



**UNIVERSIDAD  
DE ANTIOQUIA**

1 8 0 3

# **Laboratorio #1**

## **Arquitectura de Computadores**

---

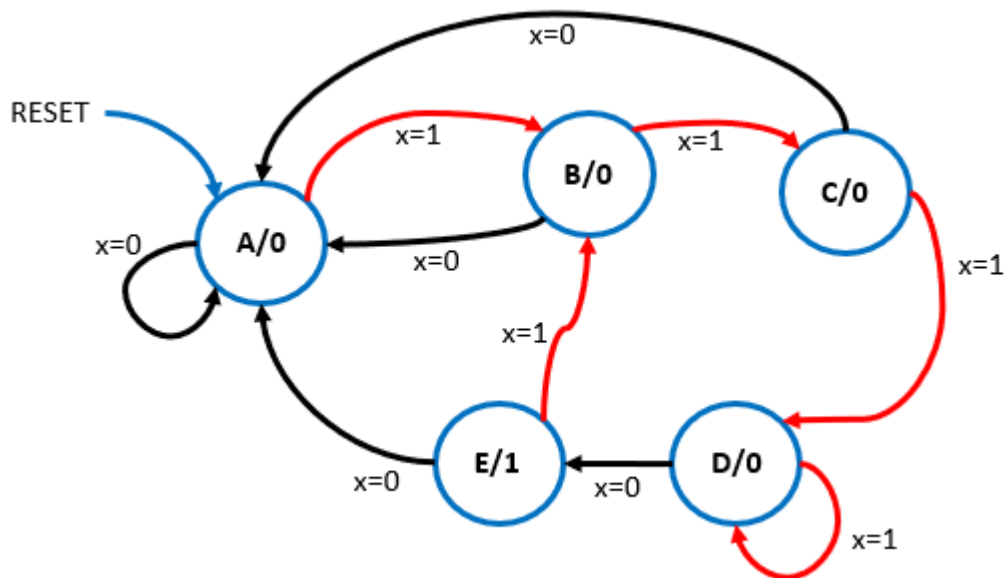
**Andrés Cardona Carmona - 1001651921**

**Sebastián Loaiza Muñoz - 1214748861**

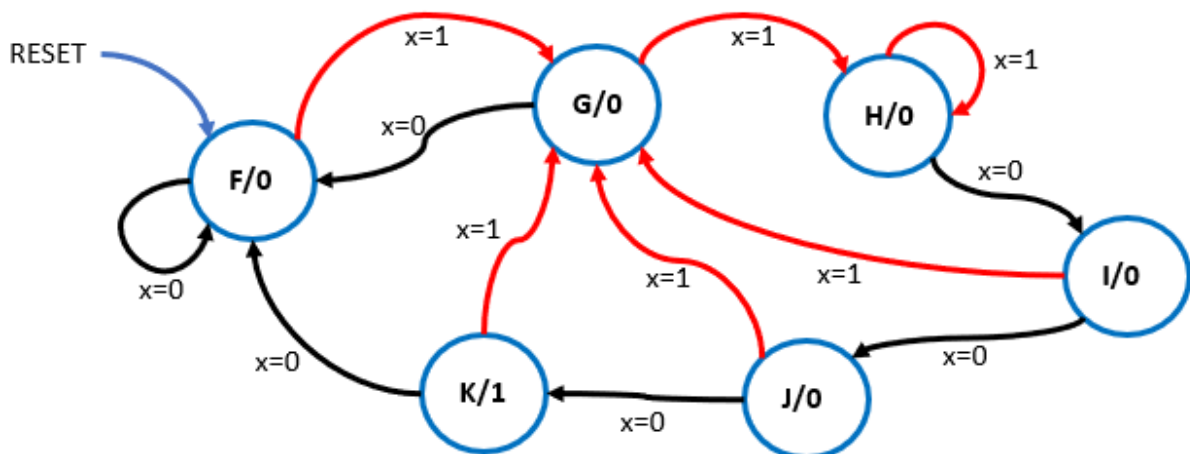
## Detector de Secuencias

A nuestro equipo (#6) le fue asignado el reconocimiento de las secuencias **1110**, **11000** y **100101** mediante el uso de Flip Flops de tipo JK. El primer paso a seguir fue la realización de los diagramas de estados para cada una de las secuencias por separado, así:

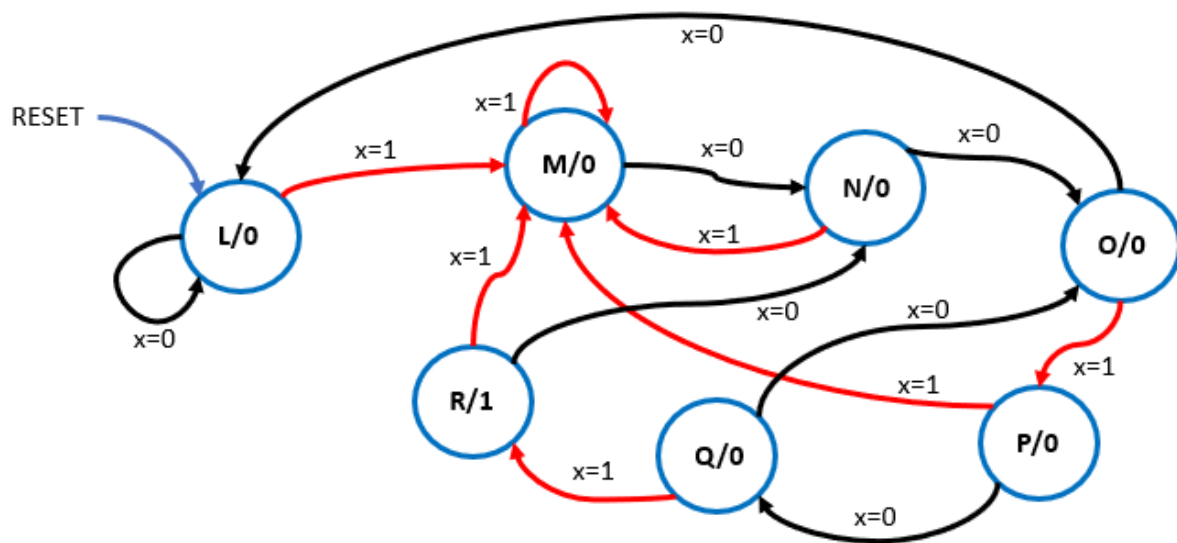
### Secuencia #1 (4 bits): 1110



### Secuencia #2 (5 bits): 11000



### Secuencia #3 (6 bits): 100101



Una vez realizados los 3 diagramas de estados, se realiza la tabla de estados y salidas y se codifican los estados en numeración binaria ascendente así:

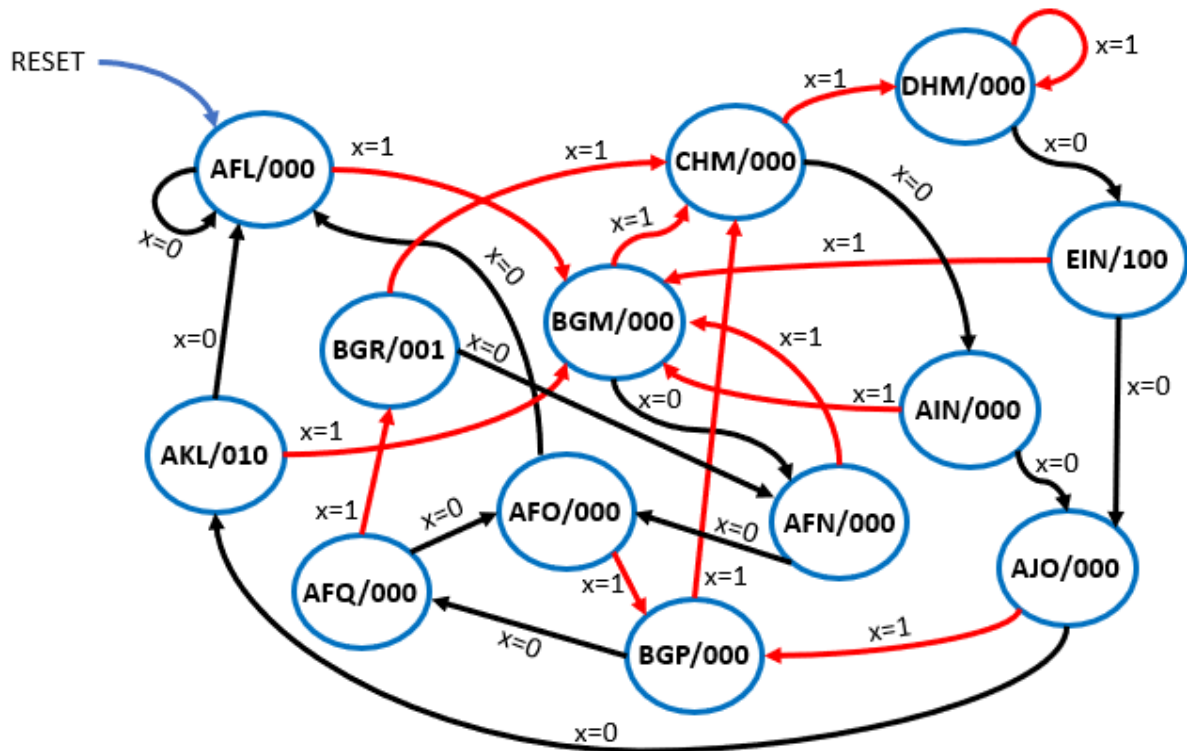
### Tabla de Estados y Salidas Codificada

Codificación	ESTADO ACTUAL	ESTADO SIGUIENTE		SALIDAS		
		x=0	x=1	S1	S2	S3
0000	AFL	AFL	BGM	0	0	0
0001	BGM	AFN	CHM	0	0	0
0010	AFN	AFO	BGM	0	0	0
0011	CHM	AIN	DHM	0	0	0
0100	AFO	AFL	BGP	0	0	0
0101	AIN	AJO	BGM	0	0	0
0110	DHM	EIN	DHM	0	0	0
0111	BGP	AFQ	CHM	0	0	0
1000	AJO	AKL	BGP	0	0	0
1001	EIN	AJO	BGM	1	0	0
1010	AFQ	AFO	BGR	0	0	0
1011	AKL	AFL	BGM	0	1	0
1100	BGR	AFN	CHM	0	0	1

El conjunto DC es {13, 14, 15}

Ahora, de manera opcional, realizamos el diagrama de estados para el reconocimiento de las 3 secuencias de forma simultánea.

## Todas las Secuencias



## Tabla Flip Flops

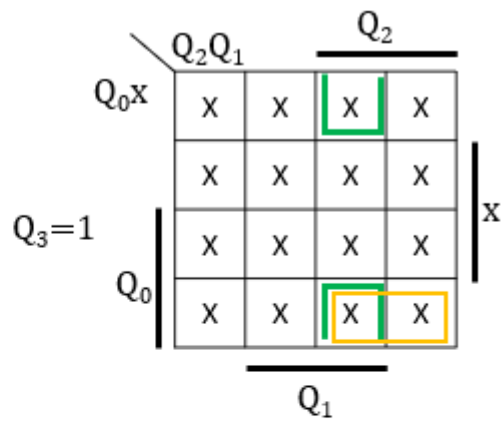
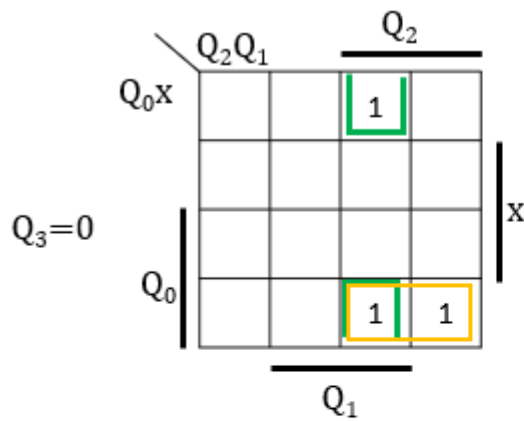
Con ayuda de la tabla de estados y salidas y de la tabla de excitación del Flip Flop asignado (JK), realizamos la tabla completa de los Flip Flops con el fin de obtener las ecuaciones de excitación del Flip Flop y de las salidas para luego ser minimizadas usando mapas de Karnaugh. La tabla se adjunta a continuación:

 **Tabla Flip Flops**

## Mapas de Karnaugh

Por último, se realizan los mapas de Karnaugh para cada una de las salidas y se simplifican todas las ecuaciones de excitación.

### Mapa para $J_3$



Implicantes primos:

$$Q_2Q_1x', Q_2Q_0x'$$

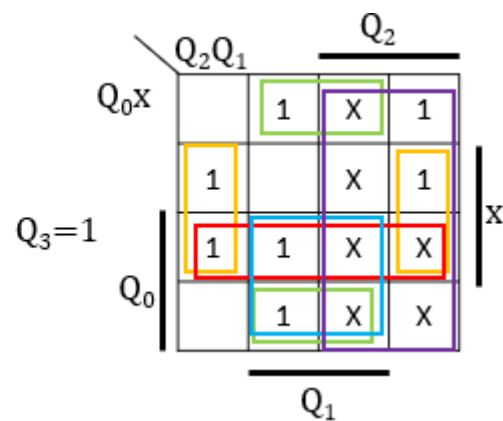
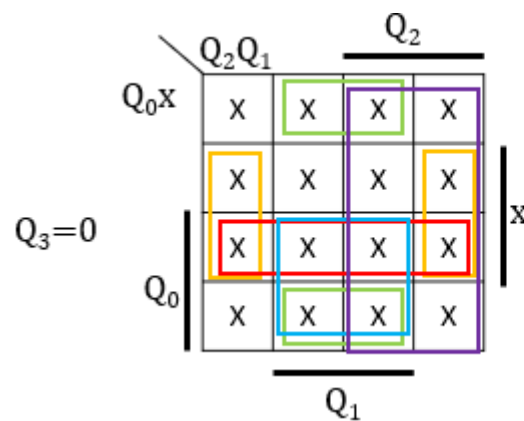
Implicantes primos esenciales:

$$Q_2Q_1x', Q_2Q_0x'$$

Expresión mínima:

$$J_3(Q_3, Q_2, Q_1, Q_0, x) = Q_2Q_1x' + Q_2Q_0x'$$

### Mapa para $K_3$



Implicantes primos:

$$Q_1x', Q_1'x, Q_2, Q_0x, Q_1Q_0$$

Implicantes primos esenciales:

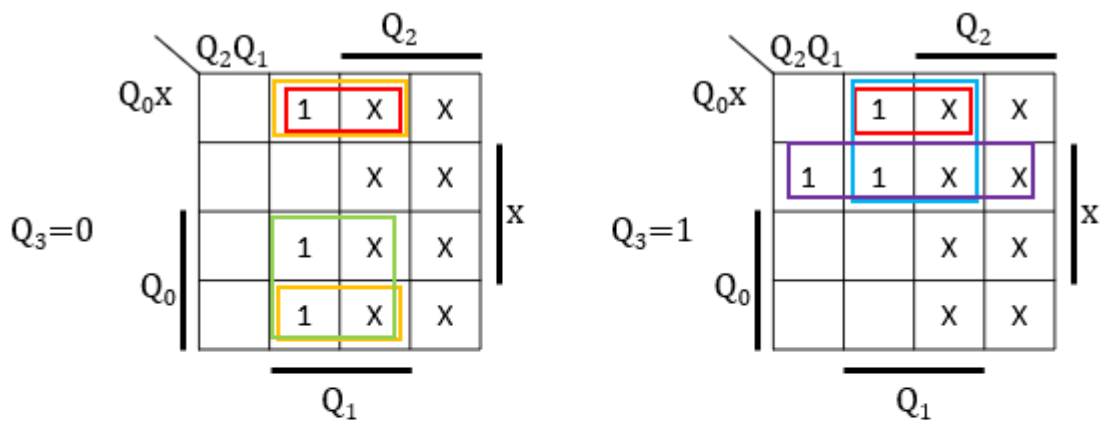
$$Q_1x', Q_1'x, Q_2$$

Expresión mínima:

$$K_3(Q_3, Q_2, Q_1, Q_0, x) = Q_1x' + Q_1'x + Q_2 + Q_1Q_0$$

$$K_3(Q_3, Q_2, Q_1, Q_0, x) = Q_1x' + Q_1'x + Q_2 + Q_0x$$

## Mapa para $J_2$



Implicantes primos:

$$Q_3' Q_1 Q_0, Q_3' Q_1 x', Q_1 Q_0' x', Q_3 Q_1 Q_0', Q_3 Q_0' x$$

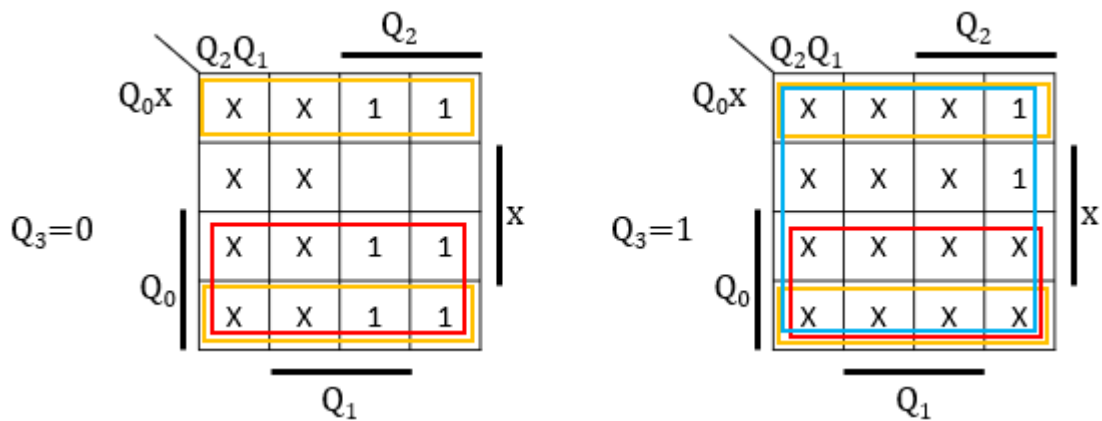
Implicantes primos esenciales:

$$Q_3' Q_1 Q_0, Q_3 Q_0' x$$

Expresión mínima:

$$J_2(Q_3, Q_2, Q_1, Q_0, x) = Q_3' Q_1 Q_0 + Q_3 Q_0' x + Q_1 Q_0' x'$$

## Mapa para $K_2$



Implicantes primos:

$$x', Q_0, Q_3$$

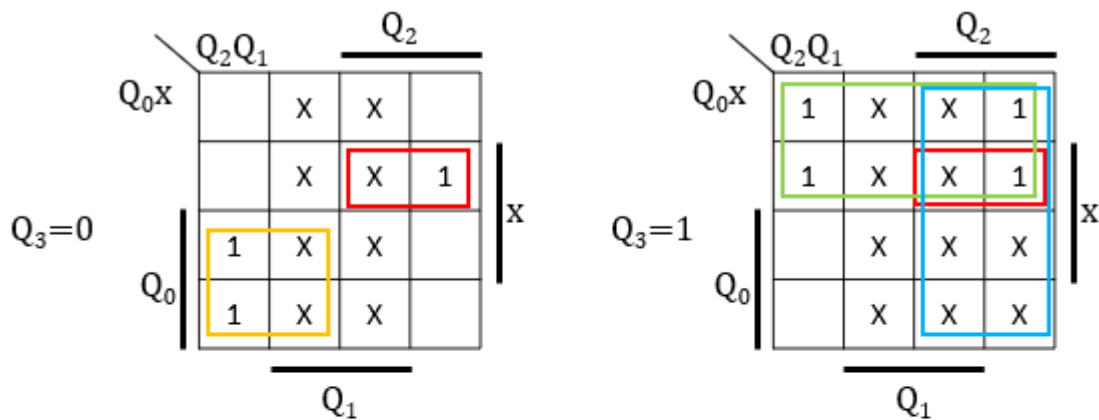
Implicantes primos esenciales:

$$x', Q_0, Q_3$$

Expresión mínima:

$$K_2(Q_3, Q_2, Q_1, Q_0, x) = Q_3 + Q_0 + x'$$

### Mapa para $J_1$



Implicantes primos:

$Q_3Q_0'$ ,  $Q_3'Q_2'Q_0$ ,  $Q_2Q_0'x$ ,  $Q_3Q_2$

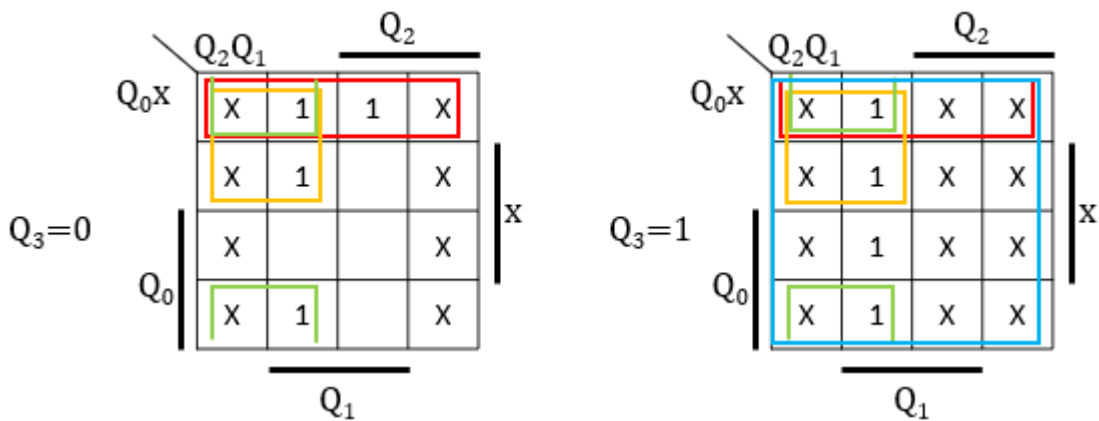
Implicantes primos esenciales:

$Q_3Q_0'$ ,  $Q_3'Q_2'Q_0$ ,  $Q_2Q_0'x$

Expresión mínima:

$$J_1(Q_3, Q_2, Q_1, Q_0, x) = Q_3Q_0' + Q_3'Q_2'Q_0 + Q_2Q_0'x$$

### Mapa para $K_1$



Implicantes primos:

$Q_2'x'$ ,  $Q_2'Q_0'$ ,  $Q_0'x'$ ,  $Q_3$

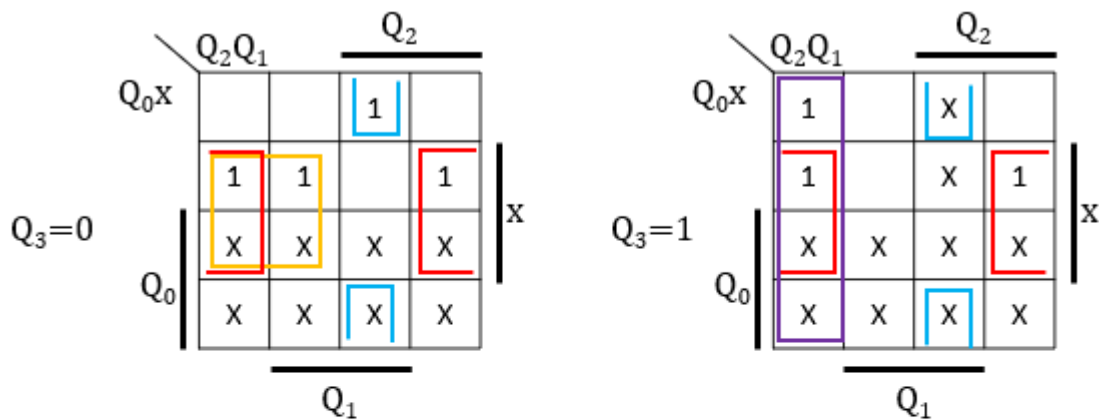
Implicantes primos esenciales:

$Q_2'x'$ ,  $Q_2'Q_0'$ ,  $Q_0'x'$ ,  $Q_3$

Expresión mínima:

$$K_1(Q_3, Q_2, Q_1, Q_0, x) = Q_2'x' + Q_2'Q_0' + Q_0'x' + Q_3$$

### Mapa para $J_0$



Implicantes primos:

$$Q_3'Q_2'x, Q_1'x, Q_2Q_1x', Q_3Q_2'Q_1'$$

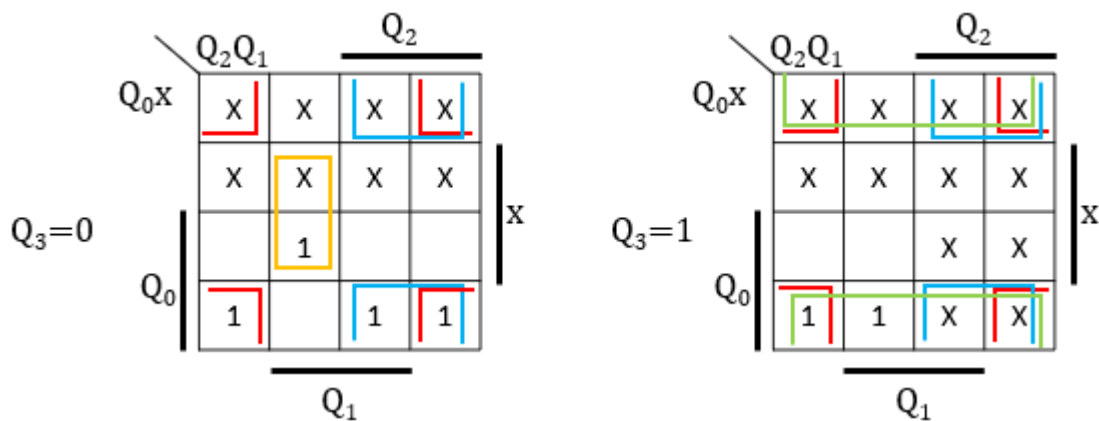
Implicantes primos esenciales:

$$Q_3'Q_2'x, Q_1'x, Q_2Q_1x', Q_3Q_2'Q_1'$$

Expresión mínima:

$$J_0(Q_3, Q_2, Q_1, Q_0, x) = Q_3'Q_2'x + Q_1'x + Q_2Q_1x' + Q_3Q_2'Q_1'$$

### Mapa para $K_0$



Implicantes primos:

$$Q_3x', Q_3'Q_2'Q_1x, Q_1'x', Q_2x'$$

Implicantes primos esenciales:

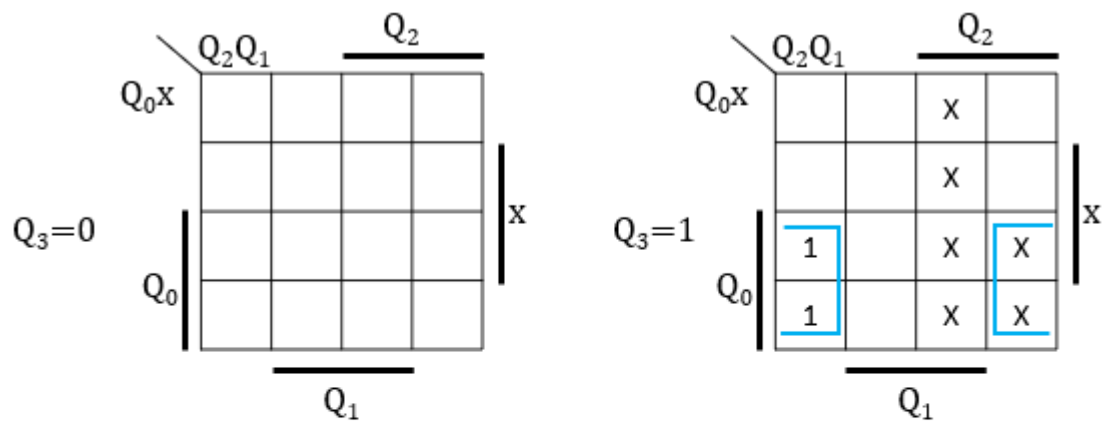
$$Q_3x', Q_3'Q_2'Q_1x, Q_1'x', Q_2x'$$

Expresión mínima:

$$K_0(Q_3, Q_2, Q_1, Q_0, x) = Q_3x' + Q_3'Q_2'Q_1x + Q_1'x' + Q_2x'$$



### Mapa para $S_1$



Implicantes primos:

$Q_3Q_1'Q_0$

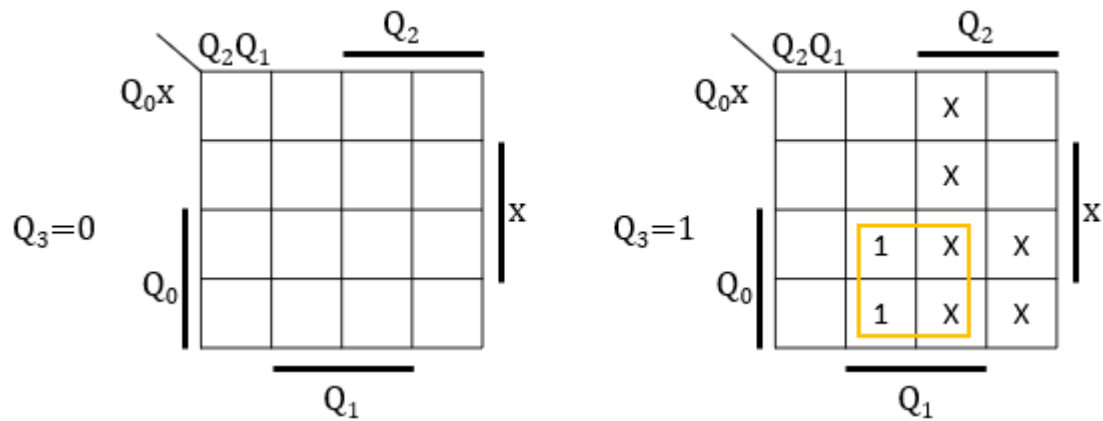
Implicantes primos esenciales:

$Q_3Q_1'Q_0$

Expresión mínima:

$$J_2(Q_3, Q_2, Q_1, Q_0, x) = Q_3 Q_1' Q_0$$

### Mapa para $S_2$



Implicantes primos:

$Q_3Q_1Q_0$

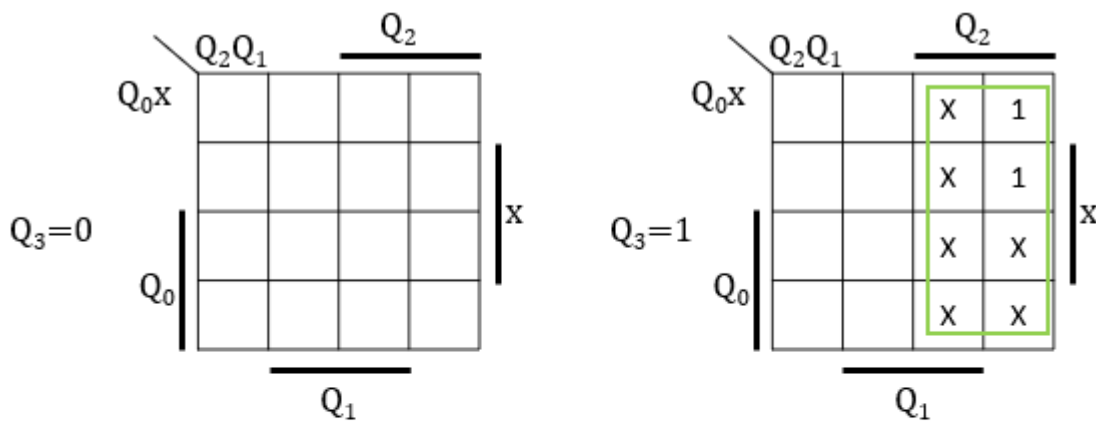
Implicantes primos esenciales:

$$Q_3Q_1Q_0$$

Expresión mínima:

$$J_2(Q_3, Q_2, Q_1, Q_0, \mathbf{x}) = Q_3 Q_1 Q_0$$

### Mapa para $S_3$



Implicantes primos:

$Q_3Q_2$

Implicantes primos esenciales:

$Q_3Q_2$

Expresión mínima:

$J_2(Q_3, Q_2, Q_1, Q_0, x) = Q_3Q_2$

Con todo lo anterior, se crea el circuito *Sequence\_Detector*, el cual tendrá como entradas la variable  $X$ , la señal del reloj universal CLK y la señal de RESET universal. Tendrá 4 salidas, siendo 3 de ellas  $S_1$ ,  $S_2$  y  $S_3$ , las cuales se corresponden a la detección de las secuencias 1, 2 o 3 respectivamente. La otra salida llamada ACTUAL servirá para visualizar en el circuito principal el estado actual de la máquina.

# Contador

Para la realización del circuito *Event\_Counter*, se realizó un contador síncrono mediante el uso de 8 Flip Flops tipo JK para cada secuencia a detectar, dando un total de 24 Flip Flops en total. Se decidió utilizar 8 Flip Flops para poder contar hasta un máximo de 255 ocurrencias para cada secuencia. La razón de lo anterior, es que a pesar de que el máximo de bits que pueden ser capturados en el tiempo especificado(30 segundos) con la frecuencia de reloj especificada(8Hz) son 240 bits, y suponiendo que todos los bits que se capturaron nos dejaran contar el máximo de ocurrencias de la secuencia 1 (60 ocurrencias), se decidió darle un valor más alto para que el circuito tuviera más libertad en caso de que se decidan usar frecuencias más altas o tiempos más largos. Sin embargo pudo haberse realizado el mismo circuito sólo con 6 Flip Flops con los cuales se podrían contar hasta 63 ocurrencias de cada secuencia, los cuales eran suficientes para cumplir con los requisitos del laboratorio.

El primer Flip Flop de cada secuencia, correspondiente al LSB del contador, depende de la entrada  $S_i$ , la cual indica si se detectó una ocurrencia de la secuencia  $i$ , y del reloj universal. Al activarse ambas entradas, el valor del Flip Flop cambia y se contaría una ocurrencia de la secuencia, la cuál será guardada en PINES de 8 bits que posteriormente serán las salidas del circuito. Los Flip Flops subsiguientes dependen no solo del reloj universal, sino también de la operación lógica AND entre la salida(Q) y la entrada(JK) del Flip Flop anterior.

Cada estado de Flip Flop se guarda en una posición de un PIN de 8 bits, el cuál, al final de la simulación, albergará el número de veces que cada secuencia fue detectada. Por ende, las salidas del circuito serán 3 PINES de 8 bits cada uno,  $C_1$ ,  $C_2$  y  $C_3$  los cuales serán el número de veces que cada secuencia fue detectada(en binario).

**NOTA:** El valor de J y K para todos los Flip Flops deben ser iguales, ya que de acuerdo a la tabla característica de los Flip Flops tipo JK sólo habrá cambio de estado respecto al estado anterior cuando ambos sean 1.

# Visualización

Para la realización del circuito *Visualization*, se requirió el uso de 2 circuitos más los cuáles serán descritos a continuación.

## “Traductor” de Binario a Decimal

Este circuito, llamado dentro del proyecto *BCD*, tiene como única función recibir como entrada el contador  $C_i$  (salida del circuito *Event\_Counter*), y separarlo en cifras decimales (Unidad, Decena, Centena) mediante divisiones sucesivas entre 10 utilizando el componente “Divisor” de la biblioteca de Logisim. Luego, envía las 3 salidas UNIDAD, DECENA y CENTENA como PINES de 4 bits cada uno.

## Configuración de LEDs

El circuito *LED\_Config* tiene como entradas la UNIDAD, DECENA y CENTENA calculadas en el circuito *BCD*; y como salidas los valores que deberían tener cada uno de los segmentos del LED de 7 segmentos de acuerdo al número que se desee mostrar. Estos valores estarán almacenados en PINES de 7 bits. Para la realización de este circuito se hizo uso de una tabla de verdad cuyas entradas son los 4 bits de cada cifra decimal a representar y cuyas salidas son los 7 segmentos del LED. La tabla se muestra a continuación:

	NÚMERO EN BINARIO				SEGMENTOS LED						
NÚMERO DECIMAL	$Q_3$	$Q_2$	$Q_1$	$Q_0$	A	B	C	D	E	F	G
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

El conjunto DC es {10, 11, 12, 13, 14, 15}

## Circuito Principal

En el circuito principal, el cual es llamado *main*, se juntan todos los demás circuitos y se agregan los LEDs, a los cuales se les coloca su respectiva configuración de acuerdo a si es unidad, decena o centena, todo esto con la salida del bloque del circuito *Visualization*.