

Introducción al diseño lógico y lógica combinacional

1. Basado en lo visto sobre **decodificadores**: obtenga las ecuaciones booleanas simplificadas (por medio de algebra booleana y mapas K) que describen un decodificador de prioridad de 8 a 3.
2. Desarrolle un sumador de números en formato **"signo y magnitud"** de 4 bits. Haga uso de un circuito de suma de rizo (ripple-carry-adder) como base. ¿Qué componentes deben añadirse para poder realizar la suma en este formato?
3. Agregue la lógica necesaria para poder obtener un sumador/restador de datos **con signo y magnitud**.
4. Describa en que casos se generan elementos de memoria NO deseados en diseños de lógica combinacional en verilog.
5. ¿Cuál es la importancia de la "lista de sensibilidad" en verilog?
6. Implemente un mux 2 a 1 en verilog usando:
 - a. Una implementación "secuencial" (procedural).
 - b. Una implementación estructural usando compuertas "primitivas".
 - c. Una implementación con asignación continua usando ecuaciones booleanas.
 - d. Una implementación con asignación continua usando el operador condicional.
7. En que casos se usan los "wires" en verilog.
8. En que casos se usan los "reg" en verilog para lógica combinacional.
9. Desarrolle **una ALU** que haga uso de una señal de control de 3 bits. Los datos de entrada son dos operandos A y B, cada uno de 4 bits, la salida también es en 4 bits. De esta manera, las operaciones posibles de dicha ALU (basadas en la señal de control) serían:
 - 0: AND
 - 1: OR
 - 2: NAND
 - 3: NOR
 - 4: SUMA
 - 5: RESTAAdemás, la ALU posee una bandera de salida cero (cuando el resultado es cero). ¿Qué estrategias de simplificación puede aplicar?