

Introdução a Sistemas Digitais

- Definição Sistemas Digitais
- Projeto

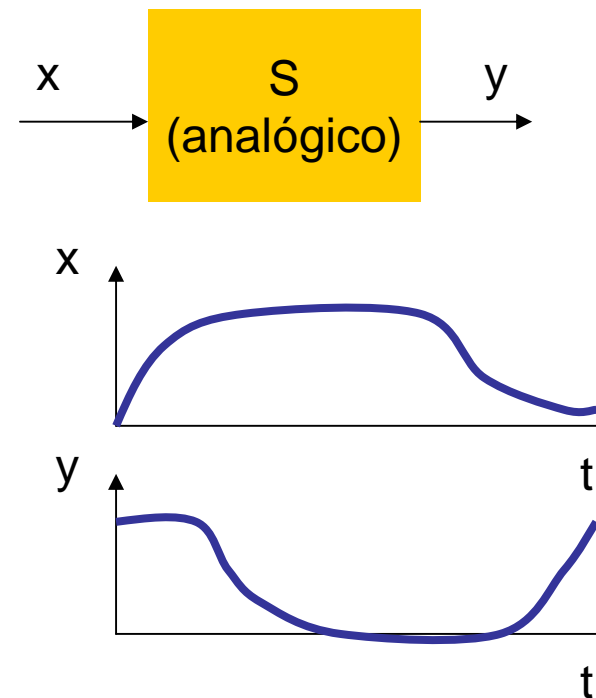
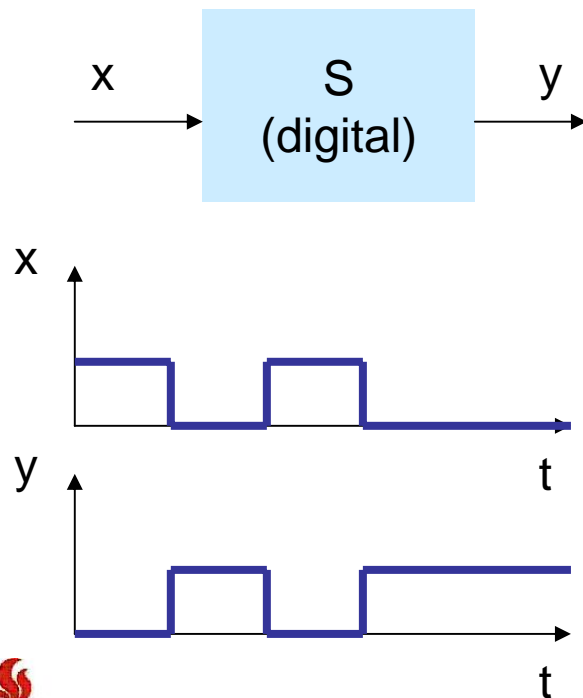
Revisão:

- Circuitos Combinacionais
- Circuitos Sequenciais
- Máquinas de Estados

Sistemas Digitais

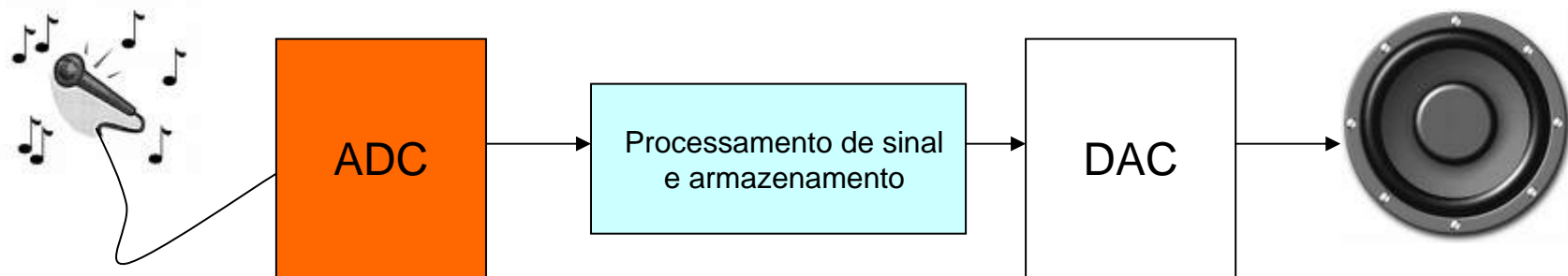
Definição

- Um sistema digital é um sistema no qual os sinais têm um número finito de valores discretos, se contrapondo a sistemas analógicos nos quais os sinais têm valores pertencentes a um conjunto contínuo (infinito).

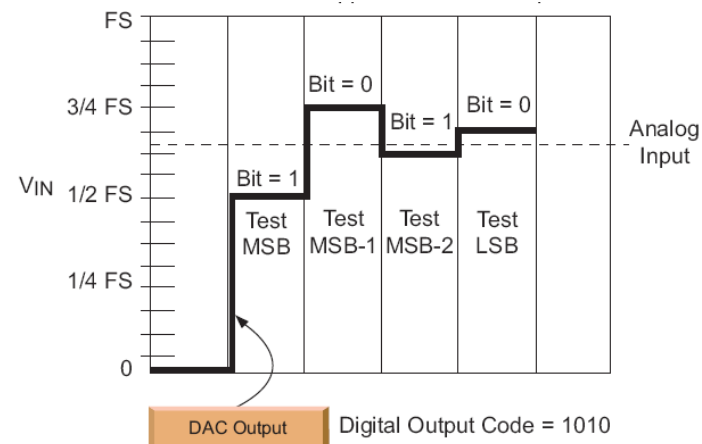
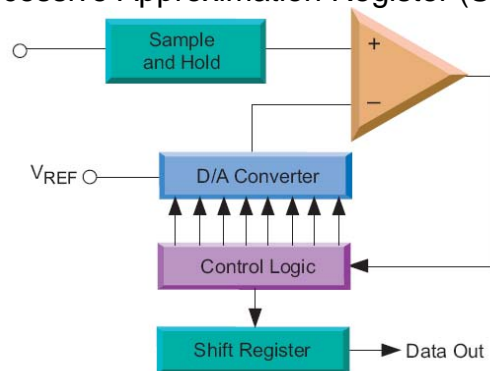


Definição (cont.)

- Uma vez que os sinais do mundo físico são analógicos, é necessários convertê-los para sinais digitais e vice-versa sempre que os sinais digitais tenham que interagir com os sinais do meio físico.

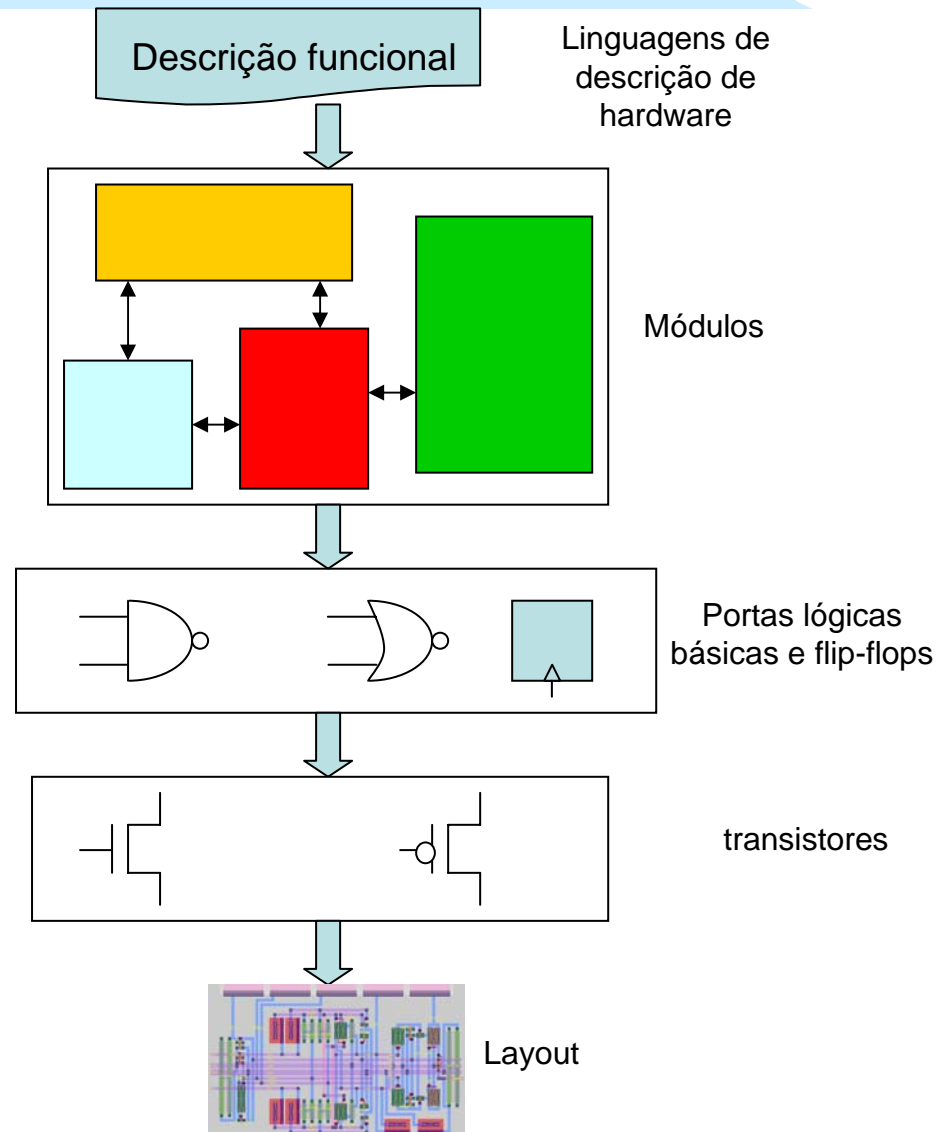


Successive Approximation Register (SAR)



Projeto

- **Abordagem Descendente:** decompõe o sistema em subsistemas que são por sua vez decompostos em subsistemas até atingir o nível de abstração desejado.
- **Desafio:** obter a decomposição adequada para cada nível para que no final os critérios de projeto (área, desempenho, potência) sejam atingidos.
- **Abordagem Ascendente:** conecta módulos disponíveis para formar subsistemas que por sua vez são conectados para formar subsistemas até que a especificação funcional seja satisfeita.
- **Desafio:** trabalhar com um conjunto muito grande de subsistemas pequenos para compor um sistema muito complexo.



Sistemas Digitais

Aula
1

Projeto: Arbodagem Descendente

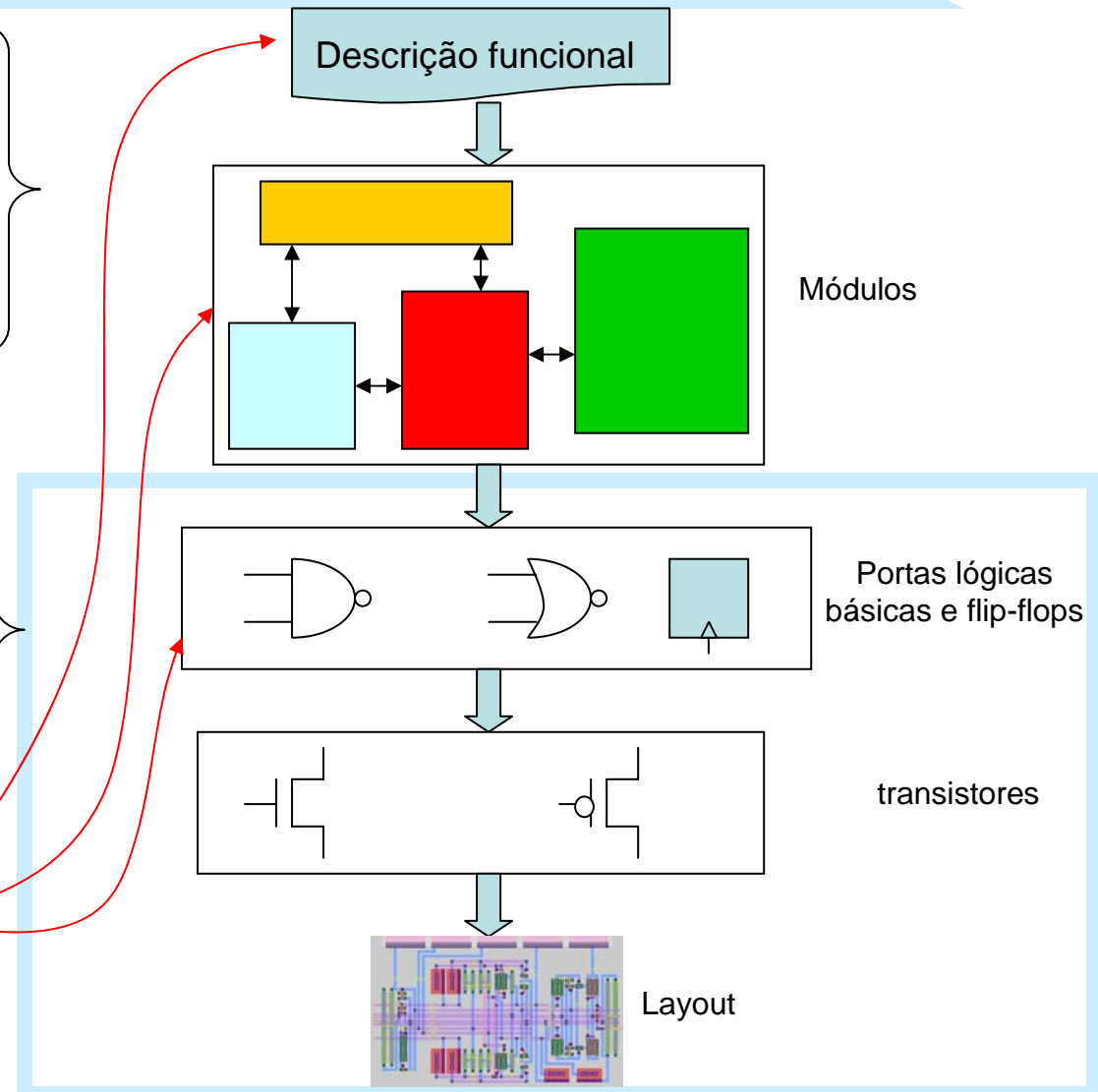
Projeto de Sistemas Digitais usando fluxogramas, grafos, máquinas de estados e diagrama de blocos

Descrever o projeto em **linguagens de descrição de hardware** como por exemplo VHDL

Usar ferramentas de síntese lógica para bibliotecas de células como o Leonardo da Mentor

Usar ferramentas de síntese lógica para plataformas programáveis como FPGAs (Xilinx – ISE, Altera – Quartus, Actel – Libero).

Verificar funcionalmente através de simulação lógica com e sem atraso.



Sistemas Digitais

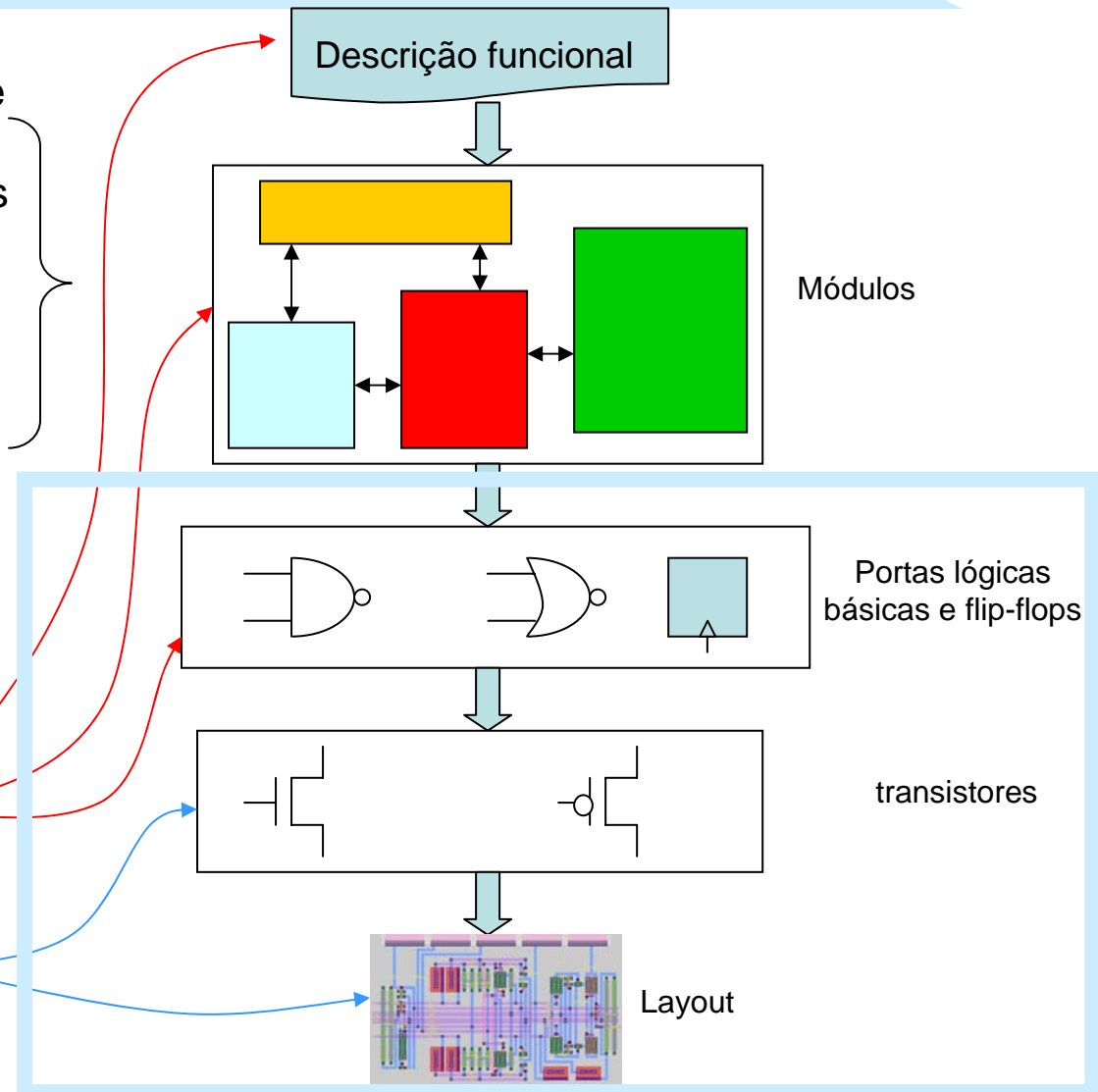
Projeto: Arbodagem Ascendente

Projeto de somadores, multiplicadores e outros subsistemas combinacionais e sequenciais de alta eficiência em termos de área, desempenho e potência para uso em sistemas digitais complexos.

Descrever o projeto em **linguagens de descrição de hardware** como por exemplo VHDL ou em esquemático

Verificar funcionalmente através de simulação lógica com e sem atraso.

Verificar eletricamente através de simulação elétrica (SPICE)



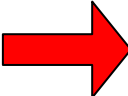
Sistemas Digitais

Aula

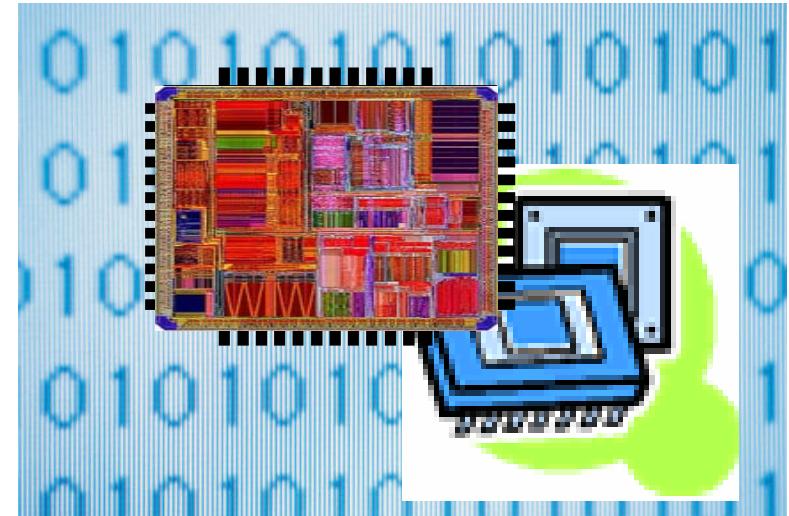
1

Projeto: considerações finais

- Levar em consideração o projeto na abordagem descendente:




ALGORITMO  HARDWARE

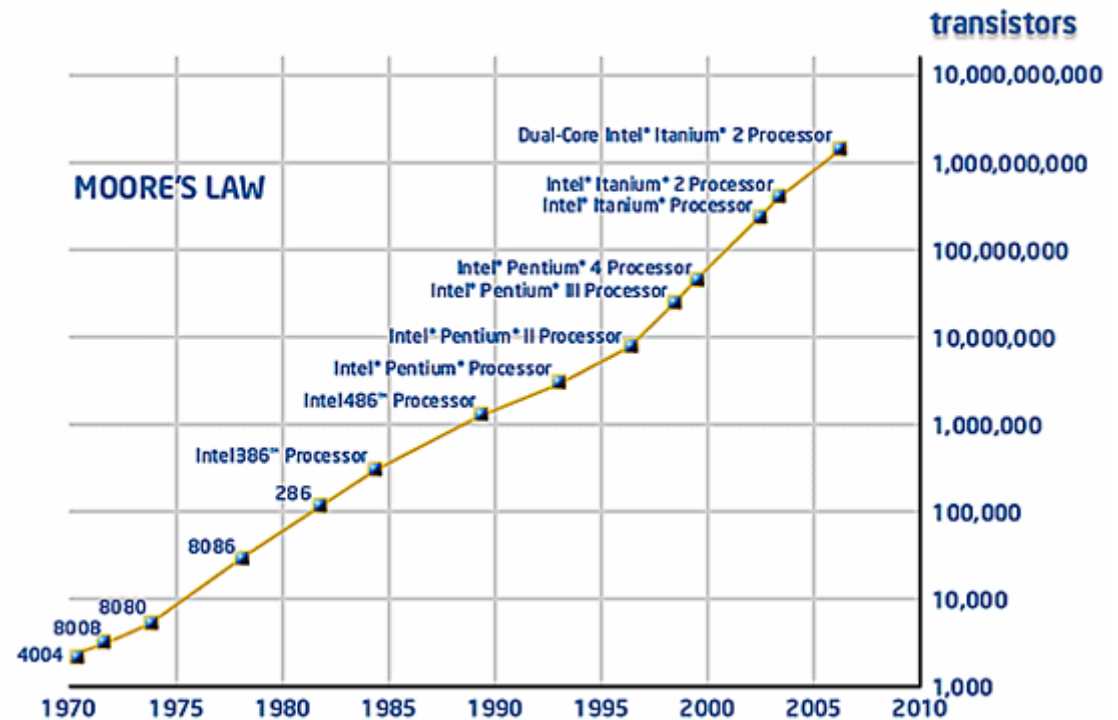
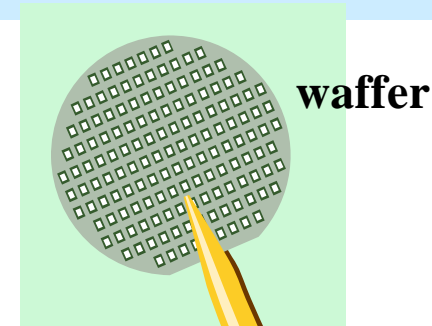
... mas também a abordagem ascendente no momento de otimizar certos blocos e subsistemas no projeto final do sistema digital visando o melhor custo x benefício.



O que esperamos de um Sistema Digital?

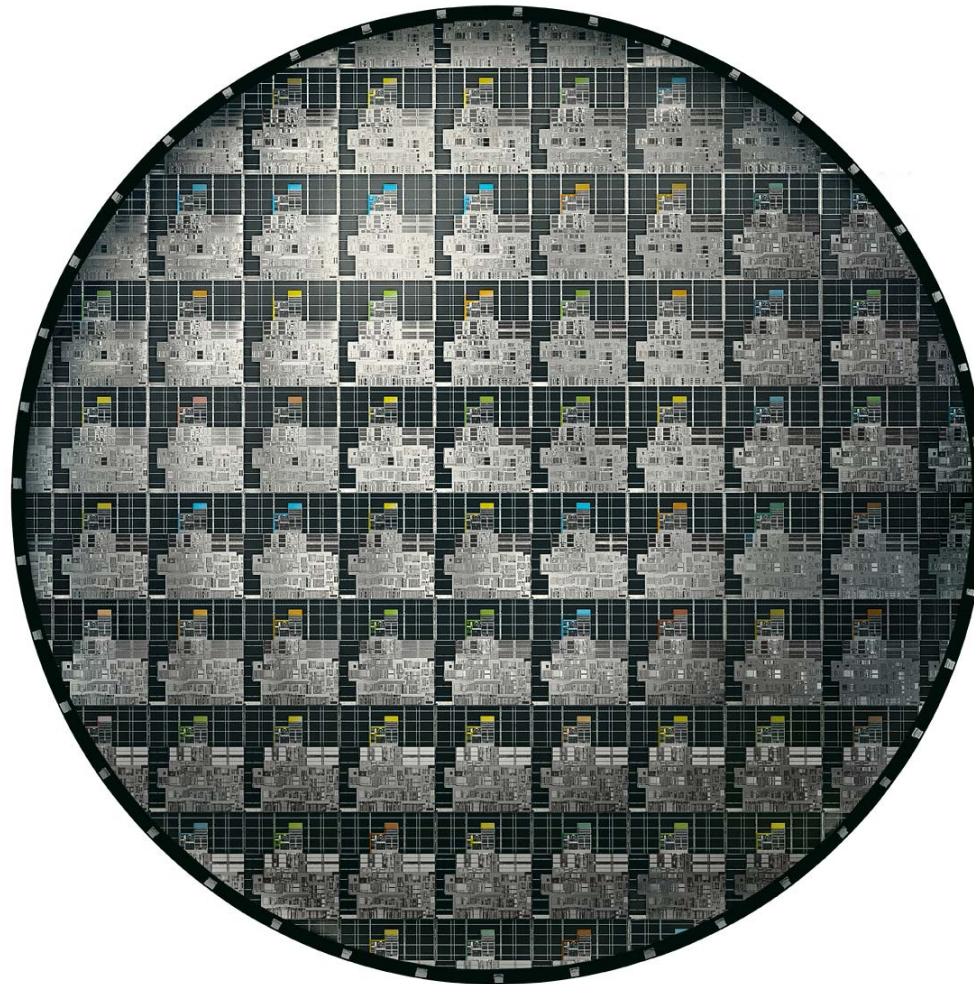
Fatores de evolução

- densidade de integração  , área ocupada
- consumo de potência 
- freqüência de operação 
- custo de fabricação



Wafer of Intel® Itanium® processors

Aula
1





In 1978, a commercial flight between New York and Paris cost around \$900 and took seven hours. If the principles of Moore's Law had been applied to the airline industry the way they have to the semiconductor industry since 1978, that flight would now cost about a penny and take less than one second.



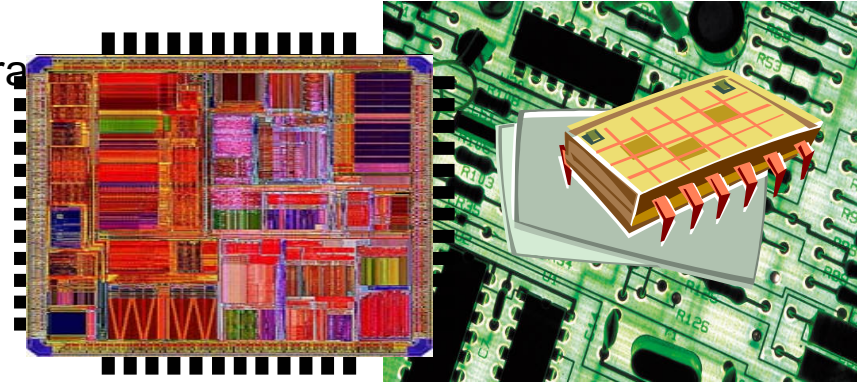
On the road to a billion transistors per chip, Intel has developed transistors so small that about 200 million of them could fit on the head of each of these pins.

Tipos de componentes

Circuito de aplicação específica (ASIC):

circuito integrado projetado especialmente para uma determinada função e sistema digital.

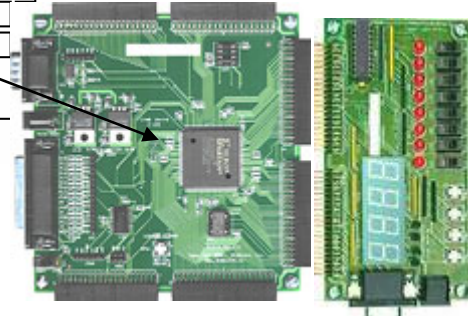
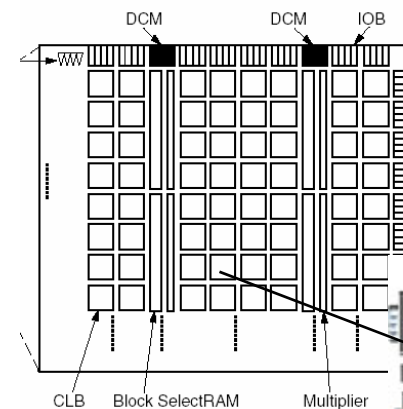
- Full-custom
- semi-custom
- Standard cell



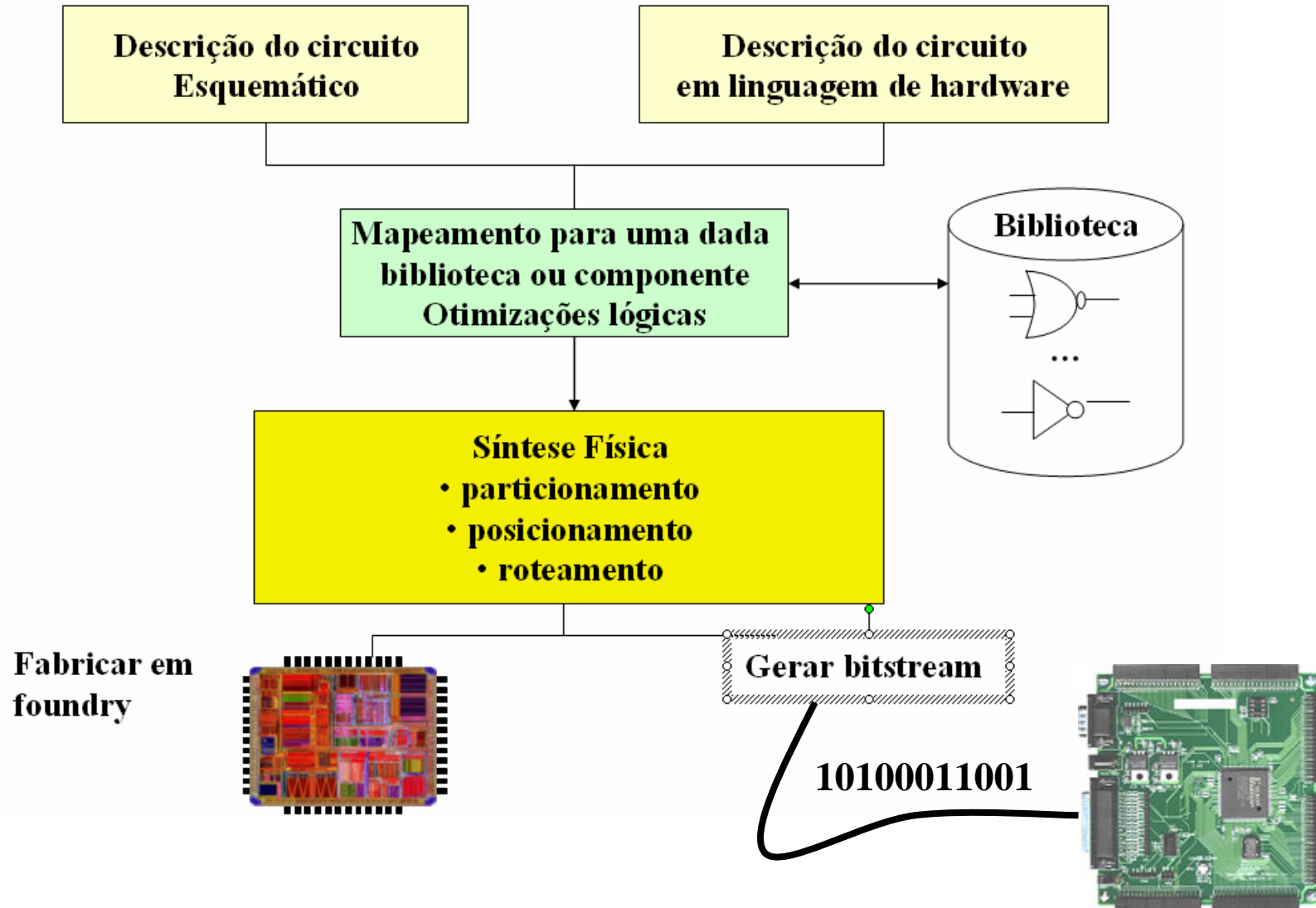
Lógica programável (FPGAs): circuito que pode ser customizado e re-programado para realizar diversas funções.

Compromisso:

Custo X tempo de projeto X desempenho

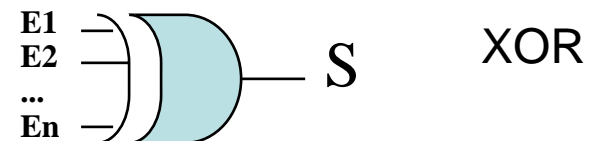
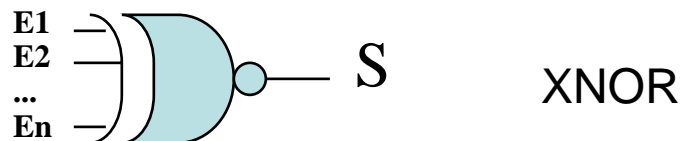
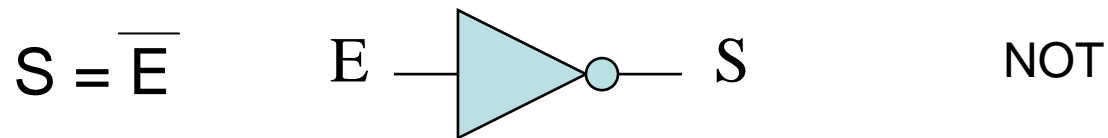


Fluxo de Projeto (simplificado)



Revisão: Circuitos Combinacionais

Portas Lógicas Básicas



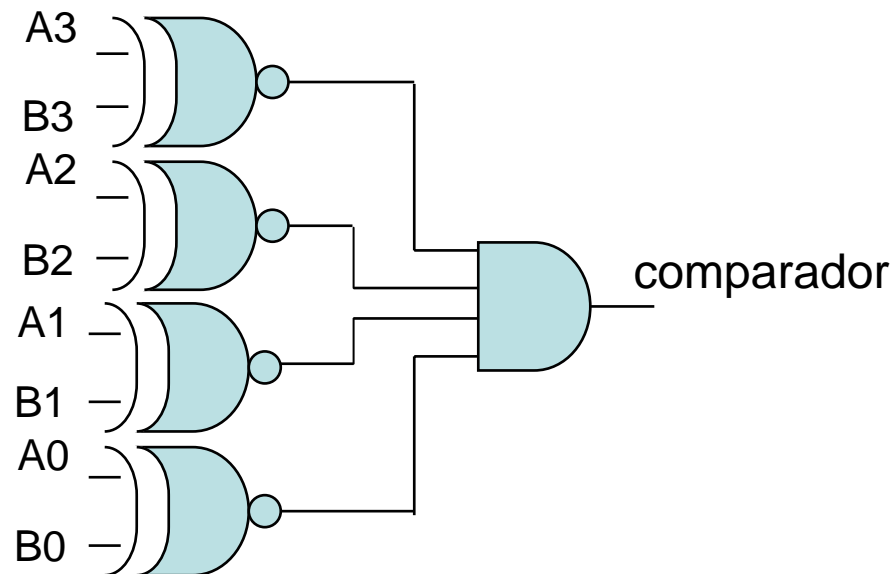
Revisão: Circuitos Combinacionais

Aula
1

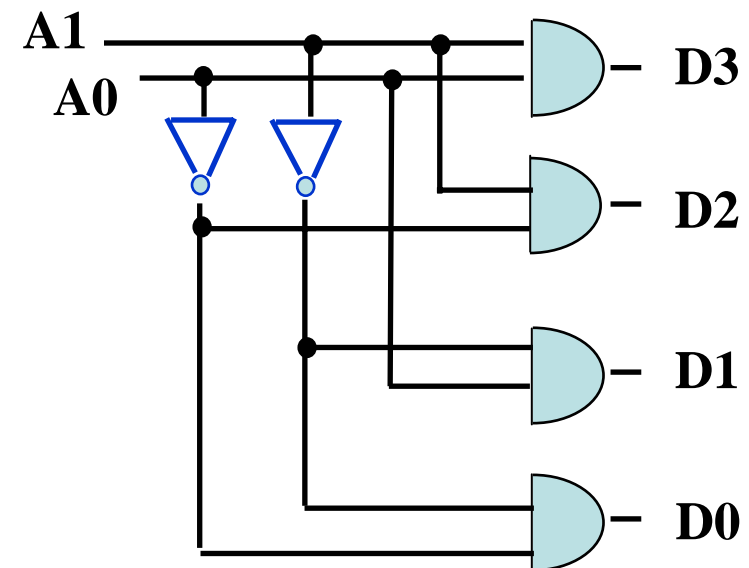
Blocos lógicos

- Saída depende apenas das entradas

Comparador de 4 bits
(A3A2A1A0 e B3B2B1B0)



Decodificador de 2 bits
(A1A0)

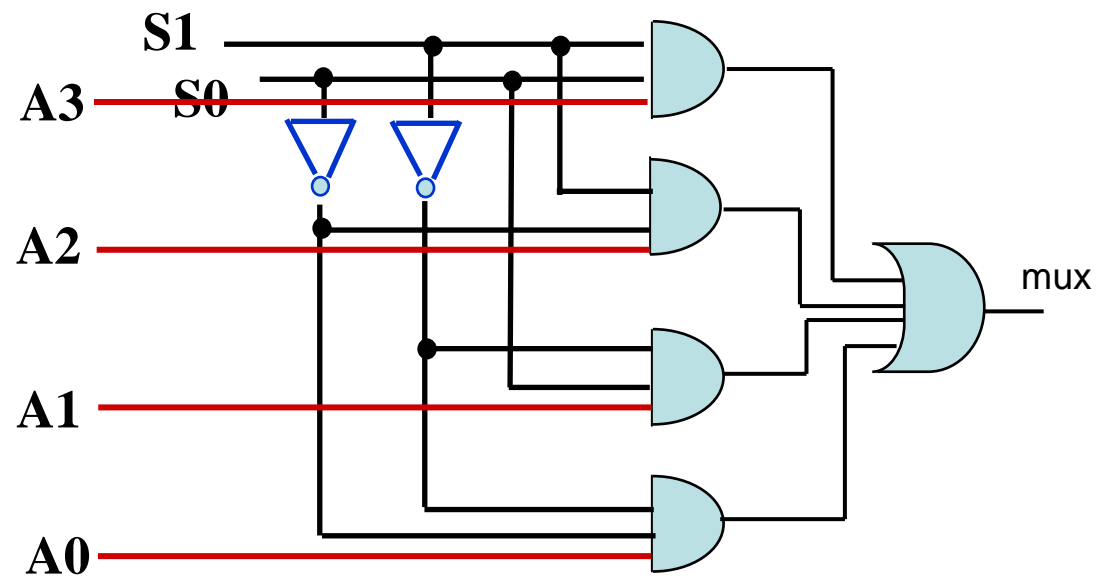
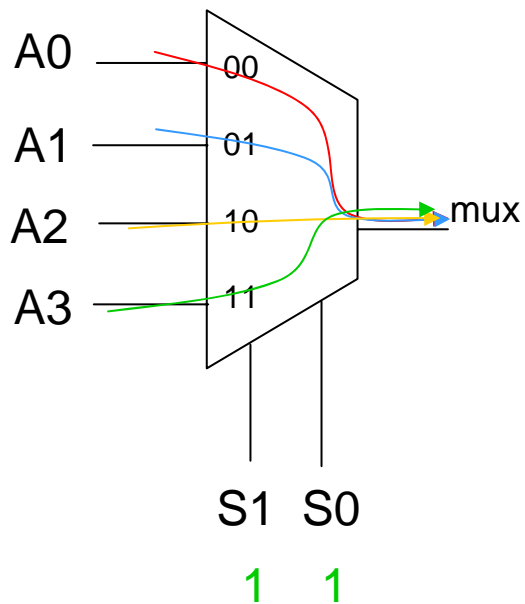


Revisão: Circuitos Combinacionais

Blocos lógicos

- Saída depende apenas das entradas

Multiplexador 4:1

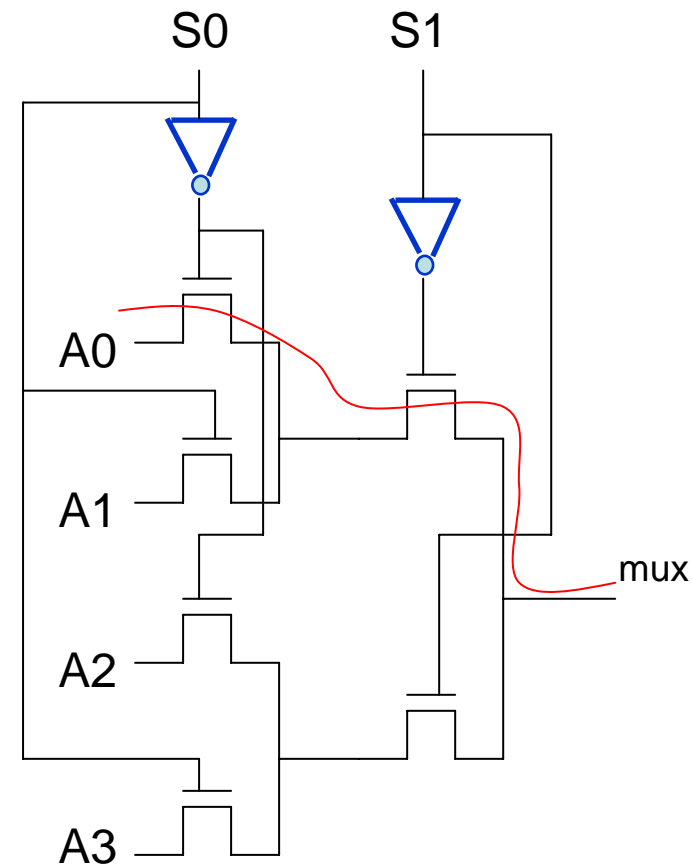
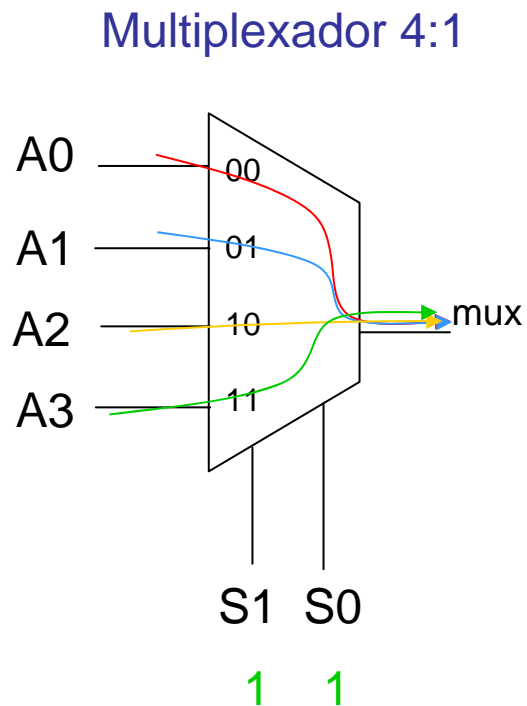


Revisão: Circuitos Combinacionais

Aula
1

Blocos lógicos

- Saída depende apenas das entradas



Revisão: Circuitos Combinacionais

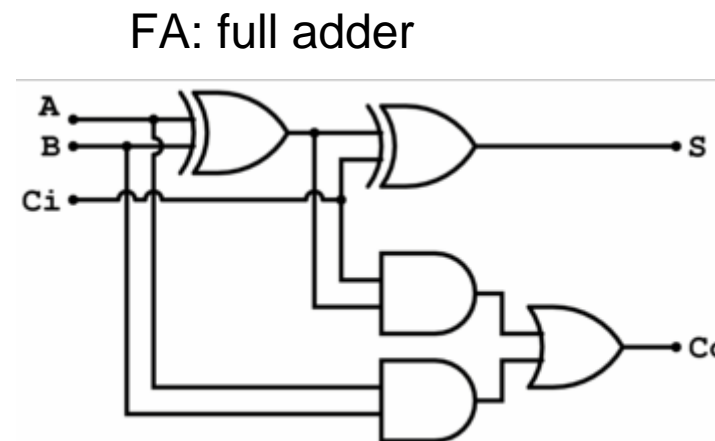
Blocos lógicos

- Somador de 1 bit completo (Full-adder):

$$S = (A \text{ xor } B) \text{ xor } C_i$$

$$C_o = (A \text{ and } B) \text{ or } (C_i \text{ and } (A \text{ xor } B)) = (A \text{ and } B) \text{ or } (B \text{ and } C_i) \text{ or } (C_i \text{ and } A)$$

A	B	C _i	S	C _o
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

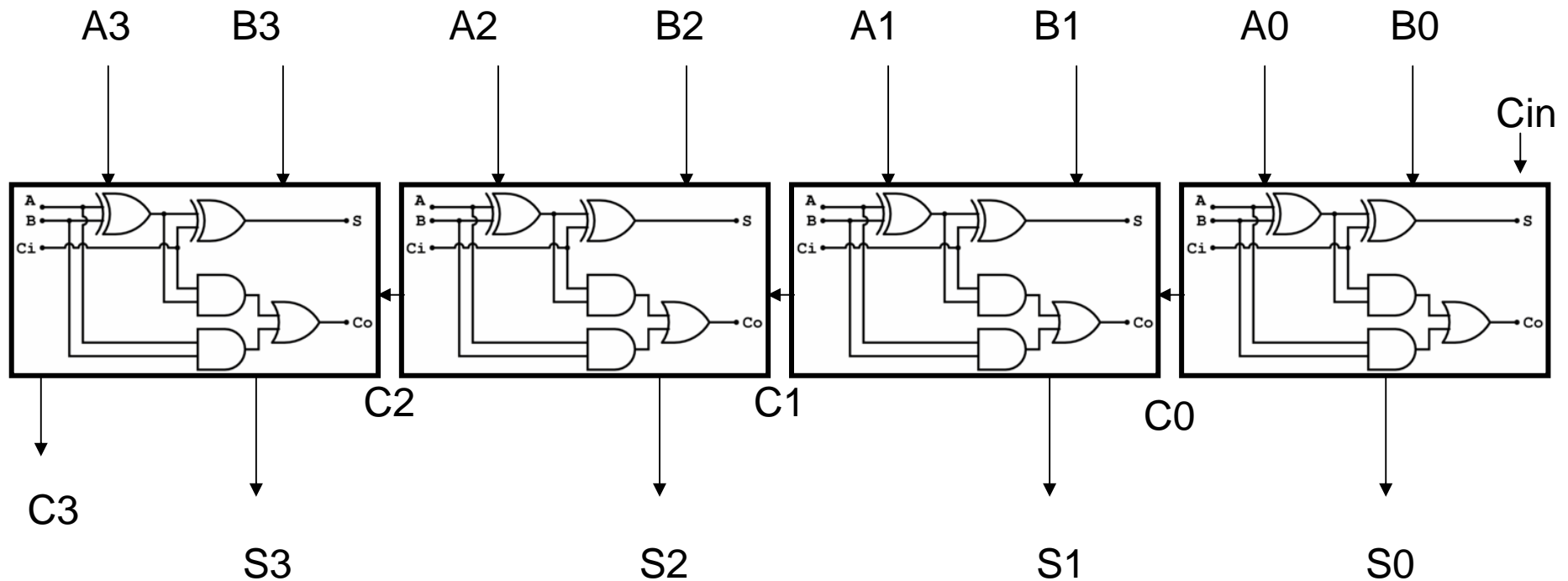


Revisão: Circuitos Combinacionais

Aula
1

Blocos lógicos

Somador Ripple-Carry

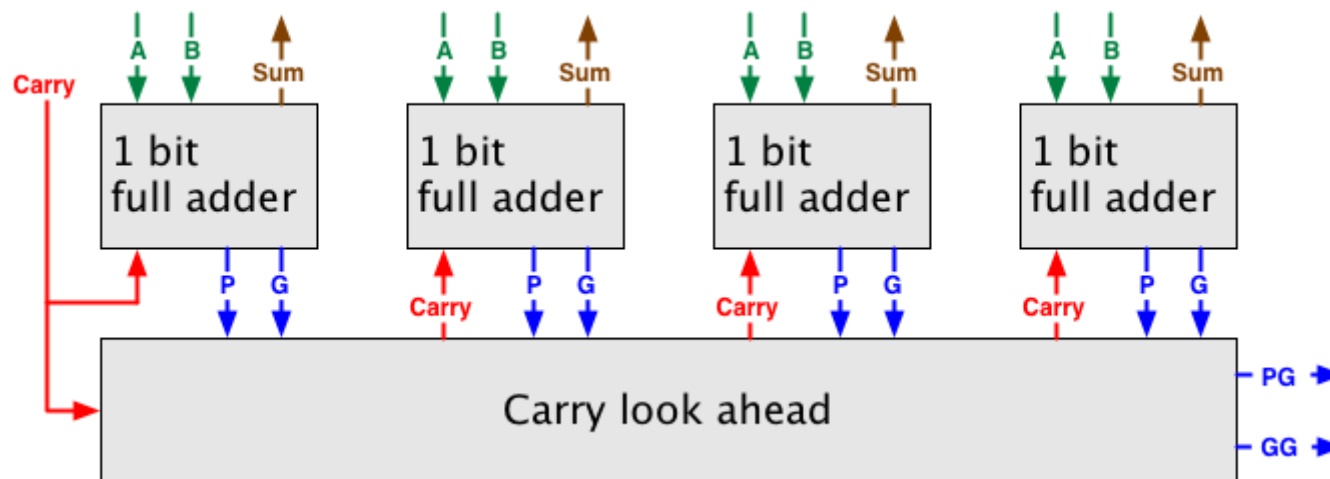


Revisão: Circuitos Combinacionais

Aula
1

Blocos lógicos

- Somador Carry Look Ahead



$$\begin{aligned} C_1 &= G_0 \text{ or } P_0 C_0 \\ C_2 &= G_1 \text{ or } P_1 C_1 = G_1 \text{ or } P_1 G_0 \text{ or } P_1 P_0 C_0 \\ C_3 &= G_2 \text{ or } P_2 C_2 = G_2 \text{ or } P_2 G_1 \text{ or } P_2 P_1 G_0 \text{ or } P_2 P_1 P_0 C_0 \\ C_4 &= G_3 \text{ or } P_3 C_3 = G_3 \text{ or } P_3 G_2 \text{ or } P_3 P_2 G_1 \text{ or } P_3 P_2 P_1 G_0 \text{ or } P_3 P_2 P_1 P_0 C_0 \end{aligned}$$

$$\begin{aligned} G &= A \text{ and } B \\ P &= A \text{ xor } B \quad (\text{also can be implemented as } P = A \text{ or } B) \end{aligned}$$

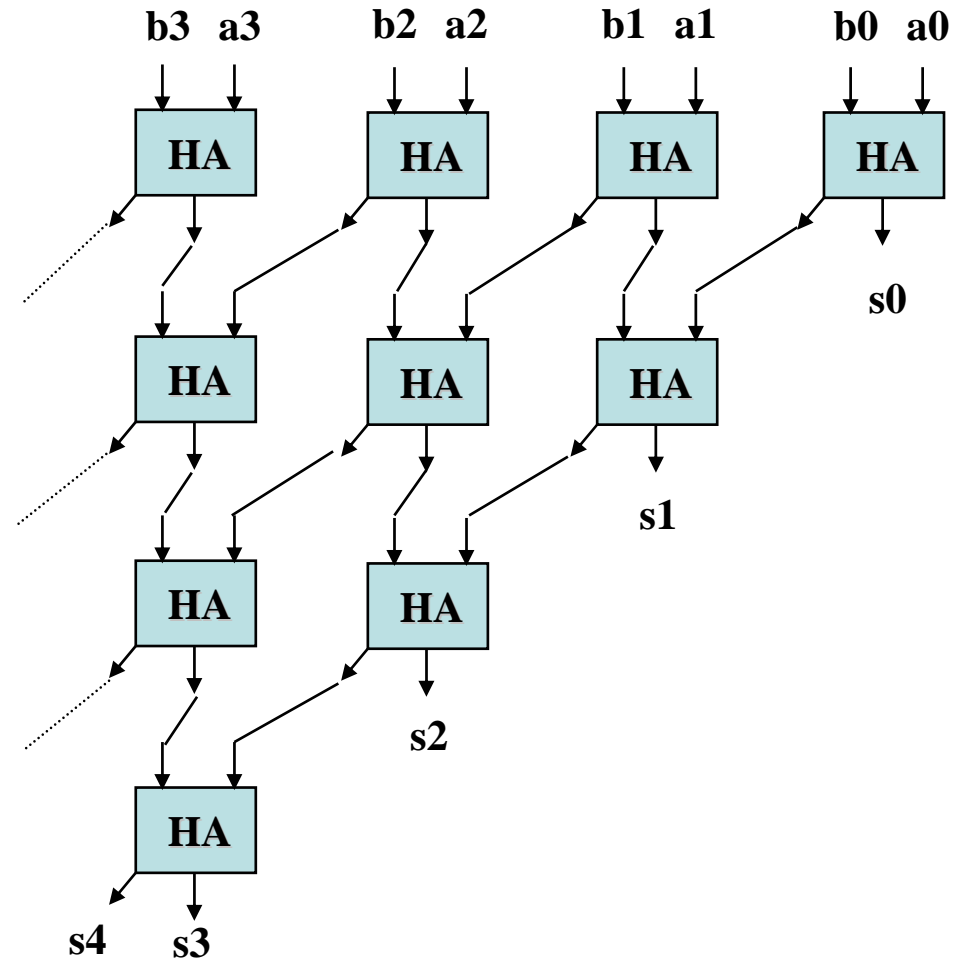
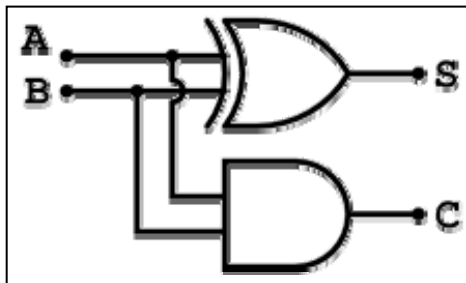
Revisão: Circuitos Combinacionais

Blocos lógicos

Somador de n bits

A ⇒	a3	a2	a1	a0
B ⇒	b3	b2	b1	b0
<hr/>				
s4(Cout)	s3	s2	s1	s0

HA: half adder



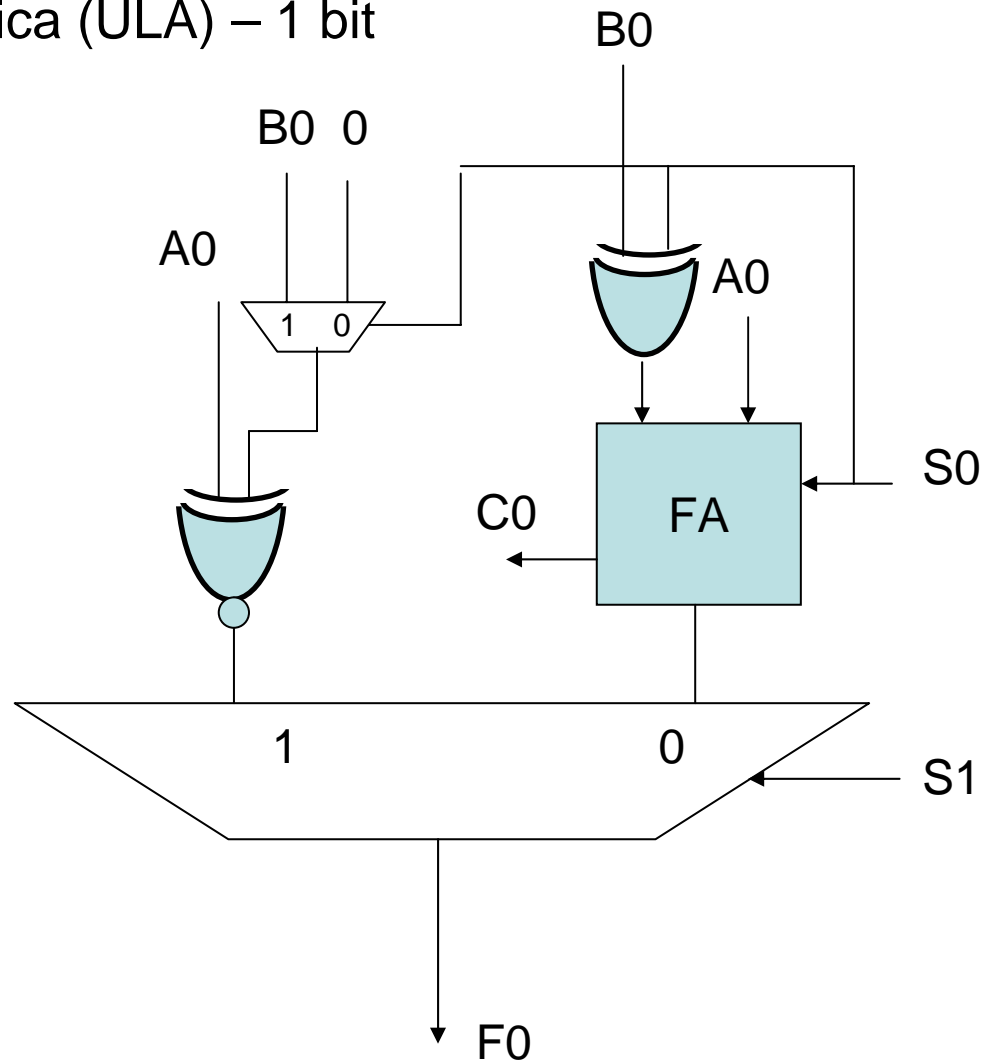
Revisão: Circuitos Combinacionais

Aula
1

Blocos lógicos

- Unidade Aritmética e Lógica (ULA) – 1 bit

S1	S0	Função
0	0	soma A+B
0	1	subtração A-B
1	0	inversão de A
1	1	comparação

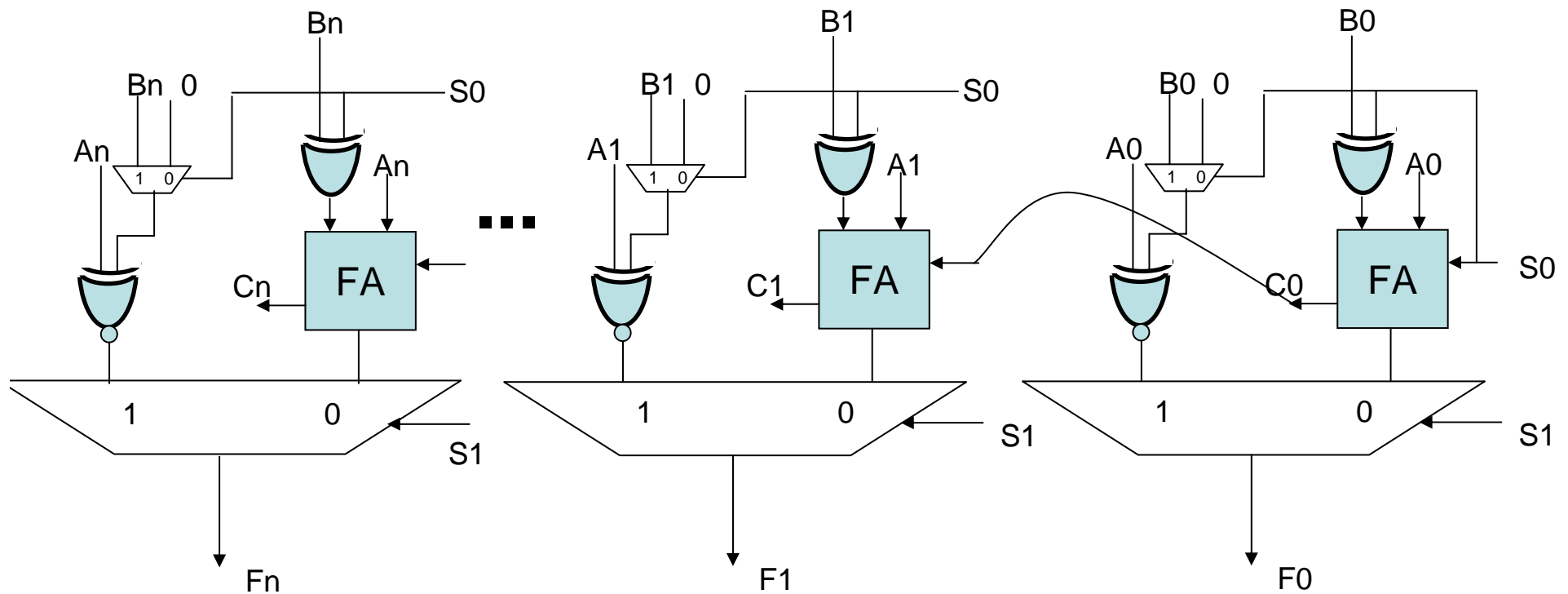


Revisão: Circuitos Combinacionais

Aula
1

Blocos lógicos

- Unidade Aritmética e Lógica (ULA) – n bit

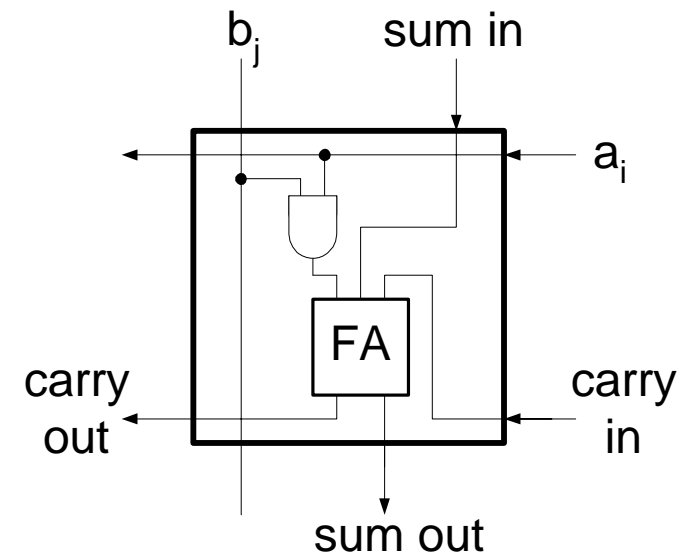
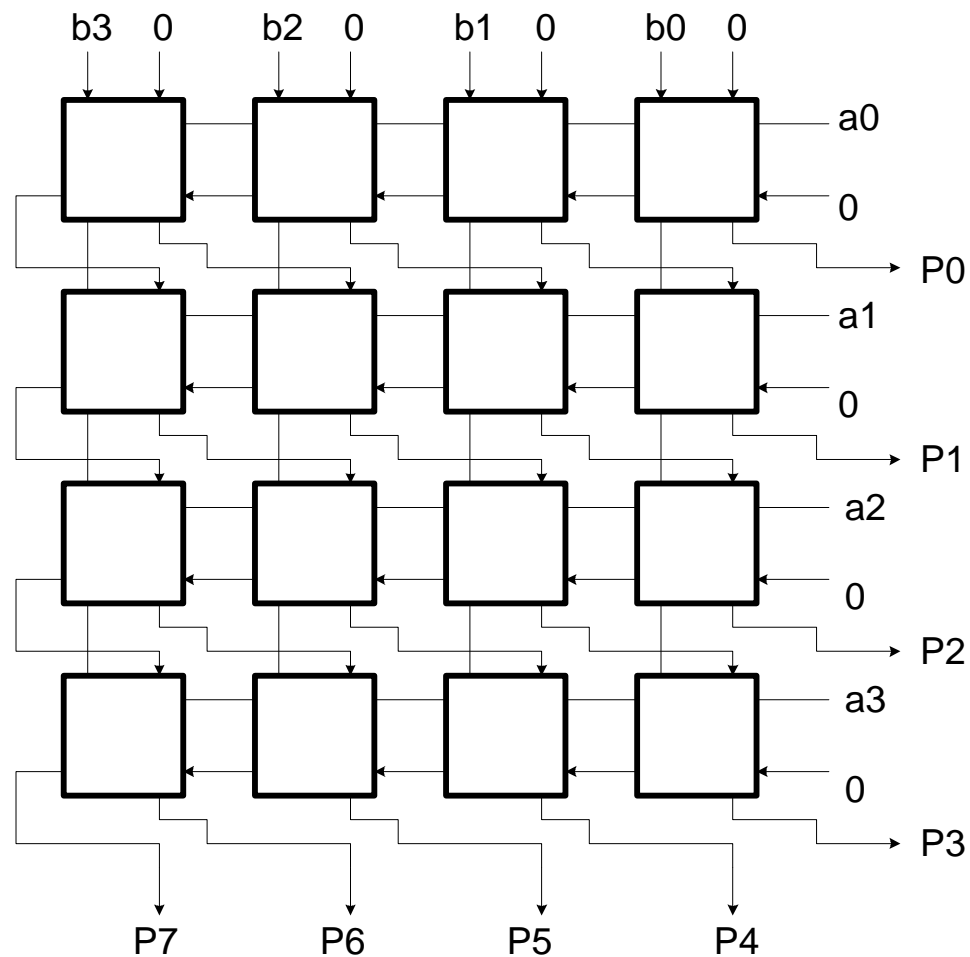


Revisão: Circuitos Combinacionais

Aula
1

Blocos lógicos

- Multiplicador

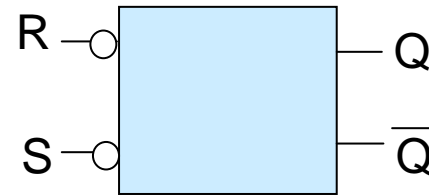


Revisão: Circuitos Sequenciais

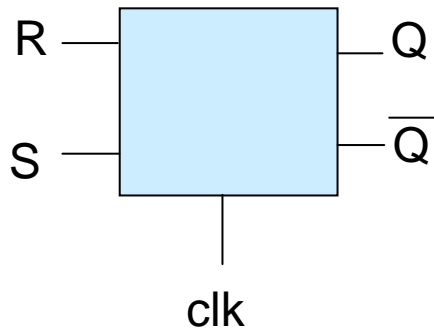
Latches: sensível ao nível do relógio (clk)



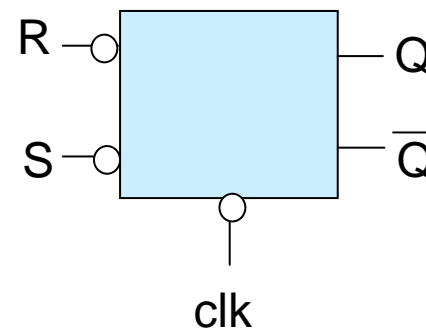
R	S	Q
0	0	Q
0	1	set
1	0	reset
1	1	invalido



R	S	Q
0	0	invalido
0	1	reset
1	0	set
1	1	Q



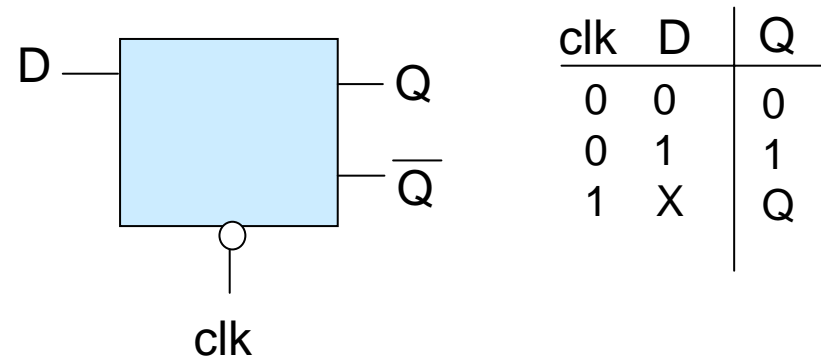
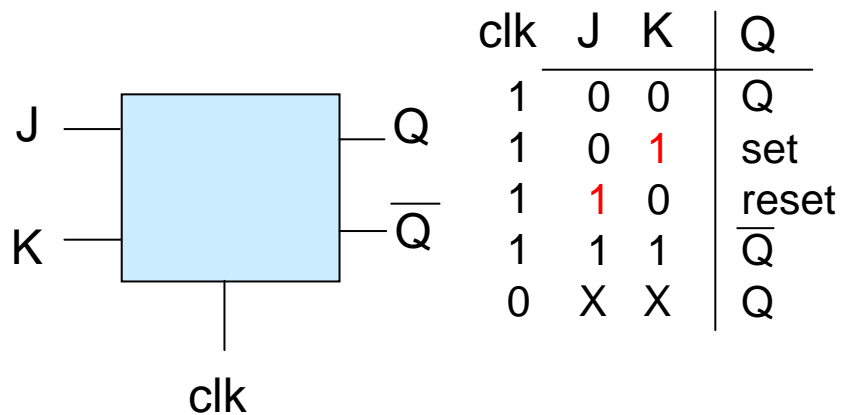
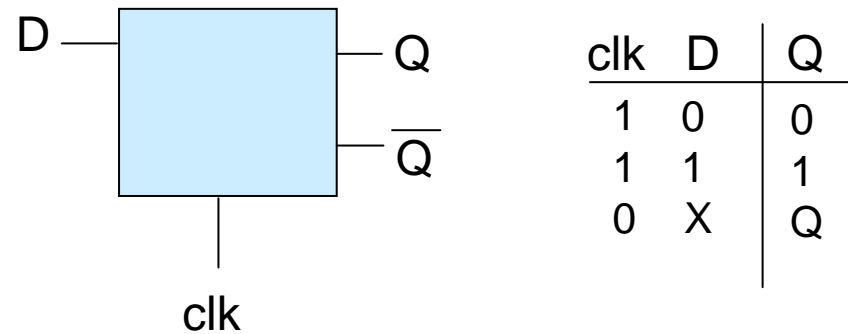
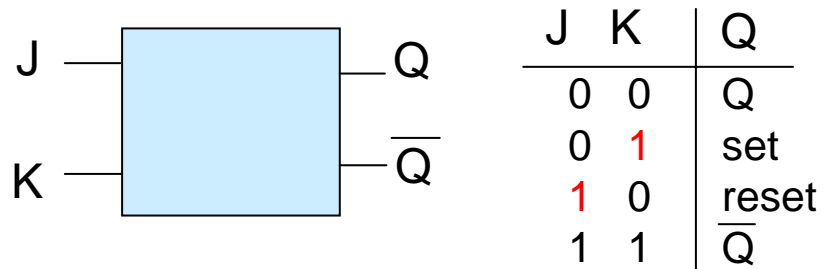
clk	R	S	Q
1	0	0	Q
1	0	1	set
1	1	0	reset
1	1	1	Invalido
0	X	X	Q



clk	R	S	Q
0	0	0	invalido
0	0	1	reset
0	1	0	set
0	1	1	Q
1	X	X	Q

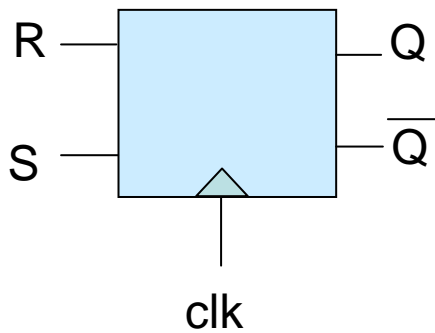
Revisão: Circuitos Sequenciais

Latches: sensível ao nível do relógio (clk)

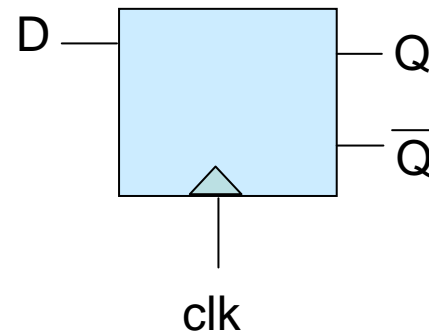


Revisão: Circuitos Sequenciais

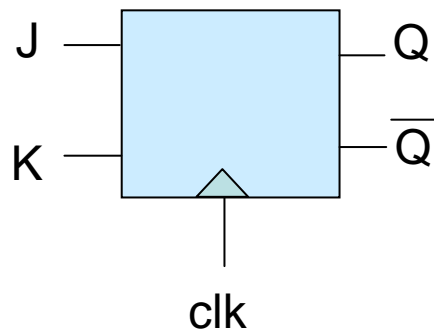
Flip-flops: sensível a borda do relógio (clk)



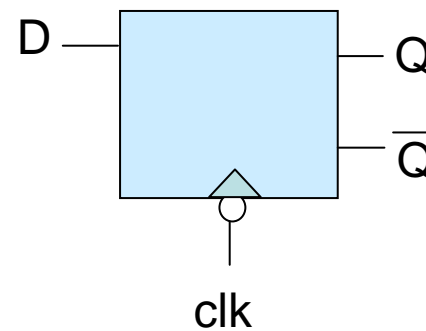
clk	R	S	Q
\uparrow	0	0	Q
\uparrow	0	1	set
\uparrow	1	0	reset
\uparrow	1	1	Invalido
X	X	X	Q



clk	D	Q
\uparrow	0	0
\uparrow	1	1
X	X	Q



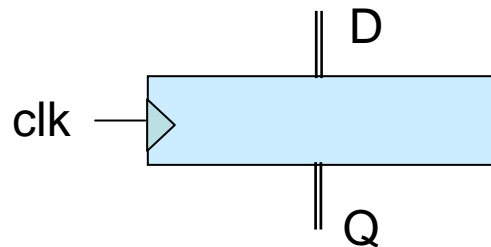
clk	J	K	Q
\uparrow	0	0	Q
\uparrow	0	1	set
\uparrow	1	0	reset
\uparrow	1	1	\overline{Q}
X	X	X	Q



clk	D	Q
\downarrow	0	0
\downarrow	1	1
X	X	Q

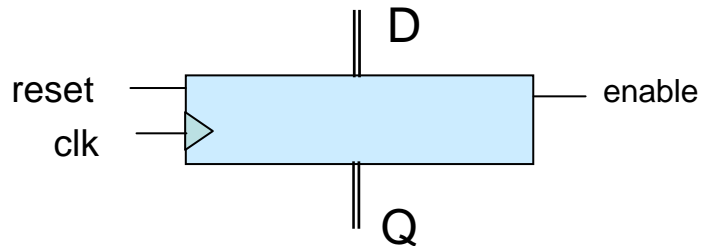
Revisão: Circuitos Sequenciais

Registradores



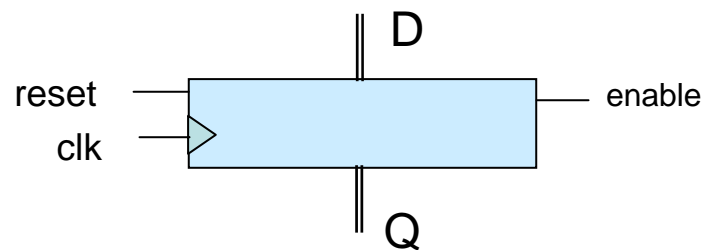
clk	D	Q
\uparrow	dado	dado
X	X	Q

Reset síncrono



clk	reset	enable	D	Q
\uparrow	1	X	X	reset
\uparrow	0	1	dado	dado
\uparrow	0	0	X	Q
X	X	X	X	Q

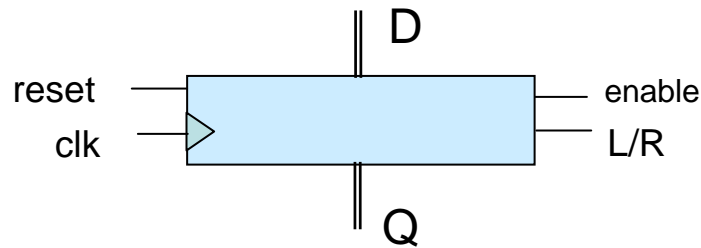
Reset assíncrono



clk	reset	enable	D	Q
X	1	X	X	reset
\uparrow	0	1	dado	dado
\uparrow	0	0	X	Q
X	0	X	X	Q

Revisão: Circuitos Sequenciais

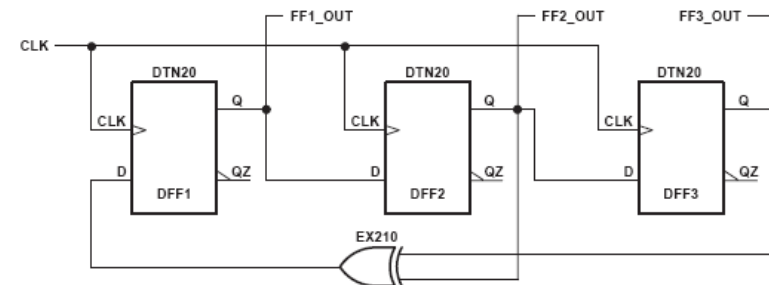
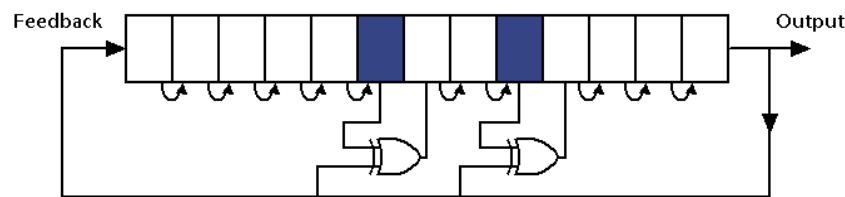
Registradores Deslocamento



clk	reset	enable	D	L/R	Q
\uparrow	1	X	X	X	reset
\uparrow	0	1	dado	0	dado
\uparrow	0	0	X	1	$Q_i \leq Q_{i-1}$
\uparrow	0	0	X	0	$Q_i \leq Q_{i+1}$
\uparrow	0	1	X	1	Q
X	X	X	X	X	Q

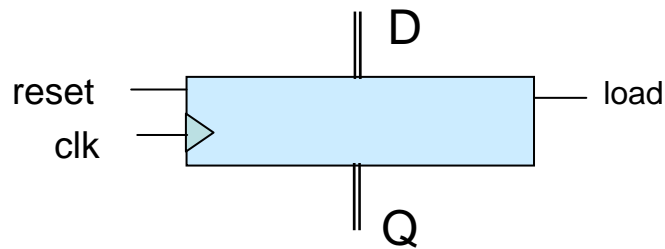
LFSR: linear feedback shift register

Uso de seed (semente)

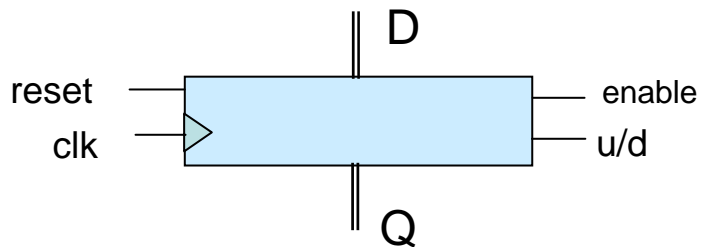


Revisão: Circuitos Sequenciais

Contadores



clk	reset	load	D	Q
\uparrow	1	X	X	reset
\uparrow	0	1	dado	dado
\uparrow	0	0	X	Q
X	X	X	X	Q



clk	reset	enable	D	u/d	Q
X	1	X	X	X	reset
\uparrow	0	0	dado	0	dado
\uparrow	0	0	X	1	Q+1
\uparrow	0	1	X	0	Q-1
\uparrow	0	1	X	1	Q
X	0	X	X	X	Q

Revisão: Máquinas de Estados

Definição

- Uma máquina de estados é uma combinação de 5 elementos:

$$(\Sigma, X, g, x_0, F)$$

Onde:

Σ é um alfabeto finito

X é um conjunto finito de estados

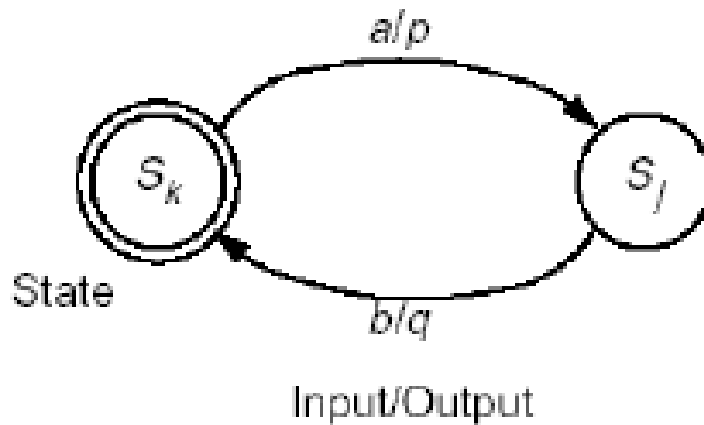
g é a função de transição de estado $g : X \times \Sigma \rightarrow X$

x_0 é o estado inicial, $x_0 \in X$

F é o conjunto de estados finais, $F \subseteq X$.

Diagrama de Estados

- O diagrama de estados representa a máquina de estados finito e contem:
 - **Circulos:** que representam os estados da máquina rotulados com o nome do estado e tambem ou não com sua codificação.
 - **Arcos diretos:** que representam as transições entre estados rotulados com entradas/saídas para a transição de estados.

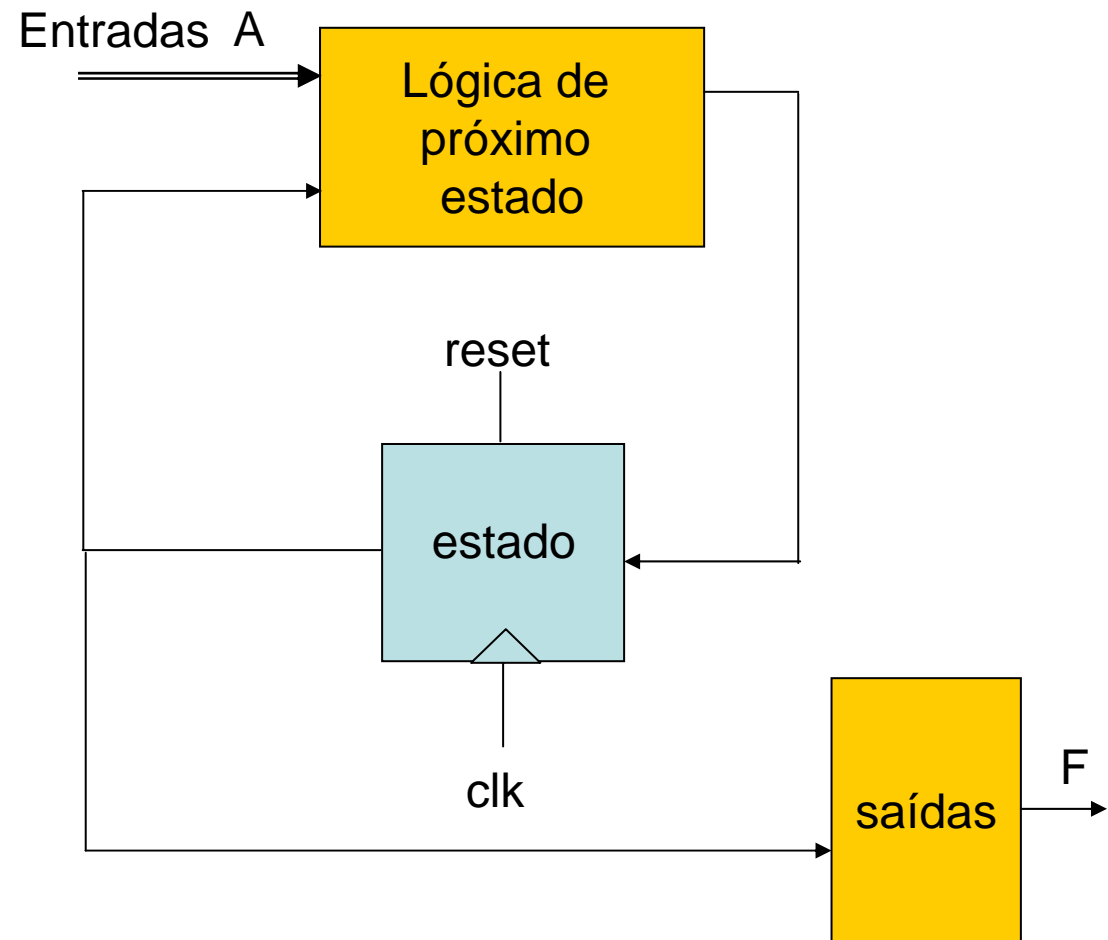
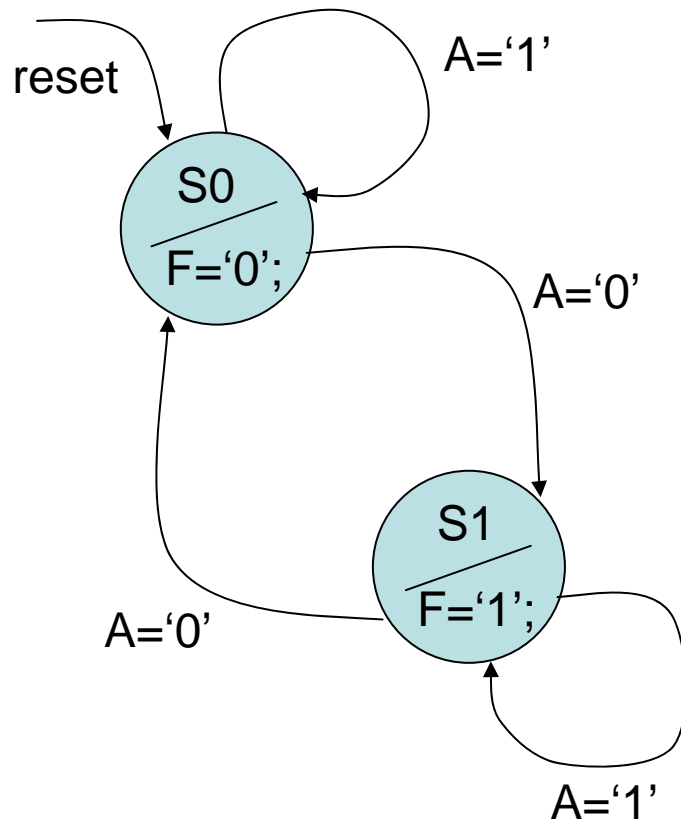


Input: $x(t) \in \{a, b\}$
Output: $z(t) \in \{p, q\}$
State: $s(t) \in \{S_k, S_j\}$
Initial state: $s(0) = S_k$

Máquina de Estados Finitos

TIPO MOORE

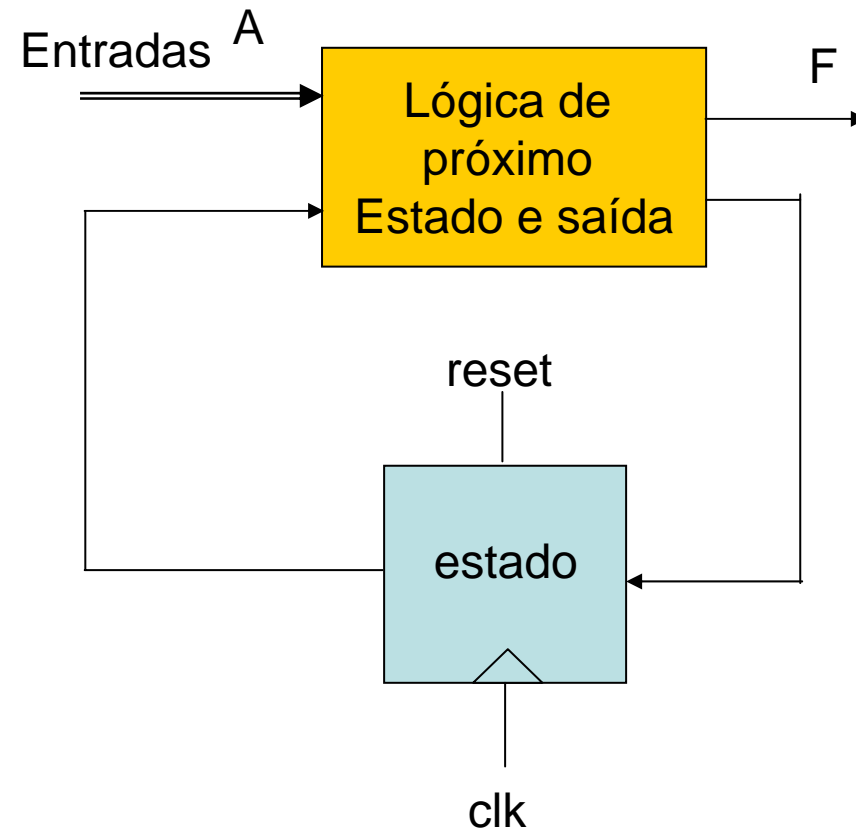
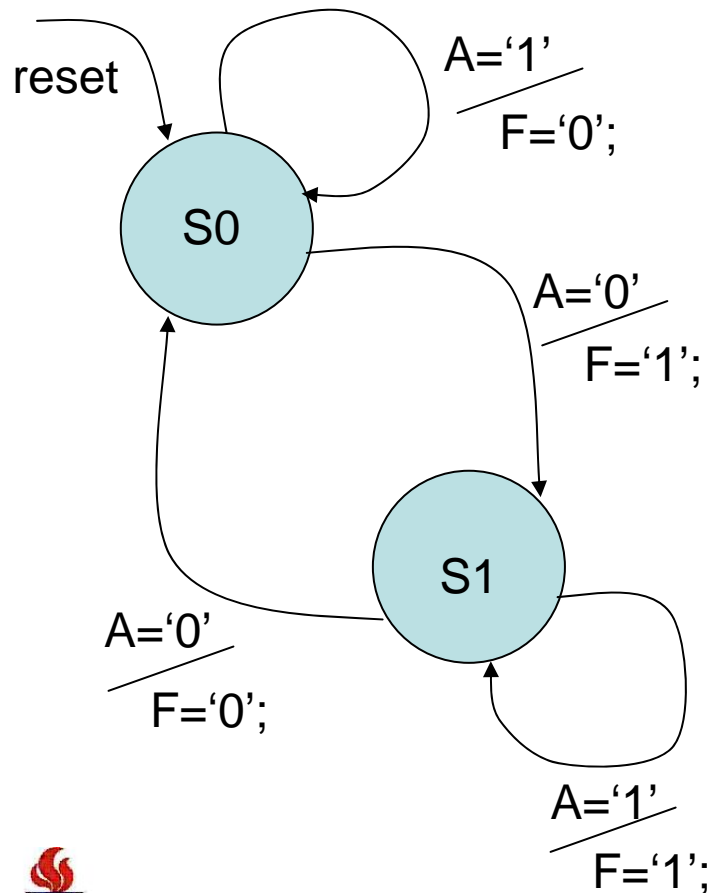
- Saída depende apenas do estado atual.



Maquina de Estados Finitos

TIPO MEALY

- Saída depende da entrada e do estado atual.

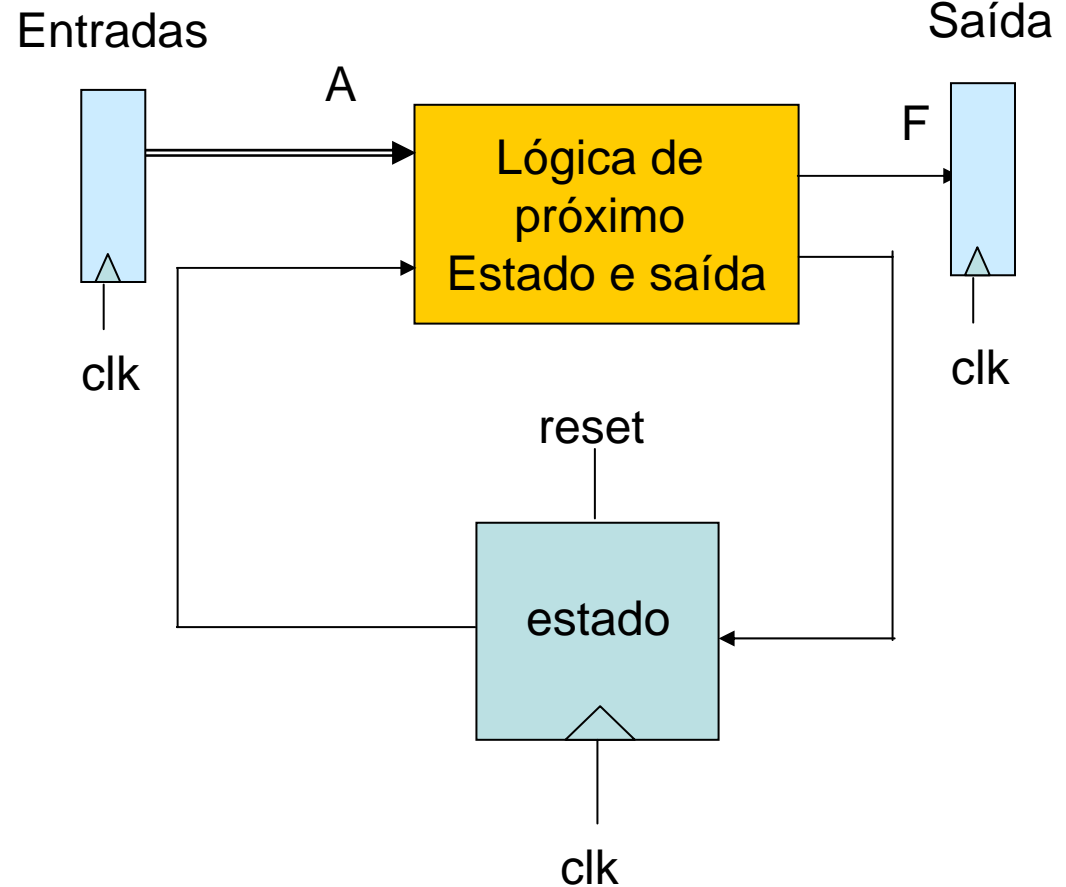
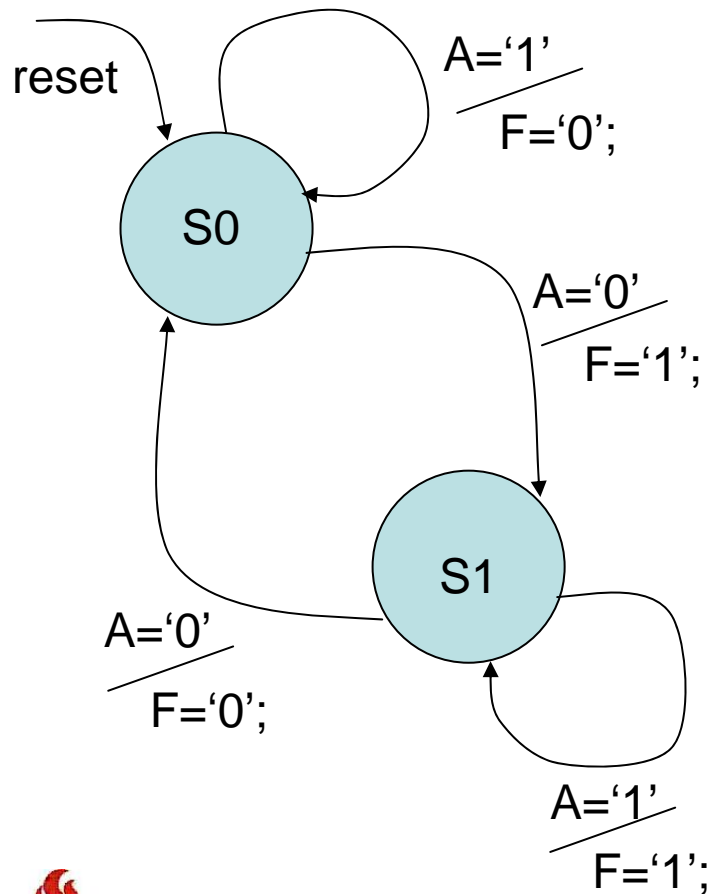


Maquina de Estados Finitos

TIPO MEALY

Solucionar problemas de estabilização

- Saída depende apenas do estado atual.



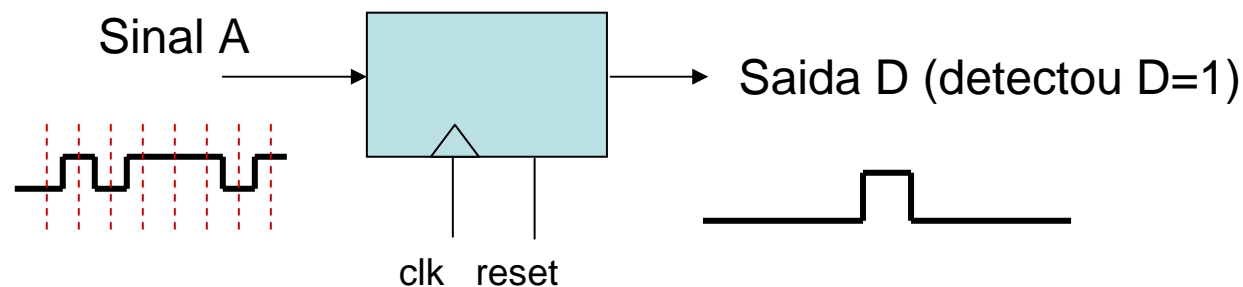
- Máquinas de estado (FSM) podem estar em apenas um estado por vez no tempo, logo há em apenas um estado ou círculo em um determinado tempo t .
- Transição de estados são permitidas apenas na transição de subida OU descida do relógio (clk), dependendo do elemento de armazenamento de estado (se é sensível a borda de descida ou subida). FSM síncronas!!!
- A representação de máquinas de Mealy e Moore são diferentes como visto.
 - Máquinas de Mealy, as entradas e saídas são definidas nos arcos (transições entre estados).
 - Máquina de Moore, as entradas são definidas nos arcos (transições entre estados) e a saída é definida no estado (dentro do círculo).

Exercício: Projeto de uma máquina de estados até o hardware

Aula
1

1. Descrição do problema em diagrama de estados
2. Montagem da tabela de proximo estados e saída
3. Descrição do esquemático lógico da FSM

Problema: Detector de sequência ...1101...



Sinal de entrada deve ser amostrado...