|  |  |  |  |
| --- | --- | --- | --- |
| Informe de trabajo  Ejercicio de entrega obligatoria TP3 | |  | G1  23 de mayo de 2023 |
| Facultad de Ingeniería UNLP  E0301 Introducción al Diseño Lógico  Curso 2023 | |  |
|  | |  |  |
| s | |  |  |
|  |  |  | |
| * Caciani Toniolo, Melina   melicaciani@gmail.com | |  | 02866/1 |
| * Chanquía, Joaquín   joaquin.chanquia@alu.ing.unlp.edu.ar | |  | 02887/7 |
| * Larsen, Mateo Emmanuel   larsenmateo.ml@gmail.com | |  | 02993/7 |
| * Ollier, Gabriel   gabyollier@hotmail.com | |  | 02958/4 |
|  | |  |  |

# ENUNCIADO

En la figura 1 se muestran cuatro Flip Flops (FF) de distintos tipos:

SR, JK, D y T

Imagen que contiene reloj

Descripción generada automáticamente

En la figura 2 se muestran tres señales, S1, S2 y CLK.

Gráfico

Descripción generada automáticamente

# CUESTIONARIO

¿Cuántos estados son necesarios?

Son necesarios 11 estados para representar las combinaciones de leds necesarias, 2 para todos los leds prendidos y apagados, 3 para los leds separados por colores, 3 para cuando se prenden 4 de cada lado y 3 para cuando se prenden 3 de cada lado.

¿Cuántos FFs?

Se necesitan 4 flip-flops para representar 16 estados que es la siguiente potencia de 2 que la cantidad de estados (11).

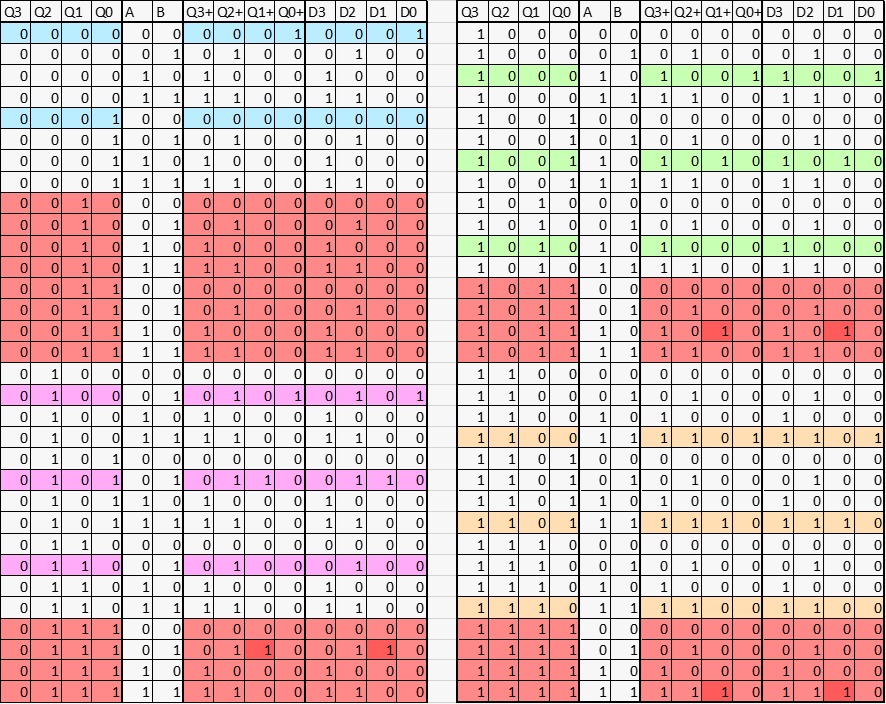
¿Cuántos estados indefinidos quedan?

Quedan 5 estados indefinidos.

Dibuje el diagrama de estados

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Foto diagrama de estados\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Haga las tablas de transición de estados y de excitaciones



Colores:

Rojo claro: Estados indefinidos

Rojo oscuro: Bits que no siguen un patron cambiados a 1 para simplificar los mapas de Karnaugh

Otros colores son los cambios de estado entre los que pertenecen a una misma entrada (los marcamos porque entre esos son los únicos que no siguen el patron general del resto de la tabla)

Celeste: Alternado entre todos los LED prendidos o apagados (A=0,B=0)

Rosa: Secuencia Rojo, Verde, Amarillo (A=0,B=1)

Verde: Secuencia de 4 Leds por lado (A=1,B=0)

Naranja: Secuencia de 3 Leds por lado (A=1,B=1)



Mapas de Karnaugh de los bit del nuevo estado.

Se muestran también los mapas de Karnaugh reducidos de Q3+, Q2+ y Q1+.

Las expresiones que aparecen son las resultantes del estudio de los mapas de karnaugh, y luego su correspondiente reducción utilizando algebra de Boole.

Diseñe la lógica de transición.

Escriba las funciones de salida y sinteticelas.



Tabla de verdad de las salidas, un 1 indica que la luz esta prendida y un 0 que esta apagada.

La primera letra de la salida indica el color de la luz (V=Verde; A=Amarillo; R=Rojo) y la segunda la posición de esta (A=Abajo; I=Izquierda; D=Derecha). (Por ejemplo, la luz Verde de la Izquierda es VI)

Para el caso de las luces rojas definimos que la luz de abajo a la derecha es la que tomamos como que se encuentra abajo y la de abajo a la izquierda es la de la izquierda.

Para las salidas de los estados indefinidos pusimos en 1 los valores que simplificaban mejor los mapas de Karnaugh. Esto resultaría en que los primeros segundos al encenderse la maquina en caso de caer en alguno de estos estados se prenderían algunas luces de forma aleatoria, pero al siguiente ciclo de reloj se corregiría e iría al caso correspondiente.



Mapas de Karnaugh de cada LED.

Texto en fondo blanco

Descripción generada automáticamente

Funciones de salida de cada LED, Utilizamos algebra de Boole y para simplificar los circuitos en Quartus utilizamos ley de Morgan para cambiar de /A./B a /(A + B).

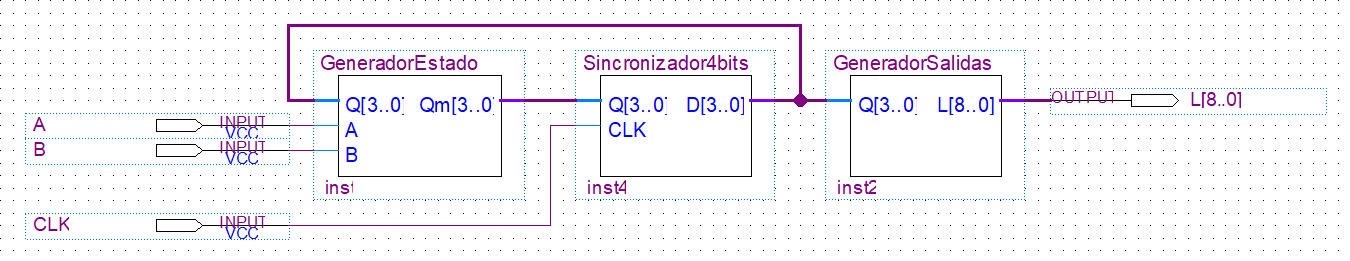
Implemente con FFs tipo D y simule en Quartus

La maquina de Moore final.

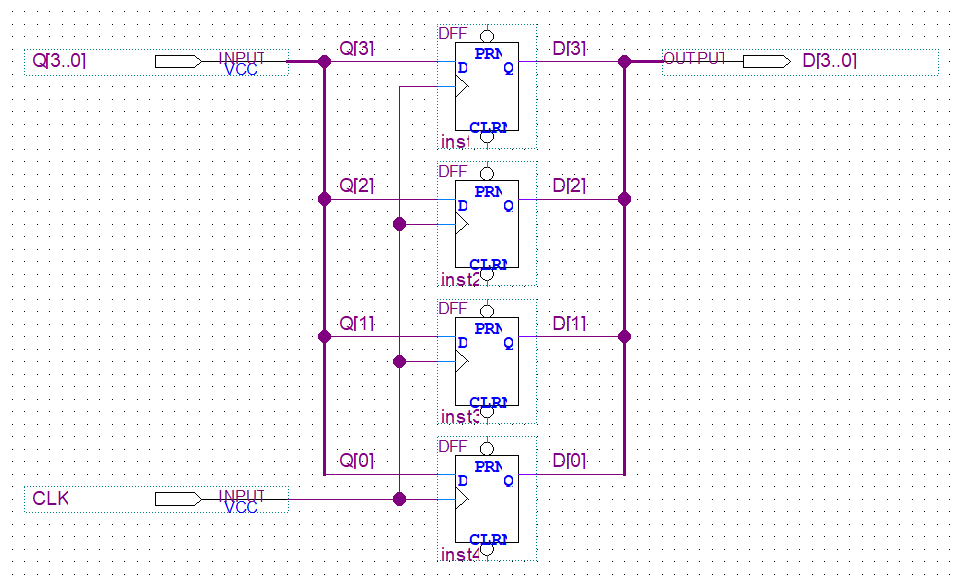
Q[3..0] es un bus que contiene el estado actual, Q[0] hace referencia al bit Q0 en las tablas de verdad y así con cada bit.

L[8..0] es un bus que tiene las salidas guardándose en este si un LED esta prendido o no así:

L[8]=VA ; L[7]=VI ; L[6]=VD ; L[5]=AA ; L[4]=AI ; L[3]=AD ; L[2]=RA ; L[1]=RI ; L[0]=RD ;

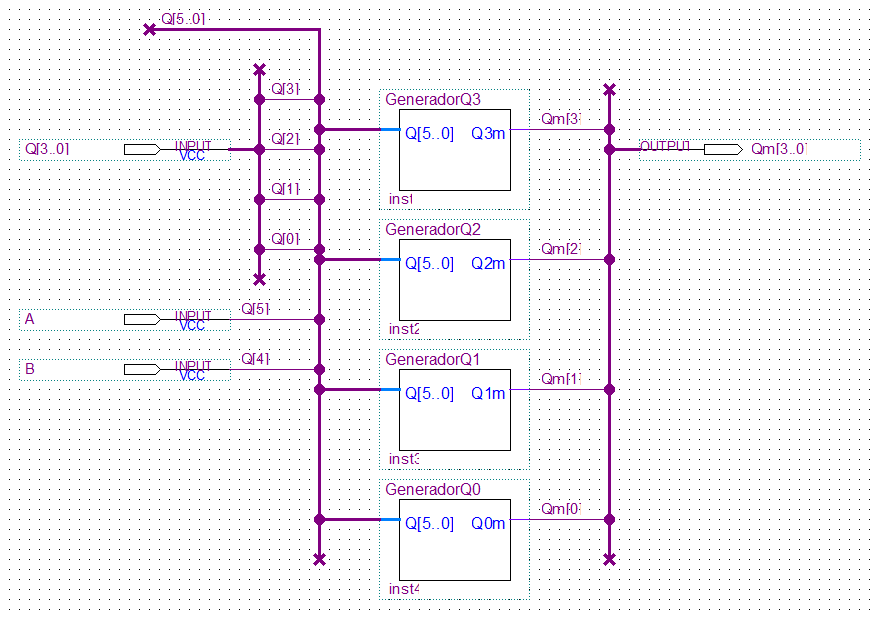


El sincronizador4bits contiene los Flip-Flop tipo D:



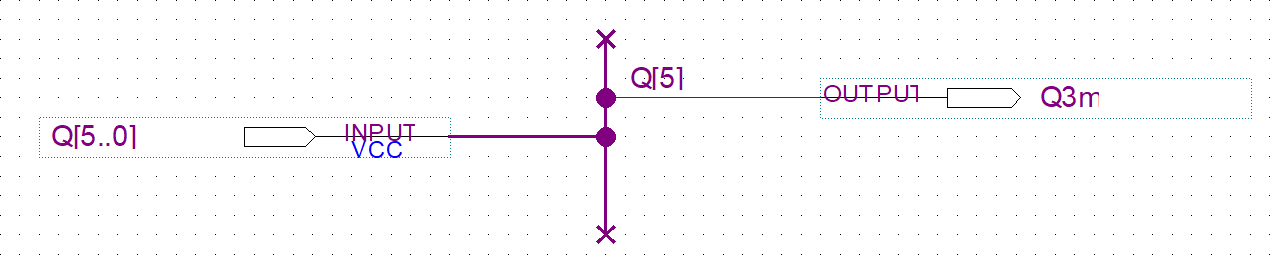
El GeneradorEstado contiene la lógica para pasar de un estado a otro:

Q[5..0] es un bus en el que están el estado anterior y las entradas A (en Q[5]) y B (En Q[4])

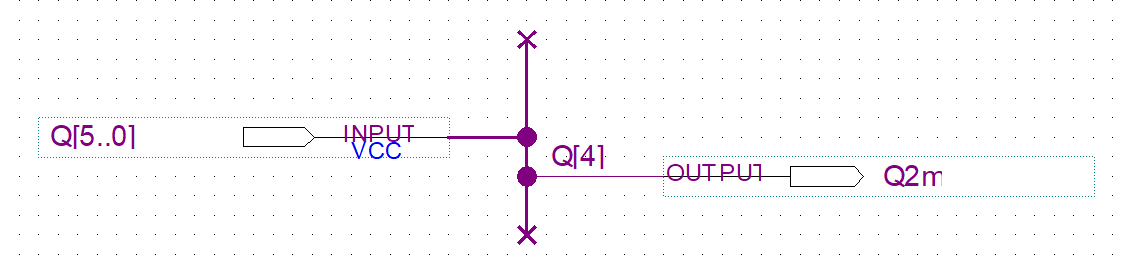


Cada caja contiene la lógica para el cambio de un bit especifico (el que dice en el nombre)

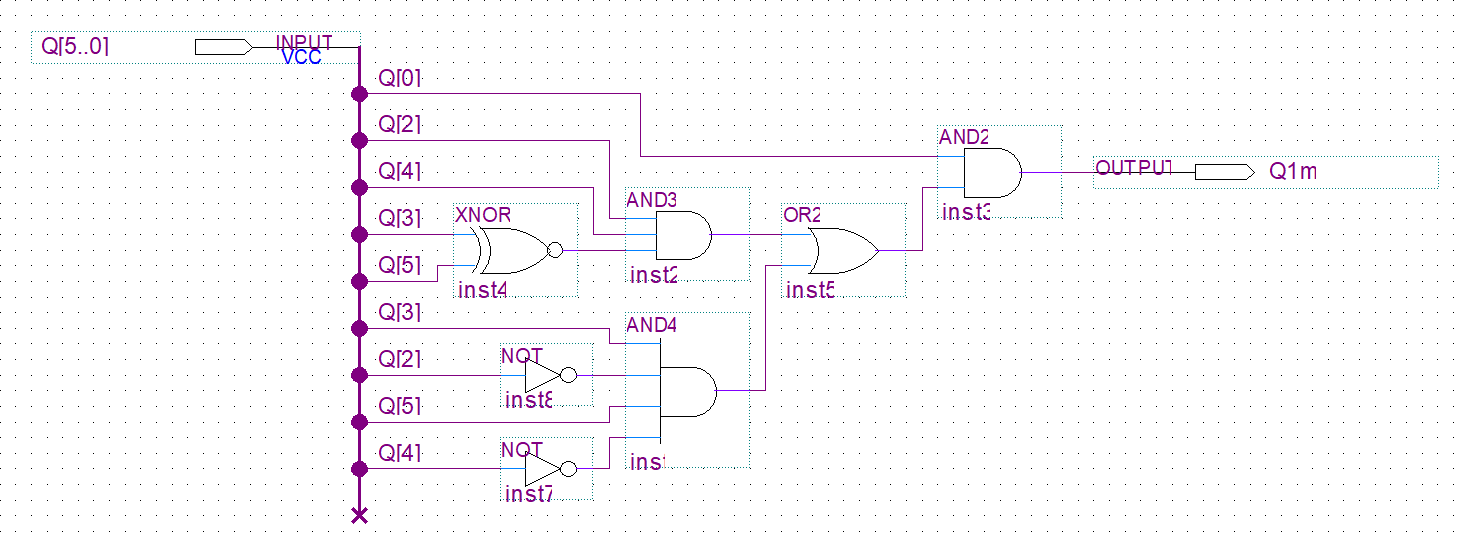
GeneradorQ3:



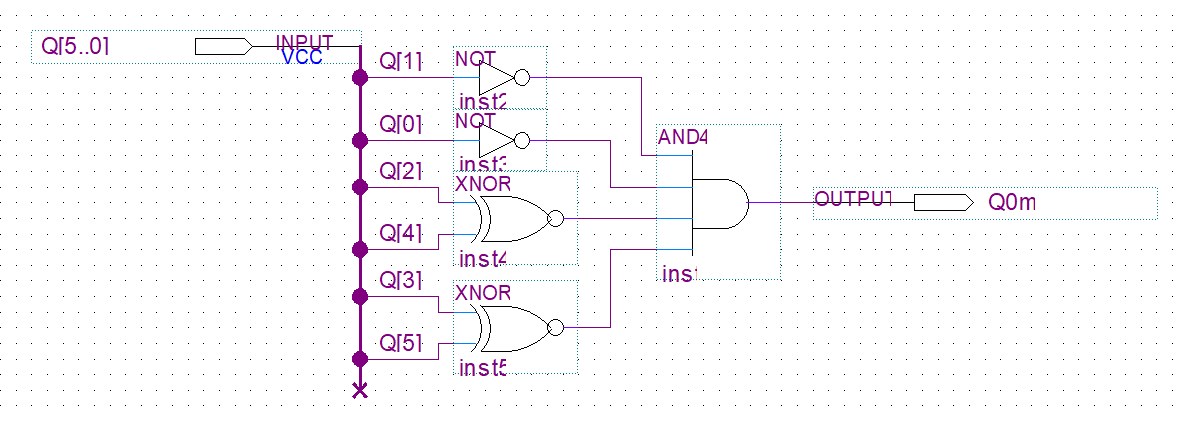
GeneradorQ2:



GeneradorQ1:



GeneradorQ0:

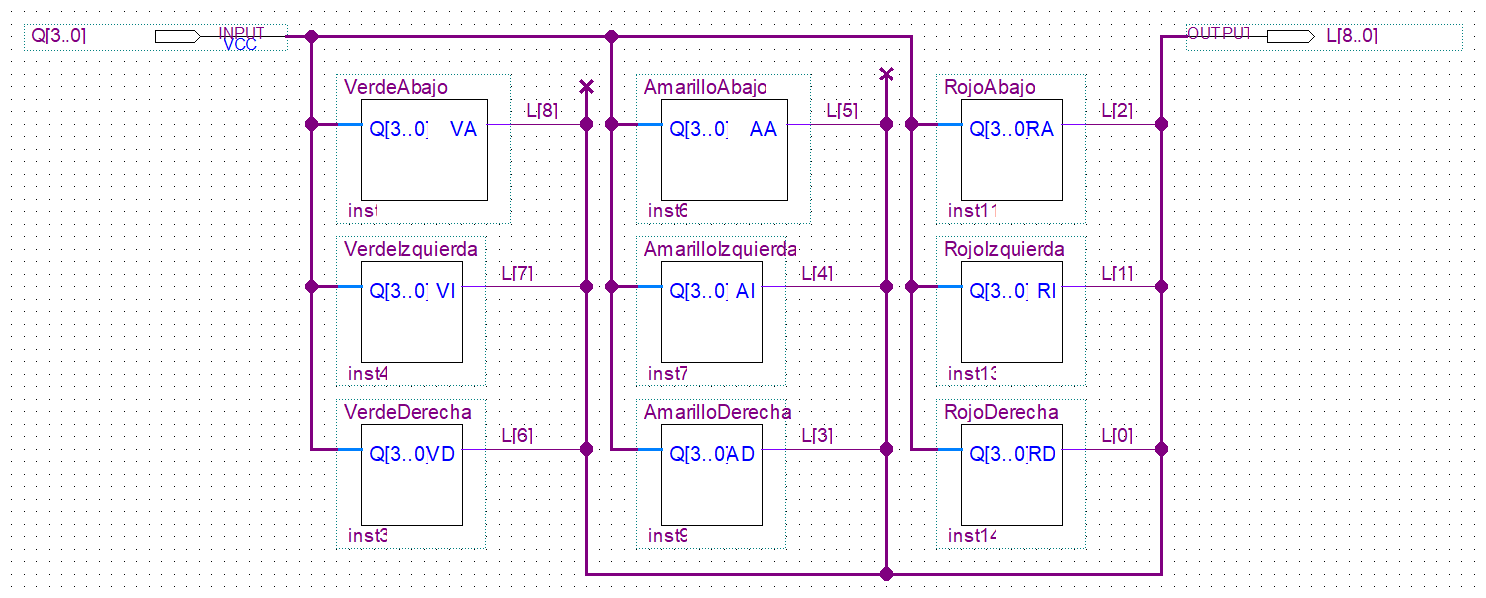


Utilizando Quartus simulamos el GeneradorEstado para ver su funcionamiento y comparando con la tabla de verdad de este nos dimos cuenta que funciona como debería:

Escala de tiempo

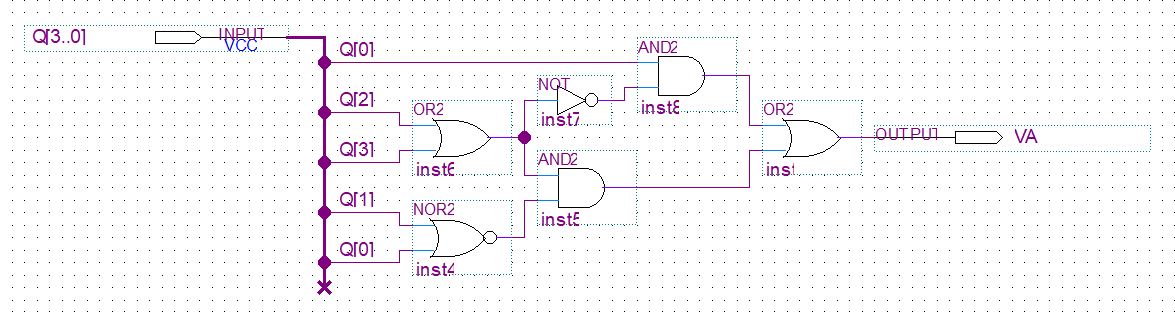
Descripción generada automáticamente con confianza media

El GeneradorSalidas tiene la lógica que indica que LEDs deben ser prendidos en cada estado:

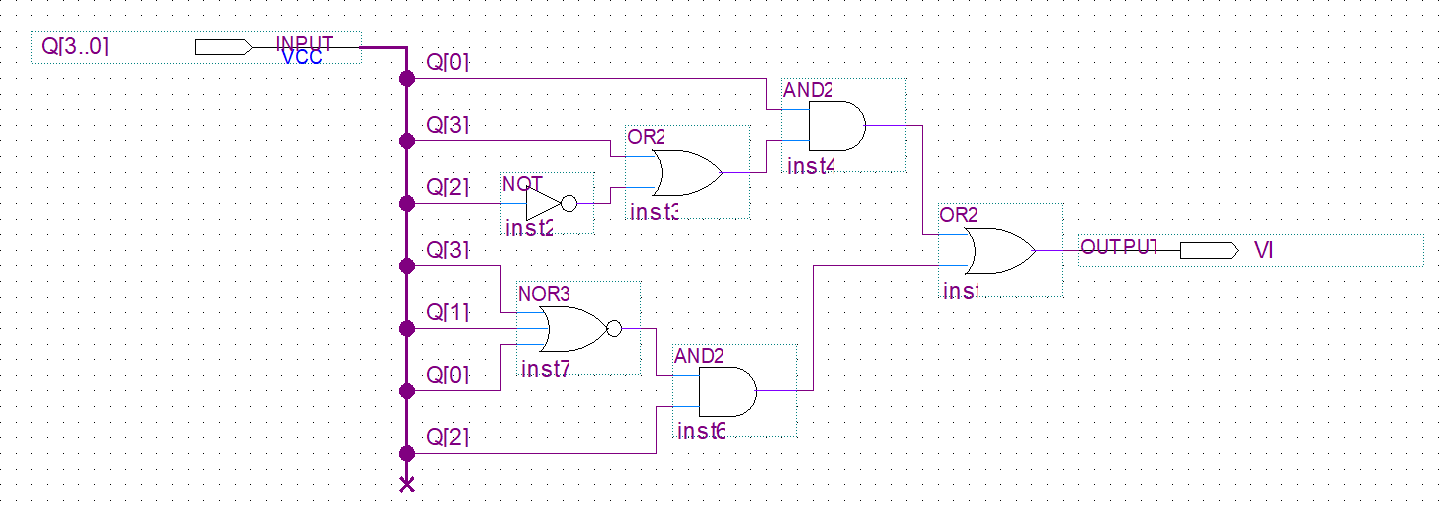


Cada caja contiene la lógica para que se prenda un LED en especifico (el que aparece en el nombre):

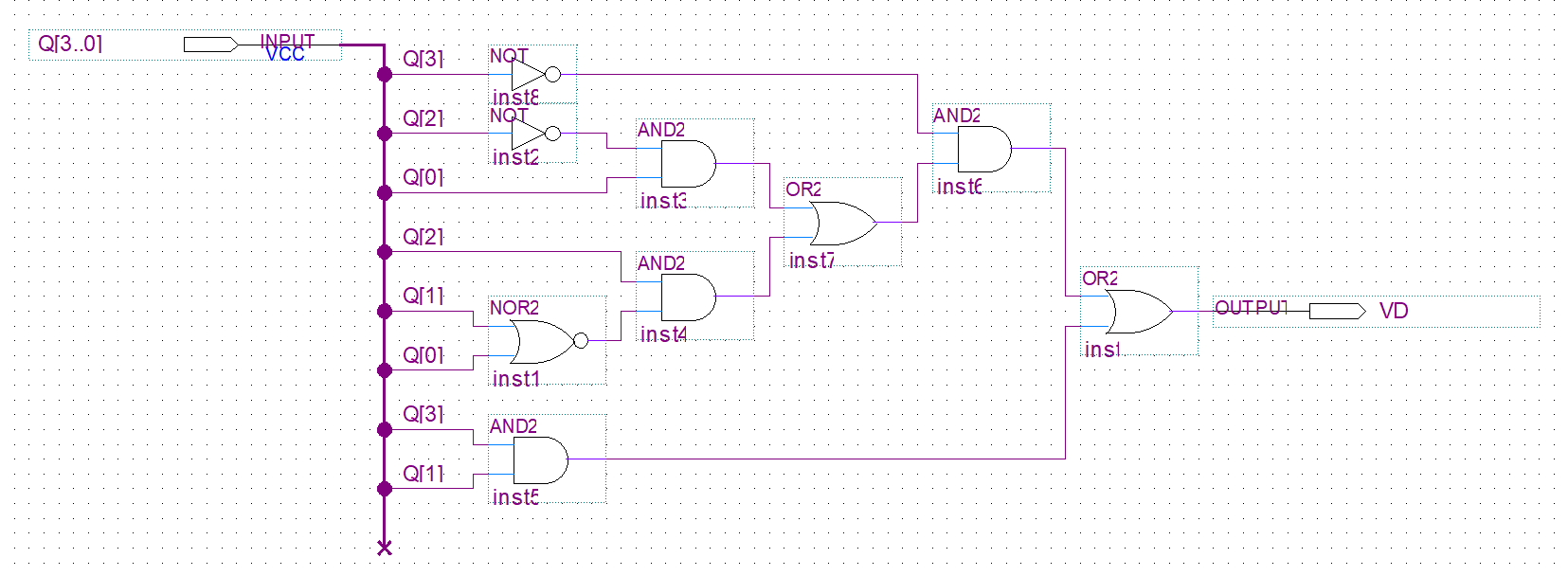
VerdeAbajo:



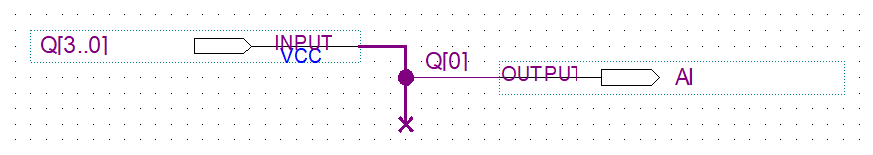
VerdeIzquierda:



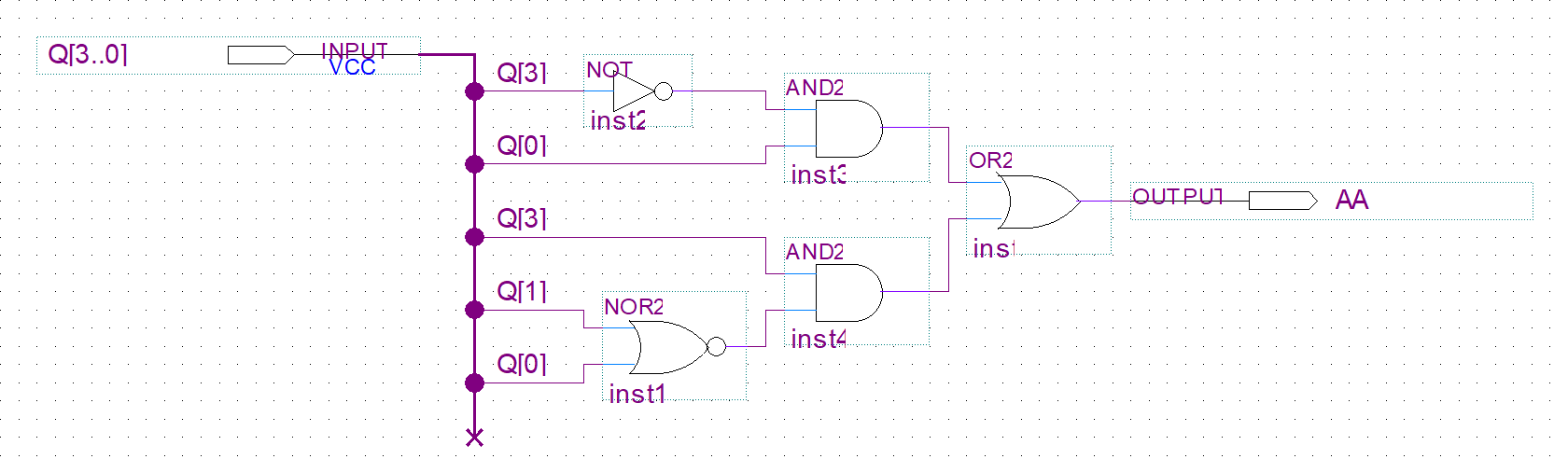
VerdeDerecha:



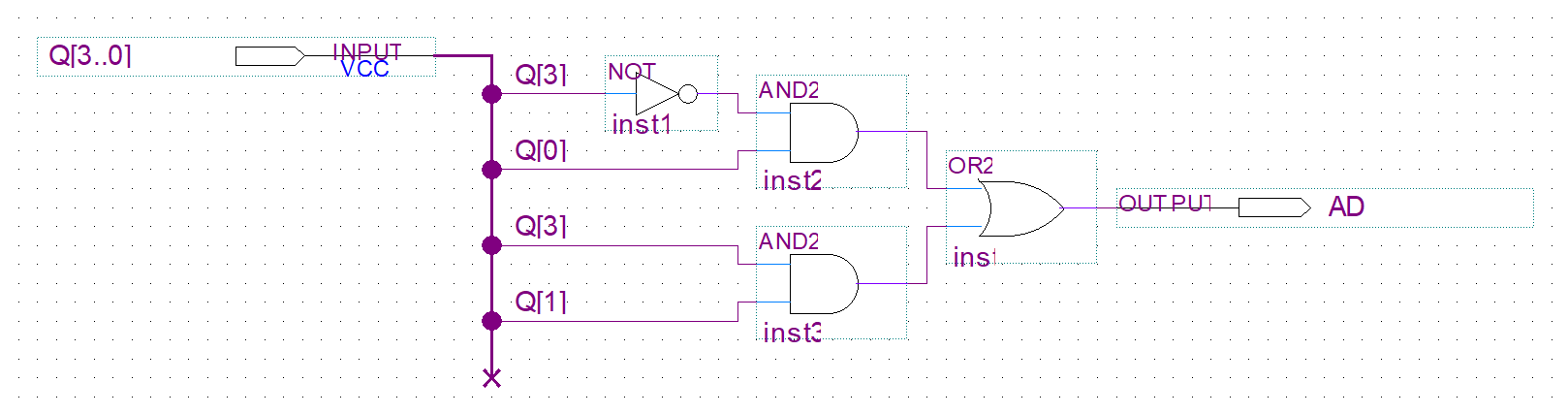
AmarilloIzquierda:



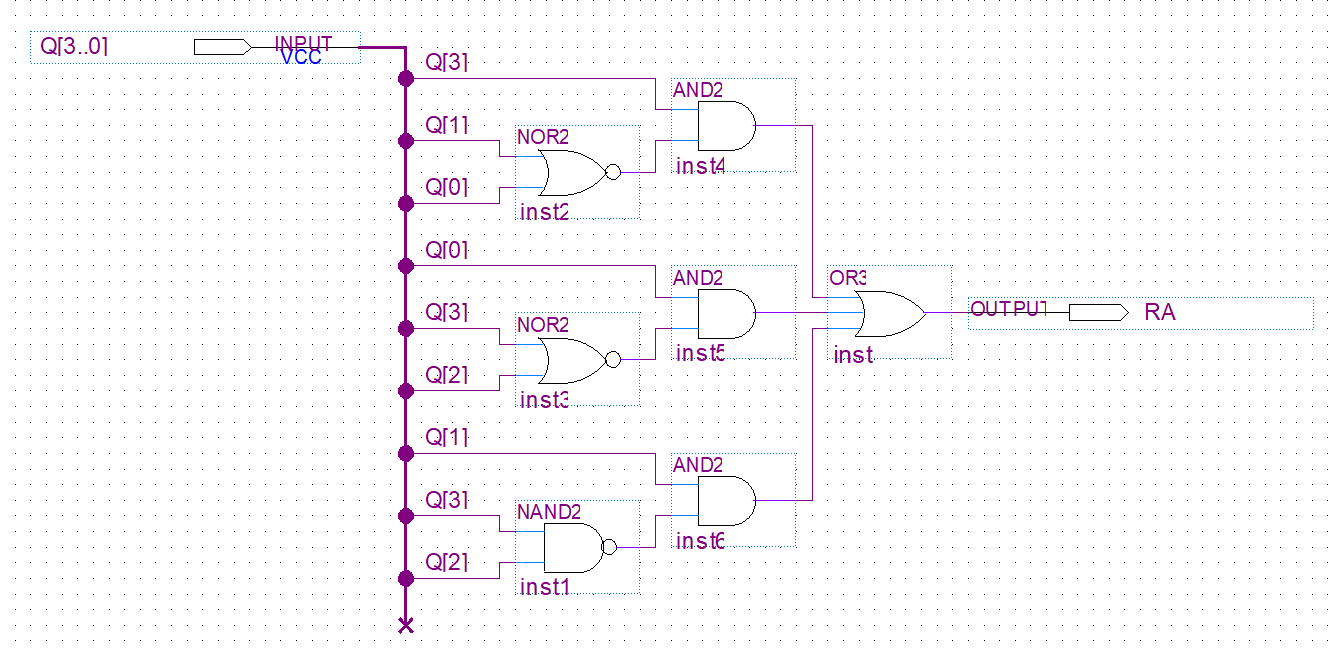
AmarilloAbajo:



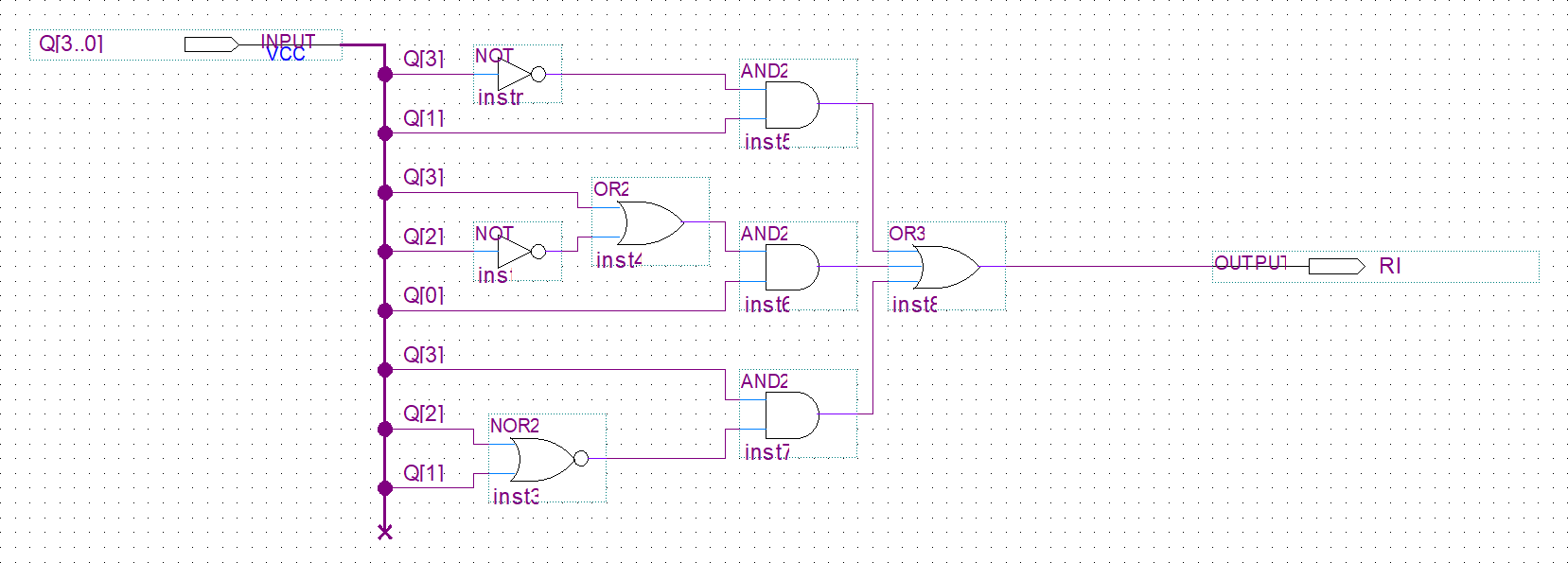
AmarilloDerecha:



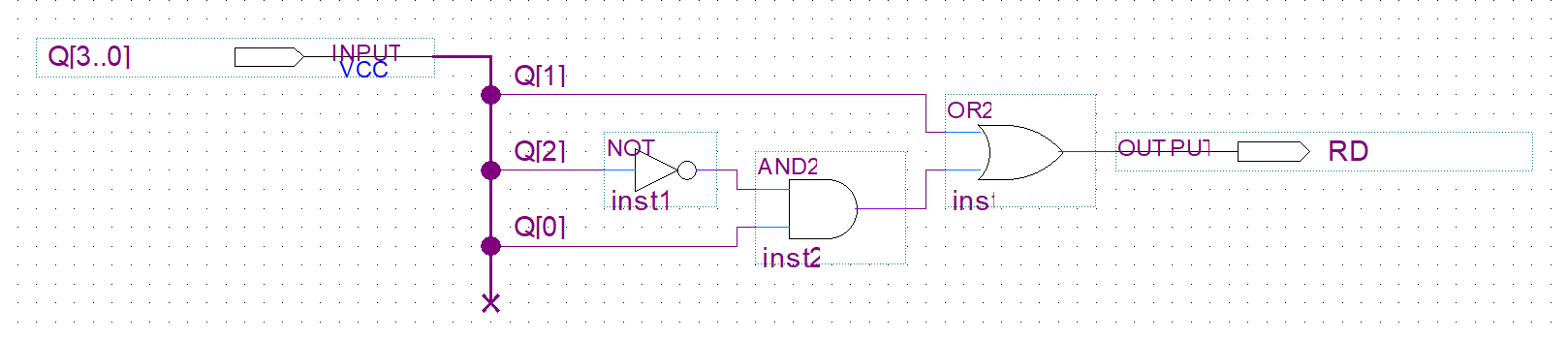
RojoAbajo:



RojoIzquierda:



RojoDerecha:



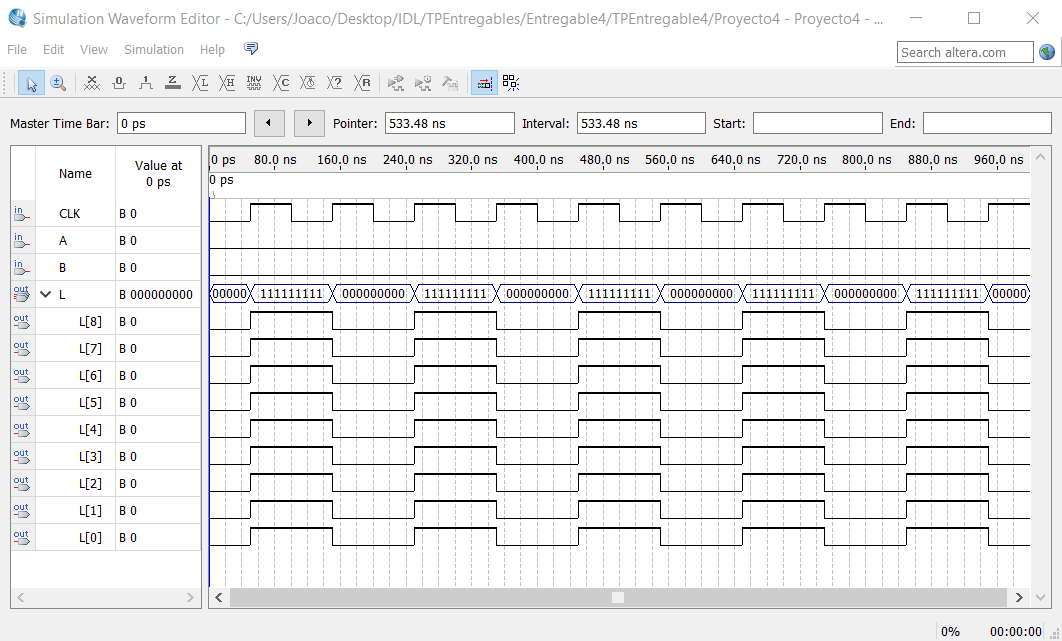
Utilizando Quartus simulamos el generador de salidas para ver que funcione:

Imagen que contiene Tabla

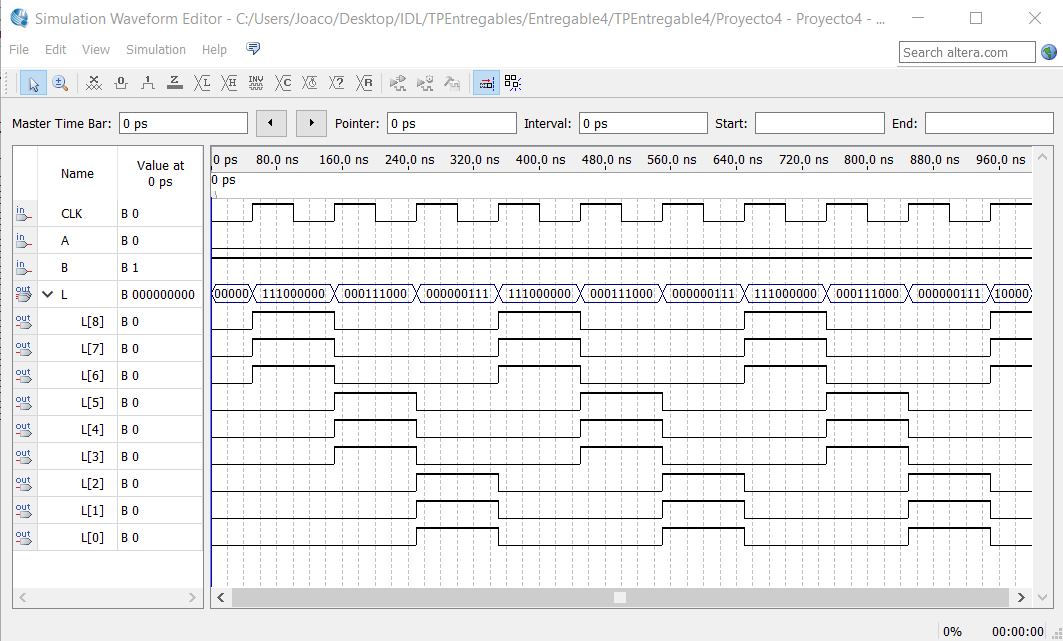
Descripción generada automáticamente

Por ultimo tenemos las pruebas que hicimos para comprobar el funcionamiento de la maquina en su totalidad:

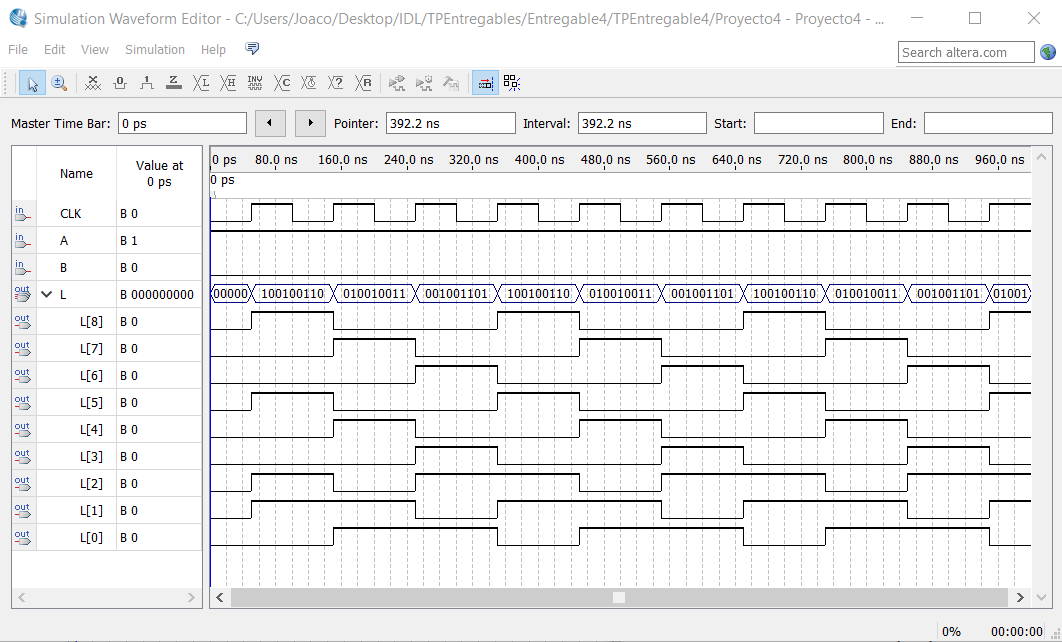
Entrada A=0 y B=0:



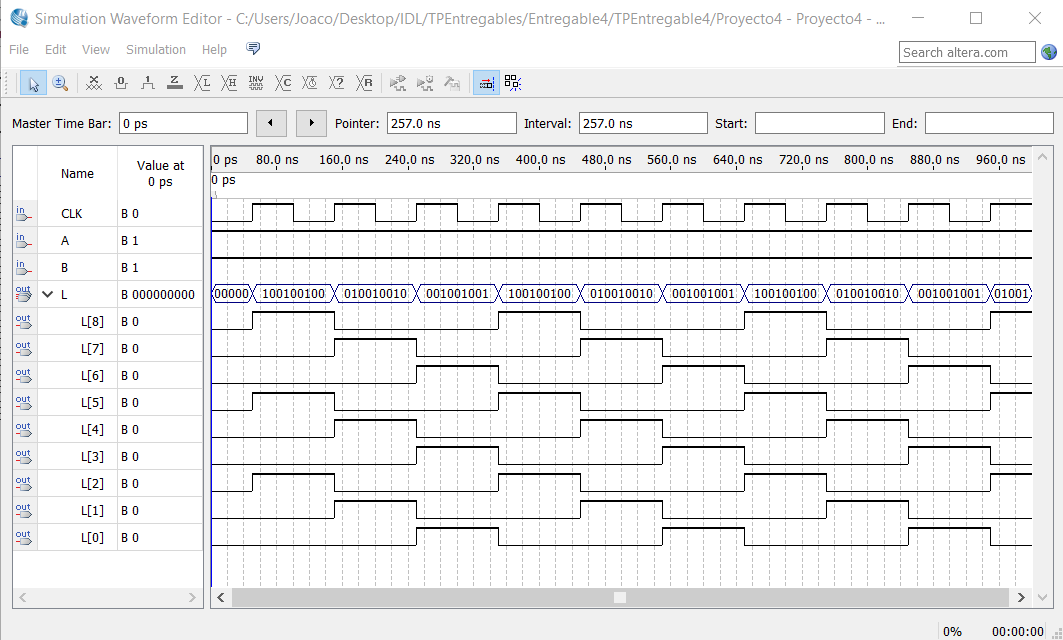
Entrada A=0 y B=1:



Entrada A=1 y B=0:



Entrada A=1 y B=1:



Y una prueba general en la que se usan las 4 combinaciones posibles de entrada para ver las transiciones:

