

Se construye la solución, se compila desde abajo, debe pasar todos los pasos que

Son : síntesis, implementación y generación de programa.

(Se validad las conexiones de los componente y los pines)

Se utilizan pines de entrada (P11 y L3) y Salida (M5 y M11)

NET "A" LOC = "P11" ;

NET "B" LOC = "L3" ;

NET "Z" LOC = "M5" ;

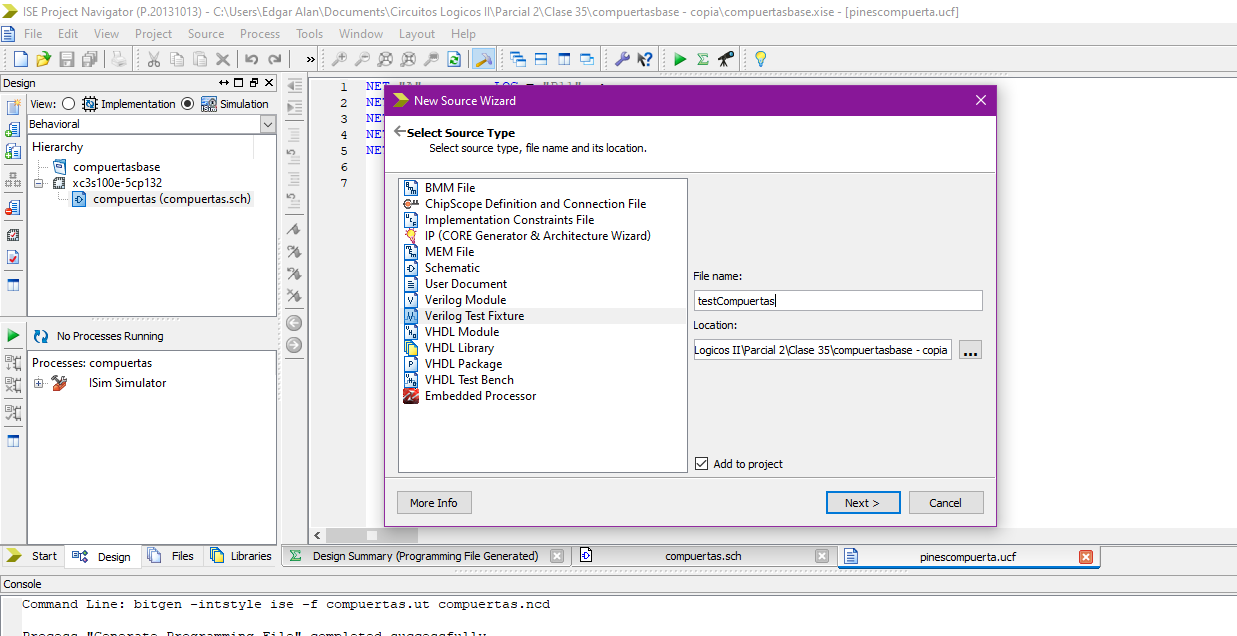
NET "C" LOC = "G12" ;

NET "X" LOC = "M11" ; # sw0



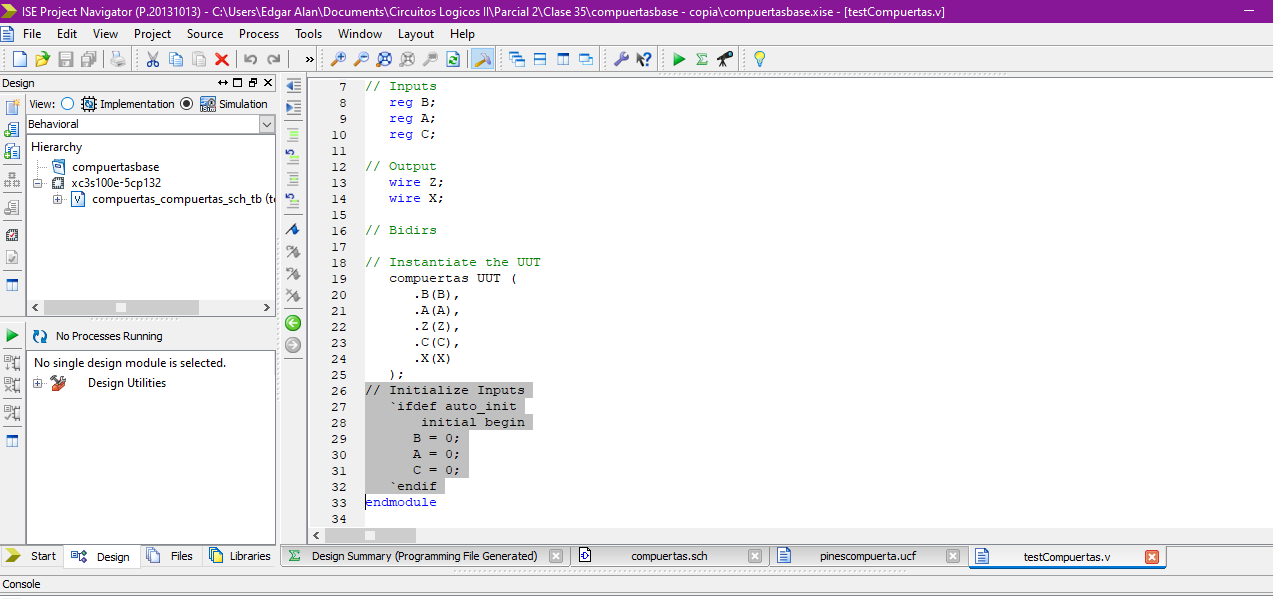
**Para la simulación, seleccionamos la vista (view) de simulación y agregamos un archivo verilog**

**Test fixture colocamos su nombre y su ubicación (que normalmente esta por defecto)**

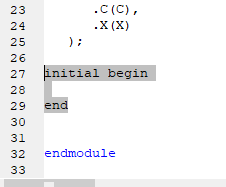


**Posteriormente editaremos el archivo agregando las variables del circuito, asignando valores de binarios (bit) 1/0 , las salidas reflejaran el comportamiento, damos en siguiente hasta finalizar**.

**Eliminamos las líneas de 26 a 32 del código**



**Colocamos el inicio y el final de el bloque donde va nuestro código**



**Colocamos el código dentro de lo antes declarado**

initial begin

A=0;

B=0;

C=0;

#5;

A=1;

B=0;

C=0;

#5; //tiempo de espera para la sig combinacion

A=0;

B=1;

C=0;

#5; //tiempo de espera para la sig combinacion

A=1;

B=1;

C=0;

#5; //tiempo de espera para la sig combinacion

A=0;

B=0;

C=1;

#5; //tiempo de espera para la sig combinacion

A=1;

B=0;

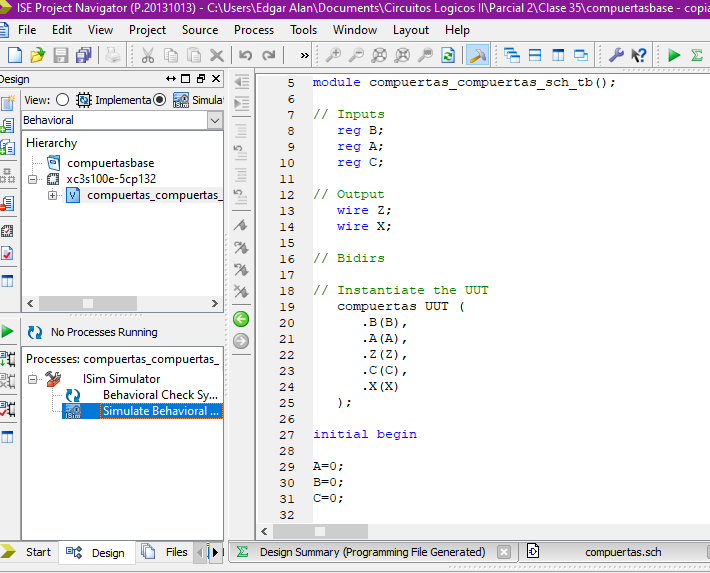
C=1;

#5; //tiempo de espera para la sig combinacion

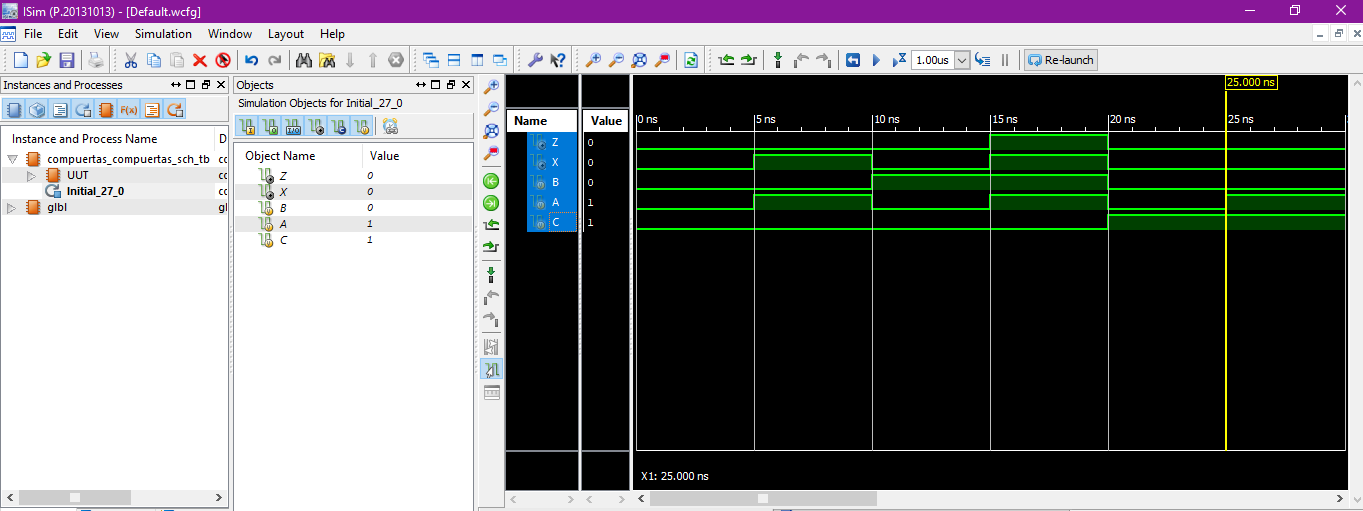
$finish; //indica fin de las combinaciones

end

**finalmente damos doble click en *SimulateBehavorial* para simular**



**La ventana de simulación se muestra así, ya que termino, solo tenemos que ajustar el Zoom usando el (-) y Zoom to full view, que se encuentran en la barra superior y ya podemos ver los estados y valores de nuestras entradas y salidas.**



**Finalmente podemos agregar colores para distinguir mejor**

