

# Portas lógicas e circuitos digitais

## **Sistemas da Computação**

Prof. Rossano Pablo Pinto, MSc.

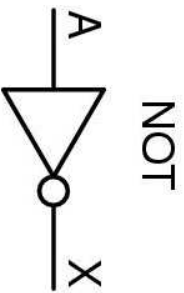
rossano at gmail com

2 semestre 2007

# Tópicos

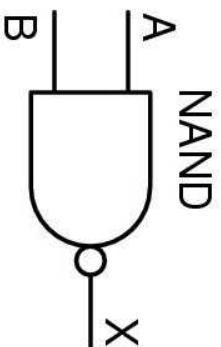
- Portas
- Circuito somador
- Circuito subtrator
- flip-flops (registradores)
- ULA de 1 bit
- Memória

# Portas Lógicas Básicas



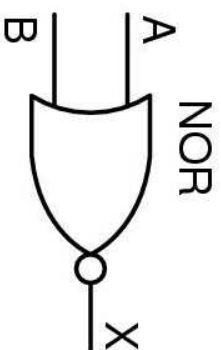
A	X
0	1
1	0

(a)



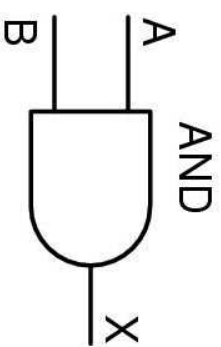
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

(b)



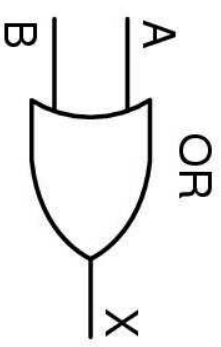
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

(c)



A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

(d)

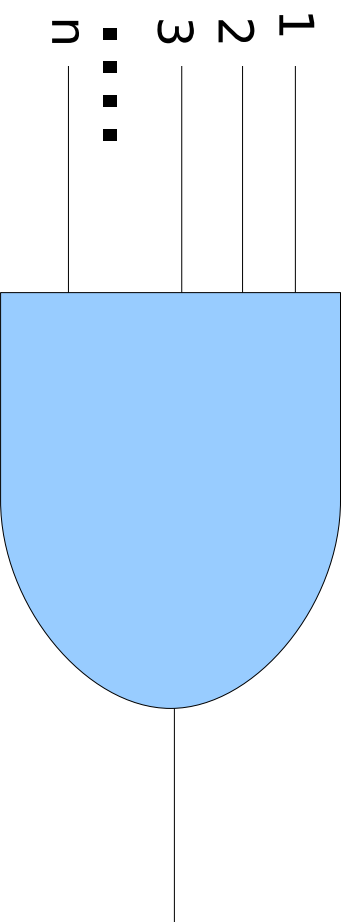


A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

(e)

# Portas Lógicas

- Características
  - As estradas não estão limitadas a 2. Podem ter quantas entradas forem necessárias.
  - A saída é sempre única



# Portas Lógicas

- Características
  - Os circuitos podem ser construídos pela combinação das portas lógicas
  - Qualquer expressão booleana pode ser representada pela combinação de portas lógicas

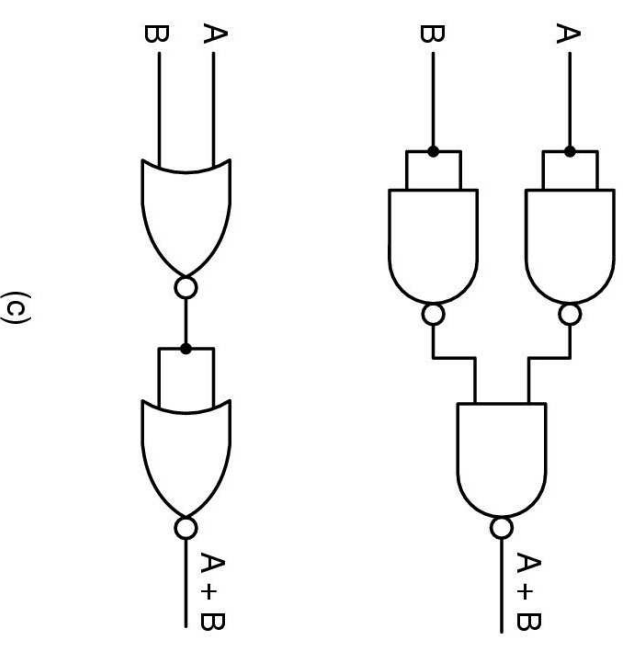
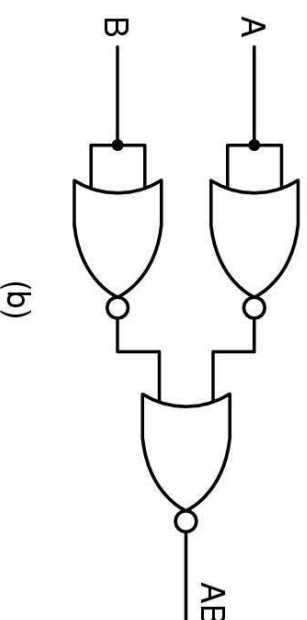
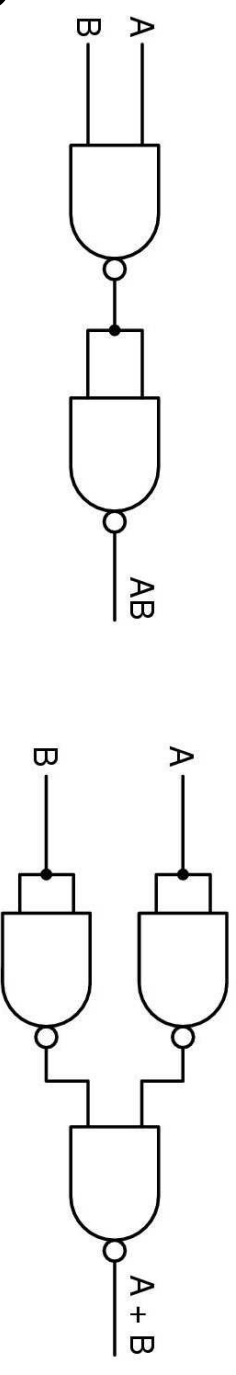
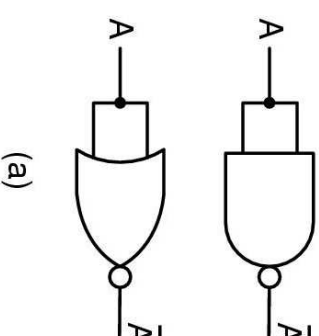
# Portas Lógicas

- **Equivalência de circuitos**
  - Um mesmo circuito pode ser construído a partir de combinações de portas diferentes

# Portas Lógicas

- (a) NOT
- (b) AND
- (c) OR

construído a partir de portas NAND ou NOR

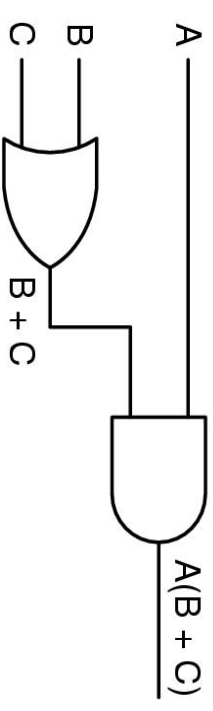
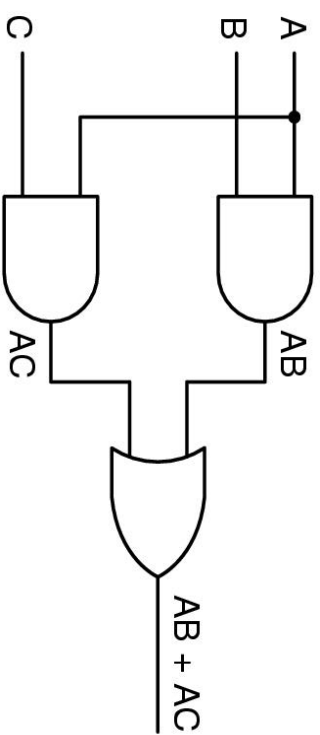


(c)

# Portas Lógicas

- 2 funções equivalentes: (a)  $AB + AC$  (b)  $A(B+C)$

$$A(B+C)$$



A	B	C	AB	AC	AB + AC
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

A	B	C	A	B + C	A(B + C)
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

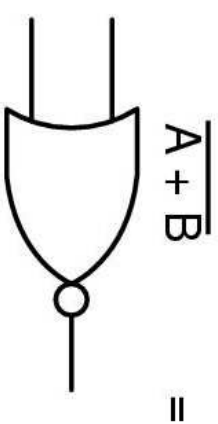
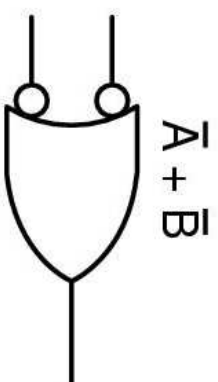
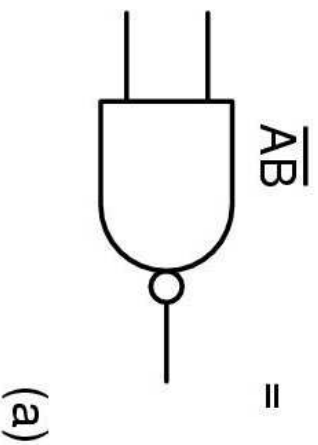
(a)

(b)

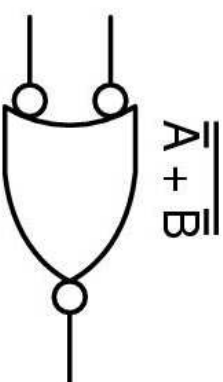
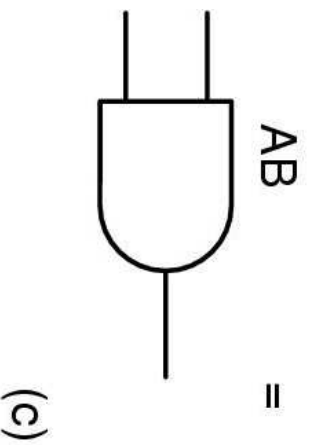
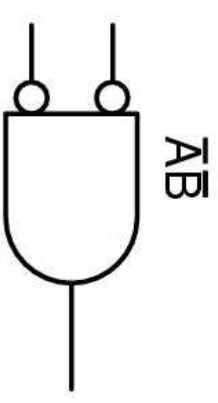


# Portas Lógicas

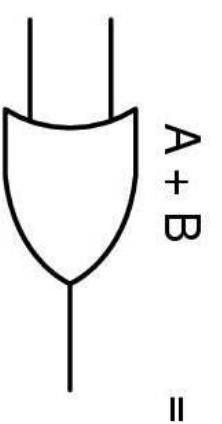
- Símbolos alternativos: (a) NAND, (b) NOR, (c) AND, (d) OR.



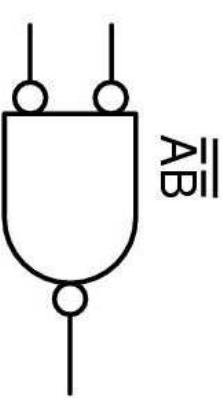
(b)



(c)



(d)

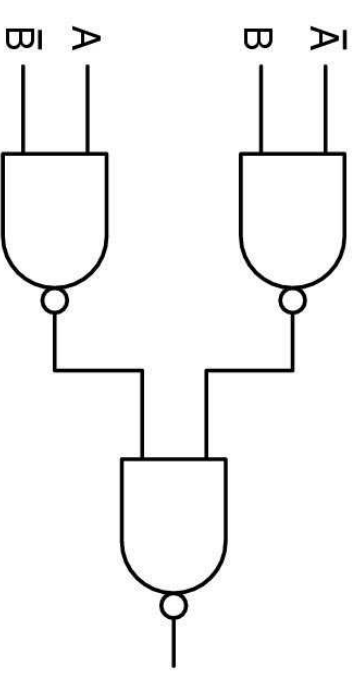
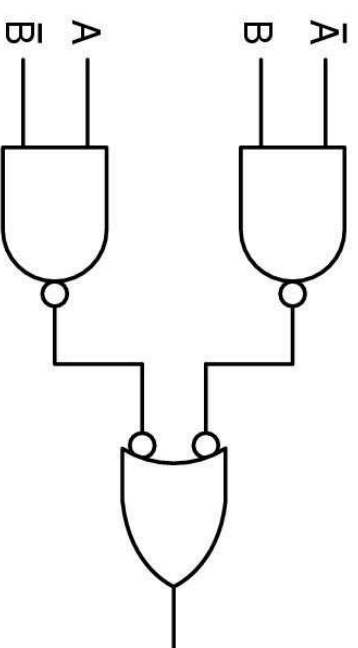
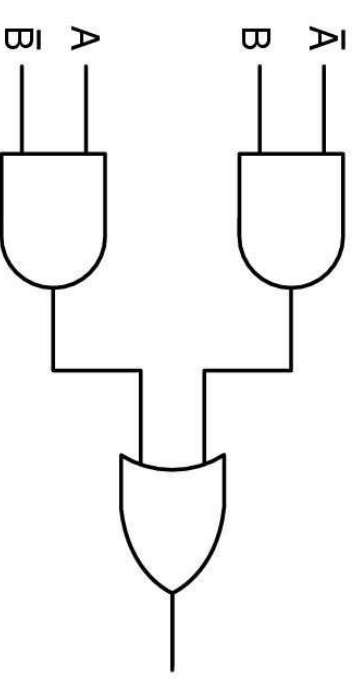


# Portas Lógicas

- Porta XOR (ou exclusivo) – 3 circuitos diferentes p/ calcular um XOR

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

(a)

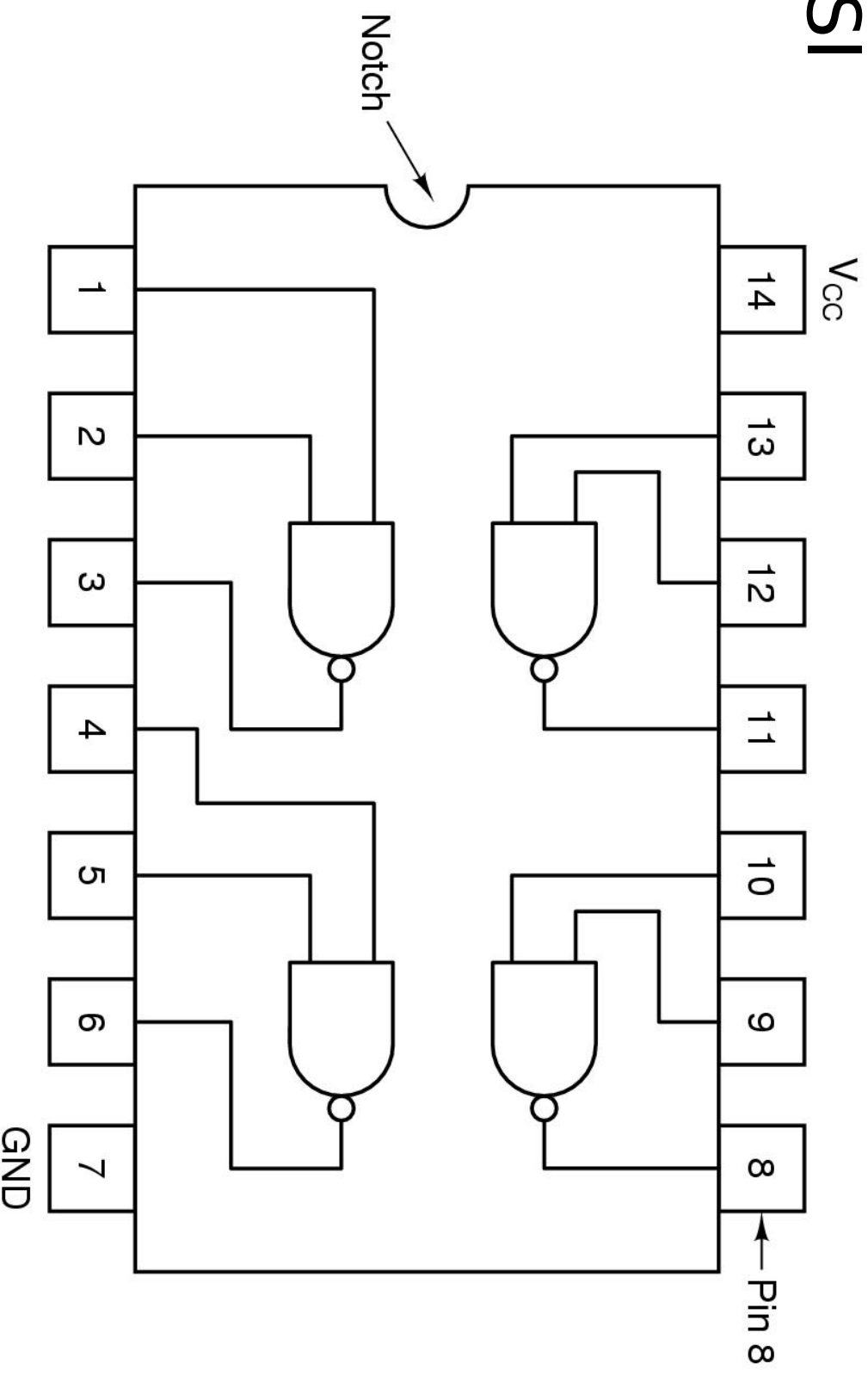


# Circuitos Integrados

- As portas não são vendidas individualmente, mas em unidades chamadas **Circuitos Integrados**:
  - SSI (Small Scale Integrated): 1 à 10 portas
  - MSI (Medium Scale Integrated): 10 à 100 portas
  - LSI (Large Scale Integrated): 100 à 100.000 portas
  - VLSI (Very Large Scale Integrated): > 100.000 portas

# Circuitos Integrados

- Chip SSI com 4 portas

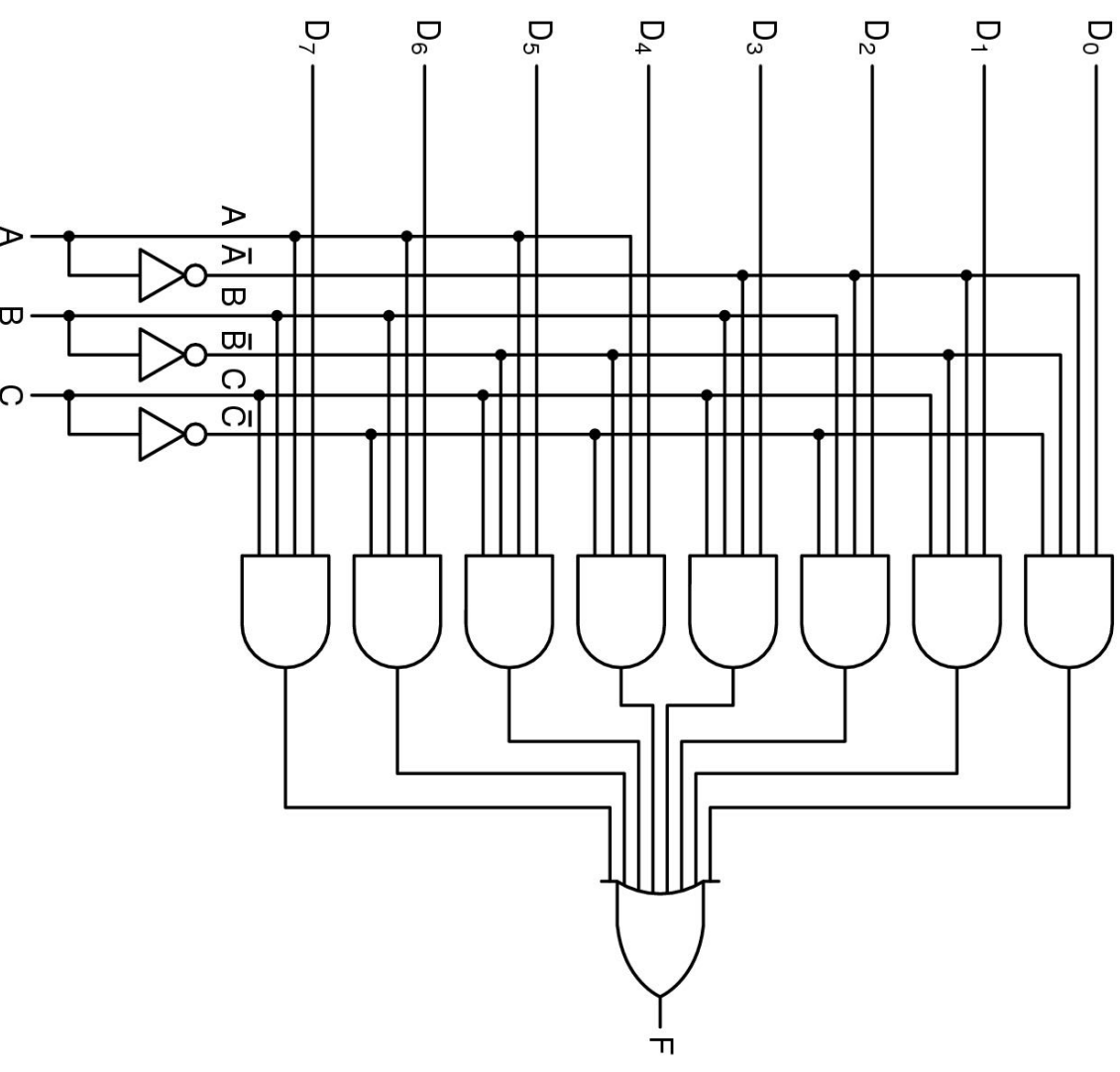


# Circuitos Combinatórios

- Multiplexador: seleciona uma das várias entradas e gera a saída
- Demultiplexador: seleciona uma dentre várias saídas

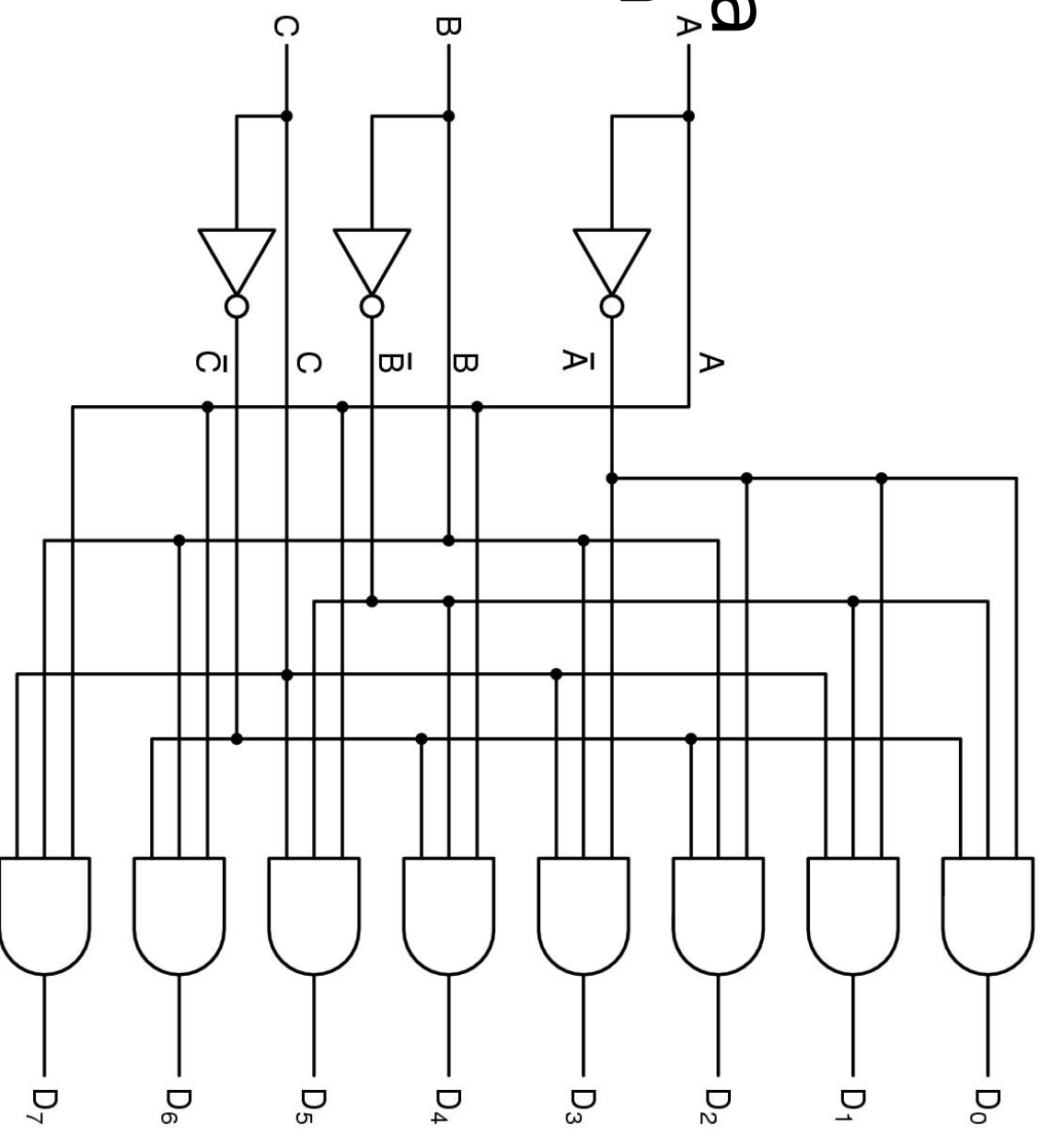
# Circuitos Combinatórios

- Exemplo de multiplexador com 8 entradas e uma saída



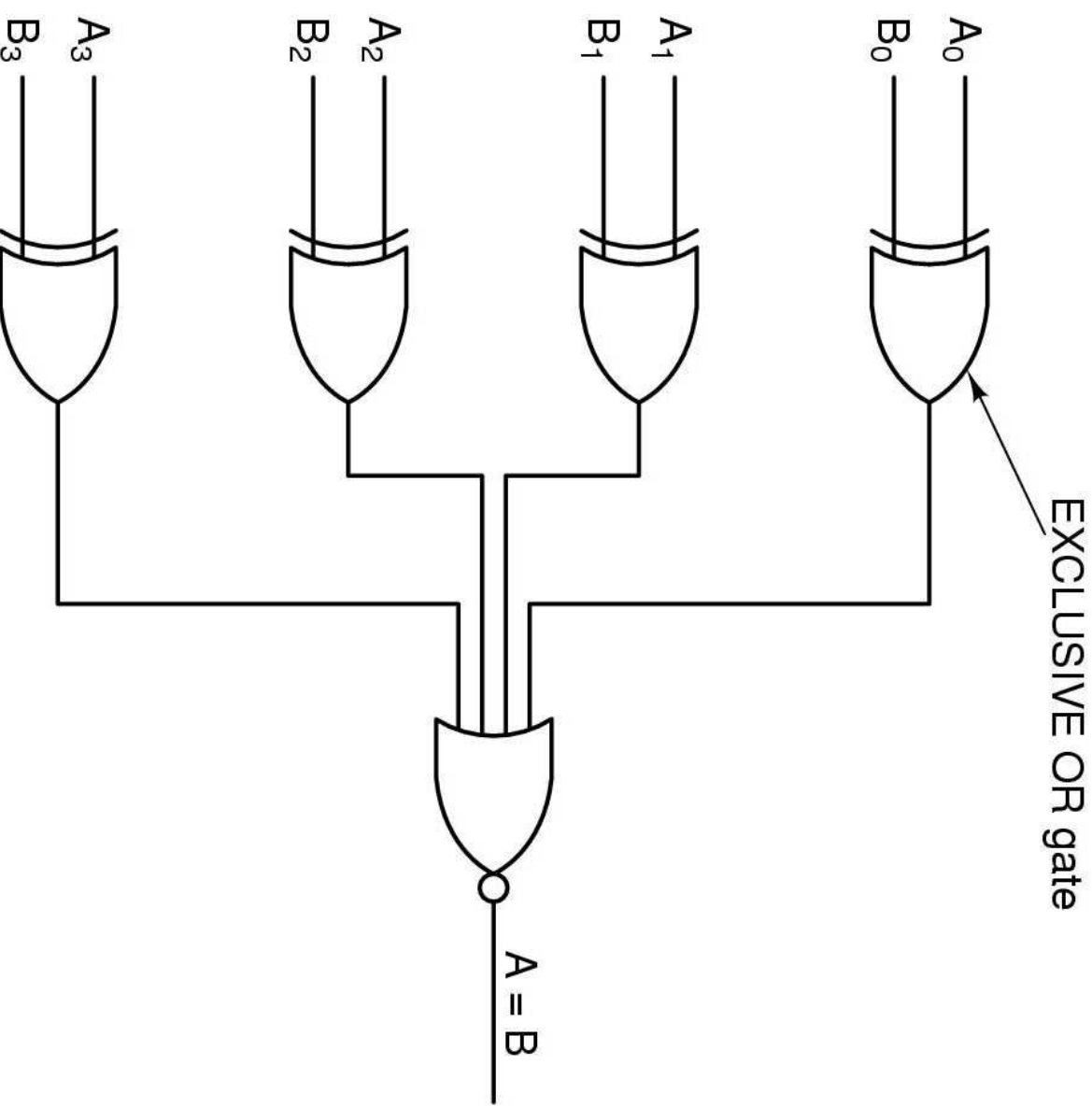
# Circuitos Combinatórios

- Decodificador:  
número de  $n$ -  
bits na entrada  
seleciona uma  
das  $2^n$  linhas  
de saída



# Circuitos Combinatórios

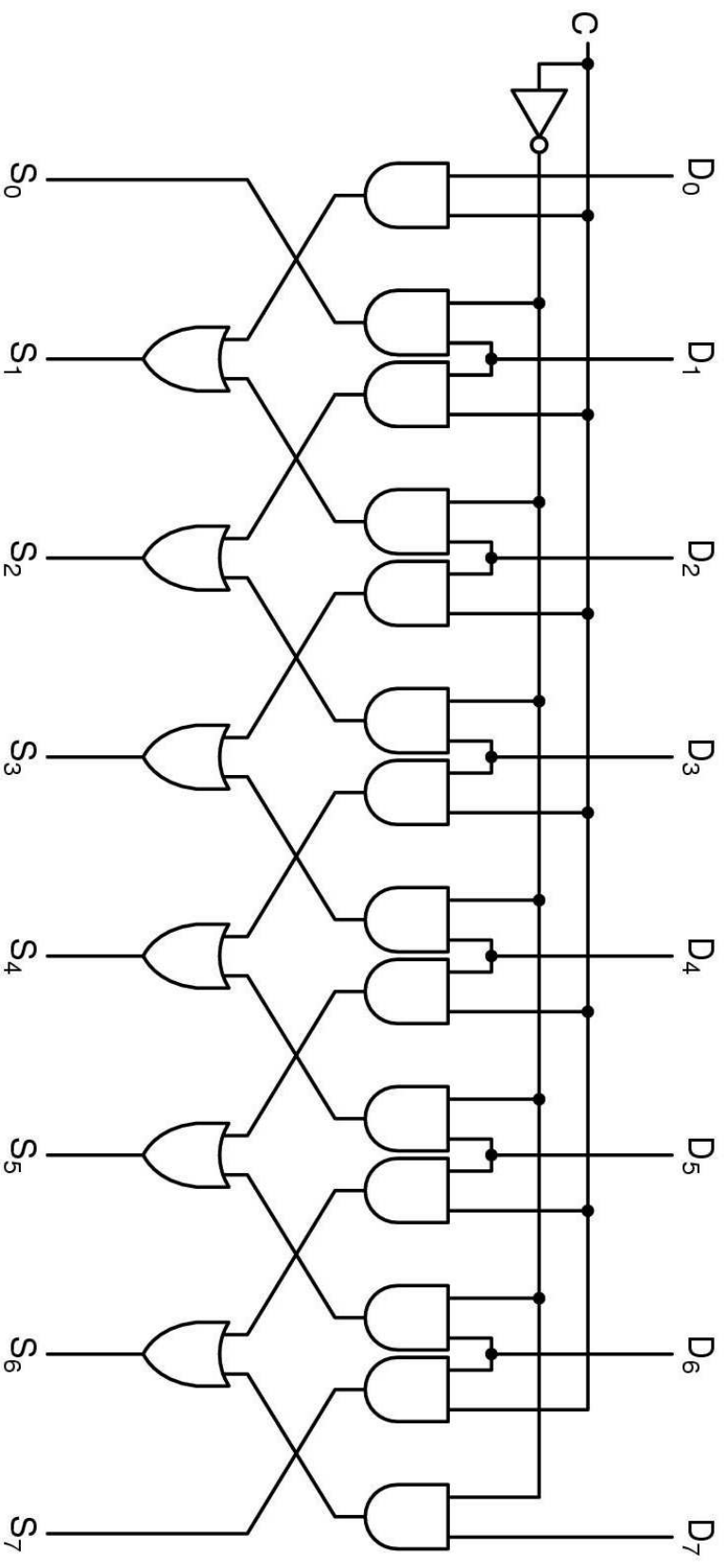
- Comparador: compara 2 palavras de entrada. EX. verifica se são iguais





# Circuitos Aritméticos

- Shifters: desloca os bits para esquerda ou direita



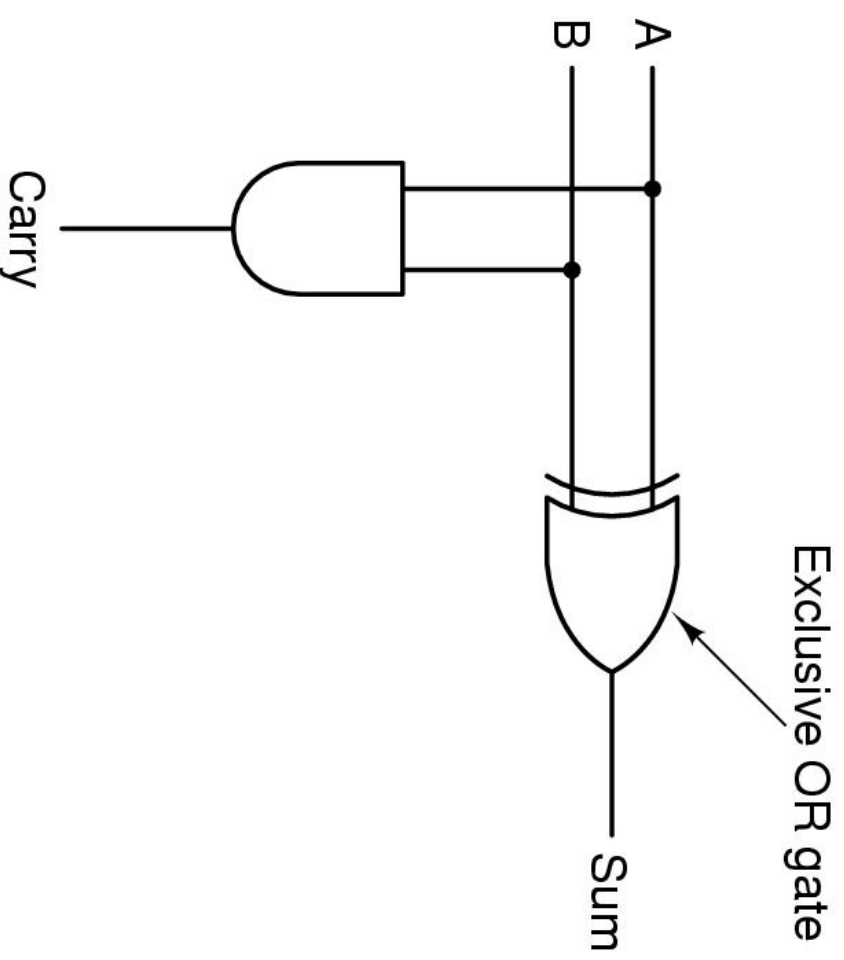
# Circuitos Aritméticos

- Somadores: soma 2 valores
  - Você saberia fazer um somador que possui como entrada o bit A e o bit B?
    - $0 + 0 = 0$
    - $0 + 1 = 1$
    - $1 + 0 = 1$
    - $1 + 1 = ?$

# Circuitos Aritméticos

- Somadores: meio somador

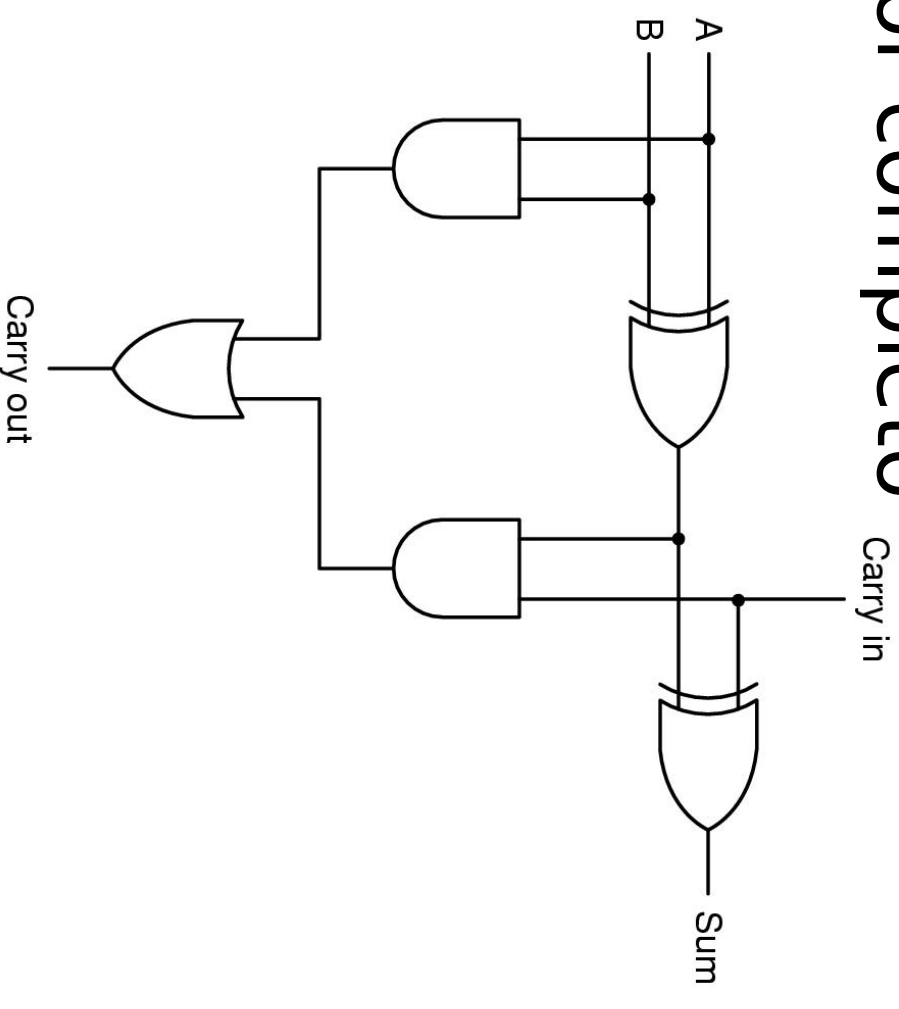
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



# Circuitos Aritméticos

- Somadores: somador completo

A	B	Carry in	Sum	Carry out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



(a)

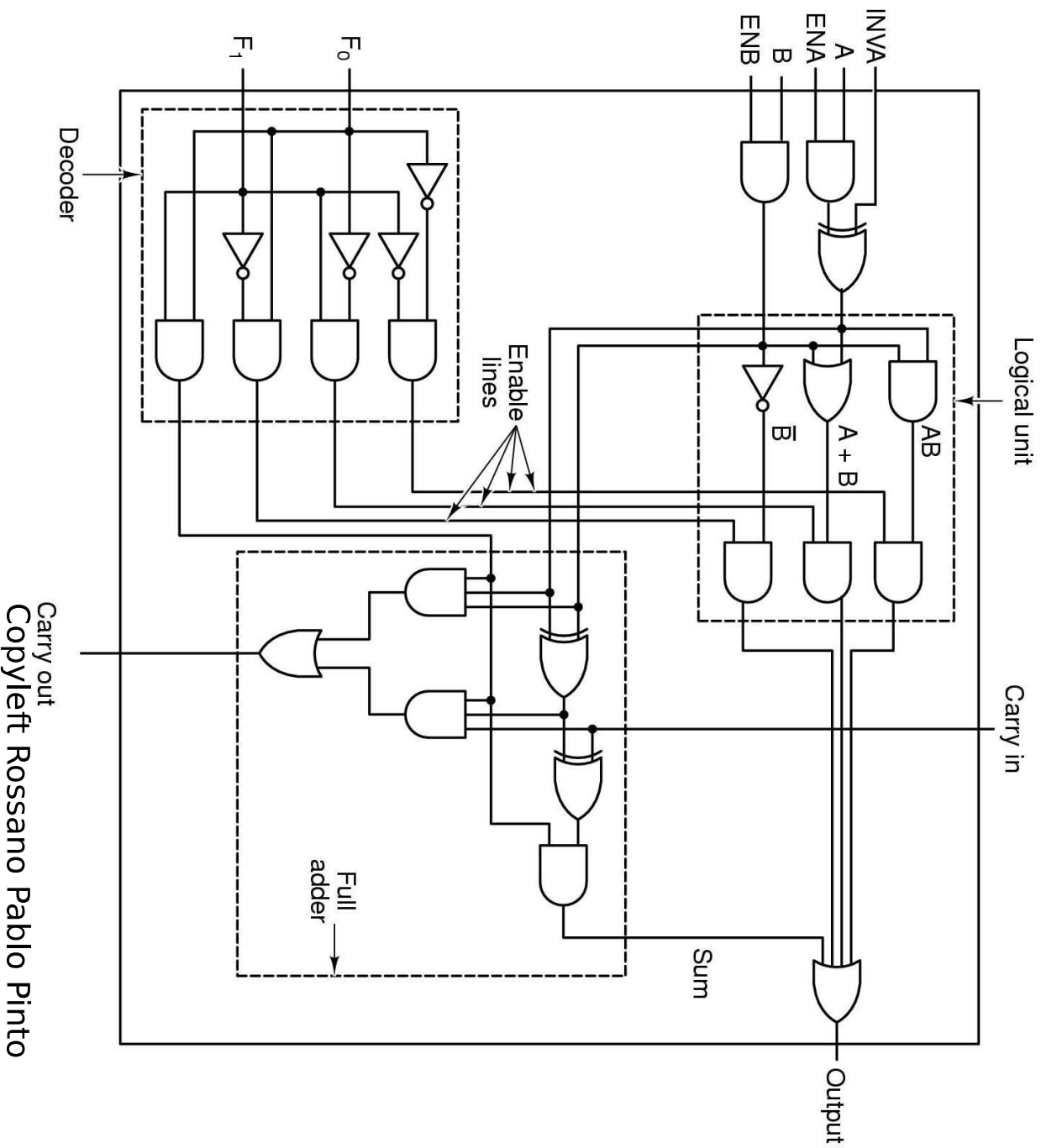
(b)

# Circuitos Aritméticos

- Unidade Lógica e Aritmética
  - opera **AND**, **OR**, **NOT** e soma de 2 palavras de máquina. No exemplo a seguir:
    - 00 -> A **AND** B
    - 01 -> A **OR** B
    - 10 -> **NOT** B
    - 11 -> **SOMA** A e B

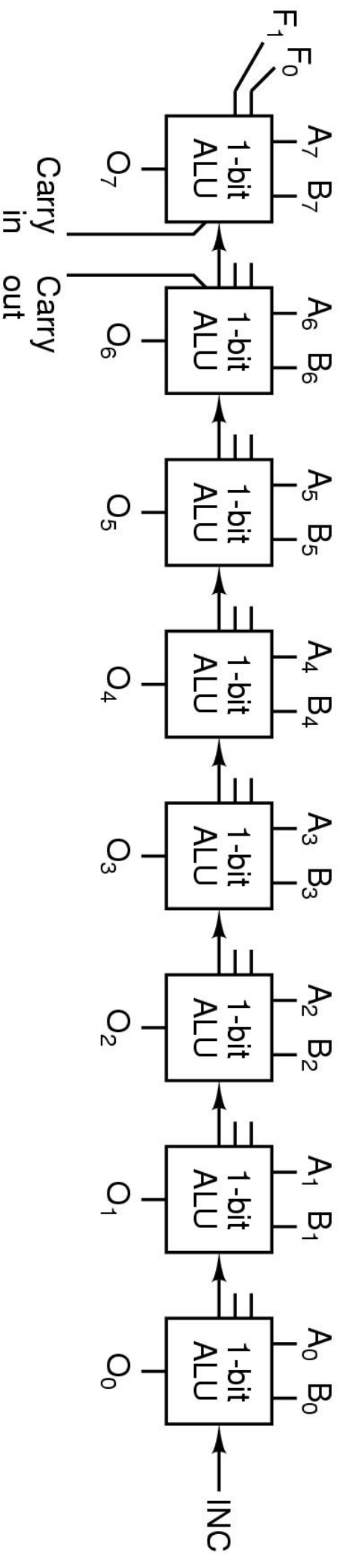
# Circuitos Aritméticos

- ULA de 1 bit



# Circuitos Aritméticos

- 8 ULAs de 1 bit conectadas p/ formar 1 ULA de 8 bits:



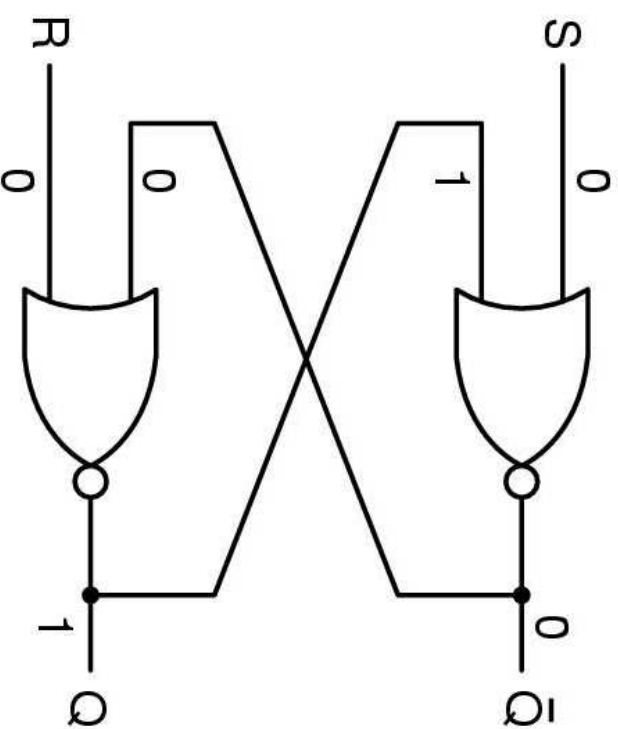
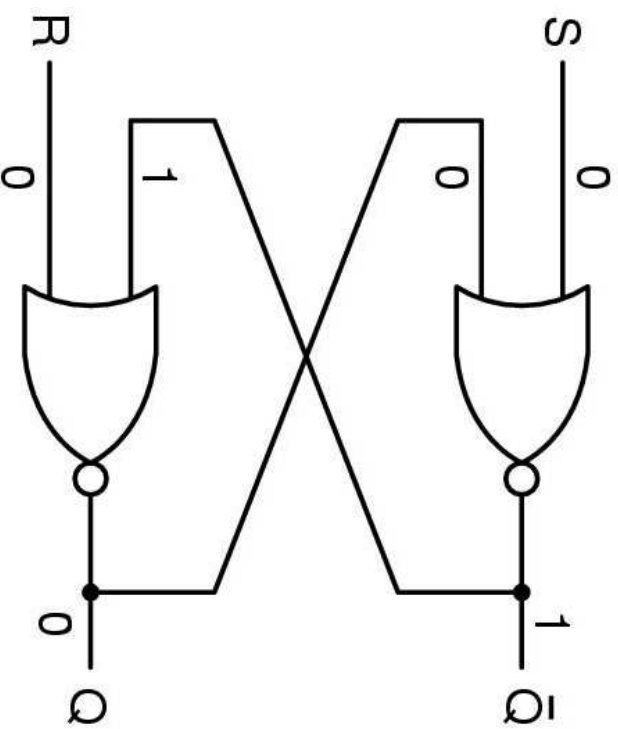
# Memória

- Componente essencial de todo computador
- Sem memória, não existiriam computadores da forma que conhecemos
- Armazena tanto dados quanto instruções
- Memórias podem ser construídas a partir de portas NOR e NAND.



# Memória

- Latches



A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

(c)

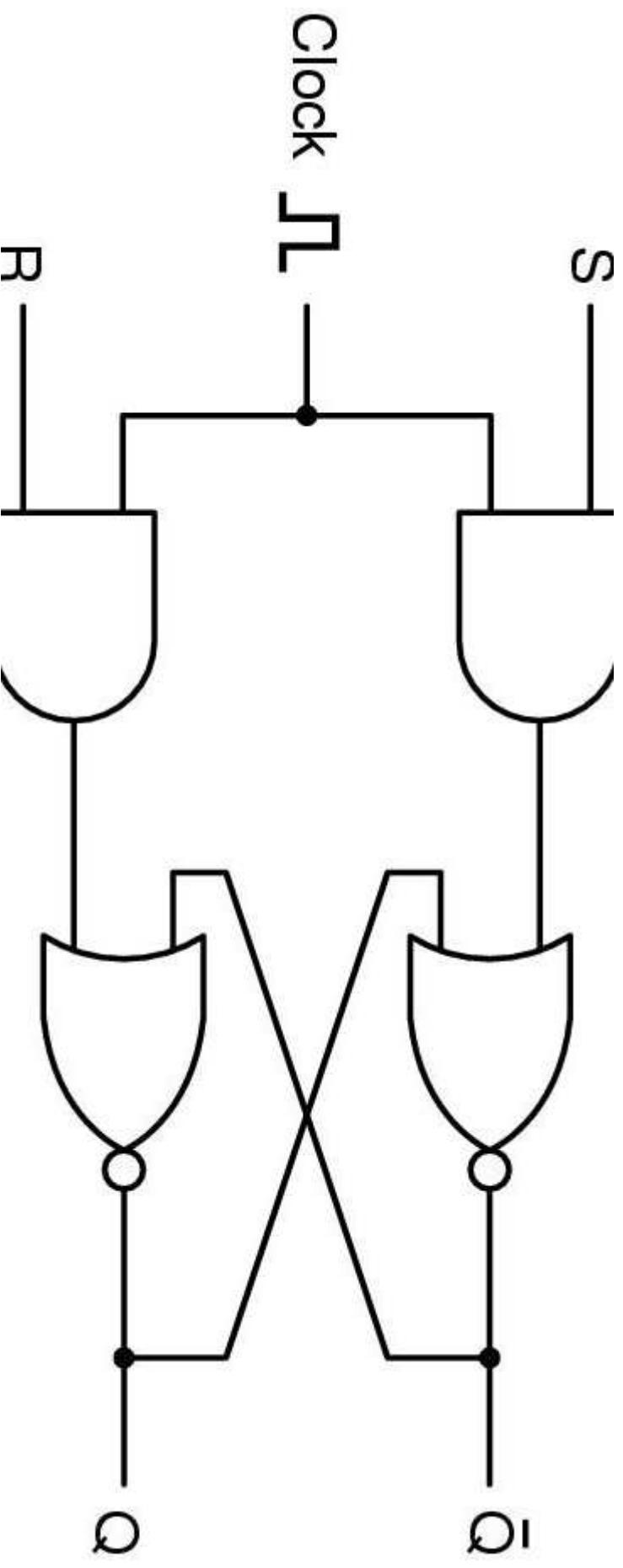
- (a) Latch NOR no estado 0. (b) Latch NOR no estado 1.  
 (c) Tabela verdade p/ porta NOR.

# Memória

- Latches: Diferença **IMPORTANTE**
  - Diferentemente de um circuito combinatório, as saídas do latch **não são determinadas unicamente pelas suas entradas** !!! O circuito leva em conta uma **entrada ANTERIOR**.
  - O circuito “**LEMBRA**” de sua entrada anterior.
  - Com este circuito é possível construir **memórias de computadores**.

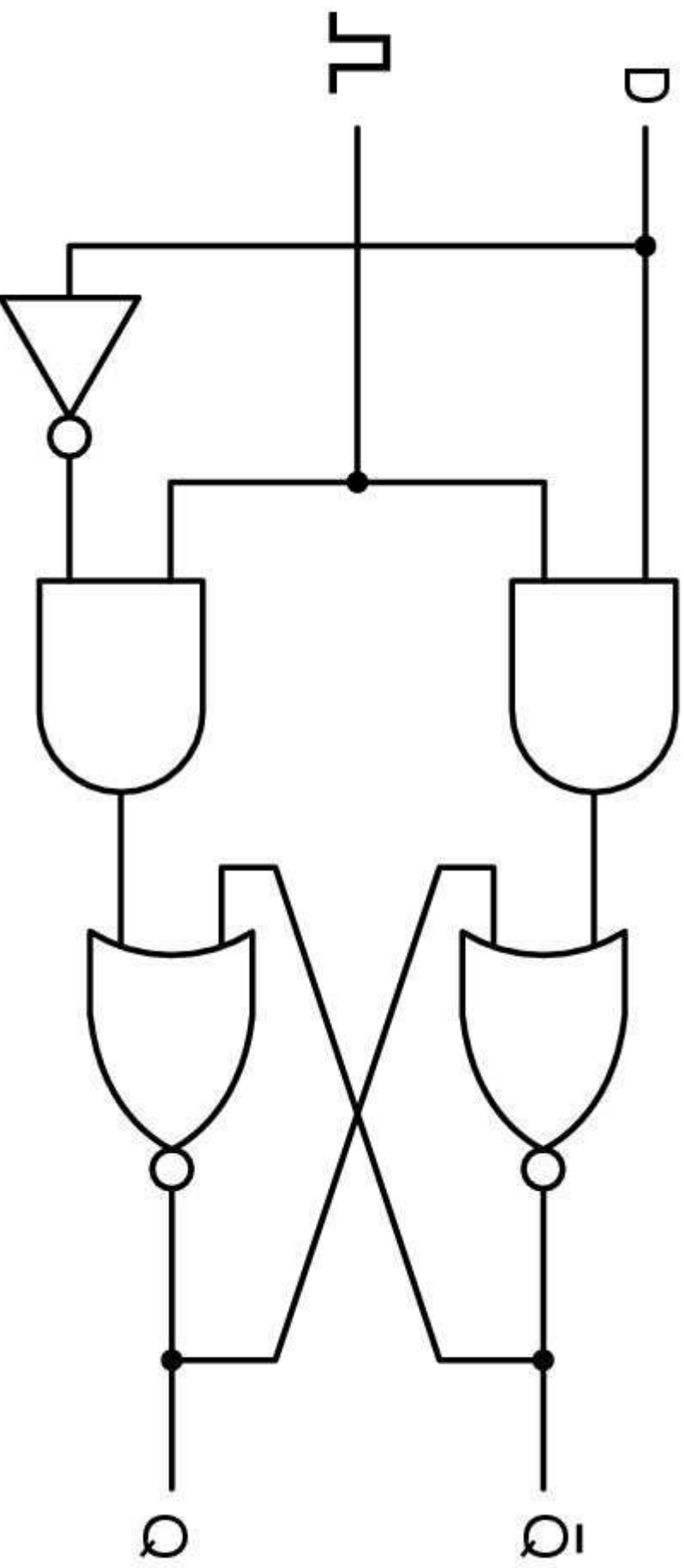
# Memória

- Latches SR com clock



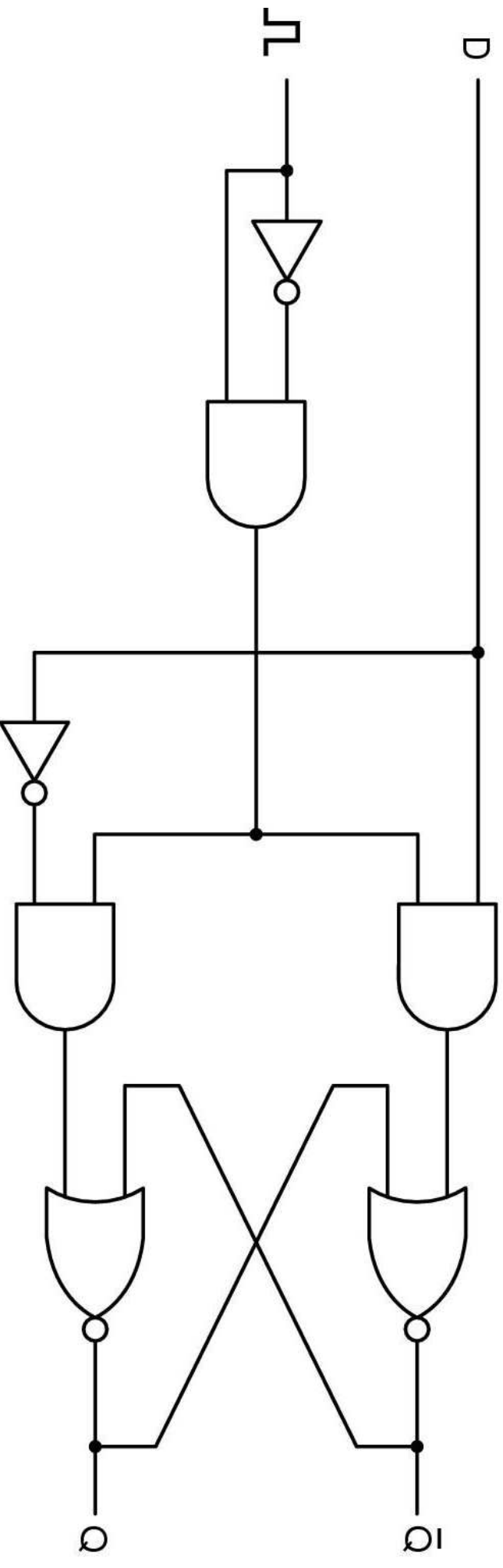
# Memória

- Latches D com clock. Motivação: evitar a entrada  $S=R=1$ . Pois é **não-determinística**.



# Memória

- Flip-flop: igual a latch só que é **edge-triggered** ao invés de **level-triggered**.



# Memória

- Prefixos métricos
  - mili (m) =  $10^{-3}$
  - micro ( $\mu$ ) =  $10^{-6}$
  - nano (n) =  $10^{-9}$
  - pico (p) =  $10^{-12}$

# Memória

- Registradores
  - São formados por vários flip-flops. EX.:
    - registradores de 8 bits são formados por 8 flip-flops
    - 16 bits -> 16 flip-flops
    - 32 bits -> 32 flip-flops
    - n bits -> n flip-flops

# Memória

- RAM – Random Access Memory – Memória de Acesso Aleatório
  - Podem ser escritas e lidas várias vezes
  - Este nome é o oposto de Memória de Acesso Sequencial de fitas magnéticas
  - Duas variedades: SDRAM (Static RAM) e DRAM (Dynamic RAM)



# Memória

- **RAM: SDRAM**
  - Construída com flip-flops D
  - Mantém seu conteúdo enquanto houver alimentação de energia
  - São muito rápidas: acesso em nano segundos ( $10^{-9}$  segundos)
  - Utilizadas para construir memórias cache nível 2

# Memória

- **RAM: DRAM**
  - Construída a partir de array de células. Cada célula é composta por 1 transistor e um capacitor
  - Necessita de ciclos de “atualização” (*refresh*) p/ manter dado a cada X mili-segundos
  - Velocidade na casa dos 60 nano segundos (mais antigas) à 5 nano segundos (DDR)
  - Utilizadas para construir memórias voláteis do sistema (“RAM”)

# Memória

- **RAM: DRAM: tipos assíncronos**
  - linhas de endereço e dados não são sincronizados por um único clock
  - FPM (Fast Page Mode)
  - EDO (Extended Data Output)

# Memória

- **RAM: DRAM: tipos síncronos**
  - linhas de endereço e dados são sincronizados por um único clock
  - SDRAM
    - SDR SDRAM (Single-Data-Rate Synchronous DRAM)
      - Híbrido de RAM estática e dinâmica
    - DDR SDRAM (Double-Data-Rate SDRAM)
      - Transfere dados tanto na subida quanto na descida do sinal de clock

# Memória

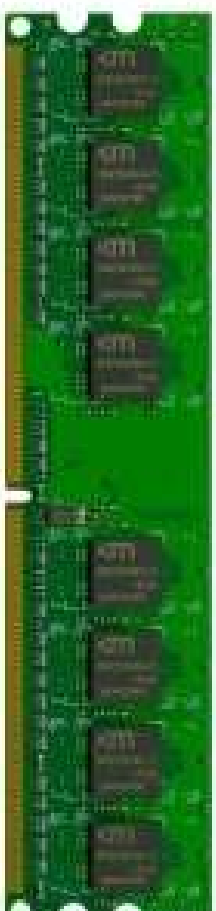
- DDR SDRAM



Standard name	Memory clock	Time between signals	I/O Bus clock	Data transfers per second	Module name	Peak transfer rate
DDR-200	100 MHz	10 ns	100 MHz	200 Million	PC-1600	1.600 GB/s
DDR-266	133 MHz	7.5 ns	133 MHz	266 Million	PC-2100	2.133 GB/s
DDR-333	166 MHz	6 ns	166 MHz	333 Million	PC-2700	2.667 GB/s
DDR-400	200 MHz	5 ns	200 MHz	400 Million	PC-3200	3.200 GB/s

# Memória

- DDR2 SDRAM



Standard name	Memory clock	Time between signals	I/O Bus clock	Data transfers per second	Module name	Peak transfer rate
DDR2-400	100 MHz	10 ns	200 MHz	400 Million	PC2-3200	3.200 GB/s
DDR2-533	133 MHz	7.5 ns	266 MHz	533 Million	PC2-4200	4.264 GB/s
DDR2-667	166 MHz	6 ns	333 MHz	667 Million	PC2-5300	5.336 GB/s
DDR2-800	200 MHz	5 ns	400 MHz	800 Million	PC2-6400	6.400 GB/s
DDR2-1066 (planned)	266 MHz	3.75 ns	533 MHz	1066 Million	PC2-8500 (planned)	8.500 GB/s

# Memória

- ROM (Read-Only Memory)
  - Permite apenas operações de leitura
  - Não são voláteis (mantém dados mesmo sem alimentação de energia elétrica)
  - Dados geralmente são gravados no processo de fabricação da ROM (material foto-sensível)

# Memória

- ROM: PROM (Programmable ROM)
  - Programável/gravável apenas uma vez – utiliza “alta voltagem” para gravar
- ROM: EPROM (Erasable PROM)
  - Similar à PROM
  - “Fotonicamente” apagável com luz ultravioleta (10 à 20 minutos de exposição)
- ROM: EEPROM (Electronic EPROM)
  - Eletronicamente apagável



# Memória

- “ROM”: EEPROM – memória flash
  - acessada como um dispositivo de bloco (PENDRIVE!!!)
  - Leitura e escrita como um procedimento “**padrão**”