

IC – Introdução a Computação

Maquinas de Estado Finito



Os circuitos sequenciais podem ser modelados como sendo uma maquina de estados finitos.

- Finite State Machine – FSM
- Maquinas de estados - MdE

Para um circuito sequencial com n variáveis de estado, temos 2^n máximos estados possíveis.

Uma máquina com $k+1$ variáveis de estado, $p+1$ entradas e $n+1$ saídas:

E_k, \dots, E_1, E_0 : são variáveis de estado atual (vetor E);

$E_k^+, \dots, E_1^+, E_0^+$: são variáveis de estado próximo (vetor E^+);

X_p, \dots, X_1, X_0 : são variáveis de entrada (vetor X);

Y_n, \dots, Y_1, Y_0 : são variáveis de saída (vetor Y);

Maquinas de Estado Finito



Funções combinacionais de Transformação

- Internamente o circuito sequencial deve:
 - Definir, em função de estado (E) atual e/ou entradas (x), qual é a saída (y);

$$y = \alpha(x, E)$$

- Definir, em função de estado (E) atual e das entradas (x), qual deve ser o próximo estado do circuito sequencial (E^+);

$$E^+ = \beta(x, E)$$

- No caso α e β são funções lógicas combinacionais.

Maquinas de Estado Finito



Funções combinacionais de Transformação

- **Funções de Armazenamento (Registro) de Estados:**

- O circuito sequencial deve armazenar (registrar) as variáveis de estado atual, para que essas possam ser utilizadas pelas funções combinacionais de transformação;
- Periodicamente, o circuito sequencial atualiza as suas variáveis de estado (variáveis de estado próximo → variáveis de estado atual);
- A atualização pode ser:
 - Natural;
 - Forçada.

Maquinas de Estado Finito

Voltando a máquina de lavar:

Temos 4 estados, então precisamos de 2 variáveis de estado.

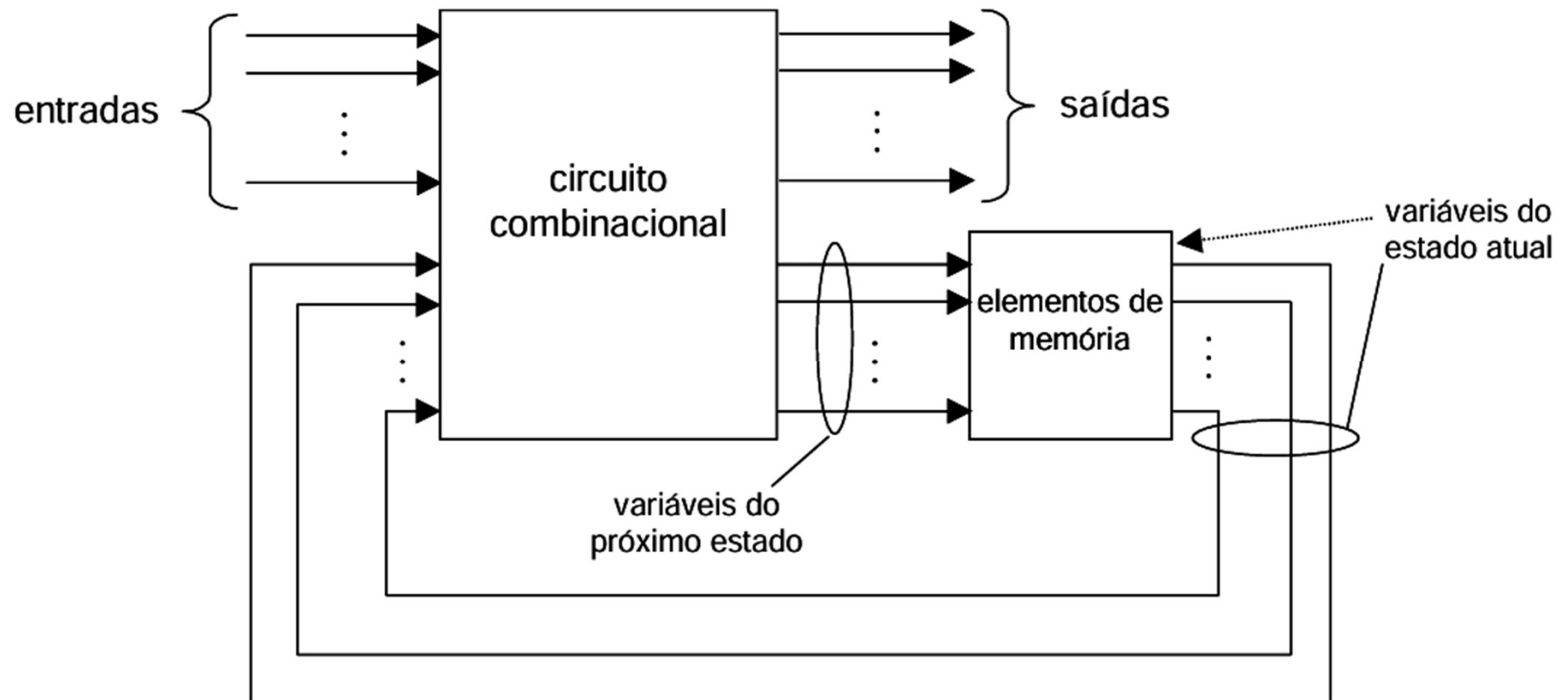
Vamos denominar em nosso curso de E_1E_0

- ❖ **Molho** - Seleção $(E_1E_0) = 00$
- ❖ **Lavar** - Seleção $(E_1E_0) = 01$
- ❖ **Enxaguar** - Seleção $(E_1E_0) = 10$
- ❖ **Centrifugar** - Seleção $(E_1E_0) = 11$



Maquinas de Estado Finito

Arquitetura de um Circuito Sequencial (Arquitetura de Huffman):



Maquinas de Estado Finito



Máquina de Moore x Mealy

☐ Moore

- Saídas dependem apenas do estado atual (ou seja das entradas)
- As saídas apenas são escritas quando os estados variam (transições de estados são síncronos)

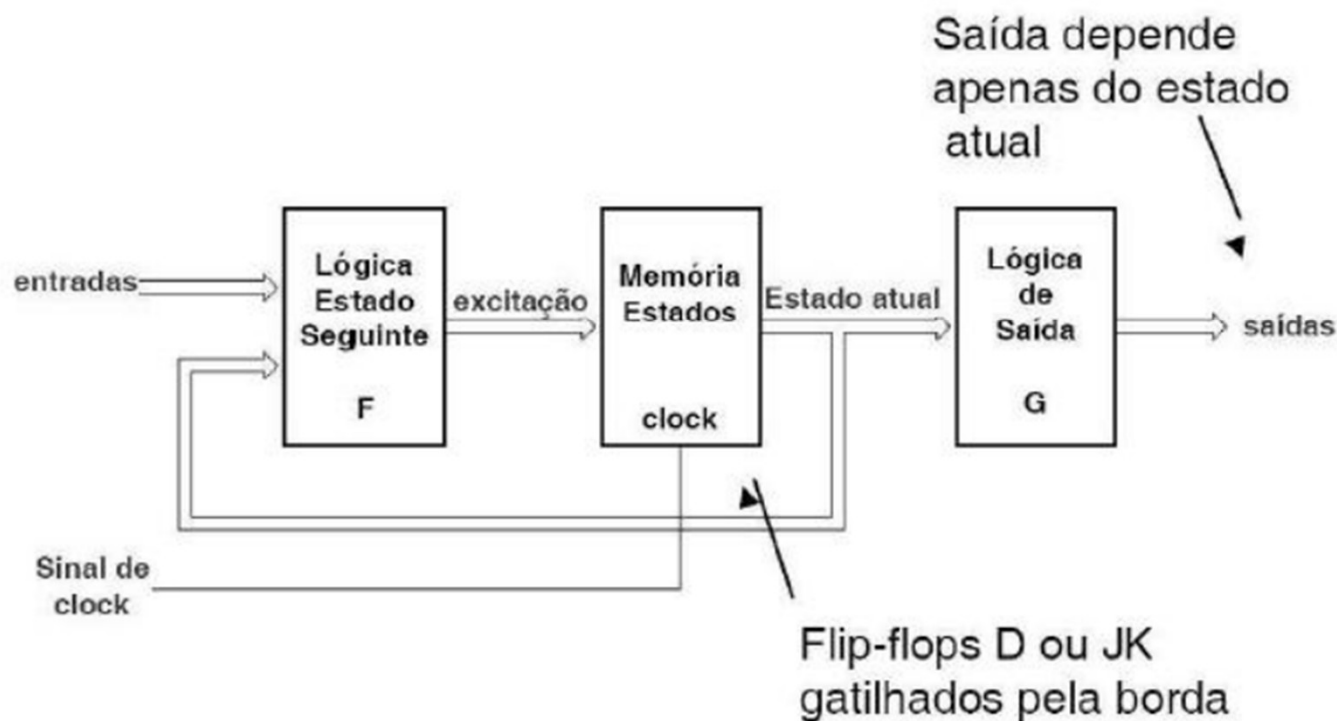
☐ Mealy

- As saídas dependem de ambos: entradas e estados
- Quando a entrada muda, as saídas são atualizadas imediatamente, sem esperar por clock

Maquinas de Estado Finito

Máquina de Moore x Mealy

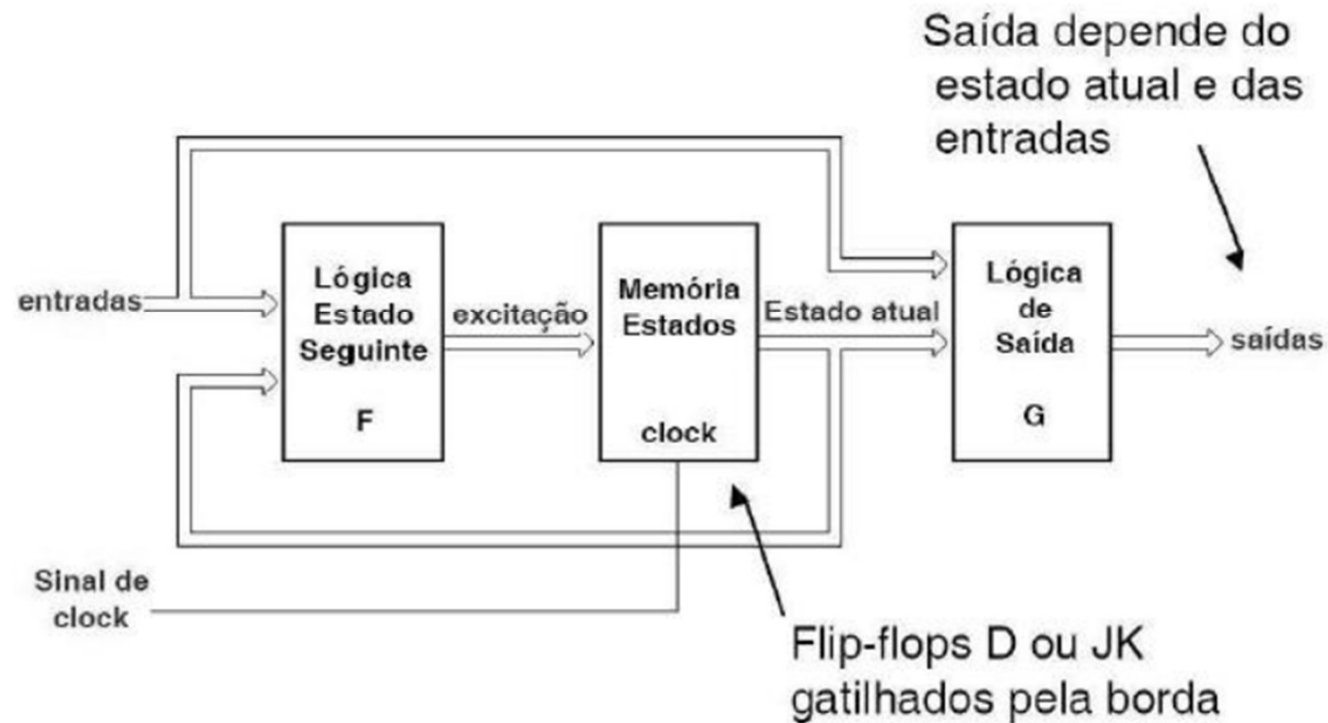
☐ Moore



Maquinas de Estado Finito

Máquina de Moore x Mealy

❑ Mealy



Maquinas de Estado Finito



Ferramenta de descrição de Máquinas de estado.

Existem duas ferramentas:

- Tabela de Transição
- Diagrama de transição

Vamos começar pela Tabela de transição entre estados.

Indicam a relação entre o estado atual, a entrada, o estado próximo e as saídas da máquina de estados em forma de tabela.

Exemplo:

Estado Atual	Estado próximo/Saida para	
	x=0	x=1
S ₀	S ₀ /0	S ₁ /0
S ₁	S ₁ /0	S ₂ /0
S ₂	S ₂ /0	S ₃ /0
S ₃	S ₃ /1	S ₀ /1

Maquinas de Estado Finito

Diagrama de Transição entre estados:

Maneira gráfica de representar a tabela de transições de estado.

Os estados são representados por círculos e as transições entre os estados são indicados por flechas.

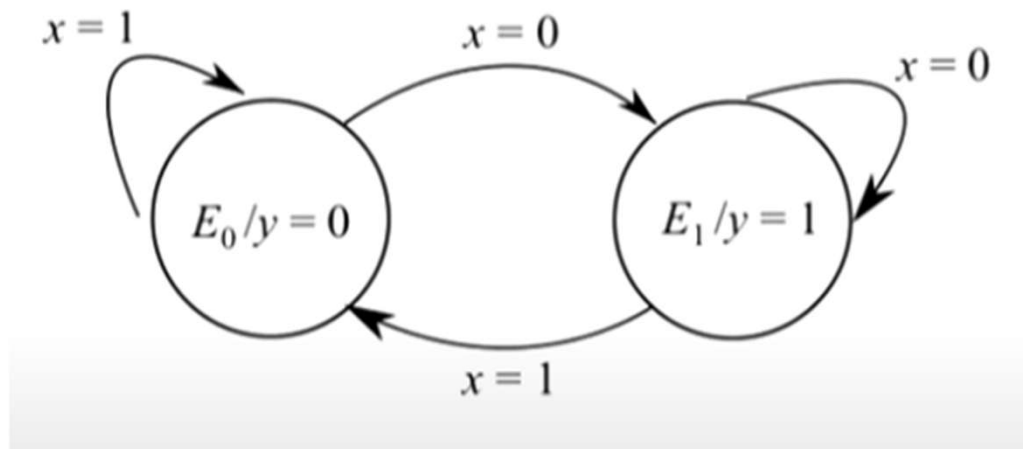


Diagrama de transição para máquina de Moore.

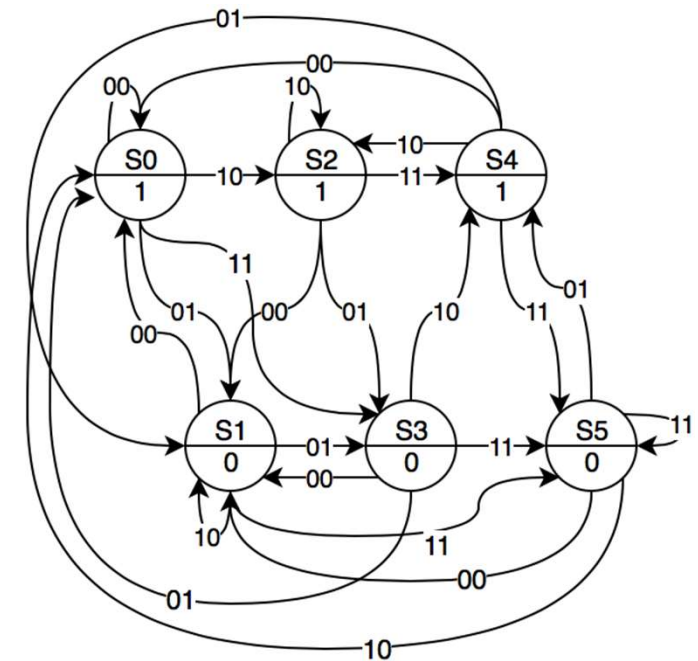


Diagrama de transição para máquina de Moore 2 bits.

Maquinas de Estado Finito

Diagrama de Transição entre estados:

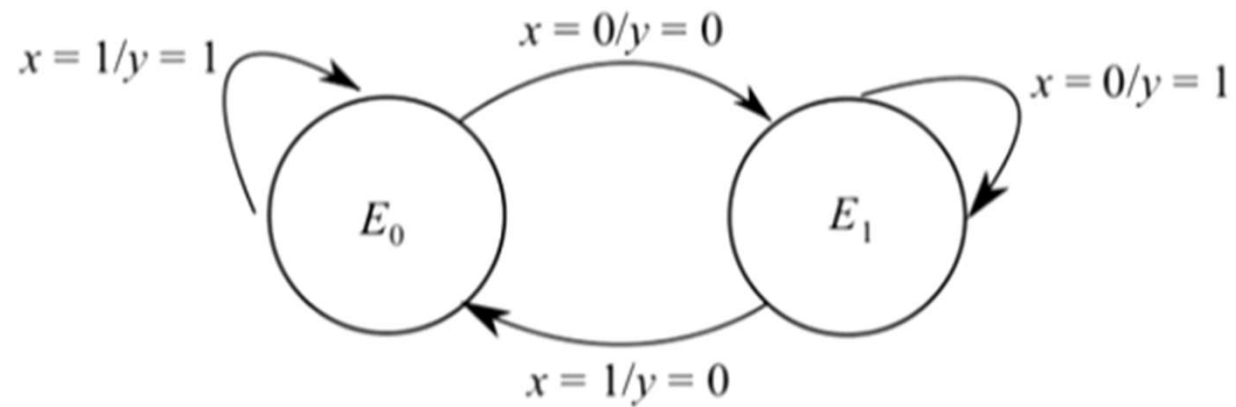


Diagrama de transição para máquina de Mealy.

Maquinas de Estado Finito

Vamos fazer um exercícios:

Converta a seguinte tabela de transição de estados em um diagrama de transição de estados

Estado Atual	Estado próximo		Saída
	x=0	x=1	
A	A	B	0
B	B	C	0
C	C	D	0
D	D	A	1

Elementos de Memória

Arquitetura de Huffman para circuitos sequenciais.

O que é um elemento de memória?

- Basicamente ele armazena uma variável de estado (armazena 1 bit) 0 ou 1.

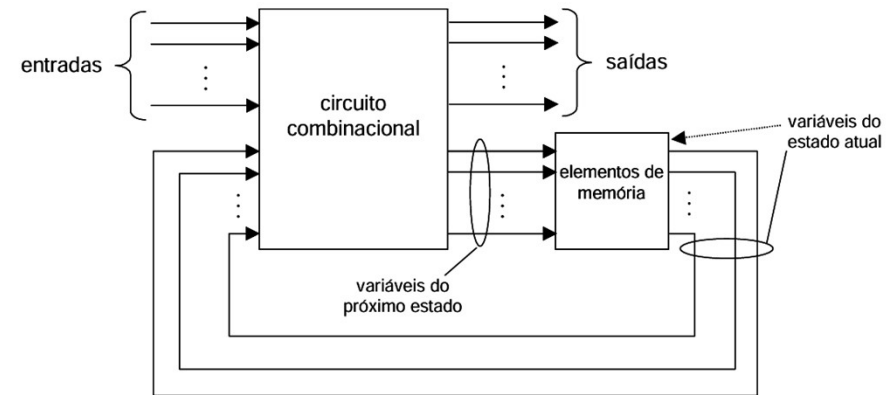
Quais são nossos elementos de memória?

- Latches
- Flip-Flops

➤ Latch:

- Latch normais: Muda os valores armazenados toda vez que suas entradas mudam;
- Latch controlado: Sensível a um nível de sinal de controle;

➤ Flip-Flops: Sensível a uma borda de um sinal de controle;



Elementos de Memória



Latches

1bit de memória ou 1 célula de memória.

O componente mais simples de memória é chamado de Latch RS ou SR. Esse componente é um dispositivo biestável de um bit que possui três entradas, uma que “SET” o dispositivo (coloca a saída como “1”) conhecida como “S”, uma entrada que ira “RESET” (colocar a saída como “0”) rotulada como “R”. Portanto a descrição SR, significa Set-Reset.

Reset: redefine o estado lógico da saída para 0

Set: redefine o estado lógico da saída para 1

E a última entrada é chamada de redefinição, converte de volta ao seu estado original com a saída Q que estará no nível lógico “1” ou lógico “0”, dependendo da condição de ajuste/redefinição.

Portanto, o Latch SR possui três entradas, Set, Reset e sua saída atual Q, relacionada ao seu estado ou histórico atual.

Elementos de Memória

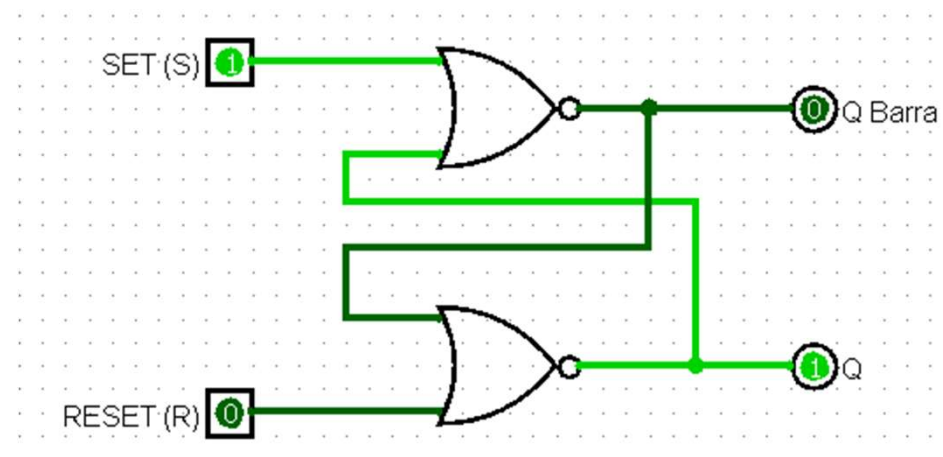
Latch SR - Resumindo

Possui duas entradas, SET e RESET, que podem mudar os valores armazenados em Q e Q`.

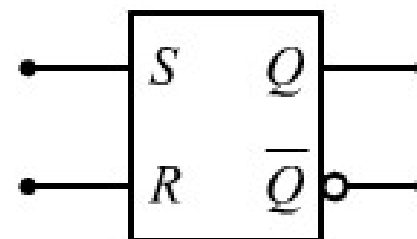
Estados possíveis:

- ☐ Estado SET: $Q = 1$ e $Q' = 0$;
- ☐ Estado RESET $Q = 0$ e $Q' = 1$;

Nosso circuito



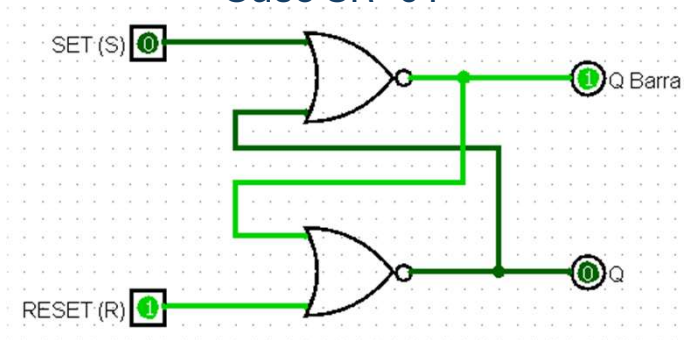
Símbolo Esquemático



Elementos de Memória

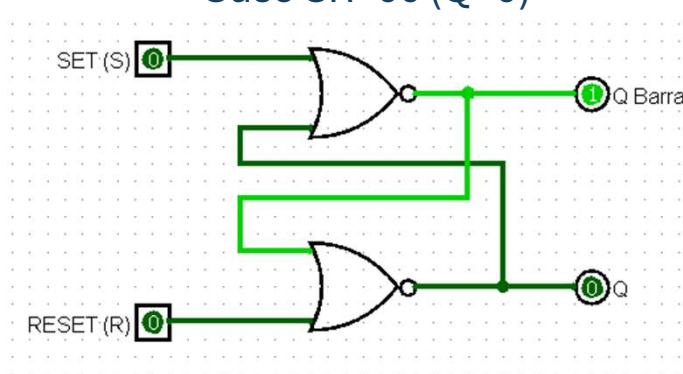
Analizando o LATCH SR

Caso SR=01



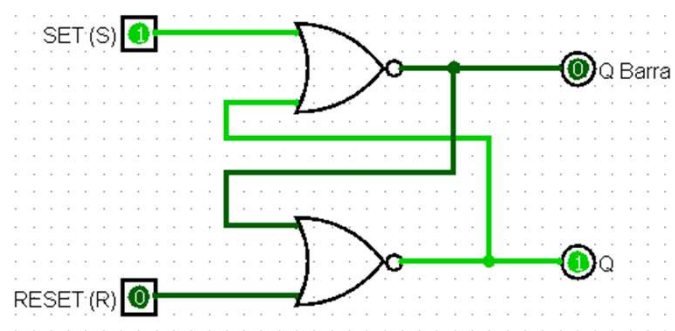
Estado de Reset

Caso SR=00 (Q=0)



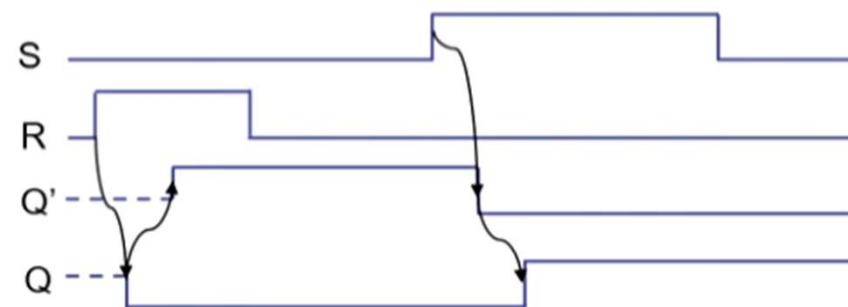
Mantém o estado de reset

Caso SR=10



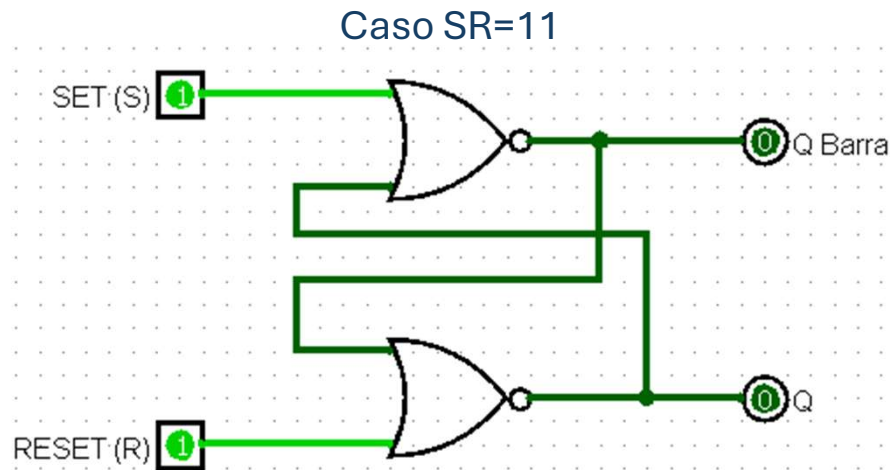
Estado de SET

Diagrama de Temporização



Elementos de Memória

Analizando o LATCH SR



Estado Impossível ou indesejado

Resumindo:

- ❑ R=1, S=0 faz com que o Latch vá para o estado de RESET;
- ❑ R=0, S=1 faz com que o Latch vá para o estado de SET;
- ❑ R=0, S=0 faz com que o Latch não mude o seu estado;
- ❑ R=1, S=1 é uma situação proibida;

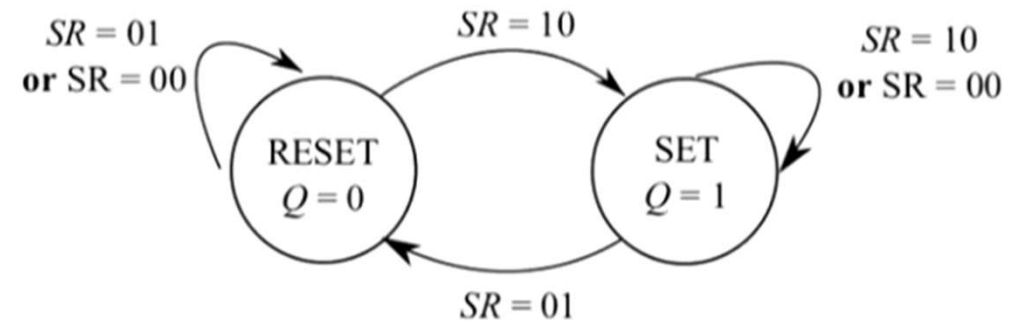
Elementos de Memória

Analizando o LATCH SR

Tabela verdade do Latch SR

S	R	Estado do Latch	Descrição
0	0	Sem alteração	Mantém o estado anterior
0	1	0 (RESET)	Força a saída para 0
1	0	1 (SET)	Força a saída para 1
1	1	Indeterminado (inválido)	Estado proibido (condição inválida)

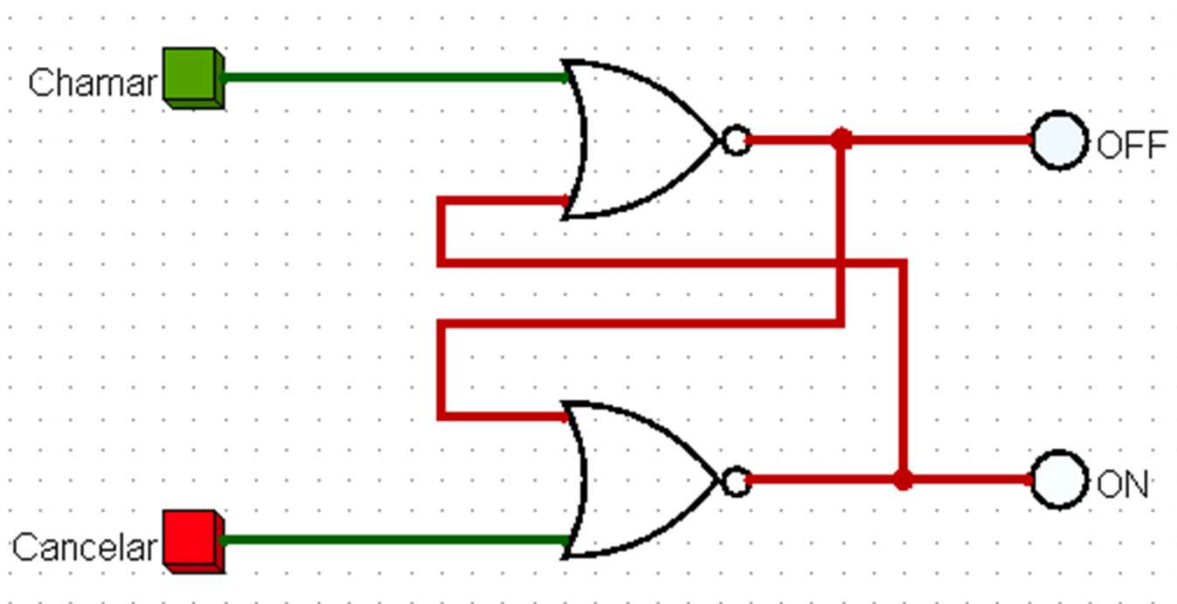
Diagrama de Transição de estados



Elementos de Memória

Um exemplo de aplicação:

Em um avião para chamar um comissário de bordo



**ATÉ A
PRÓXIMA
AULA!**



Bibliografia



TOCCI, R.; WIDMER, N.; MOSS, G. Sistemas Digitais – Princípios e Aplicações. [S.l.]: Pearson Education Limited, 2011.

FEDELI, Ricardo Daniel. Introdução à ciência da computação / Ricardo Daniel Fedeli, Erico Giulio Franco Polloni, Fernando Eduardo Peres. – 2. ed. – São Paulo: Cengage Learning, 2011.

TANENBAUM, Andrew S.. Organização Estruturada de Computadores. 6º Edição. São Paulo, Pearson Prentice Hall, 2013.



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).