**Práctica. Circuitos combinacionales usando Lenguaje de descripción de Hardware VHDL.**

Nombres: Suarez Vega Edgar Alan Grupo: 4CM13

**Objetivo.** Aplicar el lenguaje de descripción de Hardware integrado en el entorno de desarrollo para SPARTAN 3, para la solución de problemas basados en circuitos combinacionales.

Material.

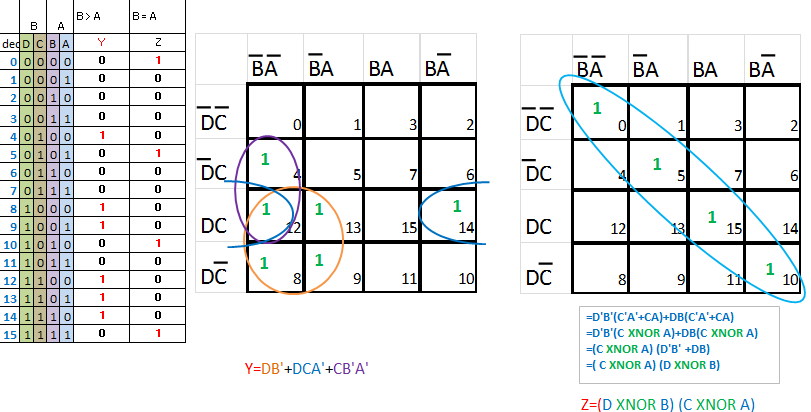
1 Cable para computadora

1 Tarjeta SPARTAN 3

**Desarrollo.**

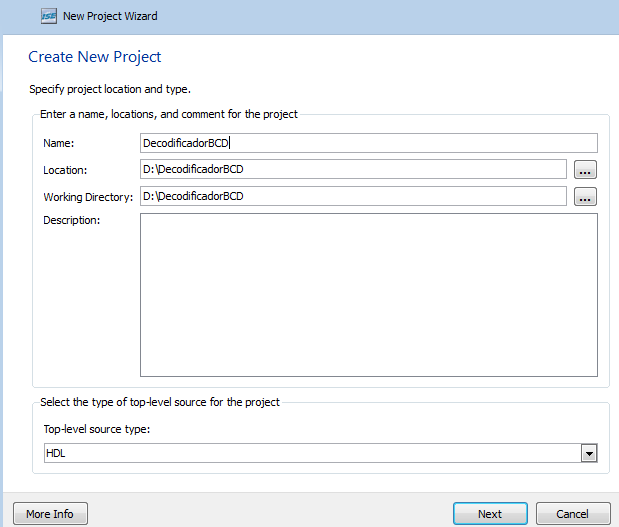
1. Realizamos los cálculos para un comparador de magnitud para dos números de dos bits.

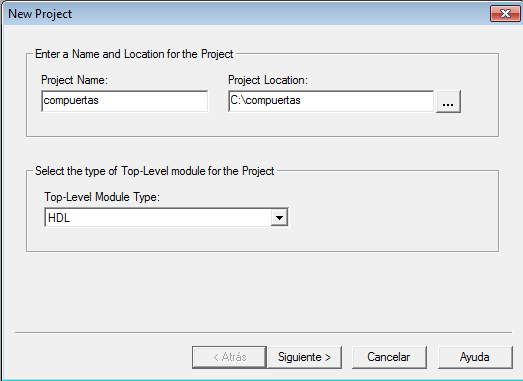
Una salida (Y) del circuito detecta cuando un número es mayor que otro ( B > A ) y otra salida ( Z ) cuando dos números sean iguales ( B = A ). El desarrollo funcional queda.

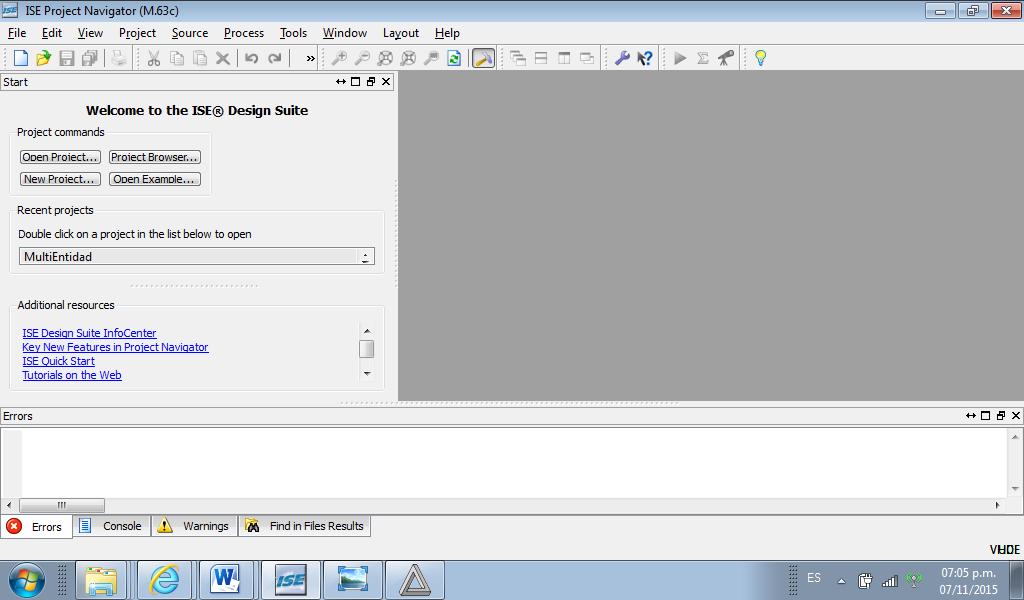
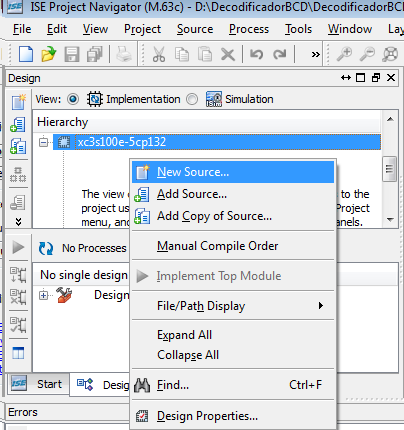


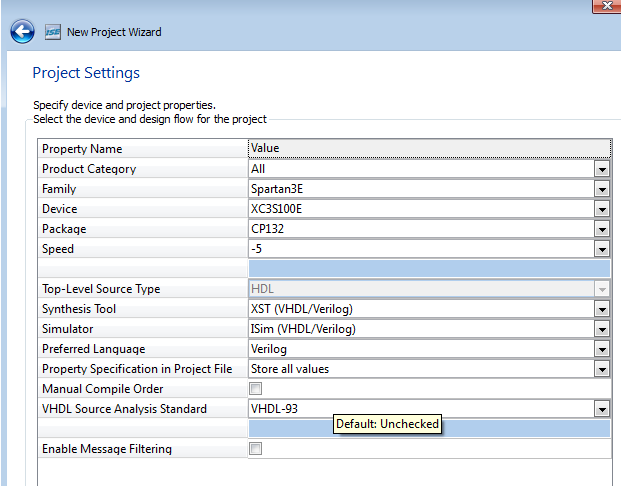
Proponemos crear una solución en el software XILINIX usando un módulo de VHDL . . Usar los pasos siguientes

1. En el manejador de proyectos usar new Project
2. Colocar el Nombre del Proyecto y el lugar de almacenamiento del proyecto
3. Dar los parámetros del integrado SPARTAN3E,X3S100E y CP132 de encapsulado.
4. Dar finalización para escribir el código
5. Agregar un archivo dando clic sobre el proyecto (imagen del chip)usando el menú de new Source

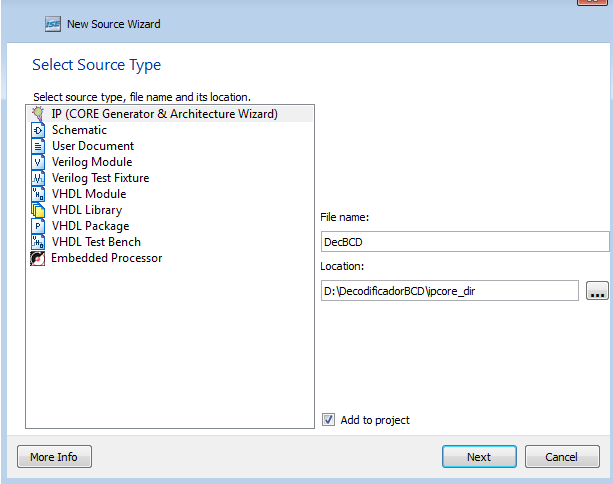
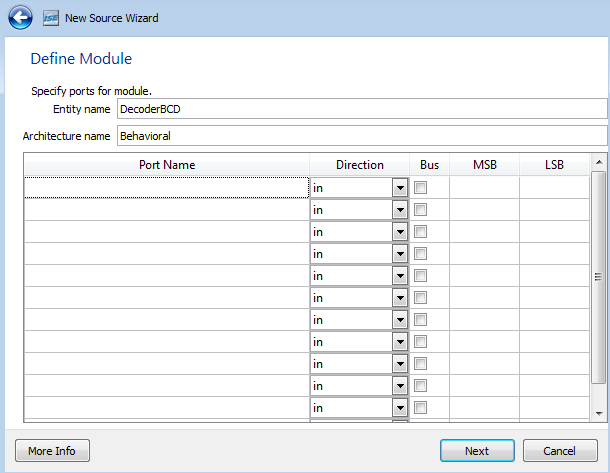




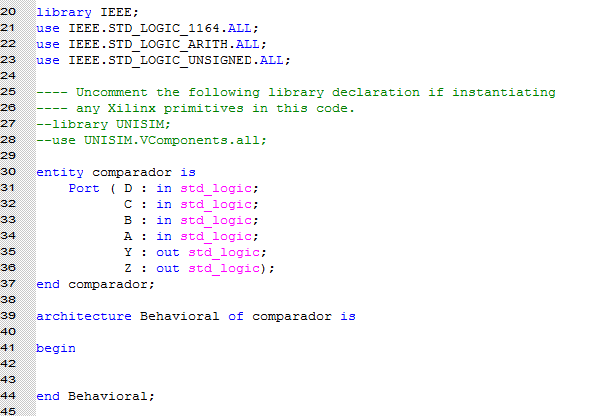




1. Se crea el modulo desde la herramienta colocando el nombre del archivo en el cuadro de Filename previo dar clic en VHDL Module y dar NEXT.
2. El programa abre el editor de pines, para este caso solo demos NEXT ya que daremos el patillaje por código.
3. Finalmente tendremos la ventana del proyecto VHDL



1. Oprimimos la pestaña de código del módulo VHDL para editar el código del comportamiento (Behavioral). En esdte debemos construir las funciones que se obtuvieron de la tabla de verdad para la salida Y , Z. El código formalmente comienza en la **línea 20.** Las líneas en color verde son comentarios, por lo que debemos empezar a codificar en la **línea 42** para el comportamiento del circuito.



Librerías de componentes

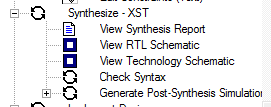
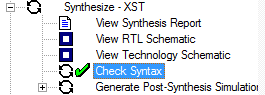
Architecture , se refiere a como esta constituido el funcionamiento del circuito (Behavioral) , dentro del cuerpo de la arquitectura, la palabra begin se usa para indicar donde empieza el funcionamiento del circuito.

Una entidad (entity)es el circuito a construir, este consta de puertos (port) de entrada (in), salida (out) y entrada salida (inout)

Para construir las funciones usamos las palabra **not** (invertir una variable) , **and** (producto) , **or** (disyunción) y **xor** (or exclusiva) Por ejemplo W= CB+BA’ tenemos la asignación de variables se hace con los signos <= , la expresión queda

Una vez realizadas las funciones debemos revisar la expresión (ecuaciones), guardamos y oprimiendo en el panel de procesos del módulo XST (Xilinix Syntesys Tecnology) buscamos

la opción check syntax , de no haber errores podremos ver una paloma verde y finalmente podremos asignar los pines de entrada y salida en el manejador.



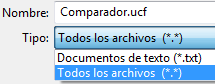
También se puede crear el **Register Transfer Level RTL,** que es la forma en que la herramienta construye los bloques lógicos.

Anote sus ecuaciones para Z e Y en la parte inferior:

Y <= (D AND (NOT B)) OR (D AND C AND (NOT A)) OR (C AND (NOT B) AND (NOT A));

Z <= (D XNOR B) AND (C XNOR A);\_\_

Las entradas y salidas se crean usando el archivo de control UCF. Para ello tenemos que conocer los pines de la tarjeta y asignarlos a los bancos internos del circuito.



Se crea con el bloc de notas un archivo , para guardar use la opción

de tipo todos los archivos , en este caso salve como comparador.ucf

En el archivo se deben escribir las líneas que indican las posiciones

de los interruptores deslizables, botones (entradas) y las salidas Led , por ejemplo

la entrada D usará el interruptor deslizable B4 sw#3 , el comentario se coloca después del;

NET "D" LOC = "B4"; # sw3 variable D

NET "\_C\_” LOC = "\_K3\_\_"; # sw2 variable C

NET "\_B\_” LOC = "\_L3\_"; # sw1 variable B

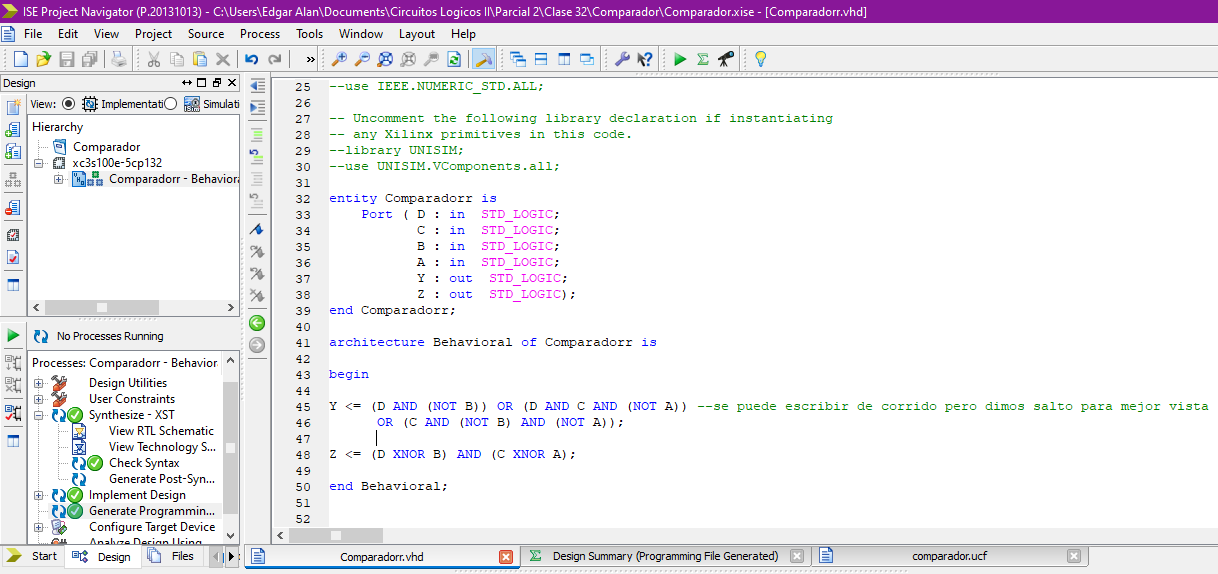
NET "\_A\_” LOC = "\_P11\_"; # sw0 variable A

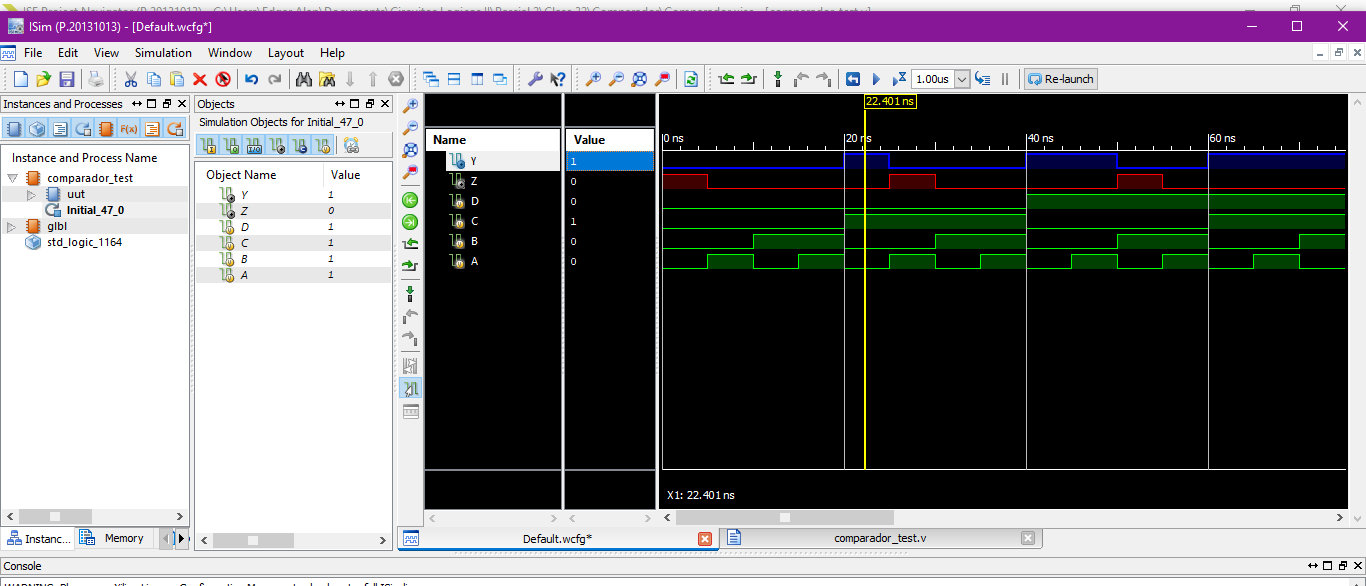
NET "Y” LOC = "M11"; # LD1 Salida Y

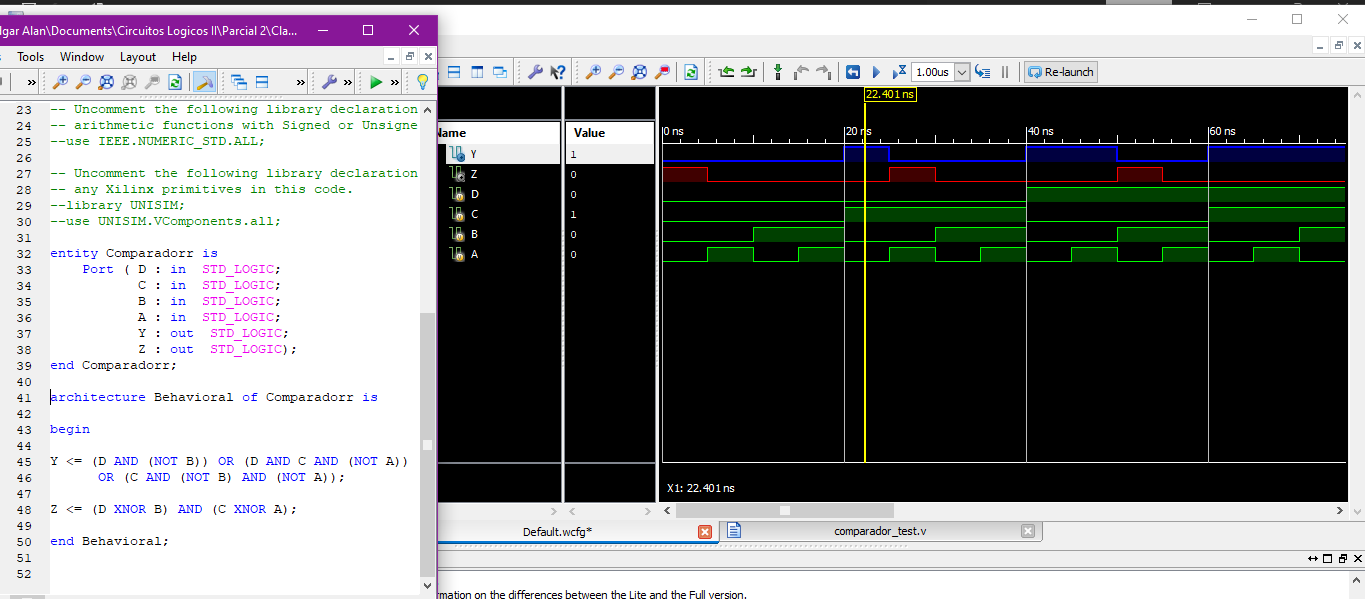
NET " Z ” LOC = "\_M5\_"; # LD0 Salida Z

**Resultados.**

Integre la captura de pantalla del paso 9 y haga las pruebas de comprobación de la tabla de verdad del comparador. Tome una foto de una combinación funcionando de cada salida para documentar su resultado.

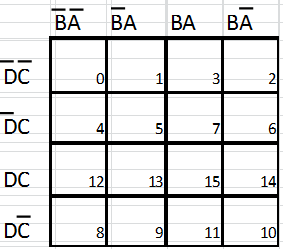






**Evaluación. Esta hoja es individual , se entrega una por cada miembro del equipo.**

Desarrolle una salida llamada **primos** que detecte los números primos de 0 a 15. Use el mapa siguiente para este fin. Use las mismas entradas y solo agregue la salida en el Led7 ( P11).

Anote la ecuación resultante en términos booleanos y en VHDL.

Agregue en los resultados una foto de una combinación.

Conteste las preguntas siguientes.

1. ¿Cuántos tipos de puertos maneja el lenguaje VHDL?
2. ¿Qué función tiene el elemento Entity?
3. ¿Qué función tiene el elemento Behaviorial?