

Introdução à arquitetura ARM



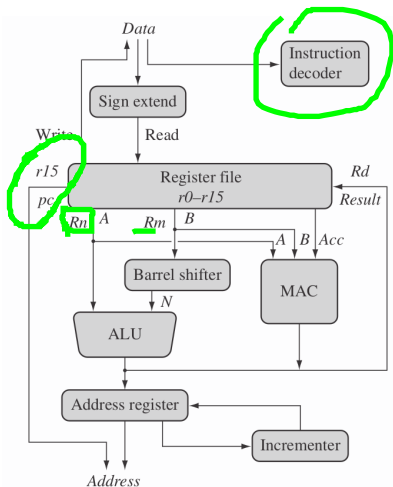
Universidade Federal do Ceará - Campus Quixadá

Roberto Cabral
rbcabral@ufc.br

03 de Dezembro de 2020

Arquitetura e Organização de Computadores II

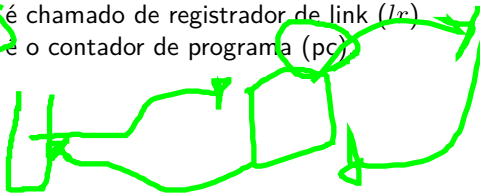
Modelo de fluxo de dados do ARM.



- Arquitetura load-store
- sign extend - conversão de dados para 32 bits.
- R_n, R_m - operandos.
- R_d - resultado.
- MAC - multiply-accumulate unit.
- ALU - arithmetic logic unit.
- Barrel shifter - deslocamento de dados.

Registadores

- Registradores de propósito geral armazenam dados ou um endereço. Eles são identificados com a letra r prefixada ao número do registrador.
- Existem até 18 registradores ativos: 16 registradores de dados e 2 registradores de status do processador. Os registradores são visíveis para o programador como r_0 a r_{15} .
- O processador ARM atribui tarefas específicas à três registradores:
 - Registrador r_{13} é usado como o ponteiro de pilha (sp).
 - Registrador r_{14} é chamado de registrador de link (lr).
 - Registrador r_{15} é o contador de programa (pc).

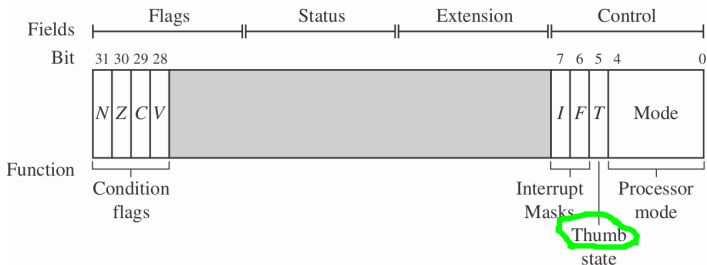


Registradores

- No estado ARM, os registradores r_0 a r_{13} são ortogonais - qualquer instrução que você possa aplicar a r_0 , você pode igualmente aplicar a qualquer um dos outros registradores.
- Além dos 16 registradores de dados, existem dois registradores de status do programa: cpsr e spsr.
- O arquivo de registrador contém todos os registradores disponíveis para um programador.
- Quais registradores são visíveis para o programador dependem do modo atual do processador.

Current Program Status Register

- O ARM usa o *cpsr* para monitorar e controlar as operações internas.
- O *cpsr* é um registrador de 32 bits dedicado e reside no arquivo de registradores.
- O *cpsr* é dividido em quatro campos, cada um com 8 bits: flags, status, extensão e controle.



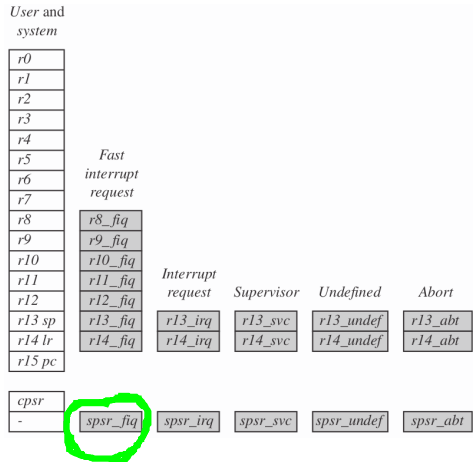
Modos do processador

- O modo do processador determina quais registradores estão ativos e os direitos de acesso ao próprio registrador *cpsr*.
- Os modos do processador podem ser:
 - privilegiado - permite acesso total de leitura/escrita ao *cpsr*
 - não privilegiado - só permite acesso de leitura ao campo de controle no *cpsr*, mas ainda permite acesso de leitura/escrita as flags condicionais.
- Existem sete modos no total: seis modos privilegiados e um não privilegiado.

Modos do processador

- modos privilegiados
 - `abortar` - quando há uma tentativa fracassada de acessar a memória.
 - `solicitação de interrupção rápida` e `solicitação de interrupção` - correspondem aos dois níveis de interrupção disponíveis no processador ARM.
 - `supervisor` - modo em que o processador está após a reinicialização
 - `sistema` - é uma versão especial do modo de usuário que permite acesso total de leitura e gravação ao *cpsr*.
 - `indefinido` - usado quando o processador encontra uma instrução que é indefinida ou não é suportada pela implementação.
- modo não privilegiado
 - `usuário` - é usado para programas e aplicativos.

Banked registers



Mudança de modo

- O modo do processador pode ser alterado por um programa que grava diretamente no *cpsr* ou pelo hardware quando o núcleo responde a uma exceção ou uma interrupção.
- As seguintes exceções e interrupções causam uma alteração de modo:
 - reset;
 - solicitação de interrupção e solicitação de interrupção rápida;
 - interrupção de software;
 - anulação de dados;
 - interrupção de prefetch;
 - instrução indefinida.
- Exceções e interrupções suspendem a execução normal de instruções sequenciais e saltam para um local específico.

Modos de Operação

Mode	Abbreviation	Privileged	Mode[4:0]
<i>Abort</i>	abt	yes	10111
<i>Fast interrupt request</i>	fiq	yes	10001
<i>Interrupt request</i>	irq	yes	10010
<i>Supervisor</i>	svc	yes	10011
<i>System</i>	sys	yes	11111
<i>Undefined</i>	und	yes	11011
<i>User</i>	usr	no	10000

Máscaras de Interrupção

- Máscaras de interrupção são usadas para impedir que solicitações de interrupção específicas interrompam o processador.
- Existem dois níveis de solicitação de interrupção disponíveis no núcleo do processador ARM - solicitação de interrupção (IRQ) e solicitação de interrupção rápida (FIQ).
- O *cpsr* possui dois bits de máscara de interrupção, 7 e 6 (ou I e F)

Flags condicionais

- As flags condicionais são atualizadas por comparações e o resultado de operações da ALU.
- Por exemplo, se uma instrução de subtração *SUBS* resultar em um valor de registrador igual a zero, então o sinalizador *Z* no *cpsr* será definido.

Flag	Flag name	Set when
<i>Q</i>	Saturation	the result causes an overflow and/or saturation
<i>V</i>	oVerflow	the result causes a signed overflow
<i>C</i>	Carry	the result causes an unsigned carry
<i>Z</i>	Zero	the result is zero, frequently used to indicate equality
<i>N</i>	Negative	bit 31 of the result is a binary 1

Execução condicional

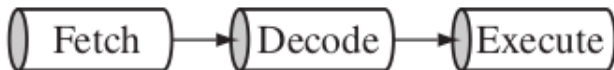
- A execução condicional controla se o processador executará ou não uma instrução.
- A maioria das instruções tem um atributo de condição que determina se o processador irá executá-la com base na configuração da flag de condição.
- Antes da execução, o processador compara o atributo de condição com a flag de condição no *cpsr*. Se eles corresponderem, a instrução é executada; caso contrário, a instrução será ignorada.

Execução condicional

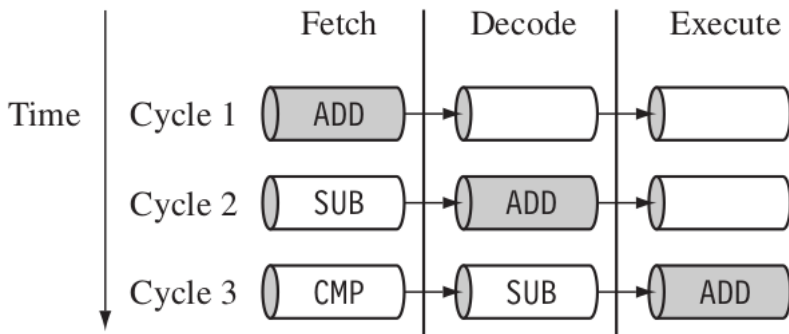
Mnemonic	Name	Condition flags
EQ	equal	<i>Z</i>
NE	not equal	<i>z</i>
CS HS	carry set/unsigned higher or same	<i>C</i>
CC LO	carry clear/unsigned lower	<i>c</i>
MI	minus/negative	<i>N</i>
PL	plus/positive or zero	<i>n</i>
VS	overflow	<i>V</i>
VC	no overflow	<i>v</i>
HI	unsigned higher	<i>zC</i>
LS	unsigned lower or same	<i>Z</i> or <i>c</i>
GE	signed greater than or equal	<i>NV</i> or <i>nv</i>
LT	signed less than	<i>Nv</i> or <i>nV</i>
GT	signed greater than	<i>NzV</i> or <i>nzv</i>
LE	signed less than or equal	<i>Z</i> or <i>Nv</i> or <i>nV</i>
AL	always (unconditional)	ignored

Pipeline

- Um pipeline é o mecanismo que um processador RISC usa para executar instruções.
- O uso de um pipeline acelera a execução buscando a próxima instrução enquanto outras instruções são decodificadas e executadas.
- Uma maneira de ver o pipeline é pensar nele como uma linha de montagem, com cada estágio realizando uma tarefa específica para fabricar o produto.



Pipeline



Pipeline

- A medida que o comprimento do pipeline aumenta, a quantidade de trabalho realizado em cada estágio é reduzido.
- Isso, por sua vez, aumenta o desempenho.
- A latência do sistema também aumenta porque são necessários mais ciclos para preencher o pipeline antes que o núcleo possa executar uma instrução.
- O aumento do tamanho do pipeline também significa que pode haver dependência de dados entre determinados estágios.
- Você pode escrever código para reduzir essa dependência.

Pipeline

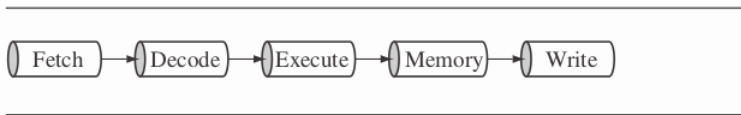


Figure 2.9 ARM9 five-stage pipeline.

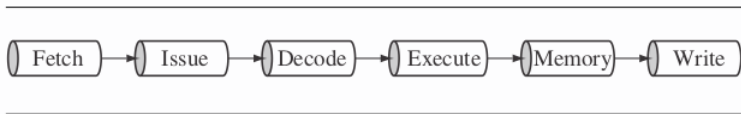
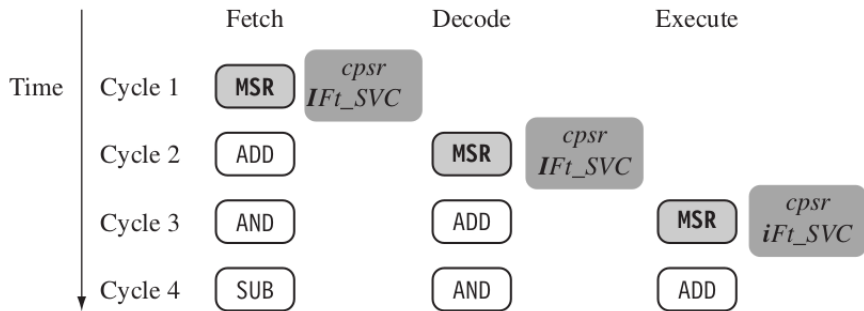


Figure 2.10 ARM10 six-stage pipeline.

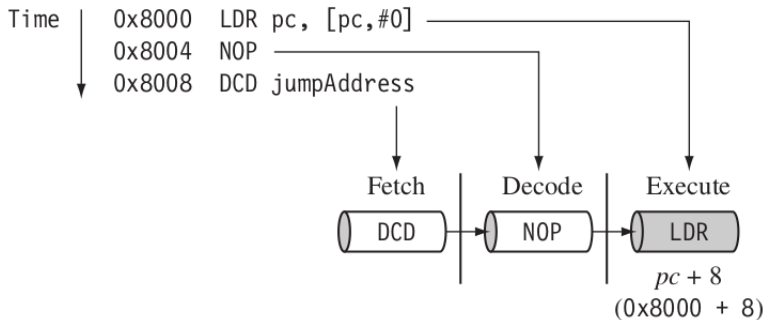
Pipeline

- O pipeline do ARM não processa uma instrução até que ela passe completamente pelo estágio de execução.
- Por exemplo, um pipeline ARM7 (com três estágios) executa uma instrução somente quando a quarta instrução for buscada.



Pipeline

- Devido ao Pipeline (busca, decodificação, execução), o PC está sempre apontando 8 bytes adiante a instrução atualmente em execução.



Pipeline

- A execução de uma instrução de salto implica em um *flush* no pipeline.
- O ARM10 usa preditor de salto, que reduz o efeito do *flush*, prevendo possíveis ramificações e carregando o novo endereço do salto antes da execução da instrução.
- Uma instrução na etapa de execução será concluída, mesmo que aconteça uma interrupção.
- Outras instruções no pipeline serão abandonadas e o processador começará a preencher o pipeline a partir da entrada apropriada na tabela de vetores.

Exceções, Interrupções e tabela de vetores

- Quando ocorre uma exceção ou interrupção, o processador define o PC para um endereço de memória específico.
- O endereço está dentro de um intervalo de endereços especial chamado tabela de vetores.
- As entradas na tabela de vetores são instruções que ramificam para rotinas específicas projetadas para manipular uma exceção ou interrupção específica.
- Quando ocorre uma exceção ou interrupção, o processador suspende a execução normal e inicia o carregamento de instruções da tabela de vetores de exceção.

Rotinas de Exceção

- *Reset vector* - é a localização da primeira instrução executada pelo processador quando a energia é aplicada.
- *Undefined instruction vector* - usada quando o processado não consegue decodificar uma instrução.
- *Software interrupt vector* - chamada quando a instrução SWI é executada.
- *Prefetch abort vector* - ocorre quando o processador tenta buscar uma instrução de um endereço sem as permissões de acesso corretas.
- *Data abort vector* - ocorre quando uma instrução tenta acessar a memória de dados sem as permissões de acesso corretas.
- *Interrupt request vector* - usado pelo hardware externo para interromper o fluxo de execução normal do processador.
- *Fast interrupt request vector* - semelhante ao anterior, mas é reservado para hardware que requer tempos de resposta mais rápidos.

Tabela de vetores de exceção

Exception/interrupt	Shorthand	Address	High address
Reset	RESET	0x00000000	0xffff0000
Undefined instruction	UNDEF	0x00000004	0xffff0004
Software interrupt	SWI	0x00000008	0xffff0008
Prefetch abort	PABT	0x0000000c	0xffff000c
Data abort	DABT	0x00000010	0xffff0010
Reserved	—	0x00000014	0xffff0014
Interrupt request	IRQ	0x00000018	0xffff0018
Fast interrupt request	FIQ	0x0000001c	0xffff001c

Extensões de hardware

- As extensões de hardware são componentes padrão colocados ao lado do núcleo do processador ARM.
- Eles melhoram o desempenho, gerenciam recursos e fornecem funcionalidade extra e são projetados para fornecer flexibilidade no manuseio de aplicativos específicos.
- Cada família ARM tem diferentes extensões disponíveis. Alguns exemplos são:
 - Cache e memória fortemente acoplada;
 - Gerenciamento de memória.
 - Interface de Co-processador.

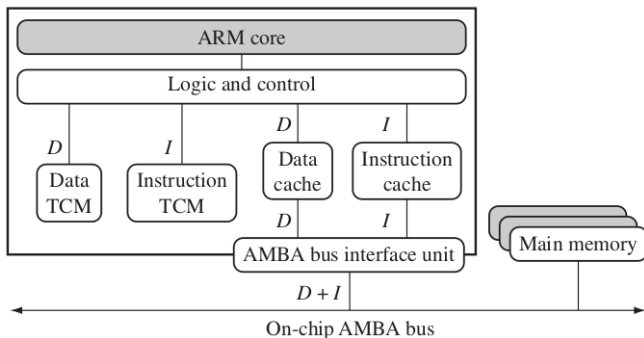
Cache e memória fortemente acoplada

- O cache é um bloco de memória rápida colocado entre a memória principal e o núcleo do processador.
- A maioria dos sistemas embarcados baseados em ARM usa um cache de nível único interno ao processador.
- O ARM possui duas formas de cache.
- O primeiro é baseado no estilo Von Neumann. Ele combina dados e instruções em um único cache unificado.
- A segunda forma baseia-se no estilo Harvard, tendo caches separados para dados e instruções.

Cache e memória fortemente acoplada

- Um cache fornece um aumento geral no desempenho, mas com uma perda de tempo de execução previsível.
- Algumas aplicações precisam que a execução do código seja determinística.
- É possível obter a execução determinística com a memória fortemente acoplada (TCM).
- O TCM é um SRAM rápido localizado próximo ao núcleo e garante os ciclos de clock necessários para buscar instruções ou dados.
- Ao combinar as duas tecnologias, os processadores ARM podem ter uma melhora no desempenho e resposta em tempo real previsível.

Cache e memória fortemente acoplada - Exemplo



Gerenciamento de Memória

- Sistemas embarcados geralmente usam vários dispositivos de memória.
- Geralmente, é necessário ter um método para ajudar a organizar esses dispositivos e proteger o sistema contra aplicativos que tentam fazer acessos inadequados ao hardware.

Gerenciamento de Memória

Os núcleos ARM possuem três tipos diferentes de hardware de gerenciamento de memória:

- *Nonprotected memory* - é fixa e oferece muito pouca flexibilidade.
- *memory protection unit* (MPU) - empregam um sistema simples que usa um número limitado de regiões de memória. Essas regiões são controladas com um conjunto de registros especiais do coprocessador e cada região é definida com permissões de acesso específicas.
- *memory management unit* (MMU) - usa um conjunto de tabelas de conversão para fornecer controle refinado sobre a memória. Essas tabelas são armazenadas na memória principal e fornecem um mapa de endereço virtual para físico, bem como permissões de acesso.

Coprocessadores

- Coprocessadores podem ser conectados ao processador ARM.
- Um coprocessador estende os recursos de processamento de um núcleo estendendo o conjunto de instruções ou fornecendo registros de configuração.
- O coprocessador também pode estender o conjunto de instruções fornecendo um grupo especializado de novas instruções.
- Por exemplo, há um conjunto de instruções especializadas que podem ser adicionadas ao conjunto de instruções ARM padrão para processar operações vetoriais de ponto flutuante (VFP).

Introdução à arquitetura ARM



Universidade Federal do Ceará - Campus Quixadá

Roberto Cabral
rbcabral@ufc.br

03 de Dezembro de 2020

Arquitetura e Organização de Computadores II