

Universidad Católica “Nuestra Señora de la Asunción”



Facultad de Ciencias y Tecnología
Ingeniería Electrónica
Sistemas Digitales

Sistemas Síncronos

Profesor:

Gerónimo Bellassai

Integrantes:

Esteban Gamarra - Y05084

Curso:

3ro.

Asunción – Paraguay
2020

Índice

Contenido	Páginas
Problema	...3
Justificación	...4
Parte Primera	...5-7
Parte Segunda (Particiones)	...8-14
Esquema del montaje	...15
Anexos (Esquemáticos de los integrados)	...16-22
74_195	...16
74_76	...17
74_74	...18
74_04	...19
74_08	...19
74_11	...20
74_21	...20
74_86	...21
74_32	...21
74_168	...22
Especificaciones LogicWork	...23-24

Problema

Diseñar un sistema secuencial síncrono de Moore consistente en un contador con dos entradas x_1 y x_2 , y una salida Z . con las siguientes características:

Inicialmente $Z = 0$ y pasa a 1 si se cumple la secuencia de entrada $(x_1; x_2) = (0;0) (0;1) (0;0) (1;0)$.

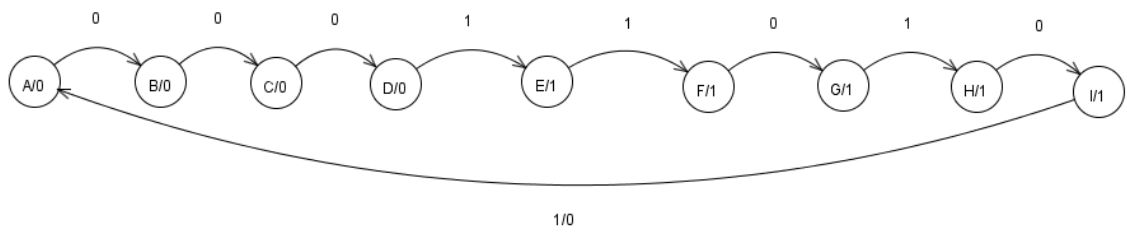
Estando en este estado la salida $Z = 1$, regresa a 0 luego de 10 ciclos de reloj o si se ha dado un par de veces la secuencia de entrada $(x_1; x_2) = (1;1) (0;0)$.

Al haberse cumplido el par de secuencias de $(1;1) (0;0)$ o el plazo de 10 ciclos de reloj el sistema se reinicializa, debiendo esperar nuevamente la secuencia de entrada.

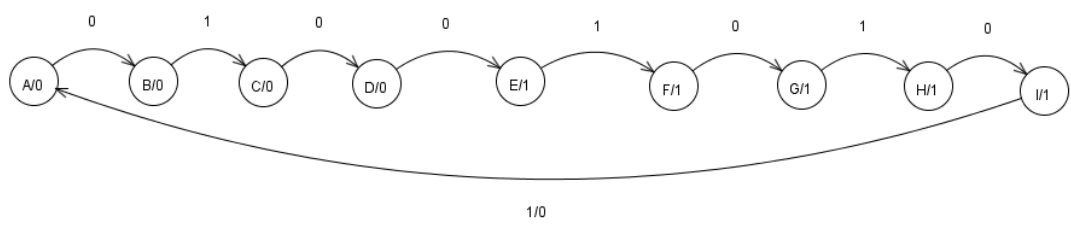
Justificación

El trabajo primera mente se pensó haciendo dos entradas por separado utilizando solamente 0's y 1's de la siguiente manera, además de haber ideado originalmente un autómata con 10 diez estados para cada uno de la siguiente manera.

Para X1



Para X2



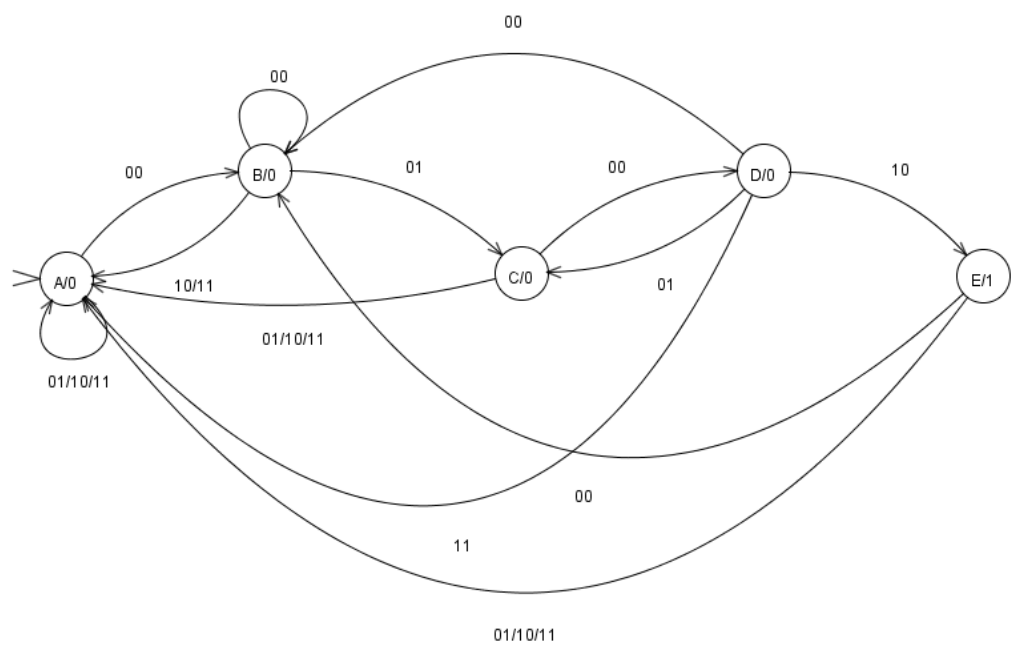
Pero esta forma resultó ser exageradamente larga y poco práctica. Además de esta forma al ser una sola solución para todo el problema no se hubiese podido separar el trabajo en partes a modo de aplicar el método de particiones en una parte del problema por separado como indica la guía de trabajo en pdf.

La solución a esto fue separar al trabajo en dos partes. Una parte cuando nuestro circuito se encuentra en el estado Z=0 y quiere pasar al estado Z=1. La segunda parte cuando nuestro circuito quiere pasar del estado Z=1 al Z=0.

Separando así el trabajo, se aplicó el método de particiones a la segunda parte, creando un autómata de la mitad del tamaño del original. Además, para ambas partes las entradas X1; X2 fueron tomadas juntas a modo de unificar los autómatas. De esta forma simplificando aún más el trabajo y la cantidad de materiales a ser utilizados a la hora del montaje tanto para una eficiencia práctica como económica.

En cuanto a la condición de que se cumplan 10 clocks para que se reinicialice el sistema, simplemente se utilizó un contador BCD tipo DIP, siendo este un parámetro para nuestra memoria, mediante un FF tipo T, que nos dirá si queremos reiniciar nuestro sistema o no.

Primera Parte: Secuencia para con Z=0 pasar a Z=1



A				
*	B			
AC				
*	BD	C		
BD	*			
AC	AE	BD	D	
AE	*	AC		
*		AE		
/	/	/	/	E

No hay simplificaciones posibles

Est. Act.	Est. Prox.				Z
	X1X2 00	X1X2 01	X1X2 11	X1X2 10	
A	B	A	A	A	0
B	B	C	A	A	0
C	D	A	A	A	0
D	B	C	A	E	0
E	B	A	A	A	1
F	/	/	/	/	X
G	/	/	/	/	X
H	/	/	/	/	X

Est. Act.	Est. Prox.				D2	D1	D0	Z
	X1X2 00	X1X2 01	X1X2 11	X1X2 10	X1X2 00-01-11-10	X1X2 00-01-11-10	X1X2 00-01-11-10	
A(000)	001	000	000	000	0000	0000	1000	0
B(001)	001	010	000	000	0000	0100	1000	0
C(010)	011	000	000	000	0000	1000	1000	0
D(011)	001	010	000	100	0001	0100	1000	0
E(100)	001	000	000	000	0000	0000	1000	1
F(101)	/	/	/	/	XXXX	XXXX	XXXX	X
G(110)	/	/	/	/	XXXX	XXXX	XXXX	X
H(111)	/	/	/	/	XXXX	XXXX	XXXX	X

X1=0

D2

		Y ₁ Y ₀			
		00	01	11	10
X2 y ₂	00	0	0	0	0
	01	0	X	X	X
	11	0	X	X	X
	10	0	0	0	0

X1=1

		Y ₁ Y ₀			
		00	01	11	10
X2 y ₂	00	0	0	1	0
	01	0	X	X	X
	11	0	X	X	X
	10	0	0	0	0

X1=0

D1

		Y ₁ Y ₀			
		00	01	11	10
X2 y ₂	00	0	0	0	1
	01	0	X	X	X
	11	0	X	X	X
	10	0	1	1	0

X1=1

		Y ₁ Y ₀			
		00	01	11	10
X2 y ₂	00	0	0	0	0
	01	0	X	X	X
	11	0	X	X	X
	10	0	0	0	0

X1=0

D0

		Y ₁ Y ₀			
		00	01	11	10
X2 y ₂	00	0	0	0	1
	01	0	X	X	X
	11	0	X	X	X
	10	0	1	1	0

X1=1

		Y ₁ Y ₀			
		00	01	11	10
X2 y ₂	00	0	0	0	0
	01	0	X	X	X
	11	0	X	X	X
	10	0	0	0	0

J2= X1./X2.Y1.Y0

J1 = /X1.X2.Y0

J0 = /X1./X2./Y1 + /X1./X2./Y2

Z = Y2 (Por deducción)

K2= 1

/K1= (/X1).(X2+/Y0).(/X2+Y0)

K1= X1 + /X2.Y0 + X2/Y0

/K0 = (/X1).(/X2)

K0 = X1+X2

Est. Act.	Est. Prox.				Z
	X1X2 00	X1X2 01	X1X2 11	X1X2 10	
A	A	A	B	A	0
B	C	A	B	A	0
C	A	A	D	A	0
D	E	A	B	A	0
E	A	A	B	A	1
F	/	/	/	/	X
G	/	/	/	/	X
H	/	/	/	/	X

Est. Act.	Est. Prox.				D2	D1	D0	Z
	X1X2 00	X1X2 01	X1X2 11	X1X2 10	X1X2 00-01-11-10	X1X2 00-01-11-10	X1X2 00-01-11-10	
A(000)	000	000	001	000	0000	0000	0010	0
B(001)	010	000	001	000	0000	1000	0010	0
C(010)	000	000	011	000	0000	0010	0010	0
D(011)	100	000	001	000	1000	0000	0010	0
E(100)	000	000	001	000	0000	0000	0010	1
F(101)	/	/	/	/	XXXX	XXXX	XXXX	X
G(110)	/	/	/	/	XXXX	XXXX	XXXX	X
H(111)	/	/	/	/	XXXX	XXXX	XXXX	X

Est.	$y_2y_1y_0$	x_1x_2 00	x_1x_2 01	x_1x_2 11	x_1x_2 10
A	000	A	A	B	A
B	001	C	A	B	A
C	010	A	A	D	A
D	011	E	A	B	A
E	100	A	A	B	A
F	/	/	/	/	/
G	/	/	/	/	/
H	/	/	/	/	/

Particiones:

P2 = ABCD, E

por y2

P1 = ABE, CD

por y1

P0 = ACE, BD

por y0

Px= AB, CD, E

producto

P2xP1

Py= AC, E, BD

producto

P2xP1

Pz= AE, C, B, D

producto

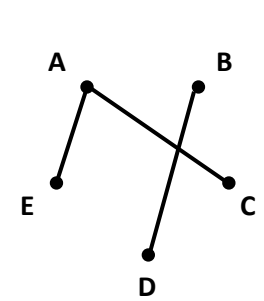
P1xP0

Pz es una partici3n cerrada π_z puesto que los estados pr3ximos son pertenecientes a estados 3nicos.

	x_1x_2 00	x_1x_2 01	x_1x_2 11	x_1x_2 10
AE	A	A	B	A
C	A	A	D	A
B	C	A	B	A
D	E	A	B	A

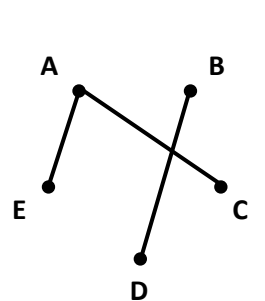
Ahora haciendo las particiones tomando los estados por pares buscaremos particiones cerradas.

Empezando por AB



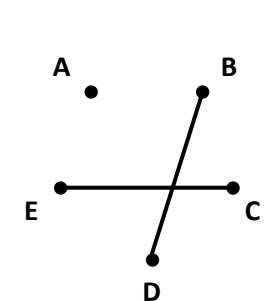
Pab = ACE, BD = Po

Empezando por AC



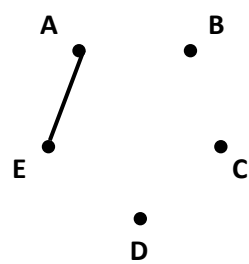
Pac = ACE, BD = Po

Empezando por BD



Pbd = A, BD, CE

Empezando por AD



$P_{AD} = \{AE, B, C, D\} = \pi_z$ y la partición es cerrada

La mayoría de las particiones llevan a la misma partición de $P_{AB} = \{ACE, BD\}$ la cual es cerrada dado que:

	X1X2 00	X1X2 01	X1X2 11	X1X2 10
ACE	A	A	BD	A
BD	AC	A	B	A

Entonces tenemos otra partición cerrada $P_0 = \pi_0$

Ahora también probamos si P_{BD} que es de las particiones más singulares, es también cerrada

	X1X2 00	X1X2 01	X1X2 11	X1X2 10
A	A	A	B	A
BD	CE	A	B	A
CE	A	A	BD	A

Y se concluye que también es una partición cerrada $P_{BD} = \pi_{bd}$

Ahora nuestro objetivo al hacer particiones es buscar particiones cerradas tal que por la regla del producto obtengamos mediante la regla del producto, la separación de todas nuestras variables, es decir:

$\pi_a \times \pi_b = A, B, C, D, E$ ó $\pi_a \times P_b = A, B, C, D, E$

Ahora analizamos nuestras particiones cerradas:

$\pi_0 = \{ACE, BD\} = 0, 1$

$\pi_z = \{AE, B, C, D\} = 00, 01, 10, 11$

$$\pi_{bd} = A, BD, CE = 00, 01, 11$$

Pero al hacer los productos ($\pi_0 \times \pi_z$; $\pi_0 \times \pi_{bd}$; $\pi_z \times \pi_{bd}$) no conseguimos entre todas las particiones cerradas no conseguimos una separación total de todas las variables. De modo que fabricamos una partición no cerrada derivada de una partición cerrada que ya poseemos, a modo de tener una distribución para nuestras tres variables $y_2y_1y_0$.

Para nuestro ejercicio, tomamos:

$$\pi_0 = ACE, BD = 0, 1 \text{ (representa 1 variable, } y_2\text{)}$$


Luego combinamos los estados a modo de tener una separación que pueda representarse con dos variables (y_1y_0)

$$P_{10} = AD, BE, C = 00, 01, 11 \text{ (Esta partición no es cerrada pero puede ser utilizada igualmente junto con la partición cerrada original para poder encontrar todas las relaciones que necesitamos)}$$

Ahora hacemos la regla del producto y verificamos si todos nuestros estados quedan separados:

$$\pi_0 \times P_{10} = ACE, BD \times AD, BE, C = A, E, C, D, B$$

Ahora reorganizamos nuestra tabla a modo de que representar nuestras particiones

Est.	$y_2y_1y_0$		Est.	$y_2y_1y_0$
A	000		A	000
B	101		E	001
C	011		/	010
D	100		C	011
E	001		D	100
F	/		B	101
G	/		/	110
H	/		/	111

Y reordenamos nuestra tabla de estados siguiente también

Est. Act.	Est. Prox.				D2	D1	D0	Z
	X1X2 00	X1X2 01	X1X2 11	X1X2 10	X1X2 00-01-11-10	X1X2 00-01-11-10	X1X2 00-01-11-10	
A(000)	000	000	101	000	0010	0000	0010	0
E(001)	000	000	101	000	0010	0000	0010	1
/ (010)	X	X	X	X	XXXX	XXXX	XXXX	/
C(011)	000	000	100	000	0010	0000	0000	0
D(100)	001	000	101	000	0010	0000	1010	0
B(101)	011	000	101	000	0010	1000	1010	0
/ (110)	X	X	X	X	XXXX	XXXX	XXXX	/
/ (111)	X	X	X	X	XXXX	XXXX	XXXX	/

X1=0

D2

		Y ₁ Y ₀			
		00	01	11	10
X2 y ₂	00	0	0	0	X
	01	0	0	X	X
	11	0	0	X	X
	10	0	0	0	X

X1=1

		Y ₁ Y ₀			
		00	01	11	10
X2 y ₂	00	0	0	0	X
	01	0	0	X	X
	11	1	1	X	X
	10	1	1	1	X

X1=0

D1

		Y ₁ Y ₀			
		00	01	11	10
X2 y ₂	00	0	0	0	X
	01	0	1	X	X
	11	0	0	X	X
	10	0	0	0	X

X1=1

		Y ₁ Y ₀			
		00	01	11	10
X2 y ₂	00	0	0	0	X
	01	0	0	X	X
	11	0	0	X	X
	10	0	0	0	X

X1=0

D0

		Y ₁ Y ₀			
		00	01	11	10
X2 y ₂	00	0	0	0	X
	01	1	1	X	X
	11	0	0	X	X
	10	0	0	0	X

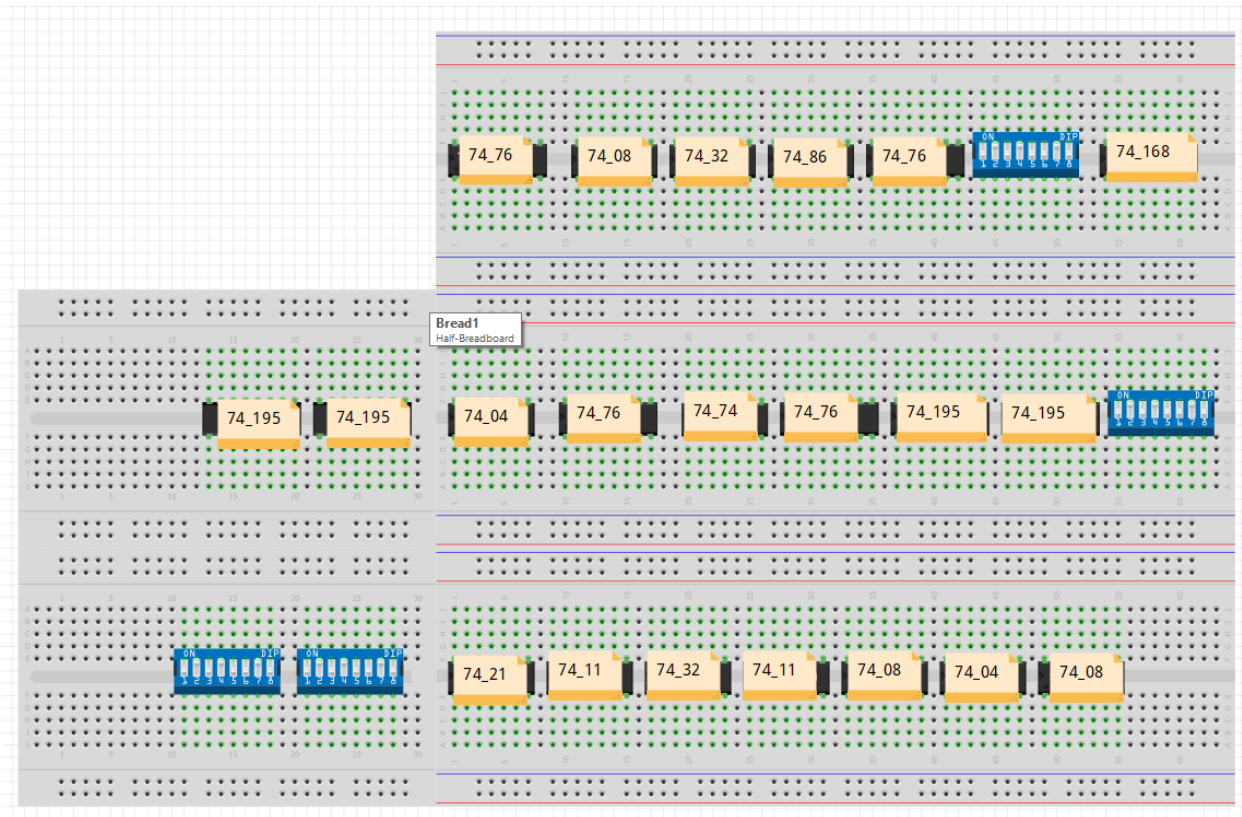
X1=1

		Y ₁ Y ₀			
		00	01	11	10
X2 y ₂	00	0	0	0	X
	01	0	0	X	X
	11	1	1	X	X
	10	1	1	0	X

$J2= X1.X2$
 $J1= /X1./X2.Y2.Y0$
 $J0= /X1./X2.Y2 + X1.X2$
 $Z=/Y2./Y1./Y0$ (Por deducción)

$/K2= (X1).(X2)$
 $K2= /X1 + /X2$
 $K1= 1$
 $/K0= (X2+Y2).(/Y1).(/X2+X1).(X2+/X1)$
 $K0= /X2./Y2 + /Y1 + X2./X2 + /X2.X1$

Mapa del montaje.



Siendo los integrados los siguientes:

- 74_195 : Parallel in-serial out shift register
- 74_76: Dual JK flip flop
- 74_74: Dual D flip-flop
- 74_04: Quad-NOT gate
- 74_08: Quad 2 AND gate
- 74_11: Triple 3 AND gate
- 74_21: Dual 4 AND gate
- 74_86: Quad 2 XOR gate
- 74_32: Quad 2 OR gate
- 74_168: 4-stage synchronous bidirectional counter
- DipSwitch

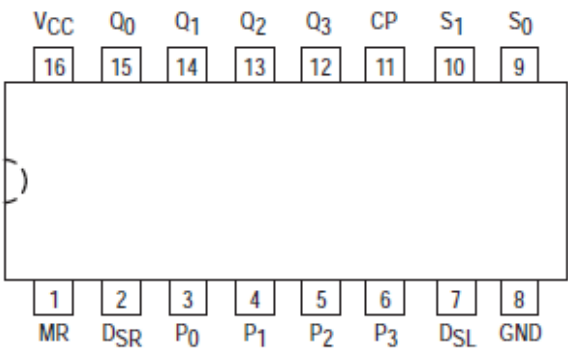
Para el clock simplemente se programó una señal controlada con el Arduino que varíe cada 1,5 segundos.

UNIVERSAL 4-BIT SHIFT REGISTER

FUNCTIONAL DESCRIPTION

The F194 contains four edge-triggered D flip-flops and the necessary interstage logic to synchronously perform shift right, shift left, parallel load and hold operations. Signals applied to the Select (S_0, S_1) inputs determine the type of operation, as shown in the Function Table. Signals on the Select, Parallel data (P_0-P_3) and Serial data (D_{SR}, D_{SL}) inputs can change when the clock is in either state, provided only that the recommended setup and hold times, with respect to the clock rising edge, are observed. A LOW signal on Master Reset (MR) overrides all other inputs and forces the outputs LOW.

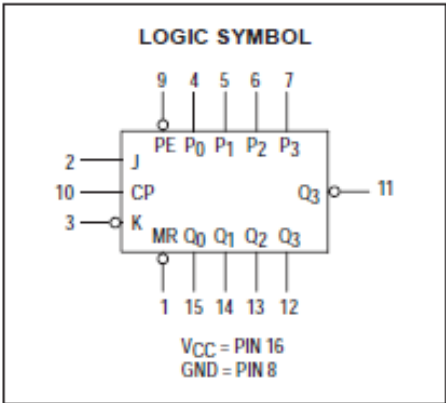
CONNECTION DIAGRAM



FUNCTION TABLE

Operating Mode	Inputs						Outputs			
	MR	S ₁	S ₀	D _{SR}	D _{SL}	P _n	Q ₀	Q ₁	Q ₂	Q ₃
Reset	L	X	X	X	X	X	L	L	L	L
Hold	H	l	l	X	X	X	q ₀	q ₁	q ₂	q ₃
Shift Left	H	h	l	X	l	X	q ₁	q ₂	q ₃	L
	H	h	l	X	h	X	q ₁	q ₂	q ₃	H
Shift Right	H	l	h	l	X	X	L	q ₀	q ₁	q ₂
	H	l	h	h	X	X	H	q ₀	q ₁	q ₂
Parallel Load	H	h	h	X	X	p _n	p ₀	p ₁	p ₂	p ₃

l = LOW voltage level one setup time prior to the LOW-to-HIGH clock transition.
h = HIGH voltage level one setup time prior to the LOW-to-HIGH clock transition.
p_n, q_n = Lower case letters indicate the state of the referenced input or output one setup time prior to the LOW-to-HIGH clock transition.
H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial



DUAL JK FLIP-FLOP WITH SET AND CLEAR

The SN54/74LS76A offers individual J, K, Clock Pulse, Direct Set and Direct Clear inputs. These dual flip-flops are designed so that when the clock goes HIGH, the inputs are enabled and data will be accepted. The Logic Level of the J and K inputs will perform according to the Truth Table as long as minimum set-up times are observed. Input data is transferred to the outputs on the HIGH-to-LOW clock transitions.

MODE SELECT — TRUTH TABLE

OPERATING MODE	INPUTS				OUTPUTS	
	$\overline{S_D}$	$\overline{C_D}$	J	K	Q	\overline{Q}
Set	L	H	X	X	H	L
Reset (Clear)	H	L	X	X	L	H
*Undetermined	L	L	X	X	H	H
Toggle	H	H	h	h	q	q
Load "0" (Reset)	H	H	l	h	L	H
Load "1" (Set)	H	H	h	l	H	L
Hold	H	H	l	l	q	q

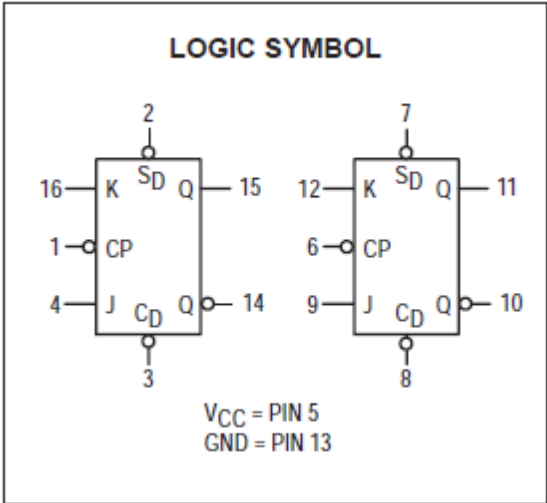
*Both outputs will be HIGH while both $\overline{S_D}$ and $\overline{C_D}$ are LOW, but the output states are unpredictable if $\overline{S_D}$ and $\overline{C_D}$ go HIGH simultaneously.

H,h = HIGH Voltage Level

L,l = LOW Voltage Level

X = Immaterial

l, h (q) = Lower case letters indicate the state of the referenced input (or output) one setup time prior to the HIGH-to-LOW clock transition

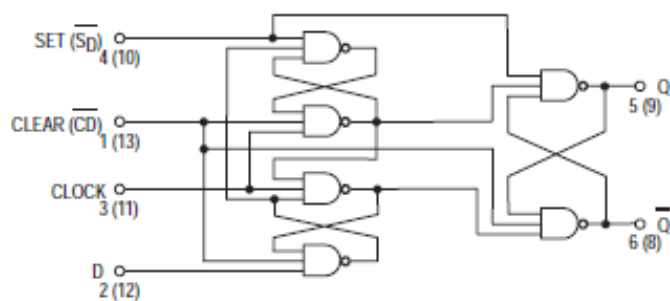


DUAL D-TYPE POSITIVE EDGE-TRIGGERED FLIP-FLOP

The SN54/74LS74A dual edge-triggered flip-flop utilizes Schottky TTL circuitry to produce high speed D-type flip-flops. Each flip-flop has individual clear and set inputs, and also complementary Q and Q outputs.

Information at input D is transferred to the Q output on the positive-going edge of the clock pulse. Clock triggering occurs at a voltage level of the clock pulse and is not directly related to the transition time of the positive-going pulse. When the clock input is at either the HIGH or the LOW level, the D input signal has no effect.

LOGIC DIAGRAM (Each Flip-Flop)



MODE SELECT — TRUTH TABLE

OPERATING MODE	INPUTS			OUTPUTS	
	S_D	$\overline{S_D}$	D	Q	\overline{Q}
Set	L	H	X	H	L
Reset (Clear)	H	L	X	L	H
*Undetermined	L	L	X	H	H
Load "1" (Set)	H	H	h	H	L
Load "0" (Reset)	H	H	l	L	H

* Both outputs will be HIGH while both S_D and C_D are LOW, but the output states are unpredictable if S_D and C_D go HIGH simultaneously. If the levels at the set and clear are near V_{IL} maximum then we cannot guarantee to meet the minimum level for V_{OH} .

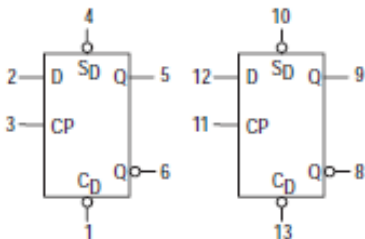
H, h = HIGH Voltage Level

L, l = LOW Voltage Level

X = Don't Care

l, h (q) = Lower case letters indicate the state of the referenced input (or output) one set-up time prior to the HIGH to LOW clock transition.

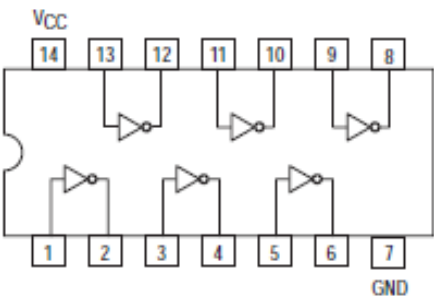
LOGIC SYMBOL



V_{CC} = PIN 14
GND = PIN 7

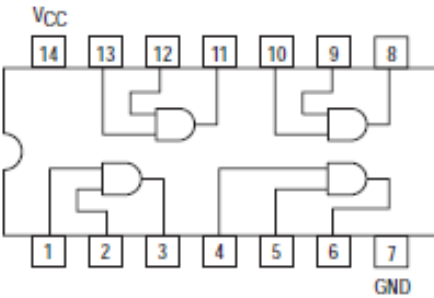
74_04

HEX INVERTER



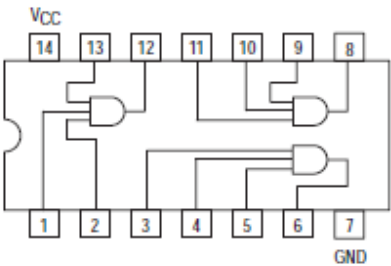
74_08

QUAD 2-INPUT AND GATE



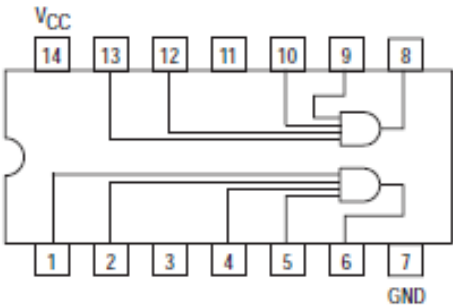
74_11

TRIPLE 3-INPUT AND GATE



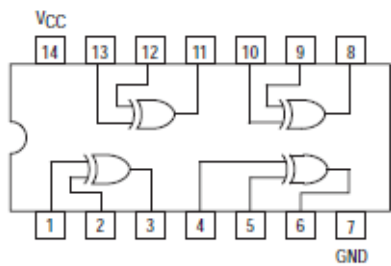
74_21

DUAL 4-INPUT AND GATE



74_86

QUAD 2-INPUT
EXCLUSIVE OR GATE

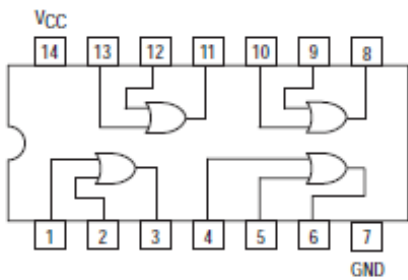


TRUTH TABLE

IN		OUT
A	B	Z
L	L	L
L	H	H
H	L	H
H	H	L

74_32

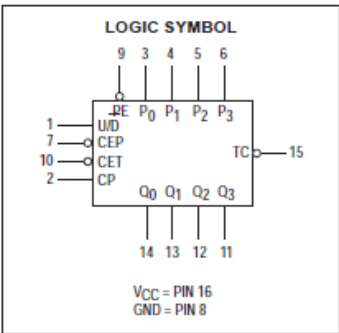
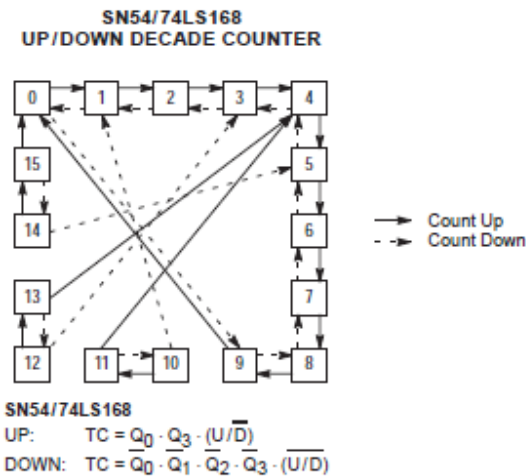
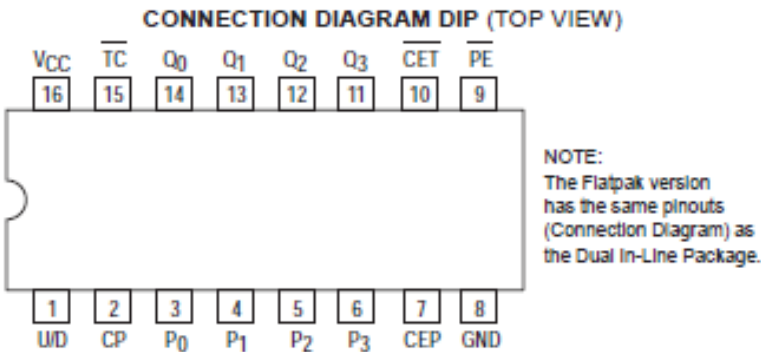
QUAD 2-INPUT OR GATE



BCD DECADE/MODULO 16 BINARY SYNCHRONOUS BI-DIRECTIONAL COUNTERS

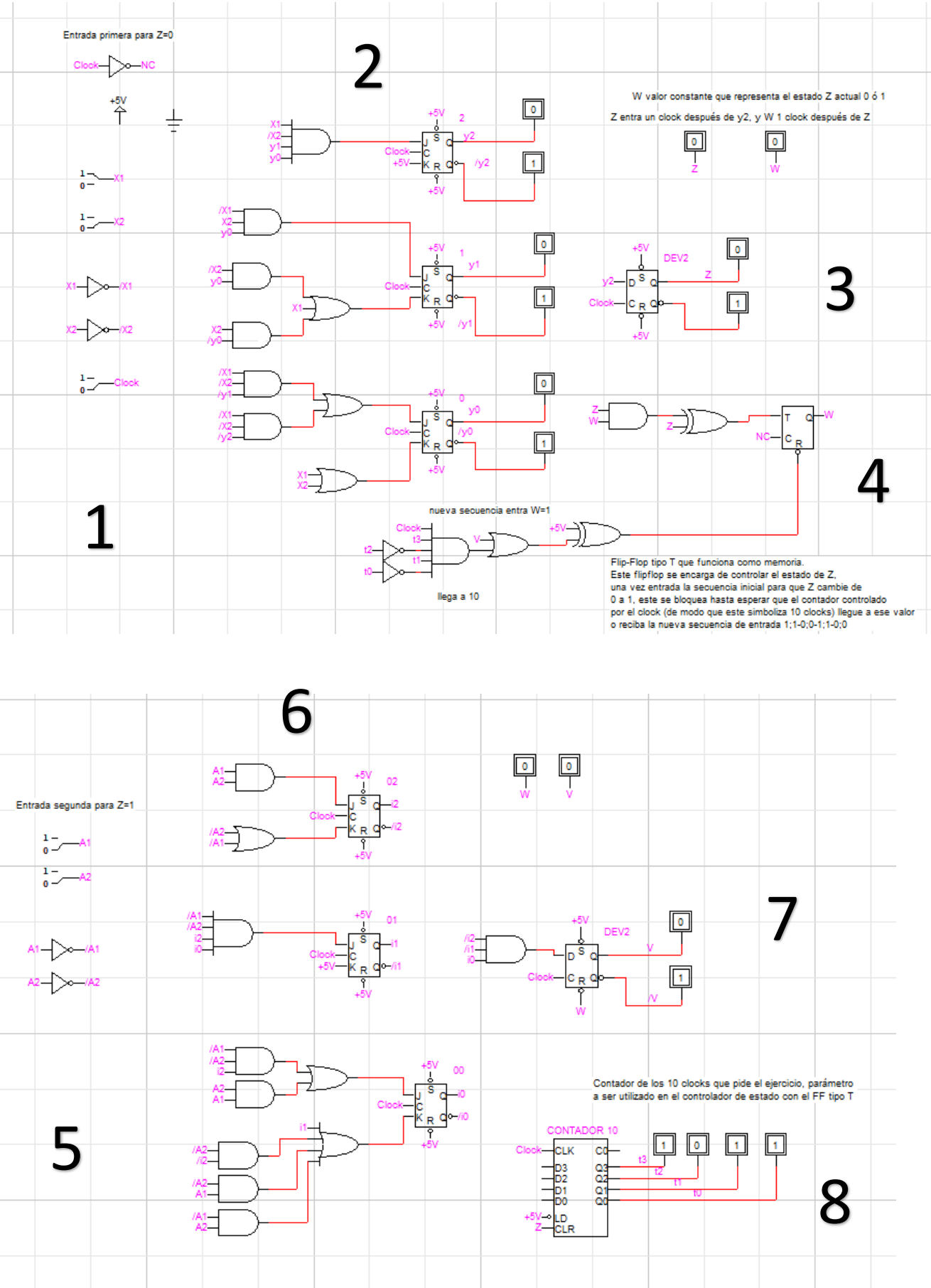
The SN54/74LS168 and SN54/74LS169 are fully synchronous 4-stage up/down counters featuring a preset capability for programmable operation, carry lookahead for easy cascading and a U/D input to control the direction of counting. The SN54/74LS168 counts in a BCD decade (8, 4, 2, 1) sequence, while the SN54/74LS169 operates in a Modulo 16 binary sequence. All state changes, whether in counting or parallel loading, are initiated by the LOW-to-HIGH transition of the clock.

- Low Power Dissipation 100 mW Typical
- High-Speed Count Frequency 30 MHz Typical
- Fully Synchronous Operation
- Full Carry Lookahead for Easy Cascading
- Single Up/Down Control Input
- Positive Edge-Trigger Operation
- Input Clamp Diodes Limit High-Speed Termination Effects



Especificaciones del LogicWork (Simulación)

El funcionamiento de la simulación es el siguiente:



1. Son las entradas X1 y X2, el clock y las alimentaciones
2. La parte fundamental que representa los Flip-flop JK para Y2, Y1 e Y0 del detector de secuencias
3. Flip-flop tipo D que aceptará la secuencia y cambiará de estado
4. Flip-flop tipo T que se encargará de actuar como “memoria”, manteniendo en W el estado actual de Z que este representa. Este posee un reset controlado por 10 clock o la secuencia segunda de X1, X2 para cambiar de estado Z=1 a Z=0 (en nuestro ejemplo esto

se controla mediante un Flip-flop tipo D que se representa por la letra “V”, la cual es un parámetro dentro del circuito para el reset)

5. Entradas X1 y X2 para cuando nuestro sistema esté en el estado Z=1, en nuestro caso representado por W
6. Flip-flop JK del detector de secuencias mientras W=1
7. Flip-flop tipo D que aceptará la secuencia y cambiará de estado V=0 a V=1
8. Contador BCD, parámetro para el reset de nuestra memoria, controlado por el clock