

Arquitetura de Computadores

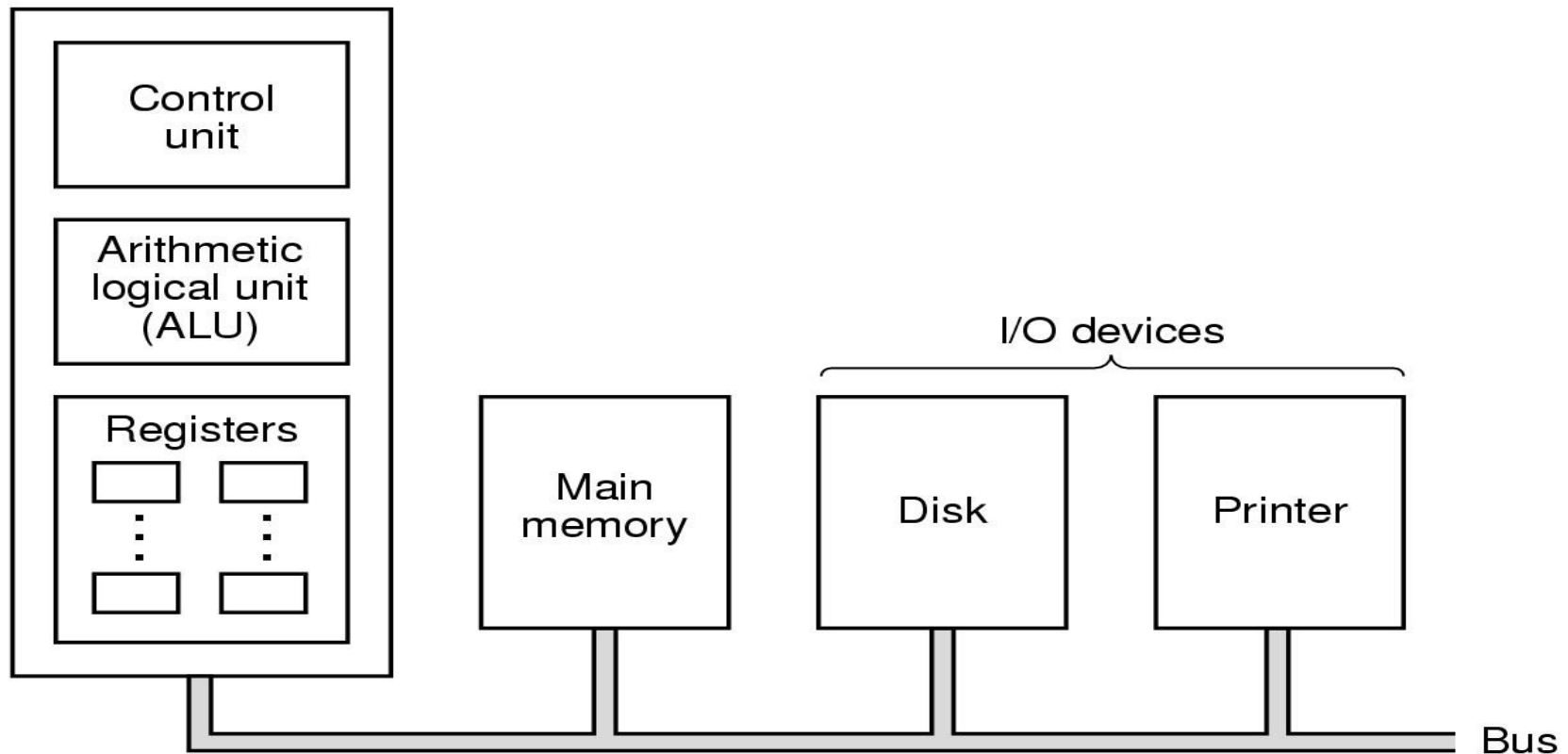
Prof. João Bosco Jr.

(CPU)

Modelo do Computador Von Neumann

- Processador
- Memórias
- E/S
- Barramentos

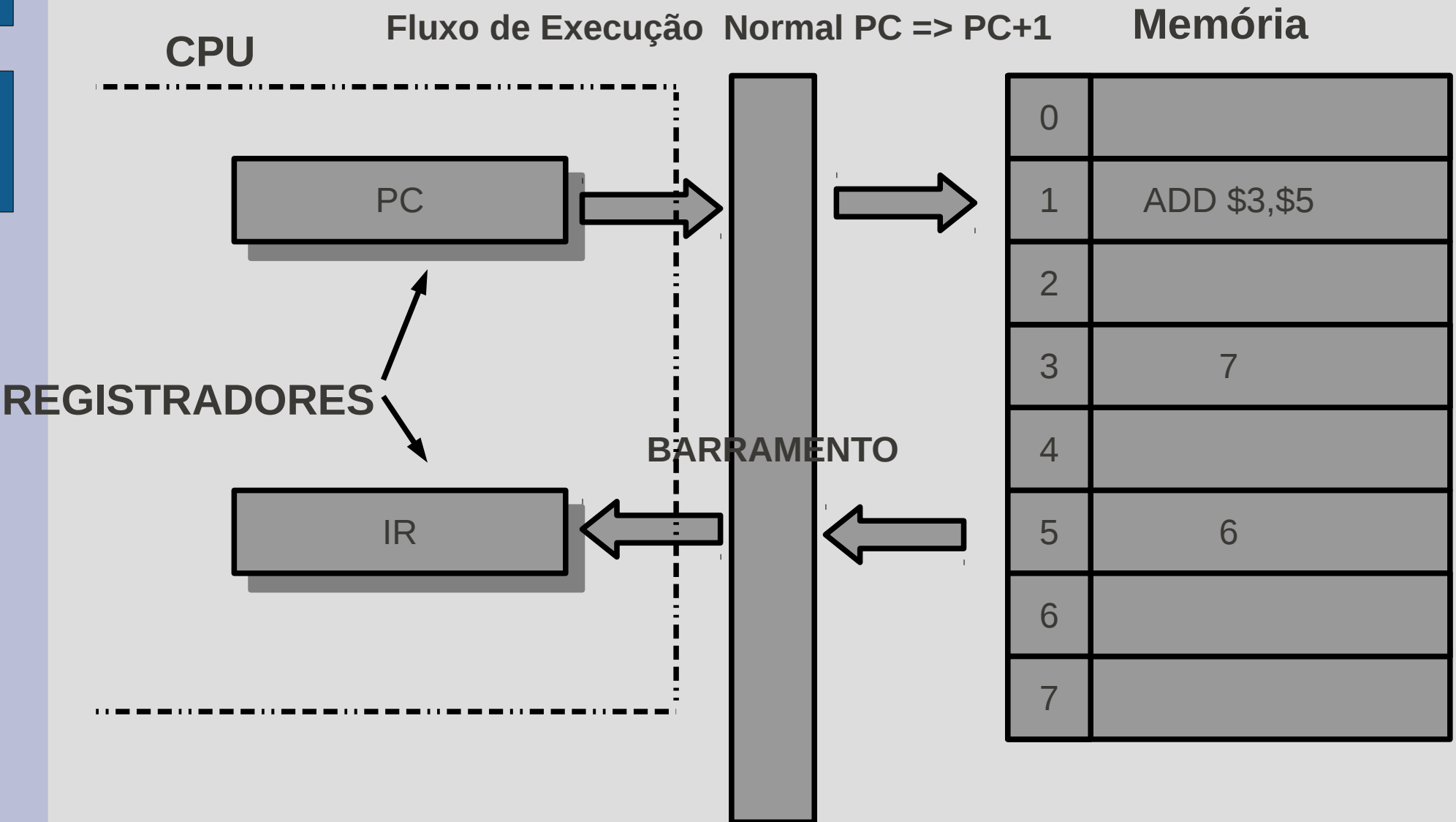
Central processing unit (CPU)



Simulação Contador

<http://courses.cs.vt.edu/csonline/MachineArchitecture/Lessons/CPU/countprogram.html>

Modelo do Computador Von Neumann



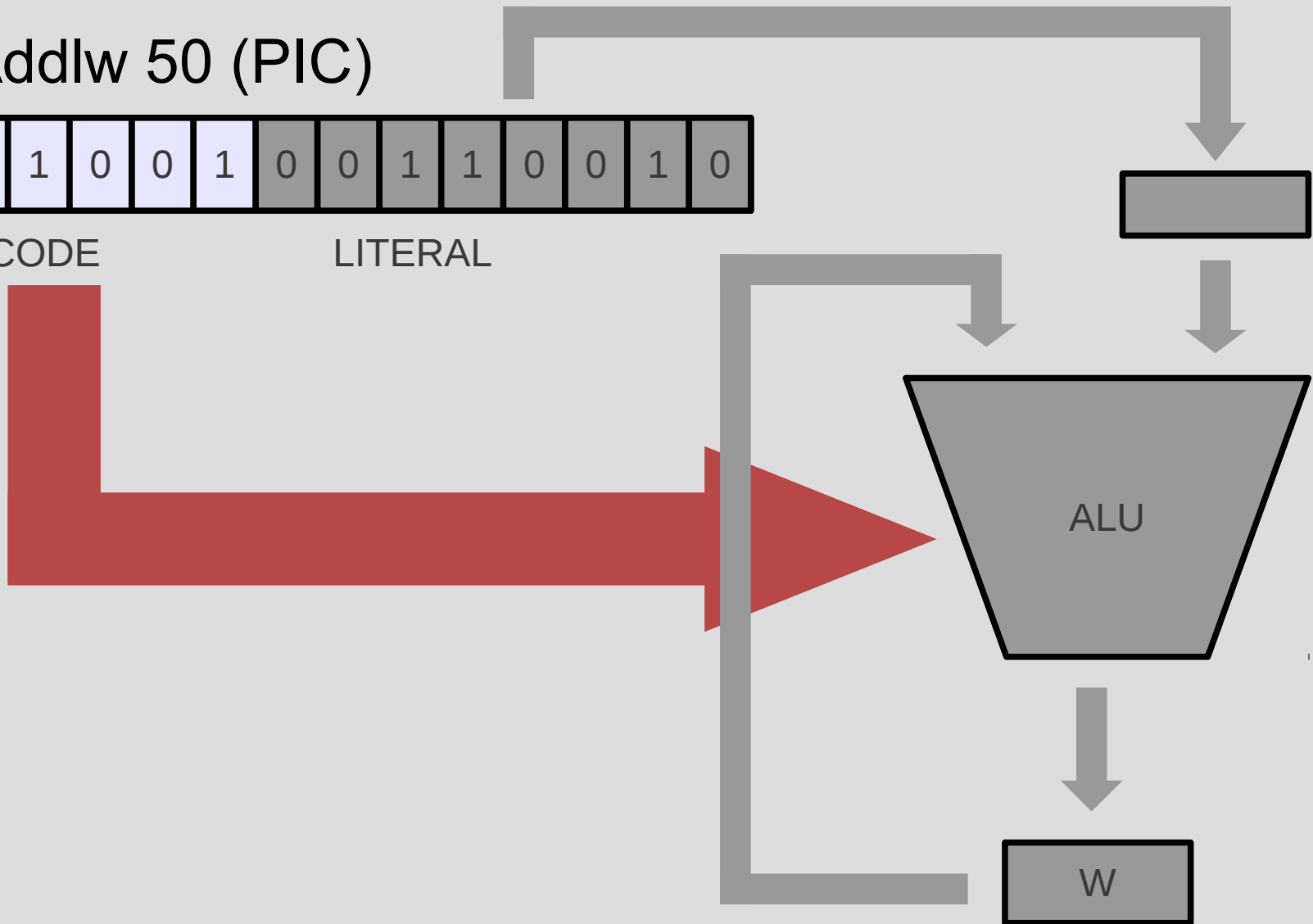
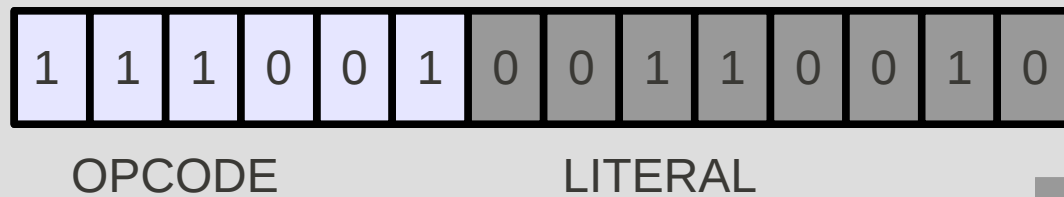
Processador Organização

- Instruções
 - OPACODE
 - OPERANDOS (0 ou mais)
- Unidade de Controle
 - Busca
 - Decodifica (Determina o tipo) instruções
 - Instruções de Desvio
 - Instruções de Movimentação de Dados
 - Instruções Lógicas e Aritméticas
- Unidade Lógica e Aritmética
 - Executa instruções Lógicas e Aritméticas (Obvio!)

Processador

Organização - Instruções

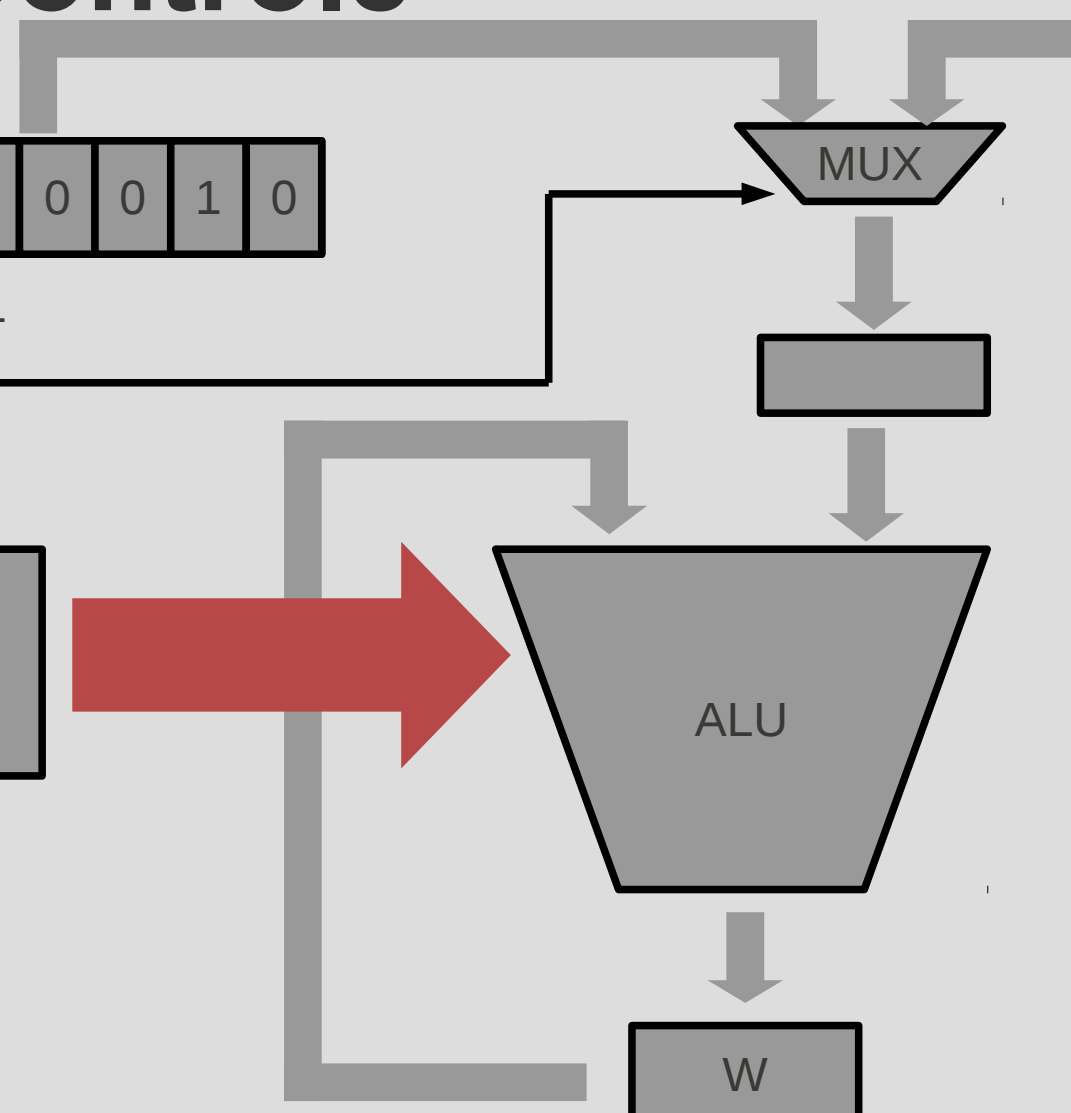
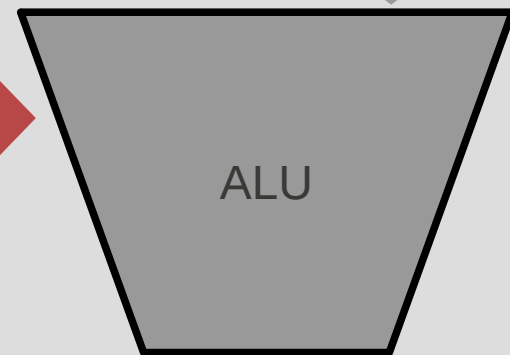
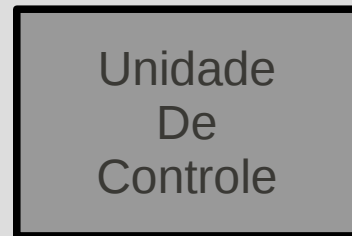
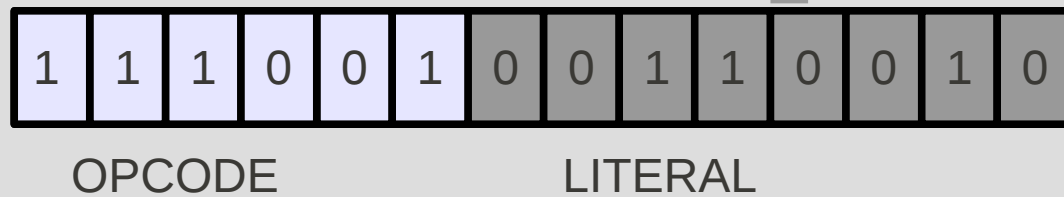
- Addlw 50 (PIC)



Processador

Organização – Unidade de Controle

- Addlw 50 (PIC)

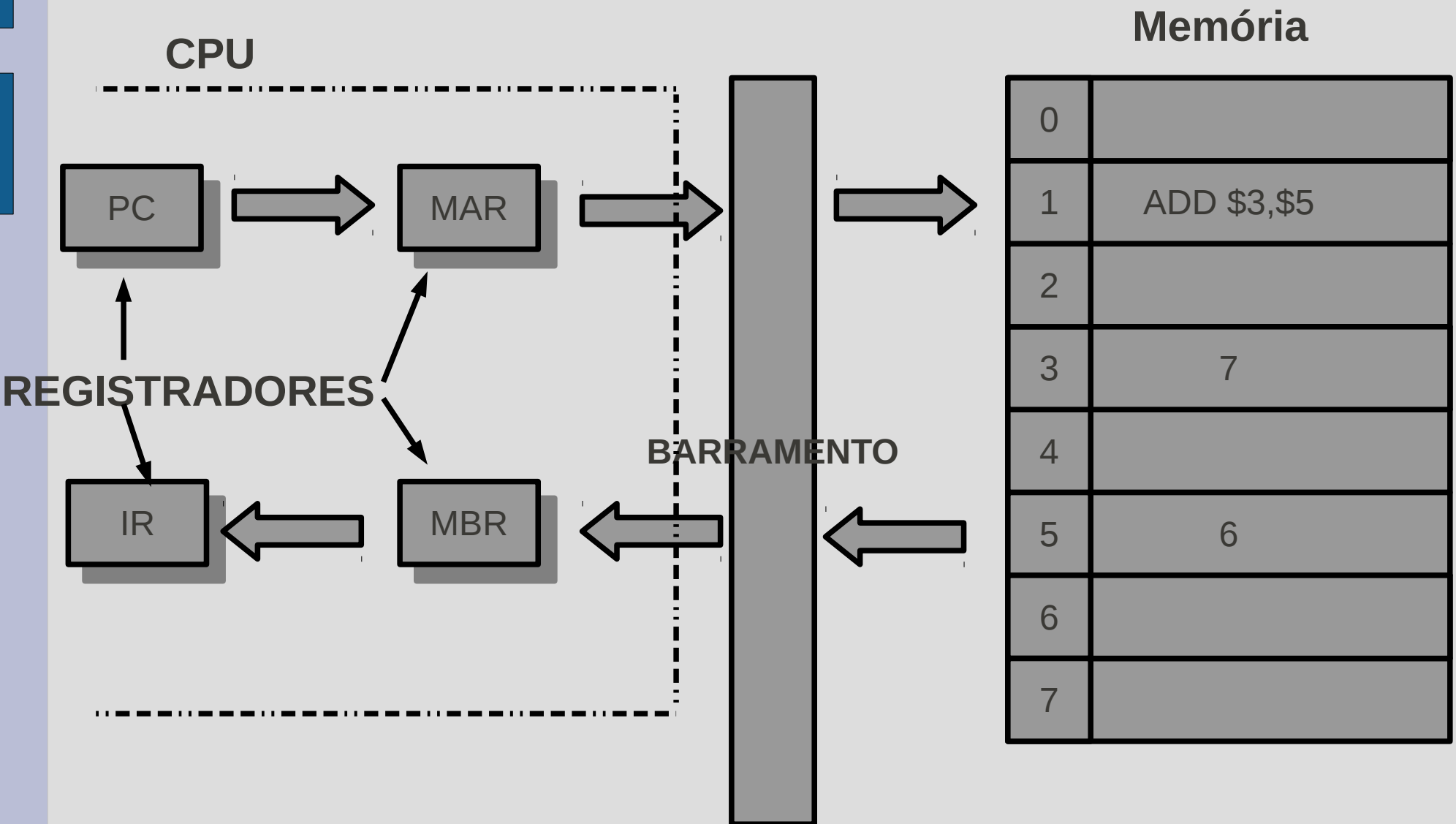


Processador

Organização (Cont...)

- Registradores
 - Memória interna a CPU de Alta velocidade
 - Usados para armazenamento temporário e controle
 - Registradores comuns (Não acessíveis geralmente)
 - PC (Program Counter): End. Próxima instrução
 - IR (Instruct Register): Instrução sendo executada
 - MBR(Memory Buffer Register): dado a ser lido/escrito na mem.
 - MAR (Mememory Address Register): endereço de/para onde o dado será lido/escrito
 - STATUS: Resultado das operações (igual, overflow), modo de execução (supervisor, normal), etc
 - Registradores de uso geral (Acessíveis)

Modelo do Computador Von Neumann

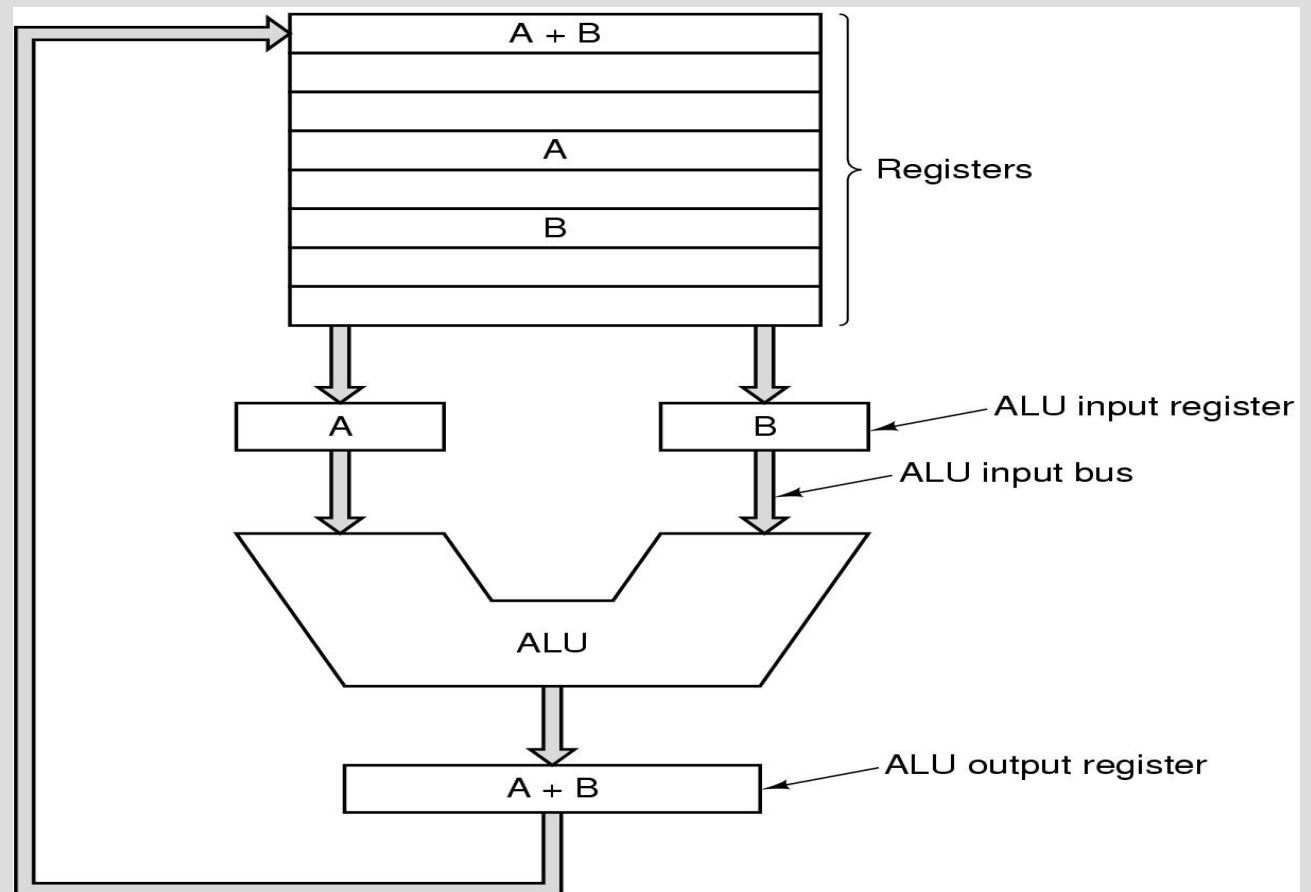


Processador

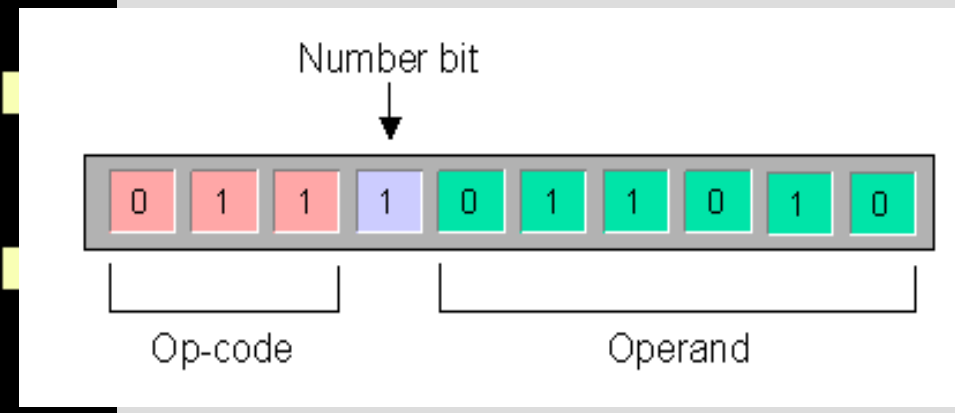
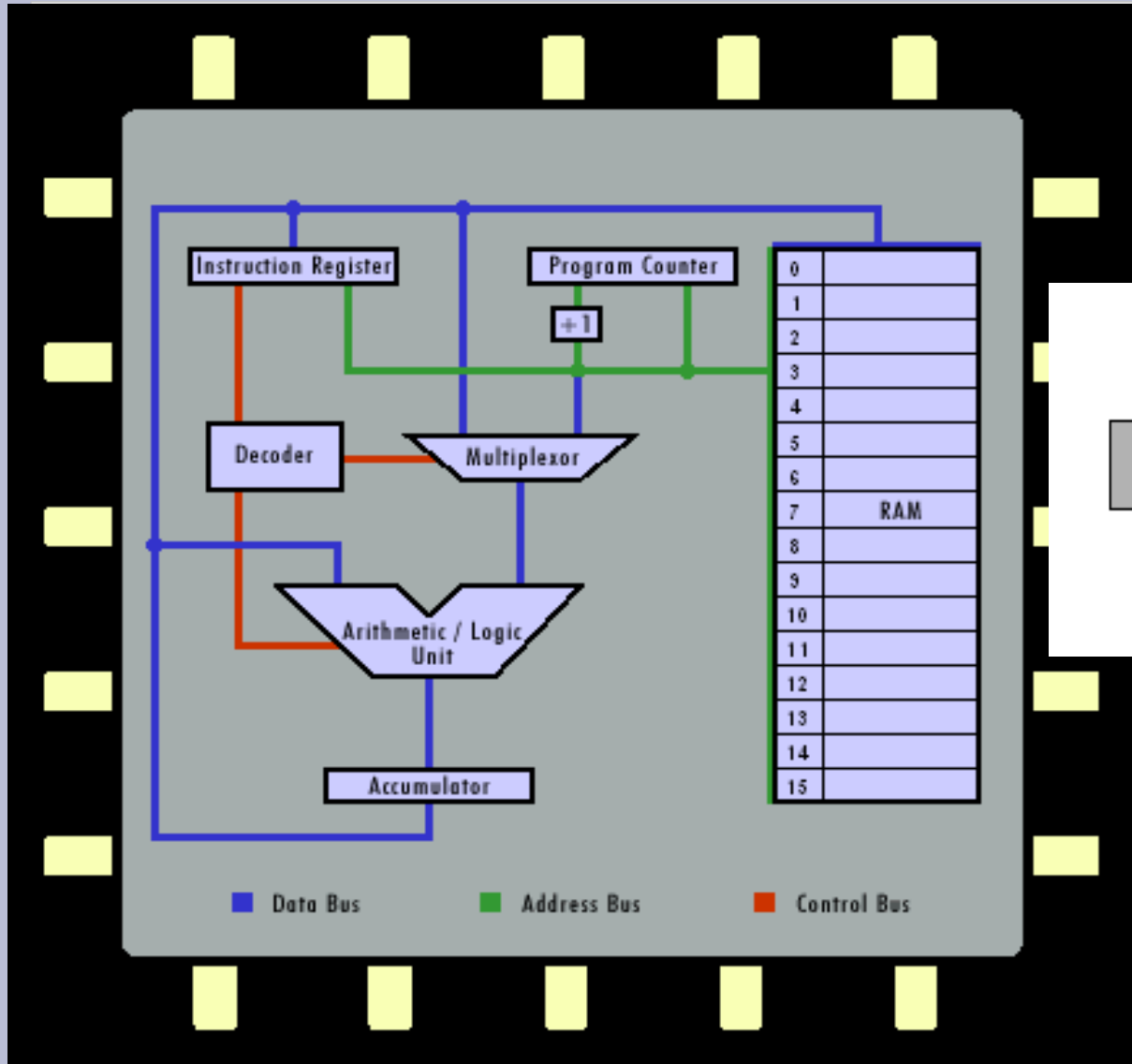
Caminho de dados

- Registradores, ALU e Barramentos

Ciclo do Caminho de Dados



Processor Demo



Processador

Execução de Instrução

- 1) Mover a próxima instrução entre reg e mem
 - 2) Alterar o Registrador PC
 - 3) Determinar o tipo da instrução trazida
 - 4) Se a instrução usar operandos verifica a localização
 - 5) Buscar operandos para registradores internos, se for o caso
 - 6) Executar a instrução
 - 7) Voltar a passo inicial
- Ciclo Buscar – Decodificar - Executar

Processador RISC x CISC

- CISC: Conjunto Complexo de Instruções
 - A evolução da tecnológica dos processadores levou ao uso de microcódigo para fornecer mais instruções ao programador
 - Instruções de máquina parecidas com as instruções de alto nível
 - Consequência: Baixa complexidade do compilador

Processador RISC x CISC

- RISC: Conjunto Reduzido de Instruções
- Registradores (até 32), ALU e Barramentos
- Instruções
 - Registrador-Memória (STORE)
 - Memória-Registrador (LOAD)
 - Registrador-Registrador (passam pela ALU)
- Compilador complexo
- Instruções executadas diretamente pelo nível da lógica digital

Processador

RISC x CISC - Resumo

- **CISC**
 - Instruções complexas em múltiplos ciclos
 - Qualquer instrução referencia memória
 - Menor possibilidade de pipeline
 - Microprograma interpreta instruções
 - Instruções de tamanho variado
 - Muitas instruções e modos de endereçamento
 - Complexidade no microprograma
 - Conjunto pequeno de registradores
- **RISC**
 - Instruções consomem somente um ciclo
 - Somente Load/Store referenciam memória
 - Uso de pipeline
 - Hardware (nível da logica digital) executa a instrução
 - Instruções de tamanho fixo
 - Poucas instruções e modo de endereçamento
 - Complexidade esta no compilador
 - Muitos conjuntos de registradores

Processador

Princípios de Projeto (RISC)

- Todas instruções devem ser executadas pelo HW (Abaixo o microcódigo!)
- Maximize a taxa de execução de instruções
- Instruções devem ser fáceis de decodificar
 - Comprimento fixo
 - Número de campos pequeno
- Acesso à memória:
 - Somente LOAD e STORE
- Registradores: Quanto mais melhor
 - Pelo menos 32

Processador Paralelismo

- Velocidade de Execução = Instruções/s
- Aumento do Clock (Limites tecnológicos)
- Aumento do número de unidade funcionais
- De Instrução
 - Pipeline, Arquiteturas Super Escalares
- De Processador
 - Maquinas Matriciais, Vetoriais, sistemas multiprocessados e MultiComputadores

Processador

Paralelismo de instrução

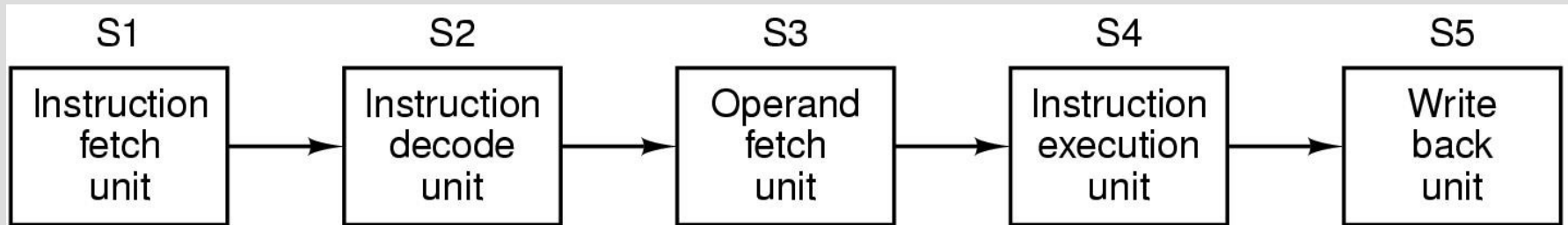
Pipelining

- Divide a execução da instrução em etapas
- Permite uso otimizado das estruturas internas do processador
- Etapas
 - Busca da Instrução
 - Decodifica a Instrução
 - Busca de Operandos
 - Execução da Instrução
 - Armazenamento do resultado no registrador

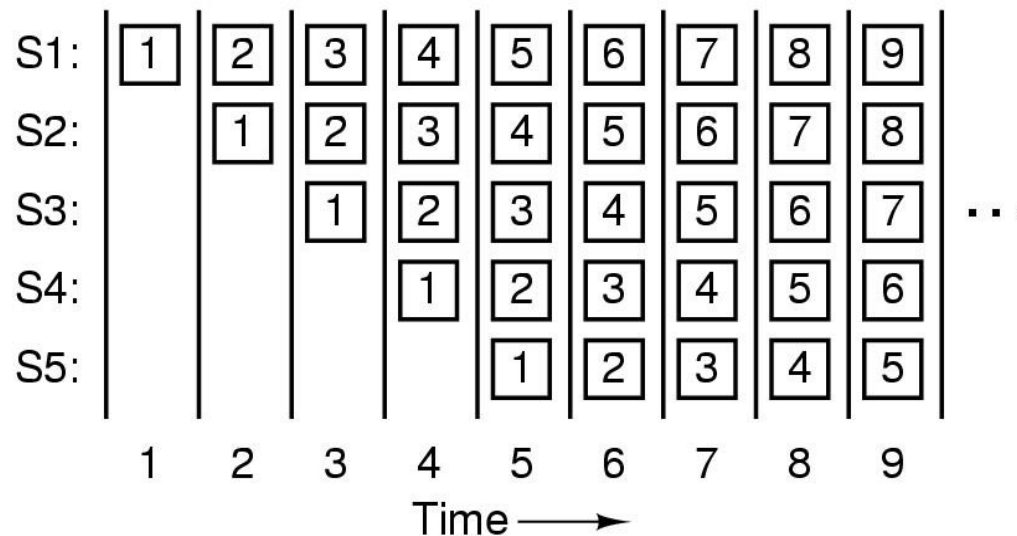
Processador

Paralelismo de instrução

Pipelining



(a)

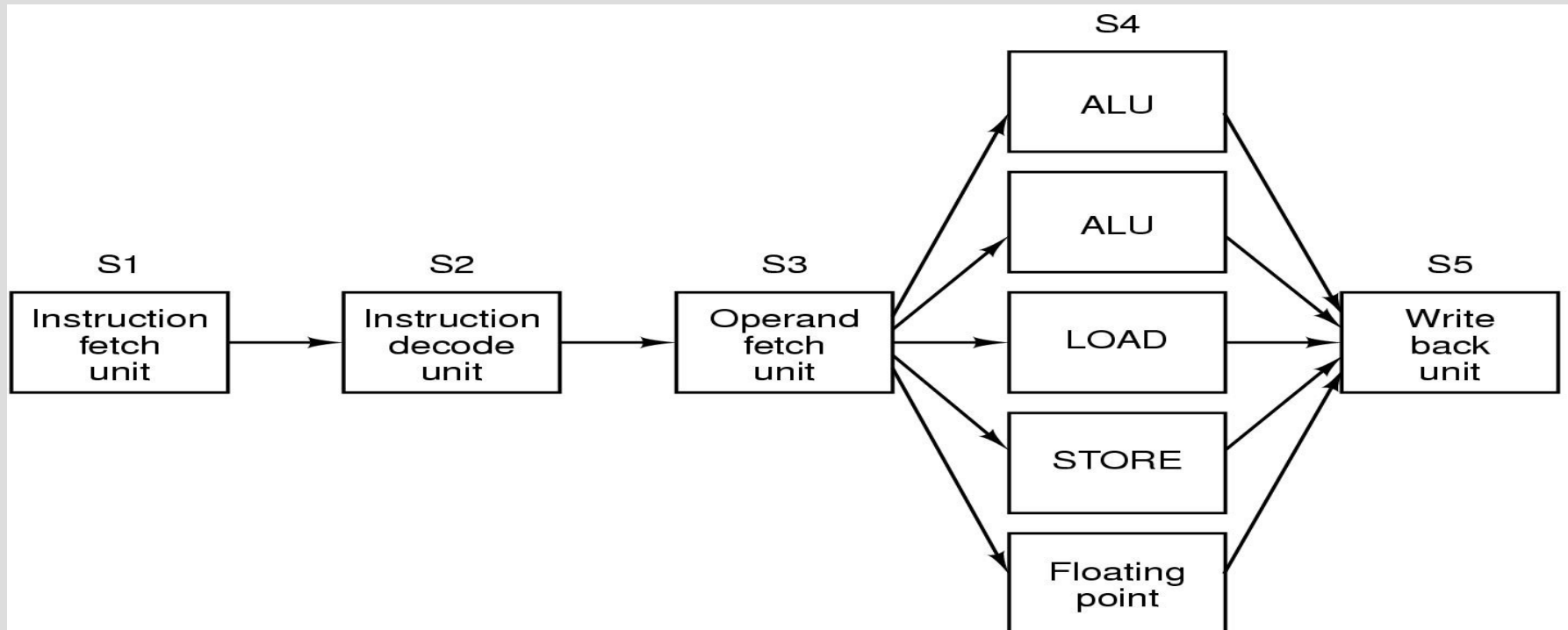


(b)

Processador

Paralelismo de Instrução

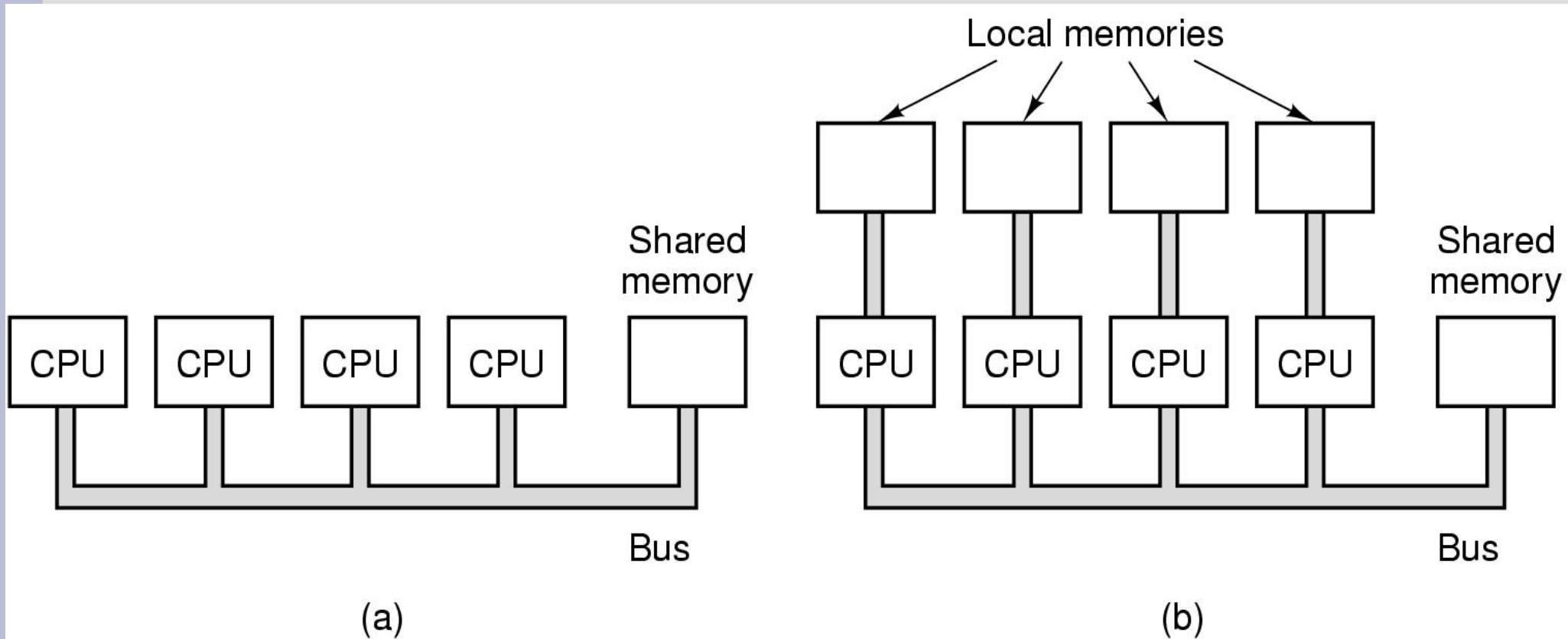
- Arquiteturas Superescalares
 - Executa várias instruções em um único ciclo
 - Várias ALUs no estágio 4
 - Neste estágio as instruções levam vários ciclos



Processador

Paralelismo de processador

- Sistemas Multiprocessados
 - UMA (SMP)
 - NUMA



Processador

Paralelismo de Processador

- Multicomputadores
- Sistemas Distribuidos
- Clusters computacionais
- Supercomputadores
- Matriciais
 - Grande número de processadores idênticos executando a mesma sequência de instruções em diferentes conjuntos de dados
- Vetoriais
 - Processadores com auto grau de paralelismo de entrada de dados
- Uso para processamento de matrizes
- Capítulo 8 (Tem mais sobre maq. paralelas)

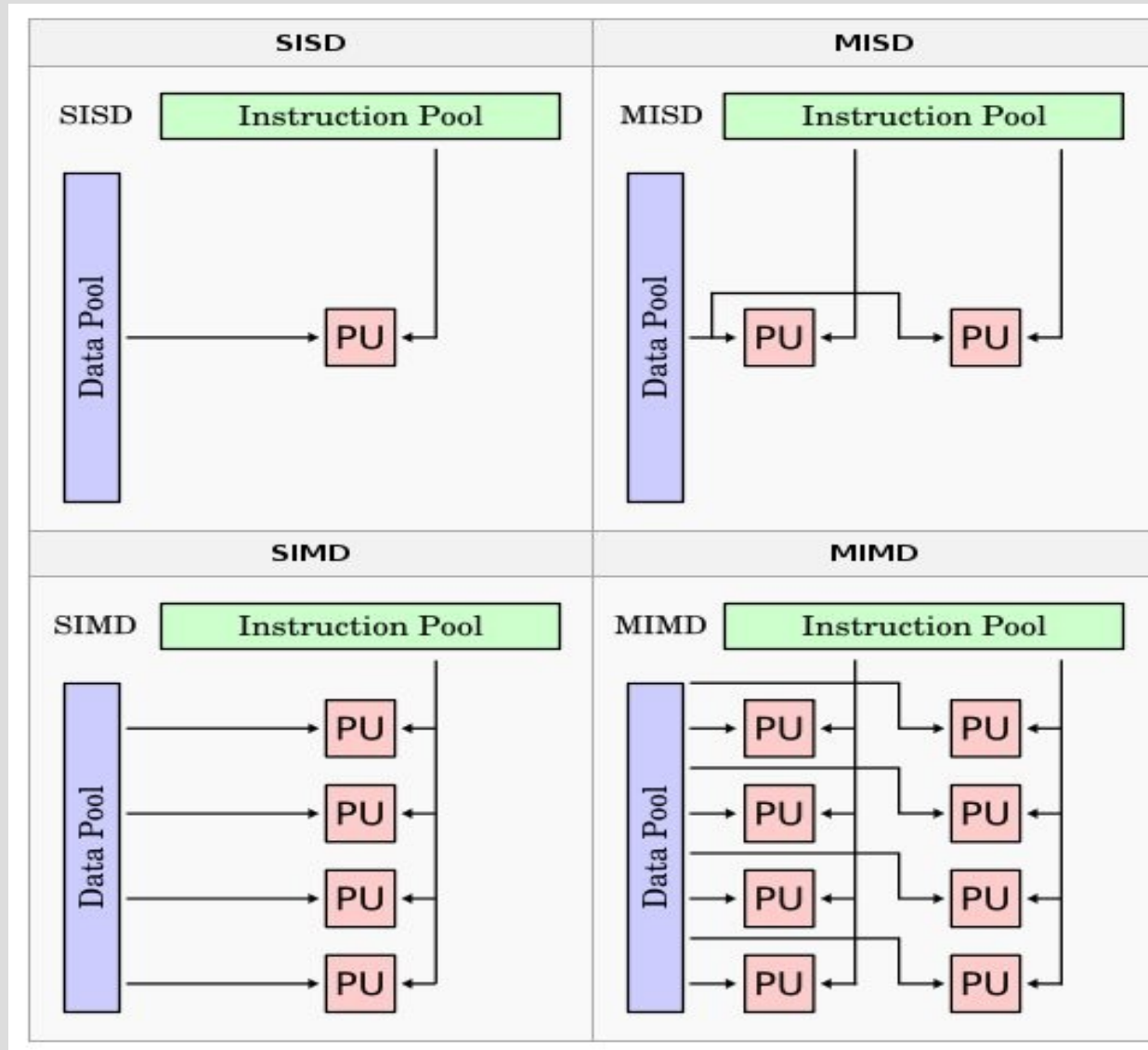
Processador

Taxonomia de Flynn

- Classificação das Arquiteturas
 - Proposta por Michael Flynn em 1966
- SISD - Single Instruction single Data
 - Uma única instrução agindo sobre um dado
- SIMD - Single Instruction Multiple Data
 - Uma única instrução agindo sobre um conjunto de dados
- MISD Multiple Instruction Single Data
 - Múltiplas instruções agindo sobre um dado
 - Nunca se ouviu falar de exemplos
- MIMD Multiple Instruction Multiple Data
 - Várias instruções agindo sobre vários dados

Processador

Taxonomia de Flynn



Processador

Taxonomia de Flynn

