Hacettepe University Computer Engineering Department

Experiment 4 - Combinational Circuits in Verilog

Due date: 05/12/2021, 22:00:00 **Via: https://submit.cs.hacettepe.edu.tr/**

BBM233 Digital Design Lab - 2021 Fall

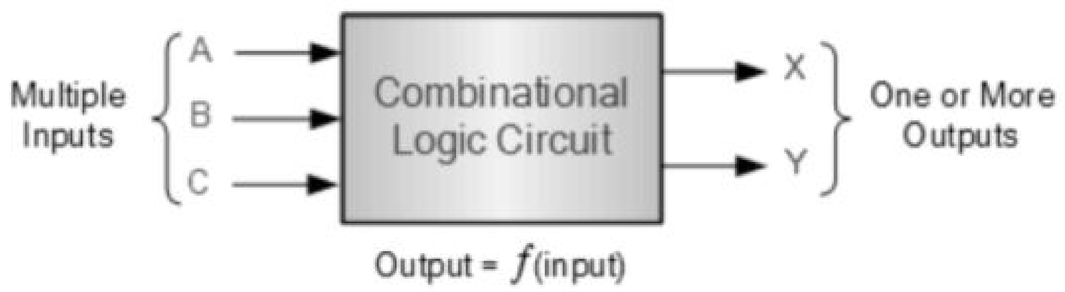
AIM

Designing and simulating combinational circuits in Verilog HDL.

BACKGROUND

Combinational Circuits

Combinational circuits are circuits whose outputs, at any instant of time, depend only on the present inputs (the combinational circuits do not use any memory elements). That is, the previous inputs or state of the circuit do not have any effect on its present state.



A combinational circuit performs a specific information-processing operation fully specified logically by a set of Boolean functions. The 'n' input variables come from an external source whereas the 'm' output variables go to an external destination. In many applications, the source or destination are storage registers.

Combinational Circuit Design Steps

To design a combinational circuit, start with the problem definition and use the following steps:

1. Identify the number of inputs and outputs of the circuit from the given specifications and assign them letter symbols.
2. Derive the truth table for each of the outputs based on their relationships to the inputs.
3. Simplify the Boolean function for each output (e.g. using Karnaugh Maps or Boolean algebra).
4. Construct the logic circuit diagram using Boolean functions obtained from the Step-3.

PLAGIARISM CONTROL NOTICE

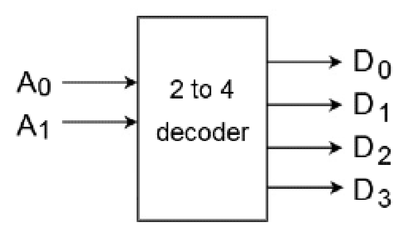
Students must implement their solutions individually. All submissions will be submitted to a plagiarism check. Any submissions that show a high level of similarity will be reported as plagiarism attempts to the ethics committee.

Part 0 - Decoders

A decoder is a combinational circuit that converts binary information from n binary inputs to a maximum of 2^n unique output lines. One of these outputs will be active **HIGH** based on the combination of inputs present, when the decoder is enabled. That means decoder detects a particular code. A decoder provides the 2^n minterms of n input variables.

2-to-4 Decoder

In a 2-to-4 decoder, two inputs are decoded into four outputs. One of these four outputs will be **HIGH** for each combination of inputs.



Input		Output			
A_1	A_0	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Experiment Steps

In this first part, you will design a **2x4 decoder** and implement it in Verilog HDL.

- Obtain Boolean equations for the outputs D_0 through D_3 from the given truth table and implement the 2x4 decoder using **dataflow design approach** (use **assign** statements to implement each output signal: D_0-D_3). Save your file as **decoder_2x4.v**
- Write a testbench and test for all 4 input combinations given in the truth table. Save your file as **decoder_2x4_tb.v**
- **You MUST download and use these starter code files before you start working! Do NOT change the I/O port names!**

Make sure to obtain a similar waveform which shows the correctness of your design.

Hacettepe University Computer Engineering Department

Experiment 4 - Combinational Circuits in Verilog

Due date: 05/12/2021, 22:00:00 **Via: https://submit.cs.hacettepe.edu.tr/**

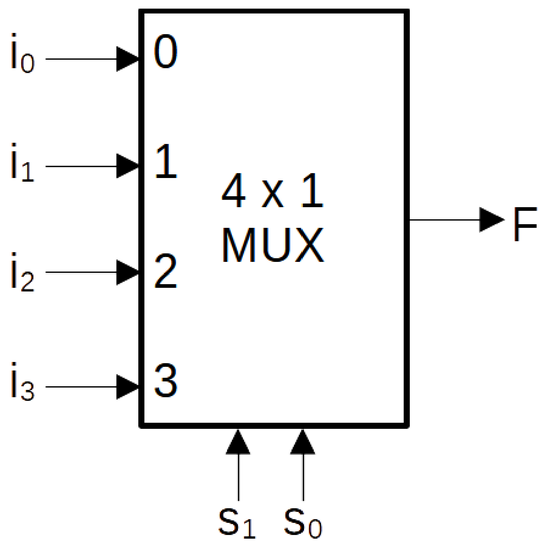
BBM233 Digital Design Lab - 2021 Fall

Part 1 - Multiplexers

A multiplexer (MUX) is a combinational logic circuit designed to switch one of several input lines through to a single common output line by the application of a control signal. A MUX has a maximum of 2^n data inputs. One of the inputs is connected to the output based on the value of the selection lines. There will be 2^n possible combinations of 1s and 0s since there are 'n' selection lines.

4-to-1 MUX

In a 4-to-1 MUX, only one out of four inputs is selected as the output based on a 2-bit select signal.



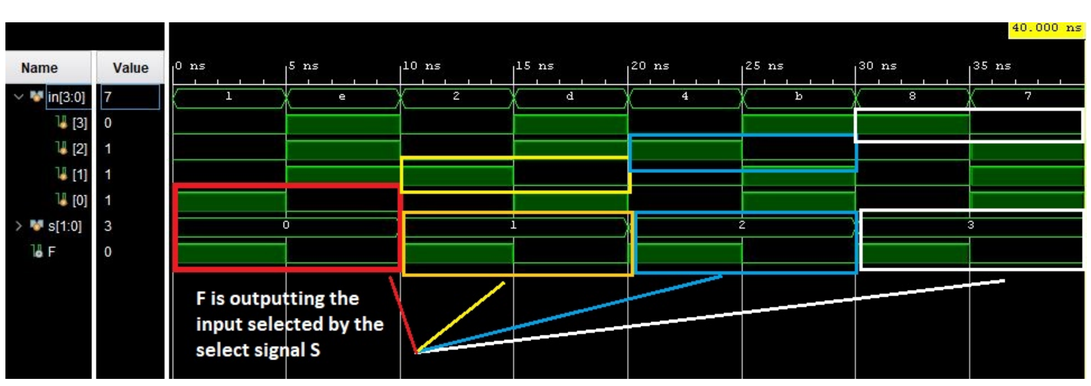
S_1	S_0	F
0	0	i_0
0	1	i_1
1	0	i_2
1	1	i_3

Experiment Steps

In this second part, you will design a **4-to-1 MUX** and implement it in Verilog HDL.

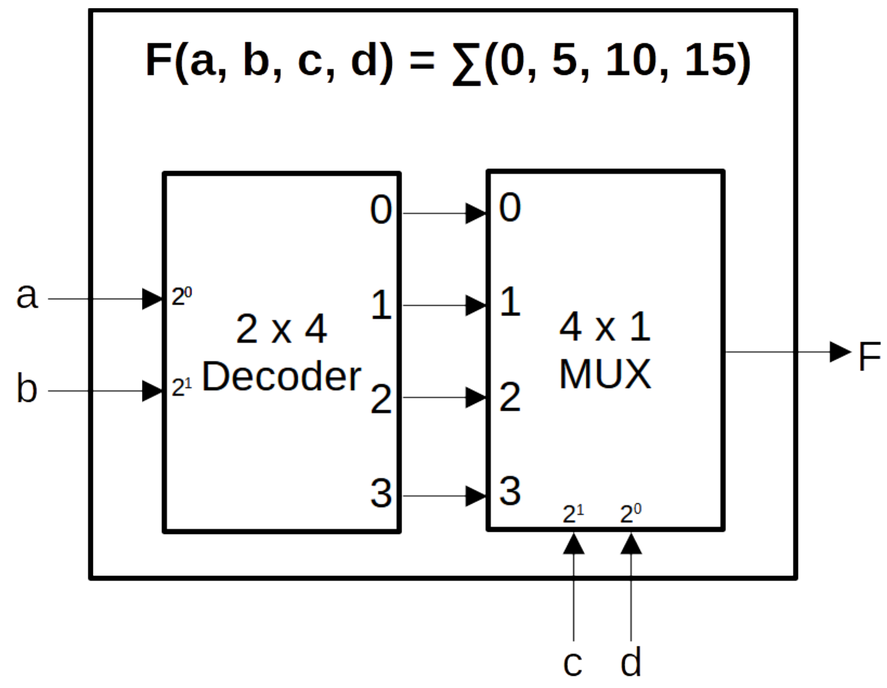
- Using the design specifications given in the truth table, implement a 4-to-1 MUX using **any design approach you want** (is it possible to use an approach with which we can have a single line of code implementation of the MUX?). Save your file as **mux_4x1.v**
- Write a testbench and test for all 4 input combinations given in the truth table. Save your file as **mux_4x1_tb.v**
- **You MUST download and use these starter code files before you start working! Do NOT change the I/O port names!**

Make sure to obtain a similar waveform which shows the correctness of your design. You should prove that the correct signal is chosen for all possible select signal inputs.



Part 2 - Combining Components

Combinational logic circuits can be connected together to form larger circuits. This is done by taking outputs from one circuit and using them as inputs to another. **Figure below shows a 2x4 decoder and a 4x1 MUX connected to implement a Boolean function** $F(a, b, c, d) = \sum(0.5.10.15)$**.**

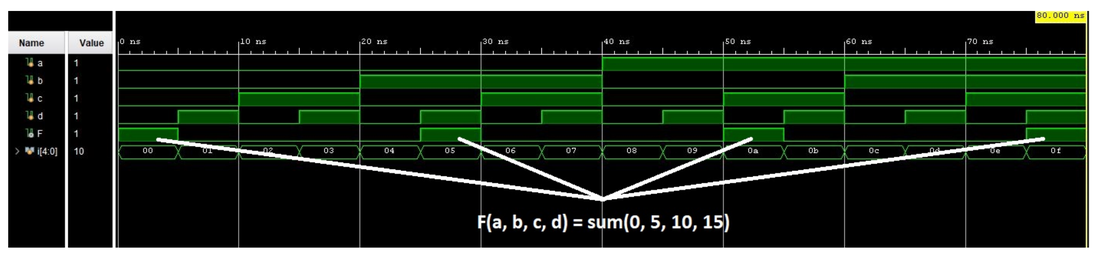


Experiment Steps

Implement the above figure in Verilog by the following steps:

1. Open a new Vivado project and add your **decoder_2x4.v** and **mux_4x1.v** files to the project.
2. Implement your module by providing the necessary Verilog code while referring to the above diagram. Use **structural design approach and explicit association** to connect the relevant ports to each other. Name your module as **circuit.v**
3. Test it by writing an appropriate testbench **circuit_tb.v** for all possible input cases.
4. **You MUST download and use these starter code files before you start working! Do NOT change the I/O port names!**

Make sure to obtain a similar waveform which shows the correctness of your design. And explain the obtained results in your report.



Hacettepe University Computer Engineering Department

Experiment 4 - Combinational Circuits in Verilog

Due date: 05/12/2021, 22:00:00 **Via: https://submit.cs.hacettepe.edu.tr/**

BBM233 Digital Design Lab - 2021 Fall

BONUS Part For Extra Credit

1. Redesign the circuit for $F(a, b, c, d) = \sum(0.5.10.15)$ using **only one inverter NOT gate and one 8-to-1 multiplexer**. Use the signals a, b, and c as the select signals s_2, s_1, and s_0 respectively.
2. Implement your design in Verilog and test it using your testbench circuit_tb.v.
3. Name your module as bonus.v and attach it to your submission.
4. Provide the waveform and your design with a detailed explanation in your report.

To receive bonus points, a well-structured Verilog code along with an explanatory report section is a **MUST**.

WHAT TO INCLUDE IN THE REPORT

This week you are expected to write a more detailed report. You are encouraged to use this Verilog Assignment Report Template and create your reports in LaTeX. We suggest using Overleaf platform for this. This is not mandatory, but make sure your report has all necessary parts and information. Check this file to see how the report should be structured: Verilog Assignment Report Template PDF

Your report needs to include the following:

1. Please, save your report as **name_surname.pdf**,
2. Include **the problem statement** (you can copy the information from these instructions) with figures of combinational circuits that are being implemented.
3. Include four Verilog code solutions for all specified modules:
   - decoder_2x4.v
   - decoder_2x4_tb.v
   - mux_4x1.v
   - mux_4x1_tb.v
   - circuit.v
   - circuit_tb.v
4. Include three waveforms obtained from each of the parts: **2x4 decoder**, **4x1 multiplexer** and your **circuit** module.
5. Explain the obtained results and how they show that your designs are working correctly. You may indicate the results directly on the waveforms.
6. You may use any resources, online or otherwise, but make sure to include the references in your report.

Submission via submit.cs

Submissions will be accepted via https://submit.cs.hacettepe.edu.tr/. Note that the deadline for submission of the report and all codes is 05/12/2021 at 22:00:00.

Your submission will include the Verilog codes and a report PDF and it must be in the following format to be accepted:

- b<studentID>.zip
  - decoder_2x4.v
  - decoder_2x4_tb.v
  - mux_4x1.v
  - mux_4x1_tb.v
  - circuit.v
  - circuit_tb.v
  - report.pdf

Note that only ZIP archives are accepted!

Automatic grading [VERY IMPORTANT!]

Your submissions will be graded using an automatic grading script. They **will not be** white-box tested. So, you are expected to **match the given wave forms 100%**, to receive full-credit.

You MUST download and use these starter code files before you start working! Do NOT change the I/O ports (names, order, bit-width, etc.)!

Grading

Report: 20%
Verilog codes: 80%
Bonus part: 10%