16 Bits Full Adder

Contreras R Orlando[1,2*] and Garcia B Iker[1,2*†]

[3]Departamento de Sistemas Electrónicos, UAA, Av. Universidad 940, Aguascalientes, 20131, Aguascalientes, México.

*Corresponding author(s). E-mail(s): {al348390,al307630}@edu.uaa.mx;
†Ambos autores contribuyeron de forma equitativa a este trabajo.

Abstract

La siguiente práctica busca replicar el comportamiento de un sumador, en este caso sería un sumador de forma binaria con 16 bits, por lo que la entradas serían de 16 bits, mientras que la salida sería de 17 considerando que la suma máxima puede desbordar el valor, a este desbordamiento se le llama Carry.

Keywords: VHDL, Adder, bits, Carry, FPGA

1 Introducción

En el diseño de circuitos digitales, los sumadores juegan un papel fundamental en diversas aplicaciones, incluyendo procesadores, unidades aritméticas y lógicas (ALU) y sistemas embebidos. Un sumador de 16 bits es un componente esencial en arquitecturas de procesamiento que requieren operaciones aritméticas de alta velocidad y eficiencia.

El presente trabajo aborda el diseño y análisis de un sumador de 16 bits. El documento está estructurado de la siguiente manera: en la Sección 2 se presentan los fundamentos teóricos sobre los sumadores, diseño y metodología utilizada; en la Sección 3 se muestran los resultados obtenidos y el circuito armado; finalmente, en la Sección 4 se exponen las conclusiones.

2 Metodología

2.0.1 Diseño de medio-sumador de 1 Bit

Para poder implementar de forma efectiva un sumador de 16 bits, deberemos realizar el primer componente siendo un sumador de 1 bit, en primer instancia se debe pensar el sumador de forma primitiva como una operación de forma manual. Como podemos apreciar en la Figura 1, al realizar la suma se obtiene un acarreo.

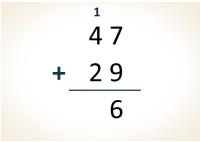

Fig. 1 Suma decimal.

A	B	C_{out}	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Table 1 Tabla de verdad de la suma de dos bits.

Si lo vemos de forma binaria podría verse como tabla de verdad (Figura 2), de aquí podemos deducir las siguientes ecuaciones

$$C_{out} = AB \tag{1}$$

$$S = A \oplus B \tag{2}$$

Basándonos en estas ecuaciones, se puede notar que la salida de suma S se obtiene mediante una operación XOR entre las entradas A y B, mientras que el acarreo de salida (C_{out}) se obtiene mediante una operación AND entre ambas entradas. A este diseño se le conoce como medio-sumador o half adder, ya que no considera un bit de acarreo/carry de entrada (C_{in}).

Éste diseño es fundamental para la construcción de sumadores más complejos.

Usando como referencia la tabla del sumador completo podemos generar una ecuación que cumpla con todos los requisitos, esta ecuación para S requiere que de 1 como salida cuando A y B y C son diferentes así que se puede intuir que es una XOR entre las tres entrada.

$$S = A \oplus B \oplus C_{in} \tag{3}$$

A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Table 2 Tabla de verdad del sumador completo.

Para la salida C_{out} la ecuación surge por que se requiere que de 1 cuando A es diferente de B por eso la XOR pero solamente cuando Cin vale 1 y para que se cumpla que de 1 cuando A y B valgan uno se usa la AND y en si cualquiera de estos casos se cumplen para que genere una salida se usa la OR aparte ver como surge la ecuación siguiente:

$$C_{out} = (AB) + (C_{in}(A \oplus B)) \tag{4}$$

3 Resultados

Se puede observar que la simulación cumplió con lo requerido, siendo el $Cout_{tb}$ la señal del Acarreo de salida. Finalmente quedaría implementarlo de forma correcta en la FPGA, para esto diseñamos la Constraints FIle quedando de la siguiente forma en el circuito implementado (Fig. 3)

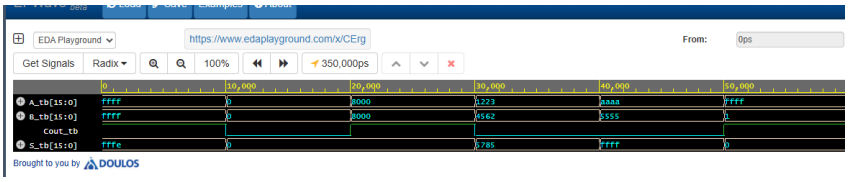

Fig. 2 Testbench realizado por EDA Playground

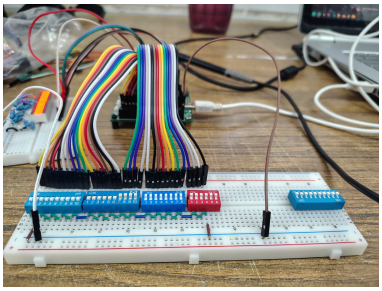

Fig. 3 Circuito Armado

4 Conclusiones

La práctica nos resulto interesante, se complico algo el hecho de que fuera de 16 bits ya que no fue posible realizarlo con la FPGA que teníamos dado a la limitante de pines y por practicad se decidió hacerlo de 16 bits ya que la practica siguiente también consiste en 16 bits y realmente no cambia funcionamiento del circuito, realmente la dificultad que nos presento fue a la hora de realizar las conexiones debido a que la descripción de hardware/ código se nos facilita.