

Sistemas Digitais 1

Tiago Alves

Faculdade UnB Gama
Universidade de Brasília



Aula 21

- Estrutura de Máquinas de Estados
- Equações Características
- Análise de Máquinas de Estados usando Flip-Flops D



Hoje estudaremos um circuito chamado “*clocked synchronous state-machine*” (ou máquina de estados síncronas ou ativadas por sinal de relógio/clock).

Máquina de Estados é o nome genérico que damos a esse tipo de circuito.

O termo **clocked** indica que os elementos de memória (os flip-flops) utilizados apresentam (respondem a) uma entrada de relógio/clock.

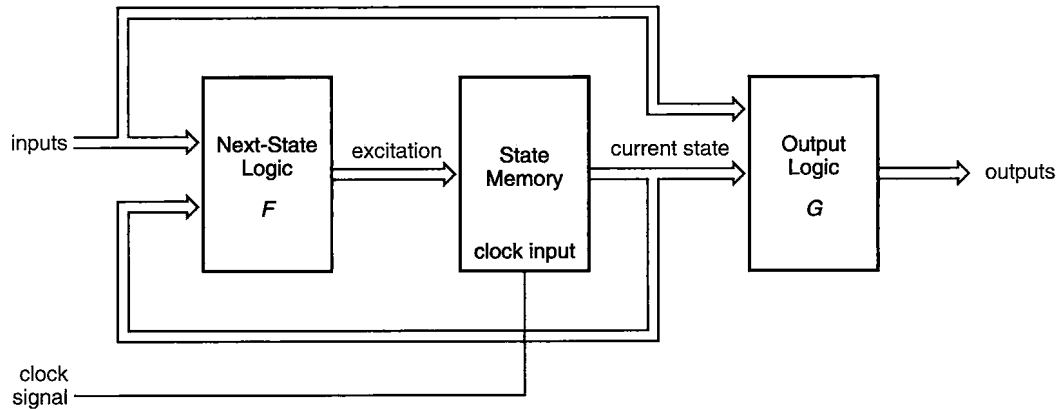
Finalmente, o termo **synchronous** indica que todos os flip-flops utilizam o mesmo sinal de relógio/clock.

Assim, o estado da máquina só muda quando se observar a borda de ativação do sinal de relógio/clock.



Estrutura de Máquina de Estados

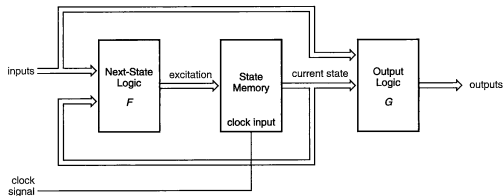
A estrutura dos circuitos que estudaremos segue a seguinte arquitetura:



A **memória** é um conjunto de n flip-flops que guarda o estado atual da máquina (portanto, podemos ter 2^n estados).

O próximo estado da máquina é determinado pelo módulo **lógica de próximo estado**, que é um circuito combinacional **F** cujas entradas são as entradas do circuito **e** o estado atual da máquina.

A saída é determinada pelo módulo **lógica de saída**, que também é um circuito combinacional **G** cujas entradas são as entradas do circuito **e** o estado atual da máquina.



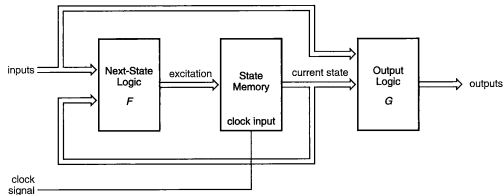
Assim, podemos expressar a arquitetura usando as seguintes equações:

$$\text{Next state} = \mathbf{F}(\text{current state}, \text{input})$$

$$\text{Output} = \mathbf{G}(\text{current state}, \text{input})$$

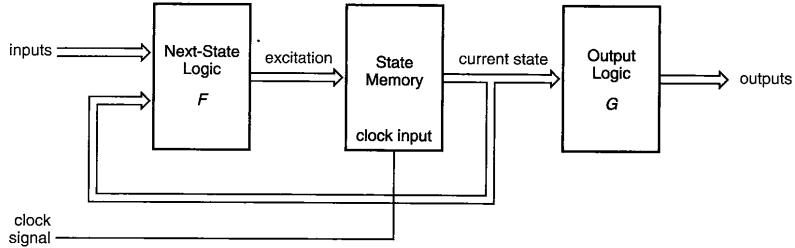
No circuito que mostramos, a saída depende tanto do estado atual quanto da entrada atual (e, portanto, a saída pode mudar independentemente do clock).

Esse tipo de máquina é chamada de *Mealy machine*, assim batizada por ter sido proposta por George H. Mealy em um artigo intitulado “A Method for Synthesizing Sequential Circuits” em 1955.

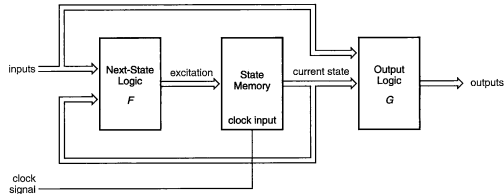


Estrutura de Máquina de Estados

Outro tipo de arquitetura é mostrado abaixo:



Qual é a diferença em relação ao modelo anterior? Confira atentamente o diagrama abaixo apresentado.

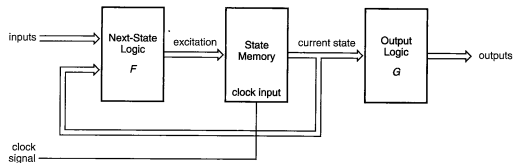


Neste caso:

$$\text{Next state} = \mathbf{F}(\text{current state}, \text{input})$$

$$\text{Output} = \mathbf{G}(\text{current state})$$

No circuito, a saída depende apenas do estado atual (logo, a saída é síncrona – só muda quando o estado muda, isto é, na borda de ativação do clock). Esse tipo de máquina é chamada de *Moore machine*, assim batizada por ter sido proposta por Edward F. Moore em um artigo intitulado “Gedanken-experiments on Sequential Machines” em 1956.

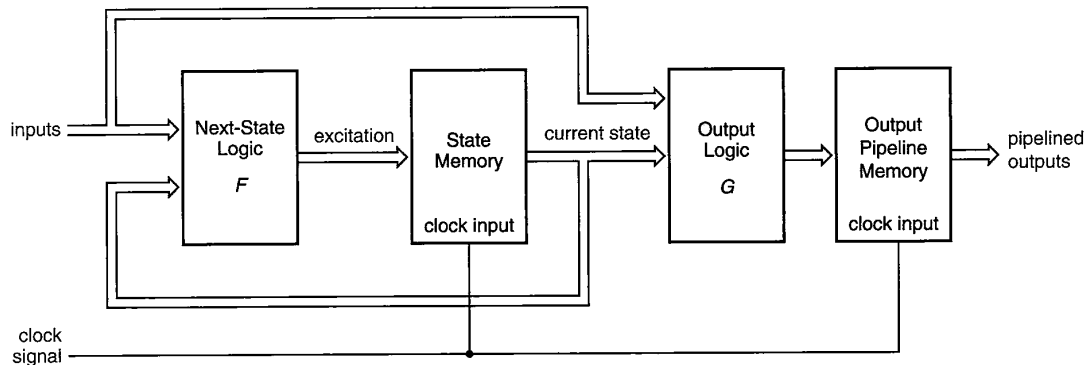


Na prática, as máquinas de estados tem algumas saídas do tipo Mealy (isto é, que dependem da entrada e do estado atual) e algumas saídas do tipo Moore (que dependem apenas do estado atual).



Estrutura de Máquina de Estados

Outra arquitetura de realização de máquina de estados é projetar a máquina de tal forma que as saídas em um período de clock dependem das entradas e suas repercussões **observadas em um período de clock anterior** (isto é, a saída é atrasada). Chamamos essa máquina de “Mealy Machine with Pipelined Outputs”.



Antes de aprofundarmos no projeto de uma máquina de estados, vamos aprender a descrever o funcionamento dos flip-flops através de um tipo específico de equações booleanas chamadas de **equações características**.

A **equação característica** de um flip-flop **descreve o próximo estado** desse flip-flop (*após* a borda de ativação do clock) **em função do estado atual e das entradas desse flip-flop**.

$$Q^* = f(Q, inputs)$$

Note que essa equação não descreve os detalhes de funcionamento do flip-flop (se é ativo em borda de subida, descida, etc..), apenas a resposta funcional às entradas de controle.



Dispositivo	Equação Característica
SR Latch	?
D Latch	?
Edge-Triggered D Flip-Flop	?
D Flip-Flop with Enable	?
Master-Slave SR Flip-Flop	?
Master-Slave JK Flip-Flop	?
Edge-Triggered JK Flip-Flop	?
T Flip-Flop	?
T Flip-Flop with Enable	?



Dispositivo	Equação Característica
SR Latch	$Q^* = S + \overline{R} \cdot Q$
D Latch	$Q^* = D$
Edge-Triggered D Flip-Flop	$Q^* = D$
D Flip-Flop with Enable	$Q^* = EN \cdot D + \overline{EN} \cdot Q$
Master-Slave SR Flip-Flop	$Q^* = S + \overline{R} \cdot Q$
Master-Slave JK Flip-Flop	$Q^* = J \cdot \overline{Q} + \overline{K} \cdot Q$
Edge-Triggered JK Flip-Flop	$Q^* = J \cdot \overline{Q} + \overline{K} \cdot Q$
T Flip-Flop	$Q^* = \overline{Q}$
T Flip-Flop with Enable	$Q^* = EN \cdot \overline{Q} + \overline{EN} \cdot Q$



Considere a definição de máquinas de estado:

$$Next\ state = \mathbf{F}(current\ state, input)$$

$$Output = \mathbf{G}(current\ state, input)$$

Lembrando que definimos o **estado** como **tudo aquilo que precisamos saber sobre o passado do circuito**, a primeira equação \mathbf{F} nos diz que o que queremos saber (o próximo estado) pode ser obtido através **do estado atual** (o que já sabemos) e **da entrada**.

Da mesma forma, a segunda equação \mathbf{G} nos diz que podemos obter a saída utilizando essa mesma informação.

Assim, o objetivo da análise de um circuito sequencial é determinar as funções de próximo estado \mathbf{F} e saída \mathbf{G} , de tal forma que o comportamento do circuito possa ser modelado.



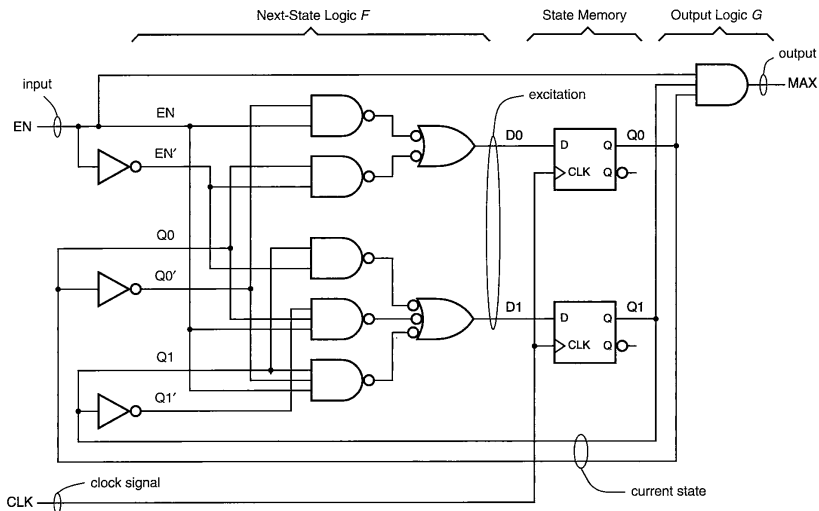
Assim, definimos três passos básicos:

- 1 Determinar as funções de próximo estado F e de saída G ;
- 2 Usar F e G para construir uma **tabela de transição de estados e saída**, que especifica completamente qual é o próximo estado e a saída para cada combinação de estado atual e entrada.
- 3 Desenhar um diagrama de estados, que apresenta essa informação em um gráfico que permite a validação do modelo que desejamos realizar.



Análise de Máquinas de Estado com Flip-Flops D

Seja o circuito abaixo. Inicialmente, queremos determinar as equações de excitação dos flip-flops (que serão usadas para determinar a função F).



Análise de Máquinas de Estado com Flip-Flops D

No caso:

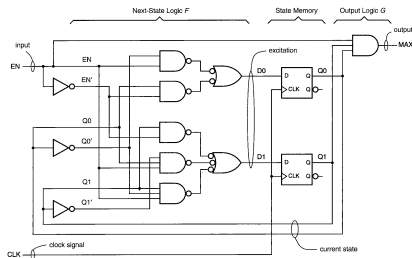
$$D_0 = Q_0 \cdot \overline{EN} + \overline{Q_0} \cdot EN$$

$$D_1 = Q_1 \cdot \overline{EN} + \overline{Q_1} \cdot Q_0 \cdot EN + Q_1 \cdot \overline{Q_0} \cdot EN$$

Especificamente, para o caso do flip-flop D:

$$Q_0^* = D_0$$

$$Q_1^* = D_1$$

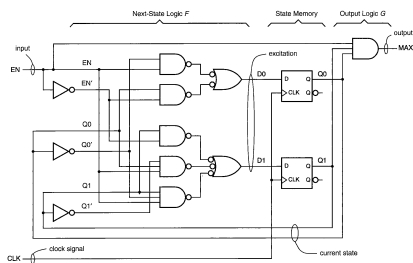


E, portanto:

$$Q_0^* = Q_0 \cdot \overline{EN} + \overline{Q_0} \cdot EN$$

$$Q_1^* = Q_1 \cdot \overline{EN} + \overline{Q_1} \cdot Q_0 \cdot EN + Q_1 \cdot \overline{Q_0} \cdot EN$$

Para cada combinação do estado atual $Q_1 Q_0$ e da entrada EN , essas equações nos dão o próximo estado $Q_1^* Q_0^*$.



Podemos escrever a tabela de transição de estados:

$$Q_0^* = Q_0 \cdot \overline{EN} + \overline{Q_0} \cdot EN$$

$$Q_1^* = Q_1 \cdot \overline{EN} + \overline{Q_1} \cdot Q_0 \cdot EN + Q_1 \cdot \overline{Q_0} \cdot EN$$

EN	Q ₁	Q ₀	Q ₁ *	Q ₀ *
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		



Podemos escrever a tabela de transição de estados:

$$Q_0^* = Q_0 \cdot \overline{EN} + \overline{Q_0} \cdot EN$$

$$Q_1^* = Q_1 \cdot \overline{EN} + \overline{Q_1} \cdot Q_0 \cdot EN + Q_1 \cdot \overline{Q_0} \cdot EN$$

EN	Q ₁	Q ₀	Q ₁ *	Q ₀ *
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0



É comum re-reescrevermos essa tabela da forma:

$Q_1 Q_0$	EN	
	0	1
00	00	01
01	01	10
10	10	11
11	11	00
	$Q_1^* Q_0^*$	



Podemos também nomear cada estado (isto vai ser especialmente útil no projeto de máquinas deste tipo):

S	EN	
	0	1
A	A	B
B	B	C
C	C	D
D	D	A
S*		



$Q_1 Q_0$	EN	
	0	1
00	00	01
01	01	10
10	10	11
11	11	00
	$Q_1^* Q_0^*$	

Essa tabela mostra a ideia do circuito: é um **contador binário** de 2 bits, controlado por uma entrada de **enable**. Quando ativado, o circuito segue os estados:

$00 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow 00 \dots$



Tabela de Transição de Estados

Finalmente, podemos analisar a saída do circuito:

$$\text{MAX} = Q_1 \cdot Q_0 \cdot \text{EN}$$

Que pode ser combinada em uma *tabela de estado/saída*.

S	EN	
	0	1
A	A,0	B,0
B	B,0	C,0
C	C,0	D,0
D	D,0	A,1
S*, MAX		

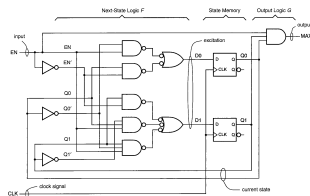
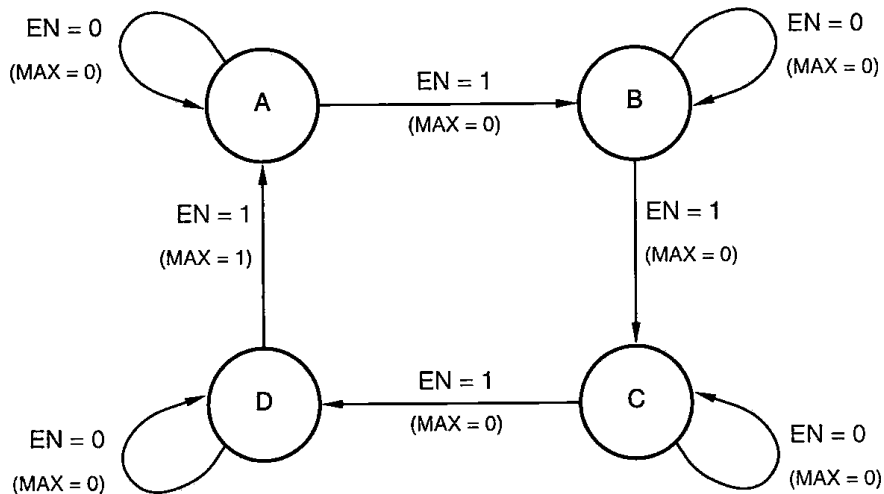


Diagrama de Estados

O Diagrama de Estados mostra essa informação de forma gráfica.



Exercício 01

Analise a máquina de estados abaixo:

