

Universidad del Istmo de Guatemala

Facultad de Ingeniería

Segundo Parcial

Arquitectura de Computadoras y Microcontroladores 1



Maximiliano González

15 de septiembre de 2025

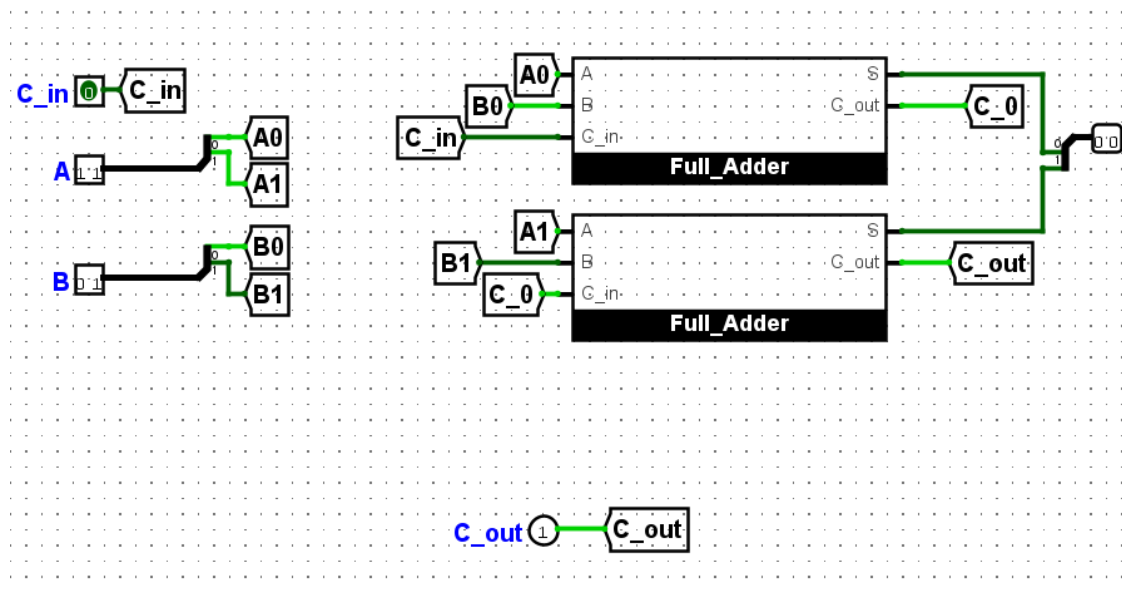
Serie 0 - (5 pts) - Instrucciones: FSM en HDL

Implemente la maquina/proceso del parcial 1 en Verilog-SystemVerilog como una FSM

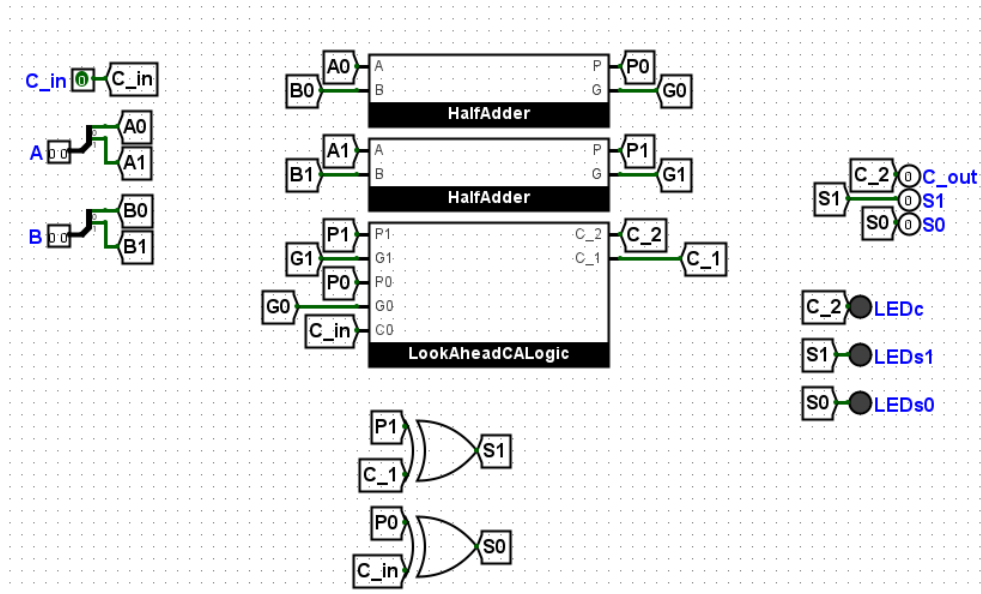
Serie 1 - (5 pts) - Instrucciones: Estudio de sumadores, simulación en Logisim

Utilizando Logisim evolution, realice la simulación de los siguientes 3 sumadores junto a su análisis de tiempo

- Ripple-Carry Adder**



- Carry-Lookahead Adder**



- Prefix Adder

Entregable: Video explicando la operación y funcionamiento de cada arquitectura incluyendo las ventajas/desventajas de cada uno. Finalmente, conteste que sumador usaría con énfasis en cantidad de compuertas y velocidad para aplicaciones:

- Lentas con restricción de espacio y presupuesto
- Rápidas sin restricción de espacio y presupuesto
- Rápidas con restricción de espacio y presupuesto

Serie 2 – (5 pts) - Instrucciones: Simulación e Implementación de ALU.

Implemente una ALU (Arithmetic Logic Unit) personalizada (Numero de bits distinto a convencional) vista en el capitulo 5 de su literatura. La siguiente figura (Figure 5.17 N - bit ALU with output flags) implementa 4 operaciones en registros de N bits (Suma, resta AND y OR).

1. Utilizando su último número de carne (1 si termina en 0) multiplicado x2 menos su penúltimo número de carne para la cantidad de bits N de los buses de datos (Tamaño de palabra, usualmente 8, 16, 32, 64... etc bits). Como por ejemplo si mi carne termina en 1998, construiré una arquitectura con buses del tamaño $8*2-9=7$ bits. [Que no sea mayor a los 10 bits]

Carne: 15990

$$|(1 * 2) - 9| = 7 \text{ bits}$$

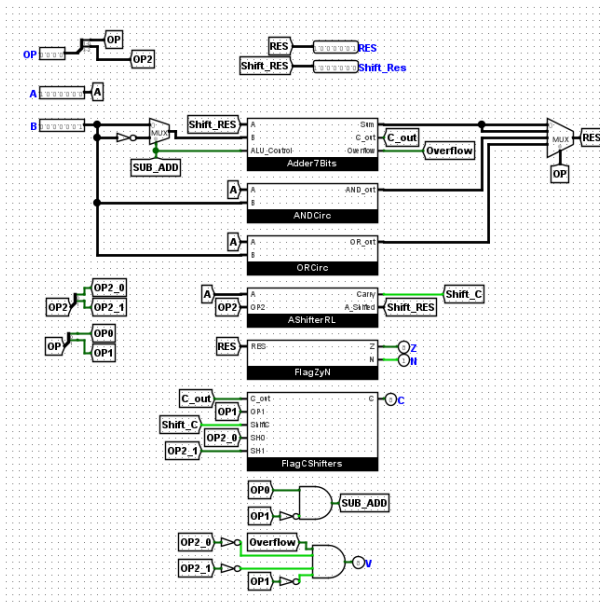
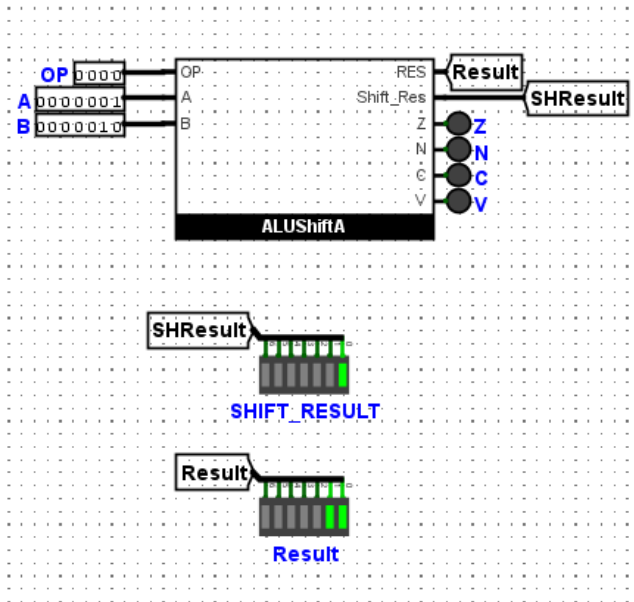
Cantidad de bits de los buses de datos: 7 bits

- a. Extienda la funcionalidad de 4 a 6 operaciones, las operaciones a agregarse son:
(Ambas explicadas en la Figure 5.18 4-bit shifters: (a) shift left, (b) logical shift right, (c) arithmetic shift right, notar uso de multiplexores)

- i. Shift right

- ii. Shift left

Considere su ALUControl debe ampliar el número de bits para poder soportar 6 operaciones y no 4. Utilice bloques jerárquicos para su simulación (De utilidad al implementarlo en HDL)

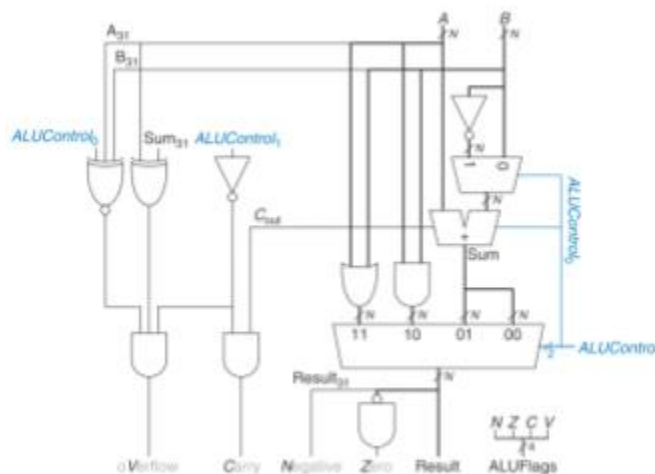


- Realice la implementación de esta ALU en Vivado utilizando SystemVerilog simulando su resultado mostrando su ALU personalizada se pudo implementar en hardware

Entregable: Video explicando la funcionalidad e implementación de la simulación en Logism Evolution. Video explicando la funcionalidad y simulación en Vivado.

Serie 3 – (5 pts) - Instrucciones: Implementación en Hardware real

1. Utilizando una FPGA, implemente el código HDL creado en el inciso anterior en hardware real utilizando entradas y salidas reales de una placa de evaluación.



2. Utilizando una FPGA, implemente el código HDL de su maquina/proceso del parcial 1

Serie 4 – (5 pts) - Instrucciones: Microcontroladores MMIO

Realice un reloj, utilizando displays de 7 segmentos, con opción a formato 12hrs y 24 hrs seleccionable con un botón. Programe una alarma, la cual realice una acción (buzzer?, LED parpadeante?) para indicar se llego a la hora de alarma.

Entregable de todo el parcial: Repositorio en github con estructura de repo formal. En el repo adjunte archivos fuente y documentación con discusión relevante. Incluya

videos cortos de explicación en cada inciso y sub inciso. Se tomará el ultimo commit antes de la fecha de entrega del parcial