

E1: UTEC-CS3501 : Computer Architecture

Instrucciones:

- Seguir las indicaciones del instructor en sala.
- **Mostrar su trabajo y justificación** en todas sus respuestas. Respuestas **no legibles** no tienen derecho a revisión.
- **Desarrollar de forma individual. No es permitido** usar material escrito o electrónico.

Question 1. Transistor level: Dibujar el circuito con transistores usando CMOS gates para un NOR y un AND de tres inputs A,B,C.

- (1pt) Obtener las tablas requeridas.
- (2pt) Dibujar los circuitos, acorde a las tablas definidas.

Question 2. Glitches: Se cree que el siguiente circuito tiene glitches. La literatura indica que una forma rápida de detectar los glitches producidos por transiciones de 1 a 0 es identificar 1s adyacentes que no estén agrupados en un mismo minterm.

- (1pt) Encontrar los glitches a partir del K-map.
- (2pt) Dibujar un diagrama de tiempo mostrando los glitches.

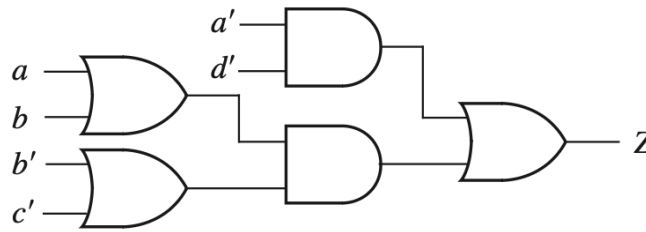


FIGURE 1. Circuito a analizar

Question 3. Timing analysis: La siguiente figura muestra un serial adder en el cual $a = 0110$ y $b = 0011$. Estas señales están incluidas en el timing diagram.

- (1pt) Asumiendo que los delays son despreciables. Completar el diagrama de tiempo.
- (2pt) Escribir la tabla de lógica de estados y output para una FSM Mealy. Dibujar la FSM Mealy.

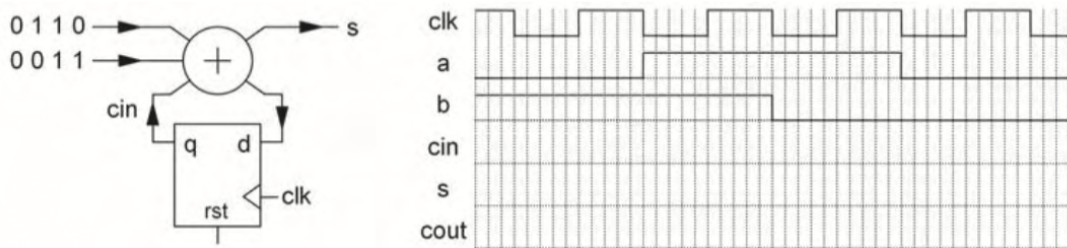


FIGURE 2. Circuito a analizar

Question 4. FSM: Diseñar un controlador para una puerta de garaje eléctrica que, además de las señales de reloj y reset, incluya cuatro entradas: remote (activada con '1' cuando se pulsa el control remoto), open ('1' cuando un sensor indica que la puerta está completamente abierta), closed ('1' cuando otro sensor confirma que está totalmente cerrada) y timer ('1' tras 30 segundos de permanecer abierta). Las salidas del sistema serán power ('1' para encender el motor) y direction ('0' para abrir la puerta y '1' para cerrarla). El controlador debe garantizar las siguientes funcionalidades:

- 1) Al pulsar el control remoto con la puerta cerrada, el motor se activa inmediatamente para abrirla.
 - 2) Al pulsar el control remoto con la puerta abierta, el motor se activa inmediatamente para cerrarla.
 - 3) Si el control remoto se pulsa durante la apertura o cierre, la puerta se detiene; al pulsarlo nuevamente, invierte su dirección.
 - 4) La puerta no puede permanecer abierta más de 30 segundos, controlado por la señal timer que fuerza el cierre automático.
- (1pt) Dibujar el diagrama de transiciones de la FSM.
 - (2pt) Escribir las tablas y lógica de estado siguiente y lógica de salida. Reducir las expresiones.
 - (1pt) Dibujar el esquemático.
 - (2pt) Escribir en Verilog la implementación de la FSM.

Question 5. ARM ISA: Implementa en ARM Assembly un algoritmo que ordene un arreglo de números enteros almacenados en memoria, comparando pares de elementos adyacentes y realizando intercambios si están en el orden incorrecto, repitiendo este proceso hasta que todo el arreglo quede en orden descendente. El resultado final debe sobrescribir los valores originales del arreglo. Asegúrate de optimizar el número de iteraciones evitando comparaciones innecesarias una vez que el arreglo esté ordenado.

- (1pt) Escribir el programa en C++ en alto nivel usando loops o loops anidados. Escribir comentarios de código. Incluir una función para crear e inicializar el siguiente array mostrado en estilo Python `A=[[1,2,3], [4,5,6], [7,8,9]]` usando un loop para este fin. Además, incluir la función para ordenar el arreglo.
- (3pt) Escribir el programa en ARM ASM. Asumir que el arreglo debe estar en el address 0x10000 (A), y el programa en 0x180000. Usar la menor cantidad de instrucciones posible y usar ejecución condicionada.
- (1pt) Dibujar el contenido de memoria y registros al finalizar el programa.
- (1pt) Calcular el porcentaje de tipos de instrucciones ARM ASM utilizadas en su programa y el tamaño en Bytes de su programa.