

Implementação da UCP

Organização de Computadores

- **Construção do caminho de dados**
- **Controle**
- **Implementação monociclo**

Conceitos Gerais para Implementação do Processador

Organização de Computadores

- **Conceito de caminho de dados e controle**
- **Caminho dos bits de instrução e dados**
- **Utilização de clock, lógica combinacional e seqüencial**
- **Começa uma implementação simples e iterativamente vai melhorando**

Performance

Organização de Computadores

- **Medida de performance baseada em:**
 - número de instruções
 - período do clock
 - ciclos de clock por instrução (CPI)
- **O primeiro é um fator do programa, mas os dois últimos são baseados na implementação do processador**

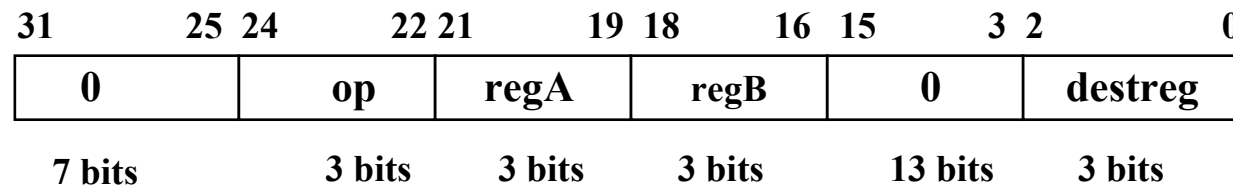
Subconjunto de instruções

- **Para simplificar o estudo do projeto do processador, o foco será dado em um subconjunto de instruções do MIPS:**
 - Memória: lw e sw
 - Aritmética: add e addi
 - Desvio: beq
- **O método de implementação das outras instruções ocorre naturalmente**

Revisão do Formato das Instruções

Organização de Computadores

- **R-FORMAT**
 - add regA regB destreg



op: código de operação

regA: registrador com primeiro operando fonte

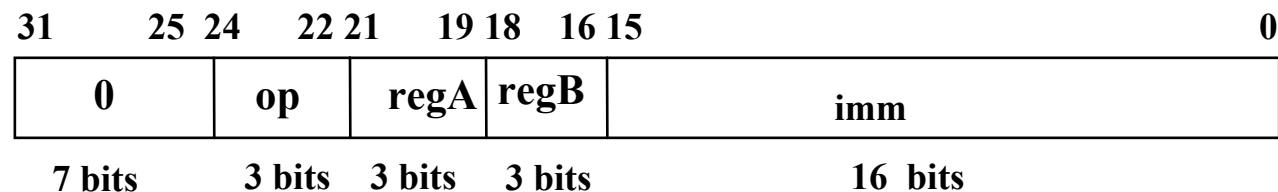
regB: registrador com segundo operando fonte

destreg: registrador que guarda resultado da operação

Revisão do Formato das Instruções

Organização de Computadores

- **I-Format**
 - lw regA regB imm
 - beq regA regB imm
 - addi regA regB imm
- **Desvio utiliza endereço PC relativo ($PC + 1 + \text{imm}$)**



Função básica da CPU

Organização de Computadores

- **Buscar uma instrução na memória**
- **Interpretar qual operação é representada pela instrução**
- **Trazer (se for o caso) os operandos para a CPU**
- **Executar a operação**
- **Armazenar (se for o caso) os dados de saída**
- **Repetir o processo com uma nova instrução**
 - ↳ etapas do Ciclo de Instrução

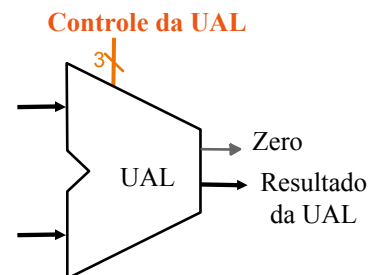
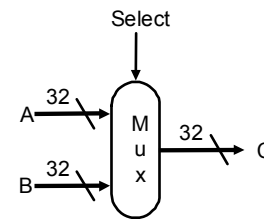
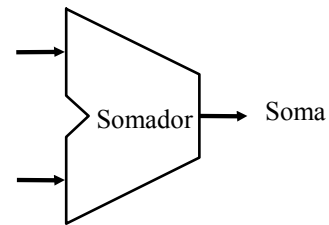
Projeto Lógico

Organização de Computadores

- **Duas definições importantes**
 - Combinacional - saída depende somente das entradas
 - Exemplo: ALU
 - Seqüencial: elementos contem informações de estado
 - Exemplo: Registradores

Elementos Combinacionais

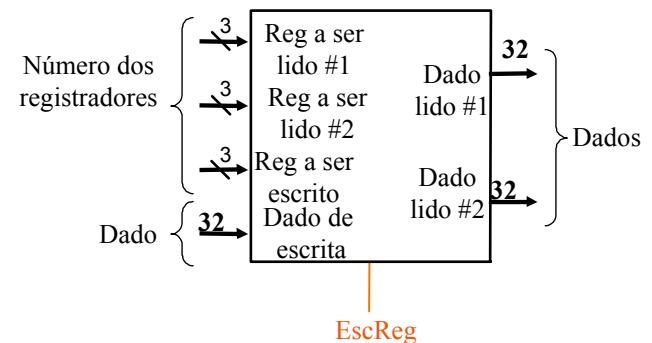
Organização de Computadores



Banco de Registradores

Organização de Computadores

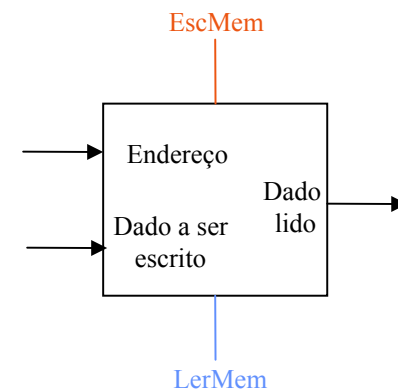
- **Contém 8 registradores**
 - Dois barramentos de 32 bits de saída
 - Dado lido #1 e Dado lido #2
 - Um barramento de 32 bits de entrada
 - Dado a ser escrito
 - Registrador 0 tem o valor 0
 - Registrador selecionado por
 - Reg a ser lido #1
 - Reg a ser lido #2
 - Reg a ser escrito seleciona registro que recebe Dado a ser escrito quando EscReg=1



Memória

Organização de Computadores

- **Um barramento de entrada: Dado a ser escrito**
- **Um barramento de saída: Dado lido**
- **Seleção de endereço**
 - Endereço seleciona a palavra a ser colocada em Dado lido
 - Para escrever no endereço, seta EscMem para 1
 - Para ler do endereço, seta LerMem para 1



Passos do Projeto

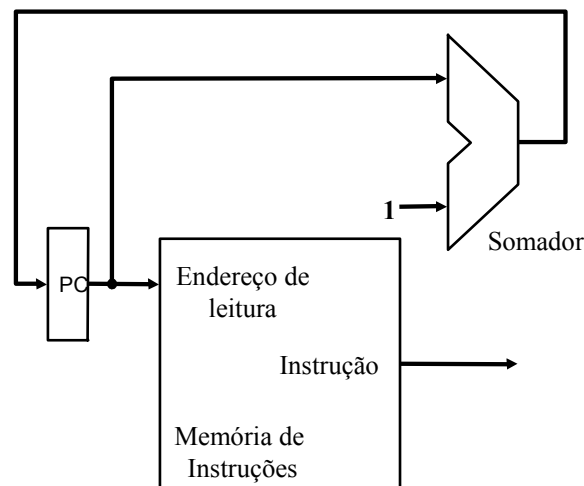
Organização de Computadores

- **De acordo com a arquitetura do conjunto de instruções, define-se uma estrutura organizacional macro (número de unidades funcionais, por exemplo)**
- **Essa estrutura é refinada para definir os componentes do caminho de dados, sua interconexão e pontos de controle**
- **Estrutura de controle é definida**
- **O projeto do caminho de dados e controle é refinado para projeto físico e validação funcional**

Busca da Instrução

Organização de Computadores

- **Busca a instrução na memória, cujo endereço está no contador de programa PC**
- **Incrementa o contador de programa PC de 1**



Instrução de Soma

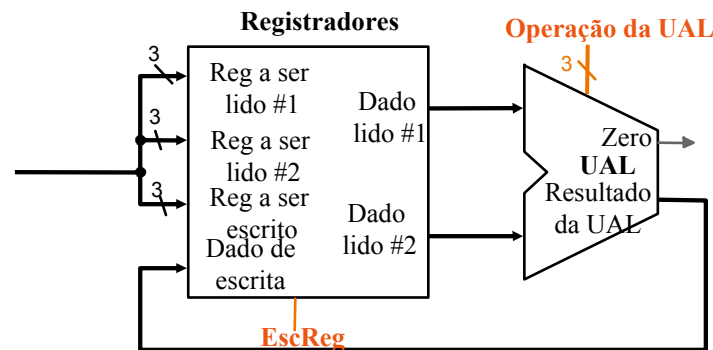
Organização de Computadores

- **add regA regB destreg**
 - Mem[PC] Obtém instrução da memória
 - $R[\text{destreg}] \leftarrow R[\text{regA}] + R[\text{regB}]$ Executa operação de soma
 - $PC \leftarrow PC + 1$ Calcula próximo endereço

Caminho de Dados para Instruções do tipo R

Organização de Computadores

- **$R[\text{destreg}] \leftarrow R[\text{regA}] \text{ op } R[\text{regB}]$**
 - Controle da UAL e de EscReg baseado na instrução decodificada
 - Reg a ser lido #1, Reg a ser lido #2, Reg a ser escrito são regA, regB, destreg



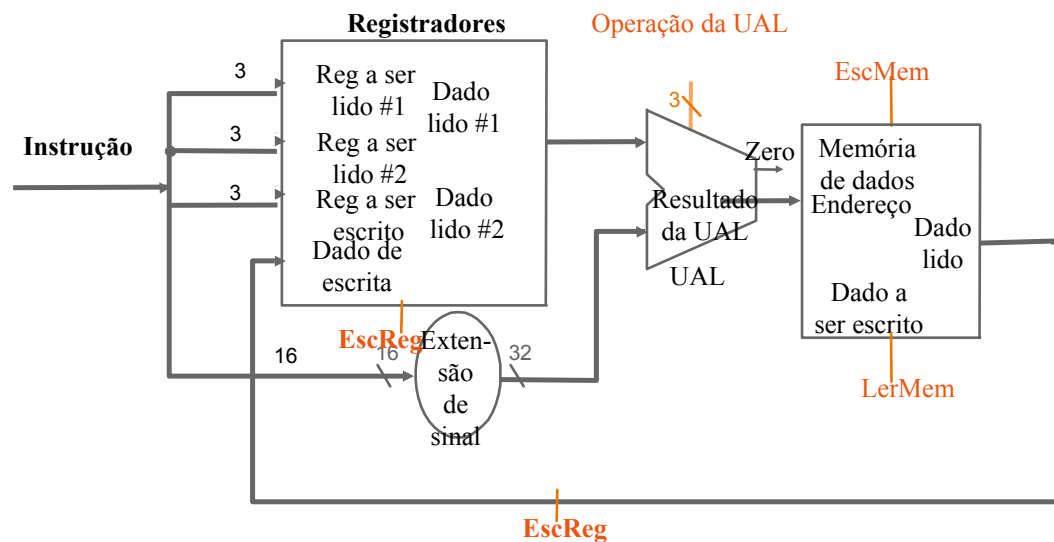
Instrução de Carga

Organização de Computadores

- **lw regA regB imm**

- mem[PC] Busca instrução na memória
- $\text{End} \leftarrow \text{R}[\text{regA}] + \text{SignExt}(\text{imm})$ Calcula o endereço da memória
- $\text{R}[\text{regB}] \leftarrow \text{Mem}[\text{End}]$ Carrega os dados no registrador
- $\text{PC} \leftarrow \text{PC} + 1$ Calcula o próximo endereço

7 bits	3 bits	3 bits	3 bits	16 bits
0	op	regA	regB	imediato



Instrução de Armazenamento

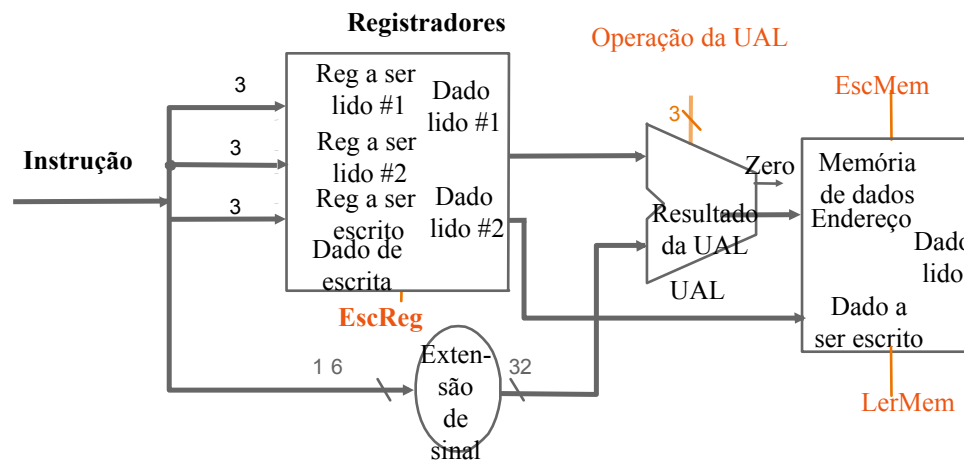
Organização de Computadores

- **sw regA regB imm**
 - mem[PC] Busca instrução na memória
 - $\text{End} \leftarrow \text{R}[\text{regA}] + \text{SignExt}(\text{imm})$ Calcula o endereço da memória
 - $\text{Mem}[\text{End}] \leftarrow \text{R}[\text{regB}]$ Carrega os dados na memória
 - $\text{PC} \leftarrow \text{PC} + 1$ Calcula o próximo endereço

7 bits	3 bits	3 bits	3 bits	16 bits
0	op	regA	regB	imediato

Caminho de Dados para Instrução de Armazenamento

Organização de Computadores



Instrução de Desvio Condicional

Organização de Computadores

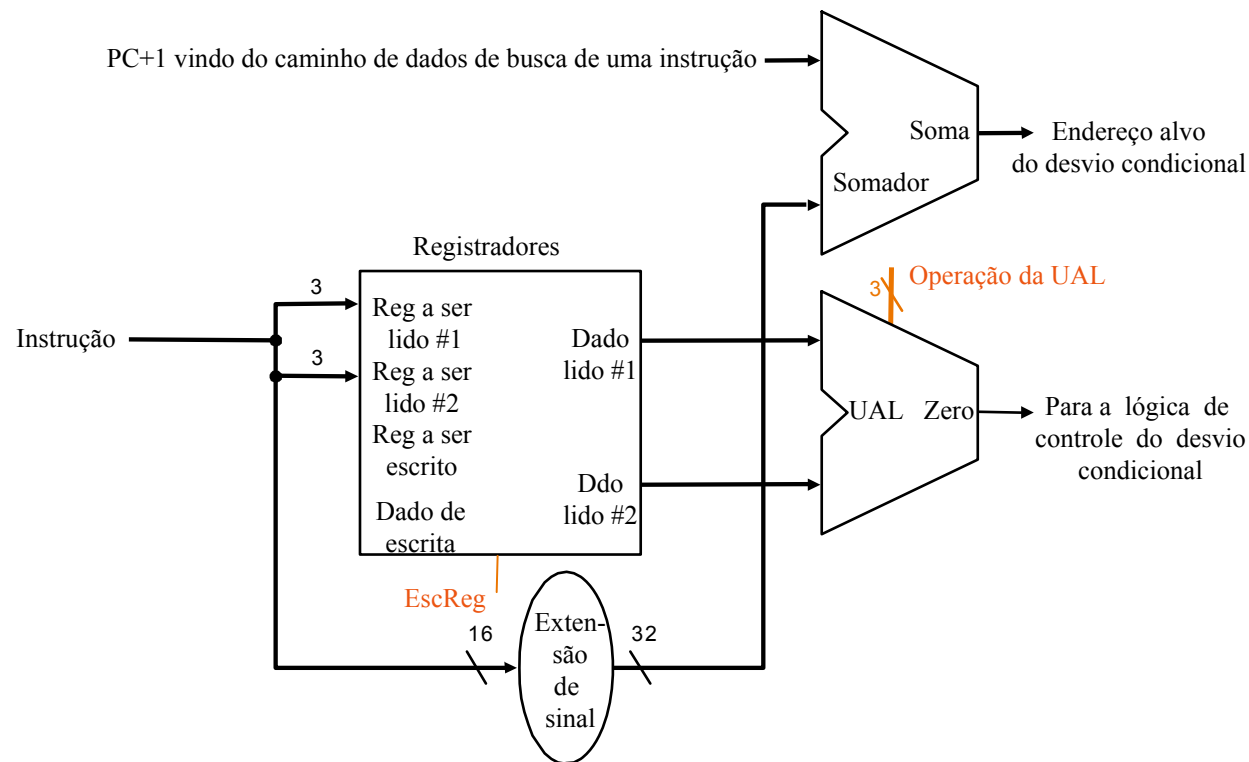
- **beq regA regB imm**

- mem[PC] Busca instrução na memória
- $\text{Cond} \leftarrow R[\text{regA}] - R[\text{regB}]$ Calcula a condição de desvio
- if (Cond eq 0)
 - $\text{PC} \leftarrow \text{PC} + 1 + \text{SignExt}(\text{imm})$ Calcula endereço PC relativo
- else
 - $\text{PC} \leftarrow \text{PC} + 1$ Calcula o próximo endereço

7 bits	3 bits	3 bits	3 bits	16 bits
0	op	regA	regB	imediato

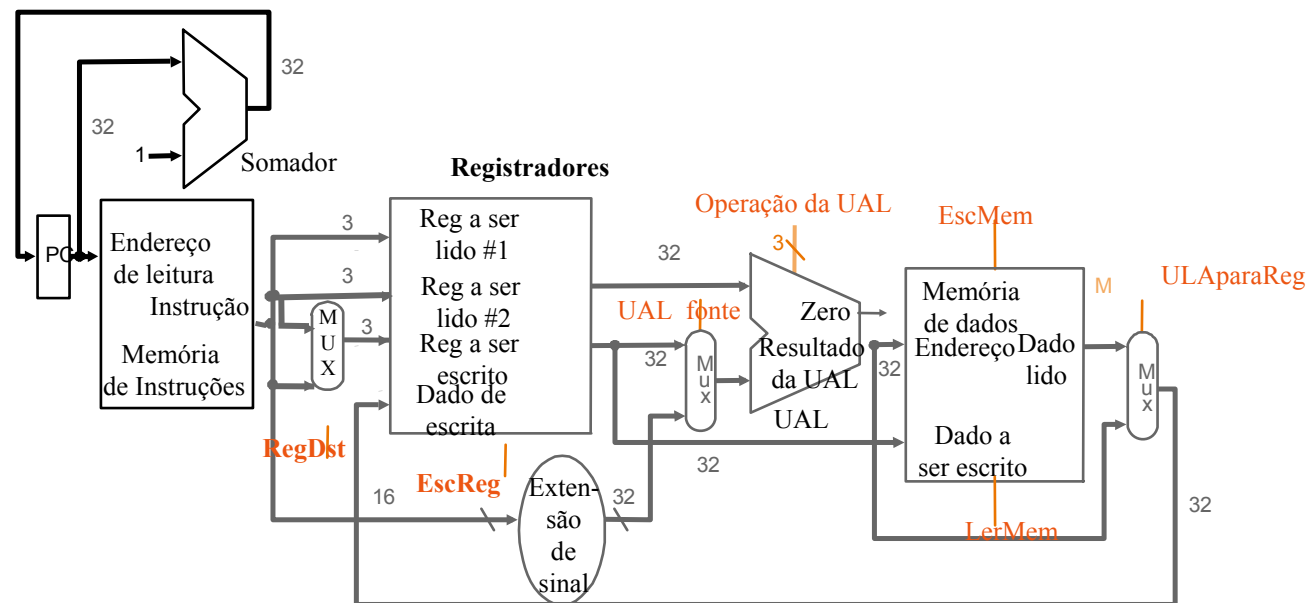
Instrução de Desvio Condicional

Organização de Computadores



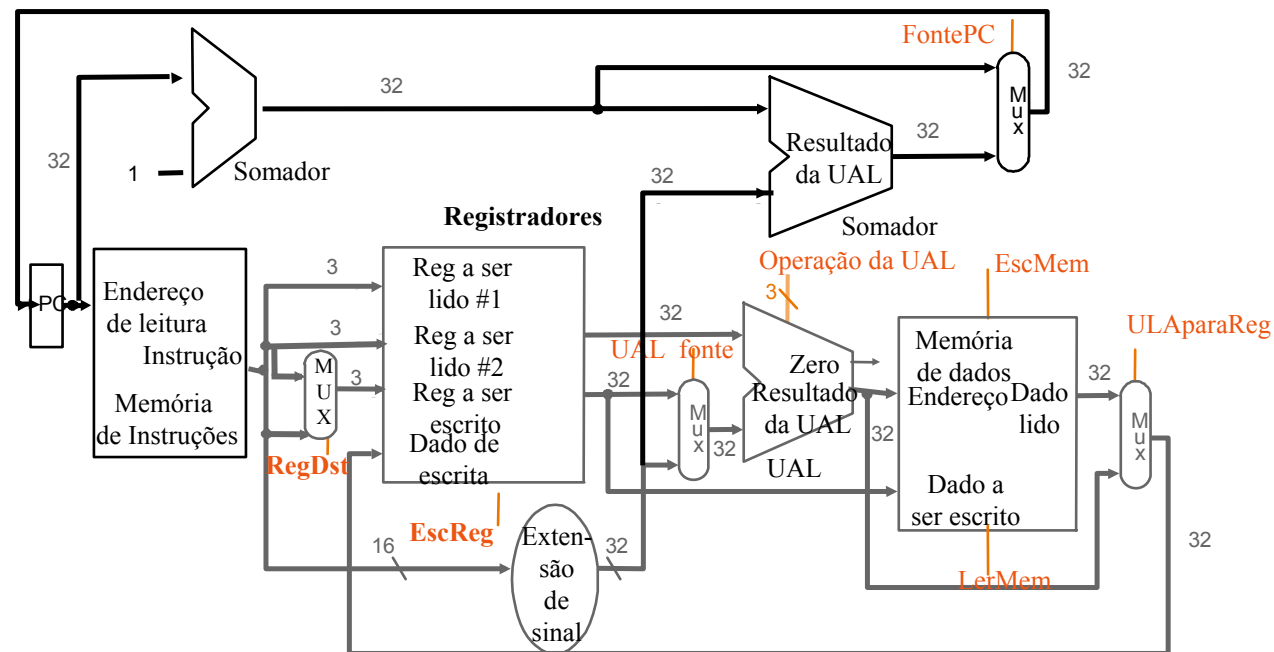
Busca de Instrução e Instruções Aritm. e Lóg. e de Referência à Memória

Organização de Computadores



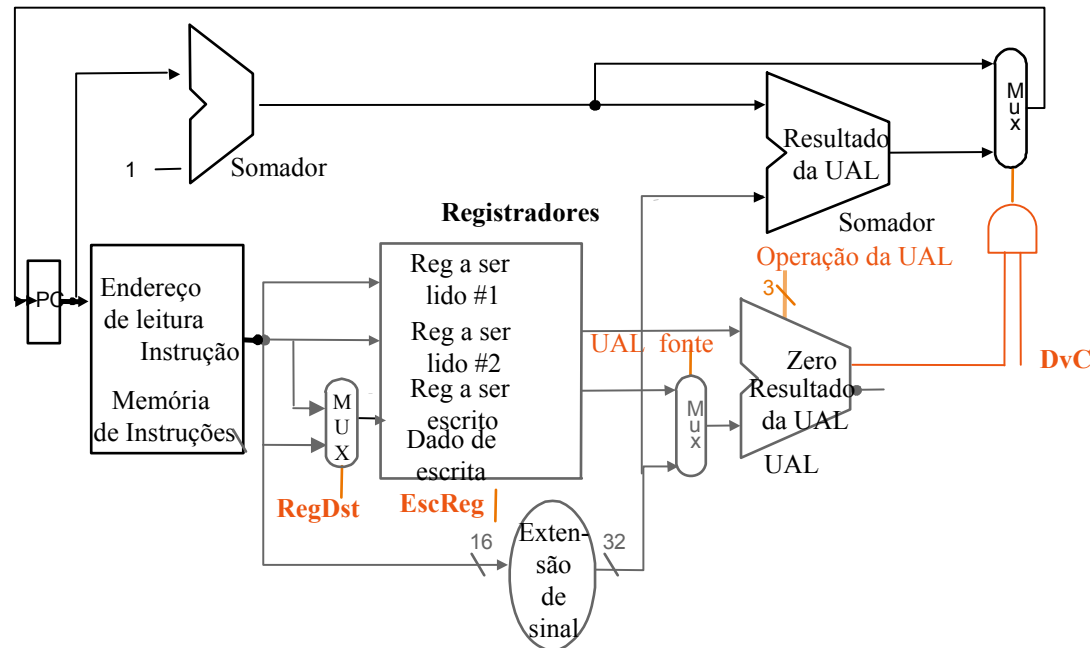
Caminho de Dados para Suportar Subconjunto das Instruções

Organização de Computadores



Caminho de Dados para Carregar PC

- **PC incrementado normalmente**
- **Se instrução beq pode adicionar imm a $PC + 1$**



Juntando Todas as Partes

Organização de Computadores

