

24 201



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES  
" ARAGON "

**MANUAL DE PRACTICAS DE DISEÑO DE  
CONTROLADORES DIGITALES**

T E S I S  
QUE PARA OBTENER EL TITULO DE:  
INGENIERO EN COMPUTACION  
P R E S E N T A :  
JESUS MANUEL RAMOS ALVAREZ

JULIO

**FALLA DE ORIGEN**

1990



Universidad Nacional  
Autónoma de México



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

INTRODUCCION.....	I
CAPITULO I	
"CIRCUITOS COMBINACIONALES Y SECUENCIALES MSI".....	1
I.1      Introducción.....	2
I.1.1. Multiplexores.....	3
I.1.2. Demultiplexores.....	6
I.1.3. Decodificadores.....	7
I.1.4. Codificadores.....	10
I.1.5. Decodificador/Manejador de des - pliegues.....	13
I.1.6. Comparadores.....	16
I.2      Circuitos Secuenciales MSI.....	18
I.2.1. Registros.....	19
I.2.2. Contadores.....	24
I.2.3. Memorias.....	27
CAPITULO II	
"CONTROLADORES TRADICIONALES" .....	33
II.1     Introducción.....	34
II.2     Controladores cableados.....	35
II.2.1. Diseño de Flíp-Flops.....	36

II.2.2. Diseño con Registros de Corri- miento.....	41
II.2.3. Diseño de contadores.....	47

CAPITULO III

"CONTROLADORES MICROPROGRAMADOS".....	53
III.1 Controladores Microprogramados sencillos..	54
III.2 Controladores con número fijo de instrucc- ciones.....	59
III.3 Controladores que manejan subrutina.....	62

CAPITULO IV

"DISEÑO CON CARTAS ASM".....	67
IV.1 Introducción.....	68
IV.2 Definición.....	69
IV.3 Reglas de diseño de una carta ASM.....	69
IV.4 Diseño con cartas ASM.....	77

CAPITULO V

"INTERFASES".....	80
V.1 Introducción.....	81
V.2 Conmutadores y teclados.....	82
V.3 Dispositivos basados en Led's.....	87
V.3.1. Fototransistores.....	89
V.3.2. Opto-aisladores.....	90

V.4	Entrada/Salida analógica.....	94
	V.4.1. Requerimientos de la interfase....	94
	V.4.2. Conversión analógico/digital.....	98
	V.4.3. Conversión de datos.....	103
	V.4.4. Transductores.....	108
	V.4.5. Actuadores.....	111
CAPITULO VI		
	"PAQUETE DE PRACTICAS" .....	117
	PRACTICA NUMERO 1.....	118
	PRACTICA NUMERO 2.....	127
	PRACTICA NUMERO 3.....	136
	PRACTICA NUMERO 4.....	146
	PRACTICA NUMERO 5.....	152
	PRACTICA NUMERO 6.....	157
	PRACTICA NUMERO 7.....	165
	CONCLUSIONES.....	174
	BIBLIOGRAFIA.....	175
	INDICE DE TABLAS Y FIGURAS.....	176
	GLOSARIO DE TERMINOS TECNICOS.....	182

---

## I N T R O D U C C I O N

---

Esta época se caracteriza por una vertiginosa carrera - hacia la automatización y con ello la electrónica ha sufrido impresionantes avances encaminándose visiblemente a la digitalización de los datos proporcionados en el mundo analógico.

Dentro de este contexto es de esperar que las instituciones educativas tengan que realizar grandes esfuerzos para permanecer actualizados e incorporar en sus cátedras los nuevos métodos, herramientas y elementos desarrollados para controlar de manera más eficiente los procesos cotidianos. Claro ejemplo de esto último es la acelerada modernización de las computadoras, - que ya se aplican en situaciones antes inimaginables.

Una parte muy importante de los sistemas digitales a los que nos hemos referido, son los controladores digitales. Dispositivos que nos permitirán mantener el gobierno sobre las interacciones existentes entre un medio determinado y ciertos elementos que actuarán sobre este medio. Un controlador digital actúa en función de ciertas señales de entrada, emitiendo ciertas señales de salida.

El objetivo de este trabajo de titulación, es proporcionar a los alumnos de carreras afines con la electrónica digital,-

un compendio de información y una propuesta que les permita conocer diferentes tipos de controladores y aprender a diseñarlos. - Es decir, se trata de que el alumno conozca los elementos y algoritmos utilizados para la construcción de estos controladores, y parte fundamental de este aprendizaje es la aplicación práctica de los elementos teóricos proporcionados.

En el capítulo I "Circuitos combinacionales y secuenciales MSI". se habla de los elementos fundamentales que constituirán a nuestros controladores, siendo muy importante conocer bien el funcionamiento de cada uno de ellos.

En los capítulos II: "Controladores tradicionales" y III: "Controladores Microprogramados", se trata ya con el proceso de diseño de diferentes arquitecturas de controladores. En el capítulo IV: "Diseño con Cartas ASM", se presenta una herramienta de software muy útil para presentar algoritmos de solución a un problema determinado antes de darle forma física con un controlador.

El capítulo V: "Interfases", trata de ciertos dispositivos auxiliares y necesarios para darle aplicación real a un controlador digital, ya que como comentamos anteriormente, el ambiente que nos rodea es esencialmente analógico.

Por último el capítulo VI: "Paquete de Prácticas" es un

grupo de prácticas planteadas con el fin de llevar al alumno - gradualmente al dominio de estos controladores. Se plantea un problema y se le da solución. Las cuatro primeras prácticas - están pensadas para solucionarse en dos sesiones de laborato - rio (4 horas) y las tres últimas en 4 sesiones cada una (8 ho - ras); diez semanas en total.

Existe gran flexibilidad tanto en los planteamientos -- como en las soluciones con el fin de que sean manejados a cri - terio del profesor.

Todos los que hemos participado en la elaboración de -- este trabajo consideramos que el mejor premio será su utilidad.



---

---

CAPITULO I

"CIRCUITOS COMBINACIONALES Y SECUENCIALES MSI"

---

---

Los circuitos integrados digitales del grupo de Media--na Escala de Integración (MSI), tienen su nivel funcional más--básico como circuitos combinacionales. Este tipo de circuitos--se llama así porque, considerándolos como una caja negra con en--tradas y salidas; una salida cualquiera es la respuesta a una -combinación de las entradas presentes en ese momento.

Como un detalle más de los circuitos combinacionales, -podemos agregar que, dada una serie de entradas y sus respecti--vas salidas, cada que se presenten las mismas entradas, se ob--tendrán las mismas salidas. Esta relación, aunque parece ser -evidente y elemental, no se debe tomar tan a la ligera pues -otro gran grupo de circuitos integrados digitales, los secuen--ciales, no cumplen con ella.

Los combinacionales MSI están formados por los arreglos de compuertas que ya conocemos, sólo que ahora los encontramos--encapsulados en un solo chip. Trataremos con 4 grupos básicos: Multiplexor, Demultiplexor, Codificador-Decodificador, Sumado -res y Comparadores.

Las aplicaciones de estos circuitos son muy variadas, -aunque, en general, no se puede decir que pueden formar un sis--

tema digital completo ya que faltarán circuitos secuenciales - (los cuales estudiaremos en la siguiente sección). Sin embargo, los circuitos combinatoriales son una parte necesaria e imprescindible en la mayoría de los Sistemas Digitales y nos permiten ya, establecer la estructura general que presentará un diseño.

---

### I.1.1

### MULTIPLEXORES

---

Los Multiplexores son elementos digitales MSI considerados como SELECTORES DE DATOS.

La Multiplexión significa transmitir varias unidades de información en pocas líneas o canales, en el caso más común es transmitir sobre una sola línea de salida.

Podemos observar que el concepto de Selector de Datos es congruente con lo explicado en el párrafo anterior, pues un Multiplexor escoge un dato de sus líneas de entrada y lo envía a la salida única. La elección del dato de entrada que se presentará a la salida se realiza por medio de un conjunto de líneas de selección.

En un multiplexor podemos encontrar  $2^n$  líneas de entrada

"CIRCUITOS COMBINACIONALES Y SECUENCIALES MSI".

da y 'n' líneas de selección, cuyas combinaciones de bit determinan cual entrada, entre todas, va a ser presentada a la salida. Una representación común del Multiplexor se muestra en la siguiente figura:

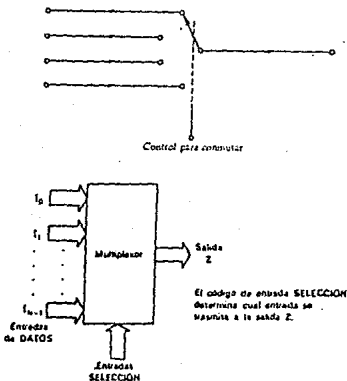


Figura 1.1 Multiplexor como selector de datos.

La dimensión de un Multiplexor se especifica comúnmente por su número de entrada ( $2^n$ ) y sus salidas, abreviándose - su nombre como MUX.

Como ejemplo, se muestra en la Figura 1.2 un MUX 4:1 y su tabla de verdad.

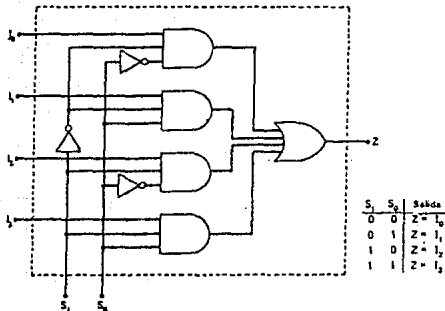


Figura 1.2 Multiplexor 4:1

Muchos Multiplexores comerciales son fabricados con una patilla de HABILITACION para controlar el chip. Esta entrada de habilitación (conocida también como Chip Select CS o Estroboscopio), al "conectar" o "desconectar" el CI, permite realizar expansiones de multiplexores para armar sistemas de más entradas o salidas.

El Multiplexor es un circuito MSI muy útil y se aplica

en gran variedad de sistemas digitales. Se usa para conectar - dos o más fuentes a un sólo destino en sistemas de bus común, además se usan como controladores de habilitaciones en sistemas digitales, y para construcciones MSI de una función booleana.

---

I.1.2

DEMULTIPLEXORES

---

Si el Multiplexor es considerado como selector de datos, el Demultiplexor deberá ser considerado como DISTRIBUIDOR DE DATOS.

Lo anterior significa que el Demultiplexor realiza la función recíproca de un Multiplexor; es decir, a un Multiplexor llegan varias líneas de entrada de datos para producir UNA SÓLA SALIDA, entonces, a un Demultiplexor llega una sola entrada de datos para ser enviado por alguna de VARIAS SALIDAS.

En la Figura 1.3 se muestra la representación común de un Demultiplexor como distribuidor de datos. Nótese que ahora se tiene una sola entrada "n" líneas de selección y  $2^n$  salidas. El dato de entrada estará presente sólo en la línea seleccionada.

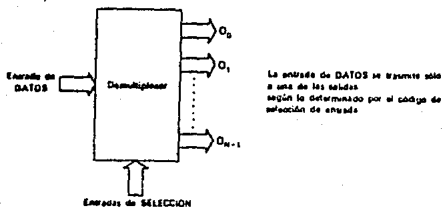


Figura 1.3 Demultiplexor como Distribuidor de Datos.

El Demultiplexor es, en realidad, un dispositivo poco usado y sus funciones son realizadas por otro componente llamado DECODIFICADOR, el cual es capaz de simular un Demultiplexor y a decir verdad, la diferencia entre ellos es sólo de filosofía de funcionamiento, es decir, meramente conceptual; aunque operativamente presentan los mismos resultados.

---

I.1.3.

DECODIFICADORES

---

Un Decodificador es un dispositivo capaz de "habili-

tar" una sola línea de salida de entre  $2^m$  salidas, de acuerdo a un código de entrada. Esto significa que un código binario se "decodifica" haciendo corresponder una sola salida a cada código de entrada. Ver Fig. 1.4.

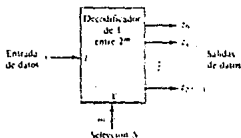
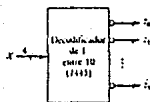


Figura 1.4 Diagrama General de un Decodificador.

Una de las funciones más importantes del Decodificador es la de "direccionar" a otros elementos, escogiéndolos de acuerdo a su código de entrada. Existen Decodificadores 1 entre 8, 1 entre 10 y entre 16, dentro de los más usados.

La Figura 1.5 muestra un Decodificador 1 entre 10, llamado Decodificador BCD a Decimal.





B	C	U	A	$O_0$	$O_1$	$O_2$	$O_3$	$O_4$	$O_5$	$O_6$	$O_7$	$O_8$	$O_9$
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1
1	0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0

(Todas las Salidas = 0)

Figura 1.5 Decodificador BCD Decimal.

Otra función muy común del Decodificador es como distribuidor, es decir, como un Demultiplexor. Esto se logra - utilizando la entrada de código como línea de dirección y la - habilitación del chip, como entrada de datos. El funcionamiento es el mismo que el de un Demultiplexor, como mencionamos anteriormente, la diferencia es sólo conceptual: en un Demultiplexor, es el dato el que sale o alguna de las líneas elegidas; en un Decodificador, se habilita una línea por medio del código y se simula el dato habilitando o deshabilitando al chip.

La siguiente Figura muestra un arreglo Multiplexor-De-

## "CIRCUITOS COMBINACIONALES Y SECUENCIALES MSI".

multiplexor, con un Multiplexor 16 a 1 y un Decodificador 1 en tre 16.

Este circuito permite llevar datos de 16 diferentes - fuentes a 16 distintos destinos, por medio de una sola línea - de transmisión.

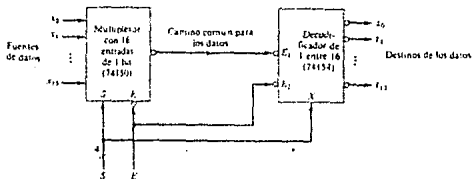


Fig. 1.6 Arreglo Mux - Demux para transmisión de datos.

---

### I.1.4

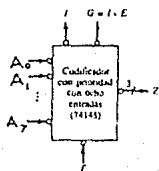
### CODIFICADORES

---

Un Codificador es un circuito que identifica una línea activa de entrada y le asigna un código binario a la salida.

Un ejemplo sencillo es un Codificador OCTAL a BINARIO, el cual cuenta con 8 líneas de entrada y 3 líneas de salida, - de tal manera que cuando una de las ocho líneas de entrada se activa, se produce su correspondiente código en tres bits.

En la siguiente Figura se presenta un Codificador Octal a Binario y su correspondiente tabla de verdad. Note que - cuando se activa  $A_0$ , la salida es 000 cuando se activa  $A_3$ , la salida es 101; es decir, existe correspondencia directa entre el número de línea activa y el código binario de salida.



$A_0$	$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	$A_7$	$O_2$	$O_1$	$O_0$
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

{ Otros casos de entrada no se permiten. }

Figura 1.7 Codificador Octal-Binario y Tabla de Verdad.

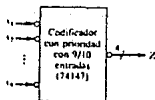
Es muy importante la consideración de SOLO UNA línea -

activada en un tiempo dado para que se le de el código correspondiente.

Como no hay manera inmediata de impedir que se activen simultáneamente varias líneas de entrada, generalmente los codificadores se diseñan para responder a una sola línea, de todas las que se encuentren activas en un momento dado. Esta selección se realiza asignando prioridades a las entradas, de tal manera que el codificador responderá sólo a la línea con más alta prioridad.

El criterio más común y sencillo de prioridad, es el de asignar un valor más alto. Así, si se activan al mismo tiempo las entradas  $X_4$  y  $X_6$ , la salida corresponderá al código de  $X_6$ , el cual tiene prioridad sobre  $X_4$ . Este criterio es válido para cualquier número de entradas activas simultáneamente.

La Figura 1.8 presenta a un Codificador DECIMAL a BCD con prioridad y salidas invertidas, con su tabla de verdad. Note las condiciones de "no importa" en la tabla de verdad.



ENTRADAS									SALIDAS			
1	2	3	4	5	6	7	8	9	0	C	B	A
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	L	H	H	H	H	H	L	L	L
X	X	X	L	H	H	H	H	H	H	L	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

Figura 1.8 Codificador Decimal - BCD

I.1.5.

DECODIFICADOR / MANEJADOR DE DESPLIEGUE

Una de las necesidades comunes dentro de los Sistemas Digitales es la de desplegar información de una forma entendible y conveniente para el usuario de tal sistema. Los indicadores luminosos son una solución muy socorrida, y los indicadores que pueden desplegar números y/o letras se han hecho muy populares.

Dentro del ámbito de experimentación en laboratorio, el

despliegue de siete segmentos a base de LED's es quizá el más-utilizado. Existen en el mercado configuraciones de un sólo - despliegue, dobles, de ánodo común, de cátodo común, etc. La - Figura 1.9 muestra un despliegue de 7 segmentos con ánodo co - mún y los caracteres generados usualmente.

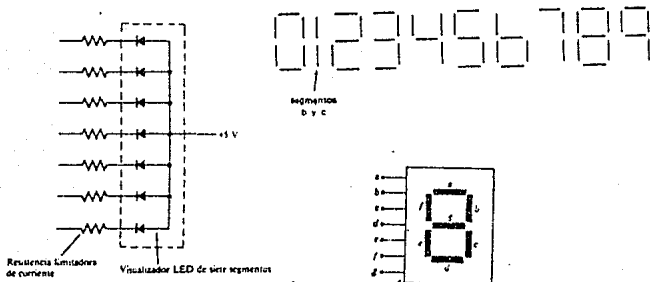


Figura 1.9 Despliegue de 7 segmentos de ánodo común.

En la figura anterior se observa que el encendido de - cada segmento está determinado por la circulación de corriente desde +Vcc, a través del LED o LED's del segmento y por una re sistencia, a tierra.

Además, el hecho de generar un carácter, implica "activar" varios caminos a tierra, y si consideramos que trabajamos con códigos binarios, necesitaremos un circuito que "transforme" el código usado en el Sistema Digital, al código correspondiente en 7 segmentos. A este circuito se le da el nombre de - Decodificador / Manejador de 7 segmentos.

El circuito de la siguiente figura es un típico dispositivo TTL 7447 Decodificador /Manejador BCD a 7 segmentos; el cual, como sugiere su nombre, acepta un código de entrada (BCD) y lo transforma a su correspondiente código de 7 segmentos, proporcionando los adecuados caminos a la corriente que proviene del ánodo común, para desplegar el número en forma decimal.

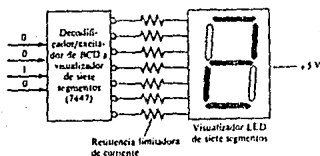


Figura 1.10 Decodificador/Manejador BCD a 7 segmentos.

Muchos Diseños Digitales exigirán la comparación de dos o más datos en el sentido de poder determinar cual es mayor, o si son iguales. Con esta finalidad se han construido circuitos MSI que realizan esta función activando una de sus tres salidas ( $A=B$ ,  $A<B$ ,  $A>B$ ), para indicar el resultado de la comparación.

La siguiente figura muestra en bloque, a un típico comparador de 4 bits TTL, el 7485.

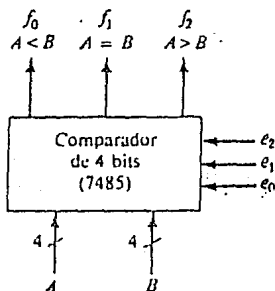


Figura 1.11 Comparador 7485 de 4 bits.



Note las entradas  $e_2, e_1, e_0$ , estas son usadas para -- construir comparadores más grandes y funcionan como "acarreo-de comparación" informando al comparador actual el resultado de la anterior comparación, de tal manera que si los datos actuales son iguales, el resultado dependerá ahora de la comparación anterior. Para mayor claridad de lo anterior, se presenta la Figura 1.12 que muestra a un comparador de 12 bits construido con 3 comparadores de 4 bits 7485.

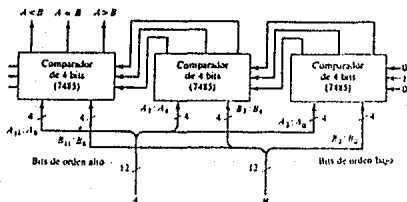


Figura 1.12 Expansión de Comparadores.

En este capítulo, analizaremos los circuitos MSI que nos permiten manipular datos "recordando" su estado o valor anterior. Es decir, circuitos de almacenamiento de datos y circuitos que actúan de acuerdo a ese dato almacenado.

En cursos anteriores seguramente se trabajó con Flip-Flops, circuitos SSI capaces de almacenar datos. Con estos circuitos se pudieron diseñar sistemas de cuenta, o lo que genéricamente llamamos CONTADORES. Estos circuitos son los que nos permiten generar diferentes estados en un sistema digital, o sea, siguen cierta SECUENCIA predeterminada que cambia con cada pulso de reloj, siempre y cuando el estado presente cumpla con ciertos requisitos que habiliten el paso a un estado futuro. -- Cuando diseñamos un sistema con estas bases, estaremos creando un Sistema Secuencial.

Dentro del campo MSI, los Flip-Flops se agrupan formando diferentes tipos de REGISTROS, y además, se nos provee de buena cantidad de contadores que facilitan bastante nuestra tarea de diseño. Estos serán pues, los circuitos a los que haremos referencia en esta parte de nuestro manual.

Sin embargo, es conveniente analizar un circuito LSI -

muy importante y vital en nuestro campo: la memoria RAM. Esto con el fin de dejar completo el panorama de los Circuitos Secuenciales, dentro de los cuales la RAM ocupa un destacado lugar.

---

### I.2.1

### REGISTROS

---

El nivel MSI es generoso en el suministro de diferentes tipos de registros que como ya sabemos, consisten de agrupamientos de Flip-Flops que permiten el almacenamiento temporal de datos y su transferencia a los demás elementos de determinado sistema digital.

Dentro de la gama existente de registros se puede establecer una clasificación basada en sus tipos de entrada y salida de datos, que en esencia pertenecen a dos grupos ya muy conocidos de transferencia de datos: transferencia en serie y transferencia en paralelo.

La transferencia en paralelo de datos no implica ninguna dificultad de comprensión si analizamos la siguiente figura, en la cual representamos a un registro paralelo típico con ocho entradas y ocho salidas.

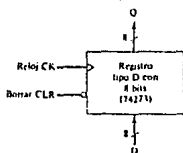


Figura 1.13 Registro Paralelo 74273.

Facilmente se observa que las ocho entradas (representadas por  $Q_m$ ) presentes en cierto momento  $T$ , se harán manifiestas a la salida, simultáneamente, cuándo pase un ciclo de reloj. Esto significa que se trata de un circuito con entrada de paralelo a 8 bits y salida en paralelo a los mismos 8 bits.

La transferencia de datos en serie, se entiende que se realiza Bit por Bit. En el caso particular de un registro en serie de 8 bits, se puede contemplar como un arreglo de 8 Flip-Flops conectados en cascada (la salida de una, a la entrada del siguiente), con una sola entrada y una sola salida. En este tipo de registro, como en el mostrado en la siguiente figura, cada bit de entrada tiene que esperar ocho ciclos de reloj para llegar a la salida.



Fig. 1.14 Registro Serie 7491

Cabe comentar que aunque en el circuito anterior habian aparentemente dos entradas, la compuerta interna del circuito las convierte en una sola entrada correspondiente a la función NAND de ambas. Así, se cumple el concepto de una sola entrada y una sola salida para una transferencia de datos en serie.

A la transferencia en serie también se le nombra como-- Desplazamiento, y este desplazamiento puede realizarse desde-- un dispositivo o hacia él. Además se pueden hacer combinaciones con entradas o salidas en paralelo, y desplazamientos hacia un lado o hacia otro.

Un registro de desplazamiento (Shift Register) con en -

trada en serie y salidas en paralelo, es el 74164 mostrado a continuación que, siendo de 8 bits se comporta de manera semejante al 7491 visto anteriormente, con la característica distintiva de tener presentes fuera del circuito, todas las salidas de los Flip-Flops. Es decir, se presenta en todo tiempo el estado general de los F.F. de desplazamiento.

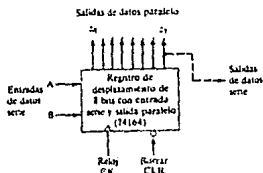


Figura 1.15 Registro SERIE-PARALELO 74164

El TTL 74165 es un Registro de Desplazamiento con 8 bits de entrada en paralelo y salida en serie.

Es utilizado regularmente para convertir datos de formato paralelo a serie, ya que cuenta con la capacidad de "cargar" (load) un dato de ocho bits y después realizar el desplazamiento de tales datos. El desplazamiento se realizará "rellenando" los espacios generados con datos introducidos por una entrada para

datos en serie.

La siguiente figura muestra la configuración de pines -  
de este registro.

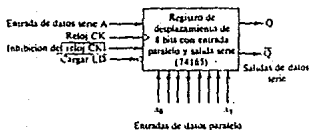


Figura 1.16 Registro de Corrimiento 74165

Todas las características de los Registros anteriores se encuentran concentradas en un Registro Universal de 4 bits -cuyas líneas de habilitación permiten que funcione como Registro Paralelo, Registro Serie, y sus combinaciones vistas; además permite realizar desplazamientos de derecha a izquierda y de izquierda a derecha. A continuación el circuito.

## "CIRCUITOS COMBINACIONALES Y SECUENCIALES MSI".

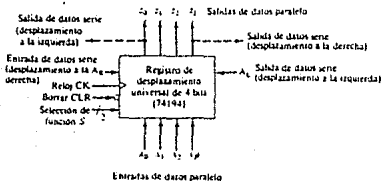


Figura 1.17 Registro Universal 74194

---

### I.2.2.

### CONTADORES

---

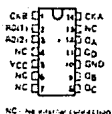
Los Contadores son circuitos basados en Flip-Flops y que permiten el paso de un estado a otro.

Cada CHIP correspondiente a un Contador, contiene normalmente un arreglo de Flip-Flops en cascada y lógica auxiliar que permitirá controlar las acciones de cuenta del circuito, -



de tal manera que se pueda contar en forma ascendente (UP) o-- descendente (DOWN). Además la lógica controlará el rango de la cuenta (desde donde hasta que número), determinando el Módulo- del Contador. De acuerdo a este criterio encontraremos Contado- res Década (cuenta hasta diez) o BCD, Contadores Módulo 16 -- (cuenta hasta 15), etc.

La siguiente figura muestra un contador sencillo de -- cuatro bits.



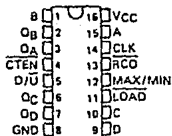
COUNT	OUTPUT			
	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H
10	H	L	H	L
11	H	L	H	H
12	H	H	L	L
13	H	H	L	H
14	H	H	H	L
15	H	H	H	H

Figura 1.18 Contador 7493 de 4 bits.

Un Contador más interesante es el 74190, el cual puede contar en orden ascendente o descendente, dependiendo del nivel lógico de su entrada U/D, además la cuenta puede iniciarse desde un número que, asincrónicamente puede ser almacenado en el

"CIRCUITOS COMBINACIONALES Y SECUENCIALES MSI".

contador al habilitar la entrada de Load (LD). A continuación se muestra la configuración de este circuito junto con su tabla de funciones.



74190

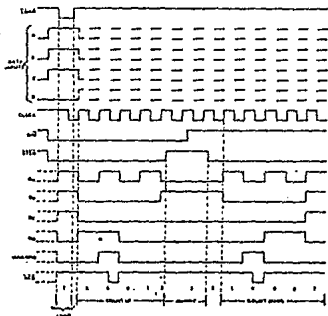


Figura 1.19 Contador BCD UP/DOWN

Los Contadores que pueden ser "cargados" con datos (como el anterior 74190), son de singular utilidad en la construcción de sistemas digitales regularmente complejos, ya que generan estados lógicos con la facilidad de 'programar' saltos de estados (del estado dos al cinco, por ejemplo), basados precisamente en esa capacidad de carga.

En resumen, los Contadores pueden ser considerados -- (aparte de su función normal o nominal de cuenta de eventos),

como generadores de estados binarios de un sistema digital.

Si usamos como ejemplo un sencillo sistema de multiplexaje, como el mostrado a continuación, podemos observar que se necesita generar tantos estados como entradas tenga el multiplexor, y para esto podemos utilizar un contador módulo- $2^n$ , donde,  $2^n$  son las entradas al Multiplexor.

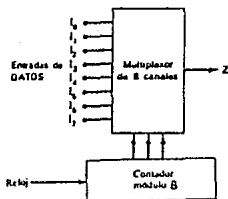


Figura 1.20 Circuito Multiplexor con Contador.

El almacenamiento temporal de datos es una de las características más importantes de los sistemas secuenciales de

## "CIRCUITOS COMBINACIONALES Y SECUENCIALES MSI".

sarrollados, por permitir conservar valores o datos nuevos o de estados anteriores y tenerlos a mano para cuando se requieran.

Para realizar con eficacia este almacenamiento, se requiere de un circuito LSI llamado "Memoria" y existen numerosas configuraciones distintas de ellas. Para comenzar, nos enfocaremos a las memorias que pueden ser escritas y leídas cuantas veces sea necesario y llamaremos por su nombre convencional RAM.

Las memorias RAM (Random Access Memory) o de acceso aleatorio, contienen cierta cantidad de localidades que pueden ser accesadas para ser leídas o para escribir un dato en ella. El acceso se logra por medio de líneas de selección o más comúnmente llamadas, líneas de dirección.

A un conjunto de líneas con un mismo fin específico se la llama genéricamente "bus" y una memoria RAM tiene 3 buses, a saber: bus de datos, bus de direcciones y bus de control.

El bus de datos lleva o trae los datos de cada localidad de memoria seleccionada mediante el bus de direcciones, y el bus de control habilita la lectura o escritura de la memoria, además de activar o desactivar el circuito en sus conjuntos.

Las memorias RAM se pueden clasificar de acuerdo a diferentes conceptos, entre ellos:

- Por su funcionamiento general: RAM dinámica (DRAM)  
RAM estática (SRAM)

- Por su tecnología de fabricación: CMOS  
NMOS  
ECL  
TTL  
etc.

- Por su capacidad de almacenamiento: 256 x 1 bit  
1K x 4 bits  
64K x 1 bit  
etc.

Se recomienda al lector interesado en este tema, consultar los manuales que los fabricantes de memorias editan, para tener un panorama más amplio de los tipos y capacidades de memorias existentes en el mercado. Es conveniente poner atención en los niveles de voltaje asociados con los estados lógicos, velocidades de acceso a un dato y potencia disipada por el dispositivo.

La siguiente figura muestra un diagrama a bloques de -

"CIRCUITOS COMBINACIONALES Y SECUENCIALES MSI".

una memoria, mostrando sus diferentes buses. Nótese que la RAM se nombra por la cantidad de localidades (Registros) que contiene y el número de bits de que consta cada localidad.

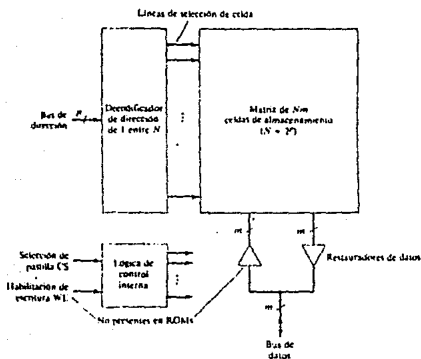


Figura 1.21 Memoria RAM a Bloques  $N \times m$

El otro grupo importante de memorias es el conocido como memorias ROM (Read Only Memory) las cuales son usadas para almacenar datos que solo serán leídos, es decir, que no serán modificados durante su operación.

La principal clasificación de memorias ROM es de acuerdo con sus características de programación:

- ROM: Memoria cuyos datos serán grabados de fábrica y sobre la cual no existe posibilidad de modificación intencional de los datos almacenados.
- PROM: Memoria cuyos datos son grabados por el usuario, pero que no existe posibilidad de modificarlos una vez que se han almacenado.
- EPROM: Memoria cuyos datos son grabados por el usuario, pero si existe posibilidad de borrar su contenido por medios ópticos (luz ultravioleta) para modificar los datos.
- EEPROM: Memoria cuyos datos son grabados por el usuario, pero si existe posibilidad de borrar su contenido por medios eléctricos para modificar los datos.

Ninguna de las memorias mencionadas pierde sus datos una vez que ha sido grabada, las tres últimas se venden "vacías", es decir con todas sus localidades con el valor lógico, el usuario programa los 0's.

Las memorias PROM son muy utilizadas para experimentar con programas y/o datos, sustituyendo a las memorias ROM. El

## "CIRCUITOS COMBINACIONALES Y SECUENCIALES MSI".

método de grabación en una PROM en normalmente por medios -  
eléctricos (impulsos de voltaje).

Las memorias EPROM pueden ser borradas mediante luz ultravioleta (típicamente de 2537 Angstroms de longitud de onda). recomendándose una exposición de 15 Watts por segundo por centímetro cuadrado. Si se utiliza una lámpara de tipo comercial de  $12 \text{ mW cm}^2$ , colocándola a 2.5 cm del chip. la memoria será borrada en 21 minutos.

Es de hacer notar que la longitud de onda de la que hablamos se encuentra también en la luz normal, así que para evitar la pérdida accidental de datos por exposición a la luz se recomienda cubrir la ventana transparente de estos chips con una película opaca.

Por último, las memorias EEPROM permiten grabar y borrar los datos por medios exclusivamente eléctricos, sin necesidad de utilizar luz UV y con niveles de voltaje TTL.

Reiteramos la necesidad de consultar los manuales de memorias para tener conocimiento de los tipos y capacidades de estos dispositivos.



---

CAPITULO II

---

"CONTROLADORES TRADICIONALES"

---

Un Controlador es un sistema que presentará un comportamiento determinado por su diseño, siempre y cuando reciba las señales de entrada previstas; es decir, es un sistema cuyas salidas responden a las entradas de acuerdo con su diseño.

Los Controladores Digitales cumplen con esta regla, ya que reciben señales digitales como entradas y responden cambiando de estado y/o presentando ciertas salidas. La diferencia básica entre un controlador y un circuito combinacional, es que el controlador puede "tomar decisiones" y responder de forma distinta a una misma entrada, ya que tiene la capacidad de "recordar" entradas anteriores. Así, mientras que un circuito combinacional responde siempre igual a una entrada, el controlador podrá responder a esa misma entrada en diferentes salidas.

Los Controladores Digitales son sistemas altamente cotizados y forman el preámbulo para comprender el funcionamiento de los Microprocesadores, la aplicación de tales controladores abarca un amplio campo dentro de la industria siendo por esto muy importante su adecuada comprensión.

Clasificaremos a los controladores digitales en dos --

grupos: Controladores cableados y controladores microprogramados. En este capítulo analizaremos los del tipo cableado o tradicionales, en el capítulo siguiente analizaremos los microprogramados.

Aunque como veremos la diferencia entre estos tipos de controladores es clara, se puede diseñar un controlador requerido mediante cualquiera de estos tipos mencionados. La elección del tipo de controlador a utilizar, dependerá de la flexibilidad necesaria, mantenimiento y costo.

---

## II.2.

## CONTROLADORES CABLEADOS

---

El concepto de Controlador Cableado define a aquellos Controladores Digitales cuyo funcionamiento (entradas-salidas) depende exclusivamente de sus conexiones internas, es decir, de los elementos usados en su construcción y de la relación (cableado) entre ellos.

Entre los elementos usados en la construcción de un Controlador deben incluirse los de almacenamiento temporal de datos, los que harán que nuestro sistema "recuerde" datos necesarios para tomar decisiones. Los principales tipos de controladores son diseñados con Flip-Flops, Registros de Corrimiento, Contadores y multiplexores directamente relacionados, se-

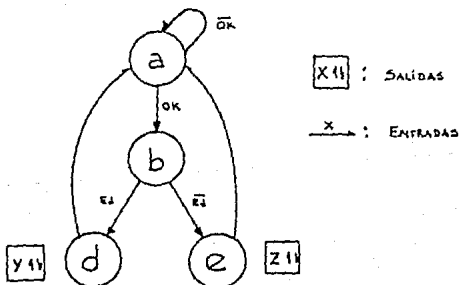
gún veremos adelante.

## II.2.1

## DISEÑO DE FLIP-FLOPS

Para diseñar un controlador mediante esta técnica, se siguen los pasos utilizados en el diseño SSI de cualquier sistema secuencial:

- 1.- Definir el diagrama de flujo del autómata deseado - considerándolo un circuito Mealy o Moore (de niveles o pulsos). Incluir las salidas deseadas y anotando cuando una variable debe ser tomada en cuenta para tomar cierta decisión. Por ejemplo:



Este diagrama indica que el sistema permanece en el estado (a) hasta que se presente la variable "O.K.", con lo que pasará al estado (b), donde tomará una decisión dependiente de la variable "EJ". Si  $EJ=1$  pasaremos al estado (d) donde se activará la salida "Y", en caso contrario pasaremos al estado (e), activándose la salida "Z". En ambos casos retornaremos al estado (a).

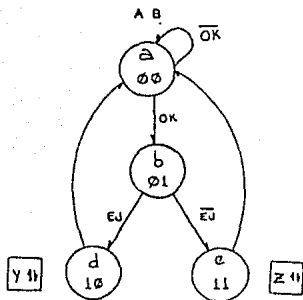
- 2.- Asignar estados binarios a cada uno de los estados presentados. Es evidente que el número de bits usados estará determinado por la cantidad de estados. Además, cada bit corresponderá a un Flip+Flop.

Es conveniente elaborar el mapa K para localizar la zona correspondiente a cada estado, además de que el mapa define en sí a cada estado. A este lo llamaremos también "Mapa de Estados Presentes".

- 3.- Elaborar el Mapa de Estados Futuros, considerando las variables que determinen el cambio de estados. De nuestro ejemplo, presentamos la siguiente figura.

Así indicamos que del estado 00 (a) pasamos al estado (b) si se presenta  $OK=1$ , que del estado 01 (b) pasamos al estado (a) si se presenta  $EJ=0$  y al estado (d) si  $EJ=1$ . Des -

## "CONTROLADORES TRADICIONALES"



A \ B	0	1
0	a	d
1	b	e

ESTOS. PRESENTES

A \ B	0	1
0	OK → b E0	a E0
1	E0 → d E1	a E0

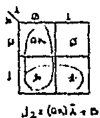
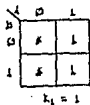
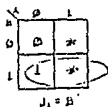
ESTOS. FUTUROS

pués pasaremos incondicionalmente al estado (a)

#### 4.- Elaborar tablas de entrada para cada Flip-Flop.

Después de elegir el tipo de F.F. que usaremos, considerando el estado de un bit, y el estado al que va a cambiar, anotar las entradas necesarias para provocar este cambio, acordes con la tabla de funcionamiento de tal Flip-Flop.

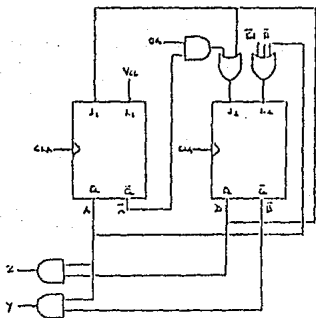
J	K	Q
0	0	Q
0	1	0
1	0	1
1	1	Q



5.- Salidas. (Elaborar tabla de salidas)

$\bar{A}$	0	1
0	*	Y
1	*	Z

6. Construcción



## "CONTROLADORES TRADICIONALES"

Si hubieramos diseñado con Flip-Flops tipo D, las tablas de diseño hubieran quedado de la siguiente manera:

D	$Q_{t+1}$
0	0
1	1

		A	
		0	1
B	0	0	0
	1	1	0

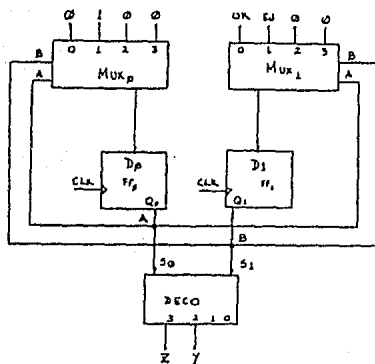
		A	
		0	1
B	0	OK	0
	1	EJ	0

Este diseño también se puede construir mediante una estructura conocida como de "Multiplexores directamente relacionados":

Las tablas anteriores muestran el dato que se debe presentar a la entrada de cada F.F. para generar el estado siguiente tras el pulso del reloj. A cada flip-flop se le asigna un multiplexor con el fin de que, usando el estado presente como líneas de selección del mux, sea presentado a la entrada de cada F.F. el dato correspondiente anotado en las tablas anteriores.

Este dato provendrá de las entradas del multiplexor y se presentará en la salida (de acuerdo a la selección) la cual se conectará con la entrada del flip-flop tipo D correspondiente, como se muestra en la siguiente figura.





## II.2.2.

## DISEÑO CON REGISTROS DE CORRIMIENTO

Esta técnica de diseño se basa en el aprovechamiento de las características de funcionamiento de un Registro Universal (típicamente el 74LS194) para utilizarlo como elemento de almacenamiento en lugar de los Flip-Flops, además de explotar sus capacidades de desplazamiento a la derecha, desplazamiento a la izquierda y carga de un nuevo dato.

Las instrucciones que podemos dar al registro 74LS194 - mediante sus pines de control son las siguientes:

SO	SI	ACCION	COMENTARIOS
0	0	HÓLD	Retiene el estado actual
0	1	SHIFT LEFT	Corrimiento a la izquierda (SL)
1	0	SHIFT RIGHT	Corrimiento a la derecha (SR)
↓	1	LOAD	Carga un nuevo dato (BR)

Los corrimientos los podemos realizar cargando un 0 o un 1 en el "hueco" que se genera, esto resulta en las instrucciones lógicas SLL, SLO, SRL, SRO. Si el corrimiento se realiza cuando se cumple cierta condición (normalmente determinada por una variable de entrada), tendremos como resultado las mismas instrucciones pero en su versión condicional: SLLC, SLOC, SRLC y SROC.

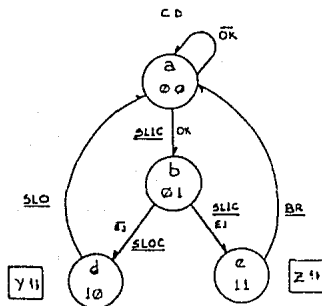
Los pasos que se deben seguir para diseñar un controlador basado en un registro de corrimiento son muy parecidos a los de un controlador de Flip-Flops, pero con las adecuaciones necesarias al registro usado. Desglosaremos este método en siete pasos:

- 1.- Elaborar el diagrama de flujo del autómata anotando las variables que intervienen en la entrada y salida del sistema.

2.- Asignar estados elaborando la tabla de estados presentes y la tabla de instrucciones a ser aplicadas a cada estado.

Utilizaremos el mismo ejemplo que en el controlador anterior para que sirva como modelo en el diseño de controladores de cada tipo que mencionaremos. Es decir, resolveremos este ejemplo mediante cada técnica que desarrollaremos, incluyendo ésta.

Dicho lo anterior, los dos primeros puntos (mencionados anteriormente) se desarrollarán como sigue:



	c	0	1
0	a	d	
1	b	e	

ESTADOS PRESENTES

	c	0	1
0	SLIC OK	SL0	
1	SL(EJ)	BR	

INSTRUCCIONES

Note que la primera instrucción (en el estado 00) es -

"CONTROLADORES TRADICIONALES"

SLIC, ya que para generar el estado 01 a partir del 00, es necesario realizar un corrimiento hacia la izquierda cargando un 1. Además, se trata de una instrucción condicional ya que para que se efectúe este cambio de estado, es necesario que se presente la variable OK.

- 3.- De acuerdo a la Tabla de instrucciones anterior y conociendo la Tabla de control de modo del registro elaborar mapas de acción para cada pin de control - del 74LS194:

S <sub>0</sub>	S <sub>1</sub>	Modo
0	0	HOLD
0	1	SL
1	0	SR
1	1	BR

	C	0	1
D			
0		0	0
1		0	1

$$S_0 = C + D$$

	C	0	1
D			
0		OK	1
1		1	1

$$S_1 = OK + C + D$$

El estado de los pines S<sub>0</sub> y S<sub>1</sub> determina la acción que realizará el registro en cada estado, cumpliendo con la tabla de control de modo anterior. Aquí se determina entonces si va-

a existir un corrimiento, retención o brinco en un estado cualquiera del sistema; estamos en cierta manera "programando" el funcionamiento del controlador.

- 4.- Elaborar mapas de carga en paralelo para determinar el dato que se deberá cargar cuando se presente un brinco:

	$\bar{C}$	$C$
$\bar{D}$	X	X
$D$	X	$\emptyset$

$C_{CARGA} = \emptyset$

	$\bar{C}$	$C$
$\bar{D}$	X	X
$D$	X	$\emptyset$

$D_{CARGA} = \emptyset$

Note que el único brinco se presenta en el estado (e)

- 5.- Elaborar mapas de carga en serie para determinar el dato que se deberá cargar cuando se presente un corrimiento.

	$\bar{C}$	$C$
$\bar{D}$	1	$\emptyset$
$D$	$\bar{E}D$	X

$SL = (\bar{E}D) \bar{C} + \bar{C}D$

	$\bar{C}$	$C$
$\bar{D}$	$\emptyset$	$\emptyset$
$D$	$\emptyset$	X

$SR = \emptyset$

Note que no existen desplazamientos a la derecha y que en el estado (d) no se presenta ningún desplazamiento.

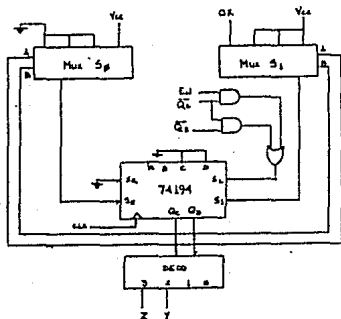
- 6.- Elaborar mapa de salidas para relacionarlas con un decodificador.

	c	0	1
D	0	x	y
	1	X	Z

Podemos simplificar nuestro diseño aplicando el concepto de Multiplexores directamente relacionados a las entradas de control del registro, evitando así tener que alambrear la función obtenida de los mapas de acción con componentes SSI.

Estos dos elementos (Decodificador de salidas y Multiplexores directamente diseñados a la entrada) serán aplicados constantemente en los subsecuentes diseños, llegando a ser determinantes para algunos de ellos.

## 7.- Construcción



### II.2.3.

### DISÑO DE CONTADORES

Otra posibilidad de estructurar un controlador es basándose en un contador con capacidad de cargar un dato y de retenerlo para comenzar, cuando sea necesario, una cuenta a partir de él.

Los estados de nuestro autómata pueden ser generados por un contador como el 74LS161, el cual cuenta con dos pines-

## "CONTROLADORES TRADICIONALES"

da control que permiten habilitar el contador para cargar --- un nuevo dato, retenerlo o contar. Esto permite que estemos -- en la posibilidad de utilizar las siguientes instrucciones lógicas:

- CI: Cuenta incondicional. El estado actual se incrementa en uno.
- CC: Cuenta condicional. El estado actual se incrementa en uno solamente si se presenta cierta variable. Si esta variable no se encuentra en estado activo, el contador retendrá el dato que se presente actualmente a la salida.
- BI: Brinco incondicional. El estado actual cambia por el dato que se presente en las entradas de carga en paralelo.
- BC: Brinco condicional. El estado actual cambia por el dato que se presente en las entradas de carga en paralelo, siempre y cuando se presente : -- cierta variable. Si esta variable no se encuentra en estado activo, el contador retendrá el dato actual.
- HD: Retención. El estado actual es retenido.



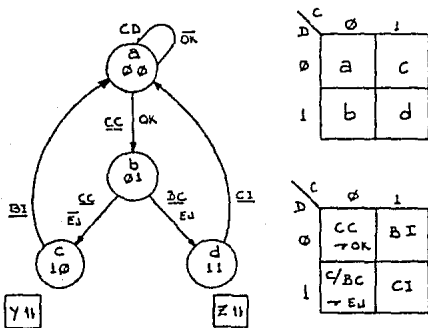
Estas instrucciones se generan a partir de la Tabla -- de control de modo del contador. Los pasos de diseño del controlador siguen la misma línea de los anteriores. A saber:

- 1.- Elaborar diagrama de flujo del autómata a diseñar.
- 2.- Asignar estados elaborando tablas de estados pre -- sentes y de instrucciones a ser aplicadas en cada -- estado.

Los dos puntos anteriores los desarrollaremos como una solo, ya que seguiremos utilizando nuestro primer ejemplo y el diagrama de flujo es el mismo.

Las instrucciones se asignarán dando evidente priori -- dad a las cuentas y tratando de evitar los brincos para simpli -- ficar el diseño. No está de más el comentar, además, que la -- solución de un controlador varía con la asignación de estados -- y que un controlador se simplifica cuando la asignación de es -- tados se presta para evitar los brincos. En el caso que nos -- ocupa, seguiremos respetando por simples, los estados que -- asignamos originalmente.

Diagrama completo y tablas de estados presentes y de instrucciones.



3.- De acuerdo a la Tabla de instrucciones anterior y conociendo la Tabla de Control de modo del contador, elaborar mapas de acción para cada pin de control de 74LS161.

De el mismo modo que con el registro de corrimiento, aquí estamos realizando la programación de nuestro controlador.

ENP	LOAD	ACCIÓN
0	0	HOLD
0	1	BRINCO
1	0	CUENTA
1	1	BRINCO

CONTROL DE MODO 74161

	C	0	1
D			
0		OK	*
1		1	1

EN P

	C	0	1
D			
0		0	1
1		EU	0

LOAD

- 4.- Elaborar mapas de carga en paralelo para determinar el dato que se deberá cargar cuando se presente un brinco.

	C	0	1
D			
0		X	0
1		1	X

C<sub>CARGA</sub> = D

	C	0	1
D			
0		X	0
1		1	X

D<sub>CARGA</sub> = D

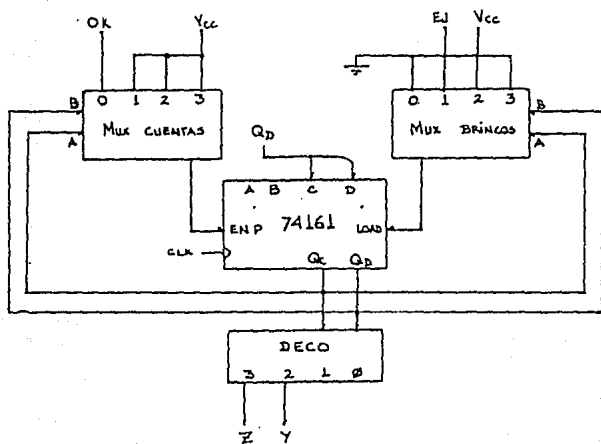
- 5.- Elaborar mapa de salidas.

Este punto ya no lo repetiremos, pues es el mismo que -  
hemos venido presentando durante todo el capítulo y la solu -

## "CONTROLADORES TRADICIONALES"

ción será con el mismo decodificador de salidas.

- 6.- Construcción utilizando multiplexores directamente relacionados para la entrada a los pines de control, extraída de los mapas de acción.



---

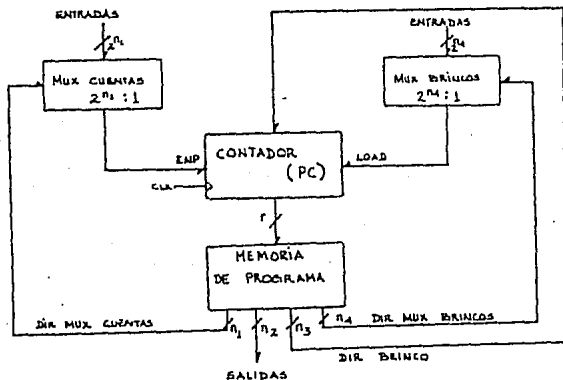
CAPITULO III

"CONTROLADORES MICROPROGRAMADOS"

---

Hasta este momento, los controladores que hemos diseñado basan su funcionamiento en sus conexiones internas, por eso se les llamó controladores "cableados". Para modificar el funcionamiento de un controlador de este tipo es necesario modificar las conexiones (el cableado) entre los circuitos que lo componen; claro está que muchas veces esto es difícil y complejo.

Se puede agregar flexibilidad al sistema incluyendo una memoria que nos permita almacenar los datos que requerimos para trabajar mediante la siguiente estructura:



Se trata de una variante de un controlador basado en un contador. La diferencia es que ahora el contador se llamará-- "contador de programa" (PC), y manejará a la memoria de programa, la cual contiene ahora, además de las salidas, las direcciones de selección de los multiplexores de variables de entrada y las direcciones de los bríncos posibles.

Esta estructura permite modificaciones en su funcionamiento mediante el cambio del programa que se tiene en memoria es decir, se puede modificar por "software". Esta es una de las razones que hacen de este tipo de controladores los preferidos por los diseñadores profesionales.

Continuando con el ejemplo utilizado durante el capítulo anterior, se puede resolver este controlador mediante la estructura microprogramada de acuerdo con los siguientes pasos:

- 1.- Diagrama de flujo del autómata deseado.
- 2.- Asignación de estados y mapa de instrucciones del contador.
- 3.- Mapas de acción para los multiplexores (Mux cuentas y Mux bríncos).

4.- Escritura del programa.

5.- Alambrado.

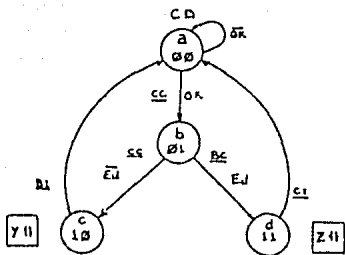
Con respecto al paso 3, no es necesario alambra los -- multiplexores  $\Sigma$ 1 y como indican los mapas K; bastará con co - locar las variables de entrada en los los Mux sin repetirlas y anotar en el programa su dirección tantas veces como las necesi - tamos. Esto se puede hacer porque el contador ya no contro - la las direcciones de selección de los Muxs.

La carga en paralelo será alimentada directamente desde la memoria, por lo que no necesitaremos mapas de carga en para - lelo y nos ahorraremos funciones para alambra.

Retomando el ejemplo del capítulo anterior, recordare-- mos los tres primeros pasos de diseño del controlador micropro - gramado como los tres primeros pasos de diseño del controlador basado en un contador.

El cuarto paso a seguir en el diseño del controlador -- microprogramado (escritura del programa) se resuelve mediante la siguiente tabla de direcciones y salidas:





	C	0	1
0	CC	OK	BI
1	C/BC	EJ	CI

INSTRUCCIONES

	C	0	1
0	OK	*	
1	1	1	

MUX CUENTAS

	C	0	1
0	0	1	
1	EJ	0	

MUX BRINCOS

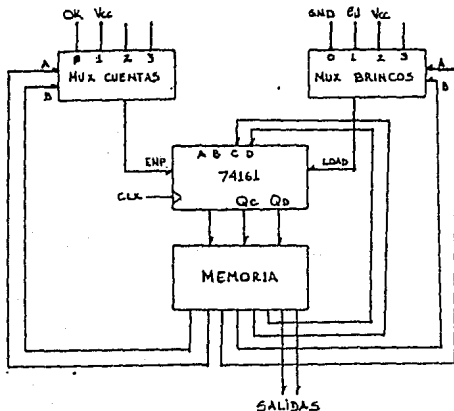
DIRECCION DEL PC	DIRECCION MUX CNTAS.		DIRECCION MUX BRCOS.		DIRECCION BRINCOS		SALIDAS Z Y	
	0	1	0	1	0	1	0	1
0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	1	0	0
1	0	0	1	0	0	0	1	0
1	1	0	0	1	0	0	0	1

La dirección del PC, es realmente el valor que envía el PC a la memoria, es decir, realmente significa o indica una -

## "CONTROLES MICROPROGRAMADOS"

localidad de memoria. En cada una de las cuatro localidades -- de memoria necesarias se debe grabar la serie de datos (direcciones) correspondientes. Por ejemplo, en la localidad de memoria 00 se deberá grabar el dato 00000000; y así sucesivamente con las localidades restantes.

El quinto paso de diseño del controlador microprogramado (alambrado), se realiza respetando la estructura mencionada al principio de este capítulo con el cuidado de colocar correctamente las variables de entrada en los multiplexores correspondientes.



La estructura de un controlador microprogramado vista - en el punto anterior, genera las instrucciones mediante la combinación de las variables de entrada a los pines de selección del contador (ENP Y LOAD). Para cambiar una instrucción por - otra, necesitamos escoger otra combinación desde los multiple- xores CUENTAS Y BRINCOS.

Se puede aumentar la versatilidad del controlador si po demos escoger la instrucción que se dará al contador mediante un código de operación (OPCODE) de tal manera que las decisio- nes dependan de una sola variable de entrada (FLAG). Con esta- estructura estaremos usando un módulo (MYCA I) que generará -- la siguiente dirección de la memoria gracias al OPCODE recibi- do de ella, considerando el valor de la bandera elegida.

Las instrucciones que recibe un módulo MYCA I pueden - ser:

"CONTROLES MICROPROGRAMADOS"

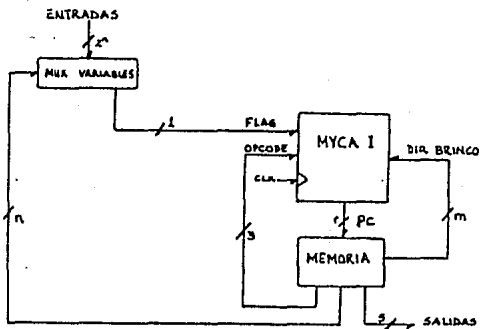
NEMONICO	FLAG	ACCION	DESCRIPCION
CC (FLAG)	0 1	RETIENE: PC ← PC CUENTA: PC ← PC + 1	CUENTA CONDICIONAL
CI	0 1	CUENTA: PC ← PC + 1 CUENTA: PC ← PC + 1	CUENTA INCONDICIONAL
BC (FLAG) (DIR)	0 1	RETIENE: PC ← PC BRINCA: PC ← DIR	BRINCO CONDICIONAL
BI (DIR)	0 1	BRINCA: PC ← DIR BRINCA: PC ← DIR	BRINCO INCONDICIONAL
C/BC (FLAG) (DIR)	0 1	CUENTA: PC ← PC + 1 BRINCA: PC ← DIR	CUENTA/BRINCO CONDICIONAL

Esta tabla es utilizada para diseñar la estructura interna del módulo MYCA I, este diseño lo realizaremos en el último capítulo de este manual, dentro de la solución a la práctica correspondiente a este controlador. Esto es con el fin de enfocarnos en la aplicación del módulo y definir cómo se puede construir un controlador auxiliados por un módulo MYCA I.

Es de hacer notar que una condicionante en el diseño con MYCA I, es que sólo se pueden tener dos estados siguientes por cada estado presente, y además, las decisiones deben depender de una sola variable o función común de las variables involucradas.

La arquitectura de trabajo para diseñar un controlador--

auxiliados de un módulo MYCA I se muestra en la siguiente figura:



Note que en la memoria se encuentran grabadas las salidas de nuestro sistema, además contiene la instrucción (OPCODE) que se aplicará al módulo MYCA para determinar la localización de la siguiente instrucción del programa en memoria. La instrucción derivada del OPCODE se ejecuta tomando en cuenta la bandera (FLAG) proveniente del multiplexor de variables de entrada, cuya selección se realiza por medio de una dirección localizada también en la memoria.

Los pasos para diseñar un controlador mediante la téc -

nica MYCA I son muy sencillos si los comparamos con los de los controladores de la sección anterior. Los podemos numerar como:

- 1.- Diagrama de flujo del autómata deseado.
- 2.- Asignación de estados y determinación de las instrucciones MYCA I necesarias para producir el cambio de estado deseado, en asociación con una única bandera.
- 3.- Escritura del programa con nemónicos mencionando las banderas asociadas y las direcciones de los brincos.
- 4.- Traducción de los nemónicos a datos binarios en el orden siguiente: OPCODE - Dirección del Mux de variables - Dirección de brinco Salidas. Este es el programa que se grabará en la memoria.
- 5.- Alambrado del controlador colocando las variables de entrada en las posiciones mencionadas en el programa.

---

### III.3

### CONTROADORES QUE MANEJAN SUBROUTINAS

---

La capacidad de manejo de estados se multiplica si podemos diseñar utilizar un controlador que maneje subrutinas, y -

más aún si podemos tener subrutinas anidadas.

El controlador denominado MYCA II es un módulo que genera la dirección de la siguiente instrucción en una memoria gracias al OPCODE recibido de ella en combinación con una bandera (FLAG) representativa de las variables de entrada. La característica a resaltar en este módulo, es su capacidad de brincar a una dirección para iniciar una rutina que será ejecutada cuantas veces sea necesario, hasta que encuentre una instrucción de retorno y la bandera esté encendida. En este caso se dará por terminada la subrutina y se continuará con el programa en la instrucción siguiente a la que se dejó al pasar a esta subrutina.

Concluyendo, el generador de direcciones MYCA II, nos permite incluir en el programa procesos repetitivos que pueden ser llamados mediante una instrucción de salto a subrutina (JSR). Saldremos de este proceso mediante una instrucción de retorno (RSR); además, podemos detener la ejecución del proceso con una instrucción de interrupción (INT).

La tabla de diseño para estas instrucciones será como sigue:

"CONTROLES MICROPROGRAMADOS"

NEMONICO	FLAG	ACCION	DESCRIPCION
JSR (FLAG) (DIR)	0 1	CUENTA: PC ← PC + 1 SALTA: PC ← DIR ; TOP ← PC + 1	SALTO A SUBROUTINA
RSR (FLAG) (DIR)	0 1	BRINCA: PC ← DIR REGRESA: PC ← TOP	REGRESO DE SUBROUTINA
INT (FLAG) (DIR)	0 1	CUENTA: PC ← PC + 1 BRINCA: PC ← DIR ; TOP ← PC + 1	SALTO POR INTERRUPCION

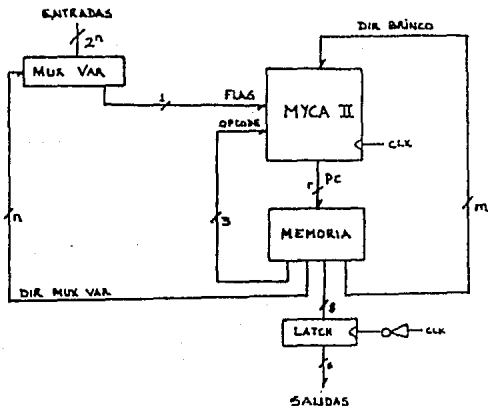
Por supuesto, estas instrucciones son adicionales a las -- cinco con las que cuenta MYCA I, de esta manera MYCA II contará con ocho instrucciones.

El diseño interno de este módulo se realizará en el último capítulo de este trabajo, en la solución a la práctica correspondiente a este tipo de controlador. Sin embargo no dejaremos de mencionar que cuando se ejecuta la instrucción JSR, mientras que se salta a la dirección donde se inicia la subrutina, se realiza también una instrucción PUSH a un stack o registro de pila en donde se guarda PC + 1; esto equivale a decir que se está guardando (en este caso en la variable TOP) la dirección de la siguiente instrucción a ejecutarse cuando se retorne de la subrutina. Esto se demuestra en la instrucción RSR en donde el PC se reinstaura la siguiente instrucción mediante una instrucción POP al stack, lo cual significa que se regresa (en este caso mediante la variable TOP) a la siguiente instrucción --



de la que se ejecutó al saltar a la subrutina.

La arquitectura de trabajo para diseñar un controlador-auxiliados de un módulo MYCA II se muestra en la siguiente figura:



Note que la estructura de este sistema es semejante a la de MYCA I, los pasos de diseño de un controlador mediante este módulo también son los mismos de tal manera que la diferencia real entre estos dos tipos de controladores estriba en su capacidad de funcionamiento relacionado con el aumento de instrucciones (aquellas que manejan subrutinas), amén de las diferencias

## "CONTROLES MICROPROGRAMADOS"

en estructura interna de los dos módulos resultantes de este mismo aumento de instrucciones.

En el último capítulo de este trabajo, presentaremos un ejemplo que será resuelto mediante este tipo de controlador. En este ejemplo abarcaremos el diseño de el módulo MYCA II.

---

---

**CAPITULO IV**

**"DISEÑO CON CARTAS ASM"**

---

Una vez que se ha planteado con palabras, diagramas, etc., la necesidad de un Sistema Digital y las características de comportamiento deseadas para este sistema, se abre ante el diseñador un amplio horizonte de soluciones posibles, tanto por los elementos de hardware que se pueden utilizar como por los distintos métodos aplicables para dar solución a los requerimientos funcionales.

En 1973 fué propuesto por Clare un método auxiliar para el diseño de sistemas; llamó a los sistemas digitales como "Máquinas de Estado Algorítmico" (ASM) y formalizó la presentación en diagramas de flujo de la relación entre los estados y el funcionamiento del sistema. A estos diagramas de flujo se les llama "Cartas ASM".

Las cartas ASM son entonces representaciones de algoritmos y las podemos relacionar con los diagramas de flujo que se elaboran antes de escribir un programa de computación. La carta ASM representa al "programa" que va a gobernar el funcionamiento del autómatas siendo posible "escribir" este programa en una memoria o de forma alambrada por hardware.

Si relacionamos los conceptos e ideas anteriores con lo

caja y el nombre del estado se anota en un círculo fuera de la caja. La combinación de variables de estado se denomina "código de estado" y se anota en binario sobre la caja.

Finalmente, el flujo entre los estados se indica por -- medio de líneas de entrada y salida. Ver figura 4.1

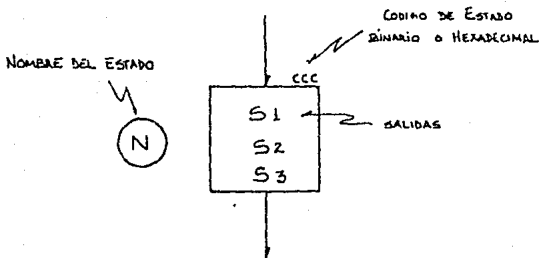


Figura 4.1 Caja de estado

b) Diamante de decisiones.

Representa un punto en el que el estado siguiente depende de el valor que adquiriera cierta variable. La variable de prueba (de la que depende la decisión) se anota dentro del diamante.

Las salidas del diamante se marcan siempre como 0 y 1 desviando el flujo de acuerdo al valor de la variable de prueba. Ver figura 4.2.

El diamante solo tendrá dos salidas y una sola entrada; las salidas se marcarán siempre como 0 y 1 por fines de seguridad, evitando la incertidumbre generada si se marcaran como -- "verdadero" o "falso" cuando la presencia de una señal se detecta en valor negado.

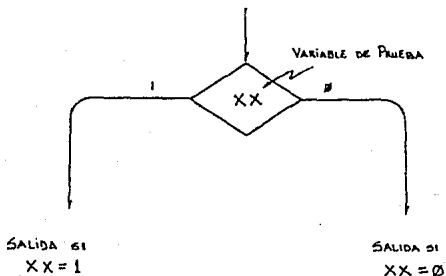


Figura 4.2 Diamante de decisión.

c) Caja de salida condicional

Representa una salida que depende de una entrada en un estado determinado. Su línea de entrada siempre proviene de un diamante de decisión que indica la condición a cumplir para que se produzca la salida de esta caja.

La salida se anota en el interior de la caja.

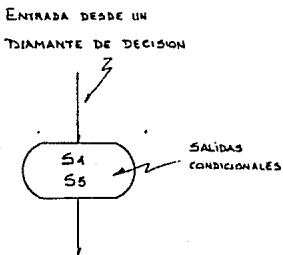


Figura 4.3. Caja de salida condicional.

d) Nodo conector.

Se utiliza para hacer más claras las representaciones -

de entradas múltiples hacia un símbolo ASM. Un nodo puede tener hasta tres entradas múltiples pero solo tiene una salida.

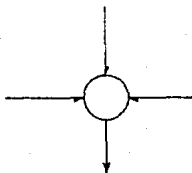


Figura 4.4      Nodo conector

e) Bloque ASM

Un bloque ASM consiste en un solo estado asociado con sus salidas, incluyendo diamantes de decisión y salidas condicionales.

Como cada caja de estado tiene un período de tiempo asociado con ella, cada bloque ASM representa un período sencillo de tiempo, o sea, un intervalo de reloj.



## "DISEÑO CON CARTAS ASM"

Un bloque ASM define sus salidas independientes, salidas condicionales y las condiciones para determinar el próximo estado.

Un ejemplo de un bloque ASM se presenta en la figura 4.5 enmarcado por un recuadro punteado.

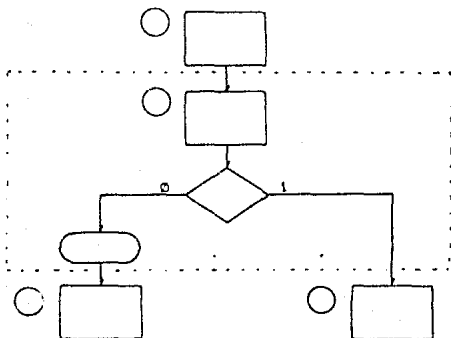


Figura 4.5 Bloque ASM.

Debido a que en una máquina de estados secuenciales solo se puede tener un estado a la vez, es decir, no se pueden tener estados simultáneos; nunca se divide una salida en dos o más estados sin utilizar diamantes de decisión.

En ocasiones es necesario dividir una salida para generar salidas condicionales simultáneas como se muestra en la figura 4.6.

Los dos casos mostrados en esta figura de ejemplo son - válidos, pero es recomendable usar la forma de la figura 4.6 b- ya que asegura que solamente una condición es probada a la vez - y evita ambigüedades en cartas más complejas. Recalcamos que - aunque los dos bloques de la figura 4.6 realizan la misma función, la forma 4.6 b es la más adecuada.

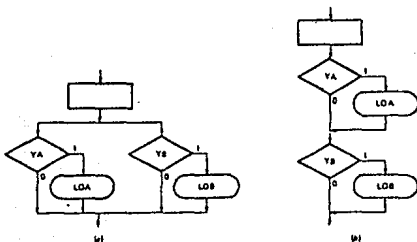


Figura 4.6 a) Conexiones paralelas para salidas condicionales.

b) Conexiones serie para salidas condicionales.

Otra abreviación usada en las cartas ASM es la de com - partir diamantes de decisión y salidas condicionales por dos - bloques ASM distintos siempre y cuando el diamante de condición y la salida condicional sean los mismos, y además, sea también el mismo destino final en los dos bloques.

Para ejemplo, obsérvese la figura siguiente (Figura 4.7) en donde se presentan dos bloques ASM que tienen diamantes de - decisión semejantes y la salida de éstos van a un mismo estado. La figura 4.7 b representa una simplificación de el primer par de bloques de tal manera que la función de éstos no se altera - Ambos casos son equivalentes y se recomienda usar el de la figu - ra 4.7 b.

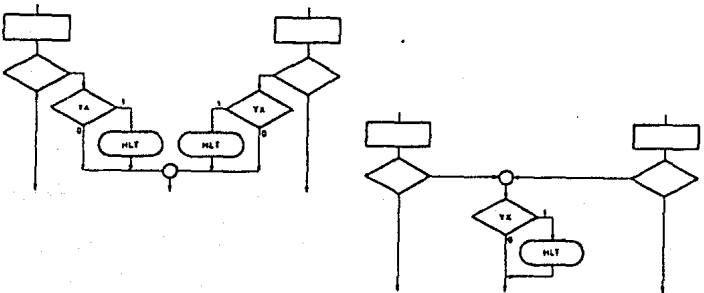


Figura 4.7 Reducción de carta ASM.

Dado que una carta ASM describe un algoritmo de solución, debemos seguir ciertos pasos para llegar al desarrollo de un buen algoritmo. La experiencia de muchos diseñadores nos ha ayudado a definir los siguientes puntos de diseño:

- 1.- Escribir las necesidades de funcionamiento del sistema así como las señales de entrada con que se cuenta y las salidas deseadas.
- 2.- Realizar un diagrama de flujo, al nivel más alto posible para auxiliarnos con la mejor abstracción del comportamiento general del sistema.
- 3.- Realizar diagramas de bloque de los subsistemas (o de el sistema completo si es muy simple) para indicar el flujo de señales entre ellos. Identificar cuidadosamente las señales de entrada y de salida.
- 4.- Describir el comportamiento de cada bloque para facilitar el desarrollo de la carta ASM respectiva, - esta descripción pueda ser escrita o mediante un sencillo diagrama.

- 5.- Desarrollar una carta ASM para cada bloque, cuidando la relación de las señales con las de otros bloques.
  
- 6.- Construir las cartas ASM mediante circuitos lógicos como módulos independientes y después relacionarlos por sus señales de entrada y salida. Esta implementación será libremente elegida por el diseñador.

Es de hacer notar que el desarrollo de los bloques impacta en la arquitectura de todo el sistema en una forma importante por lo que es bueno considerar con cuidado las funciones y relaciones de cada bloque antes de apresurar los demás pasos del diseño.

En el punto final de el proceso de diseño descrito (punto número 6) es donde vamos a aplicar los conocimientos de los capítulos anteriores, pues es donde la carta ASM definida se va a convertir en un controlador cableado o microprogramado.

Hasta este momento ya deberá ser obvio que el diseño de sistemas es complejo. Hemos presentado solo una pequeña parte de los múltiples opciones de solución a un sistema-problema planteado. La intuición y la experiencia juegan un rol muy importante en este proceso. La capacidad de imaginar un algoritmo

mo de solución y describirlo en diagramas de flujo valúan al --  
diseñador.

Es muy importante estar familiarizado con varias arquitecturas de solución para construir un algoritmo físicamente. La familiaridad con las alternativas es el primer paso para la creatividad.

Por último, es conveniente considerar en principio a los bloques del diseño en forma ideal, y además es necesario tener a mano las armas para convertirlo a su forma real cuando nos --  
enfrentemos al ruido, fan-out, retardos, etc... En este orden de ideas es importante conocer los bloques de interfase como lo son los transductores y convertidores, tema que trataremos en --  
el siguiente capítulo.

ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA

---

---

**CAPITULO V**

**"INTERPASES"**

---

Todos los Diseños Digitales que podamos realizar estarán pensados con un fin práctico específico, ya sea para abrir una puerta, mover una pieza, accionar un motor, visualizar datos, etc.

Para poder establecer una relación entre los dispositivos electrónicos que conforman nuestro sistema y el tipo de acción específica que observará el usuario, es necesario contar con una INTERFASE.

El concepto de INTERFASE abarca un gran campo de elementos y aplicaciones referidas, en forma general a la frontera o punto de contacto entre dos partes de un sistema, es decir a los puntos de conexión de señales que enlazan al sistema con el exterior.

Los circuitos de Interfase utilizados en operaciones de Entrada y Salida pueden realizar cualquiera de las siguientes funciones:

- a) Conversión de datos.



- b) Sincronización.
- c) Selección del dispositivo.

En este capítulo analizaremos principalmente elementos de Conversión de datos.

---

V.2.

CONMUTADORES Y TECLADOS

---

A continuación consideraremos algunos dispositivos de E/S básicos y sus especificaciones de interfase. Quizá el dispositivo de entrada más sencillo es un conmutador manual o tecla. La figura 5.1 muestra como un conmutador de dos terminales puede usarse para producir una señal lógica  $z$  que es adecuada para interconectarse con un sistema digital.

Cuando el conmutador está en su posición de conectado o cerrado, la señal de entrada  $X=0$  (tierra) se aplica al inversor causando que su salida  $z$  sea '1' (+5 V). Cuando el conmutador -

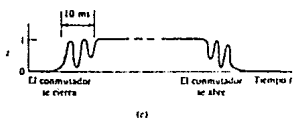
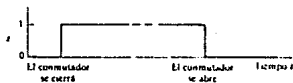
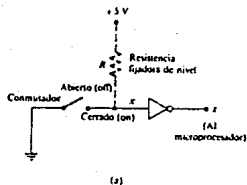


Fig. 5.1 Conmutador de dos terminales

está en la posición de desconectado o abierto, X se lleva al nivel '1' por la resistencia R, provocando que la salida del inversor sea '0'. Si, como generalmente ocurre, se utiliza un inversor TTL, R puede omitirse, ya que una línea de entrada TTL desconectada automáticamente toma el valor lógico de '1'.

Un conmutador idealmente debería tener el comportamiento lógico mostrado en la figura anterior. En la práctica los puntos de contacto físico en un conmutador (electro)mecánico

tienden a rebotar cuando son pulsados juntos o separadamente. - Estos rebotes, aunque imperceptibles por la persona que actúa-- sobre el conmutador, originan que la señal X producida por el conmutador oscile entre los niveles 0 y 1 varias veces hasta es-- tabilizarse en su valor final. La duración del intervalo de re-- botes es del orden de 10 ms; este es un tiempo muy grande compa-- rado con el tiempo de propagación de una compuerta lógica, que-- es normalmente de unos pocos nanosegundos. En consecuencia -- cualquier señal de rebote que afecte a X en el circuito de la - figura 5.1 aparecerá en la salida z del inversor, obteniéndose la forma de onda distorsionada de la figura 5.1.c Si la señal z se suministra a un sistema digital, cuando los contactos del -- conmutador estén rebotando, puede intepretarse incorrectamente el estado en que se encuentra el conmutador.

Para evitar la confusión causada por este fenómeno, que es denominado rebote de conmutación, se utiliza un proceso de - hardware o software de eliminación de rebotes.

La figura 5.2 muestra otro tipo de conmutador que está-- conectado a un sencillo circuito, eliminador de rebotes. El con-- mutador tiene tres terminales en vez de dos, de forma que dos - de las terminales se asocian con las posiciones de conectado y - desconectado. Las dos puertas NAND de la figura 5.2 constitu -- yen un biestable SR. Este biestable tiene como combinación de-- entrada inactiva '11', y es puesto a 1 o a 0 por la llegada de--

un 0 o un 1 a sus líneas de entrada  $X_1$  o  $X_2$ . La aplicación de un 0, por ejemplo, a  $X_1$  durante unos pocos nanosegundos es suficiente para llevar al biestable al estado  $z=1$ . Sucesivas aplicaciones de 0 o 1 a  $X_1$  después no afectan al estado del biestable permaneciendo  $X_2$  a 1. Por tanto, el rebote en la posición de abierto no aparece a la salida del biestable. Lo mismo ocurre con los rebotes producidos cuando el conmutador se pasa a la posición de cerrado.

Es decir, el comportamiento de este circuito se aproxima mucho al ideal.

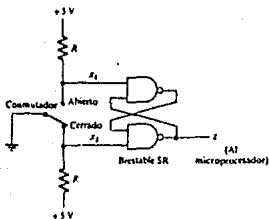


Figura 5.2 Conmutador manual de tres terminales con circuito antirrebotes.

Un teclado es una colección de conmutadores cerrado/--abierto, usualmente del tipo de los de dos terminales, cada uno de los cuales tiene una función específica, como introducir -- un carácter o una orden concreta en la microcomputadora. La figura 5.3 muestra un pequeño teclado del tipo de los utilizados en calculadoras sencillas. Cada una de sus 16 teclas es un conmutador de dos terminales que es cerrado cuando la tecla es pulsada y abierto cuando la tecla vuelve a su posición inicial. Todas las teclas tienen un común, de forma que el teclado presenta una interfase de 17 conexiones eléctricas externas.

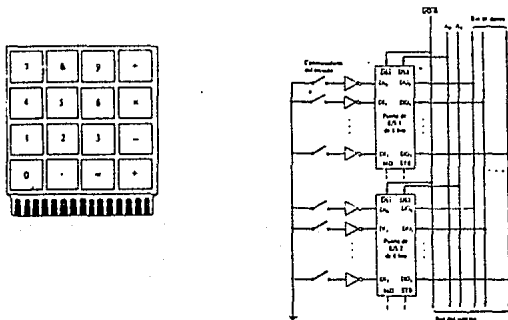


Figura 5.3 Teclado de 16 teclas tipo calculadora.

Un dispositivo de salida que puede considerarse como la contrapartida de un conmutador es una fuente de luz o lámpara - con dos estados; conectada (iluminada) y desconectada (apaga - da). Las lámparas incandescentes se utilizan ocasionalmente como dispositivos de salida en microcomputadoras. Los diodos emi - sores de luz son más adecuados en sistemas digitales por las si - guientes razones:

- 1.- Consumen menos energía eléctrica y producen menos -- calor.
- 2.- Fácilmente con LEDs se pueden construir visualizado - res de caracteres numéricos y alfanuméricos.

Aunque los requerimientos de energía eléctrica de los - LEDs son menores que los de las lámparas incandescentes, consu - men corrientes relativamente grandes en el rango de 15 a 30 mA. Además, el brillo de un LED se incrementa con la intensidad -- de corriente que fluye a través de él.

Un conjunto de LEDs puede combinarse en una única cápsu - la de forma que se pueden visualizar caracteres seleccionando -

la situación (conectado/desconectado) de cada LED.

En la figura 5.4 aparecen dos ejemplos importantes: un visualizador LED de siete segmentos diseñado para presentar los 10 dígitos decimales 0, 1, ..., 9, y una retícula de LEDs diseñada para conseguir la visualización de un conjunto completo de caracteres alfanuméricos. La matriz de visualización contiene 35 LEDs independientes formando una retícula de 7 filas y 5 columnas. Comercialmente existen cápsulas conteniendo grupos de estos circuitos de visualización LED, éstos son unos dispositivos de salida muy prácticos para calculadoras y otros sistemas digitales.

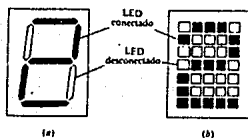


Figura 5.4. Visualizadores (a) de 7 segmentos; (b) de retícula.

Esta discusión se extenderá para incluir al fototransistor, que tiene una unión NP colector-base fotosensitiva. La corriente inducida por efectos fotoeléctricos es la corriente de base del transistor. Si nosotros asignamos la notación  $I_\lambda$  para la corriente de base fotoinducida, la corriente de colector resultante es aproximadamente:

$$I_c \cong h_{fe} I_\lambda$$

Un conjunto representativo de características del fototransistor se proporciona en la figura 5.5 con la representación simbólica del dispositivo. Note las similitudes entre estas curvas y las del transistor bipolar típico.

Algunas de las áreas de aplicación del fototransistor incluye lecturas de tarjeta perforada, circuitería lógica para computador, control luminoso, indicadores de nivel, relays y sistemas de conteo.



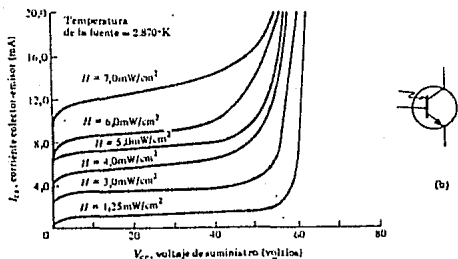


Figura 5.5. Fototransistor: (a) características típicas  
(b) símbolo.

El opto-aislador es un dispositivo que incorpora muchas de las características descriptivas en la sección precedente. - Es simplemente un paquete que contiene tanto un LED infrarrojo como un fotodetector tal como un diodo de silicio, un par tran-

sistor de Darlington, o un SCR. La longitud de onda de respuesta de cada uno de los dispositivos está ajustada para que sea tan idéntica como sea posible y permitir la medida más alta de acoplamiento. En la figura 5.6 se proporcionan dos configuraciones de pastilla posibles. Hay un gorro aislador transparente entre cada conjunto de elementos embebido en la estructura (no visible) para permitir el paso de luz. Ellos son diseñados --- con tiempos de respuesta tan pequeños que pueden ser utilizados para transmitir datos en el rango de los megahertz.

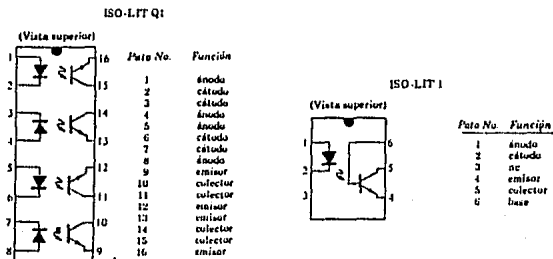


Figura 5.6 Opto-aisladores.

Los opto-aisladores o acopladores ópticos se utilizan para efectuar la interfase de sistemas digitales con otros dis-

positivos que funcionan en circuitos de elevadas tensiones o -- corrientes, con lo que son eléctricamente incompatibles. Por -- ejemplo, los circuitos eléctricos de un automóvil actúan con 12 VDC, en lugar de los 5 VDC con que actúan la mayor parte de cir -- cuitos lógicos. Además, las corrientes eléctricas absorbidas -- por ellos son del orden de amperios en lugar de miliamperios.

Un acoplador óptico resuelve el problema de incompati -- bilidad eléctrica utilizando luz para transmitir información en -- tre un circuito electrónico y un dispositivo de E/S, mientras -- que sus circuitos eléctricos permanecen desconectados o aisla -- dos uno del otro.

La figura 5.7 muestra un acoplador óptico que se usa -- como interfase entre un microprocesador y un motor eléctrico -- que funciona a 12 VDC y que absorbe una corriente elevada; es -- to es usual en los motores utilizados en los elementos acceso -- rios de un automóvil. La señal  $z$  del microprocesador indica -- si el motor debe conectarse o no, pero obviamente esta señal -- no puede conectarse directamente al motor. En su lugar  $z$  se -- aplica al LED del acoplador óptico, con lo que se convierte en -- pulsos de luz las señales que aparecen en  $z$ . Estos pulsos de -- luz son recibidos por el fototransistor  $T_1$  causando que éste -- conmute a conducción o corte en consonancia con el valor de  $z$ .

$T_1$ , a su vez excita a un transistor  $T_2$  de "potencia" que

soporta alta corriente y que puede conmutar al motor eléctrico a conducción y corte, según especifique la señal  $z$  de salida - del microprocesador. Observe también que la velocidad de rotación de la mayoría de los motores eléctricos puede controlarse variando la frecuencia de  $z$ . Por tanto, con un programa apropiado el circuito de interfase de la figura 5.7 puede usarse -- para controlar la velocidad de un motor eléctrico.

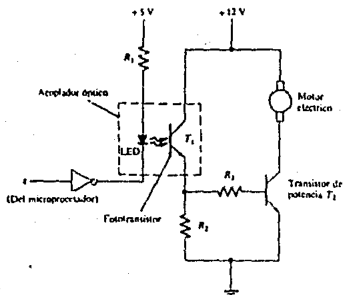


Figura 5.7 Interfase con opto-aislador.

---

V.4.

---

ENTRADA/SALIDA ANALÓGICA

---

Frecuentemente una microcomputadora, que es un sistema digital, debe interactuar con variables externas de E/S de naturaleza analógica en lugar de naturaleza digital. En esta sección se examina el diseño de interfases analógico/digitales. Se discuten algunos dispositivos de E/S representativos, incluyendo transductores y motores eléctricos, así como el problema de convertir datos entre las formas analógica y digital.

---

V.4.1.

---

REQUERIMIENTOS DE LA INTERFASE

---

Una función básica de los dispositivos de E/S es transformar información de varias formas no eléctricas a una forma eléctrica digital, así los conmutadores examinados en la sección V.2 transforman 2 posiciones físicas de un dedo del operador en dos niveles de tensión  $V_L$  y  $V_H$ , correspondientes a los valores lógicos 0 y 1 en dos niveles de intensidad luminosa (obscuridad e iluminación).

Cuando tanto las señales eléctricas como no eléctricas que intervienen en la interfase de E/S son primariamente digitales, como en conmutadores, LED, y dispositivos similares, se dice que el dispositivo es un dispositivo digital de E/S. Hay otra clase importante y numerosa de dispositivos que tratan con

señales externas analógicas, es decir, que varían continuamente, éstos se denominan dispositivos de E/S analógicos.

Entre las magnitudes analógicas más comúnmente utilizadas se encuentran:

- Posición espacial.
- Velocidad.
- Aceleración.
- Temperatura.
- Presión.
- Velocidad de flujo.
- Intensidad de radiación.

Generalmente se requieren dispositivos analógicos de entrada cuando una cantidad física continuamente variable debe medirse con precisión, mientras que los dispositivos analógicos de salida son necesarios para controlar magnitudes físicas continuamente variables. Muchas aplicaciones de los sistemas digitales utilizan medidas y control analógicos.

En la figura 5.8 se ilustran los requisitos globales para realizar una interfase entre un microprocesador y variables analógicas. En primer lugar la señal analógica no eléctrica de

entrada se convierte en una señal eléctrica proporcional (analógica) por medio de un transductor o sensor. La salida del transductor usualmente debe ser modificada por circuitos electrónicos acondicionadores que adaptan las características -- eléctricas del transductor a las del dispositivo que alimen -- tan.

Las funciones habituales efectuadas por los circuitos - acondicionadores de señales son filtrado de ruido y amplificación de tensión. A continuación la señal eléctrica acondicionada se convierte en una señal digital proporcional por medio de un conversor analógico/digital (ADC) controlado por hardware - o por software. La señal digital resultante puede ser transferida a la computadora central para su almacenamiento o procesamiento. Un conjunto de operaciones similares se necesitan para efectuar la interfase de un sistema digital con un dispositivo analógico de salida, como lo muestra la siguiente figura.

Las señales digitales de salida del microprocesador -- primero se convierten en señales eléctricas analógicas proporcionales por medio de un conversor digital/analógico (DAC). La salida del DAC, es acondicionada por circuitos apropiados (circuitos excitadores de salida, por ejemplo) y transferida al - dispositivo analógico de salida.

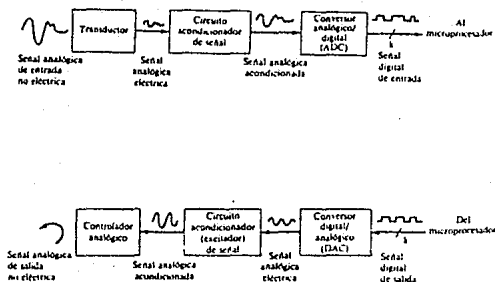


Figura 5.8 Interfases de una microcomputadora

Debería observarse que en muchos casos de interés, las magnitudes analógicas de salida pueden controlarse directamente por señales digitales, con lo que en estos casos puede hacerse la interfase de la misma forma que en los dispositivos de salida digitales. Consideremos de nuevo la interfase para el motor eléctrico de la figura 5.7. Como observamos anteriormente, la velocidad del motor puede controlarse variando la velocidad con que es conmutado a conectado y desconectado a través del acoplador óptico. De esta forma la señal digital de -



salida z controla la cantidad analógica de velocidad sin un -  
circuito DAC explícito.

Los dispositivos digitales de salida son usados frecuen-  
temente para visualizar información obtenida de fuentes analó-  
gicas. Los instrumentos "digitales" de medida emplean unidades  
visualizadoras digitales de salida, tales como LEDs o CRTs, pa-  
ra mostrar cantidades "analógicas" correspondientes a magnitu-  
des tales como tensión, corriente, temperatura y presión; en -  
lugar de unidades analógicas de salida tales como las produci-  
das por indicadores electromecánicos con el movimiento de agu-  
jas y dials (analógicos). Con los visualizadores digitales -  
se obtiene mayor precisión, y éstos son inherentemente más fá-  
ciles de interconectar a un sistema digital. Es decir, hay -  
menos necesidad de dispositivos analógicos de salida.

---

V.4.2

CONVERSION ANALOGICO/DIGITAL

---

Se conocen numerosas técnicas para convertir datos de -  
E/S entre formas analógica y digital. Estas tratan de conse-  
guir que no haya disociación entre velocidad y precisión y tam-  
poco entre la complejidad del hardware y software.

Una interfase sencilla entre los entornos analógico y -  
digital puede efectuarse con un circuito electrónico denomina-

do comparador, que se define en la figura 5.9.

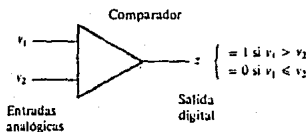


Figura 5.9 Comparador analógico

De forma similar al comparador lógico discutido en la sección I, este comparador compara sus valores relativos. En el caso actual las entradas son dos funciones analógicas  $v_1$  y  $v_2$ , y la salida  $z$  es una señal lógica binaria;  $z=1$  si  $v_1 > v_2$ , si no  $z = 0$ . Por sí mismos los comparadores son útiles en circuitos de alarma, como el ilustrado en la figura 5.10. Aquí la salida analógica  $v$  de un transductor (térnico) se aplica a un comparador cuya salida digital  $z$  es monitoreada por un microprocesador. Cuando  $v$  excede un valor predeterminado

por la tensión de referencia  $V_{ref}$ , se cambia de 0 a 1, haciendo sonar eficazmente una alarma que, en este caso, podría indicar una temperatura excesiva. Los comparadores también se utilizan en muchos otros tipos de conversión analógico/digital.

La conversión digital/analógica generalmente se hace en exclusiva por hardware con CIs específicos para ello.

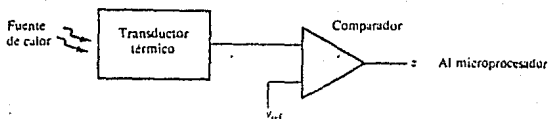


Figura 5.10 Comparador analógico.

La siguiente figura muestra un conversor DAC de resistencia ponderadas de 8 bits con transistores bipolares como conmutadores de entrada. Este tipo de DAC tiene la desventaja

de que cuando  $n$  es grande se requieren resistencias de precisión cuyos valores cubran un amplio rango. En consecuencia, - en la práctica se utiliza un tipo diferente de red resistiva - Comercialmente hay disponibles DACs adecuados para interconectar con la mayoría de tipos de microprocesador. El tamaño - usual de palabras de entrada es de 8 a 12 bits, con un tiempo de programación (tiempo de conversión digital-analógico) de -- 100 ns o más.

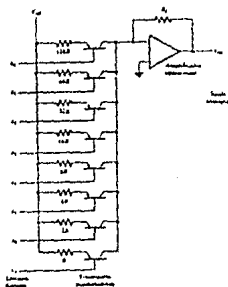


Fig. 5.11 DAC de 8 bits de resistencias ponderadas

Un método de conversión analógico/digital conceptualmen

te sencillo denominado conversión directa se muestra en la figura 5.12:

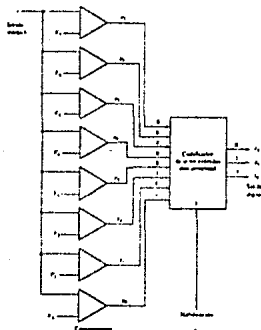


Figura 5.12 Conversión directa

En el método de conversión directa,  $V$  se compara simultáneamente con cada uno de los  $2^n$  niveles de tensión que son generados internamente por el ADC. Esta comparación se efectúa con  $2^n$  comparadores. La tensión de entrada se transforma en un número convencional de  $n$  bits por medio de un codificador de prioridad. La salida  $Z$  del codificador es la versión digital deseada de la entrada analógica  $V$ .

La conversión directa es una conversión analógico/digital relativamente rápida, consiguiéndose tiempos de conversión menores de 100 ns. A causa de que el número de comparadores - crece exponencialmente con  $n$ , el número de bits de salida, en conversión ADC directa, está limitado a  $n = 3$  o  $4$ . Por tanto, la precisión con que  $V$  puede medirse por este método es relativamente baja.

---

#### V.4.3.

#### CONVERSION DE DATOS

---

La conversión de datos se refiere al acoplo de las características físicas y lógicas de las señales de datos empleadas por el dispositivo de E/S a las empleadas por el bus del sistema. Esto incluye una conversión de señales entre las formas analógica y digital y la conversión entre el formato de transmisión de datos serie (bit a bit) y los formatos paralelo (palabra por palabra) usados por la mayoría de sistemas.

La sincronización se necesita para acomodar las diferentes velocidades operativas de nuestro sistema y los dispositivos de E/S. Esto usualmente requiere la inclusión en el circuito de interfase de una o más palabras de memoria intermedia.

Los circuitos de los dispositivos de E/S y del sistema -

funcionan independientemente en el sentido de que sus relojes-internos no están sincronizados uno con otro, por ello deben intercambiarse señales de control (listo, petición, reconocimiento, etc.), a través de los circuitos de interfase para iniciar o terminar las operaciones de E/S. La selección de dispositivo también implica intercambio de señales de control.

La figura 5.13 muestra la estructura del circuito de interfase necesario para interconectar un dispositivo de entrada específico, un teclado tipo teletipo, a un microprocesador de 8 bits. Aquí se supone que el teclado genera una única palabra de 8 bits (por ejemplo, un código ASCII) por cada tecla pulsada. También indica la disponibilidad de una nueva palabra activando la línea de control "dato listo". La conexión directa de las líneas de interfase del teclado con el bus del sistema no es posible a causa de que es imprevisible el instante de tiempo en el que las teclas son pulsadas y se produce un nuevo dato. Este tipo de circuitos de interfase tienen como objetivo, por tanto, actuar como memoria intermedia entre el teclado y el bus del microprocesador. Este microprocesador bien podría ser cualquier sistema digital tal como un controlador microprogramado.

La interfase del circuito anterior es básicamente un registro o conjunto de biestables de 8 bits, direccionales

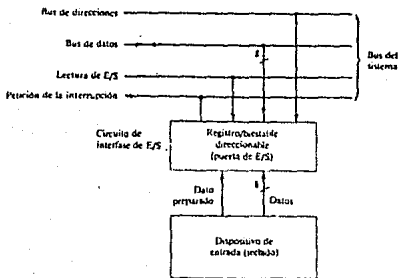


Figura 5.13 Interconexión de un teclado a un microprocesador.

con algunas señales de control lógico adicionales, y se denomina puerta de E/S. Realiza las siguientes funciones:

- 1.- Recibe y almacena palabras de datos, una a una, procedentes del teclado; cada nueva palabra de datos reemplaza a la precedente.



- 2.- Indica la presencia de un nuevo dato en la puerta de E/S enviando una señal de petición de interrupción a la CPU a través de una línea de control del bus del sistema.
- 3.- Responde a una orden de E/S procedente de la CPU transfiriendo su palabra de datos almacenada al bus de datos del sistema. Esto permite llevar el dato a ser leído por la CPU.

Observe que en este ejemplo la ejecución de un programa por la CPU, específicamente una instrucción de lectura a la puerta de E/S, controla la etapa final en la transferencia del dato a través de la interfase del microprocesador, especificando la puerta y el tipo de transferencia requerido.

Un problema de interfase más complicado se ilustra en la figura 5.14. Aquí el microprocesador se conecta a dispositivos de E/S remotos, con los que se comunica (a larga distancia) a través de una línea de teléfono, la última, por tanto, define la interfase de E/S local. La transmisión de datos a través de líneas telefónicas es en serie ya que únicamente se dispone de dos hilos, y se utilizan señales analógicas para presentar los datos sobre dicha línea. Normalmente se utiliza una señal portadora (señal eléctrica senoidal) que es modulada

para tener dos estados distintos que representen el 0 y el 1 lógicos, como muestra la siguiente figura.

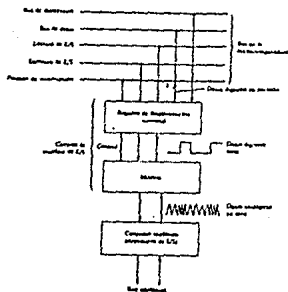


Figura 5.14 Interconexión de un dispositivo de E/S serie.

La modulación de la portadora se puede realizar en diferentes formas, una de las más comunes es la modulación en frecuencia (FSK). En líneas de teléfono convencionales, estas frecuencias pueden ser escuchadas como tonos de diferente agudeza. La transmisión de datos, por tanto, utiliza secuencias de zumbidos.

Un dispositivo electrónico denominado Modem (modulador-

demodulador) sirve para convertir los datos del formato de zumbidos al formato normal de pulsos digitales utilizado en las computadoras, y viceversa. La transmisión de datos a través del modem es en serie, mientras que la transmisión sobre el bus del sistema es en paralelo; de aquí que sea necesario un circuito que convierta los datos de tipo serie a paralelo durante las operaciones de entrada, y de paralelo a serie durante las operaciones de salida. Para realizar esta tarea puede adaptarse un registro de desplazamiento universal como puerta de E/S.

---

V.4.4.

---

TRANSDUCTORES

---

Los transductores tienen la función de convertir variables de entrada no eléctricas usualmente analógicas a variables eléctricas. Se conocen un gran número de métodos de transducción, muchos de los cuales son de gran ingenio. Por ejemplo al proceso de fabricación de cables en cierta compañía emplea un transductor ultrasónico que mide el grosor de la cubierta de aislante del cable por un proceso similar al utilizado en telemetría solar. La mayor parte de transductores son sensores. Muchos utilizan resistencias cuyo valor es alterado por la variación del parámetro de interés. Una muestra repre-

sentativa de transductores basados en resistencias se mues --  
tra en la siguiente figura:

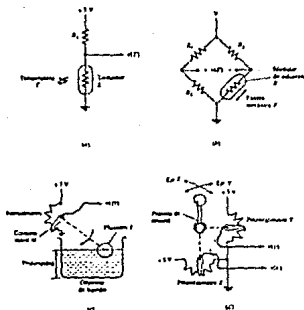


Fig. 5.15 Transductores basados en resistencias:  
a) termistor; b) medidor de esfuerzos-  
c) medidor de nivel; d) palanca de man-  
do.

Un termistor es una resistencia sensible a la temperatura, se fundamenta en el hecho de que la resistencia de la mayoría de materiales, semiconductores se decreta con la temperatura. La tensión de salida del circuito de la figura anterior es aproximadamente proporcional a la temperatura del --

termistor. La relación entre la resistencia y la temperatura puede determinarse a partir de las tablas de datos publicadas por los fabricantes de estos dispositivos.

Otro transductor importante basado en resistencias es el llamado medidor de esfuerzos. Consta de una lámina o hilo metálico, frecuentemente plegado en forma de zig zag, que es unido firmemente a la superficie objeto de medida de la tensión mecánica. La tensión mecánica provoca que el medidor de esfuerzos se expanda o contraiga, de modo que disminuye o aumenta su resistencia eléctrica. El cambio en la resistencia es muy pequeño, pero puede ser detectado por un circuito puente.

Esta configuración puente puede también utilizarse para compensar cambios indeseables en la resistencia del medidor de esfuerzos originados por la fluctuación en la temperatura. Muchas variables mecánicas incluyendo la presión, velocidad, aceleración y peso están asociadas con fuerzas que comprimen, o expanden los cuerpos sobre los que actúan las fuerzas.

Estas variables pueden ser cuantificadas por medidores de esfuerzos muy sensibles.

Algunos transductores utilizan resistencias variables o potenciómetros cuya resistencia efectiva puede alterarse des -

plazando un contacto móvil W a lo largo de la superficie de - - una resistencia lineal o circular. La resistencia R entre W - y un extremo T de la resistencia es proporcional a la distancia física D entre W y T; de aquí que R puede servir como una medida de D. Cuando se utiliza una resistencia circular, W se diseña de forma que gire alrededor del centro de la resistencia, y R también mide entonces posiciones angulares.

Los transductores mostrados en la figura anterior, en los incisos (c) y (d), pertenecen a este grupo particular de transductores ya que transforman la posición de un flotador o de una palanca de mando, en una correspondiente resistencia, y por tanto, tensión.

Muy diversos transductores reciben información del mundo exterior en forma de señales luminosas variables en el tiempo, siendo convertidas por ellos en señales eléctricas. Varios ejemplos de transductores de este tipo ya han sido citados en secciones anteriores (V.3) y son denominados como transductores ópticos.

---

#### V.4.5.

#### ACTUADORES

---

En muchas aplicaciones un sistema digital controla variables mecánicas tales como la posición u orientación de un -

objeto. Esto requiere de un dispositivo de salida, generalmente llamado actuador, que pueda transformar las señales electrónicas de control provenientes del sistema, en movimientos mecánicos.

Un relevador puede considerarse como un tipo sencillo de actuador que controla la posición de un conmutador mecánico.

Movimientos más complicados se controlan con dispositivos electromecánicos más potentes, de los que los solenoides y motores son los más importantes. El sistema de control de un proceso industrial puede utilizar, por ejemplo, tres motores eléctricos: dos para controlar la posición espacial de algún objeto y el tercero para controlar su velocidad de desplazamiento.

Un solenoide es un tipo de actuador más sencillo, que similarmente a un relevador, es un tipo de electroimán. Contiene una bobina cilíndrica de hilo esmaltado que se activa cuando pasa corriente a través de él. Esto crea un fuerte campo magnético en la cavidad del solenoide, que ejerce una fuerza atractiva sobre un émbolo deslizante de hierro dulce. De esta forma, al solenoide, cuando se activa, absorbe hacia su interior el émbolo; cuando se desactiva, por conmutarse a corte la corriente que fluye a través de la bobina, un muelle de -

recuperación hace que el émbolo salga del interior de la bobina.

La siguiente figura muestra un solenoide utilizado para actuar como cerradura de una puerta. El movimiento del émbolo del solenoide se transmite al pestillo deslizante por medio de una palanca. También puede utilizarse un solenoide para producir un movimiento de rotación con ayuda de un trinquete. Generalmente los movimientos de rotación, así como movimientos lineales grandes se afectan con ayuda de motores eléctricos.

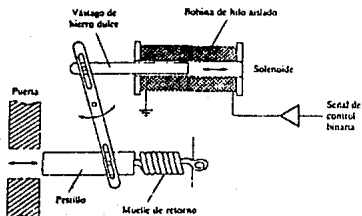


Figura 5.16 Actuador solenoide utilizado en una cerradura.



Los motores eléctricos se utilizan para controlar formas de movimiento más complicados. La Figura 5.17 muestra un motor que controla el movimiento hacia adelante y hacia atrás de una herramienta a lo largo de un camino lineal.

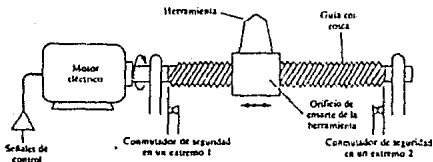


Fig. 5.17 Utilización de un motor eléctrico para controlar el movimiento de una herramienta.

El movimiento de rotación del eje del motor se convierte en movimiento lineal por medio de un eje roscado (una guía con rosca), en el que se acopla (como una tuerca) un orificio de la herramienta. La dirección del movimiento de la herramienta puede cambiarse invirtiendo la dirección de movimiento del motor eléctrico; esto último usualmente se efectúa invirtiendo

la dirección del flujo de corriente eléctrica que atraviesa el motor. Se utilizan conmutadores de seguridad en los extremos de la barra con rosca para desconectar el motor si el cabezal de la herramienta llega a ellos. Pueden utilizarse otros conmutadores para detener la herramienta en posiciones intermedias, en esta aplicación son particularmente útiles los transductores ópticos o magnéticos.

Hay disponibles una gran variedad de diferentes tipos de motores eléctricos. Todos están basados en la conversión de corriente eléctrica en campos magnéticos distribuidos de forma que se crean fuerzas magnéticas atractivas o repulsivas que producen la rotación. Un motor eléctrico tiene dos partes principales: un rotor cilíndrico que gira y un estator fijo que rodea al rotor. El rotor y/o estator usualmente están equipados con conjuntos de bobinas o devanados, que producen los campos magnéticos necesarios para que funcione el motor. Los motores eléctricos pueden diseñarse para funcionar con alimentación de potencia AC o DC. Pueden además ser caracterizados por sus velocidades de funcionamiento, su potencia y su par de fuerza de torsión características.

Un tipo especial de motor electrónico que se adapta bien a muchas tareas de control de posición es el motor de pasos. A diferencia de un motor convencional, cuyo único modo de funcionamiento es de rotación continua, un motor de pasos puede -

## "INTERFASES"

girar tan solo un ángulo o paso pequeño y preciso y después -  
pararse. Este movimiento o paso se controla por un pulso (di-  
gital) de corriente eléctrica suministrada a los arrollamien-  
tos del motor. N de tales pulsos en secuencia provocan que el  
motor gire n pasos y se detenga en una posición predetermina-  
da; una secuencia continua de pulsos produce una rotación con-  
tínua.

El tamaño del paso está predeterminado por el diseño del  
motor, y puede estar en el rango de 1.8 a 30 grados; éstos es,  
de 200 a 12 pasos por revolución.

---

CAPITULO VI

---

" PAQUETE DE PRACTICAS "

---

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO  
E.N.E.P. ARAGON INGENIERIA  
LABORATORIO DE CIRCUITOS LOGICOS  
PRACTICAS DE CONTROLADORES DIGITALES

---

PRACTICA NUMERO: 1 CONTROLADORES CABLEADOS CON F. F. J-K

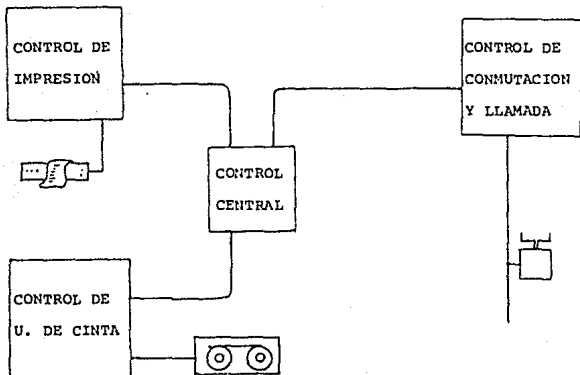
---

**OBJETIVOS:** El alumno desarrollará en base a un problema real que le sea planteado, los - diagramas de flujo que determinarán el comportamiento de un Controlador Digi - tal.

El alumno construirá su controlador - mediante Flip-Flops tipo J-K.

INTRODUCCION

Durante el presente Laboratorio de Circuitos Lógicos - intentaremos diseñar un sistema de contestación automática de - línea telefónica compuesto de los siguientes bloques:



En esta práctica diseñaremos el control central mediante Flip-Flops tipo J-K como un controlador cableado. Por tanto, comenzaremos definiendo las funciones de este controlador:

- Recibe información de las llamadas que llegan desde el control de conmutación y llamada. Es decir, el control de conmutación y llamada "avisa" al control central cuando llega alguna llamada.
- Cuando el sistema está habilitado para ello, en cuanto llegue una llamada el controlador central ordenará "descolgar el teléfono" al control de conmutación y

## "PAQUETE DE PRACTICAS"

- activará al control de cinta para emitir el mensaje - del dueño y grabar el mensaje recibido hasta que el - control de conmutación detecte y avise de que se ha concluido la comunicación en cuyo caso el control - ordenará "colgar" el teléfono.
- Cuando se ha concluido la comunicación y se ha dejado de grabar, el control central ordenará al control de impresión imprimir la hora y fecha actuales, además-- de el tiempo de duración de la llamada.
  - Si el sistema no está habilitado para contestar inmediatamente, el controlador central esperará a que el control de conmutación le indique la sucesión de cinco "RING's" (esperará a que suene cinco veces el teléfono), y si no contesta el dueño, se inicia el -- procedimiento de contestación automática.
  - El proceso de contestación automática (descolgar, activar control de cinta, grabar, imprimir, etc.) puede - ser interrumpido en cualquier momento por el dueño , - en cuanto éste descuelgue la bocina de su teléfono.

Los puntos anteriores nos indican el comportamiento del sistema, y muy específicamente el del controlador central. Los-

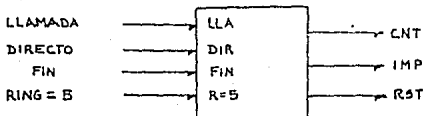
módulos llamados control de conmutación, control de cinta y --- control de impresión serán diseñados en las posteriores prácticas.

### A C T I V I D A D E S

1. Determine el diagrama de "caja negra" del controlador central descrito anteriormente.
2. Elabore el diagrama de flujo de este controlador.
3. Diseñe su controlador mediante Flip-Flops tipo J-K, como un controlador cableado. Muestre su funcionamiento al instructor.

### SOLUCION

1. El diagrama de "caja negra" del controlador central involucra solamente sus entradas y salidas.

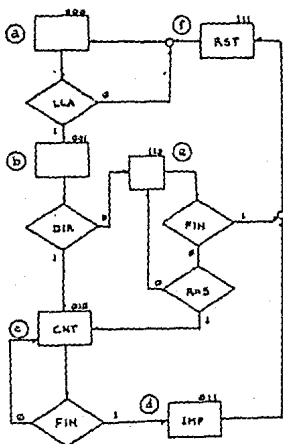




La entrada RING=5 procede de un contador de ring's incluido en el control de conmutación y llamada. Indica que ya ha sonado cinco veces el teléfono.

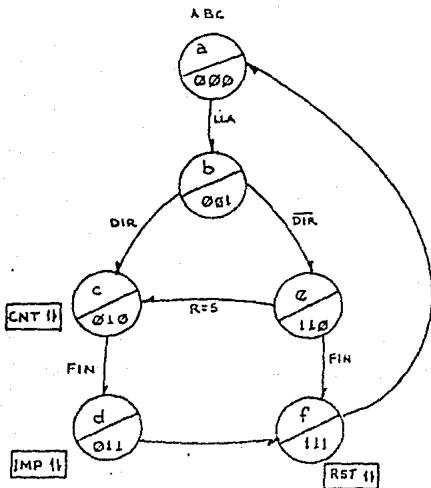
La entrada DIRECTO procede de un botón que habilita al sistema para contestar inmediatamente una llamada. La entrada LLAMADA proviene del control de conmutación, avisando la llegada de una llamada. La entrada FIN indica la finalización de la comunicación.

2. La carta ASM del controlador central puede ser la siguiente:



3. El diseño de un controlador cableado con Flip-Flops J-K se realiza siguiendo los pasos que a continuación se presentan:

a) Definir el diagrama de flujo del autómata como un circuito de Mealy, de acuerdo con el diagrama de flujo del punto anterior.



b) Tabla de estados presentes:

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

c) Mapa de estados futuros:

c \ AB	00	01	11	10
0	LLA $\rightarrow$ b 001	FIN $\rightarrow$ d 011	R=5 $\rightarrow$ c 010 FIN $\rightarrow$ f 111	a 000
1	DIR $\rightarrow$ c 010 DIR $\rightarrow$ e 110	f 111	a 000	a 000

d) Tablas de entrada para cada Flip-Flop :

J	K	$Q_{t+1}$
0	0	$Q_t$
0	1	0
1	0	1
1	1	$\bar{Q}_t$

$Q_t$	$Q_{t+1}$	J	K
0	0	0	*
0	1	1	*
1	0	*	1
1	1	*	0

C \ AB		00	01	11	10
		0	0	*	*
1		1	*	*	

$$J_1 = (\bar{C}A)C + BC + A$$

C \ AB		00	01	11	10
		0	0	*	*
1		1	*	*	0

$$J_2 = \bar{A}C$$

C \ AB		00	01	10	11
		0	11A	FIN	FIN
1		*	*	*	*

$$J_3 = (11A)\bar{A}\bar{B} + (FIN)B + C$$

C \ AB		00	01	11	10
		0	*	*	1
1		*	*	1	1

$$K_1 = (C=5)(FIN) + \bar{A} + \bar{B} + C$$

C \ AB		00	01	11	10
		0	*	0	*
1		*	0	1	*

$$K_2 = AC$$

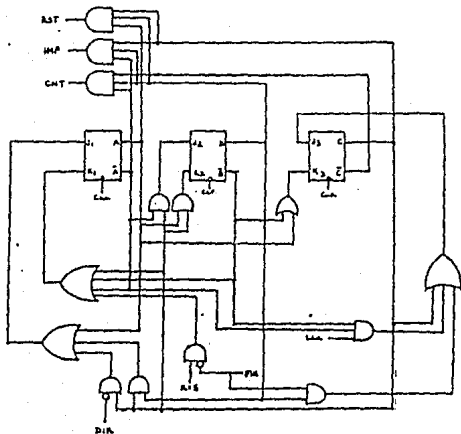
C \ AB		00	01	11	10
		0	*	*	*
1		1	0	1	1

$$K_3 = A + \bar{B}$$

e) Tabla de salidas:

C \ AB		00	01	11	10
		0	-	CNT	-
1		-	IMP	RST	-

f) Construcción



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO  
E.N.E.P. ARAGON INGENIERIA.  
LABORATORIO DE CIRCUITOS LOGICOS  
PRACTICAS DE CONTROLADORES DIGITALES

---

PRACTICA NUMERO:2 CONTROLADORES CABLEADOS CON F.F. TIPO D

---

OBJETIVOS:

El alumno desarrollará en base a un problema real que le sea planteado, todos los pasos que se suceden en el diseño de un controlador digital.

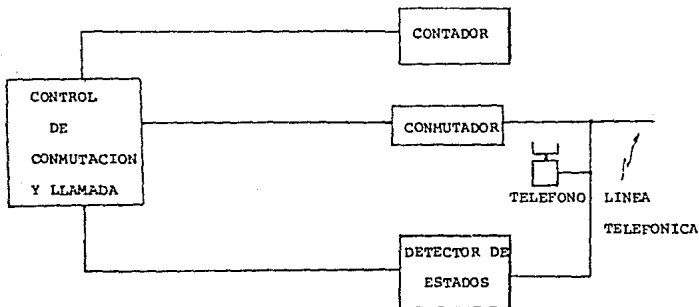
El alumno construirá su diseño con flip-flops tipo D, cableado mediante el método de multiplexores directamente -- relacionados.

INTRODUCCION

En la práctica anterior se realizó el planteamiento global del sistema de controladores que diseñaremos. En esta práctica nos enfocaremos en el desarrollo de un controlador de con-

"PAQUETE DE PRACTICAS"

mutación que contará con los siguientes bloques auxiliares:



Este controlador será diseñado mediante flip-flops tipo D y alambrado mediante el método de multiplexores directamente-relacionados por lo que vamos a comenzar definiendo las funciones que debe realizar este controlador:

- Avisa al control central cuando se recibe una llamada por la línea telefónica. La detección de la llamada la realizará el bloque auxiliar denominado Detector de Estados.

- Mediante el bloque denominado como Contador, se contarán las veces que suena el teléfono avisando al control cuando ya se hayan detectado cinco "Ring's" sin que el dueño del aparato conteste.
  
- Mediante el bloque auxiliar denominado Conmutador,-- se efectuarán las instrucciones recibidas del control central consistentes en colgar y descolgar la extensión telefónica.
  
- El Detector de Estados podrá detectar en cualquier momento cuando se presente una finalización del proceso de contestación automática, ya sea por que el dueño del aparato conteste la llamada levantando el auricular (cuando la llamada ya había sido contestada por el sistema) o porque la persona que llama cuelgue su bocina concluyendo con esto su mensaje.

#### ACTIVIDADES

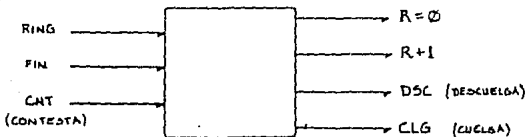
1. Determine el diagrama de "caja negra" del controlador de conmutación y llamada.
  
2. Elabore el diagrama de flujo de este controlador.



3. Diseñe su controlador mediante Flip-Flops tipo D, utilizando la técnica de multiplexores directamente relacionados. Muestre su funcionamiento a su instructor.

SOLUCION

1. El diagrama de caja negra de este controlador mostrará las entradas y salidas del sistema:

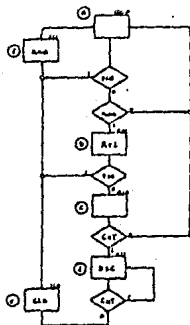


La entrada RING proviene del módulo auxiliar llamado de tector de estados, e indica que se está detectando una llamada y que el teléfono está colgado y sonando. La entrada CONTESTA - proviene del control central indicándole a este controlador que active su salida DESCUELGA hacia el módulo auxiliar denominado conmutador.

La entrada FIN proviene del módulo auxiliar llamado detector de estados, e indica que se ha detectado que la persona que llamó a nuestro teléfono en cuestión ha colgado su auricular dando por finalizado su mensaje. La misma entrada FIN indica que el dueño del aparato o usuario del sistema ha contestado personalmente la llamada. En ambos casos, la señal FIN indica la finalización del proceso de contestación automática.

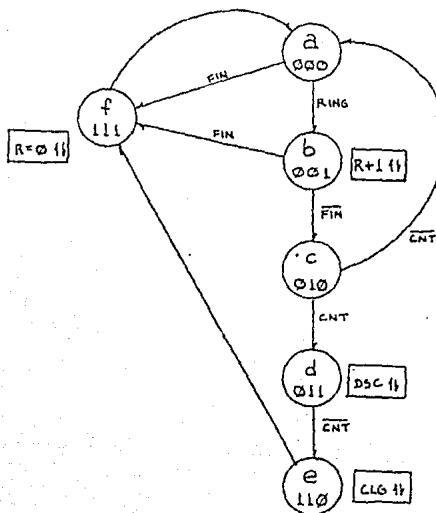
La salida R=0 se aplicará al contador de Ring's para borrar su contenido y ponerlo en cero, la salida R=R+1 será aplicada al mismo contador para incrementar en uno su contenido y así contar cuántas veces ha sonado nuestro teléfono. La entrada al controlador central denominada en la práctica No. 1 como RING=5 se tomará directamente del contador mediante compuertas lógicas.

2.- El diagrama de flujo del controlador de conmutación puede ser descrito como sigue:



3. El diseño de un controlador cableado con Flip-Flops tipo D se realiza mediante los siguientes pasos:

a) Definir el diagrama de flujo del autómata como un circuito de Mealy de acuerdo con el diagrama ASM anterior.



b) Tabla de estados presentes:

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

c) Mapa de estados futuros:

		AB			
		00	01	11	10
C	0	RING → b 001	CNT → d 011	f 111	a 000
	1	FIN → f 111	$\overline{\text{CNT}} \rightarrow a$ 000	a 000	a 000
		FIN → f 111	$\overline{\text{CNT}} \rightarrow a$ 110	a 000	a 000
		FIN → c 010			

"PAQUETE DE PRACTICAS"

d) Tablas de entrada para Flip-Flop:

AB \ C	00	01	11	10
0	FIN	0	1	0
1	FIN	$\overline{\text{CNT}}$	0	0

$D_1 \rightarrow A$

AB \ C	00	01	11	10
0	FIN	CNT	1	0
1	1	1	0	0

$D_2 \rightarrow B$

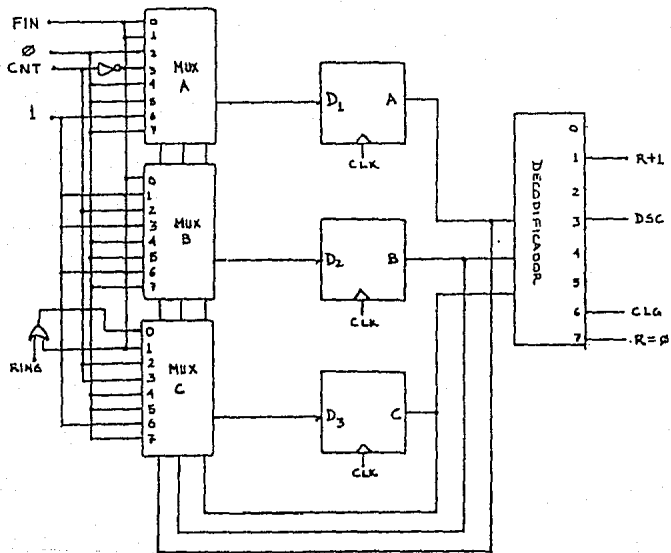
AB \ C	00	01	11	10
0	RING + FIN	CNT	1	0
1	FIN	CNT	0	0

$D_3 \rightarrow C$

e) Tabla de salidas:

AB \ C	00	01	11	10
0	-	-	CLG	-
1	R+1	DSC	R=0	-

f) Construcción



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

E.N.E.P. ARAGON

INGENIERIA

LABORATORIO DE CIRCUITOS LOGICOS

PRACTICAS DE CONTROLADORES DIGITALES

---

PRACTICA NUMERO: 3      CONTROLADORES CON REGISTROS DE CORRIMIENTO

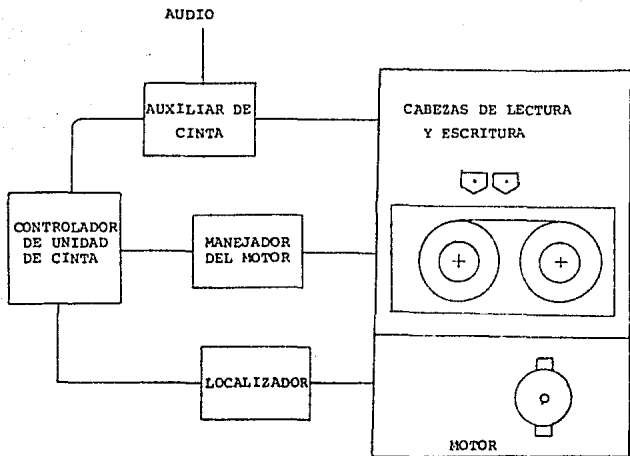
---

OBJETIVO:

El alumno desarrollará un controlador -  
cableado y aplicará el método de regis-  
tros de corrimiento en su construcción-

#### INTRODUCCION

Siguiendo el planteamiento presentado en la práctica No 1, en la presente práctica nos enfocaremos en el diseño de un controlador de unidad de cinta para el sistema de contestación-automática. Nuestro controlador contará con los siguientes bloques auxiliares:



Para poder diseñar este controlador mediante registros de corrimiento, se definirán a continuación las funciones que tiene que realizar.

- Este controlador iniciará sus procedimientos cuando reciba del control central la orden de contestar.
- Mediante los bloques auxiliar de cinta y manejador de motor, se reproducirá el mensaje del usuario para que sea escuchado por la persona que genera la llama-



da

- Mediante el bloque auxiliar denominado localizador, - se determinará la porción de cinta que queda libre pa ra ser grabada y se posicionará al principio de ella- mediante el manejador de motor. A este proceso le -- llamaremos localización de la siguiente zona de graba ción (LSZG).
  
- Una vez realizada la LSZG, se emitirá un zumbido y se procederá a la grabación del mensaje la cual podrá ser detenida en cuanto al usuario conteste personal mente o la persona que llama cuelgue el teléfono.
  
- Cuando el proceso de grabación llegue a su fin, se -- regresará la cinta hasta su principio donde estará - el mensaje del usuario, preparando el sistema para el siguiente proceso de contestación.
  
- Simultáneamente con el regreso de cinta, se actualiza rá la LSZG por medio del módulo llamado localizador.

Para comprender mejor la interrelación entre los blo- - ques auxiliares y el controlador, detallaremos un poco más las- funciones de cada bloque:

- Auxiliar de cinta: Recibe señales binarias que indican si se trata de un proceso de lectura o escritura. Actuará sobre las cabezas de lectura/escritura y será el camino de audio desde el teléfono hacia la cinta.
  
- Manejador del motor: Recibe señales binarias indicando velocidad de operación (alta o baja), movimiento del motor (sí o no), y dirección del giro (adelante o atrás). Actuará directamente sobre el motor de acuerdo con estas entradas.
  
- Localizador: Podrá determinar cuál es la siguiente zona de grabación avisando cuando se llegue a ella para emitir un zumbido. Además, podrá actualizar la LSZG cada que se termine de grabar un mensaje. Además, este bloque podrá determinar cuando se haya llegado al principio o al final de la cinta.

#### ACTIVIDADES

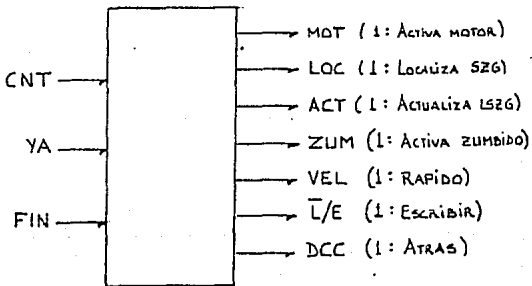
1. Determine el diagrama de "caja negra" del controlador de cinta descrito anteriormente.
  
2. Elabore el diagrama de flujo de este controlador (carta

ASM).

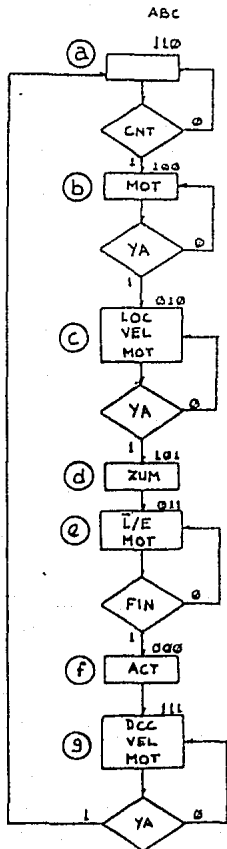
3. Diseñe su controlador mediante registros de corrimiento.
4. Muestre el funcionamiento de su circuito a su instructor.

SOLUCION

- 1.- Diagrama de caja negra:

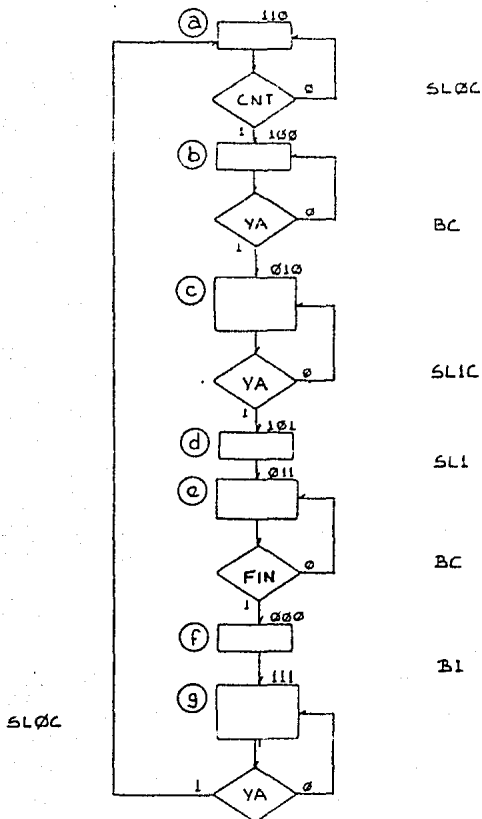


2.- CARTA ASM:



"PAQUETE DE PRACTICAS"

- a) Observaremos las instrucciones necesarias para pasar de un estado a otro:



b) Mapa de estados presentes:

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

b) Mapa de instrucciones:

	AB	00	01	11	10
c					
0		BI	SLIC	SL0C	BC
1		*	BC	SL0C	SLI

c) Mapas de Acción:

74194

S <sub>0</sub>	S <sub>1</sub>	Acción
0	0	HOLD
0	1	SL
1	0	SR
1	1	BR

	AB	00	01	11	10
c					
0		I	0	0	YA
1		*	FIN	0	0

S<sub>0</sub>

	AB	00	01	11	10
c					
0		I	YA	CNT	YA
1		*	FIN	YA	I

S<sub>1</sub>

d) Mapas de carga en paralelo:

	AB	00	01	11	10
C		00	01	11	10
0		1	x	x	1
1		x	0	x	x

$$A_{CARGA} = \bar{C}$$

	AB	00	01	11	10
C		00	01	11	10
0		1	x	x	0
1		x	0	x	x

$$B_{CARGA} = \bar{A}\bar{B}$$

	AB	00	01	11	10
C		00	01	11	10
0		1	x	x	0
1		x	0	x	x

$$C_{CARGA} = \bar{A}\bar{B}$$

e) Mapas de carga en serie:

	AB	00	01	11	10
C		00	01	11	10
0		x	1	0	x
1		x	x	0	x

$$SL = \bar{A} + \bar{B}$$

	AB	00	01	11	10
C		00	01	11	10
0		x	x	x	x
1		x	x	x	x

$$SR = 1$$

f) Salidas:

	AB	00	01	11	10
C		00	01	11	10
0		ACT	LOC	-	MOT
1		-	$\bar{L}/E$	DCC	ZUM

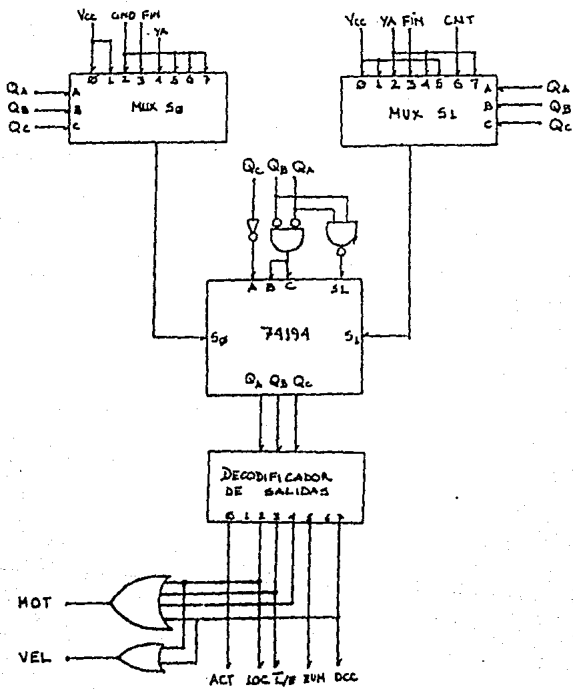
	AB	00	01	11	10
C		00	01	11	10
0		0	1	0	1
1		0	1	1	0

MOT

	AB	00	01	11	10
C		00	01	11	10
0		0	1	0	0
1		0	0	1	0

VEL

j) Construcción





"PAQUETE DE PRACTICAS"

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO  
E.N.E.P. ARAGON INGENIERIA  
LABORATORIO DE CIRCUITOS LOGICOS Y  
COMPUTADORAS

---

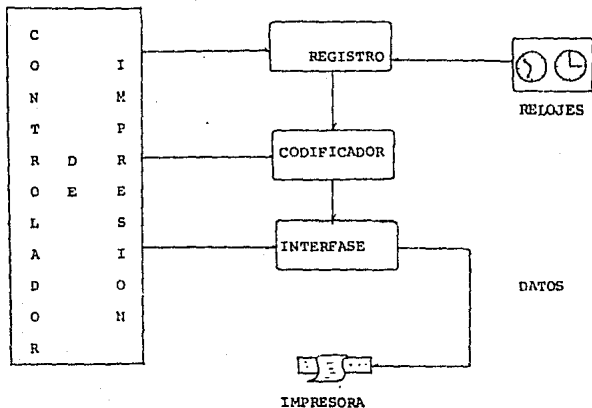
PRACTICA NUMERO: 4 CONTROLADORES CON CONTADORES

---

Objetivo: El alumno desarrollará un controlador cableado y aplicará el método de contadores en su construcción.

INTRODUCCION

En la primera práctica se realizó el planteamiento global de el sistema de controladores que diseñaremos. En esta práctica nos enfocaremos en el desarrollo de un controlador de impresión que contará con los siguientes bloques auxiliares:



a) Este controlador se activa cuando el controlador -- central le ordena imprimir. Consultará hora y fecha actuales más el tiempo de duración del mensaje recibido. Estos datos serán preparados para ser impresos en un periférico conectado a este control por medio de una interfase.

b) Los bloques auxiliares de este control serán los relojes, registros de almacenamiento temporal, codificadores para presentar en código ASCII y la interfase de salida para la impresora.

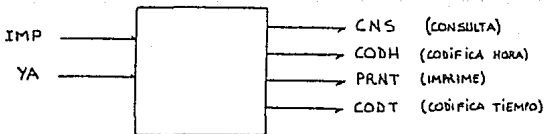
- c) Los registros y el codificador ayudarán a presentar los datos de manera adecuada a la interfase de salida.

ACTIVIDADES

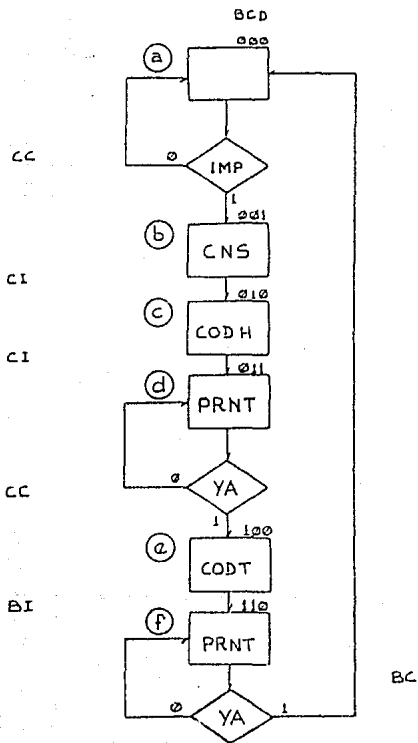
1. Determine el Diagrama de "caja negra" de este controlador.
2. Elabore los diagramas de flujo correspondientes (carta-ASM).
3. Diseñe mediante un contador.
4. Muestre su funcionamiento al instructor.

SOLUCION

1. Diagrama de Bloque :



2.- Carta ASM.



"PAQUETE DE PRACTICAS"

a) Mapa de estados presentes y Mapa auxiliar de Acciones.

	BC	00	01	11	10
D	0	a	c	f	e
	1	b	d	*	*

	BC	00	01	11	10
D	0	CC	CI	BC	BI
	1	CI	CC	*	*

b) Mapas de acción.

74161

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

	BC	00	01	11	10
D	0	IMP	1	0	1
	1	1	YA	*	*

ENP

	BC	00	01	11	10
D	0	0	0	YA	1
	1	0	0	*	*

LOAD

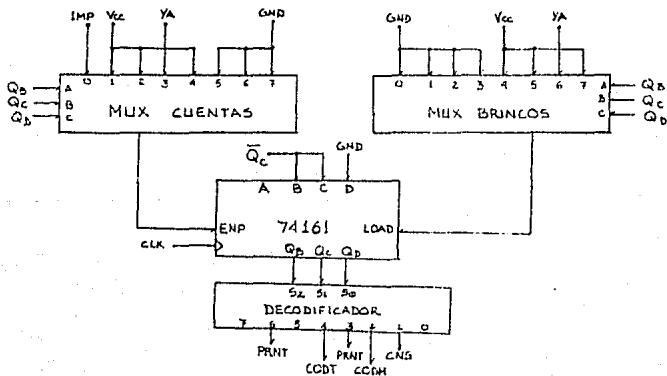
c) Mapas de Carga en Paralelo.

	BC	00	01	11	10	
D	0	x	x	0	1	$B_{CARGA} = \bar{C}$
	1	x	x	x	x	

	BC	00	01	11	10	
D	0	x	x	0	1	$C_{CARGA} = \bar{C}$
	1	x	x	x	x	

	BC	00	01	11	10	
D	0	x	x	0	0	$D_{CARGA} = 0$
	1	x	x	x	x	

d) Construcción.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO  
E.N.E.P. ARAGON                      INGENIERIA  
LABORATORIO DE CIRCUITOS LOGICOS  
PRACTICAS DE CONTROLADORES DIGITALES

---

PRACTICA NUMERO: 5                      CONTROLADOR MICROPROGRAMADO

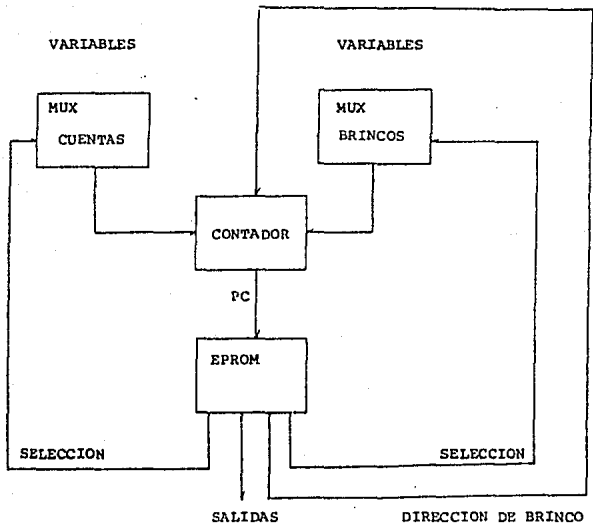
---

OBJETIVO:                      En base a un controlador con contador, el alumno podrá diseñar un controlador micro programado .

INTRODUCCION

En la práctica anterior se diseñó un controlador basado en un contador. En la presente práctica diseñaremos el mismo controlador en su versión microprogramada utilizando las mismas instrucciones del contador de la práctica 4.

La estructura es la siguiente



Las variables de entrada serán las mismas del controlador anterior.

#### ACTIVIDADES

1. Repita la carta ASM del controlador de impresión de la práctica anterior.
2. En base al punto anterior, elabore el programa que debe

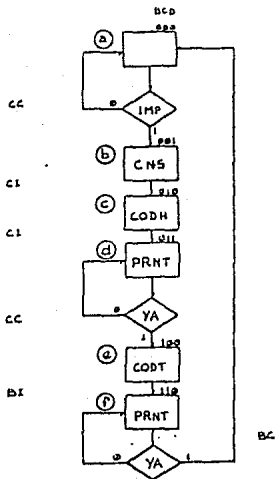


"PAQUETE DE PRACTICAS"

rá gobernar al controlador microprogramado en binario - o hexadecimal.

3. Elabore el diagrama del sistema y construya presentando su funcionamiento al instructor.

1.- Carta ASM.



2.- PROGRAMA

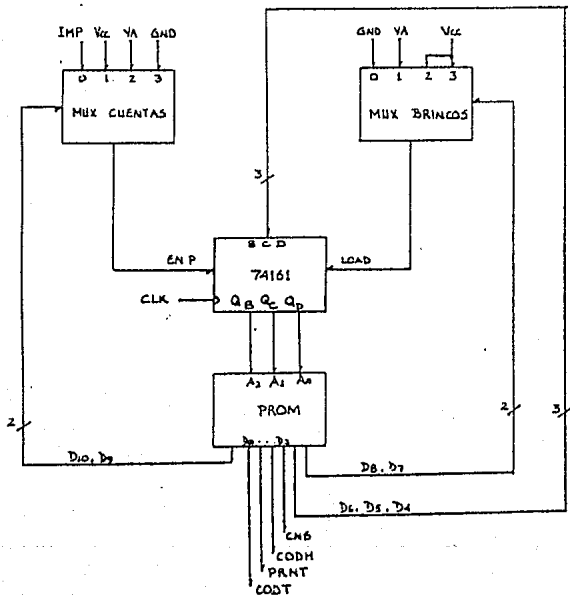
ESTADO	INSTRUCCION	ENP	LOAD
a	CC (IMP)	VAR	Ø
b	CI	1	Ø
c	CI	1	Ø
d	CC (YA)	VAR	Ø
e	BI <110>	1	1
f	BC (YA) <000>	Ø	VAR

De acuerdo con la tabla anterior se obtiene el siguiente programa:

DIR PROM	DIR MUX CUENTAS	DIR MUX BRINCOS	DIR BRINCOS	SALIDAS				HEXA
				CNS	COOH	PRNT	COOT	
000	00	00	000	0	0	0	0	F800
001	01	00	000	1	0	0	0	FA08
010	01	00	000	0	1	0	0	FA04
011	10	00	000	0	0	1	0	FC02
100	01	11	110	0	0	0	1	FB01
101	01	11	000	0	0	0	0	FBB0
110	11	01	000	0	0	1	0	FEB2

$D_{15} \dots D_0$      $D_8$   $D_7$      $D_6$   $D_5$   $D_4$      $D_3$      $D_2$      $D_1$      $D_0$

3.- Construcción



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

E.N.E.P. ARAGON

INGENIERIA

LABORATORIO DE CIRCUITOS LOGICOS

PRACTICAS DE CONTROLADORES DIGITALES

---

PRACTICA NUMERO: 6	CONTROLADOR	MYCA	I
--------------------	-------------	------	---

---

OBJETIVOS: El alumno diseñará un módulo de controlador de tipo MYCA I.

El alumno será capaz de aplicar el módulo-MYCA I en la construcción de un controlador planteado.

#### INTRODUCCION

Un módulo MYCA I es un bloque encargado de "manejar" a una memoria de programa; es una versión mejorada de contador utilizado en el controlador microprogramado de la práctica anterior.

El bloque MYCA I recibe como entradas un opcode que le indicará la instrucción a ejecutar/instrucción del contador), una bandera que representa a las variables de entrada y una di

rección de brinco procedente de la memoria de programa al --  
igual que el opcode.

La siguiente figura muestra el objetivo de este diseño:

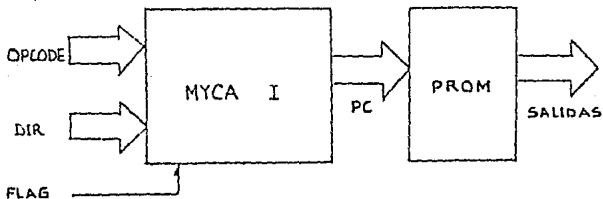


TABLA DE DISEÑO

OPCODE	NEMONICO	FLAG	ACCION	DESCRIPCION
0 0 0	CC (FLAG)	0 1	RETIENE: PC $\leftarrow$ PC CUENTA: PC $\leftarrow$ PC + 1	CUENTA CONDICIONAL
0 0 1	CI	0 1	CUENTA: PC $\leftarrow$ PC + 1 CUENTA: PC $\leftarrow$ PC + 1	CUENTA INCONDICIONAL
0 1 0	BC (FLAG) (DIR)	0 1	RETIENE: PC $\leftarrow$ PC BRINCA: PC $\leftarrow$ DIR	BRINCO CONDICIONAL
0 1 1	BI (DIR)	0 1	BRINCA: PC $\leftarrow$ DIR BRINCA: PC $\leftarrow$ DIR	BRINCO INCONDICIONAL
1 0 0	C/BC (FLAG) (DIR)	0 1	CUENTA: PC $\leftarrow$ PC + 1 BRINCA: PC $\leftarrow$ DIR	CUENTA/BRINCO CONDICIONAL

Se diseñará MYCA I en función de OPCODE, de acuerdo a -  
la Tabla de Mando del 74161; considerando además, que el OPCO  
DE será la entrada de selección para los MUX ENP y MUX LOAD --  
simultáneamente.

#### ACTIVIDADES

1. Basado en el contador 74161, diseñe el módulo MYCA I --  
presentando los pasos y tablas pertinentes.
2. Construya el módulo MYCA I diseñado en paso anterior -  
y muestre su funcionamiento al instructor.
3. Presente un diagrama del módulo diseñado y de la manera  
de conectarse a una memoria de programa.
4. Basado en el controlador de conmutación de la práctica-  
No. 2 presente la carta ASM y el programa que resuelva-  
este controlador.
5. Construya su diseño mostrando su funcionamiento al ins-  
tructor.

SOLUCION:

1.- De acuerdo a la tabla de diseño:

	OPCODE	FLAG	ENP	LOAD
<u>CC</u>	000	0	0	0
<u>CI</u>	001	0	1	0
<u>BC</u>	010	0	0	1
<u>BI</u>	011	0	1	1
<u>C/BC</u>	100	0	1	0

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

C B A

Mapas para ENP Y LOAD

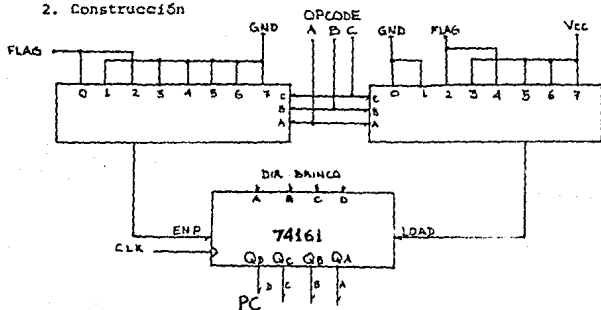
A	CB	00	01	11	10
1		FLAG	FLAG	1	1
0		1	1	1	1

ENP

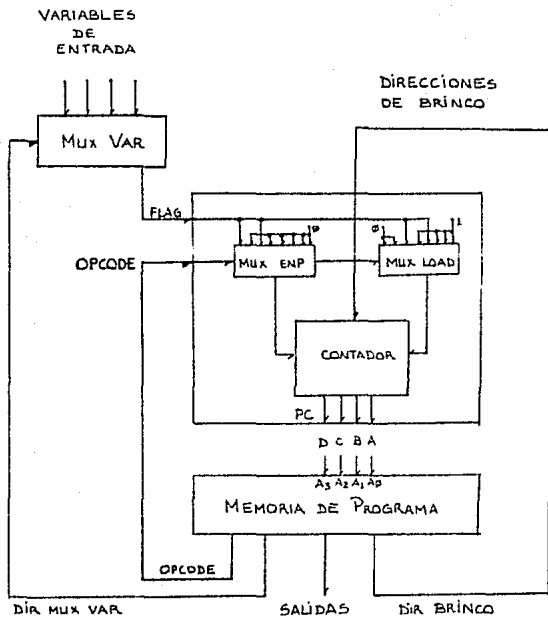
A	CB	00	01	11	10
1		0	FLAG	1	FLAG
0		0	1	1	1

LOAD

2. Construcción



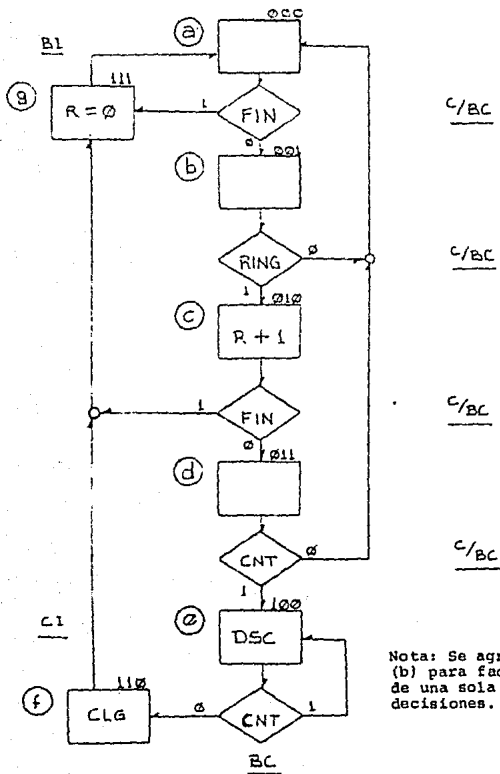
### 3. ARQUITECTURA DE TRABAJO NYCA I



En el programa irán: (Estados), códigos de operación, - direcciones de brincos y salidas.



4.- Carta ASM del controlador de conmutación.



Nota: Se agrega el estado - (b) para facilitar el uso - de una sola variable en las decisiones.

TABLA MEMORICA

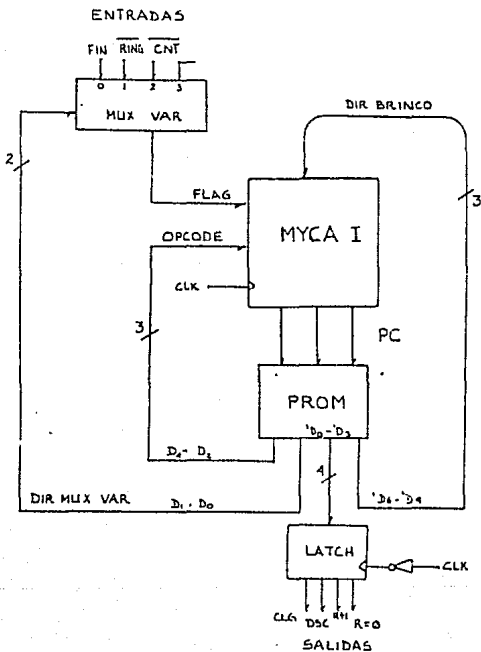
ESTADO	INSTRUCCION	
a	000	C/BC (FIN) <111>
b	001	C/BC (RING) <000>
c	010	C/BC (FIN) <111>
d	011	C/BC (CNT) <000>
e	100	BC (CNT) <110>
f	110	CI
g	111	B1 <000>

PROGRAMA SOLUCION:

DIR	OPCODE	DIR MUX VAR	DIR BRINCOS	SALIDAS				HEXADECIMAL
				R=0	R+1	DSC	CLG	
0	100	00	1 1 1	0	0	0	0	10 70
1	100	01	0 0 0	0	0	0	0	11 00
2	100	00	1 1 1	0	1	0	0	10 74
3	100	10	0 0 0	0	0	0	0	12 00
4	010	00	1 1 0	0	0	1	0	08 62
5	000	00	0 0 0	0	0	0	0	00 00
6	001	00	0 0 0	0	0	0	1	04 01
7	011	00	0 0 0	1	0	0	0	0C 08

$d_4$   $d_3$   $d_2$  |  $d_1$   $d_0$  |  $d_4$   $d_3$   $d_2$  |  $d_1$   $d_0$  |  $d_4$   $d_3$   $d_2$  |  $d_1$   $d_0$   
 PROM 1 ← PROM 2

5.- Construcción



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO  
E.N.E.P. ARAGON INGENIERIA  
LABORATORIO DE CIRCUITOS LOGICOS  
PRACTICAS DE CONTROLADORES DIGITALES

---

PRACTICA NUMERO: 7 CONTROLADOR MYCA II

---

OBJETIVO: El alumno diseñará un módulo MYCA II y se rá capaz de aplicarlo en la solución a un controlador planteado.

INTRODUCCION

Como se recordará, el módulo MYCA II es una versión mejorada de el módulo MYCA I diseñado en la práctica anterior.

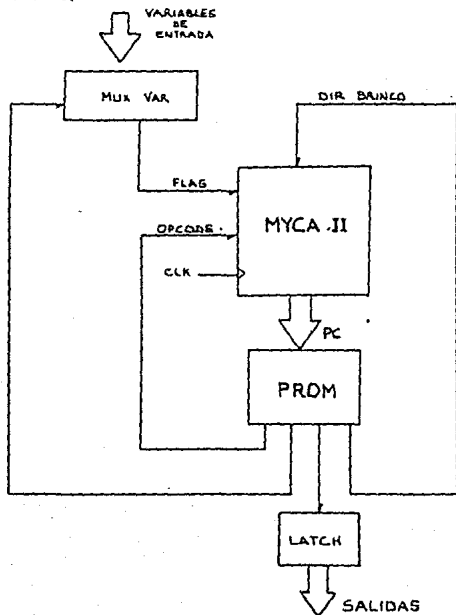
En la figura siguiente se muestra la estructura interna que deberá tener el módulo MYCA II. Note que los multiplexores Mux cuentas y Mux brincos se han eliminado y las selecciones se hacen desde una memoria interna, a la que llegan el OPCODE y la bandera como direcciones.

Mediante esta estructura resolveremos un controlador que cumpla simultaneamente con las funciones del controlador central y del control de conmutación vistos en las prácticas 1 y

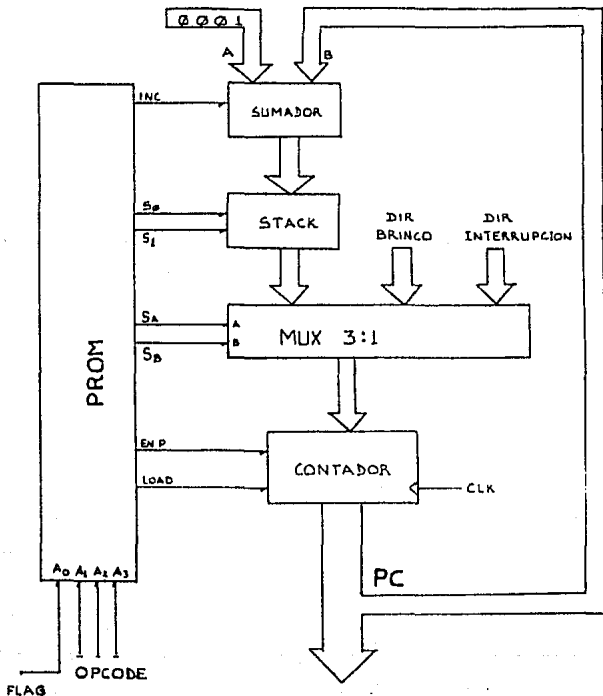
2. Ambos controladores serán diseñados como una sola de tal -- manera que se aproveche la capacidad de MYCA II para manejar subrutinas, considerando como tal el proceso de la cuenta de - 5 RINGS.

ARQUITECTURA A BLOQUES

(OBJETIVO)



ARQUITECTURA PROPUESTA PARA MYCA II.



## ACTIVIDADES

1. En base a la estructura presentada en la introducción-- presente la tabla de diseño del módulo MYCA II y el programa que deberá contener la memoria del MYCA II en hexadecimal.
- 2.- Elabore la carta ASM que de solución a la unión del controlador central y el controlador de conmutación (control común) vistos en las prácticas 1 y 2. Ordene de -- tal manera que el proceso de contar los 5 rings sea una subrutina.
3. Escriba el programa basado en instrucciones MYCA II para que se dé solución a la carta ASM del punto anterior.
4. Construya este controlador y muestre su funcionamiento al instructor.

SOLUCION

1. Nuevas instrucciones propuestas.

NEMONICO	FLAG	ACCION
JSR (FLAG) (DIR)	0 1	CUENTA: PC $\rightarrow$ PC + 1 SALTA: PC $\rightarrow$ DIR ; TOP $\rightarrow$ PC+1
RSR (FLAG) (DIR)	0 1	BRINCA: PC $\rightarrow$ DIR REGRESA: PC $\rightarrow$ TOP
INT (FLAG) (DIR)	0 1	CUENTA: PC $\rightarrow$ PC + 1 BRINCA: PC $\rightarrow$ DIR ; TOP $\rightarrow$ PC+1

OPCODE

1 0 1

1 1 0

1 1 1

Tabla del Programa de la PROM de MYCA II.

DIR	OPCODE	FLAG	S <sub>B</sub>	S <sub>A</sub>	ENP	LOAD	INC	S <sub>1</sub>	S <sub>0</sub>
0	000	0	0	0	0	1	0	0	0
1		1	0	0	1	1	0	0	0
2	001	0	0	0	1	1	0	0	0
3		1	0	0	1	1	0	0	0
4	010	0	0	0	0	1	0	0	0
5		1	0	1	0	0	0	0	0
6	011	0	0	1	0	0	0	0	0
7		1	0	1	0	0	0	0	0
8	100	0	0	0	1	1	0	0	0
9		1	0	1	0	0	0	0	0
10	101	0	0	0	1	1	0	0	0
11		1	0	1	0	0	1	1	0
12	110	0	0	1	0	0	0	0	0
13		1	1	0	0	0	0	0	1
14	111	0	0	0	1	1	0	0	0
15		1	0	0	0	0	1	1	0

DIR PROM

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

CONTADOR

S <sub>0</sub>	S <sub>1</sub>	ACCION
0	0	HOLD
0	1	PUSH
1	0	POP
1	1	LOAD

STACK



PROGRAMA MYCA II EN HEXADECIMAL:

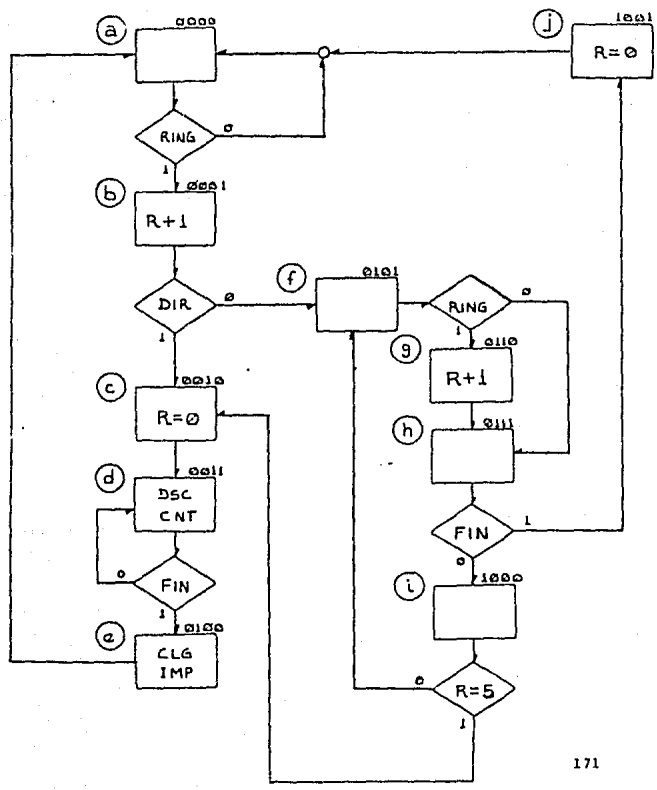
DIR PROM	VALOR HEXA
0	08
1	18
2	18
3	18
4	08
5	20
6	20
7	20
8	18
9	20
10	18
11	26
12	20
13	41
14	18
15	06

Estos valores se almacenan -  
en la PROM de MYCA II.

Para solucionar un problema-  
(construir un controlador),-  
se necesita escribir un pro-  
grama en otra memoria, la -  
del Microcontrolador en sí.

Por lo tanto, un diseño com-  
pleto requiere de 2 PROMS o-  
2 Memorias-

2.- Diseño de control común mediante MYCA II



3.- Programa Memórico.

EOO	DIR	INSTRUCCIÓN
a	0000	CC (RING)
b	0001	JSR (DIR) <0101>
c	0010	CI
d	0011	CC (FIN)
e	0100	BI <0000>
f	0101	C/BC (RING) <0111>
g	0110	CI
h	0111	C/BC (FIN) <0011>
i	1000	RSR (R=5) <0101>
j	1001	BI <0000>

Programa Binario y Hexadecimal

DIR	OPCODE	DIR MIX VAR	DIR BRINCO	SALIDAS			HEXAS
				R1	R2	CNT IMP	
0	000	000	0000	0	0	0	00 00
1	101	001	0101	1	0	0	51 58
2	001	000	0000	0	1	0	10 04
3	000	010	0000	0	0	1	02 02
4	011	000	0000	0	0	1	30 01
5	100	011	0111	0	0	0	43 70
6	001	000	0000	1	0	0	10 08
7	100	010	1001	0	0	0	42 90
8	110	100	1010	0	0	0	64 A0
9	011	000	0000	0	1	0	0 04

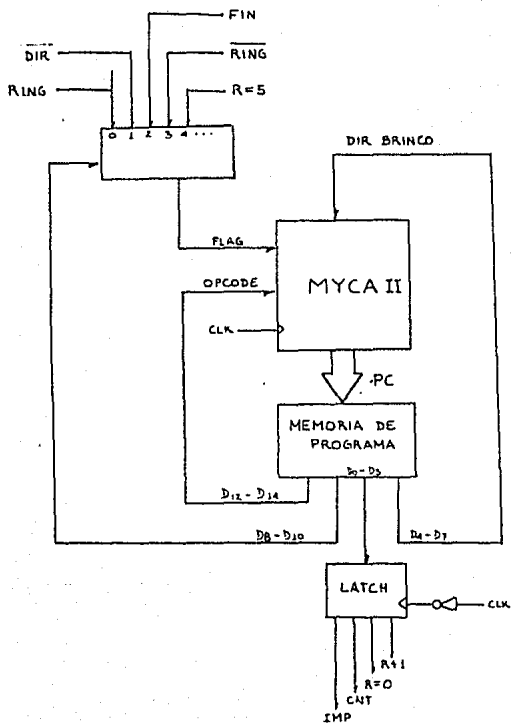
D<sub>14</sub> D<sub>13</sub> D<sub>12</sub>

D<sub>10</sub> D<sub>9</sub> D<sub>8</sub>

D<sub>7</sub> D<sub>6</sub> D<sub>5</sub> D<sub>4</sub>

D<sub>3</sub> D<sub>2</sub> D<sub>1</sub> D<sub>0</sub>

#### 4.- Construcción



Al término de este trabajo, hemos de observar que los -- capítulos II, III y VI (Controladores y paquete de prácticas) -- son los fundamentales y su estructura. Estos capítulos mencionados han de llevar al alumno al conocimiento teórico y desarrollo práctico de los dos grupos de controladores digitales que -- se estudian. Los capítulos restantes proporcionan información -- sobre los diferentes elementos que se utilizan en la estructura de estos controladores, sino que además presentan herramientas de "software" (Cartas ASM) que facilitarán el proceso de diseño y depuración del algoritmo que gobernará el funcionamiento -- del controlador. Esta aunada con los elementos de interfase -- presentados en el capítulo V, nos dá la posibilidad de crear -- sistemas completos de control digital de una considerable complejidad en gran variedad de aplicaciones.

El contenido medular de este trabajo debe mucho de su existencia a las notas de clase de Ing. David Jaime González Maxines, a quien le agradezco la información otorgada de un tema -- con poca documentación por su novedad, pero de gran importancia didáctica tanto por dar la pauta del funcionamiento de los microprocesadores como por ser directamente aplicable a la automatización desde procesos simples hasta complicados procesos -- industriales.

BOYLESTAD NASHELSKY

Electrónica Teoría de Circuitos.

Ed. PRENTICE HALL.

RONALD J. TOCCI

Sistemas Digitales Principios y Aplicaciones.

Ed. PRENTICE HALL.

CLAUDE A. WIATROWSKI

& CHARLES H. HOUSE

Logic Circuits and Microcomputer Systems.

Ed. MC GRAW HILL.

HERBERT TAUB

Circuitos Digitales y Microprocesadores.

Ed. MC GRAW HILL.

JOHN P. HAYES

Diseño de Sistemas Digitales y Microprocesadores

Ed. MC GRAW HILL.

TEXAS INSTRUMENTS

TTL Logic Databook

CAPITULO I

"CIRCUITOS COMBINACIONALES Y SECUENCIALES MSI"

1.1 MULTIPLEXOR COMO SELECTOR DE DATOS.....	4
1.2 MULTIPLEXOR 4:1.....	5
1.3 DEMULTIPLEXOR COMO DISTRIBUIDOR DE DATOS.....	7
1.4 DIAGRAMA GENERAL DE UN DECODIFICADOR.....	8
1.5 DECODIFICADOR BCD DECIMAL.....	9
1.6 ARREGLO MUX-DEMUX PARA TRANSMISION DE DATOS.....	10
1.7 CODIFICADOR OCTAL-BINARIO Y TABLA DE VERDAD.....	11
1.8 CODIFICADOR DECIMAL-BCD.....	13
1.9 DESPLIEGUE DE 7 SEGMENTOS DE ANODO COMUN.....	14
1.10 DECODIFICADOR/MANEJADOR DE BCD A 7 SEGMENTOS.....	15
1.11 COMPARADOR 7485 DE 4 BITS.....	16
1.12 EXPANSION DE COMPARADORES.....	17
1.13 REGISTRO PARALELO 74273 .....	20
1.14 REGISTRO SERIE 7491.....	21
1.15 REGISTRO SERIE-PARALELO 74164.....	22
1.16 REGISTRO DE CORRIMIENTO 74165.....	23
1.17 REGISTRO UNIVERSAL 74194.....	24
1.18 CONTADOR 7493 DE 4 BITS.....	25
1.19 CONTADOR BCD UP/DOWN.....	26
1.20 CIRCUITO MULTIPLEXOR CON CONTADOR.....	27
1.21 MEMORIA RAM A BLOQUES.....	30

## CAPITULO II

### "CONTROLADORES TRADICIONALES"

#### CONTROLADOR DISEÑADO CON FLIP-FLOPS

- DIAGRAMA DE FLUJO.....	36
- ASIGNACION DE ESTADOS Y MAPA DE EDOS. FUTUROS.....	38
- TABLAS DE ENTRADAS Y SALIDAS JK .....	39
- IMPLEMENTACION.....	40
- TABLAS DE ENTRADAS CON FLIP-FLOPS TIPO D.....	40
- CONSTRUCCION CON MULTIPLEXORES DIRECTAMENTE RELACIONADOS.....	41

#### CONTROLADOR DISEÑADO CON REGISTROS DE CORRIMIENTO

- TABLA DE INSTRUCCIONES 74194.....	42
- DIAGRAMA DE FLUJO Y MAPAS DE ESTADOS.....	43
- MAPAS DE ACCION .....	44
- MAPAS DE CARGA EN PARALELO Y EN SERIE.....	45
- MAPAS DE SALIDAS.....	46
- CONSTRUCCION .....	47

#### CONTROLADOR DISEÑADO CON CONTADORES

- DIAGRAMA DE FLUJO Y MAPAS DE ESTADOS.....	50
- MAPAS DE ACCION Y DE CARGA.....	51
- CONSTRUCCION .....	52

## CAPITULO III

### "CONTROLADORES MICROPROGRAMADOS"

#### CONTROLADOR MICROPROGRAMADO SENCILLO

- ESTRUCTURA GENERAL.....	54
- DIAGRAMA Y TABLAS DE SOLUCION.....	57
- CONSTRUCCION .....	58



## CONTROLADOR CON NUMERO FIJO DE INSTRUCCIONES

- TABLA DE INSTRUCCIONES.....	60
- ARQUITECTURA DE TRABAJO.....	61

## CONTROLADOR CON MANEJO DE SUBROUTINAS

- TABLA DE INSTRUCCIONES ADICIONALES.....	64
- ARQUITECTURA DE TRABAJO.....	65

## CAPITULO IV

### "DISEÑO CON CARTAS ASM"

4.1 CAJA DE ESTADO.....	70
4.2 DIAMANTE DE DECISION.....	71
4.3 CAJA DE SALIDA CONDICIONAL.....	72
4.4 NODO CONECTOR.....	73
4.5 BLOQUE ASM.....	74
4.6 CONEXIONES PARALELAS/SERIE PARA SALIDAS CONDICIONALES	75
4.7 REDUCCION DE CARTAS ASM.....	76

## CAPITULO V

### "INTERFASES"

5.1 CONMUTADOR DE DOS TERMINALES.....	83
5.2 CONMUTADOR MANUAL DE 3 TERMINALES.....	85
5.3 TECLADO DE 16 TECLAS TIPO CALCULADORA.....	86
5.4 VISUALIZADORES.....	88
5.5 FOTOTRANSISTOR.....	90
5.6 OPTOAISLADORES.....	91
5.7 INTERFASE CON OPTOAISLADOR.....	93

5.8	INTERFASES DE UNA COMPUTADORA.....	97
5.9	COMPARADOR ANALOGICO.....	99
5.10	COMPARADOR ANALOGICO.....	100
5.11	DAC DE 8 BITS DE RESISTENCIAS PONDERADAS.....	101
5.12	CONVERSION DIRECTA.....	102
5.13	INTERCONEXION DE UN TECLADO A UN PROCESADOR.....	105
5.14	INTERCONEXION DE UN DISPOSITIVO E/S SERIE.....	107
5.15	TRANSDUCTORES BASADOS EN RESISTENCIAS.....	109
5.16	ACTUADOR SOLENOIDE UTILIZADO EN UNA CERRADURA.....	113
5.17	APLICACION DE UN MOTOR ELECTRICO.....	114

## CAPITULO VI

### "PAQUETE DE PRACTICAS"

#### PRACTICA No. 1 CONTROLADORES CABLEADOS CON FF JK

-	PLANTEAMIENTO GENERAL.....	119
-	DIAGRAMA DE CAJA NEGRA.....	121
-	CARTA ASM .....	122
-	DIAGRAMA DE FLUJO.....	123
-	TABLA DE ESTADOS PRESENTES Y FUTUROS.....	124
-	TABLAS DE ENTRADAS Y SALIDAS.....	125
-	CONSTRUCCION .....	126

#### PRACTICA No. 2 CONTROLADORES CABLEADOS CON FF TIPO D

-	PLANTEAMIENTO.....	128
-	DIAGRAMA DE CAJA NEGRA.....	130
-	CARTA ASM .....	131
-	DIAGRAMA DE FLUJO.....	132

- TABLAS DE ESTADOS PRESENTES Y FUTUROS.....	133
- TABLAS DE ENTRADAS Y SALIDAS.....	134
- CONSTRUCCION .....	135

PRACTICA No. 3 CONTROLADOR CON REGISTRO DE CORRIMIENTO

- PLANTEAMIENTO.....	137
- DIAGRAMA DE CAJA NEGRA.....	140
- CARTA ASM.....	141
- INSTRUCCIONES EN LA CARTA ASM.....	142
- MAPAS DE ESTADOS, INSTRUCCIONES Y ACCIONES.....	143
- MAPAS DE CARGA Y SALIDAS.....	144
- CONSTRUCCION .....	145

PRACTICA No. 4 CONTROLADOR CON CONTADOR

- PLANTEAMIENTO.....	147
- DIAGRAMA DE CAJA NEGRA.....	148
- CARTA ASM.....	149
- MAPAS DE ESTADOS PRESENTES Y ACCIONES.....	150
- MAPAS DE CARGA.....	151

PRACTICA No. 5 CONTROLADOR MICROPROGRAMADO

- ESTRUCTURA.....	153
- CARTA ASM.....	154
- PROGRAMA.....	155
- CONSTRUCCION .....	156

PRACTICA No. 6 CONTROLADOR MYCA I

- OBJETIVO Y TABLA DE DISEÑO.....	158
- MAPAS DE ACCION E IMPLEMENTACION MYCA I.....	160
- ARQUITECTURA DE TRABAJO MYCA I .....	161

- CARTA ASM .....	162
- TABLA NEMONICA Y PROGRAMAS SOLUCION .....	163
- CONSTRUCCION .....	164

PRACTICA No. 7 CONTROLADOR MYCA II

- ARQUITECTURA A BLOQUES (OBJETIVO) .....	166
- ARQUITECTURA PROPUESTA PARA MYCA II.....	167
- TABLAS DE NUEVAS INSTRUCCIONES Y PROGRAMA MYCA II.....	169
- PROGRAMA MYCA II HEXADECIMAL.....	170
- CARTA ASM .....	171
- PROGRAMA NEMONICO, BINARIO Y HEXADECIMAL.....	172
- CONSTRUCCION .....	173

---

GLOSARIO DE TERMINOS TECNICOS

---

- ACTIVAR:** (una línea): cambiar el estado lógico de una salida de un dispositivo, de tal manera que se interprete como una respuesta a sus entradas. Se usa también para poner un dato de entrada en un dispositivo para que se interprete como una orden o instrucción de funcionamiento.
- AUTOMATA:** Sistema digital que funcionará automáticamente cumpliendo con un programa establecido.
- BCD:** Código binario de 4 bits usado para representar los dígitos decimales.
- BUS:** Línea física por donde circulan datos.
- CARGAR:** (un dato): acción de aceptar un nuevo dato para almacenarlo.
- CHIP:** Circuito integrado encapsulado en una sola pastilla.

**DIAGRAMA DE FLUJO:** Diagrama gráfico que muestra el comportamiento de un sistema en base a las relaciones entre los estados de tal sistema.

**DIRECCIONAR:** Señalar una dirección específica de entre varios posibles con la que se tendrá transferencia de datos. Se usa también para señalar dispositivos.

**ESTADO:** (de un sistema) situación o valor de las variables principales de un sistema entre un pulso de reloj y otro. Eta pas.

**F.F.:** (flip-flop): dispositivo básico de almacenamiento de datos. Almacena un bit.

**HABILITAR:** Permitir que un circuito funcione. Se usa también referido a entradas y salidas.

**HARDWARE:** Circuitos físicos que componen a un sistema digital.

- K.** (Mapa): Mapa de Karnaugh, tabla donde se especifica el valor de una variable en función de otras. Es auxiliar en la simplificación de funciones.
- LED:** Tipo especial de diodo que al ser polarizado directamente emite luz.
- LOGICA AUXILIAR:** Elementos combinatoriales con la función de apoyar el funcionamiento de un sistema secuencial.
- LSI:** Alta escala de integración. Grupo de dispositivos que incorporan registros, decodificadores y demás elementos MSI en un sólo chip.
- MEALY:** Grupo de sistemas digitales secuenciales cuyas salidas dependen del estado y de las entradas.
- MICROCONTROLADOR:** Controlador que funciona en base a un programa agregado a su estructura.
- MOORE:** Grupo de sistemas digitales secuenciales cuyas salidas dependen exclusivamente del estado.

**MSI:** Mediana Escala de Integración. Grupo - de dispositivos que incorporan flip- - flops y compuertas SSI en un sólo chip.

**"NO IMPORTA"** (condición) se le impone esta condi--- ción a un dato cuando es irrelevante - que valga 0 o 1. Es decir cuando su - valor no afectara el funcionamiento de un sistema. Se simboliza con X o \*.

**SOFTWARE:** Relaciones lógicas (programas) que componen un sistema digital.

**SSI:** Baja escala de integración. Grupo de- dispositivos que incorporan varios elementos semiconductores en un sólo chip.

**TABLA DE CONTROL DE MODO:** Tabla en la que se especifica - el funcionamiento que tendrá un dispo- sitivo en base al valor de sus entra - das de control.