Algoritmo de Booth - Multiplicador con signo

Ronald Esteban Duarte Barrantes, ronald@estudiantec.cr
Josué Alejandro Hernández Medina, 43214489@estudiantec.cr
Leonardo Enrique Pérez Sandoval, leoperezsandoval@estudiantec.cr
Jesús David Vargas Arias, jdvargas.arias@estudiantec.cr
Estudiantes, ITCR.

***Resumen*—Inicialmente, fue indispensable comprender el funcionamiento del algoritmo de Booth para la correcta realización de la máquina de estados. Posteriormente, se definieron las entradas y salidas para hacer el diagrama de estados y así facilitar el diseño de dicha máquina en *SystemVerilog*. Asimismo, se diseñaron otros circuitos necesarios para el funcionamiento de todo el sistema. Dichos circuitos corresponde a un circuito sincronizador y a un contador hacia abajo. Finalmente, se hicieron las pruebas necesarias para comprobar el correcto funcionamiento del sistema.**

***Index Terms*—Algoritmo de Booth, contador, flip-flops, máquina de estados, multiplicador, sincronizador, *SystemVerilog*.**

I. Introducción

Las máquinas de estado son circuitos secuenciales síncronos que consisten en tres bloques: dos de lógica combinacional (lógica de siguiente estado y lógica de salida) y un registro que almacena el estado. El objetivo de estas máquinas es controlar otros circuitos mediante sus entradas, las cuales permiten tomar decisiones y, por medio de las salidas, le indican a las otras partes del sistema cómo actuar. Por otro lado, puede ser de interés realizar un multiplicador de manera secuencial, el cual se logra mediante el algoritmo de Booth. Asimismo, el circuito que implementa dicho algoritmo puede ser controlado por una máquina de estados, siendo este el objetivo del presente trabajo.

II. Resultados experimentales

II-A. FSM Mealy

Para el diseño de la máquina de estados, se utilizó una de tipo Mealy. Para el funcionamiento de la máquina de estados, se diseñó un diagrama de flujo encargado de resumir dicho funcionamiento. El diagrama de flujo se puede observar en la Figura 1.

Luego, del propio diagrama de flujo se diseñó el diagrama de estados de la FSM. Esta se muestra en la Fig 2.

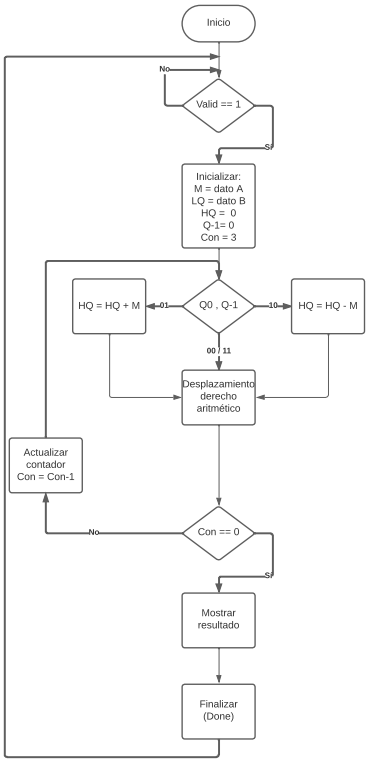

Fig. 1: Diagrama de flujo de la FSM.

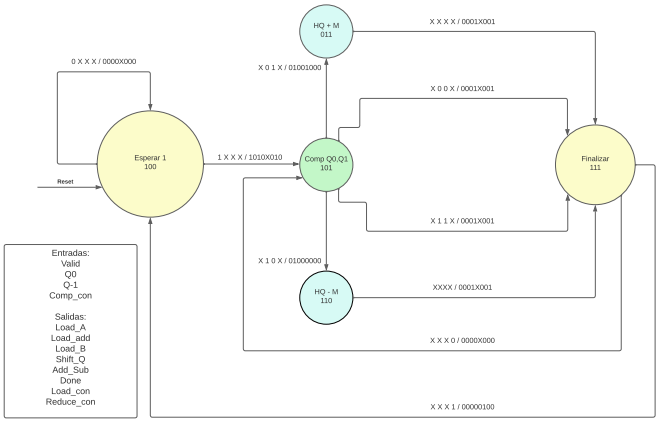

Fig. 2: Diagrama de estados de la FSM.

Por último, se obtuvo la tabla de estados la cual corresponde

a la Tabla I en conjunto con la lógica de salida de la Tabla II. A partir de estas, se realiza la descripción de hardware y se obtuvo el esquemático de la FSM, la cual se puede observar en la Figura 3.

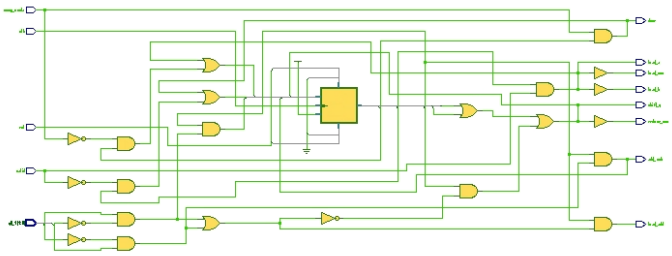

Fig. 3: Esquemático de la FSM simulado.

II-B. Datapath (Multiplicador con signo con algoritmo de Booth)

Para el caso del multiplicador tenemos el siguiente resultado con respecto a la simulación del datapath en el precision.

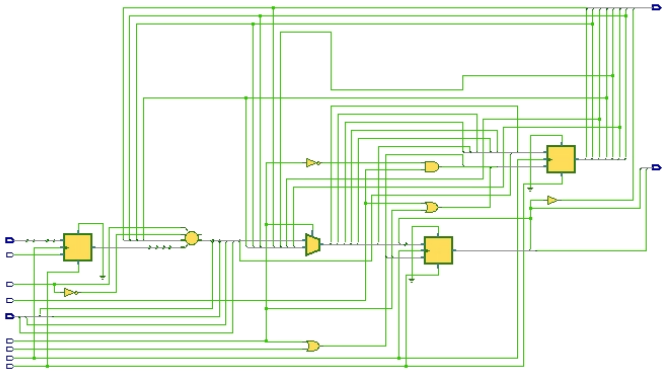

Fig. 4: Esquemático del algoritmo de Booth simulado.

II-C. Contador descendente

El presente subsistema recibe como entrada un habilitador para signar el número de iteraciones a realizar, producto de una salida de la FSM. Asimismo, otra entrada, igualmente de carácter habilitador, para hacer la reducción de las iteraciones. En la salida se encuentra un pulso que indica cuando ya se haya realizado la última iteración de la FSM. A continuación se muestra el circuito obtenido por medio de la herramienta *Precision*.

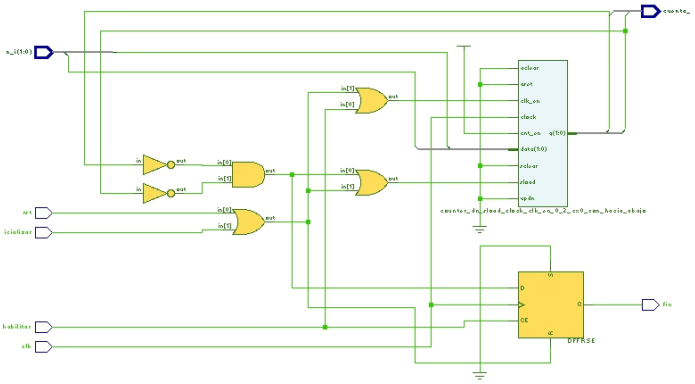

Fig. 5: Contador descendente.

II-D. Subsistema de lectura

A este subsistema le entran las variables *valid* y los operandos *A* y *B*. A continuación se muestran los resultados obtenidos, donde $valid_1$ es la entrada del subsistema y $valid_2$ la salida, es decir, la entrada de la máquina de estados. En este circuito se observan los tres flip-flops que retrasan la señal.

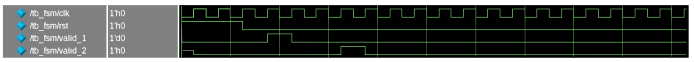

Fig. 6: Señales del subsistema del lectura

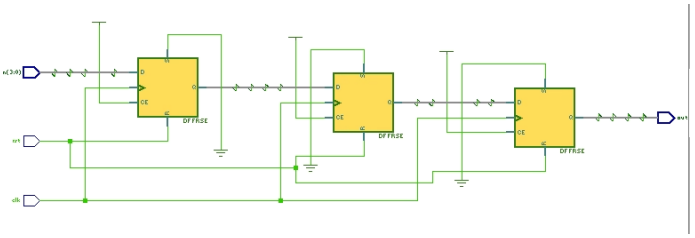

Fig. 7: Subsistema de lectura.

II-E. Curvas y datos de resultado del multiplicador

A continuación se presentan los resultados obtenidos del multiplicador.

```
a:              0100
b:              0001
Respuesta:      00000100
a:              1001
b:              0011
Respuesta:      11101011
a:              1101
b:              1101
Respuesta:      00001001
```

Fig. 8: Resultados de la terminal del multiplicador con todos los bloques en conjunto

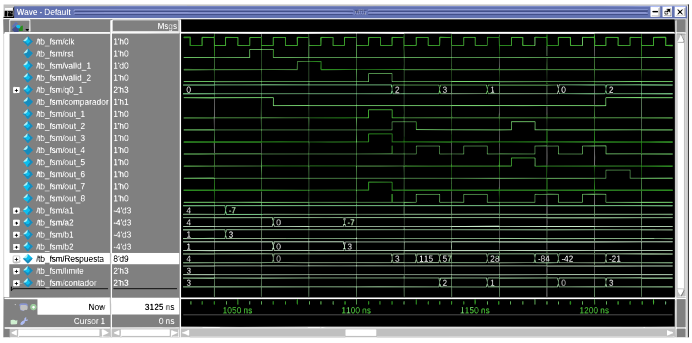

Fig. 9: Curvas del multiplicador con todos los bloques en conjunto

III. Análisis de Resultados

III-A. FSM Mealy

Tenemos la siguiente FSM proporcionado por el enunciado, el cual corresponde a la Figura 10. A pesar de que se ve simple, se tuvo que agregar una entrada más que es la encargada de llevar el funcionamiento del contador llamada comparador.

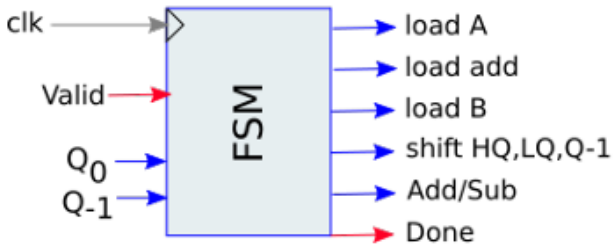

Fig. 10: Máquina de estados.

Lo primero que se debe de comprender es que la razón por la que se escogió usar una FSM de tipo Mealy es ya que permite reducir la cantidad de estados a utilizar.

La Figura 1 muestra como debe actuar el algoritmo del multiplicador. De esta forma nos damos una idea de como debe actuar la FSM con respecto al datapath y al contador.

La Figura 2 muestra el diagrama de estados de la FSM. Debido al diagrama de flujo obtenido, se establece que se usarían 4 entradas a la máquina de estados, una para validar la acción de operación, otras dos que controlan las acciones de suma y resta del datapath con el registro HQ de la Figura 11, las cuales son entradas que las controlan el propio datapath por medio de Q0 y Q-1. Por último está una entrada de comparador, la cual compara el valor de la salida del contador con respecto a 0 para controlar la cantidad de desplazamientos de todo el registro Q del datapath.

Con el diagrama de estados, se llegó a la deducción que solo es necesario utilizar 5 estados (3 estados ilegales). Los cuales son los siguientes:

- Esperar 1 (100): Se encarga de recibir un puslso de inicio del sistema. Hasta que no reciba un impulso para validar la operación del datapath, seguirá atascado en este estado.
- Comp Q0,Q1 (101): Establece la operación a llevar acabo, si es un desplazamiento (00/11), si es una suma (01) o si es una resta (10), lo cual se realiza con las entradas Q0 y Q-1.
- HQ + M (011): Es el estado en el cual se realizó la suma de M y HQ de la Figura 11 si las previas entradas Q0 y Q-1 tenían de valor 0,1. Luego realiza el desplazamiento al llegar al estado Finalizar.
- HQ - M (110): Es el estado en el cual se realizó la resta de M y HQ de la Figura 11 si las previas entradas Q0 y Q-1 tenían de valor 1,0. Luego realiza el desplazamiento al llegar al estado Finalizar.
- Finalizar (111): Corresponde al estado final en el cual se dan las iteraciones con el contador si el procedimiento del estado Comp Q0,Q1 si es que la entrada de la FSM comparador es 0, o bien se termina el procedimiento cuando comparador es 1 y vuelve al estado Esperar1.

De esta forma la FSM con sus entradas y estados controlan las salidas que están conectadas al datapath y al contador, como se muestra en el esquemático de la Figura . La Tabla I muestra el comportamiento de los estados siguientes y la Tabla II el control de las salidas para el correcto funcionamiento.

III-B. Datapath (Multiplicador con signo con algoritmo de Booth)

Primeramente analicemos el funcionamiento del datapath, tenemos el siguiente esquemático original:

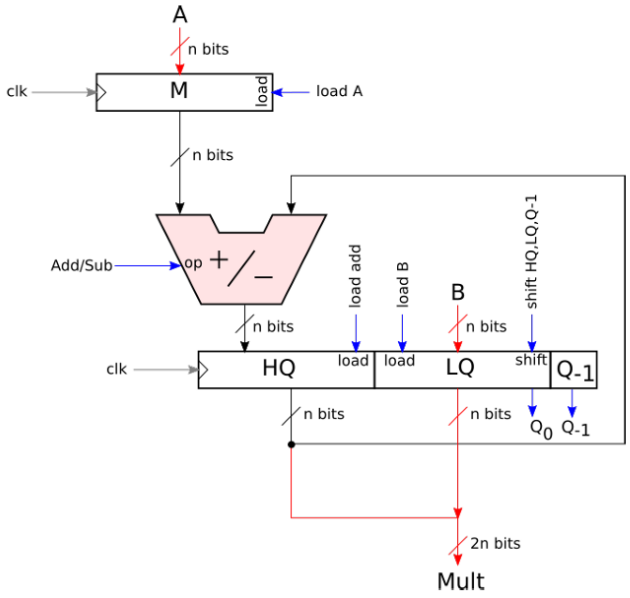

Fig. 11: Algoritmo de Booth.

Para comprender el datapath visto anteriormente, debemos centrarnos en cómo debe funcionar el algoritmo de Booth para la multiplicación binaria, para nuestro caso, trabajaremos con un bus de 4 bits de manera estándar, en otras palabras, el funcionamiento de los esquemáticos está hechos para trabajar con 4 bits.

En una primera instancia es necesario conocer los números binarios a multiplicar, usemos la misma notación para multiplicando y multiplicador que son A y B respectivamente. Para este tipo de algoritmo de multiplicación se trabaja con matrices que contempla tres filas y A+B+1 columnas, estes caso solo nos importa el tamaño del bus, en nuestro caso sería de 9 columnas, en donde los primeros bits de cada fila estarán distribuidos de esta manera y de este mismo orden:

D: bits de A
S: complemento a dos de A
P: ceros
En cuanto a los bits restantes, en cada fila se rellenan con
D: cero
S: cero
P: bits de B

Después de haber construido la matriz los valores de la columna 9 se inicializan en 0, esto es importante, debió que los dos bits menos significativos de la columna P muestran el procedimiento algebraico a seguir, tiendo las siguientes opciones

Cuando P [8:0], entonces
P [1:0] = 2'b00; No se realiza ningún cambio
P [1:0] = 2'b01: se suman P=P+D, se ignora el desbordamiento
P [1:0] = 2'b10; P=P+S ó P=P-A, se ignora el desbordamiento
P [1:0] = 2'b11; No se realiza ningún cambio

Este procedimiento se debe realizar de manera iterativa, creando nuevas filas con los nuevos valores de P, que pueden ser diferentes, es importante recalcar que para nuevos valores de P se debe hacer un desplazamiento aritmético a la derecha, conservando el signo, este proceso se detiene cuando se alcance el número de bits en un bus, en este caso se detiene al realizar 4 iteraciones.

Esta generalización del funcionamiento del algoritmo de Booth, la podemos ver en la figura 11, en donde cada bloque se encarga de una determinada parte del procedimiento, para al final tener el comportamiento completo del algoritmo de Booth.

Registro M: Este bloque permite almacenar el bus de bits anterior o carga el nuevo bus de bits, esto mediante la utilización de las señales CLK, Rst y load A, utilizando el flanco positivo del CLK y los flancos positivos de RST, sise cumple el RST entonces M es cero, en caso contrarío carga M = load A.

Adder/sub: En este bloque el bus de bits de las entradas A y B, se podrán sumar o restar dependiendo del control de la maquina de estados, que esta depende de los bits menos significativos del P como se menciono anteriormete, pero lo vemos en un bloque que contempla todos los parametros nesesarios para hacer dicho corrimiento.

Registro de corrimiento; Hace los corrimientos nesesarios, con las respectivas iteraciones, que seran de 4 iteraciones maximas.

III-C. Contador descendente

En el caso de este contador, los resultados fueron favorables, ya que, se obtuvo lo esperado, es decir, que este empiece en cuatro y termine en cero, volviendo al valor inicial y colocando la variable *fin* en uno; variable que le entra a la máquina de estados indicándole que ya se hizo el proceso. En otras palabras, esta variable *fin* tiene la función de llevar el control de cuántas veces se ha hecho el proceso.

III-D. Subsistema de lectura

En este subsistema, como se pudo esperar, la entrada se vio en la salida tres ciclos de reloj después de que esta ingresó. Esto sucede debido a que este subsistema consiste de tres flip-flops en cascada. Un detalle a notar en este subsistema, además del anteriormente mencionado, es que tiene un impacto importante a la hora de considerar metaestabilidad, ya que, si el primer flip-flop entra en este estado, no debería de haber mayor inconveniente debido a que quedan los otros dos.

III-E. Principales problemas hallados durante el trabajo

En este trabajo se produjeron diversos inconvenientes al momento de realizarlo, concretamente centrándonos en la parte del FSM y Testbench.

En cuanto a la FSM ocurrieron diversos problemas al momento de hacerla funcionar tal y como se muestra en el enunciado del trabajo, la solución utiliza en este caso fue volver a analizar todo el diagrama de estados y el diagrama de flujos, en donde comprendimos que a la maquina de estado le hacia falta una entrada más de la que habíamos propuesto, por lo cual, esta era la entrado comparadora, en donde podíamos notar las iteraciones necesarias que ocupaba la maquina para finalizar su función.

En cuanto al Testbench, este proporciono muchos errores al momento de hacer las conexiones respectivas con cada bloque, en donde se hicieron diversas modificaciones en los diferentes bloques para hacer que estos funcionen en conjunto, como lo puede ser cambiar los flancos de reloj y reset de los bloques para que todos trabajaran en un mismo flanco, analizar el comportamiento del contador y verificador porque no estaba funcionando, en este caso fue debido a que no se estaba guardando la salida del mismo, analizar la FSM y verificar que la lógica de siguiente estado sea la misma que la de los diagramas, lo cual no se estaba cumpliendo, por lo cual se modifico, con estos cambios hicimos funcionar todas las partes correspondientes a la maquina de estados.

IV. Conclusiones

- El subsistema de lectura juega un papel importante al momento de considerar metaestabilidad, aunque a nivel de simulación su impacto no sea tan significativo.
- El contador hacia abajo proporciona la iteraciones necesarias para completar el algoritmo de Booth .
- La FSM proporciona el control y alimenta la entradas del algoritmo de Booth, para ejecutar correctamente la operaciones y desplazamiento necesario.
- Para realizar conexiones se requiere observar que entradas se requieren y salidas produce cada bloque y definir que realiza una salida de un bloque en específico en otro o en todos.
- Para escoger un tipo de codificación se ha de plantear las características fundamentales de cada codificación, hasta escoger la más beneficiosa para mantener un equilibrio entre área, potencia y velocidad.
- Para establecer las señales de entrada y salida se necesita de un análisis detallado de la FSM y el diagrama de fluido, puesto que cada condicional representa una pista para cada entrada.

V. Anexos

Tabla I: Tabla de Estados de la FSM.

S(2)	S(1)	S(0)	In(3)	In(2)	In(1)	In(0)	S(2)+	S(1)+	S(0)+
1	0	0	0	x	x	x	1	0	0
1	0	0	1	x	x	x	1	0	1
1	0	1	x	0	1	x	0	1	1
1	0	1	x	1	0	x	1	1	0
1	0	1	x	0	0	x	1	1	1
1	0	1	x	1	1	x	1	1	1
0	1	1	x	x	x	x	1	1	1
1	1	0	x	x	x	x	1	1	1
1	1	1	x	x	x	0	1	0	1
1	1	1	x	x	x	1	1	0	0

Tabla II: Salidas de la FSM relacionadas a la Tabla I.

Out(0)	Out(1)	Out(2)	Out(3)	Out(4)	Out(5)	Out(6)	Out(7)
0	0	0	0	x	0	0	0
1	0	1	0	x	0	1	0
0	1	0	0	1	0	0	0
0	1	0	0	0	0	0	0
0	0	0	1	x	0	0	1
0	0	0	1	x	0	0	1
0	0	0	1	x	0	0	1
0	0	0	1	x	0	0	1
0	0	0	1	x	0	0	1
0	0	0	0	x	1	0	0

Referencias

[1] Floyd, T. L. (2018).*Dispositivos Electrónicos (Octava Edición)*.
[2] David Harris y Sarah Harris. *Digital Design and Computer Architecture.* RISC-V Edition. Morgan Kaufmann, 2022. ISBN: 978-0-12-820064-3.

VI. Link del vídeo

Copiar y pegar el link en el buscador
https://estudianteccr-my.sharepoint.com/:f:/g/personal/43214489_estudiantec_cr/EpWR1pqDWdwV4Pi7v5tg?e=eulJaw