

INE5607 – Organização e Arquitetura de Computadores

Unidade Central de Processamento

Aula 18: Processadores multithread e multicore

Prof. Laércio Lima Pilla

laercio.pilla@ufsc.br



Sumário

- Limitações de ILP
- Multithreading
- Multicore
- Considerações finais

LIMITAÇÕES DE ILP

Limitações de ILP

- O que é ILP?
 - *Instruction Level Parallelism*
- A partir de qual tecnologia começamos a ter ILP nos processadores?
 - **Pipeline**
 - Instruções em execução sobreposta
 - CPI próximo de 1
 - **Superscalar/VLIW**
 - Instruções executando em recursos em paralelo
 - IPC acima de 1

Limitações de ILP

- Limitante de ILP
 - **Dependência entre instruções**
 - Resultado anterior
 - Leitura de memória
 - Direção do desvio
 - Etc.

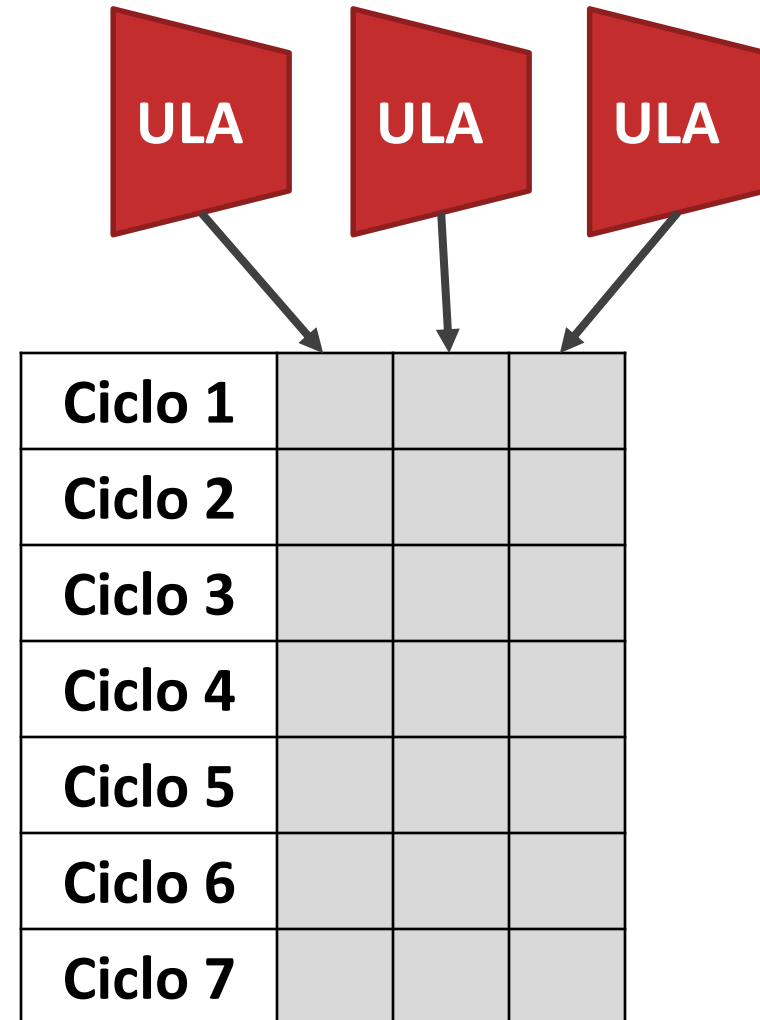
Limitações de ILP

- Outra forma de ver dependências

- Processador com 3 ULAs

- Código abaixo

1. add \$t1, \$a0, \$a1
2. add \$t2, \$t1, \$a2
3. add \$t3, \$t1, \$t3
4. add \$t4, \$t2, \$t3
5. sub \$t5, \$t2, \$t3
6. add \$t6, \$t3, \$a3
7. sub \$t7, \$t3, \$a3
8. add \$t8, \$t6, \$t7



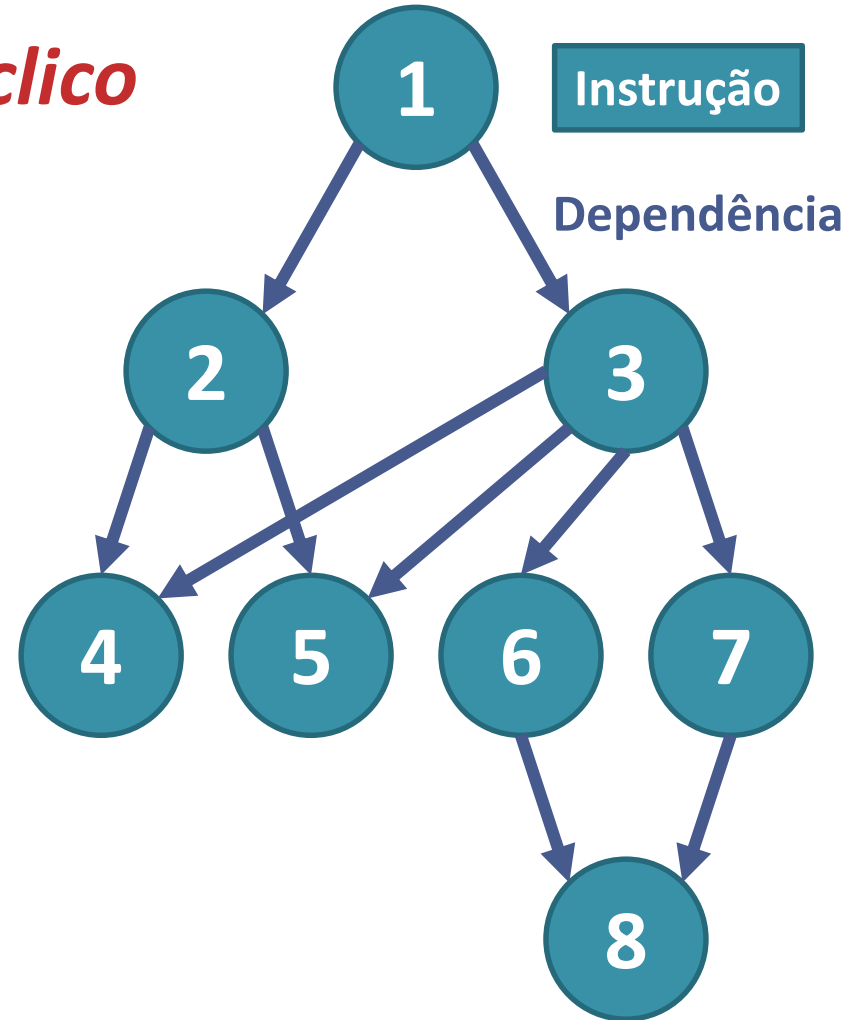
Limitações de ILP

- Código como DAG

- ***Grafo direcionado acíclico***

- *Directed acyclic graph*

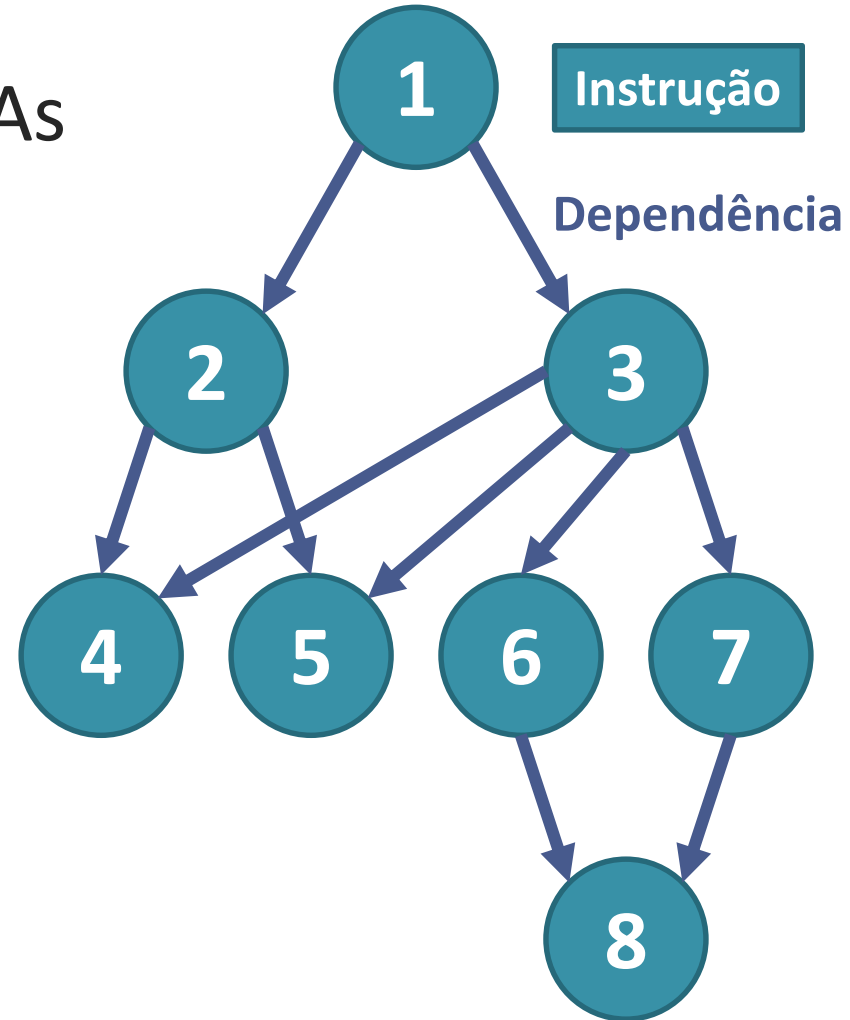
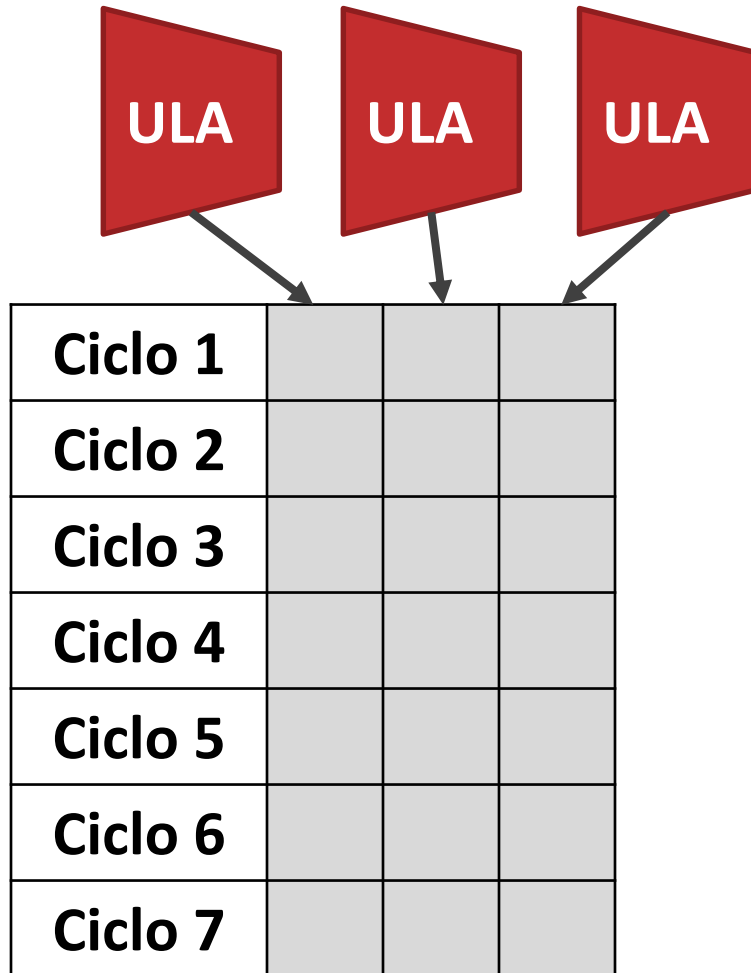
1. add \$t1, \$a0, \$a1
2. add \$t2, \$t1, \$a2
3. add \$t3, \$t1, \$t3
4. add \$t4, \$t2, \$t3
5. sub \$t5, \$t2, \$t3
6. add \$t6, \$t3, \$a3
7. sub \$t7, \$t3, \$a3
8. add \$t8, \$t6, \$t7



Limitações de ILP

- Exemplo

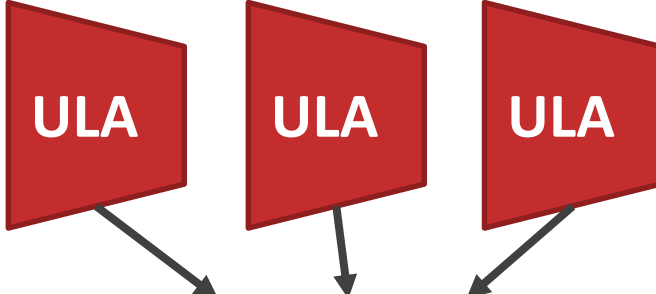
- Processador com 3 ULAs



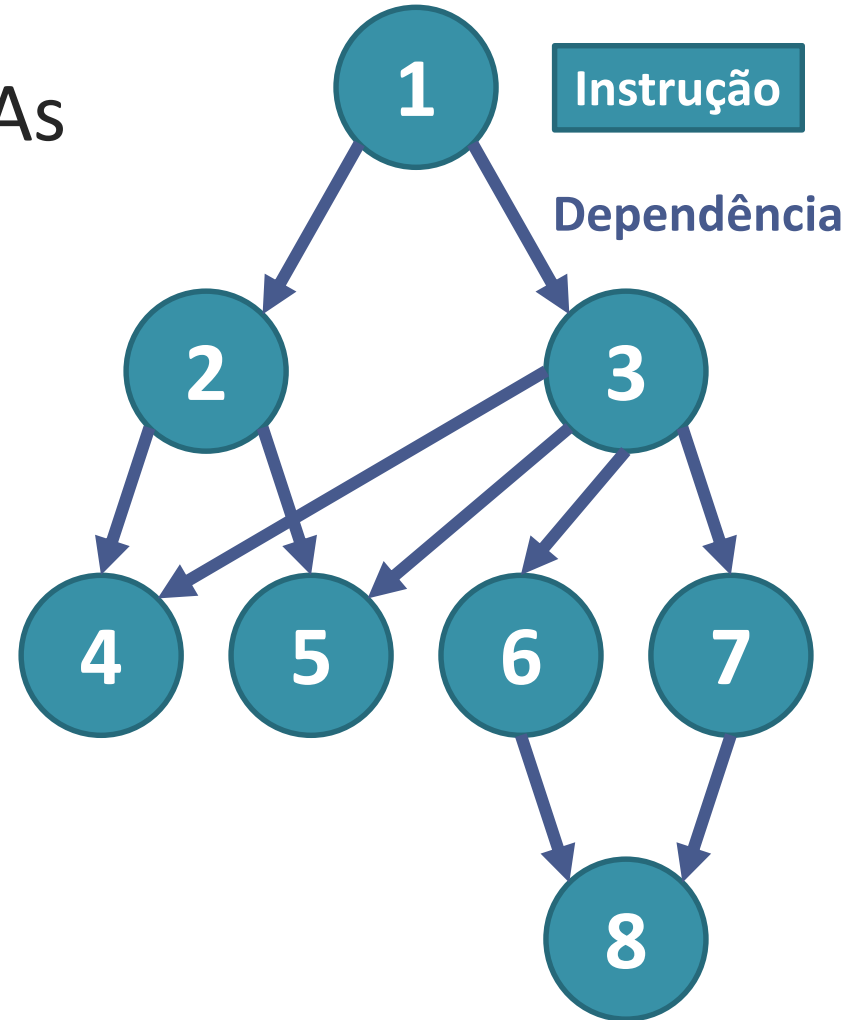
Limitações de ILP

- Exemplo

- Processador com 3 ULAs



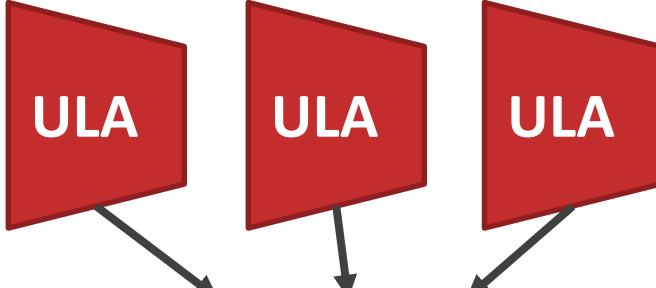
Ciclo 1	1		
Ciclo 2			
Ciclo 3			
Ciclo 4			
Ciclo 5			
Ciclo 6			
Ciclo 7			



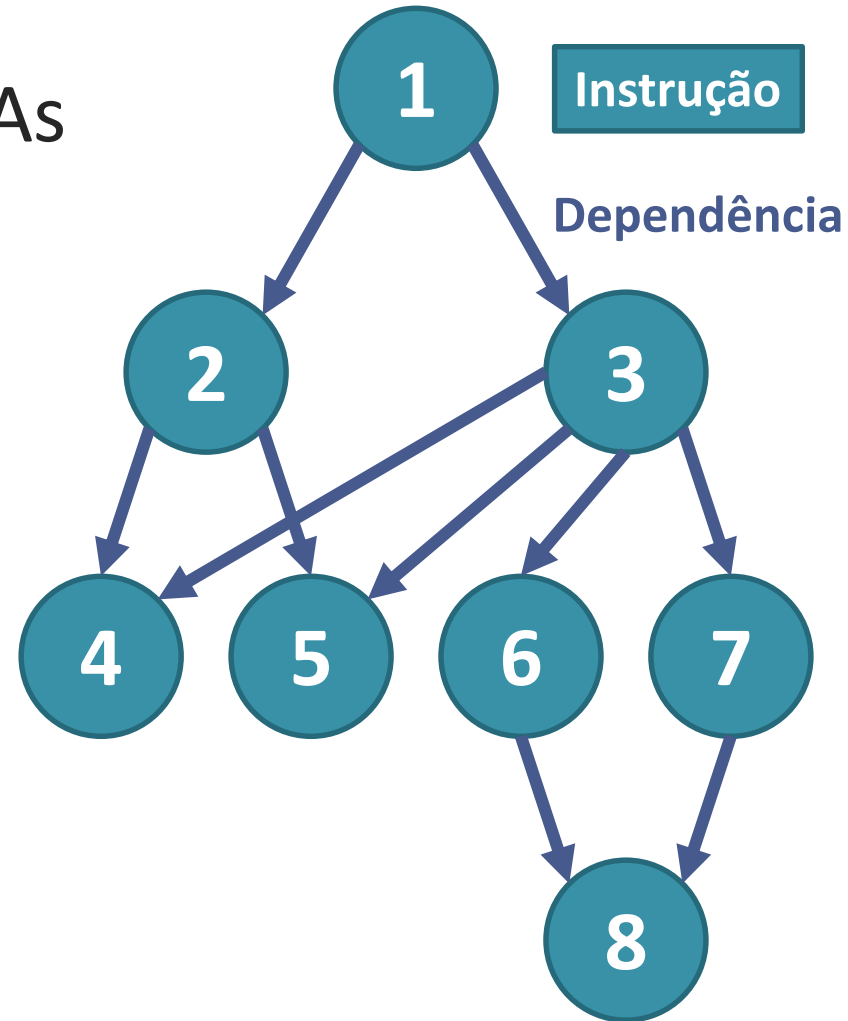
Limitações de ILP

- Exemplo

- Processador com 3 ULAs



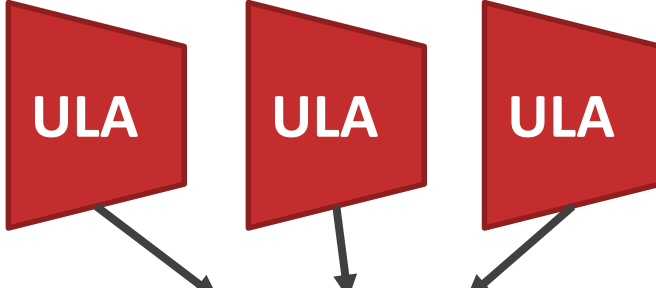
Ciclo 1	1		
Ciclo 2	2	3	
Ciclo 3			
Ciclo 4			
Ciclo 5			
Ciclo 6			
Ciclo 7			



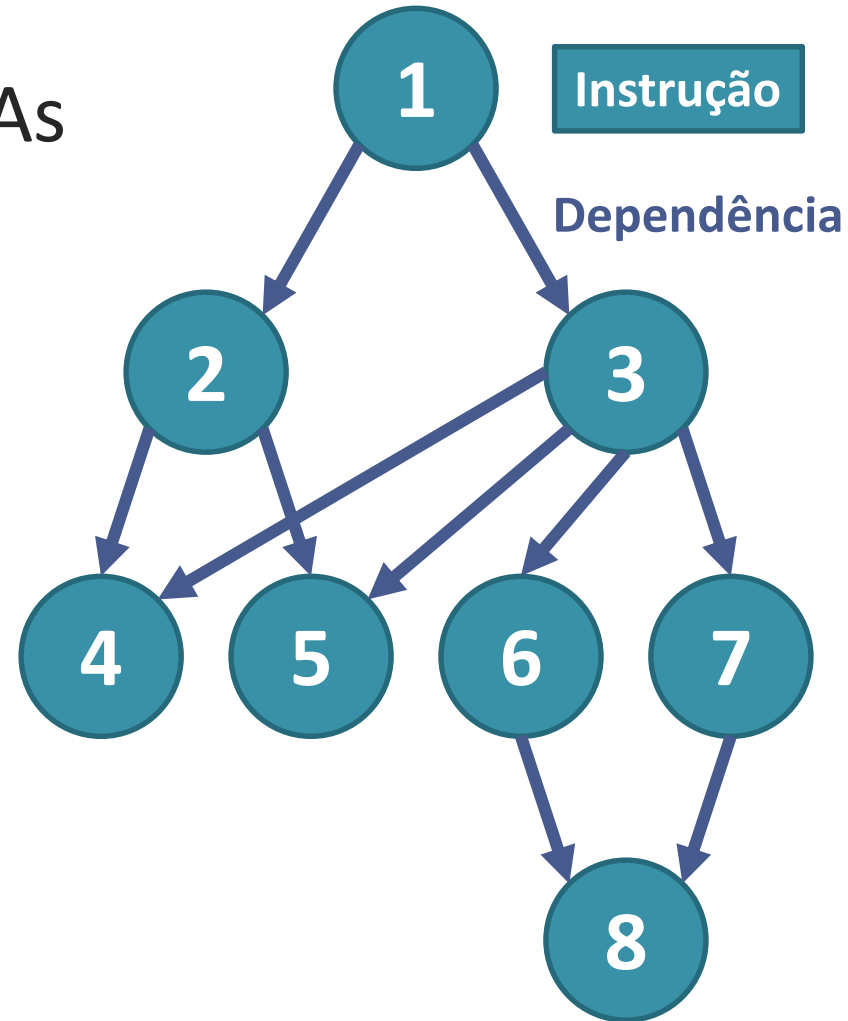
Limitações de ILP

- Exemplo

- Processador com 3 ULAs



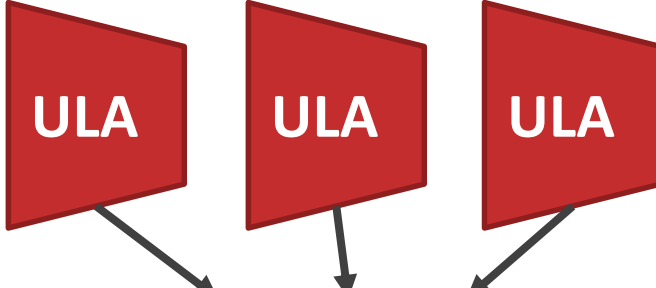
Ciclo 1	1		
Ciclo 2	2	3	
Ciclo 3	4	5	6
Ciclo 4			
Ciclo 5			
Ciclo 6			
Ciclo 7			



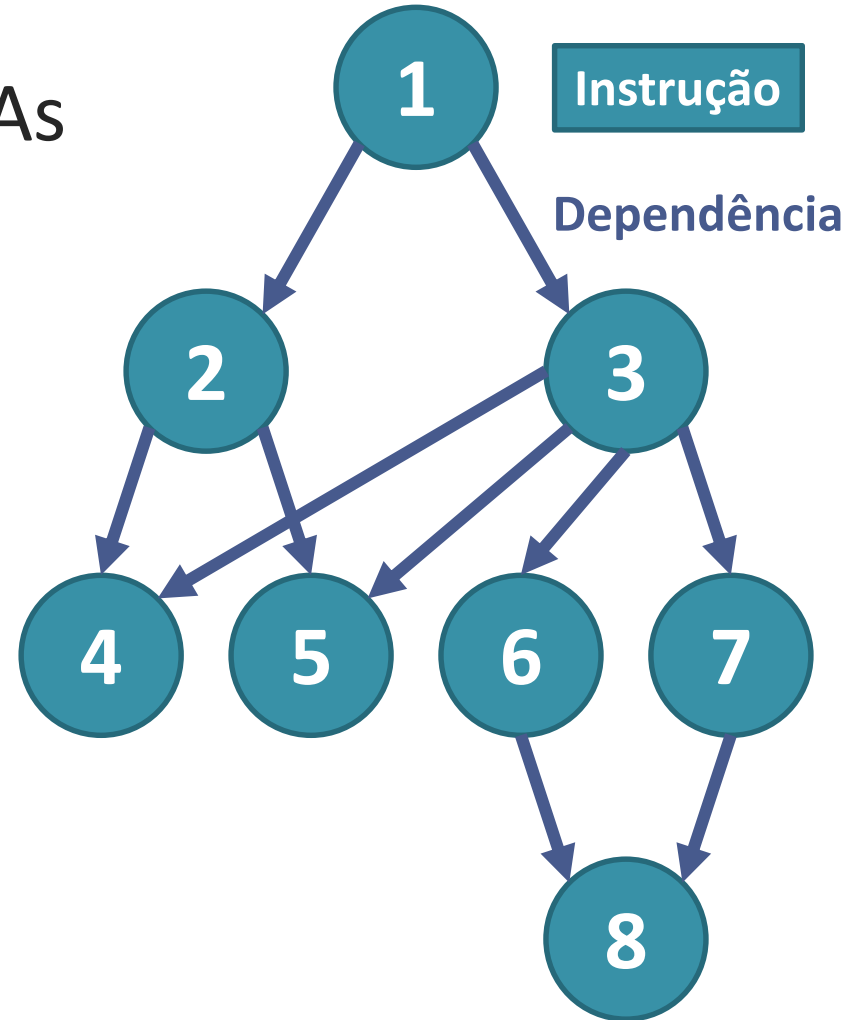
Limitações de ILP

- Exemplo

- Processador com 3 ULAs



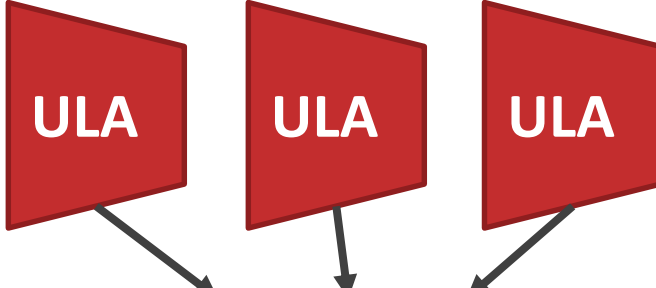
Ciclo 1	1		
Ciclo 2	2	3	
Ciclo 3	4	5	6
Ciclo 4	7		
Ciclo 5			
Ciclo 6			
Ciclo 7			



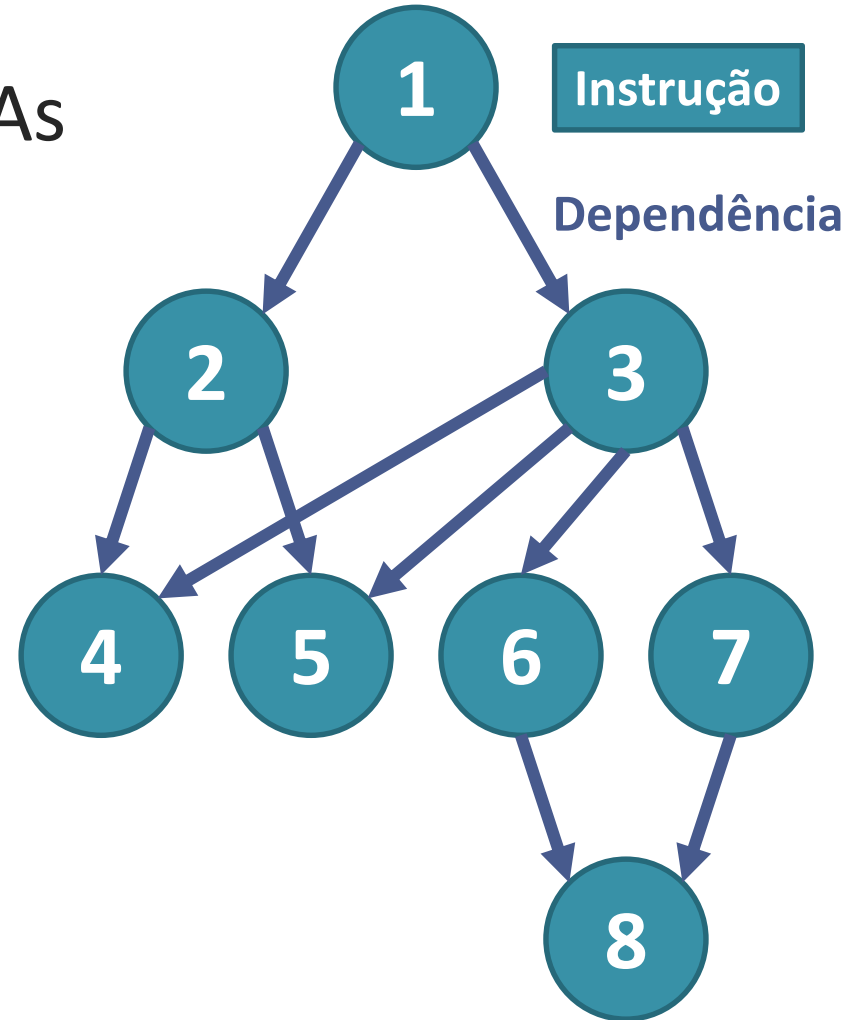
Limitações de ILP

- Exemplo

- Processador com 3 ULAs



Ciclo 1	1		
Ciclo 2	2	3	
Ciclo 3	4	5	6
Ciclo 4	7		
Ciclo 5	8		
Ciclo 6			
Ciclo 7			



Limitações de ILP

- Exemplo

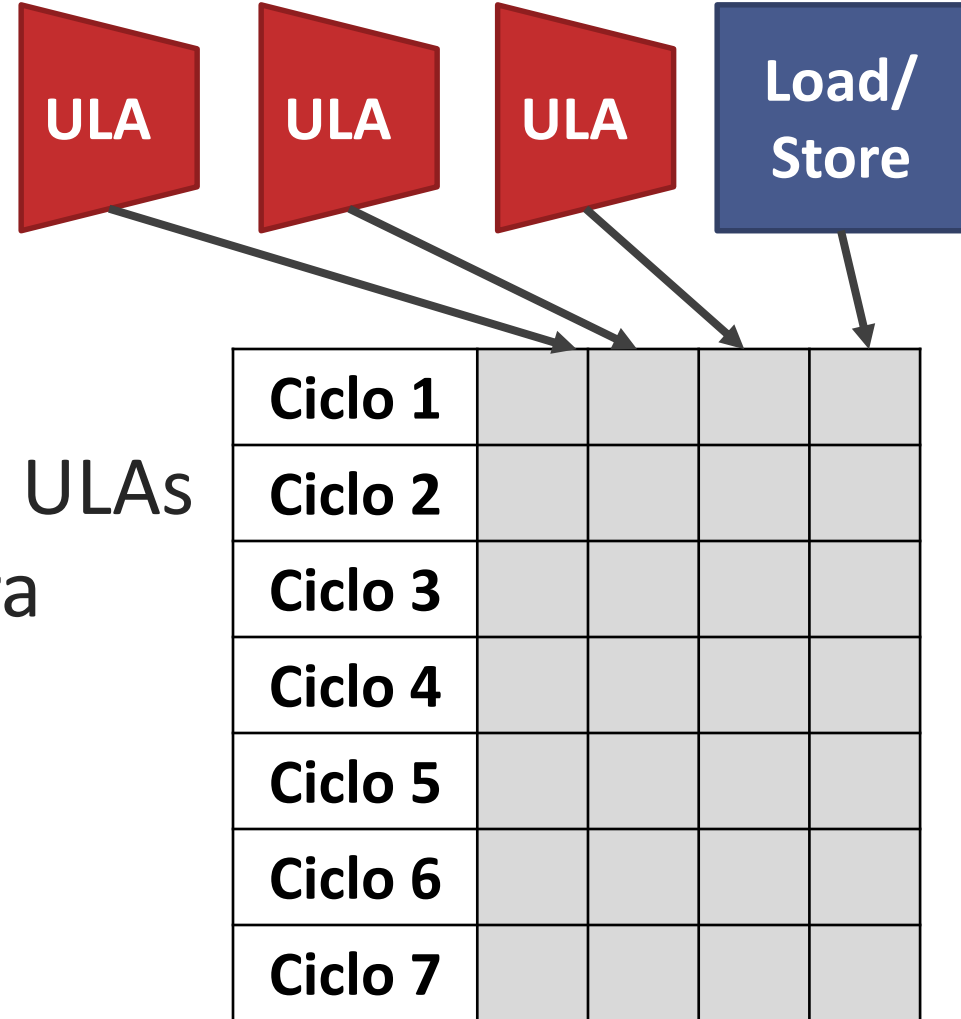
- Processador com 3 ULAs

- Número de instruções: 8
 - Tempo de execução: 5 ciclos
 - $IPC = 8/5 = 1,6$
 - $IPC\ ideal = 3\ (\#ULAs)$
 - Utilização = $1,6/3 = 53\%$

- Processador subutilizado**

Ciclo 1	1		
Ciclo 2	2	3	
Ciclo 3	4	5	6
Ciclo 4	7		
Ciclo 5	8		
Ciclo 6			
Ciclo 7			

Limitações de ILP



- Exemplo
 - Processador com 3 ULAs e uma unidade para load/store

Limitações de ILP

- Exemplo

- Processador com 3 ULAs

- Número de instruções: 8
- Tempo de execução: 5 ciclos
- $IPC = 8/5 = 1,6$
- $IPC\ ideal = 4$ (#unidades)
- $Utilização = 1,6/4 = 40\%$

– **Processador subutilizado**

Ciclo 1	1			
Ciclo 2	2	3		
Ciclo 3	4	5	6	
Ciclo 4	7			
Ciclo 5	8			
Ciclo 6				
Ciclo 7				

MULTITHREADING

Multithreading

- Problema
 - **Processador subutilizado**
 - Recursos disponíveis desperdiçados
- Solução
 - Colocar **mais de um fluxo de instruções** para executar
 - Instruções concorrem pelos recursos disponíveis
 - **Maior utilização do processador**
 - **Não aumenta desempenho sequencial!**

Multithreading

- Um fluxo de execução vs. múltiplos fluxos
 - A = instruções da thread A

A	A		
A			
A	A	A	A
A	A		A

Pipeline superescalar

A	A	B	B
A	C	B	D
C	C	B	
A	C	A	A
C	B	B	D
D	A	D	D
A	A	D	A

Multithreading

Multithreading

- O que é uma thread?
 - Qual a diferença para um processo?
 - E para um programa?

Multithreading

- Gostinho de Sistemas Operacionais

INE5611 – Sistemas Operacionais

Gerência de Processos e Threads

Threads e compartilhamento de dados

Prof. Márcio Castro

marcio.castro@inf.ufsc.br



Multithreading

- Gostinho de Sistemas Operacionais

8

Conceito de processo

- **Programa**

- Entidade estática e permanente
- Sequência de instruções
- Passivo sob o ponto de vista do S.O.

- **Processo**

- Abstração que representa um **programa em execução**
- Entidade dinâmica: altera seu estado ao longo de sua execução
- Composto pelo programa (código), dados e contexto (valores)

Multithreading

- Gostinho de Sistemas Operacionais

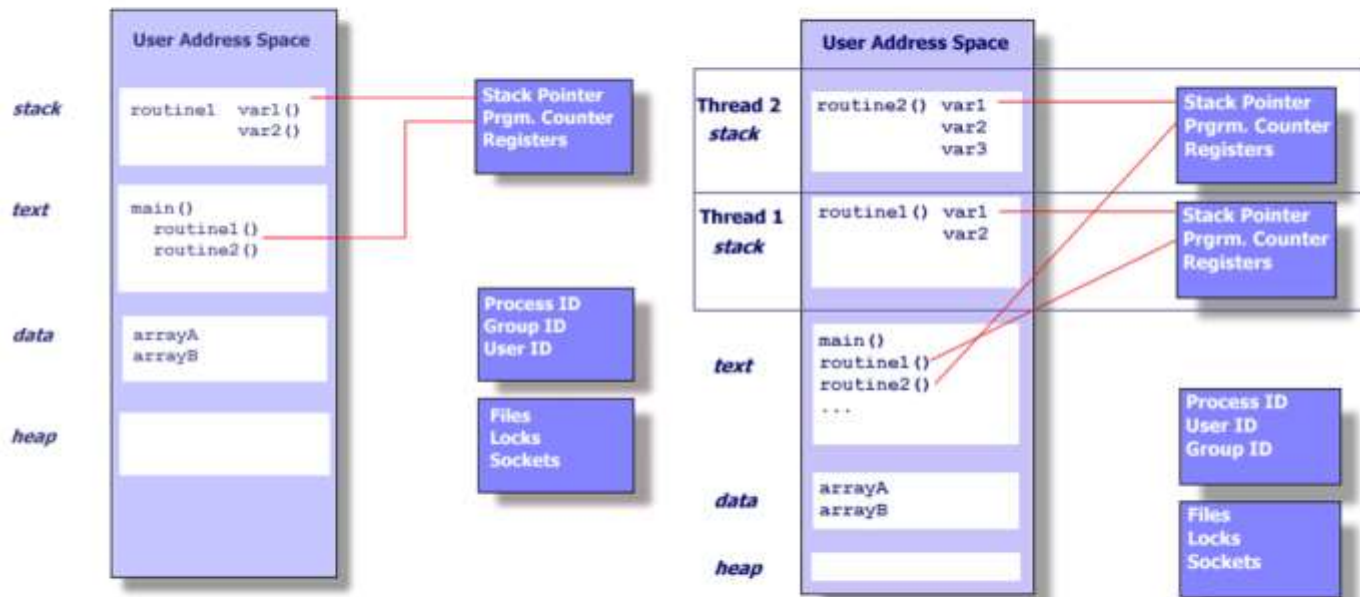
Threads e compartilhamento de dados

Itens por processo	Itens por thread
<ul style="list-style-type: none">- Espaço de endereçamento- Variáveis globais- Arquivos abertos- Informações de contabilidade do processo	<ul style="list-style-type: none">- Contador de <u>prog.</u> (PC)- Registradores- Pilha- Estado

Multithreading

- Gostinho de Sistemas Operacionais

Threads e compartilhamento de dados



Um processo UNIX

Threads em um processo UNIX

Multithreading

- O que é um thread?
 - Fluxo/linha de execução de um processo



– Em termos de hardware

- ***Program Counter***
- **Banco de registradores** (inclui *Stack Pointer*)
 - Replicando ambos, podemos ter mais de um thread em execução

Multithreading

- Multithreading resulta em **CPUs virtuais**
 - Hardware tem uma CPU com multithreading
 - Recursos de controle e datapath compartilhados
 - **Sistema operacional vê duas CPUs**
- Qual a diferença entre multithreading e o SO escolhendo qual de duas threads deve executar?
- Quando colocar threads para executar?

Multithreading

- IMT: *Interleaved Multithreading*
 - **Entrelaçado** ou de grão fino
 - A **cada ciclo muda a thread** cujas instruções serão despachadas
 - **Todas threads executam de forma mais lenta**
 - Focado em vazão
 - Servidores ou processamento gráfico

A	A		
	B	B	B
C	C		
D		D	D
A	A	A	
	B	B	
C	C		

Multithreading

- BMT: *Blocked Multithreading*
 - Em **blocos** ou de grão grosso
 - **Muda a thread** em execução a cada **evento de alta latência**
 - E.g., acesso à memória principal
 - **Troca de threads mais barata**
 - Sobrecusto de encher o pipeline

A	A		
A			
		B	B
		B	B
C	C		
D		D	D
		D	D

Multithreading

- SMT: *Simultaneous Multithreading*

- Simultâneo

- Motivado pela existência de **mais unidades funcionais do que um thread consegue usar sozinho**

- Controle mais complexo

- Diferenciar de quem é a instrução

A	A	B	B
A	C	B	D
C	C	B	
A	C	A	A
C	B	B	D
D	A	D	D
A	A	D	A

Multithreading

- Comparação

A	A		
A			
A	A	A	A
A	A		A

Pipeline superescalar

A	A		
	B	B	B
C	C		
D		D	D
A	A	A	
	B	B	
C	C		

IMT

A	A		
A			
		B	B
		B	B
C	C		
D		D	D
		D	D

BMT

A	A	B	B
A	C	B	D
C	C	B	
A	C	A	A
C	B	B	D
D	A	D	D
A	A	D	A

SMT

Multithreading

- Desempenho

- Qual a **utilização** dos processadores nos dois casos abaixo?
- Qual a **perda de desempenho** da thread A?

A	A	A	
A	A		
		A	A
A	A	A	A
	A		
	A		
A	A		A

A	A	A	B
A	A		B
	B	A	B
A	A	A	A
B	B	B	A
B	A		B
A	A		B

Multithreading

- Exemplos de multithreading
 - Intel Hyper-Threading (HT)
 - SMT
 - Desde Pentium 4 (2002)
 - Até **duas threads em execução**
 - Muitas vezes recomendado ser desligado por empresas



Multithreading

But traditional processor cores can only perform
one task at a time



<http://www.intel.com/content/www/us/en/architecture-and-technology/hyper-threading/hyper-threading-technology-video.html>

Multithreading

Intel® HT Technology enables each core to perform
two tasks simultaneously



<http://www.intel.com/content/www/us/en/architecture-and-technology/hyper-threading/hyper-threading-technology-video.html>

MULTICORE

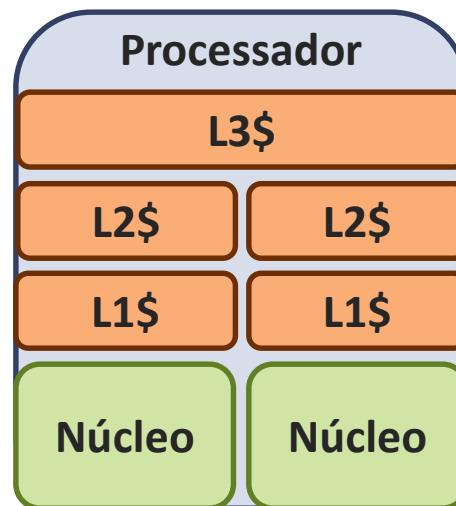
Multicore

- E se ao invés de ter apenas uma CPU, meu processador tivesse **duas (ou mais) CPUs reais e independentes?**

A	A	B	B
A		B	
		B	B
A	A	B	
A	A		
		B	B
A	A		B

Multicore

- CPUs independentes em um mesmo processador
 - **Cores ou núcleos**
 - Possuem datapaths e controles separados
 - Podem compartilhar algum nível de memória cache

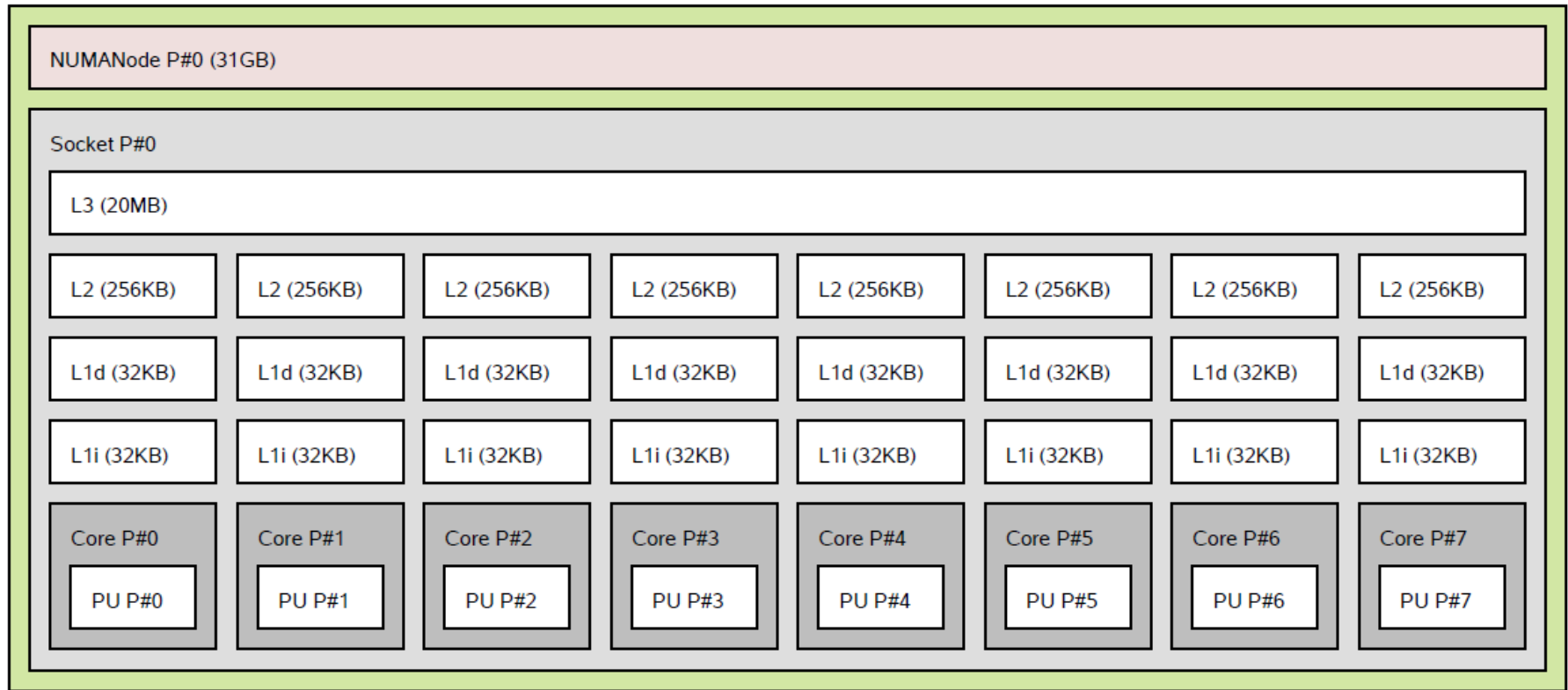


Multicore

- Exemplo: Intel Xeon E5-4640
 - Oito núcleos

Machine (744GB)

Group0 (62GB)



Multicore

- Motivação
 - **Aumentos de frequência não eram mais possíveis**
 - Design de processadores muito complexo
- Solução
 - **Múltiplos processadores com menor frequência de relógio**
 - Processadores mais simples
 - Cada um usando menos transistores

Multicore

- Desempenho sequencial
 - Não é beneficiado
- Desempenho paralelo
 - **Maior vazão**
 - Ótimo para programas com paralelismo
 - Tarefas ou dados independentes
 - **Necessita de programação paralela**
 - **Necessita de programação paralela**
 - **Necessita de programação paralela**
 - **Necessita de programação paralela**

Multicore

- **Ganho de desempenho ideal = nº de núcleos**
- Ganho de desempenho real
 - Dependente do paralelismo da aplicação
 - **Lei de Amdhal**

INTERLÚDIO: LEI DE AMDAHL

Lei de Amdahl

- **Lei de Amdahl**

- **Ganho máximo** com aprimoramento de um sistema
- **Divide algoritmo em parte sequencial e paralela**
 - Parcela serial do algoritmo = **B** $\in [0,1]$
 - Número de threads = **n**

$$\text{Tempo}(n) = \text{Tempo}(1) * (B + (1/n) * (1 - B))$$

Lei de Amdahl

- ***Speedup***

- Ganho de desempenho com paralelismo

$$\text{Speedup}(n) = \text{Tempo}(1) / \text{Tempo}(n)$$

- Com Amdahl

$$\text{Tempo}(n) = \text{Tempo}(1) * (B + (1/n) * (1 - B))$$

logo

$$\text{Speedup}(n) = 1 / (B + (1/n) * (1 - B))$$

Lei de Amdahl

- Exemplo

- Um programa executa em 10 segundos em um único núcleo e em 4 segundos usando quatro núcleos. Qual o speedup obtido?
- $\text{Speedup}(n) = \text{Tempo}(1) / \text{Tempo}(n)$
- $\text{Speedup}(4) = 10 \text{ s} / 4 \text{ s} = 2,5$

Lei de Amdahl

- Exemplo
 - Um programa tem 90% de seu código paralelizável. Qual o ganho de desempenho possível usando
 - 10 núcleos?
 - 100 núcleos?
 - Infinitos núcleos?
 - **$\text{Speedup}(n) = 1/(B+(1/n)*(1-B))$**

Lei de Amdahl

- Lei de Amdahl sem paralelismo
 - Divisão da aplicação em partes
 - 90% em CPU, 10% em disco
 - 70% em CPU, 20% em rede, 10% em memória
 - 40% no telefone, 60% no servidor

Lei de Amdahl

- Exemplo

- O processo de abrir um filme em um computador leva 15 segundos. 70% desse tempo é gasto no disco. Se eu comprar um disco 2 vezes melhor, quanto tempo levará abrir um filme?

- $T = 15 \text{ s} * (0,3 + (1/2)*0,7) = 15 \text{ s} * 0,65 = 9,75 \text{ s}$

Lei de Amdahl

- **Eficiência**

- O quanto é aproveitado da plataforma paralela

- $$\text{Eficiência}(n) = \text{Speedup}(n)/n$$**

- Número entre 0 e 1

- Fora caso de speedup superlinear

FIM DO INTERLÚDIO

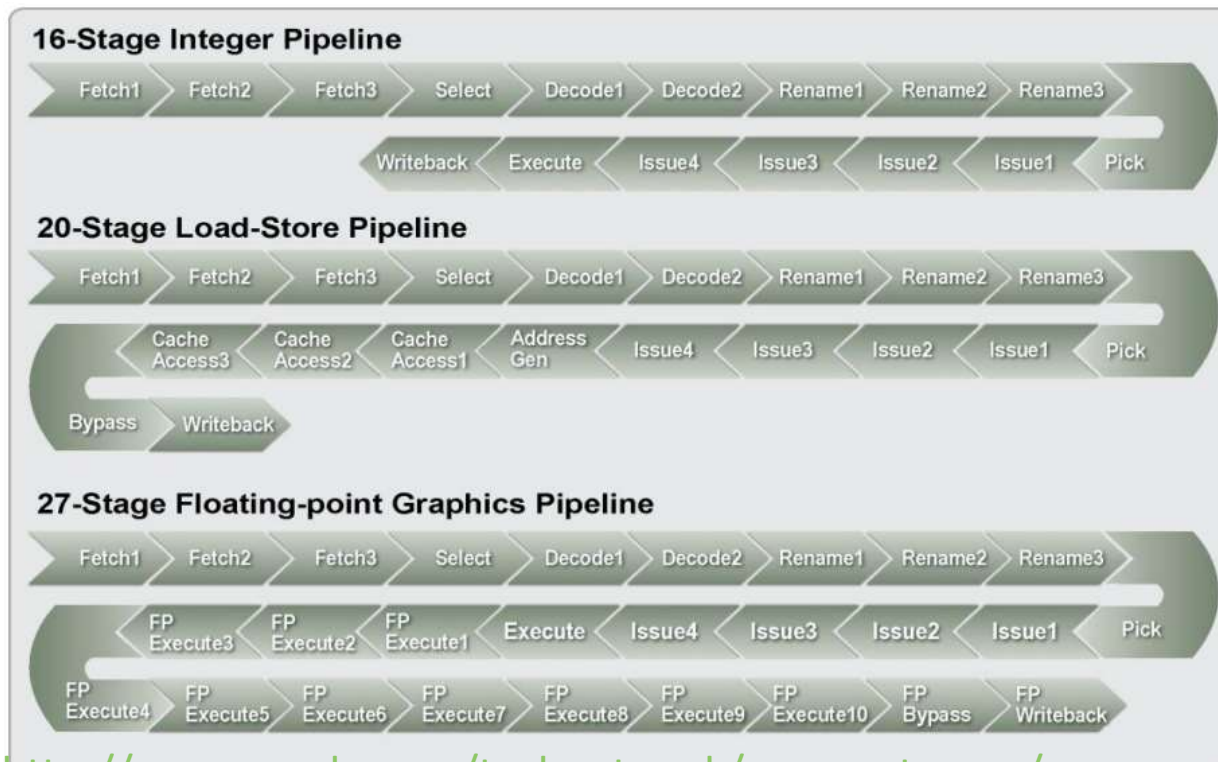
Multicore

- Exemplo de processador multicore
 - Oracle Sparc T5
 - Foco em servidores
 - **16 núcleos**
 - Até **128 threads**
 - 8 por core
 - **Mistura de IMT e SMT**
 - Duas threads no mesmo ciclo
 - Diferentes threads no ciclo seguinte



Multicore

- Exemplo de processador multicore
 - Oracle Sparc T5
 - Pipeline com execução fora de ordem



<http://www.oracle.com/technetwork/server-storage/sun-sparc-enterprise/documentation/o13-024-sparc-t5-architecture-1920540.pdf>

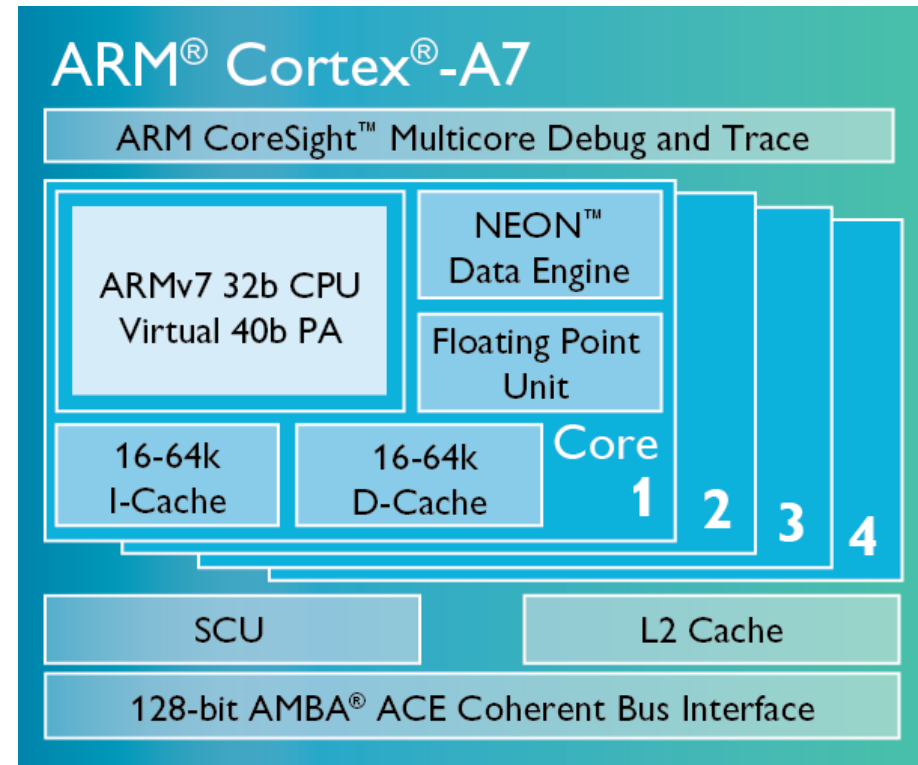
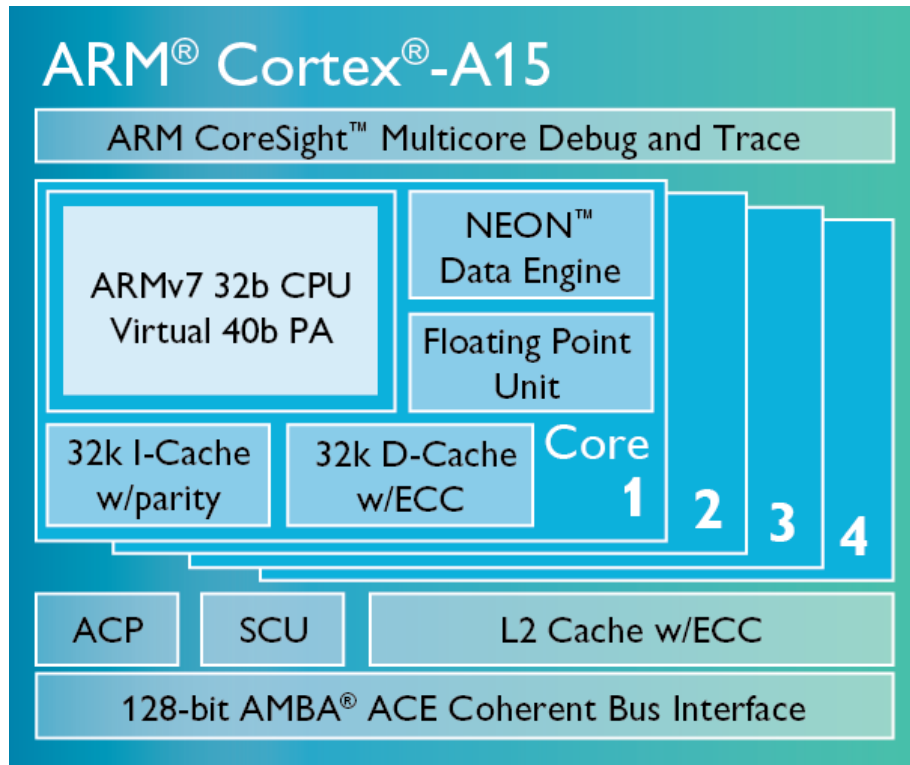
Multicore

- Exemplo de processador multicore
 - ARM big.LITTLE
 - **Multicore assimétrico/heterogêneo**
 - **big**: Núcleos Cortex A-15
 - Desempenho
 - **LITTLE**: Núcleos Cortex A-7
 - Eficiência energética
 - Oito núcleos ao todo
 - Quatro de cada
 - Sem multithreading



Multicore

- Exemplo de processador multicore
 - ARM big.LITTLE



CONSIDERAÇÕES FINAIS

Considerações finais

- **Multithreading**
 - Aumento da utilização do processador
 - Núcleos virtuais
- **Multicore**
 - Múltiplas CPUs em uma só
 - Núcleos reais
- **Lei de Amdahl**
 - Limites de ganhos com paralelismo

INE5607 – Organização e Arquitetura de Computadores

Unidade Central de Processamento

Aula 18: Processadores multithread e multicore

Prof. Laércio Lima Pilla

laercio.pilla@ufsc.br

