

DISEÑO DE PROCESADORES DEDICADOS

Práctica 1 “ISE para el Diseño con FPGAs” Captura y Simulación

Dr. Juan Carlos Herrera Lozada
jlozada@ipn.mx

Instituto Politécnico Nacional



Centro de Innovación y Desarrollo
Tecnológico en Cómputo

CIDETEC

Campo 1: Datos Personales.

Campo 2: Objetivos.

- Instalación de ISE WebPack.
- Familiarizarse con el ambiente de desarrollo.
- Crear macros esquemáticas con VHDL.
- Simulación de diseños simples.

Campo 3: Desarrollo de la Práctica.

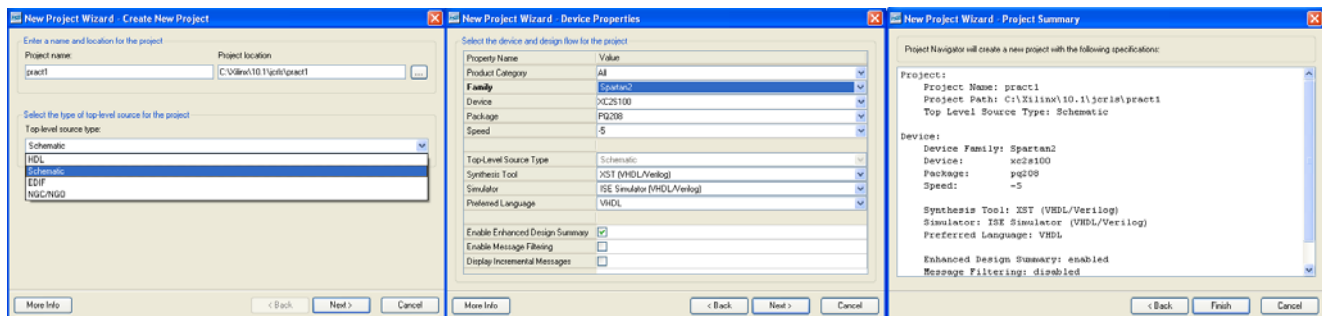
Nota: Para los diseños, anexar los respectivos diagramas, códigos y simulaciones según el caso.

1. (2 puntos) Instala ISE 14.7. Revisa el archivo de texto “*Instalación de ISE WebPack*” disponible en la página web de la materia. Descarga de la misma página, el archivo *p1.zip* que contiene los códigos en VHDL que requieres para realizar la práctica; imprímelos y analiza su funcionamiento.

Accede a ISE Project Navigator y genera un nuevo proyecto; te recomiendo crear una carpeta personal para todos tus proyectos. En la opción *Top - Level Module Type* selecciona *Schematic*. Lo anterior para indicar que el nivel jerárquico máximo estará definido por un diagrama esquemático.

Habilita los componentes necesarios y las bibliotecas que correspondan a la familia *Spartan 2*, con el dispositivo *XC2S100*, con empaquetado *PQ208* y grado de velocidad *-5*. El lenguaje de tu preferencia tiene que ser VHDL.

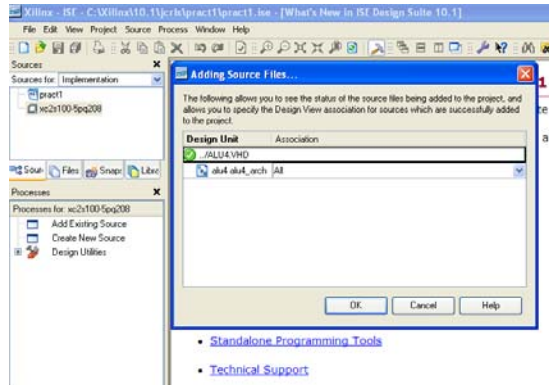
En las siguientes pantallas, presiona *next* para omitir insertar archivos. La última pantalla te mostrará la información del proyecto generado.



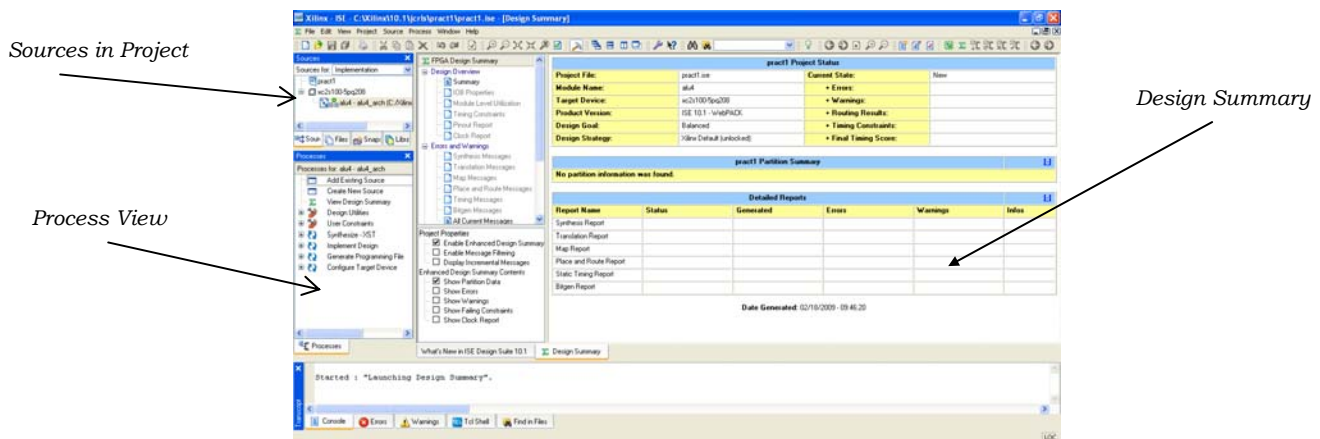
Una vez que estés en la ventana del *Project Navigator*, utiliza el explorador de Windows para descomprimir los archivos *.vhd contenidos en *p1.zip*, en la carpeta de tu proyecto para que formen parte de él.



Regresa a la ventana del *Project Navigator* y accede a la opción *Project* del menú y selecciona *Add Source* para incorporar a tu proyecto el archivo *ALU4.vhd*; acto seguido se te preguntará cómo quieres asociar tu archivo al proyecto (*associate*), selecciona la opción *All*.



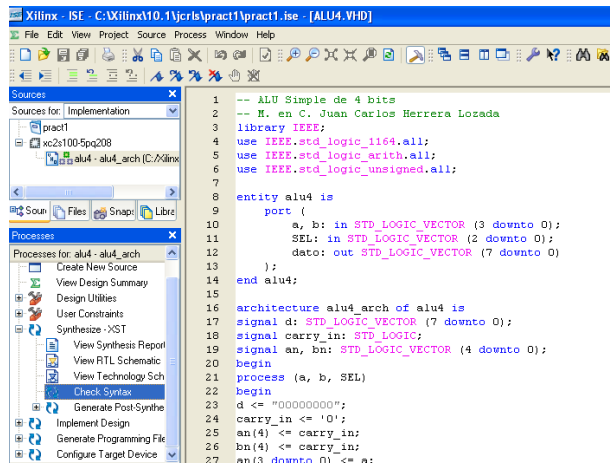
Identifica la sección *Sources in Project* (a la derecha de la pantalla), notarás que es posible observar qué archivo es el que estás utilizando de momento. *Sources for:* tiene 3 posibles opciones: implementación, simulación comportamental y simulación posterior al ruteo. La opción *Implementation* debe ser la activa. En la parte de debajo de este módulo está la sección *Process View* que te permite analizar los procesos válidos para el archivo utilizado. Ambas son muy importantes para tus diseños.



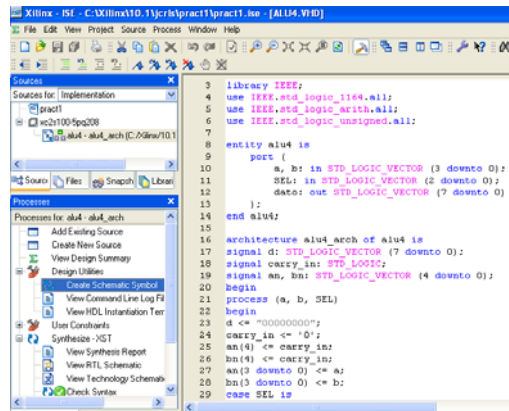
Creando una macro esquemática.

En la sección *Sources in Project*, da doble clic sobre *ALU4.vhd* para abrir el editor de código.

En *Process View*, expande la opción *Synthesize - XST* y presiona *check Syntax* para revisar la sintaxis del código.



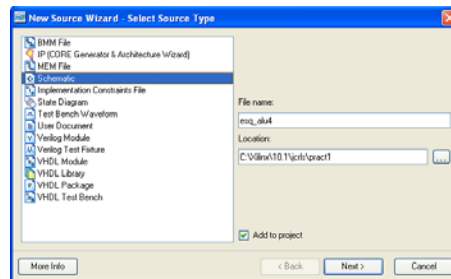
En el mismo *Process View*, en *Design Utilities*, localiza y selecciona la opción *Create Schematic Symbol*, para generar una macro reutilizable con tu código. Por omisión, la macro tendrá el mismo nombre del archivo, que por lo general es el mismo de la entidad (*entity*) en VHDL.



Si se realizó exitosamente el procedimiento, se indicará puntualmente en la consola de mensajes en la parte inferior de la pantalla del *Project Navigator*.

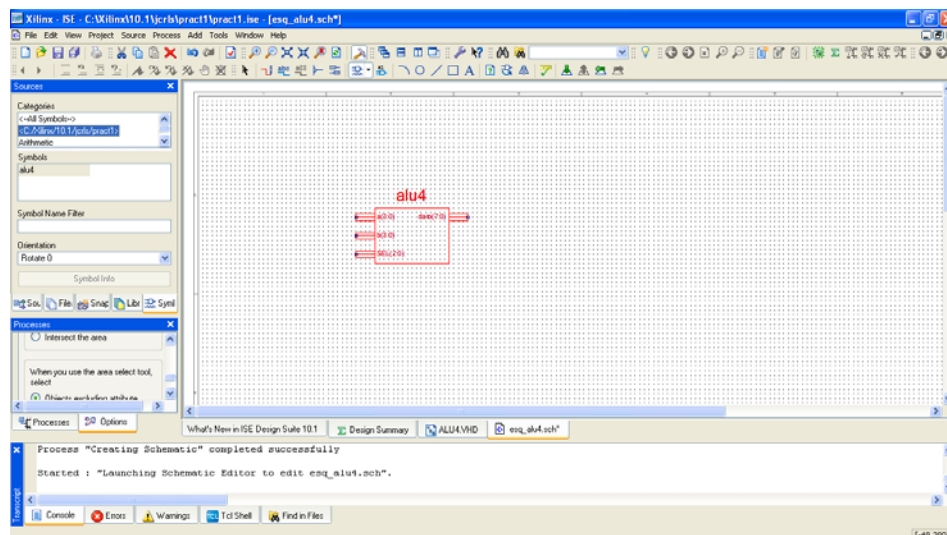
Creando un diagrama esquemático para la macro generada.

Accede al rubro *Project* del menú principal y presiona *New Source*, elige *Schematic* y dale un nombre a tu diagrama. Recuerda que no puedes repetir nombres ni para el proyecto, ni para los archivos implicados en él.



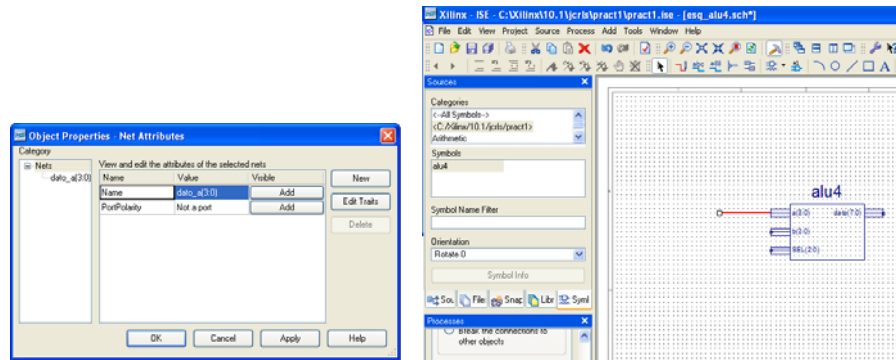
Automáticamente se abrirá el editor de esquemáticos y el archivo se adicionará a *Sources in Project*.


Para colocar tu símbolo esquemático generado, habilita la pestaña *Symbols* del área *Sources in Project*. Desplázate dentro de la biblioteca de símbolos y localiza la opción *Categories* y busca la ruta donde está tu proyecto, ahí se encontrará la macro creada.



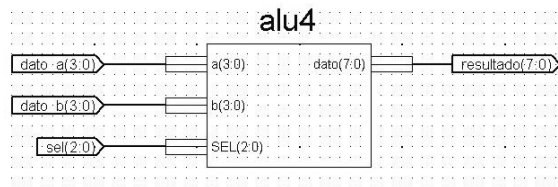
El orden recomendado que se sigue para agregar terminales de entrada y salida en la macro es el siguiente:

- a. Adiciona cable (*Add Wire*), referido con un lápiz en el menú. De manera automática se definirá si se trata de líneas individuales o de buses.
- b. Adiciona nombre a la línea o bus, presionando dos veces sobre el objeto a identificar. En esta macro en particular se trabaja con buses por lo que puedes cambiar el nombre de identificación del bus, más no las dimensiones (líneas individuales) del bus, por ejemplo (3:0). Presiona *Apply* para validar. La etiqueta asignada no se verá, sino hasta que adiciones los conectores, que es el paso siguiente.

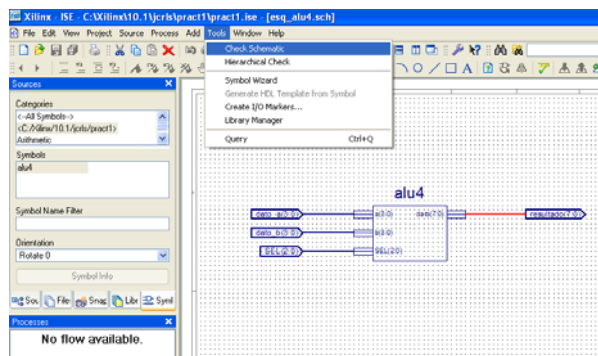


- c. Adiciona los conectores (*Add I/O Marker*) en las terminales, la opción en el menú está referida por el botón que muestra unos pads . Por omisión, el tipo del conector es automático, para no definir de manera individual si se trata de una entrada o de una salida, o de una sola línea o conjunto de ellas.

El esquemático final se muestra a continuación.



Guarda tu diagrama y desde el menú de la edición esquemática, accede a *Tools* y selecciona *Check Schematic* para revisar que tu diagrama esquemático esté correctamente conectado.



Simulando el diseño.

NOTA: COMPLEMENTA ESTA PRÁCTICA DESCARGANDO EL Tutorial para el Simulador ISIM, EN LA PÁGINA WEB DE LA ASIGNATURA.

2. (2 puntos) Realiza el mismo procedimiento del numeral 1, generando una macro con el archivo *CONTA.vhd* descargado de p1.zip. Estando en el área de *sources*, es posible remover parcialmente los archivos que no requieras en el proyecto. Como se trata de un contador binario, requieres simular una señal de reloj; refiérete al tutorial de simulación para cumplir con el circuito secuencial síncrono.

3. (6 puntos) Complementos.

3.a. Realizar una lista de los componentes más comunes encontrados en la biblioteca de símbolos esquemáticos. Verifica el acceso a la información de cada símbolo.

3.b. Modifica el código de la ALU para un diseño de 8 bits en vez de 4.

3.c. Modifica el código del contador para que funcione de manera ascendente – descendente, con 4 bits.

3.d. Simula el sumador en cascada de 4 bits, disponible en la página web de la asignatura. Nota: posiblemente se debe adecuar el código.

3.e. Simula el multiplicador de 8 bits, disponible en la página web de la asignatura. Nota: posiblemente se debe adecuar el código.

Campo 4: Conclusiones individuales.