

Organização e Arquitetura de Computadores

Pipeline de Instruções.

Pipeline de Instruções

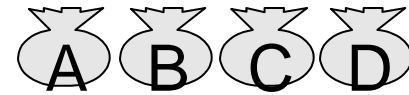
- À medida que os sistemas computacionais evoluem, um melhor desempenho pode ser obtido tirando vantagens das melhorias na tecnologia.
 - Circuitos mais rápidos
- Melhorias organizacionais no processador podem melhorar o desempenho.
 - múltiplos registradores
 - memória cache
 - Pipeline da Instrução.

Pipeline de Instruções

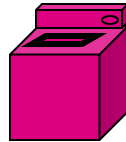
- Pipeline de instrução é semelhante ao uso de uma linha de montagem numa planta industrial.
 - um produto passar por vários estágios de produção
 - produtos em vários estágios podem ser trabalhados simultaneamente.

Processo de Pipelining (exemplo da lavanderia)

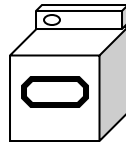
- Ana, Bruno, Carla, Luiz têm roupas sujas a serem lavadas, secadas, dobradas e guardadas



- Lavadora leva 30 minutos



- Secadora leva 30 minutos



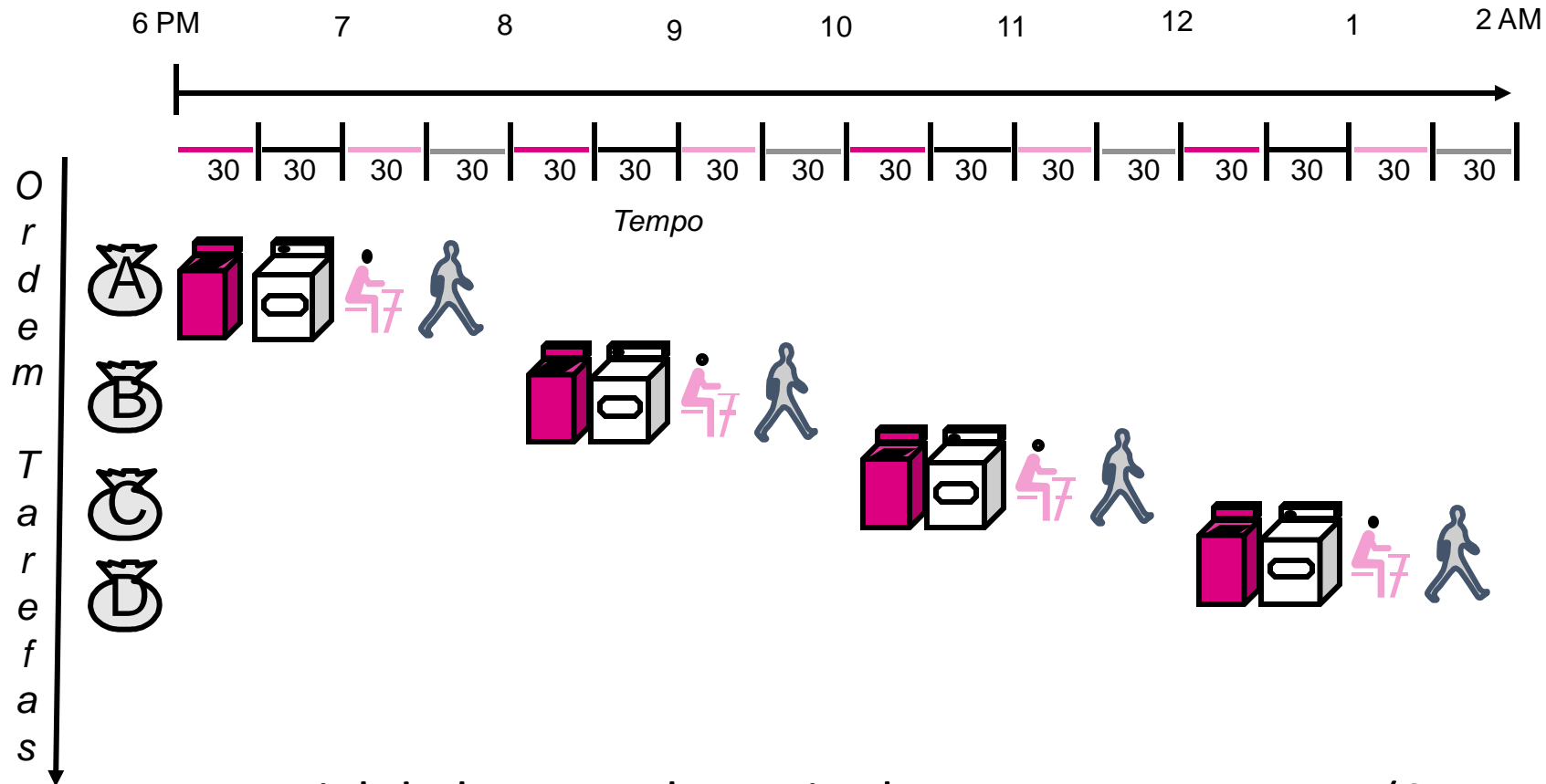
- “Dobrar” leva 30 minutos



- “Guardar” leva 30 minutos

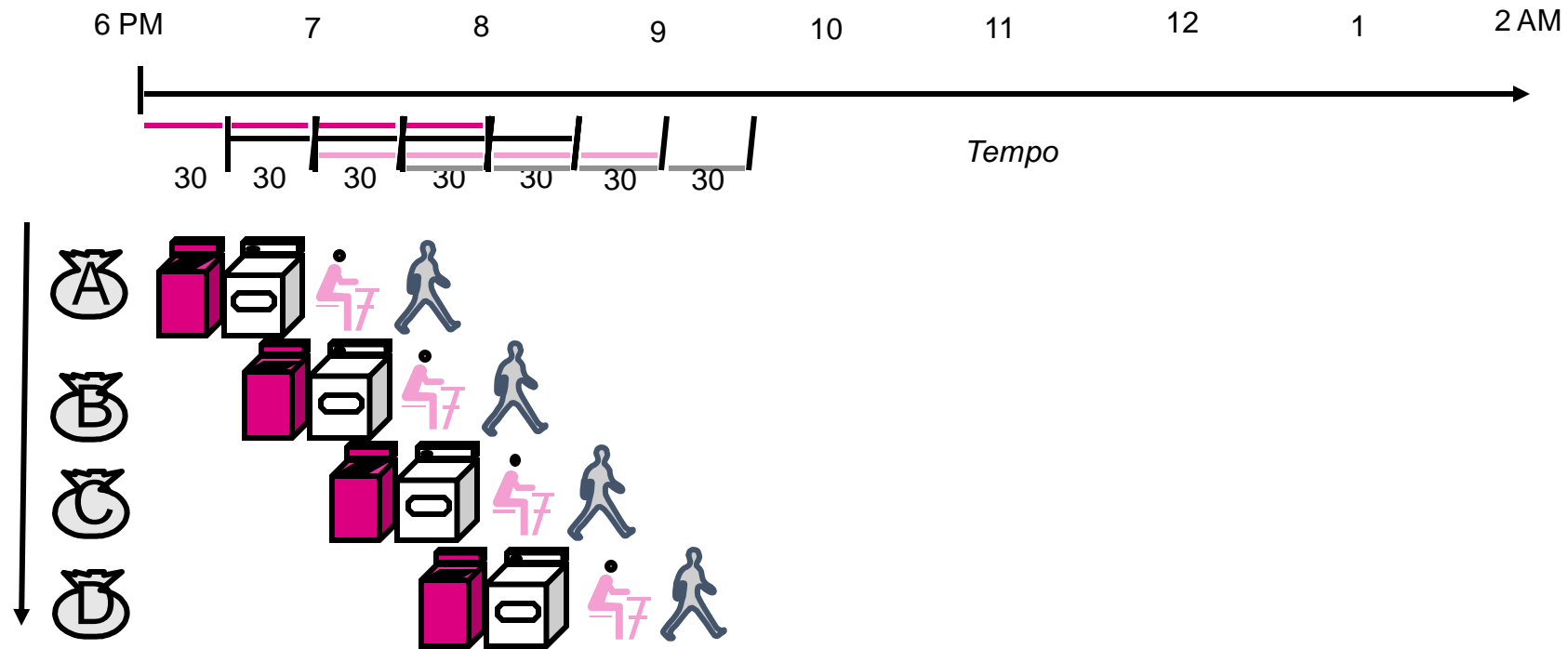


Pipelining (exemplo da lavanderia)



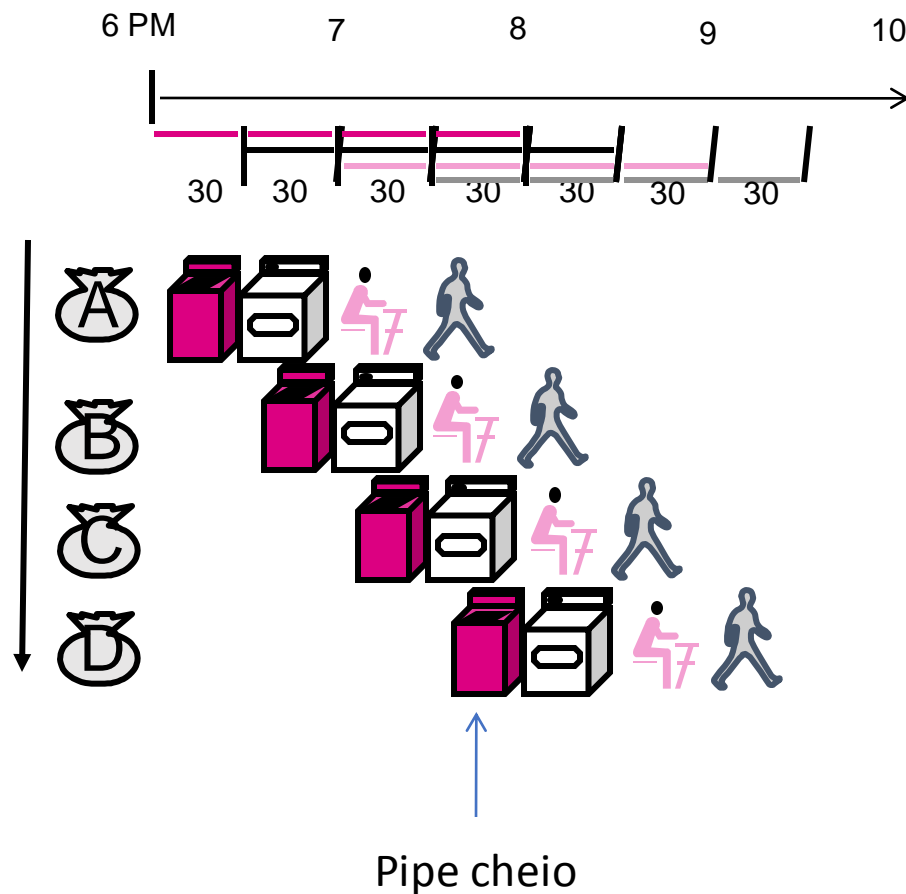
- Processo sequencial de lavagem leva oito horas para os quatro (2 para cada)
- Quanto tempo levaria, utilizando-se pipelining ?

Pipelining (exemplo da lavanderia)



- Utilizando-se a técnica de pipeline consome-se 3,5 horas no processo de lavagem !

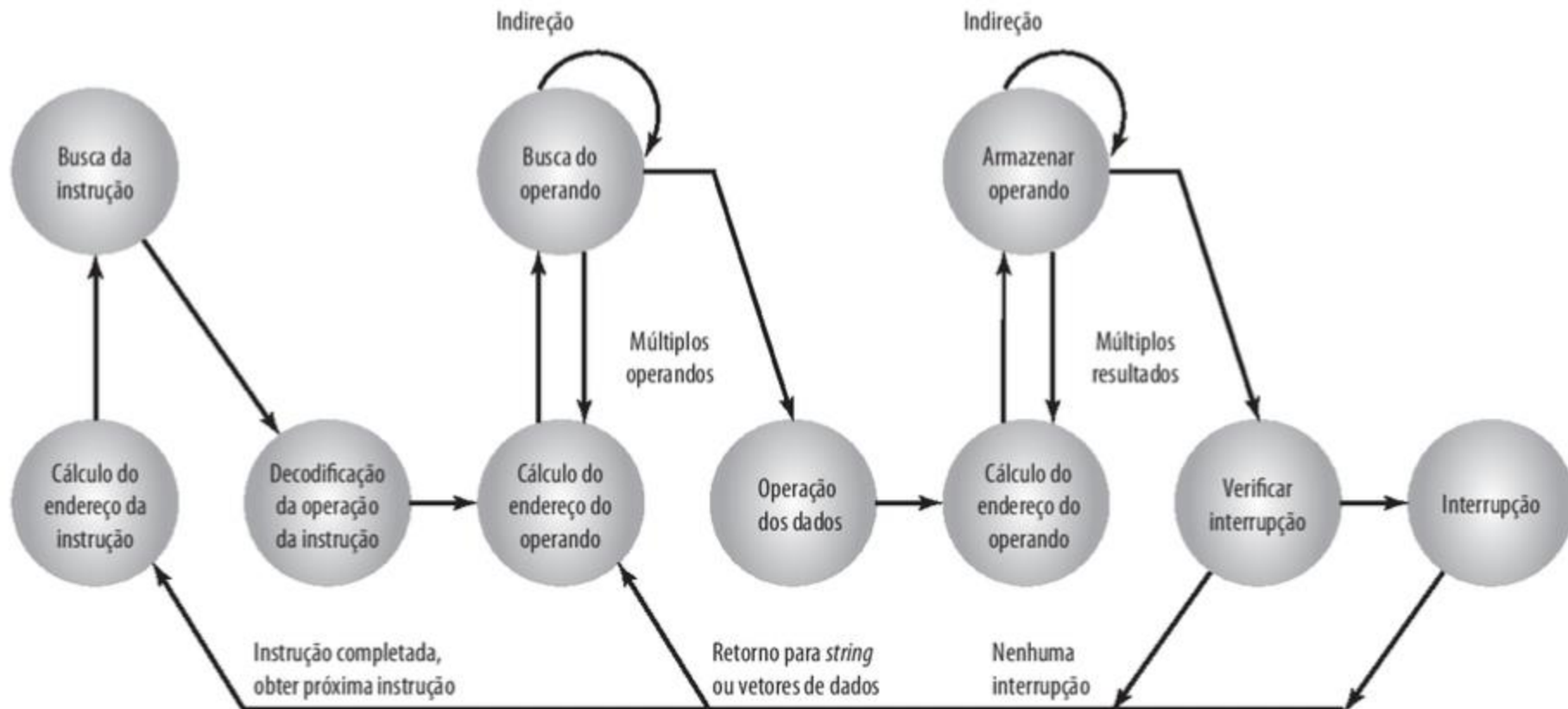
Observações sobre Pipelining



- Pipelining não ajuda a melhorar a latência de uma atividade, mas aumenta o throughput (total)
- Várias tarefas operando em paralelo utilizam recursos diversos
- Aceleração potencial = Número de estágios de pipe (ex: 4)
- Taxa de pipeline limitada pelo estágio mais lento (ex: se lavar demora mais que as outras tarefas)
- Desequilíbrios na duração dos estágios reduz a aceleração
- Tempo para “encher” o pipeline e para “esvaziá-lo” reduz a aceleração
- Pode parar por dependências entre as tarefas

Oportunidade de Aplicar Pipeline

- Diagrama de estado do ciclo da instrução



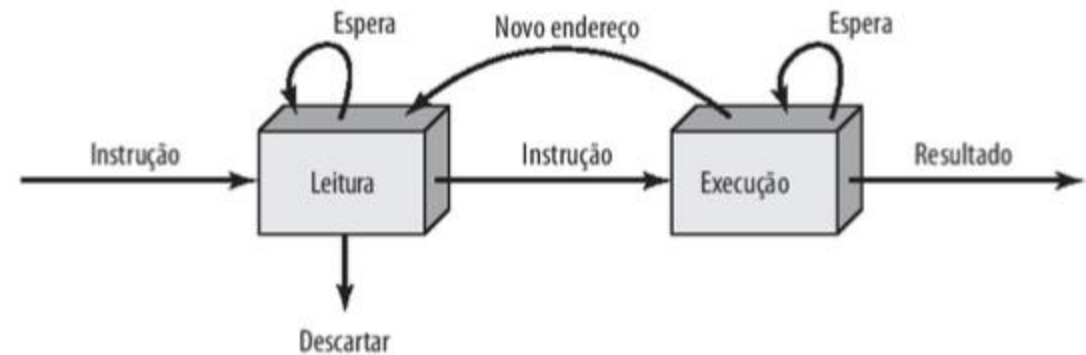
Pipeline

- Pipeline de instrução de dois estágios

- Busca antecipada (prefetch): leitura aproveita ciclo de memória de execução atual para obter a próxima instrução.
- Requer mais registradores.
- Tempos de leitura e execução não iguais:
 - tempo de execução é maior (ler e armazenar operandos)
- Instrução de desvio não conhece o endereço de desvio. Estágio de leitura aguarda o endereço da próxima instrução.
- Regra: estágio de leitura sempre obtêm a próxima instrução de desvio.
- Se desvio ocorrer, instrução é descartada e uma nova é lida.



(a) Visão simplificada



(b) Visão expandida

Pipeline

- Para obter mais velocidade o pipeline deve ter mais estágios:
- Decomposição em 6 estágios:
 - **Buscar instrução (FI, do inglês *Fetch Instruction*)**: ler a próxima instrução esperada em um buffer.
 - **Decodificar instrução (DI)**: determinar o *opcode* e os especificadores dos operandos.
 - **Calcular operandos (CO)**: calcular o endereço efetivo de cada operando de origem. Isto pode envolver endereçamento por deslocamento, registrador indireto, indireto ou outras formas de cálculo de endereço.
 - **Obter operandos (FO, do inglês *Fetch Operands*)**: obter cada operando da memória. Operandos que estão nos registradores não precisam ser lidos da memória.
 - **Executar instrução (EI)**: efetuar a operação indicada e armazenar o resultado, se houver, na posição do operando de destino especificado.
 - **Escrever operando (WO, do inglês *Write Operands*)**: armazenar o resultado na memória.

Pipeline

- Diagrama de tempo para operação do pipeline da instrução
- Operação ideal do pipeline.
- decomposição do processamento da instrução:
 - Busca da Instrução (FI)
 - Decodifica instrução (DI)
 - Calcula Operandos (CO)
 - Busca Operandos (FO)
 - Executa a instrução (EI)
 - Escreve operando (WO)

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

Pipeline

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

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

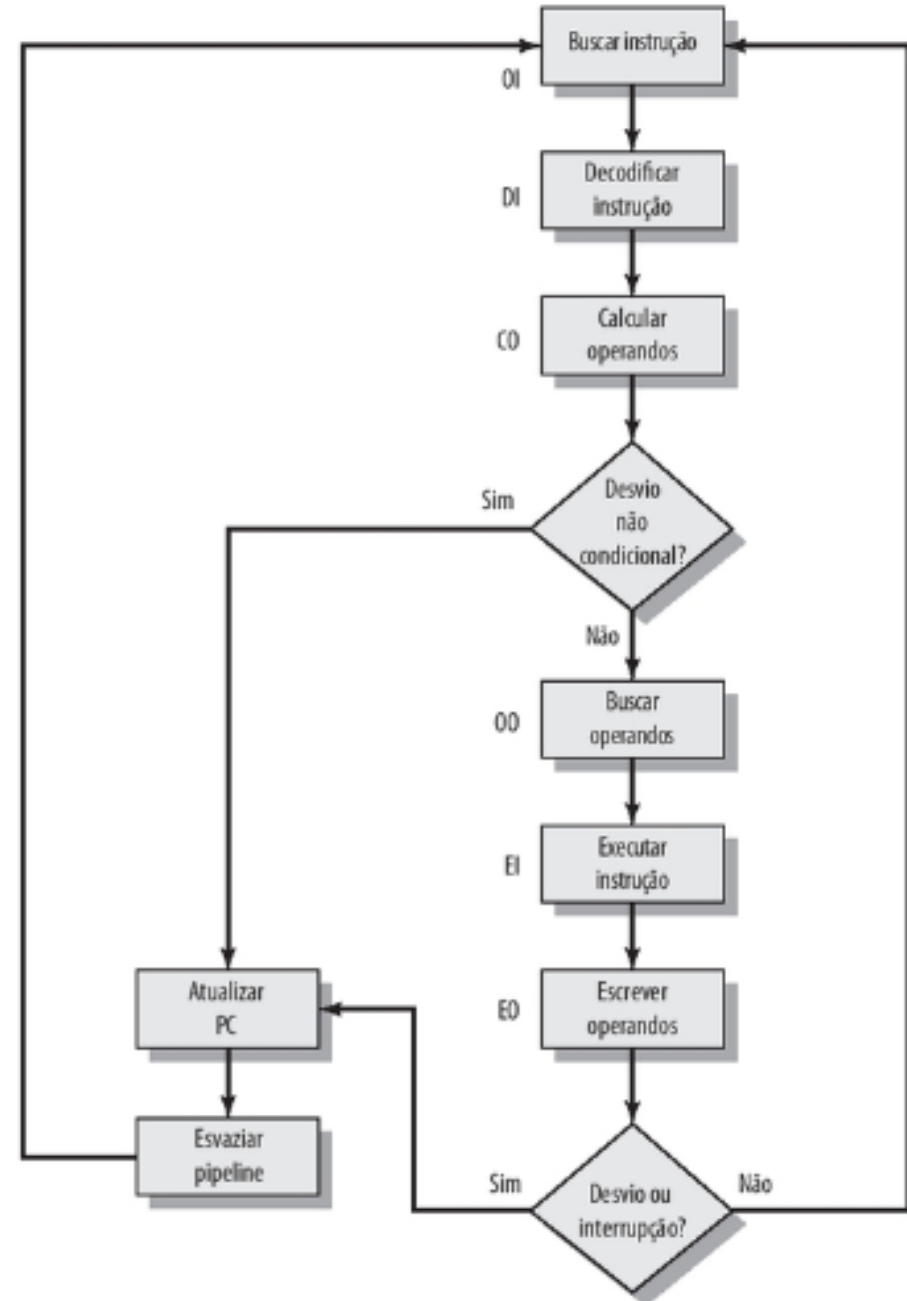
(a) Sem desvios

	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

(b) Com desvios condicionais

Pipeline

- Pipeline de instrução de uma CPU de seis estágios.



Pipeline

- Da discussão anterior seria possível concluir que quanto maior o número de estágios no pipeline, maior será a taxa de execução.
- Dois fatores que frustam este padrão para projetos de alto desempenho:
 1. Em cada estágio do pipeline, existe algum esforço extra envolvido para movimentação de dados de buffer para buffer e para efetuar várias funções de preparações e entregar de dados. Esse esforço extra pode desacelerar sensivelmente o tempo total de execução de uma única instrução. Isso é significativo quando instruções sequenciais são dependentes logicamente umas das outras, ou pelo uso pesado de desvios ou pelas dependências de acesso à memória.
 2. A quantidade de lógica de controle necessária para lidar com dependências de memória e registradores e para otimizar o uso do pipeline aumenta imensamente com o número de estágios. Isso pode levar a uma situação onde a lógica que controla a passagem entre os estágios é mais complexa do que os estágios sendo controlados.

Dúvidas

