



## **CURSO: ELECTRÓNICA DIGITAL**

### **UNIDAD 4: CONTROLADORES DIGITALES - TEORÍA**

#### **PROFESOR: JORGE ANTONIO POLANÍA**

En esta unidad se aprenderá a diseñar controladores digitales con base en la lógica combinatoria y secuencial estudiada en las unidades anteriores. Los diseños se harán con base en flip flop tipo D, registros de corrimiento y contadores. Este tipo de controladores nos permitirá realizar circuito de control para la automatización de procesos y se expone su aplicación en una máquina de servicios de líquidos.

La metodología como en todas las unidades de los cursos se realiza mediante explicaciones teóricas, simulaciones en laboratorio virtual, prácticas en laboratorio y la correspondiente evaluación

Los controladores digitales son circuitos que utilizando flip flops, registros o contadores pueden realizar un control a una máquina con base en la lógica combinatoria o secuencial vista en las anteriores unidades. Se aplicará esta metodología para resolver el problema de automatización de una máquina de servicios de líquido. Si se quiere profundizar en este tema sugiero el libro de Fletcher William "An engineering approach to digital design" editado por Prentice Hall.

### **1. MÓDULOS DE LA MÁQUINA DE SERVICIOS DE LÍQUIDO**

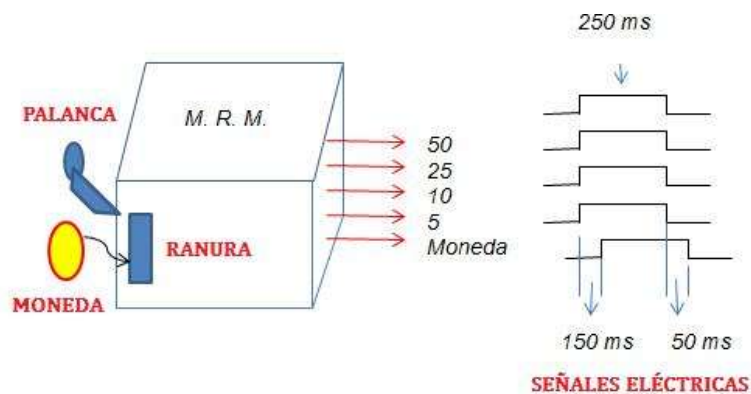
Para su estudio la máquina se dividirá en los siguientes módulos:

1. Módulo de recepción de monedas
2. Módulo de cambio y
3. Módulo de servicio

Los módulos tienen las siguientes características:

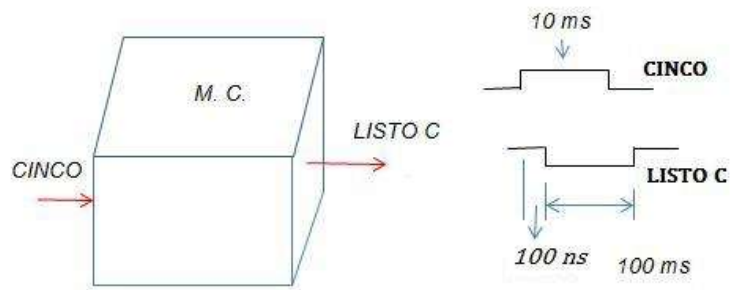
## 1. MÓDULO RECEPCIÓN DE MONEDAS

- a) Tiene una sola ranura para la inserción de monedas
- b) Hace detección electrónica de monedas
- c) Reconoce monedas de 50¢, 25¢, 10¢ y 5¢ centavos de dólar
- d) Tiene un sistema para prevenir el sobreflujo de monedas
- e) tiene una palanca para devolución de monedas
- f) Existen señales eléctricas asociadas a las monedas reconocidas



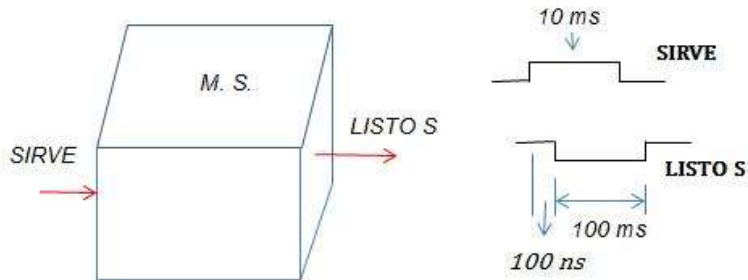
## 2. MÓDULO DE CAMBIO

- a) Acepta una señal tipo TTL (CINCO) para entregar una moneda de 5¢ como cambio.
- b) Tiene una señal eléctrica de salida que indica cuándo el módulo de cambio está listo para iniciar un nuevo proceso de cambio (LISTO C)



### 3. MÓDULO DE SERVICIO

- a) Acepta una señal (SIRVE) y
- b) Entrega una señal de listo servicio (LISTO S)



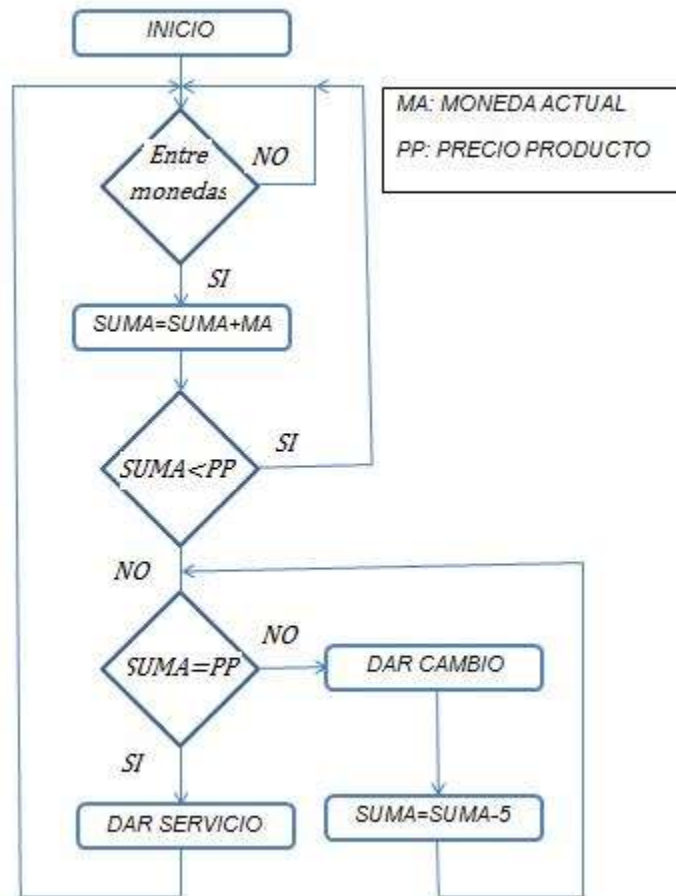
### OBJETIVO DEL PROBLEMA

Diseñar un controlador digital que interactúe con los módulos tal como se indica en la figura.

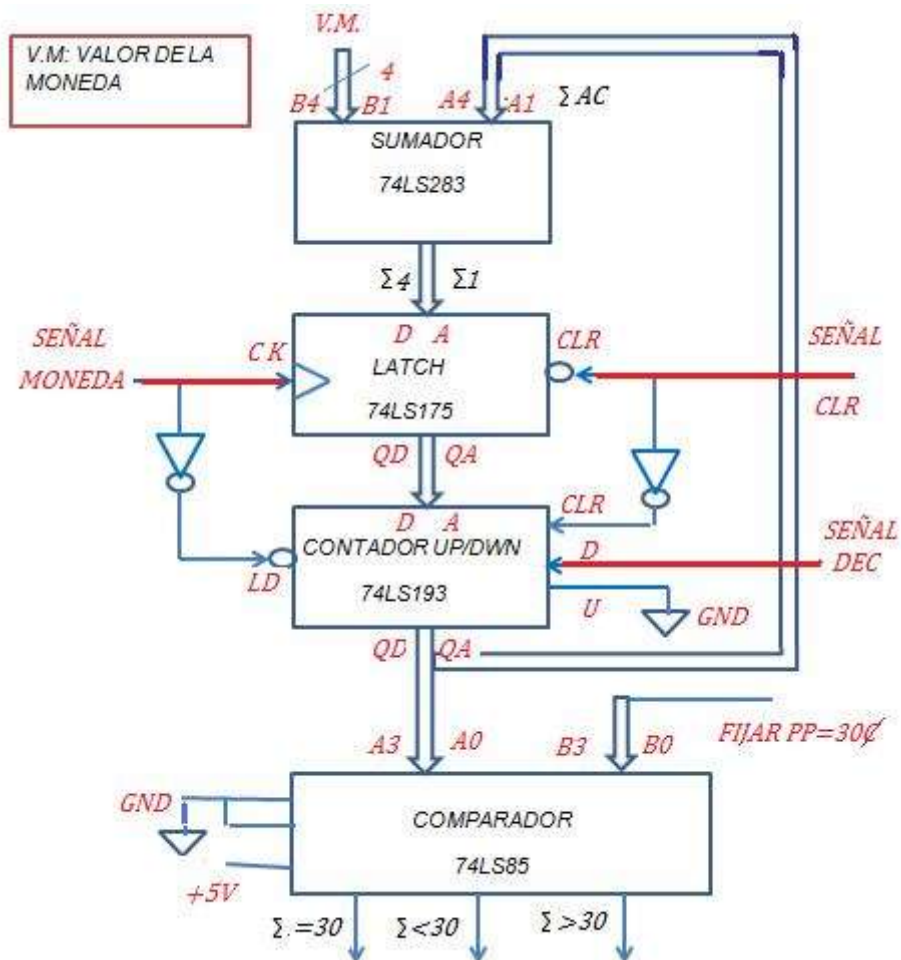


## 2. DIAGRAMAS INICIALES

### DIAGRAMA DE FLUJO INICIAL



## CIRCUITO QUE EVALÚA LA SUMA ACUMULADA

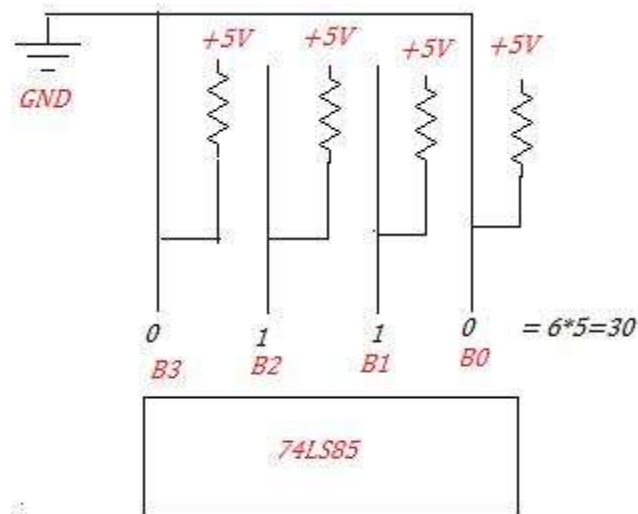


## HARDWARD ADICIONAL

Este es el hardware que se debe adicionar al circuito anterior.

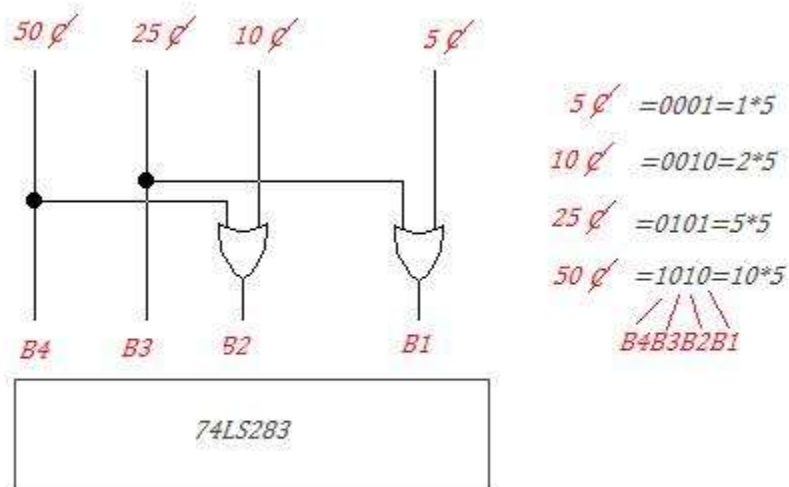
FIJAR PP = 30¢

Dato binario B3B2B1B0 de entrada al comparador 74LS85 que es el precio del producto.

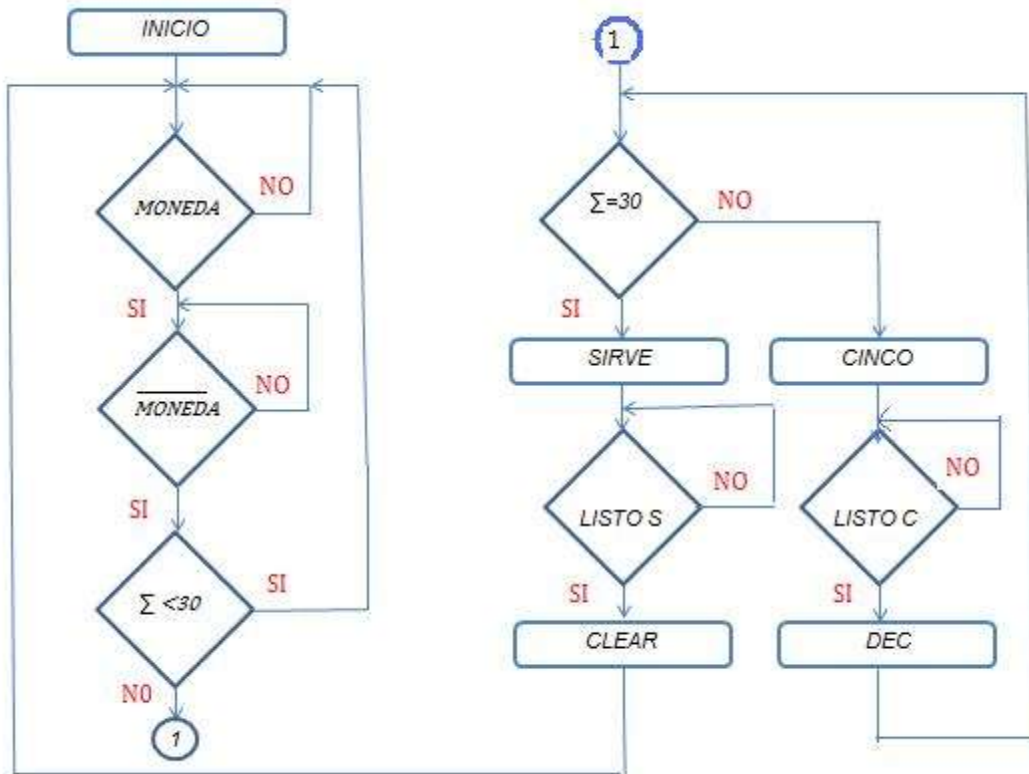


## V.M. VALOR DE LA MONEDA

Dato binario B4B3B2B1 de entrada al sumador de 4 bits 74LS283 que da un código según el valor de la moneda.



### 3. DIAGRAMA DE FLUJO DEL CONTROLADOR



### 4. DISEÑO DEL CONTROLADOR CON LATCH

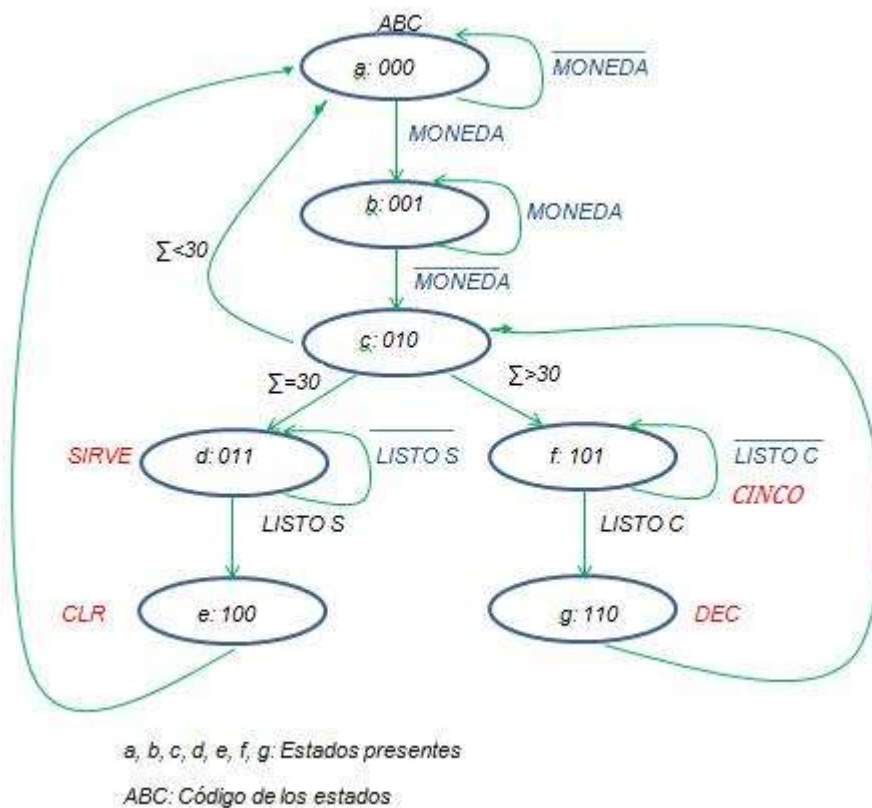
#### PASOS DE DISEÑO

- 1) Asignar códigos a los estados del autómata respetando la regla de distancia mínima utilizada en los Mapas de Karnaugh
- 2) Hacer el mapa del estado presente (E.P.)
- 3) Hacer el mapa del estado siguiente (E.S.) indicando los estados siguientes así como las variables que indican el cambio. Indicar el código del estado siguiente.
- 4) Obtener los mapas del estado siguiente para cada una de las variables de estado del problema en cuestión

- 5) Hacer mapas de salidas
- 6) Implementar la solución más adecuada con:
  - a) Lógica combinatoria para codificadores de entrada y salida
  - b) Multiplexores como decodificadores de entrada y decodificadores para las salidas
  - c) Combinaciones de a) y b)

## SOLUCIÓN AL PROBLEMA - AUTÓMATA

1) Realizar el autómata donde se presenten sus estados. Se recomienda asignar códigos con distancia mínima, esto es, 000, 001, 011, etc. De esta forma se simplifican las funciones en los mapas.





2) Mapa del estado presente (M.E.P.)

		<i>AB</i>			
		00	01	11	10
<i>C</i>	0	<i>a</i>	<i>c</i>	<i>g</i>	<i>e</i>
	1	<i>b</i>	<i>d</i>	*	<i>f</i>

*M.E.P.*

3) Mapa del estado siguiente (M.E.S.)

		<i>AB</i>			
		00	01	11	10
<i>C</i>	0	<i>MON'</i> -> <i>a:000</i> <i>MON</i> -> <i>b:001</i>	$\Sigma <$ -> <i>a:000</i> $\Sigma =$ -> <i>d:011</i> $\Sigma >$ -> <i>f:101</i>	<i>c:010</i>	<i>a:000</i>
	1	<i>MON'</i> -> <i>b:001</i> <i>MON</i> -> <i>c:010</i>	<i>LISTO S'</i> -> <i>d:011</i> <i>LISTO S</i> -> <i>e:100</i>	*	<i>LISTO C</i> -> <i>g:110</i>

*M.E.S.*

4) Mapa del estado siguiente de las variables.

Las variables de los estados con don't care (\*) se dejan = 0.

		<i>AB</i>			
		<i>00</i>	<i>01</i>	<i>11</i>	<i>10</i>
<i>C</i>	<i>0</i>	<i>0</i>	$\Sigma >$	<i>0</i>	<i>0</i>
	<i>1</i>	<i>0</i>	<i>LISTO S</i>	<i>0</i>	<i>1</i>

$$D_A = A'BC'(\Sigma >) + A'BC(LISTO S) + AB'C$$

		<i>AB</i>			
		<i>00</i>	<i>01</i>	<i>11</i>	<i>10</i>
<i>C</i>	<i>0</i>	<i>0</i>	$\Sigma =$	<i>1</i>	<i>0</i>
	<i>1</i>	<i>MON'</i>	<i>LISTO S'</i>	<i>0</i>	<i>LISTO C</i>

$$D_B = A'BC'(\Sigma =) + ABC' + A'B'C(MON') + A'BC(LISTO S') + AB'C(LISTO C)$$

		<i>AB</i>			
		<i>00</i>	<i>01</i>	<i>11</i>	<i>10</i>
<i>C</i>	<i>0</i>	<i>MON</i>	$\Sigma <$	<i>0</i>	<i>0</i>
	<i>1</i>	<i>MON</i>	<i>LISTO S'</i>	<i>0</i>	<i>LISTO C'</i>

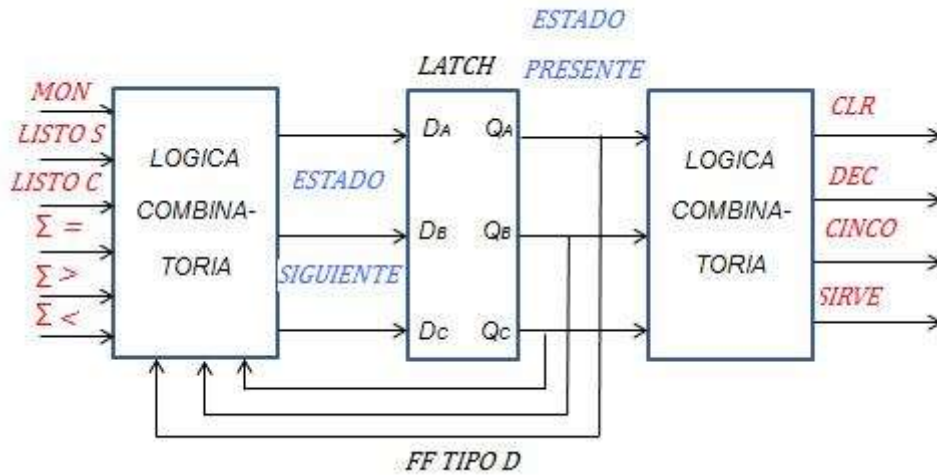
$$D_C = A'B'(MON) + A'BC'(\Sigma <) + A'BC(LISTO S') + AB'C(LISTO C')$$

### 5) Mapa de salidas

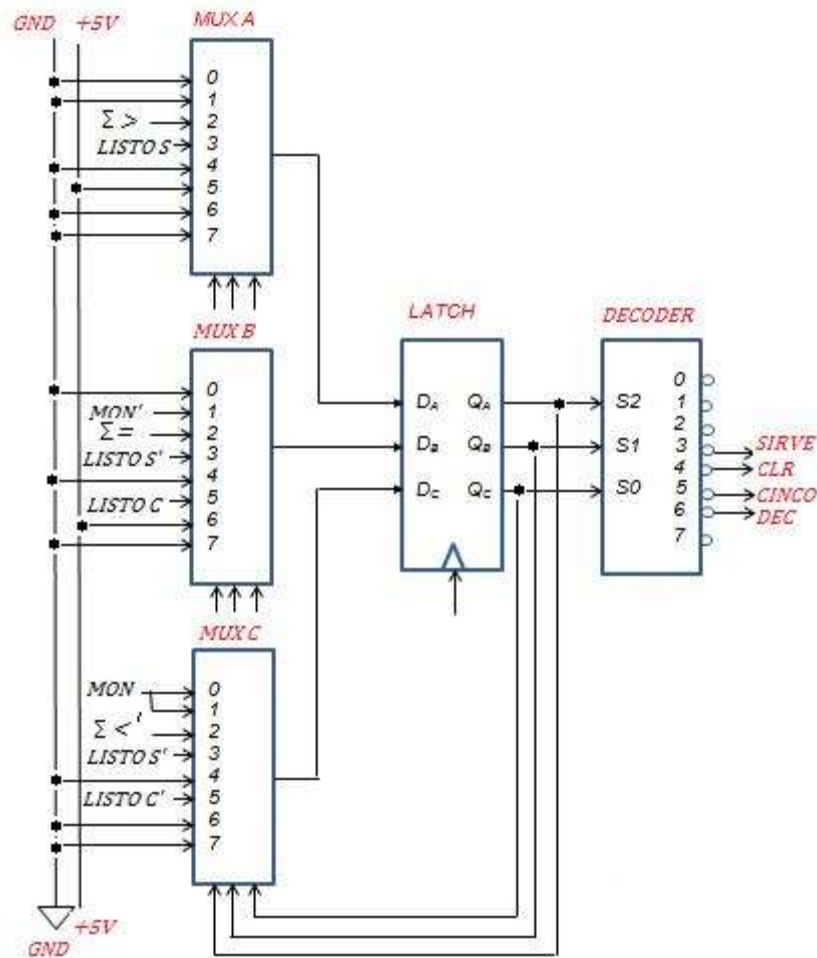
		<i>AB</i>			
		<i>00</i>	<i>01</i>	<i>11</i>	<i>10</i>
<i>C</i>	<i>0</i>	<i>*</i>	<i>*</i>	<i>DEC</i>	<i>CLR</i>
	<i>1</i>	<i>*</i>	<i>SIRVE</i>	<i>*</i>	<i>CINCO</i>

$$DEC = ABC' \quad SIRVE = A'B'C \quad CLR = AB'C' \quad CINCO = AB'C$$

### 6a) Implementación con lógica combinatoria

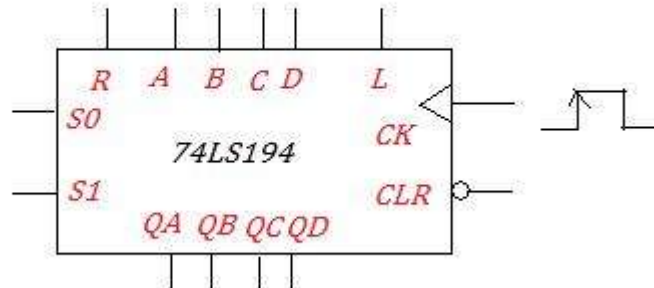


6b) Implementación con Multiplexores y Decodificador a la salida.



## 5. DISEÑO DEL CONTROLADOR CON SHIFT REGISTER

Se utiliza el registro de corrimiento 74LS194.



Tiene las siguientes características:

- Síncrono y actúa con el flanco de subida
- Permite cargar valores (LOAD) de bits por las entradas A, B, C, D
- Permite corrimientos a la izquierda (SL) y a la derecha (SR)
- Define un estado HOLD por tiempo indefinido
- Tiene 4 modos de control fijados por S0 y S1

S0	S1	ACCIÓN
0	0	HOLD
0	1	SL
1	0	SR
1	1	LOAD

### Posibles instrucciones de control

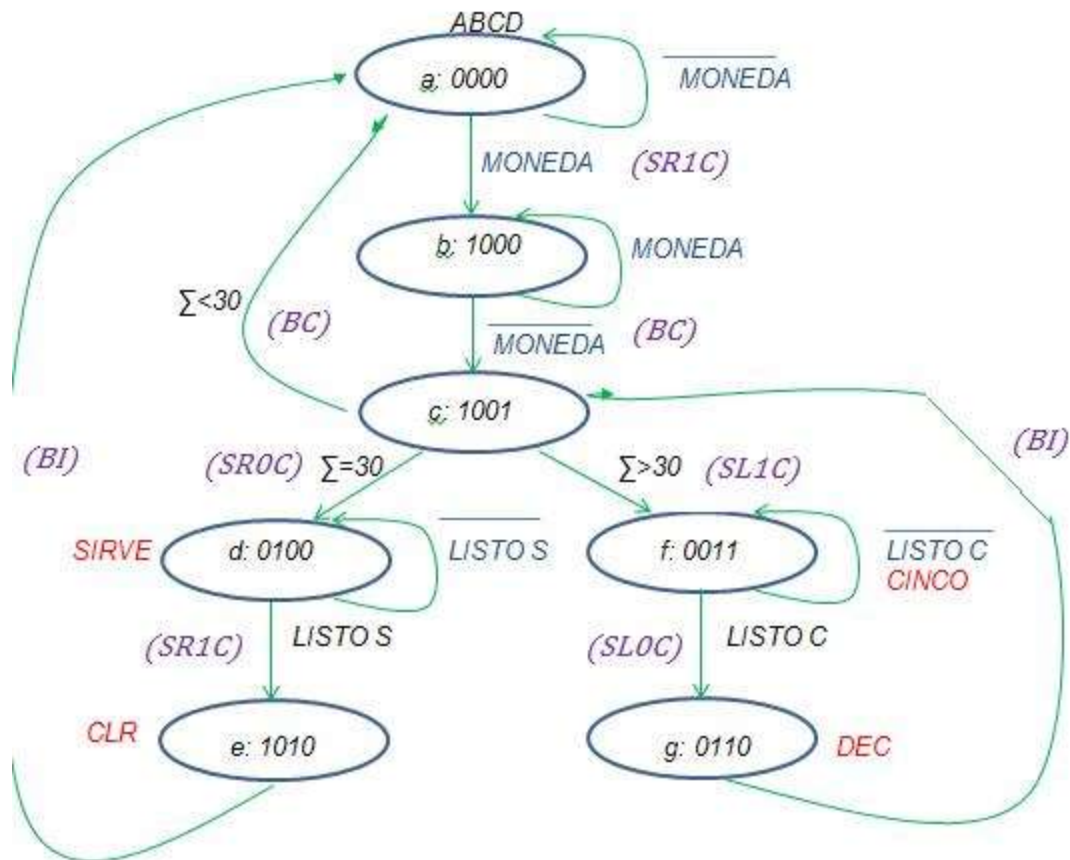
- SL0 (L=0), SL1 (L=1), SL0C, SL1C (Condicionados)
- SR0, SR1 (Incondicionados), SR0C (R=0), SR1C (R=1)
- BI (Brinco incondicional)
- BC (Brinco condicionado)

### PASOS DE DISEÑO

- Asignar código a los estados del autómata, tratando de usar en lo posible algún tipo de instrucción de las definidas
- Hacer el mapa de M.E.P.
- Hacer el mapa de M.E.S. indicando los tipos de acción del registro, así como la variable de condición
- Hacer mapas para los estados de control S0 y S1
- Hacer los mapas de los bits de carga en paralelo, así como, para R y L
- Hacer los mapas de las salidas
- Implementar con lógica combinatoria a la entrada, registro y decodificador a la salida.

### SOLUCIÓN AL PROBLEMA - AUTÓMATA

- Asignar código a los estados del autómata, tratando de usar en lo posible algún tipo de instrucción de las definidas



a, b, c, d, e, f, g: Estados presentes

ABCD: Código de los estados

b) Hacer el mapa de M.E.P.

<i>AB</i>	00	01	11	10
<i>CD</i>				
00	a	d	*	b
01	*	*	*	c
11	f	*	*	*
10	*	g	*	e

c) Hacer el mapa de M.E.S. indicando los tipos de acción del registro, así como la variable de condición

<i>AB</i>	00	01	11	10
<i>CD</i>				
00	SR1C, MON	SR1C, LISTO S	*	BC, MON' c
01	*	*	*	BC, $\Sigma < a$ SR0C, $\Sigma =$ SL1C, $\Sigma >$
11	SL0C, LISTO C	*	*	*
10	*	BI, c	*	BI, a

d) Hacer mapas para los estados de control S0 y S1. Líneas de control para el registro de corrimiento 74LS194.

<i>AB</i>	00	01	11	10
<i>CD</i>				
00	MON	LISTO S	1	MON'
01	1	1	1	$\Sigma >'$
11	0	1	1	1
10	1	1	1	1

*CONTROL S0*

	<i>AB</i>	00	01	11	10
<i>CD</i>	00	0	0	1	MON'
	01	1	1	1	$\Sigma ='$
	11	LISTO C	1	1	1
	10	1	1	1	1

*CONTROL S1*

Explicación: Observando el autómata,

estado a:

si MON=0 -->HOLD, entonces S0, S1=0 0

si MON=1 -->SR, entonces S0, S1=1 0,

deducción: S0= MON, S1=0

estado b:

si MON=1 -->HOLD, entonces S0, S1=0 0

si MON=0 -->BC, entonces S0, S1=1 1,

deducción: S0= MON', S1=MON'

estado c:

si  $\Sigma <$  -->BC, entonces S0, S1=1 1

si  $\Sigma =$  -->SR, entonces S0, S1=1 0,

si  $\Sigma >$  -->SL, entonces S0, S1=0 1,

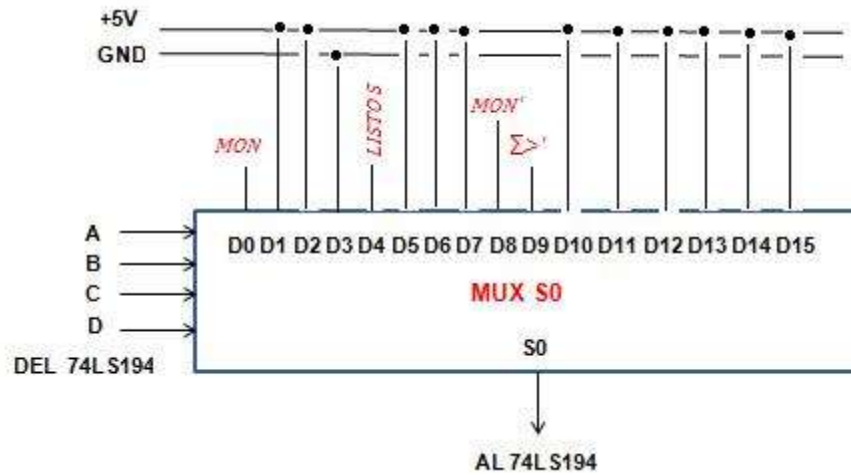
deducción: S0=  $\Sigma >$ ', S1= $\Sigma =$ '

estado e, g:

son BI, S0,S1=1 1

Los demás estados se dejan para brincos, S0, S1 = 11

Estas salidas de control para el registro se implementan con multiplexores. Por ejemplo, para S0:



e) Hacer los mapas de los bits de carga en paralelo A, B, C, D (brincos), así como, para R y L (corrimientos).

Observar los brincos del M.E.S. En estos mapas los bits verdes corresponden a los brincos. Los don't care se ponen en ceros para evitar brincos no programados.

	AB			
	00	01	11	10
CD	00	0	0	0
	01	0	0	0
	11	0	0	0
	10	0	1	0

A

	AB			
	00	01	11	10
CD	00	0	0	0
	01	0	0	0
	11	0	0	0
	10	0	0	0

B



	<i>AB</i>	00	01	11	10
<i>CD</i>	00	0	0	0	0
	01	0	0	0	0
	11	0	0	0	0
	10	0	0	0	0

*C*

	<i>AB</i>	00	01	11	10
<i>CD</i>	00	0	0	0	1
	01	0	0	0	0
	11	0	0	0	0
	10	0	1	0	0

*D*

	<i>AB</i>	00	01	11	10
<i>CD</i>	00	1	1	*	*
	01	*	*	*	0
	11	*	*	*	*
	10	*	*	*	*

*R*

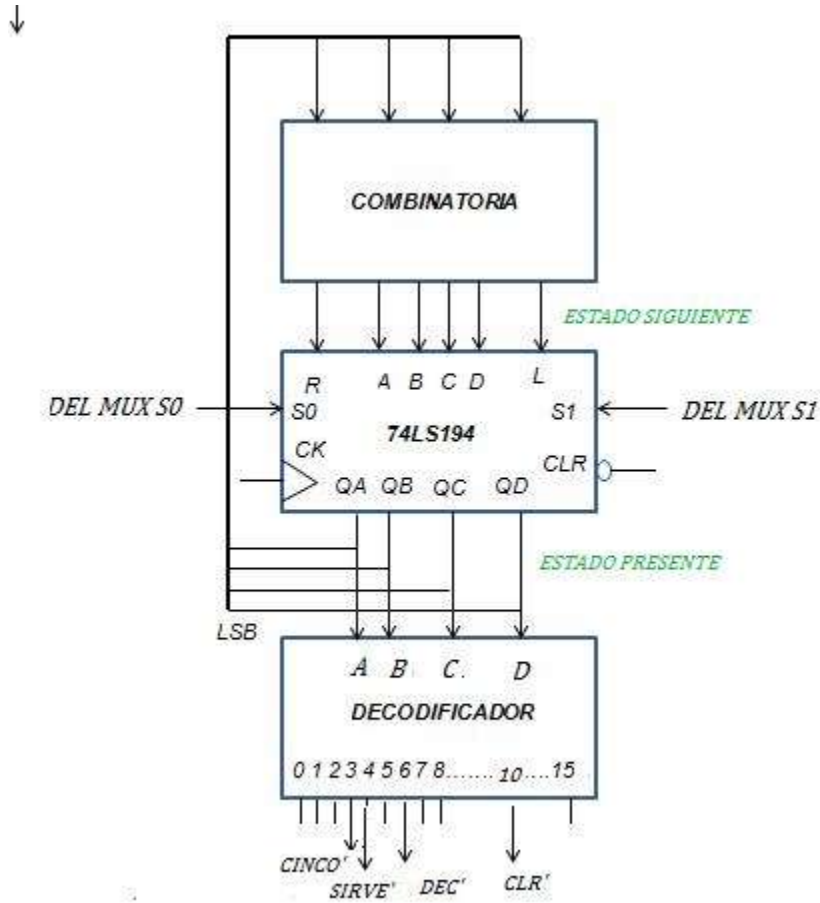
	<i>AB</i>	00	01	11	10
<i>CD</i>	00	*	*	*	*
	01	*	*	*	1
	11	0	*	*	*
	10	*	*	*	*

*L*

f) Hacer los mapas de las salidas

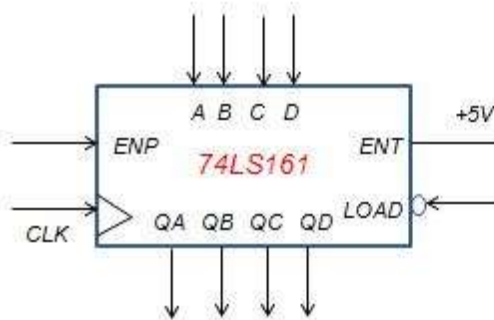
	<i>AB</i>	00	01	11	10
<i>CD</i>	00		SIRVE		
	01				
	11	CINCO			
	10		DEC		CLR

g) Implementar con lógica combinatoria a la entrada, registro y decodificador a la salida.



## 5. DISEÑO DEL CONTROLADOR CON CONTADORES

Se utiliza el contador 74LS161 que tiene la siguiente tabla de control de modo.



ENP	LOAD	ACCION
0	0	HOLD
0	1	LOAD
1	0	CUENTA
1	1	LOAD

TABLA DE CONTROL DE MODO

## Características del 74LS161:

- Es de 4 bits y posibilidad de conexión en cascada
- Tiene habilitador de cuentas ENP
- Permite hacer cargas en paralelo con LOAD
- Clear asíncrono
- Carga con el flanco de subida del reloj

## Instrucciones o comandos:

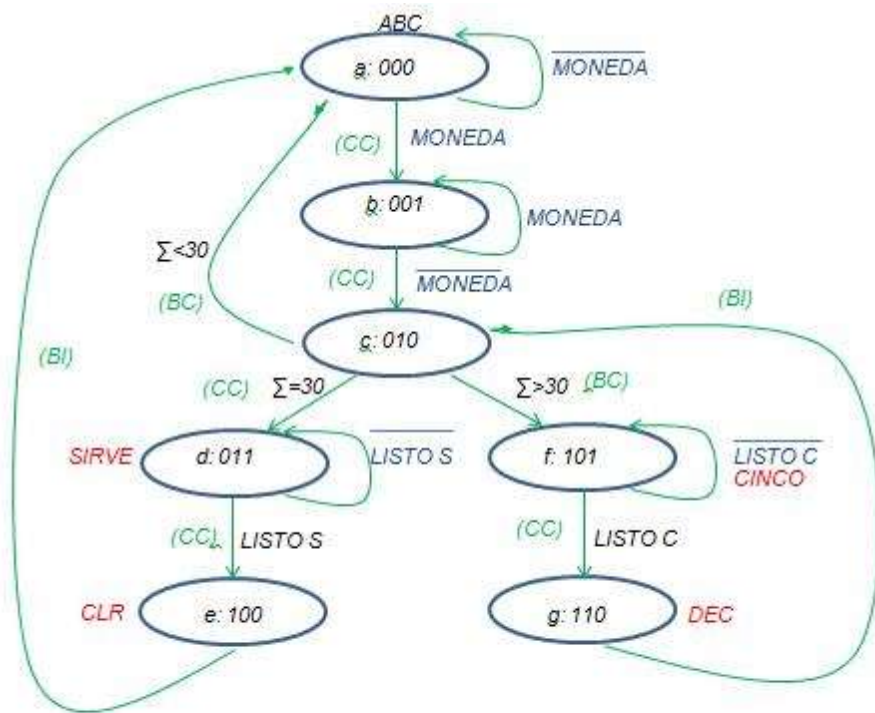
- CC, CI: cuenta condicional e incondicional
- BC, BI: brinco condicional e incondicional

## PASOS DE DISEÑO

- 1) Asignar códigos y comandos al autómata
- 2) Hacer el mapa M.E.P.
- 3) Hacer el mapa M.E.S. En caso de que se tengan brinco indicar la dirección respectiva. Indicar también las variables que indican los cambios
- 4) Hacer mapa de control de modos para ENP y LOAD
- 5) Hacer mapas de cargas en paralelo y de salida
- 6) Implementar el controlador

## SOLUCIÓN AL PROBLEMA - AUTÓMATA

- 1) Asignar códigos y comandos al autómata



a, b, c, d, e, f, g: Estados presentes  
 ABC: Código de los estados

2) Hacer el mapa M.E.P.

3) Hacer el mapa M.E.S. En caso de que se tengan brinco indicar la dirección respectiva. Indicar también las variables que indican los cambios

	AB				
	00	01	11	10	
C	0	a	c	g	e
	1	b	d	*	f

M.E.P

	AB				
	00	01	11	10	
C	0	CC, MON	$\Sigma<, BC \rightarrow a$ $\Sigma=, CC$ $\Sigma>, BC \rightarrow f$	BI $\rightarrow c$	BI $\rightarrow a$
	1	CC, MON'	CC, LISTO S	*	CC, LISTO

M.E.S

4) Hacer mapa de control de modos para ENP y LOAD

		AB			
		00	01	11	10
C	0	MON	1	*	*
	1	MON'	LISTO S	*	LISTO C

ENP

		AB			
		00	01	11	10
C	0	0	$\Sigma ='$	1	1
	1	0	0	*	0

LOAD

Estado a:

MON -> CC ENP, LOAD= 1 0

MON' -> HLT ENP, LOAD= 0 0

deducción para los mapas: ENP: MON, LOAD: 0

Estado c:

$\Sigma <$  -> BRINCO ENP, LOAD= \* 1

$\Sigma =$  -> CUENTA ENP, LOAD= 1 0

$\Sigma >$  -> BRINCO ENP, LOAD= \* 1

deducción para los mapas: ENP:1, LOAD:  $\Sigma ='$

Estado g:

BRINCO ENP, LOAD= \* 1

5) Hacer mapas de cargas en paralelo A, B, C (brincos) y de salida

		<i>AB</i>			
		00	01	11	10
<i>C</i>	0	*	$\Sigma >$	0	0
	1	*	*	*	*

$$A = A'(\Sigma >)$$

		<i>AB</i>			
		00	01	11	10
<i>C</i>	0	*	0	1	0
	1	*	*	*	*

$$B = AB$$

		<i>AB</i>			
		00	01	11	10
<i>C</i>	0	*	$\Sigma >$	0	0
	1	*	*	*	*

$$C = A'(\Sigma >)$$

Estado c:

$\Sigma < \rightarrow a$ , ABC = 000

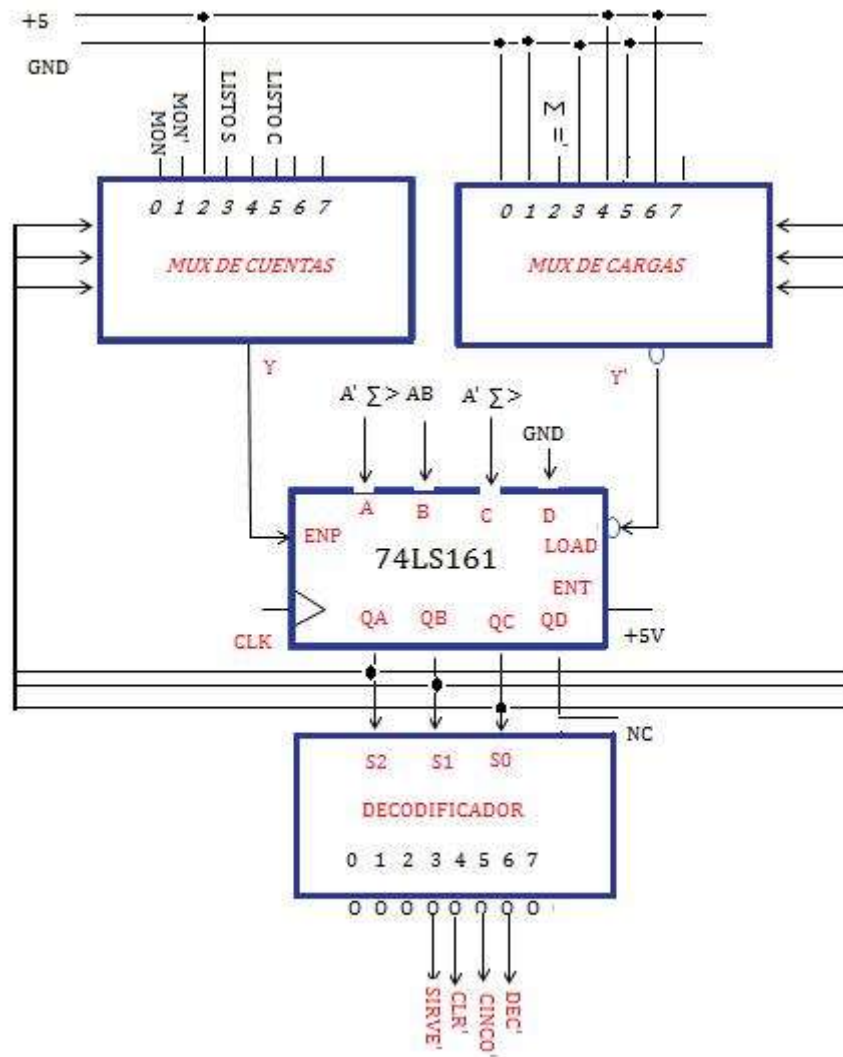
$\Sigma > \rightarrow f$ , ABC = 101

deducción:  $A = \Sigma >$ ,  $B = 0$ ,  $C = \Sigma >$

		<i>AB</i>			
		00	01	11	10
<i>C</i>	0			DEC	CLR
	1		SIRVE		CINCO

SALIDAS

6) Implementar el controlador



## CURSO: ELECTRÓNICA DIGITAL

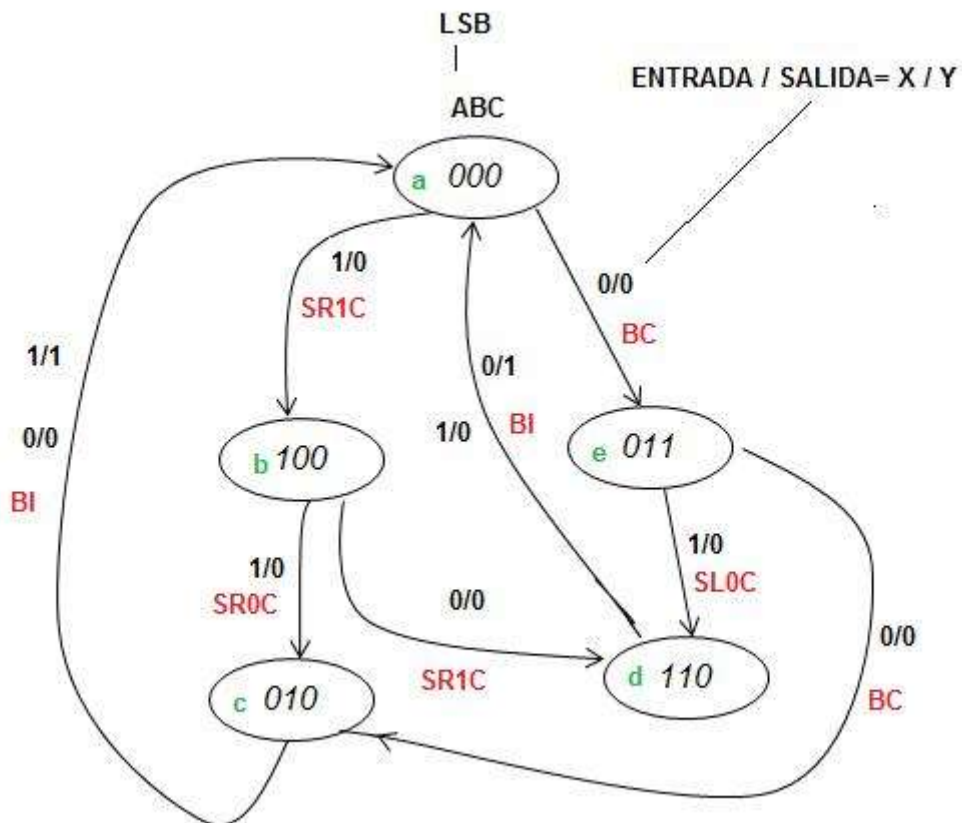
### UNIDAD 4: CONTROLADORES DIGITALES - SIMULACIÓN

En esta unidad vamos hacer dos simulaciones en Proteus para consolidar los conceptos teóricos sobre controladores digitales utilizando como controlador el registro de corrimiento 74194.

#### CONTROLADOR DIGITAL CON REGISTRO SR 74194

Simule el controlador digital que implemente el siguiente autómata utilizando como controlador el circuito integrado SR 74194.

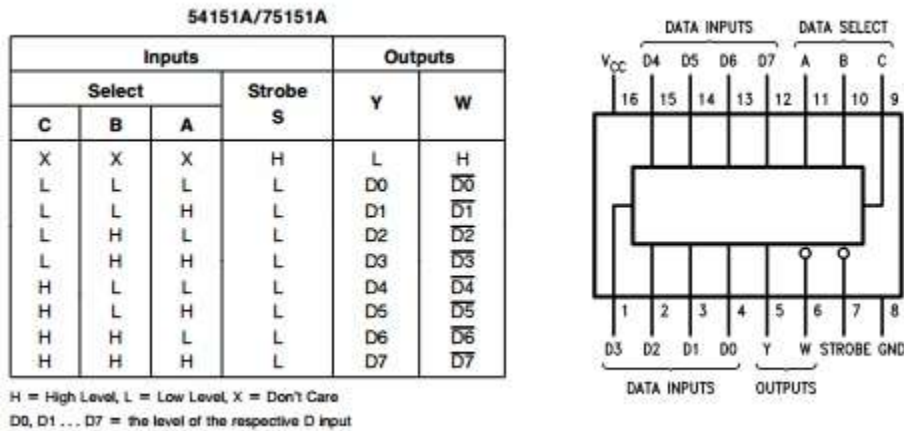
#### AUTÓMATA



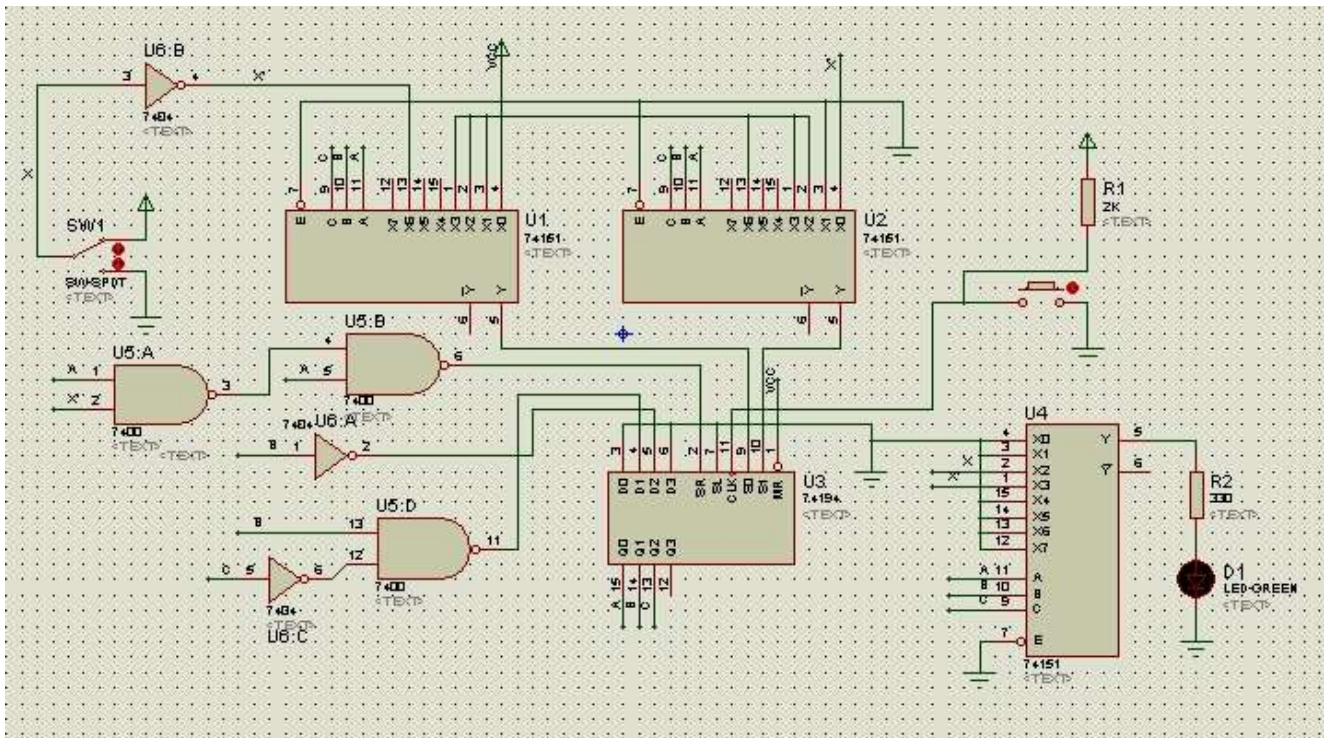


## METODOLOGÍA

1. Haga el Mapa de estados presente MEP
2. Haga el Mapa de estados siguiente MES
3. Haga el Mapa para: S1, S0, R, L
4. Haga los Mapas para los brincos: A, B, C
5. Haga el Mapa de salidas
6. Simule el siguiente circuito que es la solución del autómata
7. Compruebe el funcionamiento del autómata.



## CIRCUITO



## CURSO: ELECTRÓNICA DIGITAL

### UNIDAD 4: CONTROLADORES DIGITALES - LABORATORIO

En esta unidad vamos hacer la implementación del controlador digital usando el registro de corrimiento 74194.

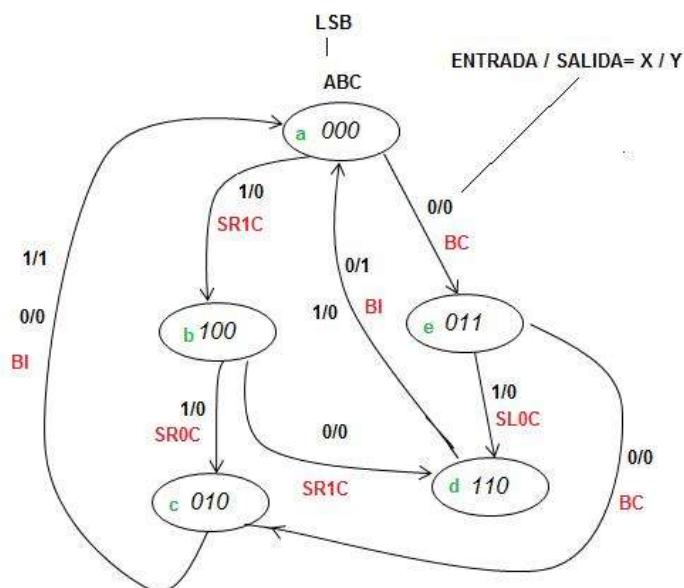
#### EQUIPO Y MATERIAL NECESARIO

- Fuente de 5V
- Protoboard
- Circuitos integrados: 555, 3x74LS151, 74LS194, 74LS00, 74LS04
- Switch SP2T
- Un pulsador
- 4 LEDs
- Resistencias a 1/4W de: 4x330Ω
- Conectores

#### CONTROLADOR DIGITAL CON REGISTRO SR 74194

Implemente el controlador digital del autómata de la figura, utilizando como controlador el circuito integrado SR 74194.

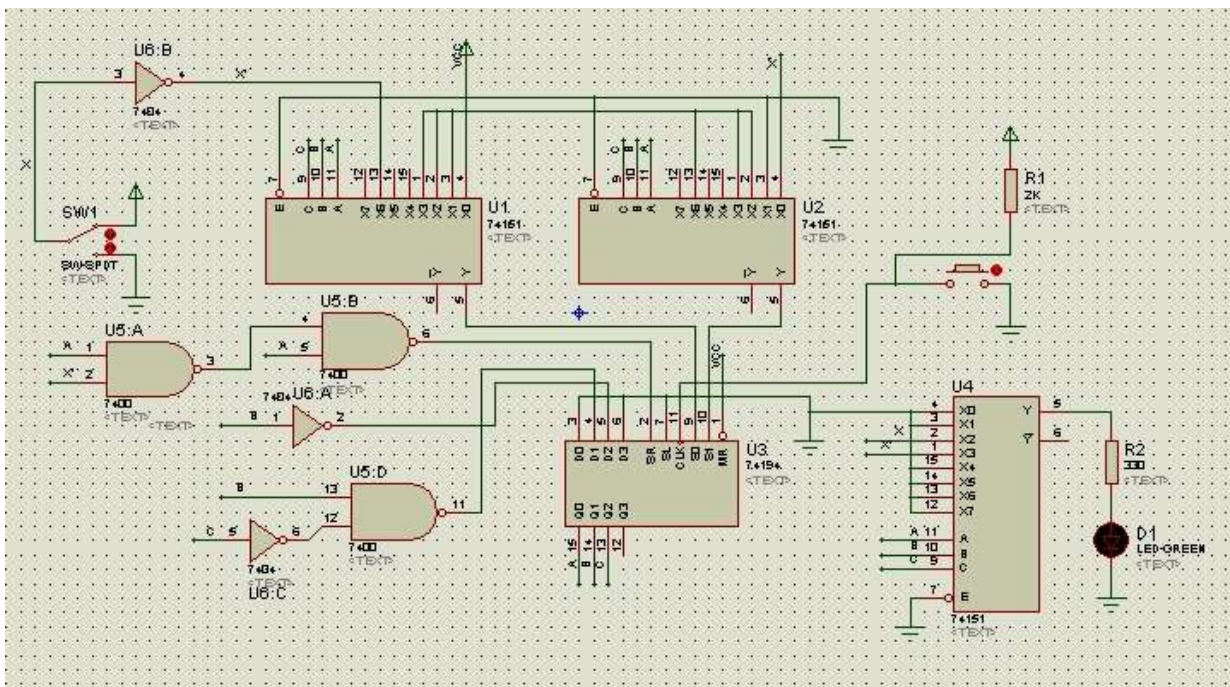
##### 1. AUTÓMATA



## 2. METODOLOGÍA

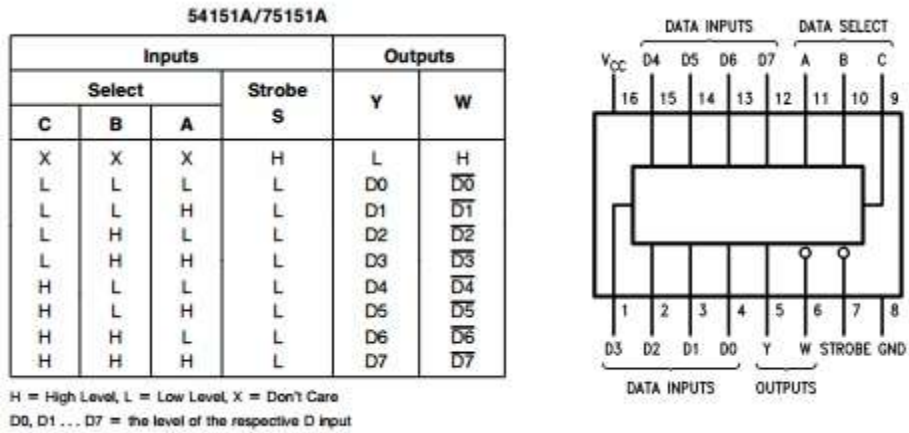
1. Haga el Mapa de estados presente MEP
2. Haga el Mapa de estados siguiente MES
3. Haga el Mapa para: S1, S0, R, L
4. Haga los Mapas para los brincos: A, B, C
5. Haga el Mapa de salidas
6. Simule el siguiente circuito que es la solución del autómata
7. Compruebe el funcionamiento del autómata.

## 3. CIRCUITO



# CIRCUITOS INTEGRADOS

## 74151 MUX



## 74194 S-R

