

Arquitetura e Organização de Computadores

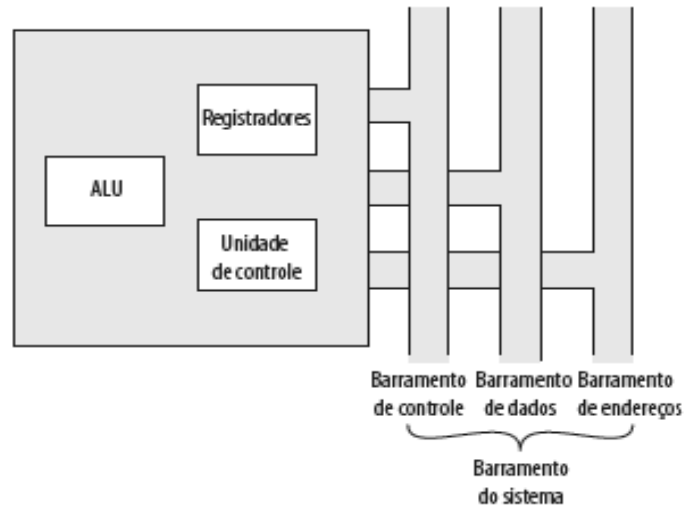
Capítulo 12

Estrutura e função do processador

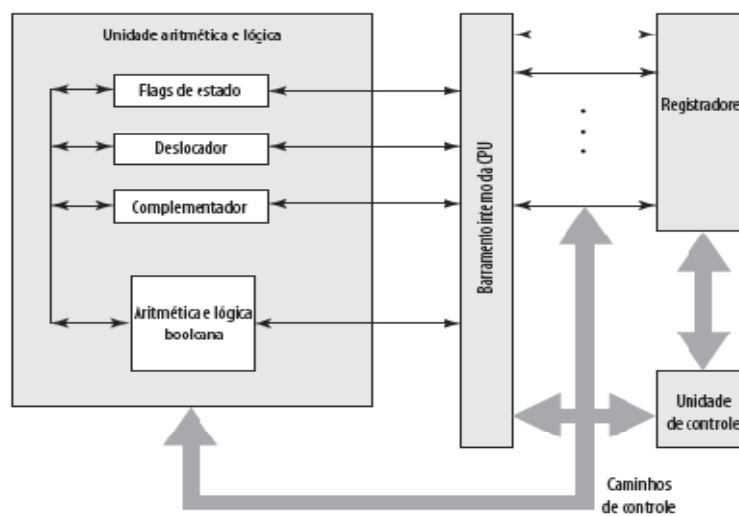
Estrutura da CPU

- CPU precisa:
 - Buscar instruções.
 - Interpretar instruções.
 - Obter dados.
 - Processar dados.
 - Gravar dados.

CPU com barramento de sistemas



Estrutura interna da CPU



Registradores

- CPU precisa ter algum espaço de trabalho (armazenamento temporário).
- Registradores chamados.
- Número e função variam entre projetos de processador.
- Uma das principais decisões de projeto.
- Alto nível de hierarquia de memória.

Registradores visíveis ao usuário

- Uso geral.
- Dados.
- Endereços.
- Códigos condicionais.

Registradores de uso geral

- Podem ser de propósito geral verdadeiro.
- Podem ser restritos.
- Podem ser usados para dados ou endereçamento.
- Dados:
 - Acumulador.
- Endereçamento:
 - Segmento.

- Torne-os de uso geral:
 - Aumente flexibilidade e opções do programador.
 - Aumente tamanho de instrução e complexidade.
- Torne-os especializados:
 - Instruções menores (mais rápidas).
 - Menos flexibilidade.

Quantos registradores de uso geral?

- Entre 8 –32.
- Menos registradores = mais referências à memória.
- Mais não reduz as referências à memória e ocupa espaço no processador.
- Veja também RISC.

De qual tamanho?

- Grande o suficiente para manter endereço completo.
- Grande o suficiente para manter palavra completa.
- Normalmente, é possível combinar dois registradores de dados.
 - Programação C.
 - Double int a.
 - Long int a.

Registradores de código condicional

- Conjuntos de bits individuais.
 - P.e., resultado da última operação foi zero.
- Podem ser lidos (implicitamente) por programas.
 - P.e., Jump if zero.
- Não podem (normalmente) ser alterados por programas.

Registradores de controle e estado

- Contador de programa.
- Registrador de decodificação de instrução.
- Registrador de endereço de memória.
- Registrador de buffer de memória.
- Revisão: o que todos eles fazem?

Palavra de estado do programa

- Um conjunto de bits.
- Inclui *flags* (códigos condicionais).
- Sinal do último resultado.
- Zero.
- *Carry*.
- Igual.
- *Overflow*.
- Habilitar/desabilitar interrupção.
- Supervisor.

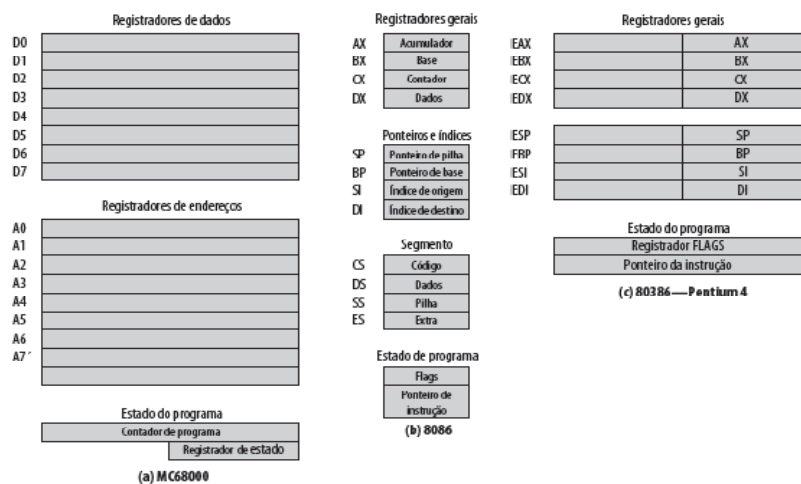
Modo supervisor

- Intel anel zero.
- Modo kernel.
- Permite execução de instruções privilegiadas.
- Usado pelo sistema operacional.
- Não disponível aos programas do usuário.

Outros registradores

- Podem ter registradores apontando para:
 - Blocos de controle de processo (ver S/O).
 - Vetores de interrupção (ver S/O).
- N.B. Projeto da CPU e projeto do sistema operacional são bastante interligados.

Exemplo de organizações de registradores



Ciclo de instrução

- Revisão.
- Stallings, Capítulo 3.

Ciclo indireto

- Pode exigir acesso à memória para obter operandos.
- Endereçamento indireto requer mais acessos à memória.
- Pode ser imaginado como subciclo de instrução adicional.

Ciclo de instrução com indireção

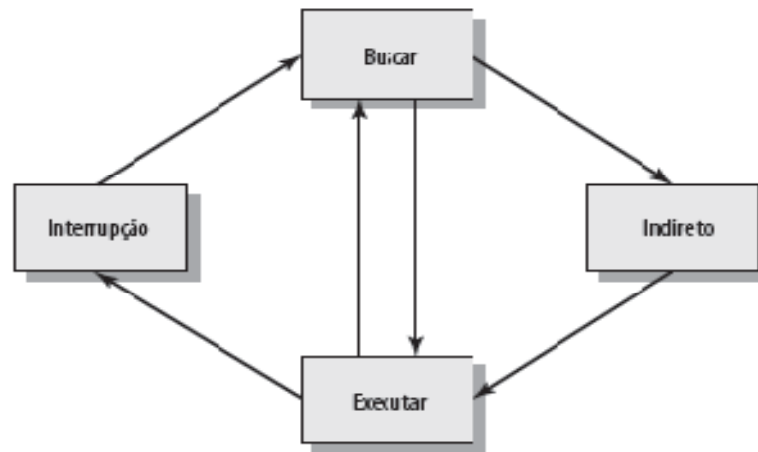
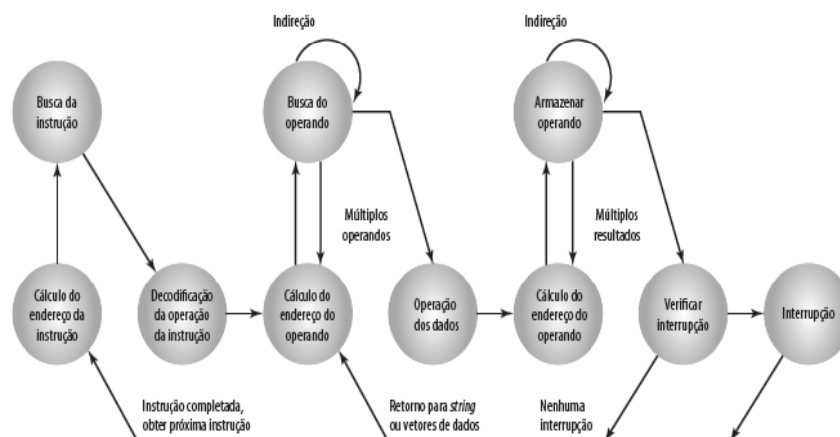


Diagrama de estado do ciclo da instrução



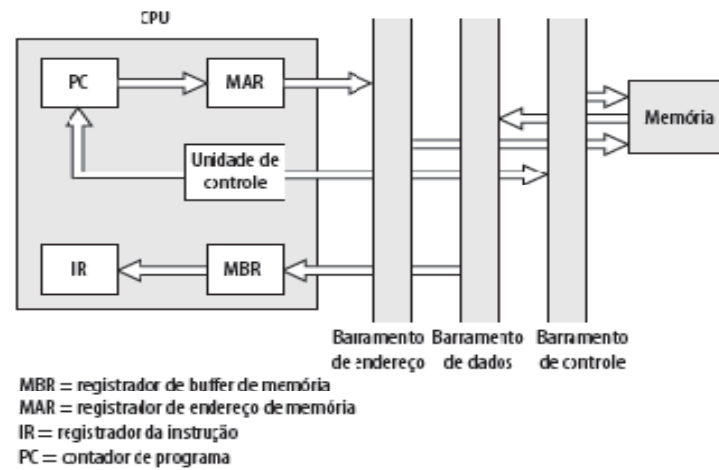
Fluxo de dados (busca de instrução)

- Depende do projeto da CPU.
- Em geral:
- Busca:
 - PC contém endereço da próxima instrução.
 - Endereço movido para MAR.
 - Endereço colocado no barramento de endereço.
 - Unidade de controle solicita leitura de memória.
 - Resultado colocado no barramento de dados, copiado para MBR, depois para IR.
 - Enquanto isso, PC incrementado em 1.

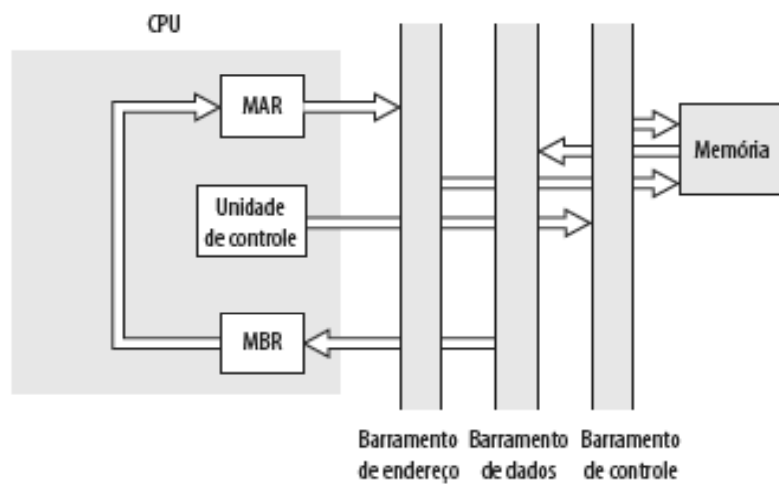
Fluxo de dados (busca de dados)

- IR é examinado.
- Se endereçamento indireto, ciclo indireto é realizado.
 - N bits da extrema direita do MBR transferidos para MAR.
 - Unidade de controle solicita leitura de memória.
 - Resultado (endereço do operando) movido para MBR.

Fluxo de dados (ciclo de busca)



Fluxo de dados (ciclo indireto)



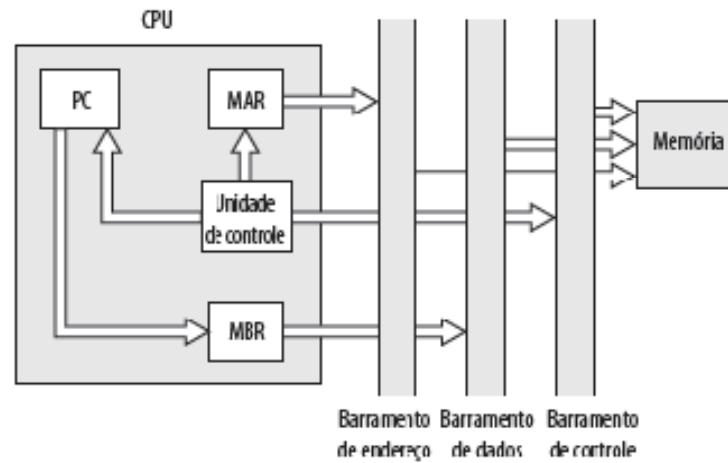
Fluxo de dados (execução)

- Pode tomar muitas formas.
- Depende da instrução sendo executada.
- Pode incluir:
 - Leitura/escrita da memória.
 - Entrada/saída.
 - Transferências de registradores.
 - Operações da ALU.

Fluxo de dados (interrupção)

- Simples.
- Previsível.
- PC atual salvo para permitir retomada após interrupção.
- Conteúdo do PC copiado para MBR.
- Local especial da memória (p.e., ponteiro de pilha) carregado no MAR.
- MBR gravado na memória.
- PC carregado com endereço da rotina de tratamento de interrupção.
- Próxima instrução (primeira do tratador de interrupção) pode ser obtida.

Fluxo de dados (ciclo de interrupção)



Busca antecipada (prefetch)

- Busca acessando memória principal.
- Execução normalmente não acessa memória principal.
- Pode buscar próxima instrução durante execução da instrução atual.
- Chamada busca antecipada da instrução.

Desempenho melhorado

- Mas, não dobrado:
 - Busca normalmente mais curta que a execução.
 - Busca antecipada de mais de uma instrução?
 - Qualquer salto ou desvio significa que as instruções com busca antecipada não são as instruções solicitadas.
- Acrescente mais estágios para melhorar o desempenho.

Pipelining

- Buscar instrução.
- Decodificar instrução.
- Calcular operandos (ou seja, EAs).
- Buscar operandos.
- Executar instruções.
- Escrever resultado.
- Sobrepor estas operações.

Pipeline de instrução de dois estágios

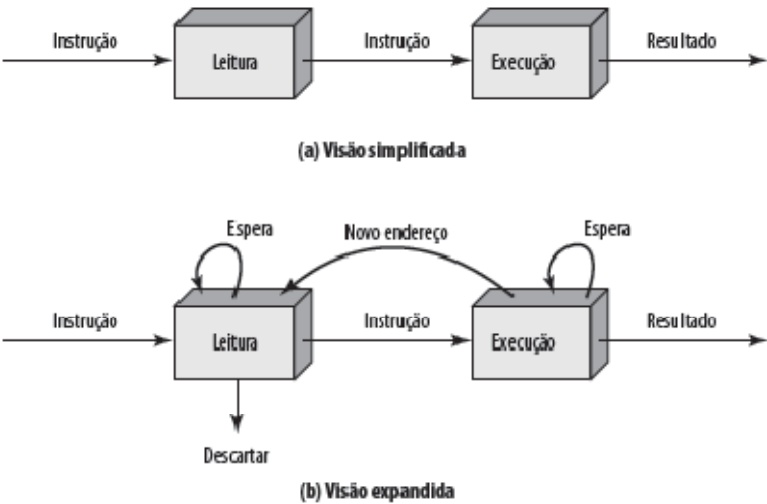
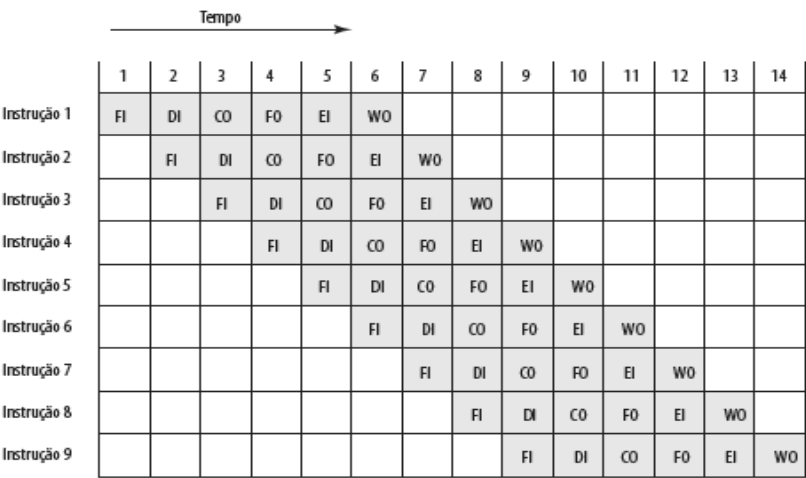


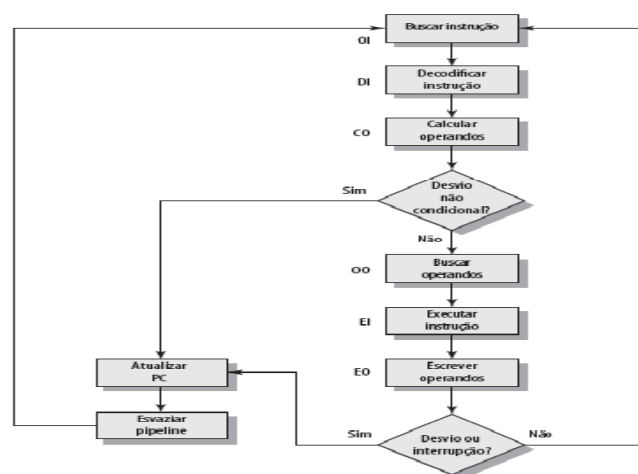
Diagrama de tempo para a operação do pipeline da instrução



Efeito de desvio condicional na operação do pipeline na instrução

	Tempo →						← Penalidade por desvio							
	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							
Instrução 5					FI	DI	CO							
Instrução 6						FI	DI							
Instrução 7							FI							
Instrução 15								FI	DI	CO	FO	EI	WO	
Instrução 16									FI	DI	CO	FO	EI	WO

Pipeline de instrução de seis estágios



Descrição alternativa de um pipeline

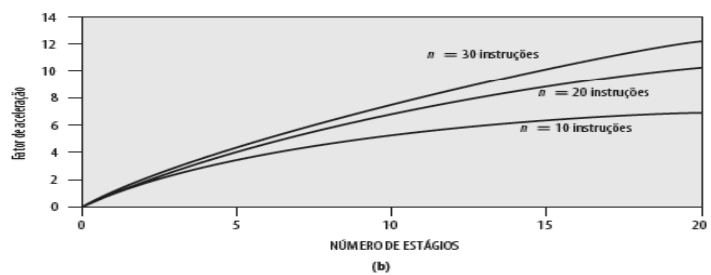
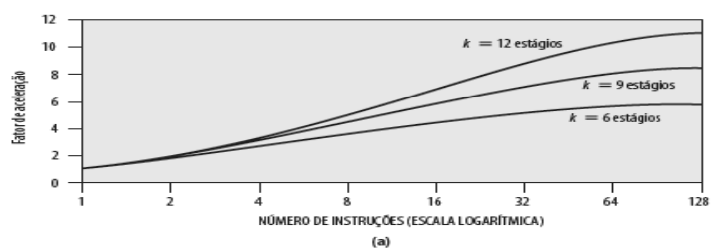
	FI	D	CO	FO	EI	WO
1	I1					
2	I2	I1				
3	I3	I2	I1			
4	I4	I3	I2	I1		
5	I5	I4	I3	I2	I1	
6	I6	I5	I4	I3	I2	I1
7	I7	I6	I5	I4	I3	I2
8	I8	I7	I6	I5	I4	I3
9	I9	I8	I7	I6	I5	I4
10		I9	I8	I7	I6	I5
11			I9	I8	I7	I6
12				I9	I8	I7
13					I9	I8
14						I9

(a) Sem desvios

	R	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	I5				
10		I6	I15			
11			I16	I15		
12				I16	I15	
13					I16	I15
14						I16

(b) Com desvios condicionais

Fatores de aceleração com pipeline da instrução



Hazards do pipeline

- Pipeline, ou alguma parte do pipeline, precisa parar.
- Também conhecida como *bolha de pipeline*.
- Tipos de hazards:
 - Recursos.
 - Dados.
 - Controle.

Hazards de recursos

- Duas (ou mais) instruções no pipeline precisam do mesmo recurso.
- Executados em série, e não em paralelo, parâmetro parte do pipeline.
- Também chamado *hazard estrutural*.
- P.e., considere pipeline simplificado em 5 estágios.
 - Cada estágio usa um ciclo de clock.
- No caso ideal, cada nova instrução entra no pipeline a cada ciclo de clock.
- Suponha que a memória principal tenha única porta.
- Considere buscas de instrução e leituras e escritas de dados uma por vez.
- Ignore a cache.
- Leitura ou escrita de operando não podem ser realizadas em paralelo com busca de instrução.
- Estágio de busca de instrução fica ocioso por um ciclo buscando I3.
- P.e., várias instruções prontas para entrar na fase de execução de instrução.
- Única ALU.
- Uma solução: aumentar recursos disponíveis.
 - Múltiplas portas da memória principal.
 - Múltiplas ALUs.

Hazards de dados

- Conflito no acesso de um local de operando.
- Duas instruções a serem executadas em sequência.
- Ambas acessam uma memória em particular ou operando do registrador.
- Se na sequência estrita, não ocorre problema.
- Se em um pipeline, valor do operando poderia ser atualizado para produzir resultado diferente da execução sequencial estrita.
- P.e., sequência de instruções de máquina do x86:
 - ADD EAX, EBX /* EAX = EAX + EBX
 - SUB ECX, EAX /* ECX = ECX – EAX
- Instrução ADD não atualiza EAX até o fim do estágio 5, no ciclo de clock 5.
- Instrução SUB precisa do valor no início do seu estágio 2, no ciclo de clock 4.
- Pipeline precisa parar por dois ciclos de clock.
- Sem hardware especial e algoritmos de impedimento específicos, resulta em uso ineficaz do pipeline.

Exemplo de hazard de dados

		Ciclo de clock									
		1	2	3	4	5	6	7	8	9	10
ADD EAX, EBX		FI	DI	FO	EI	EO					
	SUB ECX, EAX		FI	DI	Vazio		FO	EI	WO		
	I3			FI			DI	FO	EI	WO	
	I4						FI	DI	FO	EI	WO

Tipos de hazard de dados

- Leitura após escrita (RAW), ou dependência verdadeira:
 - Uma instrução modifica um registrador ou local de memória.
 - Instrução seguinte lê dados nesse local.
 - Hazard se leitura ocorre antes do término da escrita.
- Escrita após leitura (WAR), ou antidependência:
 - Uma instrução lê um registrador ou local da memória.
 - Instrução seguinte escreve no local.
 - Hazard se escrita termina antes que ocorra a leitura.
- Escrita após escrita (WAW), ou dependência de saída:
 - Duas instruções escrevem no mesmo local.
 - Hazard se a escrita ocorre na ordem contrária à sequência intencionada.
- Exemplo anterior é um hazard RAW.
- Ver também Capítulo 14.

Exemplo de hazard de recursos

		Ciclo de clock								
		1	2	3	4	5	6	7	8	9
Instrução	I1	FI	DI	FO	EI	WO				
	I2		FI	DI	FO	EI	WO			
	I3			FI	DI	FO	EI	WO		
	I4				FI	DI	FO	EI	WO	

(a) Pipeline de cinco estágios, caso ideal

		Ciclo de clock								
		1	2	3	4	5	6	7	8	9
Instrução	I1	FI	DI	FO	EI	WO				
	I2		FI	DI	FO	EI	WO			
	I3			Odoso	FI	DI	FO	EI	WO	
	I4					FI	DI	FO	EI	WO

(b) Operando de origem de I1 na memória

Hazard de controle

- Também conhecido como *hazard de desvio*.
- Pipeline toma decisão errada sobre previsão de desvio.
- Traz para o pipeline instruções que precisam ser descartadas subsequentemente.
- Lidando com desvios:
 - Múltiplos fluxos.
 - Busca antecipada do alvo do desvio.
 - Buffer de laço de repetição.
 - Previsão de desvio.
 - Desvio atrasado.

Fluxos múltiplos

- Têm dois pipelines.
- Busca antecipada de cada desvio em um pipeline separado.
- Usa pipeline apropriado.
- Ocasiona disputa por barramento e registrador.
- Múltiplos desvios exigem um fluxo adicional.

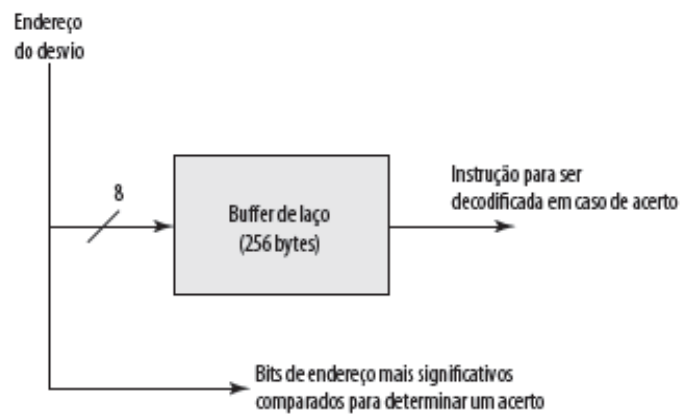
Busca antecipada do alvo do desvio

- Alvo do desvio buscado antecipadamente além das instruções após o desvio.
- Mantém alvo até que o desvio seja executado.
- Usada pelo IBM 360/91.

Buffer de laço de repetição

- Memória muito rápida.
- Mantido pelo estágio de busca do pipeline.
- Verifica buffer antes de buscar da memória.
- Muito bom para laços ou saltos pequenos.
- Compare com cache.
- Usado pelo CRAY-1.

Diagrama do buffer de laço de repetição



Previsão de desvio

- Previsão nunca tomada:
 - Assume que salto não acontecerá.
 - Sempre busca próxima instrução.
 - 68020 & VAX 11/780.
 - VAX não fará busca antecipada após desvio se resultar em falta de página (projeto do S/O *versus* CPU).
- Previsão sempre tomada:
 - Assume que salto acontecerá.
 - Sempre busca instrução alvo.

- Previsão por *opcode*:
 - Algumas instruções são mais prováveis de resultar em um salto do que outras.
 - Pode chegar até 75% de sucesso.
- Chave tomada/não tomada:
 - Baseada no histórico de desvio.
 - Boa para laços.
 - Refinada pelo histórico de desvio com dois níveis ou baseado em correlação.
- Baseado em correlação:
 - Nos desvios de laço, o histórico é uma boa forma de previsão.
 - Em estruturas mais complexas, a direção do desvio é correlacionada com a direção de desvios condicionados.
 - Também usa histórico de desvios recentes.

- Desvio atrasado:
 - Não salta até que você realmente precise.
 - Reorganiza instruções.

Fluxograma de previsão de desvio

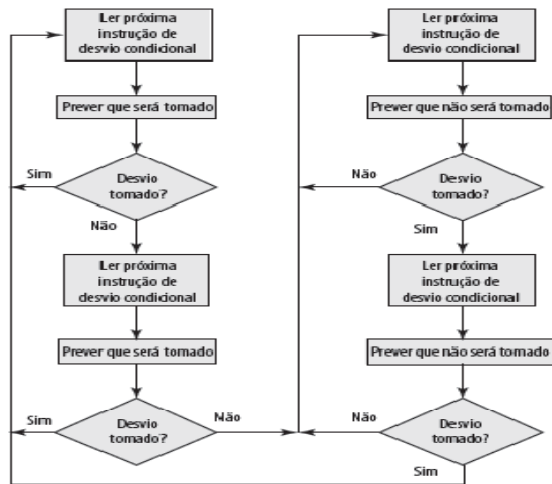
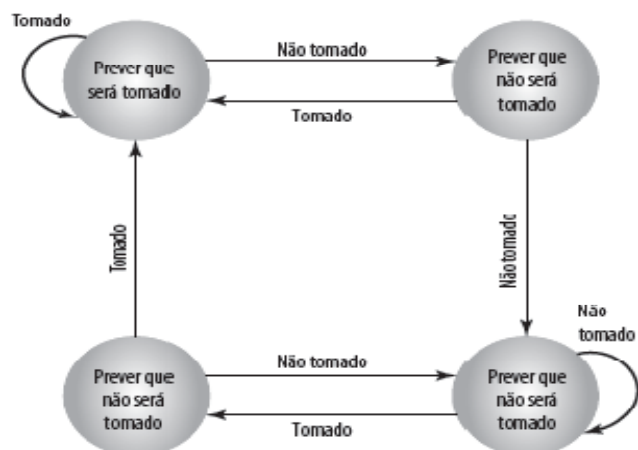
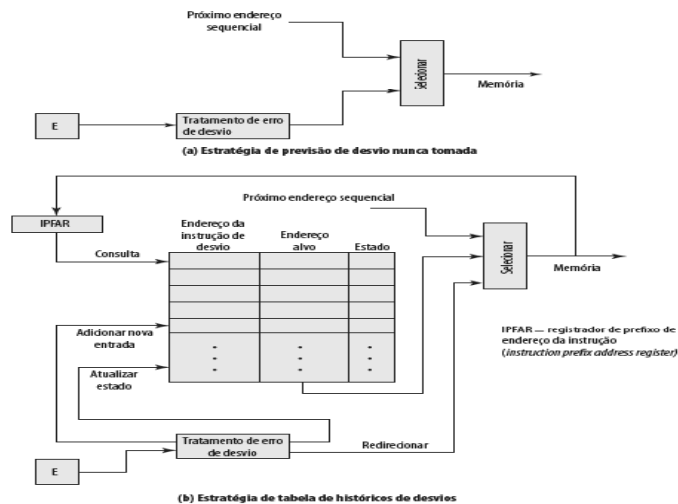


Diagrama de estados de previsão de desvio



Lidando com desvios



Pipeline de Intel 80486

- **Leitura:**
 - Da cache ou da memória externa.
 - Colocadas em um de 2 buffers de busca antecipada de 16 bits.
 - Enche buffer com novos dados quando antigos são consumidos.
 - Em média, 5 instruções lidas por carga.
 - Independente de outros estágios para manter buffers cheios.
- **Estágio de decodificação 1:**
 - *Opcode* e informação de modo de endereçamento.
 - No máximo 3 primeiros bytes da instrução.
 - Pode direcionar estágio D2 para obter restante da instrução.
- **Estágio de decodificação 2:**
 - Expande *opcode* para sinais de controle.
 - Cálculo de modos de endereçamento complexos.
- **Execução:**
 - Operações da ALU, acesso a cache, atualização de registrador.
- **Escrita:**
 - Atualiza registradores e flags.
 - Resultados enviados à cache e buffers de escrita da interface de barramento.

Exemplos de pipeline da instrução do 80486

Leitura	D1	D2	EX	Escrita			MOV Reg1, Mem1
	Leitura	D1	D2	EX	Escrita		MOV Reg1, Reg2
		Leitura	D1	D2	EX	Escrita	MOV Mem2, Reg1

(a) Nenhum atraso para carregamentos no pipeline

Leitura	D1	D2	EX	Escrita			MOV Reg1, Mem1
	Leitura	D1		D2	EX		MOV Reg2, (Reg1)

(b) Atraso para carregar ponteiro

Leitura	D1	D2	EX	Escrita			CMP Reg1, Imm
	Leitura	D1	D2	EX			Jcc Target
			Leitura	D1	D2	EX	Target

(c) Temporização da instrução de desvio

Registradores do Pentium 4

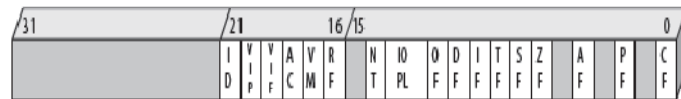
(a) Integer Unit

Type	Number	Length (bits)	Purpose
General	8	32	General-purpose user registers
Segment	6	16	Contain segment selectors
Flags	1	32	Status and control bits
Instruction Pointer	1	32	Instruction pointer

(b) Floating-Point Unit

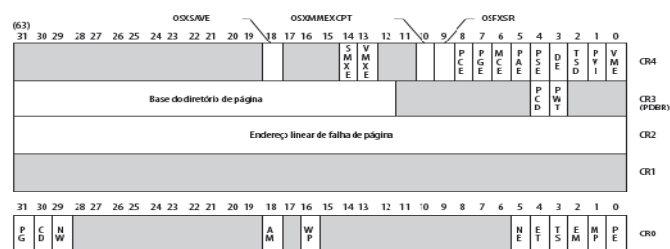
Type	Number	Length (bits)	Purpose
Numeric	8	80	Hold floating-point numbers
Control	1	16	Control bits
Status	1	16	Status bits
Tag Word	1	16	Specifies contents of numeric registers
Instruction Pointer	1	48	Points to instruction interrupted by exception
Data Pointer	1	48	Points to operand interrupted by exception

Registrador EFLAGS



ID	= flag de identificação	DF	= flag direcional
VIP	= interrupção virtual pendente	IF	= flag para habilitar interrupção
VIF	= flag de interrupção o virtual	TF	= flag de <i>trap</i>
AC	= Verificação de alinhamento	SF	= flag de <i>signal</i>
VM	= 8086 modo virtual	ZF	= flag Zero
RF	= flag de resumo	AF	= flag de <i>carry</i> auxiliar
NT	= flag tarefa aninhada	PF	= flag de paridade
IOPB	= de privilégio de I/O	CF	= flag de <i>carry</i>
OF	= flag de overflow		

Registradores de controle de x86



Áreas sombreadas indicam bits reservados.

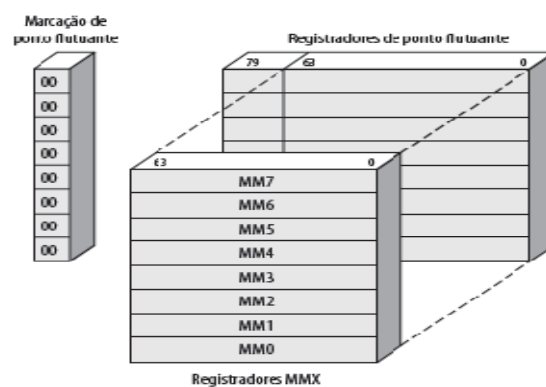
OSXSAVE = habilita bit XSAVE
 SMXSE = habilita extensões do modo de segurança
 VMXSE = habilita extensões de máquina virtual
 OSXMMEXCP = Suporta excessões SIMD/FPU não mascaradas
 OSXSRX = Suporta FPU SSE, FXSR/IO
 PCE = habilita conector de desempenho
 PGE = habilita paginação global
 MCE = habilita verificação de máquina
 PAE = extensão de endereço físico
 PSE = extensões de tamanho de página
 DE = extensões de depuração
 TSD = desabilita time stamp
 VIRT = interrupções virtuais no modo protegido
 VMXE = modo de extensão virtual de 8086

PCD = desabilita cache de página
PWT = escrita transparente em nível de página
PG = paginação
CD = desabilita cache
NW = not 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

Mapeamento de registradores MMX

- MMX usa vários tipos de dados de 64 bits.
- Usa campos de endereço de registrador de 3 bits.
 - 8 registradores.
- Nenhum registrador específico do MMX.
 - 64 bits de baixa ordem dos registradores de ponto flutuante existentes formam 8 registradores MMX.

Mapeamento de registradores MMX para registradores de ponto flutuante



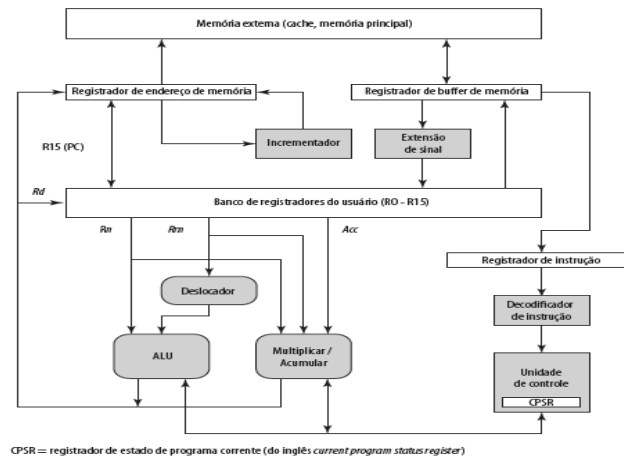
Processamento de interrupção no Pentium

- Interrupções:
 - Mascaráveis.
 - Não mascaráveis.
- Exceções:
 - Detectadas pelo processador.
 - Programadas.
- Tabelas de vetores de interrupções:
 - Cada tipo de interrupção recebe um número.
 - Indexam para tabela de vetor.
 - 256 vetores de interrupção de 32 bits.
- 5 classes de prioridade.

Atributos do ARM

- RISC.
- Array moderado de registradores uniformes.
 - Mais do que a maioria dos CISC, menos que muitos RISC.
- Modelo carregar/armazenar:
 - Operações executam com operandos nos registradores.
- Instrução uniforme de tamanho fixo:
 - 32 bits para conjunto padrão e 16 bits para Thumb.
- Deslocamento ou rotação podem pré-processar registradores de origem:
 - Separa unidades de ALU e deslocamento.
- Pequeno número de modos de endereçamento:
 - Todos os endereços de carga/armazenamento de registradores e campos da instrução.
 - Nenhum endereçamento indireto ou indexado envolvendo valores na memória.
- Endereçamento com autoincremento e autodecremento:
 - Melhora operações de laço.
- Execução condicional de instruções minimiza desvios condicionais:
 - Limpeza do pipeline reduzida.

Organização simplificada do ARM



Organização do processador ARM

- Muitas variações dependem da versão do ARM.
- Dados trocados entre processador e memória através de barramento de dados.
- Item de dados (load/store) ou instrução (leitura).
- Instruções passam por decodificador antes da execução.
- Pipeline e geração de sinal de controle na unidade de controle.
- Dados vão para arquivo de registrador:
 - Conjunto de registradores de 32 bits.
 - Um byte ou meia palavra tratados como complemento de dois estendidos com sinal até 32 bits.
- Normalmente dois registradores de origem e um resultado.
- Rotação ou deslocamento antes da ALU.

Modos do processador ARM

- Usuário.
- Privilegiado.
 - 6 modos.
 - SO pode ajustar software de sistemas utiliza.
 - Alguns registradores dedicados a cada modo privilegiado.
 - Mudanças de contexto mais rápidas.
- Exceção:
 - 5 dos modos privilegiados.
 - Entrada nas exceções dadas.
 - Substitui alguns registradores para registradores do usuário.
 - Evita corromper informações.

Modos privilegiados

- Modo do sistema:
 - Sem exceção
 - Usa mesmos registradores do modo usuário.
 - Pode ser interrompido por...
- Modo supervisor:
 - SO.
 - Interrupção de software usada para invocar serviços do sistema operacional.
- Modo de abortamento:
 - Falta de memória.
- Modo indefinido:
 - Tenta instrução que não é aceita pelo núcleo principal nem por um dos coprocessadores.
- Modo de interrupção rápido:
 - Sinal de interrupção da fonte de interrupção rápida designada.
 - Interrupção rápida não pode ser interrompida.
 - Pode interromper interrupção normal.
- Modo de interrupção.
 - Sinal de interrupção de qualquer outra fonte de interrupção.

Organização dos registradores do ARM

- 37 registradores de 32 bits.
- 31 registradores de uso geral.
 - Alguns têm propósitos especiais.
 - P.e., contadores de programa.
- Seis registradores de *status* de programa.
- Registradores em bancos parcialmente sobrepostos.
 - Modo processador determina banco.
- 16 registradores numerados e um ou dois registradores de *status* de programa visíveis.

Organização dos registradores do ARM

Modos						
Modos privilegiados						
Modos de exceção						
Usuário	Sistema	Supervisor	Abortamento	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 de 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 de programa corrente
SPSR = registrador de estado de programa salvo

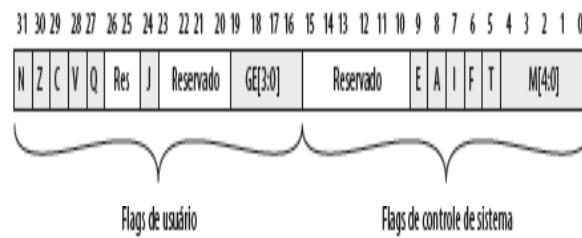
Registradores de propósito geral

- R13– normalmente ponteiro de pilha (SP).
 - Cada modo de exceção tem seu próprio R13.
- R14 – registrador de ligação (LR).
 - Endereço de retorno da sub-rotina e retornos do modo de exceção.
- R15 contador de programa.

CPSR

- CPSR processa registrador de estado.
 - Modos de exceção têm SPSR dedicado.
- 16 bits mais significativos flags do usuário.
 - Códigos de condição (N,Z,C,V).
 - Q– estouro ou saturação em instruções SMID.
 - J– instruções Jazelle (8 bits).
 - GE[3:0] SMID usam bits [19:16] como flag de maior ou igual.
- 16 bits menos significativos contêm flags para modo privilegiado.
 - E– endian.
 - Desabilitar interrupção.
 - T– instrução normal ou Thumb.
 - Modo.

Formato de CPSR e SPSR do ARM



Processamento de interrupção (exceção) ARM

- Mais de uma exceção permitida.
- Sete tipos.
- Execução forçada por vetores de exceção.
- Múltiplas exceções tratadas em ordem de prioridade.
- Processador para a execução após instrução atual.
- Estado do processador preservado no SPSR para exceção:
 - Endereço da instrução a executar colocado no registrador de ligação.
 - Retorna movendo SPSR p/ CPSR e R14 p/ PC.