Design de Computadores

Aula 3

Insper

Topologia dos Processadores

Tópicos:

- Abstração Unidade Controle + Fluxo de Dados.
- Funcionamento de um processador simplificado.
- Tipos de instruções.
- Arquitetura e operandos.
- Arquitetura e quantidade de barramentos.
- Modos de endereçamento.

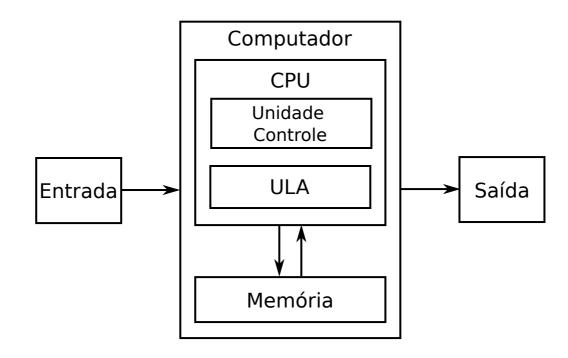
Objetivos de aprendizado:

- Descrever os tipos mais comuns de instruções.
- Descrever os modos de endereçamento.
- Descrever os tipos mais comuns de arquiteturas.

Insper

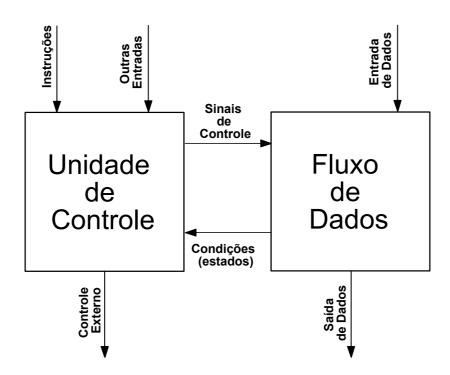
Relembrando

- O computador é composto por:
 - Processador (ou CPU);
 - Memória;
 - Entrada/Saída.

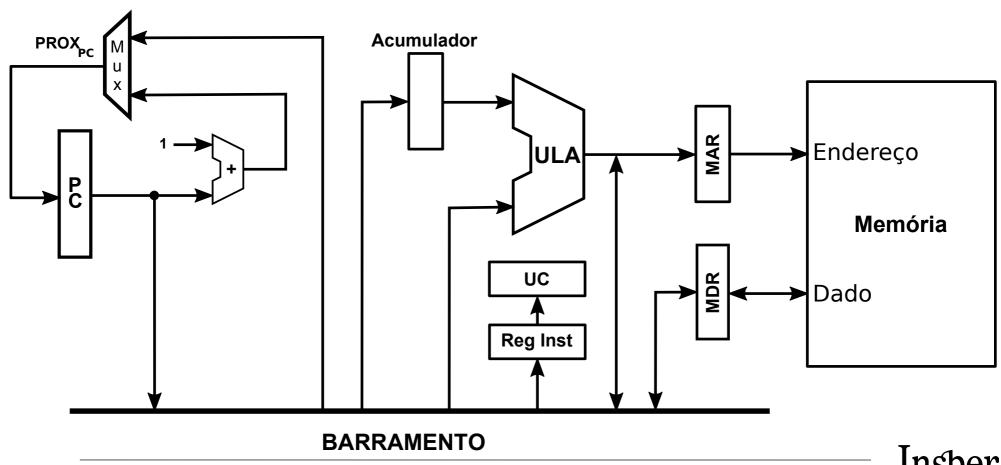




- O processador pode ser visto como:
 - Unidade de controle;
 - Fluxo de dados.



- O fluxo de dados:
 - É o responsável por executar as instruções.
- O exemplo possui um único barramento.

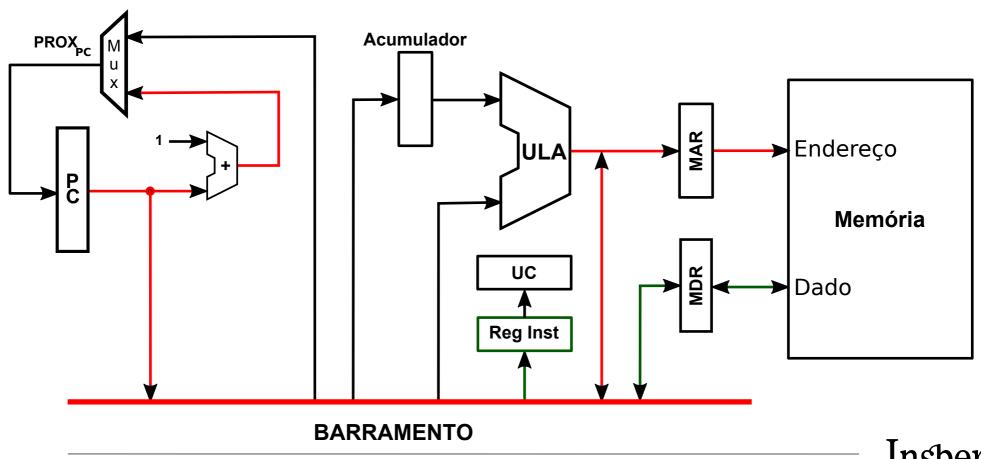


Funcionamento do Processador

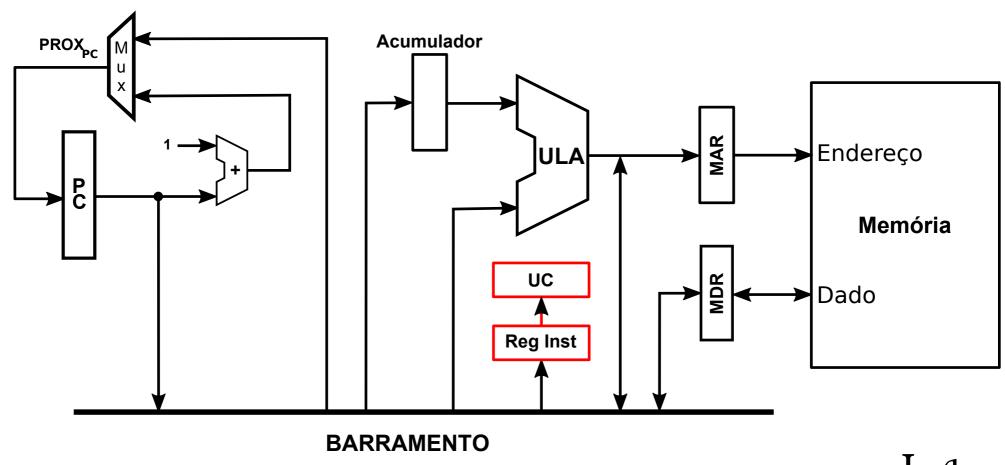
- O processador deve executar uma sequência do tipo:
 - Buscar a próxima instrução na memória (Fetch):
 - Incrementar o PC;
 - Decodificar a instrução (Decode);
 - Ler os operandos (Read);
 - Executar a instrução (Execute);
 - Escrever o resultado (Write-Back).



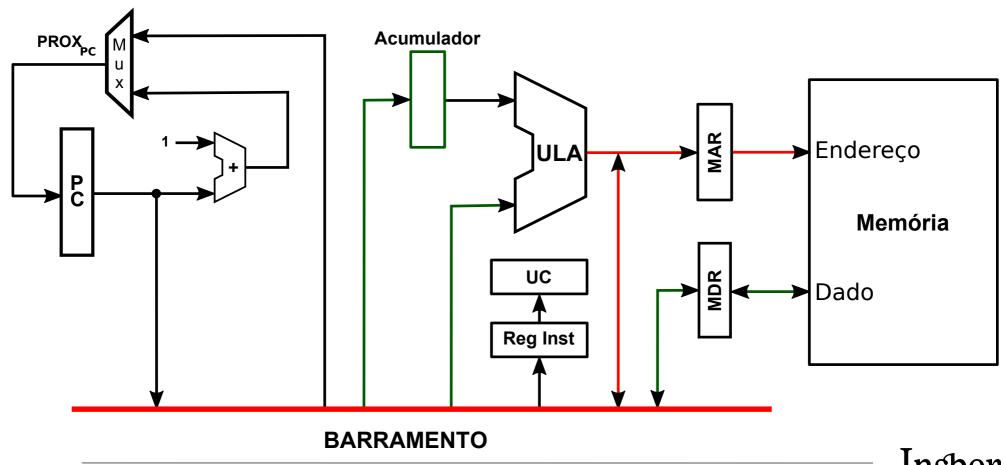
• Busca:



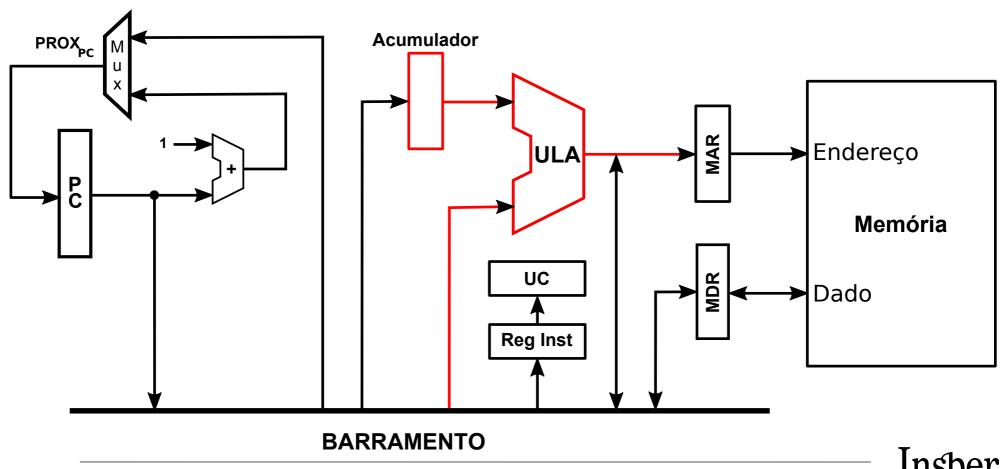
Decodificação:



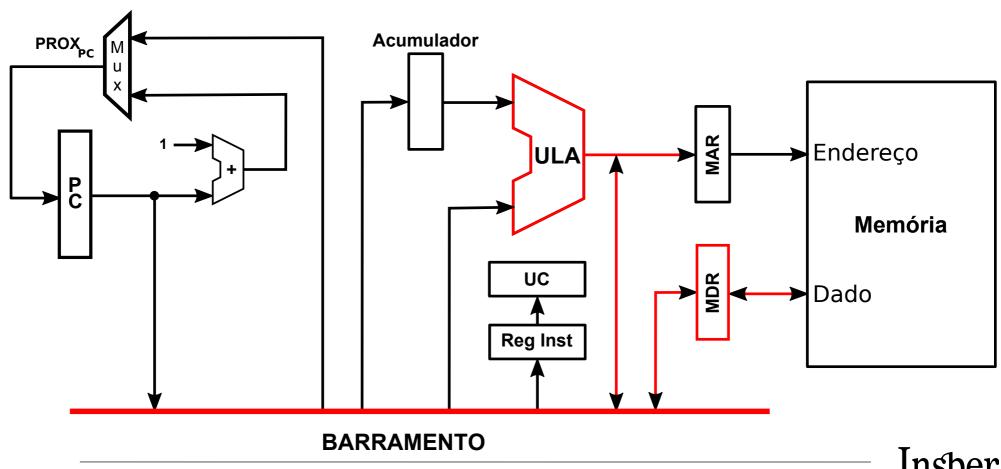
Operandos:



• Execução:



• Escrever:



Topologia

- A topologia do processador está vinculada com:
 - O tipo das instruções:
 - A sua função, quantidade de operandos e a localização deles.
 - O formato ou o comprimento da instrução:
 - Fixo: simplifica a implementação do processador ao custo de uma baixa densidade de código. Por exemplo, a instrução de incremento utilizará uma palavra inteira.
 - Variável: a implementação é mais complexa. A densidade de código pode ser ajustada para cada instrução – incremento pode ser feito com uma instrução de 8 bits.

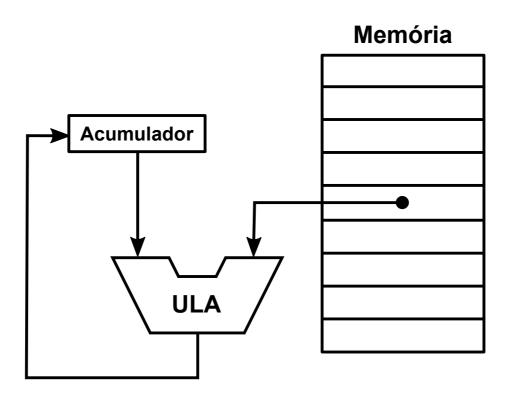


- Tipos de instruções:
 - Entrada e saída;
 - Acesso à memória (movimentação de dados);
 - Lógicas e aritméticas:
 - Manipulação de bits: deslocamento e rotação;
 - Desvios:
 - Absoluto;
 - Relativo ao PC;
 - desvio condicional (decisões).
 - Chamada de sub rotina e retorno;
 - Interrupção e retorno;
 - Parada (Halt);
 - Não faz nada: não opera (NOP).

- Pela funcionalidade das instruções, pode-se inferir quais serão as unidades de hardware necessárias:
 - Entrada e saída;
 - Memória
 - Lógica e aritmética;
 - Controle;
 - Registradores;
 - Etc ...
- Falta definir como interligá-las.

- As arquiteturas mais comuns são:
 - Baseada em acumulador:
 - O resultado sempre é armazenado no acumulador;
 - Pode haver endereçamento direto da memória:
 - Add [0x1FF] → AC = AC + Mem[0x1FF].

Arquitetura: Acumulador

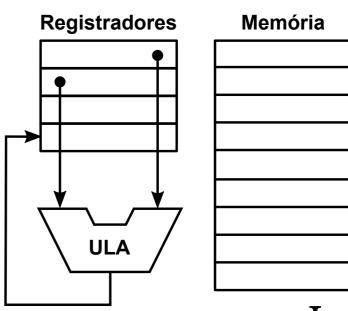




- Baseada em registradores de uso geral:
 - Chamada de registrador-registrador ou load-store.
 - As operações só ocorrem entre registradores:
 - Deve-se carregar os valores da memória (load);
 - E o resultado será armazenado em um registrador;
 - O retorno do resultado para a memória, é feito com o store.
 Arquitetura:

Registrador-Registrador

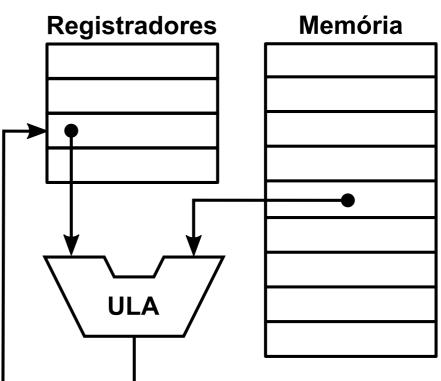
Add R1, R2, R3 \rightarrow R1 = R2 + R3



- · Baseada em registradores e memória:
 - As operações ocorrem entre uma posição de memória e um registrador;
 - O resultado é armazenado no registrador;
 - É a arquitetura do x86.

Arquitetura: Registrador-Memória

Add R1, 0x1FF \rightarrow R1 = R1 + Mem[0x1FF]

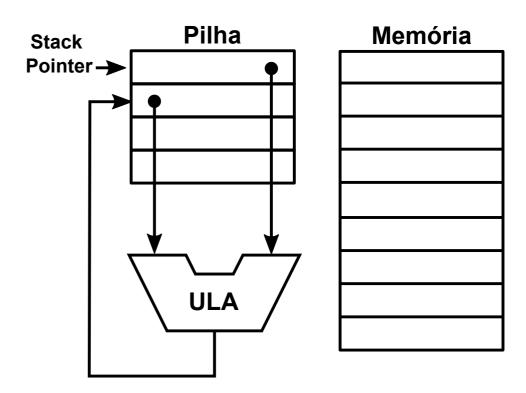


Baseada em pilha:

- Os operandos estão armazenados na pilha:
 - A operação (instrução) os retira da pilha;
 - E acerta o ponteiro da pilha.
- O resultado é escrito no topo da pilha.

Arquitetura: Pilha

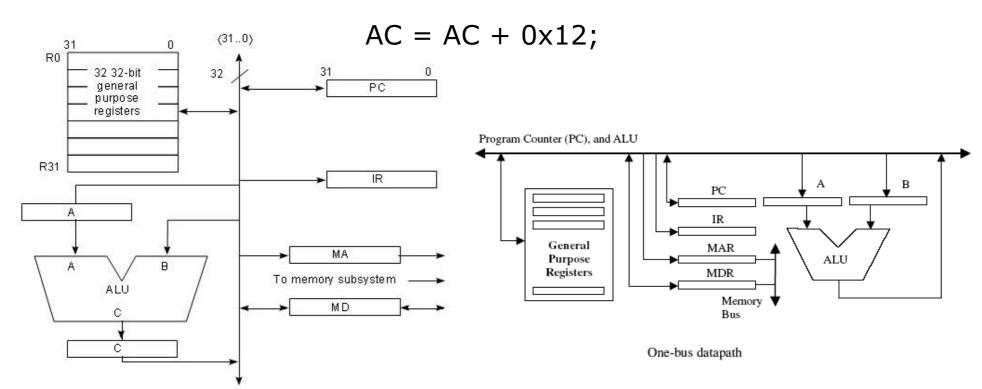
A soma de dois valores é dada por: push a; push b; add; pop c;



Essa arquitetura também é conhecida como "zero address machine".

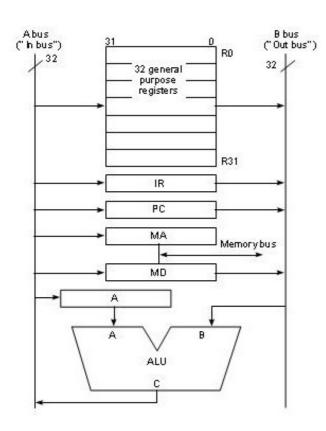


- A implementação (organização) do processador está relacionada com o ISA:
 - Um barramento interno:
 - Transferência de um único dado por vez ("clock");
 - Pode utilizar um operando por instrução:

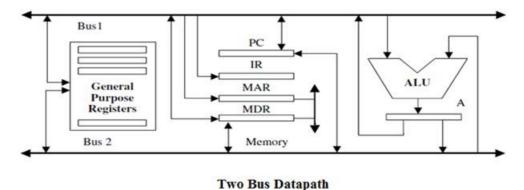


Source: Heuring - Jordan: Computer Systems Architecture and Design

- Com dois barramentos:
 - Transfere dois dados por vez ("clock"):
 - Podem ser de registradores diferentes.
 - Pode utilizar dois operandos por instrução:



$$R1 = R1 + R2;$$

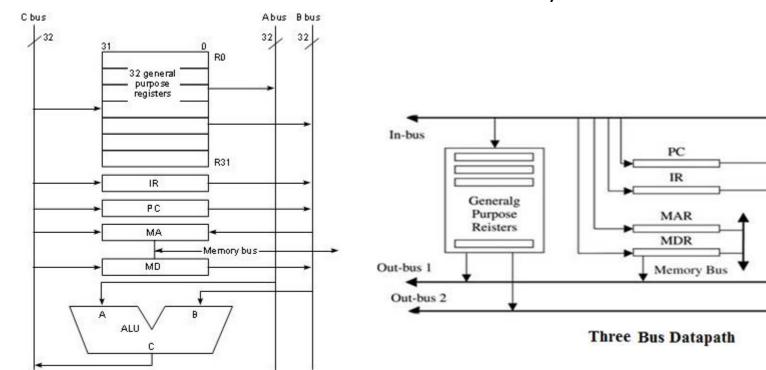


.

Source: Heuring - Jordan: Computer Systems Architecture and Design

- Com três barramentos:
 - Transfere três dados por vez ("clock"):
 - Podem ser de registradores diferentes.
 - Pode utilizar dois operandos e um destino por instrução:

$$R1 = R2 + R3;$$



Source: Heuring - Jordan: Computer Systems Architecture and Design

ALU

- O ISA define as maneiras possíveis para calcular o endereço de memória que será utilizado.
- Alguns tipos (modos) de endereçamento:
 - Imediato (valor):
 - Operando = valor
 - Registrador Ri:
 - Endereço = Ri
 - Absoluto (Local)
 - Endereço = Local
 - Indireto por registrador (Ri):
 - Endereço = [Ri]
 - Indexado X(Ri):
 - Endereço = [Ri] + X
 - Indexado referenciado a uma base (Ri,Rj) :
 - Endereço = [Ri] + [Rj]



• No caso do MIPS, são 5 modos de endereçamento: 1. Immediate addressing

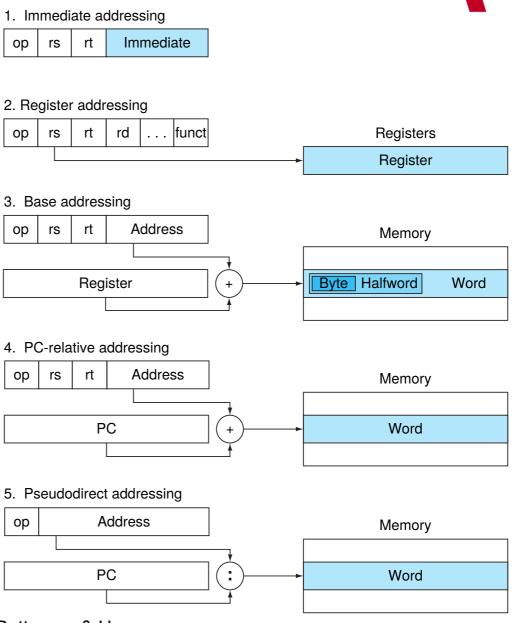


Figura 18 do capítulo 2 - C.O.D. 4 ed - Patterson & Hennessy



Atividade no Blackboard!

Insper

www.insper.edu.br

