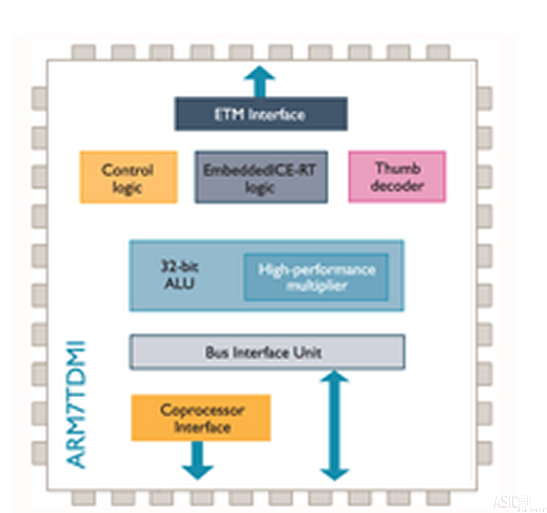
Организация на централния процесор с АRМ7 архитектура, програмен модел и режими на работа.

Архитектура:



ARM 7 има Вон Нойманова архитектура в една 32 битова шина едновременно за данни и инструкции.

Данните могат да бъдат 8bit (byte), 16 bit (halfword) или 32 (word), като те трябва да са на последователни клетки памет (4,2 или 1). Като байт може да бъде поставен на всяка една свободна клетка.

Режими на работа:

User mode - стандартен режим на работа, който се използва за изпълнение на програми.

Fast interrupt (FIQ)- режима поддържа трансфер на информация или канален процес

Interrupt (IRQ) - този режим се използва за стандартно управление на прекъсванията.

Supervisor - защитен режим за операционната система

Abort mode - в този режим влиза когато има информационен(data) prefetch abort или instruction(инструкция) prefetch abort.

System mode - системен режим - представлява привилигирован потребителски режим за операционната система.

Undefined mode - неопределен режим - в него влиза когато е изпълнена недефинирана инструкция

Всички режими на работа заедно освен User mode се наричат privileged modes ( привилегировани режими) Те се използват за обслужване на прекъсвания, изключения или за достъпване на защитени ресурси.

lpc -ARM-book.pdf

**The ARM 7 CPU Core.**

Конвейрна система: 3 стъпки.

**Fetch, decode and execute**

Това е най-простият модел на конвейерна система от такъв вид. При този вид конвейерна система са отстранени много проблеми на по-сложните pipeline-i с повече стъпки като например - проблема - прочети преди да пипеш.

Стъпките са независими от хардуера и така изпълняват една инструкция докато декодират втора и взимат трета. Този pipeline забързва магистралата на процесорните инструкции толкова ефективно, че повечето инструкции АРМ могат да бъдат изпълнени за 1 цикъл. Pipeline-a работи най-ефективно при линеен код. Когато е открито разклонение, pipeline-a се изчиства и трябва отново да се напълни за да може да продължи изпълнението. Инструкциите в АРМ имат интересни свойства които помагат за изглаждане на малки скокове в хода за да открие най-добрия поток на кода през **конвейерната система (pipeline).** Тъй като конвейрната система е част от процесора програмиста няма никакъв достъп до нея. Въпреки това трябва да се има предвид, че програмният броч (PC) е 8 байта по-напред от текущата изплъняваща се инструкция. Това трябва да се предвиди като се изчисляват offsets (отклонения) при относителното адресиране на програмният брояч.

Например инструкцията :

0x4000 LDR PC, [PC, #4]

ще зареди съдържанието на адрес PC+4 в PC. И тъй като програмният брояч е на 8 байта по-напред в съдържанието на адрес 0х400C ще бъде заредено в паметта, а не 0х4004 каквато е презумцията.

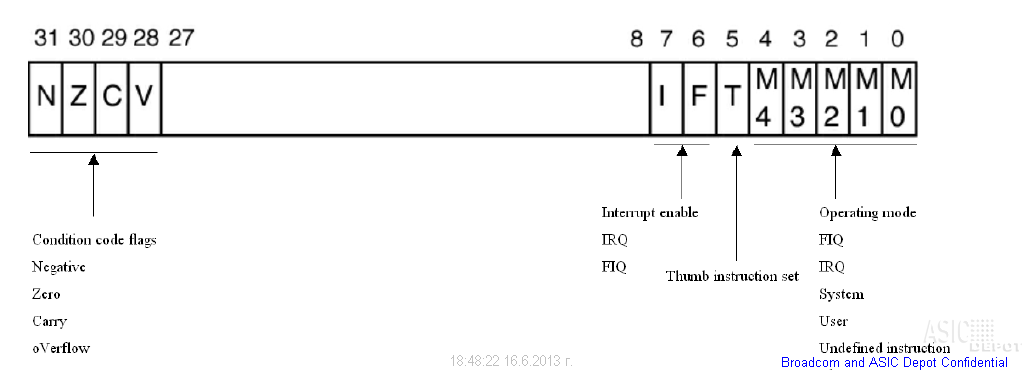
Регистри:

Архитектурата на арм 7 работи така, че зада може да се изпълни инструкция с информация, информацията трябва първо да бъде преместена от мястото в паметта в централен набор от регистри. След като е изпълнена тя отново се връща в паметта.

Централният набор от регистри представлява банка от 16 регистъра R0 - R15.(15 + PC) Всеки един от тези регистри е 32 бита. R0 - R12 са потребителски регистри и те нямат друга определена функция. Другите R13- R15 имат запазена функция в процесора:

* R13 e указател(pointer)на стека
* R14 се нарича свързващ регистър (link register). Когато е извикана фукнция адреса за връщане е автоматично се записва в него. Така в момента в който функцията се върне адреса се взима от там без допълнителни цикли. Това позволява бързо прескачане и връщане от друг leaf ( функция която няма да вика други функции). Ако извиканата функция е част от клон (branch - функция която ще викне други функции) адреса се запазва в стека R13.
* R15 e програмният брояч.

Освен банката с регистрите съществува и още една допълнителна 32бита, която се нарича CPSR - Current Program Status Register - регистър за статуса на конкретната програма. Този регистър съдържа флагове които дават информация за статуса контролират работата на ARM7 процесора.



Горните четири бита на CPSR кодове за състоянието, които са зададени от процесора. Тези кодове отчитат резултата от операцията с обработка на данни. От тези кодове може да се види дали инструкцията за обработна на данни е генерирала положителен, нула, пренос или преливане(overflow)

Битове 6 и 7 съдържат информация за прекъсвания тип съответно IRQ и FIQ. (fast interrupt и interrupt )

ARM7 процесора може да изпълни 2 типа инструкции :

- ARM инструкцията -32bit

- И THUMB инструкция - 16bit.

T bit-a в CPSR определя кой тип инструкции ще се използват.

THUMB имплементира 16 битови инструкции на 32 битова архитектура като по този начин се постига:

1. По-голяма производителност от 16 битова архитектура

2. По-голяма гъстота на кода от 32 битова архитектура

В основата си THUMB представлява извадка на най-използваните 32 битови инструкции за ARM. Всяка 16bit Thumb има отговаряща на нея 32 битова АРМ инструкция, която има същия ефект върху процесорния модел.

Формат на паметта:

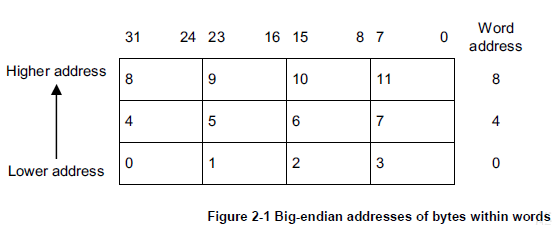
ARM7 вижда паметта като линеен сбор от байтове подредни във възходящ ред от 0.

* 0-3 - първата запазена дума
* 4-7 - втората запазена дума
* 8-11 третата запазена дума

АRM7 може да третира думите записани в паметта по един от следните начини:

* Big-endian format -

В този формат процесора записва старшия бит(most significant bit) в байта с най-малък номер в паметта, a младшият (least significant bit) в байта с най-голяма стойност



* Little-endian format -

Обратно на Big-endian. Най-малкият пореден байт в думата е младши а най-големият байт се третира като старши.

