Organização e Arquitetura de Computadores

Prof. Lúcio Renê Prade
Prof. Rodrigo Marques de Figueiredo

Competências

- Compreender os conceitos básicos relacionados à estrutura e funcionamento dos computadores digitais.
- Reconhecer o funcionamento dos microcomputadores e periféricos a partir da análise de seus componentes: Sistema de memória, sistema de interconexão, sistema de entrada/saída, unidade central de processamento.
- Relacionar o funcionamento da organização interna do processador com as características das linguagens de montagem, conjunto de instruções, montadores, carregadores, ligadores.
- Compreender a evolução das arquiteturas computacionais como a utilização de paralelismo, máquinas escalares e as tendências das arquiteturas futuras.

Programa

1 - Revisão de sistemas digitais, bases numéricas bin hex e dec, portas lógicas, aritmética, circuitos sequencias e FSM.

2 - Arquitetura e organização de computadores: visão geral, Componentes do computador e organização interna.

3 - Introdução à Arquitetura de Computadores, visão do Software, O compilador, montador, ligador, carregador e processo de boot

4 - Unidade Central de Processamento, funcionamento em alto nível dos processadores, Arquitetura de processadores Harvard e Von Neumann. Estratégias CISC e RISC

5 - Linguagem de Montagem, Instruções assembly, Operações e operandos, Instruções Lógicas e Aritméticas, Instruções de desvio, instruções de manipulação de dados. Uso do simulador MIPS assembly

6 - Linguagem de Máquina, Codificação das Instruções assembly em linguagem de máquinas, organização das instruções na memória, Anatomia de um arquivo executável.

7 - Máquinas Von Neumann, organização interna, componentes, descontruir o simulador Neander

UNISINOS
DESAFIE
O AMANHÃ.

Programa

8 - Maquinas Harvard, organização interna, componente, simulador RISC-V

9 - Sistema de Memória, Características e tipos, Estrutura e organização, Hierarquia de memória, Memória principal, Memória virtual, Memória cache, Memória secundária

10 - Sistema de Entrada e Saída, Sistema de Interconexão: Barramentos, crossbar e NOCs, Módulos de E/S, Técnicas de Controle para Transferência de Dados, PIO, Interrupção e DMA

11 - Desempenho de um processador, Métricas de desempenho, Benchmarking, Desempenho de software (compiladores)

12 - Evolução dos computadores, Máquinas Escalares e Paralelismo de instruções (pipeline)

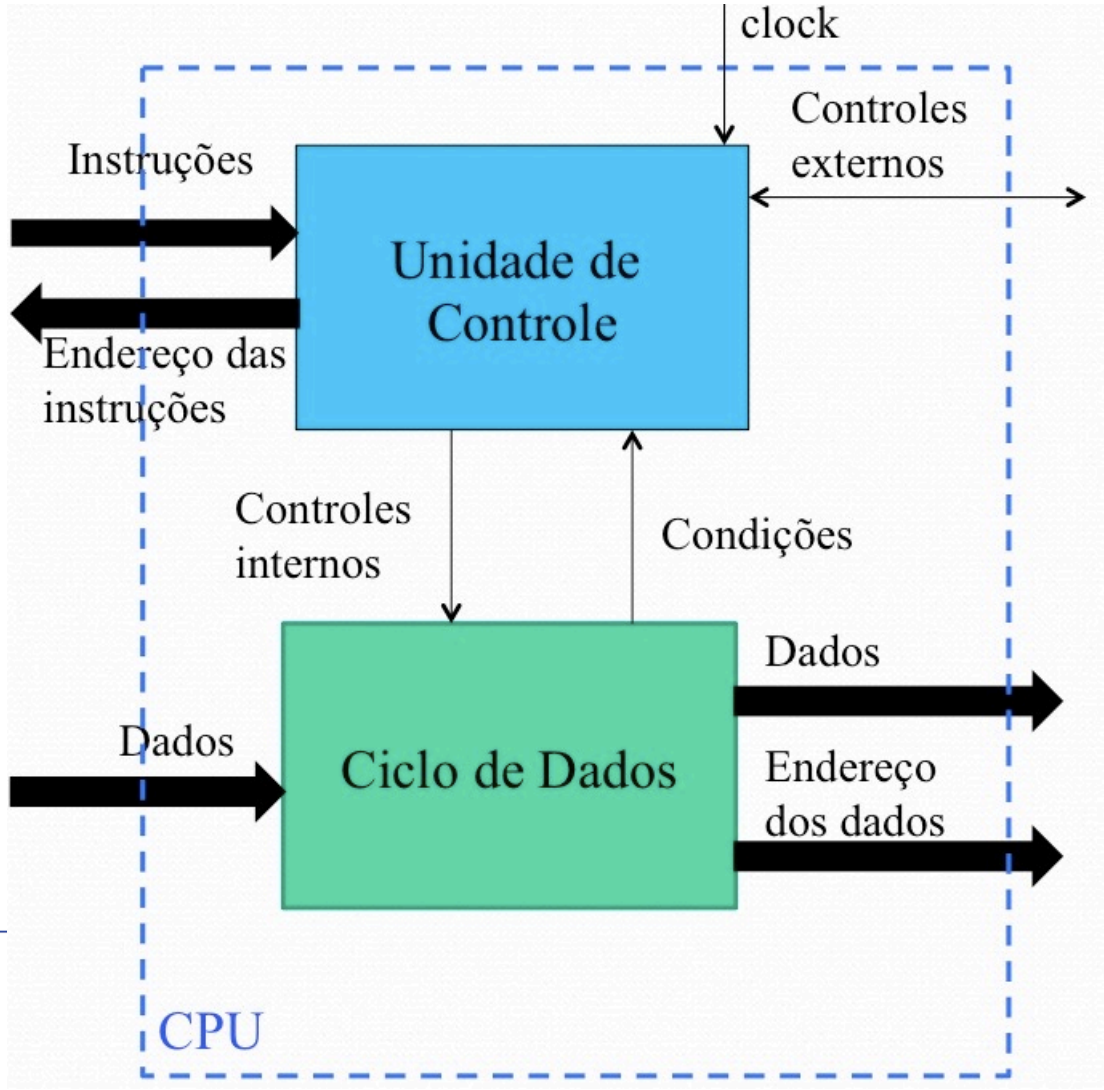
13 - Evolução dos computadores, Máquinas Escalares e Paralelismo de instruções (pipeline), conflitos de dados, controle e estruturais

14 - Evolução dos computadores, Máquinas Superescalares e VLIW

15 - Evolução dos computadores, Multicomputadores e Multiprocessadores, tendencias.

UNISINOS
DESAFIE
O AMANHÃ.

Processador



Arquitetura x Organização

Arquitetura refere-se aos atributos que são visíveis para o programador, ou seja, os atributos que tem impacto direto na execução do programa.

Atributos:

- Conjunto de instruções
- Número de bits
- Mecanismos de E/S

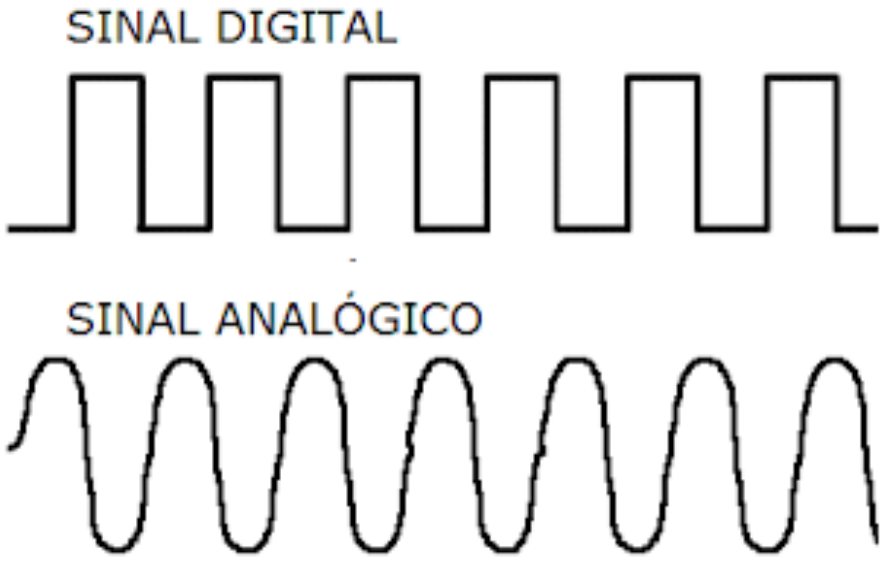
Arquitetura x Organização

Organização diz respeito às unidades operacionais e suas interconexões que implementam as especificações de sua arquitetura, ou seja, como as características da arquitetura será implementada.

Atributos:

- Sinais de controle
- Tecnologia de memória, tecnologia de transistores etc.

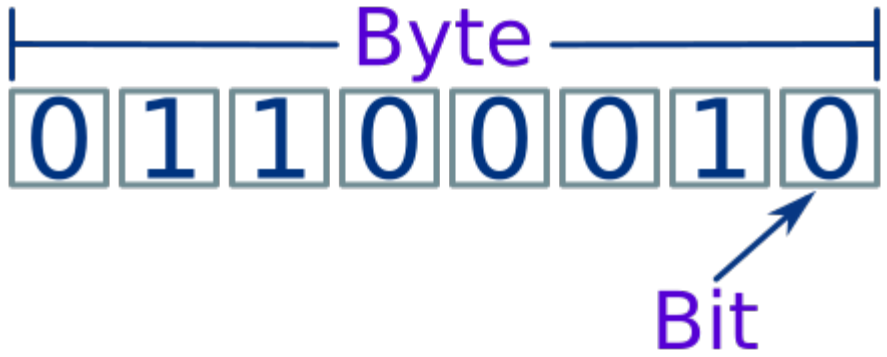
Sistemas Digitais



Sistemas Digitais trabalham com um número limitado de símbolos

UNISINOS
DESAFIE O AMANHÃ.

Sistemas Digitais Binários



Sistemas Digitais Binários trabalham com apenas 2 valores ‘0’ e ‘1’

UNISINOS
DESAFIE O AMANHÃ.

Sistemas Digitais Binários

Prefixo	Potência de Base 2	Quantidade de bytes	Símbolo
	2^0	1	B
K	2^{10}	1.024	KB
M	2^{20}	1.048.576	MB
G	2^{30}	1.073.741.824	GB

UNISINOS
DESAFIE O AMANHÃ.

Sistemas Numéricos

Base 10 (Decimal) 0,1,2,3,4,5,6,7,8,9

$$(347)_{10} = 3 * 10^2 + 4 * 10^1 + 7 * 10^0$$

$$(32)_{10} = 3 * 10^1 + 2 * 10^0$$

Base 2 (Binário) 0,1

$$(11)_2 = 1 \times 2^1 + 1 \times 2^0 = (3)_{10}$$

$$(111)_2 = 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (7)_{10}$$

$$(10111)_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (23)_{10}$$

UNISINOS
DESAFIE O AMANHÃ.

Sistemas Numéricos

Base 8 (Octal) 0,1,2,3,4,5,6,7

$$(502)_8 = 5 \times 8^2 + 0 \times 8^1 + 2 \times 8^0 = (322)_{10}$$

$$(22)_8 = 2 \times 8^1 + 2 \times 8^0 = (18)_{10}$$

Base 16 (Hexadecimal) 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

$$(12C)_{16} = 1 \times 16^2 + 2 \times 16^1 + C \times 16^0 = (300)_{10}$$

$$(BF)_{16} = B \times 16^1 + F \times 16^0 = (191)_{10}$$

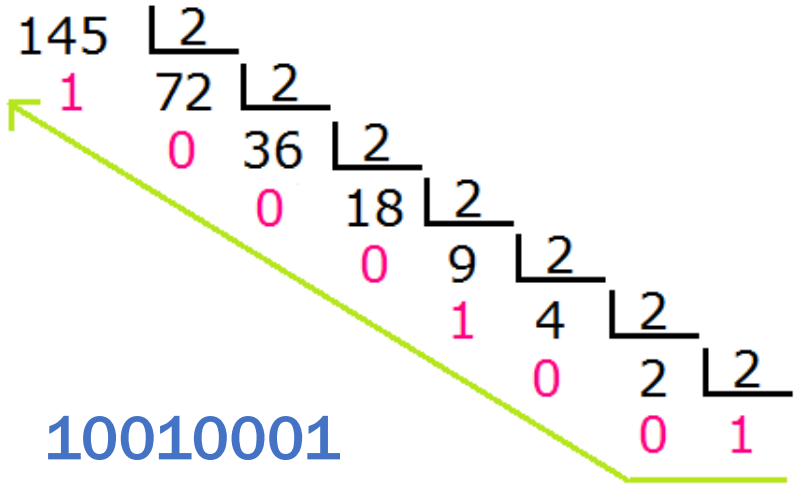
$$(54CC)_{16} = 5 \times 16^3 + 4 \times 16^2 + C \times 16^1 + C \times 16^0 = (21708)_{10}$$

UNISINOS
DESAFIE O AMANHÃ.

Conversão de Base Numéricas

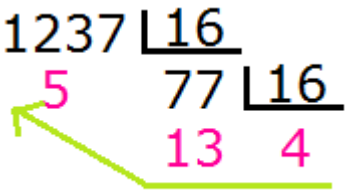
Método das Divisões – decimal para qualquer base

Decimal -> Binário



10010001

Decimal -> Hexadecimal



4D5

UNISINOS
DESAFIE O AMANHÃ.

Conversão de Base Numéricas

Método Polinomial – Qualquer base para decimal.

Binário -> Decimal

$$111010_2$$

$$1.2^5 + 1.2^4 + 1.2^3 + 0.2^2 + 1.2^1 + 0.2^0$$

$$32 + 16 + 8 + 0 + 2 + 0 = 58$$

$$111010_2 = 58_{10}$$

Hexadecimal -> Decimal

$$A3_{(16)} = 163_{(10)}$$

$$10 \times 16^1 + 3 \times 16^0$$

$$160 + 3 = 163_{(10)}$$

UNISINOS
DESAFIE O AMANHÃ.

Aritmética Binária

bit de carry (vai um)

```
    1
MSB 1010 LSB
  + 0010
  ------
    1100
```

0 + 0 = 0

1 + 0 = 1

1 + 1 = 10 | 0 +carry 1

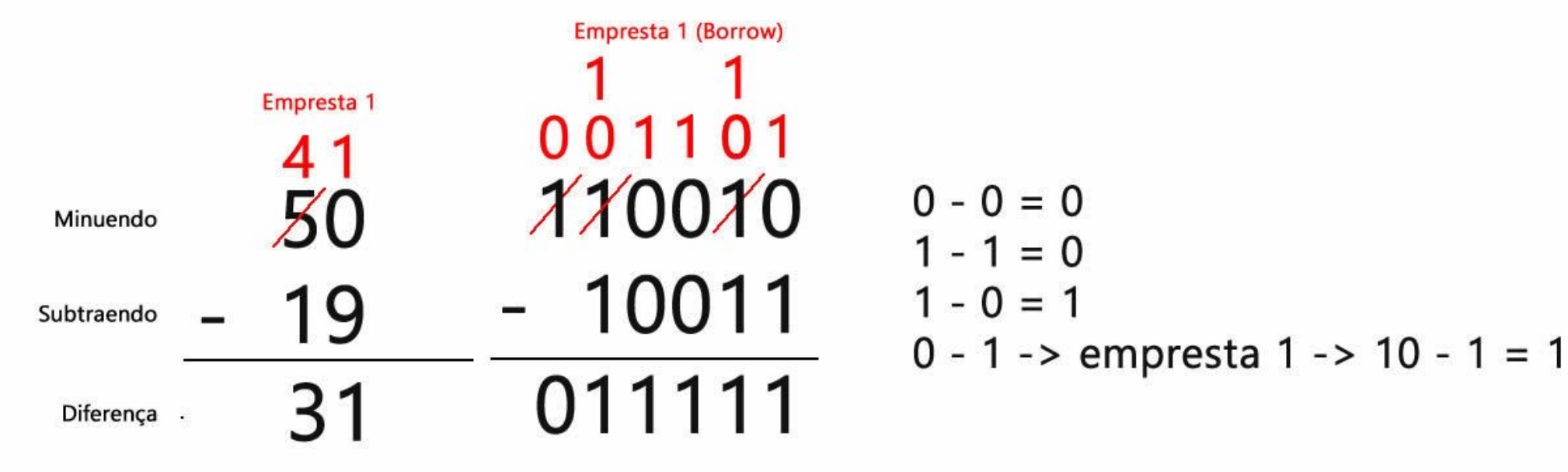
1 + 1 + 1 = 11 | 1 +carry 1

```
    1
  1001
+ 0101
------
  1110
```

Soma 1 + 1 = 0 e resta 1.
O resto sobe para soma da próxima casa.

UNISINOS
DESAFIE O AMANHÃ.

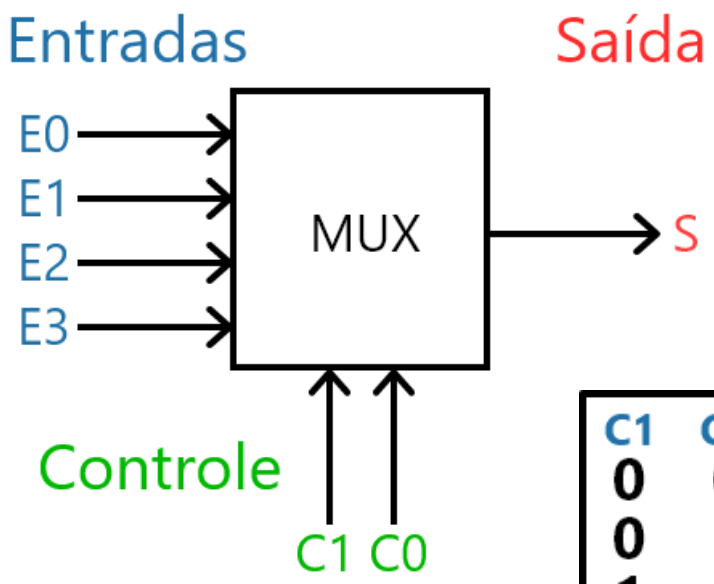
Aritmética Binária



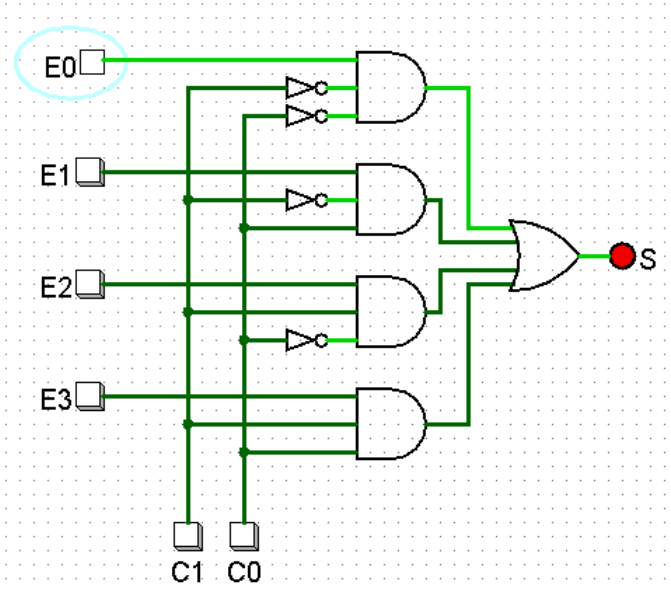
UNISINOS
DESAFIE O AMANHÃ.

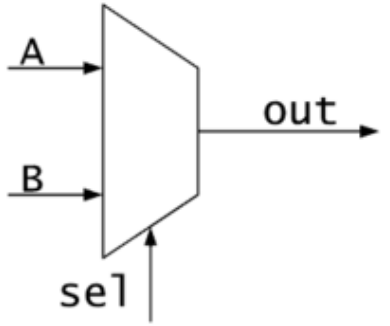
Circuitos Combinacionais Fundamentais

Multiplexador



C1	C0	S
0	0	E0
0	1	E1
1	0	E2
1	1	E3

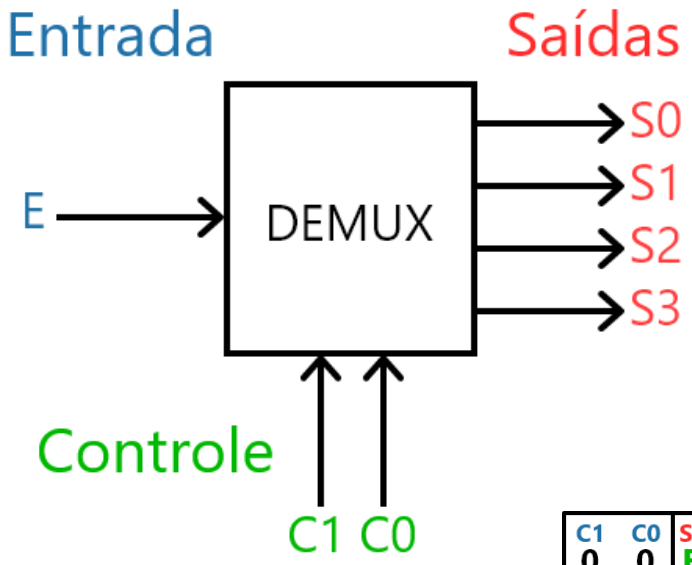




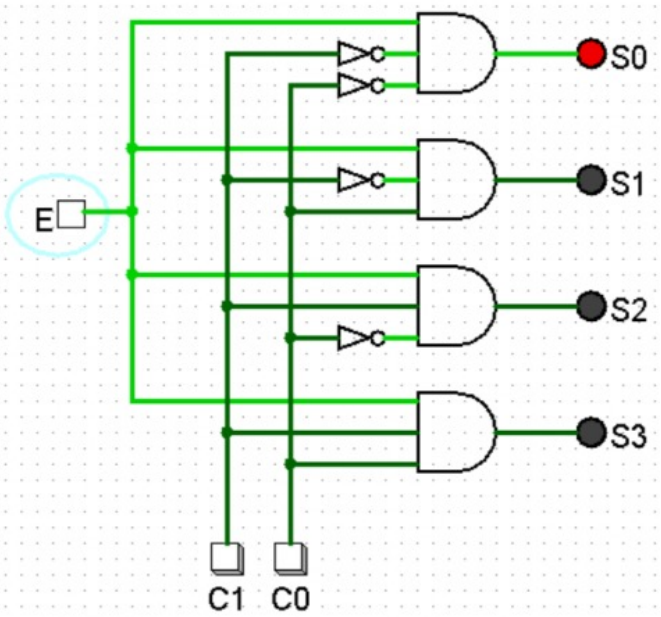
UNISINOS
DESAFIE O AMANHÃ.

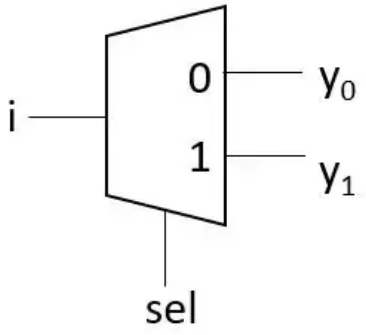
Circuitos Combinacionais Fundamentais

Demultiplexador



C1	C0	S0	S1	S2	S3
0	0	E	0	0	0
0	1	0	E	0	0
1	0	0	0	E	0
1	1	0	0	0	E





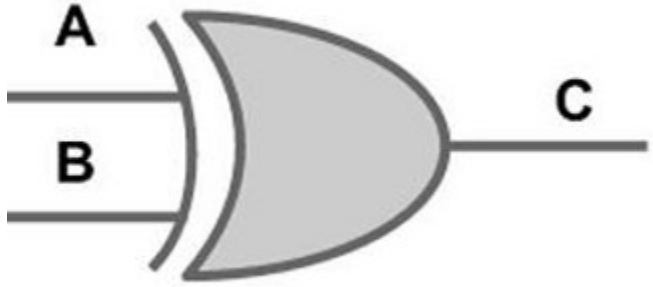
1:2 DEMUX

UNISINOS
DESAFIE O AMANHÃ.

Circuitos Combinacionais Fundamentais

Logica XOR

OU EXCLUSIVO (XOR) $C=A \oplus B$



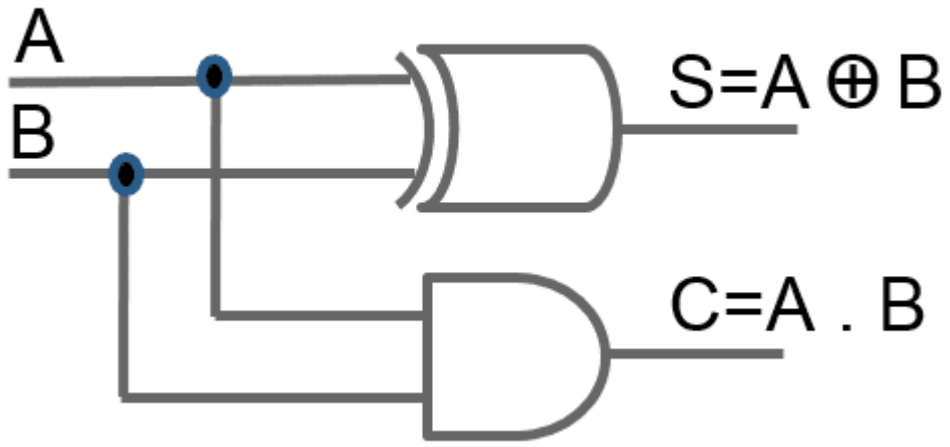
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

UNISINOS
DESAFIE O AMANHÃ.

Circuitos Combinacionais Fundamentais

Meio Somador (Half Adder)

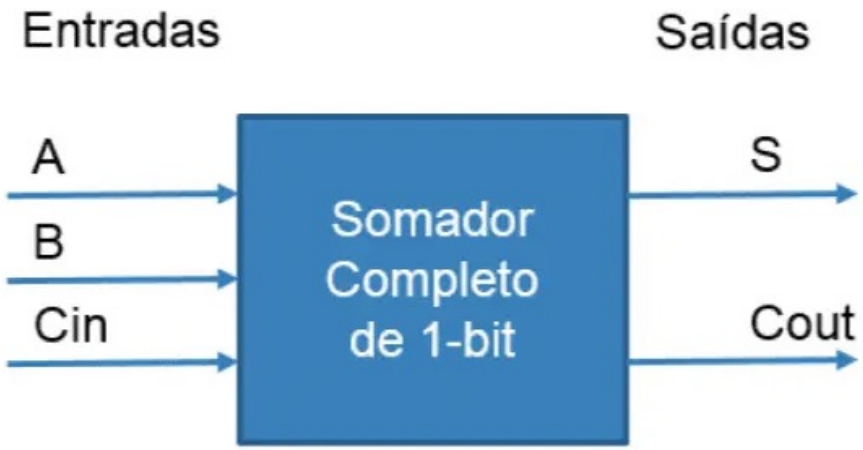
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



UNISINOS
DESAFIE O AMANHÃ.

Circuitos Combinacionais Fundamentais

Somador Completo (Full Adder)



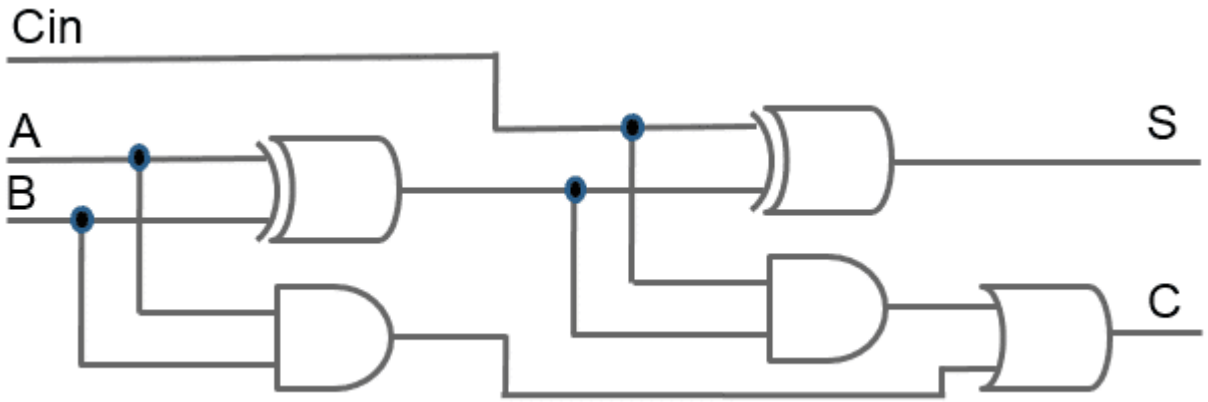
A	B	CIN	S	COUT
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

UNISINOS
DESAFIE O AMANHÃ.

Circuitos Combinacionais Fundamentais

Somador Completo (Full Adder)

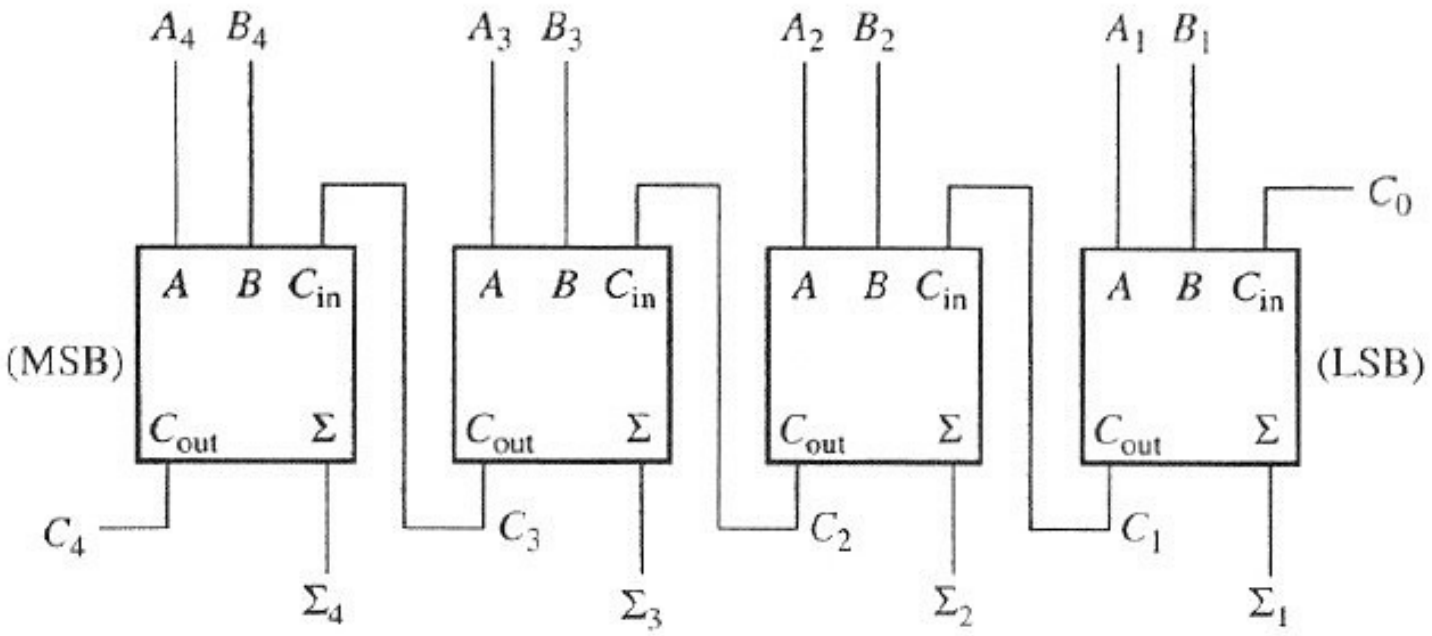
A	B	Cin	S	Cout
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1



UNISINOS
DESAFIE O AMANHÃ.

Circuitos Combinacionais Fundamentais

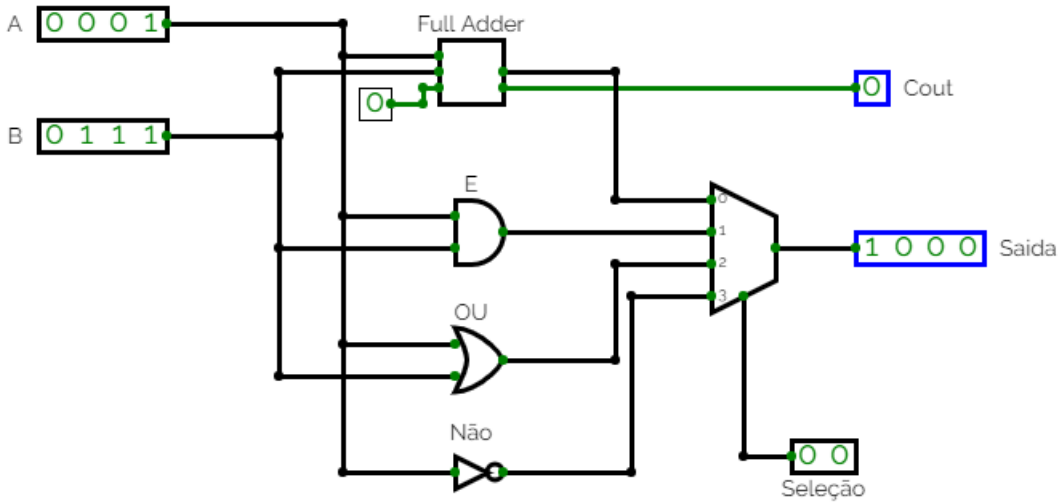
Somador Completo (Full Adder)

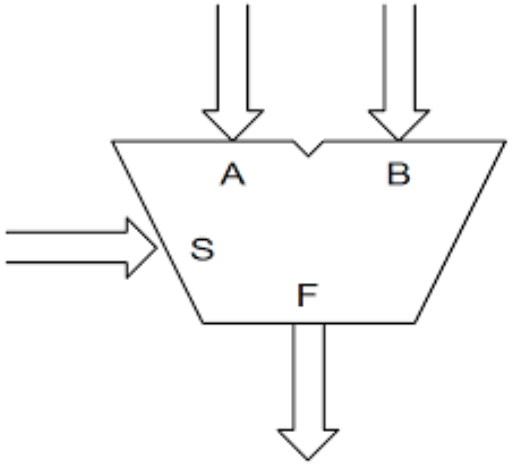


UNISINOS
DESAFIE O AMANHÃ.

Circuitos Combinacionais Fundamentais

Unidade Lógica Aritmética (ULA)





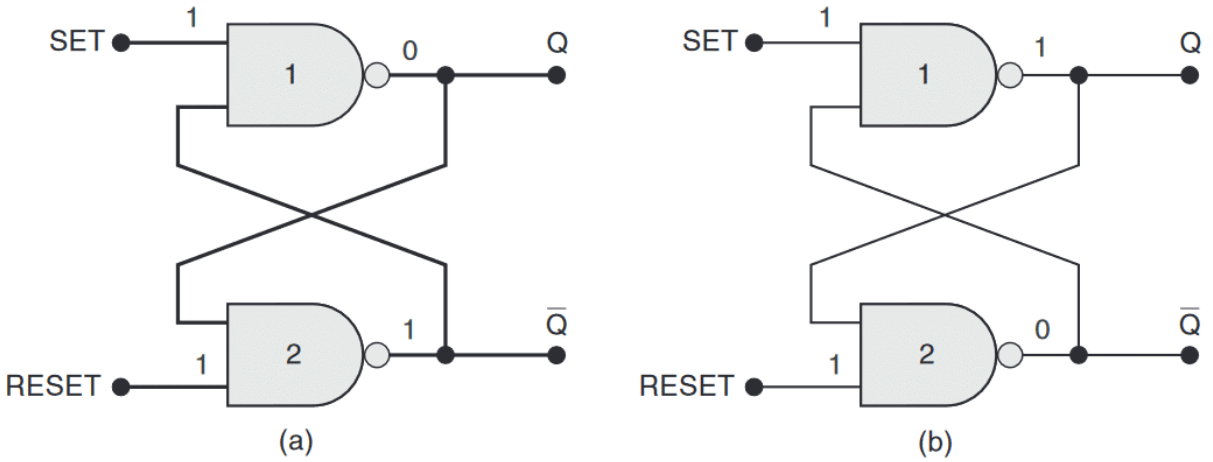
UNISINOS
DESAFIE O AMANHÃ.

Circuitos Sequenciais

Até agora os circuitos lógicos estudados são classificados como combinacionais, pois o nível lógico de saída, em qualquer instante de tempo, depende somente dos níveis lógicos de entrada naquele mesmo instante. Não há influência de nenhuma condição de entrada anterior sobre as saídas atuais, já que os circuitos combinacionais não possuem memória.

Circuitos Sequenciais

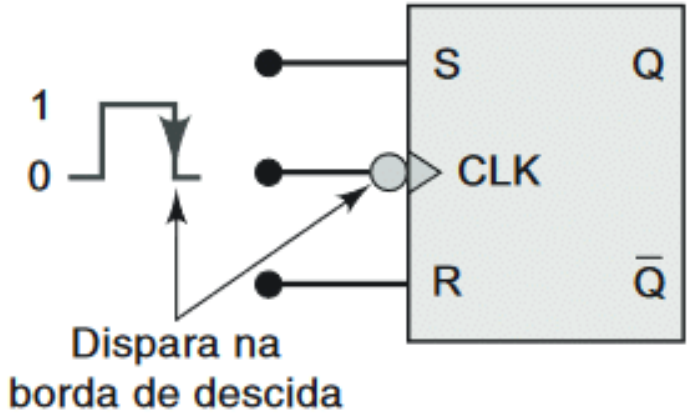
Na maioria dos sistemas digitais, há a presença de circuitos combinacionais como elementos de memória. O flip-flop é o componente de memória mais importante e é constituído por um conjunto de portas lógicas interconectadas. Embora uma única porta lógica não possua capacidade de armazenamento, algumas delas podem ser conectadas de maneira a permitir guardar informações. .



UNISINOS
DESAFIE
O AMANHÃ.

Circuitos Sequenciais

Flip Flop SR

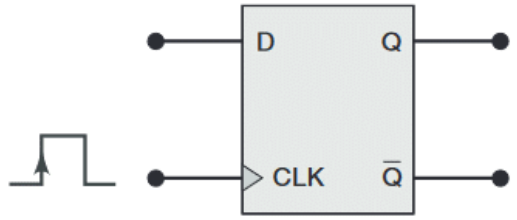


Entradas			Saída
S	R	CLK	Q
0	0	↓	Q_0 (não muda)
1	0	↓	1
0	1	↓	0
1	1	↓	Ambíguo

UNISINOS
DESAFIE
O AMANHÃ.

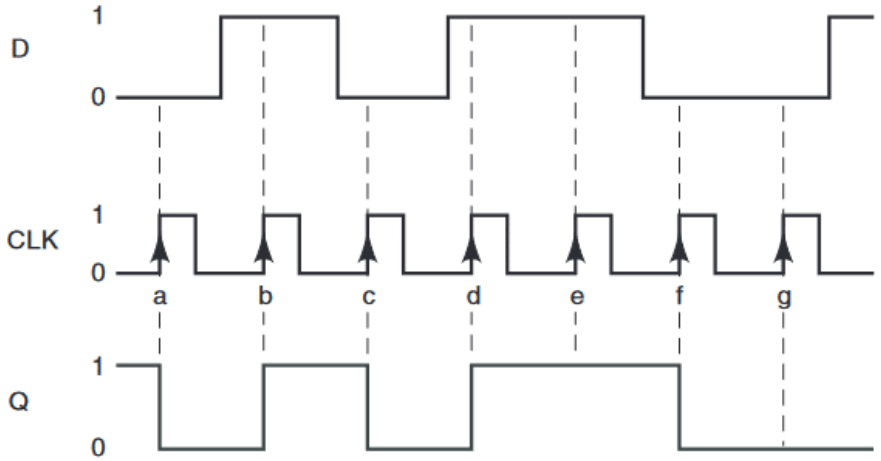
Circuitos Sequenciais

Flip Flop D



D	CLK	Q
0	↑	0
1	↑	1

(a)

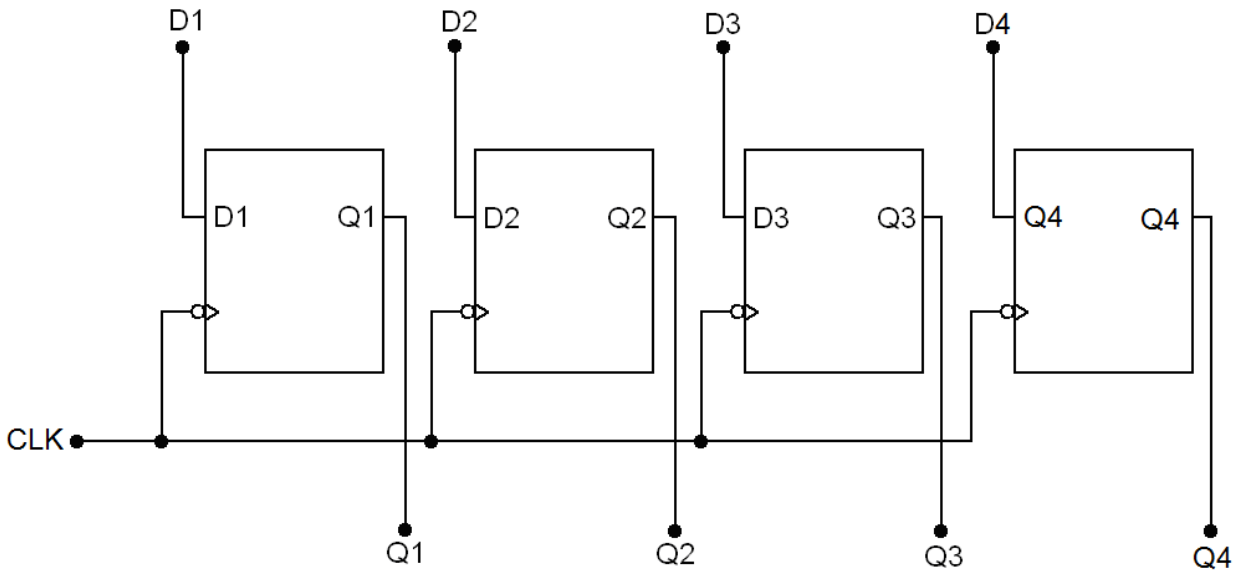


(b)

UNISINOS
DESAFIE O AMANHÃ.

Circuitos Sequenciais

Registrador



UNISINOS
DESAFIE O AMANHÃ.

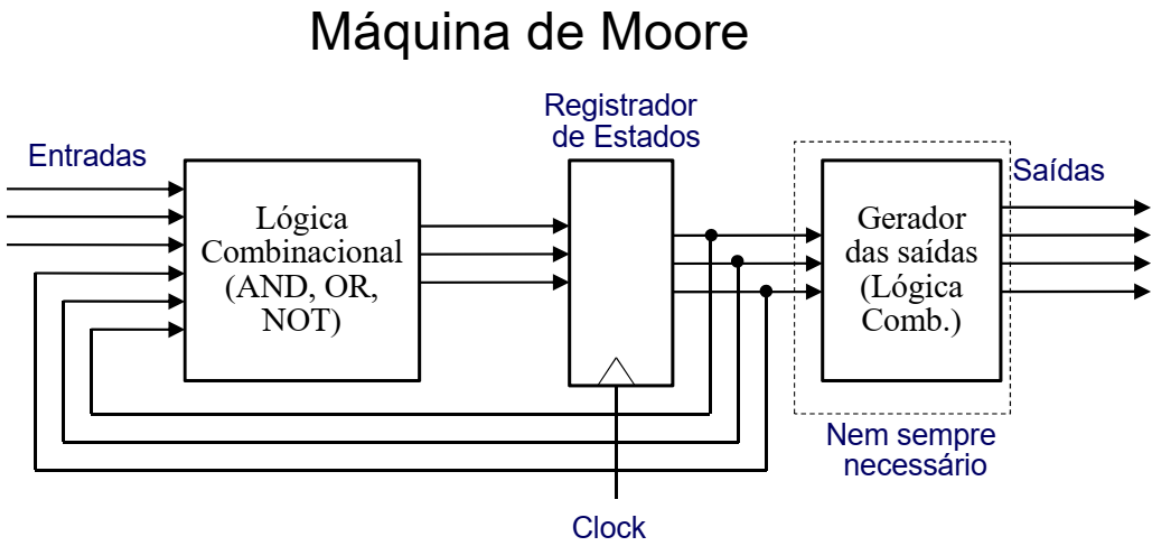
Circuitos Sequenciais

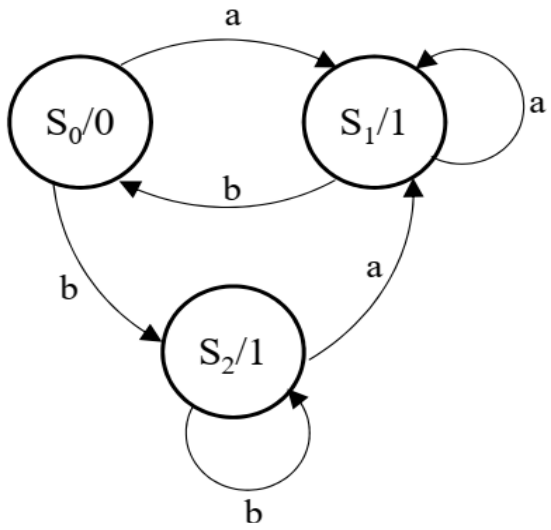
Maquinas de Estados Finitos ou (FSM – Finite State Machine)

Utilizamos o termo máquina de estados quando estamos nos referindo a um circuito sequencial que gere como saída algum tipo de sinal de controle ou de saída para algum sistema;

Podem ser de 2 tipos:
Máquinas de Mealy: A saída depende do estado e da(s) entrada(s);
Máquina de Moore: A saída depende unicamente do estado;

Circuitos Sequenciais



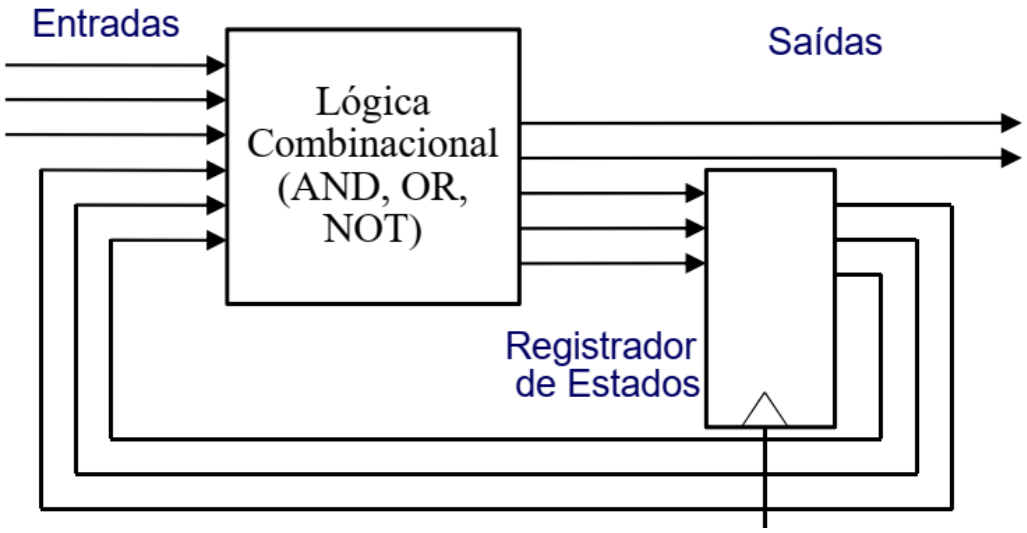


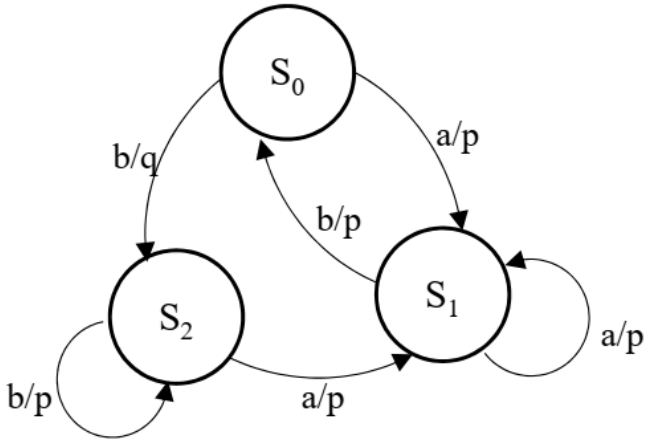
Estados	Entradas	
	a	b
$S_0/0$	S_1	S_2
$S_1/1$	S_1	S_0
$S_2/1$	S_1	S_2

UNISINOS
DESAFIE
O AMANHÃ.

Circuitos Sequenciais

Máquina de Mealy

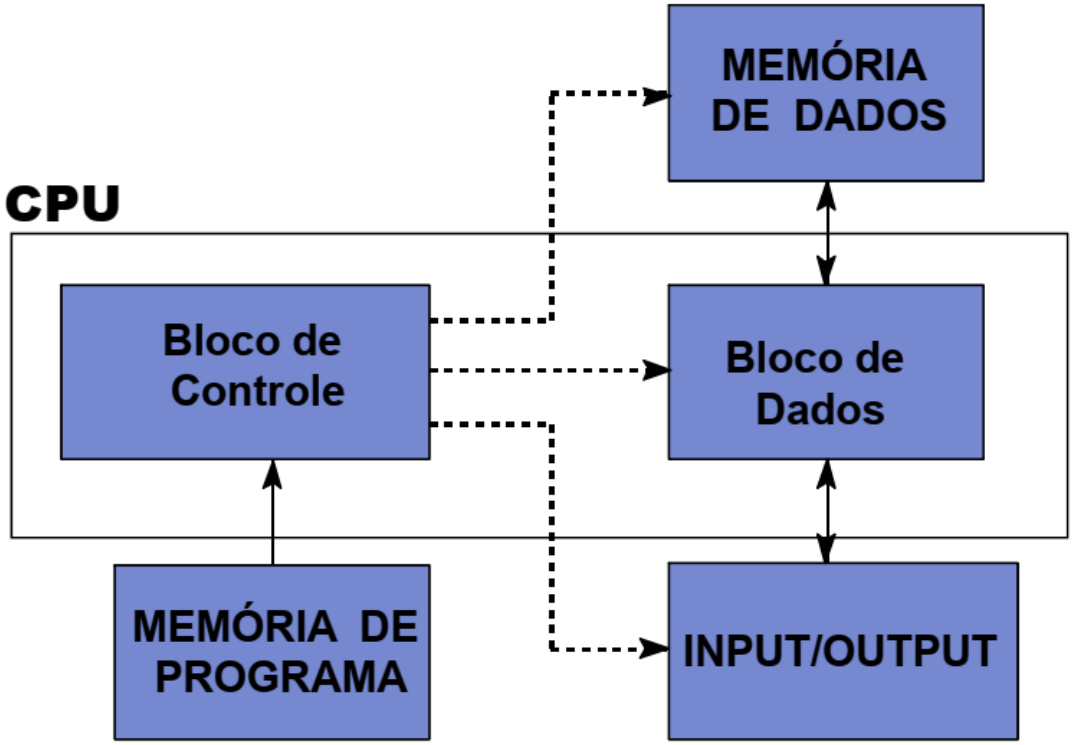




Estados	Entradas	
	a	b
S_0	S_1,p	S_2,q
S_1	S_1,p	S_0,p
S_2	S_1,p	S_2,p

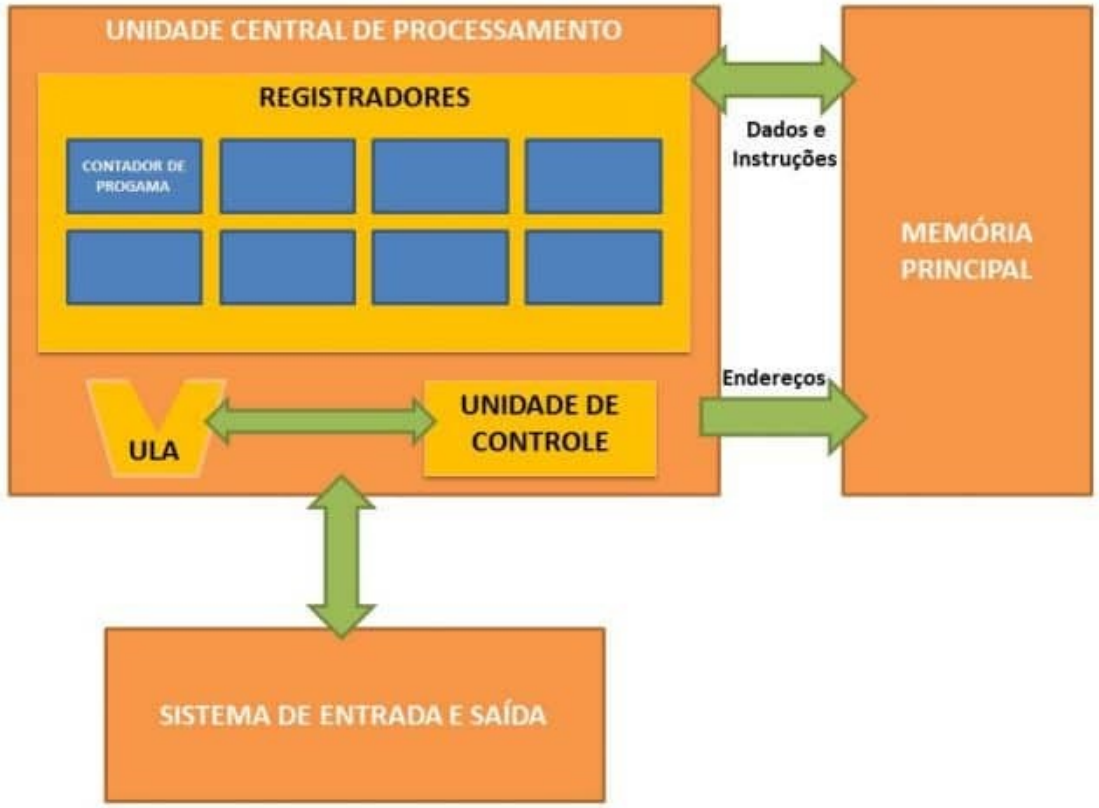
UNISINOS
DESAFIE O AMANHÃ.

Organização de Computadores



Organização de Computadores

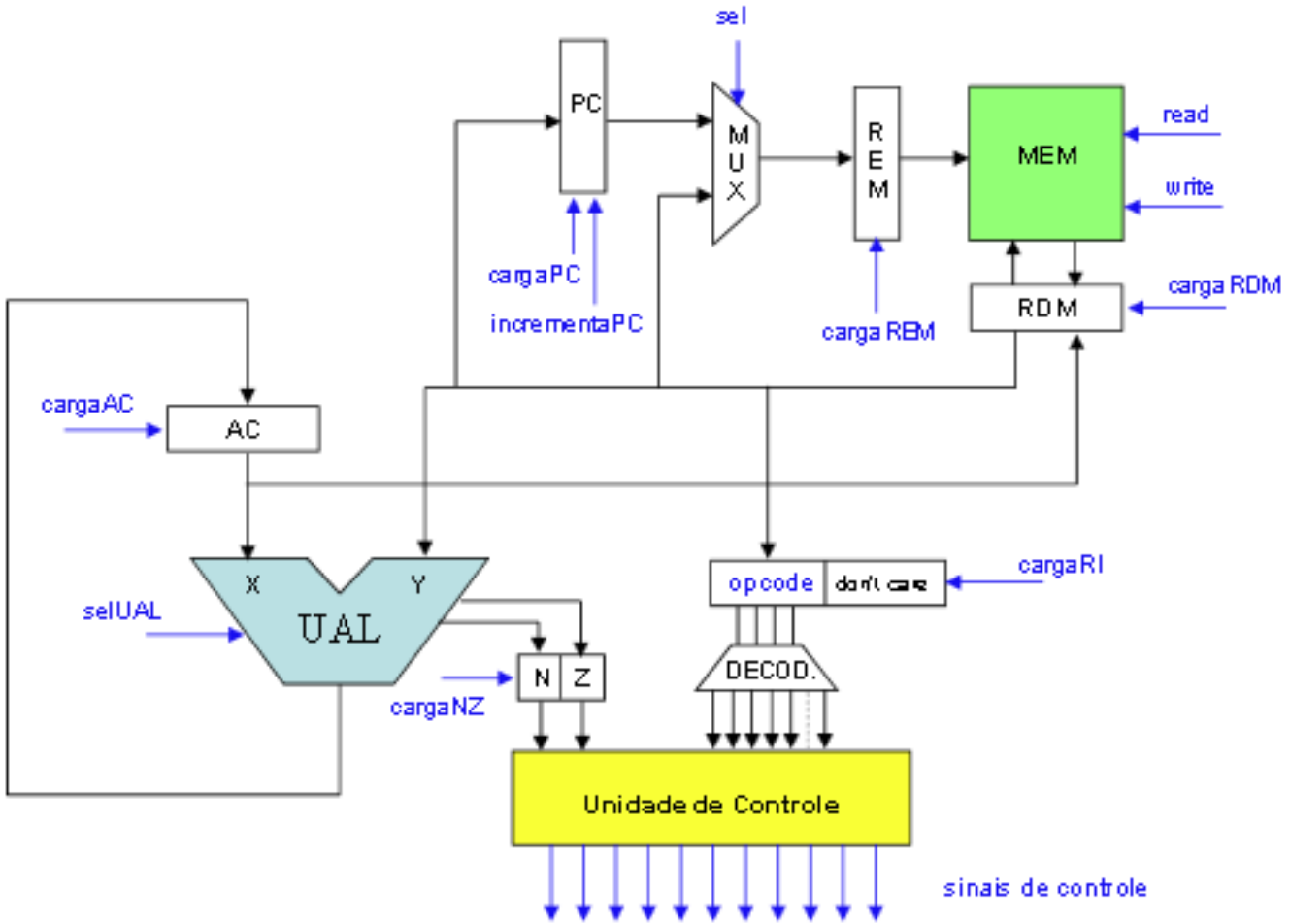
Von Neumann



UNISINOS
DESAFIE O AMANHÃ.

Organização de Computadores

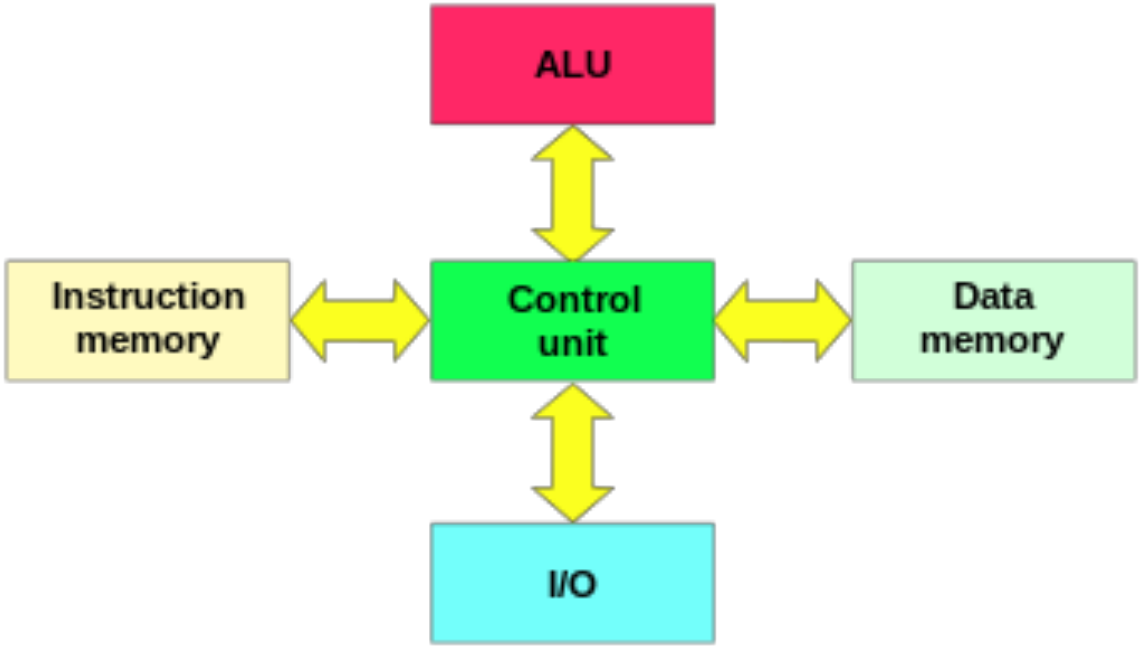
Processador Neander



UNISINOS
DESAFIE O AMANHÃ.

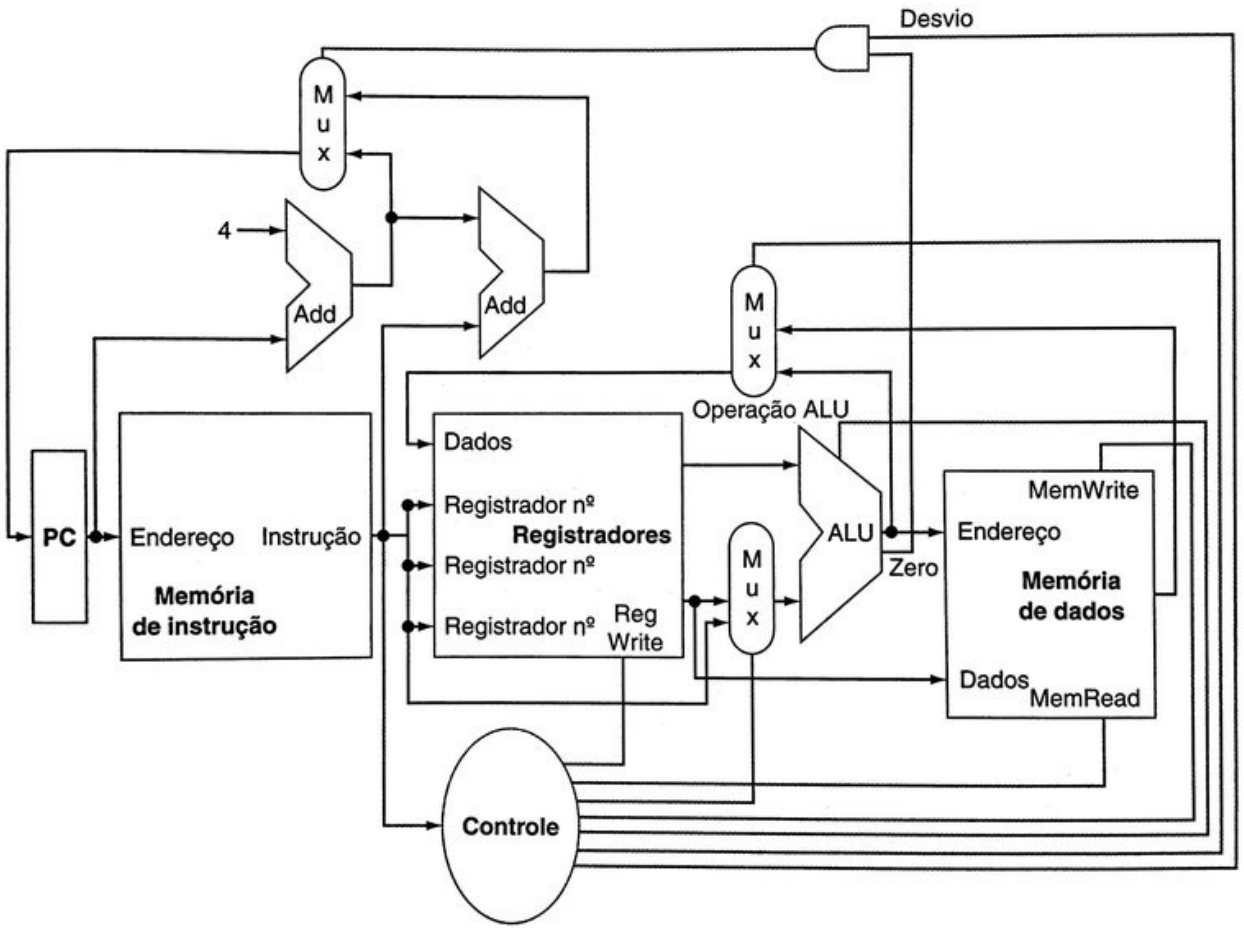
Organização de Computadores

Harvard



Organização de Computadores

Processador MIPS



UNISINOS
DESAFIE
O AMANHÃ.

Arquitetura de Computadores

Estudo de um sistema de computação sob dois pontos de vista:

Arquitetura - se refere aos atributos do sistema visíveis a um programador de linguagem de máquina (instruções, tamanho do dado, endereçamento de memória)

Organização - as unidades operacionais e sua interconexão que realizam a arquitetura, invisíveis ao programador (tipo de memória, forma de implementação da ULA)

Arquitetura de Computadores

Refere-se aos atributos de um sistema visíveis a um programador, com um impacto direto na execução de um programa.

Exemplos de atributos arquiteturais:

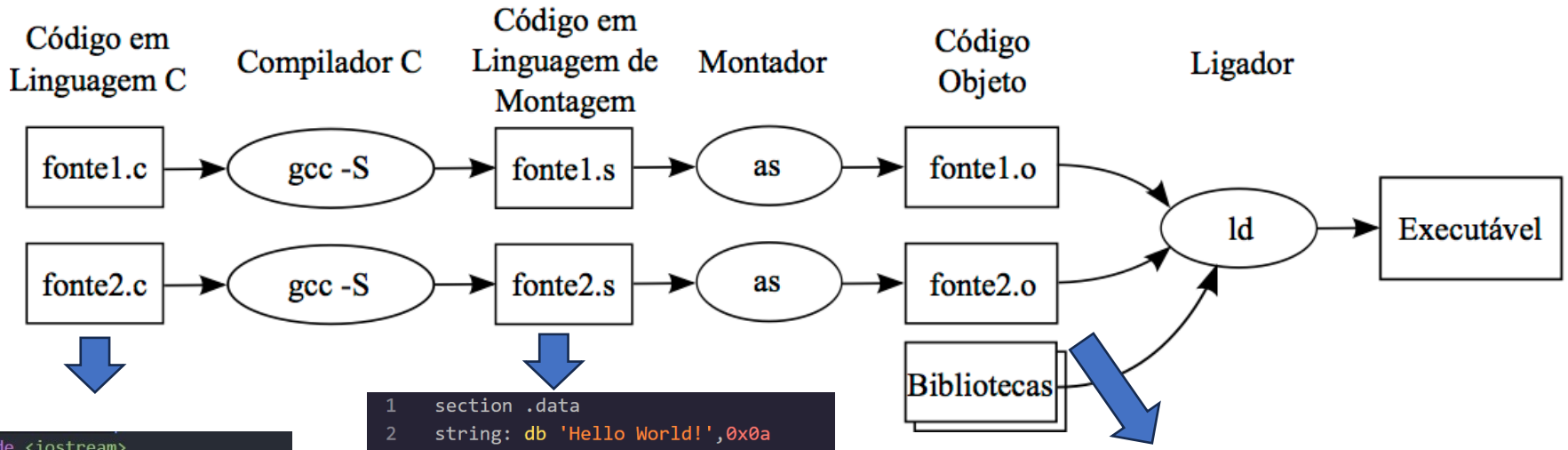
- conjunto de instruções (*instruction set*),
- número de bits usados para representar vários tipos de dados,
- mecanismos de entrada e saída e
- técnicas de endereçamento de memória.

Arquitetura de Computadores

RISC X CISC

RISC	CISC
Instruções simples levando um ciclo	Instruções complexas levando múltiplos ciclos
Apenas LOAD´s e STORE´s referenciam a memória	Qualquer instrução pode referenciar a memória
Altamente pipelined (2 ciclos: busca, executa)	Pouco pipelined
Instruções executadas pelo hardware	Instruções interpretadas pelo microprograma
Instruções de formato fixo	Instruções de vários formatos
Poucas instruções e modos	Muitas instruções e modos
Compilador complexo	Microprograma complexo
Muitos registradores (+/- 500)	Poucos registradores
Menos transistores na pastilha	Mais transistores na pastilha
Projeto mais rápido	Projeto mais lento

Arquitetura de Computadores




```cpp
#include <iostream>

using namespace std;

int main ()
{

cout << "HELLO WORLD";

return 0;
}
```

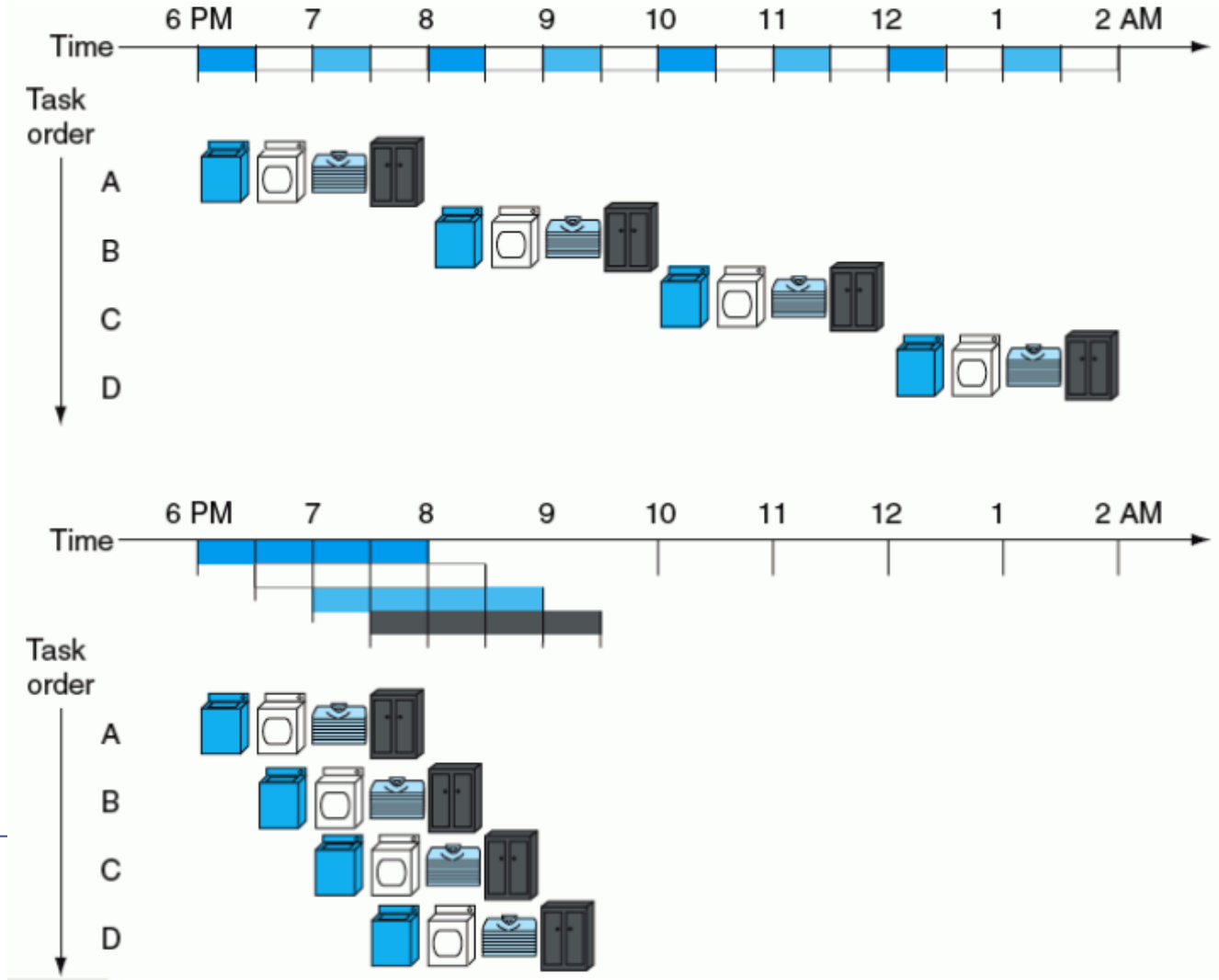
```
section .data
string: db 'Hello World!',0x0a

section .text

global _start
_start:
    mov edx,13
    mov ecx,string
    mov ebx,1
    mov eax,4
    int 0x80
    ;write()
    mov ebx,0
    mov eax,1
    int 0x80
    ;exit()

```

Assembly Language	Machine Code
add $t1, t2, $t3	04CB: 0000 0100 1100 1011
addi $t2, $t3, 60	16BC: 0001 0110 1011 1100
and $t3, $t1, $t2	0299: 0000 0010 1001 1001
andi $t3, $t1, 5	22C5: 0010 0010 1100 0101
beq $t1, $t2, 4	3444: 0011 0100 0100 0100
bne $t1, $t2, 4	4444: 0100 0100 0100 0100
j 0x50	F032: 1111 0000 0011 0010
lw $t1, 16($s1)	5A50: 0101 1010 0101 0000
nop	0005: 0000 0000 0000 0101
nor $t3, $t1, $t2	029E: 0000 0010 1001 1110
or $t3, $t1, $t2	029A: 0000 0010 1001 1010
ori $t3, $t1, 10	62CA: 0110 0010 1100 1010
ssl $t2, $t1, 2	0455: 0000 0100 0101 0101
srl $t2, $t1, 1	0457: 0000 0100 0101 0111
sw $t1, 16($t0)	7050: 0111 0000 0101 0000
sub $t2, $t1, $t0	0214: 0000 0010 0001 0100

UNISINOS
DESAFIE O AMANHÃ.

Arquitetura de Computadores

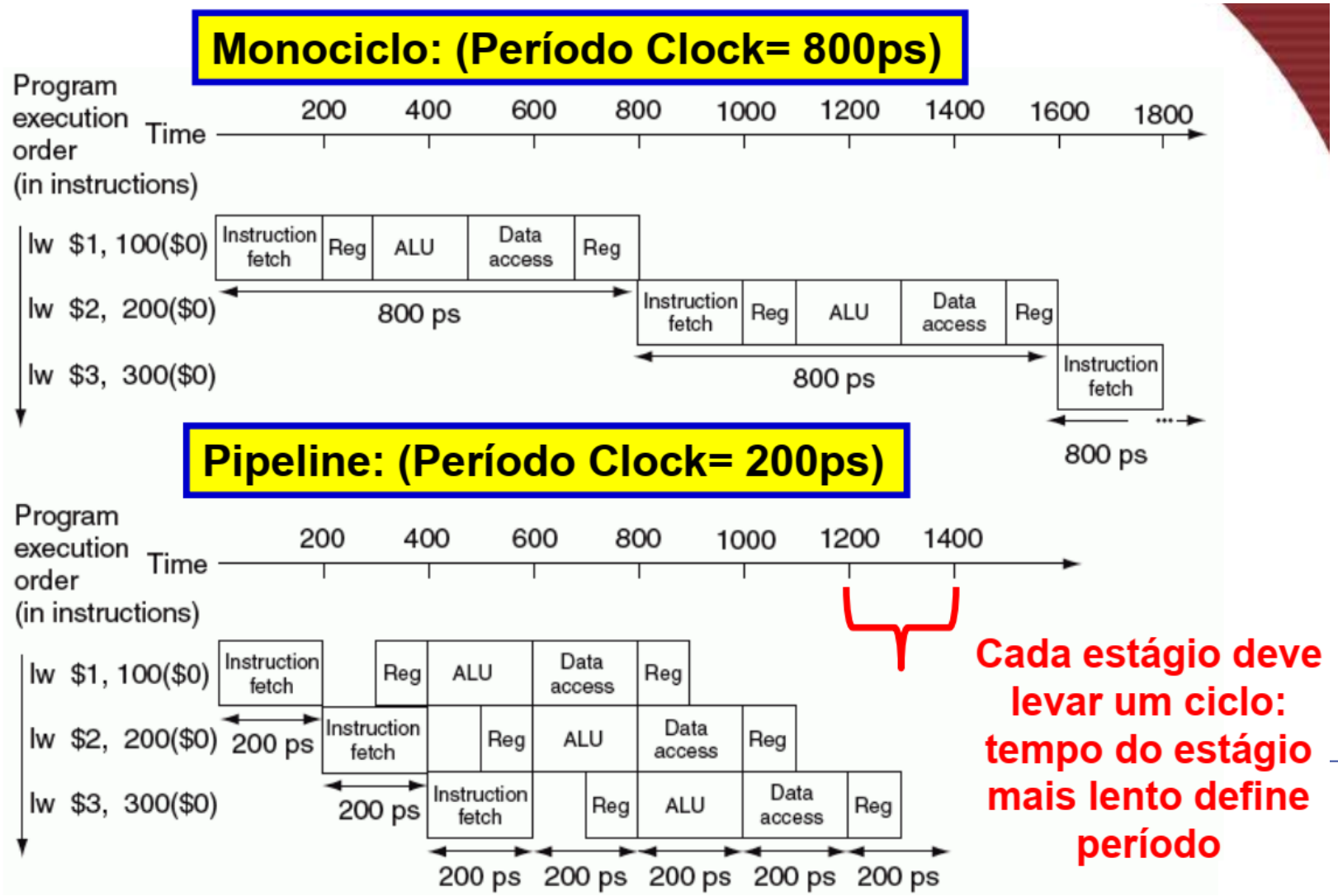
Pipeline



UNISINOS
DESAFIE O AMANHÃ.

Arquitetura de Computadores

Pipeline



UNISINOS
DESAFIE O AMANHÃ.