Микропроцесорна схемотехника

Микропроцесорите са програмно управлявани електронни цифрови устройства в интегрално изпълнение, извършващи логически и аритметични операции и предназначени за обработка на информация, приемане и предаване на данни, генериране на сигнали и управление.

Около 1970г. се създават условия за появата на първите микропроцесорни системи. По това време са разработени и намират успешно приложение миникомпютрите и калкулаторите; развива се микроелектрониката и свързаните с нея технологии за производство на различни видове електронни елементи; усвоено е производството на логически и цифрови схеми в интегрално изпълнение.

През юни 1970г. се появява първият интегрален многочипов микропроцесор F14 CADC, предназначен за вграждане в авиационен компютър.

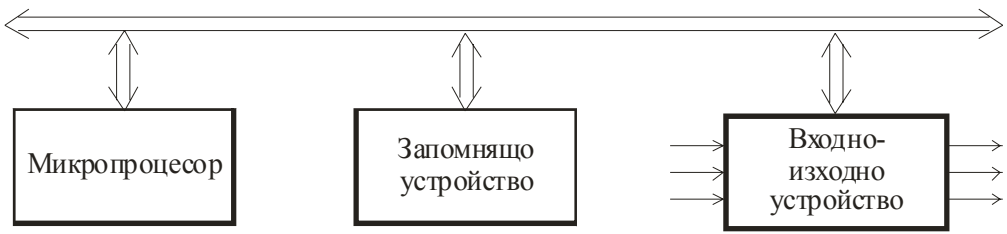
На 17 септември 1971г. фирмата Texas Instruments представя 4-битовия микропроцесор TMS1802NC,

на 15 ноември 1971г. излиза първият едночипов 4-битов микропроцесор Intel 4004,

през април 1972г. – първият 8-битов Intel 8008.

по-късно се появяват популярните и масово използвани 8-битови микропроцесори: Intel 8080 – април 1974г., Motorola 6800, MOS Technology 6502 –1975г., Zilog Z80–1976г.

Блокова схема



Фиг. 5.1.

Най-простата структурна схема (фиг. 5.1) на една микропроцесорна система включва микропроцесор, запомнящо устройство и входно-изходни устройства, които са реализирани като една или няколко големи интегрални схеми, съдържащи по-малко или повече от стотици хиляди полупроводникови елементи – основно транзистори.

В микропроцесора се изпълняват основните функции по управление и реализация на логическата и аритметична обработка.

В една част от запомнящото устройство се съхраняват програмите.

Данните или променливите се съхраняват в друга част от запомнящото устройство или постъпват от входното устройство.

В зависимост от това дали информацията в запомнящото устройство може да се променя от микропроцесора то бива ROM (**R**ead **O**nly **M**emory) – памет само за четене – или RAM (**R**andom **A**ccess **M**emory) – памет със свободен достъп за четене и запис.

ROM запомнящите устройства могат да бъдат еднократно програмируеми – PROM, или многократно електрически програмируеми – EPROM.

Запомнящо устройство от типа EPROM, което може електрически да бъде изтривано, се означава с EEPROM.

ROM запомнящите устройства са енергонезависими – при отпадане на захранващото напрежение записаната в тях информация се запазва и при подаване отново на захранващо напрежение тя е достъпна за четене.

При RAM запомнящите устройства, за да се запази информацията при отпадане на захранването, трябва да се осигури резервен постояннотоков захранващ източник – батерия или акумулатор.

В микропроцесорните системи се използва принципът на програмното управление на изчислителния процес.

Изчислителният процес се разделя на ***операции***, които се управляват с помощта на ***команди***.

Записът на командите в паметта се нарича ***програма***.

Командите се реализират последователно една след друга.

Изпълнението на всяка команда обхваща цикъл от няколко фази.

През първата фаза се прочита командата (от клетка в паметта, адресирана от брояча на команди на микропроцесора, се извлича нейното съдържание и се записва в регистъра на командите на микропроцесора).

В следващата фаза съдържанието на регистъра се преобразува от дешифратор на командите в набор от управляващи сигнали. След това се увеличава с единица съдържанието на брояча на команди.

През последната фаза се изпълнява командата.

За следващата команда цикълът се повтаря.

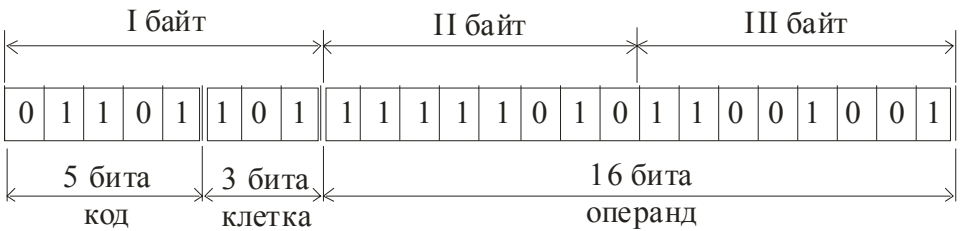
Командата представлява двоично число, което съдържа определен брой разряди (битове).

Използват се команди с различни дължини.

Броят на разрядите им се избира да е кратен на 8.

Най-късата команда се записва с число, съдържащо 8 бита.
Последователност от 8 бита се нарича ***байт***.
Данните също се избират да са с дължина, кратна на 8 бита.

На фиг. 5.2 е показан пример за структурата на една команда.



Фиг. 5.2.

Командата е образувана от три елемента:

– ***код*** на операцията – в случая се задава с 5 бита;

– адрес на ***клетка***, регистър в който се съдържа първият операнд и в който ще се намира резултатът от операцията – в примера се използват 3 бита;

– ***операнд*** – в предвидените за него разряди (на фигурата 16 бита) се записва стойността на втория операнд или адресът на клетка от паметта, в който се намира.

Записът в двоичен код на съдържанието на елементите на командата е:
код - %**01101**;
клетка - %**101**;
операнд - %**1111 1010 1100 1001**;

а в шестнадесетичен код:
код - $ **0D**; клетка - $**5**; операнд - $**FAC9**.

Със символите “%”, “$” е означено в какъв код е извършен записът.

Ако се сравнят двата записа (в двоичен и в шестнадесетичен код) се вижда, че този в шестнадесетичен код е по-кратък и е по лесен за четене.

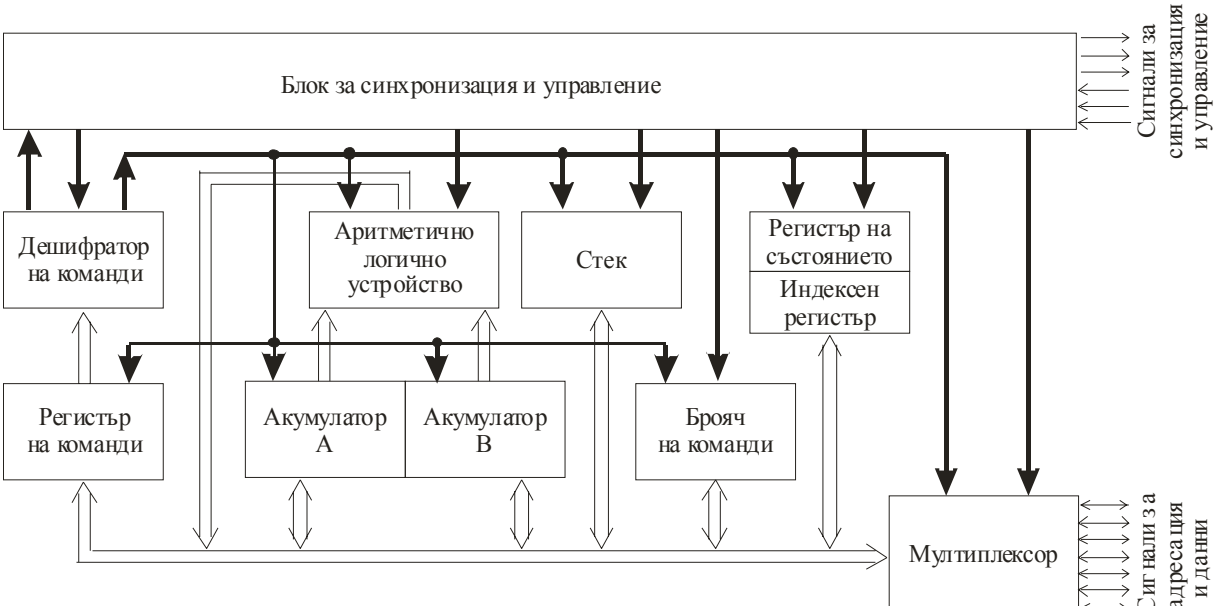
Броят на разрядите на командата са общо 24 и се записват в три последователни еднобайтови клетки от паметта, предназначена за съхраняване на програмата.

Описът на съдържанието на паметта би изглеждал така:

Адрес	Съдържание на клетката
...	...
$ **0300**	$ **6B**
$ **0301**	$ **FA**
$ **0302**	$ **C9**
…	**…**

На адрес $**0300** е записан първият байт на командата, образуван от разрядите на кода и клетката, на следващите $**0301** и $**0302** – двата байта, образувани от разрядите на операнда.

Архитектура на микропроцесора



Фиг. 5.3.

Микропроцесорът е основният блок на една микропроцесорна система. Неговата архитектура определя основните характеристики и параметри на системата. От показаната на фиг. 5.3 примерна блокова схема може да се добие представа за структурата на микропроцесора и взаимодействието между основните блокове:

Брояч на команди (Програмен брояч). Командите, съдържащи се в програмата, се съхраняват в програмната памет последователно една след друга. Задачата на брояча на команди е да зададе последователността на четене на командите и да съхранява във всеки момент адреса на командата, която предстои да се изпълни. Често в програмата се съдържат участъци от програмата – подпрограми, които се извикват многократно при изпълнение на основната програма и това налага промяна на последователността на изпълнение на командите.

Ако в програмата се появи команда “извикване на подпрограма”, адресът на следващата команда от основната програма трябва да се запази временно (до завършване изпълнението на командите, включени в подпрограмата) в някакъв регистър и едновременно с това да се запише в брояча на команди адресът на първата команда от подпрограмата. Като регистър за временно ползване се използва безадресна стекова памет (***стек***), която съдържа набор от последователни клетки от паметта. Клетките се използват последователно от най-

големия адрес към по-малките, а се освобождават от най-малкия адрес към по-големите. За съхранение на адреса на подлежащата за ползване свободна клетка на стека се използва ***указател на стека***.

При изпълнението на командата "извикване на подпрограма" следващата команда от основната програма се записва в клетка от паметта, чийто адрес се съдържа в указателя на стека, и се намалява съдържанието на указателя на стека, за да сочи следващата свободна клетка. Ако при изпълнението на подпрограмата се извиква друга подпрограма се извършват отново описаните по-горе действия. Допустимият брой на последователно извикваните подпрограми, без да се изгуби адресът на връщане, се определя от дълбочината на стека (броя на неговите клетки). Когато завърши изпълнението на подпрограмата, трябва да се осъществи преход към команда от изходната програма. В брояча на команди се зарежда съдържанието на последната заета клетка от стека и се увеличава съдържанието на стека.

Регистър на командите. При извличане на команда от паметта тя се записва в регистъра на командите и се съхранява в него докато бъде дешифрирана. Дължината на командата зависи от типа на микропроцесора и в повечето случаи тя е променлива, кратна на 8 бита. Променливата дължина позволява освен кода на операцията в командата да се запишат и адресите на една или повече променливи, които ще бъдат прочетени от или записани в паметта.

Дешифратор на командите. Задачата на дешифратора на командите е да разпознае кода на операцията, която ще се изпълнява, и да изработи сигнали за задаване на функциите на блоковете на микропроцесора и прехвърляне на информация между тях. Обикновено операциите се разделят на групи (например “аритметични операции”, “логически операции”, “операции за прехвърляне на данни” и други). Стойността на старшите битове определя типа на групата, към която се числи дадената команда. Това води до по-лесно извършване на дешифрирането.

Блок за синхронизация и управление (БСУ). Изработва тактови сигнали за всички устройства, участващи в микропроцесорната система. Честотата на тези сигнали се избира в зависимост от бързодействието на устройствата, към които се подават, така, че те да работят надеждно без грешки. При работа с външни устройства, при някои типове микропроцесори, продължителността на отделни тактове може да се увеличава с цел съгласуване на ниското бързодействие на външното устройство с това на микропроцесора.

Управлението на изпълнението на операциите се извършва съвместно с дешифратора на командите. Дешифраторът определя броя и вида на фазите, а БСУ задава тяхната последователност, като изработва необходимите управляващи сигнали.

Към функциите на БСУ се отнася и отработване на заявки за прекъсване. При определени условия (например спадане на захранващото напрежение) нормалната работа на микропроцесора (последователното изпълнение на командите от програмата) е необходимо да бъде прекъсната по сигнал от външно устройство – източник на заявката за прекъсване. Преминава се към отработване на заявката: завършва се текущата операция; в стека се записва съдържанието на всички регистри, включително и на брояча на командите; в брояча на командите се записва адресът на подпрограмата, зададен от източника на прекъсване; изпълняват се последователно командите от подпрограмата на прекъсване; след нейното завършване се възстановява съдържанието на брояча на команди и на регистрите и се продължава с изпълнение на командата, следваща командата (точката), където е получено прекъсването.

Акумулатор. Акумулаторът е регистър, съхраняващ операнда, който ще участва при изпълнението на дадена операция от аритметично-логическото устройство (АЛУ). Той е основен работен регистър, в който се записват данни от паметта, входните устройства и от АЛУ и съответно се извеждат към паметта и изходните устройства. Почти всички микропроцесори имат повече от един акумулатор.

Аритметично-логическо устройство (операционен блок). Аритметично-логическото устройство е основен блок на микропроцесора и определя неговите

функции, възможности и бързодействие. То изпълнява най-малко следните операции: целочислено събиране с пренос, изваждане със заем, побитово преместване вляво и вдясно, броене напред и назад, логическо събиране (ИЛИ) и умножение (И); сравнение на двоични числа. По-сложните АЛУ изпълняват и други операции като целочислено умножение и деление, аритметични действия с плаваща запетая, логически операции освен И и ИЛИ и др.

Регистър на състоянието. Съставен е от определен брой еднобитови регистри (флагове), в които се съхранява или записва информация за състоянието на микропроцесора, за резултатите от извършените операции, за забрана и разрешение на определени функции. Например:

– ако при извършване на събиране се получи препълване, се вдига флагът за препълване (АЛУ записва 1 в съответния еднобитов регистър). Тая информация се използва от програмата като условие за отработване на преноса.

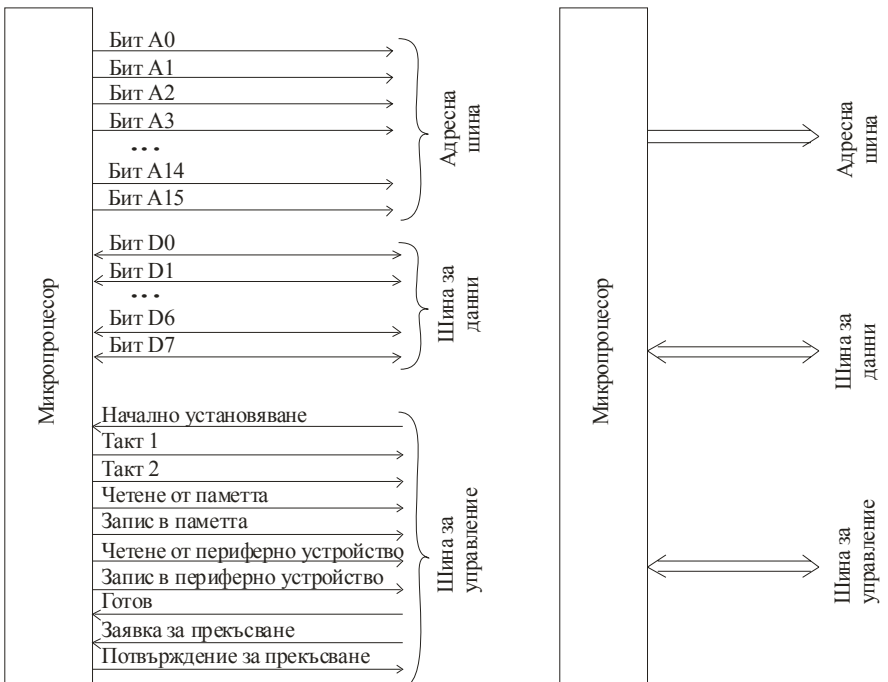
– програмата може да блокира изпълнението на заявка за прекъсване, като запише във флага за управление на прекъсването 1 и докато не бъде записана 0, няма да се изпълнява прекъсването. След премахване на флага прекъсването се изпълнява.

Индексен регистър. При някои операции (обикновено при работа със паметта) операндите се задават индексно. В индексния регистър се съдържа базов адрес, а в командата – отместването спрямо него. При извличане от

паметта на стойността на операнда истинският адрес се изчислява като сума от базовия адрес и отместването. Чрез команди съдържанието на индексния регистър може да се увеличава и намалява с 1. Съвременните микропроцесори имат повече от един индексен регистър.

Канал за пряк достъп до паметта КПДП. Чрез КПДП се осигурява механизъм за прехвърляне на пакет от данни от външно устройство в паметта и обратно без прякото участие на микропроцесора. Паралелно с прехвърлянето на данните процесорът продължава да отработва командите от основната програма.

Входни и изходни сигнали на микропроцесора



а) б)

Фиг. 5.4.

За връзка с другите блокове на микропроцесорната система (програмна памет, памет за данни, регистри за въвеждане и извеждане на данни) микропроцесорът се свързва с тях чрез набор от проводници. Проводниците, по които се пренасят еднородни сигнали, се групират в шини. Пример за организация на изводите на микропроцесора е показана на фиг. 5.4.

Двупосочна ***информационна шина (шина данни)*** се използва за прехвърляне на данни между микропроцесора и паметта и периферните устройства. Броят на паралелно предаваните разряди по шината на данни обикновено определя и разрядността на микропроцесора (4-битов, 8-битов, 16-битов). С помощта на еднопосочна ***адресна шина*** се адресират паметта и периферните устройства. В някои микропроцесори за адресната шина не се предвиждат отделни изводи и тя се мултиплексира с шината за данни. Ако адресните разряди са повече от разрядите на шината за данни, те се предават на повече тактове. При микропроцесорите, в които има мултиплексиране на адресната шина с шината за данни, се налага да се използват извън микропроцесора допълнителни демултиплексиращи схеми за разделяне на двете шини.

Управляващата шина се състои от проводниците, по които се предават управляващи сигнали. Много от микропроцесорите нямат изводи за всеки от управляващите сигнали, а те се мултиплексират с изводите за шината за данни

или помежду се. На фиг.5.4а са изчертани всички проводници на примерен 8-битов микропроцесор с 8-битова шина за данни и 16-битова адресна шина. На фиг.5.4б изводите са дадени като три шини – адресна, за данни и за управление.

Различават се два начина за организация на шините на микропроцесора. При единия от начините се използва едношинна структура (single bus system), при него адресната, данновата и управляващата шина се използват за връзка между всички устройства на микропроцесорната система – микропроцесор, памет, периферни устройства. При втория начин (двушинна структура, twu bus system) се използват две групи от шини – едната за връзка между микропроцесора и паметта, а втората – за връзка между микропроцесора и периферните устройства.

Работа с паметта и външните устройства

В запомнящото устройство се съхраняват програмата и данните, с които работи микропроцесорът.

Програмата се съхранява в памет само за четене (ROM), а данните – в памет за четене и запис (RAM).

Запомнящото устройство се изгражда от специализирани ИС ROM и RAM, характеризиращи се с организацията и обема информация, която могат да съхраняват.

Най-често паметта е организирана на клетки с по 8 бита. Ако ИС има 128 клетки по 8 бита, общият обем памет е 1024 бита или 1k памет.

В момента съществуват ИС ROM и RAM с обем и организация съответно 32k (4k x 8), 64k (8k x 8), 128k (16k x 8), 256k (32k x 8), 512k (64k x 8).

Обемът и организацията на паметта определят броя на изводите на ИС.

При 8-битова клетка шината за данни ще бъде също 8-битова – с 8 проводника.

За адресиране на 1k клетки е необходима 10-битова адресна шина.

Освен това има вход за избор (CS), с който се активизира ИС.

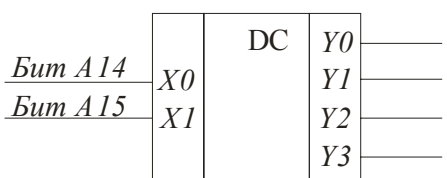
Активизирането на ИС, свързани към шината за данни, се налага от това, че нормално техните изходи са в трето състояние (имат много високо изходно съпротивление).

След активизирането изходите на ИС повтарят състоянието 0 или 1 на съответния вътрешен регистър.

Схемите RAM имат и вход (R/W) за указване на режима на четене или запис.

Таблица 5.1

A15	A14	Y1	Y2	Y3	Y4
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



Фиг. 5.5.

В една микропроцесорна система трябва да има поне една ИС ROM и една ИС RAM, които се свързват паралелно към трите шини – адресна, за данни и за управление.

При едношинна структура адресното пространство (обемът му се определя от общия брой на клетките, адресирани от микропроцесора) е общо за ROM, RAM, входни и изходни устройства.

В него клетките памет на ИС трябва да се разположат така, че да не се препокриват (да са на различни адреси).

В противен случай при опит за четене ще се задействат едновременно две ИС и изходите им ще се явят свързани паралелно, а това ще доведе до получаване на грешна информация.

Следователно входовете за избор на ИС трябва да не се задействат едновременно.

Това се осигурява от адресни дешифратори, които разделят адресното пространство на области, съответстващи на обема на ИС.

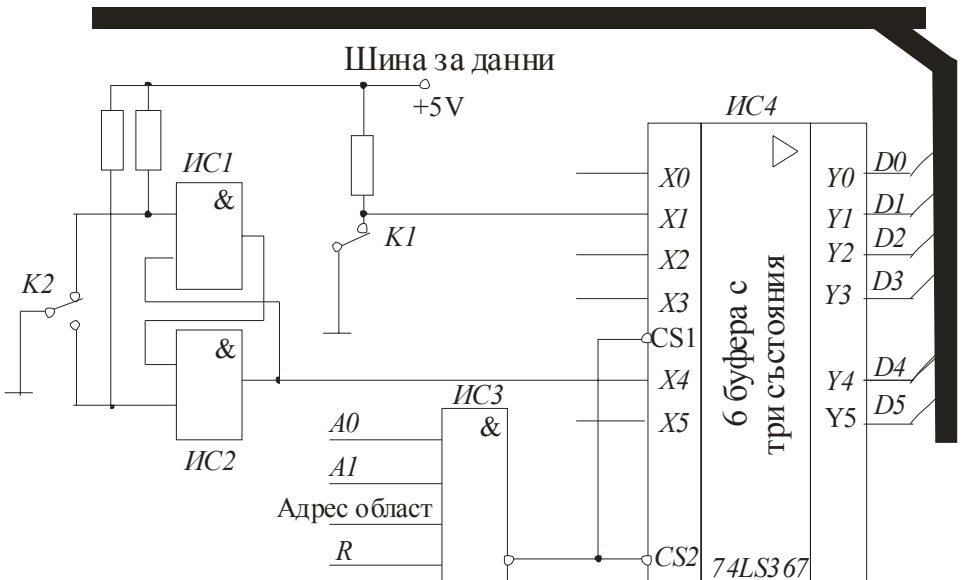
Ако използваме двата най-старши бита от адресната шина и ги подадем на двуразряден дешифратор, всеки от четирите му изхода ще се задейства при едно от четирите състояния на входните сигнали (таблица 5.1).

Задействането на изходен сигнал на дешифратора (фиг. 5.5) може да се използва за активизиране на устройство, еквивалентно на 16k клетки памет, ако общият обем памет е 64k (микропроцесорът има 16 битова адресна шина).

За разделянето на адресното пространство на по-малки области се използват няколко дешифратора, свързани един след друг, или един с повече входове.

Входните и изходните устройства, свързани към микропроцесорната система, са еквивалентни на една или няколко клетки от адресното пространство и протоколът за обмен на информация между тях и микропроцесора е същият, както с паметта.

На фиг. 5.6 е показана схема за въвеждане в микропроцесора на информация за състоянието на ключове *K1* и *K2*.

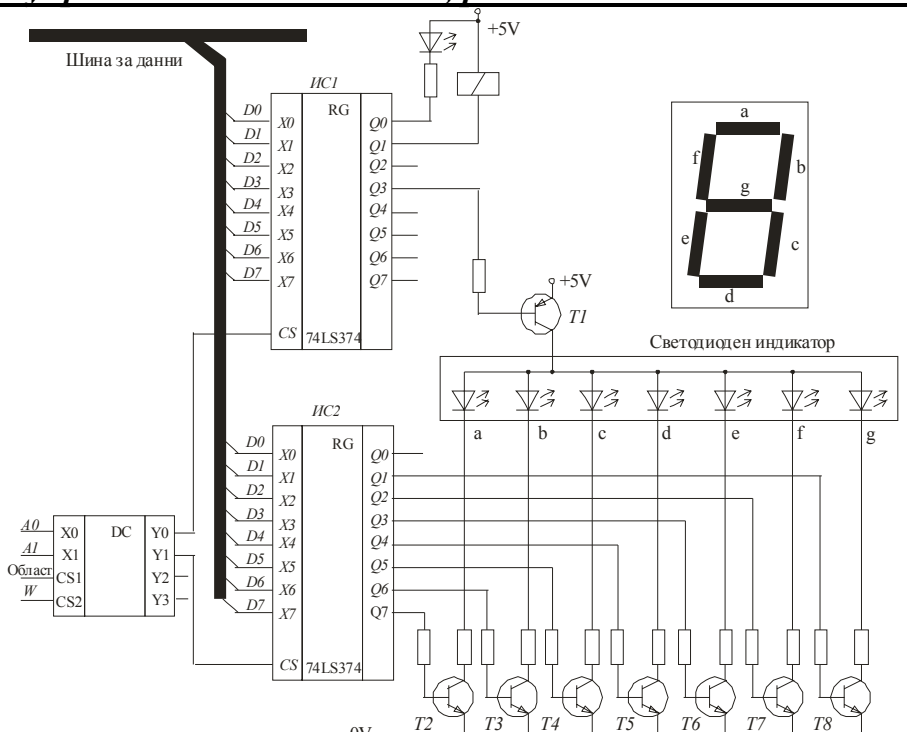


Фиг. 5.6.

Единият полюс на ключа К1 е свързан през резистор към +5V, а другият – към маса. При отворен ключ към вход *X1* на 6-входовия буфер с изходи с три

състояния с типово означение 74LS367 ще се подава логическа 1 (+5V), а при затворен – логическа 0 (0V). При подаване на входа *CS1* и *CS2* на логическа 0 буферът (*ИС4*) ще излезе от трето състояние и ще прехвърли информацията за ключа *К1* на изхода *Y1* и през проводника *A1* от шината за данни тя ще постъпи в микропроцесора. За да не се предават вибрациите на ключа, за формиране на сигнала за състоянието на ключа *К2* е използван RS-тригер, изпълнен с две схеми на съвпадение И (*ИС1* и *ИС2*). Сигналът за избор на буфера се изработва от схема на съвпадение И-НЕ (*ИС3*), на чиито входове се подават сигнала за избор на област от адресното пространство, най-младшите адреси *A0* и *A1* и сигналът за четене *R*.

Схема за управление на светодиод, реле и седемсегментна индикация.



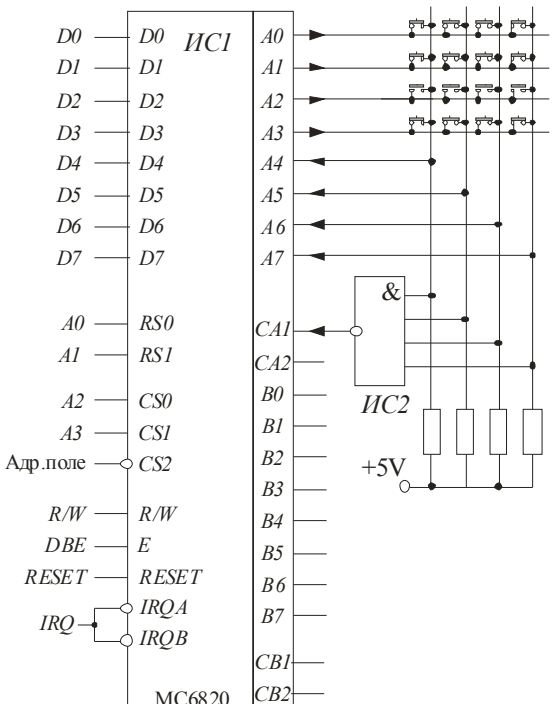
Фиг. 5.7.

За управлението на светодиода и релето микропроцесорът трябва да запише в съответния тригер на 8-битовия регистър *ИС1* (тип 74LS374) логическа 0 или 1. При запис на 0 светодиодът започва да свети, а релето се задейства. При запис на 1 светодиодът не свети, а релето не е задействано, тъй като потенциалът на изходите на регистъра, когато е записана 1, е близък до +5V и ток през веригата не тече.

За управление на седемсегментния светодиоден индикатор с общ анод е необходимо да се управлява състоянието на транзисторите *T1 – T8*. При отпушен транзистор T1 се подава напрежение +5V към общия анод на светодиодите. При отпушване на някой от транзисторите T2 – T8 през ограничителен резистор се подава потенциал 0V към съответния катод на сегмент от индикатора. За да свети сегментът е необходимо да е отпушен и транзисторът *T1*.

Транзисторът *T1* се отпушва при състояние 0 на изход *Q3* на регистъра *ИС1* (транзисторът *T1* е PNP тип и за отпушването му е необходим отрицателен потенциал спрямо емитера), транзисторите $T_2 – T_8$ се отпушват при състояние 1 на изходите Q_1 - Q_7 на регистъра *ИС2* (те са NPN тип и за отпушването им е необходим положителен потенциал на базата спрямо емитера).

Схема за въвеждане на информация от клавиатура



Фиг. 5.8.

Съществуват универсални паралелни интерфейсни схеми, чиито изводи могат да се програмират като входове или изходи чрез вграден в тях допълнителен регистър за програмиране на посоката на предаване на информацията на всеки извод. На фиг. 5.8 е показана схема за въвеждане на информация в микропроцесора от клавиатура с 16 бутона, която използва универсален паралелен интерфейсен адаптер (PIA) тип 6820 от микропроцесорната фамилия 6800 на фирмата MOTOROLA.

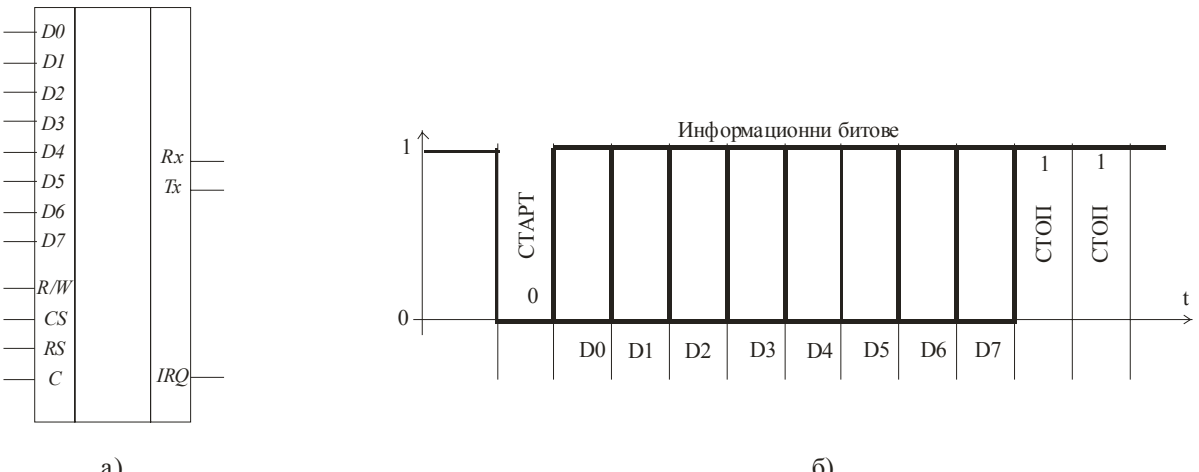
Схемата 6820 има две групи по 8 извода, които могат да се програмират като входове или изходи. Тя заема от адресното пространство на микропроцесора четири клетки, които се избират чрез адресите *A0* и *A1*, свързани към входовете *RS0* и *RS1*. За активизирането на схемата са предвидени три входа *CS0, CS1* и *CS2*, на които трябва да се подадат едновременно логическа 1 на *CS0* и *CS1* и логическа 0 на *CS2*. Тези сигнали се формират от старшите адреси. Записването и четенето се управлява от сигнала *R/W*, изработван от микропроцесора. За управление на работата на PIA са необходими сигналът *RESET* за начално установяване и сигналът *E* за синхронизиране. ИС PIA има още два изхода *IRQA* и *IRQB*, които генерират прекъсване към микропроцесора при постъпване на сигнали на изводите *CA1, CA2, CB1* и *CB2*.

Клавиатурата се свързва към страна А на ИС PIA с две групи (хоризонтална и вертикална) от по 4 проводника. В пресечните точки са свързани бутоните.

Чрез изводите *A0-A3*, програмирани като изходи, се подава към хоризонталните проводници логическа 0. Изводите *A4-A7* са програмирани като входове и получават логическа 1 от свързаните към тях вертикални проводници, към които през резистори е подадено напрежение +5V.

Когато се натисне някой от бутоните на един от вертикалните проводници се изработва логическа 0, която през схемата И-НЕ (*ИС2*) се подава като 1 към входа за прекъсване *CA1*. Изходът *IRQA* изработва заявка за прекъсване и процесорът го отработва, като извиква подпрограма за разпознаване на натиснат бутон. Алгоритъмът на подпрограмата е следният: Микропроцесорът записва на изводите *A0-A3* логическа 1 и започва последователно да подава логическа 0 на изходите *A0-A3* и се проверява има ли логическа 0 на някой от входовете *A4-A7*. По стойността на двоичното число, описващо състоянието на изводите *A0-A7*, се разпознава кой бутон е натиснат.

Асинхронен сериен интерфейсен адаптер



а) б)

Фиг. 5.9.

Освен паралелни има и последователни интерфейсни схеми, които се използват за комуникация с други микропроцесорни системи или с периферийни устройства. Информацията при тях се извежда побайтово последователно бит след бит. Пред информационните 8 или 7 бита (фиг. 5.9 б) се предава 1 стартов бит, а след тях – 1 или 2 стопови бита. Стартовият бит е логическа 0, а стоповите – логическа 1. На фиг.5.9а е показано условното

означение на ИС асинхронен сериен интерфейсен адаптер (ACIA) тип MC6850 от микропроцесорната фамилия 6800 на фирмата MOTOROLA. При получаване паралелно от микропроцесора на изводите *D1-D8* един байт той се преобразува в последователен формат, допълва се със стартов, контролен и стопови битове и се изпраща през извод *Tx* към периферното устройство. Получената информация на извод *Rx* се преобразува от последователен в паралелен формат и през изводите *D1-D8* се чете от микропроцесора.

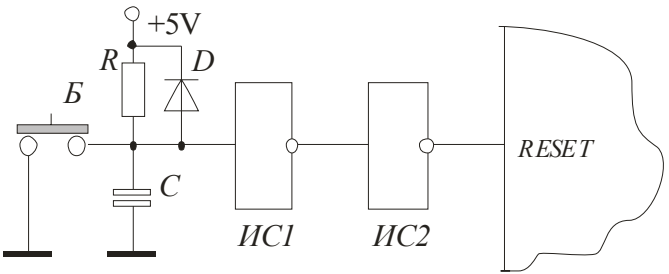
ИС ACIA заема от адресното пространство на микропроцесора две клетки, които се избират чрез адреса *A0*, свързан към входа *RS*. За активирането на схемата са предвидени три входа *CS0, CS1* и *CS2*, на които трябва да се подадат едновременно логическа 1 на *CS0* и *CS1* и логическа 0 на *CS2*. Тези сигнали се формират от старшите адреси. Записването и четенето се задава от сигнала *R/W*, изработван от микропроцесора. За синхронизиране на работата на PIA е необходим сигналът *E*. ИС ACIA има още два входа *TxC* и *DxC* за подаване на синхранизиращи импулсни сигнали за изхода *Tx* и входа *Dx*. Има изход *IRQ*, който генерира заявка за прекъсване към микропроцесора.

Освен чрез изводите *Tx* и *Dx* ИС ACIA понякога се свързва с периферното устройство и с изводите *RTS* (изход “заявка за предаване”; ACIA информира периферното устройство с предаване на логическа 0, че е готово да предава);

CTS (вход “свободно за предаване”; с предаване на логическа 0 периферното устройство информира АСІА, че е готово да приема); *DCD* (вход “открит носител на информация”; с предаване на логическа 1 периферното устройство информира АСІА, че е спряло предаването).

Работа на микропроцесора

Както беше казано по-горе, микропроцесорът работи под управлението на записаната в паметта програма. Адресът, от който се чете поредната команда, се съхранява в програмния регистър. При стартиране на микропроцесора (след подаване захранващо напрежение) е необходимо в програмния регистър да се запише началният адрес на програмата. За целта се използва входът "Начално установяване (*RESET*)", на който (примерно за микропроцесор 6800) трябва да се подаде логическа 0 веднага след включване на захранващото напрежение за не по-малко от 8 такта. На фиг. 5.10 е дадена схема за изработване на сигнала за начално установяване при подаване захранващо напрежение или натискане на бутон "*RESET*". След прехода на сигнала "*RESET*" от 0 в единица от адрес $FFFE се прочита началният адрес на програмата и той се зарежда в програмния регистър. По-нататък следва изпълнение на програмата команда след команда.



Фиг. 5.10.

Първите команди от програмата са предназначени за инициализация на микропроцесорната система – установяване на флагове, коефициенти, програмиране на изводите на периферните схеми (като входове или изходи), записване в клетките от RAM паметта началните стойности на променливите, с които работи микропроцесорът.

След блока за инициализация на системата следва основното тяло на програмата (главната програма), което се изпълнява циклично докато се изключи захранването. В края на главната програма се поставя команда за безусловен преход към нейната първа команда. Всички разклонения на главната програма трябва да се връщат към началото на нейна команда (ако командата се изписва с няколко байта, в програмния регистър трябва да се зареди адресът на първия байт.

Микропроцесор, микрокомпютър, микроконтролер

С развитието и усъвършенстването на микропроцесорната техника се появиха нови по-мощни микропроцесори с по-голяма скорост на изпълнение на командите и по-голям брой функционални възли, вградени в една ИС. Степента на интеграция е такава, че почти само с една ИС може да се създаде завършено изделие. Появиха се и понятия като *микрокомпютър и микроконтролер*. По-надолу се прави опит да бъдат обяснени различията между тези понятия.

Микропроцесорът е ИС, част от микропроцесорна система, предназначена да извършва аритметични и логически операции с цел обработка на данни.

Микрокомпютърът е ИС, включваща в себе си елементи от микропроцесорна система – микропроцесор, памет RAM и ROM и периферни схеми за управление на клавиатура и визуализация на данни. Предназначена е за обработка на информация.

Микроконтролерът е ИС, включваща в себе си микропроцесор, памет RAM и ROM и периферни схеми за въвеждане и извеждане на логически сигнали, АЦП и ЦАП за въвеждане и извеждане на аналогови сигнали, интерфейсни схеми за комуникация. Предназначена е за управление на обекти и технологични операции.

Микрокомпютърът и микроконтролерът могат да работят самостоятелно с минимален брой присъединени към тях елементи.

Приложение на микроконтролерите в електроенергетиката

Микроконтролерите намират широко приложение в измервателната, контролно-управляващата техника, медицината и много други области на човешката дейност. В зависимост от конкретната задача към тях се включват различни по брой и предназначение допълнителни устройства. За увеличаване на паметта към микроконтролера се включват различни по обем и начин на действие памети – оперативна (RAM) памет или енергонезависима (ROM, EPROM, EEPROM, FLASH) памет. За разширяване на динамичния обхват на вградените аналогово-цифрови преобразуватели се добавят мащабиращи усилватели и атенюатори (устройства за намаляване амплитудата на сигналите).

За извеждане на аналогова информация се включват цифрово-аналогови преобразуватели, а за увеличаване броя на цифровите входове и изходи се добавят паралелни или премествящи регистри.

Като пример за приложение на микроконтролер с присъединени към него периферни аналогови и цифрови устройства ще разгледаме схемата на *електронен еднофазен електромер*. Тя може да се раздели на две части – аналогова и цифрова.

Цифровата част (фиг. 5.11) е изградена на базата на микроконтролера MC68HC11 (DD_1) на фирмата Motorola. За производството му е използвана бързодействащата CMOS технология. В него са вградени следните модули:

– 8-канален 8-разряден аналогово-цифров преобразувател;

– таймерна подсистема с входове за измерване честотно-времевите параметри на импулсни сигнали и изходи за генериране на импулсни сигнали с програмно управление на техните честотно-времеви параметри;

– цифрови входове и изходи с общо предназначение;

–асинхронен (*SCI*) и синхронен (*SPI*) серийни комуникационни интерфейси.

Много от изводите на микроконтролера изпълняват повече от една функция и програмно се конфигурират в зависимост от конкретното приложение. Например входовете на аналогово-цифровия преобразувател и на таймерната подсистема могат да се използват като логически входове с общо предназначение. Изходите на таймерната подсистема могат да функционират като логически изходи с общо предназначение. Входовете и изходите на комуникационните интерфейси могат да се използват като логически входове и изходи с общо предназначение.

В схемата на цифровия електронен електромер входовете *ADC0* и *ADC1* на аналогово-цифровия преобразувател се използват за измерване стойностите на консумирания ток и на напрежението на електрическата мрежа. Към входа

ADC4, използван като цифров, се подава сигнал от външен тарифен часовник. Чрез входа *IC1* на таймерната подсистема се измерва честотата на напрежението на мрежата. Изходът *OC2* на таймерната подсистема се използва за генериране на импулси с честота, пропорционална на консумираната електрическа енергия. Посредством инвертора *DD6C* той управлява светодиода VD_4. По този начин информацията за електрическата енергия се извежда във вид на последователност от светлинни импулси.

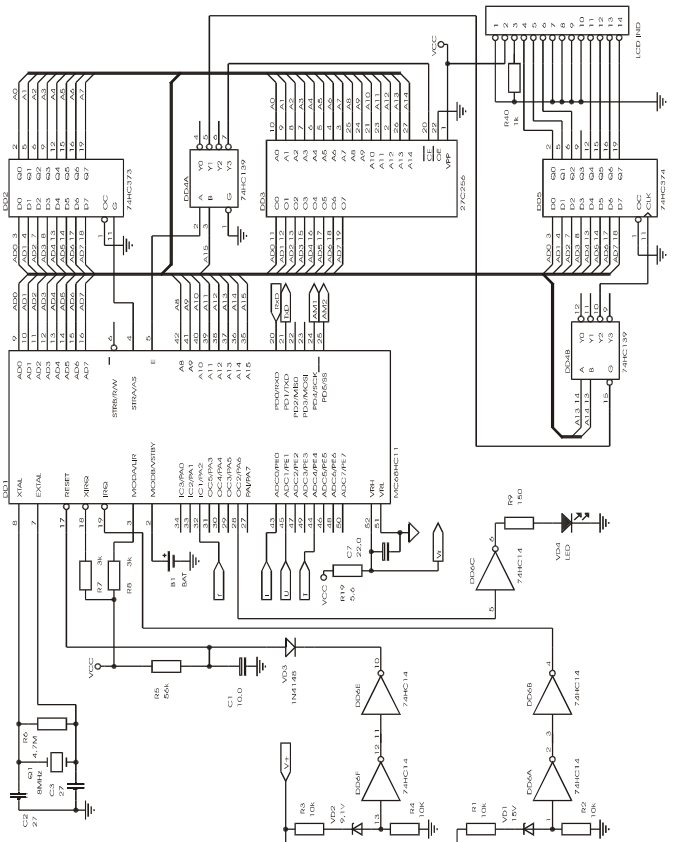
С филтровата група R_{19} и C_7 се формира опорното напрежение V_r, което е необходимо за работата на аналогово-цифровия преобразувател.

Тактовият генератор, който е необходим за работата на микроконтролера, изисква включването на външни пасивни елементи – кварцов резонатор Q_1, резистор R_6 и кондензатори C_2 и C_3. Избрана е тактова честота със стойност 8 MHz.

Сигнал за начално установяване (*Reset*) се формира след включване на захранващото напрежение от закъснителната верига R_5 и C_1. Също така сигнал *Reset* се формира и при отпадане на захранващото напрежение. За тази цел се използват резисторите R_3 и R_4, ценеровият диод VD_2, инверторите DD_{6F} и DD_{6E} , съдържащи тригер на Шмит на входа, и диодът VD_3. Стойностите на елементите са подбрани така, че сигнал *Reset* с активно ниско ниво се формира, когато напрежението V+, което се подава на входа на стабилизатора за захран-

ващо напрежение +5V, спадне под 7,5V. По този начин микроконтролерът спира да работи при стойност на напрежението, по-ниска от допустимата.

Отпадането на захранващото напрежение се следи и от схемата, включваща резисторите R_1 и R_2, ценеровият диод VD_1, и инверторите с тригер на Шмит на входа *DD6A* и *DD6B*. Тя формира сигнал за прекъсване *IRQ* с активно ниско ниво, когато стойността на напрежението на електрическата мрежа стане по-



Фиг. 5.11.

ниско от 187V. Филтровите кондензатори на захранването на електромера трябва да бъдат изчислени така, че времето между формирането на сигналите *IRQ* и *Reset* да бъде достатъчно за записването във вътрешната EEPROM памет на стойностите на измерената електрическа енергия.

Вътрешната оперативна памет при отпадане на захранващото напрежение с цел запазване на информацията се захранва от батерията B_1.

Микроконтролерът работи в разширен режим, при който адресното пространство е с обем 64kB. В това пространство се намират адресите на вътрешните памети RAM и EEPROM и на вътрешните регистри. На външните устройства трябва да бъдат присвоени адреси, които да не съвпадат с адресите на вътрешните памети и регистри.

При разширения режим за намаляване броя на изводите на микроконтролера 8-те линии за данни и 8-те младши адреса се извеждат на едни и същи изводи, но по време на различни фази на тактовия сигнал. За получаване на информация за младшите адреси е включен паралелният регистър DD_2, който изпълнява функцията на демултиплексор. Най-напред на 8-те линии за данни се извеждат младшите адреси и при получаване на високо ниво от изход *AS* те се запомнят в регистъра DD_2 – на изходите му се установяват битовете на 8-те младши адреса. След това се извежда информацията за 8-те старши бита на адреса и заедно с битовете на изходите на регистъра DD_2 се формира 16-

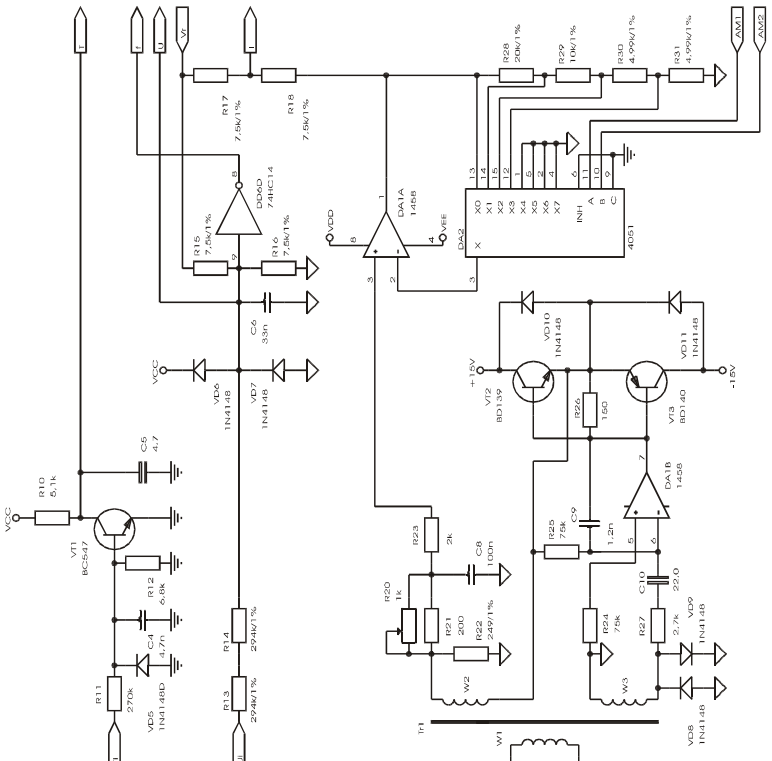
разредният адрес за достъп до клетка от адресното пространство. На изводите *AD*0÷*AD*7 се генерират битовете на данните, които ще се обменят. В този момент на изход *AS* се подава ниско ниво.

Тъй като много рядко се използват всички възможни клетки от адресното пространство, най-често се извършва непълна адресация на това пространство. Това означава, че на едно устройство се задава адрес в област от пространството – устройство може да бъде избрано на повече от един адрес. Непълна адресация е използвана и в схемата на електронния електромер.

Чрез първичния дешифратор *DD4A* адресното пространство се разделя на две половини по 32kB. Началният адрес на първата половина е $0000, а на втората - $8000. Към входовете на дешифратора са свързани най-старшата адресна линия *A15* и изходът *E*. Когато сигналът *E* е в ниско ниво, се извършва вътрешен обмен на информация, а когато е във високо ниво се прави обръщение към външните устройства. Следователно, когато изход *Y1* е в ниско ниво, обръщението е към първата половина от адресното пространство, а когато изход *Y3* е в ниско ниво – към втората половина. Цялата втора половина от адресното пространство е заета от външната EPROM памет *DD3*, която съдържа управляващата програма и необходимите за работата таблици и константи. Векторите за прекъсване на този микроконтролер се намират на най-старшите адреси от паметта. Там се съдържат началните адреси на програмите за начално

установяване и за обработка на прекъсвания. Следователно е задължително в това адресно пространство да има памет и затова постоянната памет е с начален адрес $8000 и краен адрес $FFFF.

Изходът *Y1* на *DD4A* задейства втория дешифратор *DD4B*, който разделя първата половина на адресното пространство на четири области с обем по 8kB и с начални адреси $0000, $2000, $4000 и $6000. Първата област ($0000) не се използва, защото се препокрива с вътрешната RAM памет на микроконтролера и вътрешните му регистри. Изходът *Y2* на дешифратора (адрес $4000) разрешава записа на данни в паралелния регистър *DD5*, който управлява течно-кристалния индикатор (LCD).


R17 7,5k/1%
R18 7,5k/1%
R28 20k/1%
R29 10k/1%
R30 4,99k/1%
R31 4,99k/1%
AM1
AM2
DD1D 74HC14
R15 7,5k/1%
R16 7,5k/1%
DA1A 1458
VDD
VEE
DA2
INH
4051
C6 33n
VD6 1N4148
VD7 1N4148
VD10 1N4148
VD11 1N4148
+15V
-15V
VT2 BD139
VT3 BD140
R26 150
R10 5,1k
C5 4,7
VT1 BC547
R12 6,8k
C4 47n
R11 270k
VD5 1N4148D
R14 294k/1%
R13 294k/1%
R23 2k
C8 100n
R20 1k
R21 200
R22 249/1%
R25 75k
C9 1,2n
DA1B 1458
C10 220
R24 75k
R27 2,7k
VD9 1N4148
VD8 1N4148
W2
W3
Tr1
W1

Фиг. 5.12.

Входът *RxD* и изходът *TxD* на асинхронния сериен интерфейс (SCI) се свързват към буфери, които преобразуват логически нива на микроконтролера в нива на физическата среда за комуникация.

Изводите *PD4* и *PD5* на микроконтролера са инициализирани като цифрови изходи и управляват коефициентите на усилване в аналоговата част.

Аналоговата част на схемата (фиг. 5.12) съдържа входни усилватели и преобразуватели на аналоговите входни величини.

Стъпалото с транзистора *VT1* формира логически нива, които се управляват от външен тарифен часовник. Този часовник подава променливо напрежение с ефективна стойност 220V. Наличието или отсъствието на това напрежение е информация за това, по коя тарифа се работи в момента. Резисторът *R11* ограничава базовия ток на транзистора. Захранващото напрежение в колекторната верига е +5V. Диодът *VD5* предпазва емитерния преход на транзистора по време на отрицателната полувълна на напрежението, а кондензаторът *C5* не позволява повишаване на напрежението над границата на логическата 0 по време на същата полувълна.

Резисторите R_{13}, R_{14}, R_{15} и R_{16} образуват делител на напрежение, чийто коефициент е $\frac{0{,}5R_{15}}{R_{13}+R_{14}+0{,}5R_{15}}$ и трябва да намали амплитудата на входното напрежение до стойност, по-ниска от опорното напрежение на АЦП. Тъй като напрежението има положителна и отрицателна полувълна, а аналогово-цифровият преобразувател работи само с положително напрежение, резисторите R_{15} и R_{16} отместват входното напрежение с половината от опорното напрежение V_r. По този начин входното напрежение за АЦП става еднополярно, като нулевата стойност на променливото напрежение е равна на $V_r/2$. Чрез инвертора DD_{6D}, съдържащ тригер на Шмит на входа, се формират правоъгълни импулси с честотата на входното напрежение. Те се подават към микроконтролера.

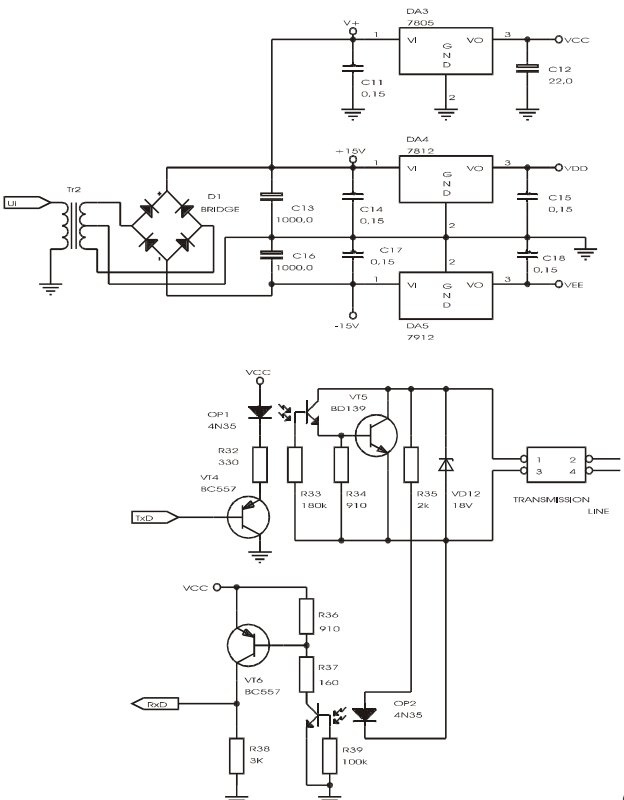
Преобразувателят на ток в напрежение е компенсационна схема с активен токов трансформатор Tr_1. Токовият трансформатор има две вторични намотки w_2 и w_3. Напрежението между изводите на намотката w_3, което се генерира при наличието на разлика между w_1I_i и w_2I_o, се подава на входа на операционния усилвател DA_{1B} и усилено към транзисторите VT_2 и VT_3 и в резултат предизвиква протичане на ток Io през намотката w_2. Генерира се магнитен поток, който компенсира потока на първичната намотка. По този начин се постига

линейност на схемата. Вторичният ток Io се преобразува в напрежение чрез шунта R_{22}. Групата R_{20}, R_{21} и C_8 се използва за настройка на дефазирането между тока и напрежението. Полученото напрежение, пропорционално на входния ток на електромера, се усилва от усилвател с програмируем коефициент на усилване, изграден с операционния усилвател DA_{1A} и аналоговия мултиплексор DA_2. В зависимост от кода, който се подава от изводите PD4 и PD5 на микроконтролера и избраните стойности на резисторите R_{28}, R_{29}, R_{30} и R_{31}, коефициентът на усилване на DA_{1A} се задава да е 1, 2, 4 или 8. Резисторите R_{17} и R_{18} изпълняват аналогична функция като резисторите R_{15} и R_{16}.

За връзка на електромера с външен компютър и включването му в локална мрежа е предвиден интерфейсен модул (фиг. 5.13). Той се състои от предавател и приемник, които са директно свързани към съответните изводи TxD и RxD на микроконтролера.

Галваничната изолация се осигурява от оптроните OP_1 и OP_2. Когато на входа на предавателя (базата на транзистора VT_4) от микроконтролера се подаде логическа нула, протича ток през светодиода на оптрона OP_1 и вследствие на отпушването на фототранзистора се насища транзисторът VT_5, който свързва на късо двата проводника на мрежовата шина (*Transmission line*). Ценеровият диод VD_{12} ограничава напрежението върху шината до безопасна за използваните

елементи стойност. При подаване на логическа единица към входа на предавателя фототранзисторът на оптрона и транзисторът VT_5 се запушват. То-


DA3
7805
V+
VI
GND
VO
VCC
C11
0,15
C12
22,0
DA4
7812
+15V
VDD
C15
0,15
Tr2
Ui
D1
BRIDGE
C13
1000,0
C14
0,15
C16
1000,0
C17
0,15
C18
0,15
DA5
7912
-15V
VEE
OP1
4N35
R32
330
VT4
BC557
TxD
VT5
BD139
R33
180k
R34
910
R35
2k
VD12
18V
TRANSMISSION
LINE
R36
910
VT6
BC557
RxD
R37
160
OP2
4N35
R38
3K
R39
100k

Фиг. 5.13.

гава токът на шината протича през входното съпротивление на приемника, състоящо се от резистора R_{35} и статичното съпротивление на светодиода на оптрона OP_2. Протичането на ток през светодиода на оптрона OP_2 води до отпушване на фототранзистора на оптрона и на транзистора VT_6. Тогава на изхода на приемника (колектора на VT_6) се формира логическа единица. Съответно, когато проводниците на шината са свързани на късо чрез електронния ключ, ток през светодиода на оптрона на приемника няма да тече, фототранзисторът и транзисторът VT_6 ще бъдат запушени и на изхода ще се формира логическа нула.

Захранващите напрежения, необходими за работата на електромера, се получават от захранващ блок, който включва мрежовия трансформатор Tr_2, мостовия изправител D_1, филтрови кондензатори и интегралните стабилизатори DA_3, DA_4 и DA_5, които осигуряват напрежения +5V, +12V и –12V.