



---

WEEK 1

**Introduction &  
Structural Model**

---

## **1 Introduction**

### **1.1 Aims**

- Get familiar with Vivado software and the FPGA development flow.
- Get familiar with FPGA Arty-Z7 board.
- Practice in designing simple digital logic circuits with Verilog.
- Understand the hierarchical design principle.
- Practice in writing test benches for a designed module.

### **1.2 Preparation**

- Read the laboratory materials before class.
- Revise chapter 0-3 about Verilog basic.
- Each group prepare at least one laptop with Vivado software installed.

### **1.3 Documents and lab materials**

- M. Morris Mano, Michael D. Ciletti, Digital System with an Introduction to the Verilog HDL, VHDL, and SystemVerilog, Pearson Education, Inc, 2017
- Lecture slides
- *Arty-Z7-20-Master.xdc*: Arty-Z7 constraint file.
- *Guide\_for\_Installing\_Vivado.pdf*: Guide for installing Vivado and getting started with Vivado and Arty-Z7.
- *dec1to2.v*, *mux2to1.v*: 2-to-1 multiplexer module and its sub-module.
- *mux2to2\_tb.v*: test bench to simulate the module mux2to1.

### **1.4 Procedure**

For each exercise (also for further labs):

- Read the requirements, then determine the input/output signals of your circuits.
- Make design idea of the circuit then using Verilog to model the circuit.
- Analysis & Synthesis the circuit with Vivado software.
- Write test bench to simulate the circuit on Vivado Simulator.
- Generate the bitstream and program the Arty-Z7 to evaluate the circuit.

## 1.5 Report requirements

- Lab exercises will be reviewed directly in class.
- Write report (with circuit/simulation screenshots inserted) in pdf.
- Must have group ID, group member's names and student IDs in the report.
- Compress the report with code files (only .v files) in only one .zip file, name the .zip the group ID (for example: Group1.zip).
- Submit on BK-elearning by deadline.

## 2 Exercises

### 2.1 Exercise 1

- a. Design a *1-to-2 decoder* using structure model as the following circuit:

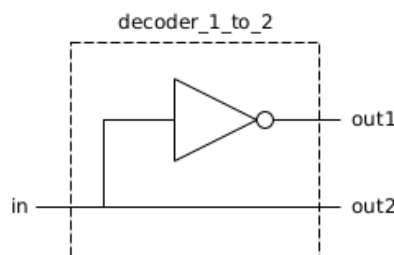


Figure 1: 1-to-2 decoder

- b. Design a *2-to-1 multiplexer* using structure model and hierarchical design (reuse the module decoder\_1\_to\_2) as following circuit:

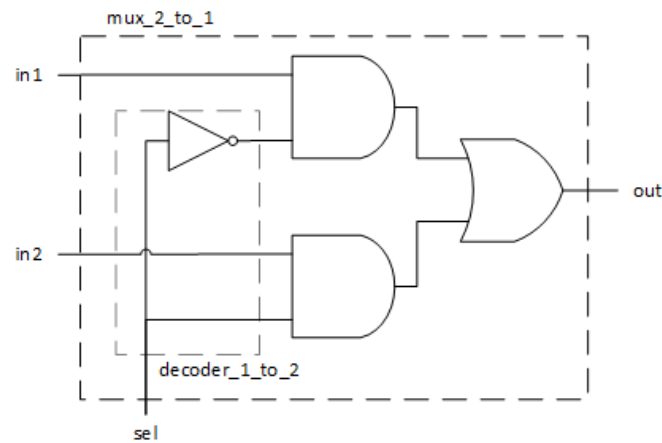


Figure 2: 2-to-1 multiplexer

Check the schematic in RTL Analysis to see the result circuit.

## 2.2 Exercise 2

a. Write a test bench for the *2-to-1 Multiplexer* in *Exercise 1* then use Vivado Simulator to simulate the design, students can use the given example source code. Let's analyse the structure of a test bench then point out the differences between an RTL code and a test bench code.

Change the **Radix**, **Format** of signals and use zoom tool to evaluate the waveform.

Check the **Tcl console** window to see output of **\$monitor** command.

b. Then, perform the Synthesis, compare the Synthesis's Schematic and the RTL Analysis's schematic.

c. After that, run the Implementation, check the Utilization report in Project Summary for used resources.

d. Add the Arty-Z7 constraint file to the project, assign pin for the design as follow:

- in1: btn[0], in2: btn[1]
- sel: sw[0]
- out: led[0]

then, generate bitstream file and program the FPGA to test the implemented circuit on board.

## 2.3 Exercise 3

a. Design a half-adder circuit using structural model.

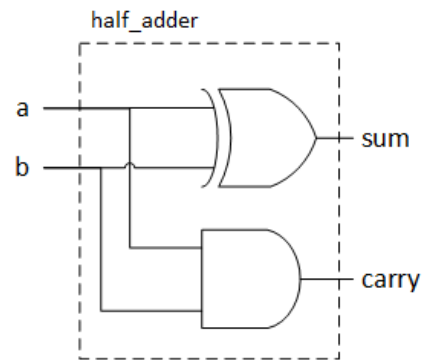


Figure 3: Half adder

**b.** Design a full-adder circuit using structural model. Reuse the half-adder module.

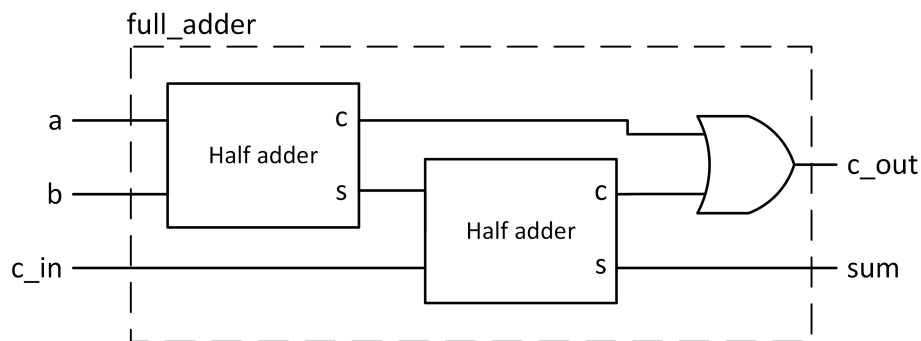


Figure 4: Full adder

Write a test bench to evaluate operations of the implemented full-adder circuit.

Test the implemented circuit on Arty-Z7 board using switches/buttons and LEDs.

*Bonus:* show the operands and sum on 7-seg LEDs.

**c.** Design a 4-bit ripple carry adder using structural model. Reuse the implemented full-adder.

Write a test bench to simulate the implemented circuit.

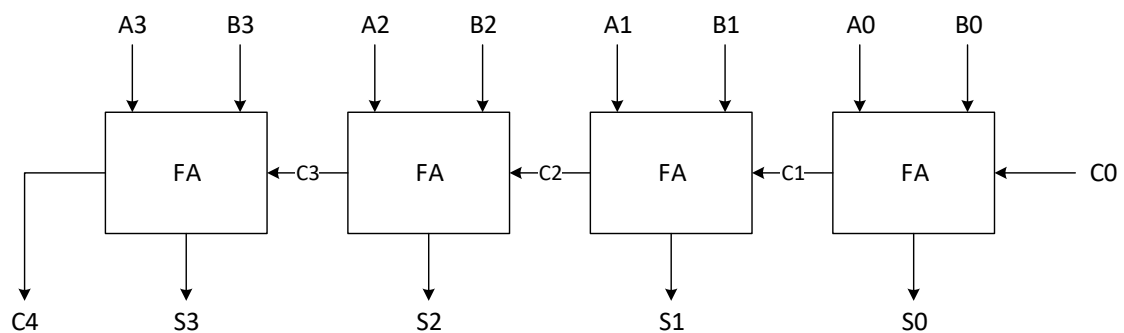


Figure 5: 4-bit ripple carry adder

## 2.4 Exercise 4

Give the 2-bit comparator circuit as Figure 6 with  $A = \{A1, A0\}$  and  $B = \{B1, B0\}$  are 2 2-bit input numbers,  $A\_gt\_B$  is active if  $A > B$ ,  $A\_lt\_B$  is active if  $A < B$  and  $A\_eq\_B$  is active if  $A = B$ .

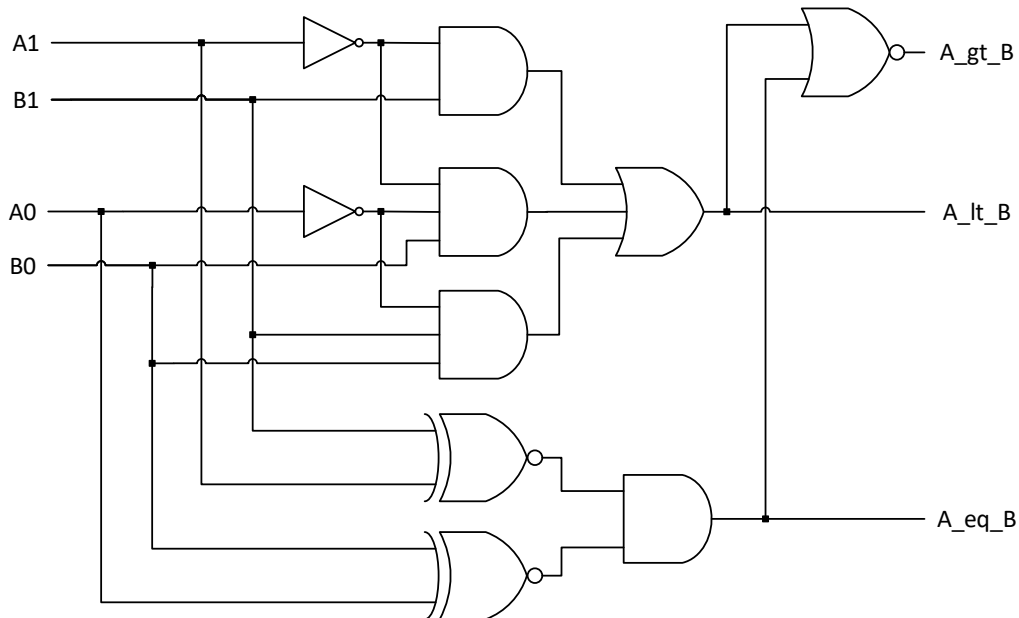


Figure 6: 2-bit comparator

Let's design a 4-bit comparator following below steps:

- Analyse the functions of each output of the 2-bit comparator, then determine the functions of 4-bit comparator outputs.
- Conceptualize the design of 4-bit comparator from 2-bit comparators (the 2-bit comparator can be partitioned into smaller blocks). Draw a block diagram that describes the idea.
- Draw a diagram to shows the designed circuit hierarchy.
- Implement the designed circuit using Verilog HDL structural model.
- Write a test bench to simulate the implemented circuit.