



# ARQUITECTURA DE COMPUTADORAS

Etapa 2

Comisión 25

Integrante 1: Luciana Vecchiotti – LU 146460

Integrante 2: Joaquín Gamonal – LU 146610

Integrante 3: Alvaro Martin – LU 145714

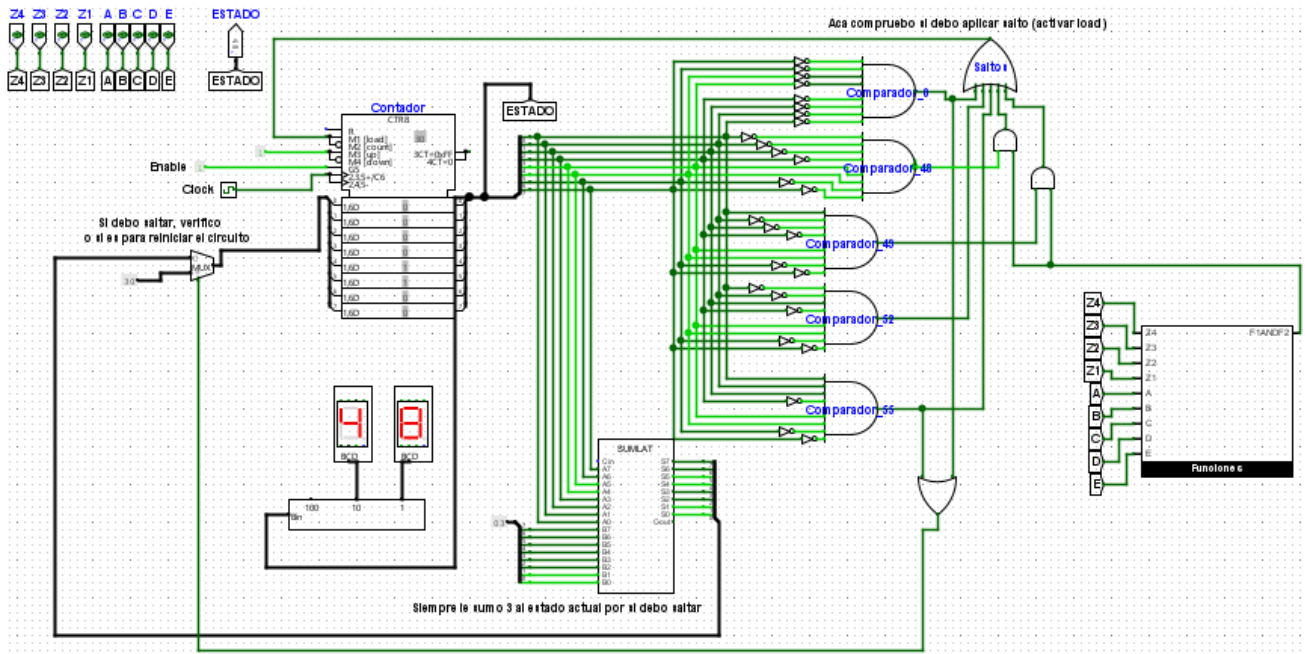
Integrante 4: Tomas Gadea – LU 146155

Fecha: 2025-05-20

Logisim-Evolution v3.9.0

## **DESCRIPCIÓN DEL CIRCUITO**

El circuito de la Etapa 2 es un **controlador de estados** de 8 bits que gestiona seis valores predefinidos (48, 49, 50, 51, 52 y 55) y permite dos modos de transición: **normal** y **alternativo**.



Su estructura principal consta de:

### **1. Contador de 8 bits**

1. Implementado con el componente Contador de Logisim-Evolution.
2. Almacena el **estado interno** actual, codificado en binario de 8 bits.
3. Salidas Q7:0 conectadas al sumador SUMLAT y a la lógica de control.

### **2. Sumador SUMLAT de 8 bits**

1. Reutiliza el look-ahead tree adder diseñado en la Etapa 1.
2. Su función principal en este circuito es **sumar 3** al valor actual del contador cuando se quiera efectuar un salto (de 52 a 55) o bien para avanzar de forma alternativa entre estados definidos.

### **3. Lógica de control de modo normal**

1. **Estados 48 → 49 → 50 → 51 → 52**

- Inicialmente, se habilita el pin **Enable** del contador, dejando pasar el pulso de reloj para que el contador incremente de uno en uno, y se habilita el pin **UpDown**, para que el contador incremente “hacia arriba”.

2. **Salto 52 → 55**

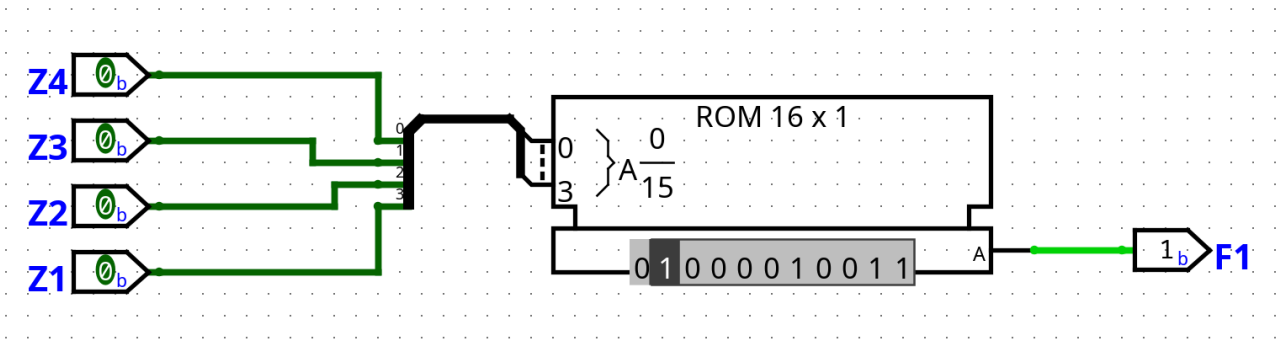
- SUMLAT recibe como operando A el valor 00110100 (52) (el estado interno del contador) y, como operando B, la constante “00000011” (3). El resultado que produce: 0011 0111 (55), actualiza el valor del pin **Data**.
- Paralelamente, cuando Q = 0011 0100 (52), se activa la señal de la compuerta lógica “**Salto**”, la cual habilita el pin de Load en el Contador, para que este cargue el valor del pin **Data**, en el próximo pulso de reloj.

3. **Reinicio 55 → 48**

- Si Q = 0011 0111 (55), en el siguiente ciclo se fuerza el pin **Load**, con el valor 0011 0000 (48) en el pin de **Data** del contador, devolviendo el sistema al estado inicial en el próximo pulso.

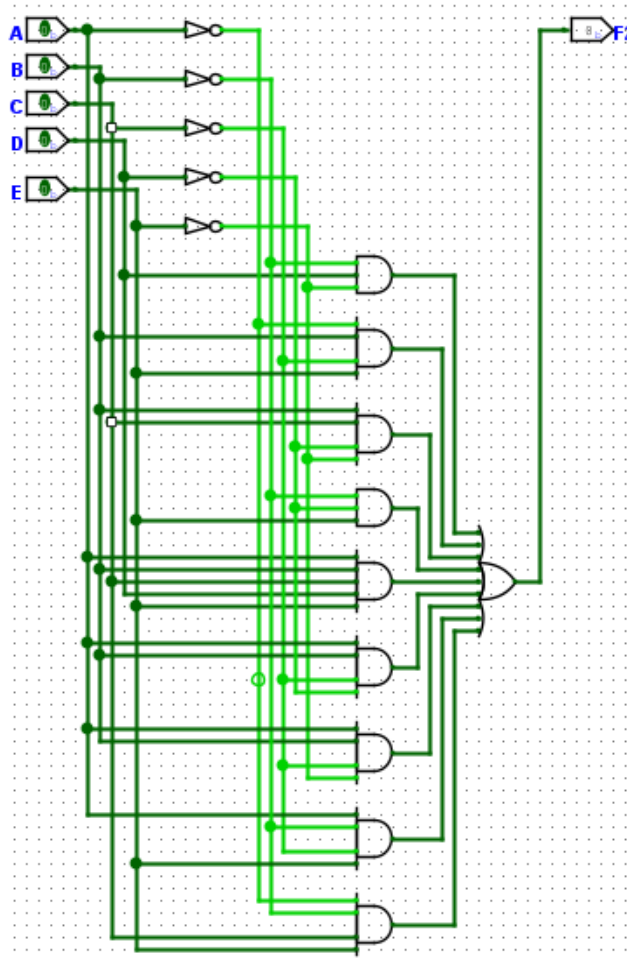
4. **Lógica de control modo alternativo**

1. Se activa mediante la salida de **F1 AND F2** (una salida de “F1 AND F2 = 1” simboliza que el modo alternativo, y su lógica correspondiente de saltos entre estados, está activa).
2. Consta de dos funciones externas **F1(Z<sub>1</sub>–Z<sub>4</sub>)** y **F2(A–E)**
  - **F1** está implementada con una ROM de 16×1, cuya tabla almacena la definición de la función sobre las cuatro señales Z<sub>1</sub>...Z<sub>4</sub>.



- F2 está realizada con puertas lógicas (AND, OR, NOT) según la tabla `f2.txt` adjunta con el enunciado.

- En el modo alternativo, se habilitan dos saltos adicionales: Estado 48 → Estado 51 y Estado 49 → Estado 52. Los valores correspondientes al resultado de los saltos, se cargan en el contador de la misma manera en la que se cargan con el modo normal, modificando la activación del pin Load según la lógica propuesta por dichos saltos.



## 5. Multiplexación de señales

La señal de selección del MUX se genera a partir de un comparador diseñado con puertas lógicas AND, el cual, al detectar el estado 55, obliga a que el contador cargue el primer estado (48) en el próximo pulso de reloj (pues simultáneamente se habilitará el pin de Load en el contador).

En el resto de los casos, el MUX estará constantemente actualizando el pin de DATA del contador, con todos los valores que el SUMLAT genera.

## 6. Flujo de señal resumido

1. El reloj está conectado siempre al contador.
2. Tabla de estados

Estado actual (Q)	Modo	Condición	Próximo estado (Q')	Acción
48	Normal	—	49	Q + 1
49	Normal	—	50	Q + 1
50	Normal	—	51	Q + 1
51	Normal	—	52	Q + 1
52	Normal	—	55	Q + 3
55	Normal	—	48	Reset
Cualquiera	Alternativo	$F1 \wedge F2 = 1$	Q + 3	Salto alternativo

## **CONCLUSIONES**

El diseño implementado cumple con todos los requisitos funcionales especificados en el enunciado del proyecto. Se lograron realizar correctamente las transiciones entre los estados definidos utilizando tanto el modo normal como el modo alternativo.

El uso del sumador SUMLAT, desarrollado en la etapa 1, demostró ser eficaz para los saltos condicionales, y la lógica de control integrada permitió automatizar los cambios de estado. Asimismo, la incorporación del circuito "modo alternativo", basado en funciones externas F1 y F2, añadió un mecanismo adicional de control que funcionó tal como se esperaba.

Este proyecto facilitó la aplicación práctica de conceptos fundamentales de arquitectura de computadoras, tales como contadores, sumadores, controladores de estado y diseño con compuertas lógicas. Se identificaron oportunidades de mejora durante la revisión de la etapa 1, algunas de las cuales fueron abordadas y otras serán consideradas en desarrollos futuros.

## **CORRECCIONES (De Etapa 1)**

Hemos recibido las siguientes sugerencias de parte de Esteban, nuestro ayudante designado en la primer etapa del proyecto, respecto a la construcción en Logisim del sumador de tipo Lookahead Tree Adder:

- SUMLAT no calcula Carry General
- El Carry-In (De SUMLAT) debería ser un pin
- Usar pines y separadores dentro del SUMLAT
- Es una mala inversión de compuertas calcular el carry-in en los circuitos FAGP si no van a ser utilizados

Se realizaron las correcciones correspondientes a las dos primeras observaciones, ajustando el diseño del SUMLAT en consecuencia. Las últimas dos sugerencias no fueron implementadas debido a limitaciones de tiempo en esta etapa del desarrollo. De todas formas, agradecemos las observaciones, ya que las tendremos en cuenta para futuras mejoras