Lab 2: Demultiplexer, Decoder, and Encoder

Today we will continue with our logic circuit design by implementing more advanced combinatorial circuits using concurrent and sequential statements. We will discuss the relationship between the number of input and output bits, see how to use the generic keyword, see further Vivado functionality, and write more logic circuits with VHDL.

14:15-14:35: Discussion – Lab 1

1. Any issues?
2. Discussion of bonus task from lab 1 a The generic keyword

14:35-15:00: Demonstration – Packaging and Export of IP, Inclusion of Entities in New Project

1. Packaging and export of block design, block design reuse
2. Creation of new project
3. Import of entities from last project

15:05-15:45: Tasks – Demultiplexer_1x8, Decoder_2x4, Encoder_8x3

Implement the logic circuits *demultiplexer_1x8*, *decoder_2x4*, and *encoder_3x8*, as of tables 1, 2, and 3, in VHDL following these steps:

1. Create the .hdl file and the VHDL entity
2. Write up the correct port configuration – think about if you should use STD_LOGIC or STD_LOGIC_VECTOR and consider the relationship between the number of inputs, outputs and selects
3. Implement the block functionality using either of the following styles:
    a. Sequential code, if/else statements
    b. Sequential code, case statements
    c. Concurrent code, when statements
    d. Concurrent code, with/select statements
4. Test and verify in behavioural simulation

s	o
0	"0000000i"
1	"000000i0"
2	"00000i00"
3	"0000i000"
4	"000i0000"
5	"00i00000"
6	"0i000000"
7	"i0000000

Table 1: Demultiplexer_1x8 truth table

i	o
"00"	"0001"
"01"	"0010"
"10"	"0100"
"11"	"1000"

Table 2: Decoder_2x4 truth table

i	o
"00000001"	"000"
"00000010"	"001"
"00000100"	"010"
"00001000"	"011"
"00010000"	"100"
"00100000"	"101"
"01000000"	"110"
"10000000"	"111"
others	"UUU"

Table 3: Encoder_8x3 truth table

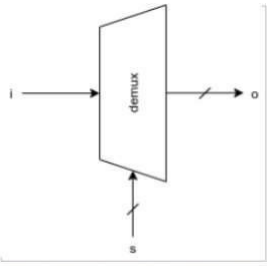

Figure 1: Demultiplexer block symbol

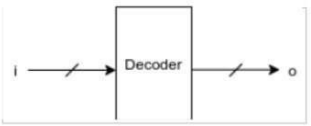

Figure 2: Decoder block symbol

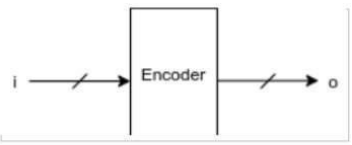

Figure 3: Encoder block symbol