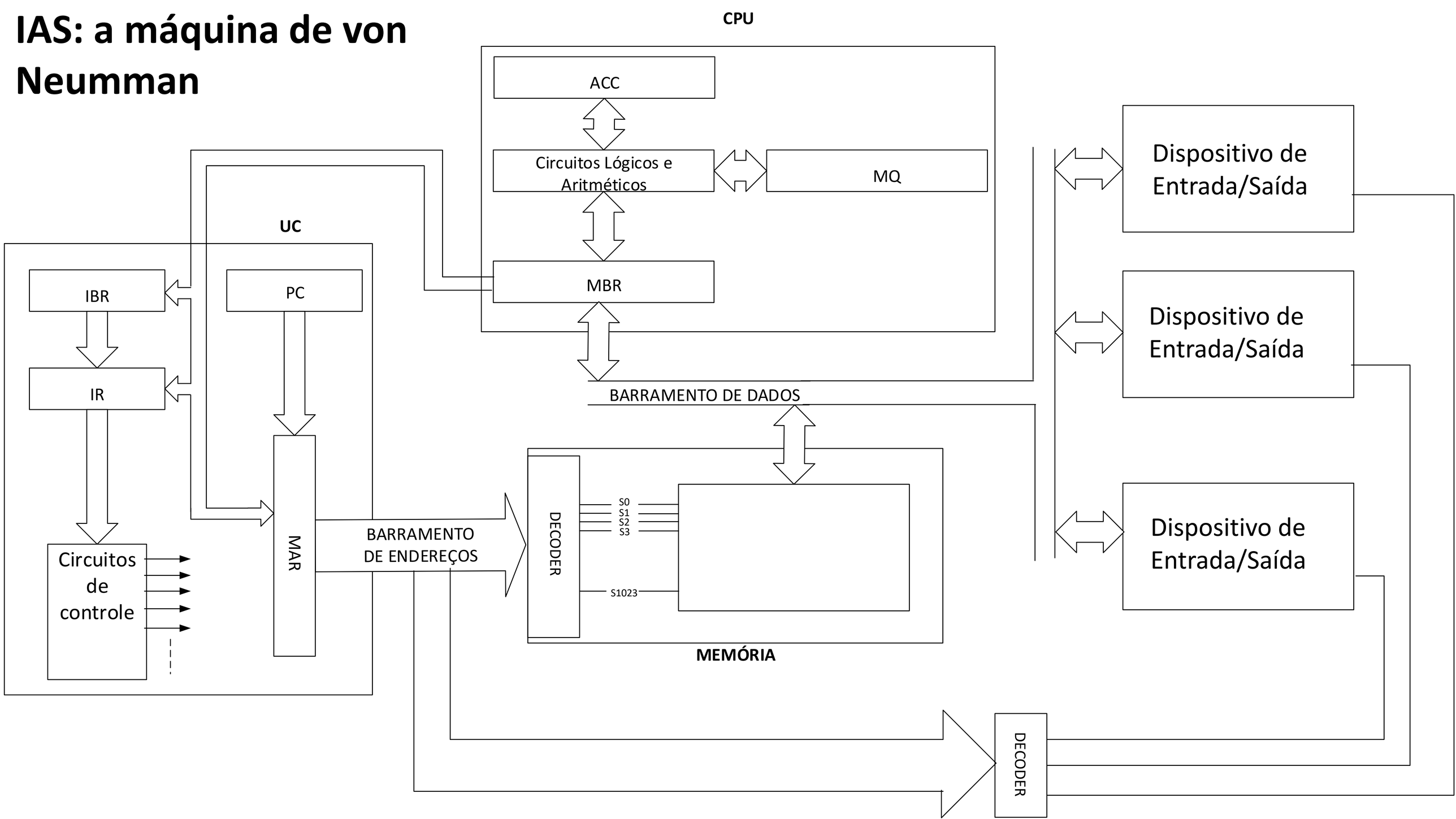
Curso: Engenharia de Computação

Arquitetura de Computadores

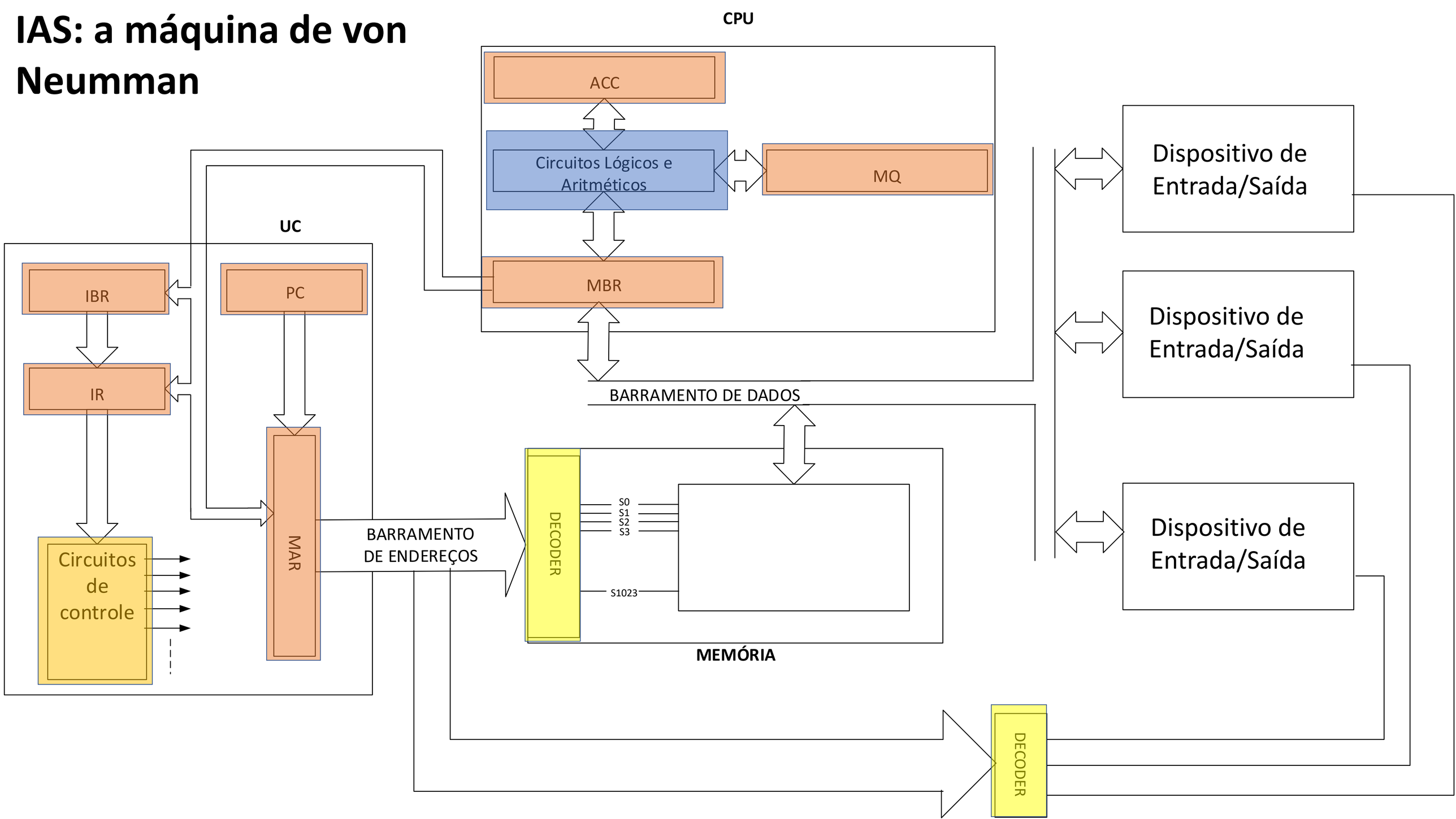
Prof. Clayton J A Silva, MSc
clayton.silva@professores.ibmec.edu.br

ibmec

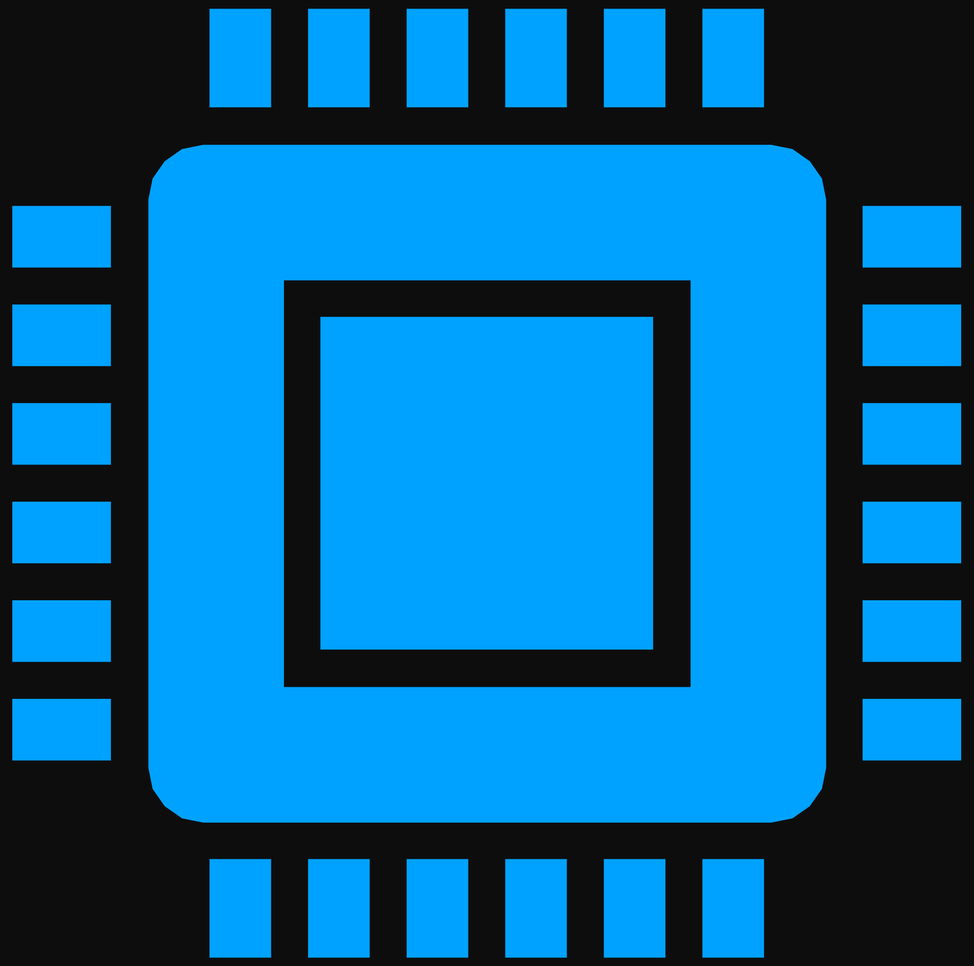
IAS: a máquina de von Neumman



IAS: a máquina de von Neumman

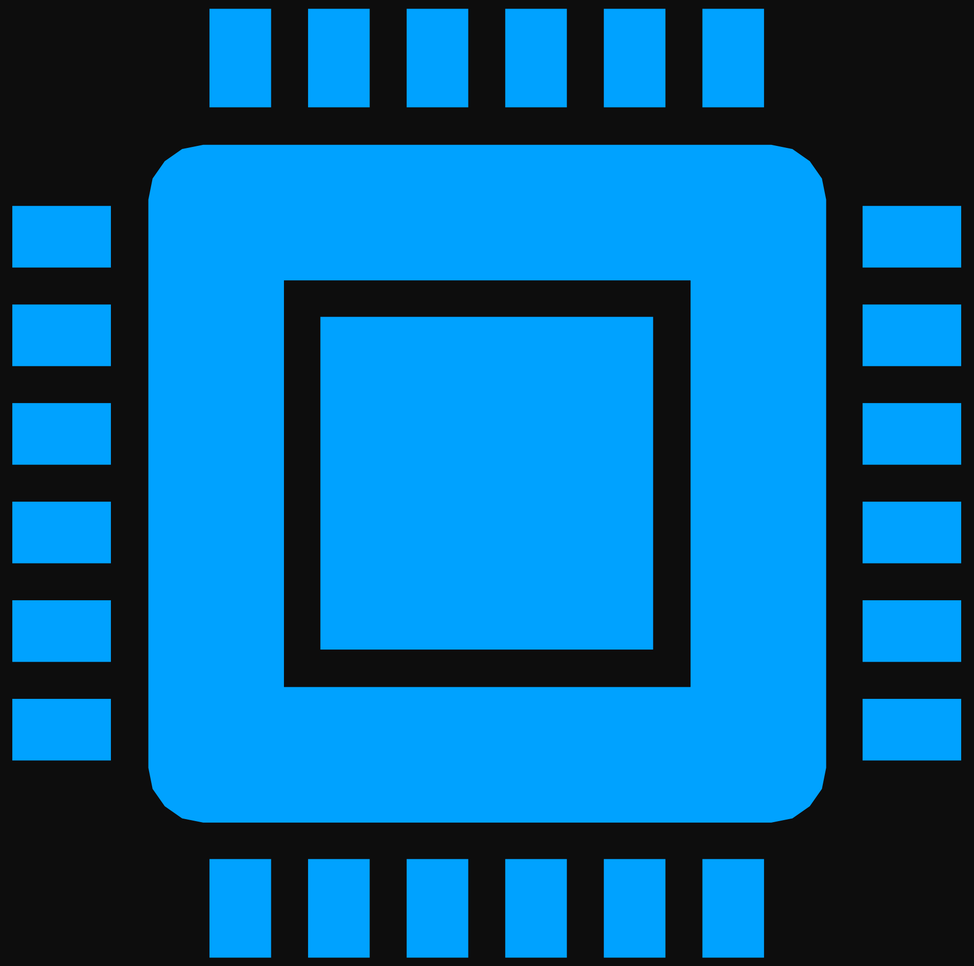


Nível de lógica digital



Circuitos digitais

1. COMBINACIONAIS
2. SEQUENCIAIS



Circuitos digitais

1. **COMBINACIONAIS**
2. SEQUENCIAIS

CIRCUITOS COMBINACIONAIS

- As saídas nas portas de um circuito combinacional em um instante qualquer são o resultado da combinação das suas entradas naquele instante.
- **Não existe memória**!

PORTAS E OPERAÇÕES LÓGICAS

OPERAÇÃO	**OPERADOR**
CONJUNÇÃO	.
DISJUNÇÃO	+
NEGAÇÃO	~

Álgebra booleana

- **CONJUNÇÃO**
  - ✓ Utiliza o operador ***and***, representado por (X*e*Y, *X.Y, X∧Y)*
  - ✓ *A expressão resulta valor lógico 0 quando pelo menos um operando é 0.*

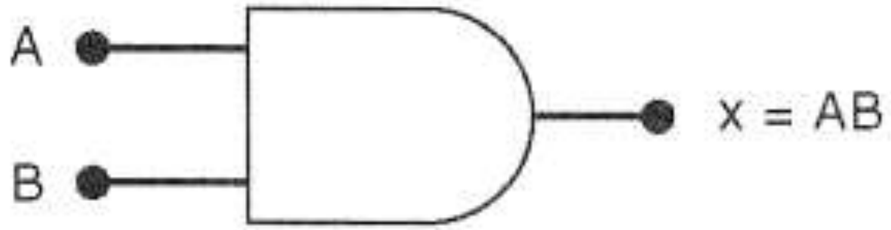
		X		
		0	1	
Y	0	0	0	*X.Y*
	1	0	1	

Porta AND

- Implementa a operação de **conjunção**

AND

A	B	x = A · B
0	0	0
0	1	0
1	0	0
1	1	1



Porta AND

Álgebra booleana

- **DISJUNÇÃO**
  - ✓ Utiliza o operador ***or***, representado por (X*or*Y, *X+Y, X∨Y)*
  - ✓ *A expressão resulta valor lógico 1 quando pelo menos um operando é 1.*

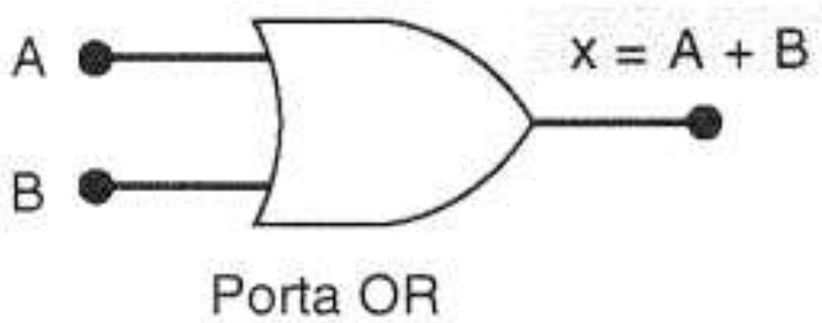
		X		
		0	1	
Y	0	0	1	X+Y
	1	1	1	

Porta OR

- Implementa a operação de **disjunção**

OR

A	B	x = A + B
0	0	0
0	1	1
1	0	1
1	1	1



Porta OR

Álgebra booleana

- **NEGAÇÃO**
  - ✓ Utiliza o operador ***not***, representado por *(~X,$\bar{X}$)*.
  - ✓ *Inverte o valor lógico do operando.*

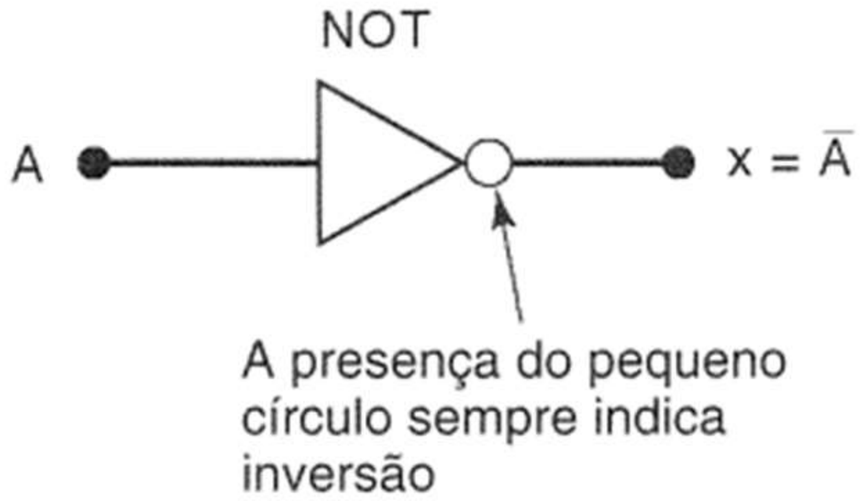
X		
0	1	
1	0	~*X*

INVERSOR

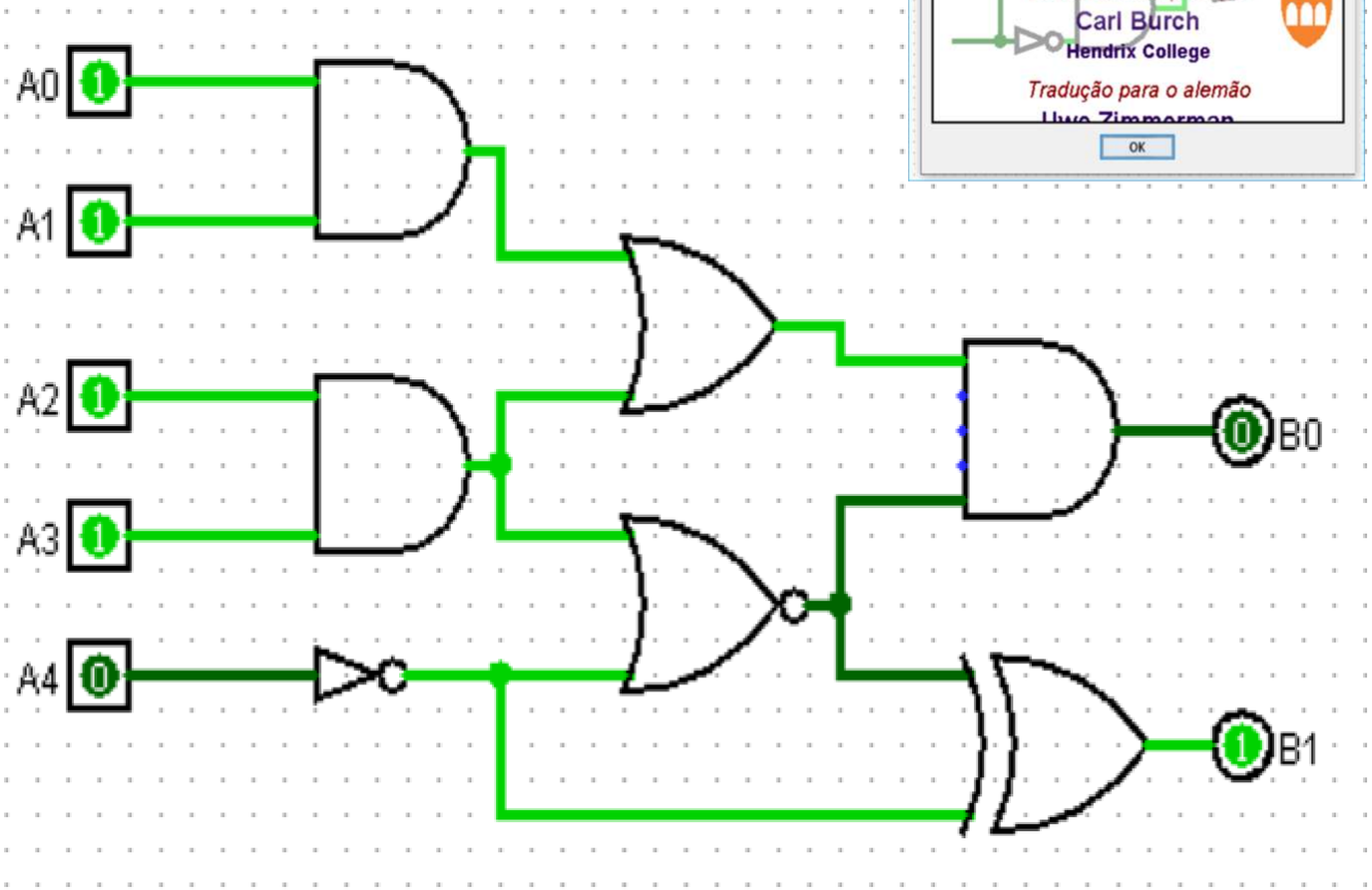
- Implementa a operação de **negação**

NOT

A	$x = \bar{A}$
0	1
1	0



A combinação de portas lógicas, organizadas em **série** e **paralelo**, constitui um **circuito digital**



Sobre circuitos...

- Entradas (A_4, A_3, A_2, A_1, A_0)
- Saídas (B_1, B_0)
- Expressão lógica (*booleana*) representa o circuito
- N entradas poderá gerar 2^N combinações de saída: **para cada saída uma expressão diferente**
- Barramento
- Posição relativa dos bits: **bits mais significativos à esquerda**

Ibmec
PROTAGONISTAS PARA O MUNDO

Expressão booleana do circuito

- **A cada saída → expressão lógica**, associada às operações lógicas das portas
- **Várias** expressões lógicas **equivalentes**
- Forma **canônica: disjunção** (operação ‘or’) dos termos da tabela-verdade que resultam valor lógico ‘1’ (**mintermos**)

Ibmec
PROTAGONISTAS PARA O MUNDO

Tabela-verdade

- Propicia **analisar** o comportamento do circuito digital para todas as possíveis combinações de entradas
- Possui 2^N **linhas**, tal que ***N* é o número de entradas**
- Possui ***M* colunas**, tal que $M=N+S$, ***S* é o número de saídas**
- Cada saída corresponde a uma coluna

Ibmec
PROTAGONISTAS PARA O MUNDO

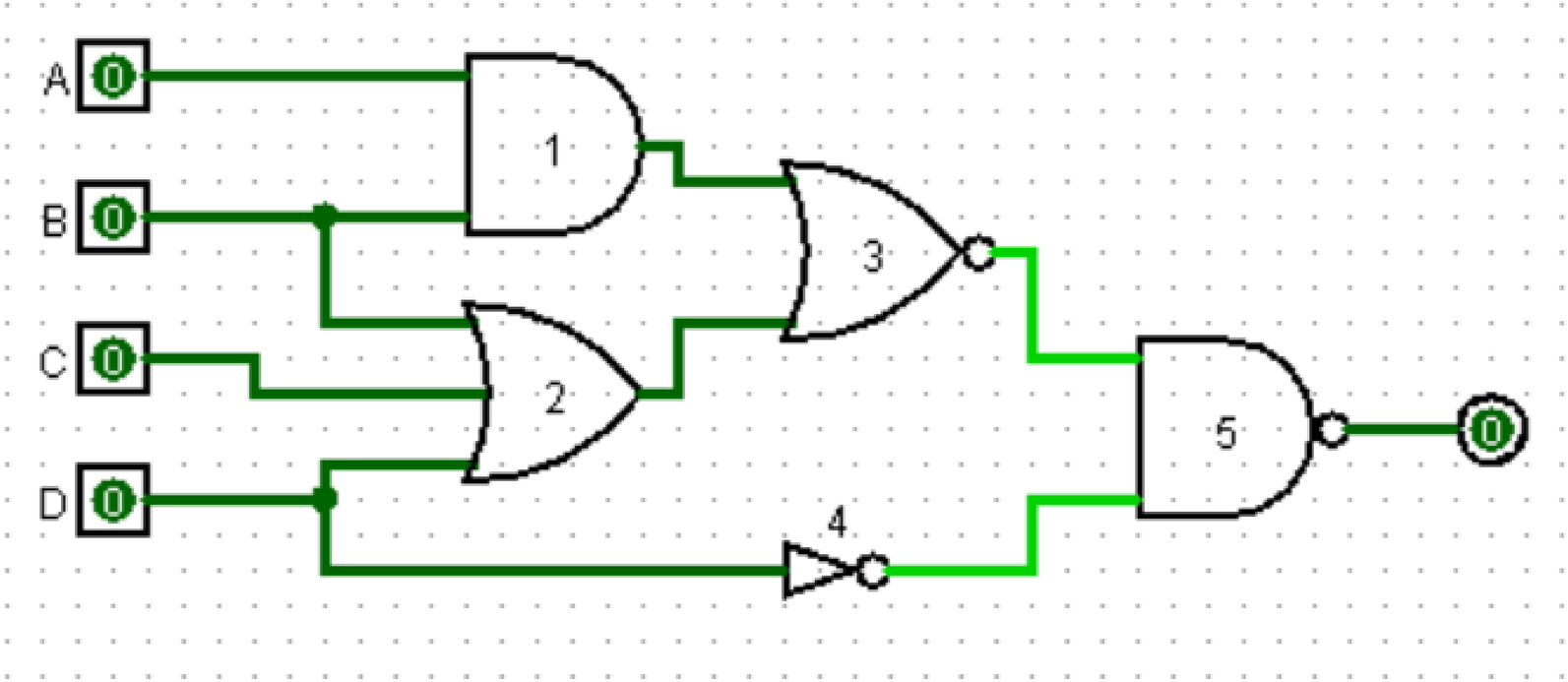
Tabela-verdade

Linhas	Entradas					Saídas	
	A0	*A1*	*A2*	*A3*	*A4*	*B0*	*B1*
1	0	0	0	0	0	0	1
2	0	0	0	0	1	0	1
3	0	0	0	1	0	0	1
4	0	0	0	1	1	0	1
5	0	0	1	0	0	0	1
6	0	0	1	0	1	0	1
...	1	0	1	1	1	0	0
25	1	1	0	0	0	0	1
26	1	1	0	0	1	1	1
27	1	1	0	1	0	0	1
28	1	1	0	1	1	1	1
29	1	1	1	0	0	0	1
30	1	1	1	0	1	1	1
31	1	1	1	1	0	0	1
32	1	1	1	1	1	0	0

Análise de circuitos digitais combinacionais

1. Enumerar as portas
2. Identificar a expressão booleana na saída de cada porta
3. Identificar a expressão booleana na saída do circuito
4. Montar a tabela-verdade relativa ao circuito

Análise de circuitos digitais combinacionais



Projeto de circuitos digitais combinacionais

1. Montar uma tabela-verdade referente ao circuito para cada saída
2. Identificar a expressão booleana na forma canônica para cada saída
3. Minimizar a expressão booleana
4. Implementar o circuito

Circuitos combinacionais típicos

Circuitos combinacionais

Aritméticos – somadores, subtratores, multiplicadores

Lógicos – Comparadores

Multiplexadores/Demultiplexadores

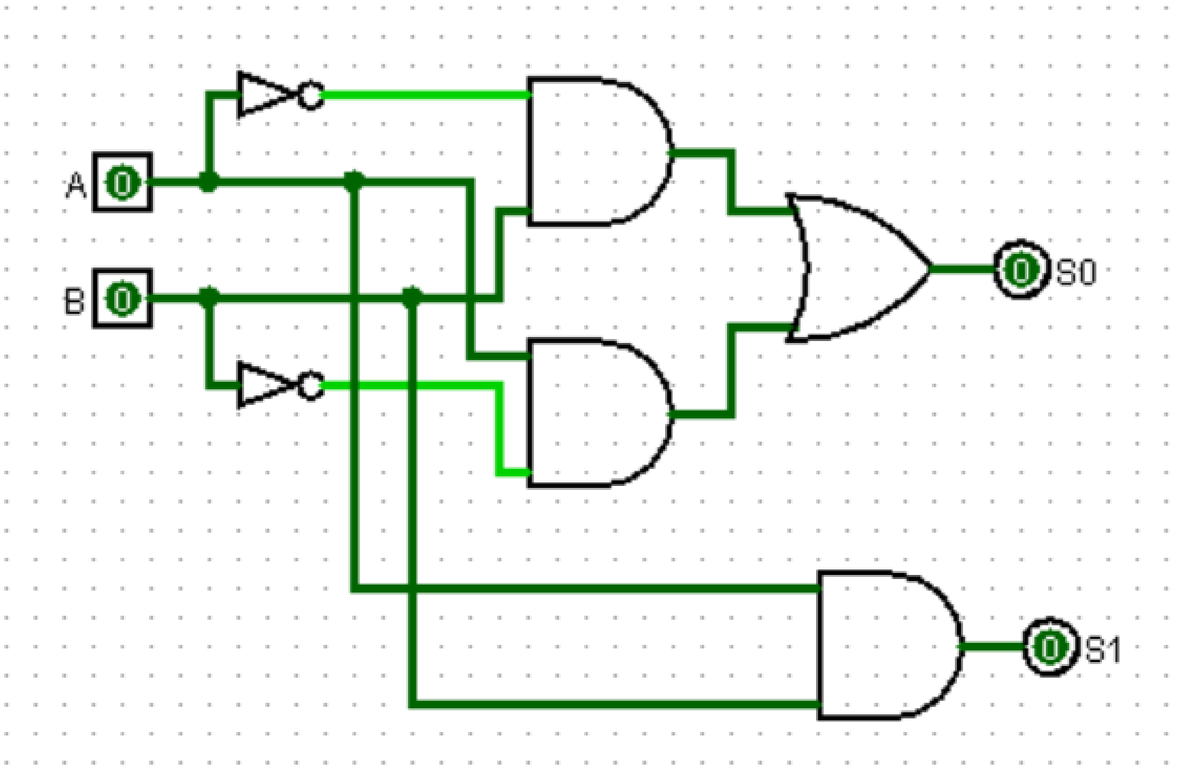
Codificadores/Decodificadores

Somadores

1. Entrada ***A***$(A_{N-1}, A_{N-2}, \ldots, A_0)$, ***B***$(B_{N-1}, B_{N-2}, \ldots, B_0)$, com *N* bits cada
2. Saída representa a soma binária de N bits
3. Implementa bit a bit a soma binária, transportando *carry*

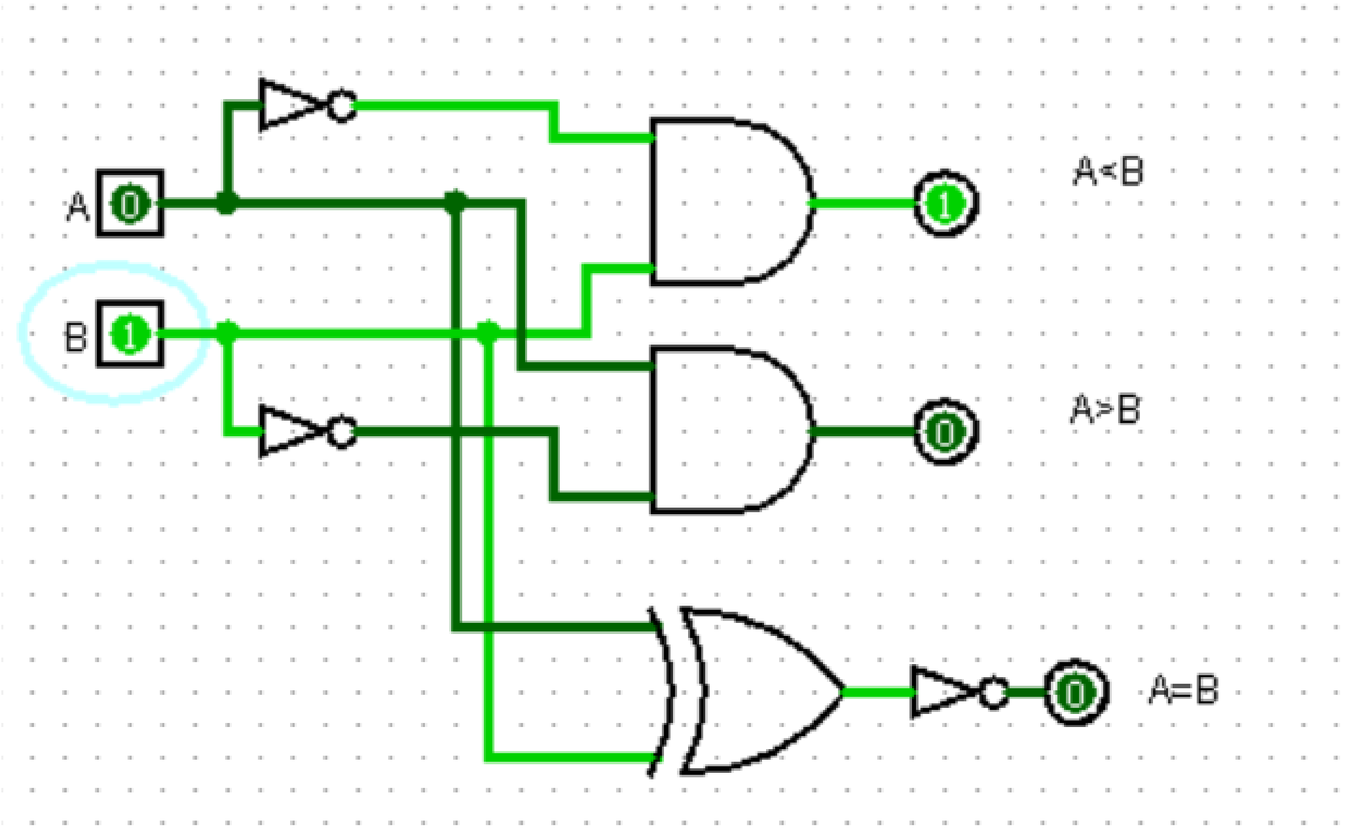
Somador de 1-bit

		SOMA	
A	***B***	***S1***	***S0***
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



Comparador de 1-bit

A	*B*	*A<B*	*A=B*	*A>B*
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0



Multiplexadores

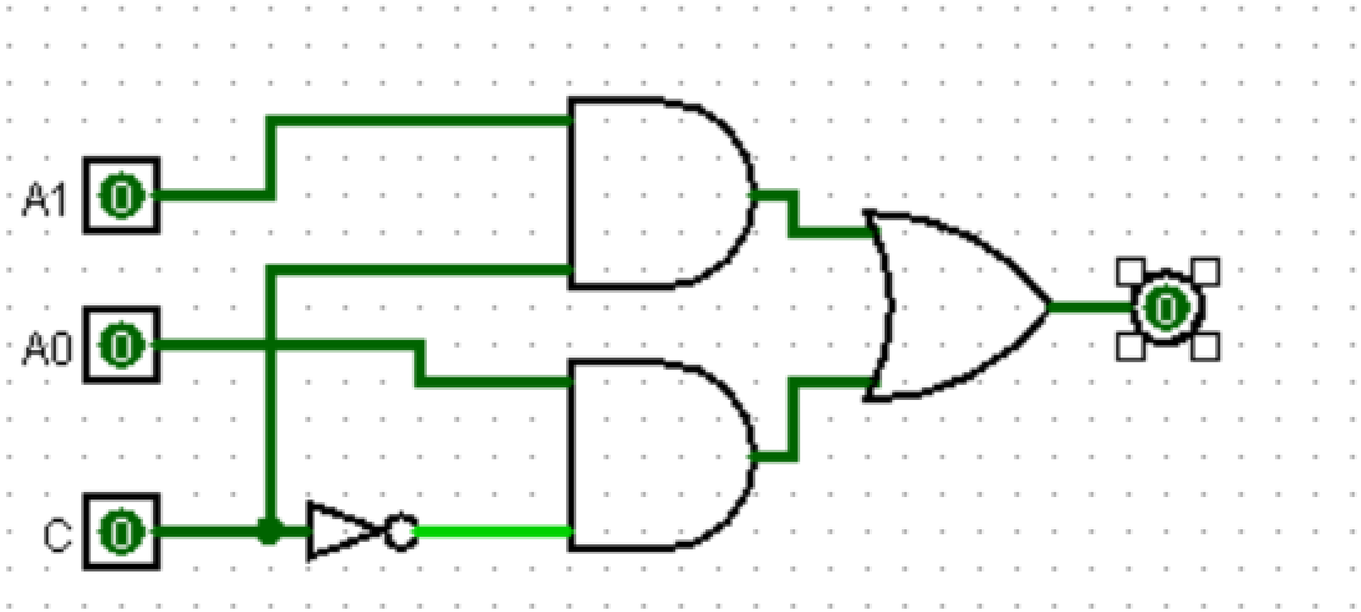
1. Entrada $\boldsymbol{A}(A_{N-1}, A_{N-2}, \ldots, A_0)$, com N bits
2. Saída com 1 bit
3. Entrada seletora de controle $\boldsymbol{C}(C_{M-1}, C_{M-2}, \ldots, C_0)$ com M bits
4. $2^M \geq N$
5. A saída reflete uma das N entradas de acordo com C

Multiplexador – 2-bits

A1	*A0*	*C*	*S*
0	0	0	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Multiplexador – 2-bits

A1	*A0*	*C*	*S*
0	0	0	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Coder

1. Entrada $\boldsymbol{A}(A_{M-1}, A_{N-2}, \ldots, A_0)$, com M bits
2. Saída $\boldsymbol{S}(S_{N-1}, S_{N-2}, \ldots, S_0)$, com N bits
3. A saída é a palavra binária de um código das M entradas
4. $2^N \geq M$

IBMEC.BR

/IBMEC

IBMEC

@IBMEC_OFICIAL

@IBMEC

Ibmec