

Author > Ômar Fontenele a.k.a f0nt_Drk

Date > 10/03/2011

Understanding the ARM Architecture.



Breve Introdução da Arquitetura ARM

Começando com uma linha básica de raciocínio e ideia vamos iniciar com o simples conceito de sistema operacional, que espero eu, todos já estejam cansados de ouvir, ler e concerteza, saber .

A ARM (antes: Acorn RISC Machine, atualmente: Advanced RISC Machine) é a arquitetura de um processador de 32 bits. Muito famoso por sua característica de oferecer um alto desempenho em pouco espaço requerido pela máquina, Observando isto percebemos a ligação entre ARM e RISC (Reduced Instruction Set Computer) que vem a ser um conjunto reduzido de instruções em um computador . Na época do início da ARM a maioria dos processadores precisavam de apenas 13 bits para alocar todas as constantes de um programa, mas, se uma vez utilizada em um sistema com 32 bits ela já estaria armazenada em 'cache' para que pudesse ser executada mais rapidamente da próxima vez. Logo isto define o porque a Arquitetura ARM trabalha com 32 Bits .

Registradores :

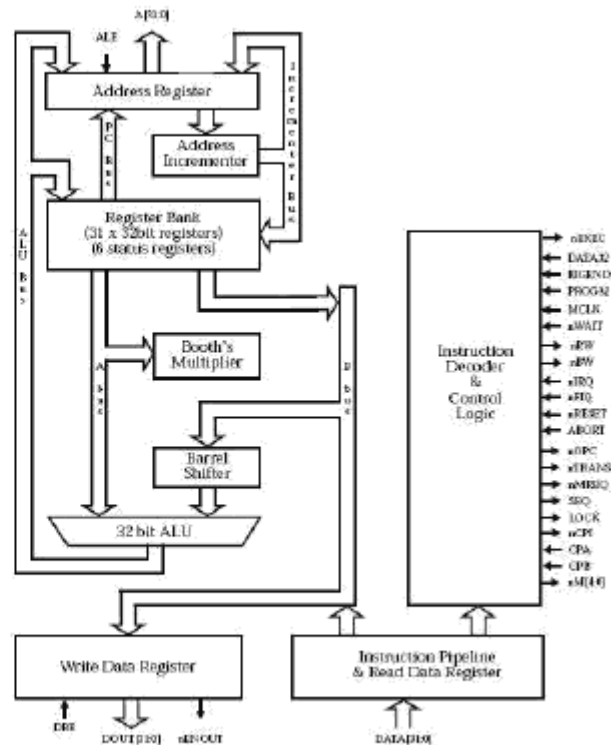
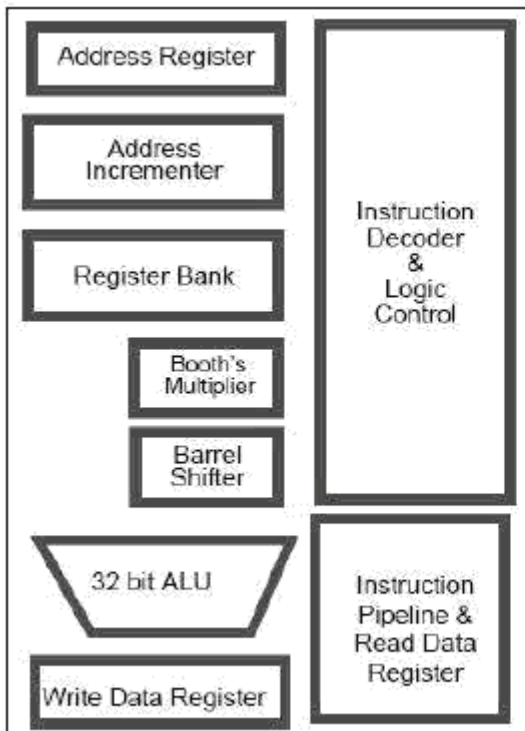
Além de Trabalhar também com 16 Registradores que os principais são :

- PSR Program Status Register
- SP Stack Pointer
- PC Program Counter
- LR Link Register

Declarados como mais principais ainda são os :

- Link Register (LR) -R14- > Cujo Contém o endereço da próxima instrução a ser executada, após uma sub-rotina - Branch And Link -
- Program Counter (PC) -R15- > Cujo Contém também o endereço da próxima instrução a ser executada, mas pelo Processador .

Instruções .



Lembre-se que a arquitetura ARM foi desenvolvida para dar mais velocidade e consumir menos energia . Por isso carrega a linha de processadores RISC , que logo recebe o adjetivo de 'Instruções Reduzidas'. Mas tomemos cuidado ao dizer isto, pois quando nos referimos a Instruções Reduzidas, não quer dizer que propriamente ditos os numeros de instruções são menos do que o de um CISC por exemplo, o que acontece é que as instruções são executadas com no maximo um ciclo de memoria além de obter muito menos dados carregados nas instruções. Mas muitos RISC's tem até mais instruções que alguns CISC's Outras características das instruções são :

- Modos de endereçamento Simples.
- Arquitetura LOAD / STORE, assim as operações interagem com o conteúdo dos Registradores, não com o da memoria diretamente
- Tamanho fixo dos campos das instruções.
- Instruções múltiplas de LOAD / STORE para melhorar a performace .

Continuando o assunto de Instruções designaremos seus tipos .

Classificando-as em 6 classes, muitas vezes denominadas de "Large Classes"

Instruções de Branch > Troca de fluxos de instruções escrevendo endereços dentro do Program Counter (PC)

Instruções de Processamento de Dados > Responsavel por instruções logicas . ex : instruções aritmeticas .

Vejam todas instruções logo mais abaixo :

Opcode	Mnemonic	Operation	Action
0000	AND	Logical AND	$Rd := Rn \text{ AND shifter_operand}$
0001	EOR	Logical Exclusive OR	$Rd := Rn \text{ EOR shifter_operand}$
0010	SUB	Subtract	$Rd := Rn - \text{shifter_operand}$
0011	RSB	Reverse Subtract	$Rd := \text{shifter_operand} - Rn$
0100	ADD	Add	$Rd := Rn + \text{shifter_operand}$
0101	ADC	Add with Carry	$Rd := Rn + \text{shifter_operand} + \text{Carry Flag}$
0110	SBC	Subtract with Carry	$Rd := Rn - \text{shifter_operand} - \text{NOT}(\text{Carry Flag})$
0111	RSC	Reverse Subtract with Carry	$Rd := \text{shifter_operand} - Rn - \text{NOT}(\text{Carry Flag})$
1000	TST	Test	Update flags after $Rn \text{ AND shifter_operand}$
1001	TEQ	Test Equivalence	Update flags after $Rn \text{ EOR shifter_operand}$
1010	CMP	Compare	Update flags after $Rn - \text{shifter_operand}$
1011	CMN	Compare Negated	Update flags after $Rn + \text{shifter_operand}$
1100	ORR	Logical (inclusive) OR	$Rd := Rn \text{ OR shifter_operand}$
1101	MOV	Move	$Rd := \text{shifter_operand}$ (no first operand)
1110	BIC	Bit Clear	$Rd := Rn \text{ AND NOT}(\text{shifter_operand})$
1111	MVN	Move Not	$Rd := \text{NOT shifter_operand}$ (no first operand)

- Instruções para transferência de registradores de status > Possuem duas, sendo elas : MRS ; MSR, a primeira move um PRS para um de nível geral, e o segundo faz o contrario .
- Instruções LOAD / STORE > são divididas em três : faz L/S de um unico registrador ou faz L/S para um Geral e o que troca o endereço do registrador pelo da memoria .
- Instruções de co-processor > são operações que se iniciam no co-processor que pode transferir dados para a memoria e a memoria para ele .
- Instruções de geração de exceções > A instrução SWI realiza uma interrupção no software muitas vezes usada para realizar uma chamada ao SO .

Modos de operações da Arquitetura ARM :

- User > Execução comum de programas e com limitações ao acesso dos Registradores .
- FIQ (Fast Interrupt) > Transferencia Rapida de Dados .
- IRQ (Interrupt) > Interrupções em Geral.
- Supervisor > Modo Restrito/Protegido para com o acesso do Sistema Operacional.
- Abort > Implementa e/ou protege a Memória Virtual .
- Undefined > Suporte para emulação de co-processadores .
- System > A partir da versão quatro a ARM implementou esse modo para executar tarefas de privilegio do Sistema Operacional .

ARM state general registers and program counter

System and User	FIQ	Supervisor	Abort	IRQ	Undefined
r0	r0	r0	r0	r0	r0
r1	r1	r1	r1	r1	r1
r2	r2	r2	r2	r2	r2
r3	r3	r3	r3	r3	r3
r4	r4	r4	r4	r4	r4
r5	r5	r5	r5	r5	r5
r6	r6	r6	r6	r6	r6
r7	r7	r7	r7	r7	r7
r8	r8_fiq	r8	r8	r8	r8
r9	r9_fiq	r9	r9	r9	r9
r10	r10_fiq	r10	r10	r10	r10
r11	r11_fiq	r11	r11	r11	r11
r12	r12_fiq	r12	r12	r12	r12
r13	r13_fiq	r13_svc	r13_abt	r13_irq	r13_und
r14	r14_fiq	r14_svc	r14_abt	r14_irq	r14_und
r15 (PC)	r15 (PC)	r15 (PC)	r15 (PC)	r15 (PC)	r15 (PC)

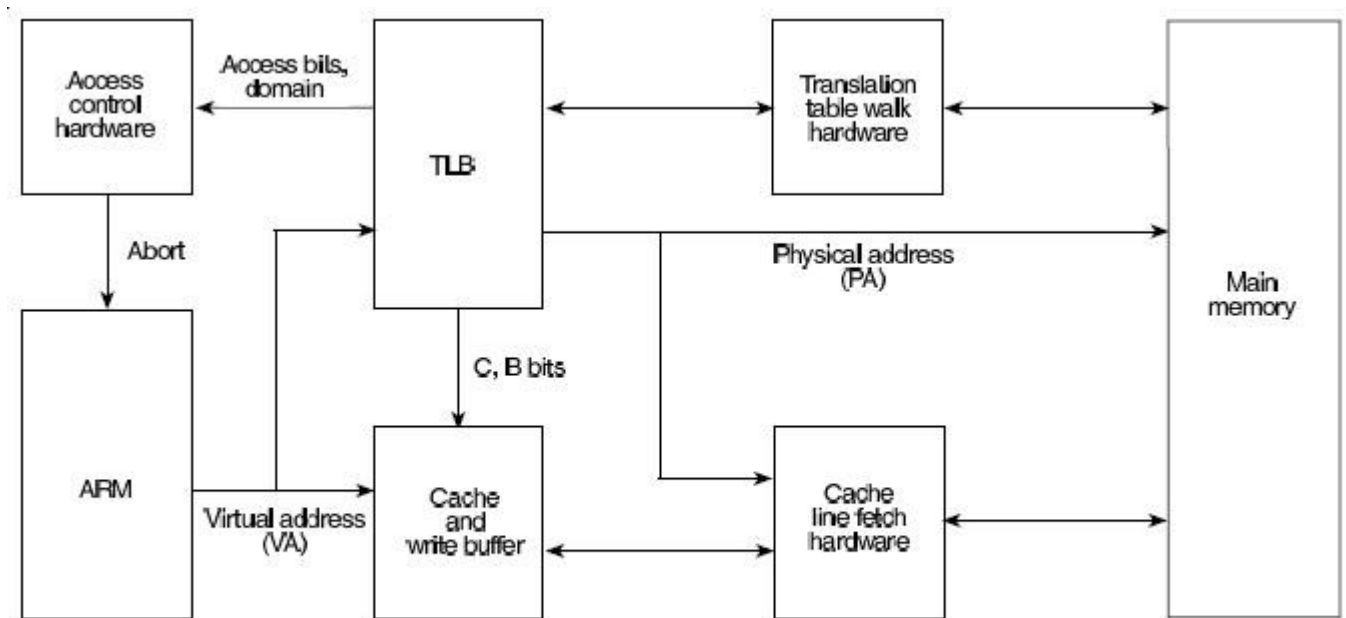
ARM state program status registers

CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
	SPSR_fiq	SPSR_svc	SPSR_abt	SPSR_irq	SPSR_und

Memoria .

Os processadores ARM foram feitos para serem endereçados de byte em byte . Além de ser dividida em duas Partes : a RAM e a ROM .

Na primeira e principal é onde tem-se : Cache de instrução e dados separados, áreas separadas de memória para instruções e dados, sistema DMA para acessar a memória, buffer de escrita e duas micro-TLBs.



Referencias

<http://pt.wikipedia.org/wiki/RISC>

http://pt.wikipedia.org/wiki/Arquitetura_ARM

http://books.google.com.br/books?id=vdk4ZGRqMskC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

- null -

Fiquem na paz , qualquer duvida, estamos ae !

[]'s