

# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO



## FACULTAD DE INGENIERÍA

### Diseño de Sistemas Digitales

Práctica #6: Diseño de un Divisor de Frecuencia

Alumno: Jesús Alfredo Hernández Alarcón

Profesor: M.I. Norma Elva Chávez Rodríguez

SEMESTRE 2014-1



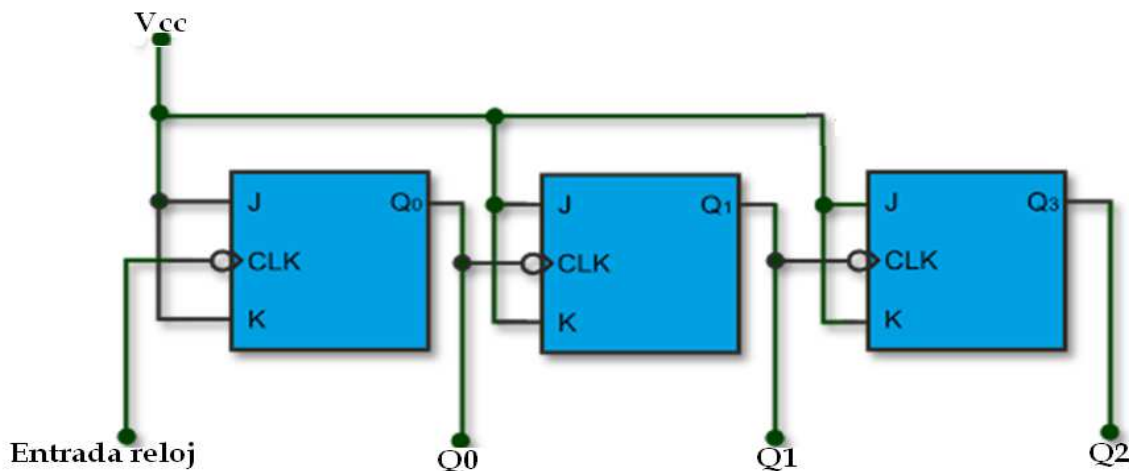
**OBJETIVO:**

El alumno aprenderá a diseñar divisores de frecuencia, con el fin de visualizar cualquier sistema secuencial en la tarjeta de desarrollo Spartan 3 la cual tiene un reloj de cristal de 50 Mhz, frecuencia muy rápida para ser detectada por el ojo humano.

Se llama divisor de frecuencia a un dispositivo que produce a su salida una frecuencia menor que la de entrada y suelen estar formados por contadores digitales. Un contador es un circuito secuencial construido a partir de biestables y compuertas lógicas.

**INTRODUCCIÓN:**

Si a un Flip-Flop JK, todas sus entradas se conectan a Vcc, excepto la entrada del reloj, a su salida, dicho Flip-Flop fluctúa entre cero y uno cada vez que el flanco activo del reloj se presente y si por ejemplo se interconectan tres Flip-Flops como se muestra en la figura 1, lo que sucede es que la salida de los tres Flip-Flops, fluctúa en diferentes tiempos, ya que cada uno tiene diferente entrada de reloj. La figura 2 muestra el diagrama de tiempos de sus salidas, suponiendo que el flanco positivo del reloj es el que activa a los Flip-Flops.



**Figura 1** Flip-Flop's interconectados mediante la unión de la salida de uno a la entrada de reloj del siguiente

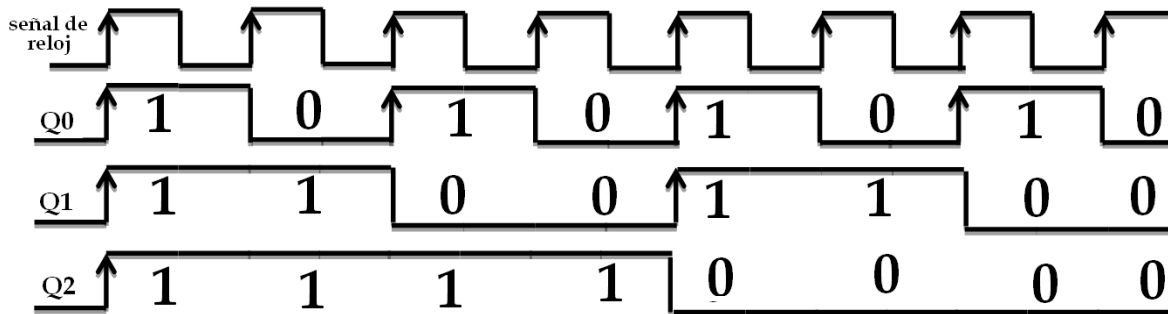


Figura 2 DIAGRAMA DE TIEMPOS DE LOS FLIP-FLOP'S INTERCONECTADOS:

Observando la figura 2 y poniendo leds a las salidas Q2 Q1 Q0 (en ese orden) se tiene un contador descendente, que cuenta del 7 al cero. También se observa que si la salida Q2, se selecciona como entrada de reloj para cualquier sistema digital, se obtuvo un divisor de frecuencia respecto a la entrada del reloj original.

#### ESPECIFICACIONES:

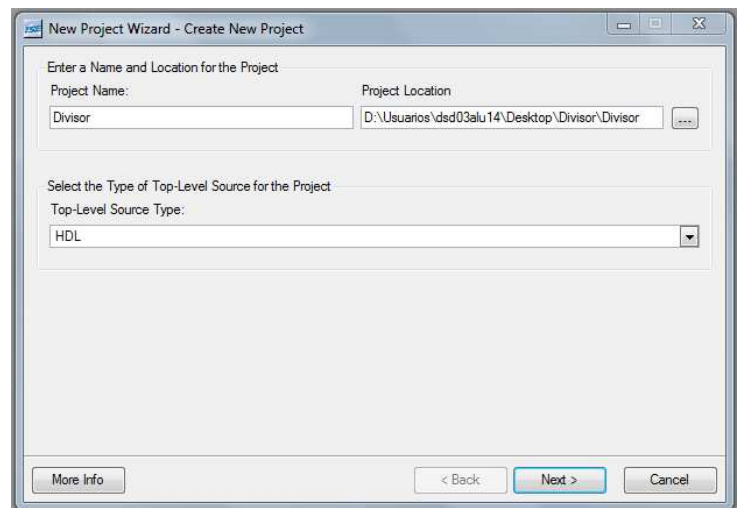
Diseñar un contador de 28 bits de salida, del cual únicamente se tomará en cuenta la salida 26 para utilizarla como el reloj de cualquier diseño secuencial implementado en la tarjeta spartan 3. Esto hará que la entrada del reloj de 50 Mhz tenga un retraso de  $2^{26}$  bits, dando un valor en hexadecimal de 48009E0 asegurando con esto un tiempo de 1.51 segundos entre cada pulso del reloj.

Notamos que hemos generado un contador binario capaz de contar hasta 1111 o 16 en decimal y además que sigue un ciclo repetitivo.

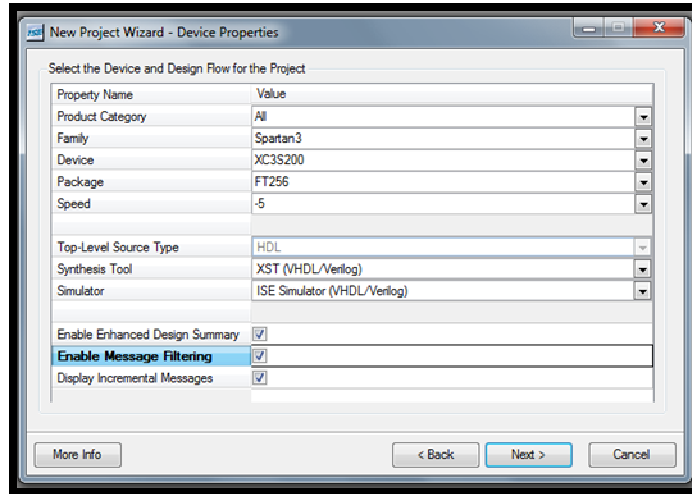
### Procedimiento:

Para el diseño vamos a hacer uso del de la plataforma ISE de Xilinx y la tarjeta SPARTAN 3. Vamos a crear una carpeta llamada "Divisor", en la cual se estará trabajando para este proyecto.

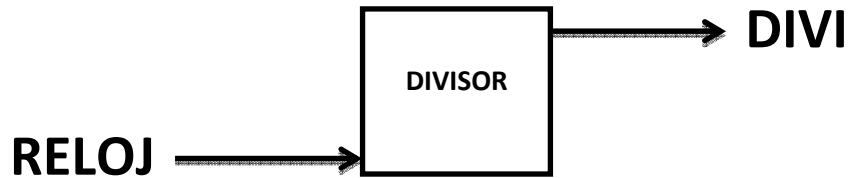
Abrimos la plataforma ISE y nos dirigimos a crear un nuevo proyecto. Lo guardaremos con el nombre "Divisor" en la misma carpeta que se creó previamente.



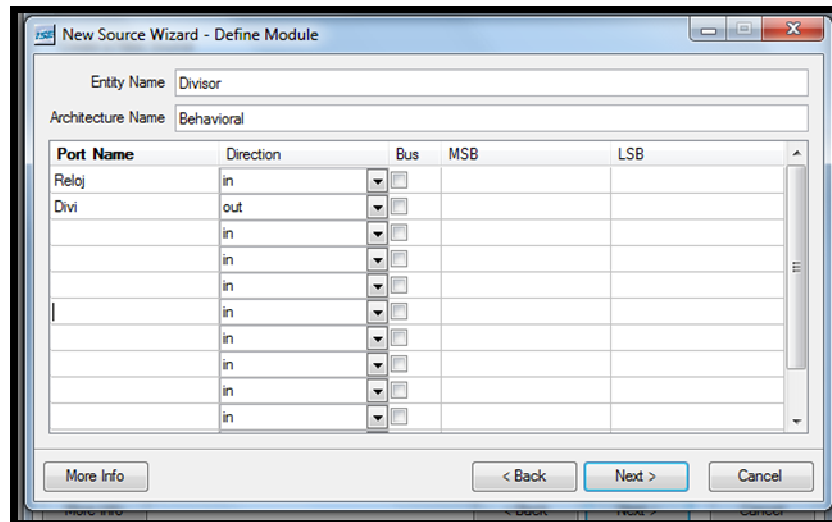
Usamos la siguiente configuración para el correcto funcionamiento:



El diseño que vamos a emplear es el siguiente, haciendo uso de una entrada y una salida. La entrada la llamaremos **RELOJ** y la salida la llamaremos **DIVI**.



En la plataforma vamos a seleccionar como entrada (in) a nuestra variable “Reloj”, y como salida (out) a nuestra variable “Divi”.



Una vez asignadas la entrada y la salida, vamos a proceder con el código, en donde se crea una variable interna, esta variable únicamente funcionará dentro de un process, dicha variable será nombrada como **“cuenta”**.

La salida 26 del contador de 28 bits será la única que se tomará en cuenta, esta será utilizada como el reloj.

Por consecuente, la entrada del reloj de 50Mhz tendrá un retraso de  $2^{26}$  bits (la variable **“cuenta”** será un contador, el cual cuando llegue al valor límite (1.51segundos), vuelve a empezar la cuenta.

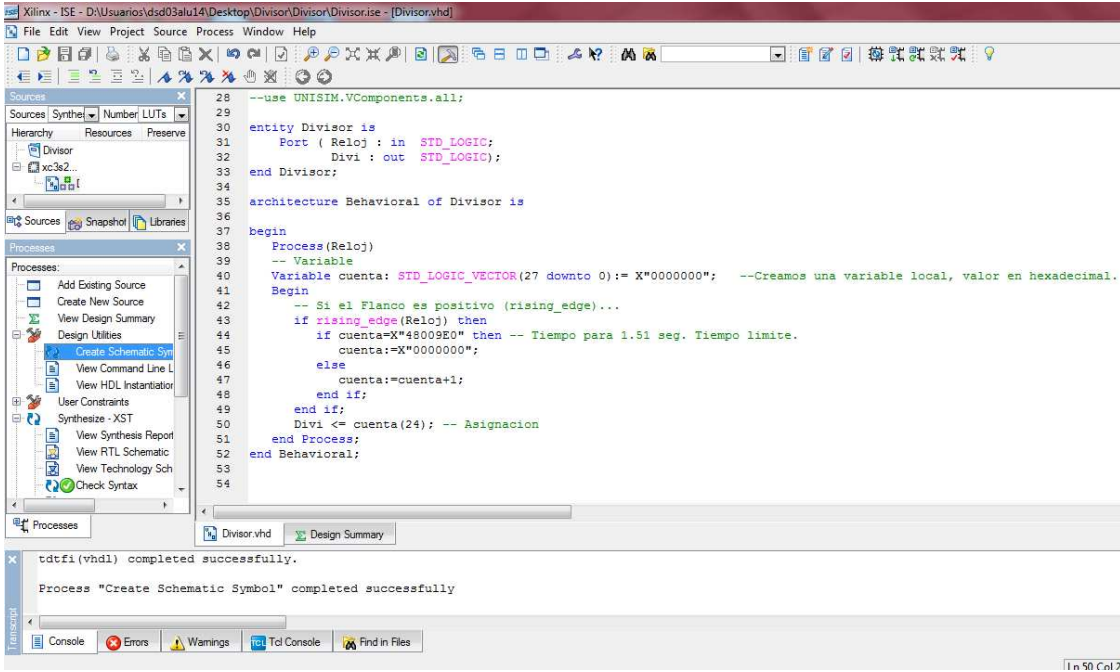
Los valores que tome la variable **“cuenta”**, están expresados en hexadecimal, con el fin de utilizar menos bits en su representación y cuando la cuenta sea 48009E0, se reinicia, este valor se volverá a cero.

El código utilizado es el siguiente:

```
libraryIEEE;
useIEEE.STD_LOGIC_1164.ALL;
useIEEE.STD_LOGIC_ARITH.ALL;
useIEEE.STD_LOGIC_UNSIGNED.ALL;
entity Divisor is
Port(Reloj:inSTD_LOGIC;
Divi:outSTD_LOGIC);
end Divisor;
architecture Behavioral of Divisor is
begin
Process(Reloj)
-- Variable
Variable cuenta:STD_LOGIC_VECTOR(27downto0):= X"0000000";
--Creamos una variable local, valor en hexadecimal.
Begin
-- Si el Flanco es positivo (rising_edge)...
ifrising_edge(Reloj)then
ifcuenta=X"48009E0"then--Tiempo limite. 1.51 seg.
cuenta:=X"0000000";
else
cuenta:=cuenta+1;
endif;
endif;
Divi<=cuenta(26);-- Asignación
endProcess;
endBehavioral;
```

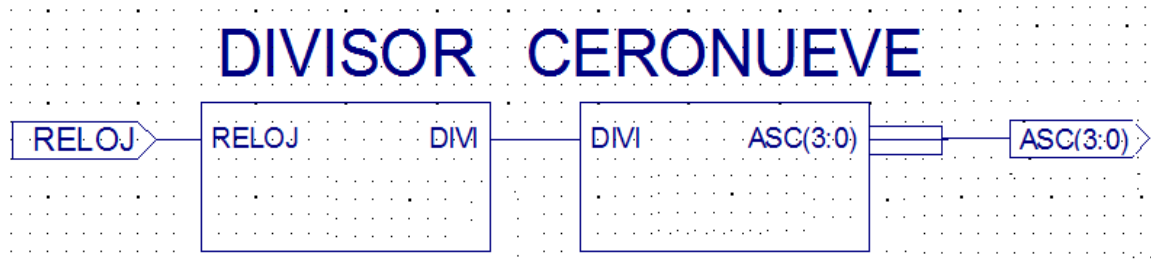
Práctica #6

Una vez terminado el código, vamos a revisar la sintaxis (**check syntax**), si no existe error alguno entonces podremos observar una palomita blanca encerrada en un fondo verde, en el menú de los procesos, indicando que se ha compilado con éxito.



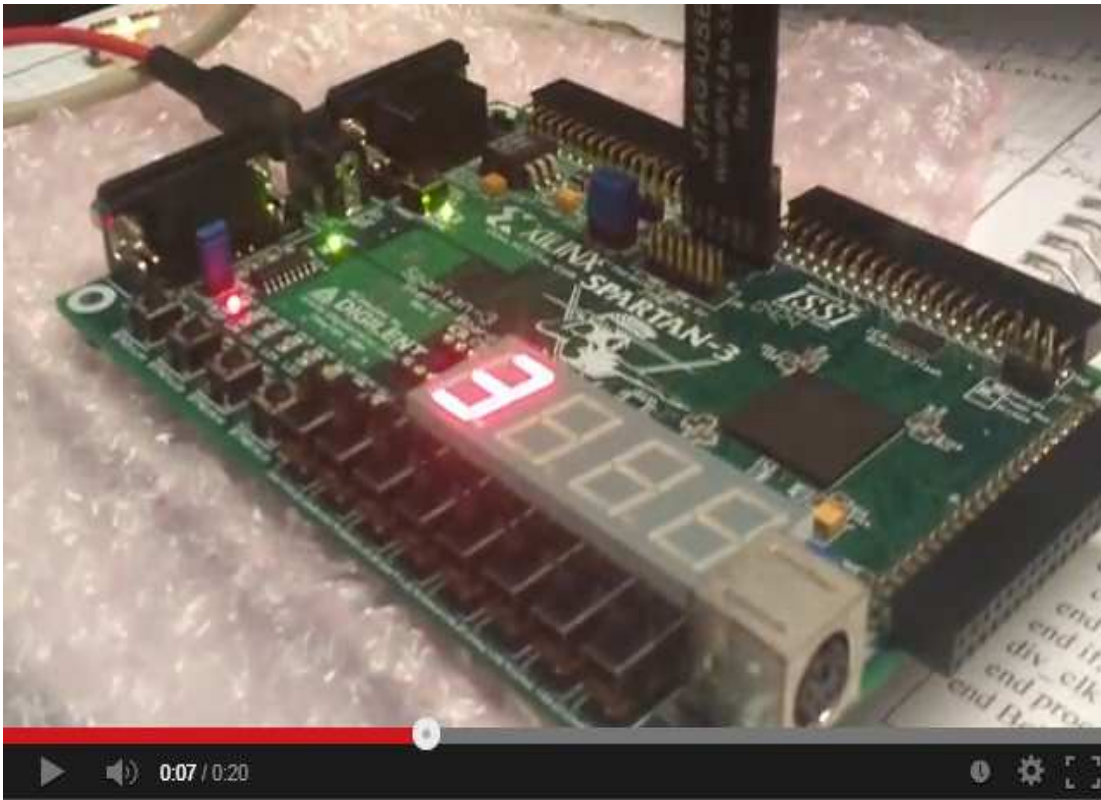
**Resultados:**

El análisis de resultados se hizo mediante la creación de dos bloques funcionales, un **divisor de frecuencia** el cual entre como reloj a un **contador** que cuenta del cero al nueve y vuelve a empezar la cuenta. La siguiente figura muestra el diagrama de bloques.

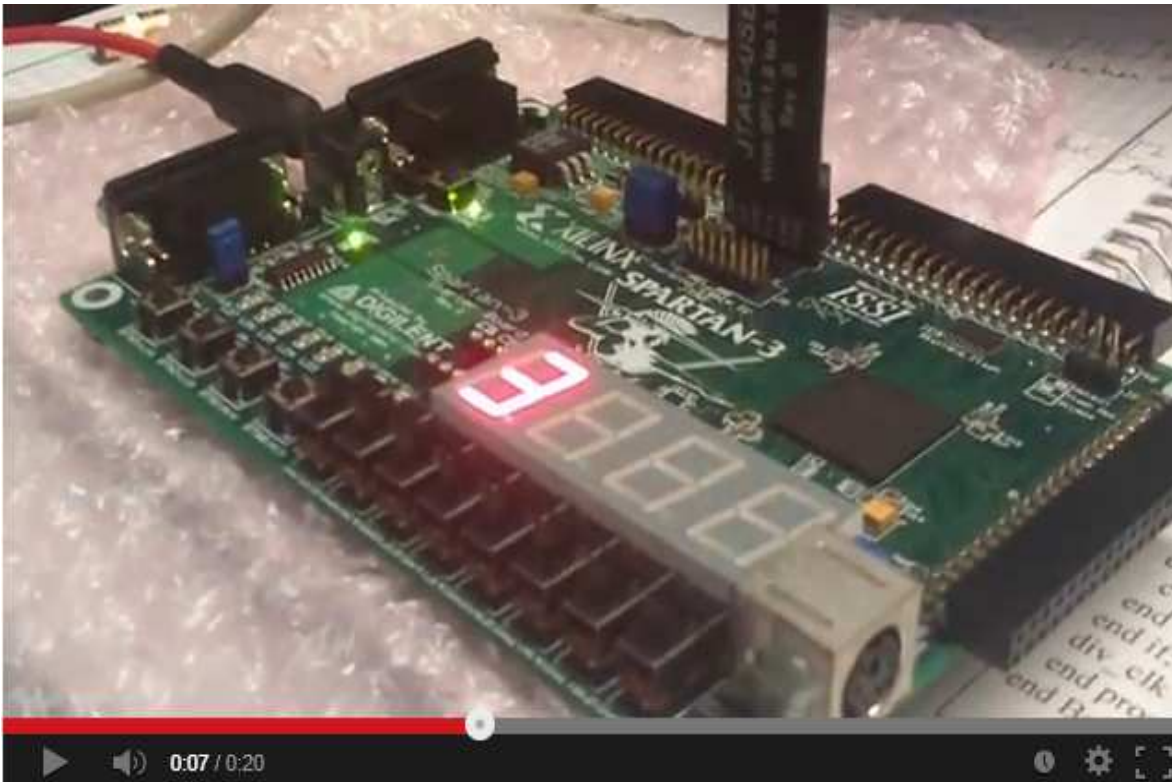


Cada vez que se prende un led, en la tarjeta (indicando el flanco positivo del divisor), el contador cambia de un número a su siguiente consecutivo, y mientras el led está pagado, no hay cambio de número por lo que en el display se mantiene el mismo número.

La siguiente fotografía muestra el cambio de cuenta estaba en el dos y en el momento de prenderse el led, la numeración cambia y se muestra el número tres en el display.



La siguiente fotografía muestra el número 3 en el display y no cambiará hasta que el led se vuelva a prender, mostrando así el número 4.



### Conclusiones:

Podemos observar que la creación de un divisor de frecuencia usando el lenguaje VHD es muy sencilla, podemos modificar valores en el reloj de una manera muy rápida y sencilla y realizar pruebas sin temor a descomponer algo.