



Processador PoliLEGv8

Processador Monociclo (1/3)

Edson S. Gomi

PCS - Departamento de Engenharia de Computação e Sistemas Digitais
Escola Politécnica da Universidade de São Paulo

Outubro, 2020

Agenda

Objetivos da aula

Visão Geral da Implementação do PoliLEG

Metodologia de Clocking

Construção do Fluxo de Dados

1 Objetivos da aula

2 Visão Geral da Implementação do PoliLEG

3 Metodologia de Clocking

4 Construção do Fluxo de Dados

Ao final desta aula você saberá:

Objetivos da
aula

Visão Geral da
Implementação
do PoliLEG

Metodologia de
Clocking

Construção do
Fluxo de Dados

- A visão geral da estrutura do PoliLEG;
- A metodologia de *clock* que utilizaremos no projeto do PoliLEG;
- A estrutura do Fluxo de Dados e os seus componentes.

Visão Geral da Implementação do PoliLEG

Objetivos da
aula

Visão Geral da
Implementação
do PoliLEG

Metodologia de
Clocking

Construção do
Fluxo de Dados

- A implementação do PoliLEG abrangerá um subconjunto das instruções do ARMv8:
 - Load Register Unscaled (LDUR)
 - LDUR X1, [X2,40] : $X1 = \text{Memory}[X2+40]$
 - Store Register Unscaled (STUR)
 - STUR X1, [X2,40] : $\text{Memory}[X2+40] = X1$
 - Instruções lógico-aritméticas ADD, SUB, AND e ORR
 - ADD X1, X2, X3 : $X1 = X2+X3$
 - Compare and Branch Zero (CBZ)
 - CBZ X1, 25 : if ($X1 == 0$) goto PC+100
 - Branch (B)
 - B 2500 : goto PC+10000

Instruções Formato-D

Objetivos da
aula

Visão Geral da
Implementação
do PoliLEG

Metodologia de
Clocking

Construção do
Fluxo de Dados

As instruções LDUR e STUR são do Tipo-D

opcode	address	op2	Rn	Rt
11 bits	9 bits	2 bits	5 bits	5 bits

- opcode + op2: código da instrução
- address : descreve o endereço Memory[Rn ± address]
- Rn : registrador do endereço base
- Rt : registrador da fonte ou do destino

Instruções Formato-R

Objetivos da aula

Visão Geral da Implementação do PoliLEG

Metodologia de Clocking

Construção do Fluxo de Dados

As instruções ADD, SUB, AND e ORR são do Tipo-R

opcode	Rm	shamt	Rn	Rd
11 bits	5 bits	6 bits	5 bits	5 bits

- opcode : código da instrução
- Rm : registrador do 2o. operando
- shamt : (shift amount) é usado em operação de shift, é zero para estas instruções
- Rn : registrador do 1o. operando
- Rd : registrador do resultado da operação

Resumo dos Formatos

Objetivos da aula

Visão Geral da Implementação do PoliLEG

Metodologia de Clocking

Construção do Fluxo de Dados

Name	Fields						Comments
Field size	6 to 11 bits	5 to 10 bits	5 or 4 bits	2 bits	5 bits	5 bits	All LEGv8 instructions are 32 bits long
R-format	R	opcode	Rm	shamt	Rn	Rd	Arithmetic instruction format
I-format	I	opcode	immediate			Rn	Rd
D-format	D	opcode	address	op2	Rn	Rt	Data transfer format
B-format	B	opcode	address				
CB-format	CB	opcode	address			Rt	Conditional Branch format
IW-format	IW	opcode	immediate			Rd	Wide Immediate format

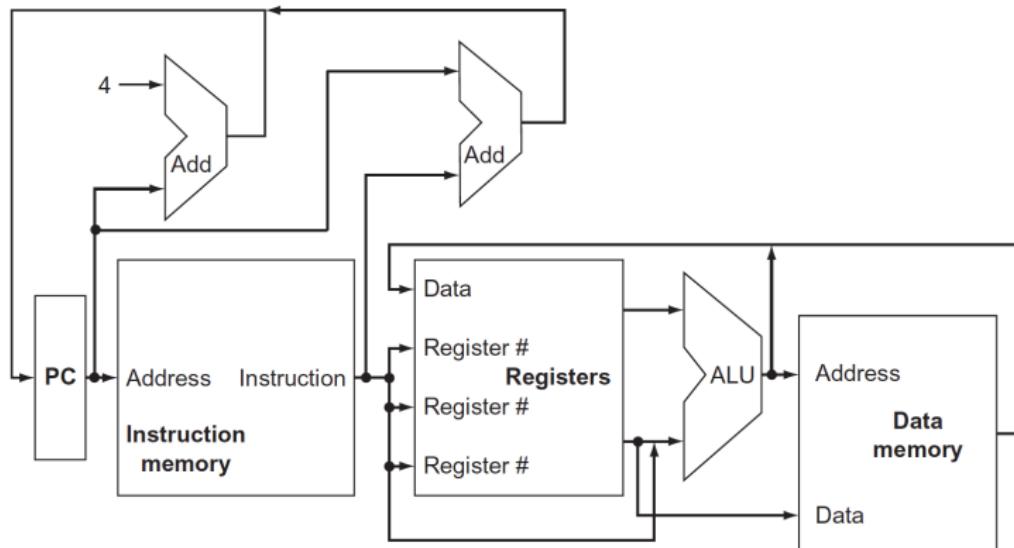
Visão Abstrata da Implementação do PoliLEG

Objetivos da aula

Visão Geral da Implementação do PoliLEG

Metodologia de Clocking

Construção do Fluxo de Dados



Execução das instruções

Objetivos da
aula

Visão Geral da
Implementação
do PoliLEG

Metodologia de
Clocking

Construção do
Fluxo de Dados

Os 2 passos iniciais das instruções são iguais:

- (1) Enviar o Contador de Programa (PC) para a memória que contém o código e transferir a instrução nela residente.
- (2) Ler 1 ou 2 registradores, usando os campos da instrução para selecionar os registradores para a leitura.
- (3) Após esses 2 passos iniciais, as ações a serem executadas dependem da instrução.

Implementação das instruções

Objetivos da
aula

Visão Geral da
Implementação
do PoliLEG

Metodologia de
Clocking

Construção do
Fluxo de Dados

O fluxo de dados funciona da seguinte forma:

- (1) O PC fornece o endereço para a memória e a instrução é transferida;
- (2) Campos da instrução definem os registradores a serem utilizados e os operandos são transferidos;
- (3) Se a instrução é logico-aritmética, o resultado é escrito no registrador;
- (4) Se a instrução é LOAD, o resultado da ULA é usado como endereço para transferir dado da memória para o registrador;
- (5) Se a instrução é STORE, o resultado da ULA é usado como endereço para transferir dado do registrador para a memória;
- (6) Se a instrução é de desvio, o resultado dos somadores determinam o endereço da próxima instrução.

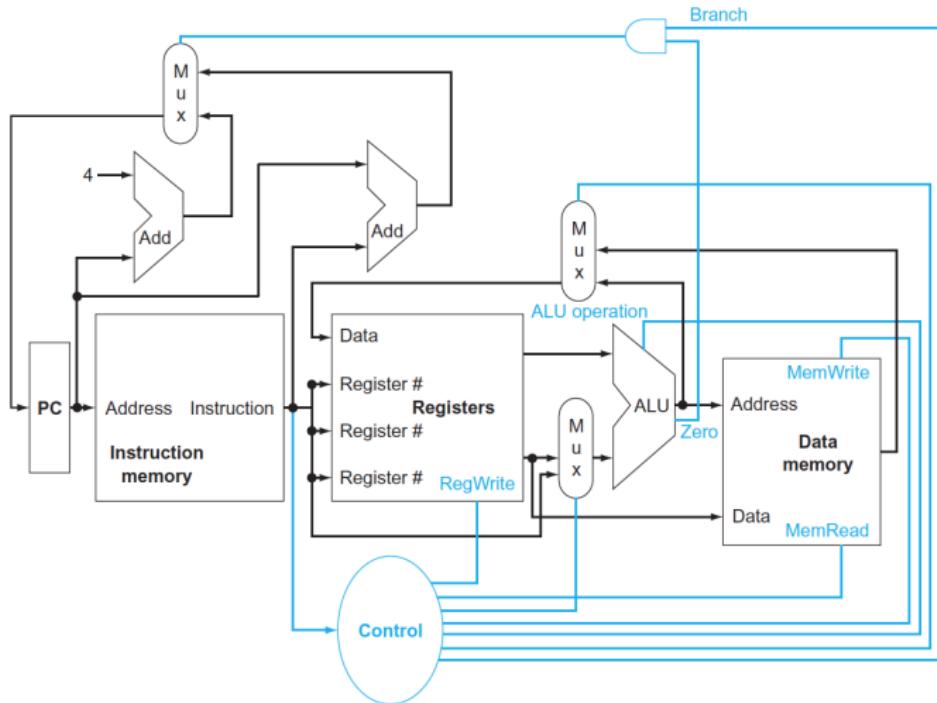
Implementação do PoliLEG incluindo multiplexadores e linhas de controle

Objetivos da aula

Visão Geral da Implementação do PoliLEG

Metodologia de Clocking

Construção do Fluxo de Dados



Acionamento por Borda

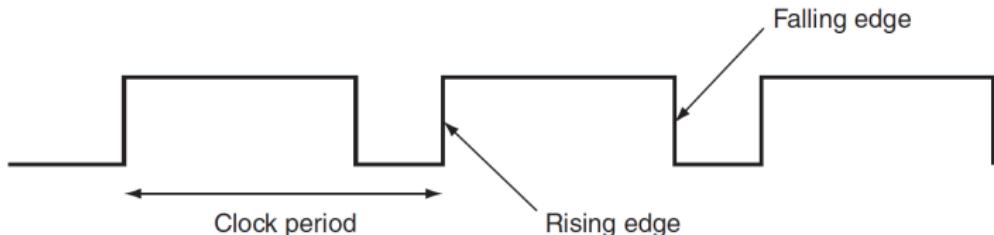
Objetivos da aula

Visão Geral da Implementação do PoliLEG

Metodologia de Clocking

Construção do Fluxo de Dados

O sinal do relógio oscila entre valores lógicos alto e baixo:



As entradas da lógica combinatória são lidas de um elemento de estado e as saídas são escritas em um elemento de estado.



Metodologia de Clocking

Objetivos da
aula

Visão Geral da
Implementação
do PoliLEG

Metodologia de
Clocking

Construção do
Fluxo de Dados

- A metodologia de *clocking* define quando os sinais são lidos e quando são escritos.
- A metodologia de *clocking* é feita para tornar o hardware previsível.
- Para o PoliLEG assumiremos que a atualização de valores nos elementos sequenciais ocorrem nas transições do relógio.
- Os valores de entradas são escritos no ciclo de relógio anterior, ao passo que os valores de saída são usados no ciclo de relógio posterior.

Convenções do Projeto

Objetivos da
aula

Visão Geral da
Implementação
do PoliLEG

Metodologia de
Clocking

Construção do
Fluxo de Dados

- Para simplificar os diagramas, os sinais de controle não são mostrados quando um elemento de estado não é escrito em toda borda de relógio.
- O termo *asserted* indica um sinal que é alto lógico.
- O termo *deasserted* indica um sinal que é baixo lógico.
- Os termos *asserted* e *deasserted* são usados porque às vezes 1 representa o alto lógico e outras vezes o baixo lógico.
- Uma metodologia de *clocking* baseada na transição do relógio, nos permite ler o conteúdo de um registrador na borda do relógio, enviar o valor através de uma lógica combinatória e escrever no registrador no mesmo ciclo de relógio.

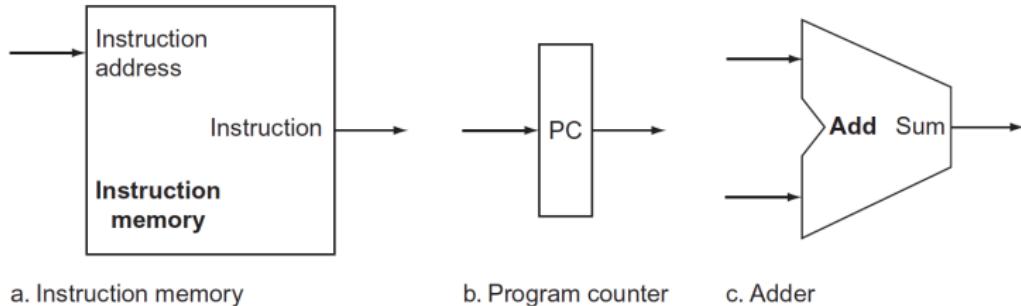
Leitura das Instruções

Objetivos da aula

Visão Geral da Implementação do PoliLEG

Metodologia de Clocking

Construção do Fluxo de Dados



- Unidade de Memória: armazena o programa.
- Program Counter (PC): contém o endereço da instrução corrente.
- Somador: feita com uma ULA configurada para ser somador.

Leitura das instruções

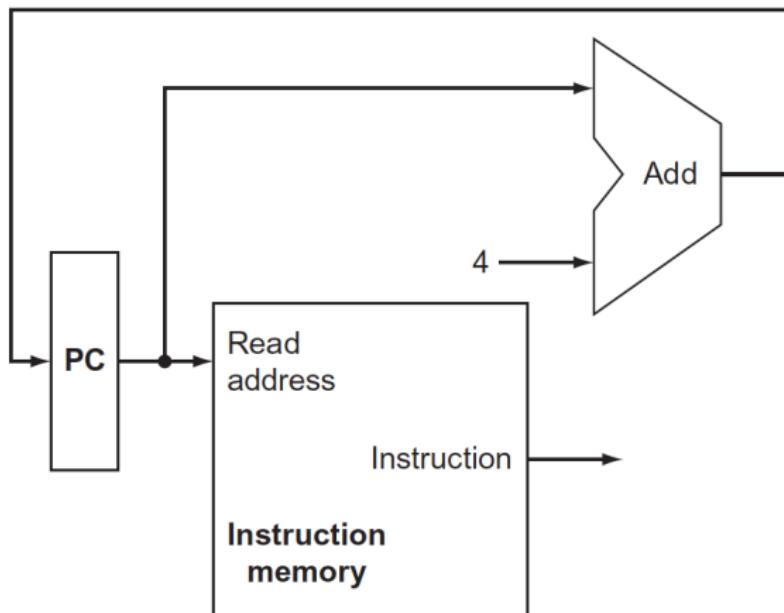
Objetivos da aula

Visão Geral da Implementação do PoliLEG

Metodologia de Clocking

Construção do Fluxo de Dados

A figura mostra como interligar os 3 elementos para formar um fluxo de dados que lê instruções e incrementa o PC para obter o endereço da próxima instrução.



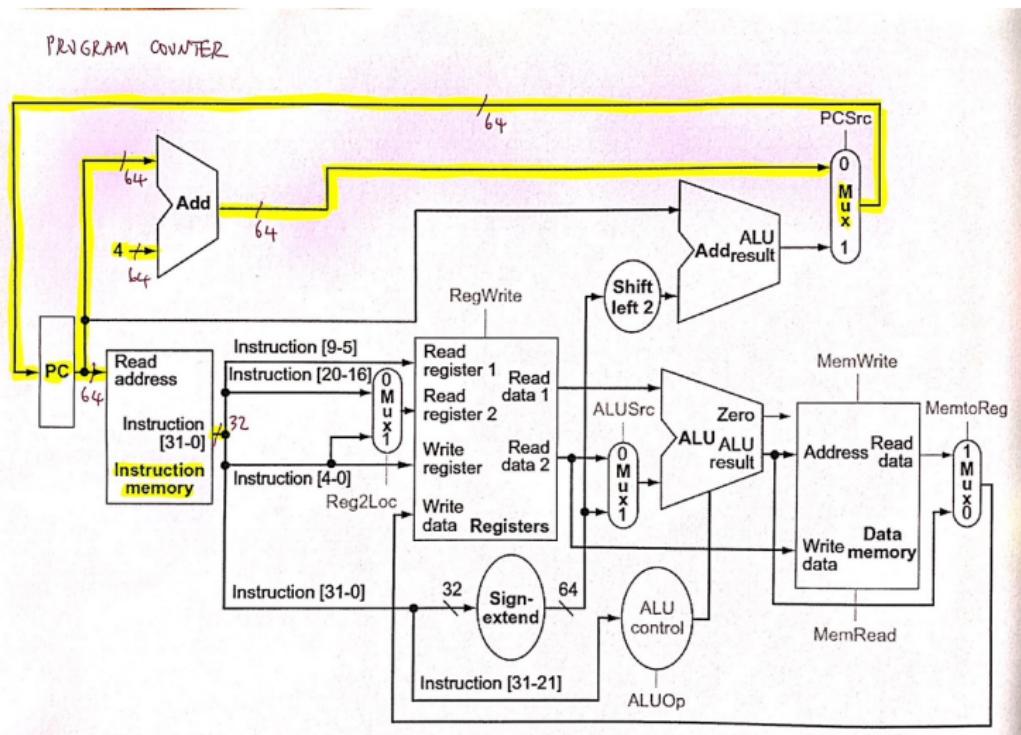
Atualização de PC

Objetivos da aula

Visão Geral da Implementação do PoliLEG

Metodología de Clocking

Construção do Fluxo de Dados



Instruções do Formato-R

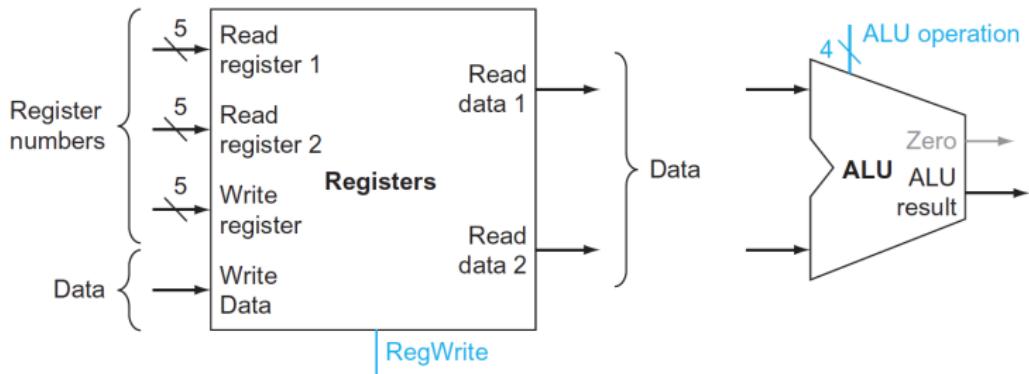
Objetivos da aula

Visão Geral da Implementação do PoliLEG

Metodologia de Clocking

Construção do Fluxo de Dados

Os 32 registradores do PoliLEG estão numa estrutura chamada *register file*. As instruções de formato R têm 3 registradores como operandos (2 para leitura e 1 para escrita). A ULA recebe 2 entradas de 64 bits e produz um resultado de 64 bits.



Instrução ADD

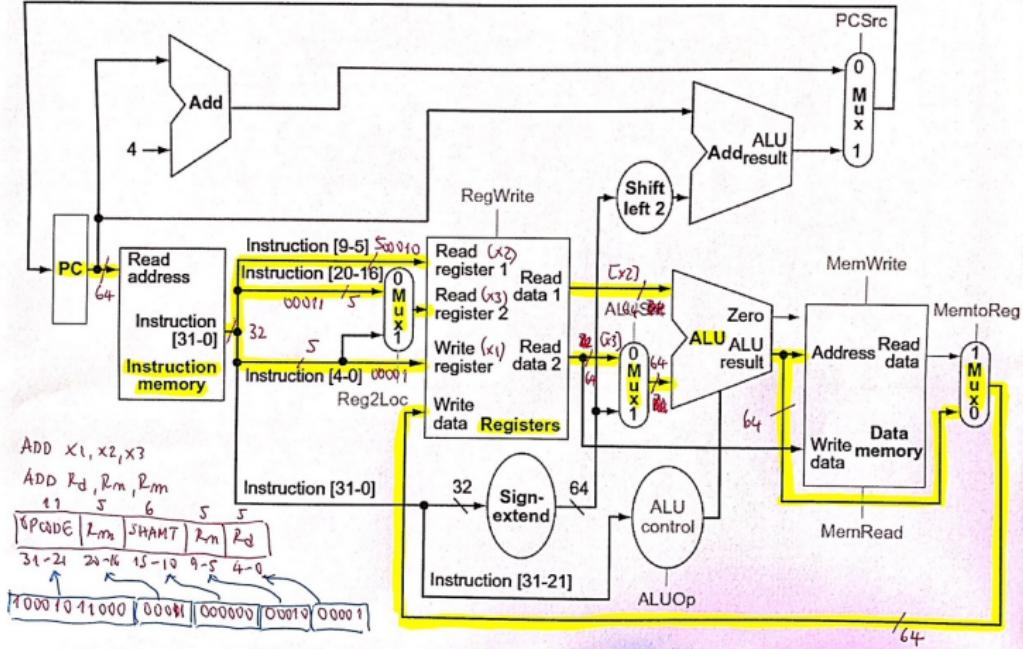
Objetivos da aula

Visão Geral da Implementação do PoliLEG

Metodologia de Clocking

Construção do Fluxo de Dados

$$ADD : \quad x_1 \leftarrow x_2 + x_3$$



Instruções do Formato-D

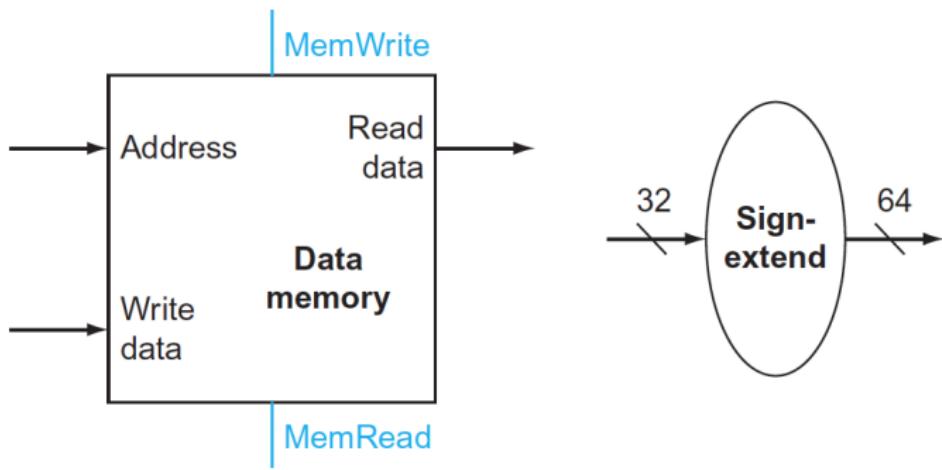
Objetivos da aula

Visão Geral da Implementação do PoliLEG

Metodologia de Clocking

Construção do Fluxo de Dados

Unidades necessárias para implementar o Load e Store:



a. Data memory unit

b. Sign extension unit

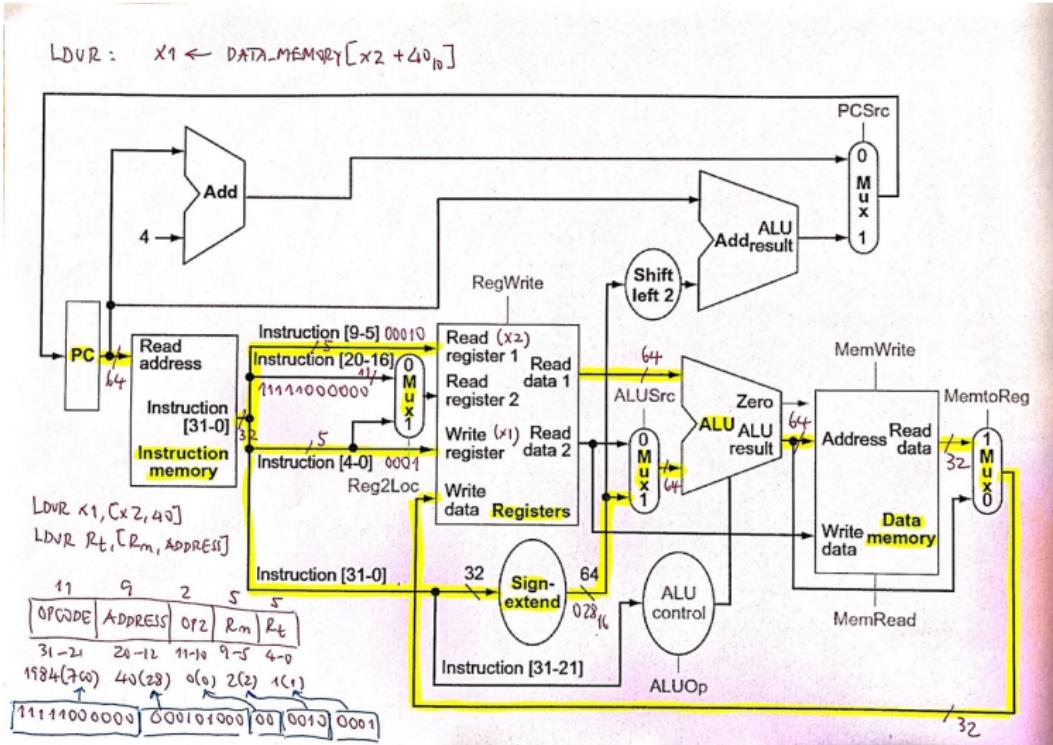
Instrução LDUR

Objetivos da aula

Visão Geral da Implementação do PoliLEG

Metodología de Clocking

Construção do Fluxo de Dados



Instruções de Salto

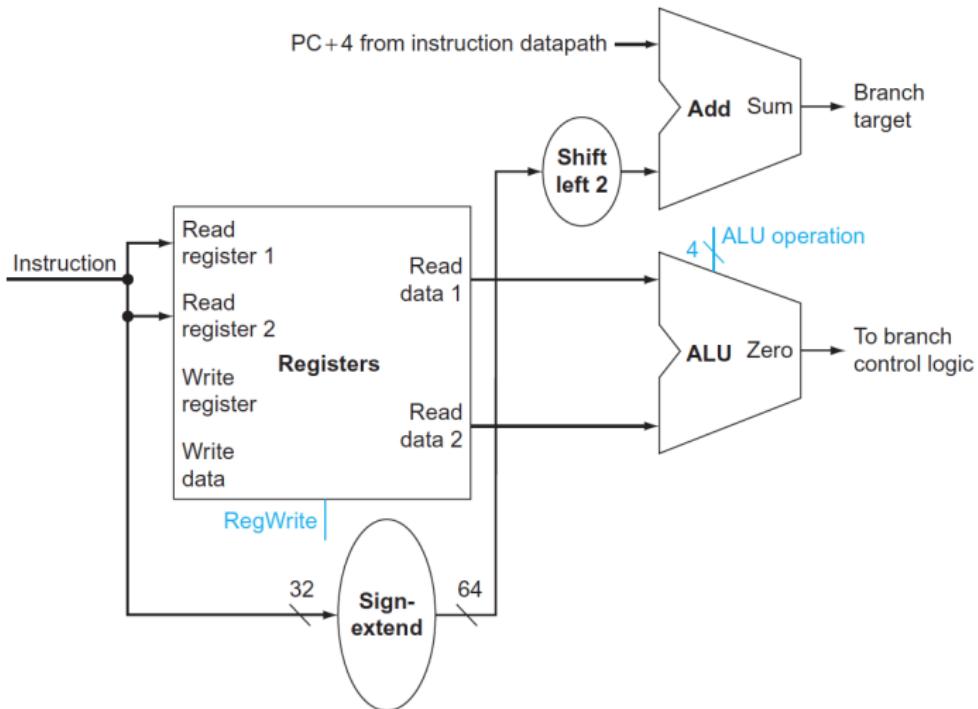
Objetivos da aula

Visão Geral da Implementação do PoliLEG

Metodologia de Clocking

Construção do Fluxo de Dados

Unidades necessárias para implementar o salto:



Juntando instruções de Salto e de Formato-R

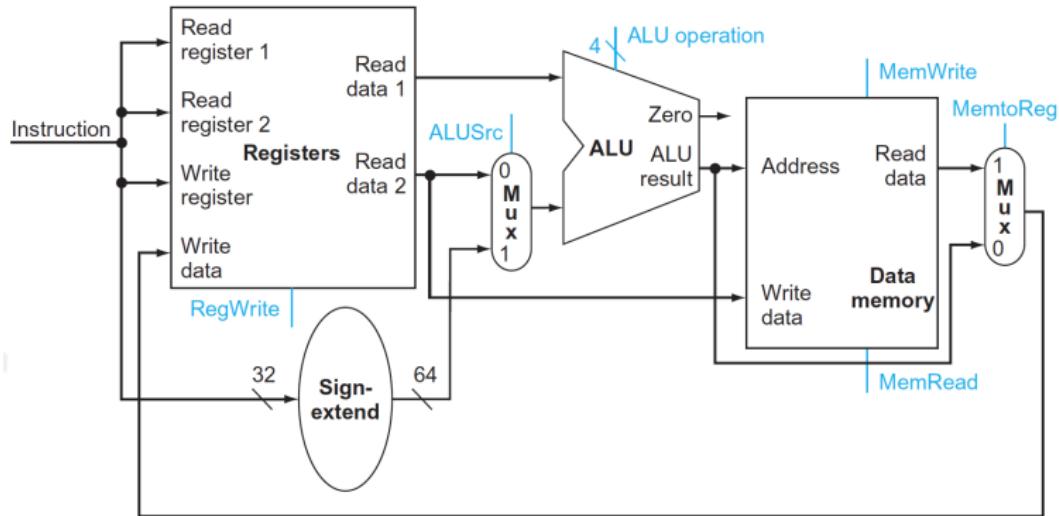
Objetivos da aula

Visão Geral da Implementação do PoliLEG

Metodologia de Clocking

Construção do Fluxo de Dados

Unidades necessárias para implementar as instruções de salto e de tipo R:



Instrução CBZ

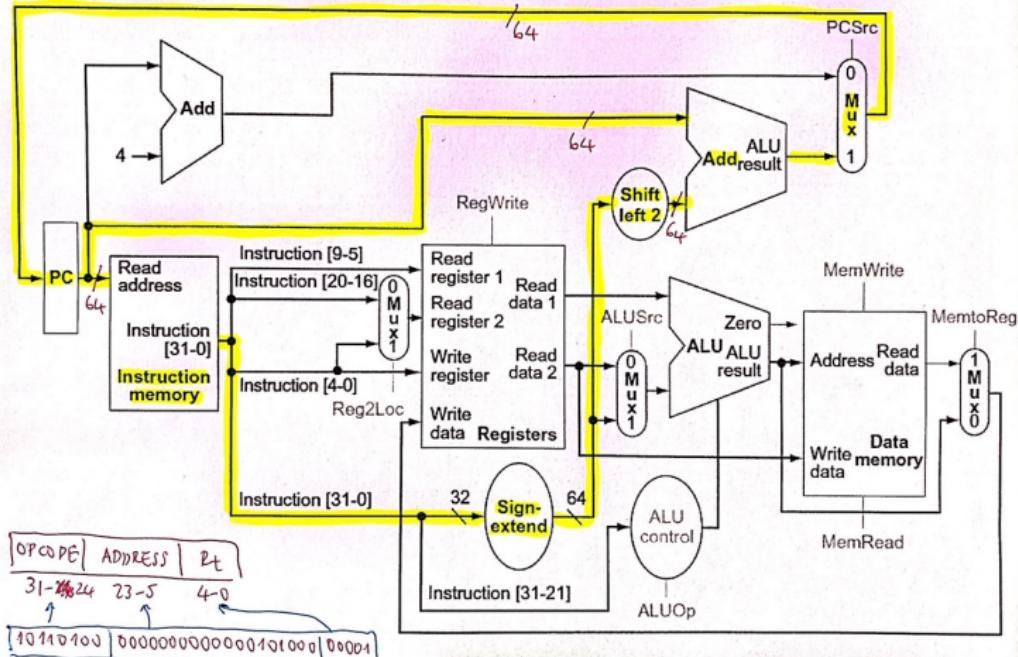
Objetivos da aula

Visão Geral da Implementação do PoliLEG

Metodologia de Clocking

Construção do Fluxo de Dados

CBZ : CBZ x1, LABEL



Fluxo de Dados para o PoliLEG

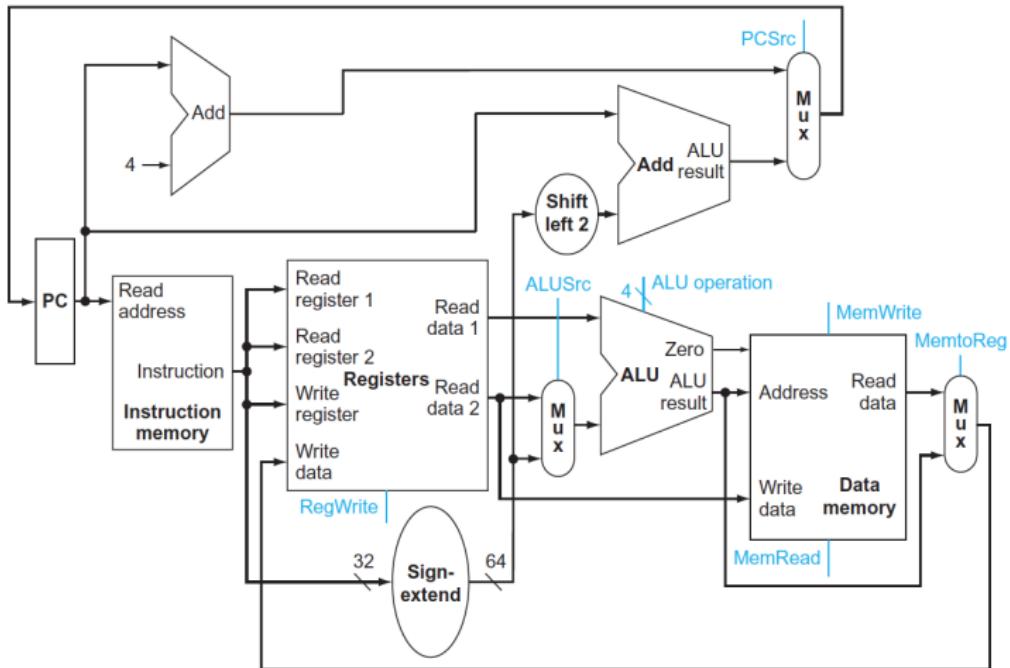
Objetivos da aula

Visão Geral da Implementação do PoliLEG

Metodologia de Clocking

Construção do Fluxo de Dados

Unidades necessárias para implementar as instruções do PoliLEG:



Leitura Recomendada

Objetivos da
aula

Visão Geral da
Implementação
do PoliLEG

Metodologia de
Clocking

Construção do
Fluxo de Dados

Esta aula foi baseada nas Seções 4.1, 4.2 e 4.3, além do Apêndice A.7 do livro *Computer Organization and Design*, do Hennessy e Patterson.

Obrigado!



Universidade de São Paulo



DEPARTAMENTO DE ENGENHARIA DE
COMPUTAÇÃO E SISTEMAS DIGITAIS

PCS



Referências

Objetivos da
aula

Visão Geral da
Implementação
do PoliLEG

Metodologia de
Clocking

Construção do
Fluxo de Dados