Arquitetura de Computadores II - 1COP012

Estrutura e função do processador

2023

Estrutura e função do processador

Objetivos

- Distinguir entre registradores visíveis ao usuário e de controle/estado, e discutir os propósitos dos registradores.
- Resumir o ciclo da instrução.
- Discutir os princípios por trás do pipeline de instruções.
- Comparar e contrastar as várias formas de hazards de pipeline.
- Apresentar uma visão geral da estrutura do processador x86.
- Apresentar uma visão geral da estrutura do processador ARM.

Estrutura e função do processador

Organização do processador

Para entender a organização do processador, vamos considerar os requisitos necessários:

- **Busca da instrução:** o processador lê uma instrução da memória (registrador, cache, memória principal).
- **Interpretação da instrução:** a instrução é decodificada para determinar qual ação é necessária.
- **Busca dos dados:** a execução de uma instrução pode necessitar a leitura de dados da memória ou de um módulo de E/S.
- **Processamento dos dados:** a execução de uma instrução pode necessitar efetuar alguma operação aritmética ou lógica com os dados.

Entrada/Saída

Organização do processador

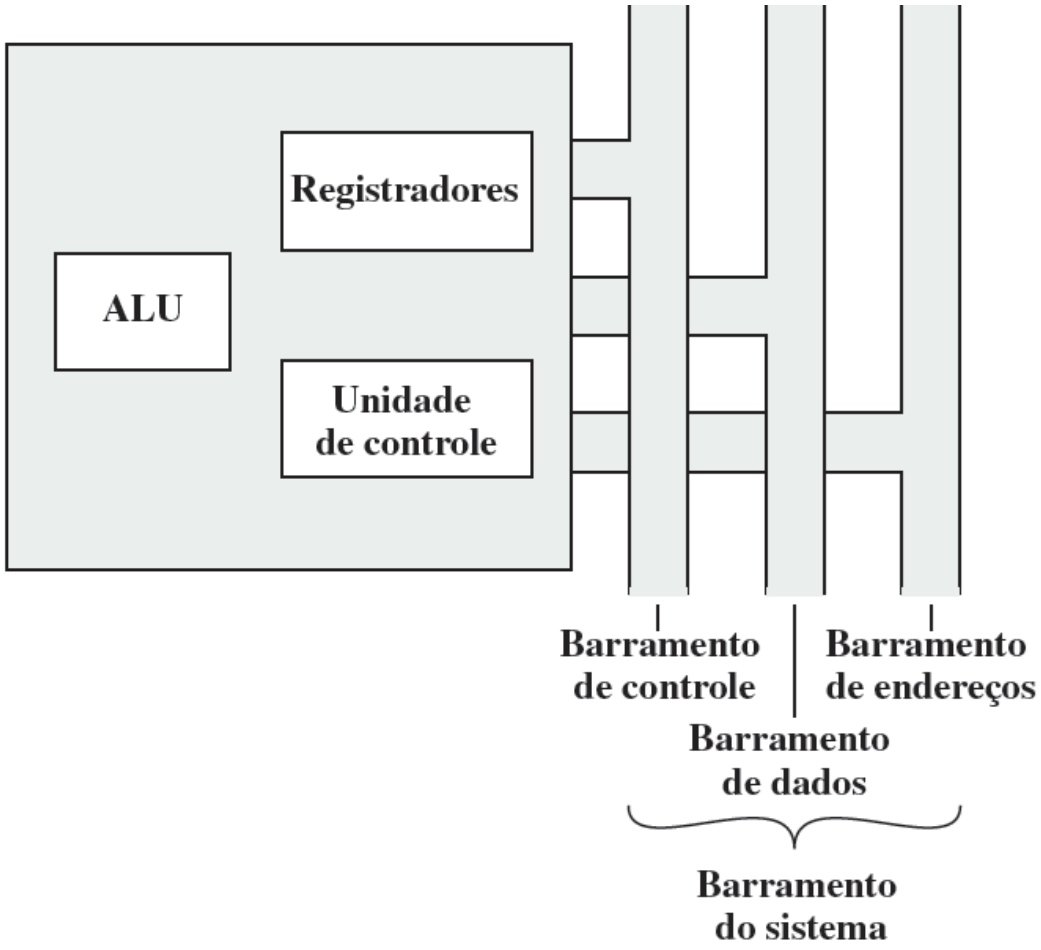
- Escrita dos dados: os resultados de uma execução podem necessitar escrever dados para a memória ou para um módulo de E/S.

Para fazer essas coisas, deve estar claro que **o processador precisa armazenar alguns dados temporariamente**.

Estrutura e função do processador

Organização do processador

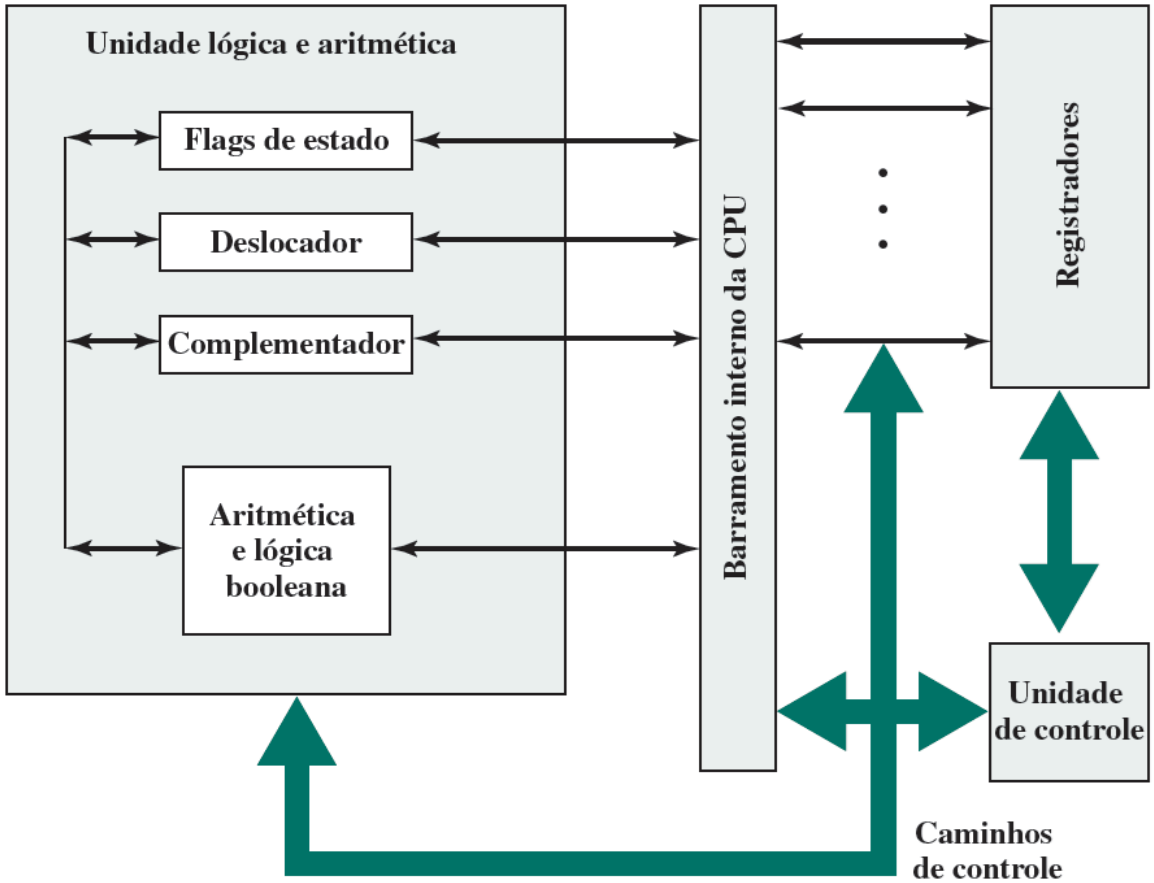
- CPU com barramento do sistema:



Estrutura e função do processador

Organização do processador

- Estrutura interna da CPU:



Estrutura e função do processador

Organização dos registradores

Os registradores no processador desempenham dois papéis:

1. **Registradores visíveis ao usuário:** possibilitam que o programador de linguagem de máquina ou de montagem minimize as referências à memória principal, pela otimização do uso de registradores.
2. **Registradores de controle e de estado:** usados pela unidade de controle para controlar a operação do processador e por programas privilegiados do Sistema Operacional para controlar a execução de programas.

Não há separação clara de registradores nessas duas categorias.

Estrutura e função do processador

Registradores visíveis ao usuário

- Um registrador visível ao usuário é aquele que pode ser referenciado pelos recursos da linguagem de máquina que o processador executa.
- Podemos dividi-los nas seguintes categorias:
  1. Uso geral.
  2. Dados.
  3. Endereços.
  4. Códigos condicionais.

Estrutura e função do processador

Registradores de controle e de estado

- Quatro registradores são essenciais:

1. **Contador de programas (PC):** contém o endereço de uma instrução a ser lida.
2. **Registrador da instrução (IR):** contém a instrução lida mais recentemente.
3. **Registrador de endereço de memória (MAR):** contém o endereço de um local de memória.
4. **Registrador de buffer de memória (MBR):** contém uma palavra de dados para ser escrita na memória ou a palavra lida mais recentemente.

Estrutura e função do processador

Registradores de controle e de estado

- Muitos modelos de processador incluem um registrador ou conjunto de registradores frequentemente conhecido como palavra de estado do programa (PSW), que contém as **informações de estado**.
- Em geral, a PSW contém códigos condicionais e outras informações de estado. Campos comuns ou flags incluem:
- Sinal: contém o bit de sinal do resultado da última operação aritmética.
- Zero: definido em 1 quando o resultado é 0.

Estrutura e função do processador

Registradores de controle e de estado

- Carry: definido em 1 se uma operação resultou em um carry ou um empréstimo de um bit de ordem maior.
- Igual: definido em 1 se uma comparação lógica resultou em igualdade.
- Overflow: usado para indicar overflow aritmético.
- Habilitar/desabilitar interrupção: usado para habilitar ou desabilitar interrupções.
- Supervisor: indica se o processador está executando no modo supervisor ou usuário.

Estrutura e função do processador

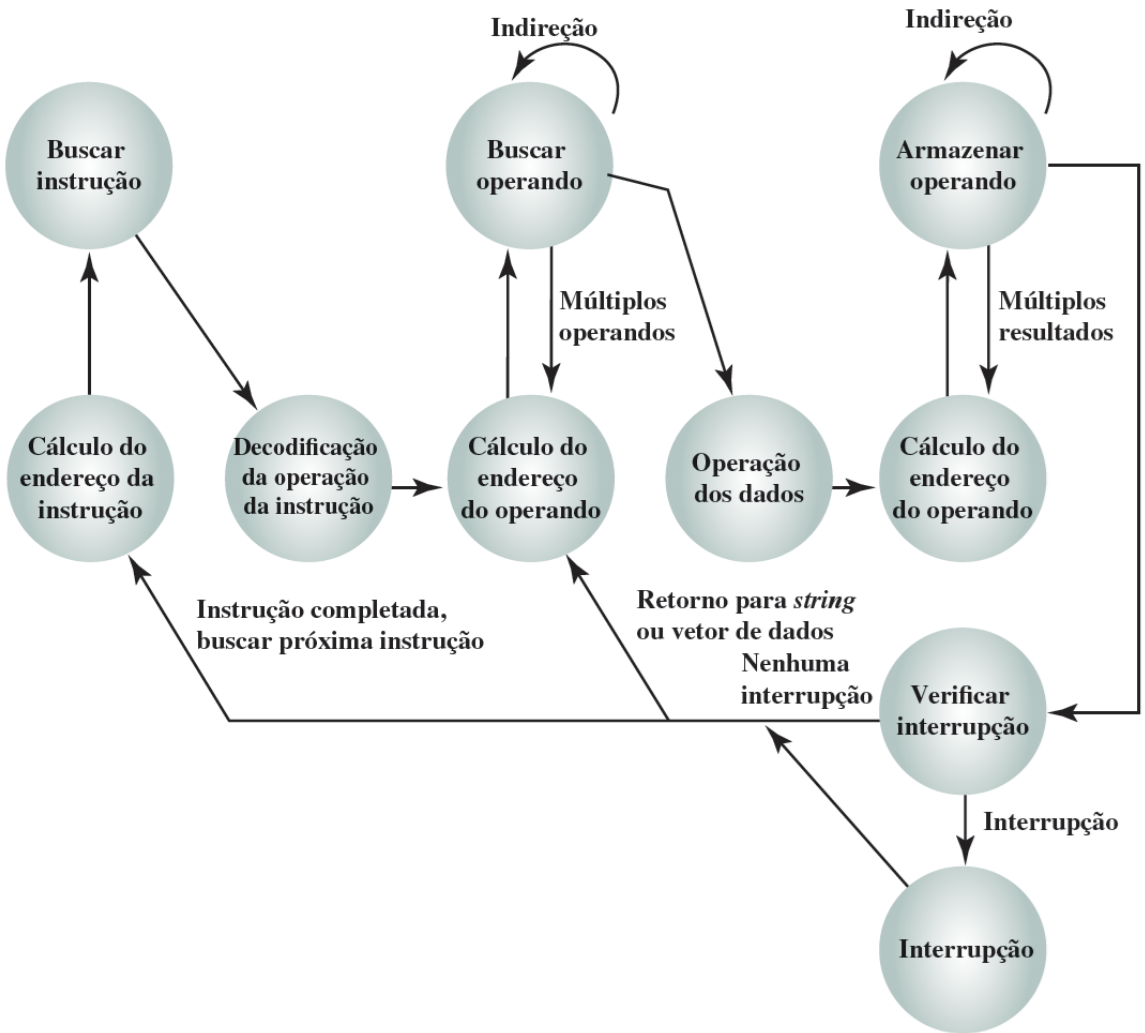
Ciclo da instrução

- Um ciclo de instrução inclui os seguintes estágios:
- Buscar: lê a próxima instrução da memória para dentro do processador.
- Executar: interpreta opcode e efetua a operação indicada.
- Interromper: se as interrupções estão habilitadas e uma interrupção ocorre, salva o estado do processo atual e atende a interrupção.

Estrutura e função do processador

Ciclo indireto

- Diagrama de estado do ciclo da instrução:



Estrutura e função do processador

Pipeline de instruções

- O pipeline de instruções é semelhante ao uso de uma linha de montagem numa planta industrial.
- Para aplicar este conceito à execução da instrução, precisamos reconhecer que, de fato, **uma instrução possui vários estágios**.
- O pipeline possui dois estágios independentes.
- O primeiro obtém a instrução e a coloca no buffer.
- Quando o segundo estágio está livre, o primeiro passa para ele a instrução do buffer.

Estrutura e função do processador

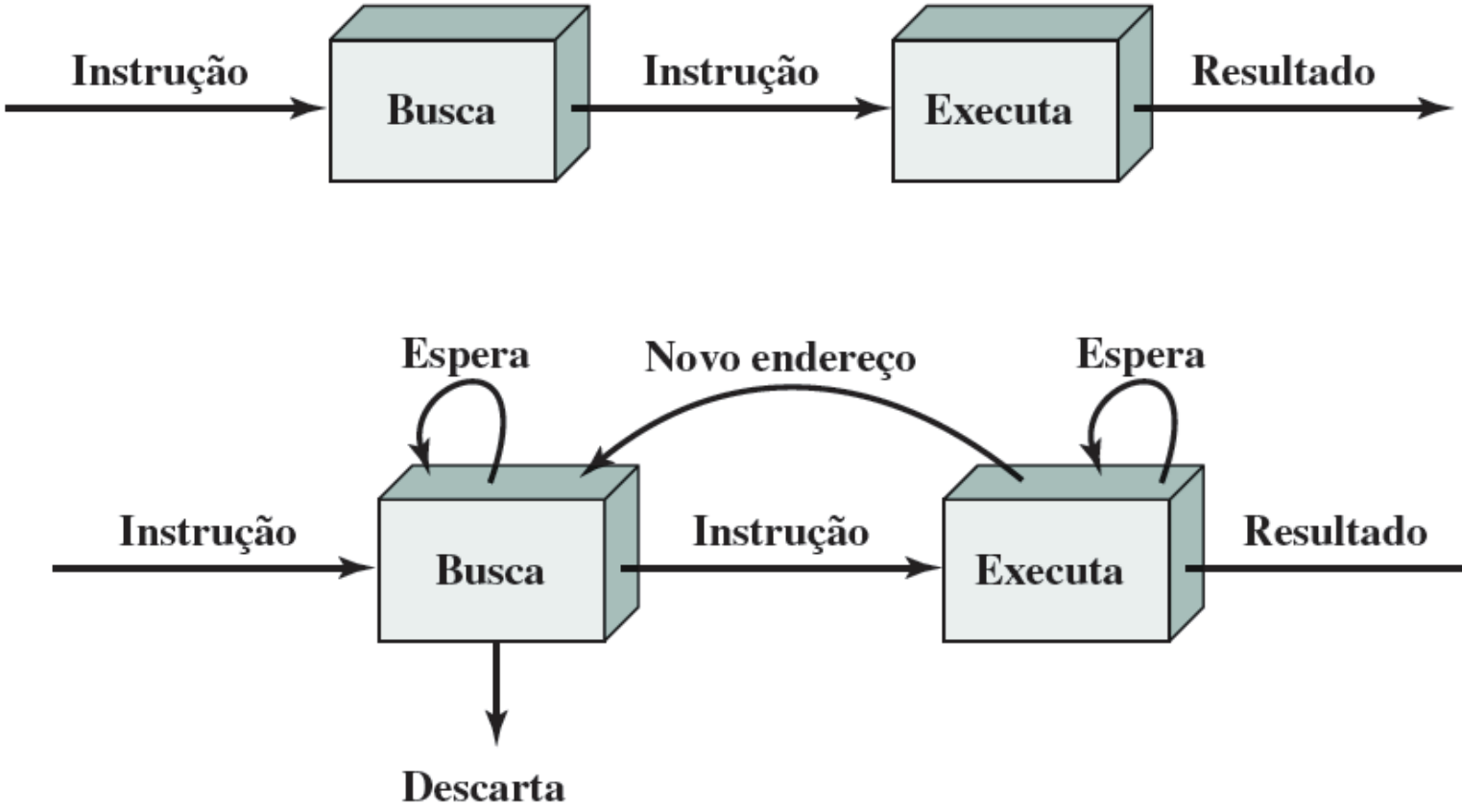
Pipeline de instruções

- Enquanto o segundo estágio está executando a instrução, o primeiro estágio aproveita qualquer ciclo de memória não utilizado para buscar a próxima instrução e colocá-la no buffer.
- Isso é chamado de busca antecipada (prefetch) ou busca sobreposta.
- Deve estar claro que este processo irá acelerar a execução das instruções.
- Se os estágios de leitura e execução forem de duração igual, o ciclo da instrução será reduzida pela metade.

Estrutura e função do processador

Pipeline de instruções

- Pipeline de instruções de dois estágios:



Estrutura e função do processador

Pipeline de instruções

- Diagrama de tempo para operação do pipeline de instrução:

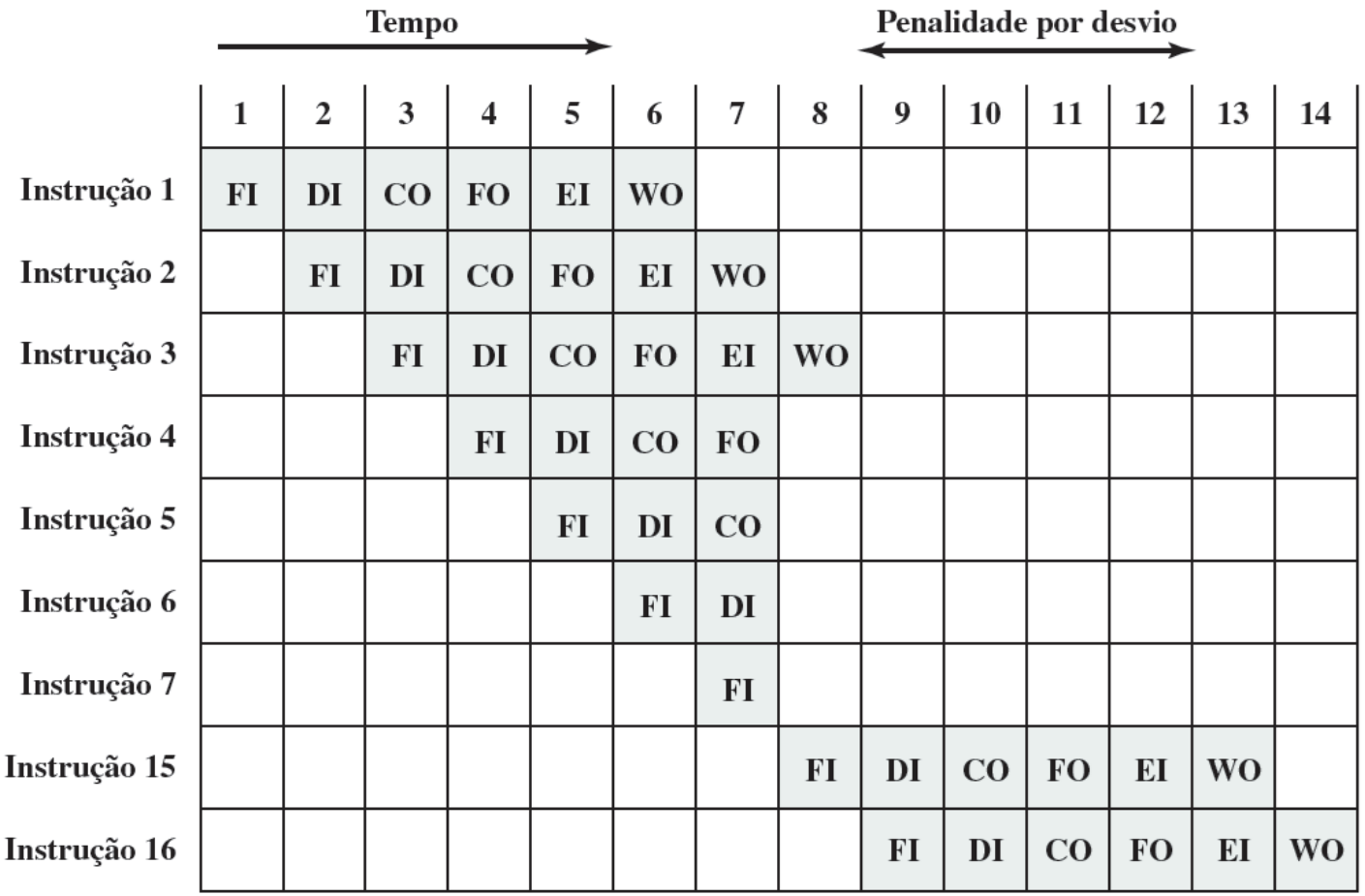
Tempo →

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Instrução 1	FI	DI	CO	FO	EI	WO								
Instrução 2		FI	DI	CO	FO	EI	WO							
Instrução 3			FI	DI	CO	FO	EI	WO						
Instrução 4				FI	DI	CO	FO	EI	WO					
Instrução 5					FI	DI	CO	FO	EI	WO				
Instrução 6						FI	DI	CO	FO	EI	WO			
Instrução 7							FI	DI	CO	FO	EI	WO		
Instrução 8								FI	DI	CO	FO	EI	WO	
Instrução 9									FI	DI	CO	FO	EI	WO

Estrutura e função do processador

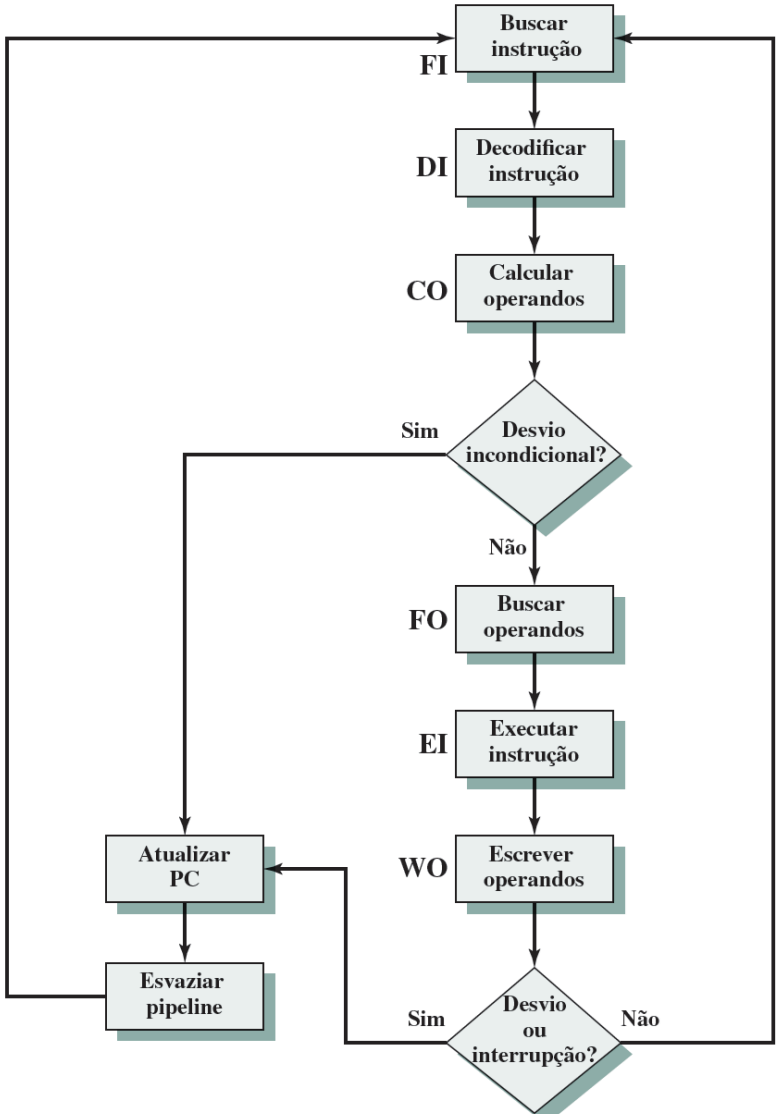
Pipeline de instruções

- O efeito de um desvio condicional na operação do pipeline da instrução:



Estrutura e função do processador

Pipeline de instruções



- Pipeline de instrução de uma CPU de seis estágios:

Estrutura e função do processador

Pipeline de instruções

- Descrição alternativa de um pipeline:

Tempo

	FI	DI	CO	FO	EI	WO
1	I1					
2	I2	I1				
3	I3	I2	I1			
4	I4	I3	I2	I1		
5	I5	I4	I3	I2	I1	
6	I6	I5	I4	I3	I2	I1
7	I7	I6	I5	I4	I3	I2
8	I8	I7	I6	I5	I4	I3
9	I9	I8	I7	I6	I5	I4
10		I9	I8	I7	I6	I5
11			I9	I8	I7	I6
12				I9	I8	I7
13					I9	I8
14						I9

	FI	DI	CO	FO	EI	WO
1	I1					
2	I2	I1				
3	I3	I2	I1			
4	I4	I3	I2	I1		
5	I5	I4	I3	I2	I1	
6	I6	I5	I4	I3	I2	I1
7	I7	I6	I5	I4	I3	I2
8	I15					I3
9	I16	I15				
10		I16	I15			
11			I16	I15		
12				I16	I15	
13					I16	I15
14						I16

Estrutura e função do processador

Desempenho do pipeline

- O tempo de ciclo τ de um pipeline de instrução é o tempo necessário para que a instrução avance um estágio dentro do pipeline.
- O tempo de ciclo pode ser determinado como

$$\tau = \max_i[\tau_i] + d = \tau_m + d \quad 1 \leq i \leq k$$

- Seja $T_{k,n}$ o tempo total necessário para que um pipeline com k estágios processe n instruções. Então

$$T_{k,n} = [k + (n-1)]\tau$$

Estrutura e função do processador

Desempenho do pipeline

- Considere agora um processador com funções equivalentes, mas sem pipeline, e suponha que o tempo do ciclo da instrução seja kτ.

- O fator de aceleração para o pipeline de instruções **comparado com a execução sem pipeline** é definido como

$$S_k = \frac{T_{1,n}}{T_{k,n}} = \frac{nk\tau}{[k + (n - 1)]\tau} = \frac{nk}{k + (n - 1)}$$

Estrutura e função do processador

Hazards do pipeline

- O **hazard de pipeline** ocorre quando o pipeline, ou alguma parte dele, **deve parar porque as condições não permitem a execução contínua**.
- Tal parada do pipeline é também conhecida como bolha de pipeline.
- Existem três tipos de hazards:

1. recursos,
2. dados e
3. controle.

Estrutura e função do processador

Hazards do pipeline

- Uma **hazard de recursos** ocorre quando duas (ou mais) instruções que já estão no pipeline precisam do mesmo recurso.
- O resultado é que **as instruções devem ser executadas em série em vez de em paralelo para uma parte do pipeline**. Uma hazard de recursos às vezes é chamada de *hazard estrutural*.
- Uma **hazard de dados** ocorre quando há um conflito no acesso de um local de operando.
- A **hazard de dados** pode ser definida da seguinte forma: duas instruções em um programa estão para ser executadas na sequência e ambas acessam um determinado operando de memória ou registrador.

Estrutura e função do processador

Hazards do pipeline

- Exemplo de hazard de dados:

```
ADD EAX, EBX /* EAX = EAX + EBX
SUB ECX, EAX /* ECX = ECX – EAX
```

- **Existem três tipos de hazard de dados**:

1) Leitura após escrita ou dependência verdadeira: uma instrução modifica um registrador ou uma posição de memória e uma instrução subsequente lê os dados dessa posição de memória ou registrador. O hazard ocorre quando a operação de leitura ocorre antes de a escrita ter sido completada.

Estrutura e função do processador

Hazards do pipeline

2) Escrita após leitura ou antidependência: uma instrução lê um registrador ou uma posição de memória e uma instrução subsequente escreve nessa posição. O hazard ocorre se a operação de escrita é completada antes da operação de leitura.

3) Escrita após escrita ou dependência de saída: duas instruções escrevem na mesma posição. O hazard ocorre se as operações de escrita ocorrerem na sequência inversa da esperada.

A **hazard de controle**, também conhecida como *hazard de desvio*, acontece quando um pipeline toma uma decisão errada ao prever um desvio e assim **acaba trazendo instruções dentro do pipeline que precisam ser descartadas logo em seguida**.

Estrutura e função do processador

Lidando com desvios

- Um dos principais problemas ao se projetar um pipeline de instruções é **garantir um fluxo estável de instruções para os estágios iniciais do pipeline**.
- **Uma série de abordagens** foram implementadas para lidar com desvios condicionais:

1. Múltiplos fluxos
2. Busca antecipada do alvo do desvio
3. Buffer de loops
4. Previsão de desvios
5. Desvios atrasados

Estrutura e função do processador

Lidando com desvios

1. Múltiplos fluxos

- Um pipeline simples tem penalidades na execução de uma instrução de desvio.
- Uma abordagem tipo força bruta é replicar as partes iniciais do pipeline e permitir que o pipeline busque as duas instruções, fazendo assim **uso de dois fluxos**.
- Problemas:1) com múltiplos pipelines, existem atrasos no acesso aos registradores e à memória. 2) instruções de desvios adicionais podem entrar no pipeline (ou em qualquer dos fluxos) antes que a decisão de desvio original seja resolvida. Cada uma dessas instruções precisa de um fluxo adicional.

Estrutura e função do processador

Lidando com desvios

2. Busca antecipada do alvo do desvio

- Quando um desvio condicional é reconhecido, **o alvo do desvio é lido antecipadamente, além da instrução que segue o desvio**. Esse alvo é então salvo até que a instrução de desvio seja executada. Se o desvio for tomado, o alvo já foi obtido.

3. Buffer de loops

- Se um desvio está para ser tomado, o hardware primeiro verifica se o alvo do desvio já está no buffer.
- Se estiver, a próxima instrução é buscada do buffer.
- O buffer de loops é semelhante, em princípio, **a uma cache dedicada para instruções**.
- A diferença é que buffer de loop **guarda apenas instruções na sequência e tem um tamanho menor, tendo assim um custo menor também**.

Estrutura e função do processador

Lidando com desvios

4. Previsão de desvios

- Várias técnicas podem ser usadas para prever se um desvio será tomado.
- Entre as mais comuns estão as seguintes:
  - Previsão nunca tomada
  - Previsão sempre tomada
  - Previsão por opcode
  - Chave tomada/não tomada
  - Tabela de histórico de desvio

Estrutura e função do processador

Lidando com desvios

5. Desvios atrasados

- É possível melhorar o desempenho do pipeline rearranjando automaticamente as instruções dentro de um programa, para que as instruções ocorram depois do que realmente desejado.
- Um exemplo instrutivo de um pipeline de instruções é o do Intel 80486.
- Ele implementa um pipeline de **cinco estágios**.
- Vejamos a seguir:

Estrutura e função do processador

Pipeline do Intel 80486

- Busca: o objetivo é preencher os buffers de busca antecipada com dados novos assim que os dados antigos tenham sido consumidos pelo decodificador da instrução.
- Estágio de decodificação 1: toda a informação de opcode e modo de endereçamento é decodificada no estágio D1.
- Estágio de decodificação 2: o estágio D2 traduz cada opcode em sinais de controle para ALU.
- Execução: este estágio inclui operações da ALU, acesso à cache e atualização de registradores.

Estrutura e função do processador

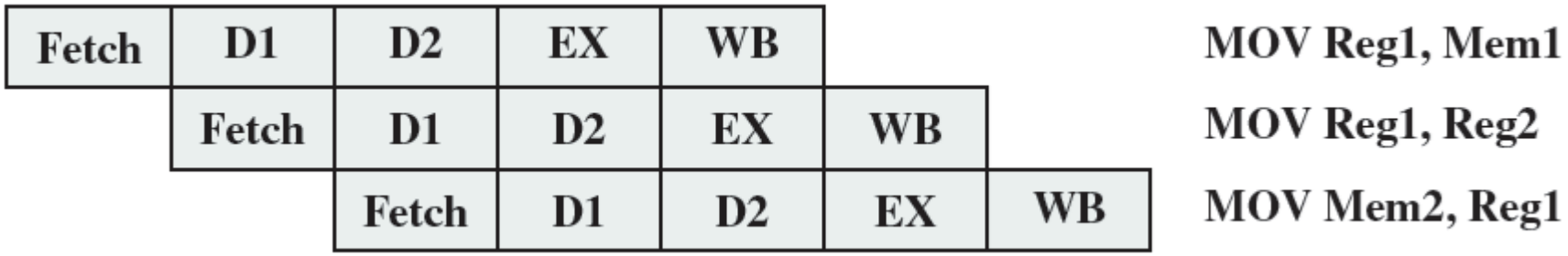
Pipeline do Intel 80486

- *Write back:* este estágio, se necessário, atualiza registradores e flags de estado modificados durante o processo da execução anterior.
- Com uso de dois estágios de decodificação, o pipeline pode sustentar uma falha de transferência de quase uma instrução por ciclo de clock.
- Instruções complexas e desvios condicionais podem diminuir essa taxa.
- As figuras a seguir mostram exemplos da operação do pipeline.

Estrutura e função do processador

Pipeline do Intel 80486

- Nenhum atraso para carregar dados no pipeline:



Estrutura e função do processador

Pipeline do Intel 80486

- Atraso para carregar o ponteiro:

Estrutura e função do processador

Família de processadores x86

Registradores do processador x86

- Unidade de inteiros no modo 32-bits

Tipo	Número	Extensão (bits)	Propósito
Geral	8	32	Registradores de uso geral
Segmento	6	16	Contém seletores de segmento
EFLAGS	1	32	Bits de estado e controle
Ponteiro de instrução	1	32	Ponteiro de instrução

Estrutura e função do processador

Família de processadores x86

Registradores do processador x86

- Unidade de inteiros no modo 64-bits:

Tipo	Número	Extensão (bits)	Propósito
Geral	16	32	Registradores de uso geral
Segmento	6	16	Contém seletores de segmento
RFLAGS	1	64	Bits de estado e controle
Ponteiro de instrução	1	64	Ponteiro de instrução

Estrutura e função do processador

Família de processadores x86

Registradores do processador x86

- Unidade de ponto flutuante:

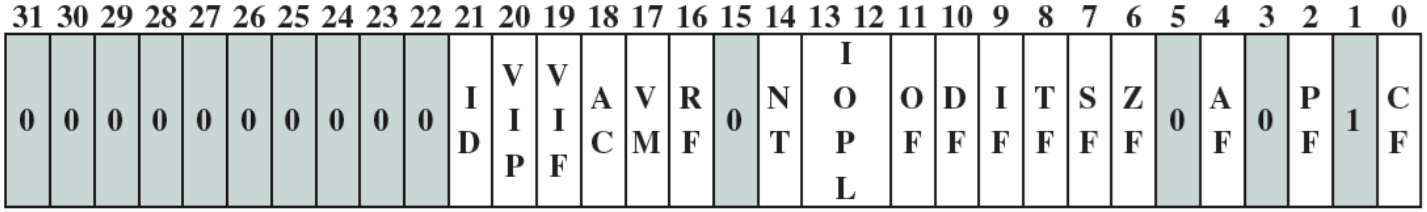
Tipo	Número	Extensão (bits)	Propósito
Numérico	8	80	Armazena números de ponto flutuante
Controle	1	16	Bits de controle
Estado	1	16	Bits de estado
Palavra de rótulo	1	16	Especifica o conteúdo de registradores numéricos
Ponteiro de instrução	1	48	Aponta para instrução interrompida pela exceção
Ponteiro de dados	1	48	Aponta para operando quando interrompido pela exceção

Estrutura e função do processador

Família de processadores x86

Registradores do processador x86

- Registrador EFLAGS do x86:



X ID = flag de identificação
X VIP = interrupção virtual pendente
X VIF = flag de interrupção virtual
X AC = verificação de alinhamento
X VM = modo virtual do 8086
X RF = flag de reinício
X NT = flag de tarefa aninhada
X IOPL = nível de privilégio de E/S
S OF = flag de *overflow*

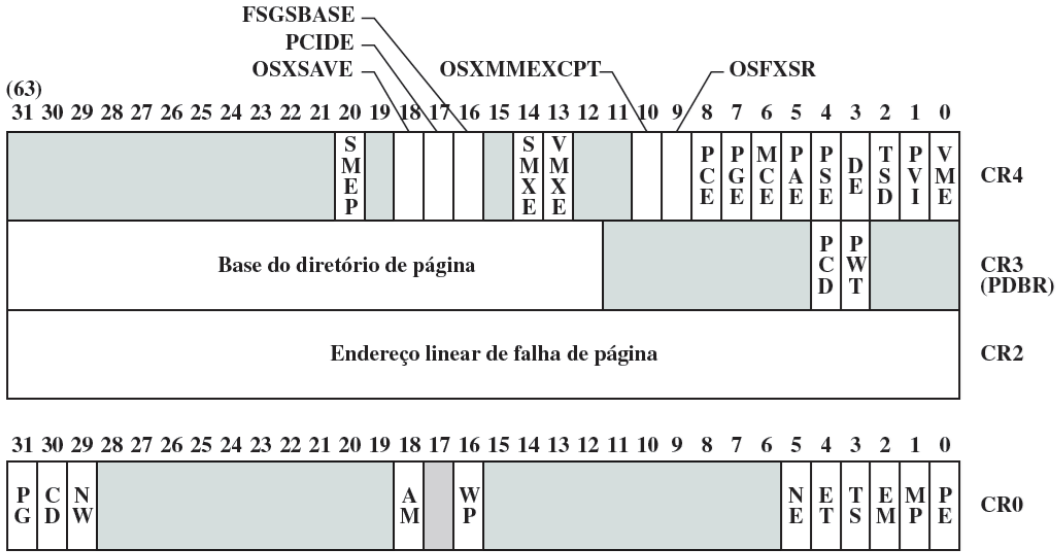
C DF = flag direcional
X IF = flag para habilitar interrupção
X TF = flag de *trap*
S SF = flag de sinal
S ZF = flag zero
S AF = flag de *carry* auxiliar
S PF = flag de paridade
S CF = flag de *carry*

S indica flag de estado.
C indica uma flag de controle.
X indica uma flag de sistema.
Os bits sombreados são reservados.

Estrutura e função do processador

Família de processadores x86

- Registradores de controle do x86



Áreas sombreadas indicam bits reservados.

OSXSAVE	=	habilita bit XSAVE
PCIDE	=	habilita identificadores contextualizados pelo processo
FSGSBASE	=	habilita instruções de segmento base
SMXE	=	habilita extensões do modo de segurança
VMXE	=	habilita extensões de máquina virtual
OSXMMEXCPT	=	suporta exceções SIMD FP não mascaradas
OSFXSR	=	suporta FXSAVE, FXSTOR
PCE	=	habilita contador de desempenho
PGE	=	habilita paginação global
MCE	=	habilita verificação de máquina
PAE	=	extensão de endereço físico
PSE	=	extensões de tamanho de página
DE	=	extensões de depuração
TSD	=	desabilita *time stamp*
PVI	=	interrupções virtuais no modo protegido
VME	=	modo de extensão virtual de 8086
PCD	=	desabilita cache de página
PWT	=	escrita transparente em nível de página
PG	=	paginação
CD	=	desabilita cache
NW	=	não *write through*
AM	=	máscara de alinhamento
WP	=	proteção de escrita
NE	=	erro numérico
ET	=	tipo de extensão
TS	=	troca de tarefa
EM	=	emulação
MP	=	monitor do coprocessador
PE	=	habilitação de proteção

Estrutura e função do processador

Família de processadores x86

Processamento de interrupções

Existem duas origens das interrupções e duas origens das exceções:

- **Interrupções mascaráveis** => recebidas no pino INTR do processador. O processador não reconhece uma interrupção mascarável se o flag habilitar interrupção (IF) não estiver definido.

- **Interrupções não mascaráveis** => recebidas no pino NMI do processador. Reconhecimento de tais interrupções não pode ser evitado.

Estrutura e função do processador

Família de processadores x86

Processamento de interrupções

Existem duas origens das interrupções e duas origens das exceções:

- **Exceções detectadas pelo processador** => resultam quando o processador encontra um erro enquanto executa uma instrução.

- **Exceções programadas** =>essas são as instruções que geram uma exceção (por exemplo, INTO, INT3, INTE e BOUND).

Estrutura e função do processador

Família de processadores x86

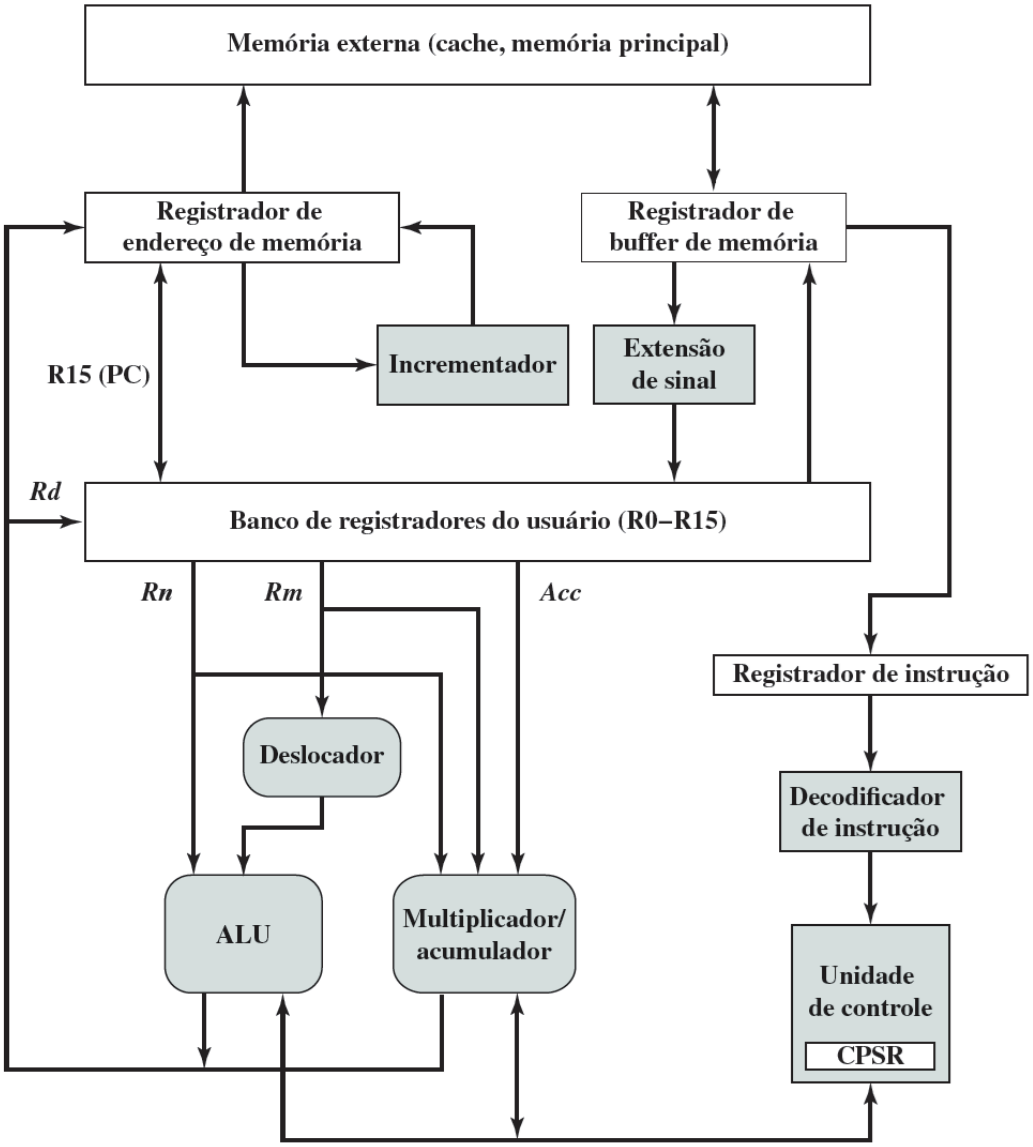
Tabela de vetores de interrupções e exceções para o x86

Número do vetor	Descrição
0	Erro de divisão; estouro de *overflow* ou divisão por zero.
1	Exceção de depuração; inclui várias falhas e *traps* relacionados à depuração.
2	Interrupção do pino NMI; sinal no pino NMI.
3	*Breakpoint*; causado pela instrução INT 3 que é uma instrução de 1 byte, útil para a depuração.
4	*Overflow* detectado em INTO; ocorre quando o processador executa INTO com o flag OF igual a um.
5	Limite em BOUND excedido; instrução BOUND compara um registrador com limites armazenados na memória e gera uma interrupção se o conteúdo do registrador está fora dos limites.
6	Opcode indefinido.
7	Dispositivo indisponível; tentativa de uso da instrução ESC ou WAIT falha por causa da demora do dispositivo externo.
8	Falha dupla; duas interrupções ocorrem durante a mesma instrução e não podem ser tratadas em série.
9	Reservado.
10	Segmento de estado de tarefa inválido; segmento que descreve a tarefa requerida não é inicializado ou não é válido.
11	Segmento ausente; segmento requerido não está presente.
12	Falha de pilha; limite do segmento da pilha excedido ou segmento da pilha ausente.
13	Proteção geral; violação da proteção que não causa outra exceção (por exemplo, escrever no segmento que é somente para leitura).
14	Falha de página.
15	Reservado.
16	Erro de ponto flutuante; gerado por uma instrução aritmética de ponto flutuante.
17	Verificação de alinhamento; acesso a uma palavra armazenada em um endereço de byte ímpar ou uma palavra dupla armazenada em um endereço não múltiplo de 4.
18	Verificação de máquina; específico para cada modelo.
19–31	Reservado.
32–255	Vetores de interrupções de usuário; fornecidos quando sinal INTR é ativado.

Sem sombra: exceções.
Com sombra: interrupções.

Estrutura e função do processador

Processador ARM



Organização ARM simplificada

Estrutura e função do processador

Processador ARM

Modos do processador

- A arquitetura ARM suporta **sete modos de execução**:
- A maioria dos programas aplicativos executa em modo usuário.
- Enquanto o processador está no modo usuário, **o programa sendo executado é incapaz de acessar os recursos protegidos do sistema ou de alterar o modo**, causando uma exceção nesse caso.
- Os seis modos de execução restantes são referidos como **modos privilegiados**.
- São usados para executar o software de sistema.

Estrutura e função do processador

Processador ARM

Modos do processador

- Os modos de exceção têm acesso total aos recursos do sistema e podem mudar os modos livremente:

1. **Modo supervisor** => normalmente é o modo que executa o SO. Ele á ativado quando o processador encontra uma instrução de interrupção de software (chama os serviços do sistema operacional no ARM).

2. **Modo aborto de acesso** => ativado como resposta a falhas de memória.

3. **Modo indefinido** => ativado quando o processador tenta executar uma instrução que não é suportada nem pelo núcleo principal nem por um dos coprocessadores.

Estrutura e função do processador

Processador ARM

Modos do processador

- **Os modos de exceção têm acesso total aos recursos do sistema e podem mudar os modos livremente**:

1. Modo interrupção rápida => ativado sempre que o processador recebe um sinal de interrupção a partir de uma fonte designada de interrupção rápida. Uma interrupção rápida não pode ser interrompida, porém uma interrupção rápida pode interromper uma interrupção normal.

2. Modo interrupção => ativado sempre que o processador recebe um sinal de interrupção a partir de qualquer outra origem de interrupção (diferente da interrupção rápida).

- O modo privilegiado restante é o modo sistema.
- O modo de sistema é usado para executar certas tarefas privilegiadas do sistema operacional.

Estrutura e função do processador

Processador ARM

Organização dos registradores

- 37 registradores de 32 bits (visíveis ao usuário), sendo 31 de propósito geral e 6 de estado de programa.

- R0 a R7, R15 (PC) e registrador de estado de programa corrente (CPSR) são visíveis e compartilhados por todos os modos.

- R8 a R12 são compartilhados por todos os modos exceto interrupção rápida, a qual possui seus próprios registradores dedicados R8_fiq até R12_fiq.

- Todos os modos de exceção tem suas próprias versões de registradores R13 e R14.

- Todos os modos de exceção têm um registrador próprio de estado de programa dedicado salvo (**SPSR – Saved Program Status Register**)

Estrutura e função do processador

Processador ARM

Organização dos registradores

Modos						
	Modos privilegiados					
		Modos de exceção				
Usuário	**Sistema**	**Supervisor**	**Aborto de acesso**	**Indefinido**	**Interrupção**	**Interrupção rápida**
R0	R0	R0	R0	R0	R0	R0
R1	R1	R1	R1	R1	R1	R1
R2	R2	R2	R2	R2	R2	R2
R3	R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6	R6
R7	R7	R7	R7	R7	R7	R7
R8	R8	R8	R8	R8	R8	R8_fiq
R9	R9	R9	R9	R9	R9	R9_fiq
R10	R10	R10	R10	R10	R10	R10_fiq
R11	R11	R11	R11	R11	R11	R11_fiq
R12	R12	R12	R12	R12	R12	R12_fiq
R13(SP)	R13(SP)	R13_svc	R13_abt	R13_und	R13_irq	R13_fiq
R14(LR)	R14(LR)	R14_svc	R14_abt	R14_und	R14_irq	R14_fiq
R15(PC)	R15(PC)	R15(PC)	R15(PC)	R15(PC)	R15(PC)	R15(PC)
CPSR	CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
		SPSR_svc	SPSR_abt	SPSR_und	SPSR_irq	SPSR_fiq

Sombreado indica que o registrador normal usado pelo modo usuário ou sistema foi substituído por um registrador específico para modo de exceção.

SP = ponteiro de pilha
LR = registrador de ligação
PC = contador de programa
CPSR = registrador de estado do programa corrente
SPSR = registrador de estado do programa salvo

Estrutura e função do processador

Processador ARM

Processamento de interrupções

- A arquitetura ARM suporta sete tipos de exceções.
- A tabela a seguir lista os tipos de exceções e o modo do processador que é usado para processar cada tipo.
- Quando uma exceção ocorre, a execução é forçada de um endereço fixo de memória correspondente ao tipo de exceção.
- Esses endereços fixos são chamados de vetores de exceção.
- Se mais do que uma interrupção está aguardando, elas são tratadas em ordem de prioridade.

Estrutura e função do processador

Processador ARM

Processamento de interrupções

Tipo de exceção	Modo	Endereço normal de entrada	Descrição
Reset	Supervisor	0x00000000	Ocorre quando o sistema é inicializado.
Aborto de dados	Aborto de acesso	0x00000010	Ocorre quando um endereço de memória inválido foi acessado, como se não houvesse memória física para o endereço, ou falta permissão correta de acesso.
FIQ (interrupção rápida)	FIQ	0x0000001C	Ocorre quando um dispositivo externo ativa o pino FIQ no processador. Uma interrupção não pode ser interrompida exceto por uma FIQ. A FIQ é projetada para suportar uma transferência de dados ou processo de canal e tem registradores privados suficientes para que não haja a necessidade de salvar os registradores em tais aplicações, economizando a sobrecarga da troca de contexto. Uma interrupção rápida não pode ser interrompida.
IRQ (interrupção)	IRQ	0x00000018	Ocorre quando um dispositivo externo ativa o pino IRQ do processador. Uma interrupção não pode ser interrompida, exceto por uma FIQ.
Aborto de busca antecipada	Aborto de acesso	0x0000000C	Ocorre quando uma tentativa de buscar uma instrução resulta em uma falha de memória. A exceção é ativada quando a instrução entra no estágio de execução do pipeline.
Instruções indefinidas	Indefinido	0x00000004	Ocorre quando uma instrução que não esteja no conjunto de instruções atinge o estágio de execução do pipeline.
Interrupção de software	Supervisor	0x00000008	Geralmente usado para permitir que os programas do modo usuário chamem o SO. O programa de usuário executa uma instrução SWI com um argumento que identifica a função que o usuário quer executar.

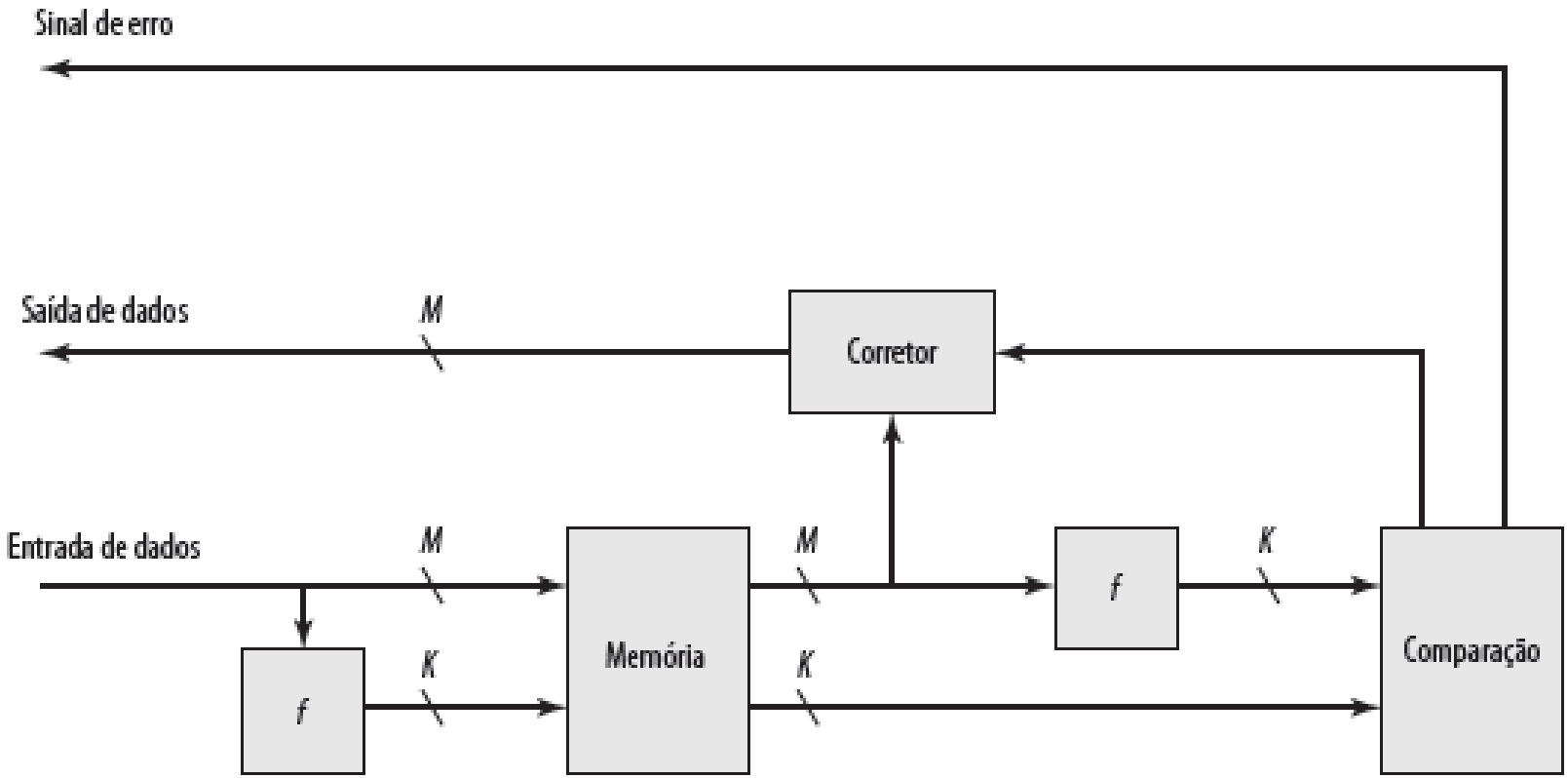
Correção de erro

Correção de erro

- Falha permanente
  - Defeito permanente
- Erro não permanente:
  - Aleatório, não destrutivo
  - Sem dano permanente à memória
- Detectado usando código de correção de erro de Hamming.

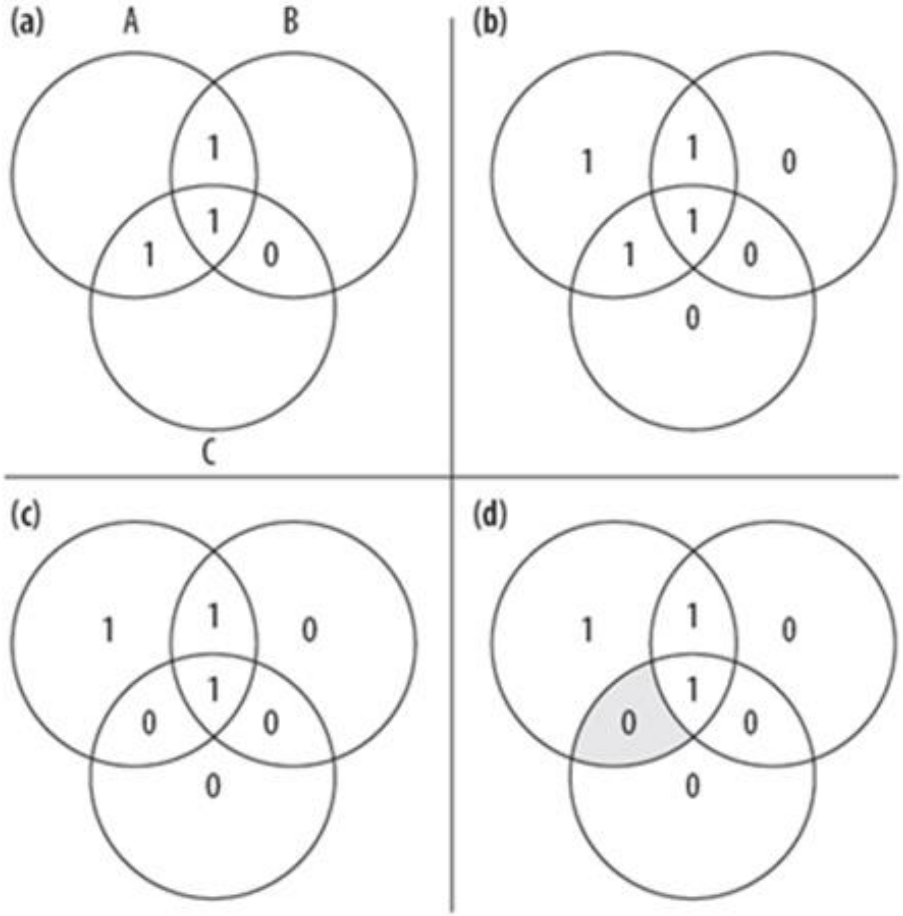
Correção de erro

Função do código de correção de erro



Correção de erro

Código de correção de erro de Hamming

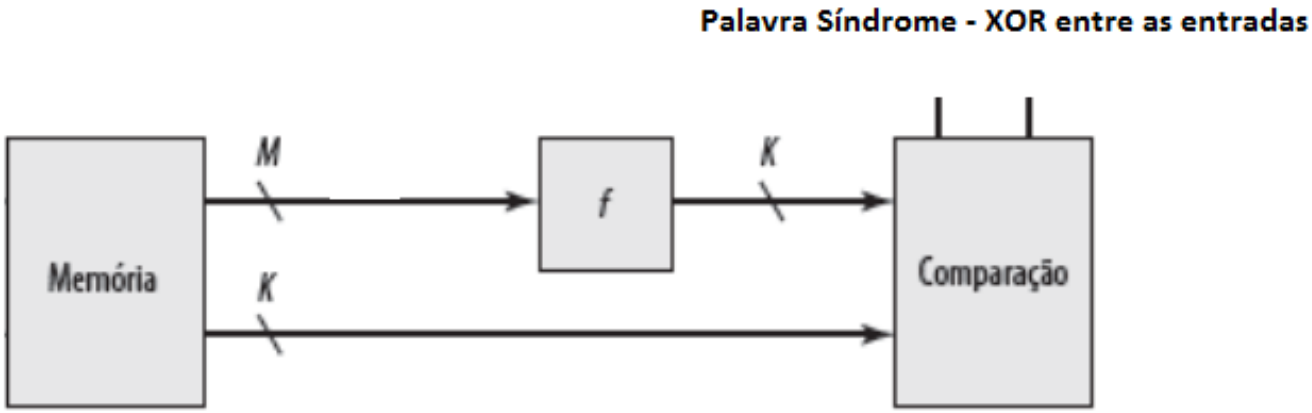


Correção de erro

Código de correção de erro de Hamming

Número de bits necessários para corrigir um erro de um único bit em uma palavra de M bits

$$2^K - 1 \geq M + K$$



Correção de erro

Código de correção de erro de Hamming

Palavra Síndrome – K bits

- se a palavra síndrome contiver apenas 0s, nenhum erro foi detectado.
- se a palavra síndrome contiver um e apenas um bit definido como 1, então houve um erro em um dos 4 bits de verificação. Nenhuma correção é necessária.
- se a palavra síndrome contiver mais de um bit definido como 1, então o valor numérico da palavra síndrome indica a posição do bit de dado com erro. Esse bit de dado é invertido para correção.

Correção de erro

Código de correção de erro de Hamming

Layout de bits de dados e bits de verificação

Posição de bit	12	11	10	9	8	7	6	5	4	3	2	1
Número da posição	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Bit de dados	D8	D7	D6	D5		D4	D3	D2		D1		
Bit de verificação					C8				C4		C2	C1

- As posições de bit cujos números de posição são potência de 2 são designadas como bits de verificação.

$C1 = D1 \oplus D2 \oplus D4 \oplus D5 \oplus D7$

$C2 = D1 \oplus D3 \oplus D4 \oplus D6 \oplus D7$

$C4 = D2 \oplus D3 \oplus D4 \oplus D8$

$C8 = D5 \oplus D6 \oplus D7 \oplus D8$

Correção de erro

Código de correção de erro de Hamming

Layout de bits de dados e bits de verificação

Posição de bit	12	11	10	9	8	7	6	5	4	3	2	1
Número da posição	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Bit de dados	D8	D7	D6	D5		D4	D3	D2		D1		
Bit de verificação					C8				C4		C2	C1

- Palavra de entrada
  00111001

C1= 1 ⊕ 0 ⊕ 1 ⊕ 1 ⊕ 0 = 1

C2= 1 ⊕ 0 ⊕ 1 ⊕ 1 ⊕ 0 = 1

C4= 0 ⊕ 0 ⊕ 1 ⊕ 0 = 1

C8= 1 ⊕ 1 ⊕ 0 ⊕ 0 = 0

Correção de erro

Código de correção de erro de Hamming

Cálculo do bit de verificação

Posição de bit	12	11	10	9	8	7	6	5	4	3	2	1
Número da posição	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Bit de dados	D8	D7	D6	D5		D4	D3	D2		D1		
Bit de verificação					C8				C4		C2	C1
Palavra armazenada como	0	0	1	1	0	1	0	0	1	1	1	1
Palavra buscada como	0	0	1	1	0	1	1	0	1	1	1	1
Número da posição	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Bit de verificação					0				0		0	1

Supondo que o bit de dados 3 sustente um erro e seja mudado de 0 para 1.

Correção de erro

Código de correção de erro de Hamming

Layout de bits de dados e bits de verificação

- Erro no bit 3 passando de 0 para 1
  00111101

C1= 1 ⊕ 0 ⊕ 1 ⊕ 1 ⊕ 0= 1

C2= 1 ⊕ 1 ⊕ 1 ⊕ 1 ⊕ 0 = 0

C4= 0 ⊕ 1 ⊕ 1 ⊕ 0 = 0

C8= 1 ⊕ 1 ⊕ 0 ⊕ 0 = 0

Correção de erro

Código de correção de erro de Hamming

Layout de bits de dados e bits de verificação

- Quando os novos bit de verificação forem comparados com os bits de verificação antigos, a palavra síndrome é formada:

C8	C4	C2	C1
0	1	1	1
0	0	0	1
⊕			
0	1	1	0

- O resultado é 0110, indicando que a posição de bit 6, que contém o bit de dados 3 está com erro.

Correção de erro

Código de correção de erro de Hamming

Cálculo do bit de verificação

Posição de bit	12	11	10	9	8	7	6	5	4	3	2	1
Número da posição	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Bit de dados	D8	D7	D6	D5		D4	D3	D2		D1		
Bit de verificação					C8				C4		C2	C1
Palavra armazenada como	0	0	1	1	0	1	0	0	1	1	1	1
Palavra buscada como	0	0	1	1	0	1	1	0	1	1	1	1
Número da posição	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Bit de verificação					0				0		0	1

Correção de erro

Código de correção de erro de Hamming

Aumento no tamanho da palavra com correção de erro

	Correção de erro único		Correção de erro único/ detecção de erro duplo	
Bits de dados	Bits de verificação	% de aumento	Bits de verificação	% de aumento
8	4	50	5	62,5
16	5	31,25	6	37,5
32	6	18,75	7	21,875
64	7	10,94	8	12,5
128	8	6,25	9	7,03
256	9	3,52	10	3,91