

INTRODUCCIÓN DE DISPOSITIVOS PROGRAMABLES EN PRÁCTICAS DE LABORATORIO

*Carlos J. Jiménez, Carmen Baena, Manuel Valencia, Enrique Ostúa
Universidad de Sevilla - Instituto de Microelectrónica de Sevilla.*

En esta comunicación se presenta la realización de una práctica de la asignatura Estructura de Computadores de primer curso de Ingeniería Informática en la que introduce el uso de dispositivos programables. En dicha práctica se realiza un multiplicador secuencial 4x4, diseñándolo con la herramienta ISE WebPack de Xilinx e implementándolo con placas de desarrollo de Xilinx.

1. Introducción

En esta comunicación se presenta una propuesta de implementación práctica para la docencia de diseño lógico usando tecnología compleja, con orientación a la nueva estructura de créditos europeos ECTS y para un curso universitario inicial en diseño digital. Este trabajo surge, por una parte, al tener en cuenta que la cuantificación del ECTS se nutre de actividades múltiples (docencia teórica y aplicada, trabajo personal del alumno, etc.), centrándose en las prácticas de laboratorio por ser el aspecto más innovador, y, por otra, al considerar los avances de la tecnología en el campo del diseño digital con su indudable crecimiento de complejidad. Este aumento de complejidad en la tecnología afecta directamente, entre otros, a los siguientes aspectos: el propio sistema digital, el uso de herramientas de diseño, la tecnología de implementación del sistema electrónico y el entorno de test adecuado.

La propuesta que presentamos en este trabajo persigue que un alumno de un curso inicial en diseño digital sea capaz de recorrer, en una única sesión de laboratorio de dos horas de duración, todo el proceso de diseño digital avanzado sobre FPGA para implementar un multiplicador secuencial y comprobar su funcionamiento real (no simulado). Esto requiere diversas actividades docentes: enseñanza-aprendizaje teórica en el aula, aplicación al circuito bajo estudio, desarrollado de trabajo personal por el alumno y preparación de diseños por los profesores. Esta propuesta, en nuestro caso particular, se lleva a cabo con éxito en el segundo cuatrimestre del primer curso de las titulaciones de Informática.

Los objetivos perseguidos son múltiples. En primer lugar, conceptualmente está centrada en una estructura de multiplicación digital (sumas y desplazamientos a la derecha) cuyo uso, aunque consolidado a nivel científico, no forma parte de la materia habitual de los cursos de grado sobre “diseño digital” [1]. En segundo lugar, como diseño práctico de un sistema digital a nivel RT, es un ejemplo de no muy alta complejidad para practicar con una estructura de datos y control. En tercer lugar, el alumno adquirirá experiencia cuasi-profesional al diseñar un sistema digital usando una herramienta comercial avanzada. Por último, el alumno podrá testar el sistema diseñado mediante una placa de desarrollo específica, comprobando tanto su operación global entrada-salida como la operación ciclo a ciclo.

En lo que sigue se describe, en primer lugar el comportamiento del multiplicador secuencial así como los distintos componentes en los que se ha dividido. A continuación se describen brevemente los conceptos teóricos explicados en las clases de aula necesarios para la realización de la práctica. Se sigue

con la descripción de las tareas a realizar por el alumno, tanto de preparación de la práctica como de realización en el laboratorio. Finalmente se extraen algunas conclusiones.

2. Descripción del multiplicador secuencial

El multiplicador secuencial es un sistema digital que realiza la multiplicación de dos números, en nuestro caso de cuatro bits, mediante una sucesión de sumas y desplazamientos. El tamaño del multiplicador se ha limitado a palabras de 4 bits por placa debido, únicamente, al tamaño de la placa de prueba. El esquema de funcionamiento que sigue este tipo de multiplicadores se muestra en la figura 1.

Para su diseño, se ha dividido el circuito completo en tres bloques: una unidad de datos, una unidad de control y un circuito denominado CGXS (figura 2). La unidad de datos se encarga del manejo de los datos a multiplicar, su almacenamiento, sumas, desplazamientos, etc. La unidad de control genera las señales que controlan el funcionamiento de la unidad de datos y el circuito CGXS es el encargado de la generación de la señal de comienzo para la unidad de control. La señal generada por el circuito CGXS, denominada XSCICLO, es una señal de un pulso a '1' durante un ciclo de reloj generada a partir de la activación de la señal START que es una señal activa en alto pero con una duración de un número indeterminado de ciclos de reloj. Con objeto de que la práctica sea abordable en el tiempo que los alumnos están en el laboratorio, el bloque de datos y el bloque CGXS les serán dados al alumno ya diseñados, de forma que éste únicamente tenga que diseñar el bloque de control. Para la realización de este diseño se le impone que emplee para la codificación de los estados un algoritmo "one-hot", es decir, un biestable por cada estado.

$$\begin{array}{r}
 \begin{array}{r}
 \times \\
 A_3 \quad A_2 \quad A_1 \quad A_0 \\
 B_3 \quad B_2 \quad B_1 \quad B_0 \\
 \hline
 C_{out1}=0 \quad A_3 \cdot B_0 \quad A_2 \cdot B_0 \quad A_1 \cdot B_0 \quad A_0 \cdot B_0 \\
 \quad \quad \quad \parallel \quad \quad \parallel \quad \quad \parallel \quad \quad \parallel \\
 \quad \quad \quad P_{13} \quad P_{12} \quad P_{11} \quad P_{10} \quad \text{Obtención del primer producto parcial } (A_{3-0} \times B_0)
 \end{array} \\
 + \quad \begin{array}{r}
 C_{out1} \quad A_3 \cdot B_0 \quad A_2 \cdot B_0 \quad A_1 \cdot B_0 \quad P_{10} \quad \text{Desplazamiento a la derecha del primer producto parcial} \\
 A_3 \cdot B_1 \quad A_2 \cdot B_1 \quad A_1 \cdot B_1 \quad A_0 \cdot B_1 \\
 \hline
 C_{out2} \quad P_{23} \quad P_{22} \quad P_{21} \quad P_{20} \quad \text{Obtención del segundo producto parcial}
 \end{array} \\
 + \quad \begin{array}{r}
 C_{out2} \quad P_{23} \quad P_{22} \quad P_{21} \quad P_{20} \quad \text{Desplazamiento a la derecha del segundo producto parcial} \\
 A_3 \cdot B_2 \quad A_2 \cdot B_2 \quad A_1 \cdot B_2 \quad A_0 \cdot B_2 \\
 \hline
 C_{out3} \quad P_{33} \quad P_{32} \quad P_{31} \quad P_{30} \quad \text{Obtención del tercer producto parcial}
 \end{array} \\
 + \quad \begin{array}{r}
 C_{out3} \quad P_{33} \quad P_{32} \quad P_{31} \quad P_{30} \quad \text{Desplazamiento a la derecha del tercer producto parcial} \\
 A_3 \cdot B_3 \quad A_2 \cdot B_3 \quad A_1 \cdot B_3 \quad A_0 \cdot B_3 \\
 \hline
 C_{out4} \quad P_{43} \quad P_{42} \quad P_{41} \quad P_{40} \quad \text{Obtención del cuarto producto parcial}
 \end{array} \\
 C_{out4} \quad P_{43} \quad P_{42} \quad P_{41} \quad P_{40} \quad \text{Desplazamiento a la derecha del cuarto producto parcial} \\
 \hline
 \mathbf{Resultado\ final: \quad C_{out4} \quad P_{43} \quad P_{42} \quad P_{41} \quad P_{40} \quad P_{30} \quad P_{20} \quad P_{10}}
 \end{array}$$

Figura 1. Algoritmo de multiplicación basados en sumas y desplazamientos a la derecha

El funcionamiento global de este multiplicador es el siguiente. La primera vez que se dispone a multiplicar, el usuario deberá realizar un reset del sistema activando la señal destinada a ello (RESETA). Posteriormente, activará la señal START dando así un '1' lógico al circuito CGXS, el cual proporciona la señal de comienzo para el controlador, XSCICLO. Una vez que el controlador recibe un '1' por su línea de entrada, XSCICLO va proporcionando, ciclo a ciclo de reloj, las señales de control que necesita cada elemento de la unidad de datos. Estas señales son las que gobiernan la carga y el desplazamiento de los registros en los que se almacenan los datos de entrada así como los datos en los que se van almacenando los distintos productos parciales.

La estructura del circuito de la unidad de datos, se muestra en la figura 3, Está formada por tres registros (A, SUMH y SUML), un sumador paralelo de 4 bits y un contador módulo 4 (CONT). La tabla de descripción de cada uno de estos bloques se muestra en la figura 4.

Para la realización de la multiplicación siguiendo el algoritmo de sumas y desplazamientos mostrado anteriormente, en el bloque de datos se siguen los siguientes pasos: tras una carga en paralelo de los datos A y B en los registros A y SUML respectivamente, se procede a analizar cuál es el bit menos significativo de B (SUML[0]). Según sea '0' o '1', se realiza una suma del dato de A con el dato presente en un tercer registro (SUMH) destinado a almacenar los productos parciales de la multiplicación. Tras la carga de la suma, en su caso en SUMH, se realiza una operación de desplazamiento a la derecha de los registros SUMH y SUML. Así, SUMH desplaza su bit LSB a SUML dado que en la próxima operación de suma para la obtención del siguiente producto parcial, este bit no debe sumarse sino que pasa a convertirse en el siguiente bit del resultado, resultado que se va almacenando en el registro SUML. Este proceso se realizará 4 veces, ya que el dato B posee 4 bits. Para ello, se ha dispuesto de un contador módulo-4 y será la señal de acarreo de este contador la que indique en fin de la multiplicación. El resultado final de la operación quedará guardado en los registros SUMH y SUML.

El último bloque que queda por especificar de nuestro sistema es CGXS. Consiste en un circuito que, a partir de la activación de una señal START a '1' lógico, proporciona un pulso positivo de exactamente un ciclo de reloj de duración (XSCICLO). El circuito que lo implementa se muestra en la figura 5 (a), En la parte (b) de dicha figura se muestra un diagrama temporal con el comportamiento de sus señales. En este diagrama temporal se observa que la activación de RESETA pone a '0' los biestables, incluido el asíncrono ($q_{SR} = '0'$). Cuando se activa START el biestable asíncrono almacena un '1' ($q_{SR} = '1'$), valor que ya no abandonará hasta que se termine la operación activando FIN. (Obviamente también se borrará si se activa RESETA). Una vez que $q_{SR} = '1'$ el primer ciclo de CLK hará que ese '1' se traslade a XSCICLO, pero sólo durante un ciclo de CLK.

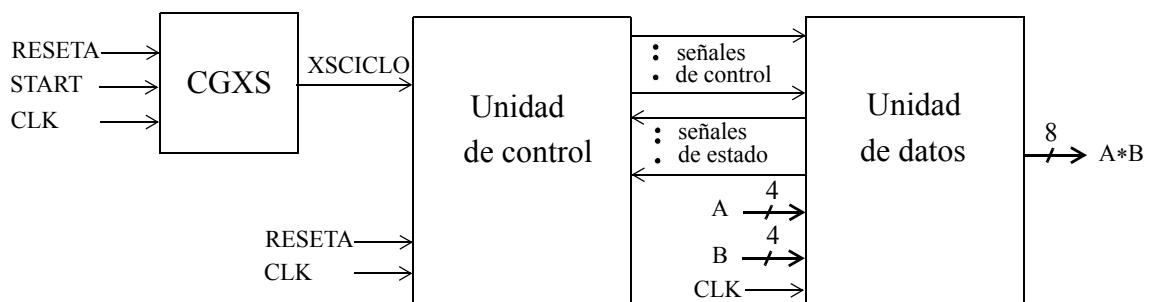


Figura 2. Representación a nivel de bloques del multiplicador implementado en el CI.

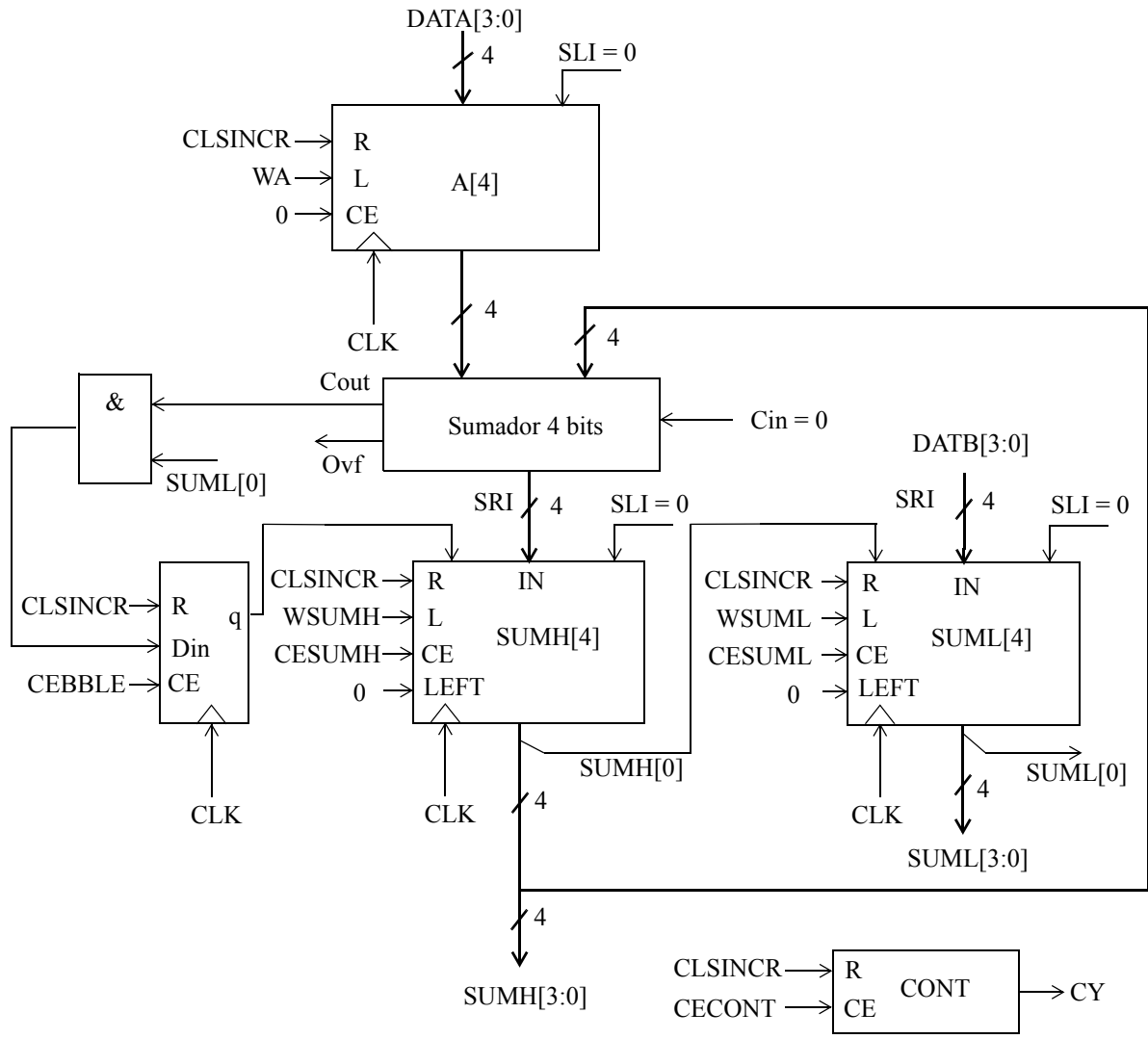


Figura 3. Unidad de datos del multiplicador secuencial.

Registro A:				Registros R (SUMH, SUML):				
R	L	CE	operación	R	L	CE	LEFT	operación
0	0	0	$A \leftarrow A$	0	0	0	0	$R \leftarrow R$
1	-	-	$A \leftarrow 0$	1	-	-	-	$R \leftarrow 0$
0	1	-	$A \leftarrow DATA[3:0]$	0	1	-	-	$R \leftarrow IN$
0	0	1	$A \leftarrow SHL(A, SLI)$	0	0	1	0	$R \leftarrow SHR(R, SRI)$
				0	0	1	1	$R \leftarrow SHL(R, SLI)$

Biestable D (DBBLE):			Contador módulo-4 (CONT)		
R	CE	operación	R	CE	operación
1	-	$DBBLE \leftarrow 0$	1	-	$CONT \leftarrow 0$
0	0	$DBBLE \leftarrow DBBLE$	0	0	$CONT \leftarrow CONT$
0	1	$DBBLE \leftarrow Din$	0	1	$CONT \leftarrow CONT+1$

Figura 4. Descripción de los registros que forman la unidad de datos del multiplicador.

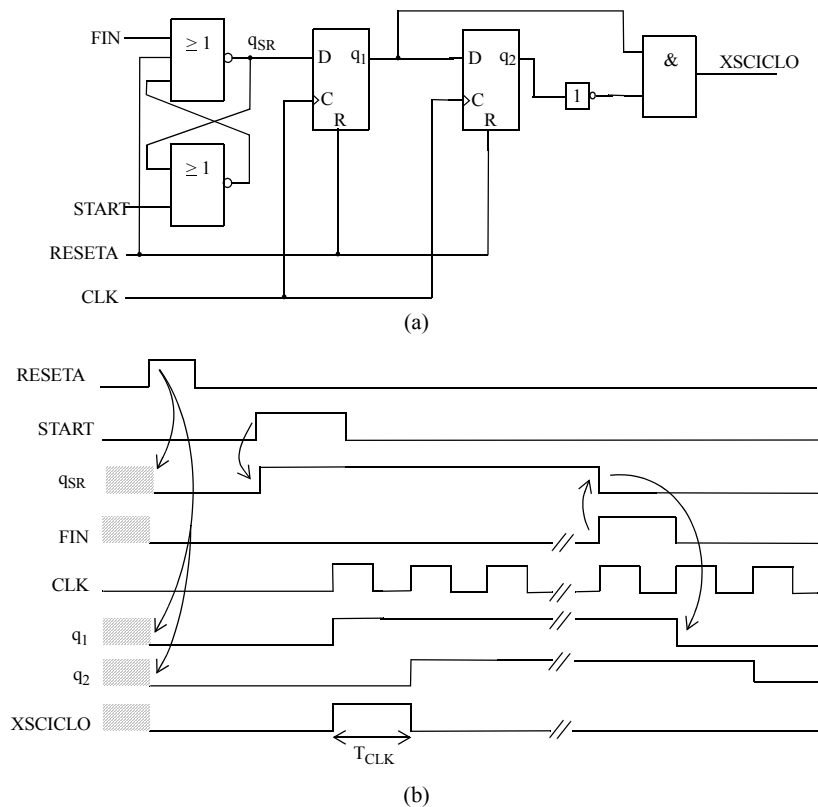


Figura 5. Bloque CGXS, (a) Esquemático, (b) Comportamiento temporal.

3. Tareas en aula

Antes de exigirle al alumno la realización del estudio teórico, ha habido que proporcionarle, en las clases de aula, los conocimientos necesarios para su realización. En estas clases se desarrolla la teoría de diseño lógico a nivel RT para sistemas con ruta de datos y control mediante cartas ASM y el diseño de dichos sistemas empleando un biestable por estado. Esta teoría se aplicará en clases de problemas al diseño de la ruta de datos del multiplicador secuencial objeto de la práctica usando los componentes con los que será implementado posteriormente [2].

Muchos de los sistemas digitales secuenciales pueden diseñarse separando entre sus componentes una ruta de datos y una unidad de control. El diseño de estos sistemas puede hacerse, además de por el método clásico de diagramas de estado, usando el llamado “Algoritmo de Máquina de Estados” (en inglés *Algorithmic State Machine*, ASM). Este algoritmo especifica mediante un diagrama de flujo los pasos del procedimiento y los caminos de decisión. Al ser un diagrama de flujo para un algoritmo hardware debe tener unas características especiales que ligen de cerca el desarrollo hardware del algoritmo.

El diagrama ASM contiene tres elementos básicos: la caja de estado, la caja de decisión escalar y la caja de salida condicional. Un estado en la secuencia de control se indica mediante una caja de estado, que se dibuja como un rectángulo y que contiene una operación de transferencia de registro o señales de salida que se activan cuando la unidad de control está en un determinado estado. La caja de decisión escalar describe el efecto de una entrada en el control. Tiene la forma de un rombo con dos caminos de

salida. La condición de entrada es una variable binaria de entrada o una expresión booleana que depende únicamente de las entradas.

El tercer elemento, la caja de salida condicional es exclusiva del diagrama ASM. Tiene una forma ovalada que la diferencia de la caja de estados. El camino de entrada a una caja de salida condicional, procedente de una caja de estado, debe pasar a través de una o más cajas de decisión. Si la condición especificada en el camino, a través de las cajas de decisión que conducen de una caja de estado a una caja de salida condicional, se cumple, se activan las transferencias de registros o salidas enumeradas dentro de la caja condicional.

4. Tarea a realizar por el alumno: estudio teórico

Una vez estudiado el funcionamiento del multiplicador secuencial, cada alumno debe realizar, antes de acudir al laboratorio de prácticas, la carta ASM de la unidad de control y el circuito que la implemente utilizando para los estados una codificación “one-hot” (un biestable por cada estado).

El comportamiento del bloque de control es el siguiente: su ciclo de funcionamiento comienza con la espera en la activación de la señal XSCICLO. La primera actuación consiste en la activación de la señal CLSINCR (clear de los registros del bloque de datos). Una vez realizado el clear se produce la activación de las señales de carga de los registros con los datos a multiplicar (WA y WSUML). A partir de aquí comienza el proceso de realización de las operaciones de multiplicación. Las señales a generar son CEEBLE (activación de la acumulación del acarreo) y WSUMH (activación de la carga). El último paso es la realización del desplazamiento de los registros. Este proceso se realiza mientras no se active la señal CY (salida de acarreo del contador módulo 4 del bloque de datos). La carta ASM que implementa este comportamiento se muestra en la figura 6.

A partir de esta carta ASM se obtiene el diseño del circuito en base a biestables y puertas lógicas que se muestra en la figura 7, donde se ha tenido en cuenta la condición de utilizar un biestable por cada estado.

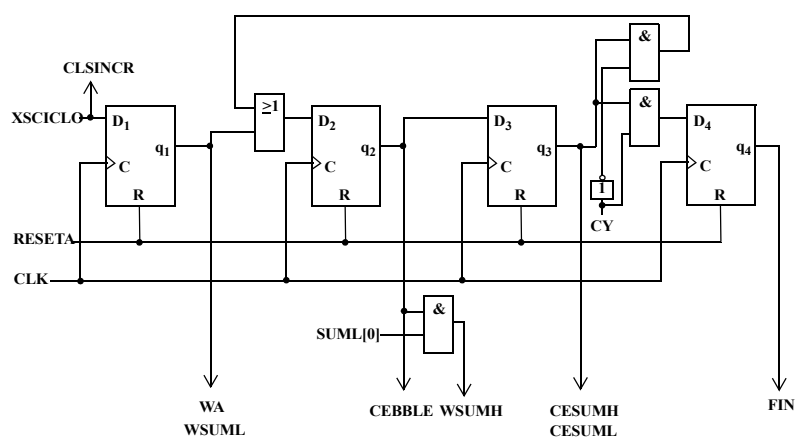


Figura 7. Esquemático de la unidad de control.

5. Tarea a realizar por el alumno: trabajo en el laboratorio

En el laboratorio, el alumno deberá introducir el diseño de la unidad de control realizada en el estudio teórico en la herramienta de Xilinx ISE WebPack [3] para su programación en una FPGA. Para simplificar el diseño, dispone de antemano del diseño del multiplicador 4x4 salvo del diseño de la unidad de control. El alumno deberá introducir el esquemático de la unidad de control y simular el circuito completo. En la figura 8 se muestra una captura del esquemático del multiplicador 4x4. En ella todos los bloques han sido descritos en VHDL. Además de los bloques propios del multiplicador se han introducido dos bloques, “controladorleds” y “controlador7seg” que sirven para mostrar las señales de salida del multiplicado en los leds y el display siete segmentos de la placa de desarrollo. De esta forma los datos a multiplicar son introducidos utilizando los conmutadores que tiene la placa. algunas de las señales internas son visualizadas utilizando leds y los datos de entrada así como la salida se visualizan utilizando los displays 7-segmentos.

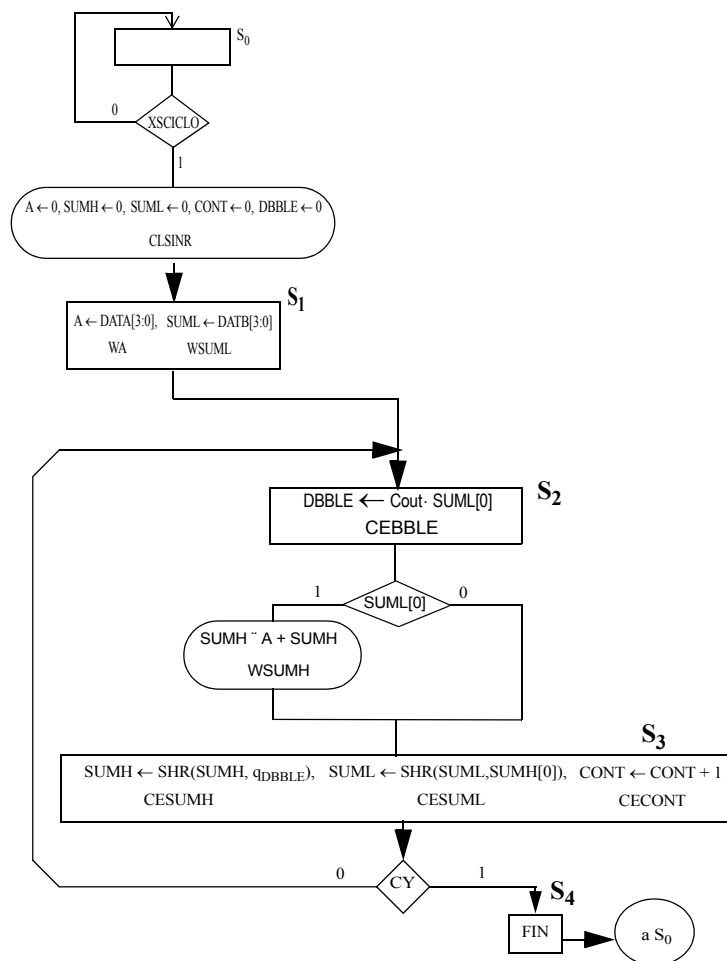


Figura 6. Carta ASM de la unidad de control.

6. Conclusiones

Esta experiencia se ha desarrollado con mucho éxito en el curso 2004/2005. Los alumnos terminan toda la experiencia en dos horas y salen muy motivados e ilusionados tanto por ‘haber manejado’ la herramienta de Xilinx, la FPGA y la placa de desarrollo, como por haber comprobado que el multiplicador funciona según los datos y señales de control estudiados teóricamente.

Referencias

- [1] Alberto Molina, Sergio Díaz, Jose I. Escudero, “Estructura y Tecnología de Computadores”, Ed. Panella, 2004.
- [2] Morris Mano, Charles Kime, “Fundamentos de Diseño Lógico y Computadoras”, Pearson Educación, 2005.
- [3] Xilinx ISE WebPack: www.xilinx.com

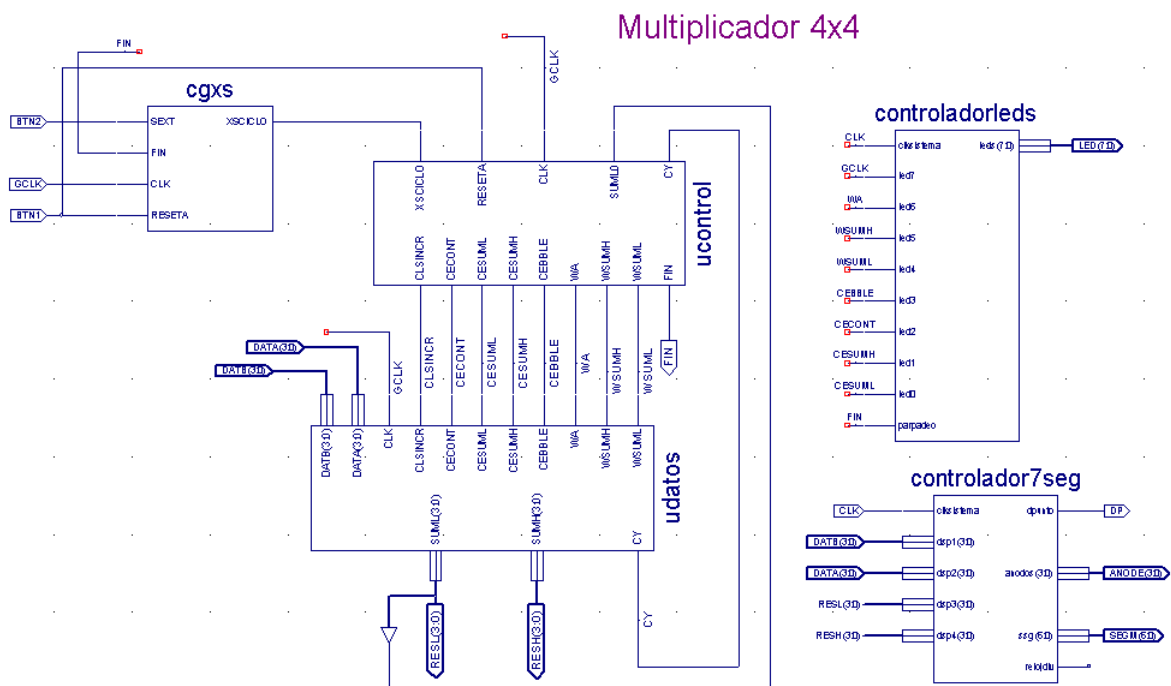


Figura 8. Esquema del multiplicador en la herramienta de Xilinx.