

# Step-by-Step Design and Simulation of a Simple CPU Architecture

## Resumo

Allex Lima   Paulo Moraes   Renan Barroncas

Bacharelado em Engenharia da Computação  
Escola de Exatas

Centro Universitário do Norte - UniNorte Laureate

Organização e Arquitetura de Computadores, 2016

# Sumário

## Introdução

## Montagem do Projeto

- Uma ULA simples

- O Datapath da CPU

- A Unidade de Controle

- Combinação dos Componentes

## Discussões e Trabalhos Futuros

# Uma breve introdução



**Derek C. Schuurman**, Ph.D., P.Eng.

Professor of Computer Science

Redeemer University College

Phone: (905) 648-2131 ext. 4273

Office: 221H

e-mail: dschuurman (at) cs.redeemer.ca

twitter: @DerekSchuurman

## *Computer science curriculum, 2008*

“A professional in any field of computing should not regard the computer as just a black box that executes programs by magic ... Students need to understand computer architecture in order to make best use of the software tools and computer languages they use to create programs”

# Sumário

Introdução

Montagem do Projeto

Uma ULA simples

O Datapath da CPU

A Unidade de Controle

Combinação dos Componentes

Discussões e Trabalhos Futuros

# A ULA

- ▶ Possui duas entradas e uma saída de um *byte*, cada;
- ▶ Executa oito operações distintas, duas lógicas e seis aritméticas;
- ▶ Possui três controladores para executar as operações ( $2^3 = 8$ );
- ▶ Implementada com o auxílio de um *Multiplexador* 1-de-8.

# A ULA

- ▶ Possui duas entradas e uma saída de um *byte*, cada;
- ▶ Executa oito operações distintas, duas lógicas e seis aritméticas;
- ▶ Possui três controladores para executar as operações ( $2^3 = 8$ );
- ▶ Implementada com o auxílio de um *Multiplexador* 1-de-8.

# A ULA

- ▶ Possui duas entradas e uma saída de um *byte*, cada;
- ▶ Executa oito operações distintas, duas lógicas e seis aritméticas;
- ▶ Possui três controladores para executar as operações ( $2^3 = 8$ );
- ▶ Implementada com o auxílio de um *Multiplexador* 1-de-8.

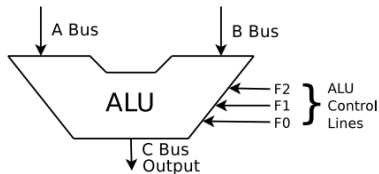
# A ULA

- ▶ Possui duas entradas e uma saída de um *byte*, cada;
- ▶ Executa oito operações distintas, duas lógicas e seis aritméticas;
- ▶ Possui três controladores para executar as operações ( $2^3 = 8$ );
- ▶ Implementada com o auxílio de um *Multiplexador* 1-de-8.



# Modelo teórico

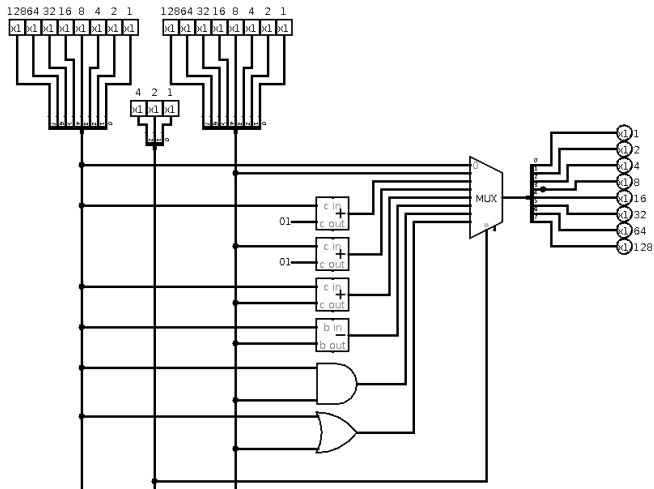
## Diagrama de Blocos



## Tabela Verdade

F2	F1	F0	Output
0	0	0	A
0	0	1	B
0	1	0	$A + 1$
0	1	1	$B + 1$
1	0	0	$A + B$
1	0	1	$A - B$
1	1	0	$A \text{ AND } B$
1	1	1	$A \text{ OR } B$

# Implementação no *Logisim*



# Sumário

Introdução

Montagem do Projeto

Uma ULA simples

O Datapath da CPU

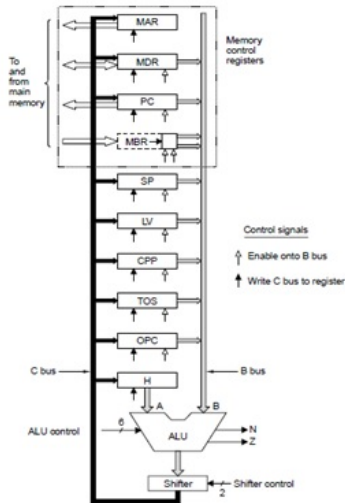
A Unidade de Controle

Combinação dos Componentes

Discussões e Trabalhos Futuros

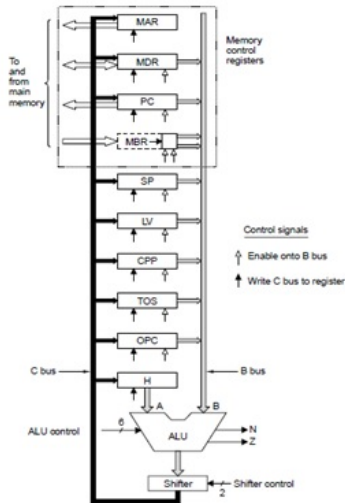
# O Datapath

- ▶ Uma versão simplificada da arquitetura *MIC-1*;
- ▶ Possui barramentos de um *byte*;
- ▶ Dispõe de sete registradores:
  - ▶ Dois de uso geral (*A* e *B*);
  - ▶ Um exclusivo para os resultados (*Out*); e
  - ▶ Quatro reservados para uso especial:
    - ▶ *MBR* Memory Byte Register;
    - ▶ *PC* Program Counter;
    - ▶ *MIR* Microprogram Instruction Register; e
    - ▶ *MIC* Microprogram Program Counter.



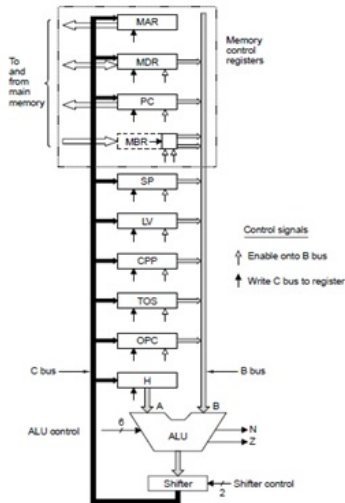
# O Datapath

- ▶ Uma versão simplificada da arquitetura *MIC-1*;
- ▶ Possui barramentos de um byte;
- ▶ Dispõe de sete registradores:
  - ▶ Dois de uso geral (A e B);
  - ▶ Um exclusivo para os resultados (Out); e
  - ▶ Quatro reservados para uso especial:
    - ▶ MBR Memory Byte Register;
    - ▶ PC Program Counter;
    - ▶ MIR Microprogram Instruction Register; e
    - ▶ MIC Microprogram Program Counter.



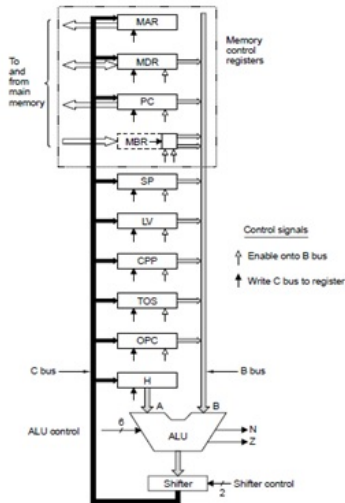
# O Datapath

- ▶ Uma versão simplificada da arquitetura *MIC-1*;
- ▶ Possui barramentos de um byte;
- ▶ Dispõe de sete registradores:
  - ▶ Dois de uso geral (A e B);
  - ▶ Um exclusivo para os resultados (Out); e
  - ▶ Quatro reservados para uso especial:
    - ▶ MBR Memory Byte Register;
    - ▶ PC Program Counter;
    - ▶ MIR Microprogram Instruction Register; e
    - ▶ MIC Microprogram Program Counter.

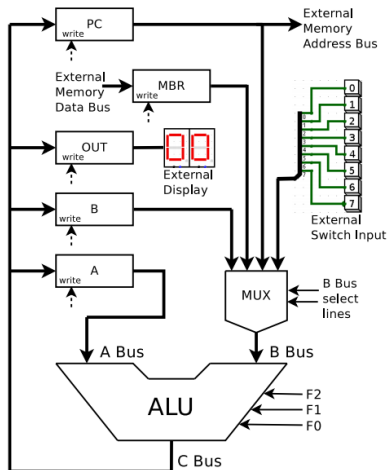


# O Datapath

- ▶ Uma versão simplificada da arquitetura *MIC-1*;
- ▶ Possui barramentos de um byte;
- ▶ Dispõe de sete registradores:
  - ▶ Dois de uso geral (**A** e **B**);
  - ▶ Um exclusivo para os resultados (**Out**); e
  - ▶ Quatro reservados para uso especial:
    - ▶ **MBR** Memory Byte Register;
    - ▶ **PC** Program Counter;
    - ▶ **MIR** Microprogram Instruction Register; e
    - ▶ **MIC** Microprogram Program Counter.



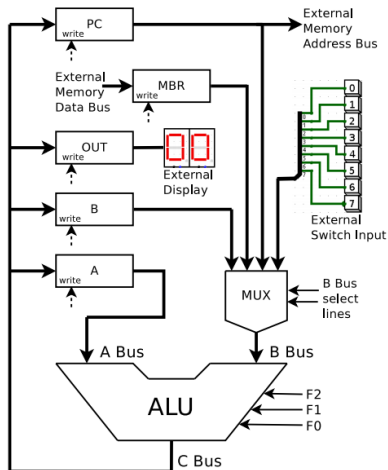
# O modelo utilizado



- ▶ A ULA sempre terá o registrador **A** associado a uma de suas entradas;
- ▶ A outra entrada da ULA possui como origem:
  - ▶ O registrador **B**;
  - ▶ O **MBR**;
  - ▶ O **PC**; ou
  - ▶ Uma fonte externa.
- ▶ A saída da ULA terá como destino o registrador que estiver habilitado a receber dados (**write** ativo).

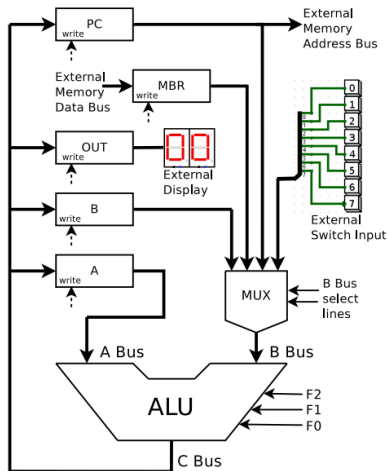


# O modelo utilizado



- ▶ A ULA sempre terá o registrador **A** associado a uma de suas entradas;
- ▶ A outra entrada da ULA possui como origem:
  - ▶ O registrador **B**;
  - ▶ O **MBR**;
  - ▶ O **PC**; ou
  - ▶ Uma fonte externa.
- ▶ A saída da ULA terá como destino o registrador que estiver habilitado a receber dados (**write** ativo).

# O modelo utilizado



- ▶ A ULA sempre terá o registrador **A** associado a uma de suas entradas;
- ▶ A outra entrada da ULA possui como origem:
  - ▶ O registrador **B**;
  - ▶ O **MBR**;
  - ▶ O **PC**; ou
  - ▶ Uma fonte externa.
- ▶ A saída da ULA terá como destino o registrador que estiver habilitado a receber dados (**write** ativo).

# Sumário

Introdução

Montagem do Projeto

Uma ULA simples

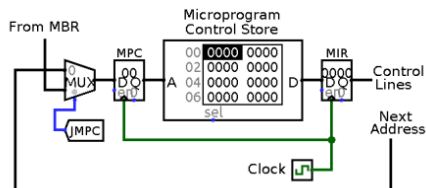
O Datapath da CPU

A Unidade de Controle

Combinação dos Componentes

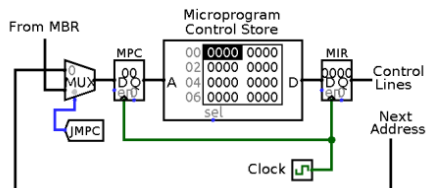
Discussões e Trabalhos Futuros

# A Unidade de Controle



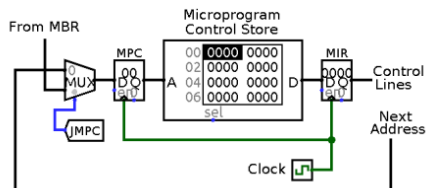
- ▶ Uma ROM (*Read Only Memory*) de 32x16-bits é utilizada para armazenar o *microprograma*;
- ▶ O registrador **MPC** contém os 5 bits do *endereço* da instrução;
- ▶ O seletor **JMPC** irá encaminhar para o **MPC** um endereço do **MBR** ou da próxima instrução;
- ▶ O **MIR** armazena os 16 bits da *instrução em si*;
- ▶ O **MPC** e o **MIR** são atualizados a cada *clock*;

# A Unidade de Controle



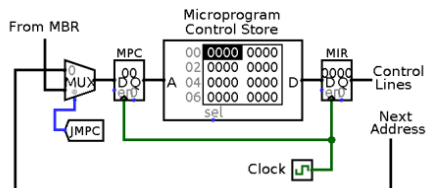
- ▶ Uma ROM (*Read Only Memory*) de 32x16-bits é utilizada para armazenar o *microprograma*;
- ▶ O registrador **MPC** contém os 5 bits do *endereço* da instrução;
- ▶ O seletor **JMPC** irá encaminhar para o **MPC** um endereço do **MBR** ou da próxima instrução;
- ▶ O **MIR** armazena os 16 bits da *instrução em si*;
- ▶ O **MPC** e o **MIR** são atualizados a cada *clock*;

# A Unidade de Controle



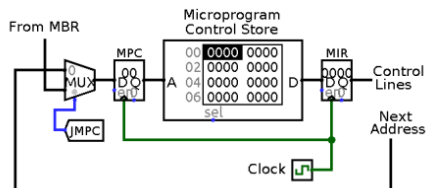
- ▶ Uma ROM (*Read Only Memory*) de 32x16-bits é utilizada para armazenar o *microprograma*;
- ▶ O registrador **MPC** contém os 5 bits do *endereço* da instrução;
- ▶ O seletor **JMPC** irá encaminhar para o **MPC** um endereço do **MBR** ou da próxima instrução;
- ▶ O **MIR** armazena os 16 bits da *instrução em si*;
- ▶ O **MPC** e o **MIR** são atualizados a cada *clock*;

# A Unidade de Controle



- ▶ Uma ROM (*Read Only Memory*) de 32x16-bits é utilizada para armazenar o *microprograma*;
- ▶ O registrador **MPC** contém os 5 bits do *endereço* da instrução;
- ▶ O seletor **JMPC** irá encaminhar para o **MPC** um endereço do **MBR** ou da próxima instrução;
- ▶ O **MIR** armazena os 16 bits da *instrução em si*;
- ▶ O **MPC** e o **MIR** são atualizados a cada *clock*;

# A Unidade de Controle



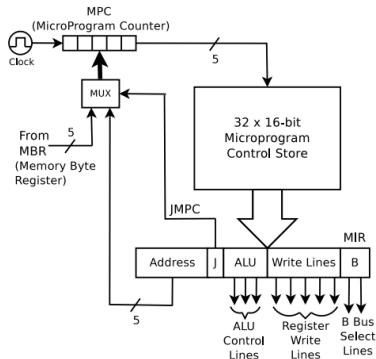
- ▶ Uma ROM (*Read Only Memory*) de 32x16-bits é utilizada para armazenar o *microprograma*;
- ▶ O registrador **MPC** contém os 5 bits do *endereço* da instrução;
- ▶ O seletor **JMPC** irá encaminhar para o **MPC** um endereço do **MBR** ou da próxima instrução;
- ▶ O **MIR** armazena os 16 bits da *instrução em si*;
- ▶ O **MPC** e o **MIR** são atualizados a cada *clock*;



# A Unidade de Controle

- ▶ Os bits que encontram-se no **MIR** possuem a organização:

- ▶ **Endereço**: 5 bits;
- ▶ Seletor **JMPC** (J): 1 bit;
- ▶ **Controladores da ULA**: 3 bits;
- ▶ **Controle de escrita**: 5 bits; e
- ▶ Seletor da entrada **B** da ULA: 2 bits.



# Sumário

Introdução

Montagem do Projeto

Uma ULA simples

O Datapath da CPU

A Unidade de Controle

Combinação dos Componentes

Discussões e Trabalhos Futuros

# Combinando os processos

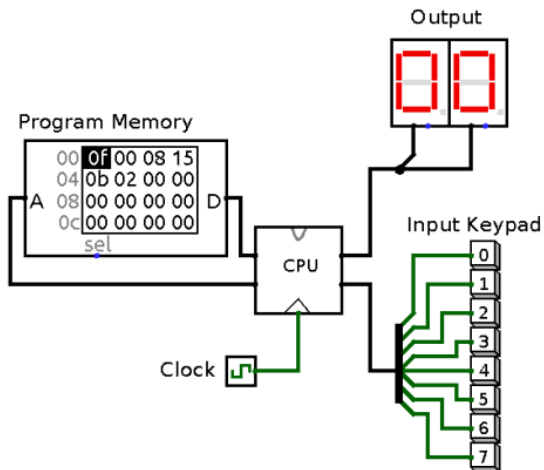


Figure: Resultado

# Discussão e Trabalhos Futuros

1. Práticas estudantis;
2. Mesclagem entre bits e logisim;
3. Incremento do processador para novos modelos;
4. Assembly no logisim.

# Discussão e Trabalhos Futuros

1. Práticas estudantis;
2. Mesclagem entre bits e logisim;
3. Incremento do processador para novos modelos;
4. Assembly no logisim.

# Discussão e Trabalhos Futuros

1. Práticas estudantis;
2. Mesclagem entre bits e logisim;
3. Incremento do processador para novos modelos;
4. Assembly no logisim.

# Discussão e Trabalhos Futuros

1. Práticas estudantis;
2. Mesclagem entre bits e logisim;
3. Incremento do processador para novos modelos;
4. Assembly no logisim.

# Discussão e Trabalhos Futuros

1. Práticas estudantis;
2. Mesclagem entre bits e logisim;
3. Incremento do processador para novos modelos;
4. Assembly no logisim.



# Referências



Derek C. Schuurman.

Step-by-step design and simulation of a simple CPU architecture

*Proceeding of the 44th ACM technical symposium on Computer science education (SIGCSE '13)*, ACM, New York, NY, USA, 335-340. 2013.