

# **КОМПЮТЪРНИ АРХИТЕКТУРИ**

**Людмила Р. Танева**



## СЪДЪРЖАНИЕ

|   |    |
|---|----|
| Глава I Класификация и основни характеристики на различни архитектурни модели микропроцесори..... | 5  |
| I.1. Основни характеристики на микропроцесорите .....   | 6  |
| I.2. Класификация на съвременни микропроцесори.....   | 8  |
| I. 2. 1. По тип архитектура.....  | 8  |
| а) МР с разширена система инструкции (CISC).....  | 8  |
| б) МР с редуцирана система инструкции (RISC).....   | 9  |
| в) МР с много голяма дължина на думата (VLIW).....  | 9  |
| г) МР с паралелно изпълнение на инструкциите (EPIC) .....   | 10 |
| I. 2. 2. По разредност.....   | 10 |
| I. 2. 3. По области на приложение.....  | 10 |
| I. 2. 4. По изпълнявани функции.....  | 17 |
| Глава II Процесори с общо предназначение (GPP).....   | 19 |
| II. 1. Структура на Pentium МР (Р6 архитектура).....  | 19 |
| II. 1. 1. Подсистема памет.....   | 20 |
| II. 1. 2. Блок за извличане/декодиране на инструкциите (FDU).....                                 | 21 |
| II. 1. 3. Област за временно съхранение на инструкциите .....                                     | 21 |
| II. 1. 4. Блок за диспечеризиране и изпълнение (DEU).....   | 23 |
| II. 1. 5. Блок за оттегляне от изпълнение (RU).....   | 23 |
| II. 2. Вътрешна архитектура (програмен модел) на Р6.....  | 23 |
| II. 3. Нишки (threads).....   | 26 |
| II. 3. 1. Супер – нишки.....  | 27 |
| II. 3. 2. Хипернишкова технология.....  | 27 |
| II. 3. 3. Многонишкова обработка.....   | 28 |
| II. 3. 4. Видове многонишкова обработка.....  | 29 |
| ГЛАВА III Микроконтролери (MCU).....  | 32 |
| III. 1. Обща структура на микроконтролер.....   | 32 |
| III. 2. Основни функционални блокове на MCU .....   | 33 |
| III. 2. 1. Процесорно ядро (CPU).....   | 33 |
| III. 2. 2. Вградена памет.....  | 33 |
| III. 2. 3. Периферни модули на микроконтролерите.....   | 34 |
| III. 3. Интегрирани развойни програмни среди (IDE).....   | 55 |
| III. 4. Съвременни микроконтролери.....   | 58 |
| III. 4. 1. Микроконтролери Intel 8051 .....   | 58 |
| III. 4. 2. Микроконтролери AVR .....  | 60 |

|   |     |
|---|-----|
| III. 4. 3. Микроконтролери на фирма Motorola/Freescale.....                 | 61  |
| III. 4. 4. Микроконтролери на Microchip.....                                | 64  |
| III. 4. 5. Микроконтролери MSP430x на Texas Instruments.....                | 65  |
| III. 4. 6. Микроконтролери ARM.....   | 67  |
| ГЛАВА IV Цифрови сигнални процесори (DSP).....                              | 70  |
| IV. 1. Предназначение и области на приложение на сигналните процесори.....  | 70  |
| IV. 2. Различия между DSP и GPP.....  | 74  |
| IV. 3. Кръгово буфериране.....  | 77  |
| IV. 4. Архитектура на цифров сигнален процесор (DSP).....                   | 79  |
| IV. 5. DSP за обработка на целочислени данни и данни с плаваща запетая..... | 82  |
| IV. 6. Конвейерна обработка на инструкциите.....                            | 88  |
| IV. 7. Организация на вътрешната памет.....                                 | 89  |
| ГЛАВА V Тестване в производството на електронните модули .....              | 92  |
| V. 1. Видове тестове.....   | 92  |
| V. 1. 1. Структурни тестове.....  | 93  |
| а) Тест на електрическите вериги (ICT).....                                 | 93  |
| б) Анализатор на дефекти в производството (MDA).....                        | 94  |
| в) Тестване чрез гранично сканираща логика (JTAG).....                      | 94  |
| г) Тест-системи с подвижна сонда (Flying Probe).....                        | 94  |
| д) Автоматичен оптичен контрол (AOI).....                                   | 94  |
| е) Автоматичен контрол с рентгенови лъчи (AXI).....                         | 95  |
| ж) Комбинационен тест.....  | 95  |
| V. 1. 2. Функционални тестове.....  | 95  |
| а) Тестове за надеждност.....   | 96  |
| б) Специални тестове.....   | 97  |
| в) Тестове, специфични за продукта (PST).....                               | 99  |
| г) Вътрешен тест (BIST).....  | 99  |
| V. 2. Тест стратегии .....  | 99  |
| Абревиатура .....   | 103 |
| Литература .....  | 107 |

## ГЛАВА I

### Класификация и основни характеристики на различни архитектурни модели микропроцесори

Архитектурата на съвременните компютърни системи е съвкупност от характеристики и параметри, определящи тяхната функционална, логическа и структурна организация. Това понятие включва принципите на построяването и функционирането, заедно с основните програмни средства за работата им. В сложната йерархия на апаратните и програмни средства, които изграждат системата, всяко ниво допуска многовариантно изпълнение и комплектоване.

Съвременните компютърни архитектури се отличават една от друга в зависимост от задачите, за които са проектирани. Микропроцесорите (MP - Microprocessor) и микроконтролерите (MCU – MicroController Unit), с които са изградени, имат близка структура на централния процесор (CPU – Central Processor Unit) и основното им предназначение е да изпълняват последователности от инструкции. Универсалният MP, за да функционира, се нуждае от външна памет и периферия (клавиатура, дисплей, таймери, контролери и др.). Той е с универсално предназначение и изпълнява огромно количество MIPS (Million Instruction Per Second – милиони инструкции в секунда) и MFLOPS (Million FLOating point instruction Per Second - милиони инструкции с плаваща запетая в секунда) (пр. MP Pentium). MCU съдържа CPU и вградени периферни устройства (ПУ): памети, серийни и паралелни интерфейси, таймери и др. модули. Предназначен е за специфични единични задачи и може да работи самостоятелно или с допълнителна периферия. От микроконтролерите не се изисква особено голяма производителност, тъй като процесите, които управляват, не са толкова бързи.

Вградените микроконтролери (Embedded MCU) са за изпълнение на специфични сложни задачи. По-бързи са от MCU и са по-ефикасни при обработка на сигнали. Повечето от тях имат в структурата си хардуерен умножител, контролер за директен достъп до паметта (DMA – Direct Memory Access) и др. модули, позволяващи обработка на сигнали в реално време. Работната им честота е от 70 MHz до 1 GHz и се използват в системи с доминиращи изчисления – в управляващи устройства, мобилни комуникации и др.

Друга група съвременни архитектури са архитектурите на мобилните интернет устройства (Mobile Internet Device - MID) на базата на сигнални процесори (Digital Signal Processor - DSP). DSP архитектурата се различава от другите микропроцесорни архитектури по уникалния тип данни, които обработва - сигнали. В повечето случаи, те са реални сензорни данни: сеизмични вибрации, визуални образи, звукови вълни и др.. Имат огромно бързодействие, което се измерва в милиарди обработени инструкции за една секунда (BIPS –

Billion Instruction per Second). Цифровата обработка, която извършват DSP включва задачи, като обработка на изображение, звукови сигнал, разпознаване и генериране на говор, компресиране на данни и др..

## I. 1. Основни характеристики на микропроцесорите:

- *Разредност* - определя се от броя на паралелно обработваните битове и възможността за паралелно приемане/предаване на данни.

- *Производителност на ядрото* – определя се от набора регистри за съхраняване на временните резултати, системата инструкции, начините на адресация на паметта, организация на процесите за избор и изпълнение на инструкциите. Измерва се в милиони/милиарди обработени инструкции за една секунда (MIPS/BIPS) и е свързана с:

- Тактова честота:  $f_t = \frac{1}{T_c}$  [MHz];
- Продължителност на един такт (цикъл):  $T_c$  [ms,  $\mu$ s];
- Среден брой тактове за изпълнение на една инструкция: CPI (Clock Cycles Per Instruction);
- Брой инструкции за изпълнение на програмата:  $N_{i\text{ prog}}$ ;
- Брой тактове за изпълнение на програмата:  $N_{c\text{ prog}}$ ;
- Време за изпълнение на програмата:  $T_{\text{prog}}$  [s];
- Време за извикване/връщане от подпрограми, обслужващи прекъсвания.

Общият брой тактове  $N_{c\text{ prog}}$  за изпълнение на цялата програма е:

$$N_{c\text{ prog}} = N_{i\text{ prog}} \times CPI \quad (1),$$

производителността:

$$MIPS = \frac{f_t}{CPI \times 1000000} \quad (2),$$

а времето за изпълнение на програмата  $T_{\text{prog}}$  [s] зависи от общия брой инструкции и скоростта на обработка:

$$T_{\text{prog}} = \frac{N_{i\text{ prog}} \times CPI}{f_t} = \frac{N_{i\text{ prog}}}{MIPS \times 1000000} \quad (3).$$

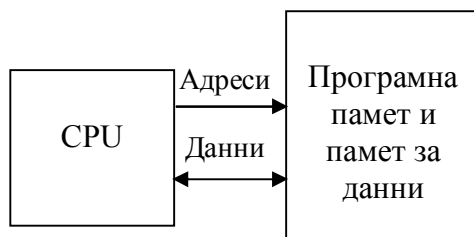
В зависимост от набора инструкции, които може да изпълнява даден MP/MCU, се разграничават две основни архитектури: CISC (Complex Instruction Set Code) – микропроцесори със сложна система инструкции и RISC (Reduced Instruction Set Code) – микропроцесори с редуцирана система инструкции. Основната идея на RISC архитектурата е подбор на такива комбинации от инструкции, които се

изпълняват паралелно за един машинен цикъл. Предимствата на този подход е рязко опростяване на апаратната реализация на CPU и значително повишаване на производителността му, намаляване броя на транзисторите и площта на кристала, което води до намаляване на себестойността и използваната мощност. Като цяло RISC архитектурите са по-ефективни от CISC и обезпечават най-малко четитикратно увеличение на бързодействието на CPU. Следствие на тяхното развитие, са суперскаларните архитектури (Pentium) [5, 9].

*Таблица 1. Сравнение на CISC и RISC микропроцесори*

|                               | CISC MP                     | RISC MP                        |
|-------------------------------|-----------------------------|--------------------------------|
| Адресни режими                | голям брой (до14)           | минимален брой (2)             |
| Регистри в CPU                | малък брой (8 рег.)         | регистров файл (16 – 128 рег.) |
| Дължина на операндите         | 1B – 16B                    | B, W                           |
| Система инструкции            | > 100                       | до 30                          |
| Време за изпълнение на инстр. | > 1 маш. цикъл              | 1 маш. цикъл                   |
| Управляваща логика            | с микропрограмно управление | твърдо зададена (hard-wired)   |
| Брой транзистори              | 3 милиона (Pentium)         | 41 000 (RISC II)               |

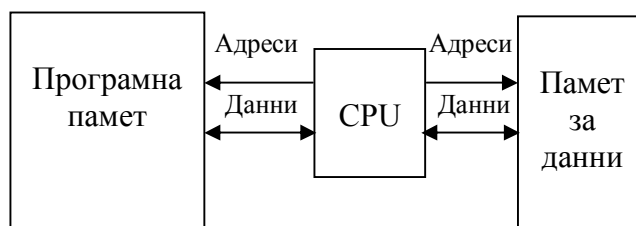
По отношение на организацията на изчислителния процес и достъпа до данните/програмата, архитектурите на монопроцесорите (SISD – Single Instruction Single Data) се разделят на класическа фон-Нойманова (Von Neumann - фиг. 1) и Харвардска (Harvard - фиг. 2) архитектури. При фон-Ноймановата архитектура съхраняването на програмата и данните е в едно общо адресно пространство. Тя е характерна за микропроцесорите с общо предназначение (Intel 80x86), при които повечето от инструкциите се изпълняват последователно.



*Фиг. 1. Фон-Нойманова архитектура*

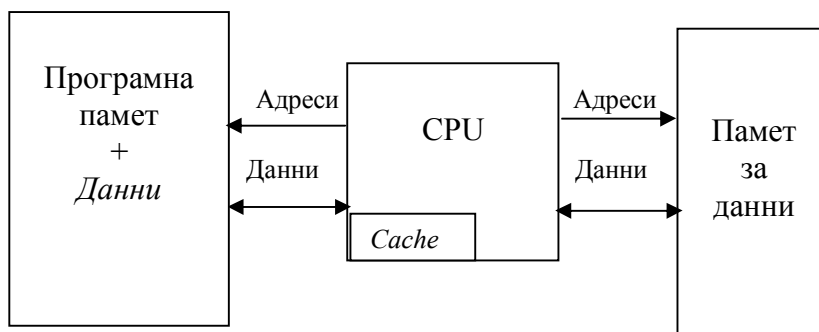
Харвардската архитектура има отделни адресни пространства за данни и програма и достъпът до тях е по отделни магистрали (Intel 8051, AVR). Поради паралелната работа на двете магистрали,

структурата има по-голяма производителност, отколкото класическата фон-Нойманова архитектура.



Фиг. 2. Харвардска архитектура

Съществува модифицирана Харвардска архитектура (SHARC - Super Harvard ARChitecture), при която инструкциите и коефициентите се записват в програмната памет, а данните – в паметта за данни (фиг. 3). Инструкциите и коефициентите се извличат в един машинен цикъл чрез двукратен достъп до програмната памет в един цикъл или чрез запис на инструкцията в допълнителен регистър (Cache). Пример за процесор с такава архитектура е DSP TMS320C54x (TI), който има една шина за инструкции, три шини за данни и четири адресни шини, специализиран набор от инструкции за реализация на по-бързи алгоритми и оптимизирана работа с езици от високо ниво.



Фиг. 3. Модифицирана Харвардска архитектура (SHARC)

## I. 2. Класификация на съвременни микропроцесори

I. 2. 1. По тип архитектура: RISC (Reduced Instruction Set Code), CISC (Complex Instruction Set Code), VLIW (Very Long Instruction Word), EPIC (Explicitly Parallel Instruction Computing).

### а) Модел МР с разширена система инструкции (CISC)

- голяма като функционалност и количество система инструкции;
- голям брой адресни режими;
- дължина на операндите – от 8 до 128 бита;



- време за изпълнение на инструкциите – повече от един машинен цикъл;
- малък брой регистри с общо предназначение (до 8);
- управляваща логика с микропрограмно управление.

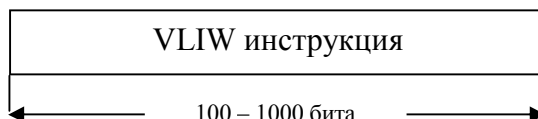
б) *Модел МР с редуцирана система инструкции (RISC)*

- минимално множество несложни инструкции;
- време за изпълнение на една инструкция – един машинен цикъл;
- дължина на операндите – до 16 бита;
- голям регистров файл – до 128 регистъра;
- минимален брой адресни режими;
- твърдо зададена управляваща логика (hard-wired).

*Post-RISC архитектурата* е характерна за съвременните МР и се базира на основните концепции на RISC-архитектурите. Тези МР имат по-висока степен на паралелизъм на ниво инструкции, дълбоки конвейри (superpipeline), усъвършенстване при предсказването на преходите, непоредно изпълнение на инструкциите, преименуване на регистрите, разширена система инструкции и др.

в) *Модел МР с много голяма дължина на думата (VLIW)*

VLIW архитектурата е подобна на суперскаларната архитектура: притежава няколко изпълнителни устройства, чрез които CPU изпълнява множество инструкции едновременно [2]. Дължината на думата е от 100 до 1000 бита (фиг. 4)

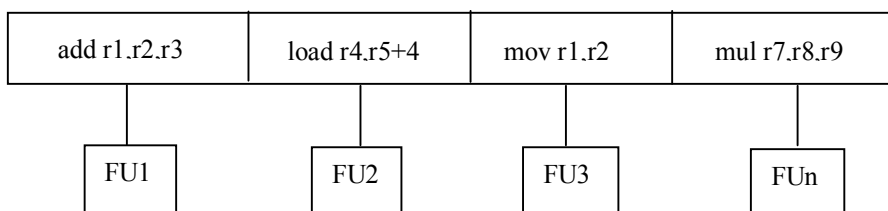


Фиг. 4. Формат на VLIW инструкция

Компиляторът определя кои операции могат да се изпълняват паралелно от функционалните устройства FUn (фиг. 5).

Особености на VLIW архитектурата:

- статично ниво на паралелизъм при изпълнение на инструкциите (под управлението на компилатора);
- инструкциите са съставени от различни машинни операции, чието изпълнение се стартира паралелно в няколко функционални устройства;
- множество паралелно работещи блокове;
- голям регистров файл



Фиг. 5. Изпълнение на VLIW инструкция

г) Модел МР с паралелно изпълнение на инструкциите (EPIC)

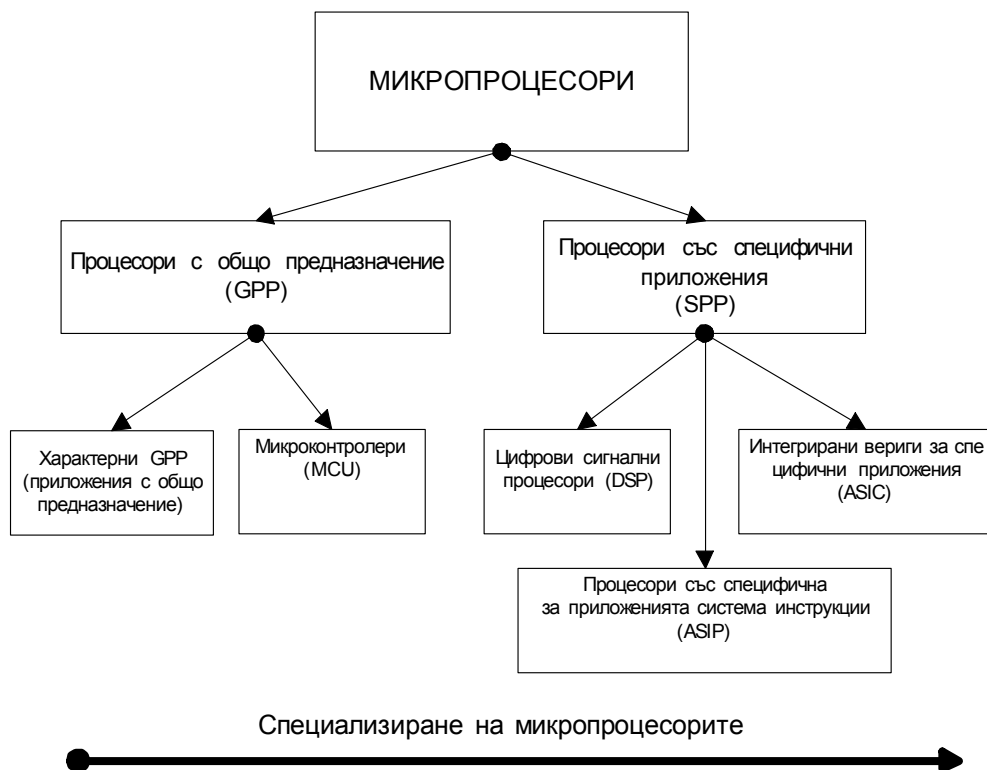
- явно изразен паралелелизъм при изпълнение на инструкциите: в процеса на компилиране, инструкциите се групират в треади, които се изпълняват паралелно (IA-64);
- оптимизиране на системата за предсказване на преходите и премахване на преходите (Predication);
- увеличен брой регистри – 128 регистъра за целочислени данни и 128 регистъра за данни с плаваща запетая;
- 64 – битово адресиране;

I. 2. 2. По разредност: 8-, 16-, 32-, 64 -, 128 – битови CPU.

Основният признак, по който се класифицират CPU, е разредността на данните, обработвани от аритметико-логичното устройство (Arithmetic Logic Unit - ALU) и за процесорите с общо предназначение те са 8/16/32/64 - битови. Отделна група са ядрата на специализираните процесори, ориентирани към системите за обработка на сигнали и графичните процесори, предназначени за видеоприложения, които притежават 128-битови CPU .

I. 2. 3. По области на приложение: МР се разделят на две основни групи (фиг. 6):

- с общо предназначение GPP (General Purpose Processors) - използват се в приложения, изпълнявани от съвременните компютърни системи, като управление на бази данни, CAD, 3D приложения и др..
- със специално предназначение SPP (Special Purpose Processors) - ASIC (Application Specific Integrated Circuit), ASIP (Application Specific Instruction Set Processor), DSP – имат особена архитектура с набор от модули, система инструкции и конструктивно-технологическо изпълнение, което позволява значително да се повиши ефективността при изпълнението на тесен кръг специализирани задачи в сравнение с други приложения.



Фиг. 6. Класификация на MP/MCU според приложението им.

#### а) Микропроцесори, характерни за РС

Типичната структура на микропроцесор за РС включва устройство за управление, едно или повече функционални устройства, набор от регистри и една или повече кеш-памети [2, 8]. Управляващото устройство извлича инструкциите от основната памет, декодира кода на операцията, след което диспечеризира инструкцията към функционално устройство. Функционалното устройство може отново да декодира кода, да чете изискваните операнди от регистрите или паметта, да изпълнява специфичната операция и да съхранява резултата в регистров файл или в паметта. След това процесът се повтаря с извличането на следващата инструкция. Програмируемостта играе огромна роля, като дава възможност да се определи коя е следващата за изпълнение инструкция.

При съвременните суперскаларни MP, инструкциите се изпълняват паралелно в различните фази на конвейрите (20-фази при Pentium IV). Чрез бързите устройства за достъп до паметта и многонивовата кеш-памет за инструкции и данни, се създава йерархична структура на паметта, чрез която се минимизират обръщенията към основната памет.

Таблица 2. Сравнителни характеристики на универсални процесори

| Характеристики                         | AMD<br>(Athlon)        | Pentium 4<br>(IA-32)                          | Power PC<br>G5                             |
|--|------------------------|---|--|
| Инструкции за един<br>машинен цикъл    | 9                      | 6   | 5  |
| Целочислени<br>конвейри                | 3                      | 4   | 2  |
| Конвейри за числа с<br>плаваща запетая | 3                      | 2   | 2  |
| Кеш-памет L1                           | 128KB                  | 12k $\mu$ op+8KB кеш за<br>данни              | 64kB за<br>инструкции,<br>32kB за<br>данни |
| Кеш-памет L2                           | 256KB                  | 256KB вградена                                | 512 kB                                     |
| Общо количество<br>кеш-памет           | 384KB                  | 264KB+12k $\mu$ op                            | 608 kB                                     |
| Скорост на<br>системната шина          | 200MHz<br>до<br>266MHz | 400MHz  | 400MHz                                     |
| Разширени SIMD<br>инструкции           | 3D Now!                | SSE2  | SSE2                                       |
| Тактова честота                        | 1 GHz                  | 2,8 GHz                                       | 2,7 GHz                                    |
| Производителност                       | 1000<br>MIPS           | 2000 SIMD MFLOPS,<br>3000 MIPS, 833<br>MFLOPS | 43 BIPS                                    |

Въвежда се динамично изпълнение на инструкциите, основаващо се на:

- многонивоно предсказване на преходите;
- динамичен анализ на потока данни;
- спекулативно изпълнение на инструкциите.

При микропроцесорите с архитектура EPIC, се въвежда нова техника на предсказване на преходите - Predication, която увеличава производителността на МР чрез избягване на преходите в програмите (табл. 2). Същността и се състои в паралелно изпълнение на всички възможни разклонения в програмата, преди реално да е изпълнено условието на прехода. Архитектурата на Intel IA-64 е базирана на EPIC и първият МР с такава архитектура – Itanium, използва метода “Predication”.

б) *Микроконтролери (MCU)* - използват се в широк кръг от индустриални приложения за изпълнение на специализирани единични задачи (повече или по-малко). Работната им честота обикновено от 32 kHz до 70 MHz и работят с много малко допълнителни схеми (табл. 3) [16, 34, 37].

Основните периферни устройства са типичните за всички MCU: таймер, входно/изходни (Input/Output - I/O) портове, сериен/паралелен интерфейс и аналогово-цифров преобразувател (Analog-to-Digital Converter - ADC) 8/10/12/16 канала).

*Таблица 3. Сравнителни характеристики на микроконтролери.*

| <b>Микроконтролер</b>        | <b>Intel 8051</b> | <b>MSP430F149</b>           | <b>PIC16F877</b>         |
|------------------------------|-------------------|-----------------------------|--------------------------|
| Максимална честота           | 12 MHz            | 8 MHz                       | 40 MHz                   |
| Архитектура                  | Харвардска        | RISC ядро; фон-Нойманова    | RISC ядро                |
| FLASH                        | -                 | 60 KB                       | 8 KB                     |
| ROM                          | 4kB               | 1 KB                        | -                        |
| EEPROM                       | -                 | -                           | 256B                     |
| RAM                          | 128 B             | 2 KB                        | 368B                     |
| ADC                          | -                 | 12 - разреден<br>12 канален | 10-разреден<br>8 канален |
| SPI интерфейс                | -                 | 2                           | 1                        |
| SCI интерфейс                | 1                 | 2                           | 1                        |
| I/O                          | 32                | 48                          | 33                       |
| таймери                      | 2                 | 2                           | 3                        |
| COP таймер                   | не                | да                          | да                       |
| JTAG интерфейс               | не                | да                          | не                       |
| PLL                          | не                | да                          | -                        |
| Компаратор                   | -                 | 1                           | 2                        |
| Режими с намалена консумация | 2                 | 5                           | 1                        |

Някои микроконтролери притежават 6/10 таймера, стражеви таймер (WatchDog Timer - WDT), компаратор, до четири DMA модула, до четири синхронни/асинхронни интерфейса (USART), до три CAN-интерфейса, до два I2C-интерфейса, JTAG интерфейс, вграден контролер на прекъсванията, до шест модула за широчинно-импулсно модулирани сигнали (Pulse Width Modulation - PWM), цифрово-аналогов преобразувател (Digital-to-Analog Converter – DAC), часовник за реално време (Real Time Clock - RTC), хардуерен

умножител и др.. Произвеждат се с програмна памет ROM от 512B до 128kB (може да бъде OTP: 4kB-16kB; EPROM: 2kB-32kB или FLASH: 1kB-256kB) и RAM памет (25B-16kB).

Типично приложение на MCU е управлението на външни събития, които се детектират на входните линии - цифрови CMOS I/O или аналогови входове. Източниците на тези сигнали са ключове, сензори, импулсни генератори, а изходните сигнали се подават към изпълнителни устройства, двигатели и др..

в) *Цифрови сигнални процесори (DSP)* - специализирани МР за обработка на сигнали.

Таблица 4. Сравнителни характеристики на DSP.

| Параметър               | Motorola DSP56F805 | TI 320LF2407A | ADI ADMC401 |
|-------------------------|--------------------|---------------|-------------|
| Производителност        | 40 MIPS            | 40 MIPS       | 26 MIPS     |
| Архитектура             | Харвардска         | Харвардска    | Харвардска  |
| 1 цикъл за MAC          | да                 | да            | да          |
| FLASH                   | 38K x 16           | 32K x 16      | 0           |
| ROM                     | 0                  | 256 x 16      | 2K x 24     |
| RAM                     | 2K x 16            | 2592 x 16     | 3K x 24     |
| PWM канали              | 12                 | 16            | 8           |
| ADC канали              | 8                  | 16            | 8           |
| SPI                     | 1                  | 1             | 2           |
| SCI                     | 2                  | 1             | 1           |
| I/O                     | 32                 | 41            | 12          |
| CAN интерфейс           | 1                  | 1             | 0           |
| Таймери                 | 16                 | 4             | 2           |
| Източници на прекъсване | 64                 | 38            | 19          |
| COP таймер              | Да                 | Да            | Да          |
| JTAG интерфейс          | Да                 | Да            | Не          |
| PLL                     | Да                 | Да            | Не          |

DSP са оптимизирани за много бързи математически обработки и производителността им се измерва в MIPS или в BIPS (Billion Instruction Per Second). Притежават високо ниво на паралелизъм. DSP изпълняват аритметичните операции паралелно с операциите LOAD/STORE, като извличат едновременно два операнда за един машинен цикъл. Основната операция умножение-натрупване

(multiply-accumulate – MAC) се повтаря циклично и определя скоростта на работа на DSP в реално време. Работната им честота е от 50 MHz до 750 MHz. Имат вграден хардуерен умножител, DMA контролер, таймери (до 16), Watchdog таймер, ADC, високоскоростни SPI, SCI, JTAG, CAN интерфейси, PWM канали (повече от шест), вградени RAM, ROM и FLASH памети, кръгови буфери, извършващи автобуфериране на входните/изходните потоци от данни (табл. 4).

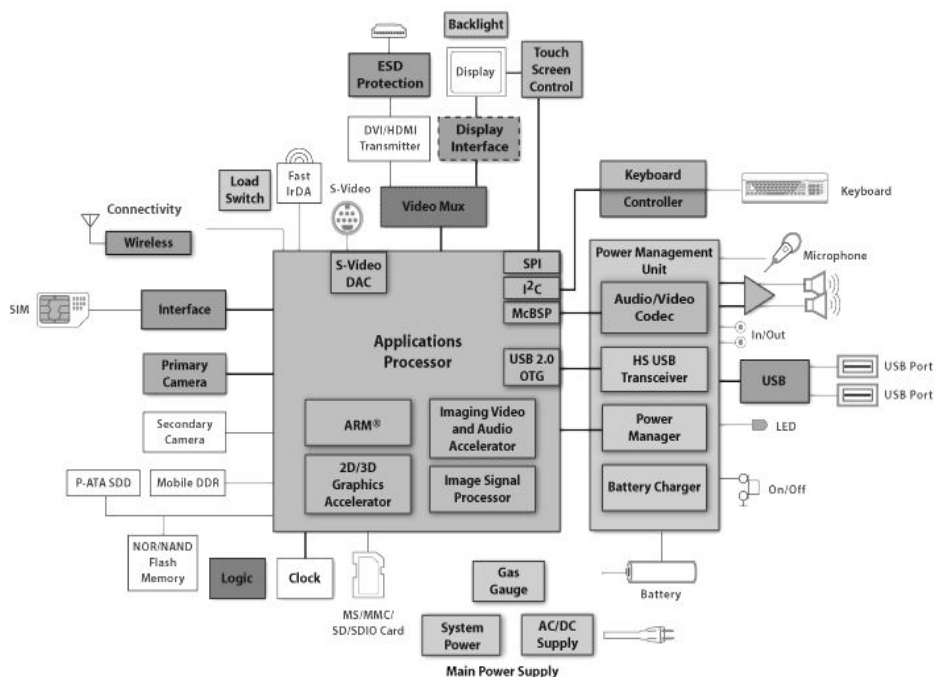
В зависимост от това дали притежават модул за операции с плаваща запетая или не, се делят на две големи групи – DSP с целочислена аритметика и DSP с плаваща запетая. Последните са много удобни, тъй като на практика решават проблема с мащабирането на данните, препълванията и др. DSP с целочислена аритметика имат поне един акумулатор с по-голяма дължина (40 бита) за операциите, в които се натрупват произведения и резултатът е много дълъг [40].

*з) Процесори със специфична за приложението система инструкции (ASIP).*

ASIP са специфични програмируеми микропроцесори, на които хардуерът и програмното осигуряване са разработени за специални приложения – клетъчни телефони, PDA (Personal Digital Assistant) и др. портативни мултимедийни устройства. Тези МР обслужват потребителски приложения, такива като управление на персонална информация, аудио/видео обработка, но не и приложения, свързани с комуникациите. Могат да работят под “отворени “ операционни системи (Windows CE, Palm OS). Повечето ASIP се базират на ARM архитектура. Разграничават се четири категории ASIP:

- *Базисни GPP.* Имат архитектура, състояща се от ARM ядро. Прямо останалите ASIP, обработката на мултимедийни приложения е по-малко ефикасна. Някои ASIP работят с високи честоти - MCU Samsung S3C2xxx (533MHz). Вградената им памет е ограничена - няколко десетки kB, но се предлагат и модели, съдържащи освен процесор, допълнителни модули памет - 32MB SDRAM, 32MB flash ROM (Samsung S3C2xxx).
- *DSP-разширени GPP.* Имат разширени възможности, ориентирани към обработката на сигнали. Новата фамилия Intel PXA27x притежава безжична MMX технология, която позволява да се извършат четири 16-битови операции тип умножение/съхраняване с една инструкция (SIMD изчисления). Работната честота на PXA27x е 624 MHz.

- *DSP* процесорите са комбинация от ARM ядро и едно или повече DSP ядро (OMAPx, TI) [50]. Имат много голямо бързодействие (фиг. 7).



Фиг. 7. Структура на мобилно интернет устройство (MID), на базата на OMAPx

#### Характеристики на OMAP 3:

- Суперскаларен MP с ARM Cortex-A8 RISC ядро
- IVA 2 + Аудио ускорител, поддържащ мултистандарт (MPEG4, WMV, Real, H.263, H.264, JPEG) кодиране/декодиране D1 (720 x 480 пиксела) 30 fps
- Вграден процесор на изображения (Integrated Image Signal Processor - ISP)

#### Характеристики на OMAP 4:

- разработен е за смартфони и мобилни интернет устройства (MIDs)
- IVA 3 хардуерни ускорители позволяващи пълнен HD 1080p, мултистандарт кодиране/декодиране
- Двухъдрен ARM Cortex-A9 симетричен микропроцесор (SMP – Simetric MicroProcessor)
- Вграден графичен ускорител POWERVR™ SGX540 за 3D игри и 3D потребителски интерфейси
- честоти от 720 MHz до 1+ GHz.



#### *Подсистеми на OMAP44х:*

- вградена система за управление на захранването, аудио кодек (пет DAC и два ADC), високоскоростен USB приемо-предавател, ас/USB зареждане, LED драйвери, ADC, RTC.

- сензорен LCD дисплей с разрешаващата способност на WVGA (854 x 480), която позволява web страницата да се показва в пълния и размер.

- Свързване с Bluetooth, WLAN и GPS

- *Програмируеми ускорители.* Те са подобни на DSP копроцесорите и предназначението им е да ускорят обработката на сигнали. Разликата е в това, че DSP копроцесорите са за широк кръг от приложения, свързани с обработка на сигнали, докато повечето програмируеми ускорители са разработени само за няколко специфични задачи. Като резултат от тази тясна специализация, те са по-производителни от DSP и имат по-малка консумация, но архитектурата им е нестандартна и сложна, което прави трудно програмирането им. Пр. NeoMagic MiMagic6 има програмируем ускорител (Associative Processing Array), състоящ се от матрица от 160 до 512 процесорни единици. Обикновено производителите обезпечават софтуерни продукти, улесняващи разработването им.

#### *д) Интегрирани вериги със специфични приложения (ASIC).*

ASIC са специфични микропроцесори, проектирани по заявка на потребителя и реализиращи строго определени функции, напр. протоколи за комуникация. Алгоритмите се осъществяват хардуерно от специално създадена за това логика с фиксирани функции. Към тях спадат хардуерните ускорители. Тези MP извършват обработката на сигнали с твърдо зададена логика. (Motorola i.MX21). Спрямо другите приложни процесори, имат по-добра енергийна ефективност, но не са гъвкави. Напр. Motorola i.MX21 има хардуерни ускорители за H.263 и MPEG-4 кодер/декодер. За декодиране на MPEG-4 имат отлична енергийна ефективност и бързодействие, но не са удобни за други видео алгоритми като H.264.

#### **I. 2.4. По изпълнявани функции:**

- Процесори на управляващи системи:
  - универсални;
  - с разширени комуникационни възможности;
  - с разширени възможности за директен вход-изход;
  - с разширени възможности за обработка на аналогови сигнали MSP (Mixed Signal Processor).

- Комуникационни процесори – апаратно поддържат интерфейси и протоколи за комуникационни системи:

- мрежови – поддържат разпространението на мрежовите и съвременните периферни интерфейси: Ethernet, HDLC, X.25, T1, ATM, USB, High Speed UART и др.. Произвеждат се от фирми Motorola (MC683xx, MPC8xx), AMD (Am186CC);

- модемни – поддържат протоколи за предаване на данни по синхронни и асинхронни модемни канали (серии V2x, V3x);

- цифрови сигнални процесори (DSP) – реализират методи за цифрова обработка на сигналите: филтрация, спектрален анализ, смесване на сигнали, мащабиране.

- Графични процесори – ориентирани са за обработка на видео и звукова информация:

- с апаратно поддържане на мултимедийната обработка и система инструкции за обработка и предаване на аудио/графични данни и видеоизображения. Използват се в персоналните компютри (PC), в приставките за игри, в битовата техника (медиапроцесор MediaGX на Cirux, видеопроцесори NV1..5 на Nvidia и др.);

- с мултимедийно разширение на системата инструкции (Intel MMX и следващите модели със SSE-, SSE2- и SSE3-инструкции, UltraSPARC на SunMicrosystem);

- Копроцесори - математически процесори и входно-изходни процесори;

- Транспютри – процесори за изграждане на паралелни системи. В кристала са вградени: локално процесорно ядро, памет и специални бързодействащи канали за връзка с други транспютри (производител Inmos);

- Други специализирани процесори:

- невропроцесори – за изграждане на невронни мрежи, които решават задачи като: разпознаване на образи, предсказване поведението на система, оптимизационни задачи.

- процесори за езици от високо ниво (Java, Forth и др.).

## ГЛАВА II

### Процесори с общо предназначение (GPP)

#### II. 1. Структура на Pentium MP (P6 архитектура)

Архитектурата P6 I80x86 съчетава най-добрите решения от предишни фамилии с нови концепции като: динамично изпълнение на инструкциите, предсказване на преходите, MMX/SSE-технологии и др. [48]. Общи характеристики на Pentium процесори:

- Комплексен набор инструкции с RISC инструкции и поддържане на по-старите X86 процесори
- Суперскаларна, суперконвейрна архитектура с два 32-битови конвейра U и V за операции с цели числа
- Вграден конвейрен модул за операции с числа с плаваща запетая
- Две отделни кеш памети - 8K за код и 8K за данни
- Динамично предсказване на преходите
- 64-битова шина данни
- Проверка по четност на данни и адреси
- Тест и възможност за отстраняване на грешки - стандарт IEEE 1149-1 за гранично сканиране
- Разширения за работа във виртуален режим
- Паралелна работа на 2 процеса
- Разширени функции за управление на захранването
- Вграден усъвършенстван програмируем контролер на прекъсванията

*Pentium 3 – характеристики:*

- 0.25  $\mu\text{m}$  технология.
- Тактова честота - 450 MHz / 500/550 до 1,1 GHz.
- 70 нови инструкции, които се използват за създаване на изображения, обработка на говор и различни мултимедийни приложения.
- Независима двойна шина.
- 512 Kb L2 обща кеш памет.
- Осем 64-битови MMX регистри.

*Pentium 4 – характеристики:*

- 0.13  $\mu\text{m}$  технология.
- 1.4 - 1.6 GHz диапазон на работните честоти.
- 144 SIMD инструкции.
- 845/850/915/945 чипсет.
- 400/533 MHz честота на системната шина.

Типичен GPP е MP Pentium 4 (Intel) със CISC архитектура (IA-32): 20-фазен конвейр, n-нивова кеш памет: към L1-кеш е добавена кеш-памет ETC (ETC - Execution Trace Cache), която съхранява 12kB

декодирани микрооперации в реда на програмното им изпълнение и 256 kB L2-кеш ATC ( ATC - Advanced Transfer Cache) памет, работеща с честотата на ядрото и обезпечаваща трансфер на данни 48 GB/s при 1.5 GHz [46].

Регистрите за операнди с плаваща запетая са 128-битови и са въведени нови типове мултимедийни данни (Single Instruction Multiple Data - Streaming SIMD Extensions 2 (SSE/SSE2/SSE3)). Те са пакетирани в 128-битови единици и се обработват с една инструкция в специализирани блокове. SIMD-обработката на MMX- и SSE-данните е допълнена с нови 144 SSE2 и 13 SSE3 инструкции, които включват 128-битова SIMD целочислена аритметика и 128-битова SIMD аритметика на числа с плаваща запетая с двойна точност.

Аритметико-логичните устройства са три: едно за сложни инструкции, което работи с честотата на ядрото и две за прости инструкции, които работят с удвоената честота на ядрото (за 2 такта изпълняват 4 инструкции).

Предсказването на преходите (Branch Prediction - BP) е основна стратегия, повишаваща производителността на модерните суперскаларни микропроцесори. За най-ефективна конвейризация на инструкциите, е необходимо всички фази на конвейрите да са запълнени с инструкции. Някои инструкции, като IF, LOOP и PROCEDURE, предизвикват промяна в нормалната последователност от инструкции и в конвейрите се появяват т.н. “балони” (“bubbles in the pipe”), които намаляват производителността на МР. Един от начините те да се намалят или изцяло да се премахнат, е механизмът BP, въведен при МР Pentium и усъвършенстван при следващите поколения микропроцесори. Чрез предсказване на преходите, се прави предположение какъв ще е адресът на инструкцията след условен преход. Ако предположението е вярно, то конвейрът ще остане пълен, а ако е грешно, то тогава конвейрът ще се преинициализира и ще има загуба на инструкции. Възможно е > 97% от преходите в програмата да са вярно предсказани от МР. При първият МР от фамилия Pentium на Intel, се въвежда еднониово BP, а при следващите МР Pentium, многониово предсказване. При новите МР, конвейрите са многофазни и се налага много по-ефективно BP. С усъвършенстване на BP, МР има възможност да разпознае закономерно повтарящи се последователности от инструкции и даже отклоненията от тях. Историята на прехода се съхранява в буфер на целевия преход ВТВ (ВТВ - Branch Target Buffer), съдържащ целевите адреси за всеки преход в програмата.

На фиг. 8 е дадена функционална блокова схема на Р6, в която са обособени пет подсистеми:

II. 1. 1. Подсистема памет: Системна шина (System Bus), L2 кеш-памет, L1 кеш-памет за инструкции, L1 кеш за данни, блок за управление на шините (BIU – Bus Interface Unit), блок за достъп до

паметта (MIU – Memory Interface Unit), буфер за преподреждане на заявките за достъп до паметта (MRB - Memory Reorder Buffer).

BIU поддържа достъпа до L1/L2 кеш-паметите, като интерфейсите между системната памет и кеш-паметите е реализиран по протокола MESI (Modified/Exclusive/Shared/Invalid). Заявките за достъп до системната памет постъпват в блок MIU и се обработват така, че се избягва блокиране на достъпа до паметта (първо се обслужва заявка от L1 кеш, ако не е възможно, заявката се пренасочва към L2 кеш, а ако и това не е възможно, заявката се насочва към системната памет). В буфера за преподреждане MRB се буферират заявките за достъп до паметта, като той променя реда на обслужването им с цел повишаване на производителността (напр. MRB може да разреши заявките за извличане на операнди да изпреварят заявките за запис в паметта).

II. 1. 2. Блок за извличане/декодиране на инструкциите (Fetch and Decode Unit - FDU): блок за извличане на инструкции (Instruction Fetch Unit - IFU), буфер на целевия преход (BTB), декодер на инструкции (Instruction Decoder - ID), блок за генериране на микроинструкции (Micro Instruction Sequencer - MIS) и таблица за разпределение на регистрите RAT (RAT - Register Allocation Table).

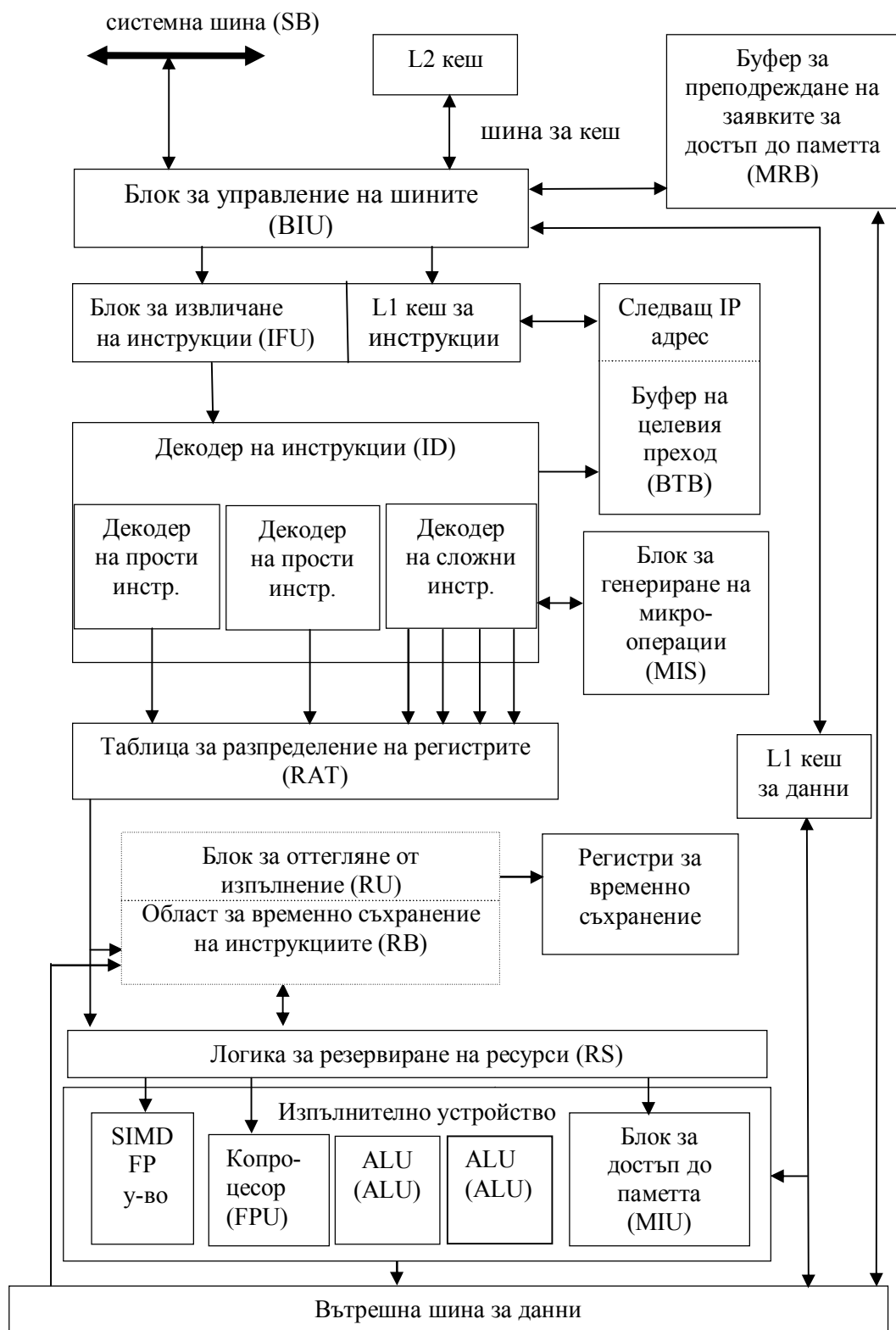
От L1 кеш за инструкциите в блока за извличане/декодиране постъпва поток инструкции, които се декодират до микроинструкции (в MIS). Те постъпват в областта за временно съхранение (instruction pool – блок RB).

Блокът за извличане на инструкции IFU пресмята стойността на IP, като изпълнява информацията от BTB. В рамките на BTB се реализира механизма на предсказване на преходите. Логиката за управление на BTB запълва 512 позиции в буфера, като проследява множество инструкции напред след текущото съдържание на IP, сочещ последната от изпълнените инструкции (завършена инструкция).

Декодерът на инструкциите ID се състои от два декодера на прости инструкции и един за сложни инструкции, като за един машинен цикъл генерират до 6 микрооперации. „Сложните” инструкции се декодират до 4 микроинструкции в блок MIS.

Блокът RAT е таблица на регистровото съответствие между логическото множество регистри и реалното множество регистри. В него има 40 регистъра с общо предназначение, използвани от микрооперациите.

II. 1. 3. Област за временно съхранение на инструкциите (Instruction Pool) или буфер за преподреждане (Reorder Buffer - RB) се състои от 40 регистъра за съхранение на микрооперации. Чрез него се осъществява непоследователното им изпълнение (out-of-order).



Фиг. 8. Вътрешна архитектура на P6 архитектура.

Всеки регистър съдържа микрооперация, която „чака“ да бъде изпълнена или е вече изпълнена, но не е маркирана като завършена.

*Завършена инструкция* е инструкция, която не е зависима от данните на друга инструкция и не е част от програмен сегмент, предвиден за изпълнение в резултат на неуспешно предсказан преход.

II. 1. 4. Блок за диспечеризиране и изпълнение (DEU – Dispatch Execute Unit). Състои се от логика за резервиране на ресурси (RS-reservation station), два блока за целочислена аритметика (Integer Unit-IU), два блока за обработка на числа с плаваща запетая (FPU) и два блока за генериране на адреси (Address Generation Unit - AGU).

Блок DEU може да обработва микрооперациите от RB в последователност, различна от оригиналния ред на инструкциите (out-of-order), като изборът е в зависимост от данните и достъпността до ресурсите. Логиката за резервиране на ресурси (Reservation Station - RS) непрекъснато следи RB и при откриване на микрооперация ( в готовност за изпълнение (всички операнди са достъпни), я насочва към незаает блок: IU, FPU или MIU.

Блок DEU поддържа механизма на спекулативно изпълнение на инструкциите: двата блока IU позволяват паралелно изпълнение на две целочислени микрооперации. Единият блок е специализиран за изпълнението на микрооперация за преход. Той разпознава грешки при генериране на преходи и генерира управляващо въздействие към ВТВ за преинициализация на конвейрите.

Когато IU изпълнява микрооперации за преход, еднозначно се определя действителния преход. Ако той е предсказаният, всички микрооперации се маркират като ”използваеми”, а ако е грешно предсказан, то тези инструкции се отстраняват от областта за временно съхранение, конвейрите се нулират и се запълват в зависимост от новото състояние на ВТВ.

Блок MIU паралелно изпълнява микрооперациите за извличане от паметта и запис в паметта в един машинен цикъл.

II. 1. 5. Блок за оттегляне от изпълнение (Retire Unite - RU) - претърсва RB за завършени инструкции и ги отстранява от него, като възстановява оригиналния ред на инструкциите. При обработка на една микрооперация, RU записва резултата в регистров файл, състоящ се от 9 регистъра с общо предназначение) и 9 регистъра за числа с плаваща запетая, с което завършва процесът на изпълнение.

## **II. 2. Вътрешна архитектура (програмен модел) на P6.**

Архитектурата на MP 80x86 съдържа 16 регистъра, които са разделни в няколко основни групи:

II. 2. 1. Регистри с общо предназначение - осем 32-битови регистъра с общо предназначение, които се използват за временно съхранение на операнди за аритметични и логически операции, адреси, указатели, междинни данни и др. (фиг. 9).

|    |    |    |    |   |   |         |         |
|----|----|----|----|---|---|---------|---------|
| 31 | 16 | 15 | 8  | 7 | 0 | 16 бита | 32 бита |
|    | AH |    | AL |   |   | AX      | EAX     |
|    | BH |    | BL |   |   | BX      | EBX     |
|    | CH |    | CL |   |   | CX      | ECX     |
|    | DH |    | DL |   |   | DX      | EDX     |
|    | BP |    |    |   |   |         | EBP     |
|    | SI |    |    |   |   |         | ESI     |
|    | DI |    |    |   |   |         | EDI     |
|    | SP |    |    |   |   |         | ESP     |

Фиг. 9. Регистри с общо предназначение на MP 80x86

Регистрите с общо предназначение на 80x86 архитектурата са EAX, EBX, ECX, EDX, EBP, ESP, ESI и EDI. 16-битовите части на тези регистри с общо предназначение са: AX, BX, CX, DX, BP, SP, SI и DI.

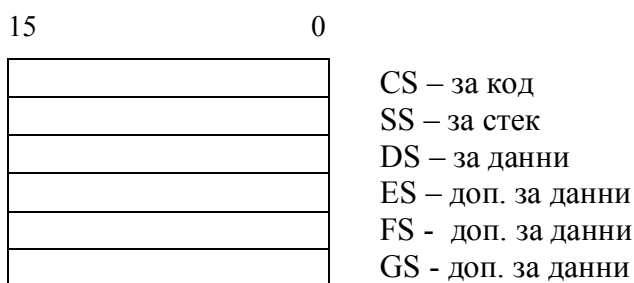
Всеки байт от 16-битовите регистри AX, BX, CX и DX може да бъде третиран самостоятелно при работа с 8-битови данни. Тези регистри са AH, BH, CH и DH (старши байтове) и AL, BL, CL и DL (младши байтове).

II. 2. 2. Сегментни (адресни) регистри – шест 16-битови регистъра, които съдържат базовите адреси на сегментите в паметта за код, за стек и за четири сегмента за данни (фиг. 10). Във всеки момент, шест сегмента от паметта могат да бъдат достъпни за изпълняващата се x86 програма. Регистрите CS, DS, SS, ES, FS и GS се използват за идентификацията на тези шест текущи сегмента. Всеки регистър уникално определя един определен сегмент от сегментите, които образуват програмата.

Сегментът, съдържащ в момента изпълняващата се последователност от инструкции, е текущия сегмент за кода на програмата и е специфициран като CS регистър. MP 80x86 извличат всички инструкции от този кодов сегмент, използвайки като отместване (offset) съдържанието на указателя на инструкции (рег. IP – Instruction Pointer ).

Подпрограмни извиквания и параметри обикновено изискват част от паметта да бъде асоциирана като стек. Всички стекови операции използват SS регистъра, за да се адресира стека. DS, ES, FS и GS регистрите позволяват спецификация на четири сегмента на данни, всеки от тях достъпен от текущата програма.

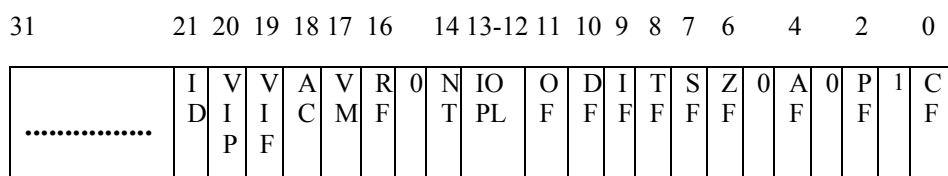




Фиг. 10. Сегментни регистри на MP 80x86

Сегментните регистри на MP 80x86 дават възможност на системните програмисти гъвкаво да избират сред множество модели на организация на паметта.

II. 2. 3. Флагов регистър *EFLAGS* – съдържа управляващи битове и битове за състояние (флагове - фиг. 11).



Фиг. 11. Флагов регистър *EFLAGS* на MP 80x86

- *флагове за състояние* – отразяват резултата от дадена инструкция:

*CF (Carry Flag)* – пренос/заем. Установява се при наличие на “пренос” след изпълнение на аритметична операция и показва препълване при целочислена аритметика без знак. Установява се и при наличие на “заем” след инструкция за изваждане.

*PF (Parity Flag)* – четност. Установява се, ако последният значещ резултат съдържа четен брой единици.

*AF (Adjust Flag)* - междинен пренос (междинен пренос между младшата и страшата тетрада). Използва се при двоично кодиране на десетични числа (BCD числа).

*ZF (Zero Flag)* – нула. Установява се при нулев резултат.

*SF (Sign Flag)* – съвпада с най-левия бит на резултата (знака). При SF=0, резултатът е положително число, а при SF=1 – отр. число.

*OF (Overflow Flag)* - препълване. Установява се, ако целочисленият резултат е твърде голямо положително число или твърде малко отрицателно число.

- *управляващи флагове* – управляват работата на MP:

*TF (Trace Flag)* – трасиране. При TF=1, MP спира след всяка инструкция. Използва се при настройка и тестване на програмата.

*IF (Interrupt Flag)* – маска на прекъсване. При установяване на IF се забранява обработката на маскируеми прекъсвания.

*DF (Direct Flag)* – посока. Определя посоката при обработка на низове. При DF=0, посоката е от младшите към старшите адреси.

*IOPL (I/O Privilege Level)* – входно- изходното ниво на привилегия. Задава нивото на привилегия за достъп до входно-изходното пространство. От това поле се определя дали да има входно-изходно прекъсване или не.

*NF (Nested Flag)* - вложена задача. Управлява вложените задачи от гледна точка на прекъсване на изпълнението и обръщение към следваща задача. Този флаг се установява, ако текущата задача е свързана по отношение на изпълнената и се нулира, ако между двете задачи няма връзка.

*RF (Resume Flag)* – възстановяване след прекъсване при настройка на приложения (debug).

*VM (Virtual Mode)*- виртуален режим. Управлява режим V'86. При VM = 1, MP работи като високопроизводителен I8086.

*AC (Alignment Check)*- адресно изравняване. При AC=1 възниква изключение “неизравнен” операнд в паметта – пр. операнд тип дума, разположен на нечетен адрес или двойна дума, разположен на некратен на четири адрес.

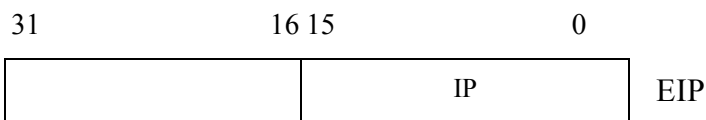
*VIF (Virtual Interrupt Flag)*- флаг за прекъсване в режим V86.

*VIP (Virtual Interrupt Pending)*- разрешаване на прекъсване в режим V86.

*ID (Identification)* – идентификатор. При ID=1 указва, че MP поддържа инструкция CPUID, чрез която е възможна идентификацията на процесора.

Останалите битове на рег. EFLAGS не се използват.

II. 2. 4. Програмен брояч EIP (EIP - Extended Instruction Pointer) - съдържа адреса на отместването от началото на текущия кодов сегмент, на което се намира следващата за изпълнение инструкция (фиг. 12). Той не е директно видим за програмиста, а се контролира неявно от инструкции за трансфер, прекъсвания и изключения.



Фиг. 12. Регистър програмен брояч EIP на MP 80x86

### II. 3. Нишки (threads)

При съвременните MP общата работа (операционна система, различни потребителски програми) се разделя на отделни процеси, а всеки от процесите се разделя на отделни *нишки*, които могат да се изпълняват едновременно, тъй като те са относително независими и

комуникацията между тях е минимална. Едновременното обработване на различни части от кода довежда до намаляване на времето за изпълнение на цялата задача в сравнение с последователната обработка.

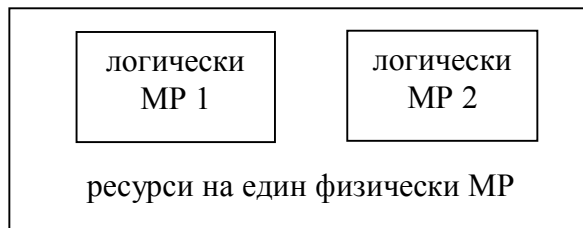
### II. 3. 1. Супер – нишки

Супернишковото изпълнение (супертрединг) е вид мултитрединг, който позволява различни нишки, да се изпълняват от един МР, без реално да ги изпълнява по едно и също време. Това се приема по-скоро за времеделение, отколкото едновременен мултитрединг (Simultaneous Multi-Threading - SMT). Функционалните блокове на МР понякога остават свободни, докато се изпълняват дълги инструкции от една нишка. Супер нишките се стремят да се възползват от неизползваните процесорни цикли, като се изпълняват инструкции от друга нишка, докато предишната нишка стане готова да възобнови изпълнението.

Подобрения на този подход се реализират чрез SMT, което позволява изпълнението на инструкциите от няколко нишки по едно и също време. Например, при двудрен супер-нишков процесор с четири функционални блока, ако нишка 1 подава три инструкции, един от блоковете остава неизползван. В SMT процесор е възможно нишка 2 да подаде инструкция към този блок, като така ще се достигне пълно използване на ресурсите на процесора.

### II. 3. 2. Хипернишкова технология

Хипернишковата технология (Hyper-Threading Technology - HT) е термин на Intel за SMT и прилагането му в процесорите Atom, Core i3, Core i5, Core i7, Itanium, Pentium 4 и Xeon. Тя се използва за подобряване на паралелизацията на изчисленията. За всяко процесорно ядро, което е налично физически, операционната система адресира два виртуални процесора, като разпределя работата между тях, когато е възможно. Хипер нишките, изискват не само операционната система да поддържа множество процесори, но също така да е специално оптимизирана за HT. Intel препоръчва изключване на HT при използване на операционни системи, които не са оптимизирани за тази функция на чипа.



Фиг. 13. HT микропроцесор

НТ работи чрез дублиране на някои части на процесора - тези, които съхраняват състоянието на архитектурата, но не дублира основните ресурси (фиг. 13). Това дава възможност на НТ процесорът да се появи като два "логически" процесора за водещата операционна система, което и позволява да управлява две нишки или процеса едновременно.

В процесор без НТ, когато изпълнителните устройства няма да бъдат използвани от текущата задача и особено, когато МР чака, при НТ процесор тези ресурси се използват за изпълнение на друга задача. (Процесорът може да чака поради липса на данни в кеша, грешно предсказване на преходите или зависимост от данни.) Минималното, което е необходимо, за да се използват предимствата на НТ технологията, е поддържане на симетрична мултипроцесорна работа (SMP) в операционната система, тъй като логическите процесори се явяват като отделни стандартни процесори.

Възможно е да се оптимизира поведението на операционната система на мултипроцесорни хипернишкови системи. Например, при SMT система с два физически процесора, които са едновременно хипер-нишковы (общо четири логически процесора), ако планирането на процесите в операционната система не разполага с НТ, тя ще третира всички четири процесора като един и същ. Ако само два процеса са в правото си да работят, може да се избере тези процеси да се обслужват на два логически процесора, които да принадлежат към един от физическите процесори. Този процесор ще стане изключително натоварен, докато другият ще бъде свободен, което води до лоша производителност. Проблемът може да се избегне чрез разглеждане на логическите процесори различно от физическите процесори.

### II. 3. 3. Многонишкова обработка

Многонишковите МР хардуерно поддържат ефективното изпълнение на множество нишки. Те се отличават от многопроцесорните системи по това, че нишките трябва да споделят ресурсите на единствено ядро: ALU, умножител, кеш-памет. При многонишковите системи целта е, да се увеличи използването на дадено ядро, като се използва паралелизма не само на ниво нишка, но и на ниво инструкция.

#### *Предимства на многонишковата обработка:*

- Ако при дадена нишка се получи кеш-пропуск (cache miss), изпълнението може да продължи с друга (чакаща изпълнение) нишка, възползвайки се по такъв начин от неизползваните изчислителни ресурси. Това ще доведе до по-бързото общо

изпълнение, тъй като тези ресурси биха останали неизползвани ако се изпълнява само една нишка;

- Ако дадена нишка не може да използва всички ресурси на процесора (защото изпълнението на инструкциите зависи от резултатите от други инструкции), изпълнението на друга нишка позволява тези ресурси да не останат неизползвани;
- Ако няколко нишки работят с един и същ набор от данни, те могат да споделят кеша, което води до по-доброто използване на кеш-паметта или тя може да се използва за синхронизация между нишките.

#### *Недостатъци на многонишковата обработка:*

- Множество нишки могат да си пречат едни на други, когато си споделят хардуерни ресурси като например кеш-памет;
- Времето за изпълнение на единични нишки не се подобрява, но може да се влоши дори и при изпълнението само на една нишка. Това се дължи на по-ниските честоти и/или на допълнителните състояния при конвейерната обработка, които са необходими за хардуерното превключване между нишките;
- Хардуерната поддръжка на многонишковата обработка е по-видима за софтуера в сравнение с многопроцесорната, и следователно изисква повече промени както при приложните програми, така и при операционните системи;
- Подредбата на нишките (thread scheduling) е важен проблем при многонишковата обработка.

### II. 3. 4. Видове многонишкова обработка

#### *а) Блокова многонишкова обработка (Block multithreading, Coarse-grained multithreading)*

Този най-прост тип многонишкова обработка се среща когато дадена нишка се изпълнява докато се появи събитие, което би довело до дълъг престой на процесора. Такова събитие може да бъде например кеш-пропуск, който довежда до достъп до памет извън МР и може да отнеме стотици процесорни тактове, докато бъдат доставени данните. Вместо да чака събитието да бъде разрешено, многонишковият процесор би превключил изпълнението на друга нишка, която е готова за изпълнение. Само, когато данните за предишната нишка бъдат извлечени от паметта, тогава предишната нишка ще бъде включена в списъка с готови за изпълнение нишки. Например:

1. Цикъл  $i$  : изпълнява се инструкция  $j$  от нишка А
2. Цикъл  $i+1$ : изпълнява се инструкция  $j+1$  от нишка А
3. Цикъл  $i+2$ : изпълнява се инструкция  $j+2$  от нишка А, инструкция за извличане на данни от паметта, която не открива данните в никоя от кеш-паметите

4. Цикъл  $i+3$ : извиква се логиката за превключване между нишките, превключване на нишка В

5. Цикъл  $i+4$ : изпълнява се инструкция  $k$  от нишка В

6. Цикъл  $i+5$ : изпълнява се инструкция  $k+1$  от нишка В

Целта на хардуерната поддръжка на многонишковата обработка е да позволи бързото превключване между блокирана нишка и готовата за изпълнение нишка. За да се достигне тази цел трябва да се мултиплицират регистрите, видими за програмата, а също така и някои контролни регистри на процесора (пр. IP).

Превключването от една нишка на друга, означава превключване от един набор от регистри на друг. Предимствата на тази архитектура са:

1) Превключването на нишката може да стане за един такт на процесора.

2) Всяка от нишките се изпълнява самостоятелно и никоя от тях не споделя хардуерни ресурси. Това намалява количеството софтуерни промени, необходими при приложните програми и операционните системи за поддържане на многонишкова обработка.

За да бъде ефективно превключването между активните нишки, всяка активна нишка трябва да има своя банка от регистри. Например, за да се превключва бързо между две нишки, регистрите трябва да са дублирани.

*б) Прескачаща многонишкова обработка (Interleaved multithreading, Fine-grained multithreading.* При нея изпълнението е следното:

1. Цикъл  $i+1$ : изпълнява се една инструкция от нишка В

2. Цикъл  $i+2$ : изпълнява се една инструкция от нишка С

Целта на този тип многонишкова обработка е да премахне от конвейъра всички спирания, зависещи от данни. Тъй като нишките са относително независими една спрямо друга, то вероятността за необходимост от прехвърляне на данни между инструкции, намиращи се на различни етапи в конвейъра, е малка. Времевият интервал, заделен за всяка активна нишка, е един такт на процесора.

*в) Едновременна многонишкова обработка (SMT)*

Това е най-сложният вид многонишкова обработка, който се прилага при суперскаларните процесори. Един обикновен суперскаларен МР изпълнява множество инструкции от една нишка на всеки такт. При SMT, суперскаларният МР може да изпълнява инструкции от множество нишки за един такт. Тъй като една нишка има ограничено ниво на паралелизъм на ниво инструкция, този тип многонишкова обработка се опитва да усвои паралелизма, наличен между различните нишки, за да намали загубите, свързани с неизползването на отделените интервали за изпълнение. Например:

1. Цикъл  $i$  : изпълняват се едновременно инструкции  $j$  и  $j+1$  от нишка А, и инструкция  $k$  от нишка В
2. Цикъл  $i+1$ : изпълняват се едновременно инструкция  $j+2$  от нишка А, инструкция  $k+1$  от нишка В, инструкция  $m$  от нишка С
3. Цикъл  $i+2$ : изпълняват се едновременно инструкция  $j+3$  от нишка А, инструкции  $m+1$  и  $m+2$  от нишка С

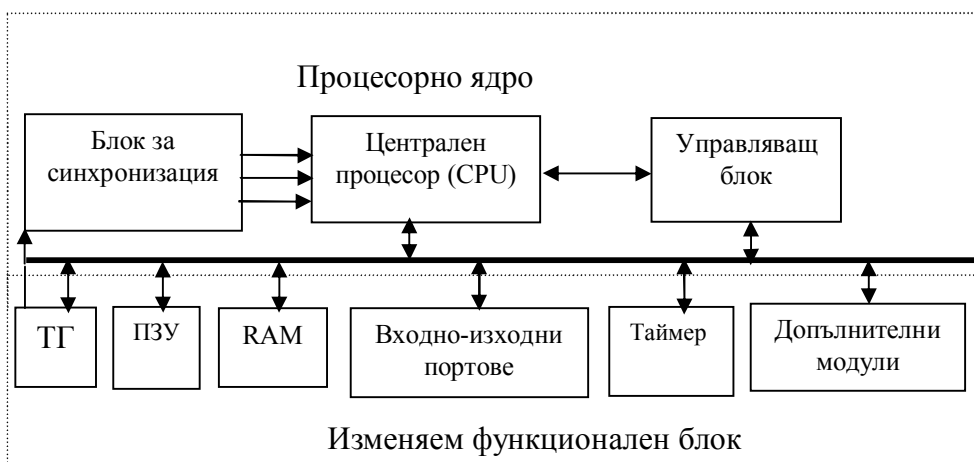
## ГЛАВА III

### Микроконтролери (MCU).

#### III. 1. Обща структура на микроконтролер

При модулния принцип на разработване, всички MCU от една фамилия съдържат процесорно ядро (CPU), еднакво за всички MCU на дадено семейство и изменяем функционален блок, по който се различават различните модели MCU (фиг. 14). *Процесорното ядро* включва: CPU, шини за адреси, данни и управление, схема за синхронизация на MCU, модул за управление на режимите на работа на MCU, поддържащ режимите с намалена консумация, включване/изключване на MCU и др. [34]

*Изменяемият функционален блок* е съдържа памети от различен тип и с различен обем, тактови генератори, таймери, компаратори, ADC, интерфейси и др. В сравнително по-простите MCU, блокът за обработка на прекъсванията влиза в състава на CPU. В по-сложните MCU, той е обособен в отделен модул с разширени възможности.



Фиг. 14. Модулна организация на микроконтролер

- Система инструкции на MCU – включва инструкции за прехвърляне на данни, аритметични и логически инструкции, инструкции за преход. За реализиране на независимо управление на портовете (регистрите), в повечето съвременни MCU са предвидени инструкции за управление на ниво битове. Тези инструкции позволяват да се съкрати обема на програмата и времето за нейното изпълнение. Друга група инструкции управлява ресурсите на MCU, като чрез тях се настройват режимите на работа на портовете, работата на таймера и др.

В сравнение със системата инструкции на универсалния МР, MCU има по-малко аритметико-логически инструкции за сметка на инструкциите за прехвърляне на данни и управление. Тази особеност е



свързана със сферата на използване на MCU, която преди всичко изисква контрол на обкръжаващата среда и формиране на управляващи въздействия.

- *Функционални възможности* – характеризират приложните качества на MCU и се определят до голяма степен от вградените периферни модули – ADC, DAC, таймери, серийни интерфейси и др.
- *Адресируемост* – показва възможността MCU да се свързва с определен брой адреси на клетки от паметта и зависи от разредността на адресната магистрала [2, 7, 11].

### **III. 2. Основни функционални блокове на MCU**

#### **III. 2. 1. Процесорно ядро (CPU)**

Съвременните процесори се реализират със CISC ядро (Motorola HC11, Intel MSC-51, AMD Am186) или с RISC ядро (Microchip PIC, Texas Instruments MSP430, Atmel AVR). Техническото решение на процесорното ядро се определя от следните параметри:

а) *Архитектура*: включва набора регистри, организацията на паметта, видовете адресации, системата от инструкции. От програмна гледна точка, това е съвкупността от инструкции, които MCU е в състояние технически да изпълни, включително с всички средства за изготвяне на програмите.

б) *Схемотехнически решения*: определят вътрешната диаграма на функциониране – последователността на предаване на данните по шините между регистрите, паметта и ALU, производителност, брой изводи и предназначението им и др..

в) *Технология на производство*: определя допустимата сложност на схемата, максималната честота на превключване, консумацията на енергия и др..

#### **III. 2. 2. Вградена памет.**

Паметта се разделя на постоянна памет и памет за временно съхранение на данните RAM. Водещо място в групата на вградените и външните постоянни паметите имат FLASH-паметите. Те са модификация на EEPROM-паметта със значително увеличаване на обема – до стотици kB. Времето за съхраняване на информацията е 10 год., а захранващото им напрежение е 1,8 V. Изтриването на паметта може да се извършва поотделно на страници или изцяло (табл. 5).

За вградена RAM памет в повечето случаи се използват модули статична памет SRAM. Тя има възможност за съхраняване на данните при намаляване на честотата до пълното спиране на процесора, което се използва в режимите с намалена консумация. Тези модули нямат голям обем – от няколко до десетки kB. В мощни системи, изискващи големи обеми RAM, се включват външни модули SRAM и динамична памет (DRAM, SDRAM).

Таблица 5. Видове програмни памет.

| Тип програмна памет                     | Начин на програмиране                          | Възможност за корекция на програмния код | Време за достъп | Скорост на програмиране/<br>Време за изтриване |
|---|--|--|-----------------|--|
| Маскируема ROM (ROM)                    | Програмиране при производство (мин. бр. 1000)  | Няма възможност                          | 10 ns (100MHz)  | 2 - 4 седмици/ няма                            |
| Еднократно програмируема ROM (OTP ROM)  | Чрез програматор                               | Няма възможност                          | 100 ns (10MHz)  | 50 думи/сек./ няма                             |
| UV програмируема ROM (UV EPROM)         | Чрез програматор или вграден програматор (ICP) | до 100 пъти                              | 100 ns (10MHz)  | 50 думи/сек./ 10-30 мин.                       |
| Електрически програмируема ROM (EEPROM) | Чрез програматор или вграден програматор (ICP) | до 10,000 пъти                           | 200 ns (5MHz)   | 100 думи/сек./ 10 ms                           |
| Flash програмируема ROM (Flash EPROM)   | Чрез програматор или вграден програматор (ICP) | До 100,000 пъти                          | 100 ns (10MHz)  | 500 words/sec/ 5 ms                            |
| Фероелектрическа RAM (FRAM)             | Чрез програматор или вграден програматор (ICP) | До $10^{12}$ пъти                        | 200 ns (5MHz)   | 2 Mbytes/sec/ не е необходимо                  |
| Статична RAM (SRAM)                     | вграден (ICP) програматор                      | неограничено                             | 20 ns (50MHz)   | 20 Mbytes/sec/ не е необходимо                 |

### III. 2. 3. Периферни модули на микроконтролерите.

#### а) Входно-изходни портове (I/O)

Определящо значение за работата на MCU имат началният и крайният етап от взаимодействието с външните устройства: въвеждане на информация от тях и извеждане на информация или въздействие върху тях. По-важните характеристики на входните портове са

свързани с възможностите им за включване на датчици – полярност, параметри на сигнала и др. При изходните портове по-важните характеристики са вида и параметрите на управляващия сигнал, подаван към външно устройство. В микропроцесорните системи, външните устройства обикновено са датчици, изпълнителни устройства, устройства за въвеждане/извеждане на информация от оператор или външна памет, индикатори.

Всеки MCU притежава определен брой външни входно-изходни линии, обединени в групи по 4, 8 или 16 извода, които представляват паралелните портове на MCU. Разредността им може да е нестандартна, напр. 5-разредния порт на PIC16F86. Чрез тях MCU взаимодейства с различни външни устройства – регистрира нивата на входните сигнали и установява нивата на изходните сигнали.

- По тип на сигнала, портовете са:

1. *Дискретни (цифрови)* - използват се за вход/изход на дискретните значения на логическата “0” или “1”.

2. *Аналогови* – чрез тях се приемат сигнали на входа на ADC или др. аналогови схеми и се извеждат изходни сигнали на DAC и др. аналогови схеми.

3. *Комбиниранни* – пренастроят се за аналогови или цифрови сигнали.

- По посока на предаване на сигнала, портовете са:

1. *Еднопосочни* – предназначени са само за въвеждане (входни портове) или само за извеждане (изходни портове).

2. *Двупосочни* – посоката им се определя в процеса на програмното управление на схемата.

3. *Портове с алтернативна функция* – отделните им линии са свързани с вградените модули. Ако съответният модул не е задействан, то линиите могат да се използват като обикновени портове. Ако модулът се задейства, то линиите автоматично или програмно се конфигурират в съответствие с функционалното предназначение и не могат да се използват с универсално предназначение.

4. *Портове с повече от една функция* - линиите се свързват с вградените модули и се конфигурират за повече от една функция чрез конфигурационни регистри от специален блок за управление на изводите (Pin Control Modul).

- По начин на обмен на сигналите, портовете са:

1. *С програмно управляем вход/изход* – установяването и отчитането на данните се определя само по време на изчислителния процес. Няма защита от повторно приемане на едно и също значение (постоянно) на извода или отчитане по време на преходен процес на извода.

2. *Със стробиране* – всяка операция за вход/изход се потвърждава със синхронизиращ импулс (строб) от страна на източника на сигнала (при извеждане – от процесора, а при въвеждане – от външно устройство). Постъпването на информация в приемниците става само чрез строб, което защитава от приемане на данни по време на преходен процес.

3. *С обмен на сигнали за готовност* - този режим най-често се използва за обмен на данни с друго устройство по паралелната шина. Освен сигнали за синхронизация от страна на предавателя, се използват сигнали за потвърждаване (готовност за следващ обмен) от страна на приемника. Това позволява да се управлява интензивността на обмена на двете взаимодействащи страни и предотвратява загуба на данни, когато една от тях е претоварена. Във вградените модули на MCU, даденият режим по-често се реализира програмно-апаратно.

- Еднопосочни портове.

А) *Входни портове*. При тях външните данни постъпват на изходите на входния порт, минават през тригер на Шмид (ТШ) или през схема за защита от смущения и по вътрешен сигнал за четене, се записват в регистър за данни, на изхода на който, последователно се приемат от CPU. В повечето MCU се използва ТШ, тъй като има хистерезис на нивото на входното напрежение и предпазва от многократно превключване на входните схеми при смущения. Схемите за защита от смущения (напр. Z8), въвеждат инерционност при превключването и премахват влиянието на кратките по дължина импулси.

Към входа може да се включат резистори, поддържащи лог. "1" (Pullup) и лог. "0" (Pulldown). Те са предназначени за осигуряване на устойчиво състояние "0" или "1" на входа и предотвратяване на превключвания от смущения в моменти, когато на входа не се подава външен сигнал, напр. при "висящи" входове (неизползвани и несвързани входове към др. схеми). Те се управляват чрез специални регистри и могат да бъдат напълно изключени или в режим "Pullup"/"Pulldown".

Б) *Изходните портове* са с двутактна изходна схема (комплементарни), с еднотактна изходна схема и вътрешен товар или с отворен изход (с отворен дрейн);

Портовете с *двутактна изходна схема* са най-разпространени – AtmelAVR, Motorola 68HC11, 68HC08, AMD AM186, Microchip PICmicro и др. Схемата функционира като изходните данни се записват в изходен регистър по вътрешен сигнал за запис (#WR) и чрез лог. схема управляват изходни транзистори T1 и T2. Ако в регистъра е записана лог. "1", е отпушен единият транзистор (T1), а другият (T2) е запушен: напрежението на изхода е Vcc (лог.1). Ако в

регистъра е записана лог."0", T2 е отпушен, T1 е запушен: изходът е свързан с масата, т.е. установена е лог. 0.

Регистърът за разрешение на изходите се управлява от сиг. ОЕ (Output Enable). Ако в него е записана лог. 0, то схемата работи както е описано по-горе. Ако е записана лог. 1, то двата транзистора са запущени и схемата преминава във високоимпедансно състояние. В това състояние, изходното съпротивление на порта е много високо и фактически той е прекъснат от MCU. Това е необходимо, ако:

- към изходния порт са включени други схеми и е необходимо да се разделят шините за предаване на данни с тези устройства. Например, ако MCU се използва като периферен контролер и неговият изходен порт е свързан към периферната шина на друг MCU (главен) и към нея са свързани още няколко периферни MCU;

- в схеми с двупосочни портове;

Предимствата на схемата са, че има значителен максимален входен (в състояние "0") и изходен (в състояние "1") ток на изхода: 2–6 mA за каскади с нормална товароспособност (например Fujitsu MB90) и 5–30 mA за каскади с голяма товароспособност (PICMicro, AVR). Съществуват и отделни микросхеми със свръхголяма товароспособност – до 60-90 mA (PIC17). Големият изходен ток позволява непосредствено от изхода на MCU да се управлява достатъчно мощен товар без схеми за усилване и съгласуване на сигнала: светодиоди, релета, мощни електронни ключове (транзистори, тиристори). Това значително опростява схемата на MCU.

Недостатъци: При програмирането е необходимо да се управлява регистър за разрешение на изходите. Има повишена консумация и ниво на шумовете при превключване, което се влияе особено много от честотата на превключване. За ограничаване на тока в момента на превключване, понякога се използват демпфиращи схеми, които понижават бързодействието на портовете. Относително сложна е вътрешната схема, което повишава сложността и себестойността на MCU като цяло.

*Изходните портове с еднотактна изходна схема и вътрешен товар* се използват при фамилия MCS-51. Когато в регистъра е записана лог.1, транзисторът е запушен и на изхода, през резистор R<sub>L</sub>, се установява V<sub>cc</sub> – лог.1. Когато в регистъра е записана лог. 0, се отпушва транзистора и на изхода се установява лог.0. Резистор R<sub>L</sub> се включва между изводите на захранването. За да се избегне голям ток през резистора и прегряването му, то съпротивлението му се избира да е с висока стойност – от 10 до 100k $\Omega$ . Големото съпротивление позволява да се свържат непосредствено няколко изхода, без опасност от тяхното насрещно включване, защото, ако на един изход, на който е установена лог. 0, се подаде лог.1, то мощността отделена в резистора

ще е малка, той няма да прегрее и изходното стъпало няма да се повреди.

Предимствата са, че схемата е проста и управлението е само с един регистър; има възможност да се организира включване на няколко такива извода в една външна шина без допълнителни схеми. Лесно може да се реализира квазидвупосочен порт за вход/изход.

Недостатък е малкият изходен ток (в състояние "1"), ограничен от  $R_L$  – стотици  $\mu A$ , което не дава възможност да се управлява относително мощен товар без допълнителни схеми.

*Изходни портове с отворен изход* се използват в много MCU (AMD Am186 – там това е един от режимите на порта; PICmicro). Изходното стъпало е реализирано чрез еднотактна схема с външен товар. Принципът на действие е аналогичен с този на еднотактната изходно стъпало.

Предимствата са, че схемата е проста и управлението е само с един регистър; външното захранващо напрежение на товара  $V_{sехт}$  може да бъде различно от захранването на MCU. Това може да бъде удобно за схеми с различни нива на лог.1, например 3,3V и 5V. Ако външното напрежение е достатъчно високо, то може непосредствено да се управляват високоволтови товари (MCU фамилия PICmicro| допуска включване на  $V_{sехт}$  до 15V, при захранване на ядрото 2,6V). Има възможност да се организира включване на няколко такива извода в една външна шина без допълнителни схеми. При това може да се избере необходимото съпротивление  $R_L$ , например, стандарта I2C изисква  $R_L$  да е 2,2 k $\Omega$ . Лесно може да се реализира квазидвупосочен порт за вход/изход.

Недостатъци: необходимост от външен товар; малък изходен ток (в състояние "1"), ограничен от външния товарен резистор;

- Двупосочни портове и портове с алтернативни функции

Схемата им представлява квазидвупосочен порт и е аналогична на схемата на изходен порт с еднотактно изходно стъпало. Регистърът за входни данни е свързан с външния изходен порт. Преди да се приемат входните данни, в регистъра за изходни данни предварително трябва да се запише лог. 1. Това запуща транзистора и игнорира влиянието на изходния порт на входния сигнал. Тази схема се използва при MCU MCS-51.

По-често се използва схема с превключваем двупосочен порт с комплементарно изходно стъпало. Схемата му обединява схемите на входен и изходен порт с двутактна изходна схема. Превключването на порта във входен режим, се осъществява чрез запис на лог. 1 във входно-изходния регистър. В този случай двата транзистора са запущени и изходния порт не влияе на входния сигнал. В

двупосочните портове, “изтеглящите” резистори pullup/pulldown се включват само във входен режим.

Освен функциите на порта въвеждане/извеждане, изводите на MCU могат да се използват за връзка с вградените модули на MCU, с ядрото, паметта и схемите за управление. Дадените функции са алтернативни и при изпълнението им обикновено основните схеми се настройват за въвеждане или се изключват.

#### *б) Таймери - броячи*

Таймерните модули са основни модули за микропроцесорните системи. Всеки MCU има в структурата си таймер, който може да участва в работата на всички останали подсистеми. Наличието на броячи и делители и работата с точна кварцово стабилизирана честота, позволява да се създават програми, измерващи времеви параметри на сигналите или генериращи последователности от импулси с програмируеми параметри.

Микроконтролерите имат един или повече таймери, с които могат да се генерират импулси с определена честота и коефициент на запълване, интервали от време и прекъсвания в системата. Това са изходните функции на таймера, при които се използва функция OUTPUT COMPARE. Тази функция се използва и за генериране на сигнали с PWM, а някои MCU имат отделен PWM модул. Чрез функция INPUT CAPTURE (IC) се детектират външни събития и могат да се измерват времеви параметри на входни сигнали. Работата на таймерната система се базира на брояч, който работи в непрекъснат режим, като обикновено може да се конфигурира за работа в:

а) инкрементиращ режим (Up mode) , в който таймерът увеличава съдържанието си на всеки такт до предварително зададена стойност, след което се нулира.

б) режим на непрекъснато броене (Continuous mode) до максималната стойност, която може да съдържа дадения таймер.

в) режим, в който таймерът увеличава съдържанието си до определена стойност и после го намалява обратно до 0 (Up/Down mode).

Таймерният модул работи като таймер или като брояч. Разредността на регистъра-брояч определя разредността на таймера-брояч.

В режим *таймер*, тактуването на брояча се изпълнява от сигнала Fint (фиг.10) за вътрешна синхронизация на MCU и обикновено се формира от генератор на основния тактов сигнал. Измерването на времевите интервали се извършва в периодите на сигнала Fint. Предварителният делител се използва за намаляване на тактовата честота, подавана на регистъра-брояч. Това позволява изчисления в по-големи интервали, но увеличава стъпката на дискретизация и съответно се намалява точността. Предварителният делител може да е с фиксиран или програмируем коефициент на

делене (1, 2, 4, 8, ...). Регистърът-бройч изчислява дължината на временния интервал като отчита тактовете след предварителния делител. Допълнителен делител след брояча (постделител) се среща рядко (PISmicro) и служи за увеличаване на периода на установяване на флага за препълване TF. Обикновено се реализира с допълнителни разреди на регистъра-бройч, недостъпни за четене/запис и може да се програмира да работи с различни коефициенти на делене.

Флагът за препълване на таймера TF се установява при преход на всички разреди на регистъра-бройч от лог.1 в лог.0 и обикновено указва края на временния интервал. По него може да се генерира заявка за прекъсване. TF участва във верига за обратна връзка, задаваща режима на работа на таймера.

1) *Режим на еднократно броене*: след препълване, в регистъра-бройч се записва лог. 0 и броенето се преустановява. Следващ цикъл на броене се стартира чрез специална инструкция от програмата.

2) *Режим на многократно броене с пълен цикъл*: След препълване, в регистъра-бройч се записва лог.0 и броенето започва отново. Пълният цикъл на броене на таймера ще бъде  $2^k$  такта, където  $k$  е разредността на брояча + постделителя.

3) *Режим на многократно броене с автозареждане*: След препълване, в регистъра-бройч се записва значението от регистъра за презареждане. Така броенето може да не започне от 0 и така да се намали дължината на цикъла на таймера.

В много процесори има специален извод INgate, който изпълнява функция на разрешаване на броене от външен сигнал. С помощта на този механизъм лесно се измерва период, който се определя от дължината на импулса на входа INgate.

За разлика от режим таймер, в режим *бройч* се избира тактуване от външен импулсен сигнал, подаван на вход INcnt, при което се отброяват импулсите на външния сигнал. Инкрементирането/декрементирането на брояча е по фронта на сигнала, който в дадения случай се явява “външно събитие”. Полярността му може да се програмира, а в останалата си част, функционирането му е аналогично на режим таймер.

в) *Таймери/бройчи със схеми за входно захващане, изходно сравнение и генериране на сигнали с PWM (CAPTURE/COMPARE/PWM (CCP модули))*.

Модули CCP се явяват усъвършенстване на таймерите/бройчите, като изпълняват подобни функции, но изискват по-малка програмна поддръжка. По-гъвкави са при настройване за различни задачи и позволяват да се постигне много високо бързодействие. Най-голяма ефективност имат при работа с външни периодични и непериодични сигнали при решаване на следните задачи:



- фиксиране на момента на настъпване на външно събитие (фронт);
- определяне на честотата и дължината на импулсите на външен сигнал, фазовото отместване на няколко сигнала;
- формиране на единични импулси с програмируема дължина;
- генериране на периодични последователности от импулси с програмируема честота, дължина, фазово отместване (при няколко изходни сигнала) на един или няколко извода;
- генериране на сигнали с PWM. При PWM, честотата на сигнала остава постоянна, а дължината на положителния и отрицателния импулс се програмира. Основна характеристика на сигнал с PWM, е коефициентът на запълване на импулса: отношението на периода към дължината на положителния импулс. Модул PWM, с включена към изхода му интегрираща верига, представлява най-простия DAC.

#### 1) Функция *OUTPUT COMPARE (OC)* - изходно сравнение.

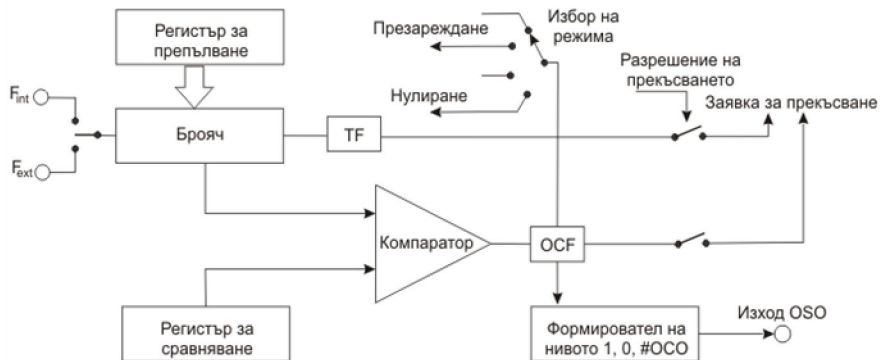
Реализира се с многоразреден цифров компаратор (фиг.15), който непрекъснато сравнява изменящата се във времето текуща стойност на брояча със стойност, записана в регистъра за сравнение. В момента, в който станат равни двете стойности, се установява флаг OCF (Output Compare Flag) и сигнала на изхода на таймера OCO (Output Compare Output) се изменя. Възможни са три варианта за програмно задаване на промяната на сигнала: установяване на лог. 0, установяване на лог.1 и инвертиране на сигнала на извод OCO. След установяване на флага за препълване OCF, броячът може да бъде нулиран или зареден с определена стойност. Освен това, по този флаг може да се генерира заявка за прекъсване, ако то е разрешено. Типовите приложения на модул CCP в режим Output Compare са:

- Генериране на сигнал с определена честота: изходът на таймера се настройва в режим на инверсия, управлението на таймера е чрез нулиране по флаг OCF, а в регистъра за сравнение се записва значение, равно на полупериода на формираната честота. При всяко ОС събитие, веднъж в полупериода сигналът OCO се инвертира и се формира предният или задният фронт на сигнала.

- Формиране на единичен импулс с определена дължина: изходът на таймера се настройва в режим на установяване на лог.0 на извод OCO, в регистъра за сравнение се записва дължината на импулса, таймерът се нулира и едновременно с това на извод OCO се установява лог.1 (преден фронт). При ОС събитие, на извод OCO се установява лог.0 (заден фронт).

- Очакване на определен брой импулси на входа на брояча (сигнал Fext): таймерът се настройва в режим "Брояч" и се нулира, в регистъра за сравнение се записва необходимия брой

импулси, разрешава се прекъсване при ОС събитие (по флаг OCF). След постъпване на зададения брой импулси, се генерира заявка за прекъсване.



Фиг.15. Модул за изходно сравнение (Output Compare)

- Делител на честота с коефициент  $N$ , кратен на 2: таймерът се настройва в режим Брояч, задава се нулиране по флаг OCF, изход ОСО се настройва в режим инверсия, в регистъра за сравнение се записва число  $N/2$ .

## 2) Функция INPUT CAPTURE (IC) - входно захващане.

Тази функция е предназначена за фиксиране на момента на възникване на външно събитие (фиг.16): когато на извод ICI (Input Capture Input) се детектира активен фронт (предварително зададен),



Фиг.16. Модул за входно захващане (Input Capture)

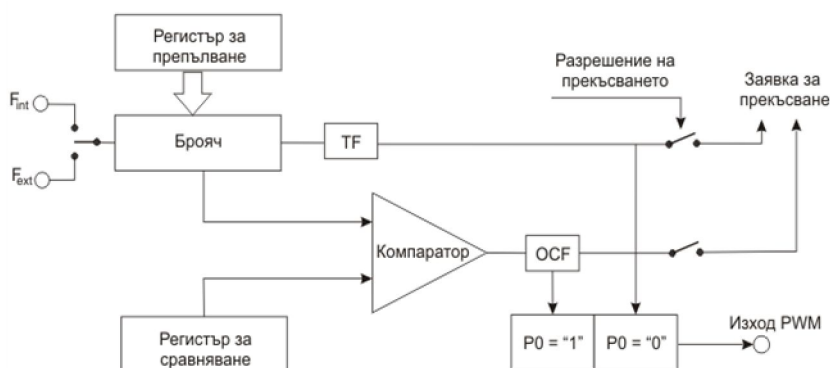
то текущото значение на брояча се записва в специален регистър (capture register), откъдето може да бъде прочетен програмно.

При някои MCU, входното захващане може да бъде управлявано програмно чрез инструкция за обръщение към специалния регистър. При настъпване на събитие IC, се установява флаг ICF и може да се генерира прекъсване. Освен това, броячът може да се нулира или да се зареди с определена стойност.

Типовите приложения на модул ССР в режим Input Capture са: определяне на периода, честотата и коефициента на запълване на сигнала на вход ICI; определяне на относителното време на възникване на различни събития.

### 3) Генериране на сигнали с PWM

Реализира се чрез схема (фиг.17), която е модифициран вариант на схемата за ОС. Разликата е в това, че изходът се управлява както от компаратор, така и от схема за препълване на брояча.

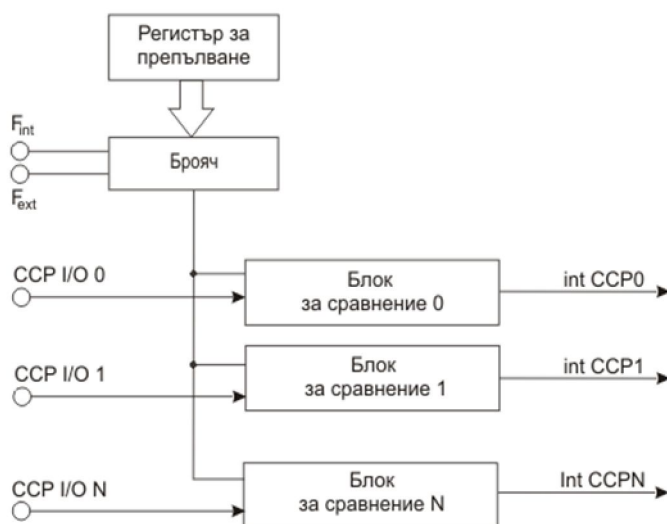


Фиг.17. Модул за генериране на PWM сигнал

Предният фронт на сигнала с PWM (0->1), се формира при настъпване на събитие ОС (стойността на брояча е равна на стойността в регистъра за сравнение), а задният фронт (1->0) – при препълване на брояча. Периодът на сигнала с PWM е равен на честотата на препълване на брояча и се определя от регистър за презареждане. Дължината на положителния импулс в периода се определя като разлика между максималното значение на брояча + 1 и съдържанието на регистъра за сравнение.

#### 4) Процесори на събития

Под управление на един брояч могат да се обединят няколко канала за входно захващане, изходно сравнение и генериране на сигнали с PWM (фиг.18), като всеки от тях може да бъде индивидуално конфигуриран за работа в някой от режимите.



Фиг. 18. Процесор на събития

Процесорите на събития (масив програмируеми броячи PCA (Programmable Counter Array) (Intel), блок CAPCOM (Infineon), блок TIM8 (Motorola)) позволяват да се формират взаимно синхронизирани изходни сигнали – с постоянна фазова разлика или с отчитане на временните разлики между събитията, а като частен случай – с фазови разлики между сигналите.

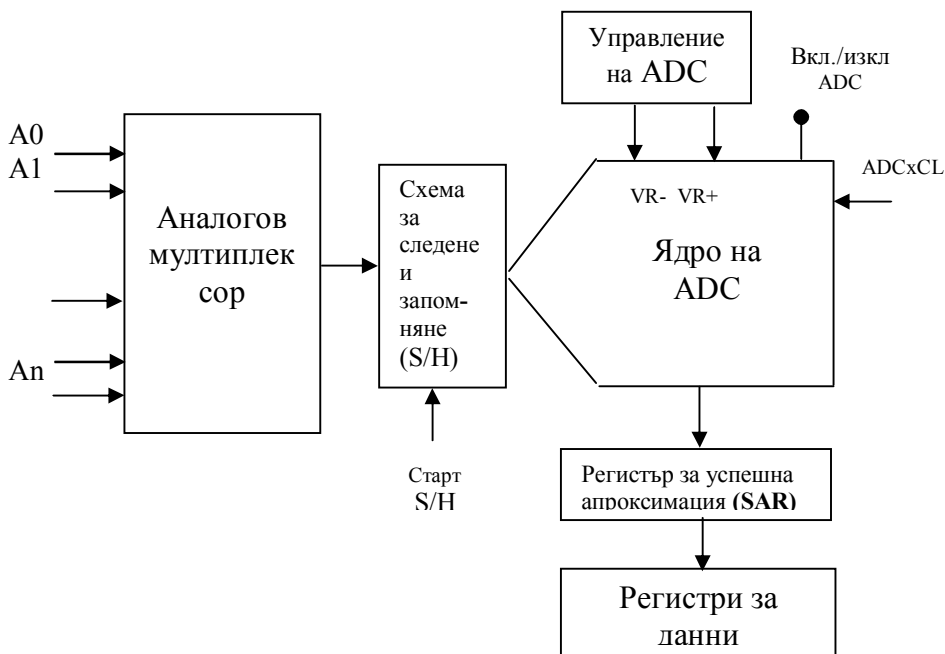
#### г) Аналогово-цифров преобразувател (ADC)

Модул ADC е предназначен за приемане от CPU на аналогови сигнали от датчици на физически величини и преобразуване на напрежението на тези сигнали в двоичен код с цел по-нататъшната им програмна обработка.

ADC преобразува аналоговото напрежение на входа в цифров код. Полученото значение се записва в регистър на данните. Обикновено вградените в MCU ADC са с поразредно кодиране на сигнала и времето им за преобразуване е няколко десетки  $\mu s$ , в зависимост от работната честота на ADC. Процесът на преобразуване завършва с установяване на флаг и ако е разрешено, се генерира заявка за прекъсване. В съвременните MP/MCU най-разпространени са 8-, 10- и 12-битовите ADC, а напоследък и 14- и 16-битовите ADC.

Аналоговият мултиплексор (фиг.19) избира един от аналоговите канали (изводи) и го включва към вътрешния вход на ADC за преобразуване, което може да се реализира програмно или апаратно (режим на сканиране на каналите). За по-голямо удобство, в режим на сканиране някои MCU (Intel 8051GB, Fujitsu MB90) имат няколко регистъра за данни (по един на канал). Източникът на опорно напрежение  $V_{ref}$  определя диапазона на напреженията на аналоговите входове и разрешаващата способност на ADC, която е равна на

$V_{ref} / 2^n$  ( $n$  – разредност на ADC). Ако напрежението на входа не е голямо, то точността на преобразуване може да се увеличи чрез намаляване на  $V_{ref}$ .



Фиг. 19. Модул за аналогово-цифрово преобразуване

Диапазонът на допустимите значения на  $V_{ref}$  е в рамките на захранващото напрежение на MCU. Опорните източници могат да са външни и се включват към специални изводи на MCU или вътрешни фиксирани/програмируеми (с помощта на DAC). Свързването на ADC към източниците на  $V_{ref}$  се извършва с помощта на комутатор. Комутаторът на сигнала за включване на ADC позволява да започне процеса на преобразуване и определя един от възможните режими на работа на ADC.

- 1) Периодично преобразуване. В този режим ADC периодично се пуска чрез сигнал от тактовия генератор или вградения таймер.
- 2) Външен старт: чрез външен сигнал – позволява точно да се определи момента на отчитане на аналоговото напрежение на входа.
- 3) Програмно-управляем старт: чрез установяване на специален бит.

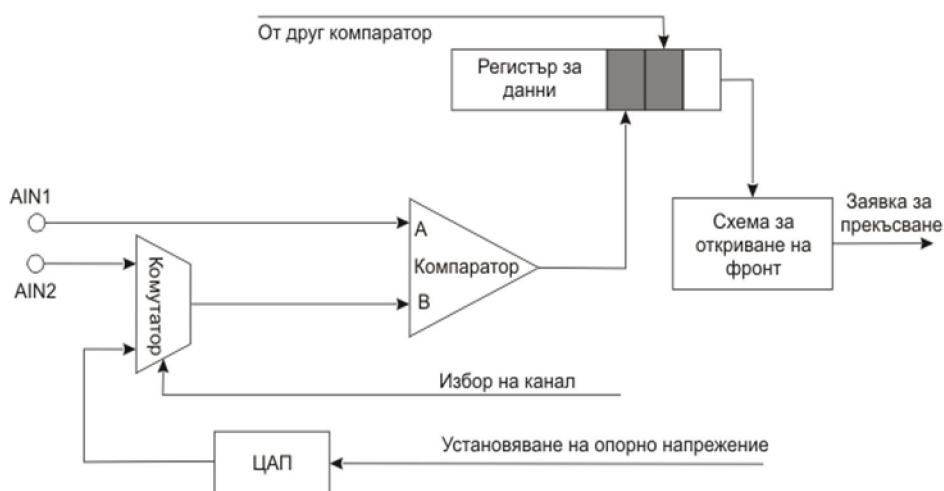
Блокът за управление на ADC конфигурира и синхронизира функционирането на останалите блокове и се управлява програмно чрез специални регистри.

#### д) Аналогов компаратор

Аналоговият компаратор се използва за сравнение на напрежението на два външни аналогови сигнала или за сравнение на външен аналогов сигнал с опорни напрежения, изработвани от процесора (фиг. 20). Резултатът от сравнението се кодира в бит от регистър със специално предназначение – например “1”, когато напрежението на вход А е по-голямо от напрежението на вход В, а “0” – при напрежение на вход А по-малко от това на вход В. В случай, че съотношението се промени, то се изменя значението на бита, а също така може да се установи флаг и да се генерира заявка за прекъсване.

Аналоговият комутатор избира аналоговите сигнали за сравнение. Един от сигналите се избира да е от външен вход (AIN1), а вторият – или да е от външен вход (AIN2), или да е образцово вътрешно напрежение, изработвано от DAC (програмируем генератор на образцово вътрешно напрежение). Компараторът е главният модул и сравнява входните сигнали. В регистърът за данни се съхранява резултата от сравнението. Схемата за откриване на фронт определя промяната на един бит в регистъра за данни (изхода на компаратора) и изработва по това събитие заявка за прекъсване.

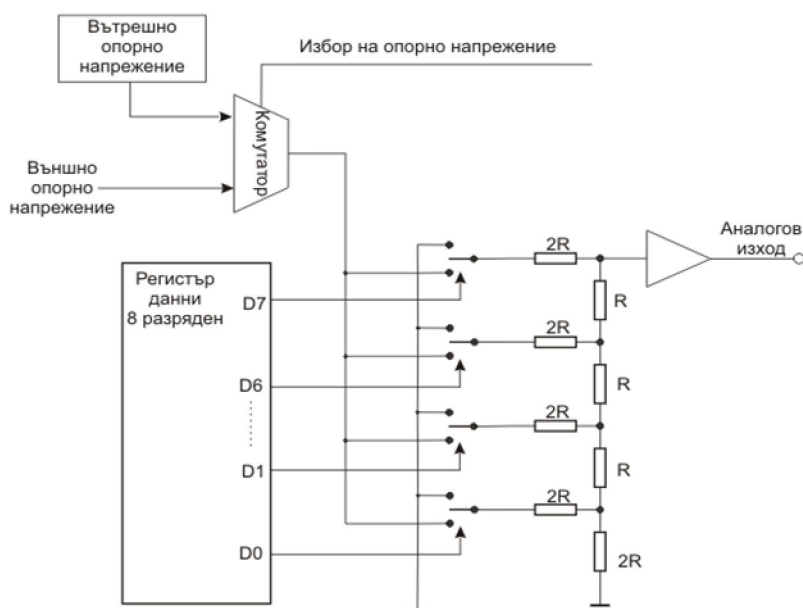
Типичните приложения на компаратора са: контролиране на превишаването на допустимите значения на температура, налягане, ток, напрежение и др. физически величини. Физическата величина се преобразува в напрежение с помощта на датчик и се контролира с помощта на аналоговия компаратор. Прагът на сработване се установява чрез вграден генератор на образцово напрежение. Може да се използва за откриване (формиране) на фронтове на външни сигнали, както и в схеми за следене на захранващото напрежение на системата.



Фиг. 20. Аналогов компаратор.

*е) Цифрово-аналогов преобразувател (DAC)*

Модул DAC е предназначен за генериране на аналогови сигнали с ниво на напрежение, съответстващо на зададен цифров код (фиг. 21). В регистър за данни се записва цифровия код и той определя разредността на DAC.



Фиг. 21. Модул DAC.

Най-разпространеният метод на цифрово-аналогово преобразуване е чрез матрица R-2R. Матрицата работи на принципа на деление на входното напрежение между входовете. Броят на входовете на матрицата е равен на броят разреди в регистъра за данни. На всеки вход чрез ключ може да се подаде опорно напрежение  $V_{ref}$  или 0V. Ключовете се управляват от разредите на регистъра за данни: при лог. 1 на матрицата се подава опорно напрежение  $V_{ref}$ , а при лог. 0 се подава 0V. Комутатор на опорно напрежение  $V_{ref}$  позволява да се избере външен или вграден източник на опорно напрежение.

На практика DAC се използва за управление на различни изпълнителни устройства и системи: постояннотокови електродвигатели с променлива скорост на въртене, източници на захранване с управляемо напрежение и др. С помощта на DAC могат да се синтезират аналогови сигнали с различна форма, напр. синусоидална.

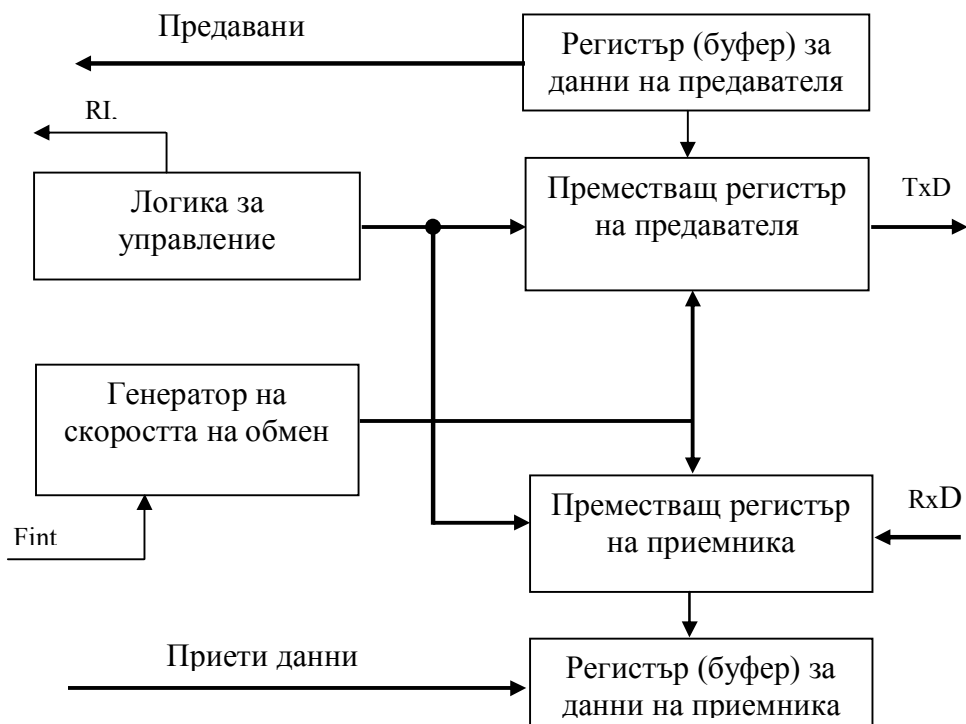
*ж) последователни интерфейси*

В съвременните MCU има един или няколко модула за последователен обмен, като стандартен е универсалния модул USART

(Universal Synchronous/ Asynchronous Receiver and Transmitter). В асинхронен режим поддържа протокол за обмен за RS232C (8N1, 9N1), а в синхронен – нестандартни синхронни протоколи, в някои случаи – протокол SPI (фиг. 22). Вградените последователни интерфейси са предназначени за решаване на следните задачи:

- връзка на микропроцесорната система с управляваща система от по-високо ниво: промишлен или персонален PC, програмируем контролер. Най-често за тази цел се използват интерфейси RS232C, RS422, USB, IrDA;
- връзка с външни по отношение на MP/MCU периферни микросхеми (памети EEPROM, RTC и др.), а също и с датчици с последователен цифров изход. За тази цел най-често се използват SPI, I2C, MicroWire, uLAN и др..
- интерфейс за връзка с локални мрежи в разпределени управляващи системи: използват се RS232C, RS485, I2C, uLAN, CAN, Ethernet.
- вътрешносистемно програмиране на вградената програмна памет (OTPROM, EPROM, FLASH) или памет за данни (EEPROM): използва се RS232C/JTAG/USB (MSP430 (Texas Instruments) или SPI (AVR (Atmel))).

Генераторът на скорост на обмен представлява делител на вътрешната тактова честота Fint с плавен или стъпков (дискретен) програмируем коефициент на делене.



Фиг. 22. Модул UART



При “плавното” програмиране, необходимата скорост може да се настрои без зависимост (в определени граници) от честотата Fint. За това се използва брояч в режим на автопрезареждане. В случай на фиксирани коефициенти на делене, за поддържане на стандартните скорости е необходимо да се избере определена честота на работа на CPU. От изхода на генератора, сигналът за синхронизация постъпва на входа за тактовия сигнал на приемния и предавателния преместващи регистри, които осъществяват последователно приемане/извеждане на данните с определена скорост. Приетият байт постъпва в буфера за данни на приемника, а байтът за предаване се разполага в преместващия регистър на предавателя.

Процесите на приемане и предаване на данни в UART са независими и така се поддържа дуплексен режим на обмен. Необходимо е приемникът и предавателят да са настроени за една и съща скорост на обмен.

В синхронен режим (SPI) всеки приет/предаден бит се стробира със специален сигнал и не е необходимо да се съгласуват скоростите на приемника и предавателя. Функционирането е опростено: приемникът и предавателят работят синхронно и на един приет бит съответства един предаден бит; началото на предаване на байт съвпада с началото на приема на байт; за един сеанс на обмен се приема един байт и се предава един байт.

В повечето случаи приемо-предавателите работят с входни и изходни сигнали с TTL нива. Формиране на физически сигнали с нива на напреженията и тока, съответстващи на реализирания интерфейс, се извършва с помощта на специални схеми – трансивери или адаптери за физически интерфейси: например MAX232 (MAXIM) – RS232C, MAX485 (MAXIM) – RS422/RS485, PCA82C251(Philips) – CAN.

Широко се използват други интерфейси (CAN, USB), но те са значително по-сложни и не са обект на изучаване в стандартните университетски курсове, а в специализирани курсове като “Системи за вход-изход”, “Разпределени системи за управление” и др.

### *з) Организация на прекъсванията в микропроцесорната система*

Източниците на прекъсване в микропроцесорната система могат да бъдат:

- *Външни източници* - заявката за прекъсване се получава при преход от лог.1 в лог.0 (или обратно) на входа или при определено ниво на напрежението (лог.0 или лог.1) на външния вход за прекъсване.

- *Вътрешни източници* – модули памет (обикновено от модул EEPROM) или от периферните устройства: таймери/бройчи – заявката се генерира при препълване; блокове за Input Capture/Output Compare – заявката се получава при IC събитие или равенство при OC събитие; ADC – заявка при завършено преобразуване; Аналогови компаратори

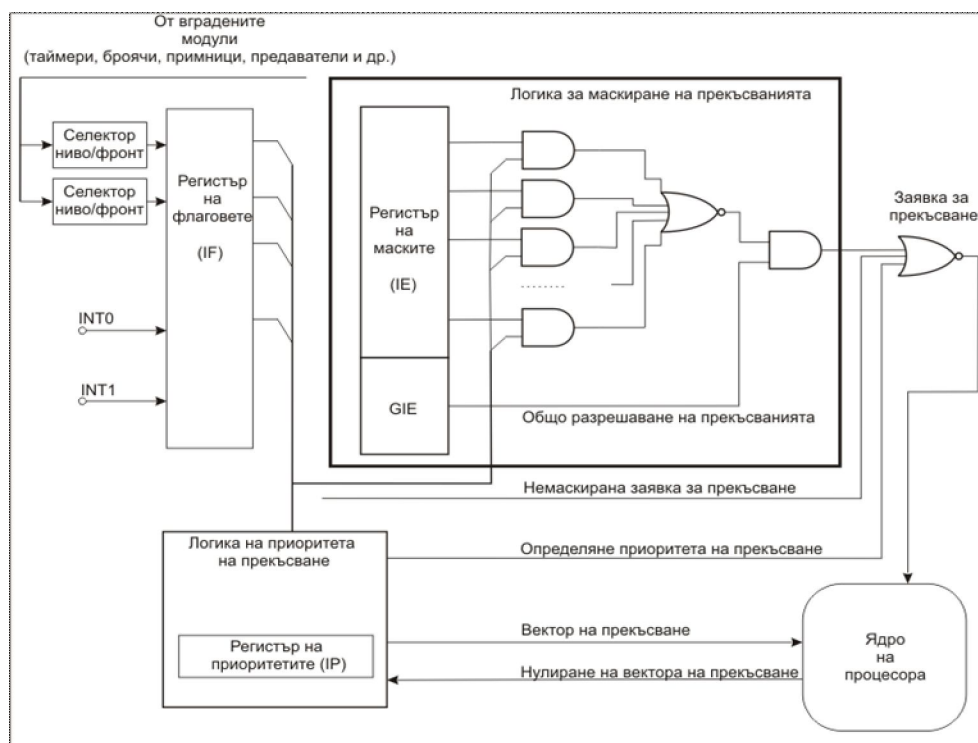
– заявка при изменение на съотношението на нивата на входните сигнали; последователни приемо-предавателни интерфейси – заявка се генерира при приемане на байт или пакет и достъпност на новоприетите данни или при завършено предаване на байт или пакет и освобождаване на предавателя;

*- Програмни прекъсвания*

В различните фамилии MCU са реализирани различни механизми за обработка на прекъсванията:

- 1) чрез вектори на прекъсванията с твърд приоритет (ST7, AVR, Am186);
- 2) чрез вектори на прекъсванията с програмируем приоритет (MCS-51);
- 3) чрез динамична таблица на векторите (M16C);
- 4) чрез общ вектор (механизъм на полинг) (PIC).

В блока за обработка на прекъсванията (фиг. 23), селекторът на ниво на външен сигнал за прекъсване избира събитие, по което генерира заявка за прекъсване при преход от лог.1 в лог. 0, при преход от лог. 0 в лог.1, при лог.1 или при лог. 0.



Фиг. 23. Обобщена схема на блок за прекъсване

При възникване на заявка на прекъсване (ЗП), в регистърът на флаговете на прекъсване IF (Interrupt Flag), се установява флаг, съответстващ на източника на прекъсване. Логиката за маскиране на прекъсванията разрешава или забранява генерирането на заявки от

определени източници или от всички източници едновременно. За разрешаване на прекъсванията, е необходимо да се установи лог. 1 в съответния бит на регистъра на маските IM (Interrupt Mask) и в бит GIE за общо разрешение на прекъсванията (GIE - Global Interrupt Enable). Логиката за маскиране на прекъсванията не влияе на немаскируемите прекъсвания. Логиката за приоритета на прекъсванията изработва вектор за най-приоритетното прекъсване и го предава към CPU синхронно със сигнала за прекъсване; следи приоритета на ЗП, които се обслужват и прекъсва тяхното обработване, ако е постъпила ЗП с по-висок приоритет. Приоритетът на всяко прекъсване може да се задава твърдо (системи с твърдо установени приоритети) или да се програмира чрез регистър на приоритетите IP (Interrupt Priority). Векторът на прекъсване се извлича по номера на ЗП от таблица с векторите на прекъсванията от паметта.

#### *и) подсистема за синхронизация*

Подсистемата за синхронизация (фиг. 24) формира устойчиви сигнали за синхронизация на вътрешните блокове на MCU и външните блокове на управляващите системи, реализирани с дадения MCU. Тя включва генератори на синхроимпулси (основен и спомагателен генератор, схема за формиране на изходния синхронизиращ сигнал), формирова̀тел на вътрешните синхросигнали на процесора (комутатори K1 и K2, делители, схема за начално установяване).

Основеният тактов генератор (ТГ) генерира сигнали за синхронизация за CPU, за повечето периферни устройства и за схемата за начално установяване. В състава на блока се включва основния ТГ, външна или вградена времезадаваща верига за ТГ и (не винаги) схема на умножител на честотата. Основната честота на синхронизация може да се изменя в широки граници – от десетки kHz до стотици MHz. Честота до 1 MHz се използва в системи с намалена консумация. В зависимост от избраната работна честота, изискванията за точност и стабилност на параметрите на сигнала за синхронизация, могат да се използват различни времезадаващи вериги, които от своя страна изискват пренастройка на режимите на работа на генератора. Изборът на режим на работа на генератора и типа на времезадаващата верига, се осъществява програмно от специални конфигурационни битове от вградената програмна памет на MCU. Времезадаващите вериги се включват към специални изводи на MCU (в някои MCU има вградени вериги). Най-често използваните вериги са:

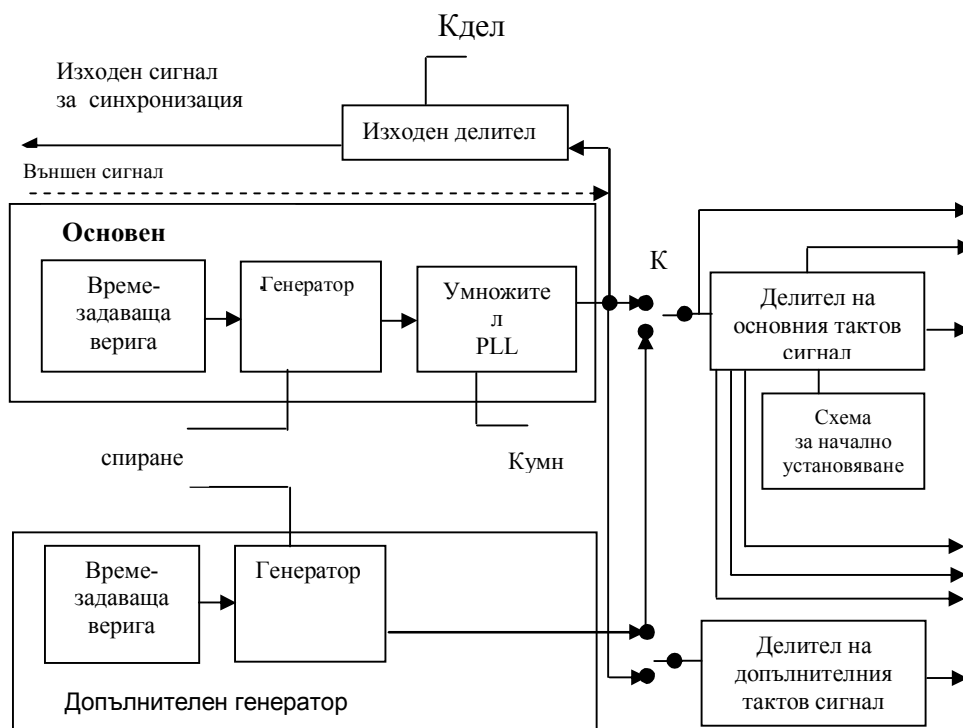
- Кварцов резонатор: честоти - от десетки kHz до десетки MHz, висока стабилност на честотата (грешка – стотици/ хиляди части от процента), относително висока цена.

- Пиезокерамичен резонатор: честоти - от десетки kHz до единици MHz, средна стабилност на честотата (грешка – десетки части от процента), ниска цена.

- LC-верига: честоти - от единици до стотици kHz, средна стабилност на честотата (грешка – десетки части/ единици от процента), ниска цена.

- RC-верига: честоти - от единици до стотици kHz, ниска стабилност на честотата (грешка – единици от процента), често се реализират като вградена времезадаваща верига.

Почти във всички 16/32-разредни MCU се използват вградени цифрови умножители на честотата с програмируем коефициент на умножение. Най-често използван умножител е схема на честотен синтезатор с фазово автоматично поддържане на честотата (PLL). От основния ТГ, синхросигналът се извежда на извод на MCU и може да се използва за тактуване на външни схеми. За намаляване на честотата на изходния сигнал в тази верига, може да се използва управляем или фиксиран делител. Допълнителният ТГ обезпечава тактуването на някои ПУ, обикновено на таймери/бройчи, но в някои режими може да бъде основен ТГ. Обикновено работи с честоти до 1 MHz и се използва като база за RTC – на честота 32 768 kHz. Структурата му е аналогична на основния ТГ, но почти никога не използва умножител на честота. Делителите на честотата на основния и допълнителен ТГ формират вътрешните сигнали за тактуване на ядрото и за начално установяване ssync и сигналите за синхронизация на ПУ – psync.

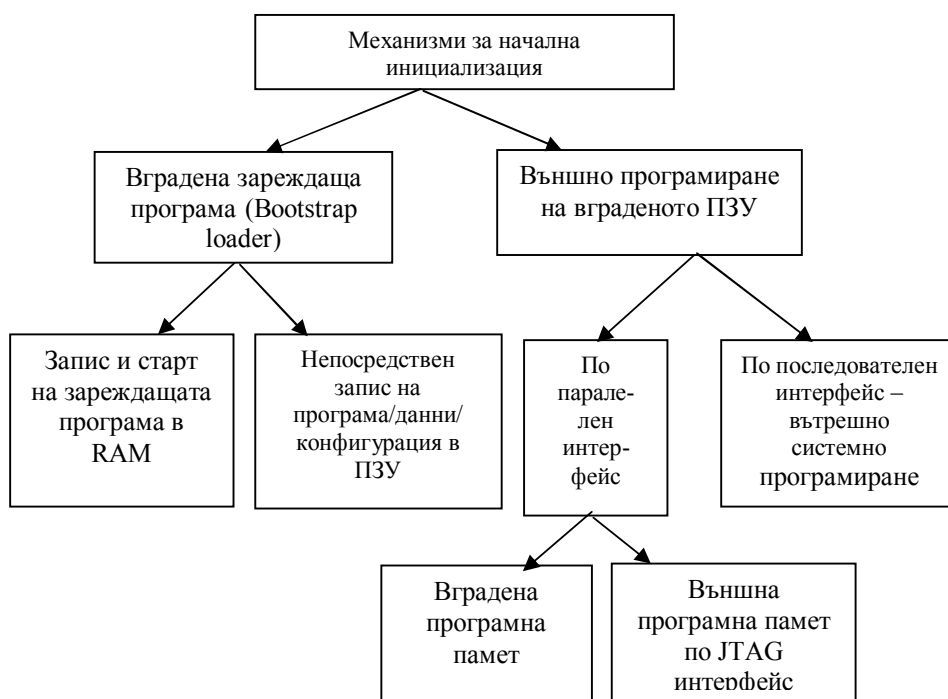


Фиг . 24. Подсистема за синхронизация

Всички вътрешни сигнали се получават чрез разделяне на честотата на фиксиран или програмируем коефициент на делене. Комутаторите на синхросигналите K1 и K2 се използват за избор на източник на тактов сигнал за ПУ: основен или допълнителен ТГ. В нормален режим повечето ПУ се синхронизират от основния ТГ, а допълнителният ТГ се използва като база за таймерите, RTC или Watchdog таймера. В случай на нестабилна работа на основния ТГ или при необходимост от преминаване на по-ниски честоти на работа, например в режим на намалена консумация, може входа на делителя на основния ТГ да се включи към изхода на допълнителния ТГ.

*й) механизъм за начална инициализация на вградената памет*

Началната инициализация (първоначалното зареждане) обезпечава запис на програмния код, данните или конфигурационните параметри във вградената енергонезависима памет на MCU (фиг. 25). Процесът на начална инициализация предполага работа само с микропроцесорната система, без помощта на системна програма (за зареждане). Зарежданият код е или самата приложна програма, или се зарежда от високо ниво. Данните обикновено са началните значения на работните параметри.



Фиг. 25. Механизми за инициализация на вградената памет

Конфигурационните параметри настройват режимите на работа на MCU и към тях се отнасят:

- типа на генератора (кварцов, с пиезокерамичен резонатор, LC или RC);
- използваната система за изключване при отпадане на захранването, автоматично включване при подаване на захранване (Power On Reset);
- използване на стражеви таймер (WDT);
- флагове за защита на вградената памет от несанкциониран достъп;
- използване и разредност на шината на вътрешната памет;

стартов адрес на програмата (Reset вектор) и др.;

Вградената програма за зареждане (Bootstrap Loader) е специална програма, записана при производството на MCU в специален блок на вградената памет (ПЗУ). При изпълнението си, тя приема записваните програми или данни чрез последователния порт (обикновено UART) и ги записва в паметта на MCU. Възможни са няколко варианта на съхраняване на зарежданата програма:

1) Програмата се зарежда в RAM паметта и веднага след това и се предава управлението. Това трябва да е зареждаща програма, която от своя страна приема и записва във вградено или външно ПЗУ (обикновено FLASH памет) приложната програма. След рестартиране, управлението се предава на приложната програма.

2) Програмата се записва непосредствено във вграденото ПЗУ и започва да се изпълнява веднага след преминаване в нормален режим (MSP430).

Преход в режим bootstrap loader обикновено се изпълнява чрез подаване на специален код на конфигурационните изводи с едновременно рестартиране на MCU.

#### *к) програмиране на паметта*

- Външно програмиране на вграденото ПЗУ.

В даденият режим, процесорът с вградено ПЗУ (ОТР, EEPROM, FLASH) се разглежда като обикновена схема ПЗУ с последователен или паралелен интерфейс. Ядрото на MCU се изключва, но при някои фамилии, например AVR (Atmel) се изисква да функционира генераторът на синхроимпулси. Превключването на MCU в режим на външно програмиране, обикновено се осъществява чрез подаване на програмиращо напрежение с ниво 12V на специален извод или чрез подаване на специален код на конфигурационните изводи с едновременно рестартиране на MCU (аналогично като прехода в режим bootstrap loader). В зависимост от типа на интерфейса, има два варианта за външно програмиране:

- Програмиране по паралелния интерфейс (поддържа се от фамилии AVR, MCS-51, Z8). MCU се програмира извън целевата система, в специални програматори. Входно-изходните портове и/или

сигналите на външните шини, се използват като шини за адреси, данни, управление на ПЗУ.

Предимства са простите алгоритми на програмиране и високата степен на защита от случайно препрограмиране в системата;

Недостатъци: необходимо е да се използва куплунг за MCU или еднократно да се програмира преди монтажа върху платката, което не позволява да се обновява програмното осигуряване на системата.

- Програмиране по серийния интерфейс (AVR, PICmicro). В този режим адресите, данните и инструкциите за достъп до ПЗУ (запис, четене, проверка и др.) се предават по специален или стандартен последователен интерфейс.

Предимства: неоглямото количество сигнали позволяват да се включи програматорът към схемата и да я програмира без да се демонтира от платката. Този режим е известен като режим на вътрешносистемно програмиране ISP (In System Programming); неограничен брой инструкции, които се предават в последователен код и увеличават функционалните възможности на програматора.

Недостатъци: относително сложен протокол на програмиране.

- Вътрешно-системно програмиране по JTAG интерфейс (MSP430).

В днешно време, в режим на вътрешно-системно програмиране, се използва серийния интерфейс JTAG - интерфейс за гранично сканиране, който позволява на изводите на MCU да се установят сигнали с определени значения и отчита сигналите, установени от външни схеми или от вътрешните подсистеми на MCU. JTAG интерфейсът представлява четири-битов тестващ интерфейс, изграден според стандарта IEEE 1149.1, който определя състава и организацията на вградената гранично сканираща логика BSL (Boundary Scan Logic). С малки изключения, тя е идентична за различните схеми и в основата и са граничните сканиращи клетки BCS (Boundary Scan Cell), представляващи гъвкави тестови точки, вградени във всеки вход и изход на схемата. BSC може да се използват за управление и контролиране на състоянието на съответния извод, осигурявайки информация за него през серийния си вход TDI и през серийния си изход TDO.

Всяка BSL има четири допълнителни извода – TDI, TDO, TMS, TCK и опционален TRST. TDI и TDO (Test Data Input/Output) обслужват пътя, по който последователната информация се извежда в/от схемата. TCK е тестов тактов сигнал, а TMS определя режима на работа - тестов или нормален. TCK и TMS управляват контролера за тестови достъп TAP, който е организиран като машина на състоянието и контролира серийното сканиране на информацията за инструкцията или данните през схемата и тяхното съвместяване с JTAG интерфейса.

### **III. 3. Интегрирани развойни програмни среди (IDE – Integrated Development Environment)**

Най-ефективният начин за разработване на управляващи програми за MCU е чрез използването на специализирани, професионални средства за настройка, които се различават от симулаторите и вътрешносхемните емулятори. Съществуващите Windows-базирани интегрирани програмни среди IDE предлагат широк спектър от средства за разработване на софтуер за всеки нов микроконтролер – асемблери, C-компилатори, симулатори, развойни системи, вградени симулатори, емулятори, програматори, дебъгери, приложни примерни програми. С тях може да се създава сорс-код, да се симулират операции, да се зареждат и изпълняват програми, да настройват и тестват до определена точка приложенията за различни MCU от дадена фамилия. Микроконтролери с Flash, OTP и EPROM памет могат да се програмират директно от средата IDE, използвайки вграден програматор или програмируем адаптер [50].

Потребителската програма, обикновено на асемблер и/или на език от високо ниво C/EC++ (други езици от високо ниво практически не се използват), се въвежда чрез вграден текстов редактор. В някои случаи, за подобряване на бързодействието, се предпочита да се редактира полученият след компилация на C-програма асемблерски листинг [22, 27].

На следващия етап, разработеното програмно осигуряване се тества върху системата за развитие с помощта на софтуерен дебъгер и/или върху друга система (може и върху разработваната) с помощта на емулятор. След отстраняване на грешките от получения изпълним код, се формира изпълним код, предназначен за зареждане в програмната памет.

IDE се стартира от PC и съдържа всички компоненти нужни за разработване на вградените системни приложения. Съвременните интегрирани програмни среди IDE са универсални и включват:

- Текстов редактор за разработване на програмния код;
- Асемблер и свързващ редактор за генериране на изпълним машинен код;
- Симулатор, позволяващ симулиране на програмата от PC или вградена симулация (с реални I/O въздействия);
- Менажер на проектите;
- Настройваща програма (DBG - debugger), позволяваща тестване на програмата;
- Програматор за програмиране на целевия MCU;

Асемблерът транслира сорс-кода в обектен код, чрез замяна на всяка асемблерска инструкция от програмата в съответен машинен код (фиг. 1.12). Обектният код не може да бъде изпълнен от MCU. Той се нуждае от свързваща програма – линкер (Linker), който свързва



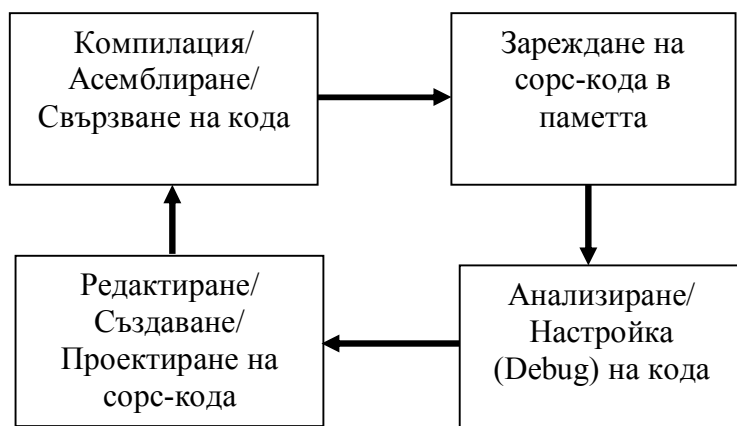
обектния файл, генериран от Асемблера, със стандартни библиотечни модули за генериране на изпълнимия файл. Може Асемблерът и Линкерът да са комбинирани в една програма (например PIC Assembler).

Процесът на създаване на приложение често се описва като цикъл на развитие и включва всички етапи от проектиране до приложение (фиг. 26). След като програмния код е написан, той се тества и след това се коригира с цел да произведе правилно изпълнение на програмата. В IDE, всички функции са интегрирани и това дава възможност за бързо и ефективно ориентиране в предлагания теоретичен материал и гъвкаво управление на инструменталните средства.

Използваните езикови средства са програми като cross-асемблерите и cross-компилаторите, които се различават от типичните компилатори с това, че се стартират на PC, но произвеждат код за MCU, който използва различен набор инструкции от PC. Също така те генерират “debug” файл, който в развойна среда позволява изпълнението на кода да се тества чрез настройваща програма (DBG) – включване на точки на прекъсване, “watch”-прозорци за визуализиране на променливи, стъпково изпълнение на кода и наблюдаване на изпълнението и др.

Вградените системни езикови средства се различават от компилаторите и по оптимизирания код. Големината на простите вградени системни програми може да бъде по-малка от kB, а средните по големина системи се нуждаят от 32kB или 64kB при сравнително сложни функции.

Настройването на програмата може да се осъществи чрез програма-симулатор. Симулаторите са част от IDE, затова програмата може да бъде тествана без допълнителен хардуер. Симулаторът изпълнява програмата по-бавно от MCU. Друг начин за настройка е чрез хардуерен DBG.



Фиг. 26. Процес на създаване на упр. програма

Хардуерните DBG могат да бъдат емулатори, които използват специализиран хардуер на мястото на MCU или могат да бъдат DBG на входна електрическа верига, които използват MCU, притежаващи “debug”- опции. Хардуерният DBG, като симулатора, позволява на обучаемия да разгледа променливите на различни места в кода, да поставя точки на прекъсване, да наблюдава изпълнение на инструкции за реално взаимодействие със специализирана електрическа верига [18].

### **III. 4. Съвременни микроконтролери.**

Към дадена фамилия принадлежат микроконтролери с еднакво ядро, под което се разбира съвкупността от системата инструкции, времедиagramата на работата на CPU, организацията на паметта, системата за прекъсване и базовите ПУ. Разликата, между отделните членове на фамилията, обикновено се състои в състава на периферните устройства и обема на паметта за данни/програма. При много фамилии броят на модификациите MCU е над 100. Най-важната особеност на една фамилия MCU е програмната съвместимост на ниво двоичен код между всичките и членове. Това позволява да се замени един MCU с друг от същата фамилия без загуба на програмно обезпечаване.

#### **III. 4. 1. Микроконтролери Intel 8051.**

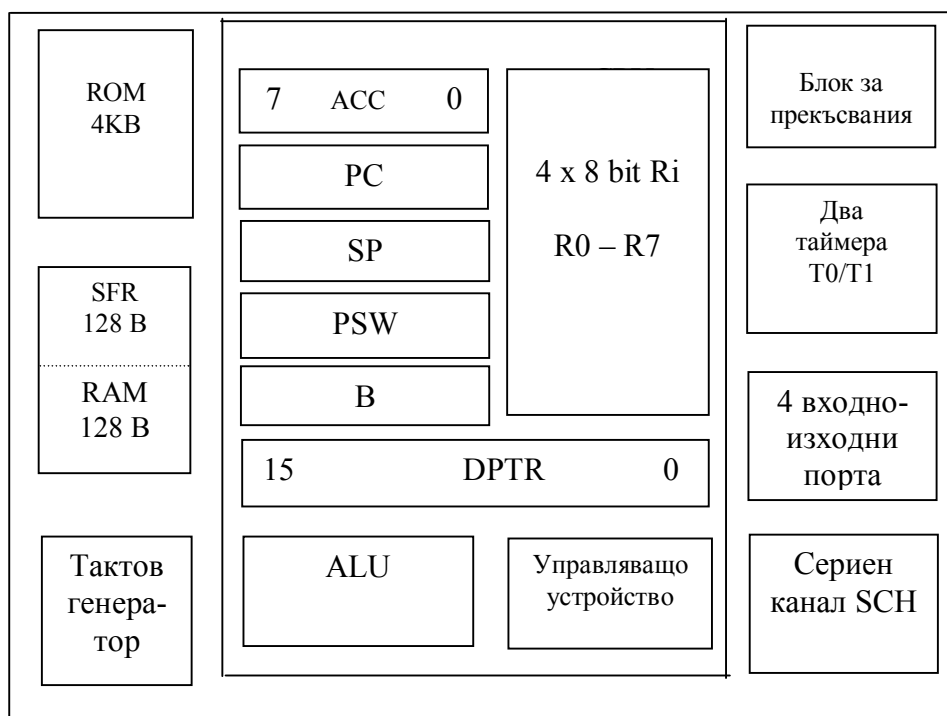
Най-разпространените MCU са Intel 8051 [5], които притежават Харвардска архитектура (фиг. 27). Фамилията 8051 се произвежда от много фирми в множество варианти с различни вградени подсистеми и типове памет. Основните производители на разновидностите на 8051 са фирмите: Philips, Siemens, Intel, Atmel, Dallas, Temic, Oki, AMD, MHS, Gold Star, Winbond, Silicon Systems и др. Някои от моделите съдържат 128, 256, 512 и 1024 SRAM; 4, 8, 16 и 32 kB PROM, EPROM или EEPROM, а други нямат програмна памет и използват външна памет.

Базовият модел 8051 е един от най-изполваните MCU в света. Има осем вида адресации и изпълнява 111 инструкции (CISC ядро). Притежава ALU; четири банки регистри, по осем във всяка, програмна памет с обем 64kB, 256B RAM; 4kB ROM; булев процесор; два 16-битови таймера; сериен канал (UART); четири 8-битови входно-изходни (I/O) порта, два от които се използват като шина за адреси/данни за достъп до външна памет (64kB) за данни и програма; вграден тактов генератор с максимална честота - до 12MHz за по-старите и до 33MHz за по-новите серии. Стандартната периферия (стандарт 8052) включва таймери T0 и T1, сериен интерфейс (SCI), паралелни портове и система за прекъсване. Всички MCU имат 16-битови програмируеми таймери, а някои от тях притежават блок за генериране на PWM сигнали. Те могат да се използват за формиране на точни времеви интервали, за броене на импулси, за формиране на

последователности от импулси, тактуване на серийния канал и др. От тях се генерират заявки за прекъсване към MCU. При някои модификации броят на таймерите достига 32 [44].

В MCU 8051 е реализиран универсален асинхронен приемо-предавател (UART) с TTL изходи и 4 режима на обмен. Поддържа протокола на стандарта RS-232C, като освен него, в съвременните MCU се вграждат RS-485, I2C, SPI, Bitbus, CAN, USB, GSC и др.

Стандартните MCU имат от 8 до 32 входно-изходни линии, които са индивидуално програмируеми като входове или изходи. Важна особеност на паралелните портове е, че могат да се програмират за различни функции. Например, изводите на портове P0 и P2 могат да се използват като обикновени изводи за вход/изход или за като шина за данни/адреси при свързване на външни устройства - допълнителна памет за данни/програма, устройства за вход/изход и др. Порт P3 може да се използва или с общо предназначение или за специални функции, обслужващи работата на последователния интерфейс, таймерите, системата за прекъсване и др.



Фиг. 27. Структура на MCU 8051.

MCU 8051 има 5 източника на прекъсване: две външни прекъсвания (INT0, INT1), две прекъсвания от таймерите/броячите (TF0, TF1) и едно прекъсване от серийния канал (SCH), който генерира прекъсване при приемане или предаване. Системата за прекъсване може да се разглежда като съставена от три части: блок за

получаване на заявки за прекъсване, блок за разрешаване на прекъсванията (общо и индивидуално) и блок за приоритет на прекъсванията [11, 12].

Масово се произвеждат MCU, които освен базовата 8051 архитектура, включват и много други функционални възможности. Фамилия 8xC552 (Philips). MCU 83C552 е с 80C51 ядро, 8kB ROM, 256B RAM, външна памет за данни/програма до 64kB, допълнителен 16-битов таймер/брояч с четири IC регистъра и три OC регистъра, два стандартни 16-битови таймера, 10-разреден ADC с 8 мултиплексирани аналогови канала, два канала за PWM, пет 8-битови I/O порта и един 8-битов входен порт, който се използва и за аналогови сигнали, I<sup>2</sup>C интерфейс, UART - пълен дуплекс, съвместим със стандартния 80C51, вграден watchdog таймер (WDT). Някои MCU имат 8-битов ADC (8xC51GB на Intel MCS-51, TSC8xC251A1 на Temic) или 10-битов (83C552) ADC.

Фамилия MCU 80C51XA (Philips Semiconductors) представлява разширение на фамилия 8051(XA). Основните характеристики са: 16-разредно ALU, 16MB памет за данни/програма, осем 16-разредни регистъра за аритметико-логически операции, разширен набор от инструкции, апаратно поддържане на многозадачност. Максимална тактова честота – 30 MHz. Вградени периферни устройства: три таймера, Watchdog таймер, PCA, интерфейси – UART, CAN, I2C, UPI, DMA, PWM модул, DRAM – контролер. MCU P51XASS3x има 80 линии и 8-канален 10-разреден ADC.

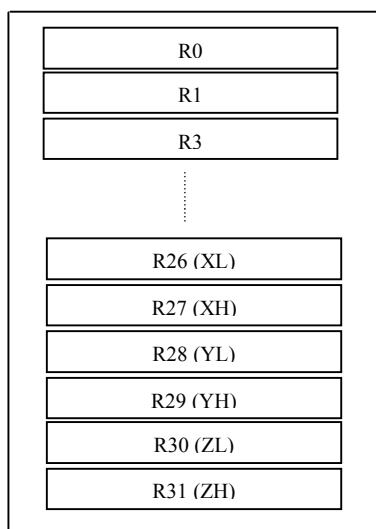
Фамилия MCU DS5000 (Dallas) са с ядро 8051 и NVRAM 128 kB, чиято енергонезависимост се обезпечава от литиева батерия в MCU. Серия DS8xCx са с повишено бързодействие – типичният машинен цикъл е съкратен от 12 на 4 такта. Допълнителна периферия: два регистъра DPTR, два модула UART и RTC таймер.

Максималната тактова честота на MCU DS80C323WDT е 20 MHz, а на останалите - 33 MHz. Всички модификации имат вградени три таймера/брояча, WDT, UART. Броят на изводите на DS87C550 е 55, а на всички други – 32. MCU DS87C550 има вграден 8-канален 10-разреден ADC. DS89C420/30/40/50 притежават до 64kB вградена Flash-памет, а DS80C400 има 24-разредна адресна шина, чрез която директно се адресира 16MB програмна памет и 16MB памет за данни.

Фамилия MCU LPC900 (Philips) са 8-битови MCU с Flash-памет, шест пъти по-бързодействащи от MCU 80C51 (167 ns за инструкция при 12 MHz). Всеки MCU притежава RTC и три 16-битови таймера/брояча, а някои имат I2C, UART, SPI – интерфейси, два 8-битови ADC (LPC935). Моделите LPC904, LPC915/916/917, LPC924/925 и LPC933/934 имат бързодействащи ADC и DAC.

### III. 4. 2. Микроконтролери AVR

Представяват 8-битови RISC MCU с подобрена Харвардска архитектура, изпълняващи 133 инструкции с формат 2B. Бързодействие - 20 MIPS при честота 20MHz.



Фиг. 28. Регистров файл на MCU AVR

Притежават програмна Flash-памет, допускаща до 1000 записа, а някои имат в състава си EEPROM памет с възможност за 100 000 цикъла за изтриване/запис на информация. Програмирането им е вградено (по SPI интерфейс). AVR функционират в широк диапазон на захранващото напрежение - от 1,8 до 6,0 V.

Аритметико – логичното устройство на MCU AVR е реализирано с регистров файл (фиг. 28), състоящ се от 32 регистъра с общо предназначение за съхранение на операндите и резултатите. В сравнение с традиционната “акумулаторна” архитектура, това спомага за получаване на по-компактен код, защото намалява прехвърлянето на данни между акумулатора и работните регистри.

Шест от регистрите могат да се използват като три 16-битови указателя на адреси при косвена адресация на данните. Един от тях (Z-указател) се използва също така за достъп до данни, записани в програмната памет на MCU. Почти всички MCU имат аналогов компаратор или 10-разреден ADC, последователни интерфейси, PWM модул, 8/16 – разредни таймери, а някои нови модели – хардуерен умножител [43].

### III. 4. 3. Микроконтролери на фирма Motorola/Freescale

#### а) 8-разредни микроконтролери

Фамилия 68HC08/68HCS08 се различават от останалите 8-разредни MCU (MCS-51, PIC, AVR) с широкия набор инструкции и

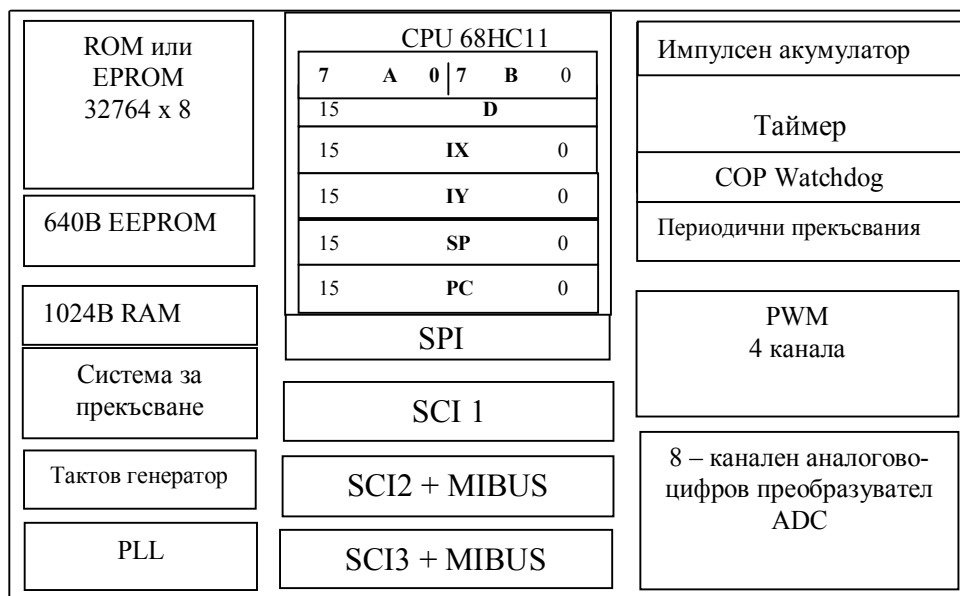
начини на адресация. Средното време за изпълнение на инструкция е 2 - 3 такта (при 8 MHz производителността е  $\sim 3$  MIPS). Апаратно е реализиран контрол на захранващото напрежение и работната честота чрез LVI модул. Всички модели съдържат WDT таймер. Моделите 68HCS08 се произвеждат в 18 разновидности, като е съхранена архитектурата и основните модули на MCU 68HC08/908, а се усъвършенстват различни параметри. Максималната честота на серии GT,GB е 20MHz. За реализация на вътрешно-схемна емуляция, е въведен режим на настройка BDM. В много от моделите са вградени интерфейси SCI и SPI, а при серии GT,GB – I2C интерфейс, 8-канален 10-разреден ADC; при серия RC – аналогов компаратор. Микроконтролерите 68HC08/908 имат модулна структура, обединяваща процесор CPU08, вътрешна памет - OTP ROM от 128B до 2kB, MPROM – до 32 kB, Flash – 1,5kB до 62kB, EEPROM – 512B до 1kB, RAM памет – 1, 2 или 4 kB, различни ПУ. Използваната Flash-памет допуска не по-малко от 10 000 цикъла на изтриване/програмиране. Захранващото напрежение на повечето модели е 3V/5V, а тактовата честота - 8MHz. [21, 49]

Едни от най-използваните за обучение микроконтролери - MC68HC11, са разработени в повече от 60 модификации с ADC, многофункционални таймери, няколко типа последователни интерфейси (SPI, SCI + MIBUS), EEPROM, MPROM или EPROM от 12 до 32 kB, RAM памет - от 192 до 1024B. Всички MCU имат възможност да работят, както само с ресурсите на чипа – режим на еднокипов MCU (Single Chip Mode), така и в разширен режим (Expanded Mode), с възможност за добавяне на външна памет. Има модели с вграден копроцесор за 16-битово умножение/деление, което увеличава бързодействието им над 10 пъти. Работна честота – до 4MHz. Програмният модел на 68HC11 се състои от два 8-битови акумулатора A и B, които се третират от някои инструкции като един 16-битов акумулатор D, два 16-битови индексни регистъра IX, IY, 16-битов стеков указател, 16-битов програмен брояч и 8-битов регистър на условията [36, 37, 45].

MCU 68HC11P2 има 32 kB ROM, 1kB RAM и 640B EEPROM, 8-канален ADC, таймерна система, включваща импулсен акумулатор, WDT, блок за периодични прекъсвания, SPI интерфейс, SCI модул, състоящ се от три самостоятелни подсистеми SC1, SC2, SC3 и два модула MIBUS. Модул MIBUS представлява поддържащ комуникационен елемент, осигуряващ връзка на едно управляващо устройство с 8 подчинени. Този модул осигурява висока степен на потискане на шумове, подходящ е при работа със средни честоти (20 kHz) и изисква само един проводник за свързване към подчиненото устройство.

Управляващият MCU може да активира модула MIBUS по всяко време с помощта на стартов бит. Чрез него MCU изпраща данни

със специален формат (PUSH сигнали) до подчинените устройства. Подчиненото устройство, приело сигналите, може да връща данни в MCU чрез втори сигнал (PULL сигнал). Модул MIBUS използва същите изводи, използвани от подсистемите SCI2, SCI3. Мониторна система в чипа проверява изпращаните от извод RxD данни и ги сравнява с тези от регистъра. Работата на MIBUS се управлява от регистрите на модул SCI (фиг. 29).



Фиг. 29. Структура на MCU 68HC11P2.

#### б) 16-разредни микроконтролери

Микроконтролери 68HC12 се произвеждат в два основни модела: MCU 68HC812A4 (112 извода), който е създаден за по-големи системи, работещи в разширен режим и 68HC912B32 (80 извода), предназначен за едночипови приложения. Двата MCU са базирани на 16-битово ядро 68HC12, програмно съвместимо с 68HC11, като значително е разширена системата инструкции. Работят с тактова честота 8 MHz в целия обхват на захранващото напрежение 2,7 – 5,5V. Структурата на тези MCU включва CPU HC12, системен интеграционен модул (SIM), 8-битов 8-канален ADC, разширен сериен интерфейс, няколко таймера.

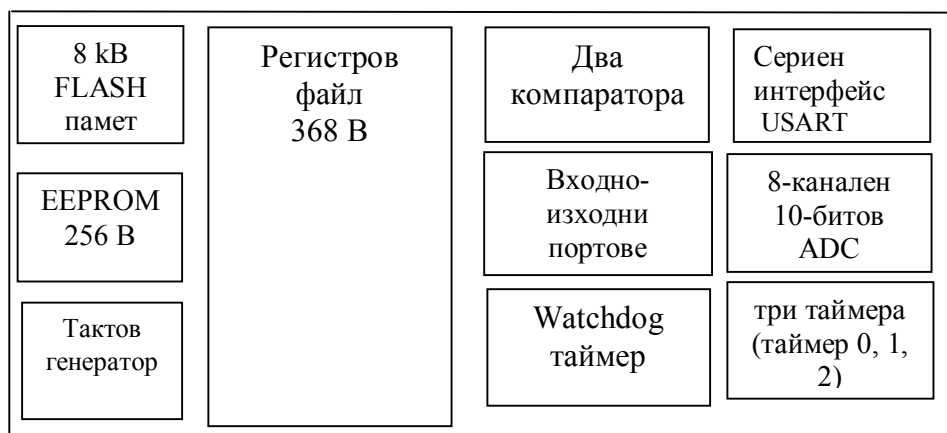
Предимства на CPU HC12: Сравнението между CPU HC11 и CPU HC12 показва, че имат еднакъв програмен модел и организация на прекъсванията. Програмният код на 68HC11 се компилира от асемблерите на HC12. HC12 е 16-битов процесор с ALU, което за някои операции е 20-битов блок. Всички шини за данни са 16-разредни, а външната шина е 16/8-битова. Опашката от инструкции, подобна на конвейър, позволява съхраняване на най-малко 2B от обектния код. Повечето инструкции се изпълняват за един машинен

цикъл без закъсненията, породени от извличане на програмна информация. HC12 има 7 вида индексна адресация спрямо един индексен режим при HC11, който позволява да се използва регистър X или Y с 8-битово отместване. При HC12 регистрите X, Y, SP и PC се използват като базови индексни регистри с 5-битово отместване със знак, с 9-битово отместване със знак и 16-битово отместване. При CPU12 има нова форма на адресиране – авто пре/пост инкрементиране/декрементиране от –8 до +8. При другите процесори, инкрементиращата/декрементиращата стойност зависи от дължината на операнда в инструкцията. Пр.: LDA ,X++ ще увеличи индекса с 1, защото операндът е тип B, а инструкция LDX ,Y++ ще увеличи индекса с 2, защото операндът е тип W [49].

### III. 4. 4. Микроконтролери на Microchip

MCU PIC имат RISC архитектура и работят в широк честотен обхват – от 32kHz до 40 MHz. Всички инструкции се изпълняват за един машинен цикъл и при 20 MHz производителността е 5 MIPS. Произвеждат се с вградена OTP, MPROM, EPROM и Flash памет.

*Фамилия PIC16(C/F)XXX.* Това е най-често използваната фамилия на Microchip. Тези MCU се предлагат в 18 - 68 изведен корпус, имат от 1kB до 4kB програмна памет (ROM/EPROM/FLASH) и от 36B до 128B RAM памет. Инструкциите, които поддържат са 14 – битови и са 35 на брой. Вградени ПУ: таймери/бройчи, 8-битов ADC, последователни интерфейси (I2C, SPI, USART), LCD контролер, паралелен порт, програмируем източник на опорно напрежение, WDT. Имат възможност да работят с ниско захранващо напрежение (2V). Най-популярният MCU от тази фамилия, използван за обучение е PIC16F84. Притежава FLASH памет, която директно се изтрива/програмира от PC чрез ICSP протокол.



Фиг. 30. Структура на MCU PIC16F77



PIC16F877 притежава високопроизводително RISC ядро и използва 35 инструкции. Оптималната му работна честота е 20 MHz, като при нея една инструкция (един цикъл) се изпълнява за 200 ns. Програмната му памет е 8 kB FLASH, а RAM паметта - 368 B. MCU притежава вградена EEPROM памет 256 B. Входно-изходните портове са 5 на брой (A, B, C, D, E), има 14 източника на прекъсване, 3 таймера, 2 компаратора, 2 серийни интерфейса (SCI, SPI), 8 канален 10 - битов ADC (фиг. 30).

Предимства: ниска цена, свободно разпространявана интегрирана програмна среда MPLAB, множество нискоструващи програматори, възможност за избор между много модели PIC с различни периферни модули и памет. Портовете с общо предназначение издържат ток 20mA и могат да управляват директно светодиоди.

Недостатъци: няма програмна съвместимост между различните PIC MCU.

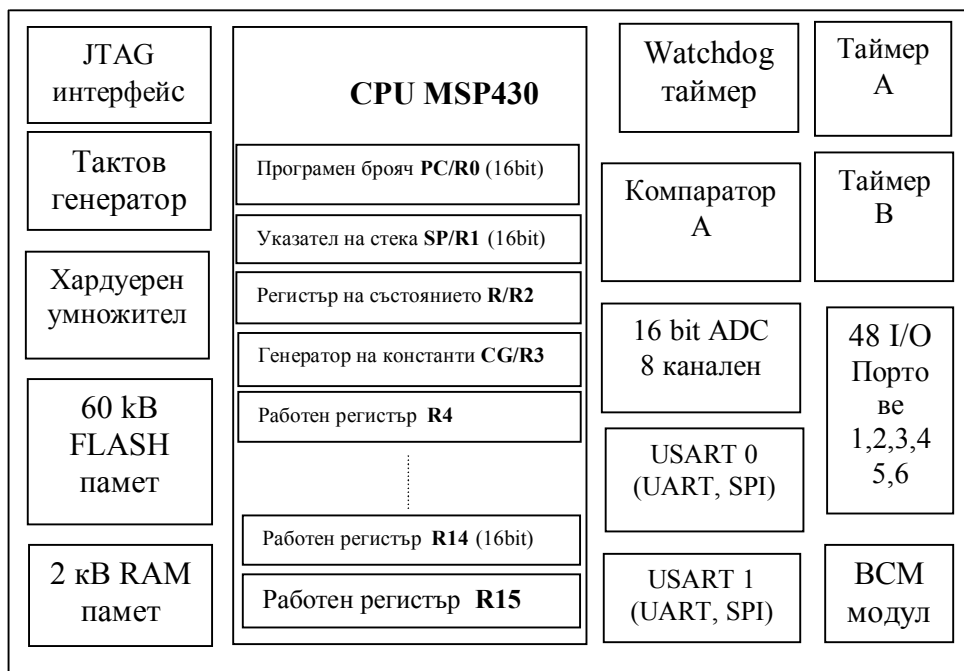
### III. 4. 5. Микроконтролери MSP430x на Texas Instruments

Фамилията MSP430 на Texas Instruments включва 16-битови микроконтролери със свръхниска консумация, проектирани за батерийно захранване и работа с честоти от 32 kHz до 16 MHz. MCU MSP430x притежават от 1 kB до 120 kB Flash памет, RAM памет до 10 kB, от 14 до 100 извода. Вградените периферни устройства са: 10-/12-/16-битов ADC, 12-битов DAC, компаратор, LCD драйвер, модул за наблюдение на захранващото напрежение, операционни усилватели, USART/I2C/JTAG интерфейси, хардуерен умножител, 8/16-битови таймера, DMA контролер и гъвкава система за управление на честотата. Характерно за всеки един MCU е, че имат пет режима с намалена консумация и възможност за работа с различни високо и нискочестотни генератори. Типичната RISC архитектура е предназначена за приложения с интензивни пресмятания, като входните данни се зареждат в 16 интегрирани 16-битови регистъра в CPU. Програмната памет, паметта за данни и паметта на периферните модули са разположени в общо адресно пространство 64kB (фон-Нойманова архитектура).

*MSP430F1xx* са 16-разредни Flash MCU с голям брой ПУ – от MCU, притежаващи само компаратор до MCU с вградени 12-битов ADC, DAC, серийни интерфейси - два USART модула, хардуерен умножител, WDT таймер, два 16-битови таймера, компаратор, модул за базова честота BCM, цифрово-управляем генератор DCO (фиг. 31).

Консумацията им е 0,1  $\mu$ A в режим на съхраняване на данните в RAM и 1  $\mu$ A при работещ RTC. 16-битовата RISC архитектура има производителност до 16 MIPS и изпълнява 51 инструкции (27 RISC инструкции и 24 емулирани инструкции), като времето за изпълнение на една инструкция е от 1 до 6 такта. Адресното пространство е общо – 64kB и в него е разположена програмната Flash памет: от 1kB до

60kB; RAM памет от 128B до 10kB; ROM 1kB със зареждащата програма и регистрите на ПУ (SFR's).



Фиг. 31. Структура на MCU MSP430F149.

Системата за генериране на тактови сигнали е гъвкава и обезпечава работа с нискочестотен генератор (32 kHz), с високочестотен генератор (8 MHz) или с вградения цифров тактов генератор DCO.

Вградените ПУ са: интерфейси I2C, SPI, UART; интелигентна система за наблюдение на захранващото напрежение с възможност за установяване на прага на сработване на сигнал RESET; разнообразна аналогова периферия: компаратор; до 8-канален 10/12 разреден ADC (200 ksmp/s); до 3-канален сигма-делта 16-битов ADC; 3-канален операционен усилвател с програмируем коефициент на усилване; 12-разреден DAC (до 2 канала); апаратен умножител; контролер за DMA; LCD-драйвер – до 160 сегмента; таймерна система с IC/OC функции; JTAG-логика за вътрешно-схемна настройка и програмиране на Flash-паметта. Произвеждат се в корпуси от 14 до 100 извода.

MSP430F20xx – нови MCU със свръхниска консумация, които са с разширен температурен диапазон от – 40 C° до + 125 C°, 16 – битов таймер и 10 I/O извода. MSP430x20x1 притежават аналогов компаратор, а MSP430x20x2 и MSP430x20x3 имат вградени SPI и I2C интерфейси и 10-битов ADC (MSP430x20x2) или 16-битов сигма-делта ADC (MSP430x20x3). Предимствата на MSP430F2xx спрямо MSP430F1xx са: намаляване на консумацията до 0,1 uA в режим RTC; излизане от режим с намалена консумация за < 1 us; програмиращо

напрежение – 2,2V; детектор на захранващото напрежение (BOR) с нулева консумация; вградени “изтеглящи” резистори; нов двупроводен интерфейс за вътрешно-схемна настройка (Spy); усъвършенстване на системата за защита на паметта; универсален последователен интерфейс, поддържащ UART, SPI, I2C, IrDA.

*MSP430F4xx* са MCU със свръхниска консумация и имат вградени: LCD контролер за 96 сегмента, 16-битов таймер с три секции за IC/OC функции, вграден компаратор, програмиране по серийния порт, памет: от 4KB до 8KB ROM или Flash памет; 256B RAM. Серия *MSP430FG461x* е с увеличено количество вградена Flash (92 – 120 kB) и RAM (4 – 8 kB) памет; вграден 12-канален 12-битов ADC; два 12-битови DAC; три операционни усилвателя, апаратен умножител, LCD драйвер (160 сегмента); два интерфейсни модула - USART, обезпечаващи универсален синхронен/асинхронен обмен и модул USCI – за обмен по протоколи UART, IrDA, SPI и I2C.

Предимства на *MSP430x*: ниска консумация, опростена архитектура, програмна съвместимост между различните модели на фамилията *MSP430x*, различни периферни подсистеми – прецизни ADC и DAC, компаратори, LCD драйвери, SPI, UART, DMA, излизане от режим с намалена консумация за < 6 us. Новите членове на фамилията *MSP430X* имат разширена система инструкции и възможност за адресиране на повече от 64KB памет [30, 50].

### III. 4. 6. Микроконтролери ARM

ARM са високопроизводителни MCU с 16/32-битово RISC ядро. Произвеждат се от фирмите: Analog Devices, Atmel, Cirrus Logic, OKI, Philips Semiconductors, ST Microelectronics, Texas Instruments, Intel, Freescale, Samsung, Sharp, Hynix и др. Заемат 75% от пазара на 32-битовите MCU.

ARM архитектурата изпълнява 32-битови ARM и 16-битови Thumb<sup>®</sup> инструкции, като поддържа технологиите за Java ускорители (Jazelle), за сигурност (TrustZone), интелигентен мениджър на консумацията (IEM), технологии SIMD и NEON.

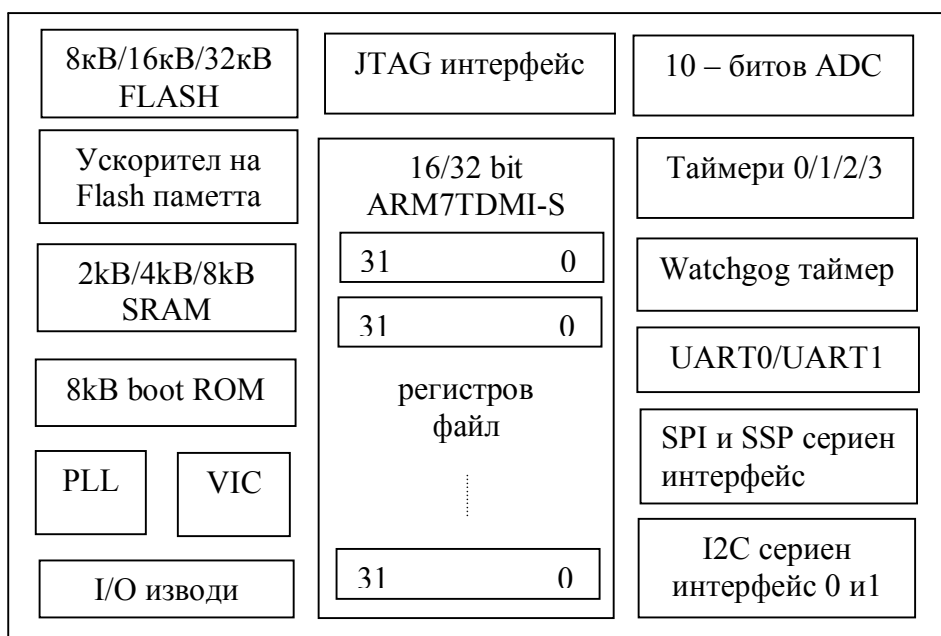
ARMv4 е най-старата архитектура, която се поддържа при някои модели ARM7 и Intel StrongARM MCU. Изпълнява 32-битови ARM инструкции. ARMv4T изпълняват 32-битови ARM и 16-битови Thumb инструкции, което позволява компилаторите да генерират компактен код (спестява се до 35% адресно пространство в сравнение с еквивалентен 32-битов код), а същевременно се използват всички предимства на 32-битовата система. Thumb е разширение на 32-битовата ARM архитектура и включва най-често използваните 32-битови ARM инструкции, които са компресирани до 16-битов операционен код. При изпълнение, тези инструкции се декомпресират до пълните 32-битови ARM инструкции в реално време. В инструкционния конвейър има Thumb-декомпресор, кодът е

компресиран, използва се по-малко програмна памет и има по-малки системни загуби. Програмистите могат да използват заедно двата вида инструкции.

Архитектурата ARMv5TE освен 32-битови ARM и 16-битови Thumb инструкции, изпълнява разширена система DSP инструкции. Новите инструкции извършват изчисления с насищане и подобряват до 70% обработката на аудио приложения. ARMv5TEJ има Jazelle-разширение, поддържащо технологията на Java ускорителите. Тя обезпечава по-добра обработка на Java-код, отколкото с Java виртуална машина (JVM) и намалява с 80% консумацията на MCU.

ARMv6 има множество подобрения в системата на паметта, обработката на изключения, поддържането на мултипроцесорна работа, поддържане на SIMD изчисления (подобряват изпълнението на аудио и видео приложения до четири пъти).

Като варианти на MCU ARMv6 са добавени Thumb-2 и TrustZone технологиите. ARM TrustZone-разширенията обезпечават хардуерно две отделни адресни пространства. Код, който се изпълнява в адресно пространство, което не е защитено, не може да получи достъп до адресно пространство, маркирано като защитено. Новият мониторинг режим поддържа превключването между двете пространства.



Фиг. 32. Структура на MCU LPC2101/2102/2103

Тази технология осигурява защита на широк кръг услуги, например банкиране по мобилен телефон, ползване на мултимедийни услуги и др. ARM Thumb-2 архитектурата дава възможност за смесване на двата режима ARM и Thumb. Добавени са нови 32-битови

инструкции, които изпълняват традиционните 16-битови инструкции в Thumb режим. Така се намалява необходимостта от балансиране между ARM и Thumb код.

Моделът ARMv7A поддържа NEON-технологията. MCU представлява хибридна 64/128-битова SIMD архитектура, предназначена да ускори обработката на мултимедийни приложения и такива, свързани с обработката на сигнали – кодиране/декодиране на видео сигнали, 3D графики, обработка на говор, компресиране на аудио сигнали, обработка на изображения, звуков синтез. R-моделите са за системи, работещи в реално време, а M-моделите са оптимизирани за MCU и нискоструващи приложения. Разширението VFP (VFP - Vector Floating Point) поддържа обработването на числа с плаваща запетая с единична и двойна точност, напълно съвместими със стандарта IEEE 754.

Възможността за векторни изчисления увеличава бързодействието при обработката на 2D/3D трансформации, цифрово филтриране и др. IEM е технология, която съдържа алгоритми за оптимално балансиране между натоварването на MCU и неговата консумация. IEM работи с OS и приложения за мобилни телефони.

LPC2101/2102/2103(Philips) са с вградени 2kB/4kB/8kB RAM памет, 8kB/16kB/32kB FLASH памет; 128-битов интерфейс, който позволява високоскоростни операции ISP/IAP за зареждащата програма при 70 MHz (изтриване на Flash-паметта се извършва за 100ms, а запис на 256 B – за 1 ms); 8-канален 10-битов ADC; два 32-битови таймера/брояча със седем capture/compare канала; два 16-битови таймера/брояча с три IC и седем OC канала; RTC с ниска консумация, работещ с честота 32 kHz и независимо захранване; серийни интерфейси – два модула UART (16C550), две бързи I2C шини (400 kbit/s), SPI и SSP с буфериране и променлива дължина на данните; векторен контролер на прекъсванията (VIC) с програмируеми приоритети на прекъсванията и векторните адреси; до 35 I/O изводи с общо предназначение; до 13 извода за външни прекъсвания с чувствителност по фронт и ниво; вграден тактов генератор, работещ външен кристал от 1 MHz до 25 MHz; максимална честота на CPU – 70 MHz от програмируемата PLL логика; три режима с намалена консумация - режим IDLE, режим Power-down с активен RTC и режим Power-down и извеждане от него чрез външно прекъсване или RTC (фиг. 32). [42, 43]

Предимствата на ARM MCU: голямо количество периферни модули: ADC, DAC, USB, SPI, UART, I2C, CAN, Ethernet, SDRAM, Flash: 32KB-1MB, RAM: 4KB-256KB, позволяващи разработване на разнообразни лабораторни упражнения; голямо бързодействие - повечето ARM7 работят с честота 60 MHz, а ARM9 – със 150 MHz; ниската консумация - изисква 0.5 -1mA на MHz. Недостатък е сложната структура и високата цена на развойните средства.

## ГЛАВА IV

### Цифрови сигнални процесори (DSP)

#### IV. 1. Предназначение и области на приложение на сигналните процесори

Цифровата обработка на сигналите (Digital Signal Processing) е една от най-мощните технологии на 21 век в науката и инженеринговата дейност. В последните години бяха направени революционни промени в области като комуникации, образна диагностика, комуникации, музикална индустрия и т.н. За всяка област е разработена DSP-технология със свои собствени алгоритми, математически апарат и специализирани техники. Цифровата сигнална обработка представлява математиката, алгоритмите и техниките, които се използват, за да се обработват тези сигнали, след като са били преобразувани в цифров вид. Това включва множество задачи, като: увеличаване на визуалните образи, разпознаване и генериране на говор, компресиране на данни и др.

##### IV. 1. 1. DSP в телекомуникациите.

При телекомуникациите се предава информация от една точка до друга. Видовете информация са много: телефонни разговори, TV-сигнали, компютърни файлове и др. типове данни. За да бъде предадена информацията, е необходим канал между две точки – дуплексна линия, радио сигнал, оптичен кабел и др. Тук са разгледани три специфични примера за DSP в телекомуникациите :

##### а) Мултиплексиране

В света има ~ 1 милиард телефона. С натискане на няколко бутона, превключващите мрежи позволяват всеки един от тях да бъде свързан с друг само за няколко секунди. До 1960, връзката между два телефона е изисквала преминаване на аналоговия гласов сигнал през механични превключватели и усилватели, а за една връзка е била необходима една двупроводна линия. DSP преобразува аудиосигнала в поток от серийни цифрови данни и много разговори могат да се предадат по един канал. Напр. при T-преносна система (*T-carrier system*) се предават едновременно 24 гласови сигнала. Всеки сигнал се преобразува в цифров вид и се отчита с 8000 дискрета/сек., използвайки 8-битово ADC (логаритмична компресия).

##### б) Компресиране

При преобразуване в цифров вид на гласов сигнал с 8000 дискрета/сек., голяма част от цифровата информация е излишна, защото информацията, която носи един дискрет, се дублира от съседните дискрети. Много от DSP алгоритмите са разработени да преобразуват цифров гласов сигнал в потоци от данни, които изискват

по-малко битове в секунда (алгоритми за компресиране на данни). Декомпресиращите алгоритми се използват да възстановят сигнала в оригиналната му форма. Тези алгоритми се различават по количеството компресирана информация и качеството на звука, което се постига като резултат. Намаляването на скоростта на компресиране на данните от 64Kbit до 32Kbit не води до нарушаване на качеството на звука. Когато се компресира до 8Kbit/sec, се получават смущения в звука, но това все още се използва при отдалечени телефонни мрежи, а най-голямата достигната скорост на компресиране, е 2 kilobits/sec, при която звука е силно изкривен, но се използва в някои случаи на военни или презокеански комуникации.

#### в) Контролиране на ехо-ефекта

Ехото представлява сериозен проблем при далечните телефонни връзки. Когато някой говори по телефона, сигналът (гласът) преминава в отсрещния приемник, където част от него се връща като ехо. Ако връзката е на разстояние няколко стотин мили, времето за получаване на ехо е няколко ms. Човешкото ухо е привикнало да чува ехо с тези малки закъснения и връзката е нормална. Но ако дистанцията е по-голяма, ехото става твърде забележимо, защото закъснението е няколко стотици ms за интерконтиненталните комуникации и е съществено.

Този проблем се решава с DSP, като се измерва обратния сигнал и се генерира обратен антисигнал, който прекъсва ехото. Същата техника позволява потребителите да чуват и говорят едновременно, без да се получава шум. Използва се и за намаляване на околния шум чрез цифрово генериран антишум.

### IV. 1. 2. DSP в аудиосистемите

#### а) Музика

DSP обезпечава филтрация, допълване и изваждане на сигналите, редактиране на сигналите и др. Едно от най-интересните приложения на DSP е изкуствена реверберация (ехтене). Ако самостоятелните канали се добавят заедно, в резултат откъсът звучи разводнено, сякаш музикантите свирят отвън. Това е, защото слушателите са под влияние на ехото или реверберацията, които обикновено са минимизирани в музикалното студио. DSP позволява да се добавя изкуствено ехо по време на смесване на сигналите, за симулиране на различни външни среди. Ехо със закъснение от няколко стотици ms дава усещане за катедрала, а при закъснение от 10-20 ms – възприятие от по-модерна зала.

#### б) Генериране на говор

Генерирането и разпознаването на говор се използва за комуникации между хората и машините. Човек използва своя говор и слух много повече от своите ръце и очи. Това е много удобно, когато

очите и ръцете са заети с нещо друго, като: каране на кола, мед. операция и др. За генериране на реч се използват два подхода: цифров запис и симулация на говорния тракт. При цифровия запис, човешкият глас се дигитализира и съхранява, обикновено в компресирана форма. По време на възпроизвеждане, съхранените данни се декомпресират и конвертират в аналогов сигнал. Един час запис на говор изисква само около 3 МВ памет.

Симулаторите на гласовия тракт са по-сложни, защото се опитват да имитират физическите механизми, чрез които хората говорят. Гласовият тракт при хората представлява акустична кухня с резониращи честоти, определени от размера и формата на камерите. Звук възниква в гласовия тракт по един от двата основни начина, наречени гласово и проходно звучене. При гласовите звуци, вибрациите на гласовите струни създават приблизително периодични импулси във въздуха в акустичната кухня. В сравнение с тях, проходните звуци произлизат от шумната турбуленция на въздуха в тесни конструкции като зъби или устни. Симулаторите на гласовия тракт генерират цифрови сигнали, които наподобяват тези два типа възбуждане. Характеристиките на наподобяваната кухня се симулират чрез пускане на възбудения сигнал през цифров филтър с подобни резонанси.

#### в) Разпознаване на реч

Автоматичното разпознаване на човешката реч е по-сложно отколкото генерирането на реч и се извършва на две стъпки: извличане на характеристиките на говора и съвпадение на характеристиките. Всяка дума от постъпващия аудио сигнал се изолира и после анализира, за да се идентифицира типа на възбуждането и резониращите честоти. Тези параметри после се сравняват с предишните примери на изговорени думи, за да се постигне най-доброто съвпадение. Често тези системи са ограничени до няколко стотици думи, допускат говор с паузи между думите и трябва да бъдат обучени индивидуално за говорещия.

#### IV. 1. 3. Ехо-локатори.

Основен метод за получаване на информация за отдалечен обект, е отразяване на вълна от него. Радарите работят чрез предаване на радиовълни и изследване на получения сигнал за ехо от самолет. При хидролокаторите, звуковите вълни се предават през водата, за да се детектират подводници и др. обекти.

#### а) Радари

Най-простата радарна система представлява радиопредавател, който предава импулс радио честотна енергия с продължителност няколко микросекунди. Този импулс се поддържа от силно насочена антена, където радиовълната се разпространява със скоростта на



светлината. Самолет, който е по пътя на тази вълна, ще отрази малка част от енергията обратно в приемаща антена, която се разполага близо до излъчващата страна. Разстоянието до обекта се изчислява като се изчислява времето между изпратения импулс и полученото ехо. Посоката до обекта се намира по-лесно – като се установи в коя посока е била насочена антената, когато е получено ехото.

Работният обхват на радарите се определя от два параметъра: каква енергия е необходима за първоначалния импулс и нивото на шум на радиоприемника. За съжаление, увеличаването на енергията на импулса обикновено изисква той да е по-продължителен, а това намалява точността при измерване на времето. Получава се конфликт между двата важни параметъра: възможността да се детектират обекти на голямо разстояние и точното определяне на разстоянието до обекта.

Чрез DSP настъпват качествени промени в три области, всяка от които, е свързана с основния проблем. Първо, DSP може да компресира импулса, след като той се получи, осигурявайки по-добро определяне на разстоянието без намаляване на работния обхват. Второ, DSP може да филтрира получения сигнал за намаляване на шума. Това увеличава обхвата, без да повлияе на точността при определяне на разстоянието. Трето, DSP позволява много бързо да се изберат и генерират различни форми и дължини на импулса. Това дава възможност да се оптимизира импулса за частен проблем с детектирането на обекта. Всичко това се прави с честоти на следене на сигнала, съвместими с радиочестотите, достигащи стотици MHz.

#### б) Хидролокатори (сонари)

Делят се на две категории – активни и пасивни. При активните сонари се излъчва звуков сигнал между 2 kHz и 40 kHz, който се предава във вода и се анализира полученото ехо. Приложение: детектиране и локализиране на тела под водата, навигация, комуникации и изследване на морското дъно. Типичният работен обхват е от 10 до 100 км. Пасивните сонари само “слушат” за подводни звуци, които могат да бъдат: естествена турбуленция, механични шумове от подводници и плавателни съдове и др. Те използват пониски честоти от активните сонари, защото те се разпространяват през водата с по-малка абсорбция. Работните обхвати могат да бъдат хиляди километри.

#### в) Сеизмология

Един от най-разпространените методи за изследване на подземните слоеве е рефлексно-сеизмичният метод. Състои се в изпращане на звуков сигнал в земята, който създава единично ехо от всеки слой, през който премине сигнала. Между слоевете се получават вторични ехо-сигнали, което прави детектирането на сигнала доста

сложно и трудно. Чрез DSP се изолират първичните от вторичните ехо-сигнали.

#### IV. 1. 4. Обработка на изображения.

Изображенията са сигнали със специални характеристики. Техните параметри могат да бъдат измервани от разстояние и съдържат огромно количество информация. Напр., за съхраняване на 1s TV видеосигнал са необходими повече от 10 MB памет. Това е хиляди пъти повече от необходимата памет за гласов сигнал със същата дължина.

DSP се използва широко в медицината. Пр. е компютърната томография, където се изследват X-лъчи, преминаващи през части от човешкото тяло. Компютърната томография е специален рентгенов метод за изследване, при който чрез тесен сноп рентгенови лъчи се пролъчва определена интересуваша ни зона от човешкото тяло. След това, чрез DSP на получения масив от данни, се изграждат детайлни образи на изследваната част на тялото. Така има възможност точно да се определи мястото и големината на болестно огнище (например тумор, инфекция), както и оглед на околните органи. Това е от голямо значение за поставяне на ранна и точна диагноза и планиране на точните лечебни методи.

При метода за диагностика с използване на ядрено-магнитен резонанс (MRI - Magnetic Resonance Imaging), се използват изкуствено генерирани магнитни полета и радиовълни, за да се сондира вътрешността на човешкото тяло. По-голямата част от човешкото тяло съдържа водни молекули, съставени от два положителни водородни атома - протони. Когато човек влезе в мощното магнитно поле, на скенера протоните се подреждат по посока на магнитното поле, в което се намират. Задейства се второ радиочестотно електромагнитно поле, което кара протоните да абсорбират част от неговата енергия. След като бъде изключено второто поле, протоните освобождават тази енергия под формата на радио-честотно излъчване, което се засича от скенер. Позицията на протоните може да бъде контролирана чрез добавянето на допълнително електромагнитно поле. Информацията се представя като образи чрез DSP.

DSP се използва във видеотелефоните, компютърни програми, които показват движещи се обекти, цифровата телевизия, космическите изследвания и др.

#### IV. 2. Различия между DSP и GPP

Компютърните системи намират приложение основно в две области:

- *обработка на данни* (бази от данни, електронни таблици, текстообработка и др.)
- *математически изчисления* (приложения в науката, техниката, цифрова обработка на сигналите).

Всички МР са способни да изпълняват и двете задачи, но е трудно и скъпо да се реализира устройство, което едновременно е оптимизирано за двете задачи. GPP (Pentium) са насочени към обработката на данни, а DSP – към математическите изчисления, използвани в цифровата обработка на сигнали [22].

Компютрите, които се използват за обработка на данни, трябва да са в състояние да записват големи обеми от данни, да ги подреждат, сравняват, местят и т.н., като времето за изпълнение на дадена операция не е критично. Разработеното за тях програмно осигуряване не е оптимизирано, поради което програмите изискват все повече и повече памет, както и все по-бързи процесори, за да могат да се изпълняват за приемливо време. Тези компютри могат да се използват за обработка на сигнали, но при тях времето за изчислителни операции не се гарантира, за това те работят трудно в реално време.

От друга страна, приложенията за цифрова обработка на сигнали в реално време изискват бързо извършване на математическите операции, времето за изпълнение на инструкциите трябва да бъде ясно дефинирано, операциите да бъдат предсказуеми. Това налага изключително висока ефективност на апаратното и програмното осигуряване. Ядрото на всички основни приложения на цифровата обработка на сигналите е от вида  $\sum h(i)x(j)$ , което представлява сума от произведения и скоростта, с която ще се изпълнява тази операция, определя до голяма степен производителността на системата. Това са бързи трансформации на Фурие, цифрова филтрация, изчисляване на корелация и конволюция, матрични умножения и др..

Табл. 6 показва най-важните разлики между двете категории приложения. Обработката на данни включва съхраняване и сортиране на информация. Например, една програма за обработка на данни има за основна задача да съхрани информацията (въведена от оператор), да я организира (премахне или премести, провери за грешки, форматира стр. и др.), след което да я запише или да я разпечата. Тези задачи са свързани с прехвърляне на данни от едно място от паметта в друго и тестване за равенства/неравенства ( $A=B$ ,  $A<B$  и др.).

Друг пример е печатането на документ. Компютърът непрекъснато тества входното устройство (мишка или клавиатура) за двоичен код, който индицира “печат”. При откриване на този код, програмата премества данните от паметта на РС в паметта на принтера. Тук също се наблюдават двете основни операции: прехвърляне на данни и проверка за равенство или не. Рядко се използва математика и изчисленията не оказват влияние на скоростта на изпълнение на програмата.

Този тип процесори са оптимизирани за много бързи математически обработки, които се повтарят циклично. Основната операция “умножение с натрупване” (Multiply-ACcumulate – MAC) се

повтаря циклично и определя скоростта на работа на процесора в реално време (MIPS, BIPS).

Таблица 6. Обработка на данни и математически изчисления.

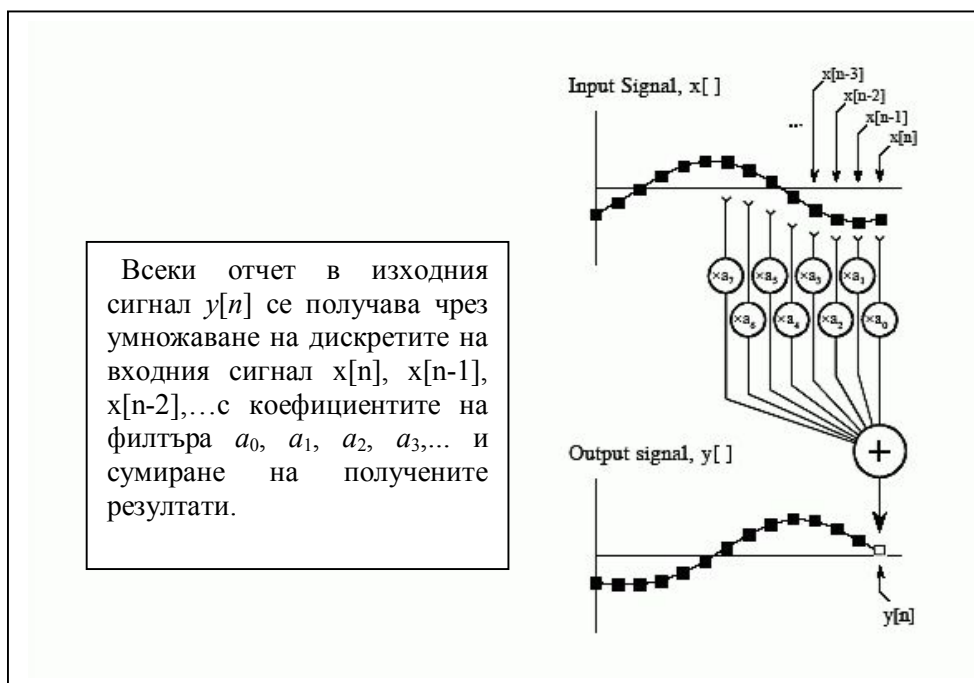
|                    | Обработка на данни  | Математически изчисления   |
|--------------------|---|--|
| Типични приложения | Обработка на думи, управление на бази от данни, операционни системи и др.                           | Обработка на цифрови сигнали, управление на задвижвания, симулации и др. |
| Основни операции   | Прехвърляне на данни: $(A \rightarrow B)$<br>Тестване: за равенства/неравенства: $(if A=B then...)$ | Събиране: $(A+B=C)$<br>Умножение: $(A \times B=C)$                       |

В зависимост от това, дали притежават модул за операции с плаваща запетая или не, DSP се делят на две групи – с целочислена аритметика и аритметика с плаваща запетая. Последните са много удобни, защото решават проблема с мащабирането на данните, препълванията и др. Процесорите с целочислена аритметика обикновено имат поне един акумулатор с по-голяма дължина – 40 бита. Напр., при умножение на две 16-битови числа, резултатът не може да бъде по-голям от 32 бита, но когато се извършва натрупване на такива произведения, резултатът може да бъде по-дълъг.

Освен бързата аритметика, DSP трябва да поддържа и бързи стандартни математически функции, бързо ALU и програмируеми преместващи (shift) регистри за предварително мащабиране на операндите (преместване с  $n$ -бита наляво или надясно е еквивалентно на умножение и деление на степените на 2).

При DSP алгоритмите, скоростта на изпълнение се ограничава почти напълно от броя умножения и събирания. Фиг. 33 показва FIR (FIR - Finite Impulse Response) филтър (филтър с крайна импулсна характеристика). Входният сигнал е означен с  $x[n]$ , а изходният с  $y[n]$ . Задачата е да се изчислят отчетите  $n$  на изходния сигнал, т.е.  $y[n]$ . FIR филтърът извършва тези изчисления чрез умножаване на съответните отчети на входния сигнал с група коефициенти на филтъра  $a_0, a_1, a_2, a_3, \dots$  и събиране на резултатите от умножението. В равенството, за  $y[n]$  се получава:

$$y[n] = a_0x[n] + a_1x[n-1] + a_2x[n-2] + \dots \quad (4).$$



Фиг. 33. Цифров FIR филтър

### IV. 3. Кръгово буфериране

Съществуват два начина за обработка на сигналите – чрез постобработка и обработка в реално време. При първият тип обработка, пълният входен сигнал е в компютъра в същото време на обработката. Напр., сеизмографите могат да използват сеизмометър за запис на движението на земята по време на земетресение. След като то завърши, информацията може да се прочете от РС и да се анализира в същото време. Друг пример за постобработка е компютърната томография или ядрено-магнитен резонанс. Данните се получават докато пациентът е в машината, но образите се обработват със закъснение. Основното при постобработката е, че информацията за обработващата програма е налична.

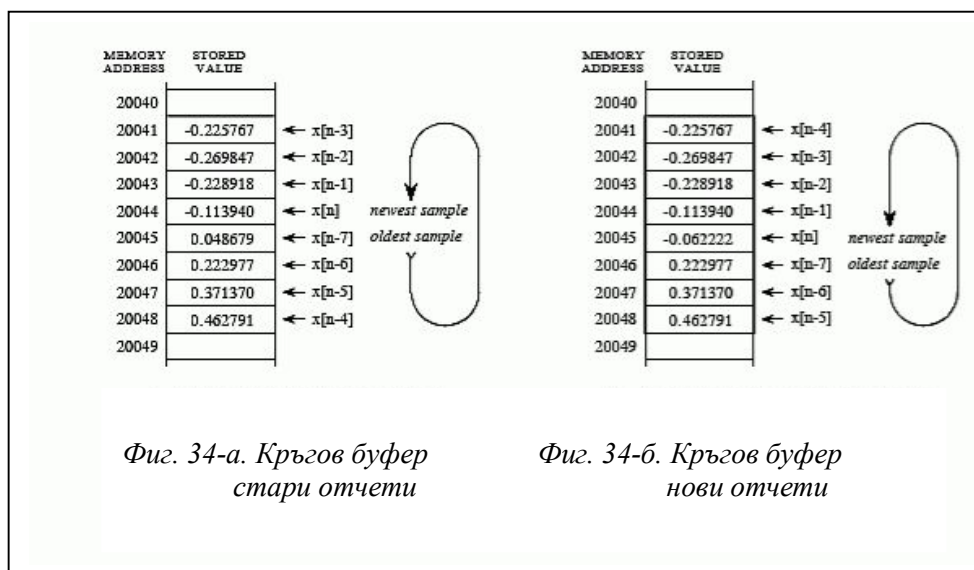
При обработката в реално време, която извършват DSP, изходният сигнал се генерира в същото време, в което се получава входният – напр. при телефонните комуникации, при радарите и др. Приложенията за обработка в реално време приемат текущия дискрет на сигнала, изпълняват алгоритъм и извеждат непрекъснато обработен вече изходен дискрет. Алтернативно, те могат да приемат група дискрети, да изпълнят алгоритъм и да изведат група изходни дискрети.

За реализиране на FIR филтъра от фиг. 33 в реално време, трябва да се изчислят изходните отчети, а за това е необходим достъп до най-новите отчети от входния сигнал. Ако се използват осем

коефициента на този филтър  $a_1, \dots, a_7$ , това означава, че трябва да се знаят текущите отчети  $x[n], x[n-1], \dots, x[n-7]$  на входния сигнал. Тези осем отчета трябва да се съхранят в паметта и да се актуализират през определен период. За целта се използва кръгово буфериране (фиг. 34).

Кръговият буфер се разполага в осем последователни клетки от паметта, пр. от адрес 20041 до 20048. Фиг. 34-а показва как осемте отчета от входния сигнал могат да бъдат разположени в буфера, а фиг. 34-б показва промяната, след като се получи следващия отчет [1, 22]. Идеята на кръговия буфер е, че край на този линеен масив е свързан с неговото начало (адрес 20041) и се разглежда като следващ след адрес 20048. Обхождането на масива е чрез указател, който показва къде се намира текущият отчет. Напр. във фиг. 34-а, указателят съдържа адрес 20044, докато във фиг. 34-б, съдържа 20045. Когато се получи нов отчет, той измества най-стария отчет в масива и указателят се премества с един адрес напред.

Компонентите, необходими за правилната работа на кръговия буфер, са четири. Първият е указател, показващ началото му в паметта (в сл. 20041). Вторият е указател, показващ края на масива (в сл. 20048) или променлива, която съдържа неговата дължина (в сл. 8). Трето, трябва да се определи стъпката при адресиране на паметта. В този случай стъпката е едно: адрес 20043 съдържа един отчет, адрес 20044 – следващият и т.н.



Тези три стойности дефинират размера и конфигурацията на кръговия буфер и не се променят по време на работата на програмата. Четвъртият компонент, указателят към текущия отчет, трябва да се модифицира при всеки нов получен отчет, т.е. трябва да има програмна логика, която контролира как четвъртата стойност се обновява на базата на първите три стойности. Когато логиката е

проста, това е много бързо, за това DSP трябва да се оптимизира при управлението на кръговите буфери и да се постигне най-високата възможна скорост на изпълнение.

Действията за реализиране на FIR-филтър чрез използване на кръгов буфер за входните сигнали и коефициентите, са следните:

1. Получаване на отчет от ADC и генериране на прекъсване.
2. Откриване и обработка на прекъсване.
3. Прехвърляне на стойността на отчета във входния кръгов буфер.
4. Актуализиране на указателя на входния кръгов буфер.
5. Нулиране на акумулатора.
6. Управление на цикъла за всеки от коефициентите.
7. Извличане на коефициентите от кръговия буфер на коефициентите.
8. Актуализиране на указателя за кръговия буфер на коефициентите.
9. Извличане на отчета от входния кръгов буфер.
10. Актуализиране на указателя на входния кръгов буфер.
11. Умножаване на коефициента с входния отчет.
12. Прибавяне на резултата в акумулатора.
13. Преместване на изходния отчет (акумулатор) в буфер.
14. Преместване на изходния отчет от буфера в DAC.

Целта е тези стъпки да се изпълняват бързо. Стъпки 6-12 се изпълняват много пъти (по един път за всеки от коефициентите на филтъра). GPP изпълняват тези стъпки последователно, докато DSP са конструирани така, че да ги изпълняват паралелно. В някои случаи, всички операции в цикъла (стъпки 6-12) могат да бъдат завършени за един машинен цикъл.

#### **IV. 4. Архитектура на цифров сигнален процесор (DSP)**

За постигане на много голямо бързодействие, не е достатъчно операцията MAC да може да се извърши бързо, необходимо е операндите, които се умножават, да се извличат достатъчно бързо – обикновено в същия цикъл, в който се извършва и MAC операцията. GPP извличат единия операнд, след това другия, умножават ги, събират произведението със съдържанието на регистър или клетка от паметта. Очевидно е, че тази процедура е много бавна. При DSP основната операция MAC е между два, независими един от друг операнда, напр.  $\sum h(i)x(j)$ , където  $h(i)$ ,  $i=1, N$  са коефициенти, а  $x(j)$ ,  $j=1, N$  – прозорец от сигнала. Едновременното извличане на операндите от паметта и извършването на операцията MAC е един от основните методи за постигане на високо бързодействие. В DSP се използват две отделни магистрали, които обслужват програмната памет и паметта за данни. Операцията MAC извлича данните

едновременно от двете памети. Наличието на две отделни памети и магистрали към тях изискват специфична архитектура (Харвардска/SHARC). Въведени са инструкционен кеш и I/O контролер.

При DSP-алгоритмите прехвърлянето на информация от/към паметта води до голяма загуба на време. Информацията включва данни, напр. отчетите на входния сигнал и коефициентите на филтъра, както и програмни инструкции. Ако трябва да се умножат две числа, които се намират в паметта, е необходимо да се извлекат три двоични стойности от паметта – числата и инструкцията, която указва какво да се изпълни. Умножаването на две числа изисква минимум три цикъла, по един за трансфер на всяко от числата по шината от паметта към CPU. Тук не се отчита времето за трансфер на резултата обратно в паметта, защото се допуска, че той остава в CPU за допълнителна обработка (като сумата на резултатите при FIR филтъра). .

Недостатък на базовата Харвардска архитектура е, че шината на паметта за данни е по-бърза от шината на програмната памет. При умножение на две числа, две двоични стойности се прехвърлят по шината за данни и само една стойност (инструкцията) - по шината на програмната памет. За да се подобри това, част от данните се преместват в програмната памет (във вторична памет). Напр. може коефициентите на филтъра да се прехвърлят в прогр. памет, докато входният сигнал се съхранява в паметта за данни .

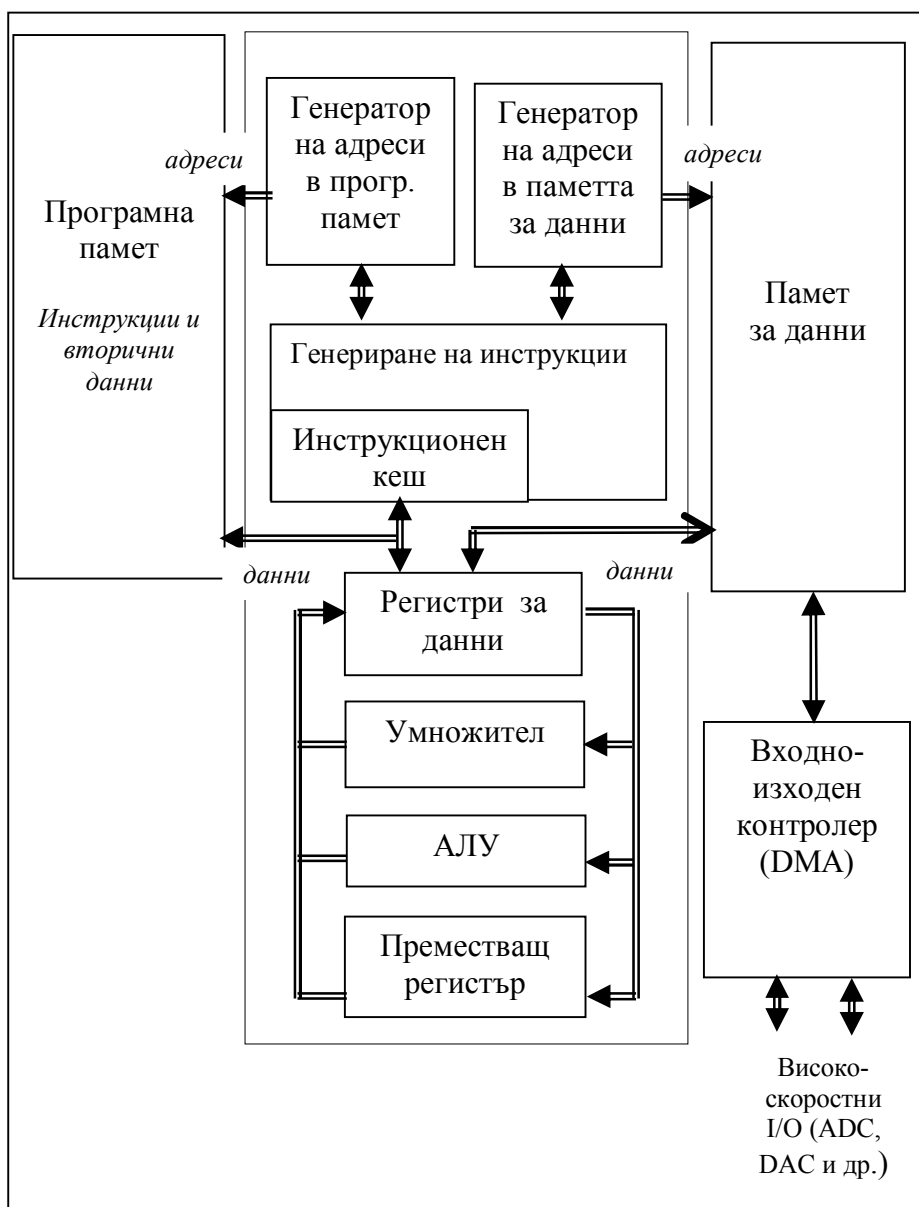
DSP алгоритмите изпълняват през повечето време цикли, като напр. инструкции 6-12. Това означава, че една и съща последователност от инструкции ще се извлича от програмната памет към CPU. При SHARC арх-та, в CPU е въведен инструкционен кеш – малка памет, която съдържа до 32 от най-често използваните инструкции. В началото на цикъла, програмните инструкции преминават през шината на прогр. памет. Това става по-бавно, защото се получава конфликт с коефициентите, които също трябва да бъдат извлечени по тази шина. В следващите изпълнения на цикъла, инструкциите се извличат от инструкционния кеш, така че цялата информация, необходима за инструкцията, може да се получи за един цикъл.

Фиг. 35 представя по-детайлно SHARC – архитектурата. Тя обезпечава серийни и паралелни комуникационни портове за много бързи комуникации. Напр., при 40 MHz тактова честота, двата серийни порта работят със скорост на предаване на данните 40 Mbits/sec. всеки, а шесте паралелни порта обезпечават трансфер на данни с 40 Mbytes/sec. всеки. Когато те се използват едновременно, скоростта на предаване на данните е 240 Mbytes/sec. Съответният хардуер позволява тези потоци от данни да се предават директно в паметта (директен достъп до паметта) без да минават през регистрите на CPU, т.е. стъпки 1 и 14 се извършват независимо и едновременно с другите стъпки, без да се изискват цикли на CPU. Главните шини



също са достъпни отвън, като има допълнителен интерфейс към външната памет и периферните устройства. Това позволява да се работи с 16 GB памет, достъпна с 160 Mbytes/sec за 32-битови данни.

Този тип високоскоростни входно-изходни комуникации са характерни за DSP. Главната цел е да се прехвърлят данните в DSP, да се обработят с математически алгоритми и да се изведат от DSP, преди да е наличен следващият отчет. Всички други задачи са второстепенни.



Фиг. 35. Типична DSP SHARC архитектура

Структурата на CPU съдържа два блока DAG (DAG - Data Address Generator), генериращи адресите на данните (по един за всяка памет). Те контролират адресите изпращани към програмната и паметта за данни, като определят къде информацията ще се чете или ще се записва. Всеки DAG може да управлява до 8 кръгови буфера.

Регистровият файл се използва като при GPP.

Математическите операции са извършват в три секции: умножител, ALU и кръгов регистър (barrel shifter). Умножителят извлича стойности от два регистъра, умножава ги и разполага резултата в друг регистър. ALU изпълнява събиране, изваждане, намиране на абсолютна стойност, логически операции (AND, OR, XOR, NOT), преобразуване между фиксиран формат и формат с плаваща запетая и др. Елементарните двоични операции се извършват от кръговия регистър – изместване, ротация, премахване и добавяне на сегменти и др. Много важна особеност е, че умножителят и ALU могат да бъдат достъпни паралелно.

За всички ключови регистри на CPU има “регистри-сянка”. Те са дублиращи регистри и са за бързо прехвърляне на информацията - за един машинен цикъл. Когато се получи прекъсване в GPP, всички данни трябва да се запишат преди прекъсването да бъде обслужено. Обикновено съдържанието на всички използвани регистри се записва в стека еднократно. При SHARC арх-та, прекъсването се обслужва чрез преместване на данните в регистрите-сянка за един цикъл на тактовия генератор. Когато завърши обслужването на прекъсването, регистрите се възстановяват много бързо, т.е. стъпка 4 се изпълнява много бързо и ефикасно.

#### **IV. 5. DSP за обработка на целочислени данни и данни с плаваща запетая**

DSP се делят на две категории според данните, които обработват – с фиксирана точка и с плаваща запетая. DSP с фиксирана запетая представят всяко число с минимум 16 бита, въпреки че се използват и други формати (Motorola DSP – 24 бита) и имат производителност 30-160 MIPS. DSP с плаваща запетая използват 32 бита за представяне на всяка стойност, но могат да обработват и числа с фиксирана точка; имат производителност 140-800 MIPS .

При DSP с плаваща запетая шумът от квантоване на сигнала е с 30 000 пъти по-малък от шума от квантоване при DSP с фиксирана точка. За програмистите е по-лесно да се разработват алгоритми за DSP с плаваща запетая за намаляване на шума от квантоване.

Предимства на DSP с плаваща запетая : висока точност, широк динамичен обхват, висока стойност на отношението сигнал-шум, лесно използване.

Недостатъци: по-скъпо, по-висока консумация, по-бавна обработка.

### *DSP TMS320C5x(TI)*

Фамилията TMS320C5x се състои от DSP с фиксирана точка, с плаваща запетая и мултипроцесорни DSP [50]. Архитектурата им е предназначена специално за обработка на сигнали в реално време. Имат гъвкава система инструкции за високоскоростна обработка на данните, иновационна паралелна архитектура, ниска себестойност и др. Структурата на DSP се състои от: MAC – блок за умножение и събиране, който изпълнява повече от една MAC операция за 1 маш. цикъл, вътрешни регистри за съхраняване на резултата и операндите, различно количество вградена памет за данни, допълнителни адресни регистри >8 и ALU за генериране на адреси, множество вътрешни магистрали и двоен достъп до паметта за данни, отделен достъп до програмната памет и паметта за данни, Харвардска архитектура, DMA – модул за директен достъп до паметта.

#### IV. 5. 1. Структура на DSP TMS320C5420 – общи характеристики:

##### *а) Ядро (CPU):*

- подобрена многошинна архитектура (SHARC) с три магистрали за данни и една за програмна памет и четири адресни магистрали;
- 40-битово ALU с 40-битов кръгов регистър и два 40-битови акумулатора А и В;
- паралелен умножител 17х17 бита и 40-битов суматор за неконвейрни операции MAC (умножение с натрупване), изпълнявани за един цикъл;
- модул за сравнение, избор и съхраняване (CSSU) за осъществяване Viterbi алгоритъм (алгоритъм за декодиране на конволюционни кодове);
- две адресни устройства, съдържащи 8 допълнителни регистъра и две допълнителни регистрови аритметични устройства (ARu);
- у-во за генериране на адреса на данните;
- у-во за генериране на адреса на инструкциите от програмата;

*б) Вградена памет:* 192 KB (64 kB – програмна памет, 64 kB – памет за данни и 64 kB I/O); DARAM – 32KB, SARAM – 168KB; (C5420);

*в) Вградени периферни устройства:* тактов генератор с фазово управление на честотата (PLL) с вътрешен или външен източник на честота; програмируем генератор; JTAG емулираща логика; блок за управление (забрана) на отделните шини; софтуерен генератор, програмируема превключваща логика, таймер, DMA контролер, сериен порт, TDM – мултиплексиран по време сериен порт; буфериран сериен порт BSP; многоканален буфериран сериен порт (McBSP), HPI:

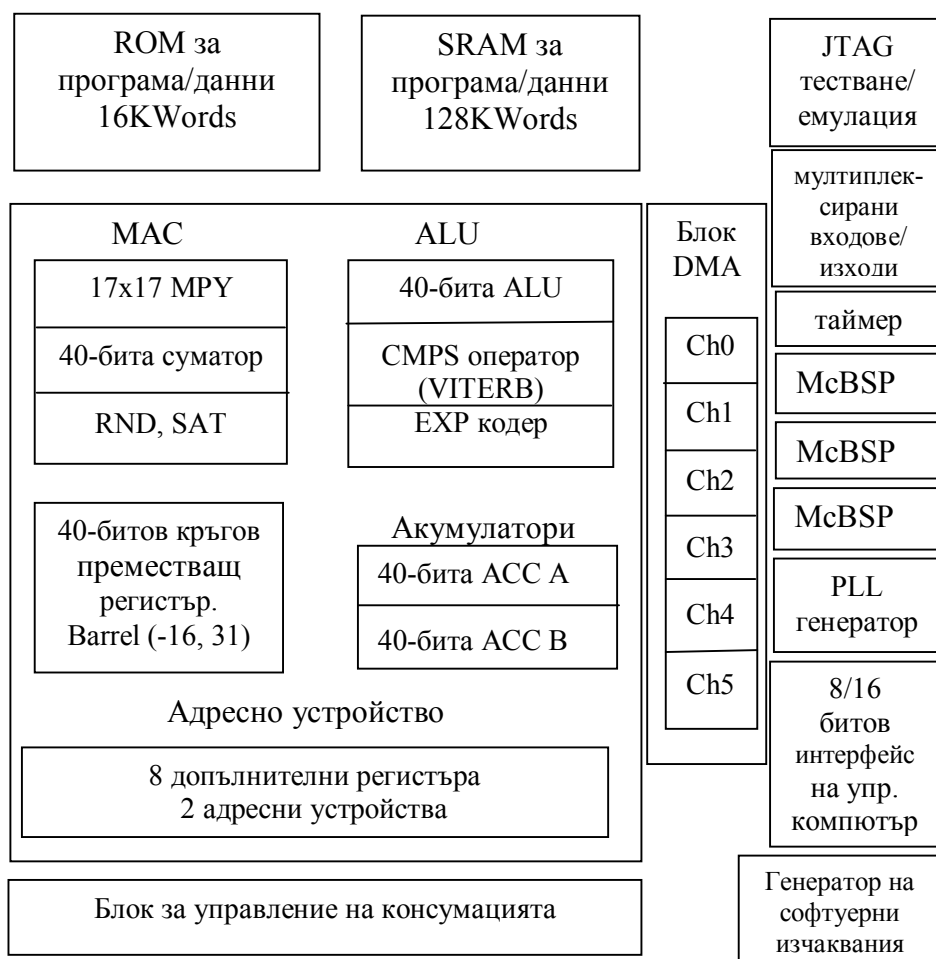
8-битов стандартен хост интерфейс (HPI), 8-битов разширен хост интерфейс (HPI8, 16-битов разширен интерфейс (HPI16).

#### IV. 5. 2. Ядро на TMS5416

а) *ALU* (40-бита) - изпълнява изчисленията заедно с два 40-битови акумулатора А и В. *ALU* може да изпълнява и Булеви операции. Входните данни, които използва са:

- 16-битова непосредствена стойност;
- 16-битова дума от паметта за данни;
- 16-битова стойност от рег. Т за временно съхранение;
- 2 16-битови думи от паметта за данни;
- 32-битова дума от паметта за данни
- 40-битова дума от всеки от акумулаторите.

*ALU* може да работи и като две 16-битови *ALU* и да изпълнява две 16-битови операции едновременно.



Фиг. 36. Блок-схема на TMS5416

#### *б) Акумулатори*

Акумулаторите А и В съхраняват резултата от ALU или блока за умножение/събиране. Могат да съдържат втория операнд за ALU. Акумулатор А може да бъде операнд за умножителя/суматора. Всеки акумулатор се състои от 3 части:

- защитни битове: битове 39 – 32
- старша дума: битове 31-16
- младша дума: битове 15-0

Има инструкции за съхраняване в паметта за данни на защитните битове, старшата и младшата дума и за прехвърляне на 32-битова стойност от акумулатора към/от паметта за данни. Всеки от акумулаторите може да се използва за временно съхранение.

#### *в) Кръгов регистър (Barrel).*

Кръговият регистър има 40-битов вход, свързан към акумулаторите или към паметта за данни (чрез СВ или DB) и 40-битов изход, свързан към ALU или паметта за данни (чрез EB). Той може да извърши преместване в ляво на битове 0-31 и преместване вдясно на 0-16 бита от входните данни. Броят на преместванията се определят от специално поле в инструкцията, от поле ASM от рег. ST1 (рег. на състоянието) или от временния рег. Т, когато се използва като брояч на преместванията.

Кръговият регистър и модулът за кодиране на експонентата нормализират стойностите в акумулатора за един цикъл. Младшите битове (LSBs) на изхода се запълват с 0-ли, а старшите (MSBs) могат да се запълнят или с 0-ли или със знак, в зависимост от състоянието на бита за режима на знаковото разширение (SXM) от рег. ST1. Допълнителните възможности за изместване позволяват CPU да извършва цифрово мащабиране, отделяне на битове, допълнителна аритметика и операции за предотвратяване на препълване.

#### *г) Модул Умножител/Суматор*

Модулът умножител/суматор извършва 17x17 умножение с 40-битово сумиране в един машинен цикъл. Този блок се състои от няколко елемента: умножител, суматор, управляваща логика за знакови и беззнакови входни операнди, детектор на 0, блок за закръгляване, управляваща логика за дробни изчисления, логика за препълване/насищане и 16-битов временен регистър за съхранение Т. Умножителят има два входа: единият вход се избира от рег. Т, от паметта за данни или акумулатор А, а другият се избира от програмната памет, паметта за данни, акумулатор А или е непосредствена стойност.

Бързият вграден умножител позволява C54x DSP да изпълнява ефикасно операции като конволюция, корелация и филтриране. Заедно с ALU изпълняват паралелно операции умножение с натрупване (MAC) и изчисления в един машинен цикъл. Тази функция се използва

за определяне на Евклидово разстояние и реализиране на LMS филтри, които се изискват при сложните DSP алгоритми.

*д) Модул за сравнение, избор и съхраняване (CSSU- Compare Select and Store Unit)*

Този модул извършва сравнения между старшата и младшата дума на акумулатора, разрешава флага за тест/ контрол ТС от рег. на състоянието ST0 и рег. на прехода TRN и избира по-голямата дума за съхранение в паметта за данни. CSSU също ускорява изчисленията на филтри тип Viterbi с диаграма butterfly чрез оптимизиран хардуер, вграден в чипа.

IV. 5. 3. Вградени периферни устройства

*а) Програмируем чакащ генератор*

Представлява генератор на единични импулси (wait-state), който увеличава външните цикли по шината до 7 машинни цикъла. Използва се за осигуряване на достъп до по-бавни периферии (I/O) и външни памети към бързите процесорни шини. Не използва външни елементи и допълнителните цикли за синхронизация от 0-7 се определят програмно чрез рег. SWWSR за всеки 32KW блок от програмната или паметта за данни или за 64KW I/O блок.

*б) Програмируема логика за превключване на памети.*

Тази логика генерира допълнителен цикъл на шината при превключване между различни банки от паметта за данни или програмната памет. По този начин се осигурява допълнително време, за да се приключи работата с едната памет, преди да се започне работа с другата и се избягват конфликти по шината. Също така, може да се вмъкне допълнителен цикъл при превключване от програмната памет към паметта за данни или от една страница на програмната памет към друга (за определени устройства). Големината на банката от паметта за превключващата логика се определя от рег. BSCR.

*в) Хардуерен таймер*

DSP C54x има 16-битов таймер с 4-битов делител. Броячът се намалява с 1 на всеки CLKOUT цикъл. Всеки път, когато той се декрементира до 0, се генерира прекъсване от таймера. Таймерът може да се спре, да му се подаде старт, ресет или да се забрани.

*г) Тактов генератор.*

За генериране на тактов сигнал, DSP C54x има два начина: чрез вътрешен кварцов резонатор или чрез фазово-затворена верига (PLL). В първият случай, тактовият сигнал на CPU се генерира чрез делене 1, 2 или 4 на входния тактов сигнал, постъпващ на извод X2/ CLKIN. Вторият начин използва PLL за генериране на тактов сигнал чрез умножаване на честотата на входния тактов сигнал. По този начин, от

външен източник с ниска честота, се генерира вътрешен тактов сигнал с висока честота, което намалява консумацията, намалява генерираните от източника ЕМІ и съдейства за използване на поевтини външни елементи. Конфигурирането на двата начина се извършва чрез предварително избиране на изводи за режима CLKMD.

*д) Контролер за директен достъп до паметта (DMA)*

DMA прехвърля данни между различни адреси от картата на паметта, без участието на ядрото (CPU). DMA позволява движение на данни от/към програмната памет и паметта за данни, вградените периферни устройства или външната памет, без това да влияе на основната работа на CPU. DMA притежава 6 независими програмируеми канала за 6 различни DMA операции.

*е) Хост – интерфейс (HPI – Host Port Interface).*

HPI представлява паралелен порт, който е за осъществяване на интерфейс с хост-процесор. Информацията между C54x и хост-процесора се обменя чрез вградената памет, която е достъпна и за двете устройства. Има 3 вида HPI: 8-битов стандартен хост интерфейс (HPI), 8-битов разширен хост интерфейс (HPI8) и 16-битов разширен интерфейс (HPI16).

*ж) Серийни портове.*

Серийните портове зависят от модела DSP и са 4 типа: синхронен сериен порт, буфериран сериен порт BSP (BSP – Buffered Serial Port), многоканален буфериран сериен порт McBSP (McBSP – Multichannel Buffered Serial Port) и мултиплексирани по време сериен порт (TDM).

1) *Синхронен сериен порт* – високоскоростен порт, пълен дуплекс, който обезпечава директна комуникация с кодиращи устройства, ADC и др. При повече от един налични серийни портове, те са идентични и независими. Всеки от тях работи с честота  $\frac{1}{4}$  от маш. цикъл. Приемникът и предавателят са двойно буферирани и индивидуално управляеми чрез маскируеми външни сигнали за прекъсване. Данните се предават като байт или дума.

2) *Буфериран сериен порт* – представлява синхронен сериен порт, който притежава автобуфериращо устройство и се тактува с CLKOUT. Той е пълен дуплекс и е двойно буфериран, за да може да предложи гъвкава промяна на дължината на предаваните данни. Автобуфериращото устройство поддържа високоскоростен трансфер и намалява претоварването от обслужване на прекъсвания.

3) *Многоканален буфериран сериен порт* – представлява разширен буфериран сериен порт, следните стандартни характеристики: буферирани регистри за данни, пълна дуплексна връзка и независимо тактуване и образуване на кадрите за приемане и предаване. Разширените възможности на McBSP са: вътрешен

програмируем тактов сигнал и генериране на кадър, многоканален режим и I/O с общо предназначение.

4) *Мултиплексиран по време сериен порт (TDM)* - синхронен сериен порт, който позволява мултиплексиране с времеделение на данните със седем други DSP C54x с TDM портове. TDM се използва в мултипроцесорни приложения и може да се конфигурира за всички синхронни операции.

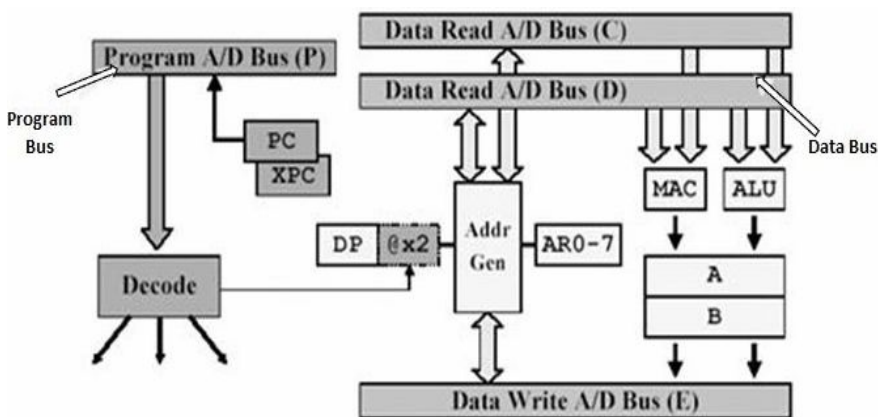
#### IV. 5. 4. Структура на шината

*PB* – програмна шина за кода на инструкцията и извличане на непосредствени операнди от програмната памет;

*CB*, *DB*, *EB* – шини за данни, които свързват CPU, логика за генериране на адресите на данните, логика за генериране на програмните инструкции, вградени периферни устройства и памет за данни (фиг. 37).

- *CB* и *DB* - шини за операндите, които се четат от паметта за данни;
- *EB* – шина за данните, които се записват в паметта

C54x DSP може да генерира два адреса в паметта за данни за един машинен цикъл, използвайки два допълнителни аритметични регистъра ARAU0 и ARAU1.

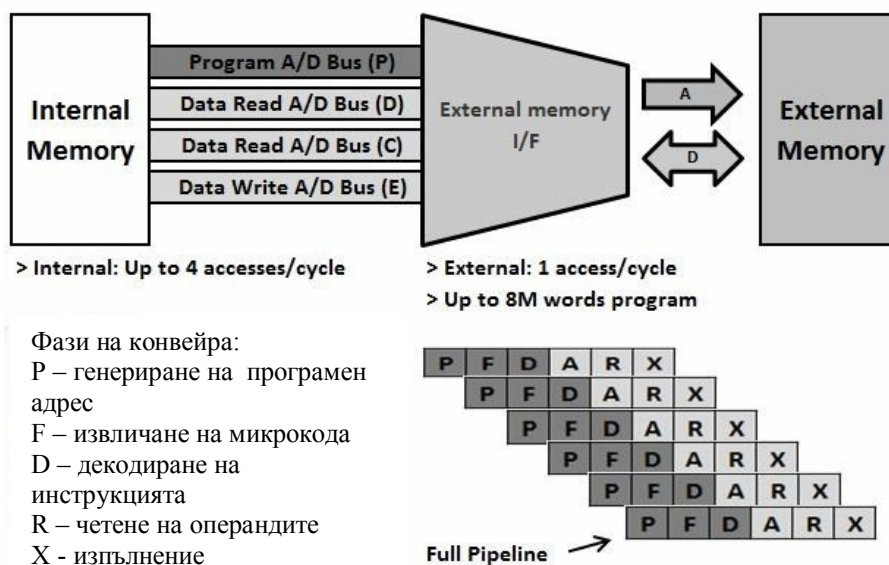


Фиг. 37. Структура на шините

#### IV. 6. Конвейрна обработка на инструкциите

C54xE DSP притежава шест-фазен конвейр, който изпълнява паралелно 6 инструкции в различна фаза на завършеност (фиг. 38).





Фиг. 38. Конвейър на TMS320 C54x

Шестте фази се изпълняват независимо и са:

- 1) генериране на програмния адрес: Програмната адресна шина (PAB) се зарежда с адреса на следващата инструкция, която трябва да бъде извлечена;
  - 2) извличане на кода на операцията от програмната шина (PB) и зареждане в рег. на инструкцията (IR – Instruction Register);
  - 3) декодиранен на съдържанието на рег. IR и определяне на типа на достъпа до паметта;
  - 4) достъп: модул DAGEN генерира адреса на данните и го разполага на шина DAB. Ако инструкцията изисква втори операнд, се зарежда другата шина за адрес на данни CAB;
  - 5) четене на операндите от шините DB и CB;
- изпълнение на инструкцията/запис на операндите

#### IV. 7. Организация на вътрешната памет

Паметта на TMS320C5416 е организирана в три адресни пространства: програмна памет, памет за данни и памет на I/O. Налична е RAM с двоен достъп (DARAM), RAM с единичен достъп (SARAM) и двойно споделена RAM (фиг. 39). DARAM или SARAM са достъпни от подсистемите на многоядрени DSP. Те могат да се конфигурират като памет за данни или памет за програма/данни. Регистрите на CPU (26 бр.) и на периферните устройства са разположени в паметта за данни.

1) Вътрешна ROM памет – част от програмното адресно пространство и в някои случаи е част от паметта за данни. Обикновено съдържа зареждащата програма.

2) Вътрешна RAM памет с двоен достъп (DARAM) – част е от паметта за данни, но може да се използва и като програмна памет за съхраняване на кода на програмата. Състои се от няколко блока, като всеки от тях е достъпен едновременно за четене и запис за един машинен цикъл.

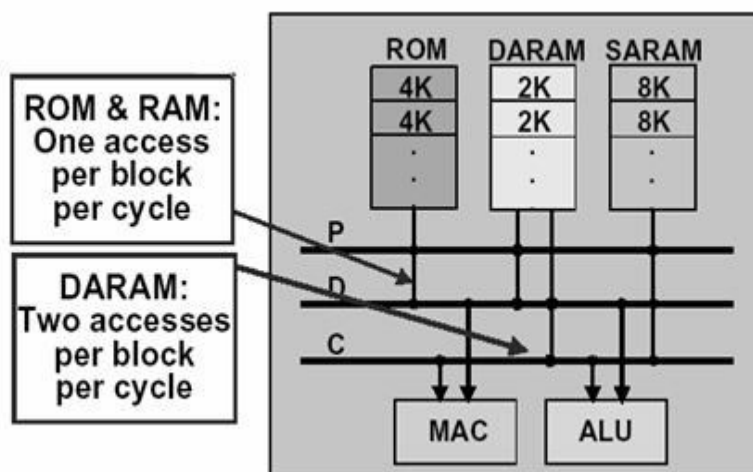
3) Вътрешна RAM памет с единичен достъп (SARAM) - състои се от няколко блока, като всеки от тях е достъпен само за четене или само за запис за един машинен цикъл. Обикновено е част от паметта за данни, но може да се използва и като програмна памет за съхраняване на кода на програмата

4) Вградена двойно споделена RAM памет.

DSP с повече ядра ( $> 1$  CPU) имат в структурата си двойно споделена RAM памет, което позволява всички CPU да имат едновременен достъп до програмната памет.

#### IV. 7. 1. Памет за данни.

Паметта за данни е 64KB, организирани в 16-битови думи. При адресиране в границите на RAM паметта, автоматично се адресира вградената RAM памет, а когато адресът е извън тях, се генерира външен достъп.



Фиг. 39. Вградена памет на TMS320 C54x

#### IV. 7. 2. Програмна памет.

Програмният код може софтуерно да се конфигурира да присъства или не на картата на програмната памет. Ако е в програмното адресно пространство, се генерира автоматичен достъп до него, ако адресът е в неговите граници. Когато модулът за генериране на програмни адреси (PAGEN) издаде адрес извън границите, се генерира автоматично външен достъп.

## ГЛАВА V

### Тестване в производството на електронните модули

Тестването на печатните платки е много важна част от процеса на производство на електронните устройства. Сложността на електронните устройства нараства с времето и това изисква да се използват все повече по-сложни и по-съвършени средства и методи за тестване. Разработването на подходящи тест-процес/процедура и/или тест-стратегия за електронните продукти е сложна задача, сравнима с разработката на самия продукт. Тестовите могат да се обособят най-общо в две големи групи – *структурни*, които са свързани с монтажа на електронните компоненти и *функционални*, които се отнасят до работоспособността на устройството.

Типовите тестове и необходимото оборудване зависят от много фактори и голяма част от тях са свързани директно с произвеждания продукт. Ето защо, се налага продуктът да се проектира по такъв начин, че да позволява оптимално тестване при възможно най-ниски цени. Технологиите за проектиране на нов продукт с предвидени възможности за тестване се нарича DFT (DFT - Design For Test) [51]. Добре проектираният продукт с възможности за тестване, осигурява лесен и удобен достъп до електрическите връзки и тестовите точки и позволява използване на най-ефективните тестове за конкретната платка, например тестове с по-малка продължителност и по-голям обхват.

Друг важен фактор е локализирането на дефекта и отстраняването му при отказалите продукти. Ако изделията не са пригодени за тестване, този процес може да бъде много дълъг и трудоемък. Това може да повиши броя на бракуваните продукти и да увеличи времето за тестване, но по-важното е, че някои дефекти ще останат неоткрити и ще се проявят при потребителя, когато ще е трудно и по-скъпо да бъдат отстранени.

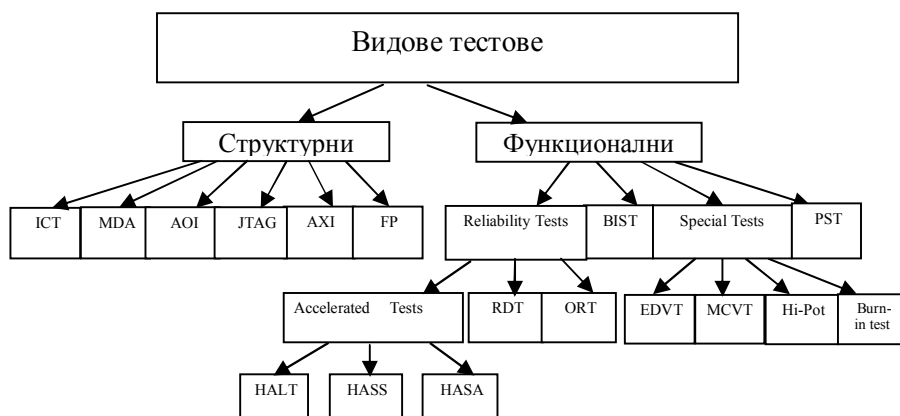
#### V. 1. Видове тестове

Тестовите, използвани в производството са две главни групи:

- *Структурни тестове* - проверяват структурата на платките и устройствата за отворени вериги, късо съединение, качество на спойките, наличие на компоненти, поляритет и ориентация на монтираните компоненти, стойност на пасивните елементи и др..

- *Функционалните тестове* - потвърждават функционалността на устройствата и тяхното действие.

На фиг. 40 е направена класификация на най-използваните тестове в производството, в зависимост от принадлежността им към двете основни групи тестове.



Фиг. 40. Класификация на най-често използваните тестове

### V. 1. 1. Структурни тестове

#### *а) Тест на електрическите вериги (In-Circuit Test - ICT)*

Тестът на електрическите вериги ICT осигурява удобна и ефикасна форма за тестване на платки и изследване на всеки компонент за това, дали е на мястото си и дали е с точната стойност. Тази форма на тестване открива повечето от проблемите на платката, защото голяма част от тях възникват по време на производствения процес и обикновено се свеждат до къси съединения, прекъснати връзки или грешни компоненти. Даже когато една интегрална схема е дефектирала, една от главните причини е статична повреда, която се появява в областите, разположени близо до изводите за връзка с околната среда. С техниките на този тест, тези дефекти се откриват сравнително лесно. ICT не е тест за функционалността на платката, но ако тя е правилно проектирана и асемблирана, трябва да работи нормално. Често по време на ICT се извършва и програмиране на някои програмируеми компоненти или се записва специфична информация за платката/устройството, т.н. cookies.

Оборудването за ICT се състои от две части – тестер и фиксатор. Фиксаторът служи за механично закрепване на тестваната платка в нужната позиция, а с тестера се извършват самите измервания. Най-общо, тестерът представлява матрица от пробници, свързани с драйвери и сензори. Драйверите подават напрежение или ток, в зависимост от конкретният тест, а сензорите регистрират тези величини след преминаването им през определени вериги на тестваната платка. По получените стойности се съди за наличието на дефекти и евентуално техният вид. Трябва да се отбележи, че тестовите установки се разработват специално за всяка отделна платка и в някои случаи могат да имат повече от 1000 пробника.

ICT-тестът има 4 нива: ниво 1 – тест на празна платка; ниво 2 – тест на платка с инсталирани куплунзи; ниво 3 – тест на платка с

монтирани компоненти без интегрални схеми; ниво 4 – тест на платка с инсталирани интегрални схеми.

*б) Анализатор на дефекти в производството (MDA-Manufacturing Defect Analyzer)*

MDA е друга форма за тестване на печатни платки и е опростен вариант на ICT-теста. Чрез него се тестват само производствени дефекти като къси съединения, прекъснати вериги и стойности на някои компоненти. Като резултат, себестойността на тези тествачи системи е много по-ниска от ICT системите, но обхватът им е много ограничен.

*в) Тестване чрез гранично сканираща логика (JTAG Boundary scan testing).*

Гранично сканиращата логика използва специални интегрирани вериги, които имат преместващ регистър в изхода. При серийно свързване на съвместими гранично сканиращи вериги към платката и разрешаване на гранично сканиращия канал, се изпраща последователно дума с данни и после се наблюдават данните на изхода. Анализът им позволява да се получи тестовата информация. По този начин се постига високо ниво на достъп при тестването, без прекъсване на веригите. Освен за тестване, JTAG логиката се използва за програмиране на вградената памет на програмируемите устройства.

*г) Тест-системи с подвижна сонда (FP – Flying Probe).*

Системи за тестване с FP се използват предимно за тестване на основна продукция, прототипи и платки, при които има проблем с достъпа. Те използват електро-механично управлявани сонди за достъп до асемблирани печатни платки. Обикновено се използват за тестване на аналогови компоненти, аналогов сигнатурен анализ, къси съединения и прекъснати вериги. Те са алтернатива на тестването с матрица от пробници за контакт с елементите от печатната платка. Точното позициониране на сондата позволява да се тестват точки на PLCC, SOIC, PGA, SSOP, QFR, някои специални куплунзи и др.

*д) Автоматичен оптичен контрол (Automatic Optical Inspection – AOI)*

AOI широко се използва в много производствени среди и е форма на автоматизиран контрол. Чрез него се обезпечава много голяма степен на повтаряемост и бързодействие, в сравнение с ръчния контрол. AOI е особено удобен, когато се прилага след окончателен монтаж на платките. Така много бързо се локализируют производствени проблеми като дефекти при запояване, грешни компоненти с неточно разположение или полярност. Всички проблеми с опроводяването на платките могат да се разрешат бързо и преди да бъдат мултиплицирани и на други печатни платки. Настройването на AOI

изисква време, за да може тестовото оборудване да изучи платката, но веднъж настроен, AOI се извършва много бързо и лесно. Този тест е подходящ за продукцията с голям обем поради бързодействието си и значителните инвестиции за тестовото оборудване.

*е) Автоматичен контрол с рентгенови лъчи (Automated X-Ray Inspection - AXI)*

AXI е много подобен на AOI, но с появяването на BGA-пакетите, се наложи да се използва форма на контрол, която да може да види обектите, които не са оптически достъпни. AXI системите могат да „видят“ през интегралните схеми и да изследват спойките под пакета, за да проверят качеството им. Особено удобен е и за тестване на куплунзи с много изводи, инсталирани върху платките, където открива огънати или недобре контактуващи изводи.

*ж) Комбинационен тест*

В днешни дни за пълно тестване се използват повече от един метода, за това тестовото оборудване осигурява тестване по няколко тест-методики. Комбинационните тестери се използват основно за тестване на печатни платки. По този начин, единичен електронен тест има много по-голямо ниво на достъп до тестваната платка и обхватът на теста е много по-широк. Освен това, комбинационният тестер може да предприеме различни тестове без да се налага да се премества платката от един тестер на друг. По този начин, единичен комплект тестове може да включва ICT или някои функционални тестове и после JTAG тест.

**V. 1. 2. Функционални тестове (Functional Tests)**

FT се изпълняват в последната фаза на производството на продукта. Обикновено се състоят в емулиране или симулиране на действителната среда на работа на продукта с цел да се провери неговата правилна работа. Средата включва, напр., всяко устройство, което комуникира с тестваното устройство, захранването, товарите, необходими за правилна работа и др. За постигане на по-висока точност и ефективност, често FT са автоматизирани. С подходящ тестващ софтуер, операторите в производството могат да изпълняват всички функционални тестове автоматично чрез компютър. За да може да се извършва това, софтуерът комуникира с външни програмируеми устройства, като програмируеми захранвания, импулсни генератори, трафик генератори, различно комуникационно оборудване, входно-изходни устройства и други. Тук са дадени най-разпространените функционални тестове:

*а) Тестове за надеждност:*

*1) Тестове за ускорено предизвикване на дефекти (Accelerated*

## *Tests)*

Целта на тези тестове е изкуствено да предизвикат дефект в лабораторни условия чрез обезпечаване на среда с големи натоварвания, която е на границата на работните параметри на устройствата или ги превишава с малко. При този тест се очаква продуктът да прояви дефект в лабораторията, така както би станало при нормалната му работа, но за много по-кратко време. Главната цел на ускоряващите тестове е да се идентифицират режимите, при които се получава отказ и да се предскаже работоспособността на продукта във времето чрез стрес-тестове в лабораторни условия.

### *1.1) Тестове за ускорено тестване на годността на устройствата (HALT- Highly Accelerated Life Testing)*

HALT обхваща четири теста. Първо се изпълняват тестове с температурно и вибрационно стъпково натоварване. Целта на тези тестове е да се намерят работните толеранси (ограничения). Те разкриват качествата на продукта като стъпка по стъпка циклично се променят променливи на средата, като температура, удар, вибрации. HALT усложнява тестването на вибрации по три оси, чрез използване на случаен честотен режим. Накрая, HALT тестът може да включва едновременно циклично променяне на множество променливи, напр., циклично променяне на температурата плюс тестване на вибрации. Този подход доближава условията до реалната работна среда. За разлика от обикновеното тестване, целта на HALT теста е повреда на продукта. Когато продуктът откаже, най-неустойчивата връзка се идентифицира, тогава инженерите знаят точно какво трябва да се направи, за да се подобри качеството на продукта. След отказ на продукт, най-слабите компоненти се подобряват и после се подлагат на друг HALT тест с повишена температура, вибрации или разтърсване, докато продуктът отново откаже. Това идентифицира следващата най-слаба връзка. След провеждане на няколко HALT теста, продуктът става много устойчив и работоспособен.

*1.2) Скрининг на продукти, подложени на ускорено тестване с повишен стрес (HASS - Highly Accelerated Stress Screening)* са форма на HALT тестовете. HASS тестът представлява непрекъснат скрининг, прилаган на редовни продукти. Идеята не е да се повреди продуктът, а по-скоро да се подsigури, че продуктите работят правилно, когато са подложени на последователна промяна на променливите на средата.

*1.3) Контрол на продукти, подложени на ускорено тестване (HASA - Highly Accelerated Test Auditing).* За продукция с голям обем, при която с увеличаване на обема, степента на дефекти намалява, трябва да се приложи HASS или HASA одит. Ключовите параметри се наблюдават, за да се определи дали HASA тестът е подходящ за стабилизиране на процеса и дали степента на откази е целта му.



Първата стъпка е да се напише HASA план. Това ще маркира решаващите параметри, напр. степен на преместване на отказите, алфа и бета риск нива и размер на мострата.

2) *Тест за демонстриране на надеждност (RDT- Reliability Demonstration Test)*. RDT е тест за демонстрация на надеждността на продукта. Обикновено се извършва на системно ниво и критерият е да е успешен. Тестът помага да се предскаже времето за нормална работа на продукта и определя дали той има готовност да бъде пуснат на пазара.

3) *Постоянен тест за надеждност (ORT On-Going Reliability Testing)* се използва за установяване на тенденцията в надеждността по време на тестване. ORT може да открие слабости и дефекти, които възникват с времето. ORT е много подобен на RDT, но RDT се изпълнява само веднъж, за да допусне продуктите до пазара, докато ORT е непрекъснат тест на продукти от производствената линия.

б) *Специални тестове:*

1) *Тест на електрическия дизайн на устройството (Electrical Design Verification Test – EDVT)* е тест, който се прилага след фазата на проектиране на продукта, преди да влезне във фаза на производство. Главната му цел е да се открият възможните слабости на дизайна - схема, платка, разположение на компонентите, охлаждане и др., чрез потвърждаване на правилното му действие в условия на стрес. Този тест също е и част от проверката на доставчиците на компоненти. Базирайки се на него, се създава лист с одобрените доставчици на компоненти AVL (Approved Vendor List). EDVT се състои от комплект от диагностични и функционални тестове, които се стартират в условия на повишени натоварвания, създадени чрез промяна на параметри като температура, захранващо напрежение и честота на тактовите генератори в различни комбинации и симулиращи най-добрия и най-лошия сценарий, който може да се развие в реални условия. Продължителността на теста може да отнеме от дни до седмици и всеки отказ се анализира най-подробно за да се открие коренната причина за него.

2) *Тест при замяна на компонент (Material Change Verification Test – MCVT)* е тест, изпълняван при смяна на компонент на устройството във фаза производство с такъв от друг тип или друг производител. MCVT е много подобен на EDVT по начина, по който се изпълнява. Той има връзка с AVL и когато компонент мине MCVT теста, вече може да влезе в списъка на одобрените доставчици.

Основните разлики между двата теста са:

- цел на теста – целта на MCTV е да оторизира нови доставчици на компоненти, блокове и дори цели продукти.

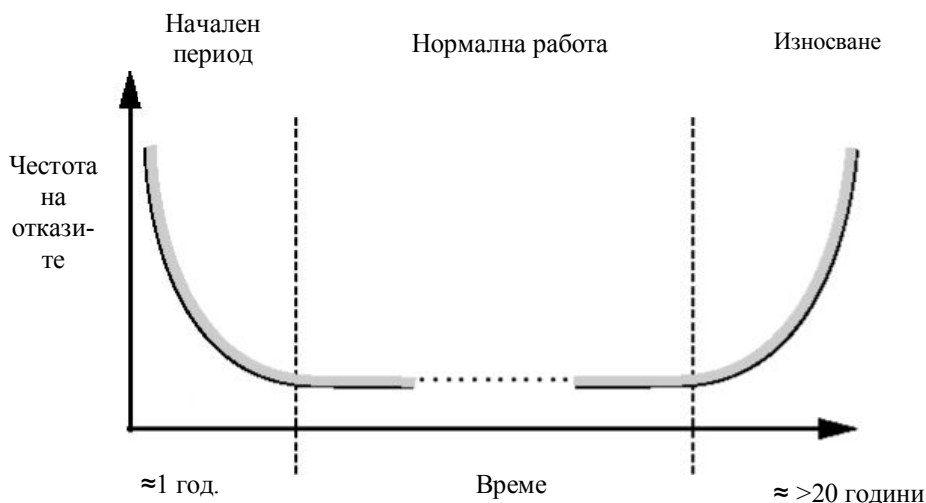
- фаза на провеждане на теста – MCVT се извършва по време на производствената фаза.

- продължителност на теста и брой параметри, които ще бъдат променяни – продължителността на MCVT може да продължи от часове до дни, а броят на контролираните променливи понякога може да бъде намален, обикновено заради цената на оборудването.

3) *Тестване на електрическа безопасност (High Potential Test - Hi-Pot)*. Този тест се изпълнява с високо напрежение (обикновено над 1000V) и целта му е да сертифицира изолацията на устройството в съответствие с регулиращите документи.

#### 4) *Изкуствено стареене (Burn-in test)*

Това е процес, при който устройството работи в стресова околна среда при повишена температура в специални камери. Стремехът е да се открият тези специфични компоненти, които биха отказали в резултат на първоначална висока степен на отказите в експлоатацията на изделието. Вероятността за отказ се моделира математически с функция на Вейбул и се изразява графично чрез кривата на „ваната“ (фиг. 41). Областите на кривата се характеризират с период на първоначална висока степен на откази, период на ниска степен на откази по време на нормалната експлоатация и ново увеличение на отказите в края на живота на изделието, дължащо се на износване.



Фиг. 41 . Крива на ваната

Ако периодът на загряване е достатъчно дълъг, след този тест системата ще е считана за надеждна откъм ранни откази. За електронните устройства, *Burn-in* тестът често се съпровожда с повишена температура и с повишено напрежение.

в) *Тестове, специфични за продукта (Product Specific Tests – PST)*. Това са истински функционални тестове и обикновено зависят от особеностите на продукта.

г) *Вътрешен тест (Built-in Self Test – BIST)*. Това е концепция за вграждане на допълнителни хардуерни и софтуерни възможности в интегралните схеми, които да им позволяват да изпълнят собствено тестване, т.е. тестване на техните собствени операции, използвайки техните собствени схеми, като по такъв начин се намалява зависимостта от външно автоматизирано тест оборудване. Чрез BIST електрическото тестване на чипа се прави по-лесно, по-бързо, по-ефективно и по-евтино.

## **V. 2. Тест стратегии**

Разработването на стратегии за тестване на електронните продукти е толкова важно, колкото и разработването на самия продукт. Това е комплексна задача и трябва да се вземат предвид всички аспекти на продукта, включително неговия дизайн, сложността и типа на платката, предназначението на устройството. Необходимо е разработването на тест стратегията да започне в самото начало да проектирането на продукта и да се извършва паралелно с него [52]. В някои случаи може да се наложи елементи на дизайна да бъдат променени, за да се постигне по-добър обхват на тестовете при по-малки разходи и по-голяма ефективност. Най-подходящото време за това е фазата на проектиране на продукта.

Създаването на тест стратегия е комплексна задача, започваща в началото на проекта и преминаваща през всичките му фази, докато се стигне до фазата на производството. Тук са обособени основните елементи на тест стратегията и те включват следните фази :

- *Фаза 1: Анализ на асемблираната платка и финансов анализ.* Една от първите задачи е анализ на предлагания продукт. Тестовете ще се базират на компонентите и технологиите, използвани в този продукт, неговата сложност, компактността на платката, броя програмируеми компоненти, особеностите на устройството и др. На тази фаза се прави и финансов анализ на оборудването, необходимо за провеждане на тестове.
- *Фаза 2: Избор на подходящо тест оборудване.* На този етап се определя необходимото тест оборудване, като с приоритет е използването на наличното тестово оборудване. Знанията за технологиите, използвани в предложения продукт, дават възможност да се направи оценка на наличното оборудване и да се подсигури, ако е подходящо за тестването на новия продукт.
- *Фаза 3: Създаване на тест стратегия.* С оглед определянето на целите, които трябва да се достигнат, е необходимо да се

създаде тест стратегия, която да осигурява оптимално покритие за тестовите, при минимално дублиране и разумно обоснована цена.

- *Фаза 4: Осъществяване на тест стратегията.* Веднъж създадена, тест стратегията трябва да бъде следвана през целия цикъл на живот на продукта – от неговото създаване до фазата на стабилно производство на продукта. Чрез подходяща тест стратегия на продукта, обхватът на теста нараства до ниво, на което много малко откази могат да се пропуснат. По този начин се създават продукти с по-високо качество при по-ниска себестойност.
- *Фаза 5: Анализ на изходните данни.* Ключов елемент на всяка тест стратегия, е събирането и анализът на получените данни. По този начин могат да се видят и отстранят най-често появяващите се откази, както и да се забележи тенденцията им в началните етапи. Само чрез събиране и анализ на резултатите, могат бързо да се направят подобрения и да се намали влиянието на дефектите върху по-голям брой продукти. Така се подобрява процесът на производство и се намалява себестойността на продукта.
- *Фаза 6: Подобряване на тест стратегията, ако е необходимо.* Един от важните елементи на всяка тест-стратегия е мониторингът на продуктите. Тя трябва да бъде гъвкава и адаптивна и да дава възможност за промени, даже и след като е започнало производството им. Тест-стратегията трябва да може да отговори на всички бъдещи промени в продуктите по време на техния живот, за да бъдат лесно и ефективно тествани.

В табл. 7 е даден пример за тестове, прилагани в различните фази на производството на даден продукт. В първоначалната фаза се прилага структурен тест ICT, при който се откриват най-голям брой дефекти К. Следва прилагане на тестовите AOI, X-Ray, FP, след които броя на дефектите К намалява. На етапа на интегриране на устройството започват да се прилагат функционалните тестове FT1, FT2, ..., FTn, като стремежът е минимизиране на К.

В примера се приема, че реалните дефекти в едно изделие са краен брой N и че се откриват съпоставим брой на всяка фаза от производството. Структурните тестове откриват голяма част от възможните дефекти N, като най-голяма е идентификацията им при ICT-теста. Следователно, ако в началната фаза на създаване на продукта, се направи стратегия за тестване с по-голямо покритие, ще останат по-малко неразкрити дефекти за следващите фази от живота на продукта, което значително ще намали цената на изделието. Популярно е така нареченото „правило 10”, според което, на всеки следващ етап от производството, разходите за откриване и

отстраняване на дефектите са 10 пъти по-високи отколкото на предишният.

Таблица 7. Примерно тестване.

| Фаза на производство                            | Тест за изпълнение                            |
|---|---|
| платка, наситена с компоненти от долната страна | Автоматичен оптичен тест (AOI)                |
| платка, наситена с компоненти от горната страна | Автоматизиран оптичен тест (AOI)              |
| Инсталирани конектори                           | Подвижна сонда (FP)                           |
| BGA инсталирани                                 | Автоматизиран тест с рентгенови лъчи (AXI)    |
| FPGA инсталирани                                | JTAG гранично сканираща логика и програмиране |
| Инсталирани електронни модули                   | Тест на електрическите връзки (ICT)           |
| Интегриране на устройството                     | Функционални тестове (FTn)                    |

От направеното проучване на автоматизираните тестове се вижда, че някои тестове изискват твърде много време за идентифициране на дефектите на продукта. Този проблем се решава с ускорените тестове. Концепцията на ускореното тестване е да компресира времето и да ускори механизмите на отказ в приемлив тестов период, така че надеждността на продукта да може да се оцени. Единственият начин да се ускори времето е като се създадат условия за отказ на потенциално ненадеждните изделия. Тези откази могат да бъдат електрически и механични. Отказ настъпва, когато натоварването надвиши здравината на продукта. Увеличавайки натоварването, се увеличава ненадеждността и се повишава шанса за поява на отказ в по-кратък период от време. Използва се фактор на ускорение  $A$ , който се дефинира математически чрез (5):

$$A = t/t' \quad (5),$$

където  $t$  е животът на изделието до отказ при нормални условия на работа, а  $t'$  е времето при ускорени тестови условия. Тъй като целта на ускорения тест е да доведе до отказ изделието за кратък период от време, то животът му при нормални условия е обикновено по-дълъг

отколкото при ускорени тестови условия и следователно  $A$  има стойност доста по-голяма от единица.

Често факторите на ускорение се моделират, а ускореният тест се използва широко за правене на прогнози. Прогнозите могат да бъдат ограничени, когато се тестват малки по размер извадки, а от друга страна прогнозите могат да бъдат неправилно базирани върху допускането, че резултатите от теста за живота на изделието представят цялата серия.

Механизмите на отказ влияещи върху дългосрочната надеждност на продуктите са относително силно зависими от температурата, влагата, работното напрежение (ток) и механичните вибрации. Чрез ускорените верификационни тестове се ускоряват специфични режими/механизми на отказ, а чрез анализ на данните, може да се прогнозира отказите в реално време и с голяма точност. Софтуерните приложения, използвайки тези данни, могат да изчислят вероятността даден компонент да откаже, а това намалява себестойността на продукта чрез минимизиране на времето/разходите за дебъгиране и намаляване на необходимото оборудване.

## АБРЕВИАТУРА

ADC - Analog-to-Digital Converter - аналогово-цифров преобразувател  
ALU – Arithmetic Logic Unit - аритметико-логично устройство  
AOI - Automatic Optical Inspection  
ASIP - Application Specific Instruction Set Processor – процесор със специфична за приложението система инструкции  
ASIC - Application Specific Integrated Circuit - интегрирани вериги със специфични приложения  
AXI - Automated X-Ray Inspection  
AVL - Approved Vendor List  
BCM - Base Clock Module – модул за базова честота  
BCS - Boundary Scan Cell - граничните сканиращи клетки  
BDM - Background Debug Module – модул за вътрешносхемна настройка  
BIST - Built-in Self Test  
BIPS - Billion Instruction Per Second – милиард инструкции в секунда  
BOR - Brown-Out Reset - модул за детектиране на захранващото напрежение  
BP - Branch Prediction - предсказване на преходите  
BSL - Boundary Scan Logic - гранично сканираща логика  
BSP – Buffered Serial Port  
BTB - Branch Target Buffer - буфер на целевия преход  
CAD - Computer-Aided Design – софтуер за инженерен дизайн  
CAN - Controller Area Network – комуникационен интерфейс между микроконтролери без участие на “главен” MCU  
CCP - CAPTURE/COMPARE/PWM – модули за ф-ии IC, ОС и PWM.сигнали  
CISC - Complex Instruction Set Code – микропроцесори със сложна система инструкции  
CPI - Clock Cycles Per Instruction - среден брой тактове за изпълнение на една инструкция  
CPU – Central Processor Unit – ядро  
CS – Chip Select – сигнал за избор на чип  
CSSU- Compare Select and Store Unit  
CTM – Configurable Timer Module - конфигурируем таймерен модул  
DAC - Digital-to-Analog Converter – цифрово-аналогов преобразувател  
DAG - Data Address Generator  
DBG – Debugger – дебъгер  
DCO – Digitally Controlled Oscillator - цифрово-управляем генератор  
DFT - Design For Test  
DMA – Direct Memory Access - директен достъп до паметта  
DSK – Development Starter Kit – демо модул с MCU за обучение  
DSP – Digital Signal Processor - цифровият сигнален процесор  
EC – Embedded C – език C за вградени микроконтролери  
EDVT - Electrical Design Verification Test

EEPROM - Electronically Erasable Programmable Read-Only Memory - електрически програмируема памет  
EPIC - Explicitly Parallel Instruction Computing - процесор с паралелно изпълнение на инструкциите  
EPROM - Erasable Programmed Read-Only Memory – програмируема памет, която се изтрива с помощта на ултравиолетова светлина  
EVB – Evaluation Board – развойна система  
FET - Flash Emulation Tool  
FIR - Finite Impulse Response  
FLASH – флаш-памет - енергонезависима препрограмируема памет  
FP – Flying Probe  
GPP - General Purpose Processor – процесори с общо предназначение  
GPT – Global Purpose Timer – таймер с общо предназначение  
GSC - Global Serial Communication - глобален последователен интерфейс  
HASA - Highly Accelerated Test Auditing  
HASS - Highly Accelerated Stress Screening  
HALT- Highly Accelerated Life Testing  
HF – High Frequency – режим с висока тактова честота  
HPI – Host Port Interface  
HWM – Hardware Multiplier – апаратен умножител  
IA-32/64 – Intel Architecture – 32-/64 - битова архитектурата на Intel  
IAP - In-Application Programming – вътрешно-схемно програмиране  
IAR EW – IAR Systems Embedded Workbench – развойна програмна среда на IAR Systems за MCU MSP430x  
IC – Input Capture – функция за входно захващане  
ICF - Input Capture Flag – флаг за IC събитие  
ICI - Input Capture Input – вход на таймера за ф-я IC  
ICSP - InCircuit Serial Programming protocol – протокол за вътрешносхемно програмиране  
ICT - In-Circuit Test  
IDE – Integrated Development Environment – интегрирана развойна среда  
IEM - Intelligent Energy Manager - интелигентен мениджър на консумацията  
I/O - Input/Output - входно/изходни портове  
IR – Interrupt Request – заявка за прекъсване  
IrDA - Infrared Data Association – протокол за комуникация чрез инфрачервени сигнали  
ISP - In-System Programming - вътрешносхемно програмиране  
I2C - Inter-Integrated Circuit – серийна компютърна шина, свързваща нискоскоростна периферия към дънни платки и др.  
JTAG - Joint Test Action Group – стандарт IEEE 1149.1 за гранично-сканираща логика  
JVM - Java Virtual Machine - виртуална машина на Java  
KCT - Компютърни Системи и Технологии



LCD – Liquid Crystal Display – дисплейс течни кристали  
 LF – Low Frequency – режим с ниска тактова честота  
 L1 – Level1-cashe – кеш от първо ниво  
 LPM – Low Power Mode – режим с намалена консумация  
 LSB – Low Significant Bit – най-младшият значещ бит  
 LVI – Low Voltage Module – модул за наблюдаване на напрежението  
 McBSP – Multichannel Buffered Serial Port  
 MCCI – Multi Channel Communication Interface – многоканален комуникационен интерфейс (MCCI), съдържащ три последователни интерфейса: един SPI и два SCI  
 MCU – MicroController Unit – микроконтролер  
 MCVT - Material Change Verification Test  
 MDA - Manufacturing Defect Analyzer  
 MFLOPS - Million Floating point instruction Per Second милиони инструкции с плаваща запетая в секунда  
 MID - Mobile Internet Device  
 MIPS - Million Instruction Per Second – милиони инструкции в секунда  
 MMX – Matrix Manipulation Extensions - пакетирани данни с фиксирана запетая  
 MP – Microprocessor – микропроцесор  
 MPROM - Mask programmable read-only memory – програмируема чрез маска ROM  
 MRI - Magnetic Resonance Imaging  
 MSP - Mixed Signal Processor – процесор, обработващ цифрови и аналогови сигнали  
 OC – Output Compare – функция за изходно сравняване  
 OCF - Output Compare Flag – флаг за OC  
 OCO - Output Compare Output – сигнал на изхода на таймера при OC събитие  
 ORT- On-Going Reliability Testing  
 OS – Operating system – операционна система  
 OTP – One-Time Programmable – еднократно програмируема памет  
 PC – Personal Computer - персонален компютър  
 PCA - Programmable Counter Array - масив от програмируеми броячи  
 PDA - Personal Digital Assistant – мобилен компютър с дизайн на органайзер  
 Peripherals - ПУ - периферни устройства  
 PIT – Periodic Interrupt Timer - таймер за периодични прекъсвания  
 PLL – Phase Locked Loop - система за фазово автоматично поддържане на честота  
 POE – Problem-Oriented Education - проблемно-ориентирано обучение  
 POR - Power On Reset - автоматично включване при подаване на захранване  
 PWM - Pulse Width Modulation - широчинно-импулсна модулация  
 QSM - Queued Serial Module - буфериран модул за последователен интерфейс

RAM – Random Access Memory - памет за съхраняване на данни  
 RDT - Reliability Demonstration Test  
 RISC - Reduced Instruction Set Code – микропроцесори с редуцирана система инструкции.  
 ROM – Programmable Read-Only Memory – еднократно програмируема памет за четене  
 RTC - Real Time Clock - часовник за реално време  
 SCH - Serial Channel - сериен канал  
 SCI – Serial Communication Interface – сериен асинхронен интерфейс  
 SDRAM - Synchronous Dynamic Random Access Memory – синхронна динамична RAM памет  
 SFR – Special Function Registers - регистри със специално предназначение  
 SHARC - Super Harvard ARChitecture – подобрена Харвардска архитектура  
 SIM – System Integration Module - системен интеграционен модул  
 SIMD – Single Instruction Multiple Data - обработка на множество данни с една инструкция  
 SISD – Single Instruction Single Data – обработка на единични данни с една инструкция  
 SP – Stack Pointer – регистър-указател на стека  
 SPI – Serial Peripheral Interface – сериен периферен интерфейс  
 SPP - Special Purpose Processor – процесор със специално предназначение  
 SSE - Streaming SIMD Extensions – пакетирани данни с плаваща запетая  
 TDMI - Thumb Instruction, Debugger, Multiplier, ICE- поддържане на инструкции Thumb, дебъгер, умножител, ICE ( при ARM CPU)  
 TPU - Time Processor Unit - таймерен копроцесор  
 TTL - Transistor–Transistor Logic – транзисторно-транзисторна логика  
 3D – 3-dimension – три-измерна графика  
 UART – Universal Asynchronous Receiver Transmitter – универсален асинхронен приемо-предавателен интерфейс  
 USART – Universal Synchronous/Asynchronous Receiver Transmitter – универсален синхронен/асинхронен приемо-предавателен интерфейс  
 USB - Universal Serial Bus – универсална серийна шина  
 USCI – Universal Serial Communication Interface – интерфейс за обмен по протоколи UART, IrDA, SPI и I2C  
 VIC – Vector Interrupt Controller - векторен контролер на прекъсванията  
 VLIW - Very Long Instruction Word  
 WDT - Watchdog Timer- стражеви таймер

## ЛИТЕРАТУРА

- [1] Атанасов А., “Основи на импулсната и цифровата схемотехника”, София, 2002г.
- [2] Атанасов А., “Основи на микропроцесорната техника”, БСУ, 2003г.
- [3] Богданов Д., И. Мустакеров, “Език за програмиране С”, Техника, София, 1991г.
- [4] Боровски Б., Б. Янков, “Справочник по изчислителна техника”, София, 1990г., изд. Техника.
- [5] Боровски Б., П. Боровска, “Архитектура на ЕИМ и микрокомпютри”, София, 1992 г., изд. Техника.
- [6] Гук Ю., "Апаратные средства IBM PC. Энциклопедия" Питер Петербург, 2001.
- [7] Закс Р., “Микропроцесори – от чипове до системи”, С., Техника, 1980.
- [8] Иванов Р., Г. Михов, “Електронно-цифрови устройства и системи”, София Техника, 1990г.
- [9] Иванов Р., О. Асенов, “Архитектура и системно програмиране на Pentium базирани компютри”, Велико Търново, изд. Абагар, 1998г.
- [10] Каракехайов З., Е. Саръмов, “Приложни микрокомпютърни системи”, Технически университет - София, 1995г.
- [11] Каракехайов З., К. Кристенсен, О. Винтер, “Проектиране на вградени микрокомпютърни системи с микроконтролери”, Пенсофт, София – Москва, 2000г.
- [12] Каракехайов З., С. Григоров, “Едночипови микрокомпютри”, 1992г.
- [13] Керезов А., “Ръководство за лабораторни упражнения по микропроцесорна схемотехника”, ТУ- София, 2000.
- [14] Колев Н.и др., “Електрически измервания”, София, Техника, 1989.
- [15] Левентал Л., “Въведение в микропроцесорите”, София, Техника, 1982.
- [16] Ленк Д., “Наръчник по микропроцесори, микрокомпютри и миникомпютри”, София, Техника, 1983.
- [17] Луканчевски М., “Системно програмиране за едночипови микрокомпютри”, Техника, 1993г., София.
- [18] Михов Г., “Контрол и диагностика на микропроцесорни системи”, Технически университет – София, 1994г.
- [19] Михов Г., “Цифрова схемотехника”, ТУ- София, 1997 г.
- [20] Николов Л., “Операционни системи”, изд. Сиела, София, 1998 г.
- [21] Ремизевич Т.В., “Микроконтроллеры для встраиваемых приложений. От общих подходов к семействам HC05 и HC08 фирмы Motorola”. М.: Додэка, 2000. 272 с.
- [22] Ружеков Г., “Сигнални процесори – особености и приложения”, Автоматика и информатика, бр.4, стр. 29 –34, 2005.
- [23] Смикаров А. И др., “Едночипови микрокомпютри”, 1999г.
- [24] Станчев В., Е. Бакърджиев, “Микропроцесорна система CM 600”, София, Техника, 1984.

- [25] Стоянова Р., А. Егоров, “Програмиране на асемблерен език за 32-битови микропроцесори”, изд. Парафлоу, София, 1998г.
- [26] Томов И. и др., “Микропроцесорни управляващи системи”, С., Техника, 1986.
- [27] Хвощ С., Т. Варлинский, Е. Попов., “Микропроцесори и Микро ЕИМ в системите за автоматично управление”, Ленинград “Машиностроене”, 1987г,
- [28] Application Report SLAA120 -2000, TI, “PWM DC Motor Control Using Timer\_A of the MSP430x”.
- [29] Application Report SLAA205 – June 2004, “MSP430 Competitive Benchmarking”.
- [30] Application Reports, TI, “MSP430 Family Mixed-Signal Microcontroller”, 2003.
- [31] Halstead, M. “Elements of Software Science”, Elsevier North Holland, 1997.
- [32] Hamrita T., W. Potter, B.Bishop, University of Georgia, Athens, “Robotics, Microcontroller, and Embedded Systems Education” Initiatives at the University of Georgia An Interdisciplinary Approach.
- [33] Hristov V., “Using the PIC16C84 in Pulse Generator for Educational Purposes”, FMNS-2005, 9-11 June, Blagoevgrad, Bulgaria.
- [34] Ismail R., I. Sayers, A. Robson, A. Adams, G. Chester, “Principles of Microprocessors”, 1991.
- [35] Ismail R., M. Slater, “A Guide to RISC Microprocessors”, 1992.
- [36] MC68HC11A8 HCMOS Microcontroller, Motorola Inc., 1990.
- [37] MC68HC11, Reference Manual, Motorola Inc., 1991.
- [38] Singer M., “TI Eyes Embedded FRAM for Memory”.
- [39] Soffe V., “The evolution of Flash Microcontrollers”, MicroController Pros Corporation, April 2003.
- [40] Williston K., BDTI, “Inside DSP on Low Power: Processors for Low-Power Signal Processing”, 2004.
- [41] [www.amd.com/gb-uk/Processors/ProductInformation](http://www.amd.com/gb-uk/Processors/ProductInformation)
- [42] [www.analog.com/en/analog-microcontrollers/ADUC7019/products](http://www.analog.com/en/analog-microcontrollers/ADUC7019/products)
- [43] [www.atmel.com/products/AVR/default\\_xmega.asp](http://www.atmel.com/products/AVR/default_xmega.asp)
- [44] [www.8052.com/tut8051.phtml](http://www.8052.com/tut8051.phtml)
- [45] [www.freescale.com/webapp/sps/site/homepage.jsp](http://www.freescale.com/webapp/sps/site/homepage.jsp)
- [46] [www.hw-group.com/application\\_notes\\_en.html](http://www.hw-group.com/application_notes_en.html)
- [48] [www.intel.com/Consumer/Learn/Desktop/core2quad-detail.htm](http://www.intel.com/Consumer/Learn/Desktop/core2quad-detail.htm)
- [49] [www.motorola.com/content.jsp?globalObject](http://www.motorola.com/content.jsp?globalObject)
- [50] [www.ti.com/\(focus.ti.com/docs/prod/folders/print/msp430f149\)](http://www.ti.com/(focus.ti.com/docs/prod/folders/print/msp430f149))
- [51] [www.radio-electronics.com/info/t\\_and\\_m/design-for-testability-dft/basics-guidelines.php](http://www.radio-electronics.com/info/t_and_m/design-for-testability-dft/basics-guidelines.php)
- [52] [www.radio-electronics.com/info/t\\_and\\_m/ate/developing-testing-automation-strategy.php](http://www.radio-electronics.com/info/t_and_m/ate/developing-testing-automation-strategy.php)