Grado en Ingeniería de Computadores

Tecnología de Computadores

Práctica 5

Universidad Rey Juan Carlos

1. Objetivo

En esta quinta sesión de prácticas se pretende que el alumno se familiarice con los circuitos combinacionales aritméticos y de almacenamiento de información. Para ello, se implementarán algunos ejemplos como un sumador completo o una memoria ROM.

Una vez terminada la práctica el alumno será capaz de:

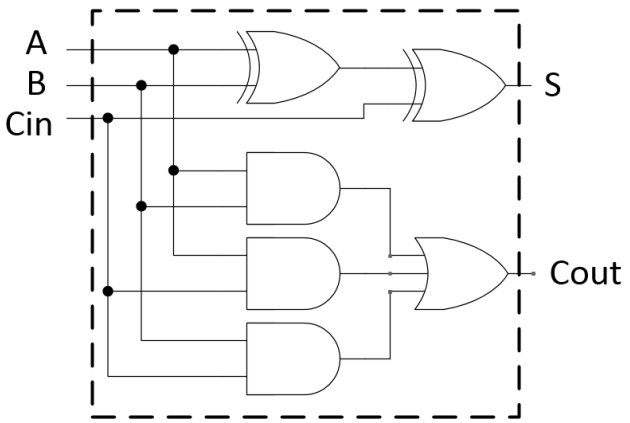
- Diseñar circuitos combinacionales aritméticos y de almacenamiento de información en VHDL
- Diseñar circuitos modulares en VHDL
- Implementar diseños con sumadores y memorias ROM en una FPGA

2. Desarrollo de la práctica

Circuitos aritméticos. El sumador.

Como se ha visto en las clases de teoría, un sumador completo es aquel que tiene, además de las señales de entrada de los operandos, señal de acarreo de entrada y de acarreo de salida. A modo de recordatorio, se presenta a continuación la tabla de verdad de un sumador completo de dos bits:

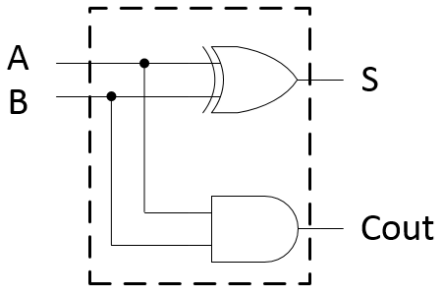
Cin	A	B	Cout	S
0	0	0	**0**	**0**
0	0	1	**0**	**1**
0	1	0	**0**	**1**
0	1	1	**1**	**0**
1	0	0	**0**	**1**
1	0	1	**1**	**0**
1	1	0	**1**	**0**
1	1	1	**1**	**1**



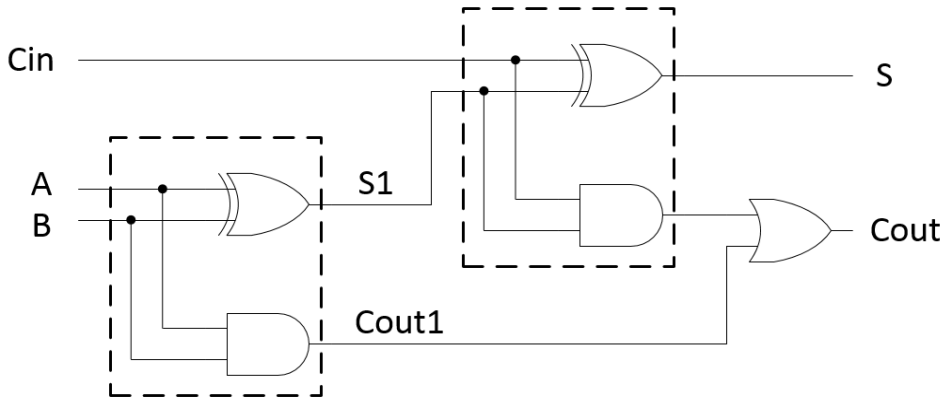
Como primer ejercicio de esta práctica, implemente en VHDL el sumador completo mostrado en la figura anterior. Obtenga el número de recursos de la FPGA utilizados por este diseño así como el tiempo de propagación de su camino crítico. Utilice el fichero de constraints "*sumador_constraints.xdc*" proporcionado.

El segundo ejercicio de la práctica consiste en implementar en VHDL el mismo sumador completo pero utilizando dos semisumadores. Para ello, implemente primero el código VHDL del semisumador de dos bits, cuya tabla de verdad y circuito se presentan a continuación:

A	B	Cout	S
0	0	**0**	**0**
0	1	**0**	**1**
1	0	**0**	**1**
1	1	**1**	**0**



Implemente ahora el sumador completo. Recuerde que se puede construir a partir de dos semisumadores como sigue:

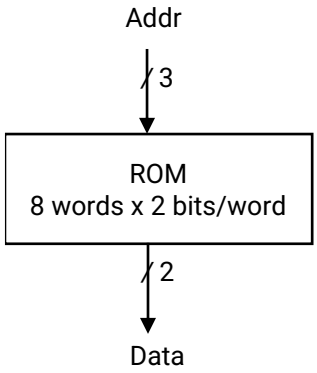


Obtenga el número de recursos de la FPGA utilizados por este segundo diseño así como el tiempo de propagación de su camino crítico. Utilice el mismo fichero de constraints del anterior ejercicio. Compare los resultados de ambas implementaciones y determine cuáles son las ventajas de cada una de ellas.

Circuitos de almacenamiento de información. La memoria ROM.

Las memorias son elementos básicos en los circuitos digitales para almacenar temporalmente información. En concreto, las memorias ROM permiten tener almacenada información de consulta que se puede leer pero no modificar. Las memorias ROM son muy útiles como tablas de consulta, de modo que tienen información precalculada de antemano. Así, no es necesario realizar la operación de cómputo cada vez, reduciendo tiempo de procesado y número de recursos utilizados por la FPGA.

Como ejemplo de esto, en este tercer ejercicio se propone la implementación de un sumador completo de dos bits como el visto anteriormente pero mediante una memoria ROM. La memoria ROM tendrá como entrada un bus de direcciones de 3 bits denominado Addr que actuará como las señales A, B y Cin. Este bus de direcciones nos permitirá acceder a la posición de memoria adecuada y leer lo que deberían valer las salidas S y Cout agrupadas en un bus de datos de 2 bits denominado Data. El diagrama del circuito a diseñar se puede ver en la siguiente figura:



Escriba el código VHDL de una memoria ROM que implemente el comportamiento del sumador completo de dos bits visto anteriormente. Añada el fichero de constraints "*ROM_constraints.xdc*". Obtenga el número de recursos de la FPGA utilizados por este diseño así como el tiempo de propagación de su camino crítico y compárelo con los dos diseños del sumador completo ya realizados.

3. Ejercicios propuestos

A continuación se proponen dos ejercicios de implementación en FPGA para validar en el hardware físico el funcionamiento de algunos de los circuitos vistos durante la práctica.

1. Genere el bitstream del primer diseño del sumador completo realizado anteriormente y cárguelo en la FPGA. Rellene la siguiente tabla con el valor de las salidas del circuito:

Interruptores de entrada			LEDs de salida	
J15	**L16**	**M13**	**H17**	**K15**
0	0	0		
0	0	1		
0	1	1		
1	1	1		

2. Modifique el decodificador de binario a 7 segmentos que diseñó en la práctica anterior para que reciba la salida del sumador completo y muestre el resultado de la suma en un display de 7 segmentos. Añada a su proyecto el fichero de constraints "*sumador_dec7seg_constraints.xdc*" proporcionado por el profesor, genere el bitstream y cargue el diseño en la FPGA. Complete la siguiente tabla indicando los segmentos que se encienden o se apagan en cada combinación de entrada.

Interruptores de entrada			Segmentos de salida						
J15	**L16**	**M13**	**A**	**B**	**C**	**D**	**E**	**F**	**G**
0	0	0							
0	0	1							
0	1	1							
1	1	1