

IMPLEMENTACIÓN EN MICROELECTRÓNICA DE UN DETECTOR DE SECUENCIA PROGRAMABLE

Edgardo Ricci, Nahuel Sede, Bruno Diez, Sebastián Rossi Grad, Walter Aróztegui, José Rapallini

UIDET – CeTAD, Calle 116 y 48 Depto. de Electrotecnia (2do piso)
Facultad de Ingeniería, UNLP. La Plata (1900). Email: [josrap@ing.unlp.edu.ar](mailto:jrap@ing.unlp.edu.ar)

INTRODUCCIÓN

En el proyecto de Microsistemas Colaborativos que se desarrolla en la UIDET–CeTAD, se proponen el diseño, análisis y ensayo de distintos módulos MEMS -(Sistemas Microelectromecánicos)-, su control y actuación con dispositivos de microelectrónica. Algunos de estos circuitos aplicados simples son desarrollados por alumnos que promedian la carrera de Ingeniería en Electrónica, cómo es el caso.

Este trabajo se enmarca dentro de la caracterización de un estándar de estos microsistemas autónomos especializados que puedan interactuar entre sí colaborando con sus distintas funciones en la resolución de un problema o acción específica sobre el medio que comparten, de la misma manera que funcionan una variedad de células especializadas dentro del torrente sanguíneo, aportando cada una característica diferente sobre un desempeño común. Se pretende que tales microsistemas sean auto-sustentados y puedan captar información del medio o accionar sobre él de acuerdo con las decisiones tomadas en base a los resultados comunicados por microsistemas vecinos con funcionalidades diferentes.

Lo presentado aquí se constituye como un subsistema de control para la etapa de comunicaciones y decisión. Tales sistemas pretenden efectuar acciones cooperativas en base a módulos de sensado y actuación sobre el medio. Módulos de sensado captan determinadas variables del medio y deben informar mediante un método sencillo de comunicación si tales variables se encuentran o no en el ambiente. Otros microsistemas vecinos deberían obtener esta información y decidir la activación de módulos específicos para la acción requerida en el medio.

Un sistema sencillo de comunicación supone el envío al medio de determinadas órdenes codificadas en números binarios, de manera continua y repetitiva. Con distintos microsistemas de sensado, pueden ser recibidas varias órdenes y los módulos receptores deberían decidir si corresponde a ellos actuar o no. El sistema de recepción y decisión más sencillo consiste en una máquina de estado que diferencie secuencias de bits determinados y los utilice de la misma manera que un procesador traduce un código de operación. Tal secuencia sólo permitirá la actuación de determinados módulos, así como secuencias distintas lo harán sobre otros. Cada uno responderá a una determinada secuencia, por lo que la máquina de estados se basará en una secuencia básica programable para cada sistema de acuerdo a su capacidad diferenciada de actuación. Una secuencia de 4 bits nos permitirá determinar la actuación de grupos de sistemas de 16 clases distintas de sensado o acción sobre el medio.

La etapa de comunicación propiamente dicha se debe determinar de acuerdo a las características de propagación del medio donde actúen los grupos de sistemas. En este trabajo sólo se centra en el circuito utilizado para la captación y decisión de las órdenes una vez recibido el flujo secuencial de bits.

Se llevará cabo entonces, un análisis del procedimiento utilizado para la síntesis del detector de secuencia (DDS) programable de 4 bits. Se analizarán los procedimientos posibles a seguir y se dará una explicación del porqué del camino seguido.

EXPERIMENTACIÓN, DISCUSIÓN Y RESULTADOS

I. CONFIGURACION DEL CIRCUITO

El fin de este detector es el siguiente: En base a una entrada de datos en serie, el usuario podrá programar según convenga, la búsqueda de una secuencia de 4 bits cualesquiera, y el circuito se encargara de avisar (dando una señal en la salida de un 1 digital) cuando encuentra dicha secuencia. Este aviso de detección de correspondencia se presentara en alto únicamente por un ciclo de Clock, luego del mismo se procederá nuevamente a la búsqueda de la secuencia seleccionada, no permitiendo en este caso el solapamiento de bits.

Para la resolución de este problema se procede a la síntesis de una máquina de estados finitos (MEF), donde, como es conocido existen dos tipos de planteo posibles, Moore y Mealy [1][2][3]. Este tipo de dispositivo requiere tanto de compuertas lógicas como de elementos de memoria (flip-flops) y de una entrada de Clock.

La diferencia entre estos dos tipos de máquina es que la salida del circuito, llamémosle Z, en el primer caso es una función lógica que depende solamente de los estados de los flip-flops, mientras que en el modelo de Mealy, la salida depende tanto de los estados de los flip-flops, como de los datos de entrada.

A. Síntesis con Mealy

En principio se decide la utilización del modelo de Mealy ya que presenta un menor número de elementos de memoria, lo que a pesar de una mayor lógica combinatoria en las distintas funciones, redundaría en una menor superficie en su implementación física. Se observa en el diagrama de la figura 1 para una máquina de tipo Mealy que el número de estados requerido para el funcionamiento es cuatro, por lo que se necesitarían únicamente dos flip-flops. En el diagrama se utilizan las siguientes notaciones:

- A-B-C-D: Secuencia que se desea leer, donde A es el bit más significativo y D el menos significativo.
- S0 a S3: Estados de los Flip-Flops.
- E: Entrada de datos.

Quedando las ecuaciones de las funciones lógicas obtenidas:

$$D1 = /Q1 ./Q0 ./A ./E ./B + E.B + Q1 ./Q0 ./A .(C.E + /C ./E + /B.C) + Q1.Q0.A ./E ./B + E.B + Q1 ./Q0.A.(C.E + /C ./E + B./C) + Q1.Q0.A.B.C.E \quad (1)$$

$$D0 = /Q1 ./Q0 ./A ./E + A.E + /Q1.Q0 ./A.B ./E + Q ./B.E + Q1 ./Q0 ./A .(C.E + B.C + /C ./E) + Q1 ./Q0.A.(C.E + /B ./C + /C ./E) + Q1.Q0 ./A.D ./E + Q1.Q0.A.(E ./D + B.C.D.E) \quad (2)$$

$$Z = Q1.Q0.(E.D + /E ./D) \quad (3)$$

Utilizando un número reducido de pruebas se pudo verificar que la MEF en principio funciona. Por lo que se procedió a su síntesis a partir de un software de CAD.

En su simulación se detectan problemas debido a que la lógica de una máquina de Mealy necesita una mejor sincronización entre el flujo de entrada de datos y los cambios en los estados, por lo tanto se presentaban falsas lecturas y glitches. Este problema ocasiona que la conveniencia prevista para la síntesis a partir de Mealy, ya no sea preponderante.

B. Síntesis por Moore:

Se procede entonces a la síntesis a partir de una máquina de estados Moore, que tendrá más elementos de memoria, pero en base a que la salida ahora sólo dependerá de los estados, simplificará las funciones lógicas, así como la sincronización, a pesar de un

pequeño incremento en el plano de los layout al implementarlo físicamente. Debido a que la secuencia consta de 4 bits, se requerirían cuatro estados, cuya función sería la detección del Bit correspondiente y un quinto estado el cual sería únicamente utilizado para indicar que la secuencia fue detectada. La cantidad de flip-flops necesaria (3) nos brindaría 8 estados posibles, utilizando únicamente 5 de estos.

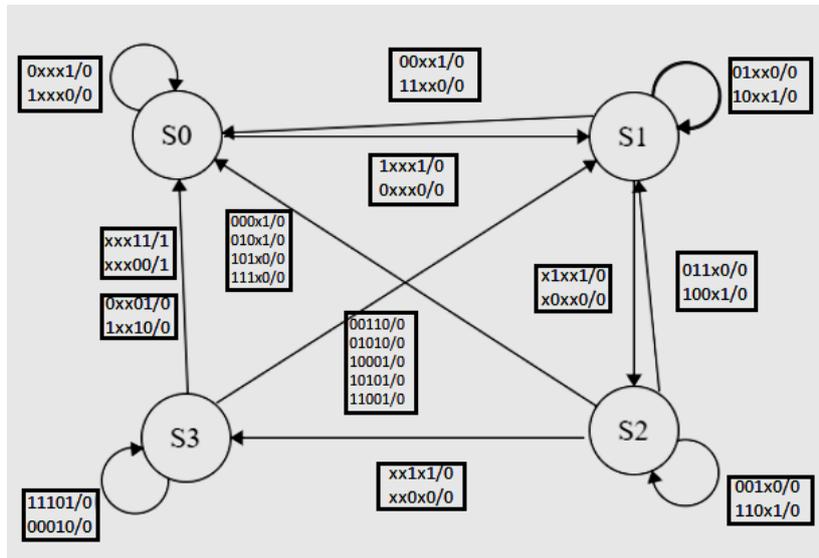


Fig.1 Diagrama de estados Mealy

A diferencia de Mealy la salida de esta máquina presenta una salida sincrónica con los estados y cumple perfectamente con los requisitos especificados del DDS. Por lo tanto procedemos a la síntesis de la misma haciendo uso del diagrama de la figura 2. Las funciones a sintetizar son las siguientes:

$$D0 = /Q2 ./Q1 ./Q0 ./(A ./E + A.E) + /Q2 ./Q1.Q0 ./(A.B ./E + A ./B.E) + /Q2.Q1 ./Q0 ./A.(C.E + B.C + /C ./E) + /Q2.Q1 ./Q0.A.(C.E + /B ./C + /C ./E) + /Q2.Q1.Q0 ./(A.D.E + A ./D.E) \quad (4)$$

$$D1 = /Q2 ./Q1.Q0 ./A ./(B ./E + B.E) + /Q2 ./Q1.Q0.A ./(B ./E + B.E) + /Q2.Q1 ./Q0 ./A.(C.E + B.C + /C ./E) + /Q2.Q1 ./Q0.A.(C.E + /B ./C + /C ./E) + /Q2.Q1.Q0.A.B.C.E ./D(2) \quad (5)$$

$$D2 = /Q2.Q1.Q0 ./(A.D ./E + A ./D.E) \quad (6)$$

$$Z = Q2 ./Q1 ./Q0 \quad (7)$$

Luego de esto se procedió a verificar la frecuencia máxima de funcionamiento del circuito. Para ello simplemente se realizaron varias simulaciones aumentando secuencialmente la frecuencia del reloj y de los datos en cada una de ellas.

Se comprobó que alrededor de 50 MHz de trabajo se encuentra la frecuencia máxima de funcionamiento. Este tope de frecuencia se debe a varios factores [4]:

- Primero por el tiempo de retardo de cada compuerta lógica del circuito, sabemos que cada una de ellas implica cierto tiempo entre la lectura de las entradas y la reacción de la salida, es así que este se convierte en el primer factor en limitar la frecuencia.

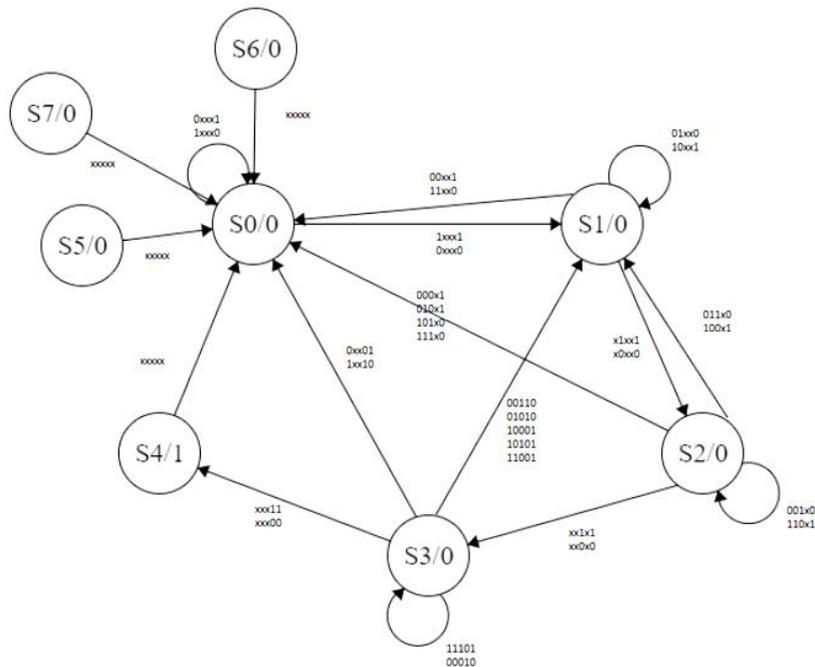


Fig.2 Diagrama de estados Moore

- En segundo lugar se encuentran los flip-flops. Para el debido funcionamiento de estas compuertas es necesaria cierta estabilidad en el tiempo en los datos de entrada para la correcta lectura de los mismos. Estos tiempos son conocidos como tiempos de set up y de hold. El tiempo de set up es el tiempo mínimo previo al flanco de reloj en el que el dato tiene que mantenerse estable para una correcta lectura, y el tiempo de hold es el tiempo posterior al flanco necesario de estabilidad del dato. En definitiva estos tiempos nos limitan cuanto pequeño puede ser el período de reloj y, en consecuencia, su frecuencia.

- Por último otro factor culpable del límite de frecuencia de trabajo son las capacidades parásitas de los transistores. Se sabe que la tecnología utilizada es CMOS -(Metal-Óxido-Semiconductor-Complementario)-, por ende cada compuerta lógica está formada por transistores PMOS -(MOS de canal P)- y NMOS -(MOS de canal N)-, los cuales acarrean consigo capacidades parásitas que, a altas frecuencias, toman valores significativos en los cálculos de los circuitos, y comienzan a alterar el paso de los datos, llevando al mal funcionamiento del dispositivo.

II. DISEÑO DEL LAYOUT

En primer lugar se procedió a plantear el diseño global. Se utilizó una tecnología de 0,5 nm y se acordó separar la máquina en 3 partes principales. Esto disminuye de manera significativa el tamaño del pitch, permitiendo trabajar con división de las conexiones. En el caso que no se hubiera separado, esto habría implicado ampliar el pitch hasta que se tenga por lo menos 24 conexiones sin contar las intracompuertas [4][5][6]. Separando los flip-flop utilizamos un pitch que contenga 15 conexiones.

De abajo hacia arriba se utilizó el siguiente esquema de conexión:

- Para ffd D0 se utilizó la línea 15 para clock, A-1 B-2 C-3 D-4 E-5 /Q0-6 Q0-7 dejando las restantes para las conexiones entre compuertas.
- Para ffd D1 se utilizó un esquema similar al anterior pero con las negaciones de las entradas, el esquema de conexión quedó: clock-15 /A-1 /B-2 /C-3 /D-4 /E-5 /Q1-6 Q2-7
- Para ffd D2 se utilizó la línea 15 para clock, como era la línea más chica se aprovechó el espacio sobrante para los inversores de las entradas y la salida.

- Para generar las conexiones se utilizó metal 2 con un ancho de 4µm para poder conectar entre metal 1, 2 y 3 respetando las condiciones que la tecnología utilizada aconseja. La separación entre cada conexión es la mínima ya que al ampliar el pitch se aumenta de manera significativa la resistencia de la capa de polisilicio. Las conexiones verticales se efectuaron con metal 3 de ancho de 6µm. Se decidió utilizar una estructura de peine de manera que los vcc y gnd queden conectados. El primer diseño planteaba también un segundo corte en cada flip flop pero no se pudo llevar a cabo ya que las conexiones con metal 3 requieren más separación que el resto de los metales.

- Una vez definidas las bases se procedió a generar de manera independiente cada una de las compuertas que se necesitarían y comprobar su correcto funcionamiento. Terminado esto se realizó la implementación de modo inverso, es decir, desde las etapas finales del circuito hasta las iniciales. Empezando con los flip flops se hicieron las conexiones separando al diseño en columnas cuyas conexiones sean similares para generarlas de manera más corta.

- Por ejemplo se alinearon en columnas todas las compuertas cuyas entradas incluían a Q0, Q1, Q2 y sus negados para después poder utilizar los canales de conexión 6 y 7 en el resto del circuito. Se generó un archivo de respaldo para comprobar que las conexiones estuvieran bien en pequeños grupos y una vez terminado se comparó con el original.

Completado el circuito y comprobado que su funcionamiento fuera igual al esquemático que se muestra en la figura 3, se procedió a terminar los detalles. Como se muestra en la figura 4, se agrupó todo de manera que ocupara el menor espacio posible, se ensancharon los canales de vcc y gnd para disminuir su resistencia.

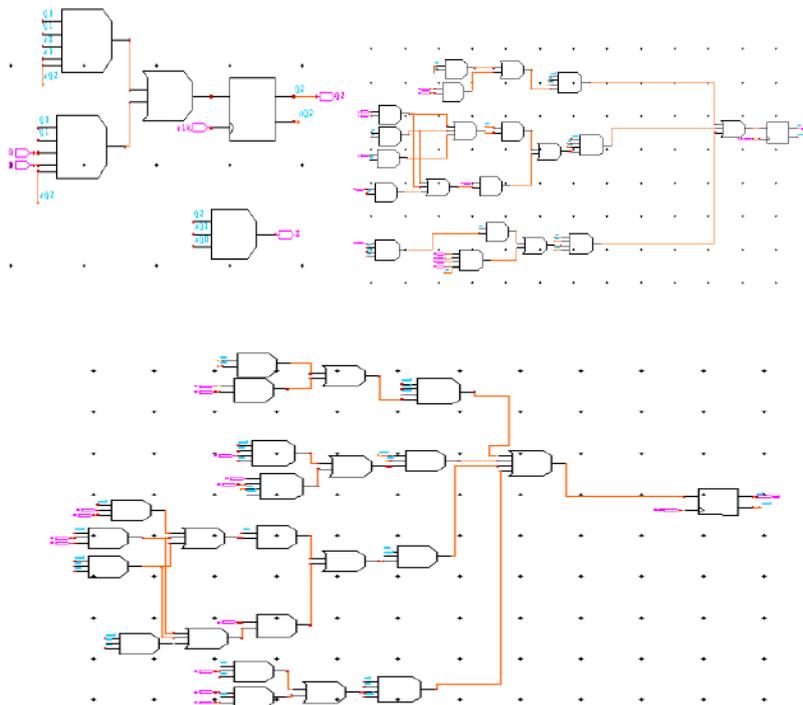


Fig.3 Circuito de simulación

CONCLUSIONES

Se ha conseguido un diseño correcto con un grado de dificultad interesante para alumnos en formación. La complejidad obtenida no es tanto en cuanto a los circuitos digitales en sí, sino a la descripción en layout obtenida, con las complicaciones que estas conllevan. Además de la utilidad que se espera para este diseño, el mismo sienta una base de conocimiento en el grupo que aquí trabajó, para continuar hacia diseños más complejos

como unidades de control en la línea de diseños de procesadores, que también se cuenta como una de las tareas abarcadas por el presente proyecto al que se ha referido anteriormente.

Se espera en un futuro, continuar con el desarrollo del mismo, con la simulación del layout para luego completar detalladamente la descripción física e integrar con el diseño completo propuesto para su fabricación.

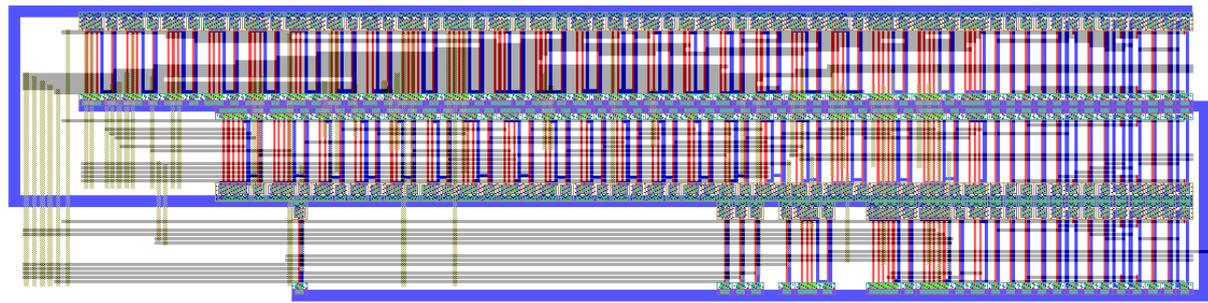


Fig 4. Layout del DDS

BIBLIOGRAFÍA

- [1] Taub – Schilling, Digital integrated circuits. McGraw – Hill. 1982.
- [2] H. Taub, Circuitos digitales y microprocesadores. McGraw – Hill. 1982.
- [3] R. Tocci, Sistemas Digitales: principios y aplicaciones. Prentice Hall. 1996
- [4] P. Julián. Circuitos Integrados Digitales CMOS, Análisis y diseño. Alfaomega. 2015.
- [5] B. Razavi, Fundamentals of microelectronics, 3rd Edition, Wiley. 2014
- [6] R. J. Baker, CMOS Circuit Design, Layout, and Simulation, 3rd Edition, 3 edition. Piscataway, NJ : Hoboken, NJ: Wiley-IEEE Press, 2010.