

## БАЗОВА СТРУКТУРА НА МИКРОПРОЦЕСОРНИ СИСТЕМИ

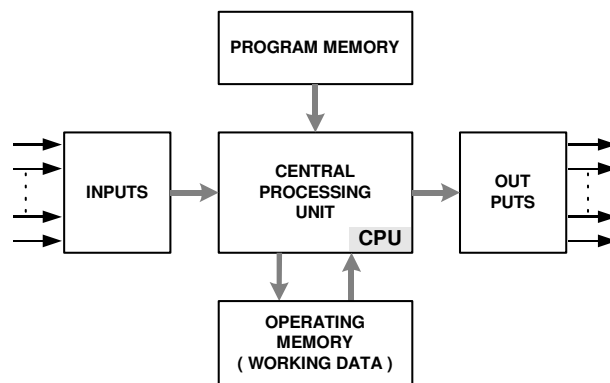
### Въведение

Концепциите, върху които се изгражда съвременната компютърна техника, се създават още през 40те/50те години на 20 в. В първата половина на 70те напредъкът във физиката и технологията на полупроводниковите прибори позволяват да бъдат разработени и масово разпространени първите *микропроцесори* - мощни универсални интегрални схеми с висока *степен на интеграция (LSI; за времето - >200 активни елемента върху монолитен кристал - тази долна граница непрекъснато се увеличава!; сега VLSI)*. Основната цел при създаването им е да се замести *твърдо опроводената логика* с програмируеми схеми, които, изпълнявайки в предварително зададена последователност (*програма*) определен набор от стандартни действия (*команди, инструкции*), да могат да бъдат прилагани за решаване на широк кръг задачи. При това, след първоначалното пускане в действие на апаратната част, за изменение на функциите на системата е необходимо да се изменя само програмата, а не опроводяването. Предимствата на такъв подход са огромни:

- електрическото проектиране и настройката на системата се изпълняват еднократно;
- адаптирането към конкретна задача или област на приложение изисква добавянето само на малка по обем и с прости функции периферия;
- разширяването на системата може да става чрез “механично” добавяне на стандартни модули, програмно управлявани от един и същи процесорен модул;
- опростяват се първоначалното пускане в действие и сервизът;
- постига се икономия на средства поради използването в големи (масови) количества на еднотипни елементи;
- пренастройката на системата за адаптиране към нови ситуации на работа или за въвеждане на допълнителни функции изисква изменение (заедно с тестването и настройката!) само на програмата, без никаква физическа намеса във вече работещата апаратна част;
- актуализирането на системата при въвеждането на нови технологии може да се осъществи само чрез смяната на малък брой елементи, често без изменения на съществуващата топология;
- става възможно на световно ниво стандартизацията на елементите, функциите и средствата за развитие (на програмното и апаратното осигуряване).

### Базова структура

Базовата структура на всяка компютърна система е представена на фиг. 1.



Фиг. 1.

Основен възел в тази структура е *централният процесор* (CPU) – той изпълнява всички логически и аритметични операции по управлението на системата.

Програмата за работата на централния процесор се съхранява в програмната памет, чиято основна характеристика е, че съхранява записаната в нея информация неограничено дълго, дори и след изключване на захранването на системата. Благодарение на това процесорът може:

- да изпълнява винаги една и съща програма (т.е. функциите на системата не се нарушават при изключване и включване на захранването);
- да открие при всяко включване под напрежение правилна последователност от изпълними команди, което гарантира правилната му работа;
- не се налага презареждане на програмата при отпадане или нарочно изключване на захранването.

Обменът на информация между процесора и програмната памет е еднопосочен. При нормалната експлоатация на системите не се предвижда (и най-често не се налага) изменение на програмата, така че не е необходимо да се осигурява възможност за запис в тази памет на място, в самата система. Информационният обем на програмната памет определя разнообразието и сложността на изпълняваните функции.

В оперативната памет непрекъснато се записват и се запазват за определено от изпълнявания алгоритъм време, оперативни данни, свързани с изпълняваната в даден момент операция (или единична команда). Това могат да бъдат данни от входовете, временно необходими за изпълнение на програмата; данни получени след обработката, определящи състоянията на изходите или междинни резултати (напр. междинна сума при операция умножение). Оперативната памет може да се използва за съхранение на историята на протичане на някакъв процес, в нея могат да се прехвърлят данни от други типове запомнящи устройства в системата, разглеждани като периферия (входно/изходни устройства), работата с която по различни причини е по-бавна. В зависимост от вътрешната архитектура на процесора, в оперативната памет могат да бъдат записвани и изпълнявани оттам и програми, което улеснява създаването и настройката на програмното осигуряване. Основна функция на оперативната памет е използването ѝ за съхранение на стека на процесора. Тъй като в стека процесорът записва и чете съдържанието на вътрешните си регистри при всяко разклонение на изпълнявания алгоритъм (програмата) или съхранява временно оперативни данни в процеса на изпълнението на по-сложни действия, той задължително трябва да е разположен в бързодействаща памет без ограничение на броя на циклите запис/изменение/запис.

*Забележка: Четенето от произволен тип памет е операция, която не изразходва ресурсите на паметта (това е неразрушаващ процес, не води до деградация на паметите). Записите и изтриването на данни, дори в един-*

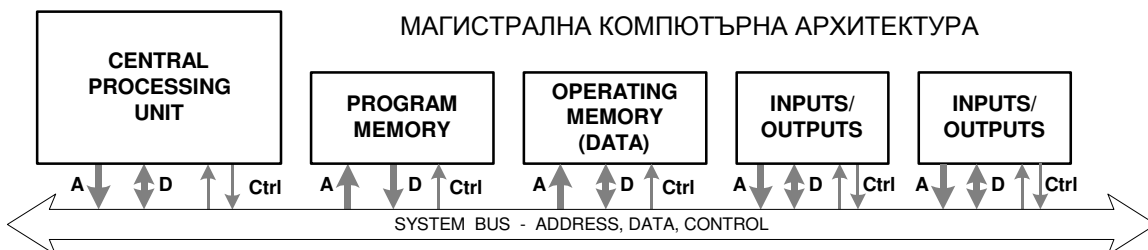
единствен бит, предизвикват в паметите от тип  $x\text{xROM}$ , необратими изменения, поради което броят  $i$ , е ограничен – обикновено до  $10^4 - 10^6$  пъти.

Този практически непрекъснат обмен между процесора и паметта поставя и едно от основните изисквания към *системната памет* – нейното бързодействие трябва да бъде съобразено с това на процесора, за да не се намалява общата производителност на системата.

За връзка на системата с реалния свят е необходимо да се предвидят и още два функционални блока – входен и изходен, чието предназначение е ясно от наименованията им. От гледна точка на електрическото им действие, те възприемат, преобразуват в подходящ вид (напрежение, полярност, товарна способност, продължителност и т.н.) постъпващата в системата информация и извеждат от нея получените след обработката резултати във вид, използваем от външните устройства.

Тази базова структура (като тип и функции на блоковете) е характерна за всички компютърни системи. В този основен вид (или с някои малки изменения – напр. управляван от процесора мултиплексор за обмен на данните между различните блокове) тя има един съществен недостатък – обменът между кои да е две (или повече) устройства се обуславя от процесора. Това от една страна забавя комуникациите вътре в системата, а от друга процесорът е непрекъснато зает с изпълнение на операции по прехвърляне на данни от едно устройство към друго, така че изчислителната му мощ не се използва пълноценно. Заедно с това, в периоди на интензивен обмен на данни, действителната работа на процесора по обработка на информацията е нарушена. Като обратен ефект може да се разглежда ситуацията, при която, ако се даде приоритет на обработката, докато тя не завърши, комуникацията с източниците на входни данни или управлението на изходите са невъзможни. Това води до конфликтни ситуации, евентуално - загуба на данни и ограничава приложимостта на подобни компютърни архитектури.

Ето защо в началото на 60те години PDP предлага за своята машина PDP8 (1964) нова архитектура на компютрите, изградена за първи път на базата на *системна магистрала (system bus)*, посредством която всички устройства в системата са свързани помежду си и с централния процесор. В тази магистрала са включени всички необходими за работата на системата сигнали: адресни, информационни (данни) и тези за управление. Тази архитектура е масово възприета в почти всички съвременни системи. Тя се



Фиг. 2.

характеризира с голяма гъвкавост, дава възможност за разширение и позволява с помощта на различни модули, свързани към магистралата, да се изграждат произволни по сложност и предназначение конфигурации – фиг. 2.

### Структура на система за развитие EVBplus2

Описаният подход е възприет и в системата за развитие EVBplus2.

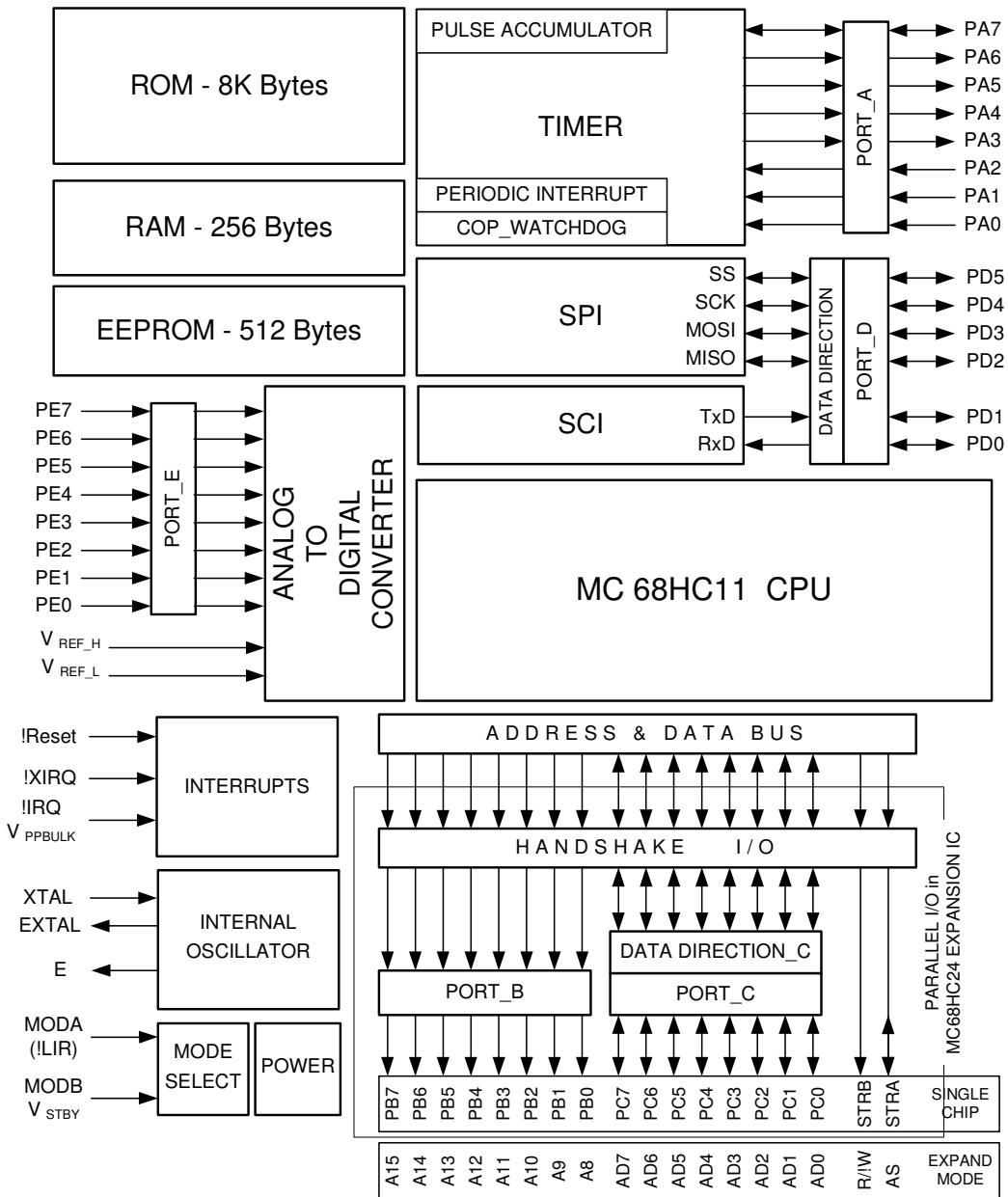
Основната част на всяка една компютърна система е централният процесор. Той може да бъде реализиран като самостоятелно устройство или да бъде част от по-голяма интегрална схема.

В разглежданата тук система е използван усъвършенстван 8-битов CMOS микроконтролер MC68HC11, работещ в разширен мультплексен режим, който освен основното процесорно ядро (CPU – *Central Processing Unit*) с мощна система команди - типична за този клас процесори от фамилията MC68xxxx, включва и богат набор от вградени периферни блокове, позволяващи изграждането на напълно функционални управляващи системи само върху един чип. Блоквата схема на микроконтролера е показана на фиг. 3.

Процесорното ядро (CPU) на MC68HC11 включва базовите функционални единици за всеки процесор, които позволяват:

- извличане на команда от паметта;
- интерпретиране на командата - управляващото устройство дешифрира командата, за да стартира необходимата обработка;
- извличане на данните (операнда) за изпълнението ѝ - изпълнението на една команда може да изисква данни, намиращи се в паметта на машината или от входно устройство;
- обработка на данните в съответствие с изпълняваната команда - изпълнение на аритметична или логическа операция с тези данни;
- запис на резултата от обработката в устройството, за което е предназначено - това може да е друг функционален блок на процесора, клетка от паметта или изходно устройство.

За да е възможно изпълнението на описаните действия, централният процесор трябва да може временно да съхранява в себе си необходимата му информация - да запомни получената команда и данните за нея, докато трае обработката, да запомни мястото (адреса) на получателя на резултата, да запомни адреса от който да извлече следващата команда, за да продължи изпълнението на програмата след текущата обработка. С други думи, в него трябва да е предвидена малка по обем бързодействаща памет.

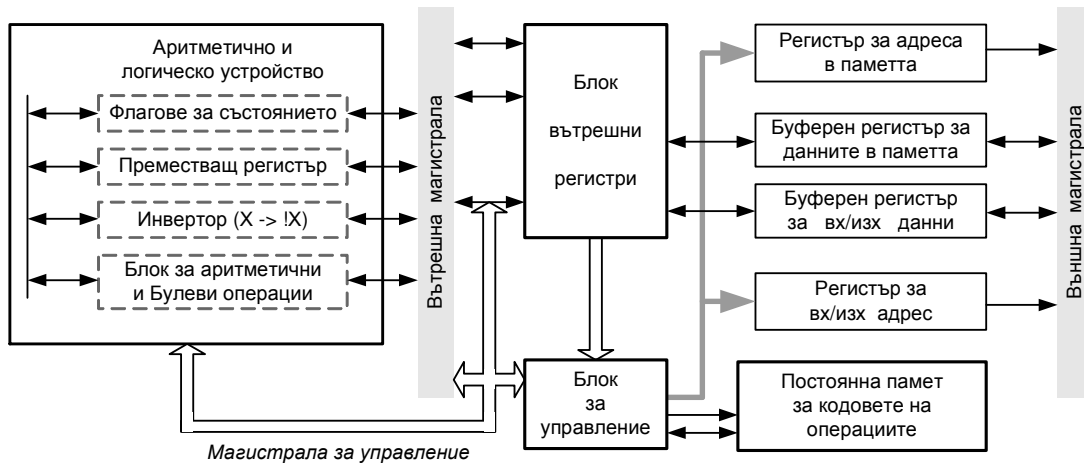


Фиг. 3. Блокова схема на микроконтролерите от фамилията MC68HC11

Освен това, в процесорът е необходимо да разполага с (фиг.4):

- изпълняващ блок - това е Аритметичното и Логическо Устройство - АЛУ (*Arithmetic Logic Unit - ALU*);
- управляващо устройство (*Control Unit - CU*) което управлява движението на данните между вътрешните функционални блокове и изпълнението на операциите;
- блок постоянна памет, в която се съхраняват кодовете на всички изпълними от процесора команди;

- вътрешна сигнална магистрала за команди, данни и управление, необходима за комуникация между отделните функционални единици.



Фиг. 4. Централен процесор - функционална схема

### Вградени периферни блокове

В монолитната интегрална схема са вградени разнообразни периферни функционални блокове, необходими за реализиране на широк кръг приложни системи:

- ◆ 8-канален 8-битов АЦП (Аналогово-цифров преобразувател / *Analog to Digital Converter - ADC*);
- ◆ Асинхронен сериен комуникационен канал **SCI** (*Serial communications interface*), позволяващ нискоскоростен обмен между компютърни системи;
- ◆ Независим синхронен сериен комуникационен канал **SPI** (*Serial Peripheral Interface*). Използва се предимно за високоскоростна връзка между микрокон-тролера и вградени в същата система периферни схеми;
- ◆ Многофункционален таймерен блок, изграден на базата на основен 16-битов цик-личен брояч с непрекъснато действие и с вътрешен (за микроконтролера) тактов сигнал с възможност за предварително делене (до  $\div 128$ ). Посредством сложна апаратна логика с програмируеми параметри са реализирани три входни канала за измерване на време (изразено в периоди на вътрешния тактов генератор) и пет изходни канала за формиране на импулсни поредици. Специален Импулсен акумулатор (*Pulse Accumulator - PA*) позволява регистрирането на външни събития (в броячен режим) или измерването на продължителността на подавани отвън импулсни сигнали - период, продължителност на ниско или високо ниво (в таймерен режим). В същия таймерен блок е вградена функция **COP** (*Computer Operating Properly Watchdog*) за мониторинг на работата на процесорното ядро при изпълнение на програмите. При навлизане в безкраен цикъл (неправилно изпълнение на програмата и загуба на контрол над процесора поради програмна грешка) се генерира вътрешен сигнал за начално установяване !Reset. Важни функции на таймерния блок са генерирането на тактовите сигнали за

сериен обмен SCI/SPI, формирането на ШИМ (широчинно-модулирани, PWM - *Pulse Width Modulated*) импулсни поредици и вътрешни времеви интервали за периодичното изпълнение на програмни модули (RTI - *Real Time Interrupts*).

- ◆ Блок за обработка и обслужване на заявки за прекъсвания от 18 различни източника - вградени в микроконтролера (свързани с всички негови периферни функции) или външни. MC68HC11 възприема както немаскируеми (от 3 изт.), така и маскируеми прекъсвания, на които може да се задава различно ниво на приоритет.
- ◆ Блок за формиране/следене честотата на основния ситемен тактов сигнал. Вграден тактов генератор задава тактовата честота за работа на процесорното ядро и периферните схеми. Възможно е използването и на външен тактов сигнал. При загуба (липса) на такта или прекалено ниска честота се генерира сигнал !Reset.
- ◆ Блок за следене на стойността на захранващото напрежение при включване и изключване. Под определена прагова стойност, при което не може да се гарантира правилната работа на MC68HC11 се генерира сигнал !Reset.
- ◆ Програмно управление на консумираната от MC68HC11 енергия чрез установяване на специални режими на работа - Stop и Wait (очакване на външно събитие), особено удобни при използване на микроконтролера в бордови системи или при батерийно захранване.

В MC68HC11 са вградени и четири различни типа памет - оперативна памет за данни RAM (до 1K байта) и постоянни паметите от различен тип за съхранение на неизменна или рядко изменяема информация: неизтриваема памет ROM (до 24K байта) и програмируеми на място паметите EPROM (до 24K байта) и EEPROM (до 8,5K байта) в зависимост от модификацията. В следващата таблица са приведени основни характеристики на паметта, вградена в някои от модификациите. За дефиниране на различните режими на работа на процесорното ядро, разположението на вградените паметите в адресното пространство, разрешаването на достъпа и на определени операции с тях, и за управление на периферните блокове служи специален, програмно достъпен на адреси от \$1000 до \$103F регистърен блок от 8-битови регистри.

Вградените паметите имат различно предназначение и различен начин на запис на данните в тях. EPROM и EEPROM паметите имат ограничен брой цикли изтриване/презапис. В ROM паметта се записва еднократно в процеса на производството мониторинг (или друга - потребителска) програма, която е невъзможно да бъде изменяна в процеса на експлоатация.

Модификация	RAM	ROM	EPROM	EEPROM	
MC68HC11A8	256	Мон.	-	512	<b>Базова модификация</b>
MC68HC11A1	256	-	-	512	A8 със забр. достъп до ROM
MC68HC11A0	256	-	-	-	A8 със забр. достъп до ROM и EEPROM

MC68HC11E9	512	12K	-	512	Увеличен обем на RAM и ROM
<b>MC68HC11E1</b>	<b>512</b>	-	-	<b>512</b>	<b>E9 със забранен достъп до ROM</b>
MC68HC11E0	512	-	-	-	E9 със забр. достъп до ROM и EEPROM
MC68HC11E8	512	12K	-	-	E9 със забр. достъп до EEPROM
MC68HC11D3	192	4K	-	-	Икон. модификация; 40 изв. корпус
MC68HC711E9	192	-	4K	-	ОТР версия на D3
MC68HC11F1	1K	-	-	512	Немулт. адреси и данни; 68 изв. корпус
MC68HC11K4	768	24K	-	640	>1MAddr; PWM; CS; 84 изв. корпус

Освен микроконтролера U12, в системата за развитие са включени и системна програмна памет с обем 8K байта - U2, потребителска програмна памет 20K байта - U3, оперативна памет U4 - 24K байта, контролери за управление на клавиатура и на индикацията и за връзка с други системи посредством паралелен и сериен интерфейс. Важен възел е системният адресен дешифратор U7, изработващ на базата на адресните и управляващите сигнали генерирани от процесора, сигнали за разрешение на достъпа до всички устройства, с които процесорът трябва да комуникира. Той е реализиран в програмируемо логическо устройство (PLD - *Programmable Logic Device*) имащо характеристиките на постоянна памет, в което са записани логическите уравнения за генериране на сигналите за разрешение на достъпа до устройствата (CS - *Chip Select*).

Предназначението на дешифратора е да формира сигналите за разрешение на достъпа в зависимост от присъстващата в даден момент на адресната магистрала комбинация и състоянието на някои контролни (управляващи) сигнали. Тази адресна комбинация се генерира от процесора според конкретните условия на работата му – типа и характеристиките на текущата команда, състоянието на външните сигнали за управление и т.н. Дешифраторът разпределя логически *адресното пространство*, достъпно за даден процесор, в съответствие с апаратните и програмните изисквания към конкретната система – архитектурата на процесора, вида и обема на паметта, входно/изходните устройства и необходимостта от бъдещо разширение на системата. Това става основно на базата на адресни сигнали. Понякога се използват и някои допълнителни системни сигнали, различни по брой и със специфични функции за всяка микропроцесорна фамилия.

Заб.: При необходимост, към адресния дешифратор могат да бъдат подавани и допълнителни сигнали, блокиращи изходите му (напр. чрез установяване в трето състояние) с цел да се избегнат конфликтите при достъп на външна система до управляваното от него адресно пространство – специфичен подход при т.нар. “директен достъп до паметта” (DMA).

#### Разпределение на адресното пространство в EVBplus2

Под “адресно пространство” на даден процесор се разбира съвкупността от клетки, на всяка от които е съпоставено число – наречено “адрес”, до които



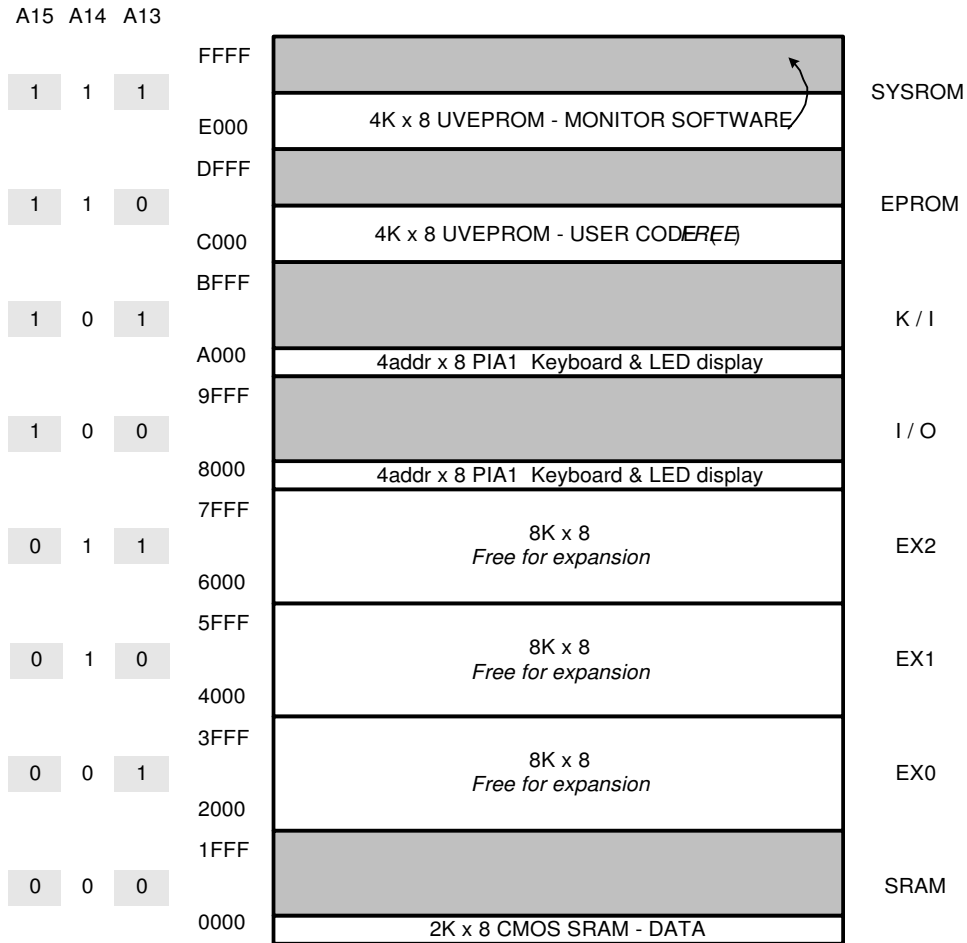
процесорът има еднозначен, директен достъп посредством адресните си сигнали, които за него са винаги изходи. В тези клетки, според характера на устройството, в което физически са разположени, процесорът може да записва информация и/или да я чете от тях. В този аспект някои клетки могат да допускат и само четене (напр. тези, принадлежащи на постоянна памет), а от други след записа, по различни съображения, четенето от страна процесора да е невъзможно (информацията се използва само вътрешно в съответното устройство – напр. Latch от типа на 573, 574). Независимо от това, за да се осъществи какъвто и да е обмен между процесора (или друг елемент, който го замества в даден момент в системата) и определено устройство, разположено в адресното му пространство, адресният дешифратор трябва да формира *сигнал за разрешение (Chip enable – CE)* на достъпа до устройството (или също така *сигнал за избор – Chip select – CS*).

Структурата и апаратната реализация на адресния дешифратор могат да бъдат най-различни, в зависимост от възприетата технология, от изискванията за бързодействие, от съображения за икономия на енергия, заемана площ върху печатната платка и др. Той може дори да бъде вграден (интегриран) в някои процесори или микроконтролери и действието му да се дефинира по програмен път по време на инициализацията на системата.

Говорейки за разпределение на адресното пространство, се има предвид по какъв точно начин определена група адресни клетки се предоставя на даден функционален блок от системата – напр. памет или входно/изходно ( I/O – *Input/Output* ) устройство.

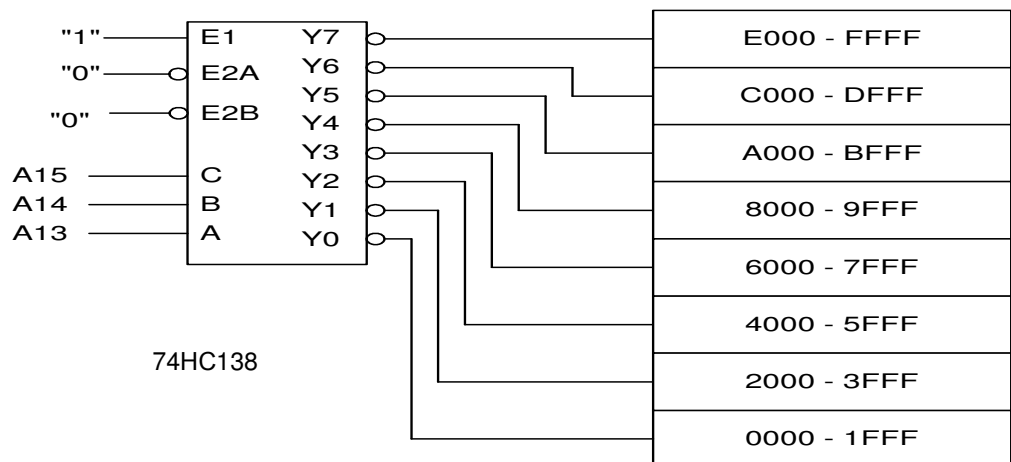
Пример за използването на адресните сигнали за разпределяне на адресното пространство в един компютър е даден на следващата фиг.6.

Ясно се виждат обособените 8 адресни области, получени използвайки в адресния дешифратор старшите адресни сигнали A15-A13 (вж. също електрическата схема).



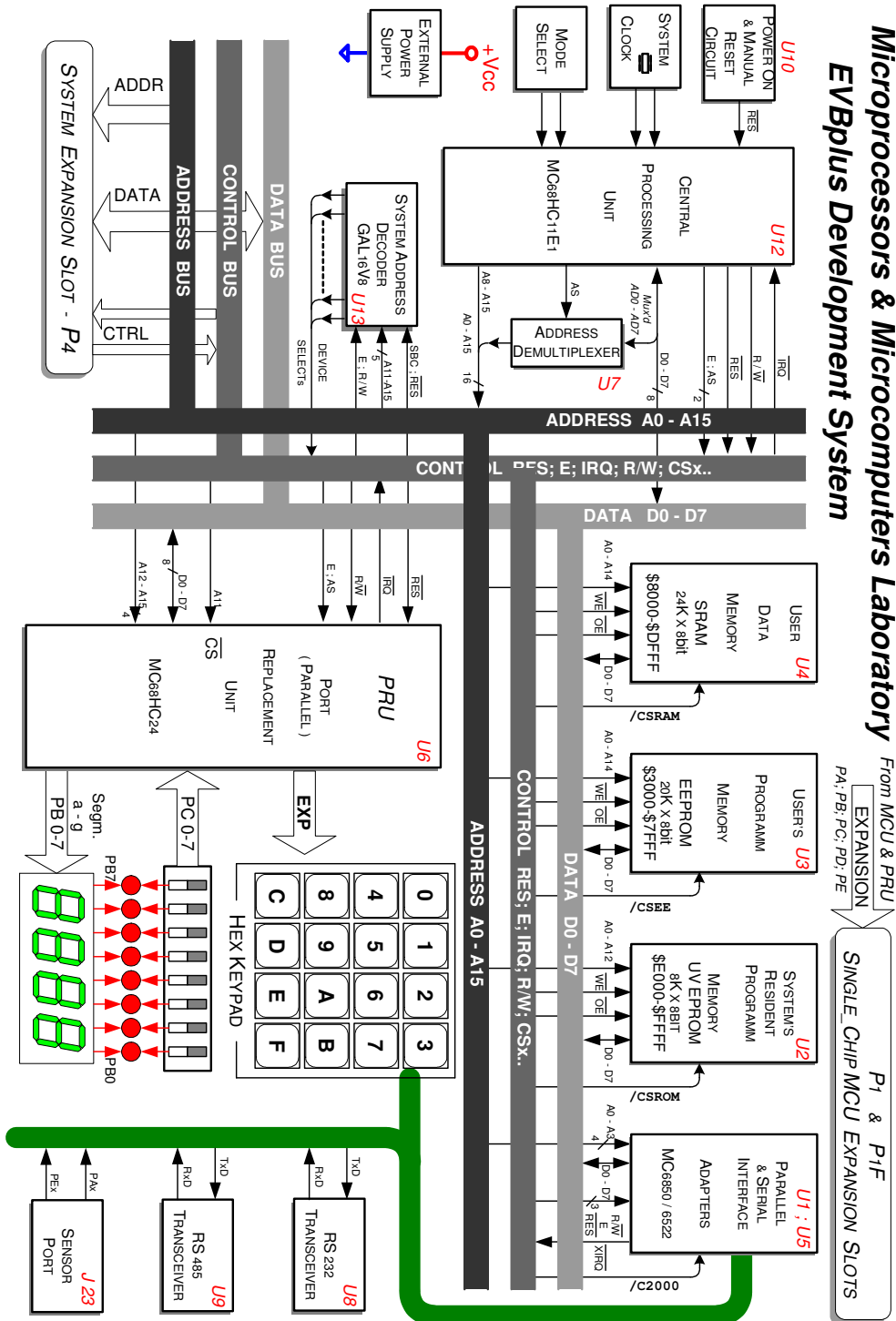
Фиг. 6. Пример за разпределяне на адресното пространство на 8 области с еднакъв обем

Апаратната реализация на адресния дешифратор е показана на фиг. 7.



Фиг. 7. Дешифратор на адресното пространство за фиг.6

# Microprocessors & Microcomputers Laboratory EV8plus Development System



Фиг. 5