична комуникационна линия, като нивата са спрямо общия проводник. В някои от приложенията си използват единични екранирани кабели, като се внимава да не се затварят общи контури с други вериги през екрана.

- **Режим на осъществяване на обмена на данни**

Приетият принцип на магистрална организация на комуникационната среда допуска само полу-дуплексен режим на работа. Времеделенето се осъществява под управлението на вграденото програмно осигуряване, като ролята на разпределител се изпълнява от водещия (главния) модул. Подчинените модули могат да отговарят само при получаване на запитване от главния модул.

• **Електрически спецификации и кодиране на сигналите. Комуникационни и управляващи сигнали**

Топологията на една система, базирана на LINbus, е показана на фиг. 2.39., както и връзката на трансивера към микроконтролера (в главния или подчинените модули) чрез серийна комуникационна магистрала SCI. От тук могат да се напра-

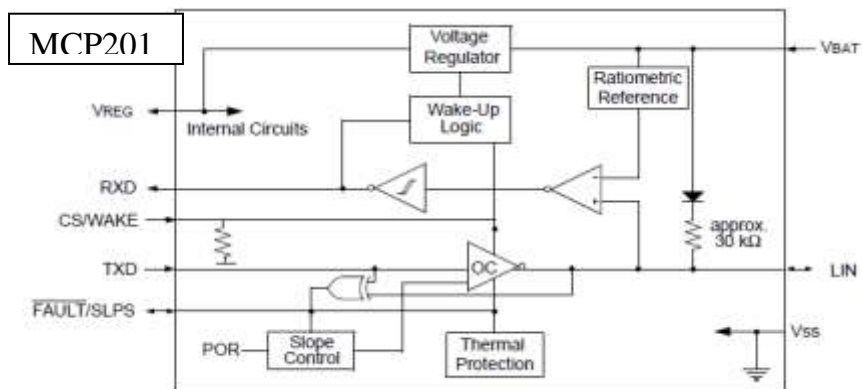


Фиг. 2.39

вят няколко съществени извода:

1. Използват се познатият ни (от разглеждането на SCI) метод на кодиране на сигналите NRZ и описаните формати за синхронизация и предаване на данните;
2. Могат да бъдат използвани стандартните скорости за предаване на данни. В случая на LIN те са ограничени между 2400 и 19200 Bit/s;
3. Електрическите сигнали между микроконтролера и трансивера (драйвера към интерфейса) отговарят на стандартните CMOS нива.

Предлагат се различни типове трансивери за LIN (напр. MCP201 – фиг.2.40, L9638 и др.). Всички те удовлетворяват електрическите характеристики на интер-



Фиг.2.40

фейса за нива на сигналите и начинна формиране на сигналите. Характерно за драйверите за LIN е, че работят с NMOS транзистори с отворен дрейн, като обединението на сигнала е по схема „жично ИЛИ“. Общият товарен резистор обикновено се поставя при водещия модул и е от порядъка на 1 КΩ. Вижда се, че се дефинират две състояния на интерфейса:

- **доминантно**, чието ниво е близо до 0V;
- **рецесивно**, чието ниво е определено от захранването V_{BAT} .

В неактивен (IDLE) режим, когато не се извършва комуникация по интерфейса, на шината има високо (рецесивно) ниво – лог.1. Началото на предаването, както е известно, започва с преход към ниско ниво – лог.0, което е доминантно.

Ако няколко драйвера започнат предаване едновременно няма да се получи колизия по линията, единствено данните няма да бъдат предадени вярно. Тази ситуация лесно се обработва от заложените в системата механизми за откриване на грешки. За всеки блок данни, който се предава по линията, се изчислява контролна сума, която се проверява в приемната част и при несъответствие се генерира флаг за възникнала грешка.

Вижда се, че нивото на рецесивното състояние зависи от захранването V_{BAT} . Праговете на превключване на входния буфер се формират от стойността на това напрежение – превключването при преход от 0 към 1 става на ниво $0,6 V_{BAT}$, а обратния преход – от 1 към 0 – се възприема при достигане на $0,4 V_{BAT}$ – т.е. хистерезисът на линейния приемник е $0,2 V_{BAT}$, което осигурява много добра шумоустойчивост на интерфейса.

- **Изисквания към канала за връзка**

Общата дължина на свързващите проводници, изграждащи комуникационната линия, не трябва да надхвърля 40 м. Има наложени ограничения на скоростта на нарастване на сигнала, които се следят и поддържат от отделен функционален модул (slope control) в трансмитера.

При изграждане на подобни системи, добре е спрямо комуникационната линия и захранващото напрежение V_{BAT} да се предвидят ценерови бариери чрез поставяне на супресори.

- **Механични характеристики на съединителите**

В стандарта на интерфейса вида и типа на свързващите куплунги не са дефинирани не се предявяват изисквания спрямо тях.

- **Галванично разделяне на отделните модули в системата**

Независимо, че стандарта не го изисква, е допустимо и дори желателно да се предвиди галванично разделяне на водещия и отделните подчинени модули от комуникационната магистрала. Това най-лесно може да стане, като се поставят оптрони между иводите на SCI на микроконтролера и трансмитера. В този случай трябва да се реши и проблема с индивидуално галванично разделено захранване за всеки модул – решение, което неминуемо ще доведе до оскъпяване на самото изделие.

- **Логически протокол**

Логическият протокол е дефиниран от стандарта и трябва стрикто да се спазва, ако се използват изделия, произведени от трети доставчици. Има достъпни изходните кодове за обите програмни модули на много разработчици, което би облекчило и ускорило процеса на разработка на специфични изделия, предназначени да работят в средата на LIN.

2.2.6. Модеми

За осъществяване на комуникация в среда с ограничена честотна лента най-често се използват най-разнообразни **модеми** – устройства, в които първо се извършва **модулация** на носещ сигнал със сигнала, който предстои да бъде предаван,

а във приемната страна се извършва обратната операция – *демодуляция*. Модулацията на носещия сигнал се изразява в моментна промяна на един или няколко негови параметри – амплитуда (амплитудна модулация), честота (честотна модулация), фаза (фазова модулация) и др.. Частен случай е, когато тази промяна може да заема само две стойности – тогава се говори за манипулация – амплитудна, честотна или фазова. Разнообразието от поддържаните режими е огромно в зависимост от преносната среда, носещата честота, предназначение, връзка към микропроцесорната система и др.

При някои от модемите като модулационен може да се използва сигнала, изработван от серийния комуникационен интерфейс. Като пример може да се разгледат *препоръките V.23* на ITU (International Telecommunication Union), които са валидни на територията на Европа. Те касаят предаване на данни по телефонни линии (комутируеми или наети), при които честотната лента на сигнала е ограничена между 300 Hz и 3600 KHz.

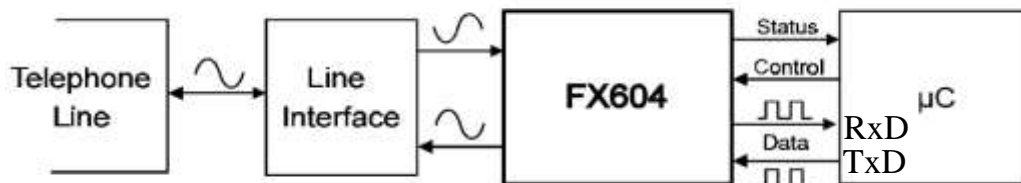
Основните характеристики съгласно препоръките, на които трябва да отговаря модема при предаване на данни, се свеждат до:

- използване на скорост на модулация до 600/1200 bit/s в комуникационния канал;
- използване на честотна модулация със синхронен или асинхронен режим на работа;
- допустимо е (по избор) използване на обратен канал при скорости на модулация до 75 bit/s за контрол на грешките.

Препоръките допускат използване на два режима (скорости) на предаване на данни, като параметрите им са указани в табл.2.4

	Скорост на предаване	Средна честота	Лог „1“ (mark)	Лог „0“ (space)
	bit/s	F_0 , Hz	F_Z , Hz	F_A , Hz
Mode 1	600	1500	1300	1700
Mode 2	1200	1700	1300	2100

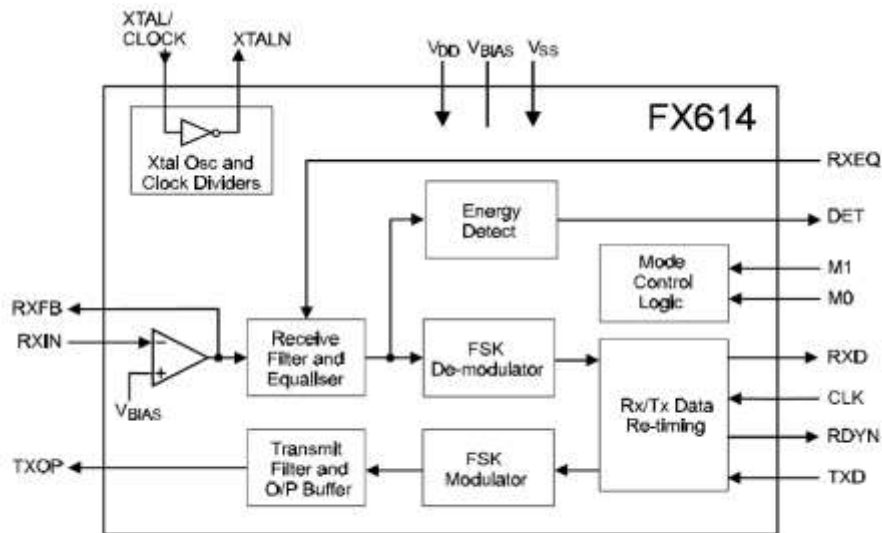
Подходящ за вграждане в микропроцесорни системи е модемът FX604, който е достъпен в интегрално изпълнение, реализиран по CMOS технология. Мястото му и начинът на свързване към една вградена система е показан на фиг. 2.41.



Фиг.2.41

Вътрешната му структура е идентична с тази на модема FX614 (фиг.2.42). Разликата между тях се състои в това, че при втория честотата $F_Z=1200$ Hz, а $F_A=2200$ Hz, с което той удовлетворява изискванията на стандарта BEL202, валиден на територията на САЩ. Обърнете внимание, че TXOP и RXIN са аналогови

хармонични сигнали (синусоидални), а не са цифрови, и са пригодени за предаване по аналогова комуникационна линия.



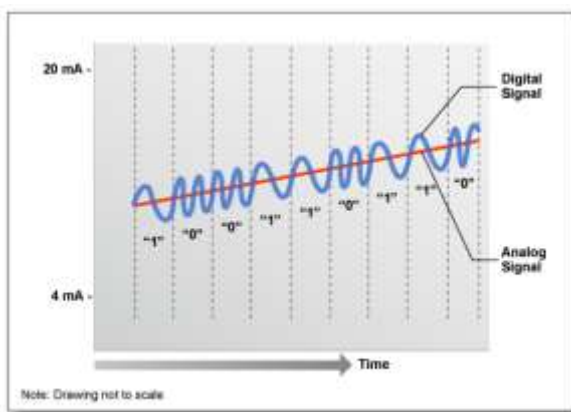
Фиг. 2.42

2.2.7. Асинхронен поледи интерфейс HART

В промишлената автоматизация много широко разпространение намира аналоговият токов интерфейс 4-20 mA както за управление на изпълнителни прибори, така и за снемане на данни от разнотипни сензори – температурни, на налягане, позиционни и пр. В течение на времето и с развитието на елементната база промишлените аналогови измерителни и управляващи системи се превръщат в „интелигентни“ разпределени цифрови системи, които наред с аналоговите канали за връзка се нуждаят и от подходящи комуникационни интерфейси. Такива възможности предоставя полевият протокол HART (Highway Addressable Remote Transducer).

- **Кодирание на сигналите**

Полевият протокол HART е базиран на споменатия по-горе стандарт за комуникация по телефонна линия BEL202. Логическите нива „1“ и „0“ от NRZ кодирания сигнал от SCI се модулират честотно, като се представят съответно с



Фиг. 2.43

един или два периода с честота съответно 1200 или 2200 Hz – модулация FSK (*frequency shift keying*). Синусоидалният сигнал с тези две алтернативно сменящи се честоти се наслагва върху аналоговия токов изходящ сигнал от сензорната система, както е показано на фиг. 2.43. Както е известно, средната стойност на FSK сигнала винаги е нула, което гарантира, че стойността на аналоговия сигнал ще остане непроменена.

• **Вид на комуникационната линия**

Както беше подчертано, модулираният сигнал се наслабва върху съществуващите в промишлеността аналогови измервателни токови линии, осигуряващи връзката между сензорите и управляващите системи. Най-често това са екранирани усукани двойки със специфичен капацитет на проводника от 20 до 70 pF/m и диаметър обикновено 1 mm . От капацитета на кабела и броя на включените устройства зависи и максималната дължина на линията – от 700 до 3000 m .

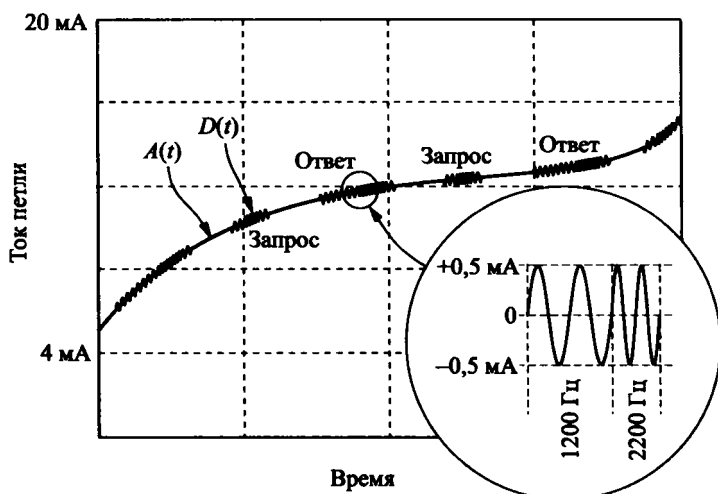
• **Режими на предаване на сигналите и възможни конфигурации**

За полевия протокол HART е характерна възможността за едновременно еднопосочно предаване на аналогови сигнали и цифрова комуникация в полу-дуплексен режим. В зависимост от конкретните условия, системата може да бъде конфигурирана като:

- Двучточкова ситема (*Point-to-Point*). Връзката се осъществява само между две устройства – водещ модул (контролер) и подчинен модул (интелигентен сензор). В този режим може да се използват възможностите на вградения в системата аналогов канал.
- Многоточкова система (*Multi-drop*). Към един водещ модул могат да бъдат присъединени до 15 подчинени модула. В този случай възможностите на аналоговите канали от подчинените модули не могат да бъдат използвани.

• **Електрически спецификации на интерфейса**

Както беше подчертано, системата е ориентирана основно за предаване на аналогови токови сигнали, които могат да приемат стойности между 4 и 20 mA .



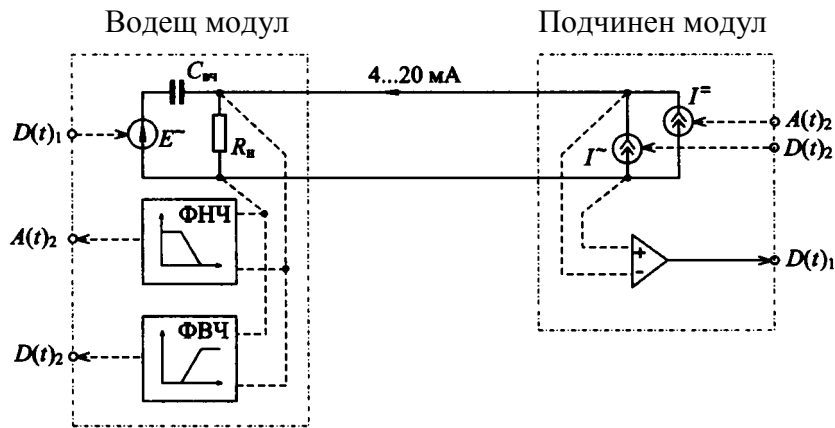
Фиг.2.44

Към тях се наслабва синусоидалният сигнал от изхода на модема, както е показано на фиг. 2.44. Амплитудните стойности на този ток достига $\pm 0,5 mA$. Терминирането на комуникационната линия се извършва с оглед осъществяване на оптимални условия за предаване на аналоговия сигнал. Най-често като товарни съпротивления се използват такива със стойности 250 Ω , гарантиращи нива на изходния аналогов сигнал от 1 до 5 V .

• **Принцип на действие на физическо ниво**

Принципът на действие на полевия интерфейс HART на физическо ниво се илюстрира от фиг. 2.45. Тук възникват три ситуации при предаване на сигналите:

- *Измерване на аналогова величина*. Тук сигналът $A(t)_2$ отразява моментната стойност на измервания параметър. Той се кодира като бавно променящ се ток I , който се предава по витата двойка. В приемната среда, протичайки



Фиг. 2.45

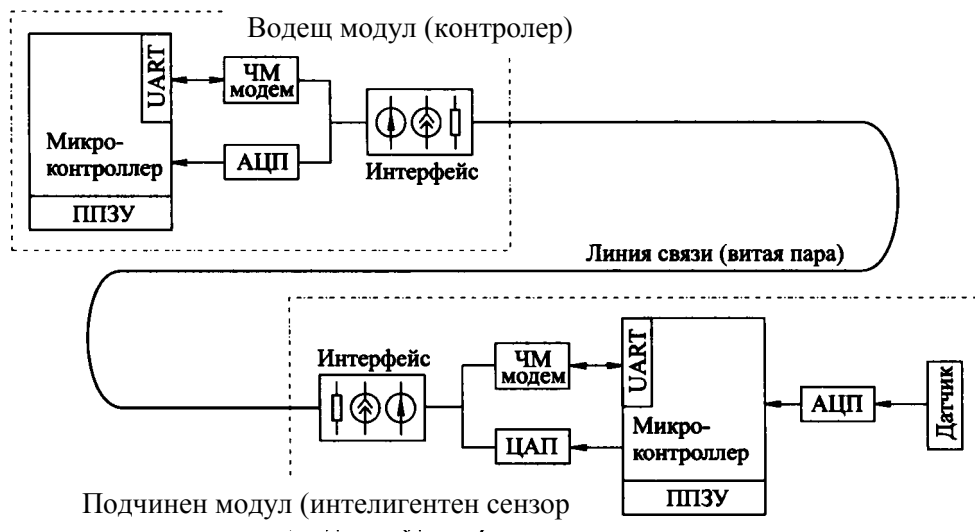
през $R_{И}$, създава напрежение върху него, като само нискочестотната съставна от него се пропуска от нискочестотния филтър НЧФ за следващо измерване и регистрация;

– *Предаване на данни от водещия към подчинения модул.* Изходният синусоиден сигнал от модема се представя като напрежителен сигнал с амплитуда 125 mV (източникът $E \sim$) и е свързан към линията чрез прехвърлящия кондензатор $C_{вч}$. Така това напрежение достига до входа на усилвателя в подчинения модул, който го подава към неговия модем и към SCI в приемната страна;

– *Предаване на данни от подчинения към водещия модул.* Изходът на модема от подчинената страна – сигналът $D(t)_2$ – управлява токовият източник $I^$, генериращ ток $\pm 0,5 \text{ mA}$, който се сумира с аналоговия сигнал и се предава по линията. Така в приемната страна върху резистора $R_{И}$ се получава синусоидално напрежение с амплитуда 125 mV и честота 1200 или 2200 Hz. То се пропуска от високочестотния филтър ФВЧ към модема на водещия модул.

• **Структурна схема на „двучкова“ измервателна система**

Всяко устройство (и водещо, и подчинено) в HART-системата задължително



Фиг. 2.46

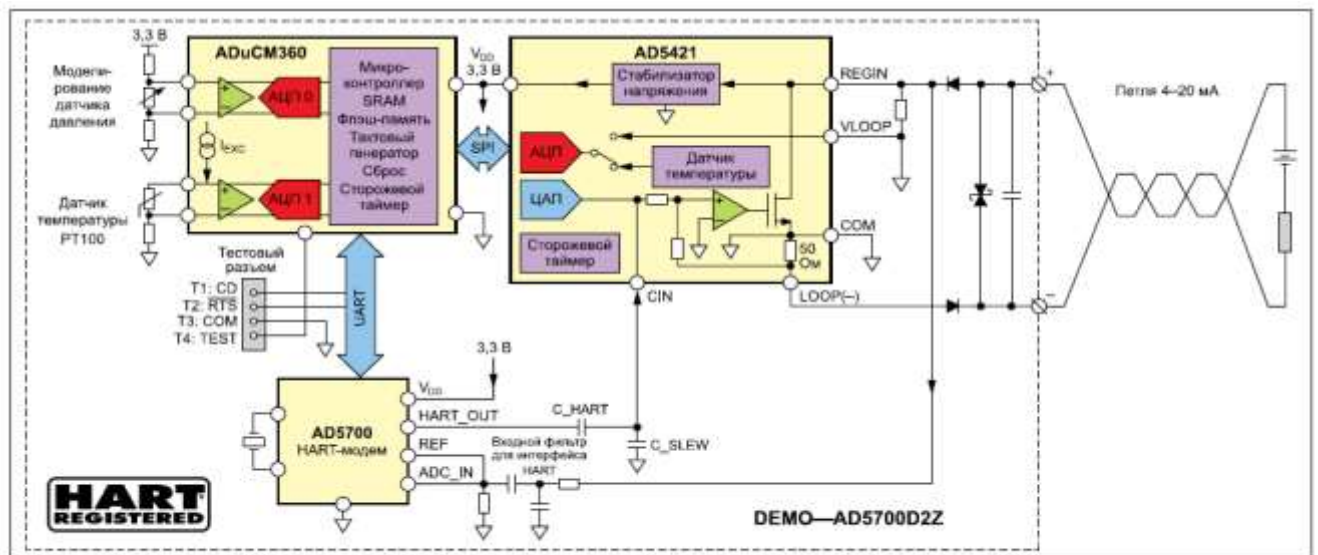
съдържа в себе си микроконтролер с вграден SCI и модем, удовлетворяващ спецификациите по BEL202 (фиг. 2.46). За да се поддържат функциите, заложен в аналоговия интерфейс 4-20 mA, в системите трябва да има вградени и АЦП (за водещия модул) и ЦАП (за подчинения модул).

В подчинения модул се съдържат допълнителни вериги – аналогови или цифрови – които позволяват следене състоянието на външни вериги или измерване на съответните физични величини – температура, налягане, позиция и др.

Когато подчиненият модул се захранва от интерфейса, сумарната консумация на всички компоненти, които го изграждат, не трябва да превишава 3,5 mA. Съвременните компоненти, изградени по CMOS технология, позволяват това условие лесно да бъде удовлетворено.

- **Структурна схема на сензор за полевия интерфейс HART**

Структурна схема на интелигентен HART-сензор (подчинен модул), илюстрираща една възможна реализация, е показана на фиг. 2.47. Тя е реализирана на базата на три достъпни интегрални схеми – микроконтролер ADuCM360, модем AD5700 и 16-разряден DAC за работа с токов интерфейс 4-20 mA.



Фиг. 2.47

- **Протокол за обмен на данни**

Стандартът дефинира много детайлно и подробно целият протокол за обмен на данни между устройствата. Като се има предвид, че повече от 190 крупни производители предлагат както интелигентни сензори, така и програмируеми логически контролери (PLC) и помощни устройства за HART базирани полеви системи, е ясно, че тези предписания трябва стриктно да се спазват, за да се гарантира съвместимост и взаимозаменяемост между отделните устройства.

Тук протоколът за обмен на данни няма да бъде разглеждан.

- **Механични характеристики. Кулунги**

Изисквания за използване на конкретни типове кулунги и съединители не се предявяват. Като се има предвид, че подобен тип системи работят в полеви условия, често в агресивни среди (взривоопасни или огнеопасни), става ясно, че трябва

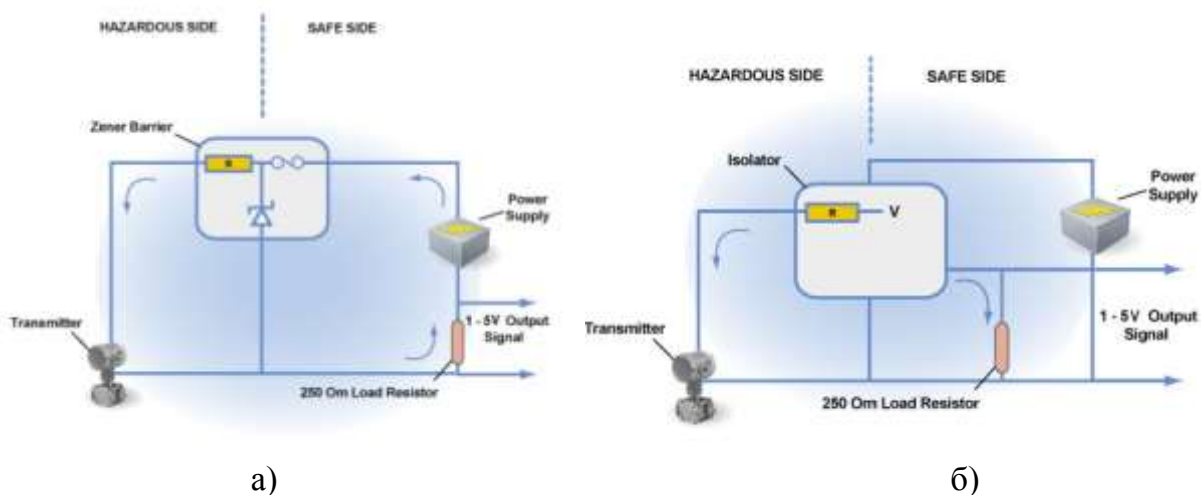
ва да се подхожда много отговорно към въпросе за присъединяване на отделните устройства.

- **Галванично разделяне на интерфейса**

Основната идея на интерфейса е устройствата да се захранват през комуникационната линия. Това оставя въпроса за галваничното разделяне на по-заден план. От друга страна голяма част от оборудването, удовлетворяващо спецификациите на HART, е експлозивен клас – т.е. е предназначено да работи във взривоопасна или пожароопасна среда. Тези приложения изискват прилагането на схемотехнични решения, които да гарантират безопасност при тяхното използване.

Първото и най-разпространено решение е поставянето на ценерови бариери на границата на двете среди – безопасната и опасната зони (вж. фиг.2.48.а). Ниската консумация на устройствата способства за по-лесно изграждане на прехода към опасната зона чрез ценерови бариери.

Понякога се налага да се използват устройства, осигуряващи пълно галванично разделяне между безопасната и защитената зони (фиг.2.48.б). При тях сигналът 4-20mA от полево устройство се пресъздава във товарната верига в безо-



Фиг. 2.48

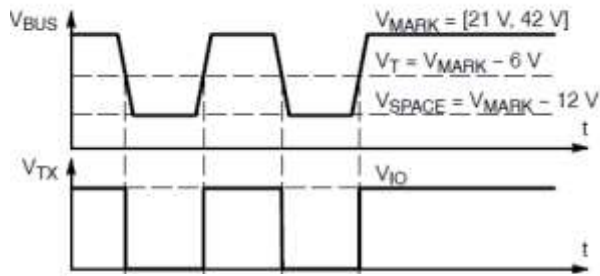
пасната зона. Трябва да се има предвид, че някои изолиращи устройства с ограничена честотна лента може да не прехвърлят сигналите от цифровия комуникационен канал.

2.2.8. Асинхронен интерфейс M-bus

Стандартът *M-bus* (*Meter Bus*) EN 13757–2 and EN 1434–3 обхваща принципите и ограниченията за физическото и комуникационното ниво при осъществяване на комуникация през усукана двойка за измервателни системи. Той е особено подходящ за термометри, разпределители на разходите за топлина, водомери и газометри. Може да се използва и за други измервателни уреди (като електромери) и за сензори и задвижващи механизми.

Като основен комуникационен интерфейс отново се използва широко разпространеният в микроконтролерите асинхронен серийен интерфейс (SCI, UART или ACIA).

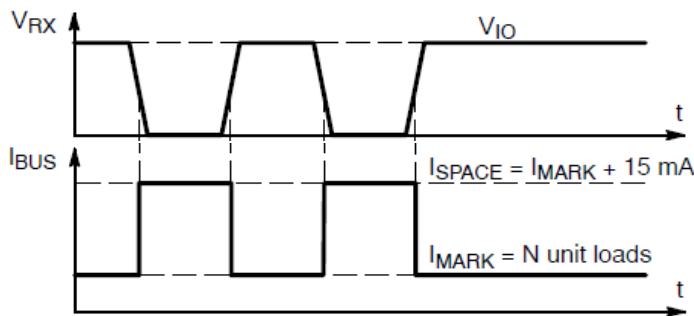
В системата са дефинирани водещ модул и подчинени модули. Комуникацията от водещия към подчинените модули се основава на промяна (понижаване) на захранващото напрежение, което се подавано по линия – вж. фиг.2.49. Обикновено линията се захранва с 21 или 42 V, като това състояние се приема за лог.“1” - V_{MARK} . За лог.“0” - V_{SPACE} - се приема ниво, с 12V по-ниско от захранващото напрежение на линията. Половината от този размах ($V_T = V_{MARK} - 6V$) се счита за нулево ниво, то се поддържа от запомнящ кондензатор – така се избягват грешки, които биха били причинени от понижаване на напрежението поради крайното съпротивление на линията при предаване от други подчинени модули.



Фиг. 2.49

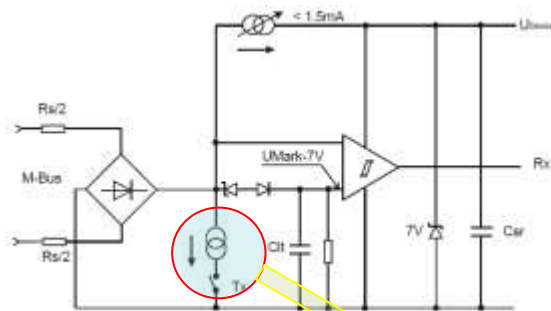
Кодирането на данните от подчинения към водещия модул става, като се променя тока, който той консумира (фиг. 2.50). За лог.“1” се приема стойността на сумарния ток (I_{MARK}), който консумират всички подчинени модули, свързани към комуникационната линия. Повишаване на този ток (I_{SPACE}) с 15 mA е указание, че в момента някой подчинен модул предава лог.“0”

Кодирането на данните от подчинения към водещия модул става, като се променя тока, който той консумира (фиг. 2.50). За лог.“1” се приема стойността на сумарния ток (I_{MARK}), който консумират всички подчинени модули, свързани към комуникационната линия. Повишаване на този ток (I_{SPACE}) с 15 mA е указание, че в момента някой подчинен модул предава лог.“0”



Фиг. 2.50

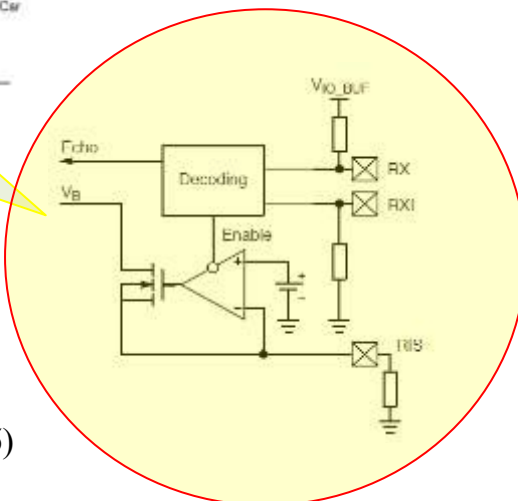
Достъпни са разнотипни трансивъри за M-bus, напр. NCN5150 и TSS721A. Обобщената функционална е представена на фиг. 2.51.а. Приемната част се реализира



а)

Фиг. 2.51

б)

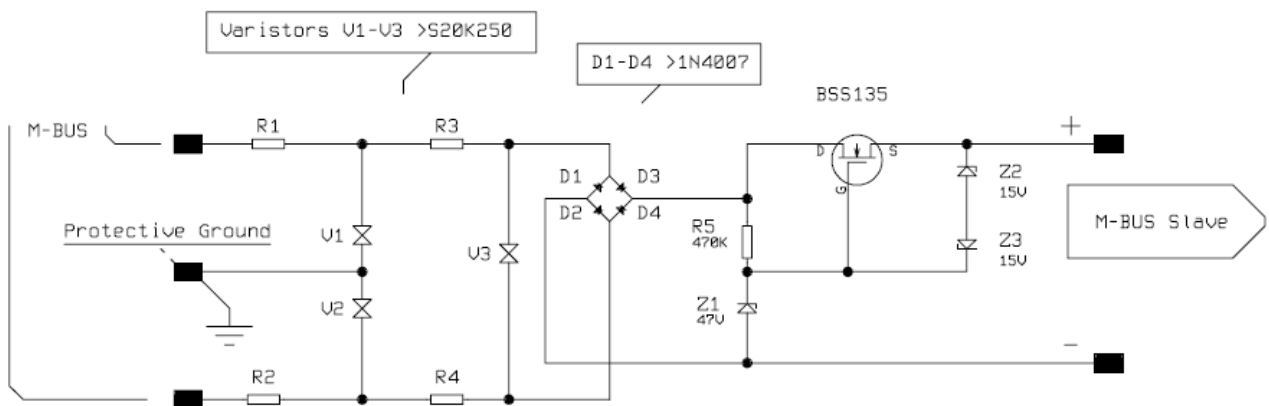


от компаратора, а логическите нива се гарантират от ценеровия диод и RC веригата C1i и паралелният му резистор (обикновено съотношението на времеконстантите заряд/разряд на C1i се подбира да бъде 1:40 – **защо?**). Стойността на кондензатора се избира така, че за времето на предаване на лог. „0“ да не се разрези и да се наруши референтното ниво.

В режим на предаване подчиненият модул използва генератор на ток, подобен на показания на фиг. 2.51.б. Той може да се управлява както от сигнала RxD, така и от неговата инверсна форма (RxDI). Стойността на комутируемия ток се настройва чрез подбор на резистора R_{IS}.

Включването на мостова схема на входа прави линията независима от полярността – т.е. двата свързващи проводника на линията равностойни, като по този начин се избягват грешки при изграждането на комуникационната среда.

За ограничаване на тока при повреда подчинените модули трябва да бъдат свързани към линията посредством два токоограничаващи резистора, най-често те са със стойност 220 Ω/0,25 W. За изграждане на надеждна ценерова бариера те могат да бъдат разделени на две (фиг. 2.52), като се използват подходящи варистори – V1 ÷ V3 – или супресори. Това свързване гарантира, че дори за продължителен период подчиненият модул може да издържи мрежови напрежения от 220V, без да възникнат необратими повреди.



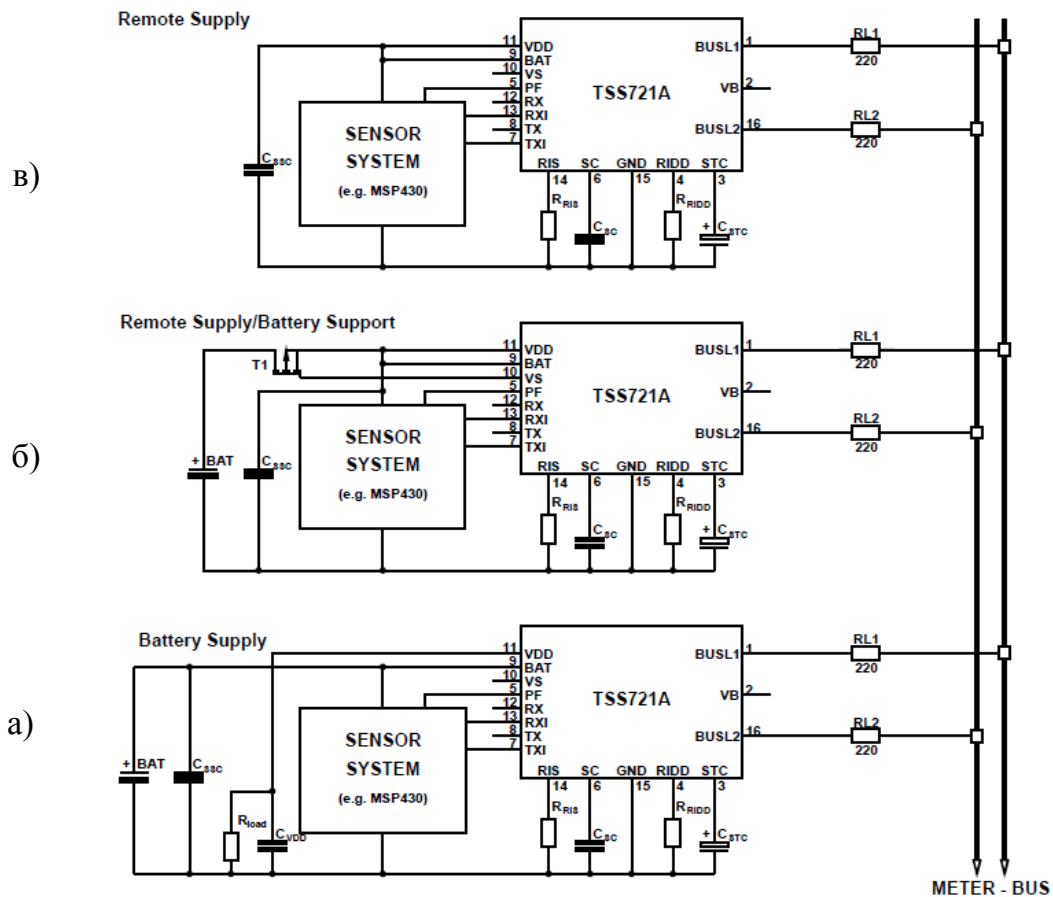
Фиг. 2.52

Стандартът M-BUS ограничава максималният ток, който подчинените устройства могат да консумират от линията, затова подчинените модули да може да се захранват от M-BUS или от автономно захранване. Съществуват няколко опции при реализиране на захранване за подчинени устройства:

1. Автономно захранване (батерия или мрежа)

Автономно захранване се налага да се изгражда в случаите, когато подчинената сензорна **система консумира ток, по-голям от разрешения** ток на интерфейсната система M-bus или трябва да се **поддържа готовност, когато шината не е активна**.

В този случай честотата на комуникация с подчинения модул може да доведе до повишаване на консумацията от локалната батерия и ограничаване на нейния живот. Сензорната система се захранва от локалното захранване, а трансмитера (напр. TSS721A) се захранва от комуникационната линия (вж. фиг. 2.53.а).



Фиг. 2.53

2. Изцяло захранване от интерфейса

Приложимо е в случаите, когато консумацията на сензорната система е в границите на разрешената ток на интерфейсната система. В този случай както сензорната система, така и трансмитера се захранват от магистралата (фиг. 2.53.в).

При тази конфигурация ограничение на интензивността на обмена с подчинения модул не се поставя. Трябва да се има предвид, че изключването на водещия модул ще доведе до отпадане на захранването и на подчинения модул и той ще престане да функционира. Така съществува опасност от загуба на данни или неизпълнение на функции по обслужване на апаратурата.

3.) Смесено захранване (локална батерия / шина)

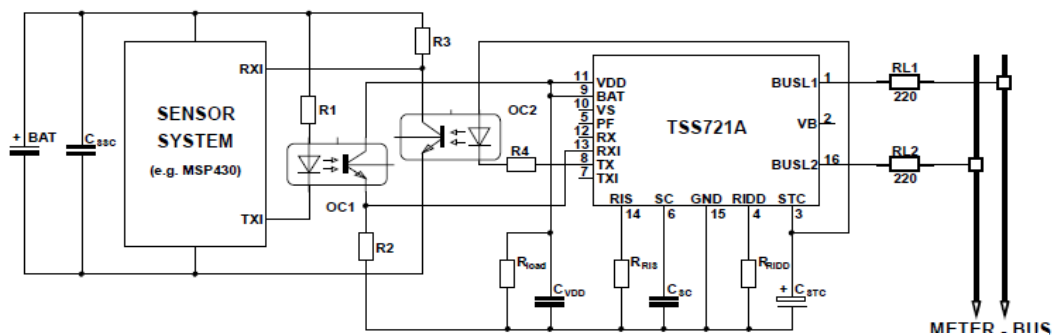
Възможно е да се изгради свързване, при което сензорната система да се захранва от локална батерия само в случаите, когато не постъпва захранващо напрежение от интерфейса (фиг. 2.53.б). Това се осъществява благодарение на ключа, реализиран с PMOS транзистора T1. В този случай отново стои изискването общата консумация да не надвишава ограниченията, наложени от системните спецификации на M-bus. Функционалността на сензорната система се запазва дори при отпадане на захранването през комуникационните линии, докато не се налагат ограничения по отношение честотата на комуникация с подчинения модул.

4.) Галванично разделяне от линията, ако подчиненият модул има връзка към земя

Системните спецификации на M-bus поставят изискване за изолационното съпротивление между всеки от комуникационните проводници на магистралата

спрямо които и да са метални части на съоръжението или земя да бъде по-голямо от 1 МΩ. В случаите, когато сензорната система има електрическа връзка към земя или към металически части на апаратурата (съоръжението) задължително се прилага галванично разделяне на сензорната система от интерфейса. Това най-често става чрез прехвърляне на сигнала с оптични методи (оптрони).

Примерната реализация на сензорна система, галванично разделена от интерфейсната част, е показана на фиг. 2.54. Тук захранването на сензорната система е локално – от батерия или с мрежов преобразувател, а захранването на трансмитера – от комуникационната линия. Галваничното разделяне на двете части се постига с оптроните OC1 и OC2. При батерийно захранване е необходимо да се ограничи честотата на обръщение към този подчинен модул, за да се обезпечи по-дълъг живот на батерията.



Фиг. 2.54

2.2.9. Предаване на данни по радиоканал

Днес много често се прилага обмен на данни между микропроцесорни системи, без да се използва проводна връзка, като се използват разнообразни средства за комуникация по радиоканал, особено на къси разстояния. За целта се използва най-често т.н ултра-късовълнов обхват (УКВ), като се прилагат модеми, използващи различни типове модулация. За честотните обхвати се прилагат лицензионни режими, като съществуват обхвати, които могат да бъдат свободно използвани, ако излъчваната мощност не превишава определена стойност.

При изграждане на връзка между микроконтролери трябва да се имат предвид някои специфични особености:

- В началото на сеанса е необходимо известно време, за да се установи както режимът на предавателя, така и приемникът да констатира наличието на носеща честота. Затова всяко излъчване започва с водещ блок, наречен „преамбюл“, по време на който не се предават данни;
- При използване на радиокомуникации е характерно, че нивото на шумовете и смущенията е високо. Затова е задължително данните да се предават на блокове, като се заложат механизми за откриване и коригиране на грешки;
- След всеки предаден блок е необходимо да се връща квитанция, декларираща дали обменът е бил успешен или неуспешен.

Напоследък се предлагат микроконтролери, които имат вградени модули за радиокомуникация – предаватели, приемници или приемо-предаватели. Те по-скоро са подходящи за използване като подчинени процесори, на чиято основа се изграждат обособени радиокомуникационни модули. Някои от тях са показани в табл. 2.5.

Таблица 2.5					
тип	предназначение	обхват	модулация		
MICRF113YM	предавател	300 – 450 MHz	ASK	MICREL	
MICRF220AY	приемник	300 – 450 MHz	ASK	MICREL	
rFRXD0420	приемник	315/433 MHz	ASK/FSK	MICROCHIP	
rFRXD0920	приемник	868/915 MHz	ASK/FSK	MICROCHIP	
ATA5428	прм./прдв.	433/886 MHz	ASK/FSK	Atmel	
CC1020	прм./прдв.	400 – 960 MHz	OOK, FSK and GFSK	TI	
CC1021	прм./прдв.	433, 868, 915, 960 MHz	OOK, FSK and GFSK	TI	

Съществува възможност да се използват готови модули за радиокомуникация, като показаните на фиг. 2.55. Най-често те са пригодени за работа с използване на NRZ формат на данните от SCI. Трябва да се отбележи, че отговорността за протокола, по който се води радиообмена, е изцяло отговорност на конструктора на системата.

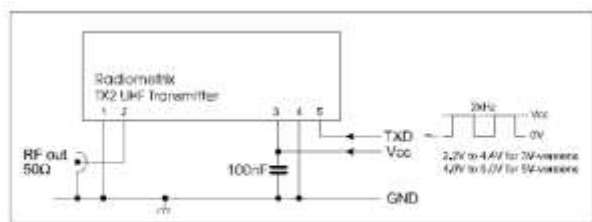
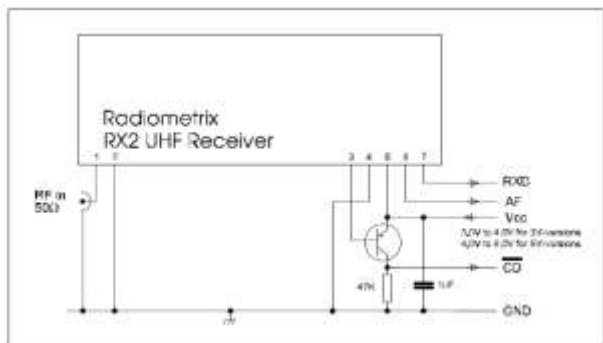


Fig.5: TX2 test circuit



left: TX2 transmitter



right:RX2 receiver

Фиг. 2.55

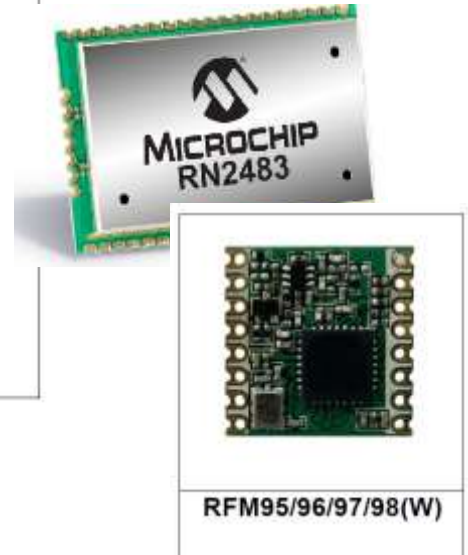
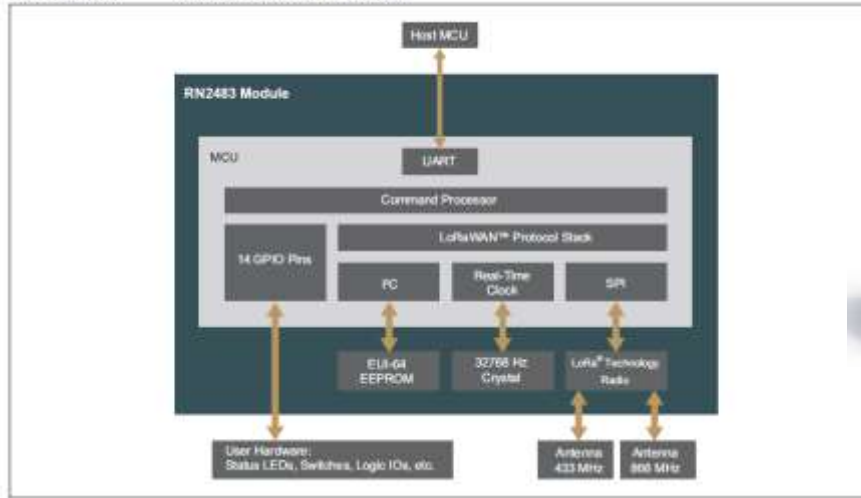
ра на системата.

В края на 70-те години на миналия век е предложен нов подход – преди модуляцията сигнала се преобразува, като „се разлива“ в много по-широк спектър. След това се прилага съответната модуляция и излъчването се извършва с много по-малка мощност. Така предаването е разпределено по цялата честотна лента на избрания канал, който е със значителна ширина. Естествено, при този подход се получава наслагане (застъпване) на каналите. Устройствата, които правилно декодират сигналите получават данните, докато други приемници „виждат“ предаванията като бял шум и не ги вземат предвид. Обратно, сигналите които проникват от припокриващите се канали за даденото устройство, имат характеристики на бял шум и се игнорират. Това позволява да се приемат и декодират множество

припокриващи се радиосигнали от различни устройства само в собствените им мрежи.

Този метод на формиране на сигналите получава развитие и днес се предлагат множество модули (вж. фиг. 2.56), които могат да излъчват в широк честотен

FIGURE 1-3: RN2483 BLOCK DIAGRAM



Фиг. 2.56

диапазон (до 500 KHz). Методът е наречен DSSS (Direct Sequence Spread Spectrum) и е популярен с търговското наименование LoRa, идващо от Low Power **Long Range**. Работят обикновено в обхвата 433 и/или 868 MHz и при излъчвана мощност 100 mW (+20 dBm) осигуряват покритие до 6 km в градска среда и до 15 km в полеви условия. Освен DSSS (LoRa™) обикновено могат да поддържат и FSK, GFSK, MSK, GMSK и OOK модуляции.

Съществуват още един клас модули за безжична връзка, които имат вграден микроконтролер с готово програмно осигуряване и поддържащи даден стандартен протокол – напр. Bluetooth, ZigBee или WiFi. Някои от тях, заедно с посъществените им характеристики, са показани в табл. 2.6.

Таблица 2.6						
тип	стандарт	интерфейс	мощност	обхват	производител	забележка
ABM-600	Bluetooth	UART, USB	+20dBm	2,4 GHz	AirLogic Co.	
RN4870/71	Bluetooth	UART	+1 dBm	2,4 GHz	Microchip	
HF-BL200A	Bluetooth	UART	+10 dBm	2,4 GHz	SHFET Co.	
XB24-BPIT	ZigBee	UART	+3dBm	2,4 GHz	DIGI INT.	
MRF24J40MB	ZigBee	SPI	+20 dBm	2,4 GHz	Microchip	
WizFi210	WiFi	UART, SPI	+8 dBm	2,4 GHz	WIZnetCo.	
HF-LPT230	WiFi	UART	+16 dBm	2,4 GHz	SHFET Co.	
ATWILC1000-MR110PB	WiFi	SPI	+18 dBm	2,4 GHz	Microchip	

При използване на такива готови модули организацията на обмена в радиомрежата е заложена от производителя в тях и не е необходимо потребителя да я разработва и залага в собственото програмно осигуряване. Те са разработени и работят в т.н. free ISM Band (обхват за свободно използване за промишлени, научни и медицински цели).

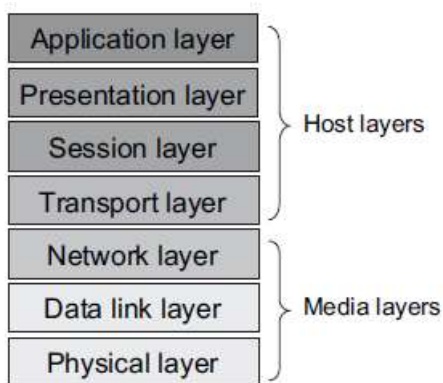
2.3. ЛОКАЛНИ КОМПЮТЪРНИ МРЕЖИ НА БАЗАТА НА АСИНХРОННИ ИНТЕРФЕЙСИ

OSI (Open Systems Interconnection Basic Reference Model) е теоретичен модел, описващ принципно начина на комуникация и строежа на телекомуникационните и компютърните мрежи. Като основна градивна единица се използват така наречените слоеве – всеки слой предоставя интерфейс и услуги към по-горния слой, като в същото време получава услуги от слоя под него.

Информацията, изпращана по мрежата, е във вид на данни или пакети от данни. Данните първо трябва да бъдат снабдени със служебна информация относно транспорта им (идентификатори за предавател и приемник) и пакетирани (капсулирани). Информацията се придвижва от предавателя към приемника, като при преминаване през отделните нива (слоеве) пакетите претърпяват промяна вследствие на работата и функциите на системата. В приемника от пакетите се премахва служебната информация, прибавена за целите на транспорта при предавателя.

OSI моделът е концептуален и съдържа описания за различните функции и операции, необходими за обмен на данни. Тези описания са групирани логически в отделни етапи (слоеве), като всеки слой съдържа близки по същност и замисъл операции, общо представяне на данните и има относителна функционална независимост от другите слоеве. Макар практическата реализация на мрежовите комуникации често да нарушава тези граници или да обединява съседни слоеве, тя винаги запазва логическата последователност и зависимости.

Във всеки един от слоевете в OSI модела (освен в първия и последния) се извършва групиране (пакетиране) на данните на порции, при което се образуват два или повече пакета. При пакетирането всеки слой добавя своята служебна информация - поредност, код за грешка, адреси, номер на услугата и други параметри в зависимост от функционалните изисквания на даденото ниво. При предаване и приемане на информация между източника и приемника строго се спазват определени правила - протокол - който представлява сбор от правила, определящи формата на данните и начина на предаване и приемане за съответния слой. За транспортирането са отговорни протоколите от по-долните нива. Обикновено протоколът не засяга повече от един или два слоя на OSI модела, т.е. предназначен е за комуникация вътре в слоя или между съседни слоеве. Това позволява оперативна съвместимост и преносимост на протоколите в различни преносни



Фиг. 2.57

среди (допустими са различни реализации на по-долните слоеве). Свързването на протоколи от различни слоеве става с помощта на предварително дефинирани интерфейси като всеки по-долен слой ограничава и определя възможностите за предаване на информация на по-горните.

Стандартът дефинира 7 слоя, които са илюстрирани на фиг. 2.57, а предназначението на всеки слой се свежда до:

– **Приложният слой (Application layer)**

е най-горният слой от OSI модела, който се отнася за приложения (програми) като интернет браузъри, мениджъри за отдалечено управление, клиенти за обмен на съобщения, HTTP-, FTP-, DNS- сървъри и др. Този слой позволява на потребителските приложения да заявяват услуги или информация, а на сървър приложенията – да се регистрират и предоставят услуги в мрежата. В този слой работят командните интерпретатори на разнотипните контролери или измерителни станции:

– **Представителният слой** (*Presentation layer*) се грижи за представяне на данните във вид, разбираем за получателя като осигурява общия им формат за различни платформи. Извършва конвертиране и “превеждане” на данните, компресиране/декомпресиране, както и криптиране/декриптиране на информацията.

– **Сесийният слой** (*Session layer*) управлява създаването и прекъсването на сесиите (диалога) между представителните слоеве на две или повече системи. Контролира и диалога между две приложения на различни сървъри и управлява потоците от данни посредством точки на синхронизация, което включва поставянето на маркери в потока от данни. При некоректно предаване на информацията или при прекратяване на връзката в мрежата, данните трябва да бъдат предадени само в частта им след последния коректно предаден маркер.

– **Транспортният слой** (*Transport layer*) осигурява комуникация от край до край (end-to-end) между процеси, изпълнявани на системи. Предоставя на по-горните слоеве услуги с или без установяване на в зависимост от застъпените протоколи. Слойът използва адресите на клиентските и сървърните портове, за да идентифицира различните процеси. Извършва сегментиране на данните, подадени му от сесийния слой, с цел правилното им подреждане от получателя като определя пореден номер за всеки сегмент.

– **Мрежовият слой** (*Network layer*) има за основна цел да задава логически адреси на източника и местоназначението, както и да определя най-добрия път за предаване на данните. За постигане на оптимална маршрутизация устройствата от мрежовия слой (маршрутизаторите) използват превключване на пакети (packet switching), което накратко се състои в това, че четейки адреса на логическото местоназначение, получен от един интерфейс, маршрутизаторът изпраща трафика (пакетите) чрез друг интерфейс към местоназначението.

– **Каналният слой** (*Data Link Layer*) има за цел да предава и да приема кадри (*frames*), а също така отговаря за тяхното физическо адресиране. Преди предаването на всеки пакет тук се прибавят заглавен символ (хедър) – в началото – и ремарке (трейлър) – в края, превръщайки го по този начин в кадър. Хедъри към данните прибавят и мрежовият и транспортният слоеве, но трейлърът се добавя само на ниво Data-link. Каналният слой се разделя на два подслоя, LLC и MAC, като първият добавя още контролна информация, служеща за правилното транспортиране на данните, а вторият осигурява достъп до преносната среда (медията). Някои от основните характеристики на слоя са контрол на достъпа до преносната среда, добавяне на хардуерни (MAC) адреси на източника и местоназначението, подготвяне на пакетите за преда-

ване (чрез капсулирането им в кадри, описано по-горе), изчисляване на контролната сума на кадъра (FCS), както и кодирането на сигналите в преносната среда. При предаване на данни изпращащите устройства изпълняват алгоритъм за циклична проверка с остатък (контролен цикличен код – CRC или контролна сума – CS) или контрол на последователността на данните (FCS), като записват получената стойност в трейлъра на кадъра. Станцията-получател изпълнява същия алгоритъм и сверява стойностите, като по този начин се установява дали кадърът се е повредил по време на преноса.

– **Физическият слой** (*Physical layer*) е най-долният слой от модела и работи само с единици и нули (битове), изграждащи кадъра. Основната му функция е да предава битове по комуникационна линия. Битовете са кодирани като физически състояния на преносната среда. При жичните кабели това са електрически сигнали, при оптичните влакна – светлинни импулси, а при безжичните връзки – електромагнитни вълни. В този слой имат значение електрическите и механичните характеристики, които определят представянето и формирането на сигналите. Към този слой има спецификации за отделните типове преносна среда и конектори, като форма, структура и състав.

При дадена реализация не е задължително да присъстват всички слоеве, дефинирани в OSI модела. Например, в рутерите са вградени само трите най-долни слоя – физически, канален и мрежов. На фиг.2.58 напр. е представен мрежовият

Layer	Function	HART	
7	Application	provides formatted data	HART instructions
6	Presentation	converts data	
5	Session	handles the dialogue	
4	Transport	secures the transport connection	
3	Network	establishes network connections	
2	Link	establishes the data link connection	HART protocol regulations
1	Physical	connects the equipment	Bell 202

ТАБЛИЦА 2.7

Модель OSI HART-протокола

Номер уровня	Название уровня	HART
7	Прикладной	HART-команды, ответы, типы данных
6	Уровень представления	Нет
5	Сеансовый	Нет
4	Транспортный	Нет
3	Сетевой	Нет
2	Канальный (передачи данных)	Ведущий/ведомый, контрольная сумма, контроль четности, организация потока битов в сообщении, контроль приема сообщений
1	Физический	Наложение цифрового ЧМ сигнала на аналоговый 4...20 мА; медная витая пара

Фиг. 2.58

модел, заложен в разгледания полеви интерфейс HART.

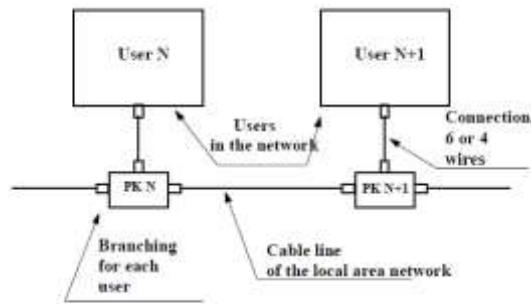
- **Малка промишлена мрежа от контролери**

Изискванията към локалните мрежи за промишлено приложение могат да бъдат сведени до:

- Сравнително бавна скорост на предаване на данни, тъй като интензивността на потоците от данни не е голяма (програми и файлове не се предават през тези мрежи).
- Галванично разделяне на всеки контролер.
- Високо ниво на електрически и електромагнитни смущения в промишлени условия, което налага да се вземат изключителни мерки за стабилност - хардуерни и софтуерни.
- Минимална цена на хардуера за локалната мрежа.

- **Структура** на мрежата

Структурата на малка локална мрежа (фиг.2.59) включва комуникационна среда (магистрала - най-често 1 или 2 усукани двойки), разклонения от магистралата, къси връзки към отделните абонати и самите устройства – промишлени кон-

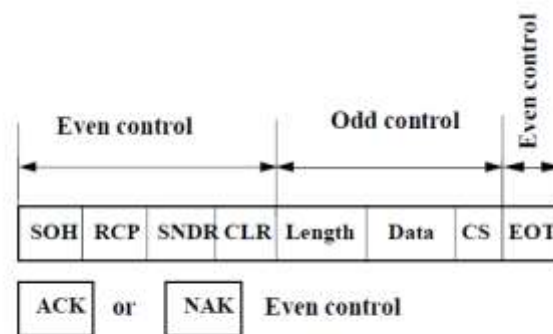


Фиг. 2.59

тролери, електронни измерителни устройства и пр. На всяко от присъединените устройства се присвоява уникален логически номер N_i . Заедно със серийните комуникационни интерфейси в отделните абонати, тези компоненти изграждат първото ниво – физическият слой на мрежата.

- **Структура** на данните потоци

Поради шумовете от околната среда данните трябва да бъдат разделени на сравнително кратки блокове. Пакетът данни (фиг.2.60) на локалната мрежа може



Фиг. 2.60

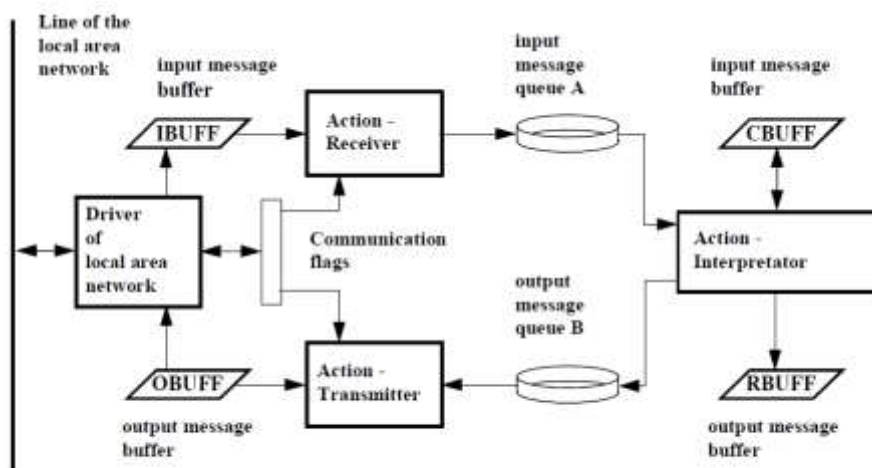
да има следната структура:

- **Водещ символ – SON or STX**, 1 байт. Той показва началото на предаването на блока данни. Има поне два типове водещи символи - начало на предаване на съобщение (SOH) и начало на блок данни предаване (STX). Първият символ се намира в началото на съобщението. Вторият е разположен в началото на следващите блокове на съобщението;
- **Адрес на приемник - RCP**, 1 байт. Той посочва адреса на получателя на съобщението. Размерът му 1 този случай ограничава брой потребители в локалната мрежа до 256;
- **Адрес на предавател - SNDR**, 1 байт. Номер, посочващ адреса на подателя на съобщението;
- **Цвят на блока - CLR**, 1 байт. Той съдържа 0 или FF₁₆. Използва се при повторение на блокове при възникване на грешки.
- **Дължина на блока - Lenght**, 1 байт. Той посочва броя на байтовете в информационната част. Обикновено дължината е ограничена до 16 или 32 байта и рядко надвишава 128.
- **Информативна част - DATA**, от 1 до 255 байта. Това са команда или данните.
- **Контролна сума - CS**, 1 байт. Това е сумата по модул 256 на блока данни (заедно с байта за дължина на блока), представена в допълнителен код. Използва се за проверка верността на получаване на блока.
- **Краен символ**, 1 байт. Той показва края на предаването на блока данни. Има два вида на крайните символи - край на предаване на съобщение (EOT) и край на предаване на блок данни (ETX). EOT се намира в края на последния блок на съобщението. ETX разположен в края на първия и следващите блокове (с изключение на последния блок) на многоредово съобщение.

За да има разлика между системната и информативната част, системната информация се предава с контрол по четност, а байтовете на информационната част са предадени с нечетен контрол.

• **Алгоритмично осигуряване**

Една възможна обобщена структура на организацията на програмното осигу-



Фиг. 2.61

ряване, реализиращо обмена на данни в малка локална мрежа, е показано на фиг. 2.61.

Тук драйверът за мрежата, приемникът и предавателят могат да се отнесат към втория слой на обобщения мрежов модел (каналния слой). Драйверът се изпълнява в прекъсване, генерирано от серийният комуникационен интерфейс в зависимост от възникващите събития (постъпил символ, празен буфер и пр.). Кодовете на приемника и предавателя се изпълняват в основната програма или в съответния таск при работа под управление на операционна система за реално време. Прехвърлянето на данни между приемника/предавателя и драйвера става чрез използване на входен/изходен буфер за съобщения, организирани в данновата памет на системата. Активирането им става чрез флагове, намиращи се в съответните регистри за състояние.

Приемникът и предавателя използват опашки, организирани като кръгови буфери, за да предоставят съобщенията към/о горното ниво – командния интерпретатор. Той е позициониран в приложния слой от обобщения OSI модел. Останалите слоеве от модела в разглеждания случай не се поддържат.

- **Арбитражиране** на достъпа до преносната среда:

При изграждане на мрежовата среда от основно значение е дали в нея ще има само един водещ (главен) модул, или се допуска работа с множество водещи модули. Във втория случай вероятността за настъпване на колизии (едновременно използване на преносната среда от няколко модула) многократно се увеличава. В този случай е необходимо да се предвиди процедура, която да гарантира устойчиво преодоляване на това противоречие.

Предполага се, че всеки модул „прослушва“ символите, които предава, и може да сравни данните в комуникационната среда и изходния поток (условието не е изпълнимо в радиомрежи!). От това следва, че решение може да се вземе, едва след като изтече времето за предаване на един символ – т.е. зависи от скоростта на предаване. При това предположение могат да се дефинират няколко състояния и действията, които са свързани с тях:

- Заемане на магистралата:

Заемането на линията за връзка (която се явява основен ресурс на мрежовата среда) започва в момента на предаване на първия водещ символ на рамката или съобщението (поредицата от служебна информация и данни). След приключване предаването на водещия символ всеки абонат в системата има информация, че линията е заета – т.е. има едно латентно време, равно на времето за предаване на един символ, от действителното заемане на линията до установяването на този факт от всички абонати. Това е критичното време, през което втори модул би могъл да инициира обмен, т.к. все още не е установил факта на заемане на магистралата;

- Установяване на колизия:

Ако втори модул започне обмен по време, докато се предава водещият символ, приетите данни във водещите модули ще се различават от предадените. Това е указание за модулите, започнали едновременно (в рамките на времето за предаване на един байт), че е настъпило застъпване на инициирането на обмен;

– Принудително прекъсване на сеанса за връзка:

Модулът (модулите), установил наличието на колизия, е длъжен да преустанови започнатата сесия за връзка и да уведоми всички останали абонати в мрежата за възникване на това състояние. Това става чрез предаване на т.н. състояние „BREAK“ – задържане на линията на нулево ниво за време, по-голямо от времето за предаване на един байт. Във всеки приемен модул ще възникне форматна грешка и той ще бъде известен за възникналата ситуация;

– Арбитражно повторение на достъп до магистралата:

След приемането на сигнала „BREAK“ всеки абонат има право да заеме линията не по-рано от изтичане на време, равно на неговия уникален номер N_i , умножен по времето за предаване на един символ, **ако през това време линията вече не е заета**. Ясно е, че устройствата с по-малък идентификационен номер N_i ще имат по-висок приоритет в така изградената мрежа.

2.4. СИНХРОННИ СЕРИЙНИ ИНТЕРФЕЙСИ

При пренасяне на данни в сериен формат основен въпрос е да се идентифицират моментите, в които се пренасят отделните битове и байтове - т.е къде е началото и края на отделния бит и съответно къде започва и къде завършва предаването на цял байт. Най-простият начин да се постигне това е данните да бъдат съпроводени от тактов сигнал, всеки период от който дефинира предаването на един бит. Явно е, че са необходими минимум две линии – една за данните и една за тактовия сигнал. Допустимо е предаване на комплексен сигнал – данни и синхронизиращ сигнал, смесени по определени правила. Прилагането на този принцип е залегнал в изграждането на „синхронна“ серийна комуникация на данни.

За по-лесно определяне началото и края на цяла дума (най-често с дължина 8 бита – байт) данните се пакетират в определен формат. За да се осигури съгласуваността комуникация е необходимо да се следва определен набор от правила, които гарантират синхронизирането и форматирането на данните. Тези набори от правила се наричат „**протоколи**“ и са основополагащи при серийната комуникация.

Устройствата, извеждащи данни към серийната връзка, обикновено се наричат „**предавател**“, а устройствата, получаващи данните – „**приемник**“. Често всяко устройство, присъединено към серийна магистрала, се нарича „**възел**“. Устройствата, които инициират обмена, се наричат „**главни**“ или „**водещи**“ модели. Те най-често изработват тактовите (синхронизиращи) сигнали. Устройствата, с които се извършва обмен, е прието да се наричат „**подчинени**“ подули.

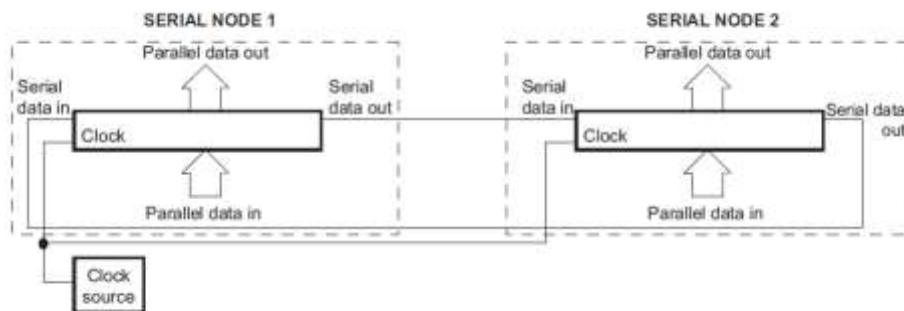
Обикновено в системите със серийна връзка има един главен и един или няколко подчинени модула. Понякога се изграждат системи, в които може да има повече от един главен модул. Характерно е, че в даден момент **само** един модул може да изпълнява ролята на главен. В такива случаи в протокола са залегнали правила, дефиниращи процедурата, по която става заемането (арбитражното) на комуникационната среда и начините за разрешаване на конфликтите (колизиите по линията).

Тук ще бъдат разгледани два от най-разпространените от тях – серийният периферен интерфейс (**Serial Peripheral Interface – SPI** и почти идентичния с него

Microwire) и (*I²C - Inter-Integrated Circuit*). Основна тяхна особеност е, че са предназначени да осигурят серийна връзка между микроконтролера и периферни схеми в рамките на една система и не се преследва връзка между отделни системи.

2.4.1. Серийен периферен интерфейс SPI

Серийният периферен интерфейс SPI е най-простата възможна серийна връзка между водещи и подчинени модули, без вградени в протокола възможности за адресация. Изгражда се на базата на преместващи регистри, по един съответно във водещия модул и един в подчинения, които позволяват паралелно да бъдат четени или записвани данни от/в тях (фиг. 2.62). Задължително двата регистра трябва да бъдат тактувани от общ тактов генератор. Всеки тактов импулс премес-

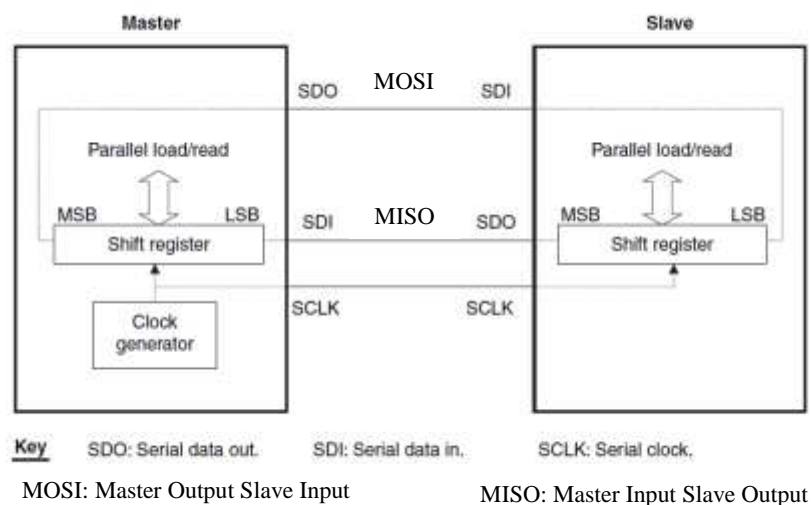


Фиг. 2.62

тва информацията в тях на 1 позиция. След постъпване на 8 такта данните, записани в съответния регистър, ще бъдат предадени в серийен формат. Съответно на тяхно място в преместващия регистър ще бъде записана входната последователност, като ще бъде преобразувана в паралелен вид и достъпна по вътрешната паралелна магистрала.

- **Сигнали:**

Интерфейсните сигнали включват входните и изходни сигнали на преместващите регистри, както и общия тактов сигнал (вж. фиг. 2.63):

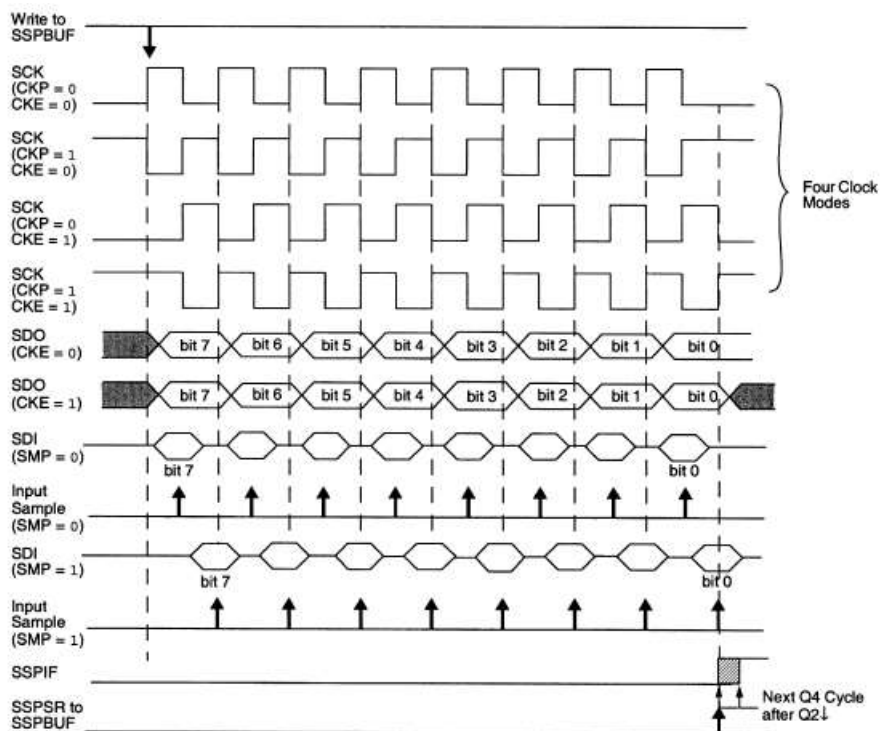


Фиг. 2.63

- **MOSI, SDO** – това е серийният изход на водещия модул, който едновременно с това се явява сериен вход на подчинения модул (Master Output Slave Input). При някои системи, както е Microware не се акцентира на това, дали модула е водещ или подчинен, тогава серийният изход се именува SDO – Serial Data Output;
- **MISO, SDI** – това е серийният вход на водещия модул, който е и сериен изход на подчинения модул (Master Input Slave Output). Алтернативното название на сигнала е SDI – Serial Data Input. Следва да се отбележи, че SDO и SDI на водещия и подчинения модул се свързват „на кръст“;
- **CLK, SCLK** – общ тактов сигнал, който се изработва винаги от водещия модул. Този сигнал е с честота обикновено от 100 KHz до 2-5 MHz, но при някои системи може да достигне и до 40 MHz. По принцип не съществува изискване, което налага да бъде симетричен, дори не се предявяват изисквания относно неговата стабилност;
- **SS**. Това е сигнал за избор на подчинен модул. В случаите, когато в системата има няколко подчинени модула, за всеки от тях трябва да се изработи индивидуален сигнал за избор. Той затваря рамката, в която става предаването на данни и пренася данните от преместващия регистър в паралелните регистри за данни (в противен случай състоянията на отделните разряди биха се сменяли, докато се извършва самият трансфер).

• **Режими (моди) на интерфейса:**

Съществуват 4 режима (моди) на работа на интерфейса (фиг. 2.64) – с положителни или отрицателни тактови импулси и с тактови импулси, изместени на



Фиг. 2.64

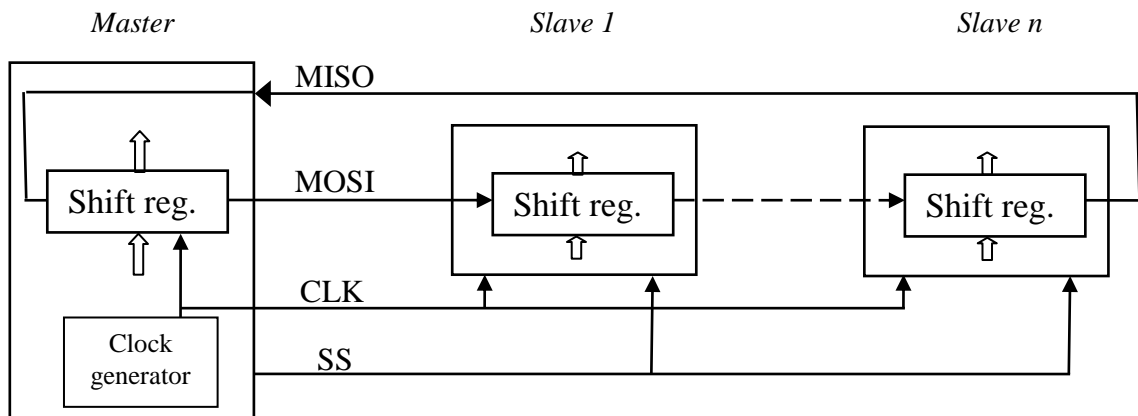
0 или 180° спрямо предаването на данните. Разликата между тях се заключава в моментите на предаване на нови данни и стробиране на данните спрямо текущата фаза и фронт на тактовия сигнал. Режимите на работа на главните и подчинените устройства трябва да бъдат съгласувани, за да се осъществи сигурен обмен на данни.

Трябва да се отбележи, че предаването на всеки байт започва със старшия бит (направете разлика спрямо SCI).

• Схеми на свързване

Когато към даден микроконтролер трябва да бъде свързан само един подчинен модул, не възникват ограничения и структурата наподобява тази, показана на фиг. 2.63. Ако се налага да има присъединени няколко подчинени модула, тогава са възможни три различни подхода при изграждане на системата:

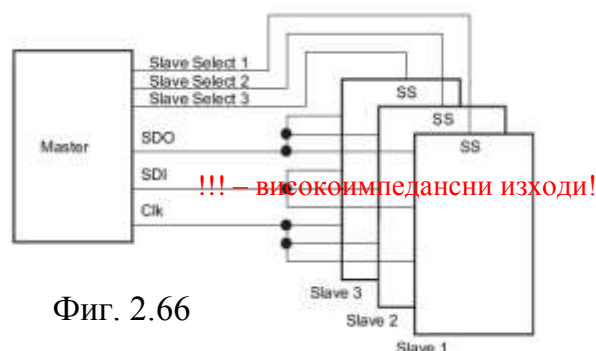
- **Последователно свързване на подчинените модули** (фиг. 2.65). Прилага се само ако подчинените модули са еднотипни и съдържат по един преместващ регистър, чиито вход/изход са изведени на корпуса на схемата. Трябва да се държи сметка, че е необходимо да бъдат предаде-



Фиг. 2.65

ни толкова байта, колкото подчинени модули са присъединени в системата. Първият байт в поредицата е предназначен за най-отдалечения подчинен модул, първият постъпил байт също е от него. Последователността на предаваните байтове се определя от отдалечеността (логически) на подчинените модули от водещия модул;

- **Паралелно свързване на подчинените модули** е показано на фиг. 2.66. Схемното решение се прилага в случаите, когато се работи с



Фиг. 2.66

разнотипни периферни схеми, на които вътрешната структура е по-сложна. В този случай е необходимо сигнала за избор да се мултиплицира в зависимост от броя на присъединените подчинени модули. Вече не съществува изискване те да са еднотипни, но изходите им **SDO** задължително трябва да са с високоимпедансно състояние и да се разрешават, само когато съответната схема получава сигнал за избор **SS_n**.

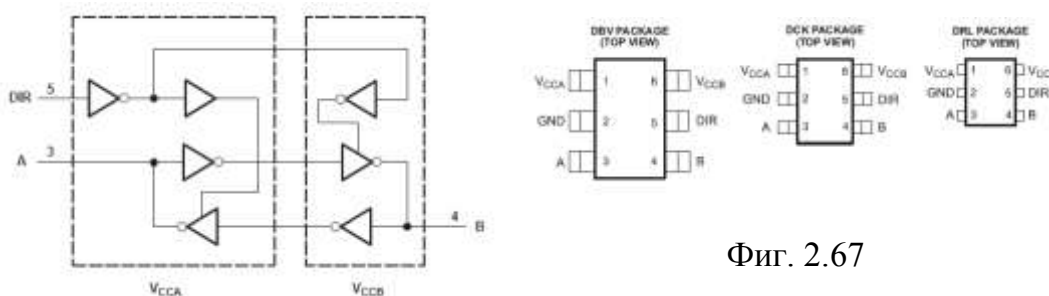
– **Смесено свързване на подчинените модули.** Допустимо е в някои от паралелните контури да бъдат свързани последователно интерфейсни схеми, например преместващи регистри. В този случай е необходимо да се спазват изискванията, разгледани по-горе за двата случая. Ако използваните преместващи регистри не притежават изходи с управление на състоянието е необходимо да бъдат използвани допълнителни буфери.

- **използване на интерфейса**

SPI интерфейса се използва основно за разширение на системите, най-вече в случаите, когато възникне необходимост от ресурси (периферни схеми), които не са вградени в използвания микроконтролер. Незаменим когато достъпа до външни ресурси трябва да става със средна до висока скорост. От друга страна не е удачно да се присъединяват външни за системата модули. Това го позиционира като серийен вътрешносистемен интерфейс за присъединяване на допълнителни памет и интерфейсни схеми.

- **схеми с различни логически нива**

Често се случва микроконтролерът да работи при различно напрежение, отличаващо се от напрежението на периферните схеми, които изграждат подчинения модул. В тези случаи могат да се използват транслатори на нива, изпълнени като двупосочни буфери – напр. **74LCV1T45DBVR** (фиг. 2.67). Трябва да се отбележи, че схемата работи с две различаващи напрежения – **V_{CCA}** и **V_{CCB}**, които определят логическите нива на двете страни. Характерна особеност е, че между двете страни съществува галваническа връзка. Такива схеми трябва да се сложат на четирите сигнала, които изграждат интерфейса – **SDO**, **SDI**, **SCLK** и **SS**.



Фиг. 2.67

- **Галванично разделяне на интерфейса**

SPI е позициониран като вътрешносистемен интерфейс, което предполага оптимизация по бързодействие и простота на изграждане на системата. Независимо от това понякога е желателно (или дори задължително) ядрото на системата да бъде галванично разделено от периферната част, където е възможно да възникват силни електромагнитни смущения или да се работи при по-високи напрежения. В тези случаи е възможно да се приложи галванично разделяне за всички сигнали на

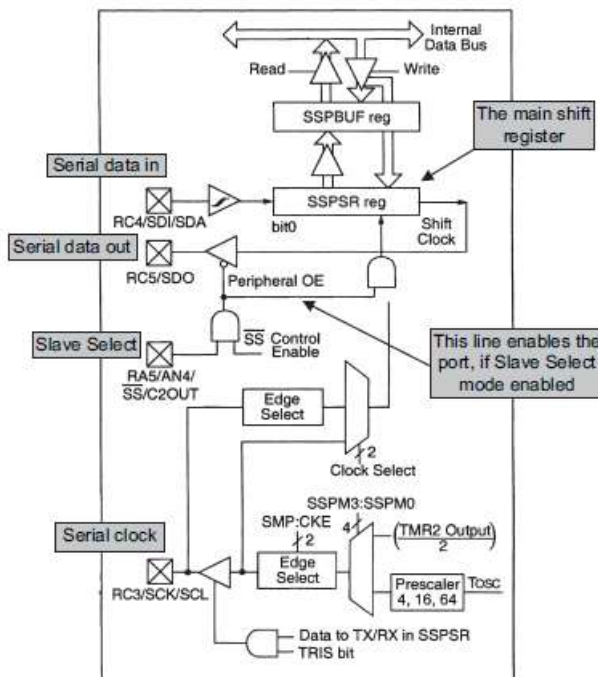
интерфейса. При това е необходимо да се използват бързодействащи оптрони, при които оптодвойката е на базата на светодиода – фотодиод. При тях се постигат бързодействия от порядъка на 1 – 10 МВ/с, при което до сериозно забавяне на интерфейса не се достига. Подходщи са оптроните от сериите 6N136, TLP2362, HCPL2630 и др.

- **Примери - SPI системи при някои микроконтролери:**

Вече болшинството микроконтролери имат вградени интерфейси за обслужване на SPI. В някои от тях има вградени по две такива системи, за да могат да се вграждат отделни управляващи или измервателни контури. Обикновено има възможност да бъдат инициализирани за работа както като водещи, така и като подчинени модули.

- SPI при микроконтролери от фамилията PIC18Fх...

Вътрешната структура на серийния периферен интерфейс е показана на фиг. 2.68. Интерфейсът е изграден около преместващия регистър SSPSR. Връзката му към вътрешносхемната магистрала на микроконтролера става чрез паралелен буферен регистър SSPBUF.



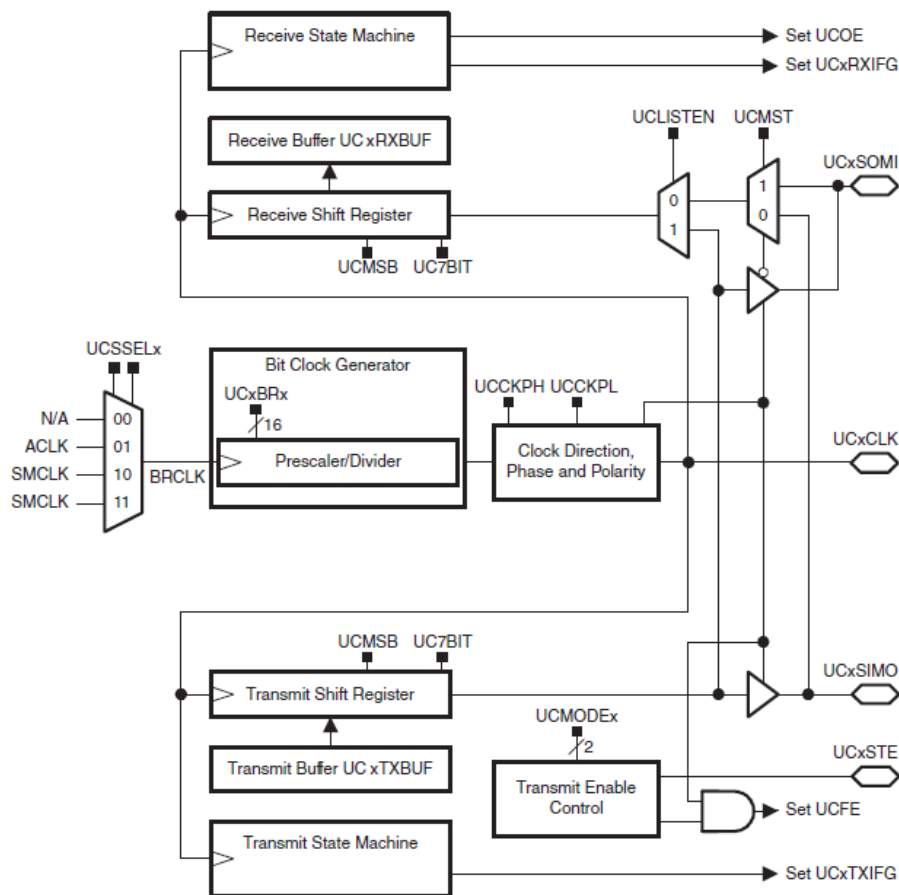
Фиг. 2.68

Когато интерфейса се използва като водещ модул, сигналът SS се генерира програмно – бит 5 от порт А (RA5) се инициализира като изход и се установява от драйверната програма. Когато е инициализиран като подчинен модул, този изход е вход и неговото ниво разрешава работата на изходния буфер за сигнала SDO. В подчинен режим тактовия сигнал постъпва от външното устройство, като може да бъде указана неговата полярност. В режим на водещ модул се използва или честотата на основния генератор, разделена на 4, 16 или 64, или сигнала от таймер 2. Това означава, че скоростта на обмен може да се настройва в много широки граници.

- SPI при микроконтролери от фамилията MSP430...

Универсалният серийен комуникационен интерфейс (USCI_A) тип А на микроконтролерите от фамилията MSP430 на TI (фиг. 2.69) може да се инициализира както за работа като водещ модул, така и като подчинен модул в SPI режим.

Характерни особености при него са, че може да работи със 7 или 8 битова дължина на думата, както и да се променя водещия бит – младшия или старшия. Притежава два преместващи регистъра, от които единият се използва в режим на предавател, а другия – в режим на приемник. Вграденият тактов генератор позволява да се поддържат и четирите режима (фаза и полярност) на интерфейса по отношение на тактовите сигнали.



Фиг. 2.69

– SPI при микроконтролери от фамилията MC68HC11

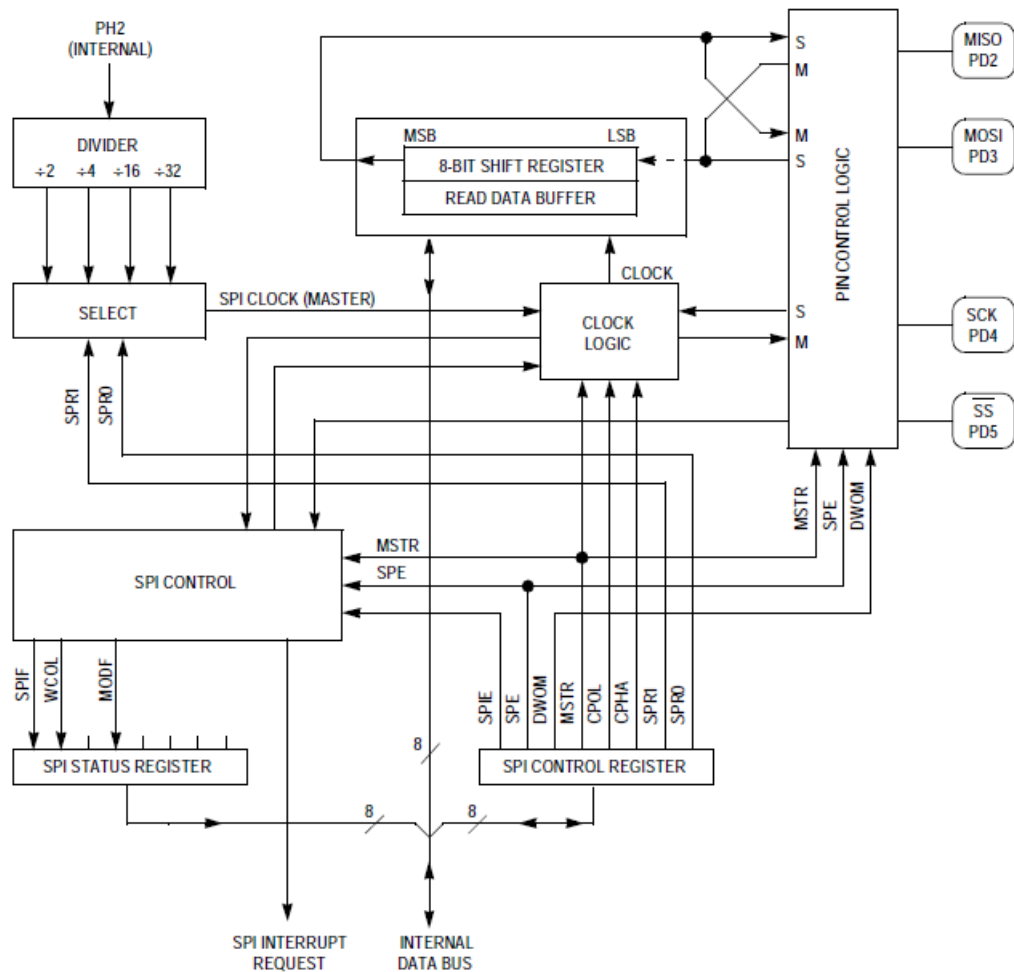
Серийният периферен интерфейс SPI също е предвиден да работи както главен или подчинен модул. Когато системата е конфигурирана като водещ модул е допустима скорост на обмен до 1 Mbit/s. При работа като подчинено устройство интерфейсет позволява работа със скорост до 2 Mbit/s.

Основен елемент също е преместващ регистър, който позволява сериен запис и четене на данни – фиг. 2.70. Системата е единично буферирана при предаване на данни и двойно буферирана (чрез READ DATA BUFFER) при приемане. Това означава, че нови данни за предаване не могат да бъдат записани в регистъра, докато предишната транзакция не бъде завършена. Получените данни в обаче се прехвърлят в паралелния буфер за четене на данни, така че преместващия регистър е готов да приема следващия байт. Входа и изхода на преместващия регистър автоматично се превключват към изводите MOSI и MISO при смяна типа на модула – главен или подчинен.

Интерфейсет позволява смяна на полярността и фазата на тактовия сигнал CLK както при работа като главен, така и като подчинен модул. Това гарантира успешна работа с повечето налични синхронни периферни устройства. При работа като водещ модул може да бъде избрана една от 4 възможни скорости за обмен на данни.

Вградената система за откриване на грешки следи за възникване на колизия на изходните данни и гарантира автоматично изключване на схемата от линията.

Така се създават предпоставки за организиране на междупроцесорна комуникация в системи с няколко водещи модули.



Фиг. 2.70

– програмен – *SPI*

При микроконтролери, в които няма вграден SPI, такъв може да се организира по програмен път, като се използват 4 разряда от паралелен порт в системата. В този случай скоростите на обмен, които могат да се реализират, са по-ниски. При работа в подчинен режим е необходимо да се използват възможностите на системата за прекъсване на съответния микроконтролер, като се следи възникването на сигнала за избор SS. Таймингът на системата (както на подчинения, така и на водещия модули) трябва да бъде много прецизно съгласуван, за да не се допусне пропускане на битове.

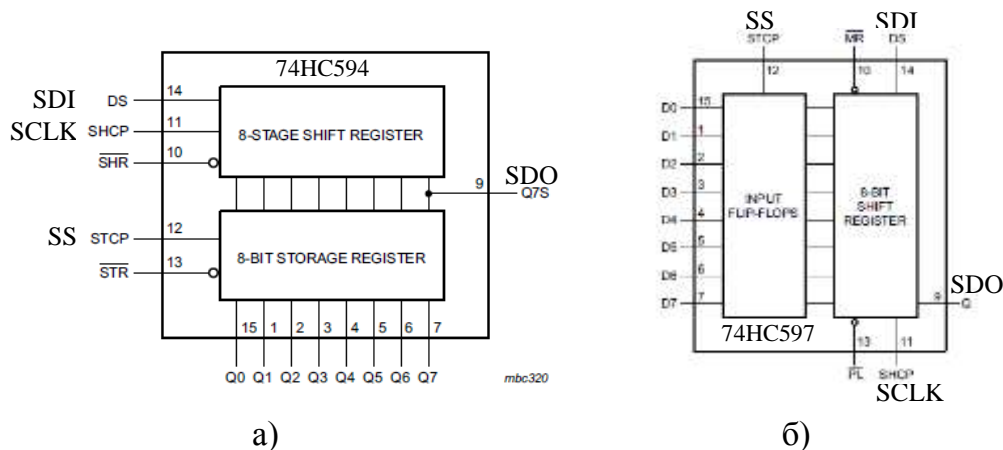
• Видове интерфейсни схеми:

Освен между микроконтролери, SPI може да се използва за връзка с най-разнообразни периферни схеми. По-долу като пример са посочени само някои от тях:

– Преместващи регистри

Най-простите схеми, които могат да се присъединят към SPI, са преместващи регистри – напр. 74HC594 (8 разряден преместващ регистър с паралелен изходен регистър – фиг. 2.71.а) и 74HC697 (8 разряден преместващ регистър с паралелен

входен регистър – фиг. 2.71.б). Те могат да бъдат използвани например за управление на отделни светодиоди или за четене състоянието на бутони и сензори.



фиг. 2.71

Изходите на тези схеми не могат да бъдат приведени във високоимпедансно състояние. Това ги прави неподходящи за изграждане на паралелни контури. От друга страна семплата им структура позволява да се изграждат последователни вериги.

– *Разширители за паралелен интерфейс*

Много често вградените в микроконтролера паралелни портове не достигат, за да се обхванат всички сигнали в системата. За тази цел са разработени т.н. разширители, връзката с които се осъществява посредством серийната вътрешносистемна магистрала SPI. Пример а такава схема е MCP23S17 – дву-портов разширител, който позволява към системата да бъдат присъединени 16 конфигурируеми входове или изходи.

– *EEPROM и FERROELECTRIC RAM*

SPI с присъщото му бързодействие е особено подходящ за присъединяване към системата на допълнителна памет. Много често се използват енергонезависими паметии от типа EEPROM – напр. 25LCxx, които се предлагат с обем от 128 байта до 128 килобайта. Поради необходимостта от изчакване между записа на два блока (обикновено с големина под 64 байта) напоследък все по-широко приложение намират т.н. FERROELECTRIC RAM, които не изискват изчакване при запис, като бързодействието се ограничава основно от възможностите на SPI. Те са енергонезависими – запазват съдържанието си при отпадане на захранването. Достъпни са с обем от 512 байта до 128 килобайта – напр. FM25L04 – FM25V10. Повечето от тях не понасят захранване, по-високо от 3,6 V. При работа с микроконтролери, работещи на 5 V, е необходимо да се използват транслатори на нива, както беше разгледано по-горе.

– *ADC, DAC и цифрови потенциометри*

Понякога дори и да има вградени аналогово-цифрови и цифрово-аналогови преобразуватели в микроконтролера, се предпочита да се използват външни такива. Това позволява по-ефективно да се раздели цифровата от аналоговата част, като по този начин силно се редуцират смущенията и се подобрява точността на

преобразуването. SPI е особено подходящ за изграждане на връзката с тях поради малкия брой използвани импулсни сигнали и високата скорост на обмен.

Разнообразието от ADC, които се предлагат на пазара, е изключително голямо. Достъпни са както ADC на базата на SAR с разрядност от 8 до 12 бита, така и $\Delta\Sigma$ ADC от 14 до 24 разряда. Има както едноканални системи, така и с вграден аналогов мултиплексор до 8 входни канала. Като примери могат да се дадат **AD5310BRTZ** (10 bit, 1 ch., SAR), **LTC1857/8/9** (12/14/16 bit, 8 ch.), **AD7799BRUZ** (24 bit, 3 ch., $\Delta\Sigma$) и мн. други ...

DAC също са достъпни както в едноканален, така и в многоканален режим, като тук увеличаване броя на каналите се постига чрез вграждане на няколко преобразувателя в една схема. Разрядността на преобразувателите варира от 8 до 14. При някои от тях опорният източник е вграден вътре в схемата, а при други се разчита на използване на прецизни външни източници. Такива са например **AD5310BRM** (10 bit, 1 ch.), **MCP4822** (12 bit, 1 ch.), **MCP4922** (12 bit, 2 ch.) и мн. други. Трябва да се отбележи, че при много модели отсъства сигнала SDO, което прави невъзможно свързването им в последователна (верижна) схема.

Цифровите потенциометри са един клас параметрични преобразуватели (код-съпротивление), което активно се налага в последно време. Представяват верига от последователно свързани съпротивления и аналогов ключ, с помощта на който може да се осъществи връзка към желана точка от този делител чрез задаване на подходящ цифров код. Обикновено настройката се извършва на 64 до 256 стъпки (6-8 разряда). В един корпус се вгражда 1 или 2 цифрови потенциометъра. Като пример могат да се посочат **MCP4131** (128 step, 1ch., 2 – 50 K Ω), **MCP4251**-(256 step, 2 ch., 2 – 50 K Ω) и др.

– други примери

SPI магистралата е предпочитана при изграждане на „интелигентни (smart)“ сензори – температурни, за влажност, налягане, акселерометри, жироскопи, магнитометри и др. Например **L3G4200** представлява MEMS, съдържащ три-дименсионен жироскоп с вграден $\Sigma\Delta$ ADC, достъпен през SPI. **BMX055** обединява в себе си три дименсионни акселерометър, жироскоп и магнитен сензор, като данните от тях се преобразуват от $\Sigma\Delta$ ADC и са достъпни чрез SPI.

• **В заключение**

От казаното по-горе се вижда, че синхронните серийни интерфейси, подобни на SPI и Microwire, могат да гарантират прост и надежден обмен на данни. Независимо от това те въвеждат известни ограничения, които се свеждат до:

- обикновено не допускат вграждане в системата на повече от един водещ модул;
- не е предвидено адресиране на подчинените модули;
- най-често няма предвиден механизъм за контрол при възникване на грешки. Водещият модул няма данни дали съобщението е достигнало до подчинения модул;
- тази системна шина не предполага лесно добавяне на нови подчинени модули.

Независимо от ограниченията, това е предпочитаният вътрешно-системен интерфейс, когато е необходимо да се гарантира високоскоростен обмен с периферни схеми.

2.4.2. Синхронен сериен интерфейс I²C

Протоколът за междусхемни връзки I²C (*Inter-Integrated Circuit*) също е ориентиран за изграждане на серийна връзка между микроконтролерите и разнотипни периферни схеми в една и съща система, а не между отделни системи. Разработен с цел да се премахнат ограниченията, присъщи на SPI и Microwire. Тук вече се преодоляват ограниченията по отношение скалиране (разширяване) на системите, работа с множество и разнотипни периферни схеми, присъединени към една и съща магистрала, работа с различни скорости на обмен и пр. Първата версия на протокола е публикувана през 1982 г. в последствие е доразвита през 1992 г. (Version 1.2) и допълнена през 1998 и 2000 г. (Version 2). Последната (към момента) версия е публикувана през 2014 г. (Version 6).

- **Характеристики и особености**

Някои от *специфичните особености и характеристики на I²C-шината* се свеждат до [13]:

- Необходими са само две свързващи линии - линия за последователни данни (SDA) и линия за тактови сигнали (SCL);
- Всяко устройство, свързано към шината, е програмно достъпно чрез уникален адрес;
- Връзката се изгражда на принципа главен/подчинен (master/slave) модул и може да се осъществи по всяко време;
- Главните (водещи) модули работят в режим на *главни предаватели* или като *главни приемници*;
- Допустим е режим на работа с множество водещи (главни) модули (multi master mode). За предотвратяване на изкривяване на данните са въведени механизми за установяване на колизии и арбитражиране на достъпа, ако два или повече главни модула едновременно инициират трансфер на данни;
- Сериеният двупосочен трансфер на 8-битови данни може да се осъществява със скорост до 100 kbit/s в стандартен режим, до 400 kbit/s в бърз режим или до 3.4 Mbit/s във високоскоростен режим;
- Сериен 8-битов едноразсочен трансфер на данни до 5 Mbit/s в ултра високоскоростен режим;
- Броят на интегралните схеми, които могат да бъдат свързани към една и съща шината е ограничен само от максимален капацитет на шината. В зависимост от режима той не трябва да надхвърля 200 или 400 pF.

Тези особености на съвместимите интегрални схеми с I²C-bus позволяват:

- бързо проектиране на системата и лесно преминаване директно от функционална блокова схема към прототип;
- Интерфейсните схеми се свързват директно към I²C-шина без необходимост от допълнителни външни схеми – буфери и пр. Това позволява прото-

типна система да бъде модифицирана или надградена просто чрез присъединяване или премахване на интегрални схеми към/от шината.

Интегралните схеми, съвместими с I²C-bus, *притежават* някои *характеристики*, които ги правят особено подходящи за вграждане в системите:

- Функционалните блокове в структурната схема съответстват на реално съществуващите интегрални схеми;
- Не е необходимо да се проектират схемотехнични решения за връзка към шината, защото интерфейсът за връзка към I²C-шината вече е интегриран в чипа;
- Интегрирането на адресна фаза и фаза за пренос на данни в единен кадър данни позволява обменът да се осъществява само под алгоритмично управление;
- Едни и същи типове IC често могат да се използват в много и най-различни приложения;
- Времето за проектиране намалява поради честото използване на еднотипни функционални блокове в интегралните схеми, съвместими с I²C-bus;
- В системата могат да бъдат добавяни или премахвани интегрални схеми, без да засягат останалите елементи към шината;
- Откриването и отстраняването на грешки е лесно;
- Неизправностите могат веднага и лесно да се проследят;
- Времето за разработка на програмното осигуряване може да бъде намалено чрез използване на библиотеки от софтуерни модули за многократна употреба.

Освен изброените предимства, CMOS интегралните схеми, съвместими с I²C-bus, са подходящи за използване в преносимо оборудване и системи, захранвани с батерии. Всички те имат:

- Изключително ниска консумация;
- Висока шумоустойчивост;
- Възможност за работа в широк диапазон на захранващото напрежение;
- Широк работен температурен диапазон.

• *Същност*

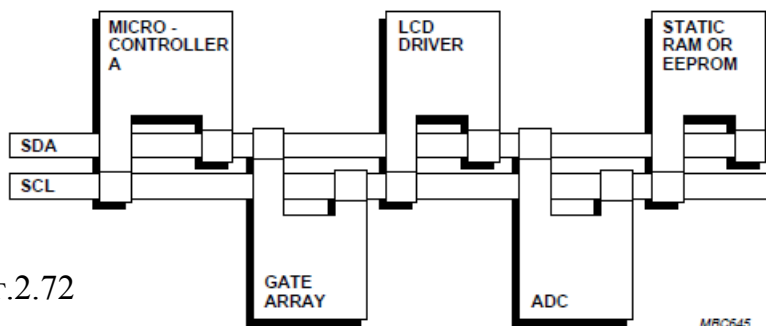
I²C е *вътрешносистемна синхронна серийна магистрала*, която, подобно на SPI или Microwire, е базирана на връзката водещ-подчинен между отделните модули. Използването на шината изцяло се контролира от водещия модул. Намира приложение в случаите, когато скоростта на обмен между микроконтролера и интерфейсите не е критична.

Всяко устройство се разпознава по уникален адрес (независимо дали става дума за микроконтролер, памет или интерфейсна схема) и може да работи както предавател или приемник, в зависимост от изпълняваната операция. Освен предаватели и приемници, устройствата могат да бъдат разглеждани като водещи (главни) или подчинени при осъществяване на трансфер на данни (вж. таблица 7). Водещо или главно (master) е устройството, което инициира трансфер на данни по

шината и генерира синхронизиращите сигнали за управление на този трансфер. По това време всяко адресирано устройство се счита за подчинено (slave).

Таблица 2.7 – термини и описанието им		
термин		описание
Transmitter	Предавател	Устройството, което изпраща данни по шината
Receiver	Приемник	Устройството, което приема данни от шината
Master	Водещ (главен)	Устройството, което инициира обмена, генерира тактовите сигнали и преустановява обмена
Slave	Подчинен	Устройството, което е адресирано от водещия модул
Multi-master	система с мн-во водещи модули	Възможност повече от един модул последователно във времето да поема управлението на шината
Arbitration	Арбитражиране на достъпа до шините	процедура, гарантираща последователен достъп да шината на потенциалните водещи модули така, че да не се допуска пропадане на съобщения
Synchronization	Синхронизиране	процедура за синхронизиране на тактовите сигнали на две или повече устройства

I²C шината използва само две линии за връзка, наречени SDA (сериен данни) и SCL (сериен тактов сигнал). Шините за обмен (SDA) и синхронизация (SCL) не са просто свързващ проводник, те удовлетворяват всички формати и процедури за комуникация в рамките на системата. Протоколът така дефинира състоянията на тези две шини, че позволява адресиране на избрано устройство, двупосочен

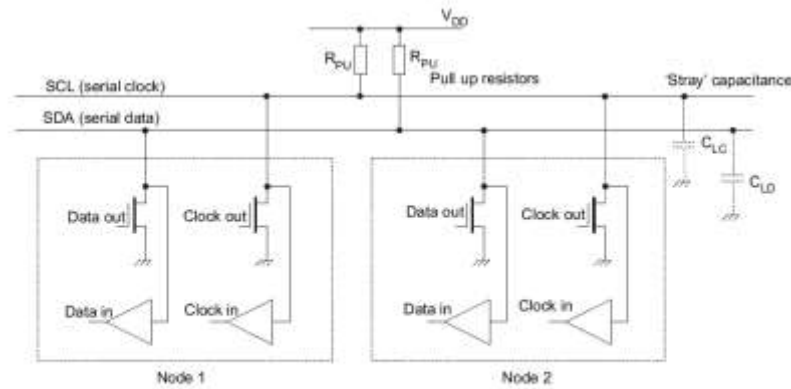


Фиг.2.72

обмен на данни (четене или запис) и потвърждение за успешен обмен при предаване на всеки байт. Една структура, при която се използва I²C шина за връзка между елементите, е показана на фиг.2.72. Тук в ролята на водещ модул влиза микроконтролерът, а останалите елементи са подчинени модули. Разбира се, всеки един от тях може да бъде премахнат или към системата да се присъедини друг модул (интегрална схема), удовлетворяващ системните спецификации на I²C.

- **Сигнали**

Вече беше казано, че сигналите, които изграждат магистралата, са SDA (serial data) и SCL (serial clock). Всеки модул, присъединен към нея (независимо дали е водещ или подчинен), притежава изход отворен колектор или отворен дрейн – фиг.2.73. За въвеждане на сигналите се използват стандартни логически буфери. Високото логическо състояние се обезпечава от товарните резистори RPU – по един за SDA и SCL съответно. Това определя ниското логическо ниво (лог.“0“) като **доминантно**, а високото (лог.“1“) – като **рецесивно**.



Фиг.2.73

Изграждането на шината по схема „жично ИЛИ“ позволява паралелното свързване на изходите без възникване на конфликтни ситуации, без да е необходимо тяхното управление. От друга страна времеконстантата, дефинирана от товарните резистори и сумарния капацитет на шините, води да ограничаване на бързодействието на интерфейса. Системните спецификации ограничават общия капацитет за всяка шина до 400 pF, а фронта на нарастване – до 1 μ s (в стандартен режим).

И двете линии са двупосочни, но тактовият сигнал SCL винаги се генерира от текущия водещ модул. Когато никой възел не е активен (в този момент няма активен водещ модул), всички изходни транзистори са запушени и линиите са във високо състояние.

- **Кодирание, фази и формат на данните.**

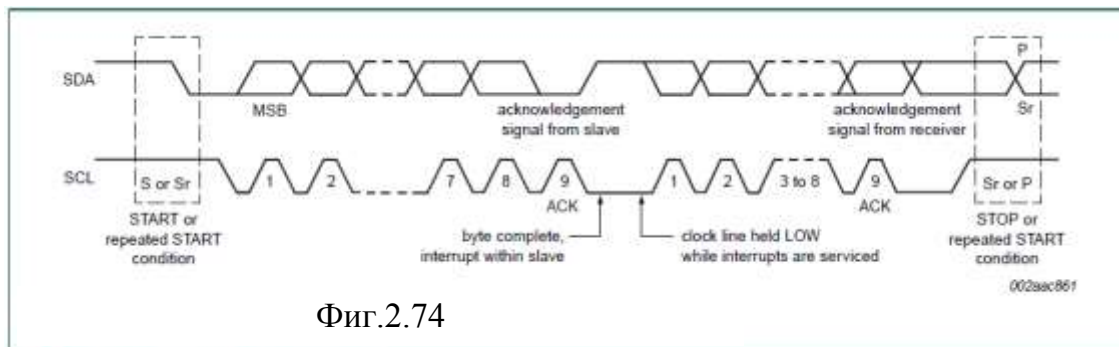
Прието е логическите „1“ да се кодират с високо ниво, т.е. на шината се поддържа рецесивното състояние, при което изходният транзистор е запушен. Съответно лог. „0“ се кодира с ниско (доминантно) ниво, като съответния транзистор е наситен. При преобразуване на данните от паралелен в сериен формат **водещ е старшият бит** (направете сравнение с SCI интерфейса!).

Прието е, че данновата шина SDA не трябва да променя състоянието си, докато тактовият сигнал SCL е във високо ниво, защото тогава става възприемането им. Данните се променят само докато SCL е в доминантно състояние. Има заложени две изключения, които ще бъдат разгледани по-долу.

При обмена на данни има ясно обособени 5 фази, което се отразява и на структурата на драйверните програми:

- **Фаза на изчакване (IDLE)** или неактивна фаза. При нея нито един от потенциалните водещи модули не извършва обмен по магистралата и двете шини са в рецесивно състояние. Това състояние може да остане неопределено дълго време, докато някой от потенциалните водещи модули не инициира началото на нов обмен;
- **Фаза „начало на обмена“ - START** – изработва се от водещия модул при заемане на шината и инициране на обмена. Изразява се в преход от рецесивно в доминантно състояние на линията SDA, докато линията SCL е в рецесивно състояние. Веднага след това и SCL преминава в доминантно съ-

тояние (вж. фиг.2.74). Този преход се възприема от всички подчинени модули като начало на нов обмен и започват процес по следене за валидно адресиране на съответния модул.



Фиг.2.74

Следва да се отбележи, че след този момент обменът е побайтов. След всеки байт приемащото устройство (независимо дали е главен или подчинен модул) „връща“ потвърждение (ACK, лог.“0“). Ако в рамките на текущия тактов импулс данновата линия SDA не бъде установена в доминантно състояние, се приема като отрицание (NACK, лог.“1“). Така се обменят 8+1 бита, докато по тактовата шина SCL водещият модул предава 9 тактови бита.

Има пет събития, които могат да предизвикат генерирането на NACK:

1. В системата не се идентифицира приемник с адрес, съответстващ на предадения по шината адрес, така че няма подчинено устройство, което да отговори с потвърждение;
2. Подчиненият модул не може да приема или предава, тъй като изпълнява някаква функция в реално време и не е готов да започне комуникация с водещия модул.
3. При комуникацията получателят получава данни или команди, които не разбира.
4. По време на обмена на данни приемникът не може да получава повече байтове с данни.
5. Водещият модул в качеството на приемник сигнализира за край на трансфера към подчинения предавател.

– **Адресната фаза** започва непосредствено след старта. През нея водещият модул предава адреса на подчинения модул, с който предстои да се извърши обмен. Стандартно адресът се състои от 7 разряда, което позволява да бъдат адресирани до 127 устройства (има режими, в които се работи и с 10-разряден адрес). Последният разряд от първия байт определя посоката на обмен – при лог.“1“ водещият модул чете данни от подчинения модул, а при лог.“0“ извършва запис в него.

– **Данновата фаза** е естествено с продължение на адресната фаза. Тактовите импулси се изработват от водещия модул. Данните при операция „запис“ са издават също от него, а при операция „четене“ – от подчинения модул. В първия случай потвърдението се издава от подчинения модул, а при „четене“ – от водещия модул. Трябва да се отбележи, че в протокола е заложена възможността за извършване на **последователен пакетен достъп** - режим, при който след адресната фаза могат да бъдат обменени последователно няколко байта данни;

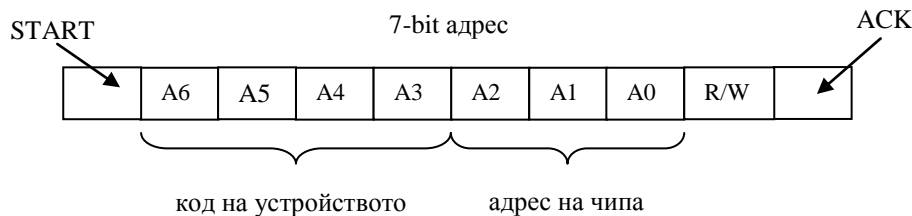
– **Фаза за край на обмена (STOP)**. Когато обменът на пакета с данни приключи, водещият модул освобождава магистралата. Това става, като се генерира състояние STOP. Аналогично на старта, тук правилото за постоянство

на линията SDA по времето, докато SCL е в рецесивно състояние, също е нарушено – извършва се преход на данновата линия от доминантно в рецесивно състояние.

Заб. Последните редакции на стандарта допускат да се генерира второ състояние старт, без да е подаден стоп за предходния обмен.

- **Формиране на адреса.**

Структурата на първия байт, предаван по време на адресната фаза в режим на 7 битова адресация, е показана на фиг. 2.75. Предназначението на отделните поле-



Фиг. 2.75

та е дадено в табл. 2.xx. Прави впечатление, че адресното поле е разделено на две части. Първата от тях, обхващаща старшите адресни линии A3-A6, е отделена за уникален **код на устройството**. Всеки от производителите, произвеждащи интерфейсни схеми за I²C, е запазил кодове за различните устройства (ADC, DAC, RTC, EEPROM и др.), които се заложили в процеса на производство.

Младшите **адресни линии A0-A2** дават възможност в **една система** да бъдат вградени по **няколко еднотипни схеми**, като всяка от тях получава уникален адрес. При някои схеми тези изводи (най-често EEPROM, DAC ...) са изведени на корпуса на схемата и външно се кодира желният адрес. При други индивидуалният адрес се задава от производителя и се избира опционно при придобиване на интегралните схеми.

Режимите за работа с 10-разреден адрес, общо повикване, смесени формати за адресиране, както и арбитражиране в мулти-мастер режим няма да бъдат разглеждани тук.

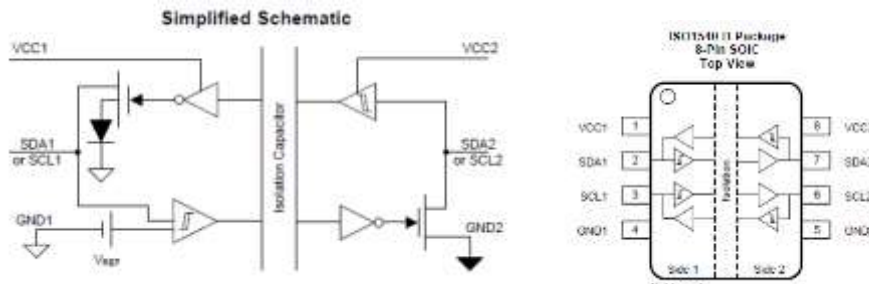
- **Използване на I²C bus**

Серийният синхронен интерфейс I²C намира най-вече приложения като вътрешен системен интерфейс във вградените системи. Той се използва най-вече за разширение възможностите на микроконтролерите, позволявайки да бъдат добавяни нови интерфейсни схеми, които ги няма вградени в нито един член на фамилията. Подходящ е за ниско-скоростни и средно-скоростни приложения поради последователния формат и не-високата скорост на обмен. Препоръчва се разширенията да се правят в рамките на една система, но е допустима и комуникация с изнесени схемотехнични модули. В този случай е необходимо да се вземат по-стриктни и адекватни мерки за електромагнитна съвместимост. Методът за изграждане на комуникацията (чрез схеми „жично ИЛИ“) позволява лесно изграждане на ефективни ценови бариери.

- **Галванично разделяне, преход към различни напржителни нива**

Тъй като шините, изграждащи интерфейса (SDA и SCL) предполагат двупосочно предаване на данни при сравнително високи скорости, то галваничното разделяне е трудно осъществимо. Независимо от това се предлагат специализирани

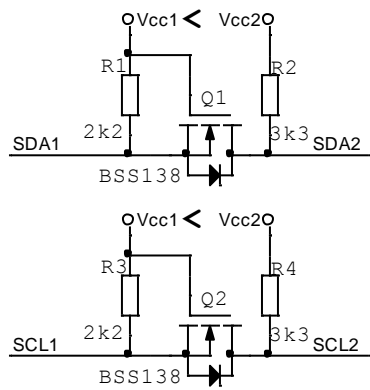
интегрални схеми, напр. ISO1540, ISO1541, които реализират тази функция. Опростената им структурна схема е показана на фиг.2.76. Галваничното разделяне се



Фиг.2.76

постига за сметка на високоволтови кондензатори от силициев диоксид с много малък капацитет – под 20 pF. Схемата осигурява галванично разделяне при напрежения до 2500 V и скорост на предаване 400 Kbps.

Много често възниква проблем, когато микроконтролерът и част от интерфейсите схеми, присъединявани към I²C магистралата, работят с различни захранващи напрежения, напр. 3,3 V и 5 V. В тези случаи е необходимо добре да се



Фиг. 2.77

проучи, доколко изводите на схемите от понисковолтовата част са толерантни към по-високо напрежение. Ако това е така се препоръчва товарните резистори да бъдат свързани към по-високото напрежение. Ако това условие не е изпълнено, се налага магистралата да бъде разделена на две части – „нисковолтова“ и „високоволтова“. За целта може да се използва схемата, показана на фиг. 2.77.

Резисторите R1 и R2 (респективно R3 и R4) играят ролята на товарни резистори за двата клона на шината, осигурявайки рецесивни нива съответно 3,3 V и 5 V. Ако на SDA1 (SCL1) се появи доминантно ниво (лог.“0“), транзисторът се насища и

доминантното ниво се прехвърля и към SDA2 (SCL). При поява на доминантно напрежение на SDA2 (SCL2) лог.“0“ води до провеждане на ток от диода, изолиращ дрейна на транзистора от подложката. Транзисторът влиза в активен режим и доминантното състояние се прехвърля към SDA1 (SCL1). Така схемата за доминантните състояния действа като двупосочен ключ, а за рецесивните състояния гарантира различни логически нива за двете страни.

Препоръчва се схемата да се използва, когато трябва да се управляват изнесени модули, свързани към основната платка, дори да работят при едно и също захранващо напрежение. В този част от външната за системата страна може да бъде поставена съответната ценерова бариера.

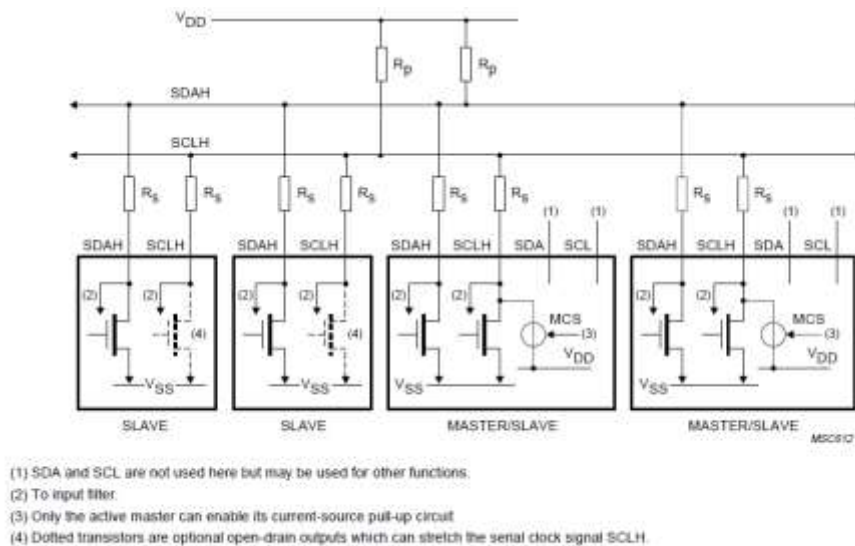
Производителите на компоненти предлагат подобни схеми в интегрално изпълнение, които също могат да бъдат използвани.

• **Скорост на обмен, опции:**

Стандартът допуска вътрешносхемната серийна магистрала I²C да поддържа различни скорости на обмен, ограничени от електрическите характеристики на 20.1.2021 г.

сегментите и възможностите на подключените към тях модули – водещи и подчинени. Това налага да се контролират параметрите на шините за гарантиране на надеждна работа. Съгласно стандарта работните режими са:

- **Стандартен режим.** Допуска работа с тактова честота до 100 Kb/s. В този режим фронтите за спадане на импулсите не трябва да превишават 300 ns а фронтите на нарастване да са не-повече от 1 μ s. Това налага, ако има свързани повече интерфейсни схеми, да се работи с по-малки товарни резистори, което води до по-голям ток.
- **Бързодействащ режим (Fast-mode).** Допустимата работна скорост до 400 kbit/s ($SCL \leq 400$ KHz). Тук вече изискването е фронтите на нарастване и спадане да не превишават 300 ns, което прави недопустимо свързването на изнесени периферни устройства. В този режим се работи с външни (за микроконтролера) памети и с някои аналогово-цифрови преобразуватели.
- **Високоскоростен режим (High-speed mode, Hs-mode)** – допуска се работа с тактова честота до 3,4 Mbit/s. Във водещите модули, които поддържат този режим, има заложен апаратни средства, които следят състоянието на шината и ускоряват преходните процеси (вж. генераторите на ток на фиг. 2.78). Магистралите обикновено се секционират, за да има по-малко интерфейсни схеми, присъединени към отделните участъци. Така еквивалентните капацитети, свързани към шините, силно се ограничават и могат да бъдат постигнати по-кратки преходни процеси.

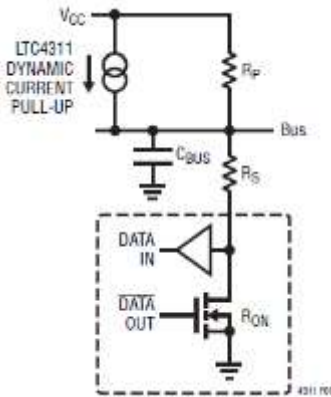


Фиг. 2.78

Във високоскоростен режим сигналите SDAH и SDLH се свързват към шините посредством нискоомни резистори R_s (вж. фиг. 2.xx). Те служат да демпфират паразитните трептящи кръгове, възникващи между паразитните индуктивности на шините и входно/изходните капацитети на отделните изводи. Така преходните процеси, възникващи при смяна на нивата на сигналите, затихват много по-бързо и могат да бъдат постигнати по-високи скорости на обмен.

• **Ускоряване преходния процес на магистралата**

Често, особено при работа с изнесени интерфейсни модули, свързани към магистрала I²C, капацитетът към шините става недопустимо голям и е трудно да се удовлетворят времевите изисквания. За тази цел са разработени и са достъпни специализирани интегрални схеми, които позволяват да се ускори процесът на заряд на магистралните капацитети, без да се използват нискоомни товарни резистори.



Фиг. 2.79

Пример за такава схема е LTC4311, която е ускорител за I²C (I²C-bus Accelerator). Схемата се свързва директно към всяка от шините SDA и SCL и не се нуждае от външно управление (вж. фиг. 2.79). Вътрешна логика следи напрежението на линиите и когато то след доминантно състояние се повиши до 0,65 V, включва вътрешен генератор на ток (Dynamic Current PULL-UP) със стойност 5 mA паралелно на товарния резистор на съответната шина. Той остава включен, докато напрежението на шината достигне V_{CC}-0,4 V, след което се изключва. Така скоростта на заряд на C_{BUS} многократно се ускорява и много по-бързо се достига до рецесивното състояние.

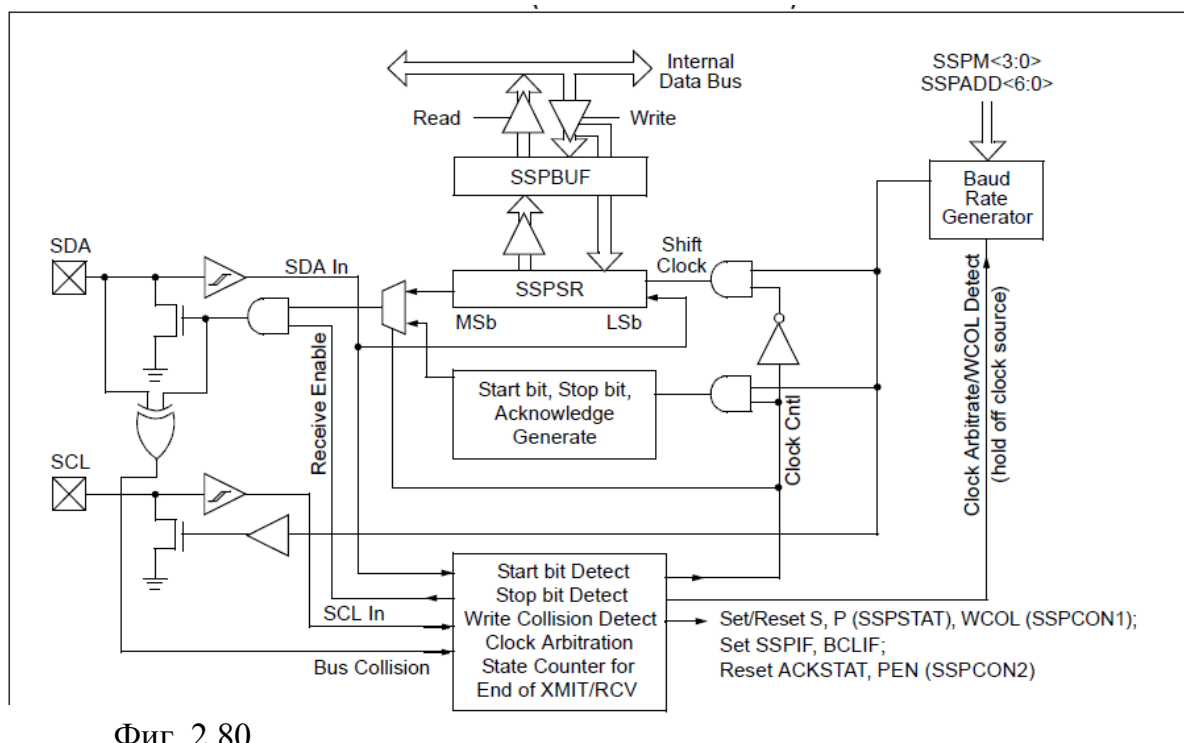
• **Примери – I²C системи при някои микроконтролери:**

В болшинството съвременни микроконтролери има вградени интерфейси, удовлетворяващи изискванията на системните спецификации на I²C bus за работа както в стандартен режим, така и поне в бързодействащ режим. Характерно е, че повечето от тези интерфейсни схеми могат да бъдат инициализирани за работа както като водещи (главни, master) модули, така и като подчинени (slave) модули. Независимо от съществените разлики между схемите, заложи в отделните микроконтролери, всички те могат да осъществят успешна комуникация с многото и разнообразни интерфейсни схеми, предназначени за работа с тази магистрала. По-долу като пример е показана опростената структура на някои от тях:

– **I²C при микроконтролерите от фамилия PIC18Fx...**

В микроконтролерите от фамилията PIC18Fxxx има вграден универсален серийен синхронен порт MSSP (Master Synchronous Serial Port), който може да бъде инициализиран за работа в режим като SPI или като I²C. Това ограничава използването в дадена система само на единия тип интерфейс. За това в повечето микроконтролери от фамилията има вградени по два серийни синхронни интерфейса - MSSP0 и MSSP1 - което дава възможност да бъдат използвани интерфейсни схеми от двата типа.

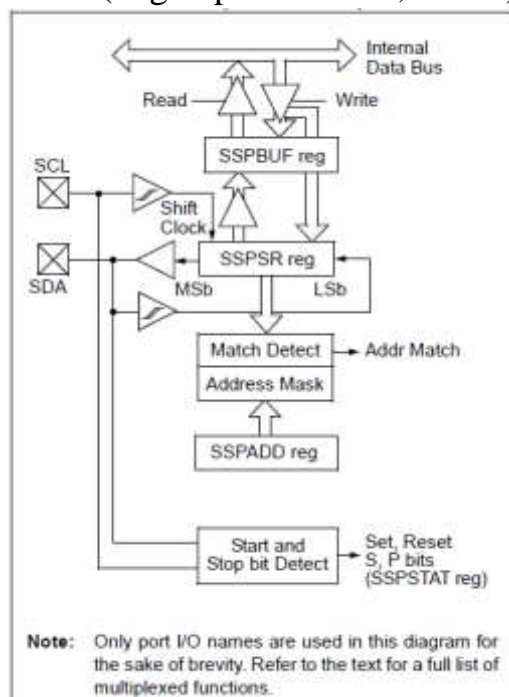
В режим като I²C MSSPx може да бъде конфигуриран за работа както като водещ, така и като подчинен модул. Еквивалентната структурна схема на модула при конфигуриране като водещ I²C модул е показана на фиг. 2.80. В основата стои преместващия регистър SSPSR. Връзката от вътрешносхемната магистрала към/от него става през буфера SSPBUF. Има логическа схема, която обезпечава генериране на състоянията за старт или стоп на обмена, както и потвърждение ACK в режим на водещ приемник.



Фиг. 2.80

За регистриране на състоянията по време на обмен се грижи допълнителна схема, като се актуализират флаговете в регистрите за състояние SSPCON1 и SSPCON2. Следи се за възникване на състояние старт и стоп, както и колизия по данновата линия SDA. По този начин е създадена възможност за реализиране на системи с няколко водещи модула.

Скоростта на обмен се определя от вградения bod rate генератор, като могат да бъдат програмно зададени скорости честоти 100 Kbit/s (standard mode, Sm), 400 Kbit/s (fast mode, Fm) или 1 Mbit/s (Fast-mode Plus Fm+). Високоскоростен режим (High-speed mode, Hsm) и еднопосочен свръхвисокоскоростен режим (Ultra Fast-mode, Ufm), които се допускат от стандарта, от тази система не се поддържат.



Фиг. 2.81

Обобщената структурна схема на интерфейса при инициализиране за работа като подчинен модул е показана на фиг. 2.81. Вижда се, че тя е много по опростена от тази при работа във водещ режим. Тук вече има регистър, в който може за де зареди код на устройство и адрес по програмен път. Модулът допуска работа със 7-разряден и 10-разряден адрес. Този адрес се сравнява (в блок Match Detect) с постъпващия в регистъра SSPSR по време на адресната фаза на обмена.

– *I²C при микроконтролерите от фамилия MSP430...*

При микроконтролерите от фамилията MSP430Fxxx има вградени универсални серийни комуникационни интерфейси (*Universal Serial*

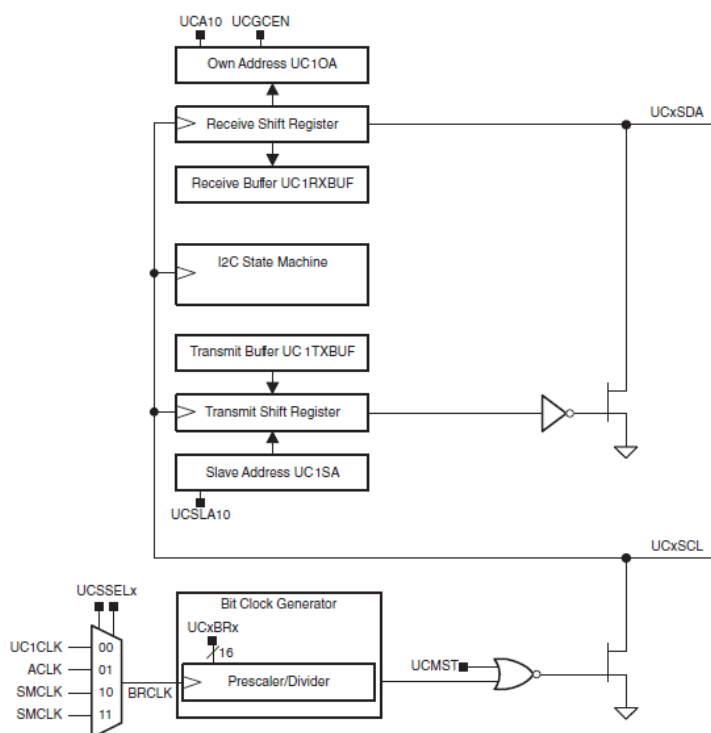
Communication Interface - USCI), който могат да бъдат два типа – USCI-A или USCI-B. Първият от тях може да се използва в режим на SCI или SPI, а вторият – като SPI или I²C.

В режим I²C, модулът USCI-B осигурява интерфейс между устройството и I²C-съвместими устройства, свързани чрез двупроводна последователна шина. Функциите на USCI-B в режим I²C включват:

- ▲ Съответствие със спецификацията на I²C v2.1 от 2000 г.;
- ▲ Поддържане на 7-битов и 10-битов режими за адресиране на устройства;
- ▲ Възможност за осъществяване на общо повикване;
- ▲ Генериране на СТАРТ / РЕСТАРТ / СТОП състояния;
- ▲ Multi-master режим на предавател / приемник;
- ▲ Подчинен режим на приемник / предавател;
- ▲ Поддържане стандартен режим до 100 kbps и бърз режим до 400 kbps;
- ▲ Програмируем SCL при работа като водещ модул;
- ▲ Следене състоянието на шината при привеждане на микроконтролера в режим на ниска консумация;
- ▲ Автоматично излизане от режим на ниска консумация при откриване на състояние СТАРТ.

Обобщената структурна схема на комуникационния модул в режим I²C е показана на фиг. 2.82. Тук има два преместващи регистъра – за приемане на данните и съответно за предаване на данните. В регистъра UC10A се зарежда собствения адрес, когато интерфейсът е инициализиран за работа като подчинен модул. Ако USCI-B работи като водещ модул, то по време на адресната фаза като адрес към

подчинените модули се изпраща съдържанието на регистъра UC1SA.



Фиг. 2.82

• **Периферни схеми за връзка към I²C:**

Към настоящия момент са разработени и се предлагат най-разнообразни схеми, удовлетворяващи системните спецификации на стандарта I²C. Благодарение на тях микроконтролерите могат да бъдат надградени, като се залагат оптималните модули за измерване и управление. По долу са изброени само някои от най-често използваните схеми:

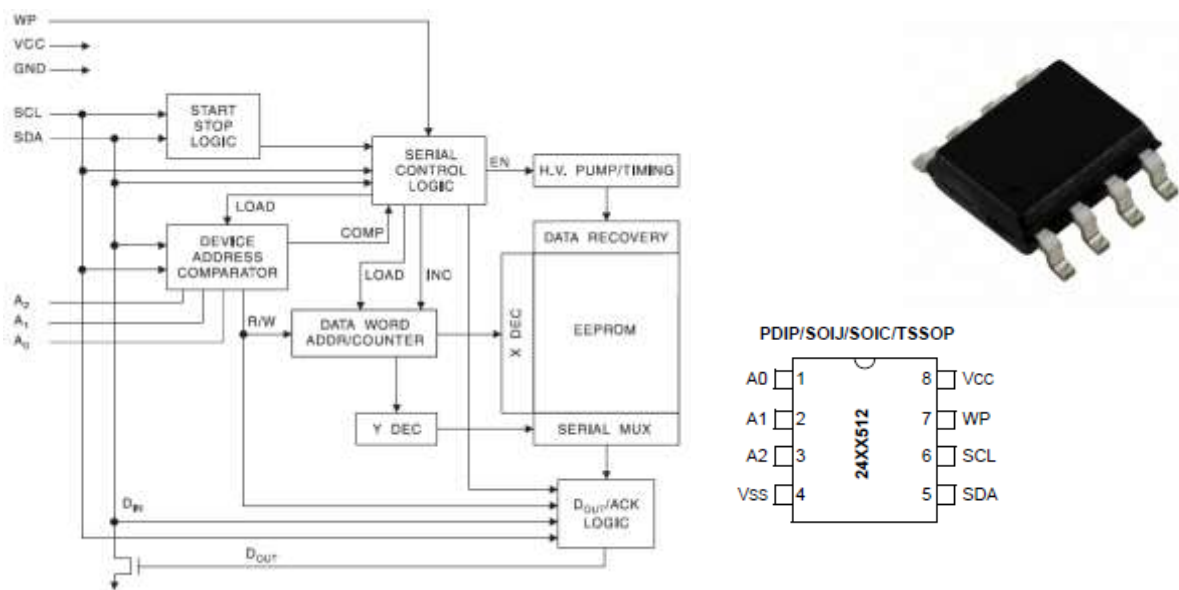
– **I²C EEPROM**

Може би най-често в системите има необходимост от доба-

вяне на допълнителна памет. За работа със системната шина I²C са разработени паметии EEPROM с обем от 128 бита до повече от 2 МВ, като обикновено се поддържа байтова организация на данните. Всички те поддържат стандартна (100 Kbps) и висока скорост на обмен. За код на устройство най-често се използва 1010 (A₁₆).

Независимо от обема на използваната памет, разположението на изводите на корпуса се запазва, нещо повече – различни производители запазват едно и също разположение за съответния вид корпус. Това позволява да бъдат използвани схеми от различни производители и дори да се прави преход към други типове (с по-голям или по-малък обем на паметта) при необходимост. Не трябва да се забравя, че при замяна вероятно ще се наложи малка промяна в драйверната програма.

Типичната структура на EEPROM, достъпен чрез I²C магистрала, е показана на фиг. 2.83 [15]. В нея ясно се разграничават функционалните модули, обезпеча-



Фиг. 2.83

ващи работата на масива памет, както и тези, осъществяващи връзката с I²C магистралата (*кои са тези модули и какви са функциите им?*). Чрез A₀, A₁ и A₂ може да се кодира адреса на съответната памет – така в системата могат да се вградят до 8 еднотипни паметии.

Още при изграждане на системата трябва да се държи сметка **за времето за достъп до паметта** в различни режими и доколко това ще удовлетвори изискванията, налагани от конкретното приложение. Интерес представлява да се оцени средното време за четене на байт и блок (примерно 16 байта), както и на запис в сериен I²C EEPROM, като се отчита и времето за предаване на служебната информация – стартов и стопов бит, адресиране на устройство, адрес в информационния масив. Нека обменът става в бързодействащ режим – 400 Kbps, което е еквивалентно на 2,5 μs за обмен на 1 бит.

При четене на 1 байт е необходимо да се **заеме два пъти шината**. При първия се предава СТАРТ (1 такт), адресиране на устройството (9 такта – 1 байт + АСК) и запис на два байта за адрес от паметта (18 такта – 2 байта + 2*АСК) – или **общо 28 такта**. При втория достъп се предава РеСтарт (вместо СТОП и СТАРТ – 1 такт),

отново адрес на устройството (9 такта – 1 байт + АСК) и четене на 1 байт – 9 такта (заедно с АСК) и генериране на състояние стоп – общо 20 такта. За обмена на байта се получават общо 48 такта по $2,5 \mu\text{s} = 120 \mu\text{s}$.

При четене на 2 байта отново е необходимо да бъдат извършени два последователни обмена по I²S шината. Първият е идентичен с разгледания по-горе и отново приключва за 28 такта. При втория след адреса на устройството (9 такта) вместо 1 се приемат 16 байта (всеки заема по 9 такта) и се губят общо 154 такта. Така целият обмен заема 182 такта по $2,5 \mu\text{s} = 455 \mu\text{s}$. Цялото време, отнесено към големината на блока, дава около 28,5 μs за 1 байт – време, което е значително по-малко, отколкото при извършване на побайтов обмен.

От тези примери може да се направи един *основен извод* – при използване на памети със *сериен достъп* е много по-изгодно така да се организират програмните процедури, че *да се работи с блоков обмен* за данните, записани в EEPROM. Ако средното бързодействие, което може да се постигне с I2C шината не е достатъчно, следва да се използват памети, ориентирани за работа чрез SPI.

При запис трябва да се има предвид, че се изпълнява вътрешна процедура, при която е необходимо време, за да се натрупат токоносителите в плаващите гейтове на запомнящите транзистори. Затова записът винаги се осъществява на блокове, които варират от 16 до 128 байта. Времето за запис при различните модели на различните производители варира от 2 до 10 ms. Ако това време е недопустимо голямо според изискванията на конкретното приложение, следва да се използват RAM или NVRAM.

– Ferroelectric NVRAM

Принципът, заложен във феро-RAM, е предсказан още през 1952 г. (в магистърска дипломна работа в MIT). Предполага се използване на фероелектрическият ефект, който обхващат някои материали. Той се изразява в появата на хистерезис при преполяризиране на определени диелектрици, което води до формиране на нелинейни кондензатори, ако са изградени с тяхна помощ.

По своето действие тези запомнящи клетки наподобяват донякъде запомнящите клетки с феромагнитни пръстени (широко разпространени през 50-те и 60-те години), където се използваше хистерезиса в характеристиката на феромагнитните материали. За разлика от феромагнитните памети тук вече се използва хистерезисът в характеристиката на фероелектричните материали (обикновено оловно-циркониев титанат - PZT).

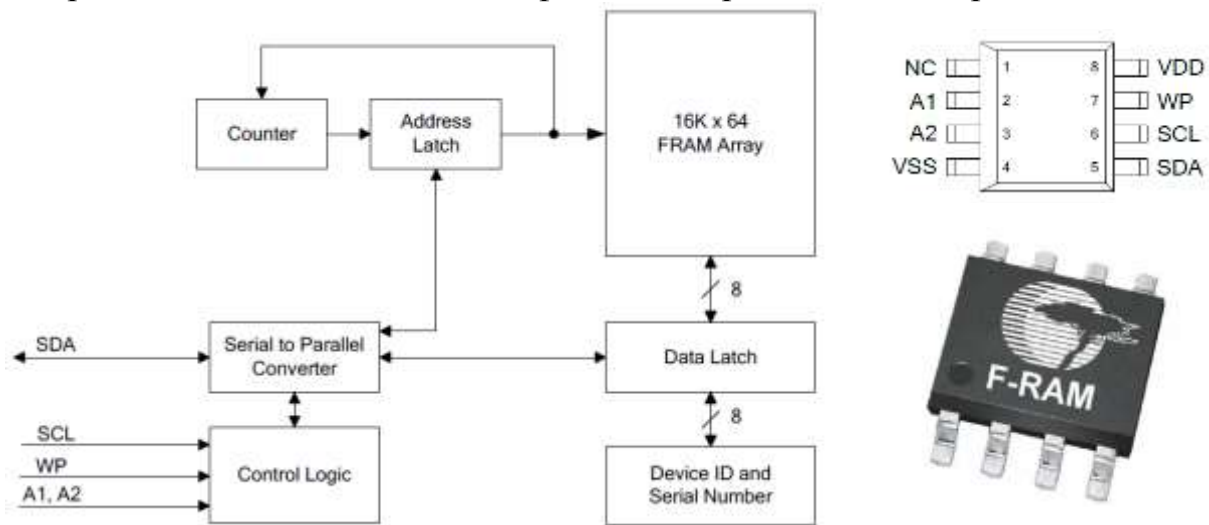
Запомнящите клетки по структура напомнят клетките, изграждащи динамичните памети (DDR) – комбинация от транзистор и запомнящ кондензатор. Тук обаче използваният кондензатор вече е нелинеен, като диелектрика се явява фероелектрик. В резултат на това не е необходимо зарядът в него непрекъснато да се опреснява, както при DDR. Нещо повече – веднъж преполяризиран, запомнящият елемент не се нуждае повече от захранващо напрежение.

Паметите от типа ferroRAM, които са фероелектрични енергонезависими памети, са достъпни с обем от 4Kbit (FM24C04) до 1Mbit (FM24V10). Някои от основните им параметри са:

- ▲ Брой цикли на презапис – повече от 100 трилиона – 10^{14} пъти;

- ▲ Запазване на данните – повече от 10 години;
- ▲ Без изчакване при запис;
- ▲ Поддържане скорости Sm – 100Kbps, Fm – 400Kbps, Hsm – 3,4Mbps;
- ▲ Директа подмяна на съществуващите EEPROM (на хардуерно ниво);
- ▲ Фабрично записан идентификатор на производителя, идентификатор на типа на устройството и индивидуален сериен номер;
- ▲ Ниски работни напрежения – 2V-3,6V или 3,6V-5V;
- ▲ Ниска консумация в работен режим - < 150 μ A (при Sm - 100KHz);
- ▲ Консумация в режим на готовност - < 90 μ A
- ▲ Консумация в режим „заспиване“ - < 5 μ A

Обобщената структурна схема на подобна памет, която може да бъде достъпна чрез I2C шина, е показана на фиг. 2.84. Трябва да се обърне внимание, че схе-



Фиг. 2.84

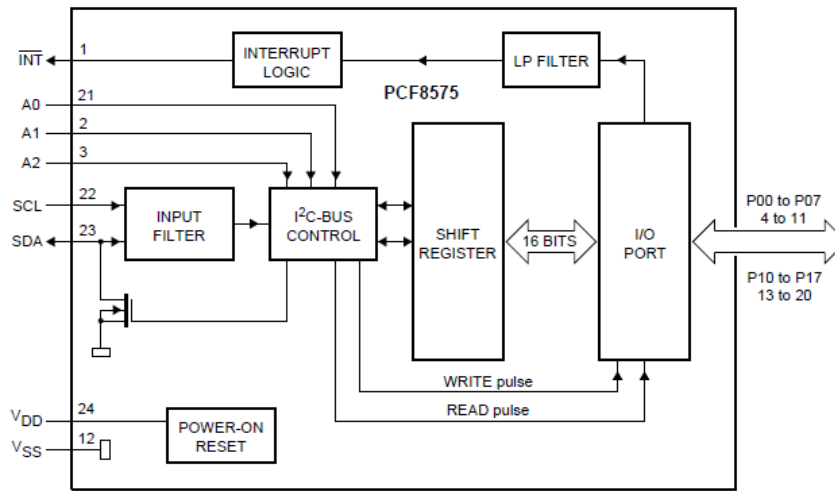
мите, които работят на по-ниски напрежения (напр. 3,6V), магистралните им изводи не са толерантни към шина, която е терминирана към по-високо напрежение – напр. 5 V.

При тези схеми времето за запис е многократно намалено спрямо EEPROM при равни други условия. Съществено предимство е много по-малката консумация във всички режими.

– Разширители за паралелни портове

Много често в дадена система броят на вградените паралелни портове (вж. гл. 3) в микроконтролера се оказва недостатъчен, за да се покрият всички входно-изходни сигнали. В такива случаи могат да се използват специализирани интегрални схеми, наречени разширители (I/O expander), чрез които „се добавят“ недостигащите ресурси към микроконтролерите. Най-често те се свързват чрез шината I²C. Предлагат се както еднопортови (напр. MCP23008), така и двупортови разширители (MCP23017, TCA6416, PCF8575). Структурната схема на един от тях – PCF8575 - е показана на фиг.2.85. Вътрешните структури и на останалите от тях са много подобни, както и основните им технически характеристики. Вижда се, че в системата могат да бъдат включени до 8 такива схеми, което позволява раз-

ширяване до 16 порта по 8 вход/изхода всеки. Това е сериозен ресурс за всяка управляваща система.



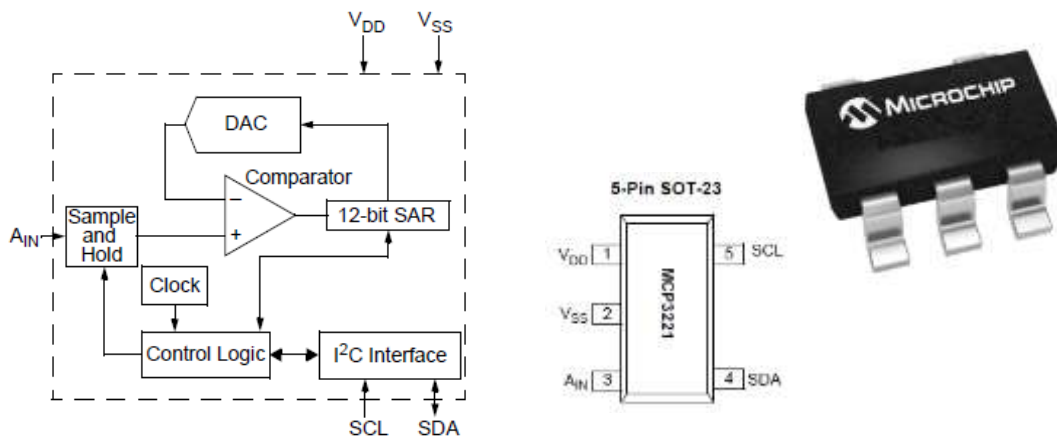
Фиг. 2.85

– **Аналогово-цифрови преобразуватели (ADC)**

В болшинството микроконтролери има вградени ADC. Независимо от това много често се използват външни преобразуватели. Това се налага по една или няколко от средните причини:

- ▲ Липса на ADC в използвания микроконтролер;
- ▲ Използване на специализирани ADC (напр. с диференциален вход или ориентирани за работа със специфични сензори);
- ▲ Използване на ADC с по-висока разрядност;
- ▲ Използване на по-добри опорни източници;
- ▲ Постигане на по-добра шумоустойчивост (чрез отдалечаване и/или екраниране на аналоговата от цифровата част).

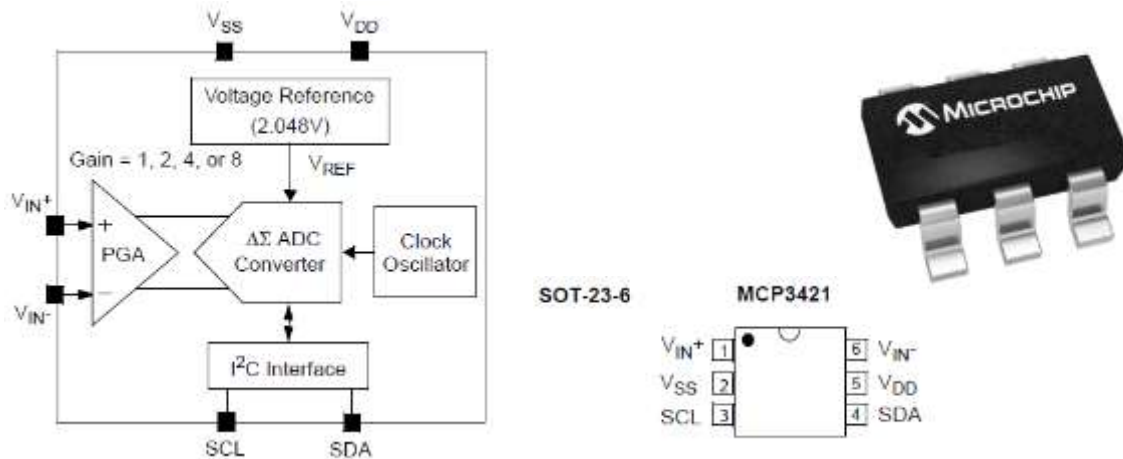
Използваните ADC с шината I²S обикновено са с не високо бързодействие – обикновено до няколко десетки преобразувания в секунда. Основно преобладават ADC с поразрядно кодиране (от 8 до 14 разряда, напр. MCP3221 – фиг.2.86) и делта-сигма ADC (от 14 до 22 разряда, напр. MCP3421 – фиг.2.87). С особено внимание трябва да се подхожда към втория тип и да се прави разлика между дискрет-



Фиг.2.86

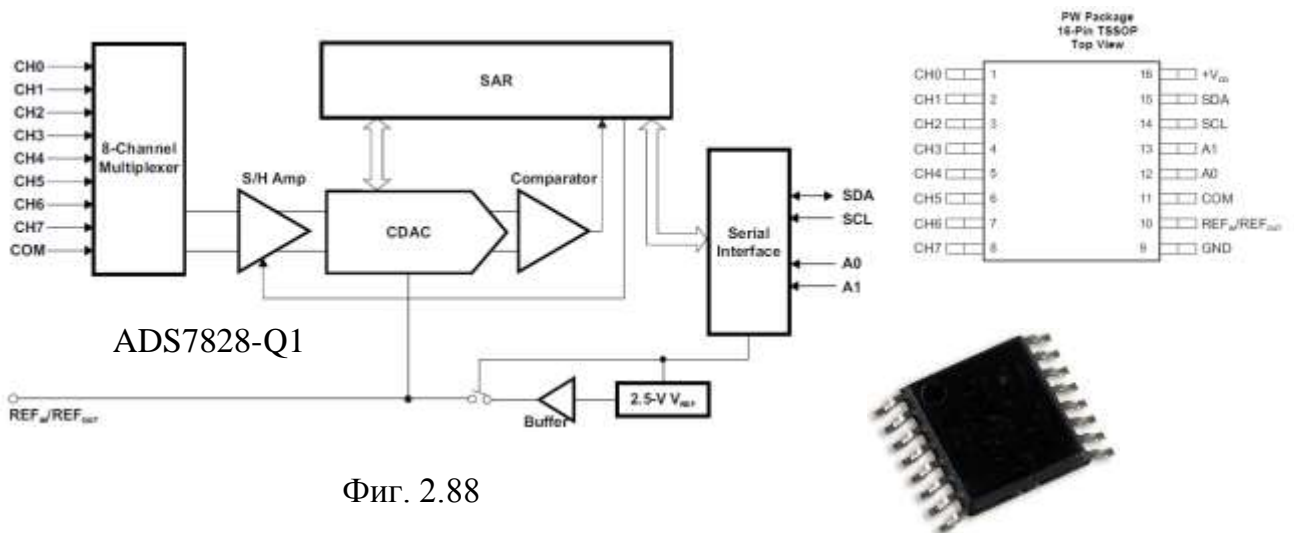
ност и точност, като особено внимание се отделя на опорния източник!

Широко се използват единични ADC в корпуси с минимален брой изводи



Фиг. 2.87

(напр. SOT23-6). Разбира се, приложение намират и многоканални (с 4 и дори 8 канала) ADC, напр. ADS7828 – фиг.2.88, като при тях организацията обикновено се свежда до аналогов мултиплексор на входа и единичен ADC.

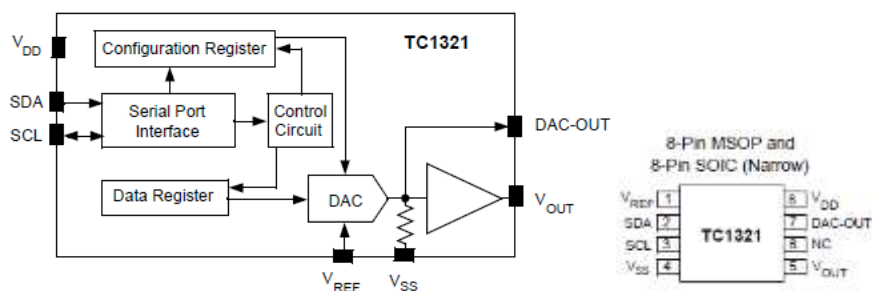


Фиг. 2.88

По-подробни сведения за използването на ADC в микропроцесорни системи може да намерите в глава 5.

– **Цифрово-аналогови преобразуватели - DAC;**

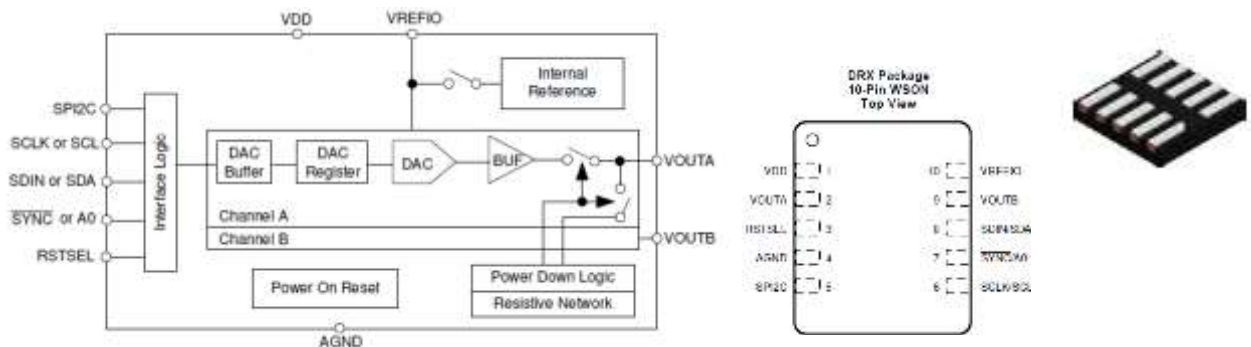
В микроконтролерите сравнително по-рядко вграждат цифрово-аналогови преобразуватели. Най-често това е резистивен делител, към който е свързан ана-



Фиг. 2.89

логов мултиплексор, захранен или от вътрешен опорен източник, или от общото захранване на микроконтролера. В общия случай от крайните параметри на такъв преобразувател има много какво да се желае. Затова набора от външни DAC, които се предлагат, е много голям. Тяхната разрядност варира от 8 до 12 разряда, например TC1321 – фиг. 2.89.

Достъпни са и преобразуватели с 14 и дори 16 разряда като DAC80502 – фиг. 2.90. При работа със схеми с повишена разрядност трябва много критично да се подхожда към използвания опорен източник. Например, в случая на DAC80502 температурната нестабилност на вградения опорен източник е $5 \text{ ppm}/^\circ\text{C}$. Това означава, че промяна на околната температура с 10°C ще доведе до промяна на изходното напрежение с повече от 0,5 МЗР.

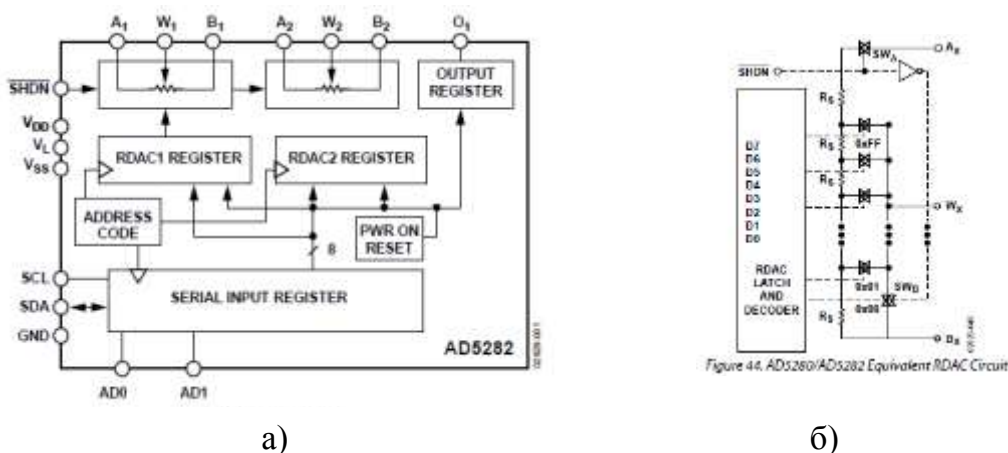


Фиг. 2.90

Цифрови потенциометри

Цифровите потенциометри все по-широко се използват в последно време като алтернатива на механичните потенциометри и тримерите. Понякога се означават като VR (Variable Resistor) или RDAC (Resistive DAC).

Структурно цифровите потенциометри (фиг. 2.91.а) се състоят от интерфейсна част, която осигурява връзката със серийната магистрала, в разглеждания случай – I²C, група от последователно свързани резистори (резистивен делител), като чрез аналогов мултиплексор може да се осъществи връзка към определена точка от делителя (фиг. 2.91.б). Достъпни са както единични (напр. AD5280), така и двойни цифрови потенциометри (AD5282). Обикновено делителят съдържа 128 или 256 резистора – толкова са и стъпките, които могат да бъдат настроени. В



Фиг. 2.91

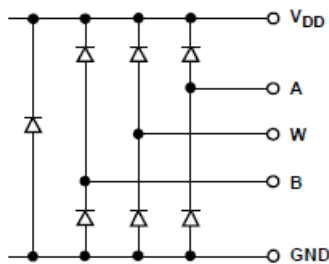
болшинството случаи е възможно използването както в потенциометричен, така и в реостатен режим. Предлагат се с обхват от 2 КΩ до 100 КΩ. Вградените резистори са идентични (еднакви по стойност), което позиционира потенциометрите в категорията на тези с линейна скала. Програмно могат да се настройват стойности, които да осигурят с много добро приближение и логаритмична скала.

Обикновено на корпуса са изведени само два адреса към вътрешния магистрален адресен дешифратор, което позволява в една система да се вградят до 4 схеми (или до 8 цифрови потенциометъра).

При използването им като елементи за настройка в аналогови схеми трябва да се има предвид, че имат сравнително ограничена честотна лента. За това свойства както паразитните капацитети в паралел на отворените аналогови ключове в мултиплексора, така и крайното съпротивление на включения ключ (обикновено няколко десетки ома).

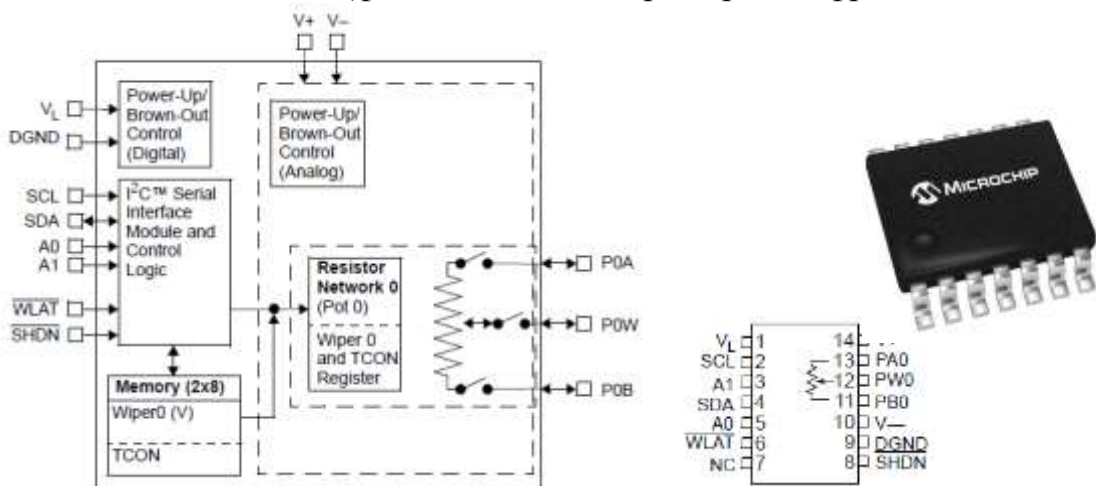
Допълнителен проблем може да предизвика началната стойност на настройката в момента на включване на захранването. Повечето цифрови потенциометри така са проектирани, че при начално включване, преди да бъдат инициализирани от микроконтролера, „плъзгачът“ да бъде позициониран в средата на делителя. Има и такива модели, при което настройката се записва във вътрешни регистри, които се намират в енергонезависима памет – така последната настройка се запазва вътре в потенциометъра.

Напрежението на трите извода на потенциометъра (A, B и W), е ограничено до положителната и отрицателната стойност на захранващото напрежение посредством диодни ограничители – вж. фиг. 2.92. Това трябва да се има предвид, когато се синтезират принципните схеми на устройствата. В случаите, когато потенциометричната част трябва да работи при по-високи напрежения, отколкото е захранването на цифровата част, е необходимо да бъдат подбрани други модели цифрови потенциометри. При тях обикновено е предвидено допълнително захранване на „аналоговата“ част – резистивните делители и аналоговия мултиплексор. В този случай е възможно работа до напрежения, определени от тези допълнителни източници (фиг. 2.93). Например, цифровият потенциометър



Фиг. 2.92

захранване на „аналоговата“ част – резистивните делители и аналоговия мултиплексор. В този случай е възможно работа до напрежения, определени от тези допълнителни източници (фиг. 2.93). Например, цифровият потенциометър



Фиг. 2.93

MCP45HVX1 може да бъде използван до напрежения $\pm 18V$.

– **RTC**

Днес широко в системите се използват схеми, които съдържат в себе си часовник с календар (real time clock – RTC). Характерното за тези схеми е, че е необходимо да се обезпечи буферно хранване, което да осигури тяхната непрекъсваема работа при отпадане на основното хранване на системата. Това най-лесно става, когато връзката с тях е осъществена през I²C шина (не се налага изграждане на вериги, които да предотвратят разреждането на батерията през обезточените управляващи вериги към системата).

Болшинството RTC съдържат кварцово стабилизиране генератори, работещи на честота 32768 KHz (2^{15}) – така се постига минимална консумация и се гарантира дълъг живот на батерията. Като примери за такива схеми могат да се посочат DS1307, DS1338, MCP79410. Някои техни параметри, касаещи шината I²C, са посочени в табл. 2.8.

Таблица 2.8							
означение	тип	Код	брой ^{3аб.1}	режим	производител	C _{IN} , pF	забележка
24(L)Cxxx	EEPROM	1010	8	Sm, Fm, Fm+	MicroChip, Atmel et all	10	128Kx 8
FM24V10	NVRAM	1010	4 ^{3аб.2}	Sm, Fm, Hsm	Cypress	8	2*64Kx8
PCF8575	I/O Expander	0100	8	Sm, Fm	NXP	7	2*8 GPIO
TCA6416	I/O Expander	0100	8	Sm, Fm	TI	7	2*8 GPIO
MCP23008	I/O Expander	0100	8	Sm, Fm, Fm+	MicroChip	n.a.	8 GPIO
MCP3221	ADC	1001	8 ^{3аб.3}	Sm, Fm	MicroChip	10	12 bit, SAR
MCP3421	ADC	1101	8 ^{3аб.3}	Sm, Fm	MicroChip	10	18 bit, $\Delta\Sigma$
ADS7828	ADC	1001 0	4	Sm, Fm, Hsm	TI	n.a.	8 Ch, 12 bit, FAR
TC1321	DAC	1001 ^{3аб.4}	8 ^{3аб.3}	Sm	MicroChip	5	10 bit
DAC80502	DAC	1001	4 ^{3аб.5}	Sm, Fm, Fm+	TI	10	16 bit
AD5280	VR	0101 1	4	Sm, Fm	Analog Device	5	256 step, $\pm 5V$
MCP45HVX1	VR	0111 1	4	Sm, Fm, Hsm	MicroChip	10	256 step, $\pm 18V$
DS1307	RTC	1101 000	1 ^{3аб.4}	Sm	Maxim	10	RTC, NV_RAM
DS1338	RTC	1101 000	1 ^{3аб.4}	Sm, Fm	Maxim	10	RTC, 56B SRAM
MCP79410	RTC	1101 111 1010 111	1 ^{3аб.4}	Sm, Fm	MicroChip	10	RTC, SRAM, 1KB EEPROM
TCN75	T ^o sensor	1001	8	Sm, Fm	MicroChip	15	T ^o sen/mon
HDC1080	Rh sensor	1000 000	1	Sm, Fm	TI	5	Rh + T ^o sensor
BME280	P sensor	1110 11	2	Sm, Fm, Hsm	Bosch	n.a.	P, Rh & T ^o
LIS302DL	MEMS-3D-g	0011 00	2	Sm, Fm	ST	n.a.	3D- accelerometer

Забележки:

1. Брой еднотипни схеми, които могат да се адресират в една I²C шина
2. A0 се използва вътрешно за избор на страница от масива памет
3. Адреса се задава в процеса на производство
4. Възможен е конфликт с някои типове ADC
5. Адреса е в зависимост от свързването на A0 към GND, Vdd, SDA или SCL

– **Сензори:**

Едни от най-често използваните компоненти за разширяване на системите, които се свързват към микроконтролерите посредством шината I²C са сензорите.

Предвид огромното им разнообразие те не могат да се изброят, само за пример могат да се посочат някои от тях:

- ▲ Температурен сензор TCN75;
- ▲ Сензор за влажност Rh – HDC1080;
- ▲ Сензор за налягане P – BME280;
- ▲ Микроелектронни механични сензори - MEMS – напр. LIS302DL за ускорение и др.

Някои техни параметри, касаещи връзката им с шината, са показани в табл.2.8.

ЛИТЕРАТУРА

- [1]. Crisp J., Introduction to Microprocessors and Microcontrollers. Elsevier, Linacre House, Jordan Hill, Oxford. 2004 - Second Edition. ISBN 0 7506 5989 0
- [2]. Hintz K., Tabak D., [Microcontrollers: Architecture, Implementation, and Programming](#), McGraw-Hill Inc, 1992. – 287 p-p. ISBN 0 7506 5989 0
- [3]. Rafiquzraman, M. [Fundamentals of digital logic and microcomputer design](#). JOHN WILEY & SONS, INC., PUBLICATION. 2005. 5th ed., - 840 p-p. ISBN 0-471-72784-9
- [4]. Барретт С. Ф., Пак Д. Дж. [Встраиваемые системы. Проектирование приложений на микроконтроллерах семейства 68HC12 / HCS12 с применением языка С](#). — М.: Издательский дом «ДМК_пресс», 2007. — 640 с. ISBN 5-9706-0034-2
- [5]. Лавентал, Л. Въведение в микропроцесорите: апаратно и програмно осигуряване, програмиране. ДИ „Техника,, – София, 1982 г. превод от английски
- [6]. Точи, Р., Л. Расковски. Микропроцесори и микрокомпютри: апаратно и програмно осигуряване. ДИ „Техника,, – София, 1982 г. превод от английски
- [7]. 16-BIT NMOS MICROPROCESSOR 8086/8086-2/8086-1
- [8]. Texas Instruments. [MSP430x5xx and MSP430x6xx Family. User's Guide](#). Literature Number: SLAU208N. June 2008–Revised May 2014
- [9]. Михов Г., Цифрова схемотехника, ТУ-София, 1999.
- [10]. Park John, St. Mackay, Ed. Wright. Practical Data Communications for Instrumentation and Control. Elsevier Science Ltd, The Boulevard, Langford Lane, Kidlington, OX5 1 GB, England, 2003.
- [11]. Денисенко, В.В., Компьютерное управление технологическим процессом, экспериментом, оборудованием. – Москва, Горячая линия – Телеком, 2009. – 608с. ISBN 978-5-9912-0060-8
- [12]. Wilmshurst Tim. Designing embedded systems with PIC microcontrollers - principles and applications. Elsevier Science Ltd, The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, UK, First edition 2010.
- [13]. UM10204. I²C-bus specification and user manual. NXP Semiconductors. 2014
- [14]. Парк Дж., Маккей С., Райт Э. Передача данных в системах контроля и управления: практическое руководство; [перевод с англ. В.В. Савельева]. – М.: ООО „Группа ИДТ“, 2007. – 480 с. ISBN 978-5-94833-023-5
- [15]. AT24C32/64 Data sheet, 2-Wire Serial EEPROM. Rev. 0336G–04/01 Atmel Corporation 2001.
- [16].

СЪДЪРЖАНИЕ

УВОД	3
ГЛАВА 1. МИКРОПРОЦЕСОРИ И МИКРОКОНТРОЛЕРИ.....	4
1.1. ВЪВЕДЕНИЕ. МИКРОПРОЦЕСОРНИ СИСТЕМИ – СЪЩНОСТ И ОРГАНИЗАЦИЯ	4
1.2. АРХИТЕКТУРА И СТРУКТУРА НА МИКРОПРОЦЕСОРНИ СИСТЕМИ	5
1.2.1. Архитектура на микропроцесорните системи.....	5
1.2.2. Структурна схема на микропроцесорна система	6
1.2.3. Методи за надграждане на системите – системни магистрали	9
1.2.4. Механизми за управление на паметта в системата.....	10
1.2.5. Механизми за директен достъп до паметта.....	10
1.3. ВЪТРЕШНА СТРУКТУРА И НАЧИН НА РАБОТА НА МИКРОПРОЦЕСОРИ И МИКРОКОНТРОЛЕРИ.....	10
1.3.1. Какво е микропроцесор ?.....	11
1.3.2. Какво е микроконтролер?	11
1.3.3. Микропроцесорни фамилии и фамилии микроконтролери	12
1.3.4. Архитектурни особености и структура на микропроцесорите	12
1.3.5. Как функционира микропроцесорът?.....	15
1.3.6. Примери за структура на микропроцесори и ядра на микроконтролери.....	16
1.3.7. Програмен модел на микропроцесорите	19
1.3.8. Специфични функционални особености на микропроцесорите	22
1.3.9. Режими на работа на микропроцесорите.....	23
1.4. ОБМЕН НА ДАННИ С ПРОГРАМАНА И ОПЕРАТИВНА ПАМЕТ И ИНТЕРФЕЙСНИ СХЕМИ.....	23
1.5. СИСТЕМИ ЗА ПРЕКЪСВАНИЯ ПРИ МИКРОПРОЦЕСОРИ И МИКРОКОНТРОЛЕРИ	25
1.6. КАРТА НА АДРЕСНОТО ПОЛЕ. ДЕКОДЕРИ НА АДРЕСИТЕ.	28
1.6.1. Симетрични дешифратори на адресното поле	28
1.6.2. Несиметрични дешифратори на адресното поле	29
1.6.3. Непълно дешифриране на адресите	30
1.7. СИСТЕМИ ИНСТРУКЦИИ.....	31
1.7.1 Видове системи от инструкции.....	31
1.7.2 Типове инструкции.....	32
1.7.3 Формати на данните	33
1.8. ВИДОВЕ АДРЕСАЦИИ.....	33
1.8.1. Непосредствена адресация.....	33
1.8.2. Регистрова адресация	34
1.8.3. Абсолютна или разширена адресация	34
1.8.4. Индексна адресация	34
1.8.5. Индексна адресация с пре- декремент или инкремент.....	34
1.8.6. Индексна адресация с пост- декремент или инкремент.....	35
1.8.7. Индиректна адресация.....	35
1.8.8. Относителна адресация	35
1.9. РЕЖИМИ НА РАБОТА С ПОНИЖЕНА КОНСУМАЦИЯ.....	35
1.9.1 Управление на работната честота	36
1.9.2 Управление на захранващото напрежение	36
1.9.3 Управление на набора от активирани ресурси в системата	36

1.9.4 Спиране на ядрото на микроконтролера и периферни схеми	37
ГЛАВА 2. СЕРИЙНИ ИНТЕРФЕЙСИ В МИКРОПРОЦЕСОРНИТЕ СИСТЕМИ.....	38
2.1. АСИНХРОННИ СЕРИЙНИ ИНТЕРФЕЙСИ	38
2.2.1. Асинхронен сериен интерфейс SCI	39
2.2.2. Асинхронен интерфейс RS232	48
2.2.3. Асинхронен интерфейс RS422/RS485.....	60
2.2.4. Токов кръг 4-20 mA.....	70
2.2.5. Асинхронен интерфейс LIN (Local Interconnect Network).....	71
2.2.6. Модеми	74
2.2.7. Асинхронен полев интерфейс HART.....	76
2.2.8. Асинхронен интерфейс M-bus.....	80
2.2.9. Предаване на данни по радиоканал.....	84
2.3. ЛОКАЛНИ КОМПЮТЪРНИ МРЕЖИ НА БАЗАТА НА АСИНХРОННИ ИНТЕРФЕЙСИ	87
2.4. СИНХРОННИ СЕРИЙНИ ИНТЕРФЕЙСИ.....	93
2.4.1. Сериен периферен интерфейс SPI	94
2.4.2. Синхронен сериен интерфейс I ² C.....	103
ЛИТЕРАТУРА.....	155