Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ingeniería Eléctrica
IEE2463 Sistema electrónicos programables

Ayudantia 02 2S23

ZYBOZ7 - Data types y Operators con VHDL

Ayudante: Catalina Sierra catalina.sierra@uc.cl
Prof. Dr.-Ing. Félix Rojas - felix.rojas@uc.cl

---

1. Objetivo de la Ayudantía

- Ejercitar conceptos de *data types* y *operators* en programación en VHDL.
- Comprender el funcionamiento y programación de una ALU en VHDL.
- Ejercitar el uso de perféricos (IP Cores) en un *Block Design* en Vivado para el desarrollo de una ALU e inicio de arquitectura de un procesador simple.

2. Actividades Previas a la Ayudantía

En el video de la ayudantía se explica y desarrolla el código de una ALU en VHDL, además de un módulo SM (*State Machine*) que permite seleccionar los operandos y operación a realizar en la ALU. Finalmente se construye un Block Design en Vivado con los módulos ALU, SM y RAM (esta última de la ayudantía 1) como IP Cores, y tras la generación del bitstream se carga el diseño a la Zybo.

- Crear un nuevo proyecto en Vivado asociado a la ZyboZ7-10. Si no encuentra la tarjeta en el menú de Vivado, descargar el archivo aquí, y descomprimirlo en el siguiente directorio dentro del archivo donde se instaló Vitis:*...\Xilinx\Vivado\2015.1\data\boards*. Reiniciar Vivado tras descomprimir los archivos.
- Incluir en el *IP Repository* los IP Cores asociados a ALU, SM y RAM. Esto se logra añadiendo el *path* relativo a la carpeta donde se encuentran los IP Cores.
- Crear un *Block Design* donde se conectarán los módulos ALU, SM y RAM de acuerdo a lo visto en el video de la ayudantía.

- Incluir archivo *Constraints* asociado a la tarjeta y editarlo de acuerdo al *Block Design* desarrollado.
- Generar el *HDL Wrapper* del *Block Design* y luego el *bitstream*, para cargarlo a su tarjeta Zybo y probar su funcionamiento.

3. Actividades Durante la Ayudantía

El ejercicio propuesto para esta ayudantía consiste en incorporar 2 nuevas operaciones a la ALU: multiplicación y división, usando para ello los códigos 0110 y 0111 respectivamente. A continuación se presentará una serie de pasos para orientar el desarrollo de esta actividad.

- Abrir el proyecto donde se desarrolló el código VHDL de la ALU para editarlo de acuerdo a lo solicitado.
- Para realizar las operaciones de multiplicación y división, se deberá hacer conversión de *data types*, ya que estas operaciones deben realizarse con números enteros. Así, es necesario ser cuidadosos sobre cómo hacer estas conversiones, ya que el resultado debe estar nuevamente como un número binario.
- Tras la edición, re-empaquetar el IP Core de la ALU. Esto se realiza, estando en el proyecto de la ALU, en la ventana *Package IP*, en la sección *Review and Package*, clickeando la opción *Re-Package IP*. Tras esto, en el *Block Design* del proyecto completo, se deberán actualizar los IP, opción que aparece tras re-empaquetar un IP core que está en uso.

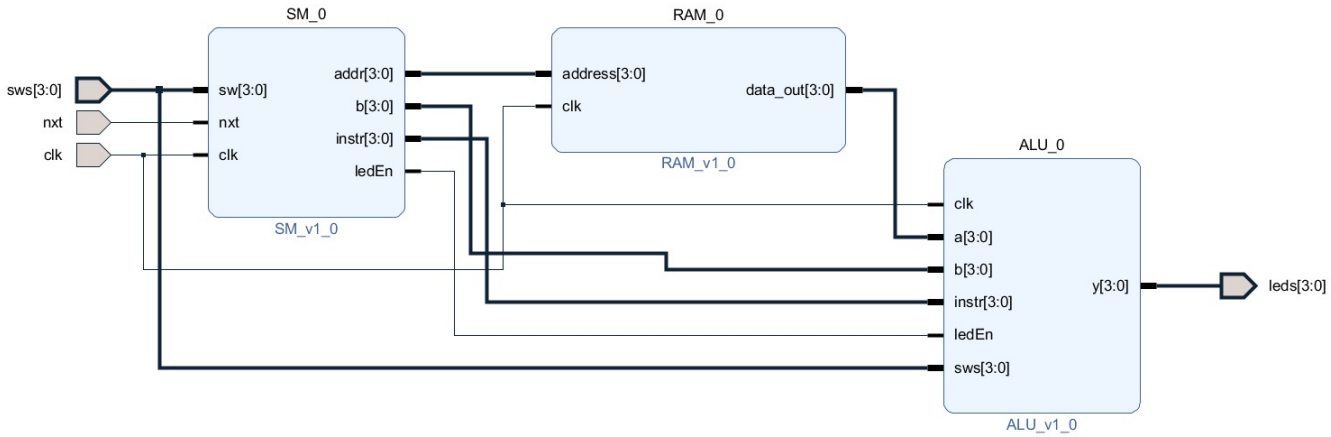

Figura 1: *Block Design* en Vivado del proyecto completo (ALU+SM+RAM)

Código	Operación
0000	y <= a
0001	y <= b
0010	y <= a+1
0011	y <= b-1
0100	y <= a+b
0101	y <= a-b
0110	y <= a*b
0111	y <= a/b
1000	y <=NOT a
1001	y <= a AND b
1010	y <= a OR b
1011	y <= a NAND b
1100	y <= a NOR b
1101	y <= a XOR b
1110	y <= sl a
1111	y <= rr a

Tabla 1: Tabla de códigos ALU