

## Reporte Contador de 0 a 9675

Jhontan Alexander Gomez Gamboa

Profesor : Luis René Vela García

### 1 Introducción experimental

---

El diseño y la implementación del circuito digital tiene la funcionalidad de incrementar en una unidad el contador con cada accionar de un interruptor, mostrando el conteo en 4 display de 7 segmentos uno para cada bit.

### 2 Ambiente experimental

---

- Quartus (Quartus Prime 17.1) Lite Edition.
- FPGA Cyclone IV de la marca ALINX.

### 3 Programa.

El módulo principal top\_contador contendrá la instanciación y la interfaz de conexión de los módulos del circuito.

El diseño consiste en un módulo divisor de reloj el cual permitirá tener 2 frecuencias distintas, una para el contador a 2.98 Hz, y la segunda señal de 3333.3 Hz utilizado para encender y apagar por ciclo cada uno de los display, a la primera señal se conectarán 4 módulos contadores de 4 bits los cuales serán circuitos secuenciales los cuales se evaluarán cada flanco positivo de reloj y tendrán un reset síncrono ,misma salida que se dirigirá tanto al módulo controlador como a sus respectivos decodificadores a 7 segmentos,

El módulo controlador el cual manejará las condiciones bajo las que operan los contadores, este será de lógica combinacional.

Los módulos decodificadores permitirán tomar una entrada combinada de 4 bits los cuales representan un número y de salida nos entregará la codificación que representa el cómo se encenderán los segmentos del display.

Modulo reg scan el cual toma una frecuencia la cual cambiara la salida que estará conectada a los display de 7 segmentos, esta señal cambiara rápidamente permitiendo encender y apagar cada uno de los display de forma secuencial provocando que el ojo humano lo perciba como un encendido constante.

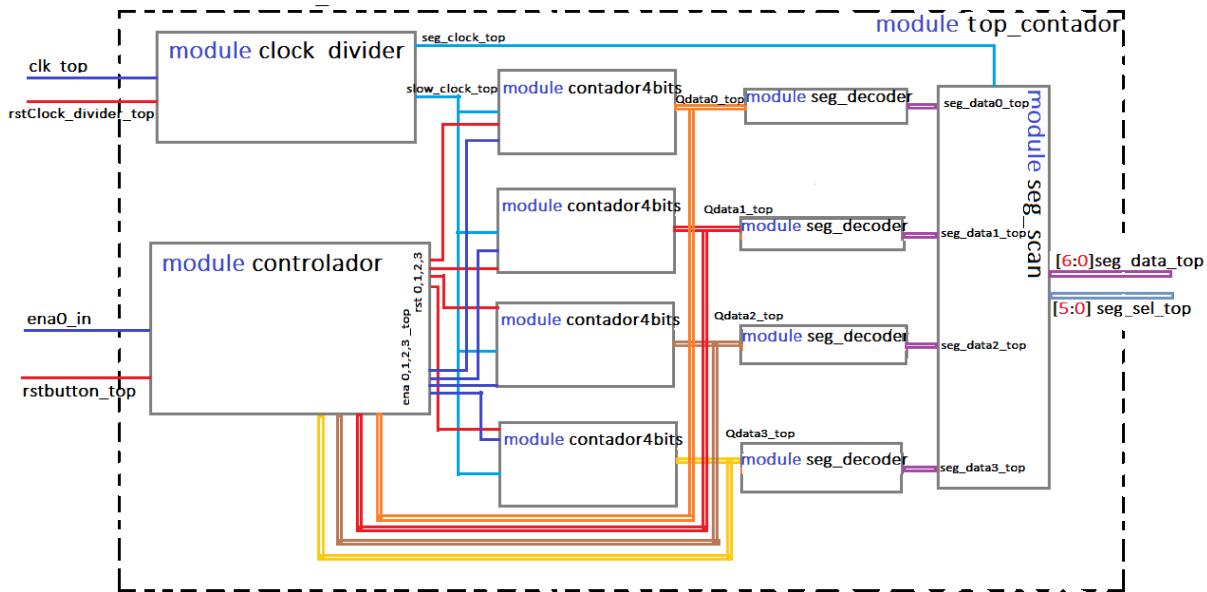


Figure 1: imágenes

#### 3.1 Diseño de módulos.

La función principal del divisor de reloj es retrasar la señal de oscilación del cristal del FPGA con el objetivo de realizar cambios a una frecuencia observable por los sentidos humanos, como entrada tendremos dos señales, la señal de un oscilador a 50MHz, y una señal de reset para inicializar los valores del divisor. De salida tendremos dos señales, con diferentes frecuencias, una para poder hacer los cambios en los contadores y otra que nos permitirá mostrar las salidas en los display 7 segmentos de la tarjeta.

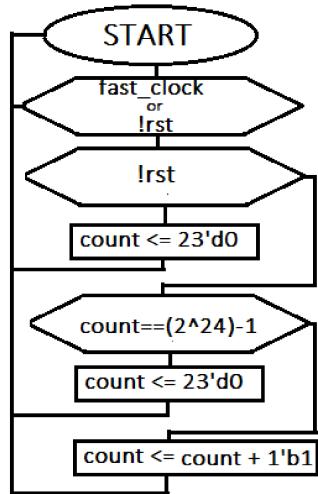


Figure 2: Diagrama de flujo para clock divider

Table 1: module clock\_divider

Señal	Dirección	Tamaño	Descripción
<i>fast_clock</i>	in	1	señal de reloj de 50 Mhz de la tarjeta FPGA
<i>rst</i>	in	1	señal para inicializar el contador colocandolo a cero
<i>slow_clock</i>	out	1	señal de reloj retrazada (2.98 Hz) para contador
<i>seg_clock</i>	out	1	señal de reloj retrazada (3333.3 Hz) para display

Modulo que permite describir el funcionamiento del controlador así como sus entradas y salidas, cumple la función de poder gestionar las señales de habilitación de los contadores dependiendo ciertas condiciones de entrada así como del valor alcanzado, será consciente de las salidas de los contadores para resetearlos y habilitar su conteo.

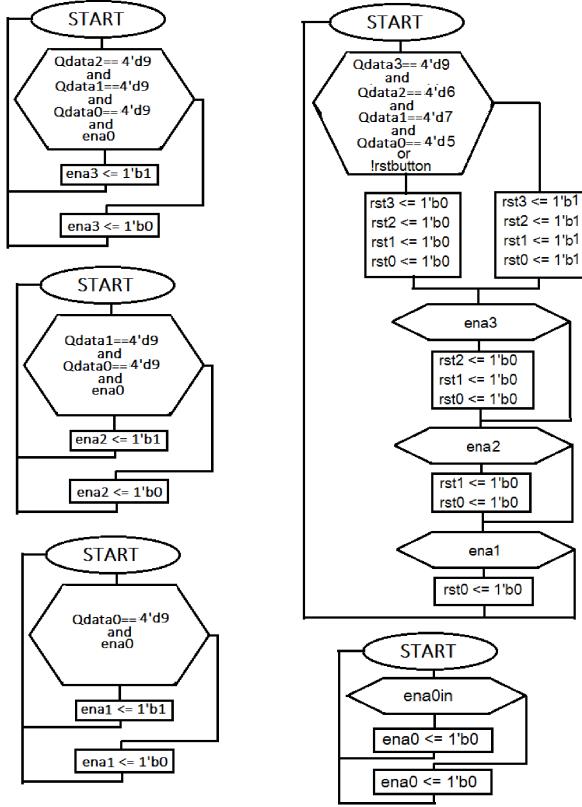


Figure 3: diagrama de flujo del módulo controlador

Table 2: module controlador

Señal	Dirección	Tamaño	Descripción
$Qdata3$	in	4	señal de control para resetear y habilitar cambio en contador 3
$Qdata2$	in	4	señal de control para resetear y habilitar cambio en contador 2
$Qdata1$	in	4	señal de control para resetear y habilitar cambio en contador 1
$Qdata0$	in	4	señal de control para resetear y habilitar cambio en contador 0
$rstbutton$	in	1	inicializar todo los contadores a su valor inicial
$ena0in$	in	1	aumentar el contador del bit menos significativo
$ena3$	out	1	habilita el conteo del contador 3
$ena2$	out	1	habilita el conteo del contador 2
$ena1$	out	1	habilita el conteo del contador 1
$ena0$	out	1	habilita el conteo del contador 0
$rst3$	out	1	resetear a su valor inicial el contador 3
$rst2$	out	1	resetear a su valor inicial el contador 2
$rst1$	out	1	resetear a su valor inicial el contador 1
$rst0$	out	1	resetear a su valor inicial el contador 0

Modulo que describe el funcionamiento de un contador de 4 bits para poder incrementar de 0-9, la entrada de reloj nos permite evaluar cada una de las salidas, una señal de reset (síncrona) para inicializar el conteo o colocar la salida a 0 cuando a complete un ciclo, su señal de habilitación obtenida del controlador, nos dará de salida una secuencia de bits que representara los números dentro del rango especificado.

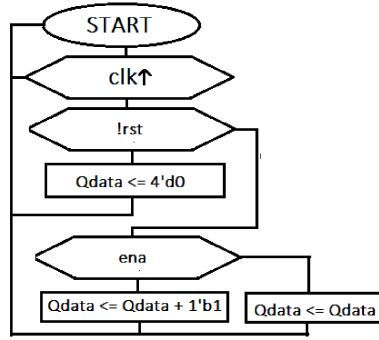


Figure 4: Diagrama de flujo para el modulo contador de 4 bits

Table 3: module contador4bits

Señal	Dirección	Tamaño	Descripción
<i>clk</i>	in	1	señal de reloj como flanko de reloj para evaluar señales
<i>rst</i>	in	1	señal que permite inicializar los valores del contador a 0
<i>ena</i>	in	1	señal que permite el aumento de la cuenta en 1 bit
<i>Qdata</i>	out	4	señal que representara la cuenta en binario

La función principal de este módulo ( *tabla 4* ) es convertir el código binario de 4 bits en una señal de 7 bits para poder habilitar los display de 7 segmentos de la tarjeta.

Table 4: module seg\_decoder

<b>Señal</b>	<b>Dirección</b>	<b>Tamaño</b>	<b>Descripción</b>
<i>bin_data</i>	in	4	recibe un numero en binario del contador conectado
<i>seg_data</i>	out	7	señal que representa el encendido de los segementos

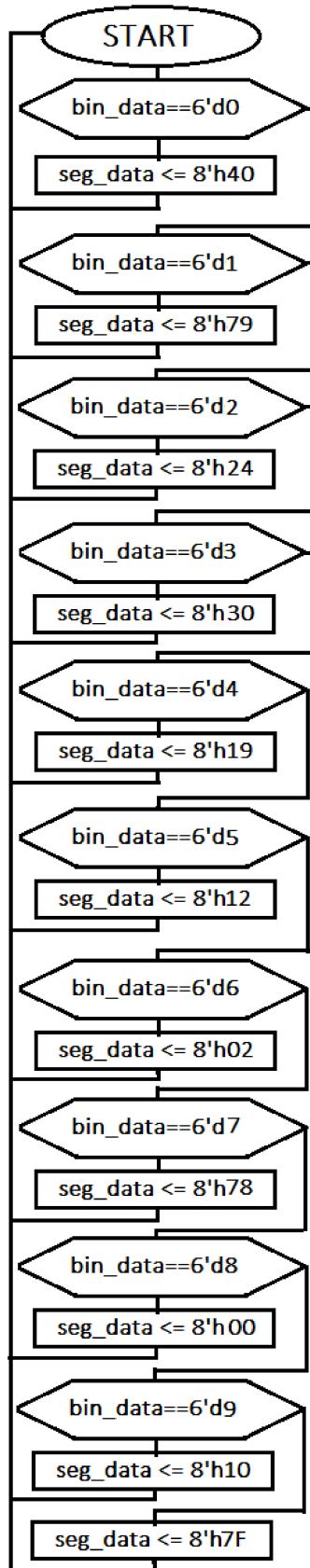


Figure 5: Diagrama de flujo del modulo decodificar para display 7 segmentos  
<sub>7</sub>

Módulo que permite transmitir el control de señales de los segmentos en los múltiples display de salida. Para poder utilizarlos es necesario que se encienda un display a la vez a una alta frecuencia con el objetivo de que sea imperceptible este cambio, dentro de las limitaciones de la vista humana todos los display estarán encendidos al mismo tiempo. Para esto es necesario altas frecuencias obtenidas del divisor de reloj utilizando el cambio del bit 14 de 24.

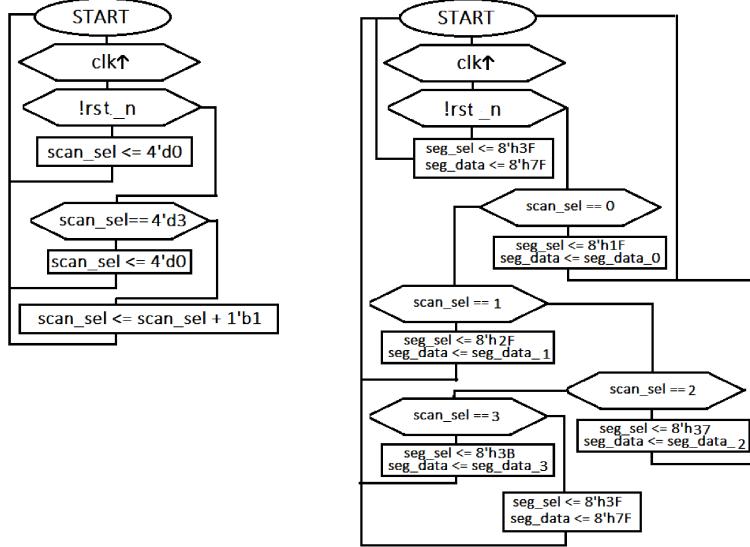


Figure 6: Diagrama de flujo del modulo para desplegar los modulos en los display

Table 5: module seg\_scan

Señal	Dirección	Tamaño	Descripción
clk	in	1	señal de reloj para iterar sobre cada bloque de 7 segmentos
rest_n	in	1	señal para inicializar el conteo e iteración
seg_data_3	in	1	representa la señal de 7 bits (display) del contador 3
seg_data_2	in	1	representa la señal de 7 bits (display) del contador 2
seg_data_1	in	1	representa la señal de 7 bits (display) del contador 1
seg_data_0	in	1	representa la señal de 7 bits (display) del contador 0
seg_sel	out	6	señal indicadora del display a habilitar (1 de 4)
seg_data	out	7	señal con información de que segmentos encender

## 4 Resultados simulados.

A continuación, se mostrarán los resultados obtenidos de la simulación utilizando el software de ModelSim - Altera. Es importante recalcar que se hicieron un par de cambios en el código del divisor de reloj para que sean apreciables las señales en el simulador, el divisor de reloj paso de utilizar el 24 bit a utilizar 3 bits, y la señal de display a utilizar el bit 0 en lugar del bit 14.

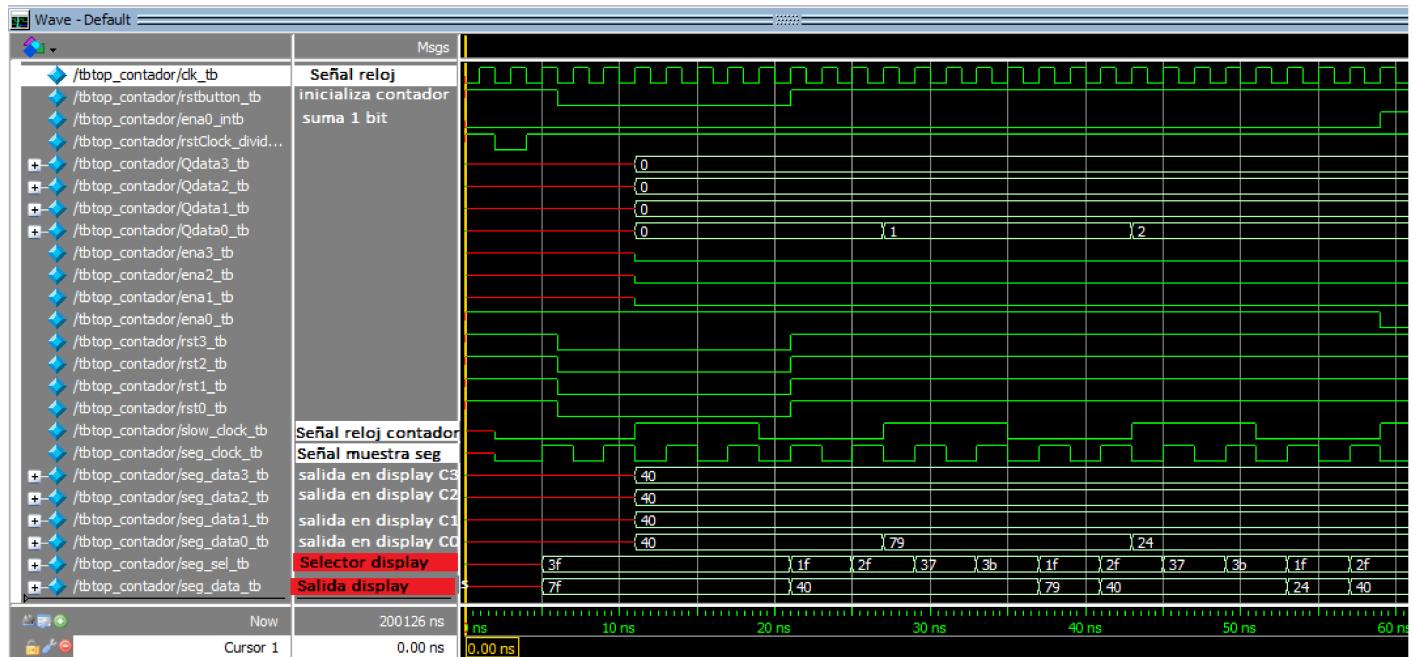


Figure 7: inicio de la simulación

Para efectos explicativos los valores en la simulación de las señales conectadas al display se colocaron en formato hexadecimal, así como la señal de selección de los display. Consultar tablas 6 y 7

Table 6: valores del selector de display

Selector binario	selector hexadecimal	selección
6'b111_111	8'h3F	N/A
6'b011_111	8'h1F	Display 1
6'b101_111	8'h2F	Display 2
6'b110_111	8'h37	Display 3
6'b111_011	8'h3B	Display 4

Codificación para las distintas salidas considerando los segmentos del display.

Table 7: codificacion para activacion de segmentos

Selector binario	selector hexadecimal	selección
7'b111_1111	8'h7F	OFF
7'b100_0000	8'h40	0
7'b111_1001	8'h79	1
7'b010_0100	8'h24	2
7'b011_0000	8'h30	3
7'b001_1001	8'h19	4
7'b001_0010	8'h12	5
7'b000_0010	8'h02	6
7'b111_1000	8'h78	7
7'b000_0000	8'h00	8
7'b001_0000	8'h10	9

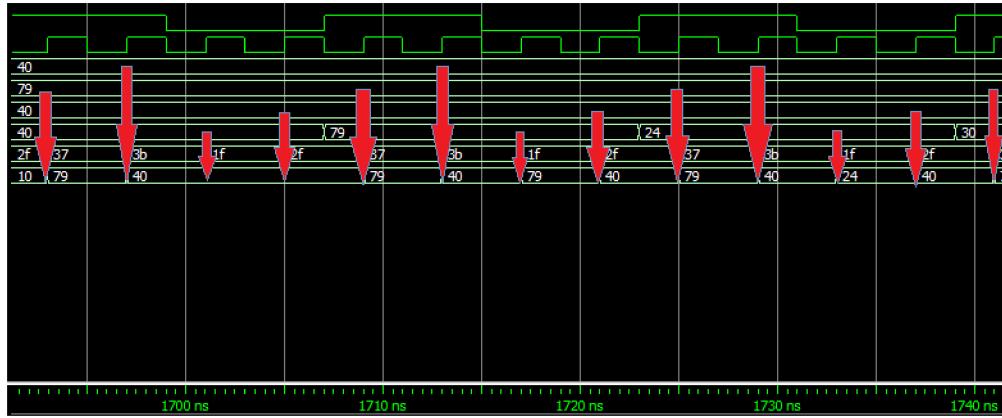


Figure 8: Señal selectora de display

Se puede apreciar en la imagen (figura 8) como se inicializan tanto el divisor de reloj tanto sus dos señales y como se inicializa el contador de 4 bits, el objeto de mostrar cómo se comportan las salidas del **selector de display** y **salida del display** es dar a entender cómo se pueden mostrar en los 4 display del FPGA, solo se puede imprimir en un display así que se necesita colocar una señal extra para seleccionar en qué display mostrar el número, esto se logra utilizando frecuencias grandes, para que a simple vista todos los display estén encendidos cuando se enciende uno, se apaga y se enciende el siguiente.

Se puede apreciar (figura 9) como la transición de los dígitos se hace de manera esperada, sin estados intermedios indeseados, el contador respeta la enumeración correcta de términos sucesivos al sumar 1 bit.

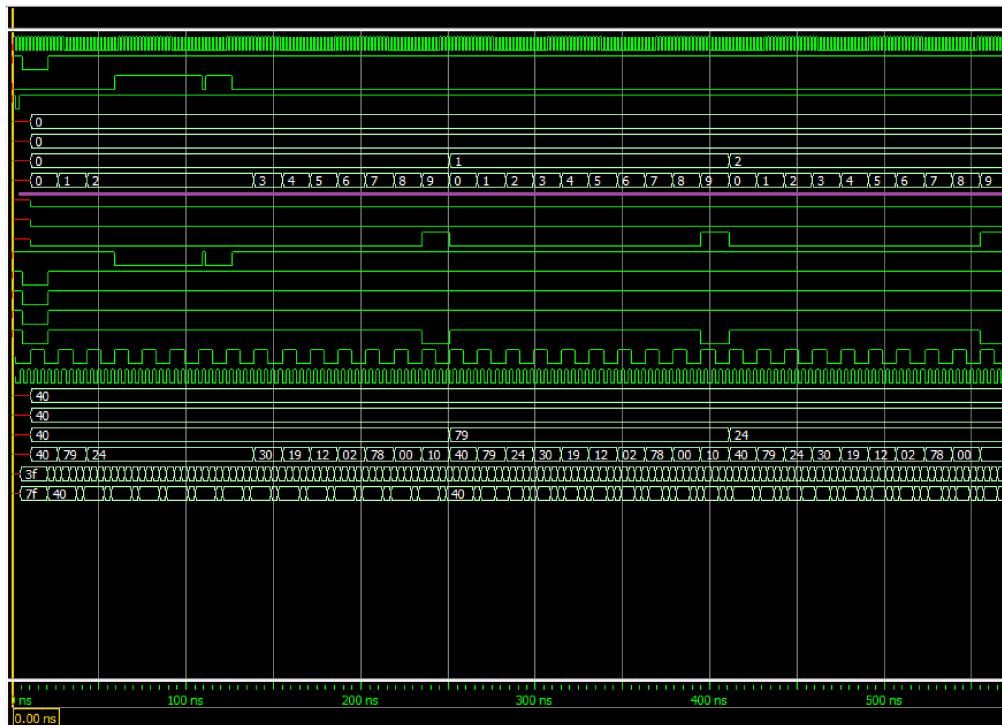
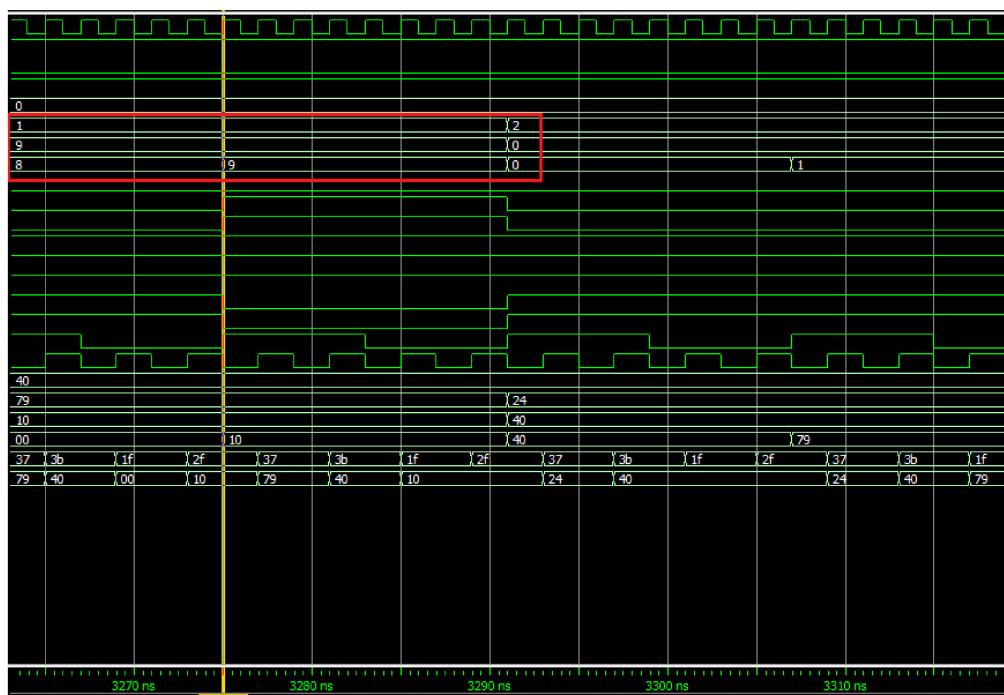
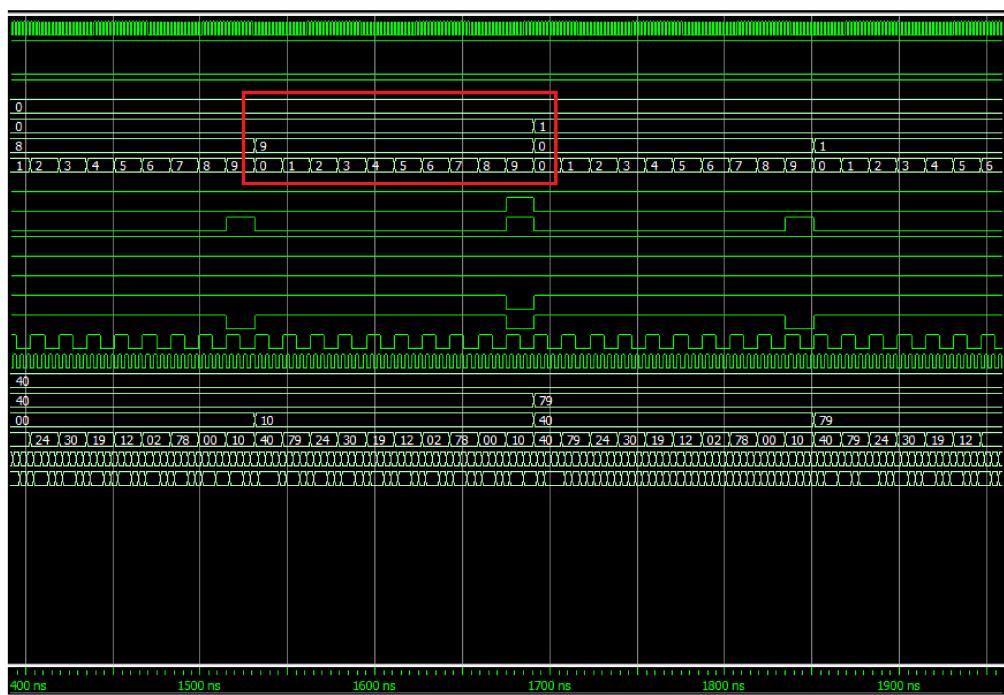


Figure 9: Conteo de los primeros numeros

#### 4.1 Condiciones de cambio correctas

A continuación, se mostrar como las transiciones al dar un ciclo completo uno de los bits, este cambio respeta los requerimientos del proyecto.

Cumplimiento de la condición de reinicio del contador al llegar al numero 9675.



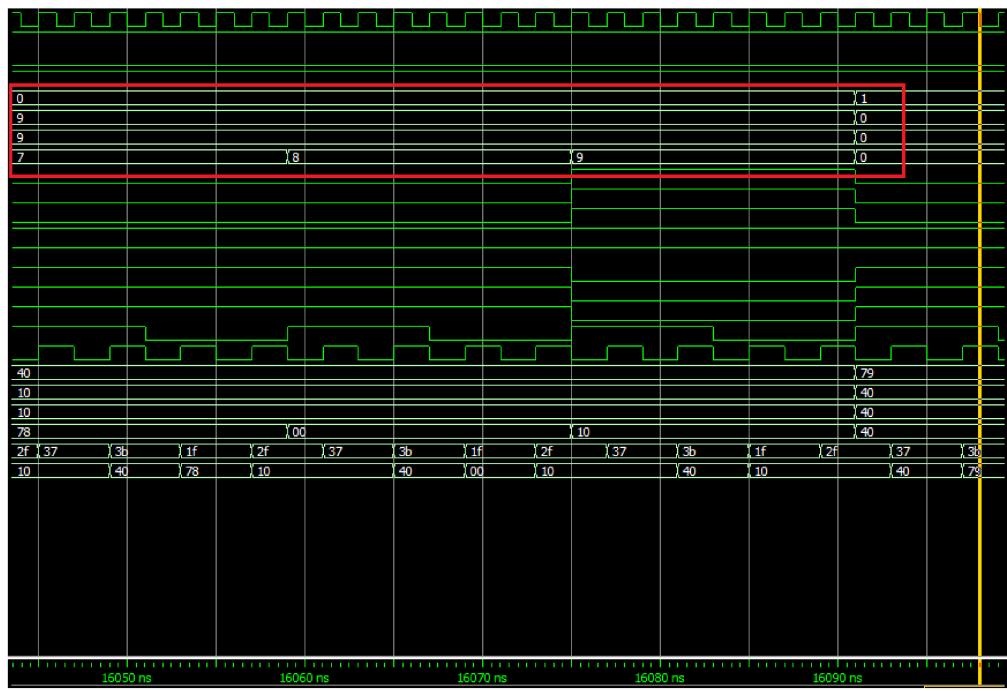


Figure 12: correcta transición de 999 a 1000

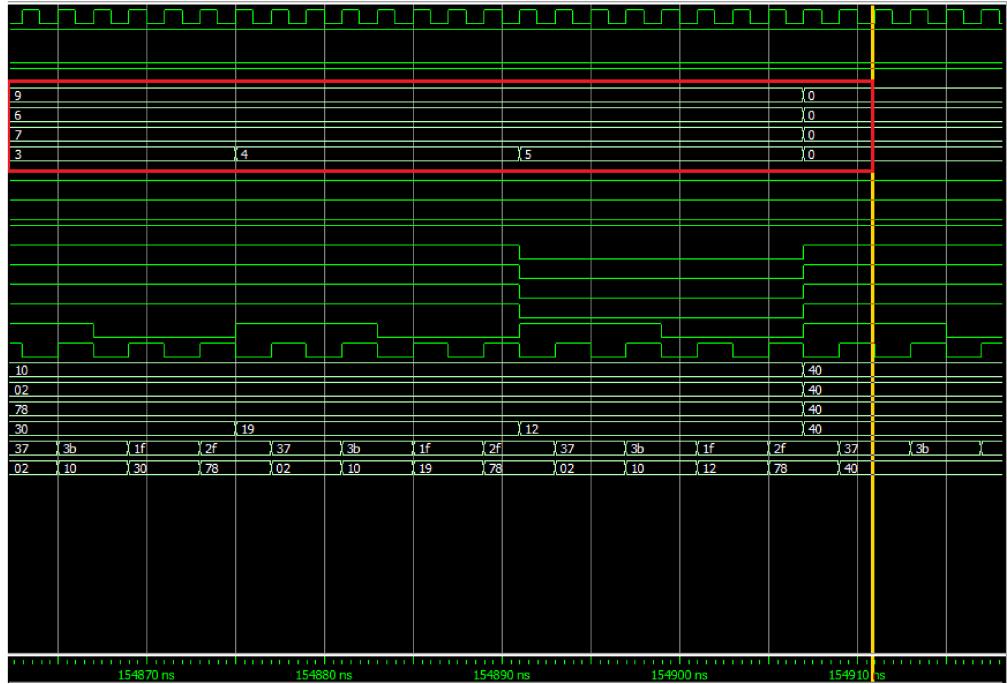


Figure 13: correcto reinicio del contador al llegar al numero 9675

video demostrativo: (<https://drive.google.com/drive/folders/1Ql4Qg8HUKSYrCj1CsZl6MhGdvE0TpKE7?usp=sharing>)

## 5 Resultados experimentales.

---

A continuación se mostraran los resultados experimentales sobre la tarjeta FGPA, mostrando una funcionalidad exitosa.

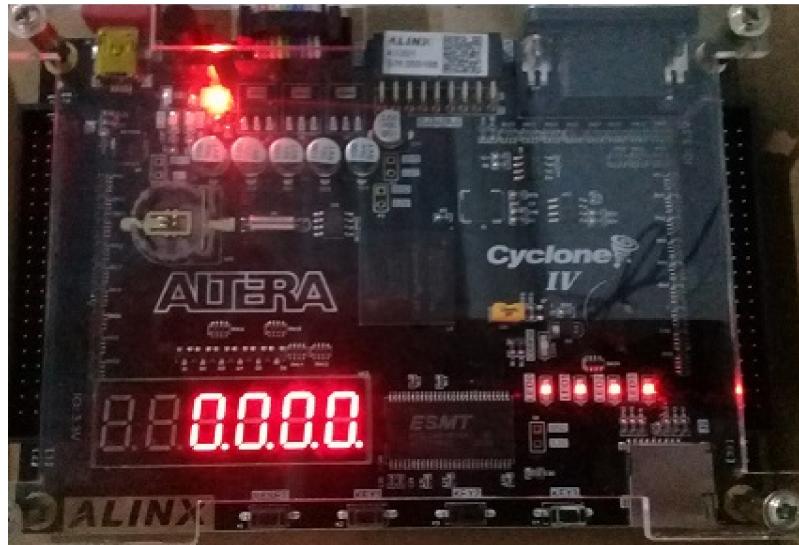


Figure 14: displays del FPGA mostrando estado inicial

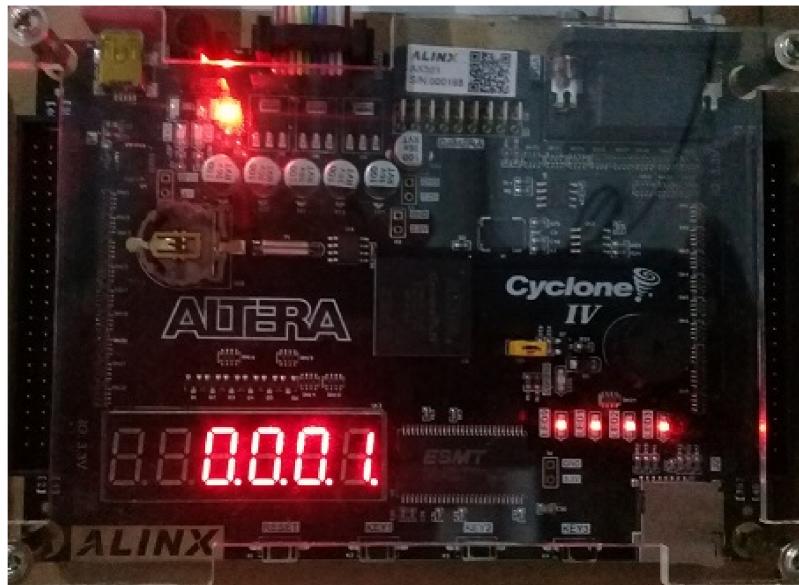


Figure 15: Display del FPGA mostrando el primer estado del conteo

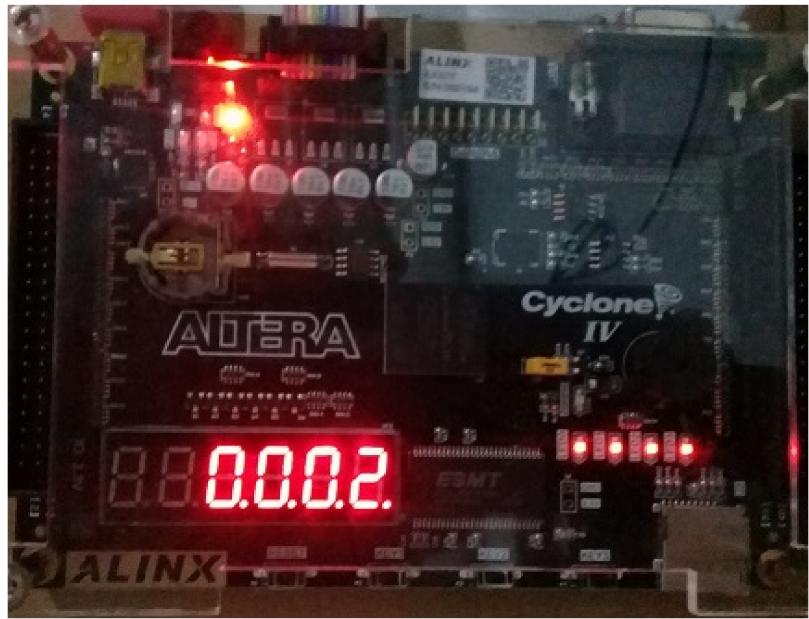


Figure 16: Display del FPGA mostrando el segundo estado del conteo

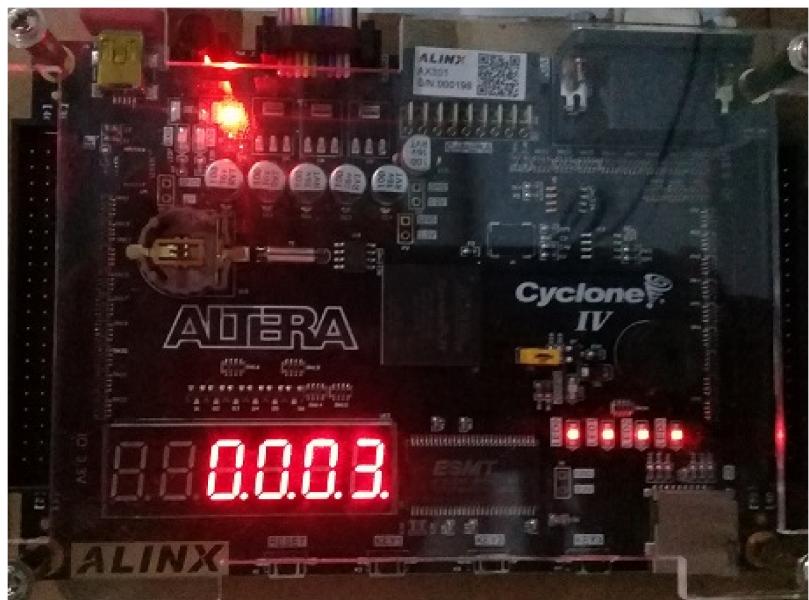


Figure 17: Display del FPGA mostrando el tercer estado del conteo

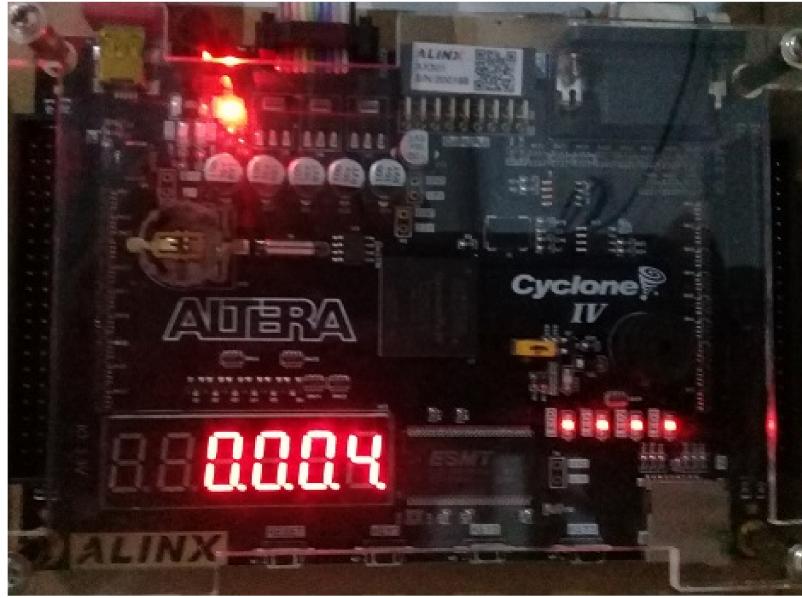


Figure 18: Display del FPGA mostrando el cuarto estado del conteo



Figure 19: Display del FPGA mostrando el quinto estado del conteo

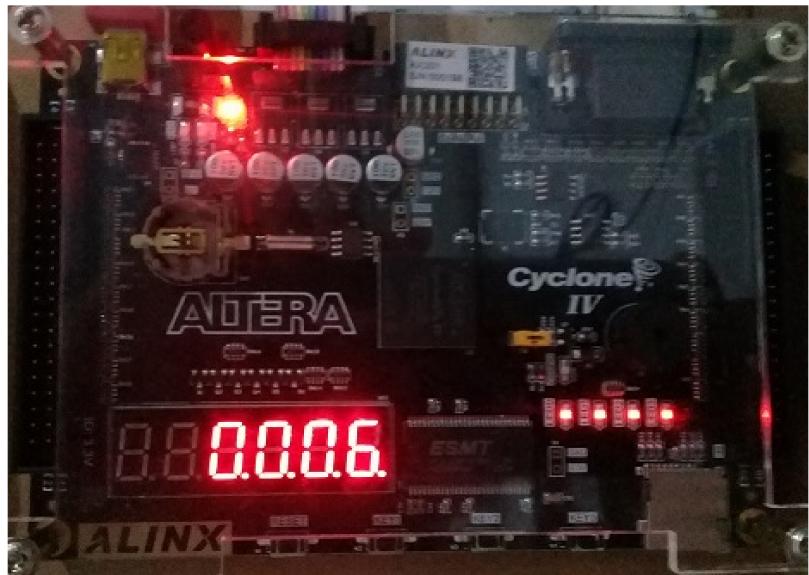


Figure 20: Display del FPGA mostrando el sexto estado del conteo

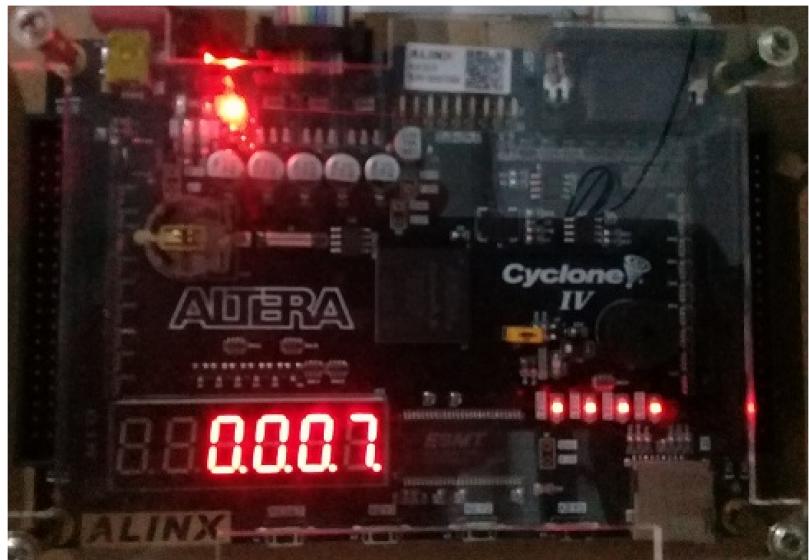


Figure 21: Display del FPGA mostrando el septimo estado del conteo

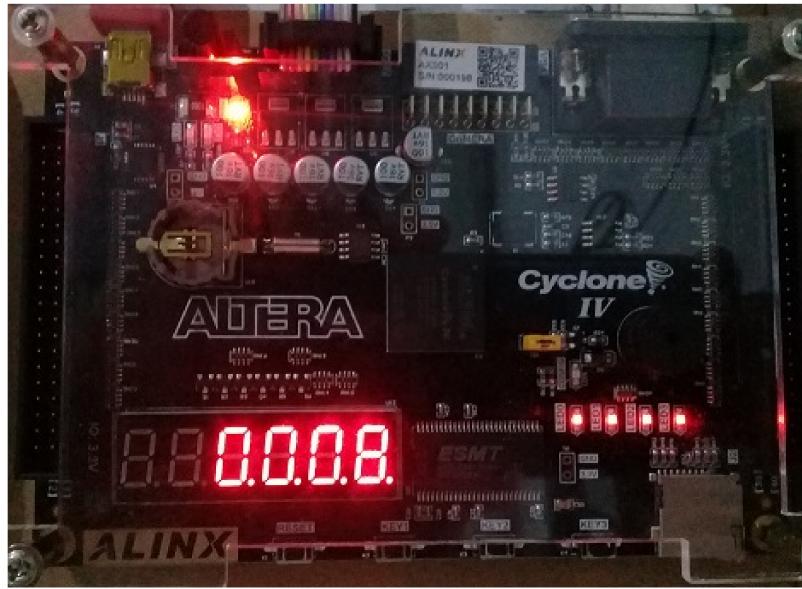


Figure 22: Display del FPGA mostrando el octavo estado del conteo

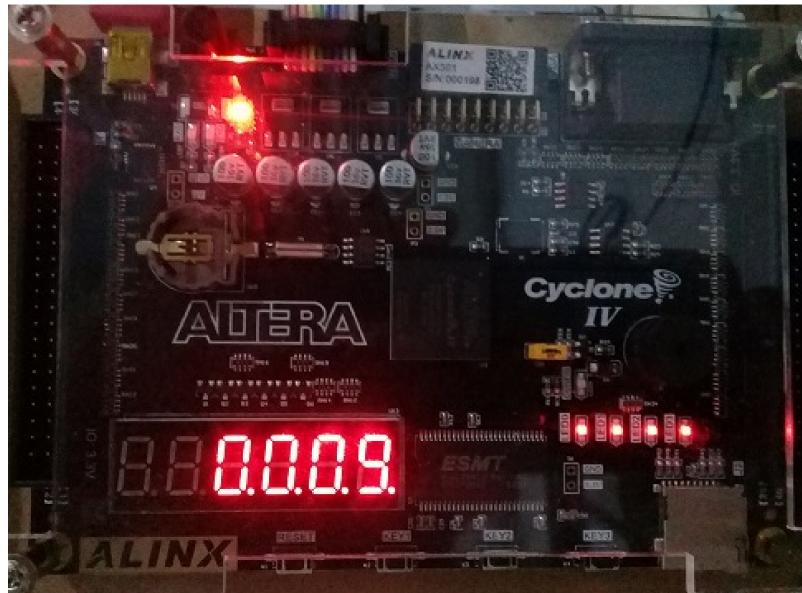


Figure 23: Display del FPGA mostrando el noveno estado del conteo

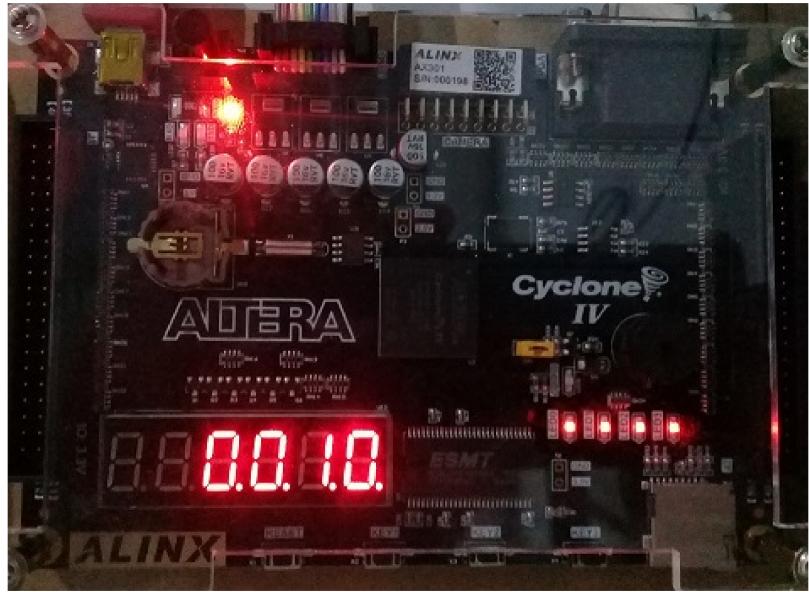


Figure 24: Display del FPGA mostrando el decimo estado del conteo

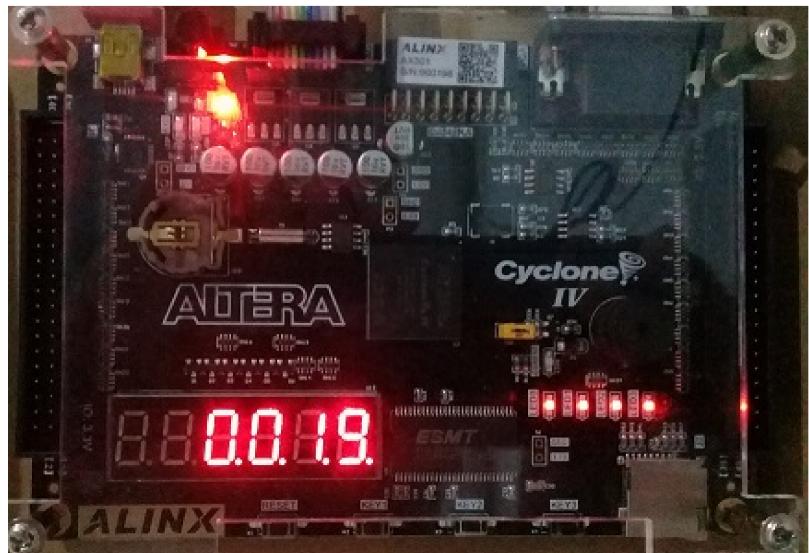


Figure 25: Display del FPGA mostrando el estado diecinueve del conteo

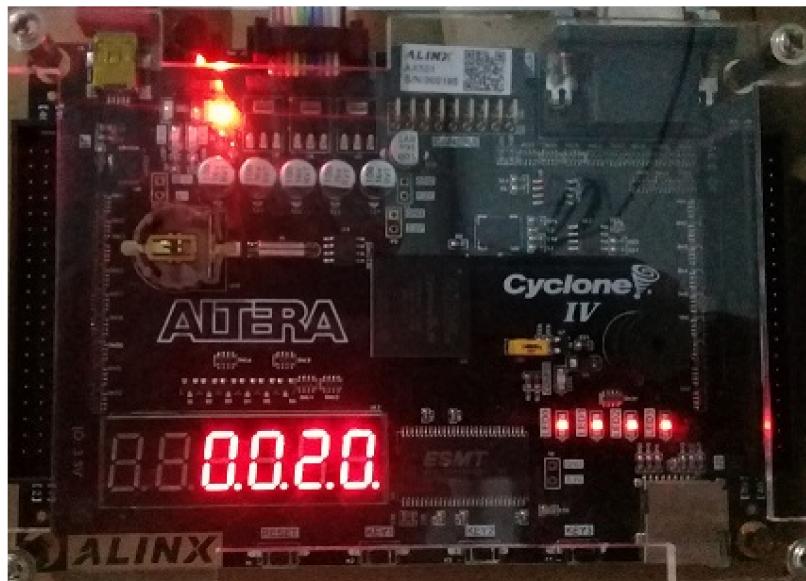


Figure 26: Display del FPGA mostrando el estado veinte del conteo