

SÍNTESE DE SISTEMAS SEQÜENCIAIS POR DIAGRAMA DE ESTADOS

O diagrama de estados é uma ferramenta de análise de circuitos seqüenciais que permite descrever o comportamento dinâmico de uma máquina de estados. É um grafo e portanto visual em forma de diagrama muito útil e prático no projeto de máquinas cuja descrição do funcionamento é produto de um conjunto de especificações bem elaboradas. Através de alguns símbolos conforme figura o diagrama de estados é apresentado.

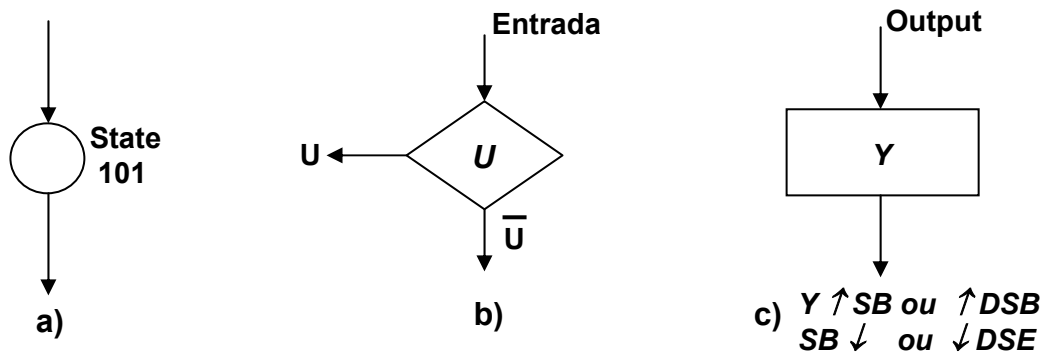
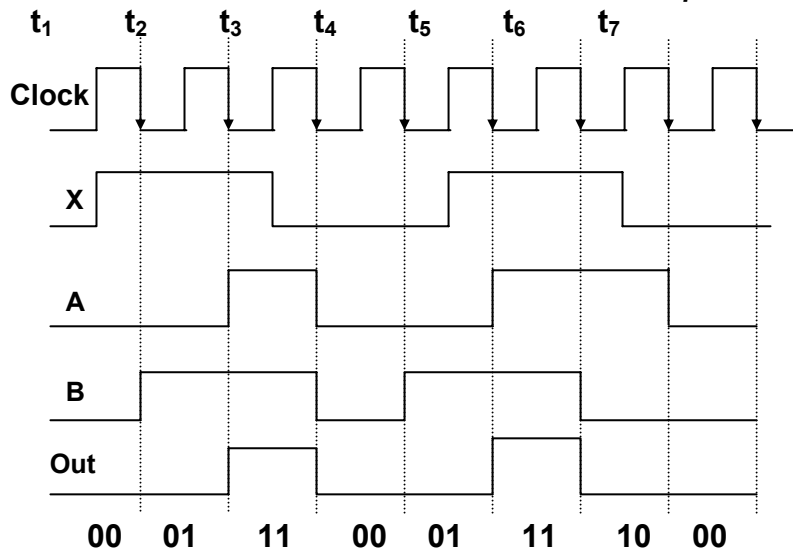
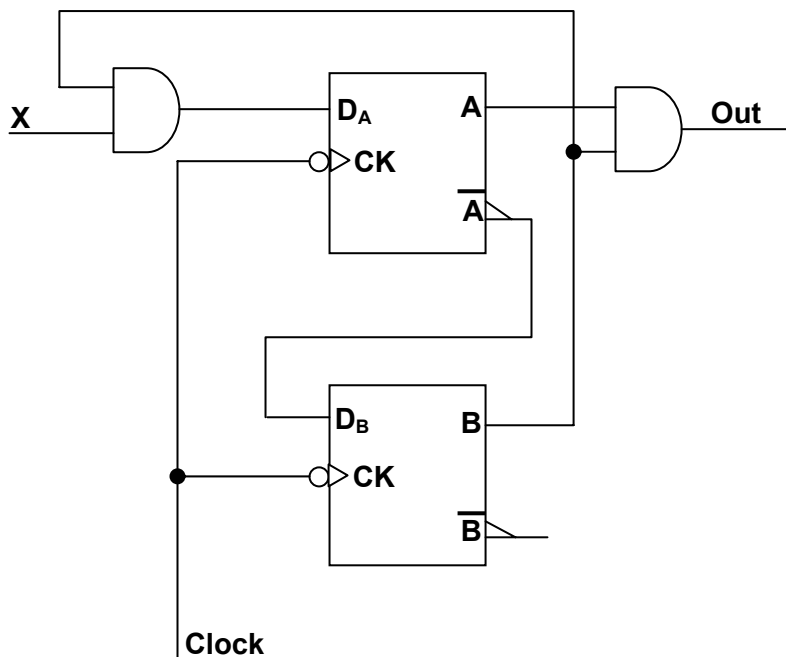


Figura Símbolos do diagrama de estado.

O símbolo a) circular representa o estado, internamente ao círculo vem o nome do estado e do lado de fora próximo ao círculo a identificação do estado. Os arcos indicam o caminho de entrada e de saída. Podem existir múltiplos arcos e portanto muitos caminhos de entrada, porém um único caminho de saída. O símbolo b), um losango, representa a variável de entrada e internamente contém o nome da variável. O símbolo terá um único caminho de entrada e terá 02 caminhos de saída determinados pelo valor lógico da variável ou U ou \bar{U} . O símbolo c) em retângulo representa a saída e internamente é inscrito o nome da variável de saída. Os arcos de entrada e saída indicam os caminhos de entrada e saída e adjacente ao retângulo as indicações **Set** e **Reset** da saída. A indicação após nome da variável de saída arco ascendente indica que a saída é **Set**, enquanto o arco descendente indica que a saída é **Reset**. Se a saída é **Set**, no início do estado, a notação $Y \uparrow SB$ é usado. Se a saída é **Set** no meio-estado, a notação será $Y \uparrow DSB$ indicando que o **Set** é atrasado do início do tempo do estado. As notações $Y \downarrow SE$ e $Y \downarrow DSE$ significam **Reset** no final do tempo do estado e respectivamente sobre a próxima transição do clock após o final do tempo do estado. Para ilustrar o uso do diagrama de estado, no exemplo a seguir, será construído o diagrama de estado do circuito de uma máquina de estados. As formas de ondas definem a dinâmica da máquina de estados em função da variável de entrada X. O uso de 02 F/Fs mostra que o circuito é do tipo seqüencial e o comum dos clock que é um circuito síncrono. A figura a seguir mostra as formas de ondas da máquina de estados.



O circuito a seguir gerou as formas de ondas apresentadas na figura. A seguir apresentamos a tabela de estados presentes e futuros, de descrição da máquina de estados.

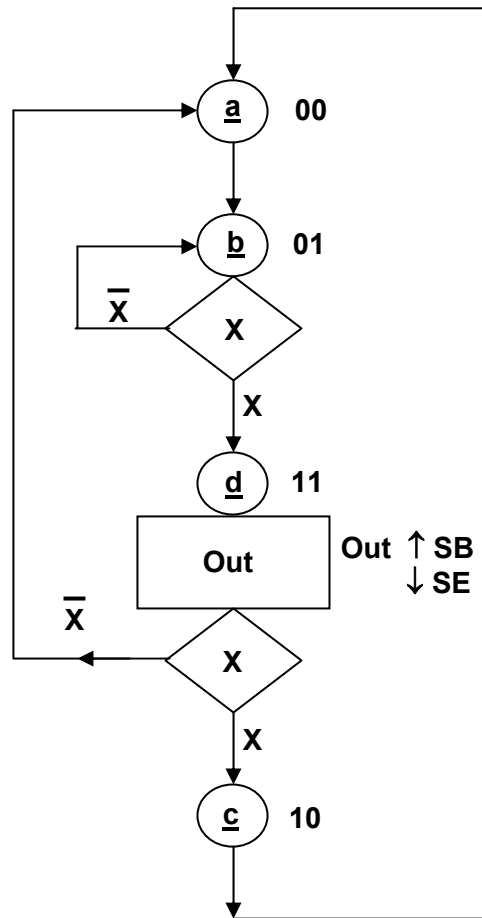


IFL – INPUT FORMING LOGIC
OFL – OUTPUT FORMING LOGIC

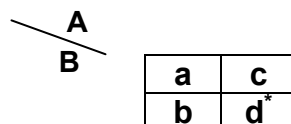
Consiste de uma lógica de entrada das variáveis e as saídas dos F/Fs utilizados como variáveis internas e a lógica combinacional formada por gates. O IFL gates consiste de portas lógicas AND de entrada e o OFL de gates AND de saída, com os sinais lógicos X e A e B.

Fila	Estado Atual		Entrada	Entradas F/Fs		Próximo Estado		Saída
	A	B	X	D _A	D _B	A	B	Out
1	0	0	0	0	1	0	1	0
2	0	0	1	0	1	0	1	0
3	0	1	0	0	1	0	1	0
4	0	1	1	1	1	1	1	0
5	1	0	0	0	0	0	0	0
6	1	0	1	0	0	0	0	0
7	1	1	0	0	0	0	0	1
8	1	1	1	1	0	1	0	1

O diagrama de estados a seguir mostra a dinâmica do circuito.



Para a implementação da máquina de estados um presente mapa de estados é utilizado para definir o nome dos estados. Como são 02 F/Fs somente 04 estados são possíveis. O mapa de estados presentes mostra estes estados possíveis em função das saídas dos F/Fs A e B. O mapa de estado define como **a** A = 0 e B = 0 ou 00, o estado **b** como 01, estado **c** como 10 e o estado **d** como 11. A designação de estado no mapa presente de estados será arbitrária, embora a designação de estados deve obedecer 02 princípios da adjacência para obter minimização do circuito. O asterisco no estado **d** identifica-o como estado de saída.



A implementação com F/Fs do tipo D pode ser realizada, com a tabela de estados presentes A e B e a variável de entrada X. A saída Out = A.B.

		AB			
		00	01	11	10
X	0	0	0	0	0
	1	0	1	1	0

$D_A = XB$

		AB			
		00	01	11	10
X	0	1	1	0	0
	1	1	1	0	0

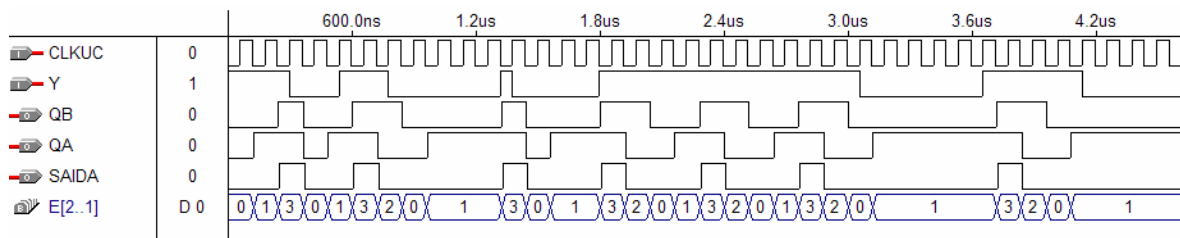
$D_B = \bar{A}$

As equações de estado e saída implementadas no MAX-PLUS II, são :

SUBDESIGN EXEMPLO_1_UC

```
(
CLKUC : INPUT;
Y : INPUT;
QA,QB,SAIDA : OUTPUT;
)
Variable E[2..1] : DFF;
BEGIN
E[].CLK=!CLKUC;
E[1] = !E2;
E[2] = (Y & E1);
QA = E1;
QB = E2;
SAIDA = E1 & E2;
END;
```

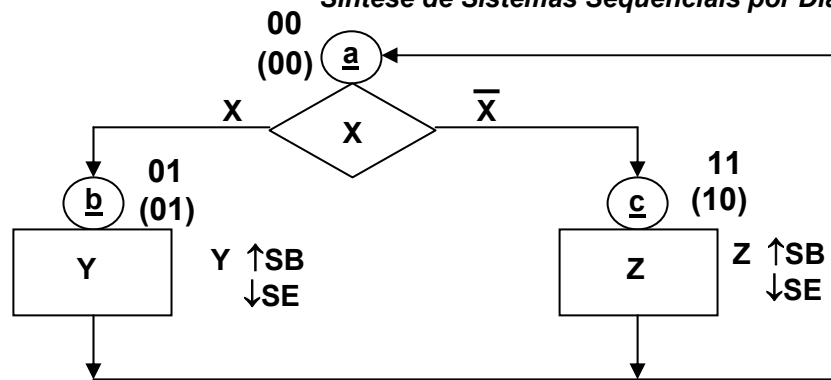
As formas de ondas apresentadas no MAX-PLUS II, são apresentadas, a seguir :



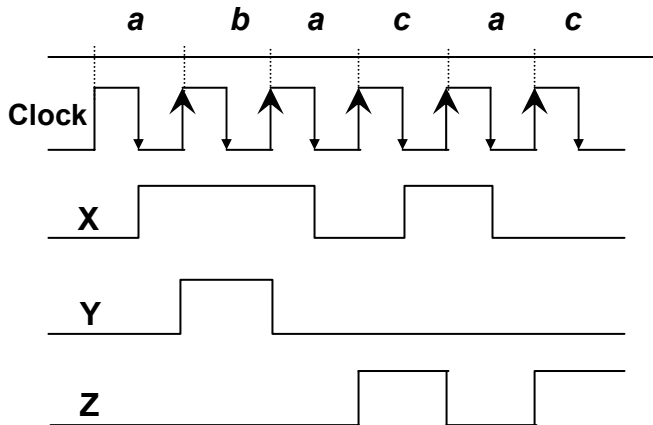
O próximo exemplo mostra um diagrama de estados com vários estados internos e várias saídas com muitos caminhos que resultam no seguinte diagrama de estados, apresentado na figura .

Exemplo : Construir um sistema seqüencial para uma linha serial de dados X sincronizada com a borda de descida do clock. Uma máquina de estados será projetada para detetar o próximo valor de X. Se X = 1, a saída Y deve ser Set. Se X = 0, a saída Z deve ser Set. O valor de saída será Set por um tempo de estado e então o próximo valor de X deve ser conferido.

SOLUÇÃO : O diagrama de estado a seguir reflete o comportamento especificado na figura . Como as trocas de estados do dado X ocorre com a borda de descida do clock, a troca de estado da variável X se dará somente com a transição negativa do clock. O sistema inicia no estado **a** e se X = 1 o sistema se move para o estado **b**; se X = 0, o sistema se move para o estado **c**, todas trocas de estados se darão com a transição positiva do clock. A saída em ambos estes estados existem por um tempo de clock após o qual o sistema retorna ao estado **a** para conferir o dado bit alternado. O diagrama de estado a seguir mostra a evolução do sistema seqüencial.



A solução do problema é de um circuito seqüencial pois a saída somente deve ser Set por um tempo de clock, mas poderia ser implementado com um circuito combinatório em conjunto com um sinal periódico do clock. As formas de ondas a seguir da figura mostram o comportamento do sistema com as variáveis X de entrada e as saídas Y e Z.

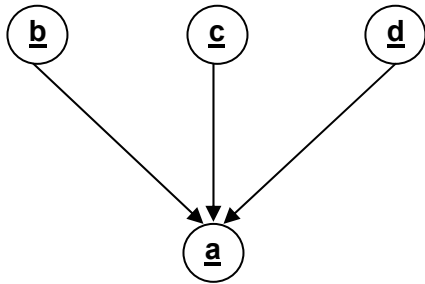


Os 03 estados que satisfazem as condições do problema, podem ser solucionados por 02 F/Fs A e B, onde A representa o MSB do código de estado. O projeto IFL, requer a designação de estado e será feito obedecendo os 02 princípios básicos para a minimização do circuito final. Usando métodos tradicionais para a implementação dos circuitos digitais combinacionais, os princípios da adjacências de estados leva a uma minimização final do circuito. Para comparar os resultados finais, o exemplo que estamos estudando será implementado obedecendo e não, os princípios das adjacências com designação correta e incorreta de estados.

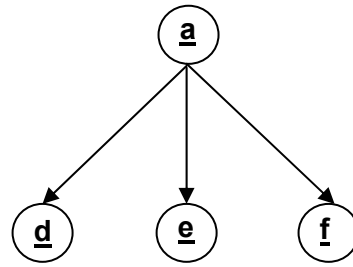
PRINCÍPIO 1 : Estados tendo o mesmo próximo estado para uma dada condição de entrada deve ter logicamente designação de estado adjacente.

PRINCÍPIO 2 : Estados que são os próximos estados de um único estado deve ter logicamente designações adjacentes.

Princípio 1



Princípio 2



Os estados **b, c, d** são adjacentes.

Os estados **d, e, f** são adjacentes.

Montamos 02 mapas de designações de estados, sendo uma correta e outra arbitrária para demonstração dos resultados. O mapa a) será corretamente designado obedecendo os princípios das adjacências conforme a seguir. Os estados **b** e **c**, obedecem o princípio 1, pois independente de qualquer condição estes levam para o mesmo estado **a**. E também obedecem o princípio 2 pois são os próximos estados **b** e **c** do produto de um único estado **a**. O mapa b) os estados serão montados arbitrariamente. Uma das designações corretas é apresentada no mapa a), mas existe outra solução corretamente designada.

		A	
		0	1
B	0	a	⊕
	1	b	c

		A	
		0	1
B	0	a	c
	1	b	⊕

a) Estados designados corretamente. Onde ⊕ é um estado irrelevante.

b) Estados designados arbitrariamente.

O próximo passo é desenvolver os mapas para os F/Fs A e B. Os códigos designados para os estados **a** – 00, **b** – 01, **c** – 11. Os estados apresentados entre parênteses no diagrama de estados são incorretamente escolhidos e são **a** – 00, **b** – 01 e **c** – 10.

a) Estados designados corretamente.

		A	
		0	1
B	0	\bar{X}	⊕
	1	0	0

F/F A

		A	
		0	1
B	0	1	⊕
	1	0	0

F/F B

b) Estados arbitrariamente designados.

		A	
		0	1
B	0	\bar{X}	0
	1	0	⊕

F/F A

		A	
		0	1
B	0	X	0
	1	0	⊕

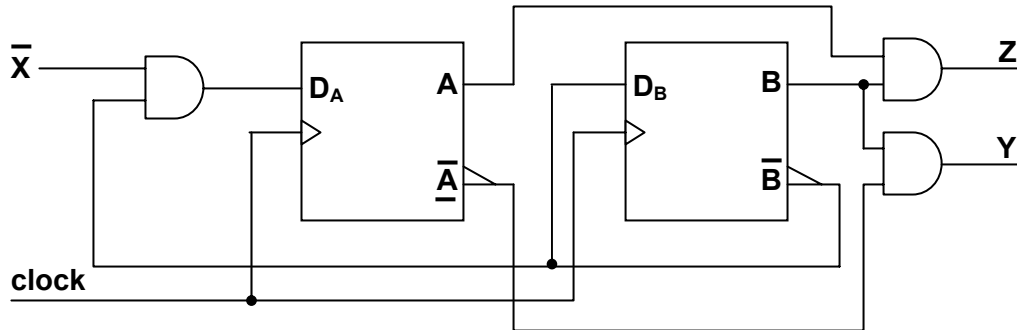
F/F B

O próximo passo é escolher o tipo de F/F a ser utilizado na implementação do circuito. Escolhemos inicialmente os F/Fs do tipo D, mais fácil de se implementar.

Para o item a) retiramos 02 funções D_A e D_B entradas de cada F/F A e B. Assim sendo :

$$D_A = \bar{X} \cdot \bar{B} \text{ e } D_B = \bar{B}$$

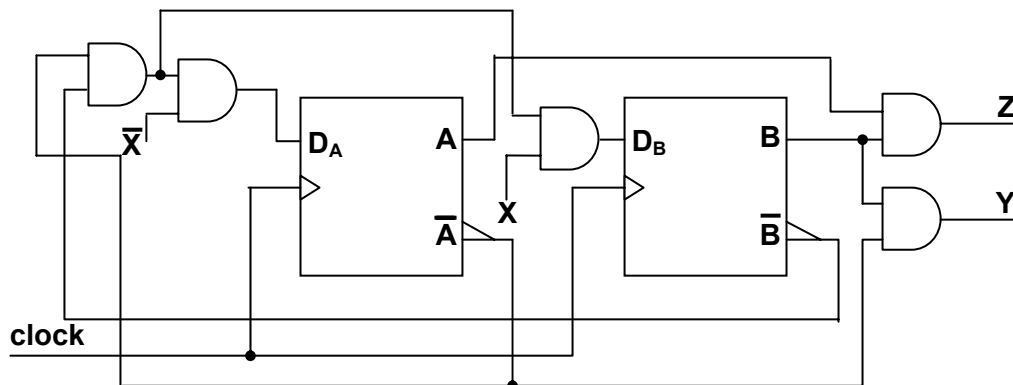
A implementação do circuito será :



Para o item b) retiramos 02 funções D_A e D_B entradas de cada F/F A e B. Assim sendo :

$$D_A = \bar{X} \cdot \bar{A} \cdot \bar{B} \text{ e } D_B = X \cdot \bar{A} \cdot \bar{B}$$

Pela expressão pode-se notar que não são minimizadas as expressões de D_A e D_B pois há mais termos. O circuito a seguir mostra a implementação do item b).



Uma outra forma de implementação do problema anterior é mostrada a seguir, onde após a designação de estados, a tabela de estados é assim montada.

ATUAIS	ENTRADA K		SAÍDAS	
	0	1	Y	Z
$Q_B Q_A$	$Q_B Q_A$	$Q_B Q_A$		
00	11	01	0	0
01	00	00	1	1
10	XX	XX	X	X
11	00	00	0	1

Após a montagem da tabela, o circuito combinatório a ser projetado é tirado do mapa de Karnaugh.

AB	00	01	11	10
X				
0	1	0	0	X
1	0	0	0	X

AB	00	01	11	10
X				
0	1	0	0	X
Pág. 1	1	0	0	X

$$D_A = \overline{X} \overline{B}$$

$$D_B = \overline{B}$$

As saídas Y e Z serão :

$$Y = \overline{Q_B} Q_A \quad \text{e} \quad Z = Q_A + Q_B$$

A tabela de estados presentes e futuros será montada a seguir.

Fila	Estado Atual		Entrada X	Entradas F/Fs		Próximo Estado		Saída	
	A	B		D _A	D _B	A	B	Y	Z
1	0	0	0	1	1	1	1	0	0
2	0	0	1	0	1	0	1	0	0
3	0	1	0	0	0	0	0	1	0
4	0	1	1	0	0	0	0	1	0
5	1	0	0	⊖	⊖	⊖	⊖	⊖	⊖
6	1	0	1	⊖	⊖	⊖	⊖	⊖	⊖
7	1	1	0	0	0	0	0	0	1
8	1	1	1	0	0	0	0	0	1

A implementação com outros tipos de F/Fs, como tipo JK é necessário da tabela de transição de estados ou tabela de excitação.

J	K	Q _n → Q _{n+1}
0	X	0 → 0
1	X	0 → 1
X	1	1 → 0
X	0	1 → 1

A tabela de estados presentes e futuros será montada a seguir para F/F JK.

Fila	Estado Atual		Entrada X	Próximo Estado		Entradas dos F/Fs			
	A	B		A	B	J _A	K _A	J _B	K _B
1	0	0	0	1	1	1	⊖	1	⊖
2	0	0	1	0	1	0	⊖	1	⊖
3	0	1	0	0	0	0	⊖	⊖	1
4	0	1	1	0	0	0	⊖	⊖	1
5	1	0	0	⊖	⊖	⊖	⊖	⊖	⊖
6	1	0	1	⊖	⊖	⊖	⊖	⊖	⊖
7	1	1	0	0	0	⊖	1	⊖	1
8	1	1	1	0	0	⊖	1	⊖	1

A próxima etapa é a implementação da lógica formadora de entrada dos F/Fs, construindo 04 mapas de Veitch Karnaugh.

X		0	1	0	⊖	⊖
		1	0	0	⊖	⊖

X		0	⊖	⊖	1	⊖
		1	⊖	⊖	1	⊖

$$J_A = \overline{B} \overline{X}$$

$$K_A = 1$$

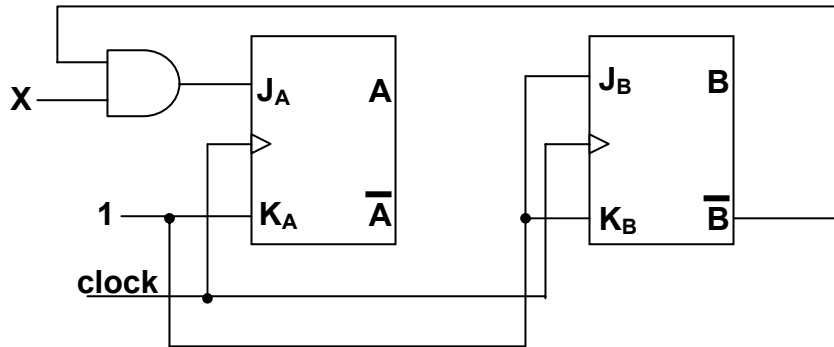
AB X		00	01	11	10	
		0	1	⊖	⊖	⊖
		1	1	⊖	⊖	⊖

AB X		00	01	11	10	
		0	⊖	1	⊖	⊖
		1	⊖	1	⊖	⊖

$$J_B = 1$$

$$K_B = 1$$

O circuito implementado com F/F do tipo JK é apresentado na figura .

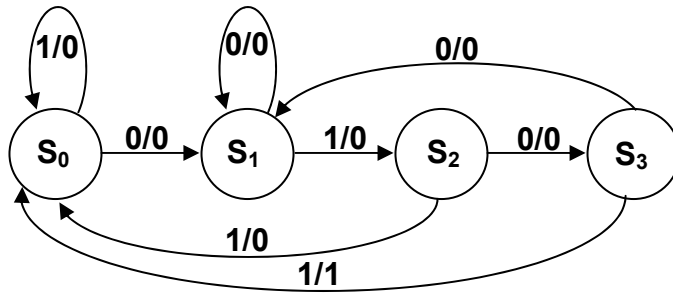


Exercício : Construir um circuito capaz de receber seqüencialmente através de uma única entrada X, (=0 ou =1), os 4 bits seriais de entrada. A saída produz um 1 sempre que a seqüência na entrada for igual a 0101. Pede-se :

- Diagrama de estados do conversor pelo modelo de Mealy.
- A tabela de estados pelo modelo Mealy
- Designação de estados.
- Implementação por F/F tipo D.
- Repetir item a) pelo modelo de Moore
- Repetir item b) pelo modelo de Moore

Solução :

- Diagrama de estados para o modelo de Mealy.



b) Implementação do sistema descrito acima pela tabela de estados, presentes e futuros e saída.

A primeira etapa a fazer é escolher os estados codificados, daí deveremos designar os estados S_0, S_1, S_2 e S_3 .

b.1) Designação de estados.

Designação de Estados S	Estados Atuais Q_1Q_0	Entrada Externa X		Entradas dos F/Fs D_1D_0		Saída Z	
		0	1			0	1
S0	00	01	00	01	00	0	0
S1	01	01	10	00	10	0	0
S2	10	11	00	11	00	0	0
S3	11	01	00	01	00	0	1

b.2) A implementação do sistema pelo Modelo de Mealy, a partir da tabela de estados usando F/F do tipo D.

Q_1Q_0	00	01	11	10
X 0	0	0	0	1
1	0	1	0	0

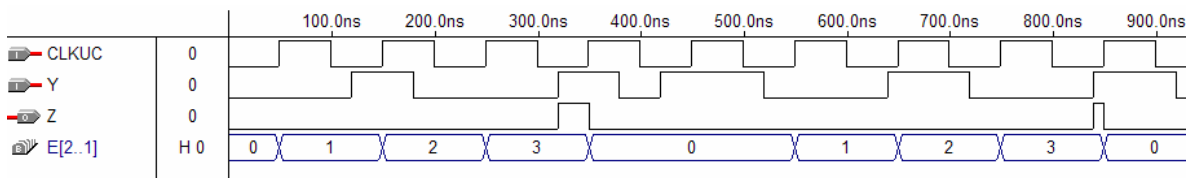
Q_1Q_0	00	01	11	10
X 0	1	1	1	1
1	0	0	0	0

$D_1 = Q_1'Q_0X + Q_1Q_0'X'$

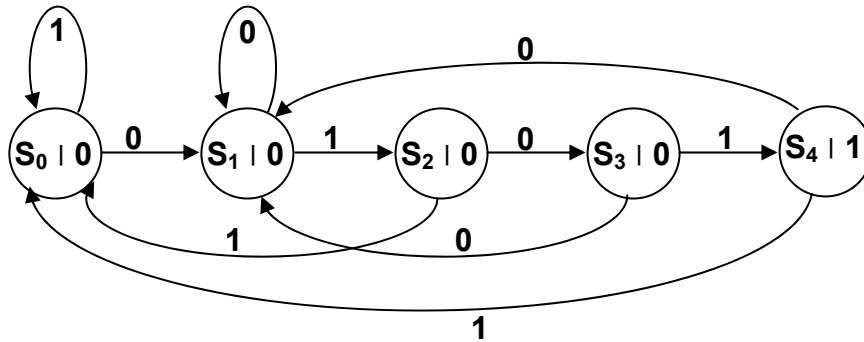
$Z = Q_1Q_0X$

$D_0 = X'$

c) O diagrama de tempo a seguir foi simulado no MAXPLUS II conforme as condições de entrada.



d) Diagrama de estados pelo Modelo de Moore.



Exercício : Construir um conversor BCD-Excesso 3. Sabendo-se que o conversor recebe seqüencialmente através de uma única entrada X, (=0 ou =1), os 4 bits seriais de entrada. A seqüência dos 04 bits de entrada é iniciando pelo bit LSB e a a cada bit de entrada deverá se ter uma saída do conversor. Pede- se :

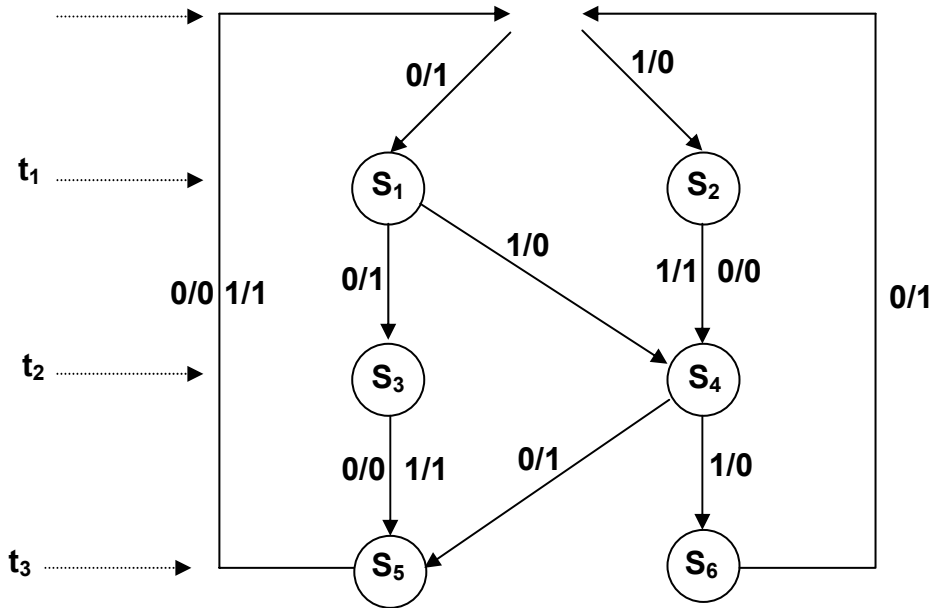
- a) Diagrama de estados do conversor pelo modelo de Mealy.
- b) A tabela de estados pelo modelo Mealy
- c) Designação de estados.
- d) Implementação por F/F tipo D.
- e) Repetir item a) pelo modelo de Moore
- f) Repetir item b) pelo modelo de Moore
- g) As formas de ondas com um clock de entrada borda de subida, dos estados e saídas para os 02 modelos.

Solução : Monta-se inicialmente a tabela da verdade de conversão de BCD-X3

Entrada				Saída			
T ₃	T ₂	T ₁	T ₀	T ₃	T ₂	T ₁	T ₀
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

a) Diagrama de estados Modelo Mealy

Serão sempre 02 caminhos alternativos para cada estado da rede, pois a entrada é de 1bit ou é 0 ou é 1 e no máximo 04 transições t_0 , t_1 , t_2 e t_3 nesta ordem. A montagem do diagrama de estados traz um macete, quando o nível lógico da entrada for resultar em um mesmo nível lógico na saída, a transição será para um mesmo estado na fase seguinte t_i , caso contrário existirá 02 transições sendo uma para cada estado da fase seguinte.



b) Tabela de Estados para o modelo de Mealy.

Estado Atual	Próximo Estado		Saída Z	
	Entrada X		X = 0	X = 1
	X = 0	X = 1		
S ₀	S ₁	S ₂	1	0
S ₁	S ₃	S ₄	1	0
S ₂	S ₄	S ₄	0	1
S ₃	S ₅	S ₅	0	1
S ₄	S ₅	S ₆	1	0
S ₅	S ₀	S ₀	0	1
S ₆	S ₀	-	1	-

c) Regras para designação de estados

Regra I : Estados os quais tem o mesmo próximo estado, para uma dada entrada devem ser designados adjacentes. (Olhar as colunas da tabela de estados).

Regra II : Estados os quais são os próximos estados do mesmo estado, devem ser designados adjacentes. (Olhar as filas da tabela de estados).

Regra III : Estados os quais tem a mesma saída para uma dada entrada, devem ser adjacentes.

Da tabela de estados, para as regras I, II e III, temos :

Regra I : (1,2), (3,4) e (5,6) → Na coluna X = 1, S₁ e S₂ tem estado futuro S₄; Na coluna X = 0, S₃ S₄ tem o estado futuro S₅; e para X = 0 S₅ S₆ tem o estado futuro S₀.

Regra II : (1,2), (3,4), (5,6) → S₁ S₂ são estados futuros de S₀; S₃ S₄ são estados futuros de S₁; S₅ S₆ são estados futuros de S₄.

Regra III : (0,1,4,6) e (2,3,5)

Designando – se os estados $S_0, S_1, S_2, S_3, S_4, S_5$ e S_6 , teremos :

		$Q_1 Q_2$			
		00	01	11	10
Q_3	0	S_0	S_6	S_4	S_1
	1	-	S_5	S_3	S_2

d) Implementação por F/F tipo D.

Estado Atual	Próximo Estado		Saída Z	
	Entrada X		X = 0	X = 1
	X = 0	X = 1		
$S_0 - 000$	100	101	1	0
$S_1 - 100$	111	110	1	0
$S_2 - 101$	110	110	0	1
$S_3 - 111$	011	011	0	1
$S_4 - 110$	011	010	1	0
$S_5 - 011$	000	000	0	1
$S_6 - 010$	000	---	1	-

		$Q_1 Q_2$			
		00	01	11	10
Q_3	X 00	1	0	0	1
	01	1	X	0	1
	11	X	0	0	1
	10	X	0	0	1

$$D_1 = Q_2'$$

		$Q_1 Q_2$			
		00	01	11	10
Q_3	X 00	0	0	1	1
	01	0	X	1	1
	11	X	0	1	1
	10	X	0	1	1

$$D_2 = Q_1$$

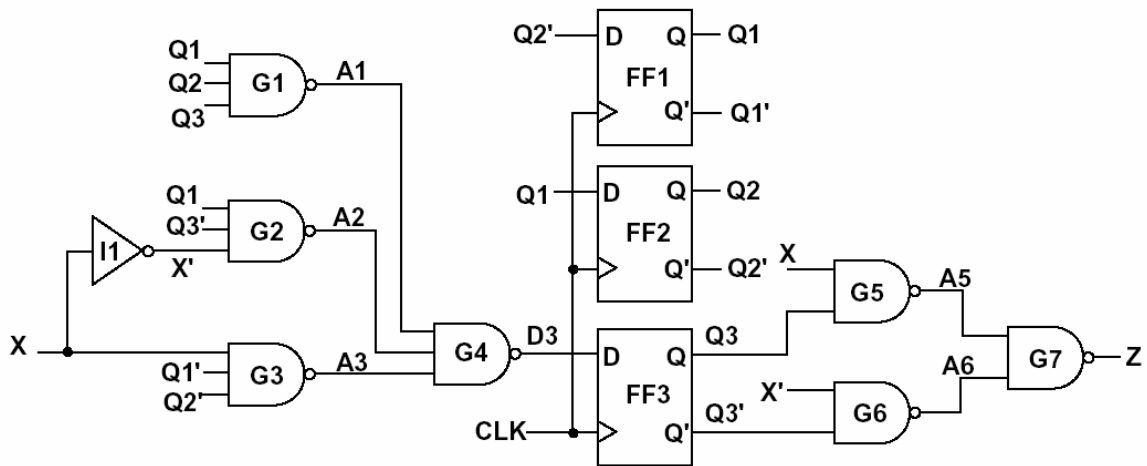
		$Q_1 Q_2$			
		00	01	11	10
Q_3	X 00	0	0	1	1
	01	1	X	0	0
	11	X	0	1	0
	10	X	0	1	0

$$D_3 = Q_1' Q_3' X + Q_1 Q_2 Q_3 + Q_1 Q_3' X'$$

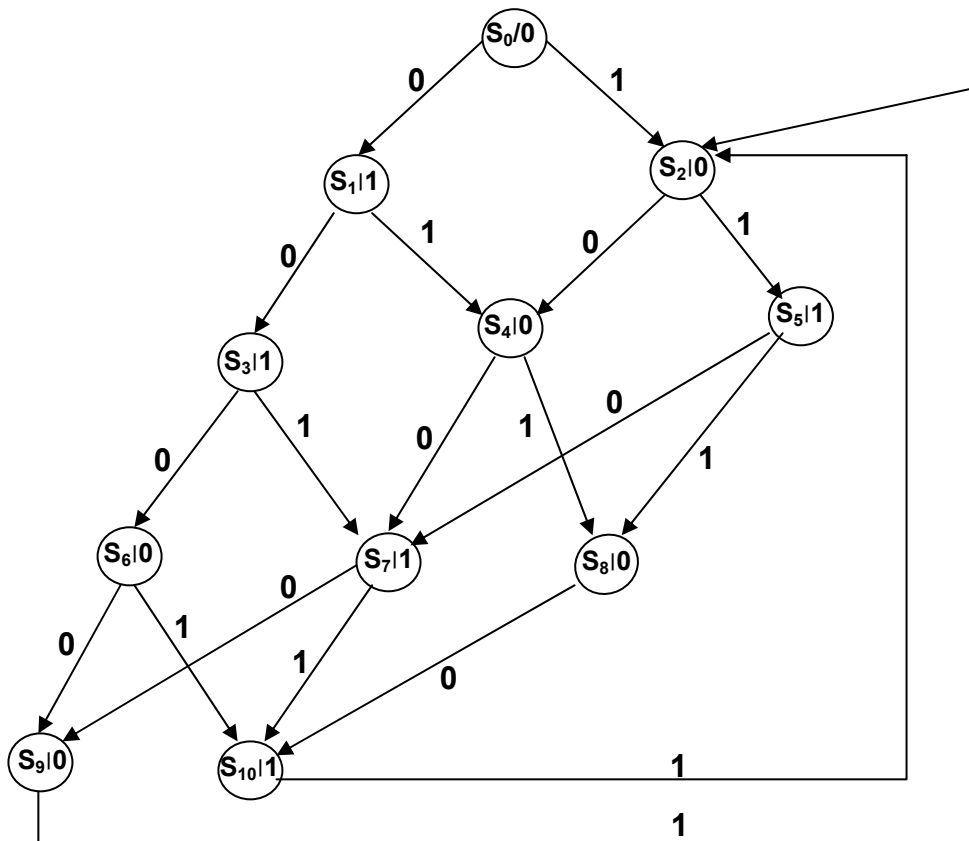
		$Q_1 Q_2$			
		00	01	11	10
Q_3	X 00	1	1	1	1
	01	0	X	0	0
	11	X	1	1	1
	10	X	0	0	0

$$Z = Q_3' X' + Q_3 X$$

O circuito desenhado com F/F do tipo D é apresentado a seguir,



e) Diagrama de Estados pelo modelo de Moore.



f) Tabela de Estados para o modelo Moore

Estado Atual	Próximo Estado		Saída Z
	Entrada X = 0	Entrada X = 1	
S ₀	S ₁	S ₂	0
S ₁	S ₃	S ₄	1
S ₂	S ₄	S ₅	0
S ₃	S ₆	S ₇	1
S ₄	S ₇	S ₈	0
S ₅	S ₇	S ₈	1

S ₆	S ₉	S ₁₀	0
S ₇	S ₉	S ₁₀	1
S ₈	S ₁₀	- - -	0
S ₉	S ₁	S ₂	0
S ₁₀	S ₁	S ₂	1

g) Designação de Estados para o modelo de Moore.

Regras para designação de estados

Regra I : Estados os quais tem o mesmo próximo estado, para uma dada entrada devem ser designados adjacentes. (Olhar as colunas da tabela de estados).

Regra II : Estados os quais são os próximos estados do mesmo estado, devem ser designados adjacentes. (Olhar as filas da tabela de estados).

Regra III : Estados os quais tem a mesma saída para uma dada entrada, devem ser adjacentes.

Da tabela de estados, para as regras I, II e III, temos :

Regra I : (0,9,10), (4,5) e (6,7) → Na coluna X = 0, S₀ S₉ e S₁₀ tem estado futuro S₁; Na coluna X = 1, S₄ S₅ tem o estado futuro S₇; e para X = 0 S₆ S₇ tem o estado futuro S₉.

Regra II : (1,2), (3,4), (4,5), (6,7), (7,8),(9,10) → S₁ S₂ são estados futuros de S₀ e de S₉ e de S₁₀; S₃ S₄ são estados futuros de S₁; S₄ S₅ são estados futuros de S₂; S₆ S₇ são estados futuros de S₃; S₇ S₈ são estados futuros de S₄; S₇ S₈ são estados futuros de S₅; S₉ S₁₀ são estados futuros de S₆ e de S₇;

Regra III : (0,2,4,6,8 e 9) e (1,3,5,7 e 10)

Designando – se os estados S₀,S₁,S₂,S₃,S₄,S₅,S₇, S₈, S₉ e S₁₀, teremos :

Q ₃ Q ₄ \ Q ₁ Q ₂		Q ₁ Q ₂			
		00	01	11	10
Q ₃ Q ₄	00	S ₉	S ₁₀	S ₈	-
	01	-	-	-	S ₅
	11	-	S ₁	S ₃	S ₄
	10	S ₀	S ₂	S ₇	S ₆

Q ₃ Q ₄ \ Q ₁ Q ₂		Q ₁ Q ₂			
		00	01	11	10
Q ₃ Q ₄	00	S ₀	S ₂	S ₇	S ₆
	01	-	S ₁	S ₃	S ₄
	11	-	-	-	S ₅
	10	S ₉	S ₁₀	S ₈	-

São apresentadas 02 soluções de designação de estados, conforme tabela de designações, escolheremos a primeira tabela.

A seguir montamos a tabela de estados com as designações escolhidas para cada estado.

Estado Atual	Próximo Estado Entrada X		Saída Z
	X = 0	X = 1	
S ₀ - 0010	0111	0110	0
S ₁ - 0111	1111	1011	1
S ₂ - 0110	1011	1001	0
S ₃ - 1111	1010	1110	1
S ₄ - 1011	1110	1100	0
S ₅ - 1001	1110	1100	1
S ₆ - 1010	0000	0100	0
S ₇ - 1110	0000	0100	1
S ₈ - 1100	0100	- - - -	0
S ₉ - 0000	0111	0110	0
S ₁₀ - 0100	0111	0110	1

Para X = 0, temos :

Q ₁ Q ₂		Q ₃ Q ₄			
		00	01	11	10
Q ₃ Q ₄	00	0	0	1	-
	01	-	-	-	1
	11	-	0	1	1
	10	0	0	1	1

$$D_1 = Q_1$$

Q ₁ Q ₂		Q ₃ Q ₄			
		00	01	11	10
Q ₃ Q ₄	00	0	1	1	-
	01	-	-	-	0
	11	-	1	1	0
	10	0	1	1	0

$$D_2 = Q_2$$

Q ₁ Q ₂		Q ₃ Q ₄			
		00	01	11	10
Q ₃ Q ₄	00	0	0	-	-
	01	-	-	-	0
	11	-	1	1	1
	10	0	1	1	1

$$D_3 = Q_3$$

Q ₁ Q ₂		Q ₃ Q ₄			
		00	01	11	10
Q ₃ Q ₄	00	0	0	-	-
	01	-	-	-	1
	11	-	1	1	1
	10	0	1	0	0

$$D_4 = Q_4$$

Para X = 1, temos :

Q ₁ Q ₂		Q ₃ Q ₄			
		00	01	11	10
Q ₃ Q ₄	00	0	0	-	-
	01	-	-	-	1
	11	-	1	1	1
	10	0	1	0	0

$$D_1 = Q_4 + Q_1'Q_2Q_3$$

Q ₁ Q ₂		Q ₃ Q ₄			
		00	01	11	10
Q ₃ Q ₄	00	1	1	-	-
	01	-	-	-	1
	11	-	0	1	1
	10	1	0	1	1

$$D_2 = Q_1 + Q_2' + Q_3'$$

Q ₁ Q ₂		Q ₃ Q ₄			
		00	01	11	10
Q ₃ Q ₄	00	1	1	0	-
	01	-	-	-	0
	11	-	1	1	0
	10	1	0	0	0

$$D_3 = Q_1'Q_2' + Q_2Q_4 + Q_1'Q_3'$$

Q ₁ Q ₂		Q ₃ Q ₄			
		00	01	11	10
Q ₃ Q ₄	00	0	0	0	-
	01	-	-	-	0
	11	-	1	0	0
	10	0	1	0	0

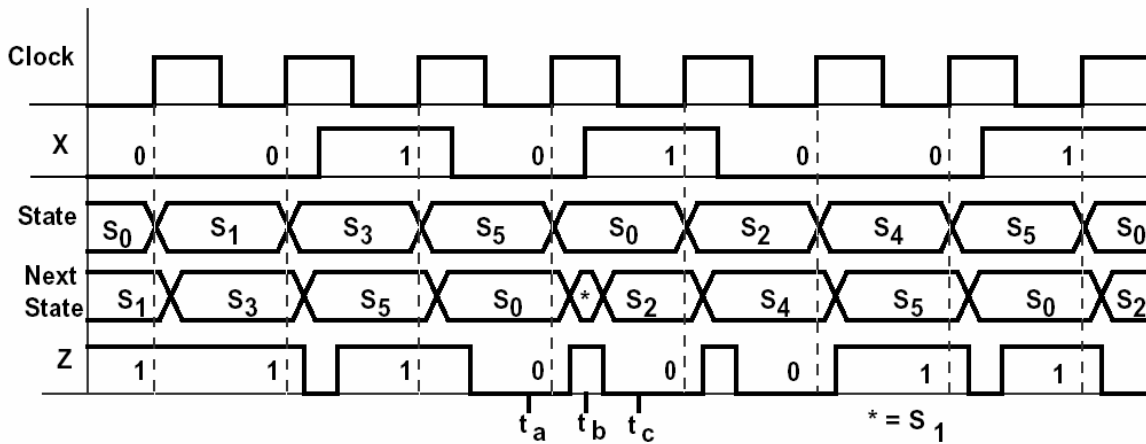
$$D_4 = Q_1'Q_2Q_3$$

Q_1Q_2		00	01	11	10
		0	1	0	-
Q_3Q_4	00	0	1	0	-
	01	-	-	-	1
	11	-	1	1	0
	10	0	0	1	0

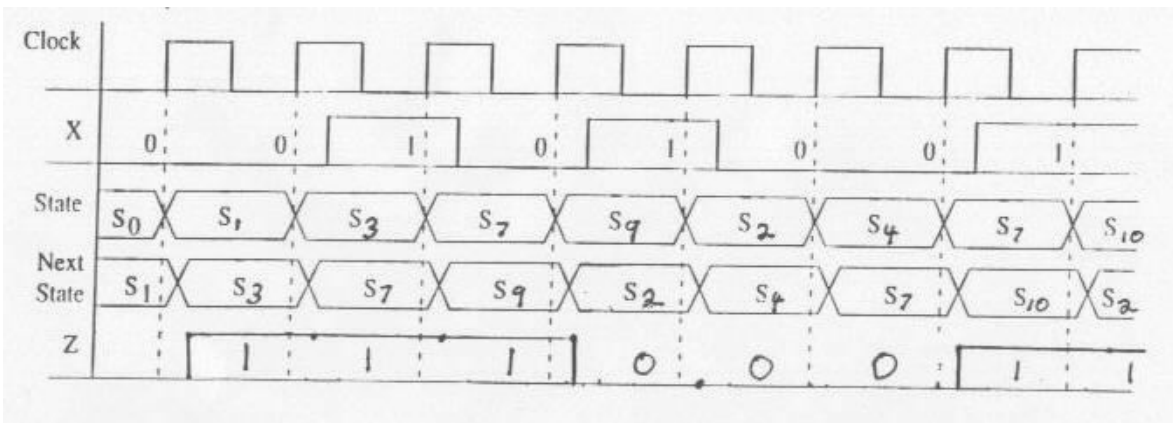
$$Z = Q_2Q_4 + Q_1Q_2Q_3 + Q_1'Q_2Q_3' + Q_3'Q_4$$

g) As formas de ondas dos modelos de Mealy e de Moore para a conversão BCD – X3

Mealy

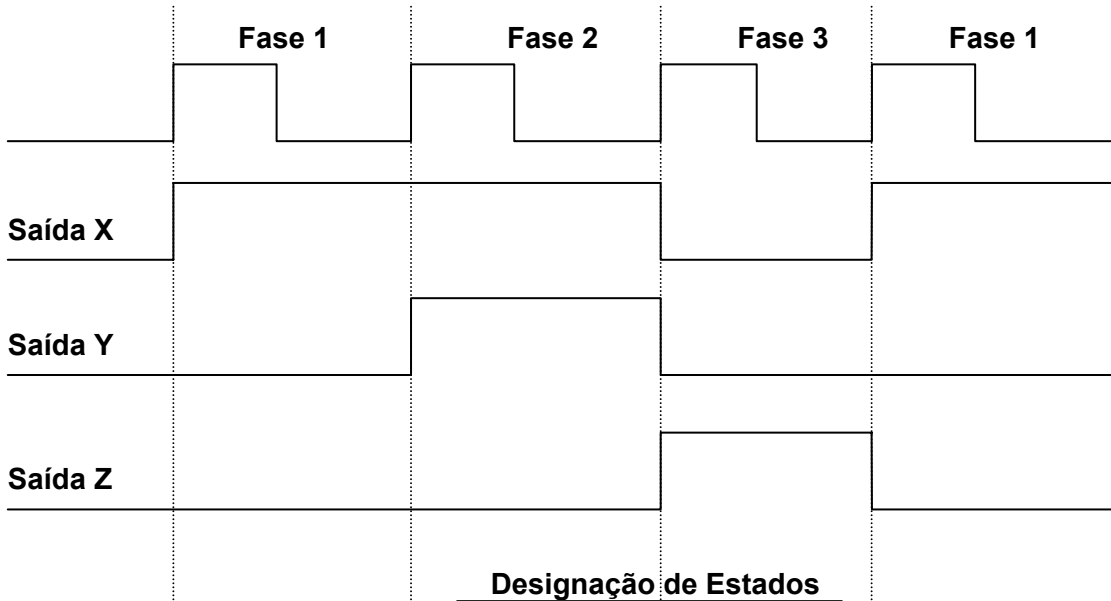


Moore



EXERCÍCIO : Usando F/Fs do tipo D e a designação de estados da tabela, projetar o sistema digital mínimo cujas saídas estão especificadas no diagrama de tempo. Para tanto determine :

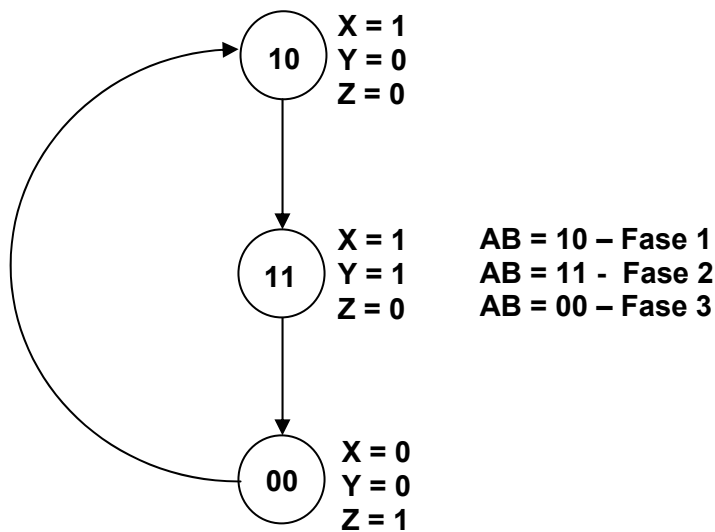
- As funções das entradas dos F/Fs;
- As funções lógicas das saídas X, Y e Z.



Designação de Estados

	Variáveis de Estado	
	A	B
Fase 1	1	0
Fase 2	1	1
Fase 3	0	0

Solução : Utilizando o diagrama de estados para a solução do problema, temos :



Designação de estados

	A	0	1
B	0	F ₃	F ₁
	1	X	F ₂

Implementação da máquina de estados utilizando F/F do tipo D.

A		0	1
B	0	1	1
1		X	0

$$D_A = B' = Q_B'$$

A		0	1
B	0	0	1
1		X	0

$$D_B = AB' = Q_A Q_B'$$

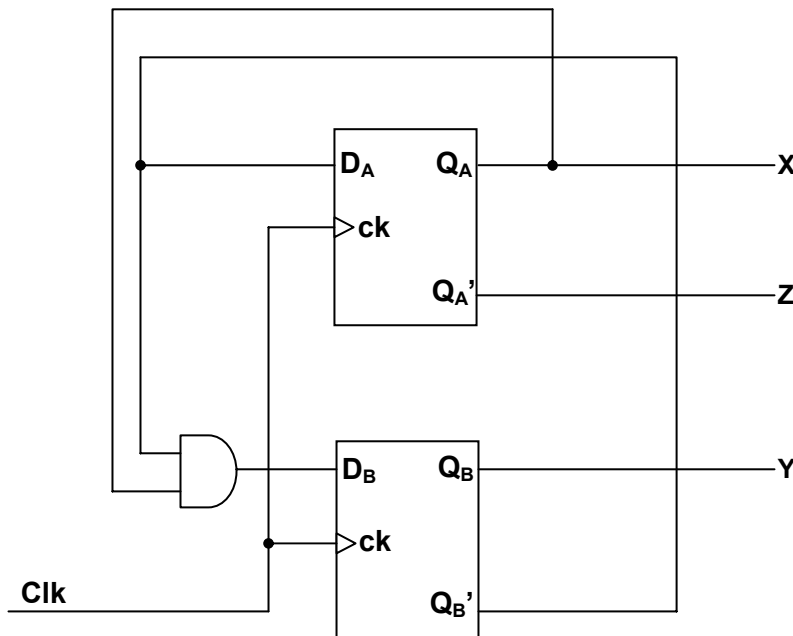
As saídas serão expressas por :

$$X = A = Q_A$$

$$Y = B = Q_B$$

$$Z = A' = Q_A'$$

A implementação por F/F do tipo D é apresentada a seguir.



EXEMPLO : Projetar um circuito que trabalha as seguintes especificações :

- O circuito possui uma única entrada, in, e uma única saída Out;
- Todas as trocas no circuito ocorrem na borda de subida do sinal de relógio;
- A saída Out é igual a 1 se durante 02 ciclos precedentes de relógio a entrada in foi igual a 1. Caso contrário, o valor de Out é igual a zero.

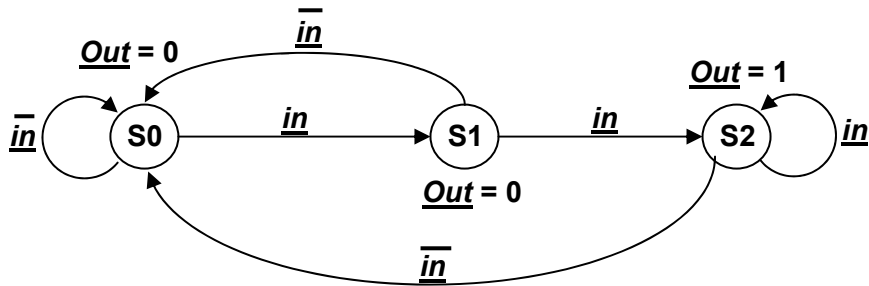
Seguindo-se a seqüência dos sinais entrada e saída.

Ciclo de relógio	:	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
In	:	0	1	0	1	1	0	1	1	1	0	1
Out:	:	0	0	0	0	0	1	0	0	1	1	0

SOLUÇÃO : O primeiro passo é elaborar a FSM, a fim de determinar quantos estados são necessários e quais transições são possíveis de um estado para outro. O diagrama de estados será a ferramenta de descrição da seqüência e o caminho é iniciar o diagrama

de estado com um estado inicial, o qual o circuito deve cair quando a energia é ligada ou quando um Reset é aplicado.

a) Diagrama de estados



c) Implementação por F/F do tipo D.

Designação de estados.

		0	1
Q ₁	Q ₀		
0		S0	S2
1		S1	X

Os F/Fs D₁ e D₀ são implementados da forma :

		00	01	11	10
Q ₁ Q ₀	in				
0		0	0	0	0
1		0	1	0	1

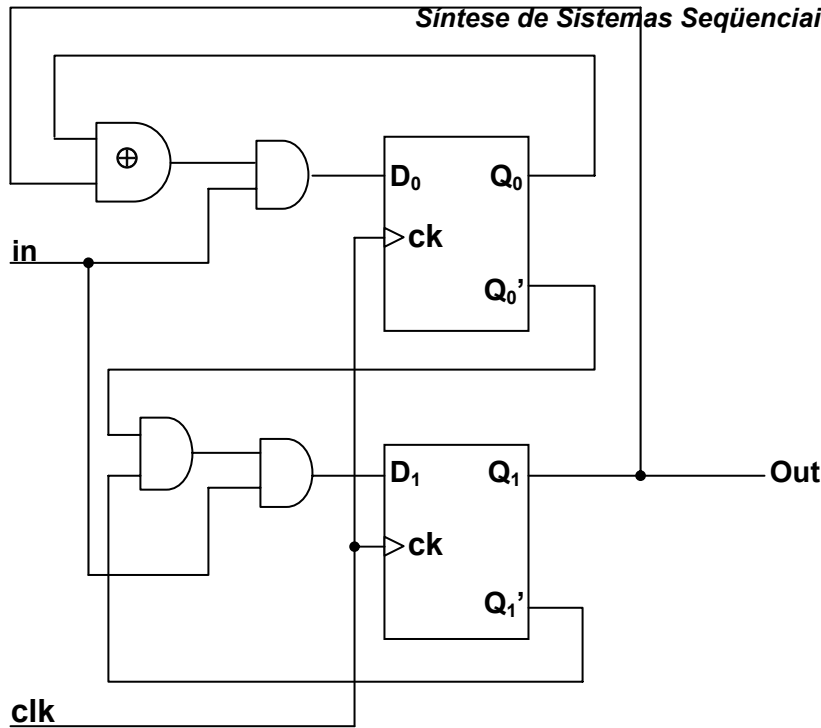
		00	01	11	10
Q ₁ Q ₀	in				
0		0	0	0	0
1		1	0	0	0

$$D_0 = \underline{in} (Q_1 \oplus Q_0) .$$

$$D_1 = \underline{in} \bar{Q}_1 \cdot \bar{Q}_0 .$$

$$\underline{Out} = Q_1$$

d) O circuito implementado fica :



EXEMPLO : Projetar um circuito que trabalha as seguintes especificações :

- 1) O circuito possui uma única entrada, in , e uma única saída Out ;
- 2) Todas as trocas no circuito ocorrem na borda de subida do sinal de relógio;
- 3) A saída Out é igual a 1 se durante 03 ciclos precedentes de relógio a entrada in foi igual a 0. Caso contrário, o valor de Out é igual a zero. Pede-se :

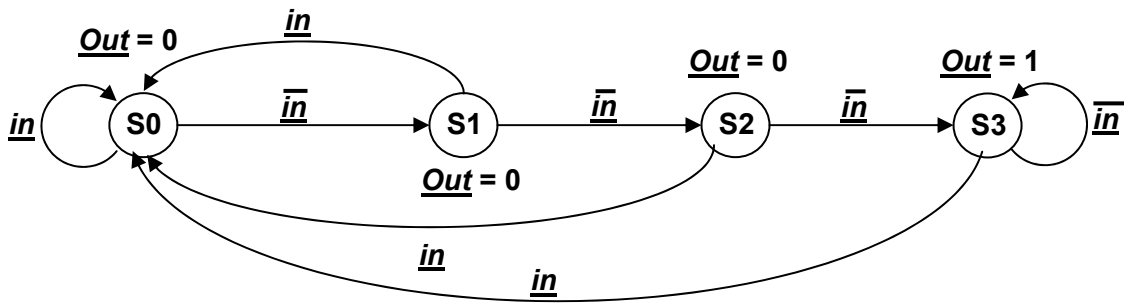
- a) Modelagem do problema por Moore.
- b) Implementação da FSM usando F/F do tipo D.
- c) Implementação da FSM usando ROM.
- d) Modelagem do problema por Mealy.
- e) Implementação da FSM usando F/F do tipo D.
- f) Implementação da FSM usando ROM.
- g) Implementação com o MAX-PLUS.

Seguindo-se a seqüência dos sinais entrada e saída.

Ciclo de relógio	:	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10		
		In	:	0	0	1	0	0	0	1	1	1	0	1
		Out:	:	0	0	0	0	0	1	0	0	0	0	0

SOLUÇÃO : O primeiro passo é elaborar a FSM, a fim de determinar quantos estados são necessários e quais transições são possíveis de um estado para outro. O diagrama de estados será a ferramenta de descrição da seqüência e o caminho é iniciar o diagrama de estado com um estado inicial, o qual o circuito deve cair quando a energia é ligada ou quando um Reset é aplicado.

a) Diagrama de estados pelo Modelo de Moore



b) Implementação por F/F do tipo D .
 b.1) A designação de estados

Sendo $m = 4$ estados $S(3..0)$, $m \leq 2^n \Rightarrow 2^n \geq 4 \Rightarrow n = 2$. Assim Q_1 e Q_0 , o qual :

		Q_1	
Q_0		0	1
0	S0	S3	
1	S1	S2	

Estado Atual	Próximo Estado Entrada in		Saída OUT
	in = 0	in = 1	
S ₀ - 00	S ₁ - 01	S ₀ - 00	0
S ₁ - 01	S ₂ - 11	S ₀ - 00	0
S ₃ - 10	S ₃ - 10	S ₀ - 00	1
S ₂ - 11	S ₃ - 10	S ₀ - 00	0

Os F/Fs D_1 e D_0 são implementados da forma :

		Q_1Q_0			
in		00	01	11	10
0		0	1	1	1
1		0	0	0	0

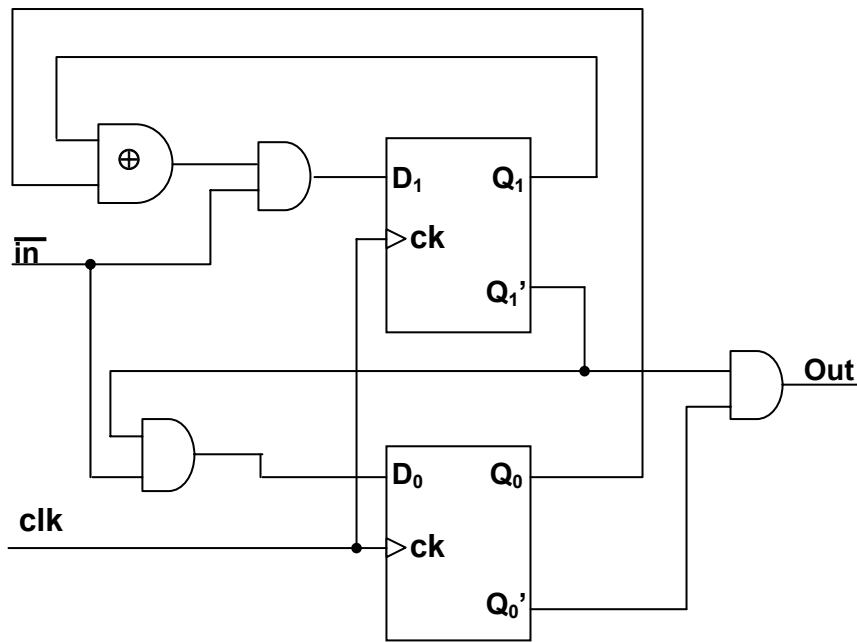
		Q_1Q_0			
in		00	01	11	10
0		1	1	0	0
1		0	0	0	0

$$D_1 = \overline{in} (Q_1 \oplus Q_0) .$$

$$\underline{Out} = Q_1 \overline{Q_0}$$

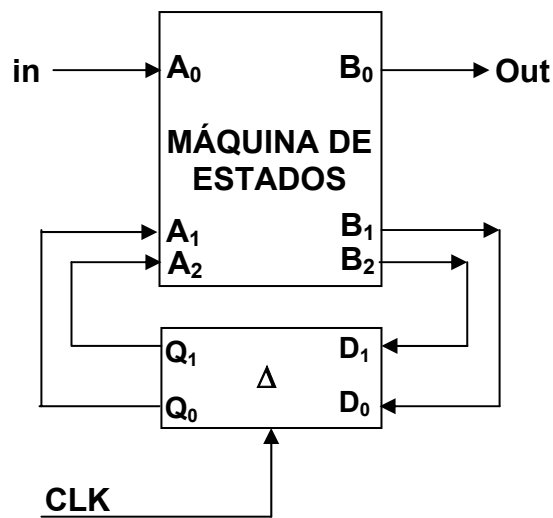
$$D_0 = \overline{in} \overline{Q_1} .$$

O circuito implementado fica :



c) Implementação da FSM com ROM

c.1) Representação da FSM.

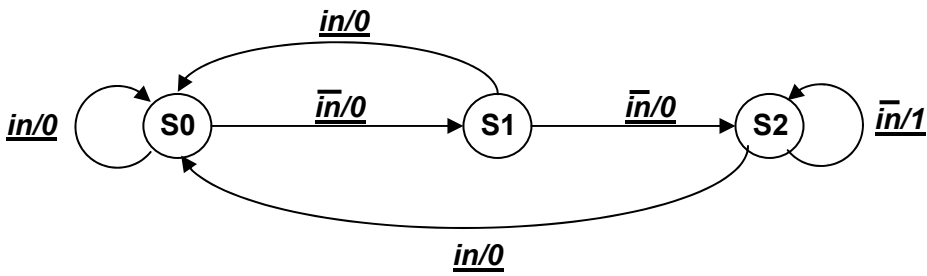


A tabela da ROM, será :

A ₂	A ₁	A ₀	B ₂	B ₁	B ₀	Endereço	Conteúdo
0	0	0	0	1	0	0	2
0	0	1	0	0	0	1	0
0	1	0	1	1	0	2	6
0	1	1	0	0	0	3	0
1	0	0	1	0	1	4	5
1	0	1	0	0	1	5	1
1	1	0	1	0	0	6	4
1	1	1	0	0	0	7	0

C = 8 X 3

d) Diagrama de estados pelo Modelo de Mealy



e) Implementação por F/F do tipo D .

e.1) A designação de estados

Sendo $m = 4$ estados $S(3..0)$, $m \leq 2^n \Rightarrow 2^n \geq 4 \Rightarrow n = 2$. Assim Q_1 e Q_0 , o qual :

Q_1		0	1
Q_0		0	1
0		S0	X
1		S1	S2

Estado Atual	Próximo Estado Entrada in		Saída OUT	
	in = 0	in = 1	in = 0	in = 1
S ₀ - 00	S ₁ - 01	S ₀ - 00	0	0
S ₁ - 01	S ₂ - 11	S ₀ - 00	0	0
S ₃ - 10	S ₀ - 00	S ₀ - 00	0	0
S ₂ - 11	S ₂ - 11	S ₀ - 00	1	0

Os F/Fs D_1 e D_0 são implementados da forma :

$Q_1 Q_0$		00	01	11	10
\overline{in}		0	1	1	0
		0	0	0	0

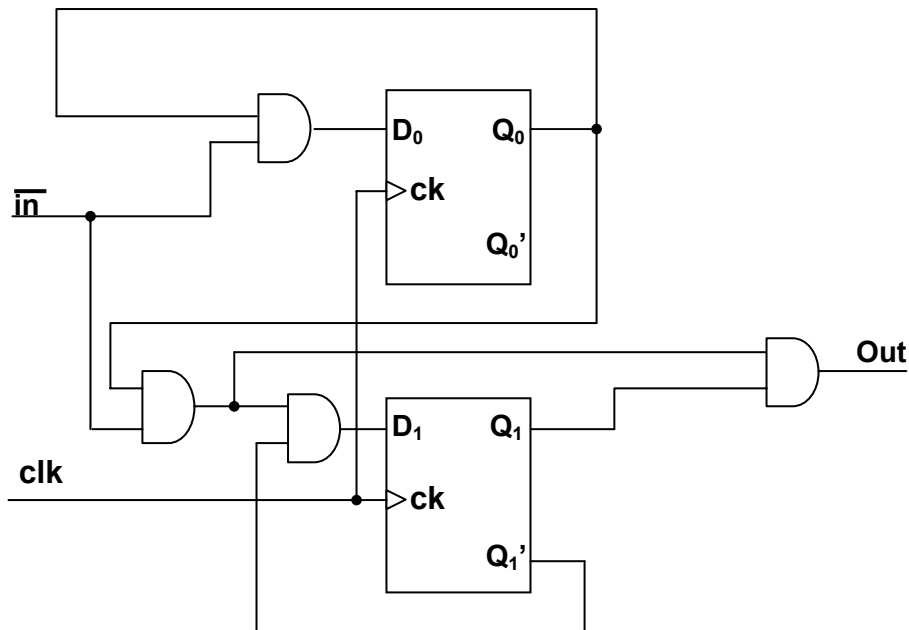
$$D_1 = \overline{in} \cdot Q_0$$

$$\underline{Out} = Q_1 Q_0 \cdot \overline{in}$$

$Q_1 Q_0$		00	01	11	10
\overline{in}		1	1	1	0
		0	0	0	0

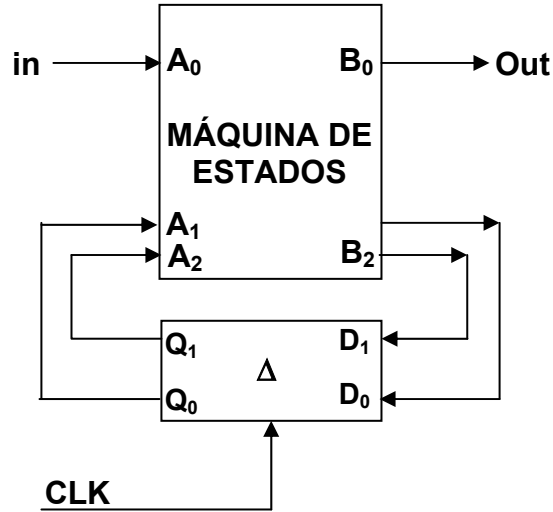
$$D_0 = \overline{in} \cdot (\overline{Q_1} \cdot Q_0)$$

O circuito implementado fica :



f) Implementação da FSM com ROM

f.1) Representação da FSM.



A tabela da ROM será :

A_2	A_1	A_0	B_2	B_1	B_0	Endereço	Conteúdo
0	0	0	0	1	0	0	2
0	0	1	0	0	0	1	0
0	1	0	1	1	0	2	6
0	1	1	0	0	0	3	0
1	0	0	1	0	0	4	4
1	0	1	0	0	0	5	0
1	1	0	1	1	1	6	7
1	1	1	0	0	0	7	0

$C = 8 \times 3$

g.1) O programa realizado no MAX-PLUS temos :

```

SUBDESIGN sequencia_000_moore
(
    in,clk : INPUT;
    Out    : OUTPUT;
)
VARIABLE
ss: MACHINE OF BITS (Out)
    WITH STATES (s0 = 0,
                 s1=0,
                 s2=0,
                 s3=1);

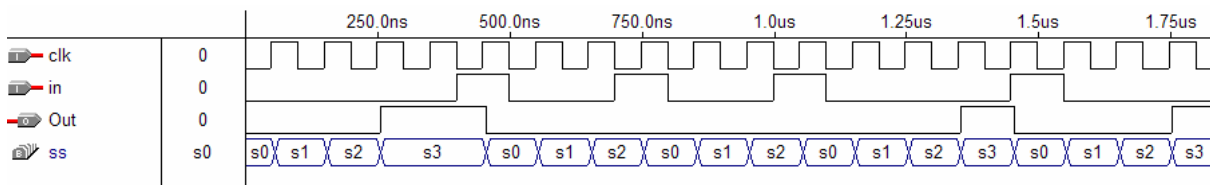
BEGIN
    ss.clk = clk;

    TABLE
    ss,          in    => ss;

    s0,          1    =>  s0;
    s0,          0    =>  s1;
    s1,          1    =>  s0;
    s1,          0    =>  s2;
    s2,          1    =>  s0;
    s2,          0    =>  s3;
    s3,          1    =>  s0;
    s3,          0    =>  s3;

    END TABLE;
END;
    
```

g.2) As formas de ondas pelo Modelo de Moore são apresentadas a seguir :



g.3) O projeto realizado no MAX-PLUS temos :

```

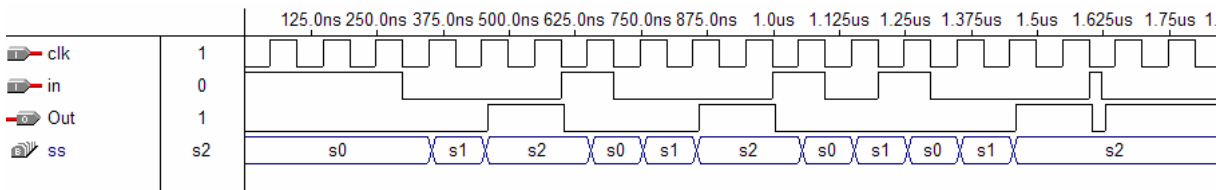
SUBDESIGN sequencia_000_mealy
(
    in,clk : INPUT;
    Out    : OUTPUT;
)
VARIABLE
ss: MACHINE WITH STATES (s0,s1,s2);
BEGIN
    ss.clk = clk;

    TABLE
    ss,          in    => Out,          ss;

    s0,          1    =>  0,            s0;
    s0,          0    =>  0,            s1;
    s1,          1    =>  0,            s0;
    s1,          0    =>  0,            s2;
    s2,          1    =>  0,            s0;
    s2,          0    =>  1,            s2;
    
```

END TABLE;
END;

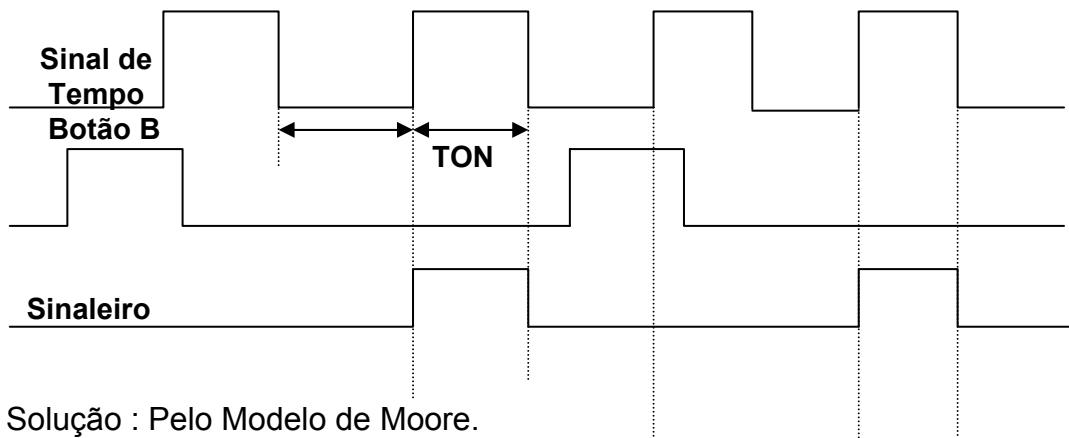
g.4) As formas de ondas pelo Modelo de Mealy são apresentadas a seguir :



Exercício : Construir um comando de um sinaleiro. Dispõe-se de um sinal de tempo o qual sempre abre o sinaleiro por um dado período de tempo, para a passagem de pedestre. Dispõe-se para abertura do sinaleiro, um botão de pedestre que quando $B = 1$, o sinaleiro deverá parar sob certas condições descritas a seguir.

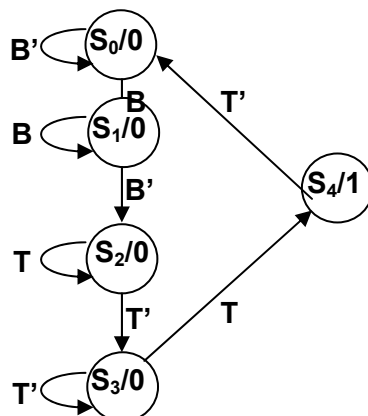
O sinaleiro somente abre a passagem para os pedestres, B tiver sido pressionado em qualquer instante e quando o sinal de tempo T estiver iniciando o TON e o botão B já tiver sido liberado pelo pedestre. Caso contrário o sinaleiro não abre a passagem para os pedestres e aguarda até satisfazer as condições para a abertura do sinaleiro. Pede-se :

- Definir as variáveis de entrada e saída e a lógica das variáveis.
- Modelagem do comando por diagrama de estados, modelo de Moore.
- Implementação por F/F do tipo D.
- A esquematização .gdf (Max-Plus).
- A simulação .scf (Max-Plus).

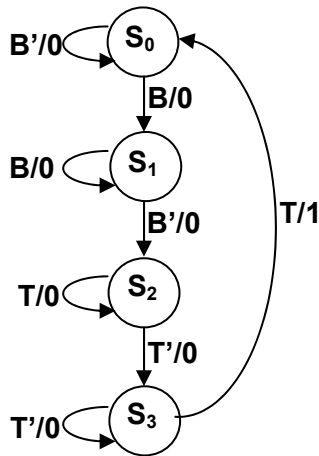


Solução : Pelo Modelo de Moore.

- Variáveis de entrada : T,B
- Variáveis de saída : S



b) Pelo Modelo de Mealy



ATUAL		FUTURO ENTRADA B,T				SAÍDAS ENTRADA B,T			
Q ₁	Q ₀	00	01	10	11	00	01	10	11
0	0	00	00	01	01	0	0	0	0
0	1	10	10	01	01	0	0	0	0
1	0	11	10	11	10	0	0	0	0
1	1	11	00	11	00	0	1	0	1

Q ₁ Q ₀	00	01	11	10
BT 00	0	1	1	1
01	0	1	0	1
11	0	0	0	1
10	0	0	1	1

$$D_1 = Q_1'Q_0B' + Q_1T' + Q_1Q_0'$$

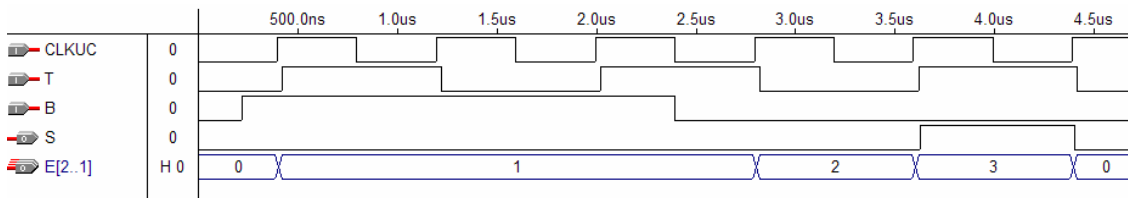
Q ₁ Q ₀	00	01	11	10
BT 00	0	0	1	1
01	0	0	0	0
11	1	1	0	0
10	1	1	1	1

$$D_0 = Q_1'B + Q_1T'$$

Q ₁ Q ₀	00	01	11	10
BT 00	0	0	0	0
01	0	0	0	0
11	0	0	1	0
10	0	0	0	0

$$S = Q_1Q_0T$$

As formas de ondas simuladas no MAX-PLUS são mostradas a seguir.



EXEMPLO : Projetar um circuito de alarme de carro, com as seguintes características. Quando uma porta ou janela é aberta Ch = 0, um temporizador com 30s deve ser disparado e no término do tempo de 30s o alarme é disparado intermitentemente de 1s em 1s. O alarme é desarmado por ativar uma chave K. Pedem-se :

- Definição das Variáveis de entrada e saída e alógica de funcionamento;
- Representação em blocos do processo;
- Diagrama de estados do alarme;
- Representação em bloco do sistema por máquina de estados;
- Implementação por máquina de estados usando F/F do tipo JK;
- Implementação por máquina de estados usando ALP.

SOLUÇÃO :

- Variáveis de entrada e saída do processo.

Variáveis de Entrada : Ch,K, FT
Variáveis de Saída : LT, AL.

A lógica das variáveis do processo.

Variáveis de entrada :

Variáveis de saída :

Ch = 0 ⇒ Porta Aberta.

AL = 0 ⇒ Alarme desligado.

Ch = 1 ⇒ Porta Fechada.

AL = 1 ⇒ Alarme ligado.

K = 0 ⇒ Desarma alarme.

LT = 0 ⇒ Temporizador desligado

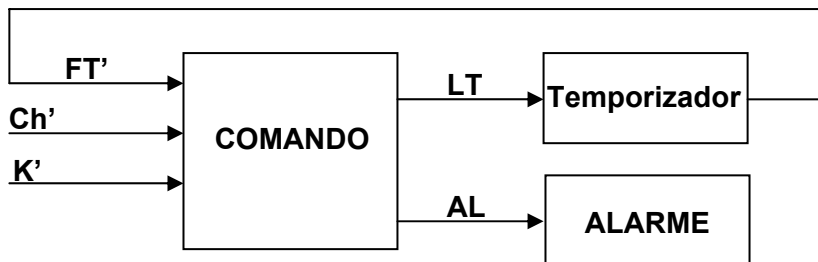
K = 1 ⇒ Alarme em atividade.

LT = 1 ⇒ Temporizador ligado.

FT = 0 ⇒ Fim temporização.

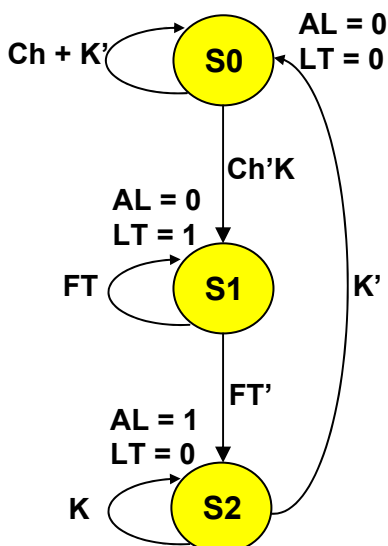
FT = 1 ⇒ Temporização inacabada.

b) Representação em bloco do processo.



c) Diagrama de estados do processo.

O diagrama de estados deste processo deverá ser montado conforme as condições impostas pelo problema. O diagrama de estados parte inicialmente de um estado inicial S0, que de acordo com o circuito é o estado onde o sistema quando energizado parte inicialmente.



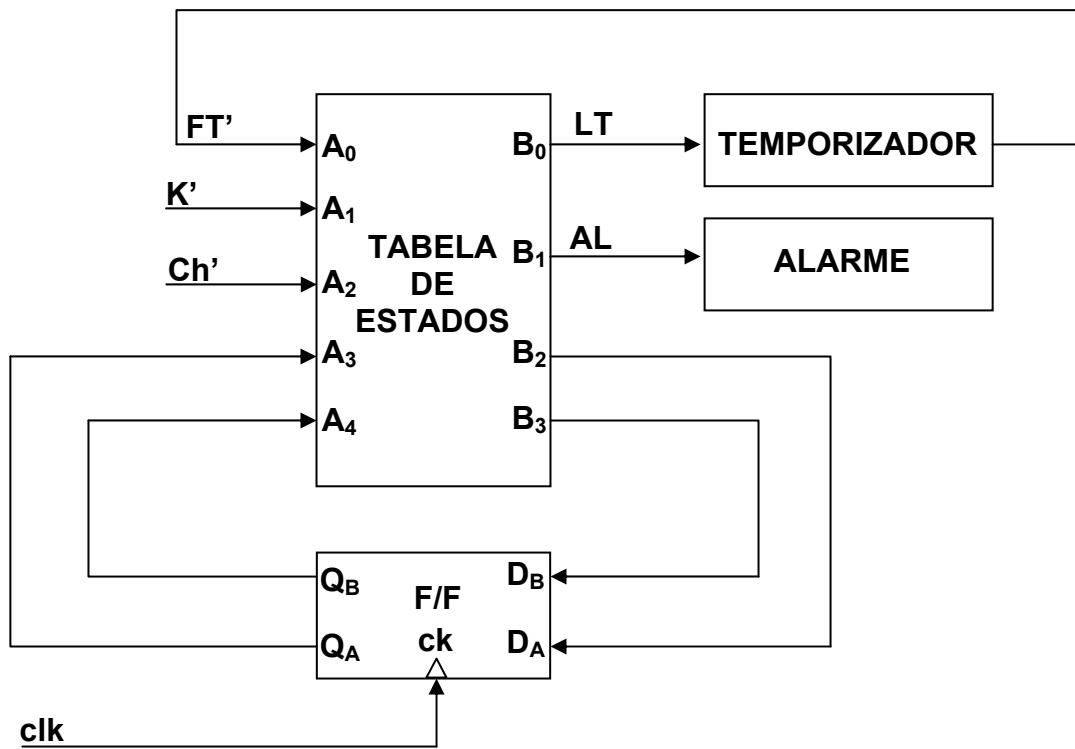
Designação de Estados
Por designação de estados teremos :

		Designação de Estados	
		0	1
Q _B	Q _A 0	S0	S2
	Q _A 1	S1	X

Os estados conforme a designação são codificados.

S0 = 0- 00, S1 = 1- 01 e S2 = 2-10

e) Representação em bloco por máquina de estados.



f) Implementação por F/F tipo JK

A tabela da verdade mostra a tabela da verdade e de excitação do JK.

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	Q_n

J	K	$Q_n \rightarrow Q_{n+1}$
0	X	$0 \rightarrow 0$
1	X	$0 \rightarrow 1$
X	1	$1 \rightarrow 0$
X	0	$1 \rightarrow 1$

A seguir é montada a tabela de estados presentes e futuros e saída.

Fila	Estado Atual					Entradas Externas		Próximo Estados		Entradas dos F/Fs				Saídas	
	Q _A	Q _B	Ch	K	FT	Q _A	Q _B	J _A	K _A	J _B	K _B	AL	LT		
00	0	0	0	0	0	0	0	0	X	0	X	0	0		
01	0	0	0	0	1	0	0	0	X	0	X	0	0		
02	0	0	0	1	0	0	1	0	X	1	X	0	0		
03	0	0	0	1	1	0	1	0	X	1	X	0	0		
04	0	0	1	0	0	0	0	0	X	0	X	0	0		
05	0	0	1	0	1	0	0	0	X	0	X	0	0		
06	0	0	1	1	0	0	0	0	X	0	X	0	0		
07	0	0	1	1	1	0	0	0	X	0	X	0	0		
08	0	1	0	0	0	1	0	1	X	X	1	0	1		
09	0	1	0	0	1	0	1	0	X	X	0	0	1		
10	0	1	0	1	0	1	0	1	X	X	1	0	1		
11	0	1	0	1	1	0	1	0	X	X	0	0	1		
12	0	1	1	0	0	1	0	1	X	X	1	0	1		
13	0	1	1	0	1	0	1	0	X	X	0	0	1		
14	0	1	1	1	0	1	0	1	X	X	1	0	1		
15	0	1	1	1	1	0	1	0	X	X	0	0	1		
16	1	0	0	0	0	1	0	X	1	0	X	1	0		
17	1	0	0	0	1	1	0	X	1	0	X	1	0		
18	1	0	0	1	0	1	0	1	X	0	X	1	0		
19	1	0	0	1	1	1	0	1	X	0	X	1	0		
20	1	0	1	0	0	0	0	X	1	0	X	1	0		
21	1	0	1	0	1	0	0	X	1	0	X	1	0		
22	1	0	1	1	0	1	0	1	X	0	X	1	0		
23	1	0	1	1	1	1	0	1	X	0	X	1	0		
24	1	1	0	0	0	0	0	X	1	X	1	0	0		
25	1	1	0	0	1	0	0	X	1	X	1	0	0		
26	1	1	0	1	0	0	0	X	1	X	1	0	0		
27	1	1	0	1	1	0	0	X	1	X	1	0	0		
28	1	1	1	0	0	0	0	X	1	X	1	0	0		
29	1	1	1	0	1	0	0	X	1	X	1	0	0		
30	1	1	1	1	0	0	0	X	1	X	1	0	0		
31	1	1	1	1	1	0	0	X	1	X	1	0	0		

A fim de implementar o circuito de alarme serão construídos 04 mapas de Karnaugh com 05 entradas sendo 03 entradas externas e 02 estados internos Q_A e Q_B, e 04 saídas para os 02 F/Fs J e K.

As equações de saída serão : $AL = Q_A Q_B'$ e $LT = Q_A' Q_B$

Ch	Q _A Q _B		FT			
	K		00	01	11	10
000			0	1	X	X
001			0	0	X	X
011			0	0	X	1
010			0	1	X	1
110			0	1	X	1
111			0	0	X	1
101			0	0	X	X
100			0	1	X	X

$$J_A = Q_A + Q_B FT'$$

Ch	Q _A Q _B		FT			
	K		00	01	11	10
000			X	X	1	1
001			X	X	1	1
011			X	X	1	X
010			X	X	1	X
110			X	X	1	X
111			X	X	1	X
101			X	X	1	1
100			X	X	1	1

$$K_A = 1$$

Ch	Q _A Q _B		FT			
	K		00	01	11	10
000			0	0	X	X
001			0	0	X	X
011			1	0	X	X
010			1	0	X	X
110			0	0	X	X
111			0	0	X	X
101			0	0	X	X
100			0	0	X	X

$$J_B = Ch' K Q_A'$$

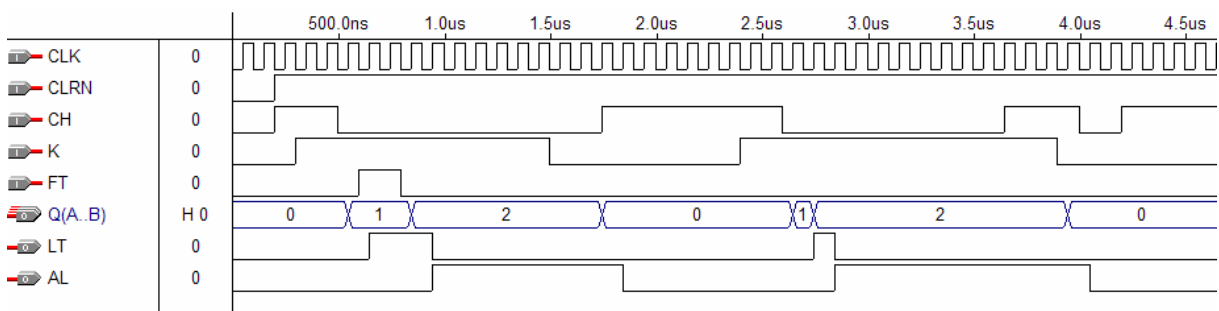
Ch	Q _A Q _B		FT			
	K		00	01	11	10
000			X	0	1	1
001			X	0	1	1
011			1	0	1	1
010			1	0	1	1
110			X	0	1	1
111			X	0	1	1
101			X	0	1	1
100			X	0	1	1

$$K_B = Q_A$$

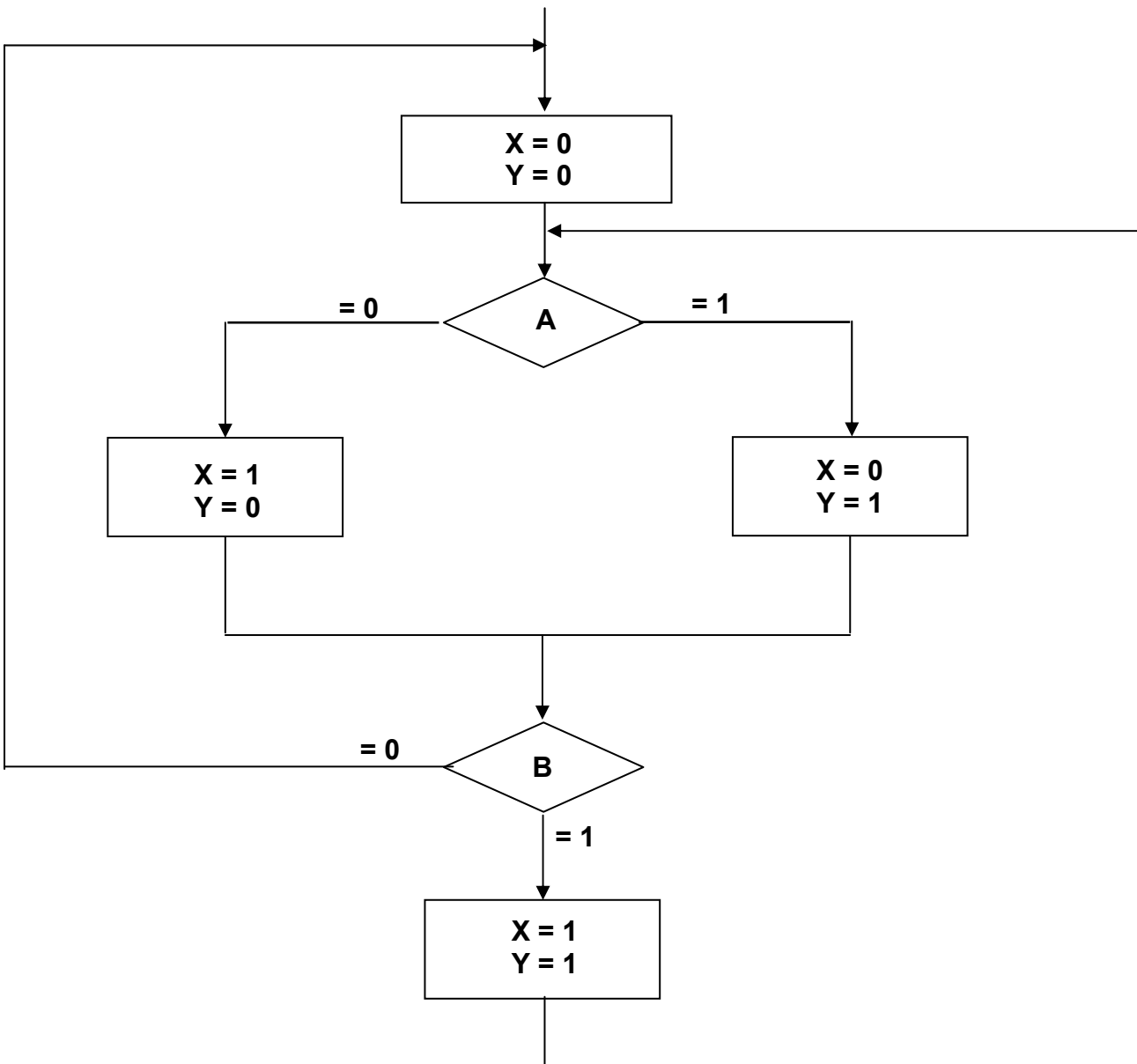
g) Implementação por tabela (ALP)

Fila	Estado Atual					Entradas Externas		Próximo Estados		Saídas dos F/Fs		C
	Q _A	Q _B	Ch	K	FT	Q _A	Q _B	AL	LT	C		
	A ₄	A ₃	A ₂	A ₁	A ₀	B ₃	B ₂	B ₁	B ₀			
00	0	0	0	0	0	0	0	0	0	0	0	
01	0	0	0	0	1	0	0	0	0	0	0	
02	0	0	0	1	0	0	1	0	0	4	0	
03	0	0	0	1	1	0	1	0	0	4	0	
04	0	0	1	0	0	0	0	0	0	0	0	
05	0	0	1	0	1	0	0	0	0	0	0	
06	0	0	1	1	0	0	0	0	0	0	0	
07	0	0	1	1	1	0	0	0	0	0	0	
08	0	1	0	0	0	1	0	0	1	9	0	
09	0	1	0	0	1	0	1	0	1	5	0	
0A	0	1	0	1	0	1	0	0	1	9	0	
0B	0	1	0	1	1	0	1	0	1	5	0	
0C	0	1	1	0	0	1	0	0	1	9	0	
0D	0	1	1	0	1	0	1	0	1	5	0	
0E	0	1	1	1	0	1	0	0	1	9	0	
0F	0	1	1	1	1	0	1	0	1	5	0	
10	1	0	0	0	0	0	0	1	0	2	0	
11	1	0	0	0	1	0	0	1	0	2	0	
12	1	0	0	1	0	1	0	1	0	A	0	
13	1	0	0	1	1	1	0	1	0	A	0	
14	1	0	1	0	0	0	0	1	0	2	0	
15	1	0	1	0	1	0	0	1	0	2	0	
16	1	0	1	1	0	1	0	1	0	A	0	
17	1	0	1	1	1	1	0	1	0	A	0	
18	1	1	0	0	0	0	0	0	0	0	0	
19	1	1	0	0	1	0	0	0	0	0	0	
1A	1	1	0	1	0	0	0	0	0	0	0	
1B	1	1	0	1	1	0	0	0	0	0	0	
1C	1	1	1	0	0	0	0	0	0	0	0	
1D	1	1	1	0	1	0	0	0	0	0	0	
1E	1	1	1	1	0	0	0	0	0	0	0	
1F	1	1	1	1	1	0	0	0	0	0	0	

As formas de ondas geradas no MAX-PLUS, mostram o projeto-alarme



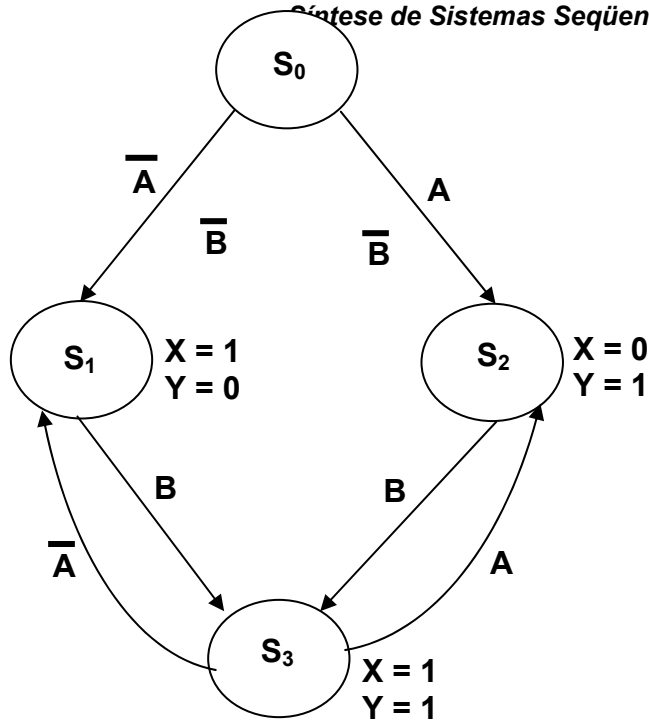
Exemplo : Para o fluxograma a seguir, traduzir em um diagrama de estados e implementar utilizando F/Fs do tipo D. Em seguida implementar como máquina de estados, modelo Moore.



Do fluxograma esboçado, pode-se construir o diagrama de estados. As entradas externas do processo são A e B e o sistema possui 4 estados S_0, S_1, S_2 e S_3 , codificados como $S_0 - 00, S_1 - 01, S_2 - 11$ e $S_3 - 10$.

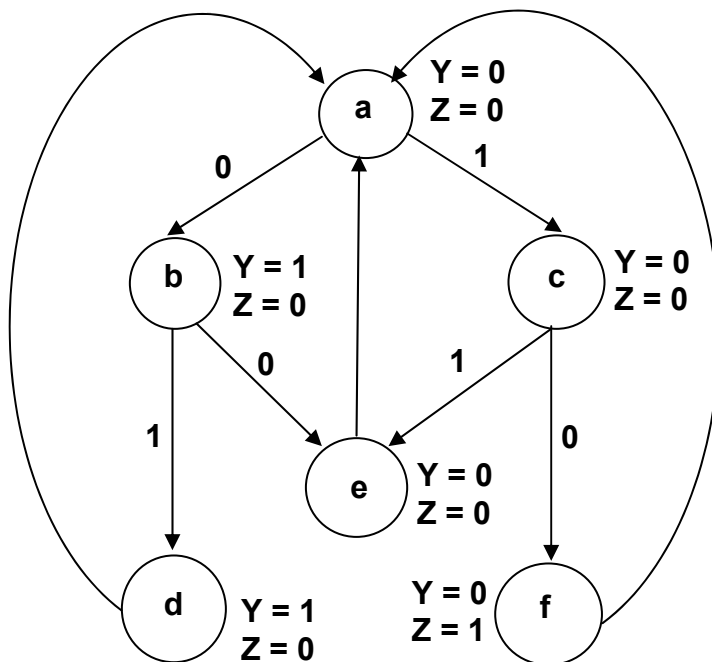
O diagrama de estados é apresentado a seguir.





Exemplo : Uma linha de dados serial X é permitida para alterar sobre o sinal de descida do clock. Projetar o IFL (circuito combinatório para uma máquina de 02 estados, utilizando F/F do tipo D, para detetar a seqüência 10 ou 01 sobre a linha de dados. Se 10 ocorrer, saída Y deve ser SET para o estado 1 antes da próxima verificação for iniciada. Se 01 ocorrer, saída W deve ser SET para o estado 1. Se 00 ou 11 aparecer como os 2 bits seqüenciais sobre X , nenhuma saída deve ser gerada e a próxima verificação deve iniciar com um atraso de um tempo de estado, pede-se:

- O diagrama de estado do processo.
- Implementação com F/F
- Implementação com máquina de estados



b) Designação de Estados

O preenchimento da tabela de designação de estados, deve seguir os 02 princípios da adjacências. Vemos que b e c obedecem o segundo princípio da adjacência e d, e e f

obedecem o primeiro princípio da adjacência e assim devem ser adjacentes. A escolha das células do mapa onde deverá localizar estes estados deverá observar também que o estado **b** deve ser próximo ao estado **d** e o estado **c** próximo ao **f**

	00	01	11	10
0	a	b	d	f
1	x	c	e	x

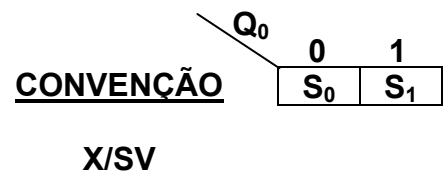
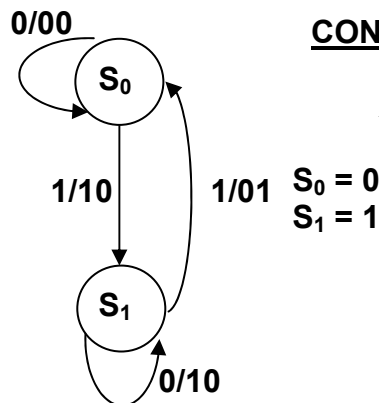
c) Implementação do circuito IFL

ATUAIS ENTRADA X SAÍDA

Exemplo : Construir um sistema seqüencial binário de uma única entrada X, que seja capaz de produzir uma saída S que é a soma binária entre X e Q acumulado. Produzir outra saída V que é o “vai um” da soma dos 2 bits binários. A tabela da verdade a seguir, mostra entradas e saídas binárias. Pede-se :

- Diagrama de estados do somador pelo modelo de Mealy.
- Implementação do somador por FSM.
- Repetir o item a) pelo modelo de Moore.
- Repetir o item b).

X	Q	S	V
0	0	0	0
0	1	1	0
1	1	1	0
1	1	0	1



Q ₀ X	Q ₀ S V
A ₁ A ₀	B ₂ B ₁ B ₀
0 0	000
0 1	110
1 0	110
1 1	001

Atual	Entrada X		Saída S,V	
Q ₀	0	1	0	1
0	0	1	00	10
1	1	0	10	01

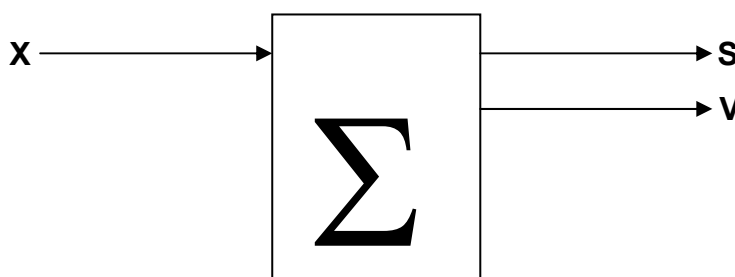
As equações :

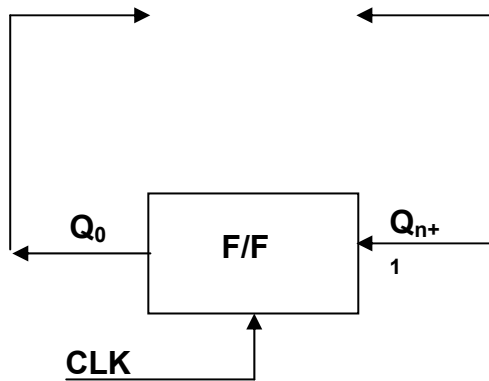
$$B_0 = Q_0 \oplus X$$

$$S = Q_0 \oplus X$$

$$V = Q_0 \cdot X$$

b) Representação por FSM, será :

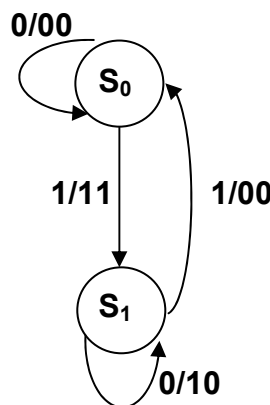




Exemplo : Construir um sistema seqüencial binário de uma única entrada X, que seja capaz de produzir uma saída S que é a subtração binária entre X e Q acumulado. Produzir outra saída M que é o “menos um” da subtração dos 2 bits binários. A tabela da verdade a seguir, mostra entradas e saídas binárias. Pede-se :

- Diagrama de estados do subtrator pelo modelo de Mealy.
- Implementação do somador por FSM.
- Repetir o item a) pelo modelo de Moore.
- Repetir o item b).

X	Q	S	M
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



CONVENÇÃO

Q ₀	0	1
	S ₀	S ₁

X/SV

S₀ = 0
S₁ = 1

Atual	Entrada X		Saída S,M	
Q ₀	0	1	0	1
0	0	1	00	11
1	1	0	10	00

Q ₀ X	Q ₀ S M
A ₁ A ₀	B ₂ B ₁ B ₀
0 0	000
0 1	111
1 0	110
1 1	000

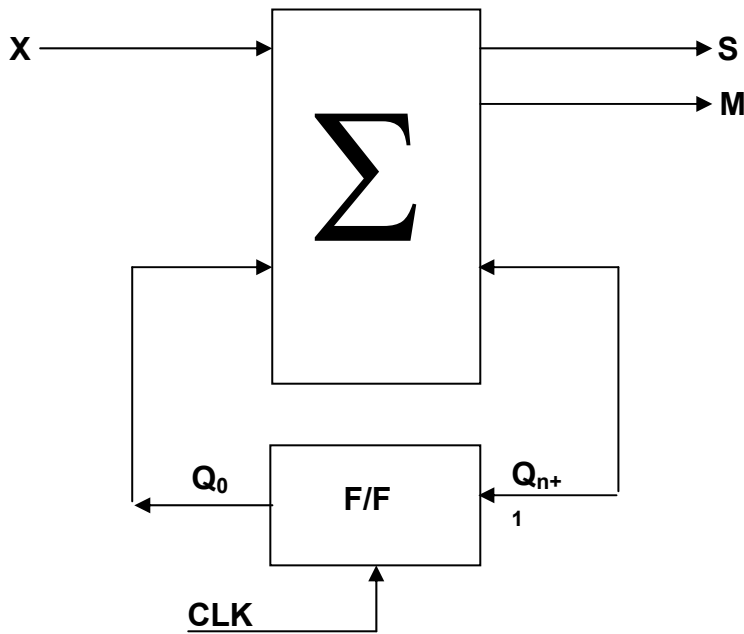
As equações :

$$B_0 = Q_0 \oplus X$$

$$S = Q_0 \oplus X$$

$$M = Q_0' \cdot X$$

b) Representação por FSM, será :

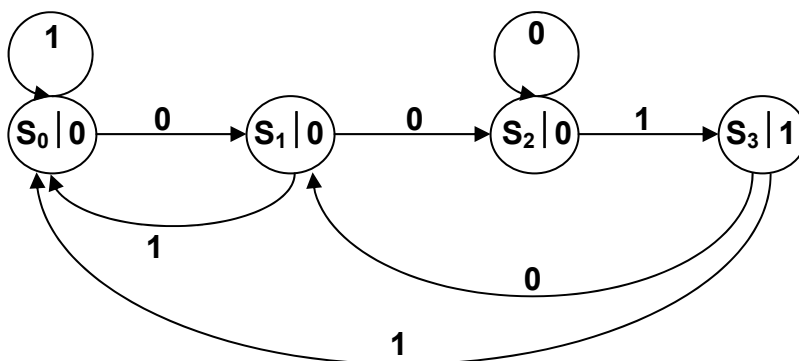


EXEMPLO : Construir um sistema seqüencial síncrono que produz uma saída ALTO quando a seqüência de bits de entrada for 0 – 0 – 1. Pede-se :

- a) A representação do sistema através de um diagrama de estados usando o modelo de Moore.
- b) A implementação do sistema, da lógica combinatória e F/Fs usando tipo D.
- c) O diagrama de tempos do sistema, simulando uma entrada de dados.
- d) A representação do sistema através de um diagrama de estados usando o modelo de Mealy.
- e) A implementação do sistema, da lógica combinatória e F/Fs usando tipo D.
- f) O diagrama de tempos do sistema, simulando uma entrada de dados.

Solução : O diagrama de estados a seguir é montado segundo o modelo de Moore, onde a saída é uma combinação de uma lógica com o estado atual.

a) Diagrama de Estados Modelo de Moore.



b) Implementação do sistema descrito acima pela tabela de estados, presentes e futuros e saída.

A primeira etapa a fazer é escolher os estados codificados, daí deveremos designar os estados S_0, S_1, S_2 e S_3 .

b.1) Designação de estados.

Designação de Estados S	Estados Atuais Q_1Q_0	Entrada Externa X		Entradas dos F/Fs D_1D_0		Saída Z
		0	1			
S0	00	01	00	01	00	0
S1	01	10	00	10	00	0
S2	10	10	11	10	11	0
S3	11	01	00	01	00	1

e.2) A implementação do sistema pelo Modelo de Moore, a partir da tabela de estados usando F/F do tipo D.

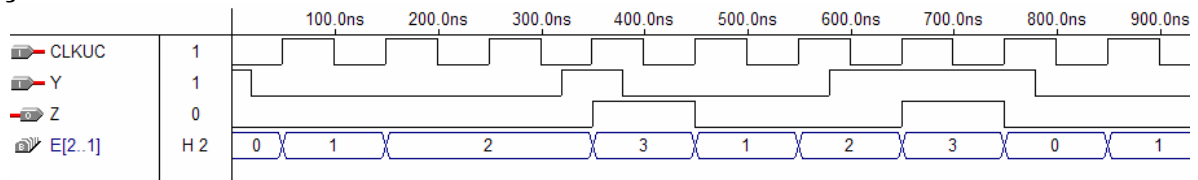
Q_1Q_0	00	01	11	10
X 0	0	1	0	1
1	0	0	0	1

Q_1Q_0	00	01	11	10
X 0	1	0	1	0
1	0	0	0	1

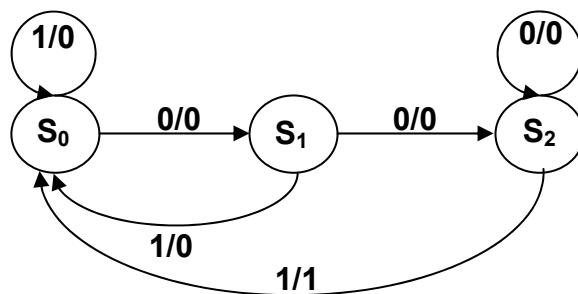
$$D_1 = Q_1'Q_0X' + Q_1Q_0X$$

$$Q_1Q_0' Z = Q_1Q_0 \quad D_0 = Q_1'Q_0'X' + Q_1Q_0X' + Q_1Q_0'X$$

c) O diagrama de tempo a seguir foi simulado no MAXPLUS II conforme as condições de entrada.



d) Diagrama de Estado Modelo de Mealy



e) Implementação do sistema descrito acima pela tabela de estados, presentes e futuros e saída.

A primeira etapa a fazer é escolher os estados codificados, daí deveremos designar os estados S_0, S_1, S_2 e S_3 .

e.1) Designação de estados.

Designação de Estados S	Estados Atuais Q ₁ Q ₀	Entrada Externa X		Entradas dos F/Fs D ₁ D ₀		Saída Z e X	
		0	1			0	1
S0	00	01	00	01	00	0	0
S1	01	10	00	10	00	0	0
S2	10	10	00	10	11	0	1
S3	11	00	00	00	00	0	0

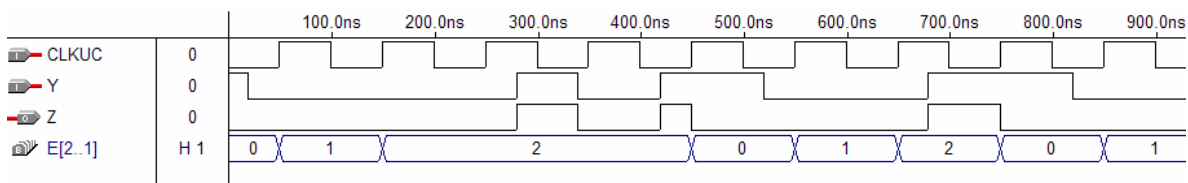
e.2) A implementação do sistema pelo Modelo de Mealy, a partir da tabela de estados usando F/F do tipo D.

	Q ₁ Q ₀				
		00	01	11	10
X	0	0	1	0	1
	1	0	0	0	0

	Q ₁ Q ₀				
		00	01	11	10
X	0	1	0	0	0
	1	0	0	0	0

$D_1 = Q_1'Q_0X' + Q_1Q_0'X$ $Z = Q_1Q_0'X$ $D_0 = Q_1'Q_0'X$

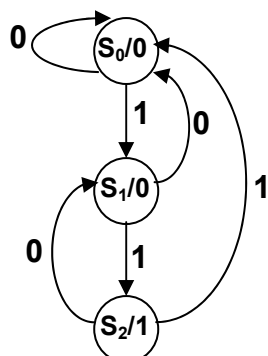
f) O diagrama de tempo do sistema pelo Modelo de Mealy é apresentado a seguir.



Exemplo : Construir um sistema seqüencial para o diagrama de estados representado pelo modelo de Moore, conforme a seguir. Implementar usando F/F do tipo D, onde X é a entrada do sistema. Pede-se :

- a) Tabela de estados
- b) Mapas de Karnaugh
- c) Esquemático por circuito
- d) Formas de ondas.

Dados : Diagrama de estados pelo Modelo de Moore.



a) Tabela de Estados

Q ₁ Q ₀	X		Y
	0	1	
00	00	01	0
01	00	10	0
10	01	00	1
11	00	00	0

Designação Estados

$S_0 = 00$
 $S_1 = 01$
 $S_2 = 10$

	Q ₁		
		0	1
Q ₀	0	S ₀	S ₂
	1	S ₁	X

b) Implementação com F/F – TIPO D

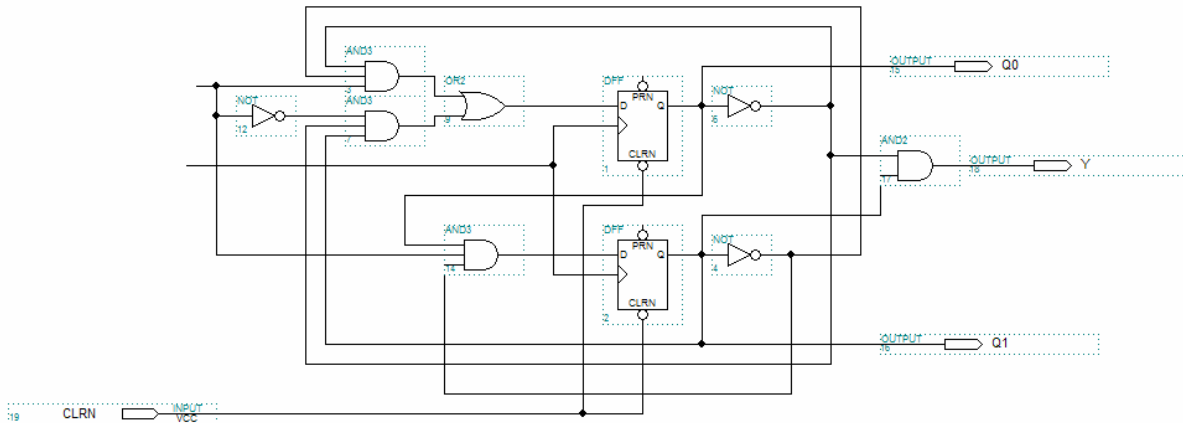
	Q ₁ Q ₀				
		00	01	11	10
X	0	0	0	0	0
	1	0	1	0	0

	Q ₁ Q ₀				
		00	01	11	10
X	0	0	0	0	1
	1	1	0	0	0

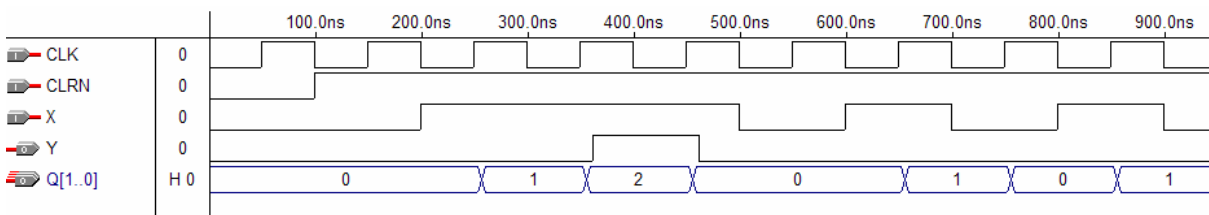
$D_1 = Q_1'Q_0X$ e

$D_0 = Q_1'Q_0'X + Q_1Q_0'X'$

O circuito é mostrado a seguir e foi editado no MAXPLUS.



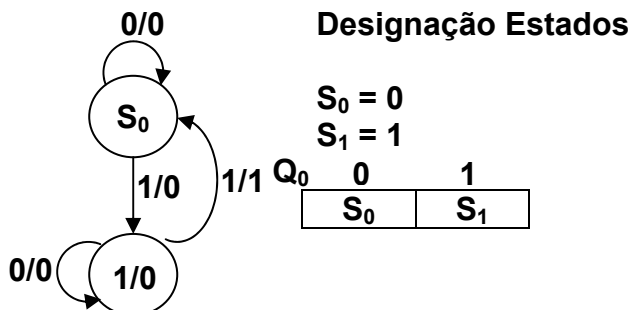
As formas de ondas geradas de acordo com a entrada X, mostra a evolução de estados, conforme o diagrama de estados.



Exemplo : Construir um sistema seqüencial para o diagrama de estados representado pelo modelo de Mealy, conforme a seguir. Implementar usando F/F do tipo D, onde X é a entrada do sistema. Pede-se :

- Tabela de estados
- Mapas de Karnaughs
- Esquemático por circuito
- Formas de ondas.

Dados : Diagrama de estados pelo Modelo de Mealy.



a) Tabela de Estados

	X		Y	
Q_0	0	1	0	1
0	0	1	0	0
1	1	0	0	1

b) Implementação com F/F – TIPO D

Q_0	0	1
X	0	1
1	1	0

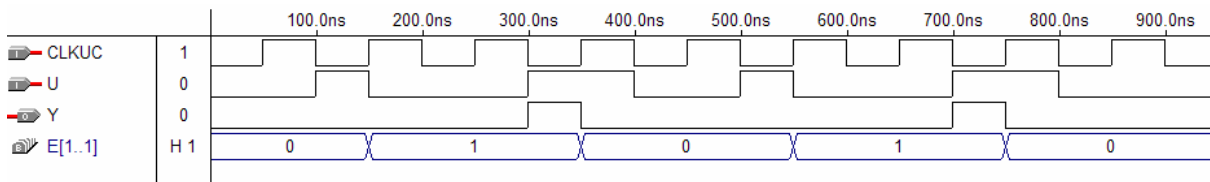
$$D_0 = Q_0 \oplus X$$

O programa foi editado no MAXPLUS no editor de texto e apresentado a seguir.

```

SUBDESIGN UNIP_EXP_1B
(
CLKUC : INPUT;
U : INPUT;
Y : OUTPUT;
)
Variable E[1] : DFF;
BEGIN
E[].CLK=CLKUC;
E[1] = (E1 & !U # !E1 & U);
Y = E1 & U;
END;
    
```

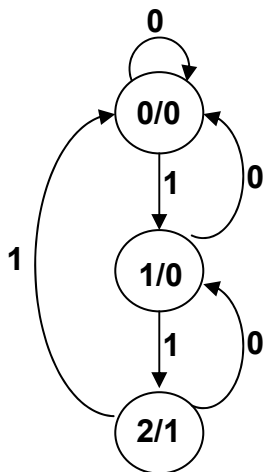
As formas de ondas geradas no diagrama de estado pelo Modelo Mealy, são apresentadas a seguir.



Exemplo : Construir um circuito seqüencial pelo Modelo de Moore capaz de gerar uma saída ALTO sempre que o sistema atingir o estado 2. O sistema possui uma única entrada e num total de 3 estados. Pede-se :

- a) O diagrama de estados
- b) A simulação da máquina de estados.

Solução : O diagrama de estados do sistema é apresentado a seguir.



A representação em blocos do sistema por máquina de estados é apresentado a seguir e consideramos uma entrada, 03 estados representados por 02 bits Q_1 e Q_0 e uma saída.

A tabela de estados é apresentada a seguir.

Q_1	Q_0	Ent	Q_1	Q_0	S
A_2	A_1	A_0	B_2	B_1	B_0
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	1
1	0	0	0	0	1
1	0	1	0	1	1
1	1	0	0	0	0

```
SUBDESIGN mapa_rom_exemplo1
```

```
(  
A2,A1,A0 : INPUT;  
B2,B1,B0 : OUTPUT;  
)
```

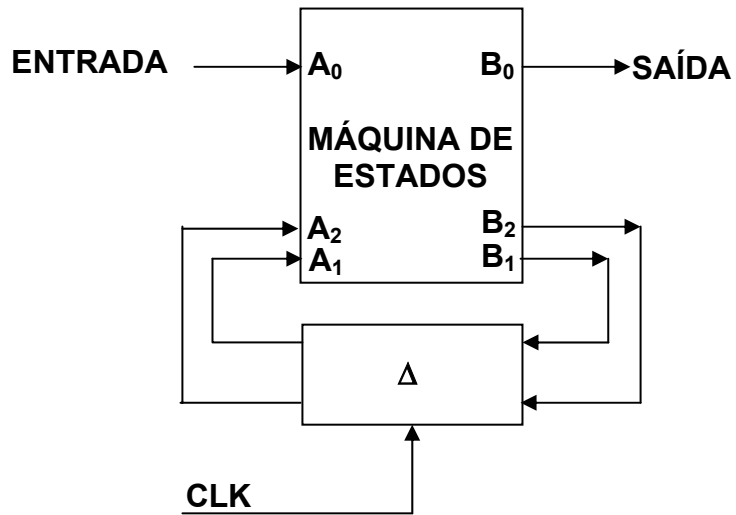
```
BEGIN
```

```
B2 = !A2 & A1 & A0;
```

```
B1 = !A2 & !A1 & A0 # A2 & !A1 & !A0;
```

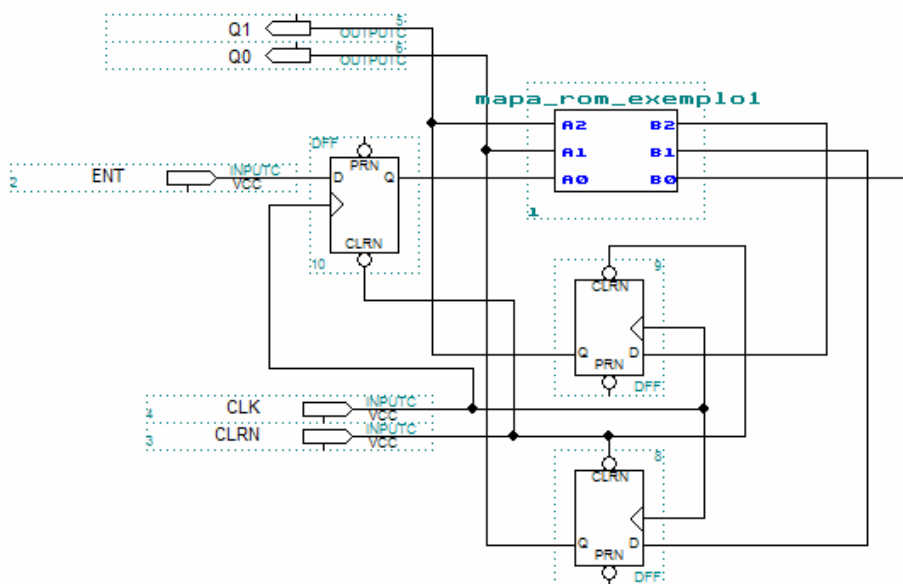
```
B0 = A2 & !A1 & !A0 # A2 & !A1 & A0;
```

```
END;
```

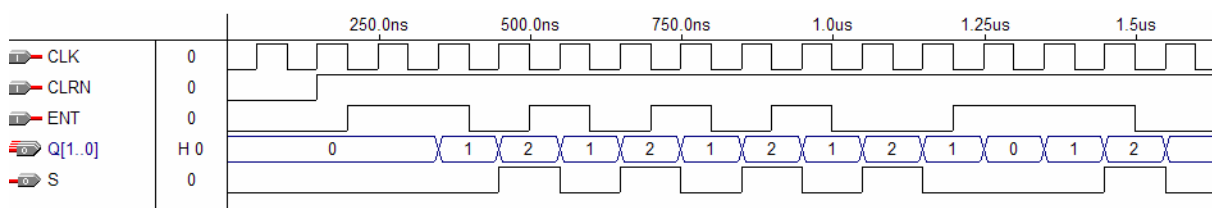


A implementação do circuito capaz de gerar o sistema é apresentado a seguir.

EXEMPLO - EVOLUÇÃO DO CONTEUDO - MÁQUINA DE ESTADOS



As formas de ondas geradas na simulação é apresentada a seguir.



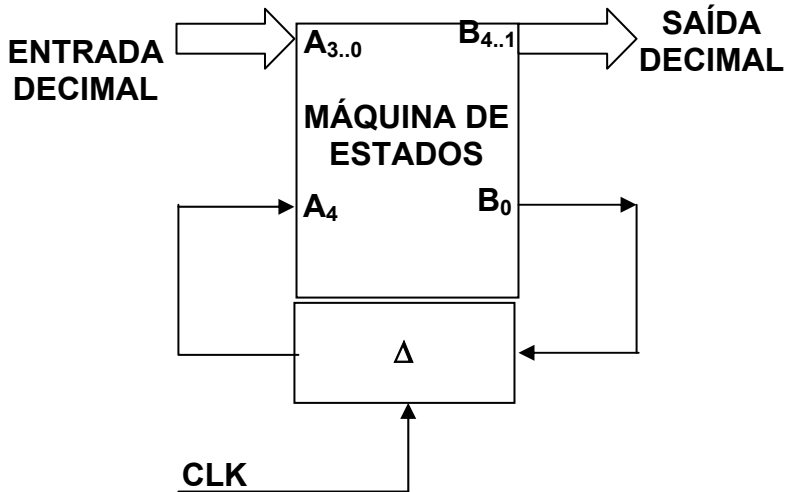
Exemplo : Projetar uma máquina de estados capaz de converter números decimais BCD – 8421 em números binários. Pede-se :

- A tabela de fluxo do conversor
- Implementação por máquina de estados

Solução : A tabela ser convertida deverá receber os números de entrada decimal e a cada conversão (divisão por 2), o resultado parcial deverá ser armazenado e o quociente realimentado para a entrada da tabela para ser convertido. O processo só finaliza quando o quociente se tornar zero. O resultado final será lido do último bit convertido MSB para o primeiro bit LSB.

Implementação da máquina de estados para a conversão de números na base 10 para a base 2.

- Entrada do número decimal de 0 a 9.
- Saída do número decimal (quociente da divisão) de 0 a 9.
- Resultado número binário 0 ou 1.
- Representação por máquina de estados finitos da conversão de base.



b) Representação do algoritmo de conversão por uma tabela de fluxo. A tabela possui entrada dos números de 0 a 9 portanto 10 colunas. A conversão é para a base 2 portanto 2 estados internos, assim a tabela é de 10 x 2.

0	1	2	3	4	5	6	7	8	9
0,0	1,0	0,1	1,1	0,2	1,2	0,3	1,3	0,4	1,4
0,5	1,5	0,6	1,6	0,7	1,7	0,8	1,8	0,9	1,9

c) Para mostrar o funcionamento da tabela acima, vamos simular o número $(151)_{10}$.

Entrada	Estado	Saída
1	1	0
5	1	7
1	1	5

Realimentando o quociente $(75)_{10}$ para a entrada

Entrada	Estado	Saída
7	1	3
5	1	7

Realimentando o quociente $(37)_{10}$ para a entrada

Entrada	Estado	Saída
3	1	1
7	1	8

Realimentando o quociente $(18)_{10}$ para a entrada

Entrada	Estado	Saída
1	1	0
8	0	9

Realimentando o quociente $(9)_{10}$ para a entrada

Entrada	Estado	Saída
9	1	4

Realimentando o quociente $(4)_{10}$ para a entrada

Entrada	Estado	Saída
4	0	2

Realimentando o quociente $(2)_{10}$ para a entrada

Entrada	Estado	Saída
2	0	1

Realimentando o quociente $(1)_{10}$ para a entrada

Entrada	Estado	Saída
1	0	0

1 1 0

O resultado é lido no estado assim o número convertido é $(10010111)_2$

A implementação por máquina de estados usando uma PLA, fica :

A ₄	A ₃	A ₂	A ₁	A ₀	B ₄	B ₃	B ₂	B ₁	B ₀
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	0	0	1	0
0	0	0	1	1	0	0	0	1	1
0	0	1	0	0	0	0	1	0	0
0	0	1	0	1	0	0	1	0	1
0	0	1	1	0	0	0	1	1	0
0	0	1	1	1	0	0	1	1	1
0	1	0	0	0	0	1	0	0	0
0	1	0	0	1	0	1	0	0	1
0	1	0	1	0	X	X	X	X	X
0	1	0	1	1	X	X	X	X	X
0	1	1	0	0	X	X	X	X	X
0	1	1	0	1	X	X	X	X	X
0	1	1	1	0	X	X	X	X	X
0	1	1	1	1	X	X	X	X	X
1	0	0	0	0	0	1	0	1	0
1	0	0	0	1	0	1	0	1	1
1	0	0	1	0	0	1	1	0	0
1	0	0	1	1	0	1	1	0	1
1	0	1	0	0	0	1	1	1	0
1	0	1	0	1	0	1	1	1	1
1	0	1	1	0	1	0	0	0	0
1	0	1	1	1	1	0	0	0	1
1	1	0	0	0	1	0	0	1	0
1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	X	X	X	X	X
1	1	0	1	1	X	X	X	X	X
1	1	1	0	0	X	X	X	X	X
1	1	1	0	1	X	X	X	X	X
1	1	1	1	0	X	X	X	X	X
1	1	1	1	1	X	X	X	X	X

As equações são :

$$B_4 = A_3A_4 + A_4A_3'A_2A_1;$$

$$B_3 = A_4'A_3 + A_4A_3'A_2' + A_4A_3'A_2A_1'$$

$$B_2 = A_4'A_2 + A_4A_3'A_2'A_1 + A_4A_3'A_2A_1'$$

$$B_1 = A_4'A_1 + A_4A_1'$$

$$B_0 = A_0$$

Exemplo : Projetar um conversor números na base 2 para a base 10, por máquina de estados. Pede-se :

- a) Tabela de fluxo
- b) Mapa da PAL
- c) Equações da PAL

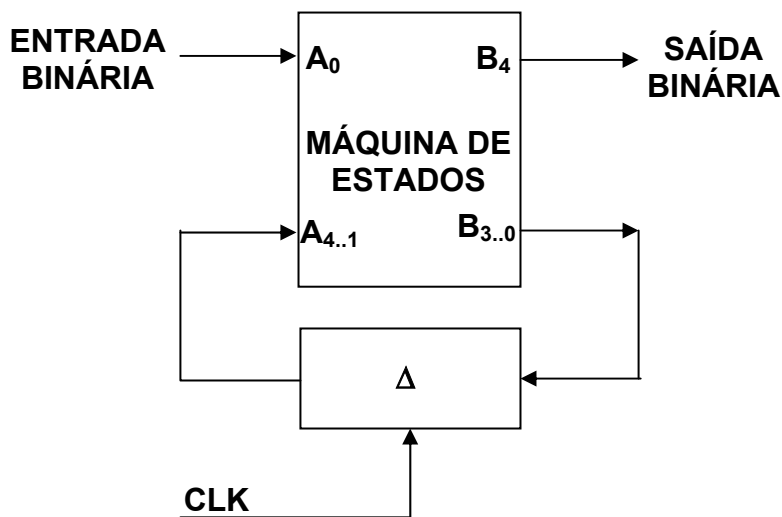
Conversão de números inteiros na base 2 para a base 10.

	0	1
	0,0	1,0
	2,0	3,0
	4,0	5,0
	6,0	7,0
	8,0	9,0
	0,1	1,1
	2,1	3,1
	4,1	5,1
	6,1	7,1
	8,1	9,1

O algoritmo de conversão para a base 10, segue :

$$N = a_1 / 2^1 + a_2 / 2^2 + \dots + a_{n-1} / 2^{n-1} + a_n / 2^n$$

Representação por máquina de estados do conversor binário para decimal.



O mapa da PAL, do conversor é mostrada a seguir.

A ₄	A ₃	A ₂	A ₁	A ₀	B ₄	B ₃	B ₂	B ₁	B ₀
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	0	0	1	0
0	0	0	1	1	0	0	0	1	1
0	0	1	0	0	0	0	1	0	0
0	0	1	0	1	0	0	1	0	1
0	0	1	1	0	0	0	1	1	0
0	0	1	1	1	0	0	1	1	1
0	1	0	0	0	0	1	0	0	0
0	1	0	0	1	0	1	0	0	1
0	1	0	1	0	1	0	0	0	0
0	1	0	1	1	1	0	0	0	1
0	1	1	0	0	1	0	0	1	0
0	1	1	0	1	1	0	0	1	1
0	1	1	1	0	1	0	1	0	0
0	1	1	1	1	1	0	1	0	1
1	0	0	0	0	1	0	1	1	0
1	0	0	0	1	1	0	1	1	1
1	0	0	1	0	1	1	0	0	0
1	0	0	1	1	1	1	0	0	1
1	0	1	0	0	X	X	X	X	X
1	0	1	0	1	X	X	X	X	X
1	0	1	1	0	X	X	X	X	X
1	0	1	1	1	X	X	X	X	X
1	1	0	0	0	X	X	X	X	X
1	1	0	0	1	X	X	X	X	X
1	1	0	1	0	X	X	X	X	X
1	1	0	1	1	X	X	X	X	X
1	1	1	0	0	X	X	X	X	X
1	1	1	0	1	X	X	X	X	X
1	1	1	1	0	X	X	X	X	X
1	1	1	1	1	X	X	X	X	X

As equações de saída da PAL, são :

$$B_4 = A_4 + A_3A_2 + A_3A_2'A_1$$

$$B_3 = A_3A_2'A_1' + A_4A_3'A_2'A_1$$

$$B_2 = A_3'A_2 + A_3A_2A_1 + A_4A_3'A_2'A_1'$$

$$B_1 = A_4'A_3'A_2'A_1 + A_3'A_2A_1 + A_3A_2A_1' + A_4A_3'A_2'A_1'$$

$$B_0 = A_0$$

Exemplo : Projetar um contador síncrono modo crescente/decrescente módulo 4 e 8, sendo selecionados através de 02 chaves K e V. Pede-se :

a) Tabela de fluxo do contador;

b) Implementação por máquina de estados com PAL.

A tabela a seguir mostra a configuração do contador.

K	V	TIPO
0	0	CM4
0	1	CM8
1	0	DM4
1	1	DM8

Obs.:

CM4 = Crescente Módulo 4.

CM8 = Crescente Módulo 8.

DM4 = Decrescente Módulo 4.

Dm8 = Decrescente Módulo 8.

KV

Est.	00	01	10	11
0	1,0	1,0	3,0	3,0
1	2,0	2,0	0,1	0,1
2	3,0	3,0	1,0	1,0
3	0,1	4,0	2,0	2,0
4	X	5,0	X	3,0
5	X	6,0	X	4,0
6	X	7,0	X	5,0
7	X	0,1	X	6,0

$$\begin{aligned}
 A_4 &= K & B_3 &= Q_2 \\
 A_3 &= V & B_2 &= Q_1 \\
 A_2 &= Q_2 & B_1 &= Q_0 \\
 A_1 &= Q_1 & B_0 &= S \\
 A_0 &= Q_0 & &
 \end{aligned}$$

O mapa da PAL, para o contador acima, fica :

A ₄	A ₃	A ₂	A ₁	A ₀	B ₃	B ₂	B ₁	B ₀
0	0	0	0	0	0	0	1	0
0	0	0	0	1	0	1	0	0
0	0	0	1	0	0	1	1	0
0	0	0	1	1	0	0	0	1
0	0	1	0	0	X	X	X	0
0	0	1	0	1	X	X	X	0
0	0	1	1	0	X	X	X	0
0	0	1	1	1	X	X	X	0
0	1	0	0	0	0	0	1	0
0	1	0	0	1	0	1	0	0
0	1	0	1	0	0	1	1	0
0	1	0	1	1	1	0	0	0
0	1	1	0	0	1	0	1	0
0	1	1	0	1	1	1	0	0
0	1	1	1	0	1	1	1	0
0	1	1	1	1	0	0	0	1
1	0	0	0	0	0	1	1	0
1	0	0	0	1	0	0	0	1
1	0	0	1	0	0	0	1	0
1	0	0	1	1	0	1	0	0
1	0	1	0	0	X	X	X	0
1	0	1	0	1	X	X	X	0
1	0	1	1	0	X	X	X	0
1	0	1	1	1	X	X	X	0
1	1	0	0	0	1	1	1	0
1	1	0	0	1	0	0	0	1
1	1	0	1	0	0	0	1	0
1	1	0	1	1	0	1	0	0
1	1	1	0	0	0	1	1	0
1	1	1	0	1	1	0	0	0
1	1	1	1	0	1	0	1	0
1	1	1	1	1	1	1	0	0

As equações da PAL, são mostradas a seguir.

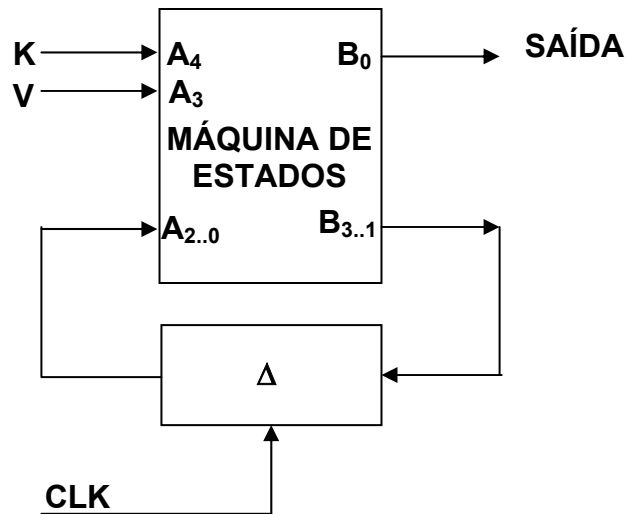
$$B_3 = A_4'A_3A_2'A_1A_0 + A_4'A_3A_2A_1'A_0' + A_4'A_3A_2A_1'A_0 + A_4'A_3A_2A_1A_0' + A_4'A_3A_2'A_1A_0 + A_4A_3A_2'A_1'A_0'$$

$$B_2 = A_4'A_3'A_2'A_1'A_0 + A_4'A_3'A_2'A_1A_0' + A_4'A_3A_2'A_1'A_0 + A_4'A_3A_2'A_1A_0' + A_4'A_3A_2A_1'A_0 + A_4'A_3A_2A_1A_0' + A_4A_3'A_2'A_1'A_0' + A_4A_3'A_2'A_1A_0 + A_4A_3A_2'A_1'A_0' + A_4A_3A_2'A_1A_0 + A_4A_3A_2A_1'A_0' + A_4A_3A_2A_1A_0$$

$$B_1 = A_0'$$

$$B_0 = A_4'A_3'A_2'A_1A_0 + A_4'A_3A_2A_1A_0 + A_4A_3'A_2'A_1'A_0 + A_4A_3A_2'A_1'A_0$$

Representação por máquina de estados do contador módulo 4/8 cresc./decresc.



Exemplo : Projetar um sistema digital de controle de um disco. O disco se encontra em repouso no ponto X. Quando o operador aciona o botão B, o disco se movimentava através do motor M, gira 1 volta completa e depois pára no ponto X. Pede-se :

- a) Tabela de fluxo do sistema digital
- b) Implementação por máquina de estados usando PAL.
- c) Diagrama de estados do sistema digital.

Solução :

a) Tabela de fluxo

B,X	00	01	10	11
0	X	0,0	X	1,1
1	X	X	2,1	1,1
2	3,1	X	2,1	X
3	3,1	0,0	X	X

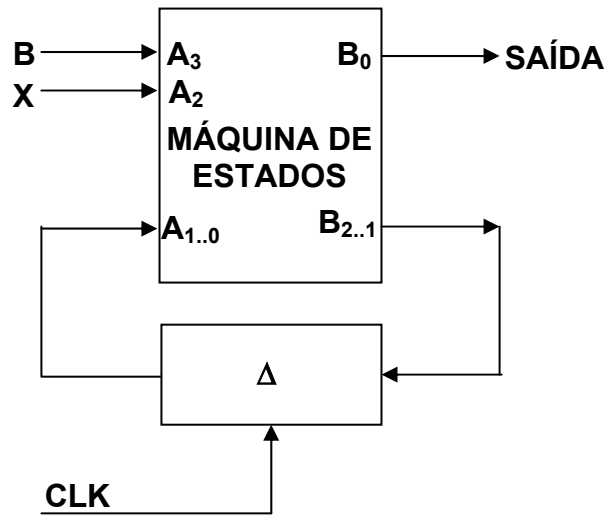
Obs.:

- X = Ponto X
- X = 0 – Fora do ponto X;
- X = 1 – No ponto X;
- B = Botão de Acionamento;
- B = 0 – Botão Não Acionado;
- B = 1 – Botão Acionado.

b) Implementação por máquina de estados.

A implementação de uma tabela combinatória com F/Fs do tipo D como memória, sincronizado por um sinal de relógio.

Representação por máquina de estados do controle do disco.



b) A PAL é mostrada a seguir.

A ₃	A ₂	A ₁	A ₀	B ₂	B ₁	B ₀
0	0	0	0	X	X	0
0	0	0	1	X	X	0
0	0	1	0	1	1	1
0	0	1	1	1	1	1
0	1	0	0	0	0	0
0	1	0	1	X	X	0
0	1	1	0	X	X	0
0	1	1	1	0	0	0
1	0	0	0	X	X	0
1	0	0	1	1	0	1
1	0	1	0	1	0	1
1	0	1	1	X	X	0
1	1	0	0	0	1	1
1	1	0	1	0	1	1
1	1	1	0	X	X	0
1	1	1	1	X	X	0

As equações da PAL, são :

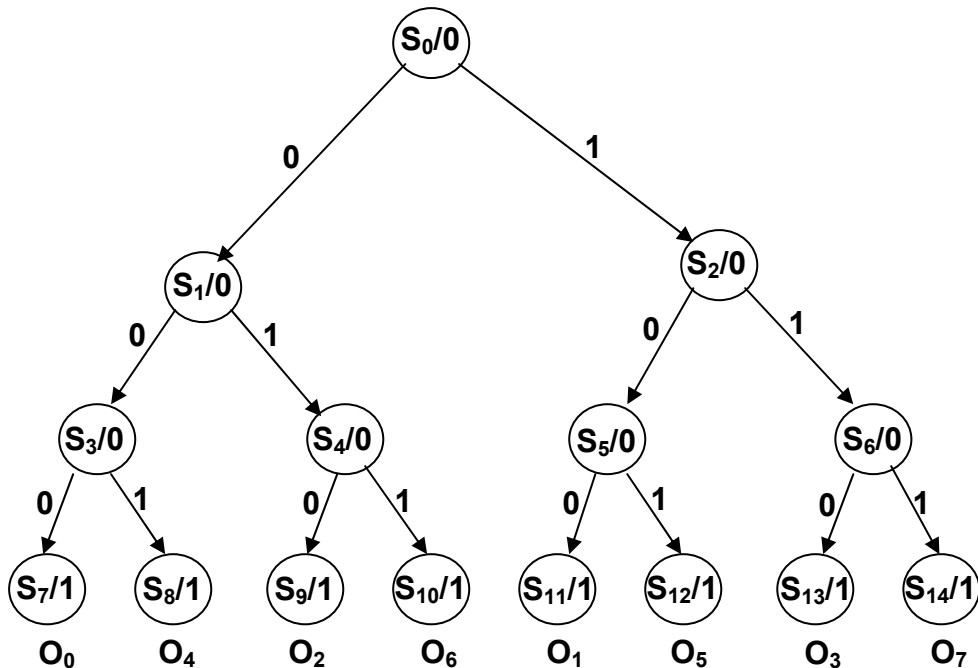
$$B_2 = A_3'A_2'A_1 + A_3A_2'$$

$$B_1 = A_3'A_2' + A_3A_2$$

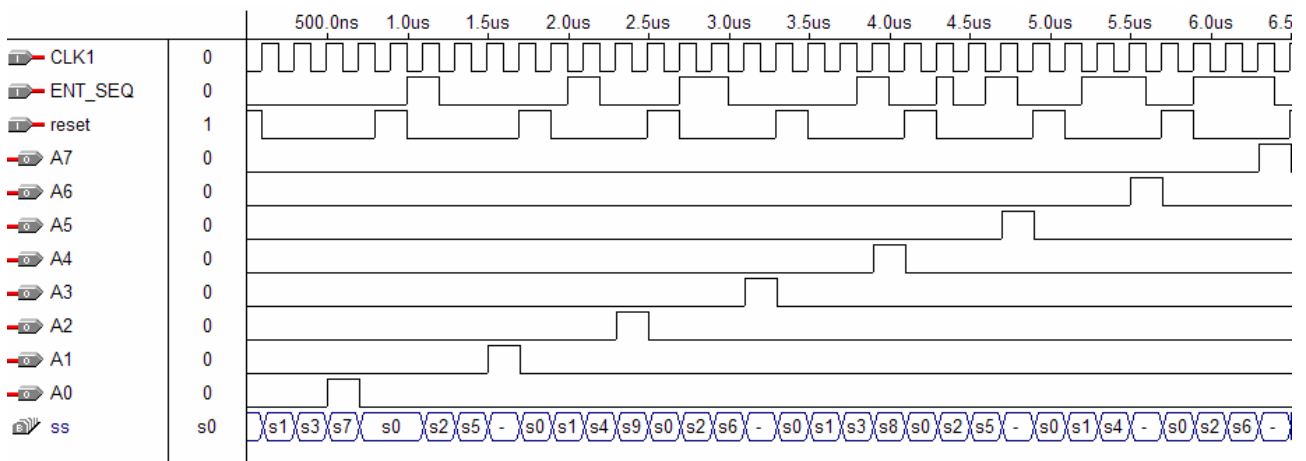
$$B_0 = A_3'A_2'A_1 + A_3A_2'A_1'A_0 + A_3A_2'A_1A_0' + A_3A_2A_1'$$

Projetar um decodificador seqüencial de 3 x 8 saídas. A saída opera na lógica positiva de A_{7...0}. Pede-se :

a) Diagrama de estados pelo Modelo de Moore.



As formas de ondas geradas no MAX-PLUS para o decodificador, sendo uma entrada seqüencial, uma entrada reset e um clock.



O projeto a seguir mostra em AHDL no MAX-PLUS a estrutura de programa para o decodificador seqüencial para 3 bits.

SUBDESIGN decodificador_seqüencial

```
(
    ENT_SEQ      : INPUT;
    CLK1,reset: INPUT;
    A0,A1,A2,A3,A4,A5,A6,A7  :output;
)
VARIABLE
    ss      : MACHINE WITH STATES (s0,s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13,s14);
    A0,A1,A2,A3,A4,A5,A6,A7  : NODE;

BEGIN
```

ss.CLK = CLK1;
ss.reset = reset;

TABLE

%ATUAL %ESTADO ss,	reset reset,	ENT. ENT_SEQ	=>	PROXIMO ESTADO % ss,	SAIDA % A0,A1,A2,A3,A4,A5,A6,A7;
s0,	0,	0	=>	s1,	0,0,0,0,0,0,0;
s0,	0,	1	=>	s2,	0,0,0,0,0,0,0;
s1,	0,	0	=>	s3,	0,0,0,0,0,0,0;
s1,	0,	1	=>	s4,	0,0,0,0,0,0,0;
s2,	0,	0	=>	s5,	0,0,0,0,0,0,0;
s2,	0,	1	=>	s6,	0,0,0,0,0,0,0;
s3,	0,	0	=>	s7,	0,0,0,0,0,0,0;
s3,	0,	1	=>	s8,	0,0,0,0,0,0,0;
s4,	0,	0	=>	s9,	0,0,0,0,0,0,0;
s4,	0,	1	=>	s10,	0,0,0,0,0,0,0;
s5,	0,	0	=>	s11,	0,0,0,0,0,0,0;
s5,	0,	1	=>	s12,	0,0,0,0,0,0,0;
s6,	0,	0	=>	s13,	0,0,0,0,0,0,0;
s6,	0,	1	=>	s14,	0,0,0,0,0,0,0;
s7,	0,	0	=>	s0,	1,0,0,0,0,0,0;
s7,	0,	1	=>	s0,	1,0,0,0,0,0,0;
s8,	0,	0	=>	s0,	0,0,0,0,1,0,0;
s8,	0,	1	=>	s0,	0,0,0,0,1,0,0;
s9,	0,	0	=>	s0,	0,0,1,0,0,0,0;
s9,	0,	1	=>	s0,	0,0,1,0,0,0,0;
s10,	0,	0	=>	s0,	0,0,0,0,0,0,1;
s10,	0,	1	=>	s0,	0,0,0,0,0,0,1;
s11,	0,	0	=>	s0,	0,1,0,0,0,0,0;
s11,	0,	1	=>	s0,	0,1,0,0,0,0,0;
s12,	0,	0	=>	s0,	0,0,0,0,0,1,0;
s12,	0,	1	=>	s0,	0,0,0,0,0,1,0;
s13,	0,	0	=>	s0,	0,0,0,1,0,0,0;
s13,	0,	1	=>	s0,	0,0,0,1,0,0,0;
s14,	0,	0	=>	s0,	0,0,0,0,0,0,1;
s14,	0,	1	=>	s0,	0,0,0,0,0,0,1;

END TABLE;

TABLE

%ATUAL %ESTADO ss,	reset reset,	ENT. ENT_SEQ	=>	PROXIMO ESTADO % ss,	SAIDA % A0,A1,A2,A3,A4,A5,A6,A7;
s0,	1,	0	=>	s0,	0,0,0,0,0,0,0;
s0,	1,	1	=>	s0,	0,0,0,0,0,0,0;
s1,	1,	0	=>	s0,	0,0,0,0,0,0,0;
s1,	1,	1	=>	s0,	0,0,0,0,0,0,0;
s2,	1,	0	=>	s0,	0,0,0,0,0,0,0;
s2,	1,	1	=>	s0,	0,0,0,0,0,0,0;
s3,	1,	0	=>	s0,	0,0,0,0,0,0,0;
s3,	1,	1	=>	s0,	0,0,0,0,0,0,0;
s4,	1,	0	=>	s0,	0,0,0,0,0,0,0;
s4,	1,	1	=>	s0,	0,0,0,0,0,0,0;
s5,	1,	0	=>	s0,	0,0,0,0,0,0,0;
s5,	1,	1	=>	s0,	0,0,0,0,0,0,0;
s6,	1,	0	=>	s0,	0,0,0,0,0,0,0;

Síntese de Sistemas Seqüenciais por Diagrama de Estados

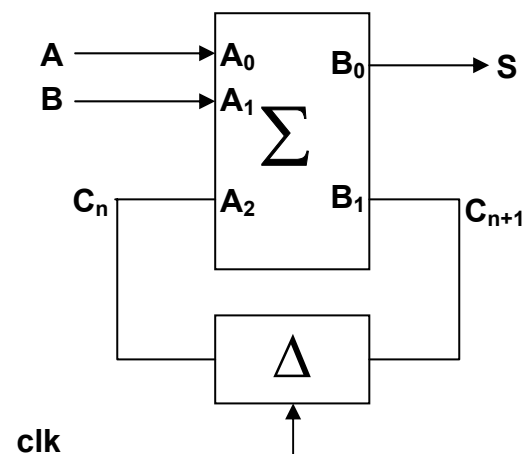
s6,	1,	1	=>	s0,	0,0,0,0,0,0,0,0;
s7,	1,	0	=>	s0,	0,0,0,0,0,0,0,0;
s7,	1,	1	=>	s0,	0,0,0,0,0,0,0,0;
s8,	1,	0	=>	s0,	0,0,0,0,0,0,0,0;
s8,	1,	1	=>	s0,	0,0,0,0,0,0,0,0;
s9,	1,	0	=>	s0,	0,0,0,0,0,0,0,0;
s9,	1,	1	=>	s0,	0,0,0,0,0,0,0,0;
s10,	1,	0	=>	s0,	0,0,0,0,0,0,0,0;
s10,	1,	1	=>	s0,	0,0,0,0,0,0,0,0;
s11,	1,	0	=>	s0,	0,0,0,0,0,0,0,0;
s11,	1,	1	=>	s0,	0,0,0,0,0,0,0,0;
s12,	1,	0	=>	s0,	0,0,0,0,0,0,0,0;
s12,	1,	1	=>	s0,	0,0,0,0,0,0,0,0;
s13,	1,	0	=>	s0,	0,0,0,0,0,0,0,0;
s13,	1,	1	=>	s0,	0,0,0,0,0,0,0,0;
s14,	1,	0	=>	s0,	0,0,0,0,0,0,0,0;
s14,	1,	1	=>	s0,	0,0,0,0,0,0,0,0;

END TABLE;

END;

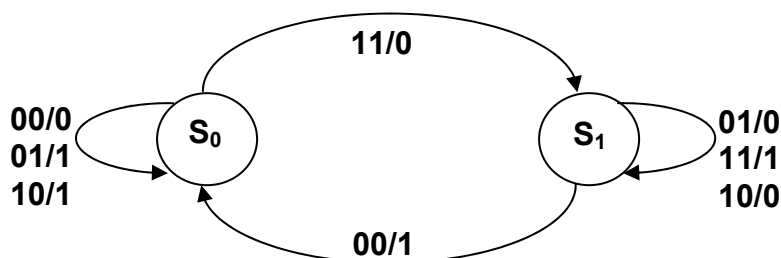
Exemplo : Construir um somador seqüencial de 02 bits A e B. Pede-se :

a) Diagrama de estados pelo Modelo de Mealy.



a) Tabela de fluxo a seguir, mostra o somador.

AB/C _n	00	01	11	10
0	0,0	0,1	1,0	0,1
1	0,1	1,0	1,1	1,0



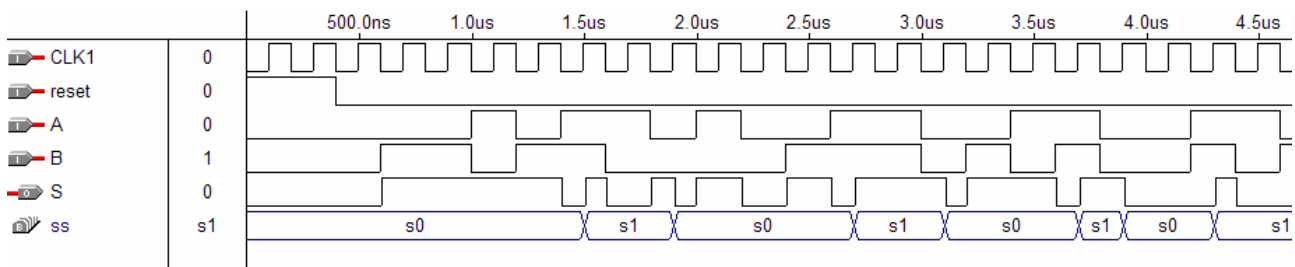
A tabela de estados do somador, será :

Atual	Entrada				Saída			
C _n	00	01	10	11	00	01	10	11
0	0	0	0	1	0	1	1	0
1	0	1	1	1	1	0	0	1

A tabela de estados implementada com Arranjo lógico Programável, será :

A ₂	A ₁	A ₀	B ₁	B ₀	End.	Cont.
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	0	1	2	1
0	1	1	1	0	3	2
1	0	0	0	1	4	1
1	0	1	0	1	5	1
1	1	0	1	0	6	2
1	1	1	1	1	7	3

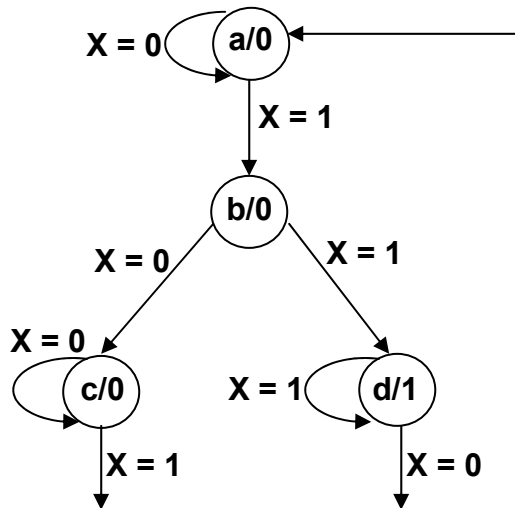
As formas de ondas do somador seqüencial, será :



ELIMINANDO GLITCHES NA SAÍDA DOS CIRCUITOS SEQÜENCIAIS POR DESIGNAÇÃO DE ESTADOS.

O exemplo a seguir mostra como evitar os glitches na saída por designação de estados.

Exemplo : Para o diagrama de estados a seguir, implementar o circuito de saída.



1) Designação de Estados

		0	1
Q ₁	0	a	b
	1	c	d*

De b → c

1) Q₀ = 1 (antes de Q₁ = 0)

Estados 10 – 11** – 01

2) Q₁ = 0 (antes de Q₀)

Estados 10 – 00 – 01

** glitch na saída por passar pelo estado d.

2) Nova designação de Estados

		0	1
Q ₁	0	a	d*
	1		

Pág. 55

3) Nova designação de Estados

		00	01	11	10
Q ₂	0	a	d*	x	x
	1	b	x	x	c

Nos itens 1 e 2) não se consegue evitar glitch, daí o item 3) com a introdução de um F/F consegue-se evitar o glitch na saída porém se sacrifica a minimização e criou-se com os 03 F/Fs mais estados e daí estados adjacentes o que evita glitches.

Princípio 3 : “ESTADOS COM A MESMA SAÍDA SÃO ADJACENTES E PORTANTO NÃO HÁ GLITCHES”.