
Practica No. 1 Circuitos Secuenciales

Objetivo: Conocer la estructura y características de la tarjeta de los dispositivos lógicos programables que se dispone en el laboratorio, tarjeta TerAsic, el software de operación de esta, Quartus, y su programación en lenguaje VHDL y gráfico.

Repasar el diseño de circuitos secuenciales básicos y algoritmos de máquinas de estados.

Duración: Dos semanas

Desarrollo: Realizar los siguientes apartados:

1.- Siga los pasos siguientes para configurar un contador de 4 bits usando el ambiente de desarrollo Quartus.

En la computadora seleccionar el simbolo de Quartus y ejecutar el programa. Para crear un proyecto nuevo seleccione la pestaña FILE -> New Project Wizard

En el recuadro “What is the working directory for this project” indique el directorio en donde quedara el proyecto, ejemplo c:\estudiante\materia\practicass\practica1. En el recuadro “What is the name of the project” escribir practica1, ver figura 1.

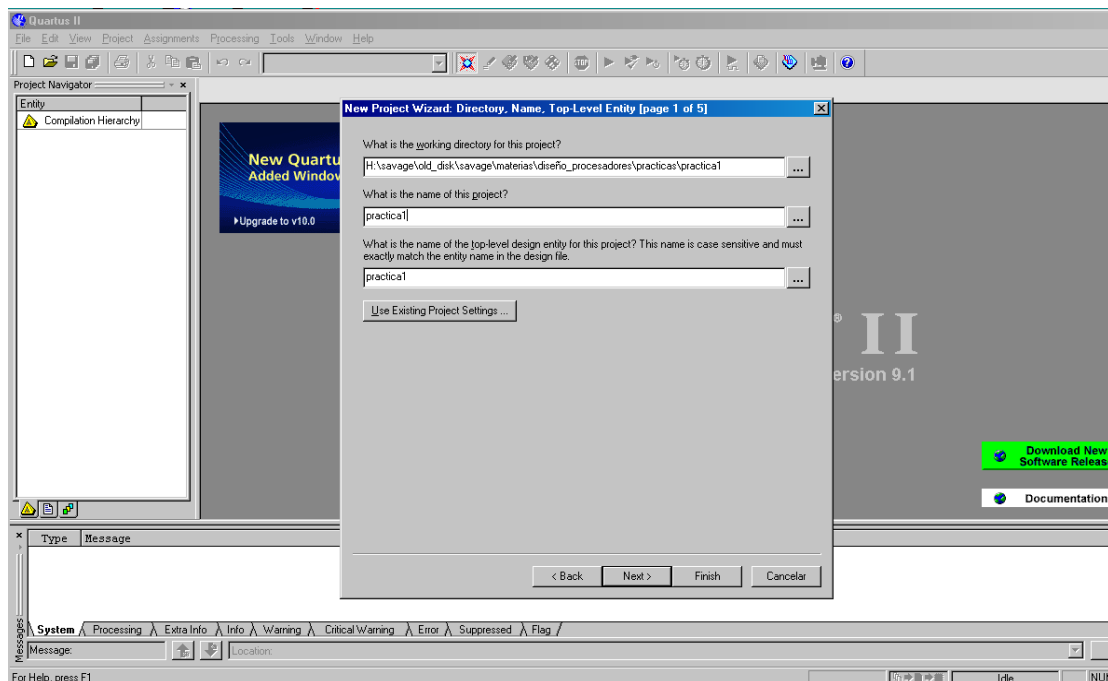


Figura 1. New Project Wizard

Después seleccione next, así como para la siguiente ventana, ahora seleccione el dispositivo que se utilizara, ver figura 2:

Device Family
Family MAXII
Devices all
Target device

X Specified device selected in 'Available list'

Seleccionar: EPM2210F324C3

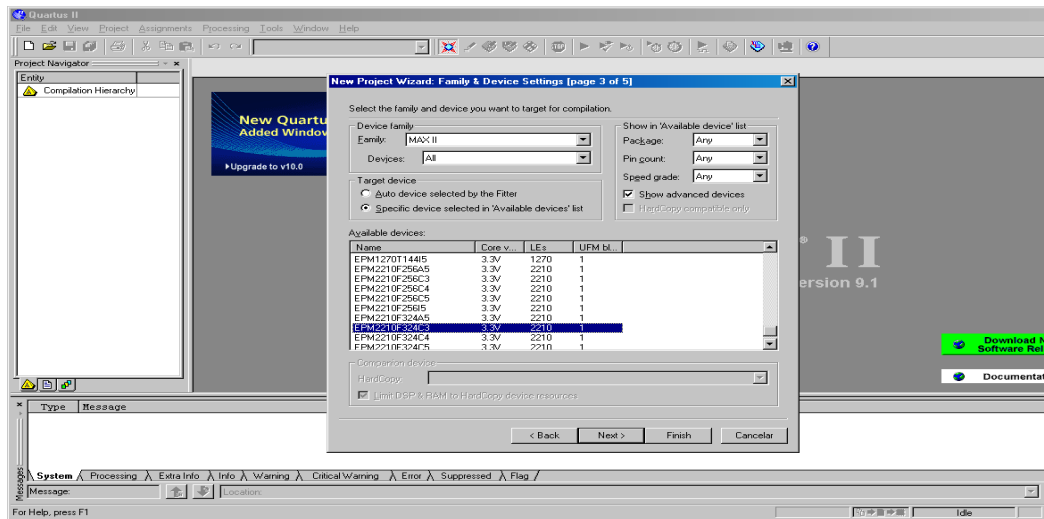


Figura 2. Selección del dispositivo.

Después seleccionar next, lo mismo que para la siguiente ventana, por último seleccionat finish

1.2 Introducción de los componentes en un diagrama esquemático.

Para crear el diagrama esquemático seleccionar en File -> new -> Block Diagram/ Schematic, como se muestra en la figura 3.

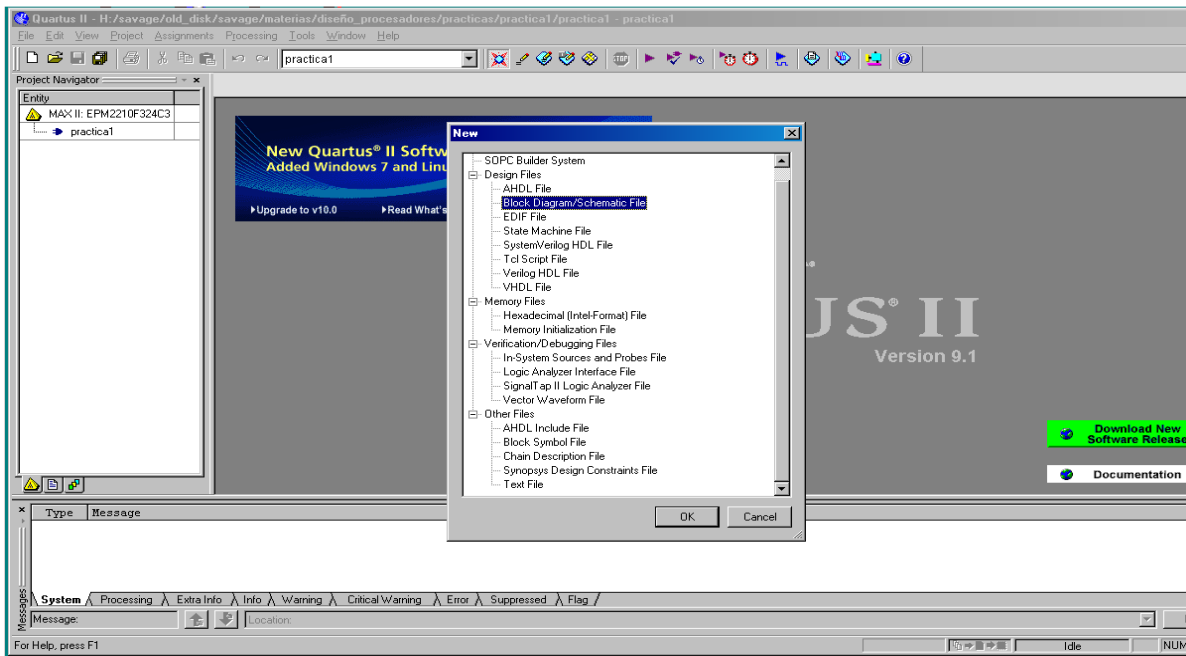


Figura 3. Creación del diagrama esquemático

Aquí aparecerá una nueva ventana en donde se irán colocando los componentes del sistema. Para ingresar un nuevo elemento en esa ventana posicionarse primero en Block1.bdf y con el botón derecho del mouse seleccionar: insert -> symbol, expandir [h:/quartus/libraries](http://quartus/libraries) y después seleccionar others -> maxplus2, ahí escoger el contador 74193, dar ok, ver figura 4.

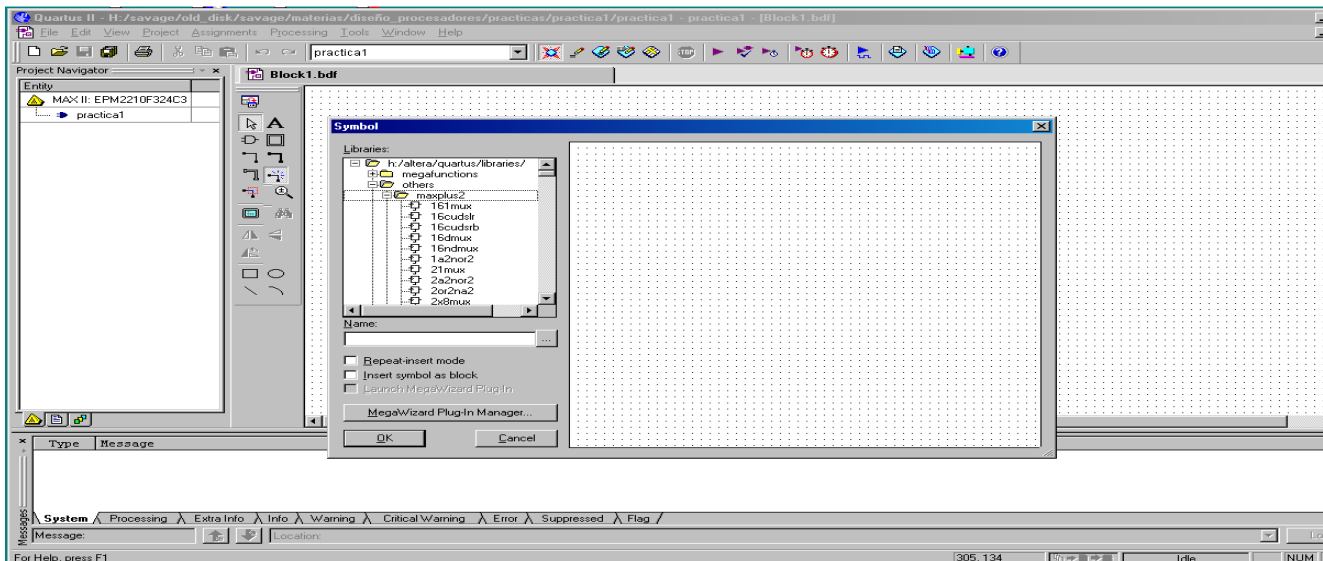


Figura 4. Selección de componentes

Coloque el dispositivo en el diagrama esquemático. Almacene lo que se ha hecho seleccionando el icono de disco y dar guardar. Inserte ahora un pin de entrada, en la ventana practica1.bdf oprima el botón derecho, seleccione insert -> symbol -> primitives -> pin -> input, ver figura 6.

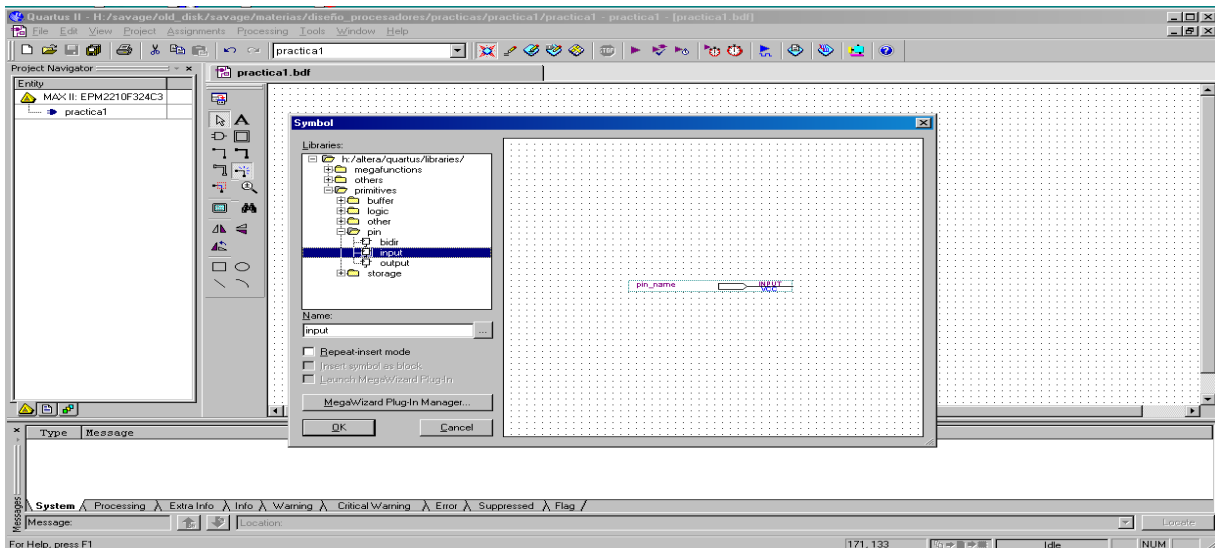


Figura 6. Selección de un pin de entrada.

Coloquelo en el esquemático y conéctelo a la entrada UP del contador, seleccionando con el botón del mouse izquierdo el final del conector y llevando una línea hasta esa entrada. Seleccione de nuevo el conector de entrada con el mouse usando el botón izquierdo, después oprimiendo el botón derecho seleccione properties y cambie el PIN name a clk. Repita el procedimiento para colocar otro pin de entrada conectándolo al CLR del contador y con el nombre reset. Coloque pines de salida en QA, QB, QC y QD del contador, nombrándolos A,B,C y D respectivamente, ver figura 7. Guarde su diseño

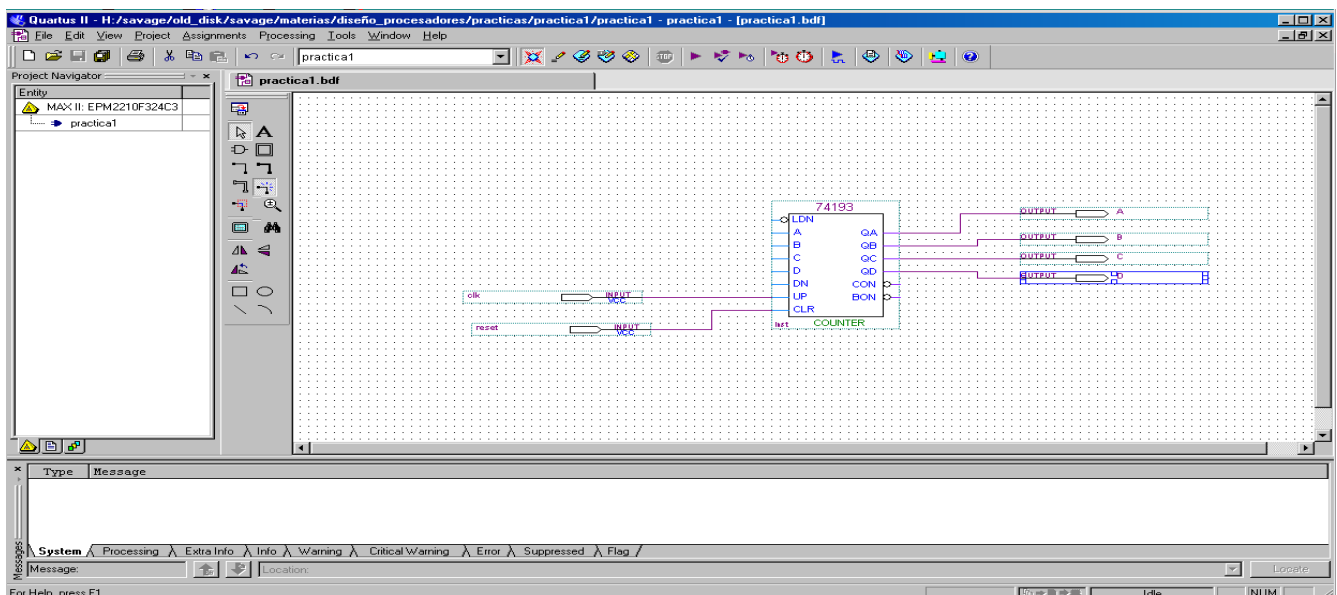


Figura 7. Contador

Para compilar el diseño hecho hasta ahora, seleccione en la pestaña de Processing -> Start

Compilation. Si todo esta bien no debió haber marcado ningun error, en caso contrario, trate de resolver el problema o consulte al instructor.

1.3 Simulación

Para ver si el diseño es correcto, antes de grabarlo en el tarjeta que lo contendra, se utiliza el simulador. Para invocar el simulador seleccione en File -> New -> Vector Waveform File.

Aquí aparecera una nueva ventana, colocar el mouse en el lado izquierdo (abajo de Name) y usando el botón derecho seleccionar Insert -> Insert Node or Bus. Después seleccionar Node Finder y en su ventana seleccionar List escogiendose las variables que se desean ver simuladas. En este caso escojanse todas las variables, ver figura 7.

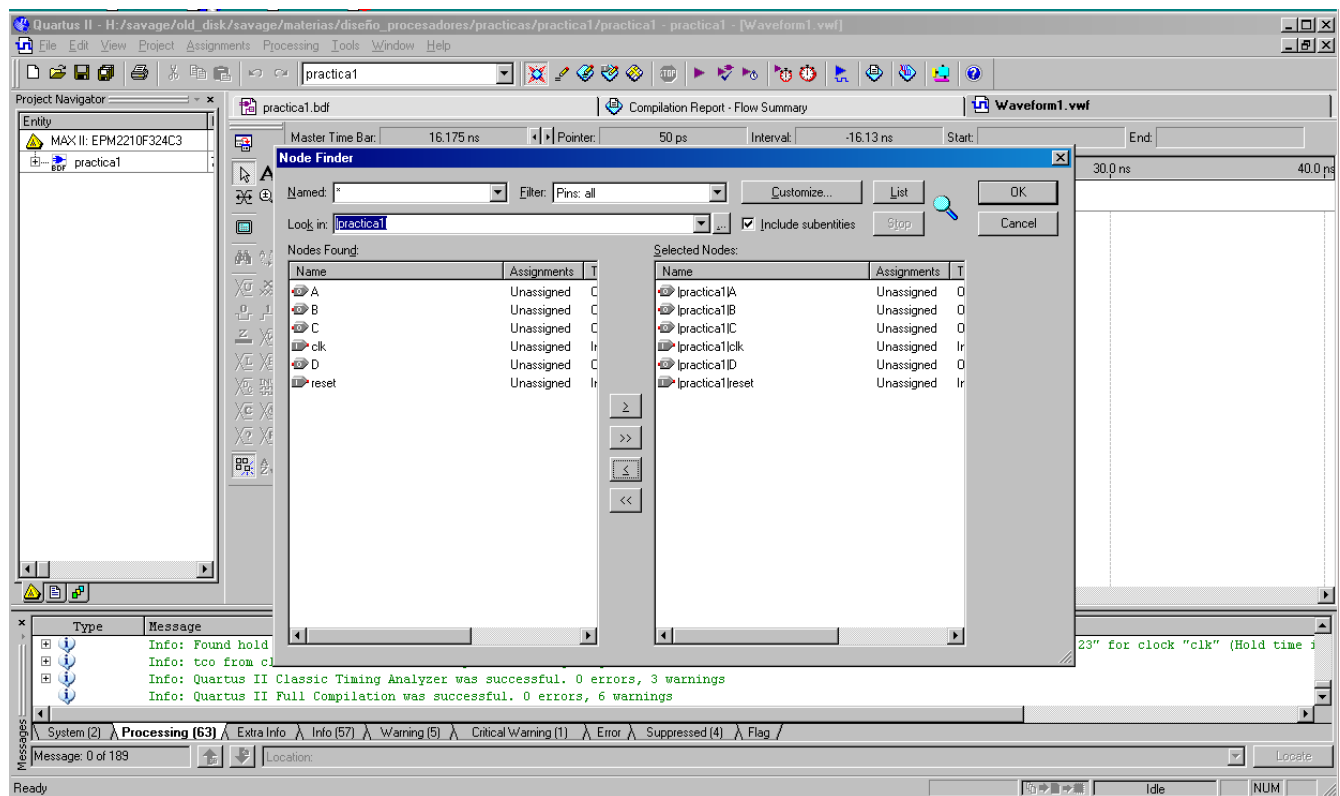


Figura 7. Selección de variables de simulación

En la ventana de las formas de onda de simulación, seleccione la variable clk con el botón izquierdo y después con el botón derecho del mouse, value -> clock, seleccionar OK. Guarde el archivo de simulación seleccionando el icono de guardar archivos. Después active la simulación con Processing -> Start Simulation.

Más adelante aparecerá una ventana, figura 8., la cual tiene el resultado de la simulación, vea que los resultados concuerdan con un contador de 4 bits. Modifique los valores del reloj, clk, y de la variable reset. Comente sus resultados.

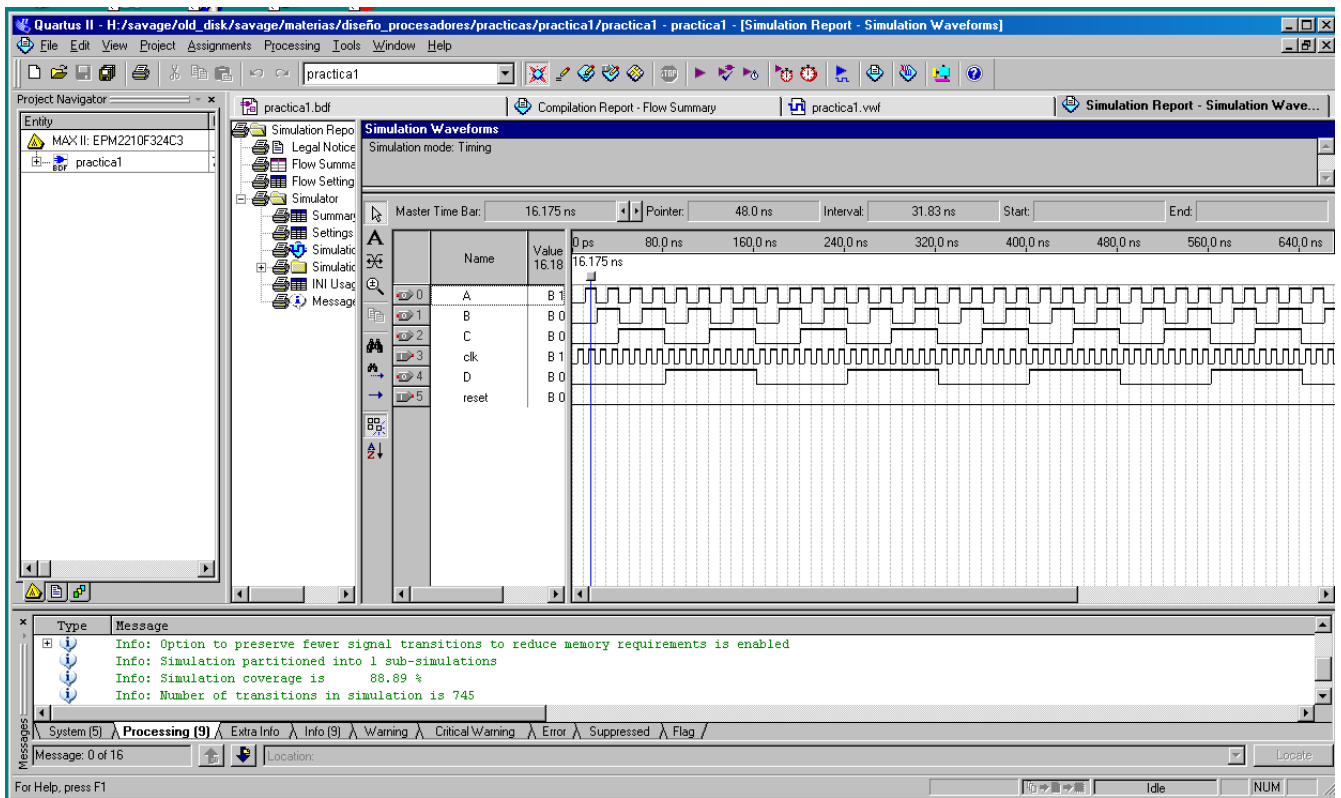


Figura 8. Simulación

1.4 Divisor del reloj

Para poder ver funcionando este diseño en la tarjeta en donde se programará, es necesario hacer un divisor del reloj. El siguiente código en VHDL hace esta función:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity divider is
    Port ( reloj : in std_logic;
          div_clk : out std_logic);
end divider;

architecture Behavioral of divider is
begin
    process (reloj)
        variable cuenta: std_logic_vector (27 downto 0):=X"0000000";
    begin
        if rising_edge (reloj) then
            if cuenta=X"4000000" then
                cuenta:=X"0000000";
            else
                cuenta:= cuenta+1;
            end if;
        end if;
        div_clk <= cuenta (25);
    end process;
end Behavioral;
```

Introduzca este código seleccionando File -> New -> VHDL. Guarde su diseño como divider.vhd. Una vez introducido compruebe que este código no tiene errores de sintaxis con Processing ->Analyze Current File.

Para crear un simbolo de este divisor haga lo siguiente, seleccione de nuevo la ventana de divider.vhd y despues seleccione la pestaña File -> Create/ Update -> Create Symbol files for current file

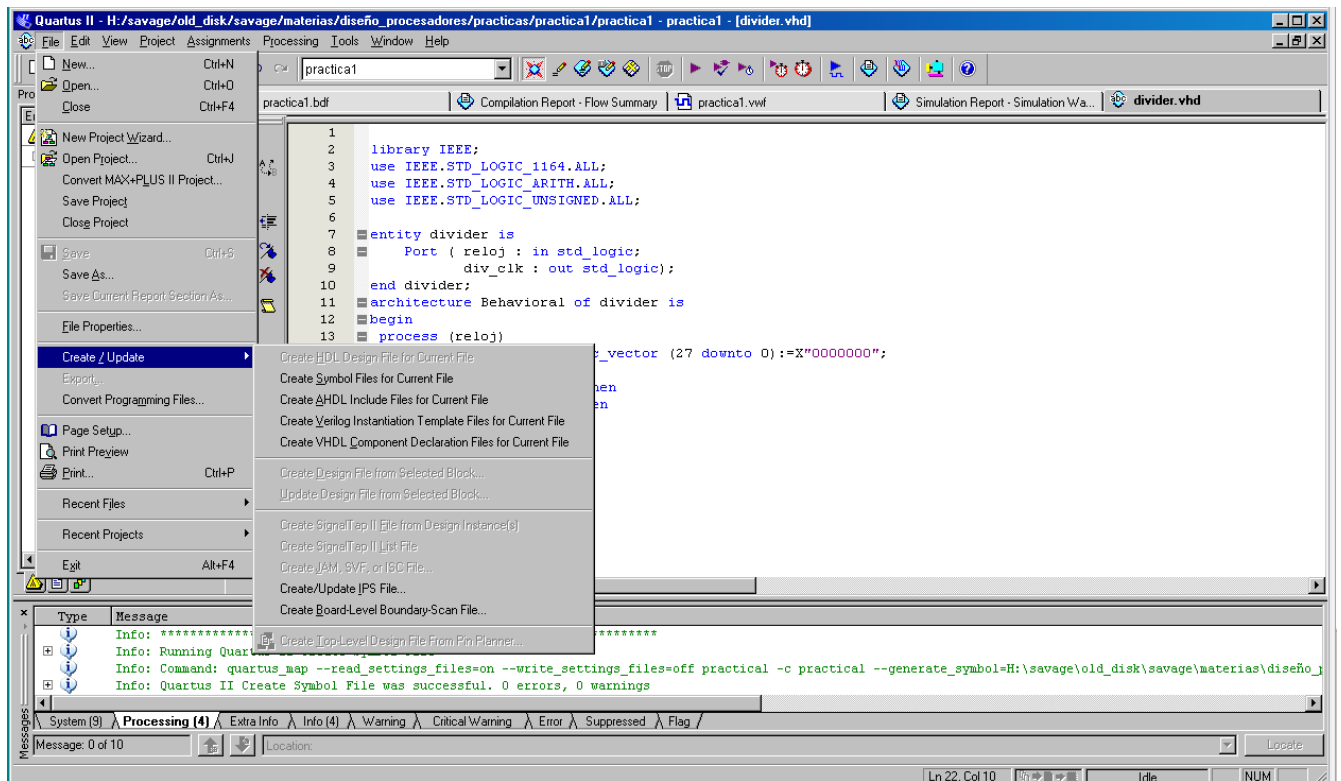


Figura 9. Creación del simbolo del divisor

Para insertar este nuevo simbolo en el esquemático seleccione la ventana practica1.bdf, después con el botón derecho del mouse Insert -> Symbol -> Project -> divider, ver figura 10.

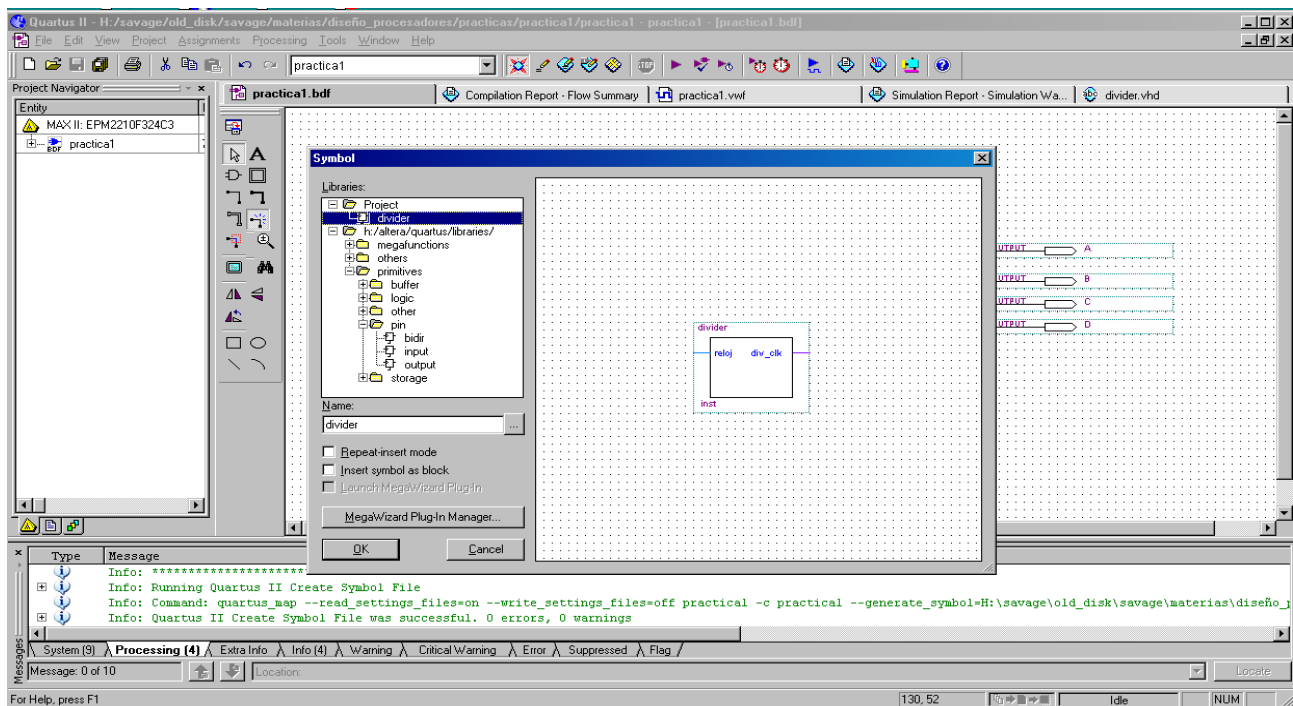


Figura 10. Insertando el simbolo de divider.

Coloque el simbolo y conecte su entrada a la linea clk y su salida al reloj UP del contador, como se observa en la figura 11.

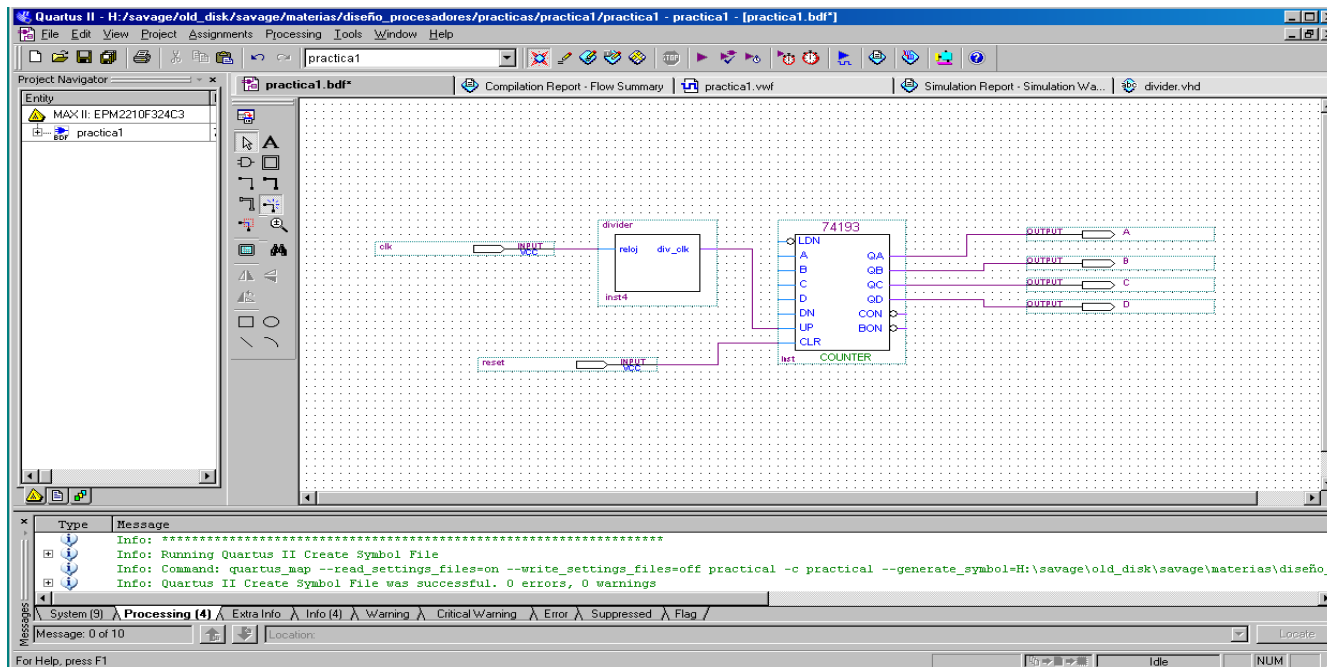


Figura 11.

En la tarjeta en donde se programara este diseño las salidas del contador, ABCD, se observaran usando LEDs, pero estas necesitan ser negadas para ver encendidos los LEDs cuando sus valores sean 1. Lo mismo sucede con la linea de entrada reset. Inserte NOTs en esos pines, como se muestra en la figura 12 y vuelva a compilar su diseño.

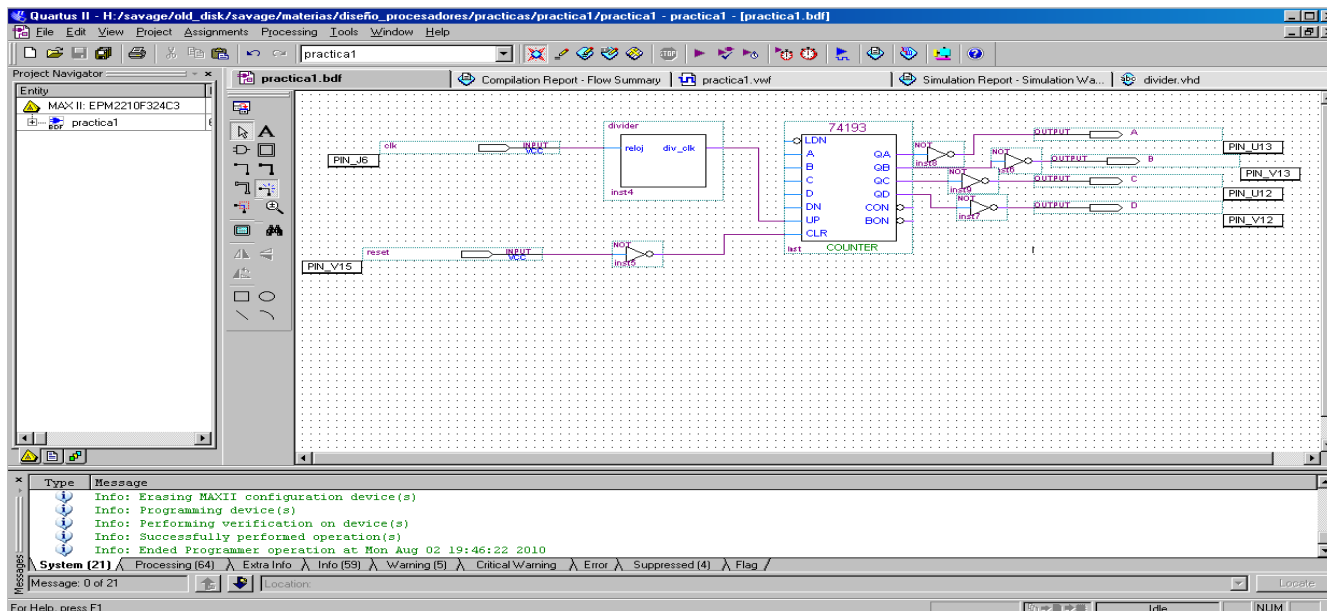
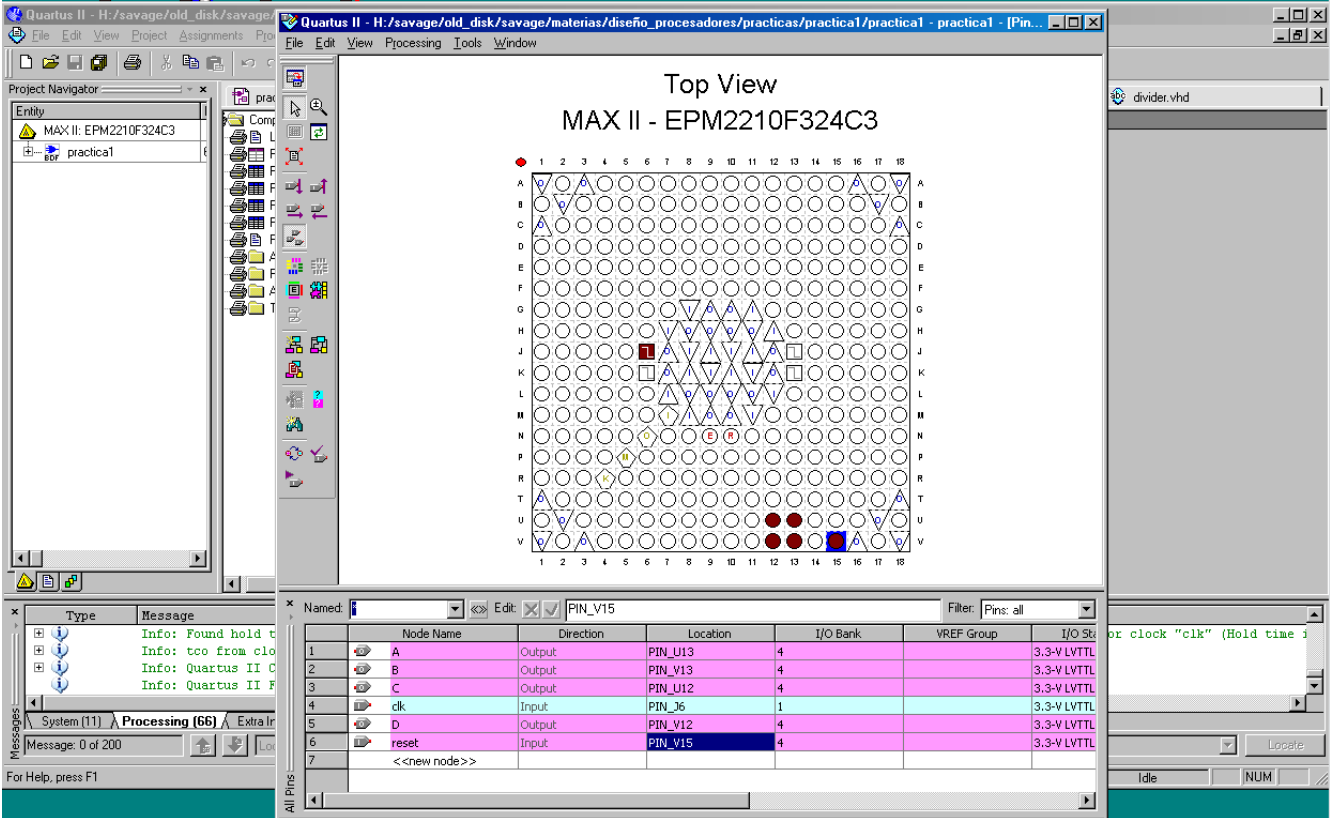


Figura 12. Colocación de NOTS a la entrada y salidas

1.4 Programación física del diseño

Primero se tienen que asignar los pines de entrada, salidas y del reloj, para hacer esto seleccionar la pestaña Assignments -> Pin Planner

En la ventana en donde aparecen los nombres de las variables seleccionar en Node Name clk y en la columna location teclear j6, el cual conecta el reloj al dispositivo. Seleccionar ahora A asignandole el pin U13, repetir para B con V13, C con U12, D con V12 y reset con U15. En el apéndice A se muestra la asignación de pines para la tarjeta de Altera utilizada en el laboratorio.



Node Name	Direction	Location	I/O Bank	VREF Group	I/O Str
A	Output	PIN_U13	4		3.3-V LVTTTL
B	Output	PIN_V13	4		3.3-V LVTTTL
C	Output	PIN_U12	4		3.3-V LVTTTL
clk	Input	PIN_J6	1		3.3-V LVTTTL
D	Output	PIN_V12	4		3.3-V LVTTTL
reset	Input	PIN_U15	4		3.3-V LVTTTL
<<new node>>					

Figura 13. Asignación de pines

Cerrar esta ventana y después seleccionar la ventana practica1.bdf en donde apareceran estas asignaciones que se acaban de hacer. Compilar de nuevo.

Finalmente para programar el dispositivo conecte la tarjeta TerAsic a uno de los puertos USB de la computadora. Seleccionar ahora la pestaña Tool-> Programmer, después en Hardware Setup seleccionar USB-Blaster, active el cuadro Program/Configure y después seleccione Start. Una vez programado se debera ver en los LEDs la cuenta del contador, oprima el botón 1 de la tarjeta y deberá observar como la cuenta regresa a cero.

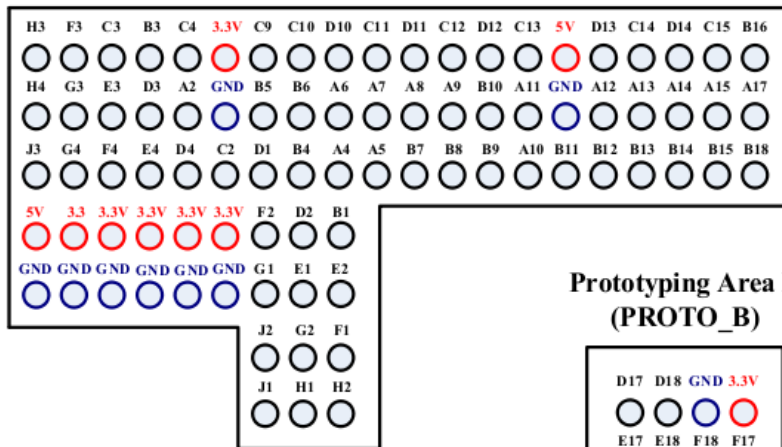
APENDICE A

Configuración de PINES de la Tarjeta de Altera

Signal Name	FPGA Pin No.	Description
LED[0]	PIN_U13	Blue LED
LED[1]	PIN_V13	Green LED
LED[2]	PIN_U12	Yellow LED
LED[3]	PIN_V12	Red LED
LED[4]	PIN_V5	Blue LED
LED[5]	PIN_U5	Green LED
LED[6]	PIN_V4	Yellow LED
LED[7]	PIN_U4	Red LED
KEY[0]	PIN_U15	Button1
KEY[1]	PIN_V15	Button2
KEY[2]	PIN_U14	Button3
KEY[3]	PIN_V14	Button4
CLOCK_50	PIN_J6	50 MHz clock input

Table 2.1. Pin assignments for the LEDs, Buttons, and Clock inputs.

Prototyping Area A (PROTO_A)



Prototyping Area B (PROTO_B)

