

# SOLUCIÓN SISTEMÁTICA DE SISTEMAS DIGITALES SECUENCIALES MODO NIVEL

Juan Angel Garza Garza<sup>1</sup>  
Cesar Augusto Leal Chapa<sup>2</sup>  
Guadalupe Ignacio Cantú Garza<sup>3</sup>

## RESUMEN

Esta ponencia propone un método sistemático que nos permite solucionar sistemas digitales secuenciales de modo nivel (asíncronos), mediante una secuencia de pasos y con la ayuda de la computadora personal y programas de aplicación que nos permiten minimizar, sintetizar y simular dichos sistemas y asegurar su correcto funcionamiento antes de su implementación física ya sea con dispositivos lógicos programables (PLD's) o con controladores lógicos programables (PLC's).

## INTRODUCCIÓN

Los Sistemas secuenciales asíncronos caracterizados mediante niveles, son los primeros sistemas secuenciales que se desarrollaron y se diseñan mediante una metodología que puede considerarse como intuitiva. Se pueden realizar por medio de retroalimentación directa o mediante celdas asíncronas básicas activadas por niveles, que reciben el nombre de biestables asíncronos, o activadas por cambios de nivel o flancos (Mandado Perez, 2008).

Otra definición de un Sistema Secuencial modo nivel: es aquél sistema en donde los valores de salidas ( $Z1...Zn$ ) no dependen únicamente de las combinaciones de entrada ( $X1...Xn$ ) sino también del estado presente ( $Y1...Yp$ ) sin incluir una señal adicional de sincronía (Roth, Fundamentals Logic Design, 1992), tal como se muestra en la figura 1.

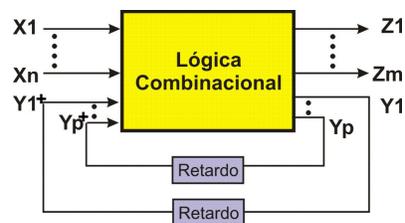


Figura 1. Descripción a Bloques de un sistema secuencial asíncrono

<sup>1</sup> Profesor y Coordinador de Informática, Facultad de Ingeniería Mecánica y Eléctrica, U.A.N.L.

<sup>2</sup> Profesor, Facultad de Ingeniería Mecánica y Eléctrica, U.A.N.L.

<sup>3</sup> Profesor, Facultad de Ingeniería Mecánica y Eléctrica, Univ. Autónoma de Nuevo León.

Estos sistemas son utilizados en aplicaciones en donde los eventos que determinan la secuencia no dependen del tiempo sino de valores específicos de una variable; por ejemplo, en los circuitos de arranque, paro y control de aceleración para motores eléctricos (Cantu Garza & Garza Garza, 2010). Las formas más comunes de representación gráfica del comportamiento de un sistema digital secuencial de modo nivel son:

- Diagrama de Tiempos figura 2.
- Diagrama de Transición figura 3.

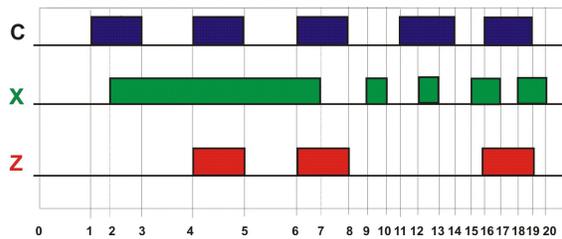


Figura 2. Diagrama de Tiempos

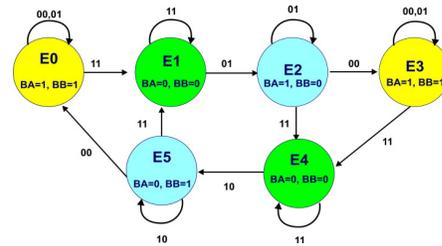


Figura 3. Diagrama de transición

### Características de los sistemas secuenciales asíncronos:

- No necesita del reloj como señal de control.
- La salida cambia cuando una variable de entrada cambia (por eventos).
- Se implementan a partir de operadores And, Or y Not, utilizando retroalimentación.
- También a partir de Flip Flop R-S básico (sin reloj).
- No requieren de Flip Flop's estándar (J-K, T o D).
- Más rápidos en su respuesta por no depender de una señal adicional de sincronía, sólo de un cambio de nivel.

**Tabla de flujo primitiva.** Es la manera más simple para describir el comportamiento relativo al sistema digital a diseñar, consiste en la construcción de una tabla en donde se consideran todas las posibles transiciones en el contexto de un problema de diseño.

También se considera que tiene únicamente un estado estable por fila, con salidas especificadas sólo para estados estables (Balabanian, 2002).

En una tabla de flujo (o transiciones) se tiene la misma información que en un diagrama de estados, pero organizada de forma tabular. Una tabla de flujo se considera "primitiva" si en la construcción de ella se establecen las siguientes condiciones:

- Solo debe de haber un estado estable por fila.
- Solo cambia una variable de entrada a la vez.

## **OBJETIVO**

Que el estudiante de ingeniería de los programas educativos relacionados con las especialidades de electrónica y automatización desarrollen la competencia (Martinez Alonso, Garza Garza, & Portuondo Padrón, 2008) del diseño de sistemas digitales modo nivel teniendo como recurso el método propuesto en esta ponencia y lo puedan aplicar en su ámbito profesional.

## **JUSTIFICACIÓN**

El tradicional proceso de diseño está basado en la intuición, habilidad y experiencia de quien diseña, limitando el desarrollo de aplicaciones solo para quienes cumplen ese perfil, considerándose un recurso artesanal no necesariamente ingenieril. En este escrito se presenta una solución sistemática de diseño con la que se resuelve la dificultad planteada anteriormente con la posibilidad de que se pueda en una forma didáctica diseñar dichos sistemas sin necesidad de tener una gran experiencia

## **METODOLOGIA**

La propuesta de esta ponencia es aplicar un método sistemático para diseñar sistemas secuenciales modo nivel, para la automatización de procesos industriales con el uso de la programación por eventos discretos, en donde su implementación física puede realizarse en un Dispositivo Lógico Programable (PLD) por medio de un Lenguaje de Descripción de Hardware (HDL) o también en Controlador Lógico Programable (PLC) utilizando el diagrama escalera, las ecuaciones obtenidas o el diagrama esquemático (Garza Garza, Sistemas Digitales y Electrónica Digital, 2006). Por medio de este método es posible solucionar problemas de automatización de procesos industriales, que tradicionalmente se basan en módulos o soluciones anteriormente ya planeadas sin conocer la eficiencia de estos, además presentan una lógica en la que otro diseñador le tomaría tiempo comprenderla para hacer adecuaciones o en su caso reparaciones.

A continuación se muestran 14 pasos sugeridos para el desarrollo y diseño de sistemas secuenciales asíncronos:

- |  |   |
|--|---|
| 1.- Especificar el Sistema.              | 9.- Completar Tabla de Salidas.                             |
| 2.- Tabla de Flujo Primitiva.            | 10.- Obtención de las ecuaciones por medio de minimización. |
| 3.- Eliminación de estados redundantes.  | 11.- Elaborar el archivo en formato ABEL-HDL.               |
| 4.- Mezcla de filas.                     | 12 Simulación.  |
| 5.- Expandir tabla de salidas.           | 13.- Representación grafica.                                |
| 6.- Tabla de estados internos.           | 14.- Implementación   |
| 7.- Asignación de valores a los Estados. |   |
| 8.- Tabla de Estados Totales.            |   |

Para la demostración de la efectividad del método, a continuación se muestra su aplicación en un ejemplo, así como los resultados obtenidos. Seleccionamos un sistema digital secuencial asíncrono capaz de detectar el sentido de giro de un motor por medio de dos sensores llamados A y B, en el que la dirección de giro se indica por medio del valor de una salida Z.

El sistema cuenta con un dispositivo con dos círculos concéntricos distribuidos en código GRAY, de modo que si el giro es a favor de las manecillas del reloj (Derecha,  $Z=1$ ) la secuencia de valores que se obtendrían en A y B serían: 00, 01, 11 y 10 sucesivamente; pero si el giro es en sentido contrario de las manecillas del reloj (Izquierda,  $Z=0$ ) la secuencia de valores de A y B serían: 00, 10, 11 y 10 sucesivamente. En cualquier instante que cambie la dirección de giro, la salida Z deberá de cambiar su valor como lo muestra la figura 4.

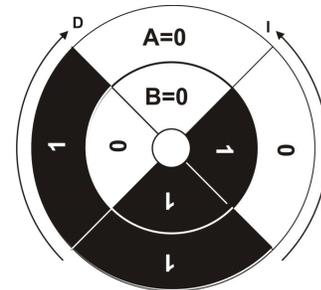


Figura 4. Dispositivo sensor

### 1.- Especificar el Sistema.

Para describir el funcionamiento del método propuesto, utilizaremos como recurso el diagrama de transición descrito en la figura 5.

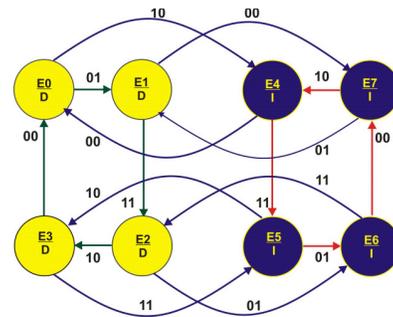


Figura 5. Diagrama de transición

### 2.- Tabla de Flujo Primitiva.

En la tabla siguiente se describe completamente el comportamiento del sistema y además se comprueba que todas las transiciones estén definidas.

En la tabla hay que considerar que los estados subrayados son estados persistentes (estables) y los no subrayados son estados transitorios (inestables), además las casillas marcadas con X se refieren a transiciones que no son posibles porque implica un cambio simultáneo de dos o más entradas (A y B).

	Entradas AB				Salida Z
	00	01	11	10	
0	<u>E0</u>	E1	X	E4	D
1	E7	<u>E1</u>	E2	X	D
2	X	E6	<u>E2</u>	E3	D
3	E0	X	E5	<u>E3</u>	D
4	E0	X	E5	<u>E4</u>	I
5	X	E6	<u>E5</u>	E3	I
6	E7	<u>E6</u>	E2	X	I
7	<u>E7</u>	E1	X	E4	I

### 3.- Eliminación de Estados Redundantes (equivalentes).

Dos estados son equivalentes y por lo tanto uno de ellos redundante si:

- a) Son estados estables en la misma columna (misma combinación de entradas).
- b) Tienen la misma salida.
- c) Sus estados siguientes son equivalentes.

En la tabla anterior podemos apreciar que en la columna 00 los estados E0 y E7 son estables pero tienen salida diferente, así mismo en la columna 01 con los estados E1 y E6, en la 11 con E2 y E5, también en la columna 10 con E3 y E4, por lo que podemos concluir que no hay estados equivalentes por contar con salidas diferentes.

### 4.- Mezcla de Filas.

En este proceso, la tabla de flujo primitiva se transforma en una tabla de estados totales (Aquel formado por la combinación de las entradas y el estado presente).

Dos filas o mas se pueden mezclar siempre y cuando no haya ningún conflicto sobre qué estado debe ocupar cada columna, entendiéndose por conflicto la ocupación simultánea de una columna por dos estados diferentes. La salida no se considera como un factor de conflicto en la mezcla de filas. Esto es, dos filas con salidas diferentes pueden mezclarse.

Con el propósito de tener una visualización completa sobre las posibilidades de mezcla de las filas, se construye un diagrama de mezcla, el cual se inicia asignando un punto por cada fila (figura 6) y se termina uniendo esos puntos por líneas, cuando los estados pueden mezclarse (figura 7), sin considerar el valor de la salida.

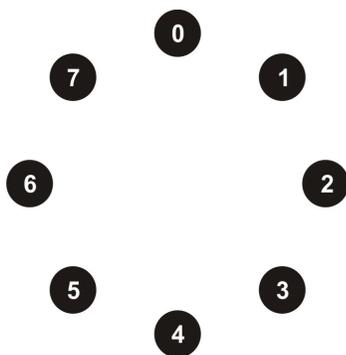


Figura 6. Diagrama inicial de mezcla de filas

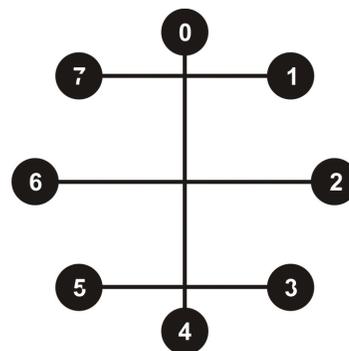


Figura 7. Diagrama final de mezcla de filas

La siguiente tabla muestra las filas ya mezcladas sin considerar la salida.

Entradas AB				
	00	01	11	10
0, 4	<u>E0</u>	E1	E5	<u>E4</u>
1, 7	<u>E7</u>	<u>E1</u>	E2	E4
2, 6	E7	<u>E6</u>	<u>E2</u>	E3
3, 5	E0	E6	<u>E5</u>	<u>E3</u>

### 5.- Expandir Tabla de Salidas.

Si para una hilera se tiene la posibilidad de mezclar filas, pero los estados estables tienen salidas diferentes, es necesario expandir las salidas como se muestra en la siguiente tabla.

	Entradas AB				Salida Z			
	00	01	11	10	00	01	11	10
0, 4	<u>E0</u>	E1	E5	<u>E4</u>	D			
1, 7	<u>E7</u>	<u>E1</u>	E2	E4	I	D		
2, 6	E7	<u>E6</u>	<u>E2</u>	E3		I	D	
3, 5	E0	E6	<u>E5</u>	<u>E3</u>			I	D

En las salidas la D significa giro a favor de las manecillas del reloj (Derecha) e I giro contrario a las manecillas del reloj (Izquierda).

### 6.- Tabla de Estados Internos.

La tabla de estados totales obtenida al mezclar las filas, se debe convertir en una tabla de estados internos (Aquella que incluye solo al estado presente). Para ello, se sustituye cada fila por una variable, quedando asignados con esa variable los estados estables de esa fila. De modo que, en este caso:

La fila marcada E0 y E4 =a,  
 E1 y E7 =b,  
 E6 y E2 =c y  
 E3 y E5 =d.

Ocurriendo las siguientes transiciones:

En la columna 00: c→b, d→a  
 En la columna 01: a→b, d→c  
 En la columna 11: a→d, b→c  
 En la columna 10: b→a, c→d

	Entradas AB				Salida Z			
	00	01	11	10	00	01	11	10
a	<u>a</u>	b	d	<u>a</u>	D			I
b	<u>b</u>	<u>b</u>	c	a	I	D		
c	b	<u>c</u>	<u>c</u>	d		I	D	
d	a	c	<u>d</u>	<u>d</u>			I	D

### 7.- Asignación de Valores a los Estados.

Para cumplir con las transiciones descritas anteriormente para cada columna, se asignan valores a los estados de modo que, entre aquellos estados en donde haya una transición, cambie una sola variable, para evitar una condición de carrera.

La forma más simple de cumplir estas condiciones es aprovechar la estructura de un mapa de Karnaugh, como puede apreciarse en figura 8 en donde se ha establecido una asignación que cumple con las transiciones presentes en el diseño.

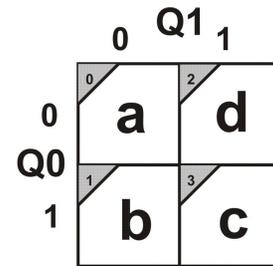


Figura 8. Mapa de asignaciones

De modo que los valores de Q1 y Q0 para cada estado son:

	Q1	Q0
a	0	0
b	0	1
c	1	1
d	1	0

### 8.- Tabla de Estados Asignados.

En esta tabla se sustituyen los estados internos por el valor de la asignación propuesta en el paso anterior.

	Entradas AB				Salida Z			
	00	01	11	10	00	01	11	10
00	<u>00</u>	01	10	<u>00</u>	D			I
01	<u>01</u>	<u>01</u>	11	00	I	D		
11	01	<u>11</u>	<u>11</u>	10		I	D	
10	00	11	<u>10</u>	<u>10</u>			I	D

### 9.- Completar Tabla de Salidas.

En este diseño las salidas no están completamente definidas y esto puede generar transiciones no deseadas (Glitches), inconvenientes para el sistema. Por tal razón, es necesario asignar un valor a las salidas no asignadas de modo que no se presenten dichas transiciones no deseadas.

	Entradas AB				Salida Z			
	00	01	11	10	00	01	11	10
00	<u>00</u>	01	10	<u>00</u>	D=1	D=1	I=0	I=0
01	<u>01</u>	<u>01</u>	11	00	I=0	D=1	D=1	I=0
11	01	<u>11</u>	<u>11</u>	10	I=0	I=0	D=1	D=1
10	00	11	<u>10</u>	<u>10</u>	D=1	I=0	I=0	D=1

Nota: La salida definida como D se asignó el valor 1 y a la definida como I el valor 0

## 10.- Obtención de las ecuaciones por medio de minimización.

De la tabla anterior se obtiene la siguiente tabla de verdad, que utilizando el programa de aplicación LogicAid se obtienen las siguientes ecuaciones mínimas;

m	A	B	Q1	Q0	Q1+	Q0+	Z
0	0	0	0	0	0	0	1
1	0	0	0	1	0	1	0
2	0	0	1	0	0	0	1
3	0	0	1	1	0	1	0
4	0	1	0	0	0	1	1
5	0	1	0	1	0	1	1
6	0	1	1	0	1	1	0
7	0	1	1	1	1	1	0
8	1	0	0	0	0	0	0
9	1	0	0	1	0	0	0
10	1	0	1	0	1	0	1
11	1	0	1	1	1	0	1
12	1	1	0	0	1	0	0
13	1	1	0	1	1	1	1
14	1	1	1	0	1	0	0
15	1	1	1	1	1	1	1

Ecuaciones mínimas:

$$Q1 = B Q1 + A Q1 + A B$$

$$Q0 = A'Q0 + A'B + B Q0$$

$$Z = A Q1 Q0 + B Q1'Q0 + B'Q1 Q0' + A'Q1'Q0'$$

$$Z = A B Q0 + A B'Q1 + A'B Q1' + A'B'Q0'$$

En donde la salida Z tiene dos posibles resultados.

## 11.- Elaborar el Archivo en formato ABEL-HDL.

Por medio de esta herramienta en el programa ISPliver Projet Navigator de la compañía Lattice Semiconductor nos permite efectuar la simulación y si cumple correctamente con lo esperado implementarlo físicamente en un Dispositivo Lógico Programable (PLD).

```

MODULE SIMULA
"Entradas
A,B pin 1,2;
"Salidas
Z,Q1,Q0 pin 19..17 istype 'com';
equations
"Q1+ = B Q1 + A Q1 + A B
"Q0+ = A'Q0 + A'B + B Q0
"Z = A B Q0 + A B'Q1 + A'B Q1' + A'B'Q0'
Q1 = B&Q1# A&Q1# A&B;
Q0 = !A&Q0# !A&B#B&Q0;
Z = A&B&Q0#A&B&Q1#!A&B&!Q1#!A&B&!Q0;

```

```

test_vectors
([A,B]->Z)
[0,0]->.x.;
[0,1]->.x.;
[1,1]->.x.;
[1,0]->.x.;
[0,0]->.x.;
[1,0]->.x.;
[1,1]->.x.;
[0,1]->.x.;
[0,0]->.x.;
END

```

## 12 Simulación.

En la figura 9 se muestra la simulación del diseño, en donde se propone una secuencia de los valores de entrada A y B en donde se obtiene la salida Z, adicionalmente se obtienen los valores de Q1 y Q0.

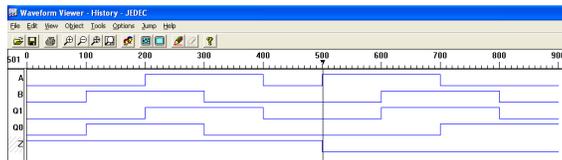


Figura 9. Simulación por medio de un diagrama de tiempo

### 13.- Representación grafica.

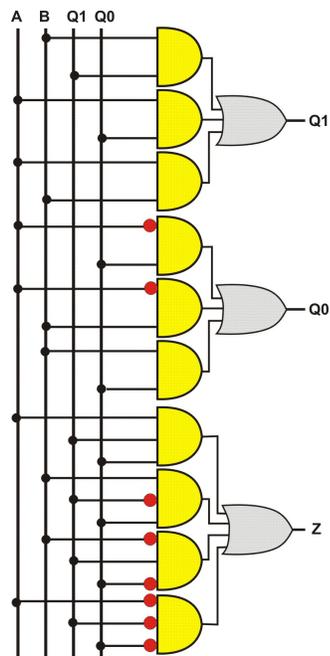


Figura 10. Diagrama esquemático

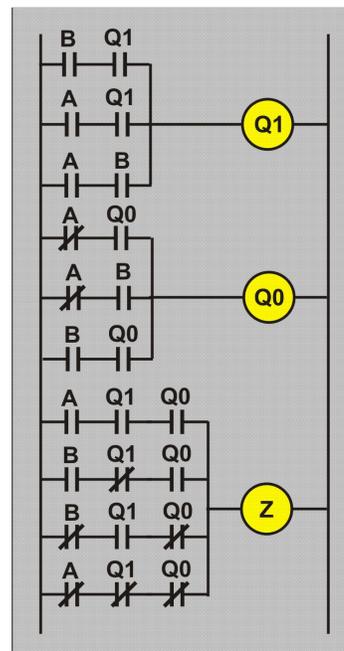


Figura 11. Diagrama escalera

### CONCLUSIONES

Como hemos demostrado en este documento a través de un ejemplo sencillo e ilustrativo, la aplicación del método de diseño de sistemas digitales secuenciales asíncronos (Modo nivel), presenta la ventaja de la sistematización, comparado con el método de diseño tradicional, en donde el proceso es más intuitivo que sistemático y; por tanto, exige un alto grado de experiencia para llevar a cabo un diseño exitoso.

Con un proceso de diseño sistemático, se obtienen soluciones simplificadas, ahorrando elementos en la implementación y tiempo en el diseño; así mismo, con el uso de herramientas computacionales, el diseño propuesto puede probarse, usando el recurso de la simulación, antes de la implementación física.

Por otro lado, la tendencia de los sistemas educativos actuales es el desarrollo de competencias. Una de las principales en el área de ingeniería es la aplicación del conocimiento, y con respecto a ello, un recurso para lograrlo es la sistematización del conocimiento teórico, ya que esto facilita que el alumno pueda producir e implementar diseños exitosos a bajo costo, aprendiendo por cuenta propia.

## **Bibliografía**

Balabanian, N. (2002). Diseño Logico Digital. Florida, EUA: CECSA.

Brown, S. (2006). Fundamentos de Logica Digital con diseño VHDL. Toronto, Canada: McGraw Hill.

Cantu Garza, G. I., & Garza Garza, J. A. (2010). Control Electrónico de Motores (Primera edicion ed.). México: FIME UANL.

Garza Garza, J. A. (2006). Sistemas Digitales y Electrónica Digital. México: PEARSON.

Garza Garza, J. A., Cantu Garza, G. I., Hernández Venegas, J. E., & Martínez Luna, S. (2002). Laboratorio de Electrónica Digital. Monterrey: FIME UANL.

Leal Chapa, C. A. (1990). Fundamentos de Diseño Digital. Monterrey, México: FIME UANL.

Mandado Perez, E. (2008). Sistemas Electrónicos Digitales. Barcelona, España: Marcombo.

Martínez Alonso, G. F., Garza Garza, J. A., & Portuondo Padrón, R. (2008). El Currículo Basado en competencias y su implementación en cursos de ingeniería. Ingenierías XI núm 41 , 40-50.

Roth, C. H. (1992). Fundamentals Logic Design. EUA: WEST.