

Ejercicio No.1

Caja Negra

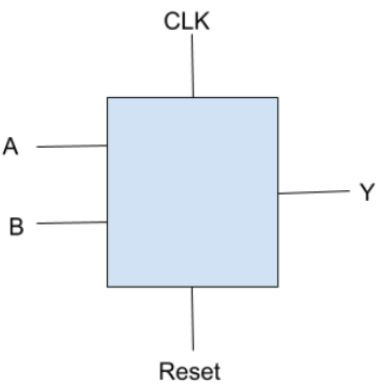


Tabla de transiciones sin codificar

Estado Actual	A	B	iguiente Estad	Y
S0	0	x	S0	0
S0	1	x	S1	0
S1	x	0	S0	0
S1	x	1	S2	0
S2	1	1	S2	1
S2	0	0	S0	0

Tabla de transiciones codificada

Estado Actual	A	B	iguiente Estad	Y
0 0	0	x	0 0	0
0 0	1	x	0 1	0
0 1	x	0	0 0	0
0 1	x	1	1 0	0
1 0	1	1	1 0	1
1 0	0	0	0 0	0

Logic Friday

File Operation Truthtable Equation Gates View Help

Funci... Inputs Outputs True False DC PI Gates

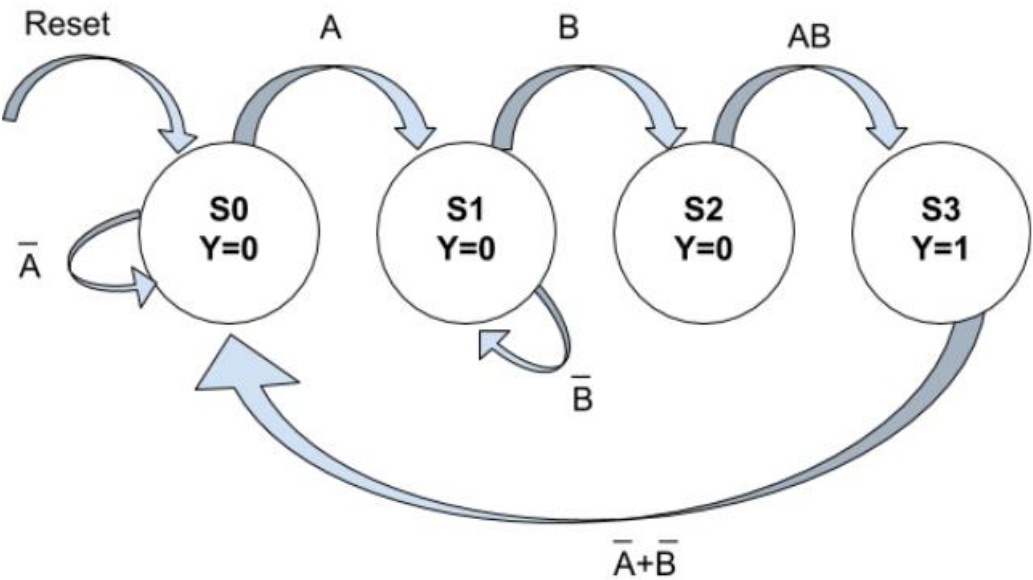
S'0-Y 4 3 3, 2, 1 9, 10,... 4, 4, 4 4 Not mapped

S0	S1	A	B	=>	S'0	S'1	Y
X	1	X	1		1		
1	X	1	1		1		
0	0	1	X			1	
1	X	1	1				1

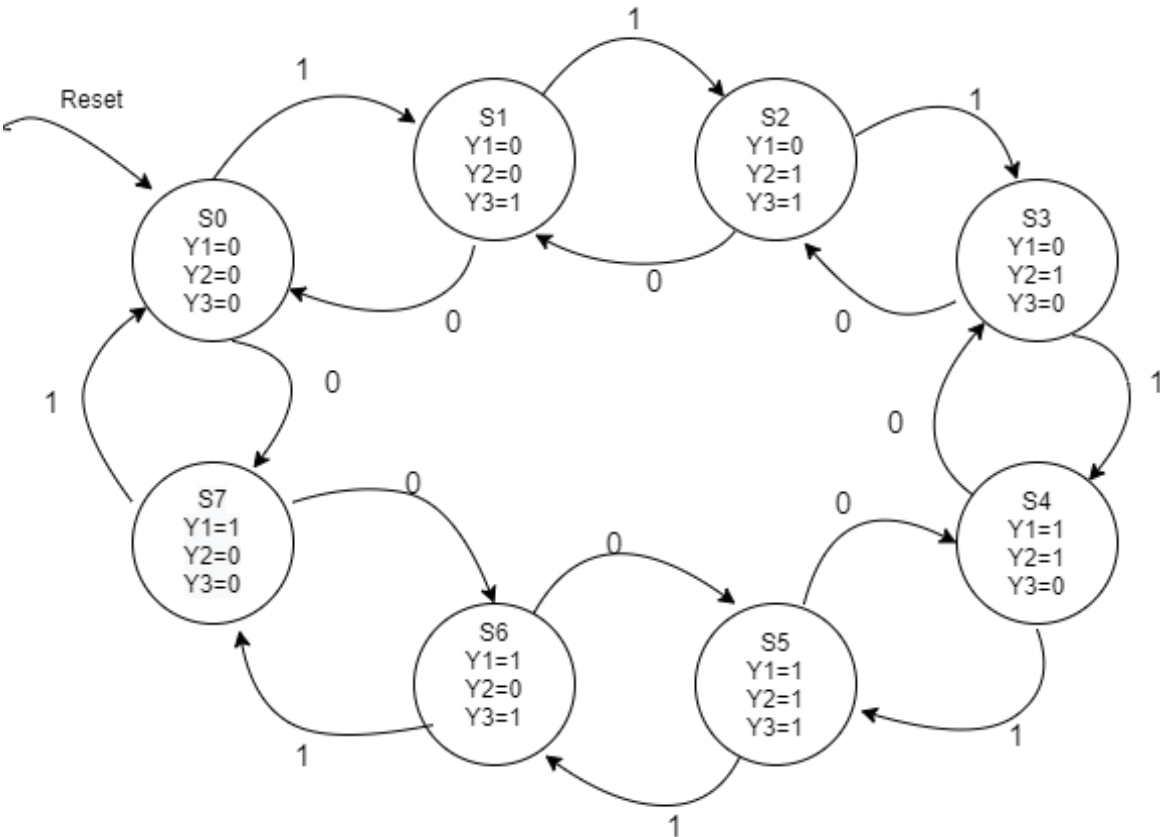
Entered by truthtable:
S'0 = S0' S1 A' B + S0' S1 A B + S0 S1' A B;
S'1 = S0' S1' A B' + S0' S1' A B;
Y = S0 S1' A B;

Minimized:
S'0 = S1 B + S0 A B;
S'1 = S0' S1' A ;
Y = S0 A B;

Ejercicio No.2



Ejercicio No.3



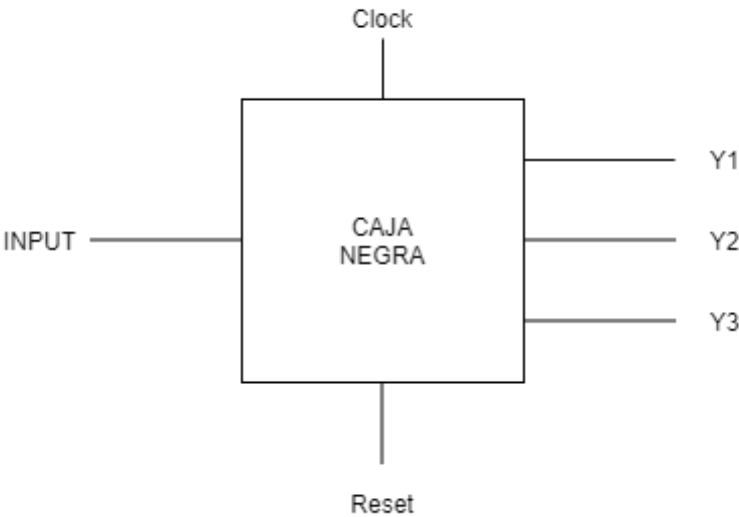


Tabla de transiciones sin codificar

Estado Actual	Input	Siguiente Estado	Y1	Y2	Y3
S0	0	S7	0	0	0
S0	1	S1	0	0	0
S1	0	S0	0	0	1
S1	1	S2	0	0	1
S2	0	S1	0	1	1
S2	1	S3	0	1	1
S3	0	S2	0	1	0
S3	1	S4	0	1	0
S4	0	S3	1	1	0
S4	1	S5	1	1	0
S5	0	S4	1	1	1
S5	1	S6	1	1	1
S6	0	S5	1	0	1
S6	1	S7	1	0	1
S7	0	S6	1	0	0
S7	1	S0	1	0	0

Tabla de transiciones codificada						
Estado Actual			Input	Siguiete Estado		
A	B	C		S'0	S'1	S'2
0	0	0	0	1	1	1
0	0	0	1	0	0	1
0	0	1	0	0	0	0
0	0	1	1	0	1	0
0	1	0	0	0	0	1
0	1	0	1	0	1	1
0	1	1	0	0	1	0
0	1	1	1	1	0	0
1	0	0	0	0	1	1
1	0	0	1	1	0	1
1	0	1	0	1	0	0
1	0	1	1	1	1	0
1	1	0	0	1	0	1
1	1	0	1	1	1	1
1	1	1	0	1	1	0
1	1	1	1	0	0	0

Tabla de transiciones codificada						
Estado Actual			Input	Salidas		
A	B	C		Y1	Y2	Y3
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	0	1	1
0	1	0	1	0	1	1
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	1	1	0
1	0	0	1	1	1	0
1	0	1	0	1	1	1
1	0	1	1	1	1	1
1	1	0	0	1	0	1
1	1	0	1	1	0	1
1	1	1	0	1	0	0
1	1	1	1	1	0	0

Logic Friday

File Operation Truthtable Equation Gates View Help

Function Inputs Outputs True False DC PI Gates

S'0-Y3 4 6 8, 8, ... 8, 8, ... 0, 0, ... 15 Not mapped

A	B	C	Input	=>	S'0	S'1
0	1	1	1		1	
0	0	0	0		1	
1	0	1	X		1	
1	X	0	1		1	
1	1	X	0		1	
X	0	1	1		1	
X	1	0	1		1	
X	1	1	0		1	
X	0	0	0		1	
X	X	0	X			
1	X	X	X			
0	1	X	X			
1	0	X	X			
X	0	1	X			
X	1	0	X			

Entered by truthtable:
 $S'0 = A' B' C' \text{ Input}' + A' B C \text{ Input} + A B' C' \text{ Input} + A B' C \text{ Input}' + A B C' \text{ Input}' + A B C \text{ Input}' + A B C' \text{ Input}' + A B C \text{ Input}'$
 $S'1 = A' B' C' \text{ Input}' + A' B' C \text{ Input} + A' B C' \text{ Input} + A' B C \text{ Input}' + A B' C' \text{ Input}' + A B' C \text{ Input} + A B C' \text{ Input}' + A B C \text{ Input}'$
 $S'2 = A' B' C' \text{ Input}' + A' B' C' \text{ Input} + A' B C' \text{ Input}' + A' B C' \text{ Input} + A B' C' \text{ Input}' + A B' C' \text{ Input} + A B C' \text{ Input}' + A B C' \text{ Input}$
 $C' \text{ Input}' + A B C' \text{ Input};$
 $Y1 = A B' C' \text{ Input}' + A B' C' \text{ Input} + A B' C \text{ Input}' + A B' C \text{ Input} + A B C' \text{ Input}' + A B C' \text{ Input} + A B C' \text{ Input}' + A B C \text{ Input}'$
 $+ A B C \text{ Input};$
 $Y2 = A' B C' \text{ Input}' + A' B C' \text{ Input} + A' B C \text{ Input}' + A' B C \text{ Input} + A B' C' \text{ Input}' + A B' C' \text{ Input} + A B' C' \text{ Input}' + A B' C \text{ Input}'$
 $\text{Input}' + A B' C \text{ Input};$
 $Y3 = A' B' C \text{ Input}' + A' B' C \text{ Input} + A' B C' \text{ Input}' + A' B C' \text{ Input} + A B' C \text{ Input}' + A B' C \text{ Input} + A B C' \text{ Input}' + A B C' \text{ Input}$
 $\text{Input}' + A B C' \text{ Input};$

Minimized:
 $S'0 = A' B C \text{ Input} + A' B' C' \text{ Input}' + A B' C + A C' \text{ Input} + A B \text{ Input}';$
 $S'1 = B' C \text{ Input} + B C' \text{ Input} + B C \text{ Input}' + B' C' \text{ Input}';$
 $S'2 = C' ;$
 $Y1 = A ;$
 $Y2 = A' B + A B' ;$
 $Y3 = B' C + B C' ;$

Ejercicio No. 4

- Se utiliza el nonblocking assignments para modelar logica synchronous sequential.
 - Ejemplo1:


```
always_ff @(posedge clk)
begin
    n1 <=d;
    q <=n1;
end
```
- El blocking assignment nos sirve para modelar lógica combinacional más complicada donde el estado always es de ayuda.
 - Ejemplo:


```
always_comb
begin
    p=a^b;
    g=a&b;
    s=p^cin;
    cout=g | (p& cin);
end
```