

PCS 3225

Sistemas Digitais II

Módulo 03 – ASM – *Algorithm State Machine*

Andrade, Marco Túlio Carvalho de
Professor Responsável

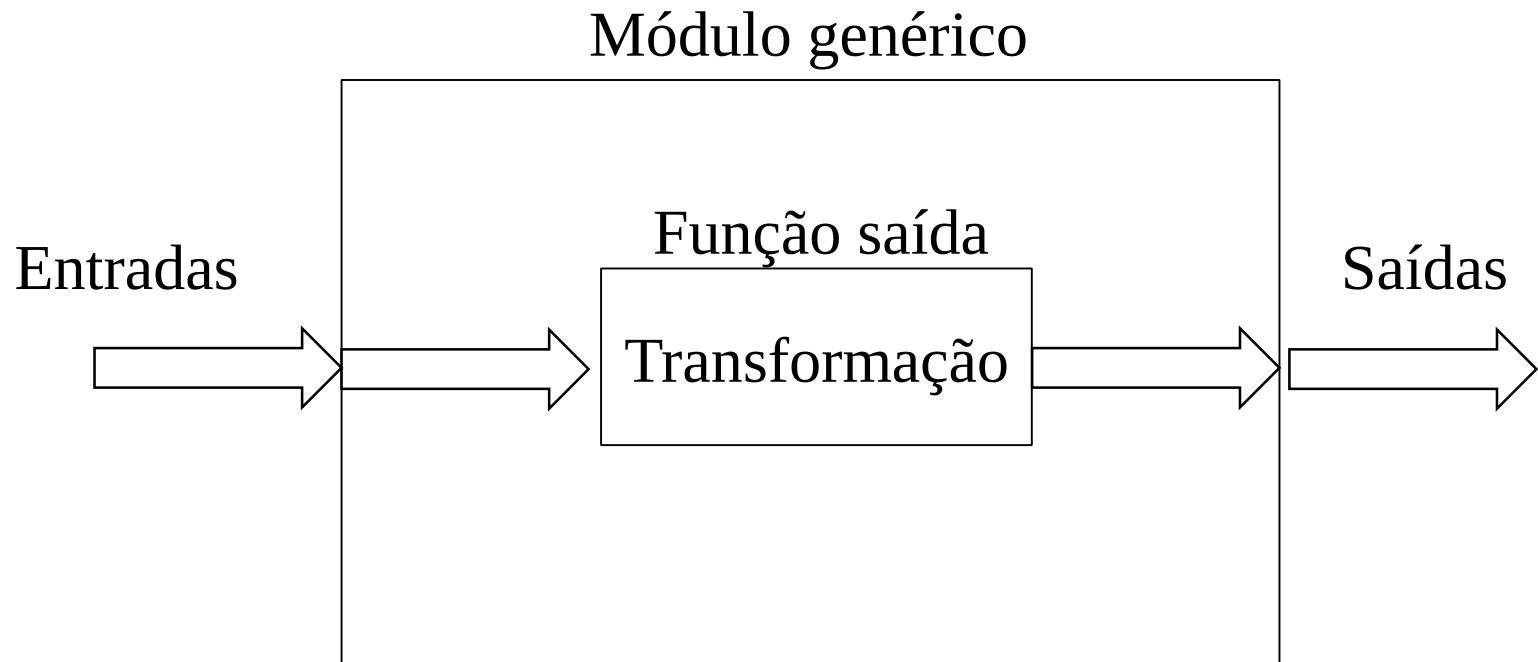
Adaptado por Glauber (2018)

Conteúdo

- 1. Máquinas de Estado – Introdução
- 2. Máquinas de Estado e Circuitos Digitais
- 3. Diagrama ASM – Algorithmic State Machine
 - Introdução
 - ASM – Mapeamento em Portas e Biestáveis
 - 3.1. Diagrama ASM – Exemplos
 - 3.2. ASM e Máquinas de Estado

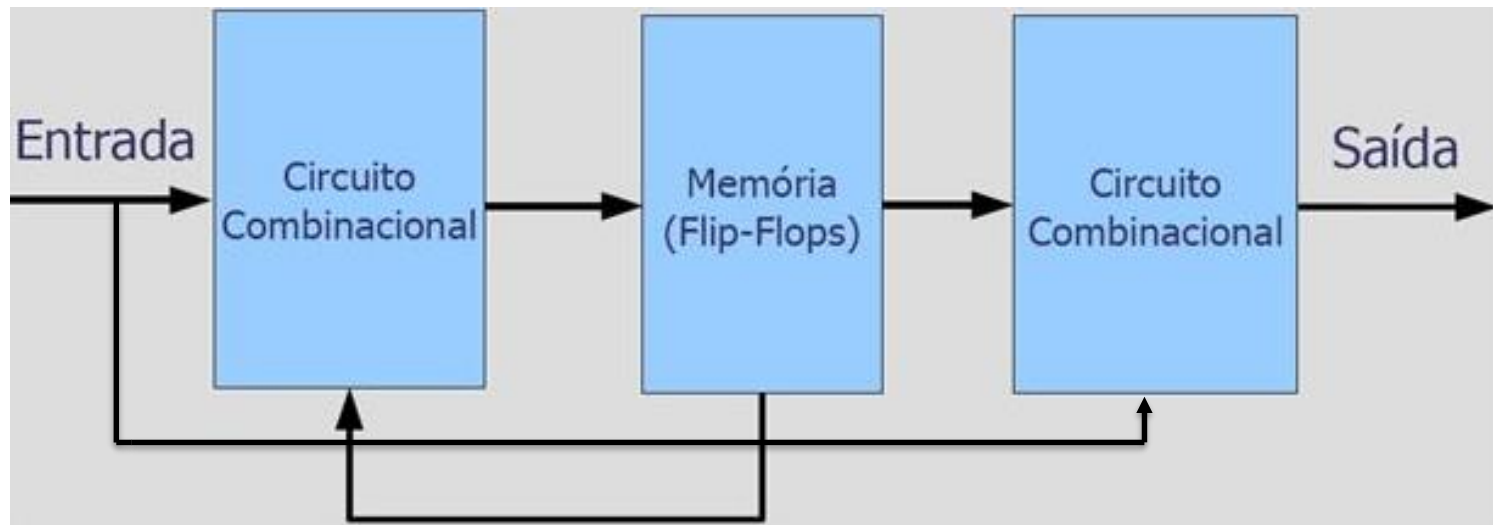
1. Máquinas de Estado – Introdução

■ Circuitos combinatórios: Entradas; Saídas.



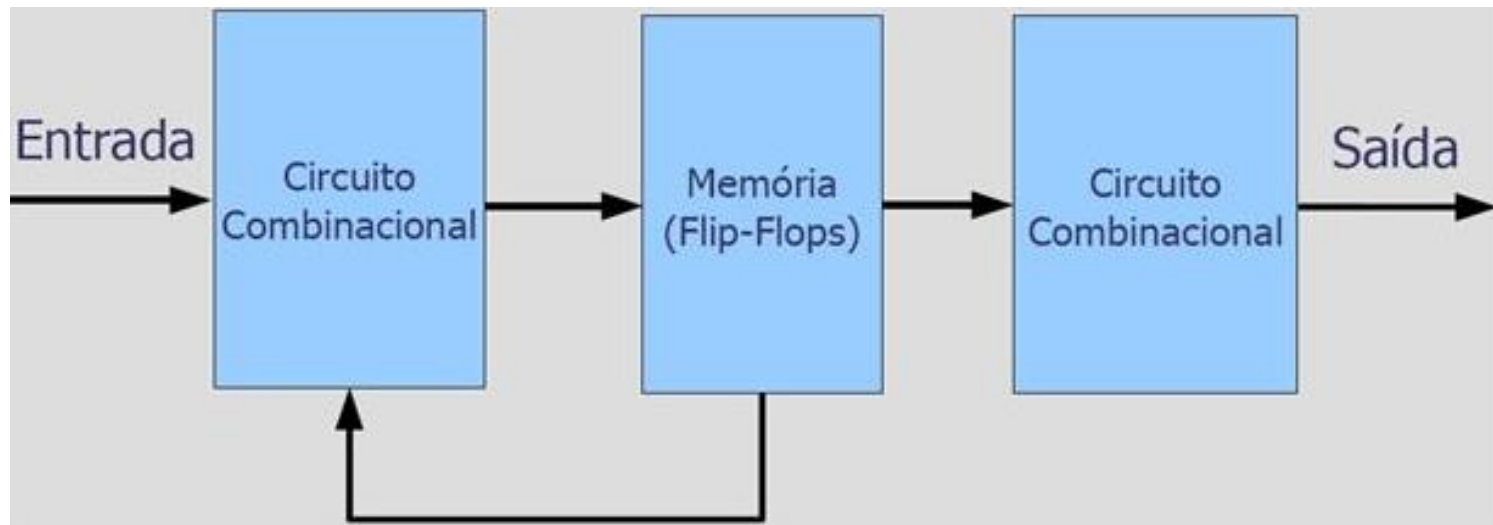
1. Máquinas de Estado – Introdução

- Circuitos sequenciais: Entradas, Saídas e memória.
- **Máquina de Mealy:** Saída depende do conteúdo da memória e das entradas.



1. Máquinas de Estado – Introdução

- Circuitos sequenciais: Entradas, Saídas e Memória.
- **Máquina de Moore:** saída só depende do conteúdo da memória.



1. Máquinas de Estado – Introdução

- Circuitos sequenciais **síncronos**: só interessa o comportamento em determinados instantes de tempo (ticks do clock), tempo é abstraído como discreto.
- No tempo t , as entradas e o conteúdo da memória determinam a saída no tempo t e o conteúdo da memória no tempo $t+1$.
- A memória é finita, e seu conteúdo (um conjunto de bits) é abstraído para um **estado**.

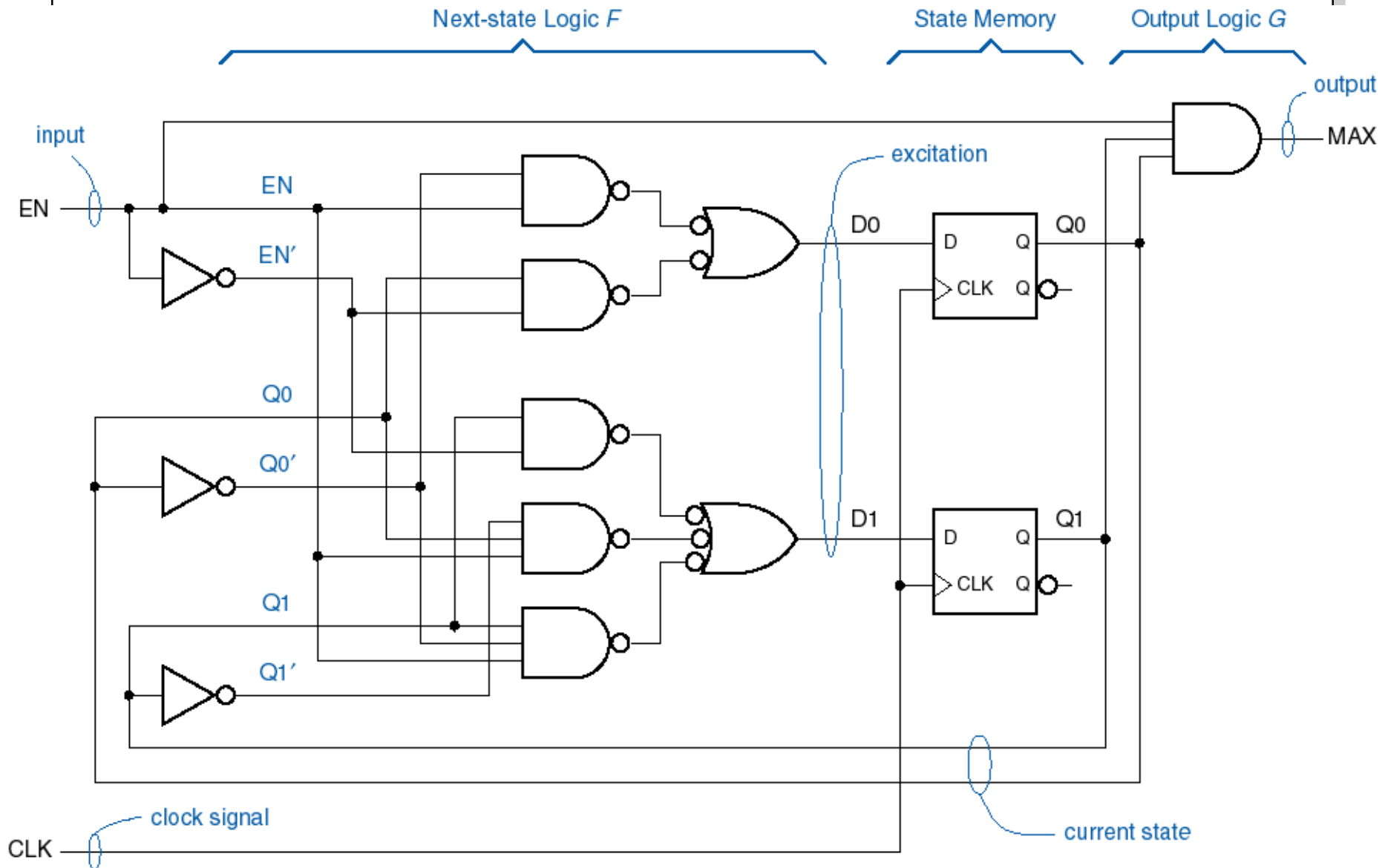
1. Máquinas de Estado – Introdução

- **Estado:** é a memória de suficiente história do passado que permite determinar o comportamento futuro, i.é., conhecidas as entradas e o estado atual permite determinar:
 - Saídas;
 - Próximo estado (ou p. estado).
- Em sistemas digitais, o estado é codificado em binário, implementado com um conjunto de flip-flops (registrador de estados):
 - Flip-flops do Estado – Variáveis de Estado.

1. Máquinas de Estado – Introdução

- O próximo Estado é função do Estado atual e das entradas.
- O próximo Estado (codificado em binário) determina o próximo valor (estado) de cada variável de estado (bit 0 ou 1).
- O próximo estado no tempo t (um tick no clock) será o estado atual no tempo $t+1$ (no próximo tick).

1. Máquinas de Estado – Introdução



1. Máquinas de Estado – Introdução

■ **Máquina de Estado Finita (FSM):**

Formalismo matemático que descreve a sequência de estados e saídas determinada por uma sequência de entradas.

Discretizando o tempo, **qualquer circuito sequencial síncrono** pode ser modelado como uma **FSM!**

1. Máquinas de Estado – Introdução

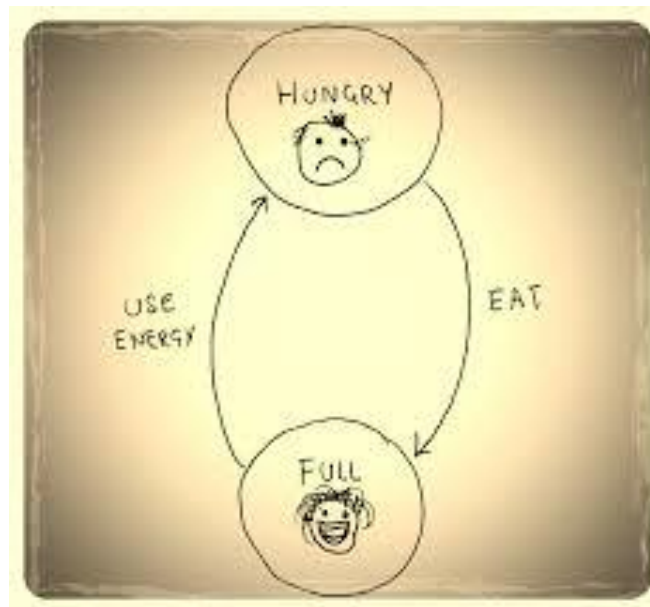
- Por que Máquina de Estados, “**finita**”?
 - Resposta: Porque o **número de Estados**, do sistema representado, é **finito**.
- Alguns autores usam a denominação “Máquina de Estados Finitos” com a expressão “Estados Finitos” sendo entendida como um **número finito de Estados**.

1. Máquinas de Estado – Introdução

■ Representação via diagramas:

- Nós (círculos) são estados, arcos (setas) são transições.

Exemplo:



1. Máquinas de Estado – Introdução

- **Definição:** Uma **Máquina de Estados Finitos** é uma tupla $\mathbf{M} = (I, O, S, f, g, \sigma)$ onde:
 - Um conjunto finito I de símbolos de entrada;
 - Um conjunto finito O de símbolos de saída;
 - Um conjunto finito S de estados;
 - Uma função próximo estado $f: S \times I \rightarrow S$;
 - Uma função de saída $g: S \times I \rightarrow O$;
 - Um estado inicial $\sigma \in S$.

1. Máquinas de Estado – Introdução

- **Ex:** $I=\{a,b\}$, $O=\{0,1\}$, $S=\{\sigma_0, \sigma_1\}$ e $\sigma = \sigma_0$. As funções f e g são definidas pelas regras dadas na **Tabela (de Transição) de Estados**

$f(S_t, I_t) = S_{t+1}$ é o próximo estado

$g(S_t, I_t) = O_t$ é a saída

$S_t \backslash I_t$	f		g	
	a	b	a	b
σ_0	σ_0	σ_1	0	1
σ_1	σ_1	σ_1	1	0

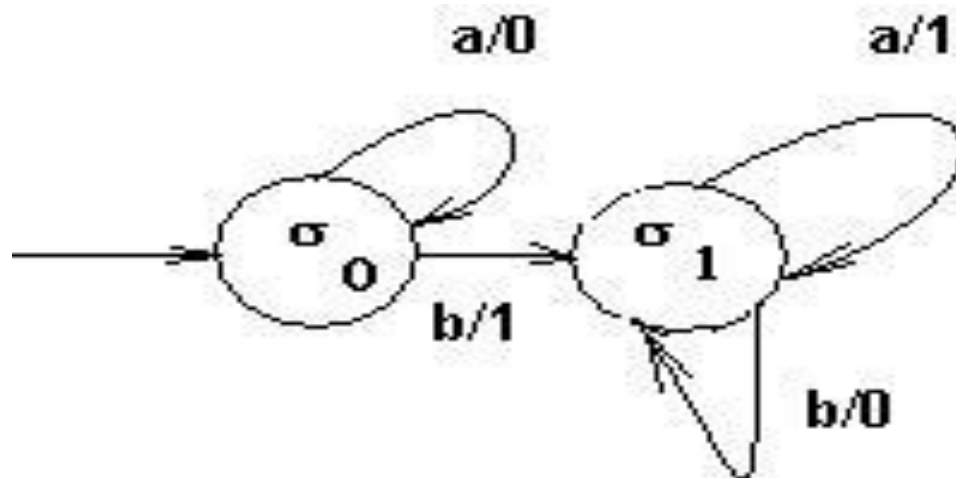


$$\begin{array}{ll} f(\sigma_0, a) = \sigma_0 & g(\sigma_0, a) = 0 \\ f(\sigma_0, b) = \sigma_1 & g(\sigma_0, b) = 1 \\ f(\sigma_1, a) = \sigma_1 & g(\sigma_1, a) = 1 \\ f(\sigma_1, b) = \sigma_1 & g(\sigma_1, b) = 0 \end{array}$$

1. Máquinas de Estado – Introdução

- **Definição:** Seja $M = (I, O, S, f, g, \sigma)$ uma FSM. O **Diagrama (de Transição) de Estados** de M é um digrafo onde há um nó para cada estado em S , e arcos de S_j para S_k sse $f(S_j, I_t) = S_k$ para algum I_t em I . Tais arcos são rotulados com a entrada e saída correspondentes: $I_t/O(S_j, I_t)$. Uma seta indica o estado inicial.

		f		g	
$S_t \backslash I_t$		a	b	a	b
σ_0		σ_0	σ_1	0	1
σ_1		σ_1	σ_1	1	0



1. Máquinas de Estado – Introdução

- Da **Tabela de Transição de Estados** de um circuito sequencial (mais o estado inicial) pode-se extrair seu **Diagrama de Transição de Estados**, e vice-versa.

1. Máquinas de Estado – Introdução

- Seja $M = (I, O, S, f, g, \sigma)$ uma FSM. A cada cadeia (sequência de símbolos) de entrada $x_1 \dots x_n$ sobre I **corresponde** uma cadeia de saída $y_1 \dots y_n$ sobre O tal que existem estados

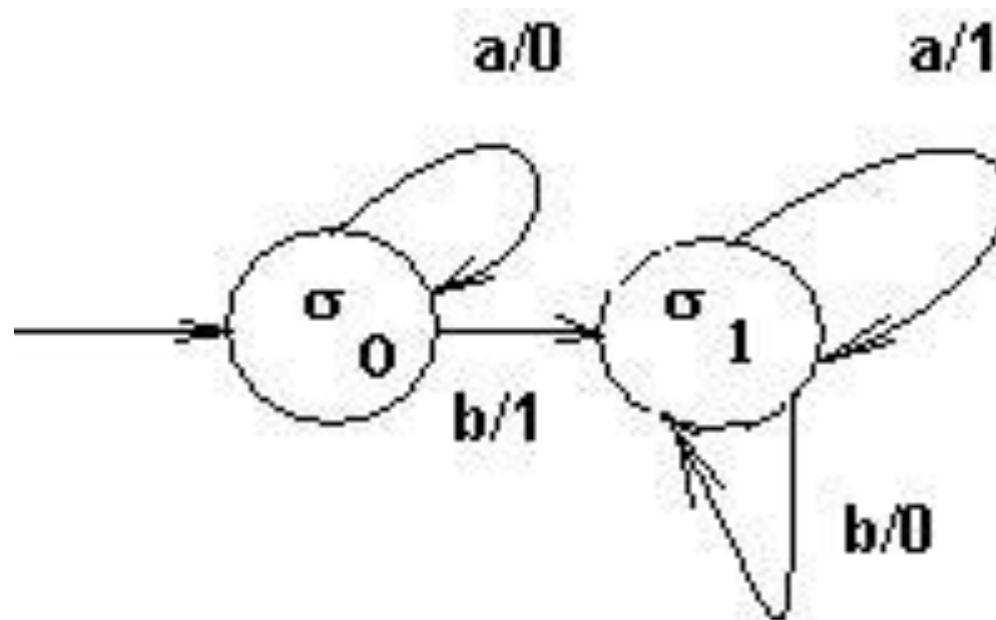
$\sigma_0, \sigma_1, \dots, \sigma_n \in S$ com:

$$\sigma_0 = \sigma, \quad \sigma_i = f(\sigma_{i-1}, x_i), \quad y_i = g(\sigma_{i-1}, x_i) \quad \text{para } i=1, \dots, n;$$

Olhando para o diagrama de estados, $y_1 \dots y_n$ corresponde a $x_1 \dots x_n$ se, partindo do estado inicial, há uma caminho de arcos rotulados com $x_1/y_1, \dots, x_n/y_n$

1. Máquinas de Estado – Introdução

- **Exemplo:** Encontre a cadeia de saída correspondente à cadeia de entrada *aababba* para a FSM abaixo:



1. Máquinas de Estado – Introdução

- **Tabela de Transição de Estados** de um circuito sequencial – dados tabelados facilitam visualizar o comportamento do sistema (estado futuro e saída) em um dado estado para uma dada entrada.
- **Diagrama de Transição de Estados**, de um circuito sequencial – Representação gráfica que facilita a visualização da sequência de passos (transições de estado e saídas).

1. Máquinas de Estado – Introdução

- **Tabela de Transição de Estados** de um circuito sequencial do tipo *Mealy*.

	$S_{t+1} = f(S_t, I_t)$		$O = g(S_t, I_t)$	
$I_t \backslash S_t$	a	b	a	b
σ_0	σ_0	σ_1	0	1
σ_1	σ_1	σ_1	1	0

1. Máquinas de Estado – Introdução

- **Tabela de Transição de Estados** de um circuito sequencial do tipo *Mealy* – Representação alternativa:

		$S_{t+1}/O = f(S_t, I_t)/g(S_t, I_t)$	
$S_t \backslash I_t$		a	b
σ_0		$\sigma_0/0$	$\sigma_1/1$
σ_1		$\sigma_1/1$	$\sigma_1/0$

1. Máquinas de Estado – Introdução

■ Diagrama de Transição de Estados de um circuito sequencial do tipo *Mealy*.

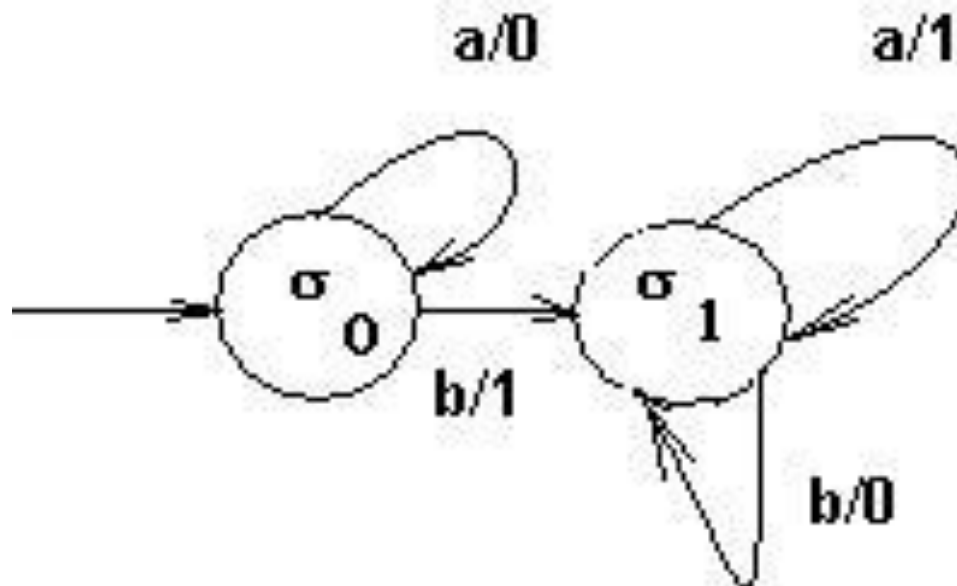
$$S_{t+1}/O_t = f(S_t, I_t)/g(S_t, I_t)$$

Rótulos nos arcos:

Input/Output = I_t/O_t

Rótulos no nós:

EstadoAtual = S_t



1. Máquinas de Estado – Introdução

- **Tabela de Transição de Estados** de um circuito sequencial do tipo *Moore*.
- Saídas são iguais para todas entradas.

		$S_{t+1} = f(S_t, I_t)$		$O_t = g(S_t)$	
$S_t \backslash I_t$		a	b	a	b
σ		σ	A	0	0
A		A	B	1	1
B		B	A	0	0

1. Máquinas de Estado – Introdução

- **Tabela de Transição de Estados** de um circuito sequencial do tipo *Moore*.
- Saídas são iguais para todas entradas.

		$S_{t+1} = f(S_t, I_t)$		
S_t	I_t	a	b	$O_t = g(S_t)$
σ		σ	A	0
A		A	B	1
B		B	A	0

1. Máquinas de Estado – Introdução

- **Diagrama de Transição de Estados**, de um circuito sequencial do tipo *Moore*.

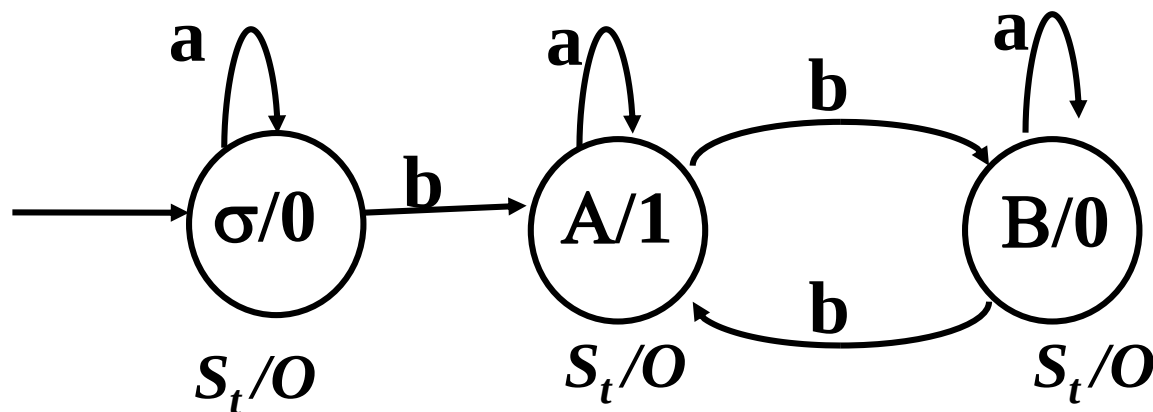
$$S_{t+1}/O_t = f(S_t, I_t)/g(S_t)$$

Rótulos nos Arcos:

$$Input = I_t$$

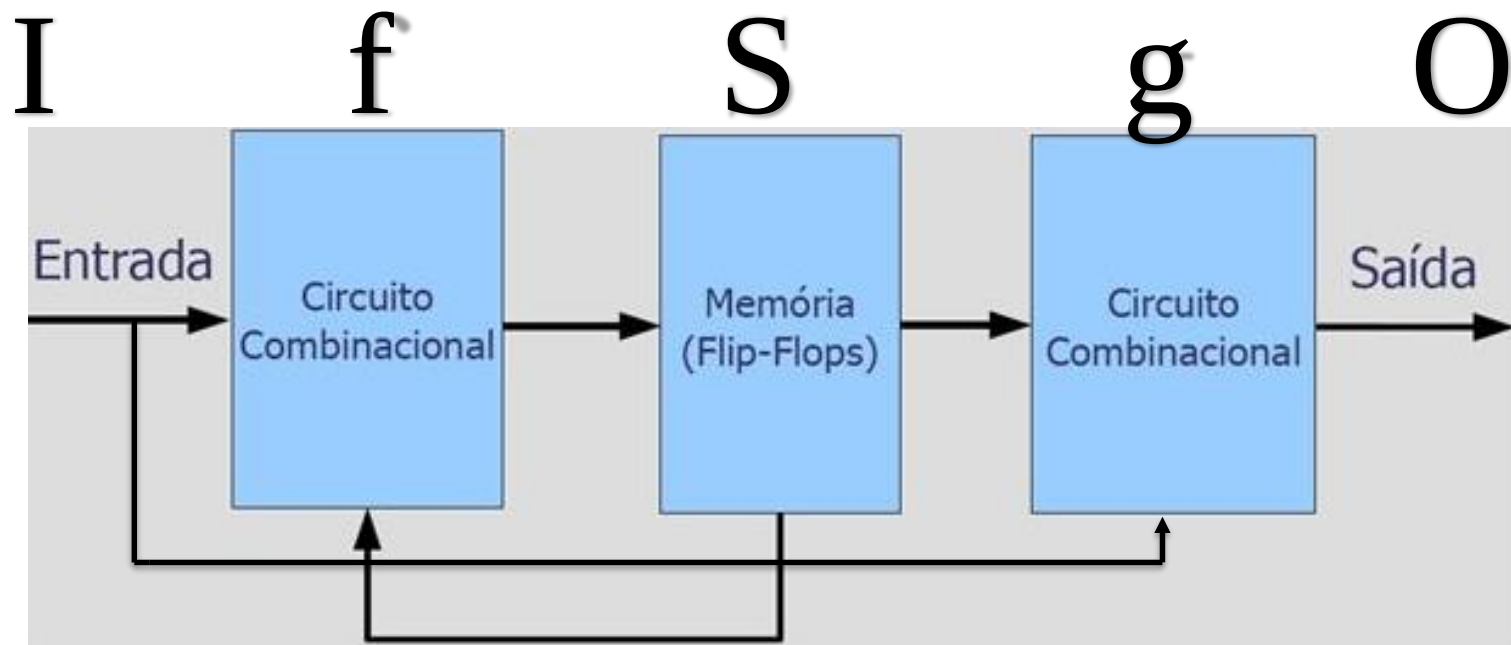
Rótulos no nós:

$$EstadoAtual/Output = S_t/O_t$$



2. Máquinas de Estado e Circuitos Digitais

- $M = (I, O, S, f, g, \sigma)$: cada entrada possível é um elemento de **I**, **S** contém configurações possíveis para FFs, e cada saída possível é um elemento de **O** (Tudo em **binário**). **f** e **g** são circuitos combinatórios.



2. Máquinas de Estado e Circuitos Digitais

- Técnicas de verificação de **Equivalências** no âmbito de **Circuitos Combinatórios**:
 - Tabelas da Verdade;
 - Equações Algébricas;
 - Mapas de *Karnaugh*.
- **Equivalências** para **Circuitos Sequenciais**:
 - Máquinas de Estados Finitos.

2. Máquinas de Estado e Circuitos Digitais

- Técnicas de verificação de **Equivalências** no âmbito de **Circuitos Combinatórios**:
 - Tabelas da Verdade;
 - Equações Algébricas;
 - Mapas de *Karnaugh*.
- **Equivalências** para **Circuitos Sequenciais**:
 - Máquinas de Estados Finitos.

2. Máquinas de Estado e Circuitos Digitais

- Pode-se utilizar **Máquinas de Estado** para manipular circuitos combinatórios?
- Se sim, quantos estados teríamos?

Exercício

- Projetar uma FSM que indique a paridade de uma cadeia de bits:
 - » Entrada: cadeia de bits, $I = \{0,1\}$
 - » Saída: 1 sse cadeia lida até então tem número ímpar de 1s.

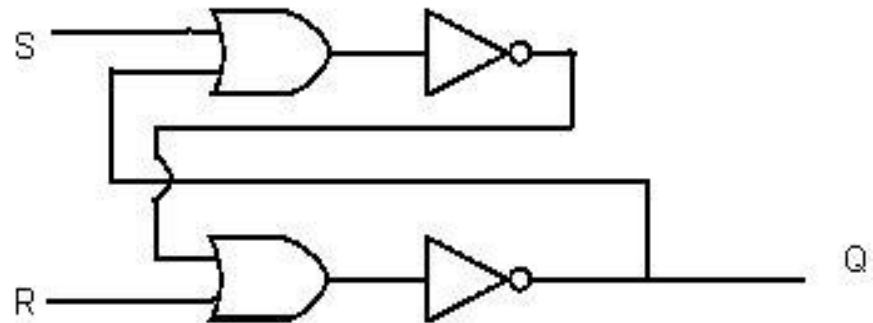
Exercício

- Projetar uma FSM que faça uma soma bit a bit propagando o carry.
 - » Entrada: pares de bits, do menos pro mais significativo, $I=\{00,01,10,11\}$
 - » Saída: 0 ou 1, conforme a soma e o vem-um

Exercício

– Projetar um flip-flop RS como FSM

S	R	Q
1	1	Not allowed
1	0	1
0	1	0
0	0	1 if S was last 1 0 if R was last 1



3. Diagrama ASM – Algorithmic State Machine

- Representação gráfica do algoritmo que descreve o comportamento do sistema digital, ou seja, uma ferramenta para descrever uma **Máquina de Estados Finita**.
- É uma maneira diagramática de representar a Função de **Saída** e a Função de **Próximo Estado**, relacionando-as ao **Estado Atual** da máquina.
- Assumem-se entradas/saídas binárias.

3. Diagrama ASM – Algorithmic State Machine

- Cada diagrama ASM corresponde diretamente a um circuito sequencial síncrono.
- Utilizando diagramas ASMs para descrever máquinas de estado finitas, obtemos uma implementação imediata em hardware.
- **Inconveniente:** No método mais direto, o número de flip-flops é igual ao número de estados (*one-hot encoding*).

3. Diagrama ASM – Algorithmic State Machine

■ Dispõe-se de quatro **Elementos Primitivos**:

- Bloco de Estado;
- Bloco de Decisão;
- Bloco de Junção;
- Bloco de Saída Condicional.

3. Diagrama ASM – Algorithmic State Machine

- A cada um destes **Elementos Primitivos** associa-se um símbolo gráfico (para elaborar o **Diagrama ASM**).
- Estes últimos, por sua vez, são associados a **Elementos Primitivos de *Hardware*** (para elaborar o **Diagrama Lógico** correspondente).

3. Diagrama ASM – Algorithmic State Machine

- Bloco de Estado:

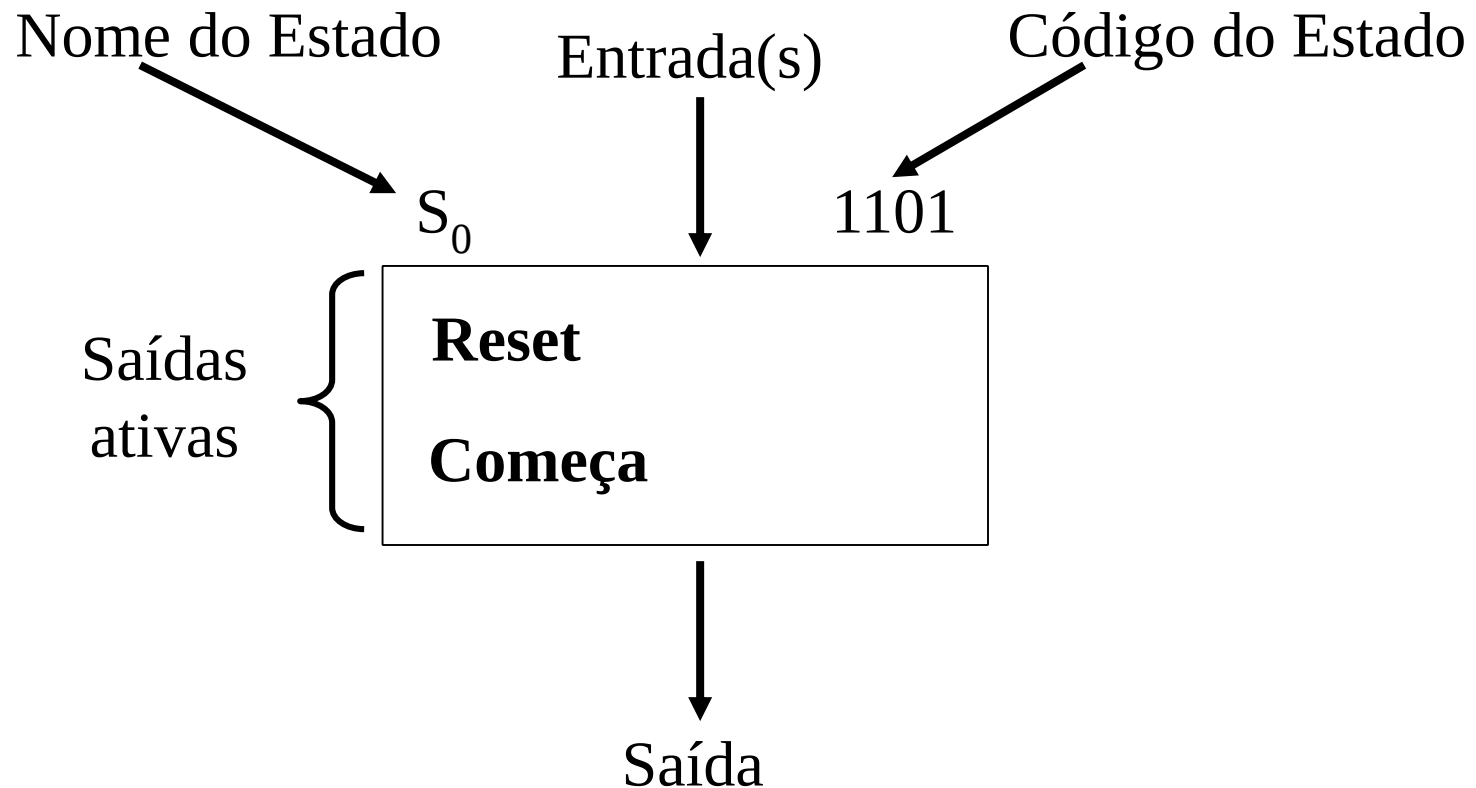
- Lista das saídas de Estado (Moore);
- Nome do Estado (código);

- Lista das saídas:

- Nomes das saídas que serão iguais a 1, independentemente da entrada.
- Podem definir um conjunto de operações;

3. Diagrama ASM – Algorithmic State Machine

■ Bloco de Estado:

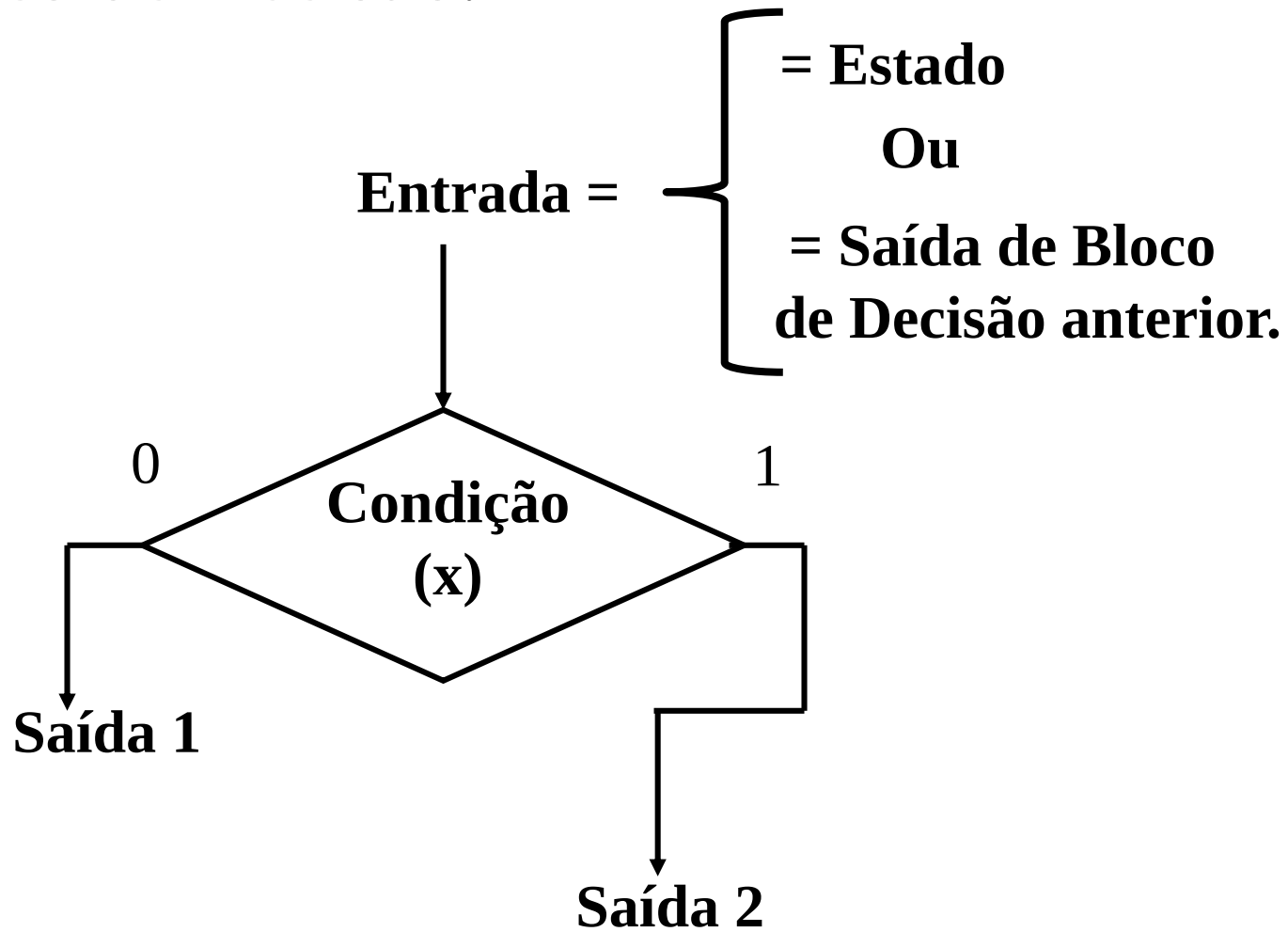


3. Diagrama ASM – Algorithmic State Machine

- **Bloco de Decisão** – Descreve as entradas (externas) para a máquina de estados e possui 2 caminhos de saída:
 - Condição verdadeira (“ $x = 1$ ”) –
Entrada externa $x = 1$;
 - Condição falsa (“ $x = 0$ ”) –
Entrada externa $x = 0$.
- Representa o efeito das entradas no próximo estado e na saída.

3. Diagrama ASM – Algorithmic State Machine

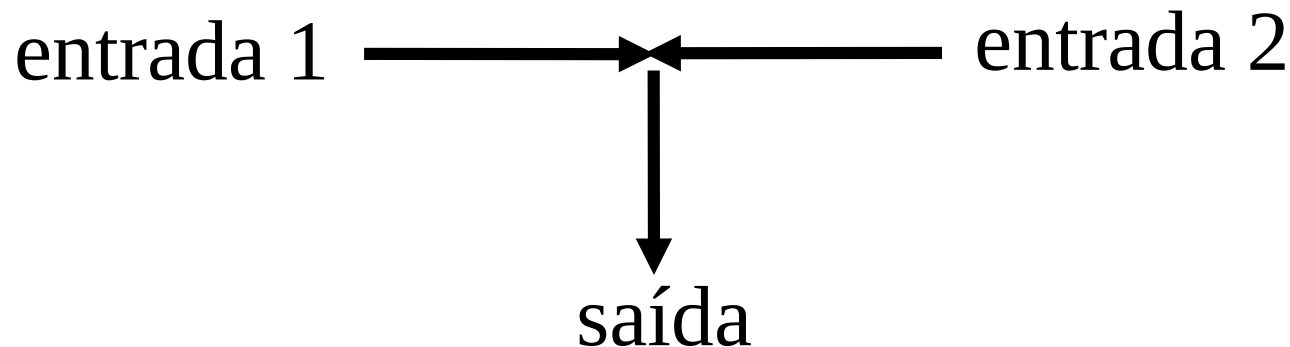
■ Bloco de Decisão:



3. Diagrama ASM – Algorithmic State Machine

- **Bloco de Junção** – Este elemento descreve e trata um conjunto de uma ou mais entradas que provocam a mesma saída (transições que levam ao mesmo estado):

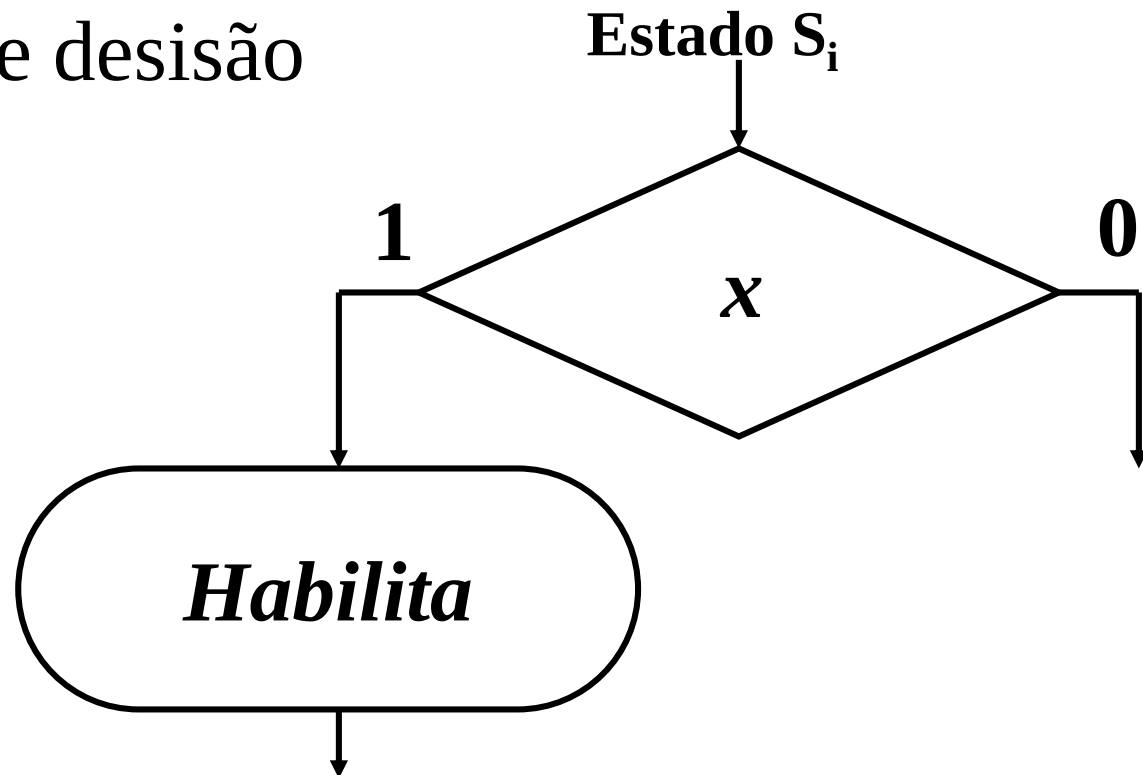
– Exemplo:



3. Diagrama ASM – Algorithmic State Machine

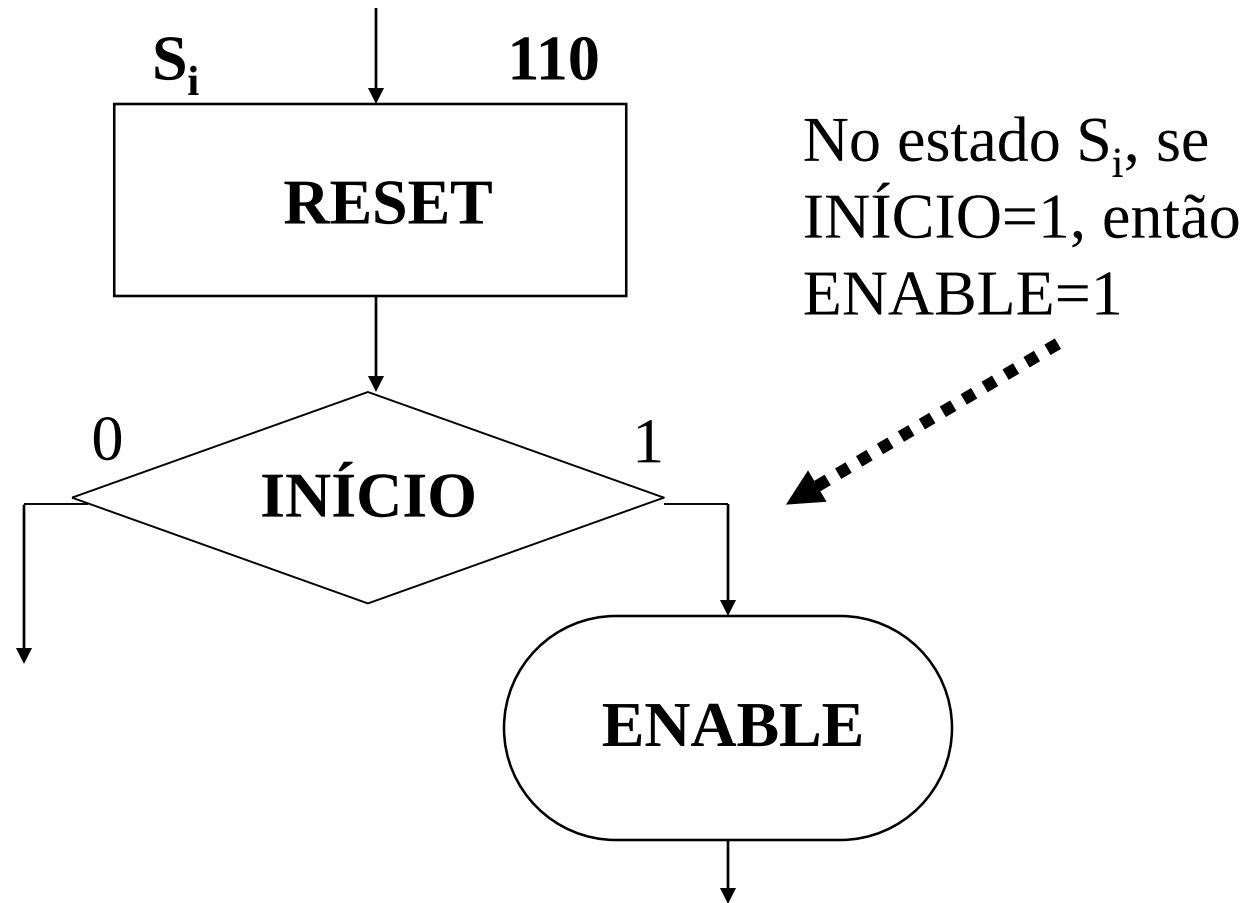
■ Bloco de Saída Condicional:

- Lista saídas que devem ser iguais a 1 para o valor das entradas especificados pelos blocos de desisão



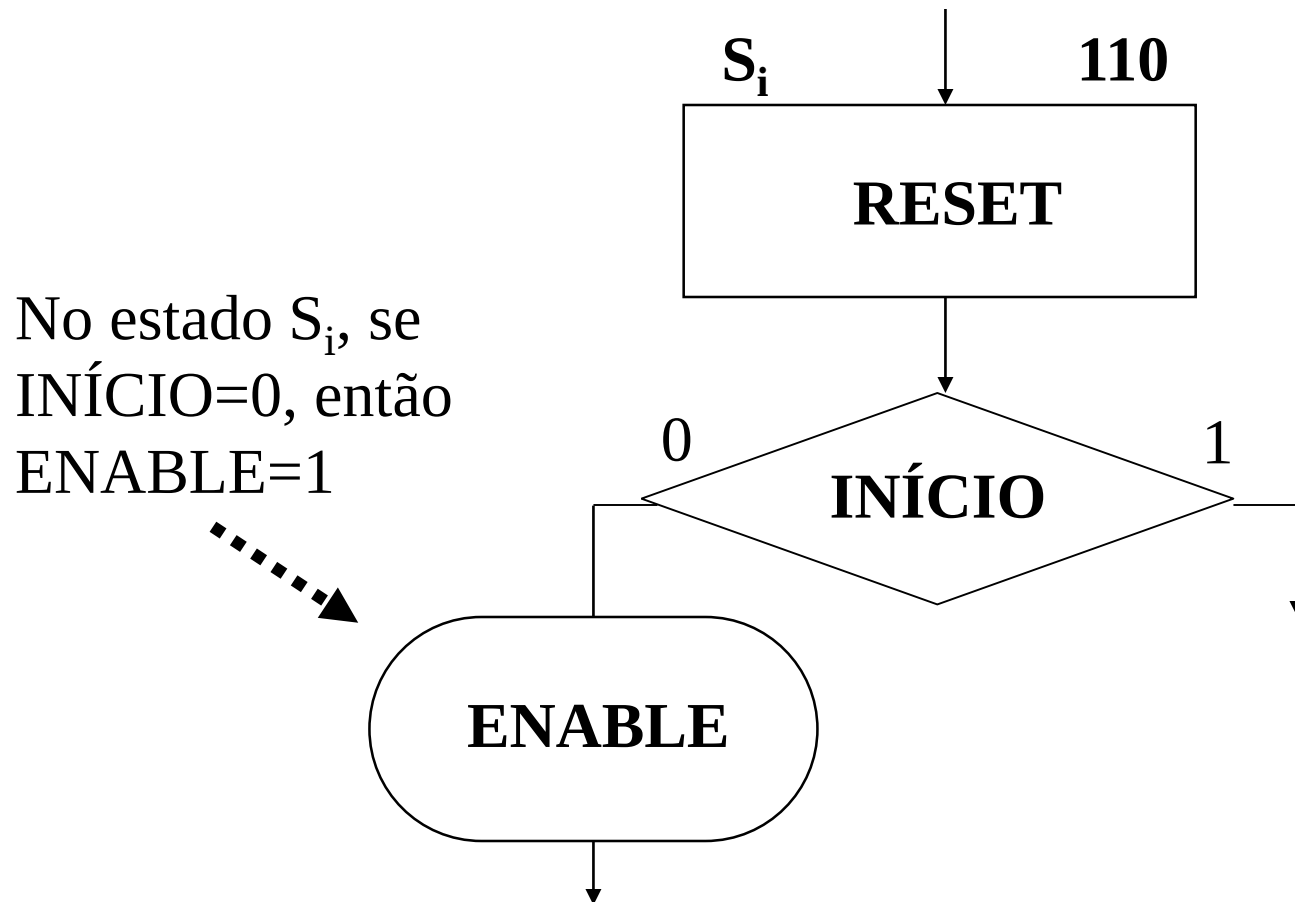
3. Diagrama ASM – Algorithmic State Machine

■ Bloco de Saída Condicional (inserido em um Diagrama ASM):



3. Diagrama ASM – Algorithmic State Machine

■ **Bloco de Saída Condicional** (inserido em um Diagrama ASM):



3. Diagrama ASM – Algorithmic State Machine

- **Bloco ASM** – Estrutura que consiste de um Bloco de Estado e uma rede de Blocos de Decisão, Blocos de Saída Condicional e Blocos de Junção:
 - Descreve a operação da Máquina de Estados durante um estado de tempo (período do clock);
 - Representa o estado atual, o próximo estado e as saídas (=1).
 - As demais saídas são iguais a zero.

3. Diagrama ASM – Algorithmic State Machine

Exemplo:
Bloco ASM

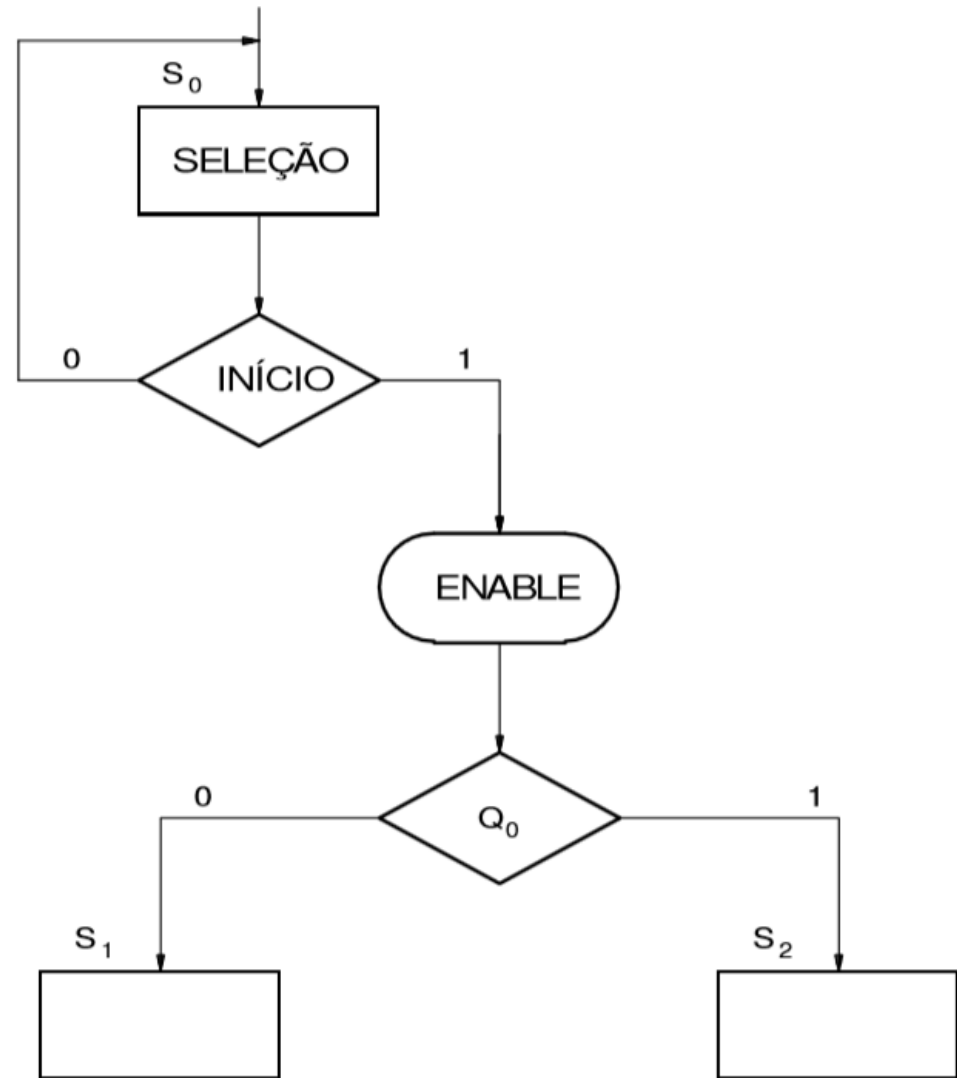


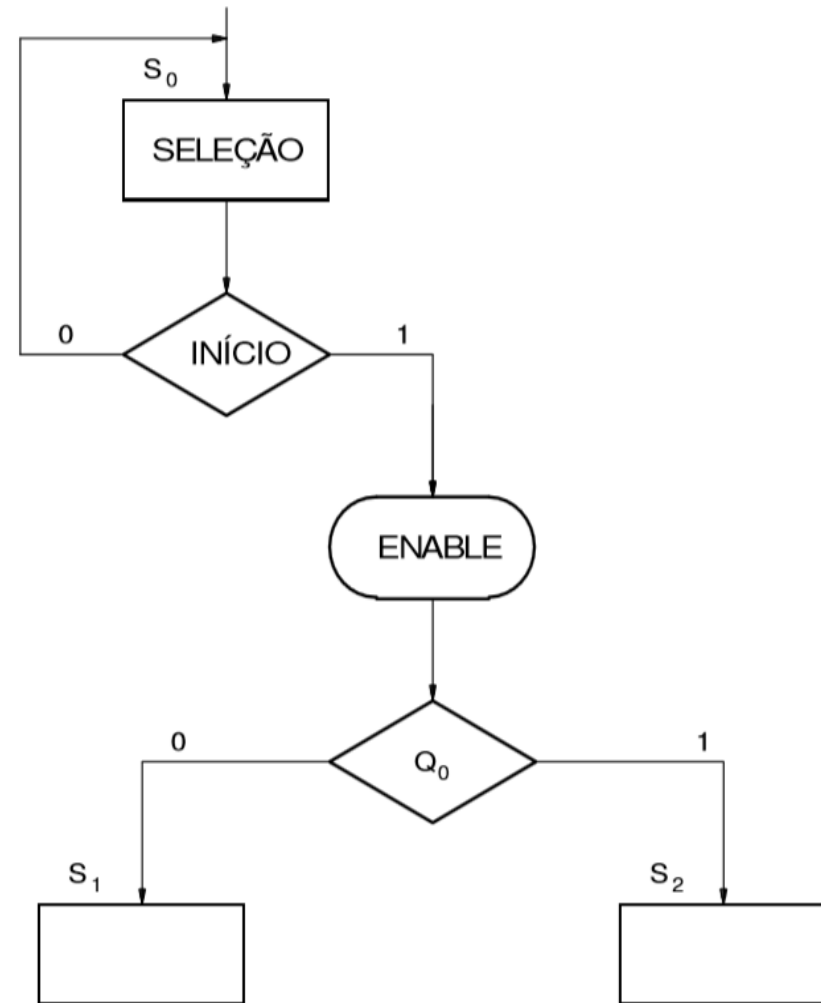
Diagrama ASM:
1 Bloco por Estado

3. Diagrama ASM – Algorithmic State Machine

Em **Hardware**:

Um estado por FF,
 $Q=1$ sse estado
atual.

Caminhos são sinais
(fios), ativos (=1)
sse ligam estado
atual ao futuro (FF)

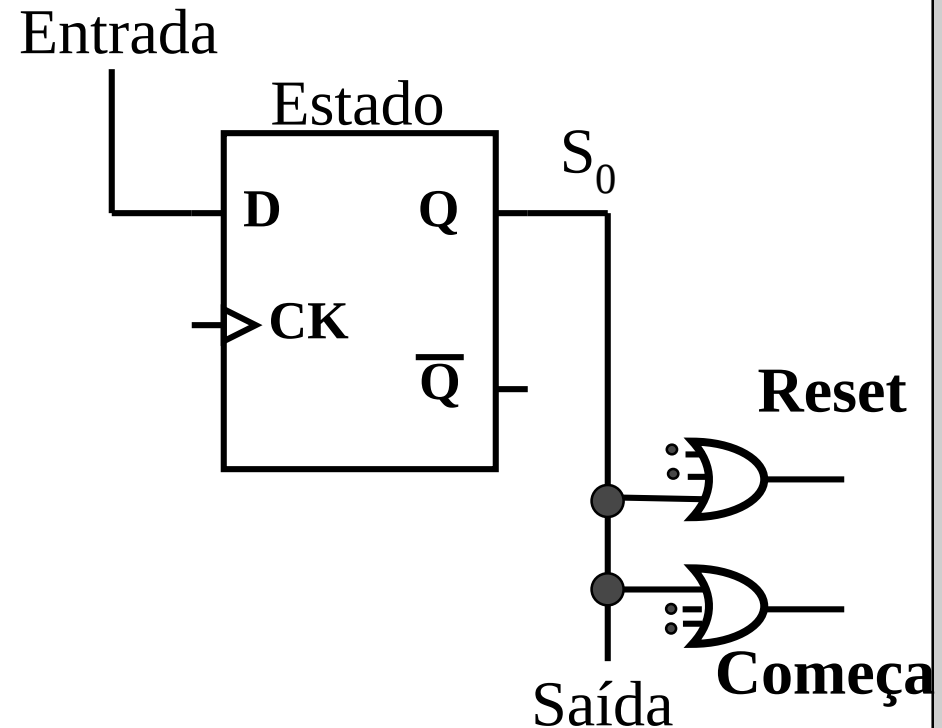
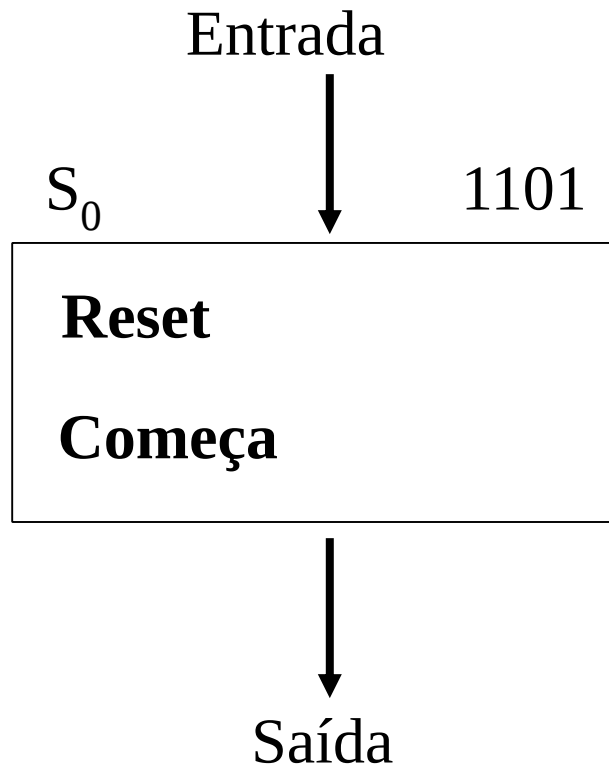


3. Diagrama ASM – Algorithmic State Machine

- Cada símbolo é mapeado em *hardware*:
 - Bloco de Estado: *Flip-Flop*
 - Bloco de Decisão: *AND* e *NOT*
 - Bloco de Junção: *OR*
 - Saída (Condicional ou de estado): *OR*, descartado se com 1 entrada.

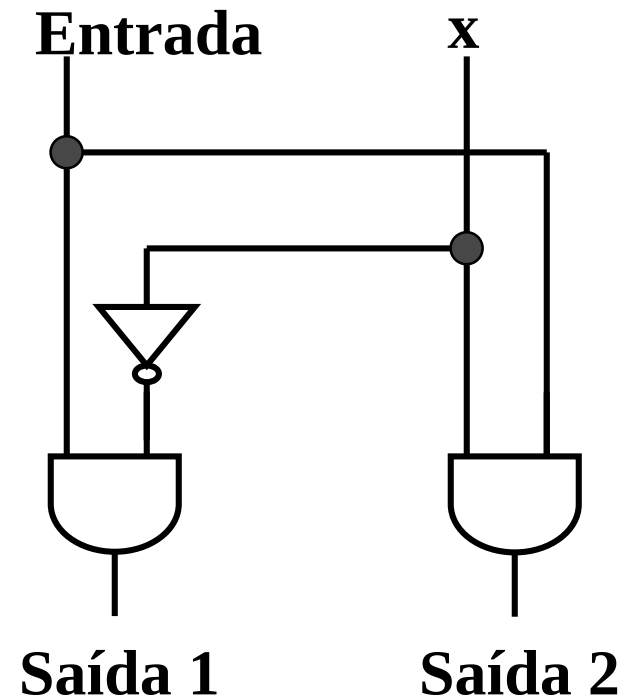
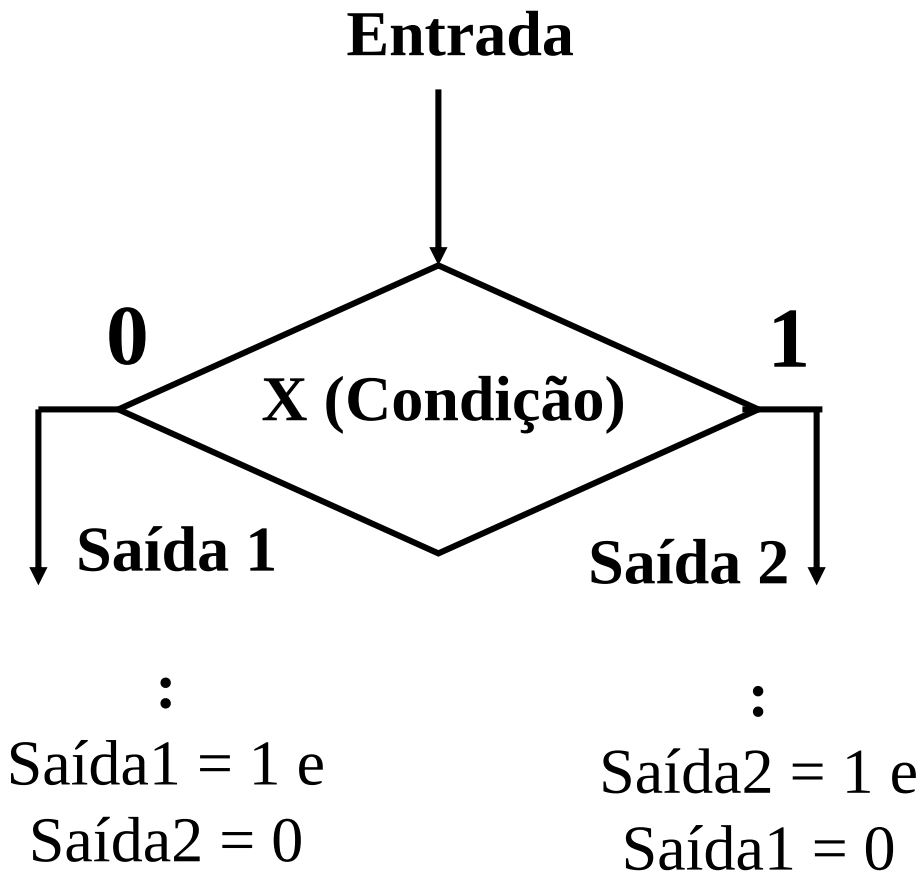
3. Diagrama ASM – Algorithmic State Machine

- Cada símbolo é mapeado em *hardware*:
 - Bloco de Estado: *Flip-Flop D*



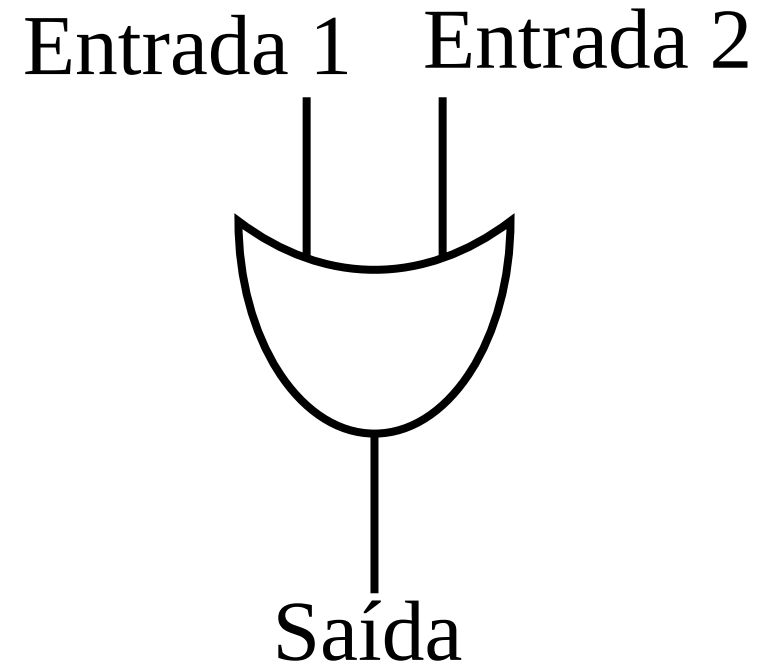
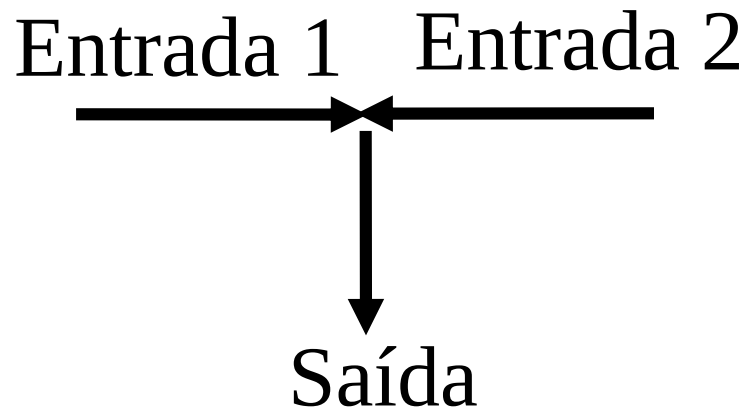
3. Diagrama ASM – Algorithmic State Machine

- Cada símbolo é mapeado em *hardware*:
 - Bloco de Decisão: *AND+NOT, DEMUX 1:2*



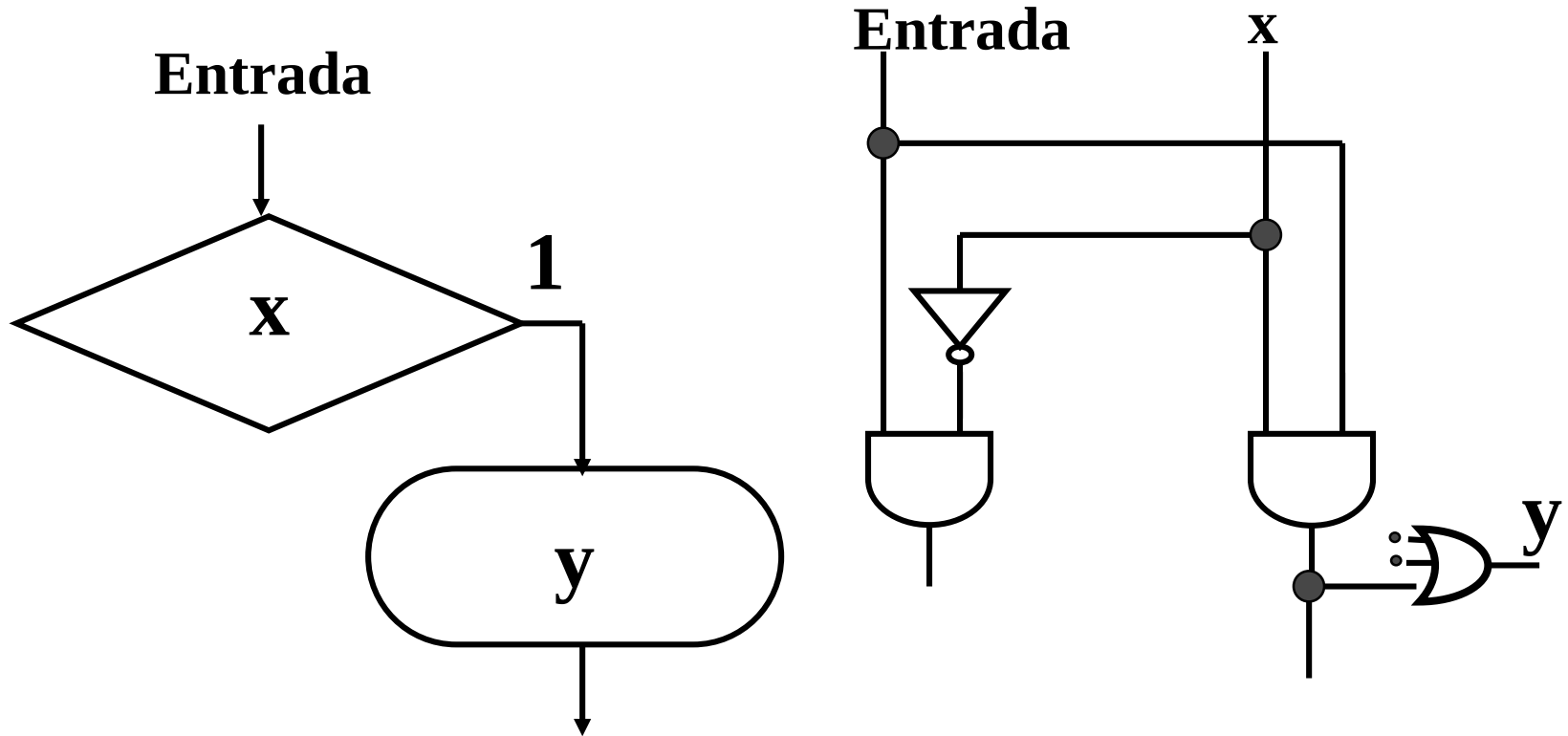
3. Diagrama ASM – Algorithmic State Machine

- Cada símbolo é mapeado em *hardware*:
 - Bloco de Junção: *OR*



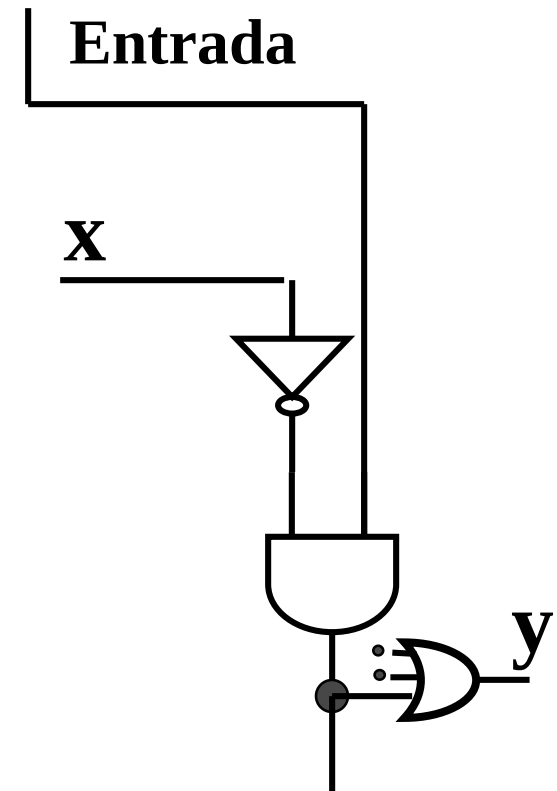
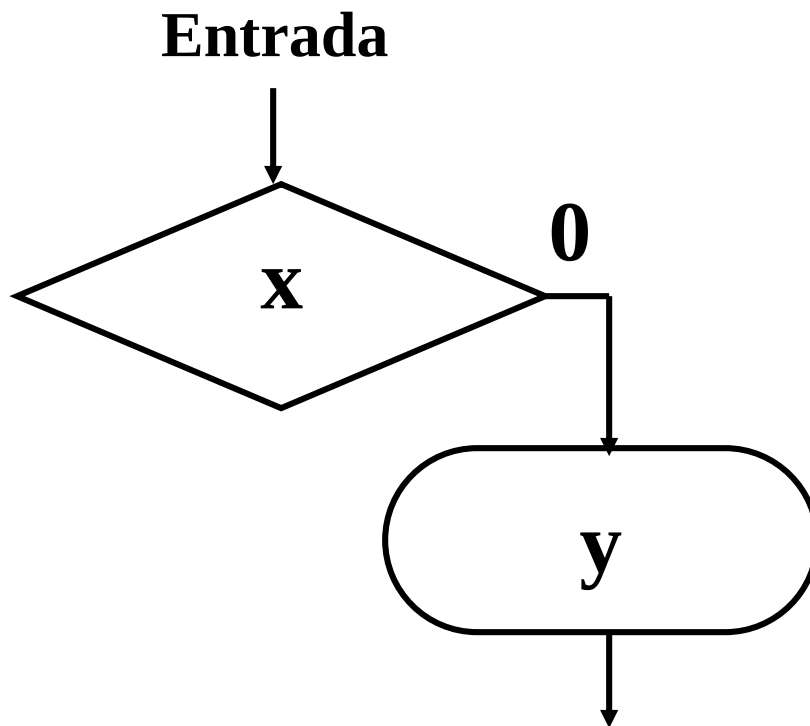
3. Diagrama ASM – Algorithmic State Machine

- Cada símbolo é mapeado em *hardware*:
 - Bloco de Saída Condicional: fios até a *OR*

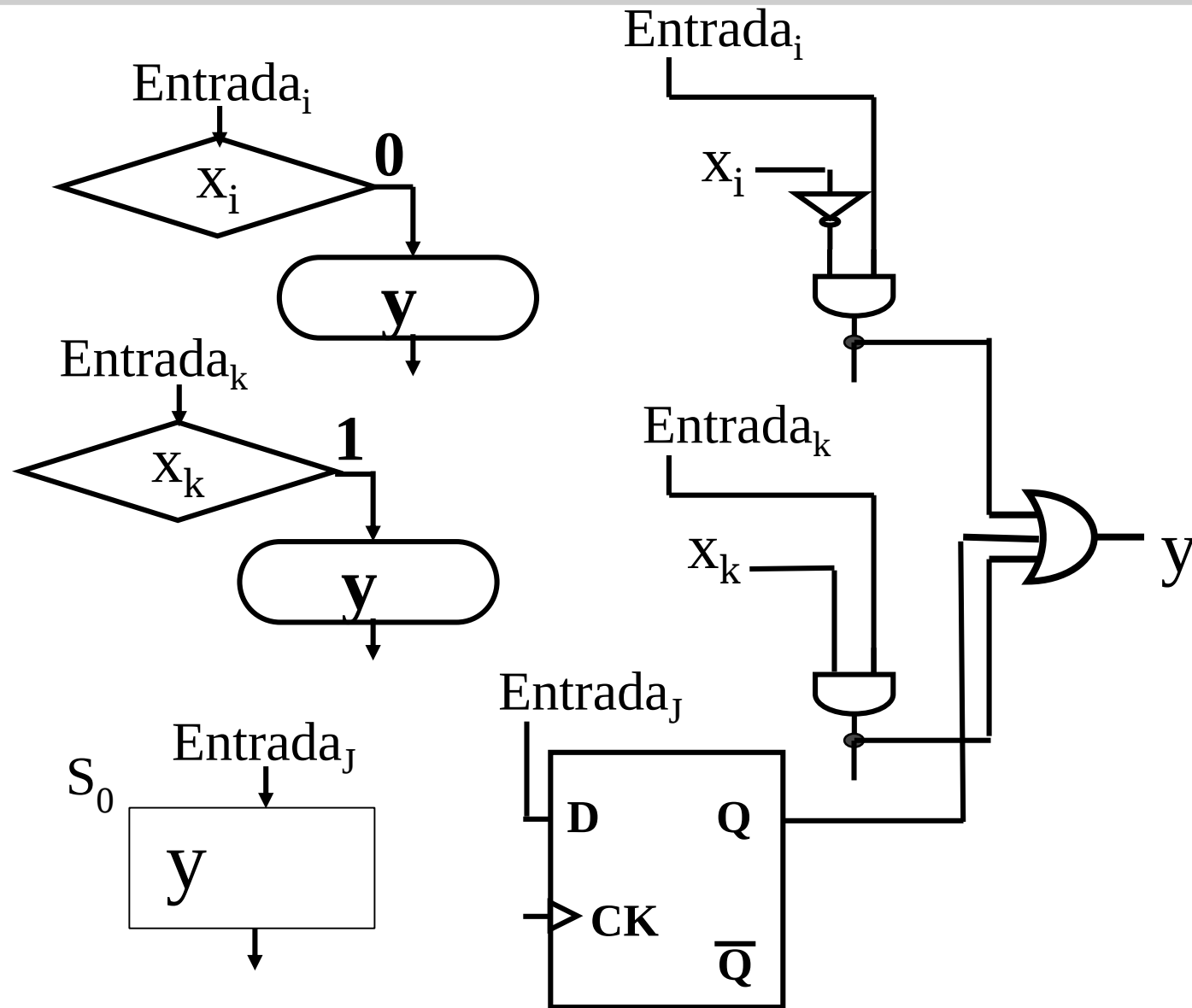


3. Diagrama ASM – Algorithmic State Machine

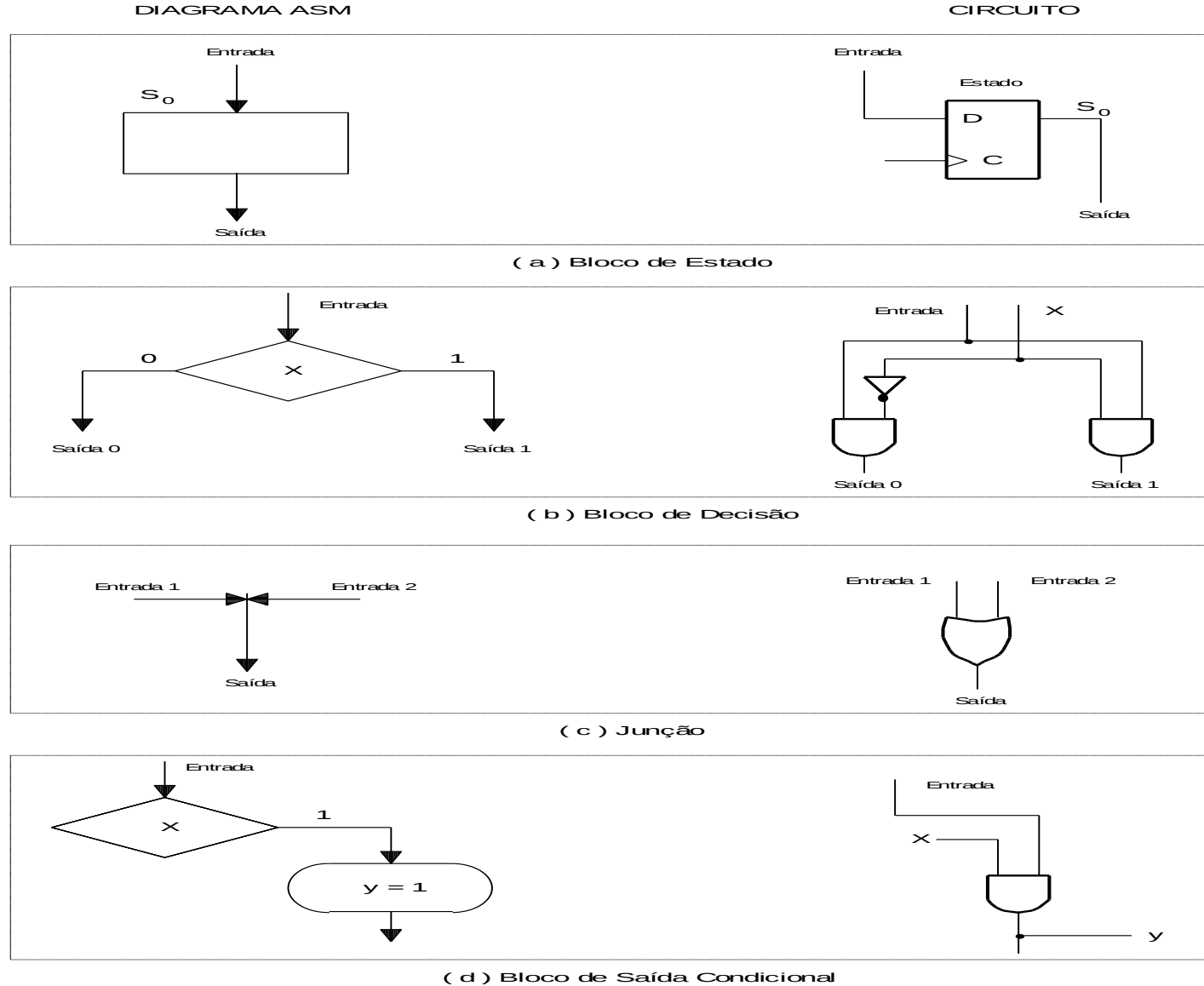
- Cada símbolo é mapeado em *hardware*:
 - Bloco de Saída Condicional: fios até a *OR*.



3. Diagrama ASM – Algorithmic State Machine



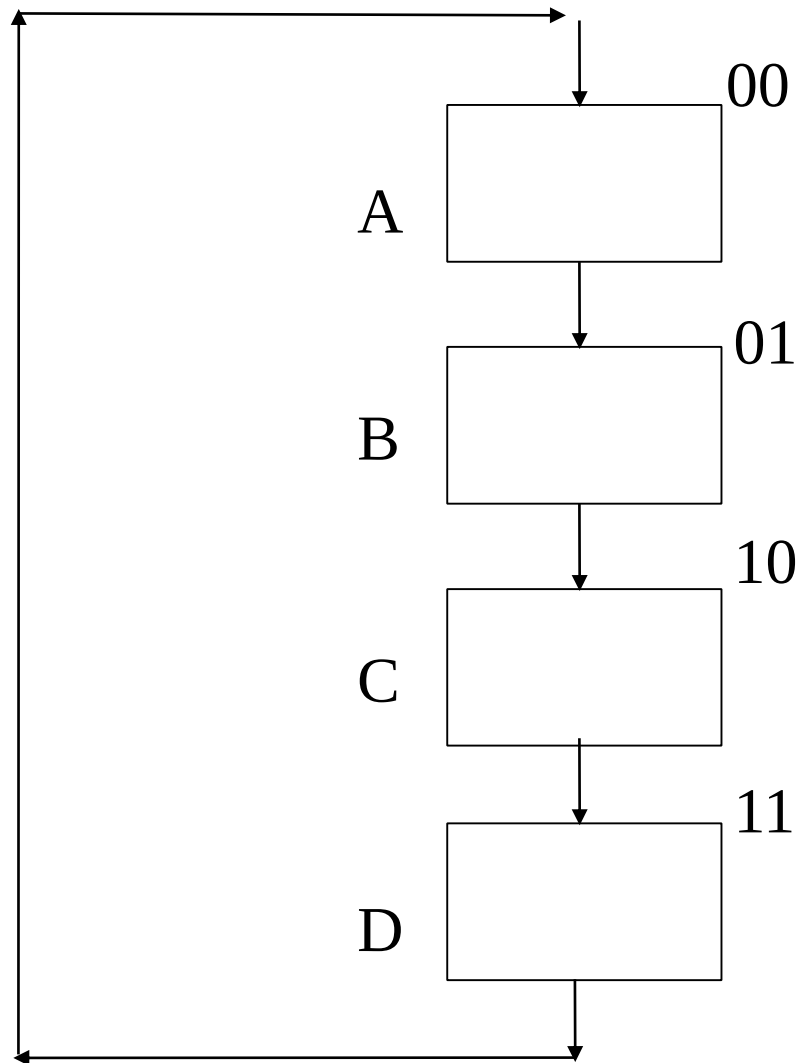
3. Diagrama ASM – Algorithmic State Machine



3. Diagrama ASM – Algorithmic State Machine

- Restrições dos diagramas ASM:
 - Blocos de saídas condicionais devem vir depois de Blocos de Condição.
 - Blocos de Condição não podem fazer loops sem passar por estados.
 - Blocos de Condição num mesmo caminho entre dois estados devem testar diferentes entradas.

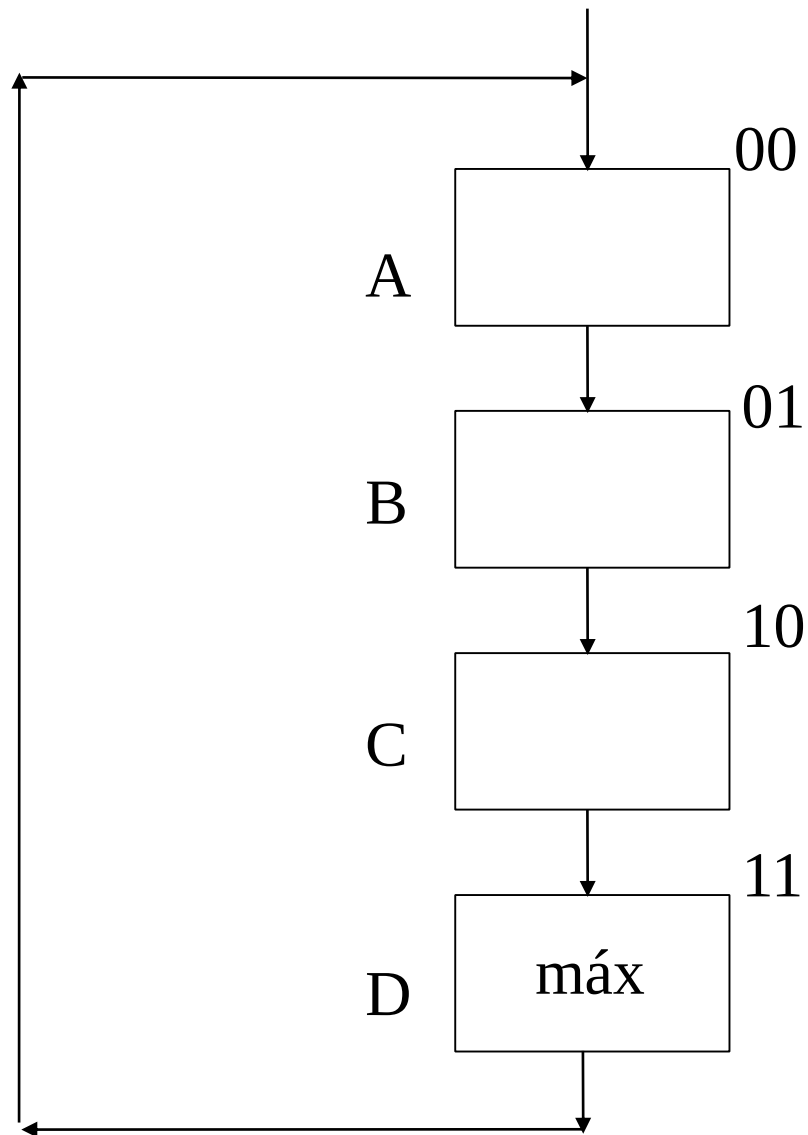
3.1. Diagrama ASM – Exemplos:



O que esse
Sistema faz?

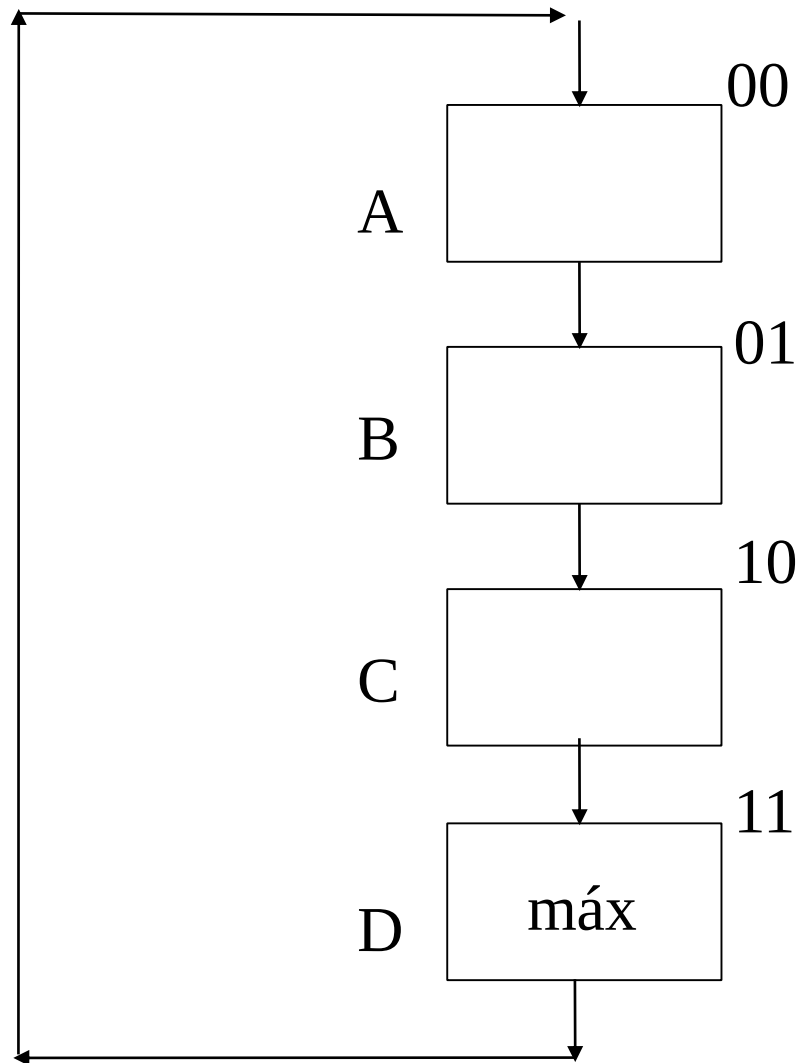
contador de
módulo 4

3.1. Diagrama ASM – Exemplos:



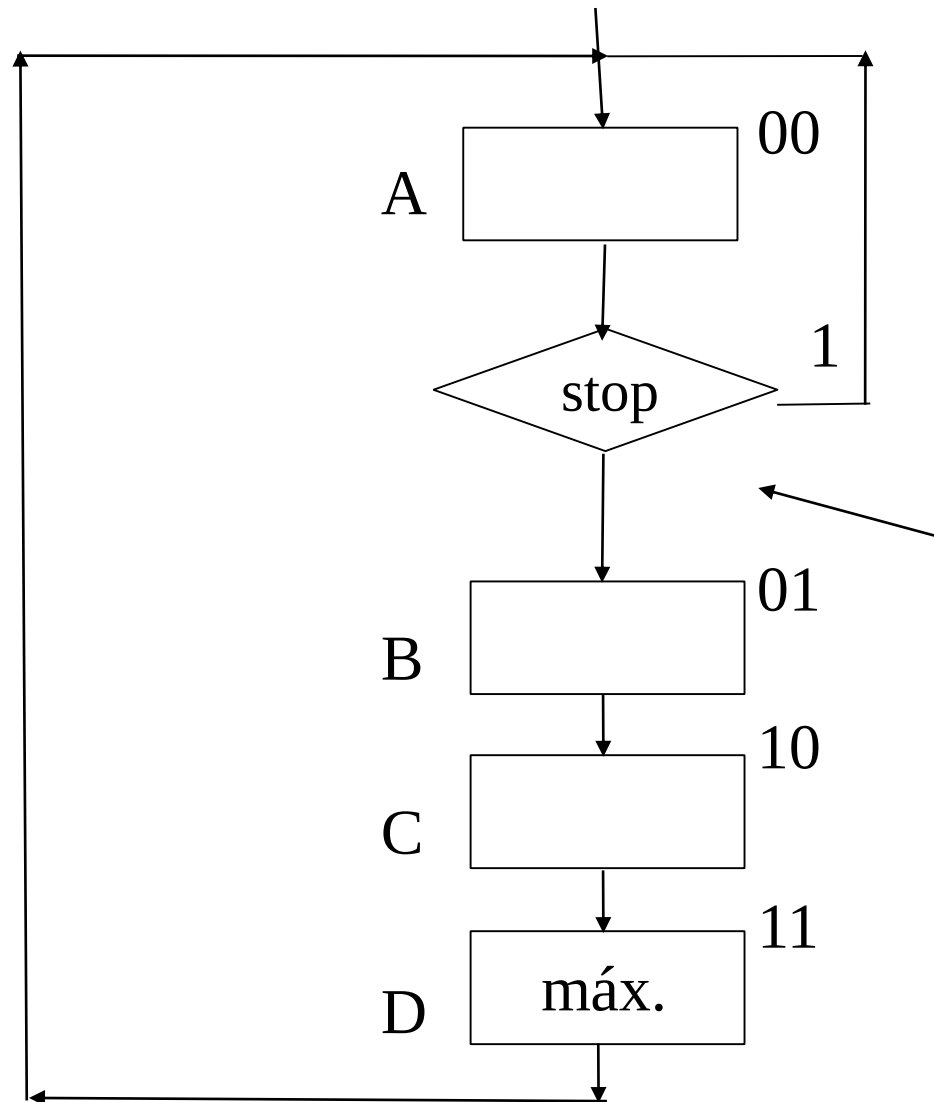
Quero que o Contador avise em uma saída (máx) quando chegar a 11. Como faço isso?

3.1. Diagrama ASM – Exemplos:



Quero que o Contador pare no estado A se a entrada stop=1. Como faço isso?

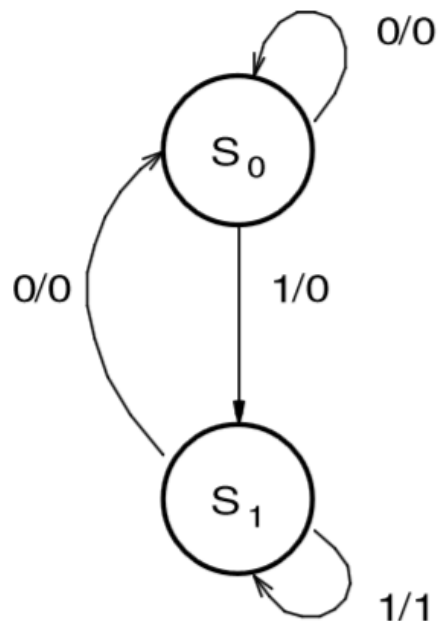
3.1. Diagrama ASM – Exemplos:



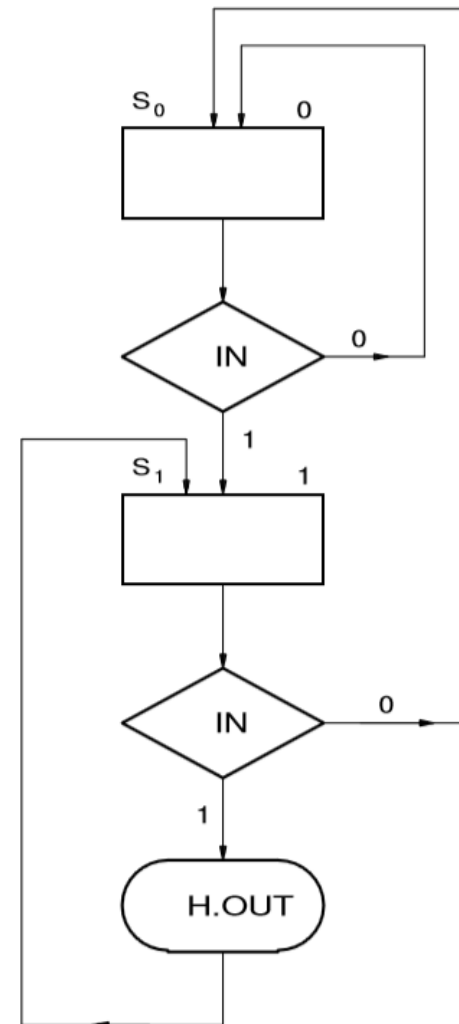
Quero que o contador não passe do estado A se a entrada **stop** for 1. Como faço isso?

3.2. ASM e Máquinas de Estado

Relacionando máquinas de estado e diagramas ASM

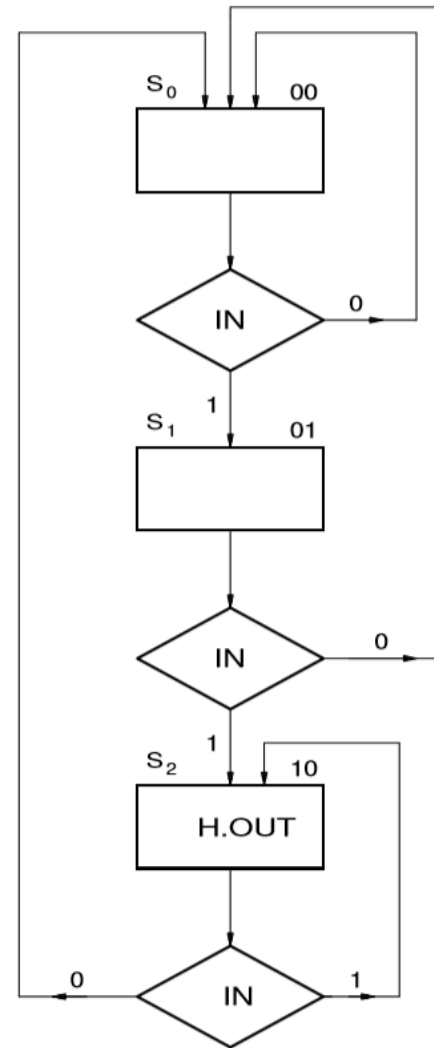
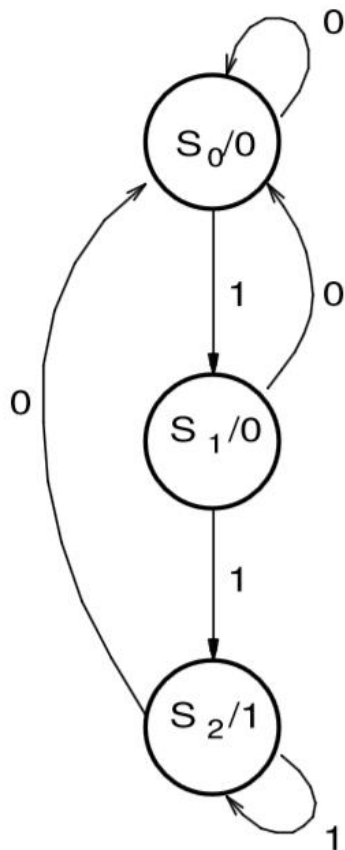


Entrada: IN
Saída: H.OUT



3.2. ASM e Máquinas de Estado

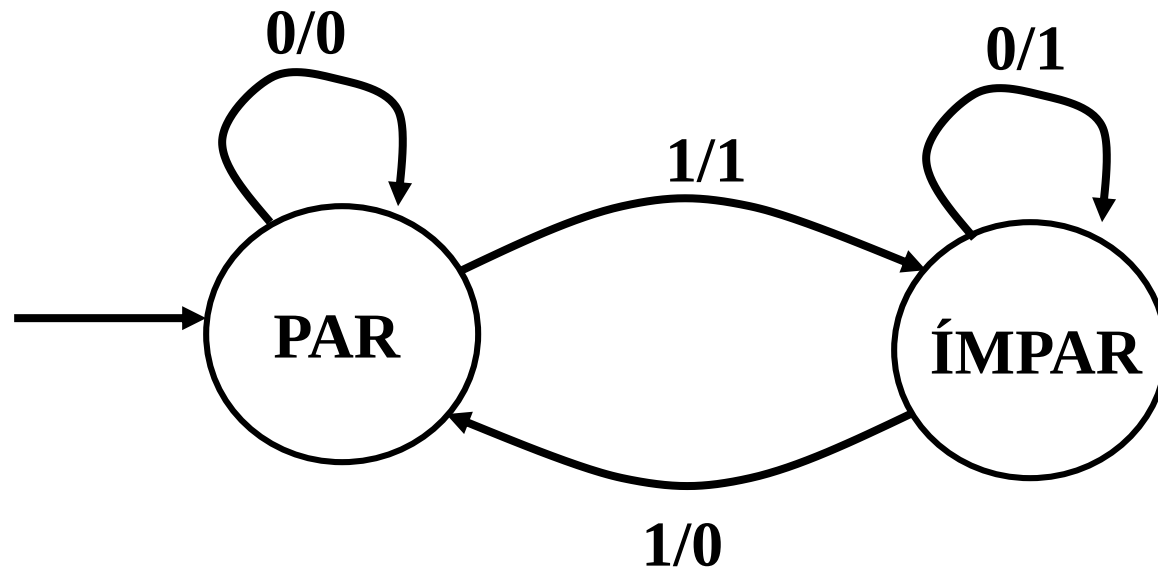
Relacionando máquinas de estado e diagramas ASM



Exercício

- Projetar uma FSM que indique a paridade de uma cadeia de bits:
 - » Entrada: cadeia de bits, $I = \{0,1\}$
 - » Saída: 1 sse cadeia lida até então tem número ímpar de 1s.

3.1. Diagrama ASM – Exemplos:



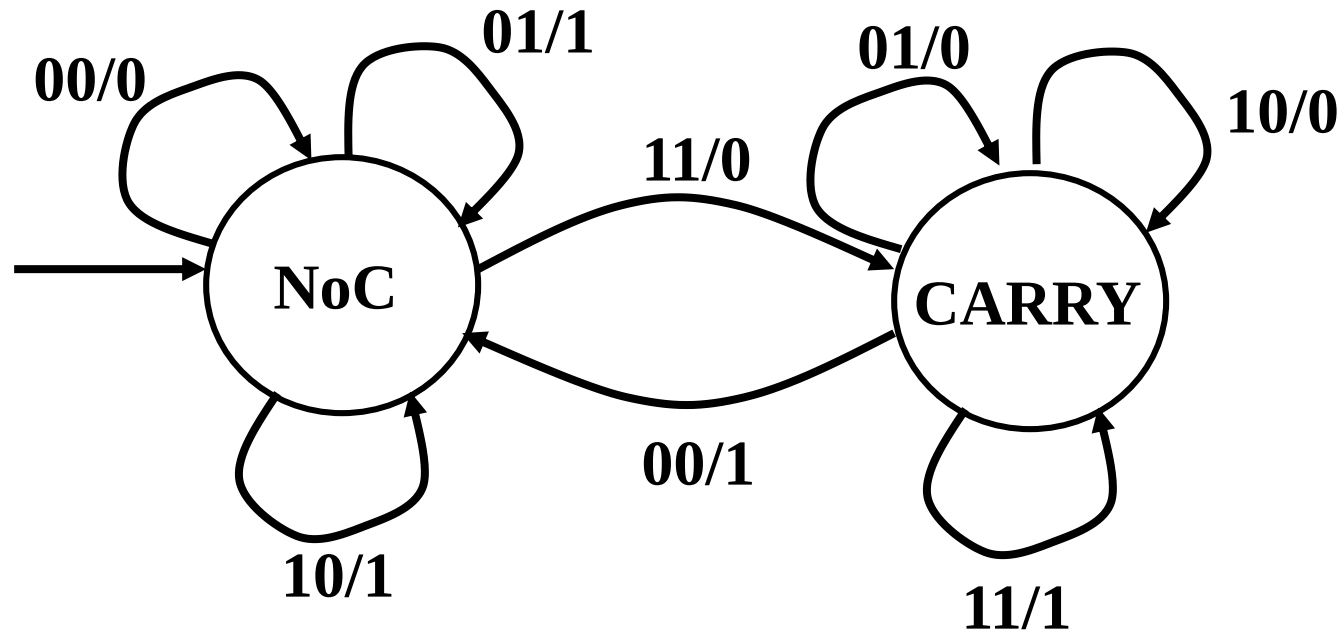
- Desenhar o diagrama ASM correspondente e então o circuito.

Exercício

- Projetar uma FSM que faça uma soma bit a bit propagando o carry.
 - » Entrada: pares de bits, do menos pro mais significativo, $I=\{00,01,10,11\}$
 - » Saída: 0 ou 1, conforme a soma e o vem-um

3.1. Diagrama ASM – Exemplos:

**Legenda: CARRY = Houve Carry;
NoC = Não houve Carry.**



– Desenhar o diagram ASM correspondente e então o circuito.

Bibliografia Adicional Deste Assunto

- Clare, Christopher R.; *Designing Logic Systems using State Machines*; McGraw-Hill, 1973.