

1 Ejercicio 6: Implementación de una ALU

Para este ejercicio se pidió implementar una ALU de 4 bits con las operaciones SUMA, RESTA, AND, OR, NOT, XOR, Complemento a dos y Shift Left. Para esto se decidió crear un modulo encargado de sumar 2 bits, este a su vez devuelve un bit de respuesta y un bit de carry. Utilizando ese modulo, se procedió a hacer un nuevo modulo que haga una suma de 3 bits. Esta decisión simplificó mucho el desarrollo del modulo de suma de la ALU de 4 bits, ya que como se puede ver en el código “sum.v” encontrado en la carpeta src, se puede ver comentado dentro del código como fue nuestro desarrollo sin la función sum3Bits y como quedo finalmente el código con la implementacion de sum3Bits.

Para el caso de la resta se decidió utilizar el modulo hecho en el ejercicio 4 del complemento a 2 y así, se utilizó en conjunto ese modulo con el modulo suma.

En el caso de las operaciones AND, OR, NOT y XOR, se opto por utilizar las funciones predefinidas por verilog y reutilizarlas bit a bit.

1.1 Definiciones

Para la suma, se definio un bit extra llamado carry que justamente se enciende en caso de que la suma se pase del rango máximo representable por cuatro bits. Por otro lado, en el caso del Shift left, se definio que se ingresa un 0 al bit menos significativo del nibble en cuestión y el bit más significativo previo al shift, se devuelve como carry.