

UNIVERSIDAD NACIONAL DE TUCUMÁN

FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍA

ELECTRÓNICA II



Actividad 5: Contadores

Autor: Chaves, Gonzalo Efraín - Estudiante de Ing. Electrónica

Resumen:

El trabajo aborda el diseño e implementación de máquinas de estado finito (FSM) aplicadas a contadores binarios, utilizando dos enfoques distintos:

1. Método basado en tablas de estados y salidas.
2. Método basado en datapath.

Primero se repasan los conceptos de lógica secuencial, máquinas de Mealy y Moore, y el datapath, entendido como la parte del hardware que procesa y mueve datos, mientras la unidad de control define las operaciones.

Se diseña un contador binario de 3 bits con reset sincrónico mediante el método clásico de estados y transiciones, elaborando:

Tablas de estado/salida, transición y excitación.

Codificación binaria de estados.

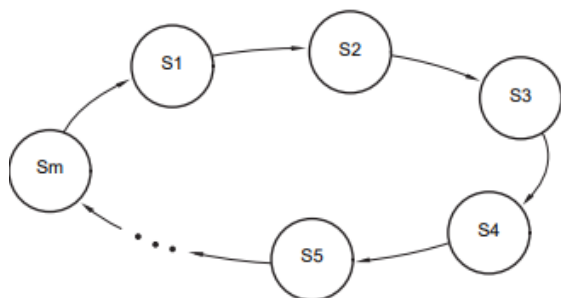
Síntesis lógica mediante mapas de Karnaugh.

En la segunda parte, se aplica el método del datapath para diseñar un contador binario de 4 bits con reset, carga y habilitación sincrónicos. Este enfoque simplifica el diseño al expresar el comportamiento mediante expresiones lógicas o VHDL(lenguaje que se utiliza para el curso), en lugar de construir tablas complejas, especialmente útiles para sistemas con muchas entradas y estados.

Introducción:

Concepto de Contador

Según Wakerly (2008) el nombre "contador" se utiliza generalmente para cualquier circuito secuencial sincronizado cuyo diagrama de estados contiene un solo ciclo, como se muestra en la Figura 1.1. Un contador con m estados se denomina contador módulo- m o, a veces, contador de división por m .



Uno de los modelos más utilizados de contadores es el contador binario de n bits. Este tipo de contador tiene la secuencia $0, 1, 2, 2^n-1, 0, 1, \dots$. Cada uno de estos estados se codifica como el entero binario de n bits.

Figura 1.1 - Estructura general de un diagrama de estados de un contador.

Para hablar correctamente de un contador, primero veremos lo que es la lógica secuencial, que es distinta a la lógica combinacional con la que se vino trabajando.

La lógica combinacional se la puede representar con tablas, siendo que un determinado cambio en la entrada tiene ya fijado un cambio o resultado a la salida. La lógica combinacional no tiene memoria ni dependencia temporal de las entradas, solo depende del estado actual de las mismas, es instantánea.

A diferencia de ello, la Lógica Secuencial, tiene memoria, depende de las entradas actuales y del estado anterior para poder determinar su salida, así como en orden y el tiempo de las entradas influyen en la salida. Como estos cambios son muy variados en el tiempo, normalmente depende una de una señal de reloj para cambiar de estado.

Algunos Latch, y especialmente los Flip-Flops, así como los registros son elementos principales en la implementación de lógica secuencial, ya que se utilizan como elementos de memoria y responden a una señal de reloj para cambiar o no de estado.

Donde la señal de reloj, es un detector de flanco que ve una señal, generalmente, cuadrada donde puede utilizar el flanco ascendente o descendente, dependiendo de las necesidades.

Máquina de Estado Finito

Una Máquina de Estado Finito es un circuito secuencial, donde sus estados actuales y salidas, dependen de las entradas y estados anteriores.

Un circuito secuencial cuyas salidas dependen tanto del estado como de las entradas se conoce como una máquina de Mealy Salidas = $G(\text{estado actual, entradas})$, como se muestran en la Figura 1.2.

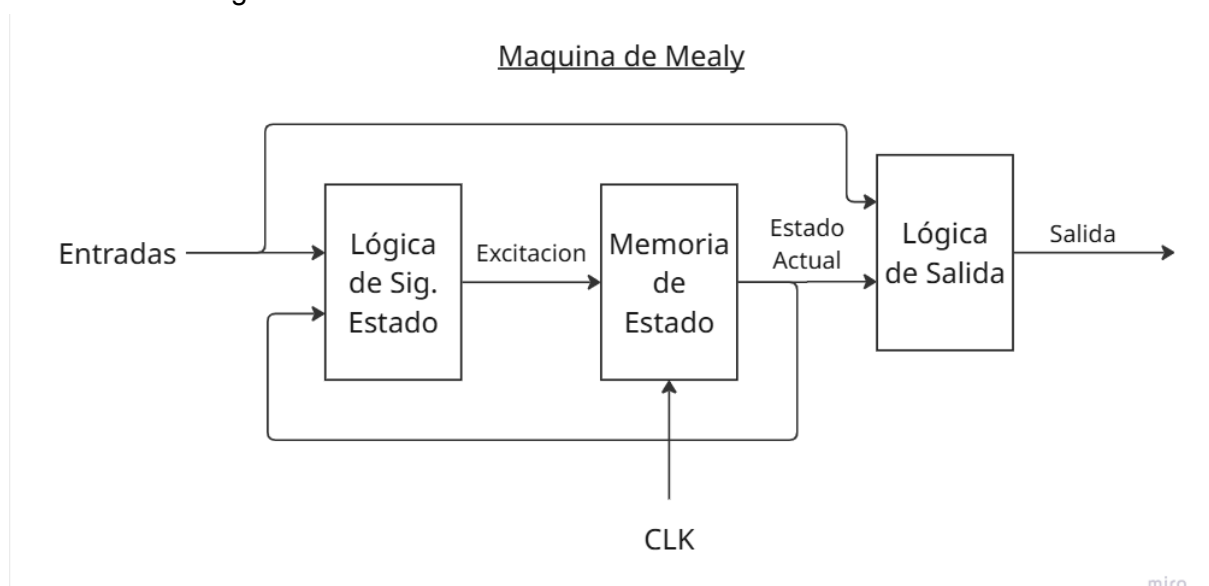


Figura 1.2 - Estructura de una Máquina de Estado Finito Sincrónica de Mealy

En algunos circuitos secuenciales las salidas dependen solamente del estado, tal circuito se conoce como una máquina de Moore: Salidas = $G(\text{estado actual})$, como en la Figura 1.3.

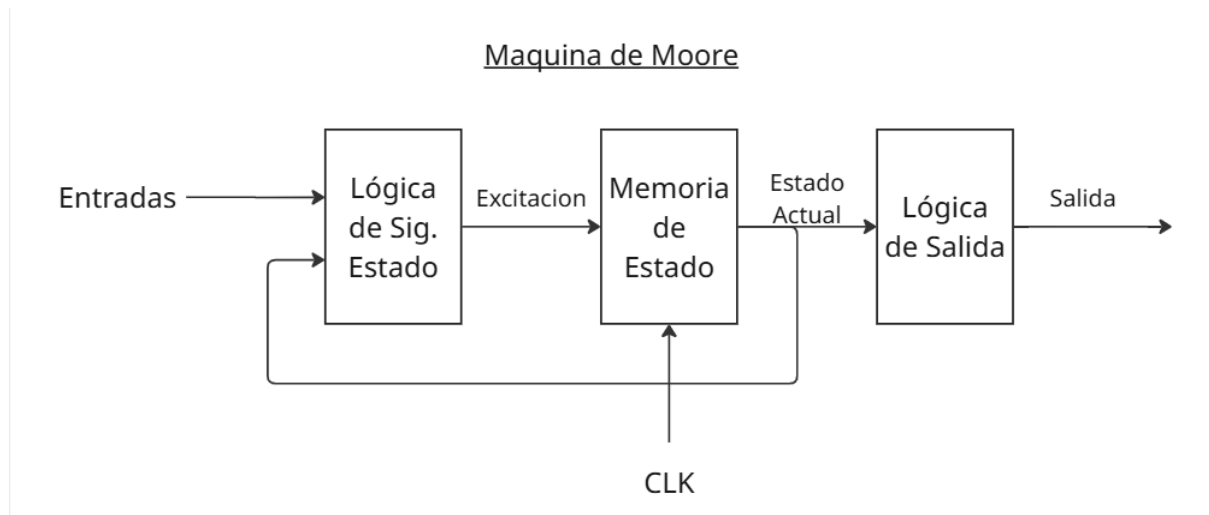


Figura 1.3 - Estructura de una Máquina de Estado Finito Sincrónica de Moore

Concepto de Datapath

Datapath o camino de datos, se refiere al conjunto de componentes de hardware que procesa datos, mueve datos entre registros y lee o escribe datos en la memoria.

El datapath trabaja en conjunto con la unidad de control, que le indica qué operación realizar en cada momento.

Resumir una FSM por el método del datapath es una forma estructurada de diseñar sistemas secuenciales, dividiendo claramente: la parte que procesa datos (datapath) y la parte que controla las operaciones (FSM).

Este método permite modular el proceso, facilita el diseño de sistemas complejos como CPUs, controladores o sistemas digitales personalizados.

Este método, facilita la síntesis de sistemas secuenciales donde las tablas de estados y salidas, las tablas de transición se vuelven poco óptimas o imposibles de hacer.

Materiales y Métodos:

Explica los procesos de diseño de una máquina de estado finito en base a estados y transiciones y por el método de datapath.

Los pasos para el diseño de una Máquina de Estado Finito sincrónica en base de los estados y transición::

1. Definir estados y salidas.
Si los mismos se presentan en las especificaciones ya están determinados, en caso contrario se definen como elección de diseño.
2. Definir la topología de la Máquina de Estado Finito.
Determinar si la misma es una Máquina de Moore o de Mealy, también ver la mejor conexión de la misma, pues para la máquina de Moore las entradas pueden interactuar de modo sincrónico o asincrónico. Elección de diseño.
3. Construir la tabla de estado/salida correspondiente a las especificaciones..

Estados	Transición			Salida		
	Entrada			Entrada		
	a	b	c	a	b	c
s_0	s_0	s_3	s_2	0	1	1
s_1	s_1	s_1	s_3	0	0	1
s_2	s_1	s_1	s_3	1	1	0
s_3	s_2	s_3	s_0	1	0	1

En una Máquina de Estados/Salidas, como en la Figura 2.1, se colocan los Estados Actuales, Entradas, Salidas y los Estados Siguiendo que dependen de las Entradas y Estados Actuales.

Figura 2.1 - Tabla de Estados/Salida de una Máquina de Estados.

- Elegir un conjunto de variables de estado y asignar combinaciones de variables de estado a los estados nombrados.
Se deben codificar los Estados y Salidas, pues los mismos se encuentran con un nombre genérico, pero para poder utilizarlos a través de alguna descripción de hardware, se los debe determinar codificándolos (en binario natural, binario con signo, hexadecimal, etc).
Obs. al utilizar una topología de una Máquina de Moore, se puede optar por elegir una codificación para los estados de igual modo que los de salida. Esto puede simplificar la Lógica de Salida.
- Sustituir las combinaciones de variables de estado en la tabla de estado/salida para crear una tabla de transición/salida
De este modo la tabla ahora no tiene variables, ni nombres genéricos, sino una codificación, de modo que los elementos de memoria pueden interpretarlos.
- Construir una tabla de excitación que muestre los valores de excitación requeridos para obtener el siguiente estado deseado para cada combinación de estado/entrada
- Realizar la síntesis lógica de los bloques combinacionales de Estados Siguiendo y Salida
De éste modo no tendremos sólo una tabla de transición, sino de expresiones lógicas, de modo que se puede representar los mismos a través de circuitos lógicos, los cuales luego se pueden implementar.
- Se implementan las funciones sintetizadas mediante circuitos lógicos(diagrama esquemático)

Los pasos para el diseño de una Máquina de Estado Finito sincrónica por el método de Datapath:

Los primeros pasos son iguales que a través de estados y transición.

Una vez diseñada la Tabla de Estados/Salidas como en la Figura 2.1.:

Se debe buscar algún tipo de patrón, de modo que se pueda resumir la Tabla en forma de expresiones, y predecir las Salidas y Estados Siguiendo en Función de las Entradas y Estados Actuales.

Es un método que ayuda a implementar funciones más complejas, con muchos más estados y más entradas que en el método de Estados y Transición.

Resultados

En el punto 2 de la actividad se pedía diseñar una máquina de estado finito que implemente un contador binario de tres bit con reset síncrono. Utilizar el enfoque basado en estados y transiciones. Realizar las síntesis de la máquina de estado finito. Realizar el diagrama de tiempo de la máquina de sintetizada.

Transcribir las indicaciones

- Contador binario de 3 bit, salida codificado a 3 bits.
- Reset síncrono, si Reset activo entonces Estado siguiente es "000".(Entrada RST)

1 Definir Estados y Salidas

Salidas: S0, S1, S2, S3, S4, S5, S6, S7

Estados: Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7

2 Definir la topología de la Máquina de Estado Finito

Se eligió para este problema una Máquina de Moore(ver Figura 1.3), ya que el reset es síncrono, la salida actual sólo depende del estado actual.

3 Construir la tabla de estado/salida correspondiente a las especificaciones.

TABLA 3.1- Tabla de Estados y Salidas

	Entradas		
	RST		
Estados	0	1	Salidas
E0	E1	E0	S0
E1	E2	E0	S1
E2	E3	E0	S2
E3	E4	E0	S3
E4	E5	E0	S4
E5	E6	E0	S5
E6	E7	E0	S6
E7	E0	E0	S7

En la Tabla 3.1 se colocó estados actuales, entradas y salidas contenido entre las columnas Estados y Salidas, bajo RST, es la zona de los Estados Siguientes, de modo que los mismos son función de los Estados y Entradas.

4 Elegir un conjunto de variables de estado y asignar combinaciones de variables de estado a los estados nombrados.

TABLA 3.2 - Codificación Salidas y Estados

Salidas	Y2	Y1	Y0		Estados	Q2	Q1	Q0
S0	0	0	0		E0	0	0	0
S1	0	0	1		E1	0	0	1
S2	0	1	0		E2	0	1	0
S3	0	1	1		E3	0	1	1
S4	1	0	0		E4	1	0	0
S5	1	0	1		E5	1	0	1
S6	1	1	0		E6	1	1	0
S7	1	1	1		E7	1	1	1

Se decidió codificar ambos dos con el mismo código, de lo que la lógica de salida se simplifique, siendo una lógica identidad,

5 Sustituir las combinaciones de variables de estado en la tabla de estado/salida para crear una tabla de transición/salida

Se sustituye las Variables en la Tabla de estado y salida(Tabla 3.1) obteniendo la Tabla 3.3

Tabla 3.3 - Tabla de Estados y Salidas, luego de reemplazar la codificación de Estados y Salidas

			Entradas				
Estados			RST		Salidas		
Q2	Q1	Q0	0	1	Y2	Y1	Y0
0	0	0	001	000	0	0	0
0	0	1	010	000	0	0	1
0	1	0	011	000	0	1	0
0	1	1	100	000	0	1	1
1	0	0	101	000	1	0	0
1	0	1	110	000	1	0	1
1	1	0	111	000	1	1	0
1	1	1	000	000	1	1	1

6 Construir una tabla de excitación que muestre los valores de excitación requeridos para obtener el siguiente estado deseado para cada combinación de estado/entrada.

7 Realizar la síntesis lógica de los bloques combinacionales de Estados Siguiente y Salida

Para la síntesis de las bloques combinacionales, se utilizó el método del mapa de Karnaugh.

Tabla 3.4- Mapa de Karnaugh referido a Q*2

RST Q2 \ Q1 Q0	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	0	0	0	0
10	0	0	0	0

De la tabla 3.4, se puede sintetizar Q*2 como:

$$Q^*2 = RST'Q2Q1' + RST'Q2Q0' + RST'Q2'Q1Q0 = RST'Q2(Q1' + Q0') + RST'Q2'Q1Q0$$

Tabla 3.5- Mapa de Karnaugh referido a Q*1

RST Q2 \ Q1 Q0	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	0	0	0
10	0	0	0	0

De la Tabla 3.5, se puede sintetizar Q*1 como:

$$Q^*1 = RST'Q1'Q0 + RST'Q1Q0' = RST'(Q1 \text{ XOR } Q0)$$

Tabla 3.6- Mapa de Karnaugh referido a Q*0

RST Q2 \ Q1 Q0	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	0	0	0	0
10	0	0	0	0

De la tabla 3.6, se puede sintetizar Q*0 como:

$$Q^*0 = RST'Q0'$$

Con esto se sintetiza los Estados Siguientes Q*2, Q*1 y Q*0

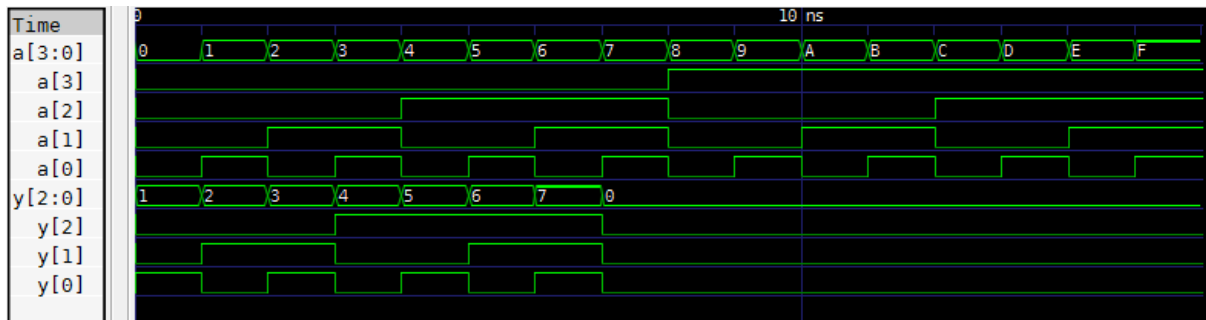


Figura 3.1 - Forma de Onda resultante al utilizar la síntesis realizada

En la Figura 3.1 se utilizó $y[2:0]$ como las Salidas o Estados siguientes, donde $y(2)=Q*2$, $y(1)=Q*1$ e $y(0)=Q*0$. Además $a[3:0]$ como el RST y Estados Actuales, donde $a(3)=RST$, $a(2)=Q2$, $a(1)=Q1$ y $a(0)=Q0$.

En el punto 3 de la actividad, se nos pide utilizar la técnica de datapath para diseñar un contador binario de 4 bit con reset, carga y entrada de habilitación. Todas las entradas de control son sincrónicas. Realizar la síntesis del contador, utilizando para la suma un sumador de 4 bit con propagación de acarreo. Realizar el diagrama de tiempo de la máquina sintetizada.

Si se hubiese utilizado el método anterior también para éste punto no sería la mejor opción de trabajar.

Porque:

1. Tiene salida de 4 bits, un bit más que el anterior por lo que pasa de 8 estados a 16 estados, eso acompleja el mapa de Karnaugh a una de 5 variables.
2. Este contador tiene además de Reset sincrónico, cuenta con una habilitación y además una carga de valor también un número de 4 bits

De todo ello, hay que buscar un método de trabajo que simplifique las cosas, por ello se utiliza el método de datapath.

Especificaciones:

- Entradas: RST(Reset sincrónico), hab(Habilitación), Carga, Valor (número de 4 bits)
- Salidas: un número de 4 bits.
- RST vuelve a "0000"
- hab comienza a contar, o va al estado siguiente
- carga carga un valor como estado siguiente

1 Definir la Arquitectura

Se eligió utilizar una Máquina de Moore(ver Figura 1.3)

2 Seleccionar los códigos

Tabla 4.1 - Tabla con codificación de Salidas y Estados

Salidas	Código		Estados	Código
S0	0000		S0	0000
S1	0001		S1	0001
S2	0010		S2	0010
...
S15	1111		S15	1111

3 Construir la tabla de estado/salida correspondiente a las especificaciones, definir las expresiones

No haremos la tabla, nos guiaremos por las especificaciones para deducir expresiones de los estados siguientes.

Estas expresiones las haremos en lenguaje VHDL ya que es más fácil orientarse.

- Reset que devuelve "0000"

```
S <= "0000" when RST;
```

- Habilitación comienza a contar o va al estado siguiente

```
S <= estado_sig when hab;
```

- Carga carga un valor como estado siguiente

```
S <= valor when carga;
```

- Si no tengo ninguna señal me mantengo en el estado actual

```
S <= estado;
```

De esta manera cada una de las especificaciones se la puede codificar como expresiones en vez de hacer una tabla.

Ahora juntamos todo:

```
S <= "0000" when RST else
    estado_sig when hab else
    valor when carga else
    estado ;
estado_sig <= estado + 1 - - pues es un contador;
```

Esta arquitectura utilizada, tiene una jerarquía implícita, donde se necesita saber el resultado del primer "when else" para continuar y ver los demás, por lo que aquí las entradas de control están jerarquizadas y es decisión de diseño el orden de colocarlas, siempre evitando tener ambigüedades, en caso de tenerlas se puede consultar con el cliente para quien deba hacerse esta proyecto.

4 Implementar dichas expresiones como un bloque combinacional

Conclusiones

Con esta actividad pudimos comparar la complejidad de implementar una máquina de estados finitos mediante el uso de tablas de estados y salidas y a través del método de datapath.

El desarrollo mediante tablas de estados resultó ser más simple, rápido y visualmente comprensible. Este método permite diseñar máquinas sin conocer en detalle su funcionamiento interno, ya que a partir de la tabla se puede sintetizar y construir el circuito secuencial correspondiente. Sin embargo, su aplicación es práctica solo para máquinas con pocas entradas y un número reducido de estados, ya que la complejidad crece de forma exponencial, volviéndose un método ineficiente y poco manejable para sistemas más grandes.

Por otro lado, el método de datapath requiere identificar el patrón del proceso para resumir el comportamiento de la máquina. Una vez hallado dicho patrón, el diseño se simplifica considerablemente, sin importar la cantidad de entradas o estados. Además, permite predecir el valor de los estados a partir de las entradas, sin necesidad de elaborar una tabla completa de estados y salidas.

Este enfoque supera las limitaciones del método de tablas, ya que posibilita trabajar con mayor número de entradas y estados de manera más eficiente, siempre que se haya encontrado el patrón y las expresiones correspondientes.

Referencias

- John F. Wakerly. (2005). *Digital Design: Principles and Practices*. Prentice Hall. Partes del Capítulo 7 y 8
- Harris & Harris (2013). *Digital Design and Computer Architecture*. Partes del Capítulo 3