

# Hw #08 { Contadores }

## Lenguajes de descripción de hardware

Hernández Castellanos César Uriel

### 1 Sección uno

La siguiente figura nos muestra un contador sincrono de 4 bits en flip-flops tipo T:

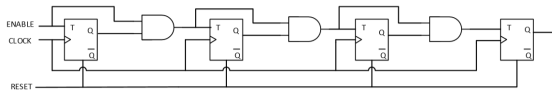


Figure 1: Contador de 4 bits

Este contador incrementa su cuenta en 1 cada flanco de activación si la señal enable es igual a 1. Recordar que existen contadores síncronos y asíncronos, ¿Cuál es la diferencia? Los contadores son muy útiles para contar pulsos, dividir frecuencia, generar señales de reloj, temporizadores y frecuencímetros entre otras aplicaciones.

1.- Escribir un código en lenguaje VHDL o verilog basado en el diagrama de la figura anterior para un contador de 16 bits. ¿Cuántos elementos lógicos y que frecuencia máxima de operación reporta la compilación?

El siguiente código muestra la implementación de un contador de N bits.

```
Array0(0) <= ENABLE;
FFTCycle: FOR I IN 0 TO N-1 GENERATE
    Array0(I+1) <= Array1(I) and Array0(I);
    FFTi: FFT
    PORT MAP (
        T => Array0(I),
        CLK => CLK,
        RESET => RESET,
        Q => Array1(I)
    );
    Q(I) <= Array1(I);
END GENERATE;
```

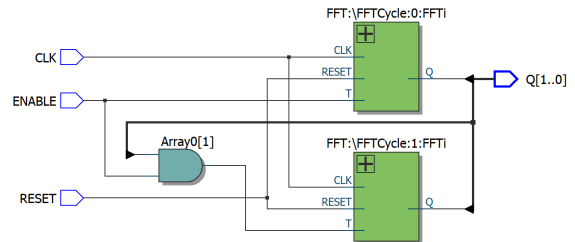


Figure 2: Contador de 2 bits

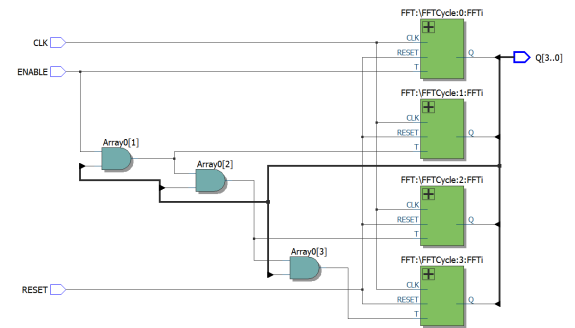


Figure 3: Contador de 4 bits

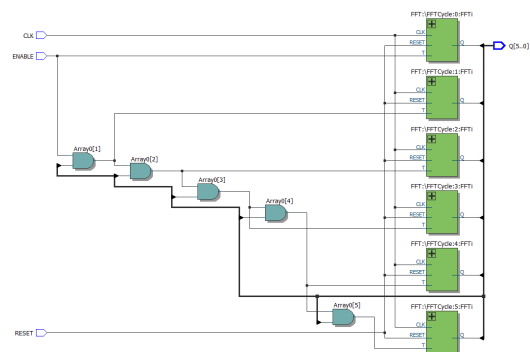


Figure 4: Contador de 6 bits

Número de elementos lógicos para un contador de 16 bits.

Total logic elements	22 / 6,272 ( < 1 % )
Total combinational functions	22 / 6,272 ( < 1 % )
Dedicated logic registers	16 / 6,272 ( < 1 % )
Total registers	16
Total pins	19 / 92 ( 21 % )
Total virtual pins	0
Total memory bits	0 / 276,480 ( 0 % )
Embedded Multiplier 9-bit elements	0 / 30 ( 0 % )
Total PLLs	0 / 2 ( 0 % )

Frecuencia máxima del contador de 16 bits.

	Fmax	Restricted Fmax	Clock Name
1	320.0 MHz	250.0 MHz	CLK

Usar el visor RTL y comentar si el resultado de la síntesis difiere de la figura anterior.

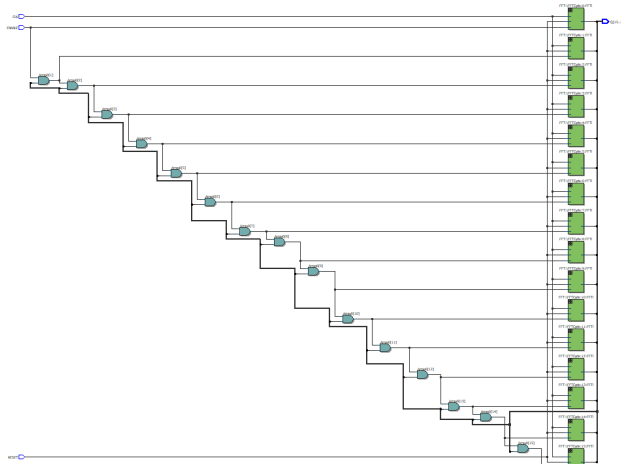


Figure 5: Contador de 16 bits

En la figura anterior es posible observar el resultado del visor RTL, para un contador de 16 bits, lo que es una extensión del contador de 4 bits que proporcionó inicialmente, solamente que con más compuertas y flip flops.

## 2 Sección dos

Realizar un contador hexadecimal, usando el contador anterior.

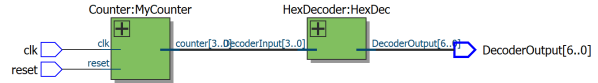


Figure 6: Contador hexadecimal

```
signal counter_up: std_logic_vector(3 downto 0);
```

```
MyCounter : Counter
PORT MAP (
    clk => clk,
    reset => reset,
    counter => counter_up
);
```

```
HexDec: HexDecoder
PORT MAP (
    DecoderInput => counter_up,
    DecoderOutput => Output
);
```

## 3 Sección tres

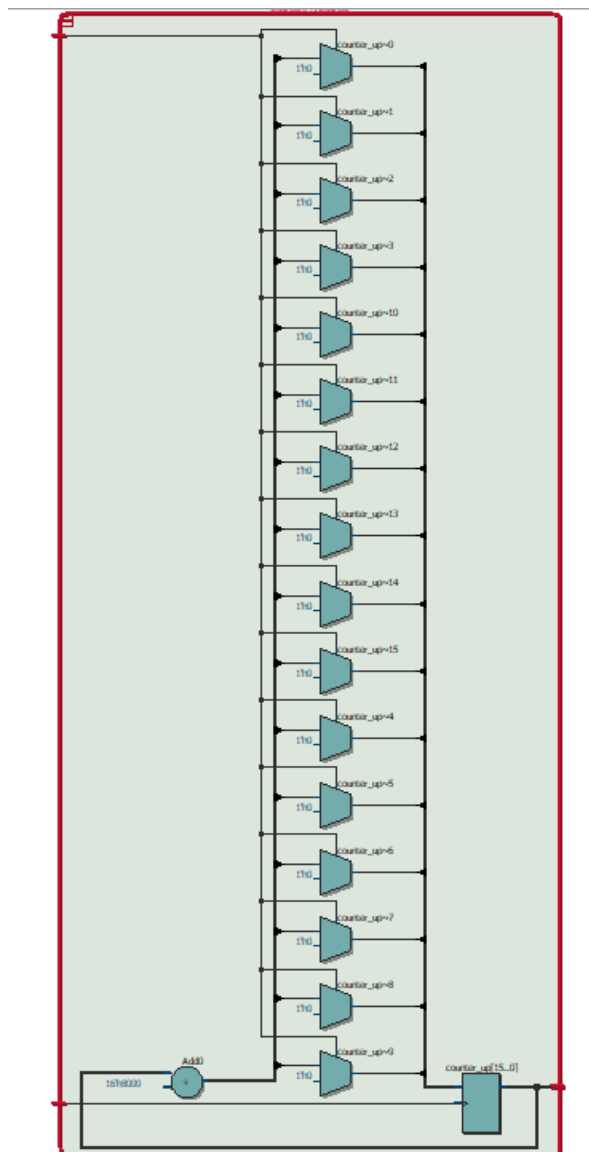
Repetir el inciso 1 pero usar descripción por comportamiento. ¿Cuántos elementos lógicos y que frecuencia máxima de operación reporta la compilación? En descripción por comportamiento debe utilizarse una línea como la siguiente:

$$Q \leq Q + 1;$$

A continuación se muestra la implementación de un contador de 16 bits por comportamiento.

```
process (clk, reset)
begin
    if (rising_edge(clk)) then
        if (reset='1') then
            counter_up <= x"0";
        else
            counter_up <= counter_up + x"1";
        end if;
    end if;
end process;
counter <= counter_up;
```

Usar el visor RTL, ¿Qué diferencias en el resultado de la síntesis se observan respecto al resultado del inciso 1?



**Figure 7:** *Contador de 16 bits*

La figura anterior muestra el resultado del visor RTL con un contador de 16 bits, descrito por comportamiento.

Es posible ver que se implementa una serie de multiplexores que tienen como señal de control la señal reset, además de un sumador que incrementa en uno nuestro contador.

Finalmente es posible observar que se implementa un flip flop tipo D.

En la figura siguiente es posible observar el número de elementos lógicos del contador descrito por comportamiento, lo que es un número menor de elementos lógicos.

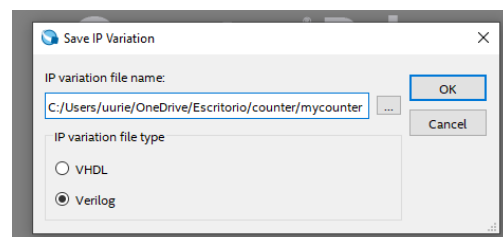
Flow Status	Successful - Fri May 15 22:59:01 2020
Quartus II 64-Bit Version	13.1.0 Build 162 10/23/2013 SJ Web Edition
Revision Name	HexCounter
Top-level Entity Name	HexCounter
Family	Cyclone IV GX
Total logic elements	5 / 14,400 (< 1 %)
Total combinational functions	5 / 14,400 (< 1 %)
Dedicated logic registers	4 / 14,400 (< 1 %)
Total registers	4
Total pins	6 / 81 (7 %)
Total virtual pins	0
Total memory bits	0 / 552,960 (0 %)
Embedded Multiplier 9-bit elements	0
Total GXB Receiver Channel PCS	0 / 2 (0 %)
Total GXB Receiver Channel PMA	0 / 2 (0 %)
Total GXB Transmitter Channel PCS	0 / 2 (0 %)
Total GXB Transmitter Channel PMA	0 / 2 (0 %)
Total PLLs	0 / 3 (0 %)
Device	EP4CGX15BF14C6
Timing Models	Final

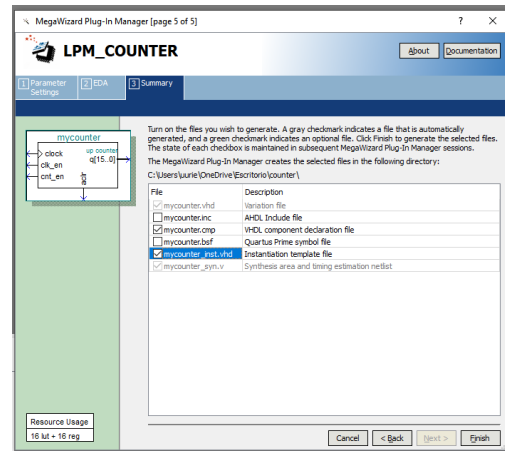
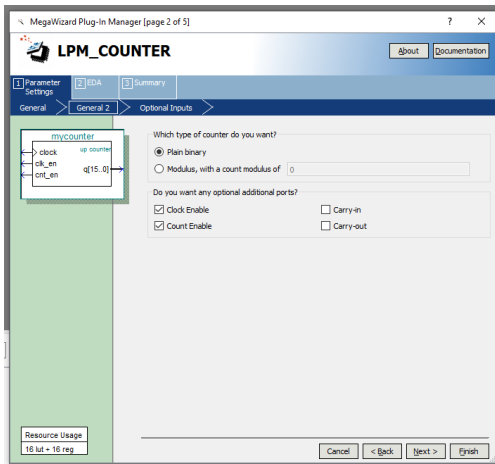
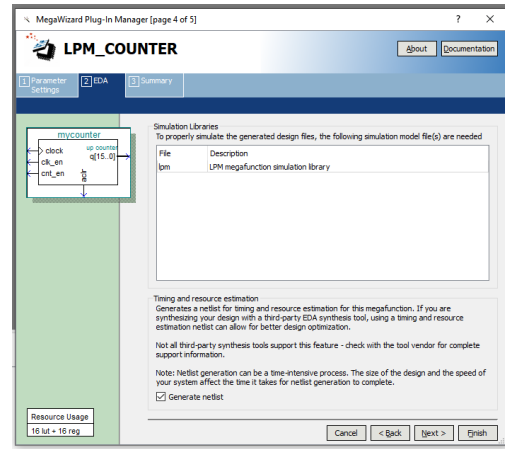
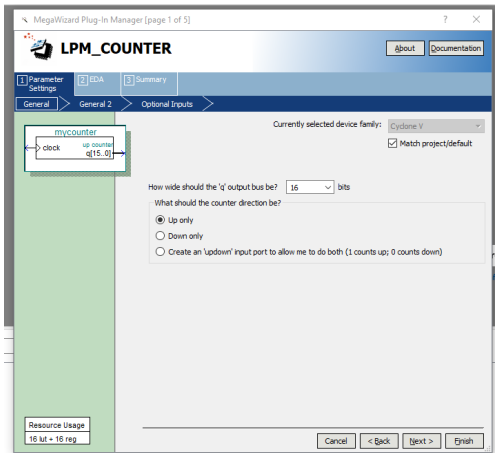
Finalmente vemos la frecuencia máxima del circuito contador descrito por comportamiento.

	Fmax	Restricted Fmax	Clock Name
1	837.52 MHz	250.0 MHz	clk

## 4 Sección cuatro

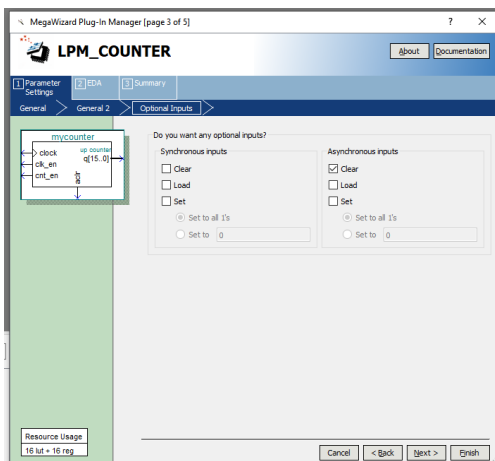
Investigar como describir un contador de 16 bits basado en un módulo LPM Library of Parameterized Modules. Usar el módulo LPMCOUNTER. En las versiones más recientes de Quartus el módulo se obtiene en la sección IP core. Elegir opciones para ser consistentes con los dos incisos anteriores (nivel del enable y reset asíncrono).





## 4.1 Contador generado

```
q<= sub_wire0(15 DOWNT0 0);
LPM_COUNTER_component : LPM_COUNTER
GENERIC MAP (
    lpm_direction => "UP",
    lpm_port_updown => "PORT_UNUSED",
    lpm_type => "LPM_COUNTER",
    lpm_width => 16
)
PORT MAP (
    aclr=>aclr,
    clk_en=>clk_en,
    clock=>clock,
    cnt_en=>cnt_en,
    q=>sub_wire0
);
```



## 5 Sección cinco

Usar el reloj de 50MHz que tiene la tarjeta DE2 para alimentar a un contador que permita encender y apagar un dígito hexadecimal (codificado con 4 interruptores: sw0 a sw3, y mostrado en un display de 7 segmentos ) a razón de aproximadamente 1 segundo (medio segundo encendido y medio segundo apagado). ¿De cuántos bits debe ser el contador?

**R:** Únicamente se necesitaría un contador de 1 bit, ya que los estados posibles del dígito hexadecimal son dos (prendido y apagado)

## 6 Sección seis

Describir un circuito que recorra un mensaje como en la práctica número 1 pero que rotará el mensaje en forma automática a razón de aproximadamente una posición por segundo (usar el contador del inciso 4 para indicar la posición)

### 6.1 Implementación en VHDL

```
mycounter_inst : counter PORT MAP (  
    aclr=>reset,  
    clock=>clk,  
    q=>MarqueePosition  
);  
  
CM : CombinationalMarquee  
PORT MAP (  
    C0 => C0,  
    C1 => C1,  
    C2 => C2,  
    C3 => C3,  
    C4 => C4,  
    MarqueePosition => MarqueePosition,  
    D0 => D0,  
    D1 => D1,  
    D2 => D2,  
    D3 => D3,  
    D4 => D4,  
    D5 => D5,  
    D6 => D6,  
    D7 => D7  
);
```

### 6.2 Resultado del visor RTL

