Department of Computer Science
Bldg 315
900 Fifth Street
Nanaimo, BC
V9R 5S5
Canada

Lab #3
Advanced Combinational Logic

Date Due: Saturday, October 5, 11:59 pm **Total Marks: 32**

General Instructions

- **This lab assignment is individual work. You may discuss questions and problems with anyone, but the work you hand in for this assignment must be your own work. Put your name, student number and instructor's name on the cover page of your report. Failure to follow these conventions will result in your grade deduction. Do not submit folders, or zip documents, even if you think it will help.**
- **Assignments must be submitted to VIULearn.**
- **VIULearn will not let you submit work after the assignment deadline. It is advisable to hand in each answer that you are happy with as you go. You can always revise and resubmit as many times as you like before the deadline; only your most recent submission will be graded.**
- **Partial credit will be given as appropriate, so hand in all attempts.**

1. Objectives

i. Familiarization with Full Adder
ii. Familiarization with Parity Generator and Checker
iii. Construct Full-Adder using NAND and XOR only logics
iv. Construct NAND equivalent for combinational circuit
v. Construct NOR equivalent for combinational circuit

2. Health and Safety

This section tries to highlight the health and safety concern while conducting lab activities. Any laboratory environment may contain potentially hazardous conditions to a person's health if not handled appropriately. The Department of Computer Science laboratory Bld:315/115 obviously has electrical potentials that may be lethal and must be treated with respect. One of our objectives is to educate all laboratory users to handle laboratory materials and situations safely, thereby ensuring a safe and healthy experience for all. Watch for posted information in and around the laboratories, and on the class website.
https://adm.viu.ca/sites/default/files/viu_safety_design_for_facilities_standard_draft.pdf

3. Background

3.1 Boolean Function Representation

A Boolean expression is an expression consisting of variables, constants (0-false and 1-true) and logical operators which result in true or false. A Boolean function is an algebraic form of a Boolean expression. The Boolean expressions are standardized using two standard forms.

i. SOP form – Sum Of Products form
ii. POS form – Product Of Sums form
iii. Canonical form

There are two types of canonical forms:

i. Sum-of-min terms or Canonical SOP
ii. Product-of-max terms or Canonical POS

SOP form representation is the most suitable to use in FPGA (Field Programmable Gate Arrays)[1].

3.2 Half-Adder

The circuit shown in Figure 3-1 is known as a half-adder. It has two 1-bit unsigned binary inputs, A and B, and two 1-bit outputs, sum (S) and carry (C).

[1] https://www.electronicshub.org/

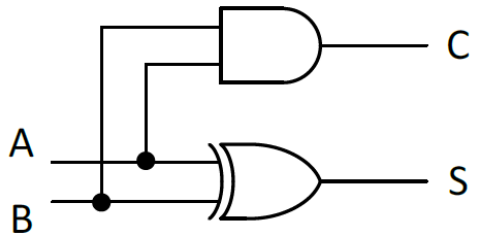

Figure 3-1: Half-Adder

3.3 Full-Adder

Two half-adders can be combined to form a full adder as shown in Figure 3-2. The full-adder adds two input bits (x_i and y_i) as well as a "carry-in" bit (c_i) and generated sum (s_{i+1}) and "carry-out" (c_{i+1}). N full-adders can be used to add two n-bit input numbers $x_{n-1}x_{n-2}\cdots x_0$ and $y_{n-1}y_{n-2}\cdots y_0$ (set c_0=0).

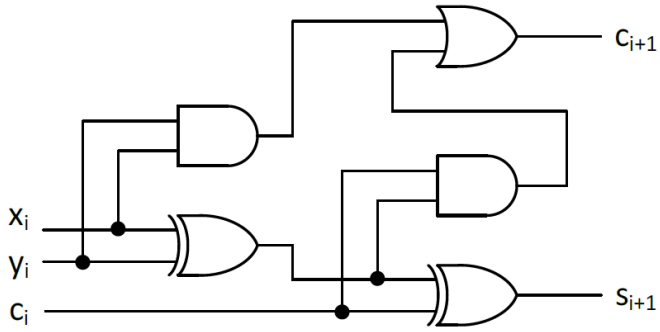

Figure 3-2: Full-Adder

4. Pre-Lab Exercises

4.1 Boolean Function

4.1.1 Use algebraic manipulation to prove

3

i. $x + yz = (x + y) \cdot (x + z)$

ii. $xy + yz + \bar{x}z = xy + \bar{x}z$

iii $\overline{\left((a+b)+\bar{c}+\bar{d}\right)\cdot\left(\overline{a\cdot b}\cdot\bar{c}\cdot\bar{d}\right)} = \overline{(a+b)}\cdot c\cdot d + a\cdot b + c + d$

4.1.2 Use algebraic manipulation to find the minimum sum-of-products (SOP) expression for the function:

2

- $f = x_1\overline{x_2}\,\overline{x_3} + x_1x_2x_4 + x_1\overline{x_2}x_3\overline{x_4}$
- $f(x_1, x_2, x_3) = \sum m(1, 3, 4, 6, 7)$

4.1.3 Use algebraic manipulation to find the simplest products-of-sums (POS) circuit that implements the function:

1

$f(x_1, x_2, x_3) = \prod M\,(0, 2, 5)$

2

4.1.4 Find the minimum-cost SOP and POS forms for the function

$f(x_1, x_2, x_3) = \Sigma m(1, 2, 3, 5)$

4.2 Full Adder Implementation

2

For this lab, you have NAND, NOR, and XOR gates. Show how the full adder circuit shown in Figure 3-2 can be implemented using just NANDs and XORs without increasing the total number of gates used (i.e., just using five gates).

4.3 Parity Generator

2

Review the previous lab's section on 4-input XOR. Furthermore, read over the section regarding "Parity bit" from Wikipedia.org. Based upon this information, is the circuit shown in section 7.2 generating an "even" or "odd" parity bit for the input 5-bit binary number?
What is the purpose of a parity generator? Consider that the parity bit is added to the input number to form a 6-bit binary number while sending from the transmitter to the receiver. Give the reasons behind your answer.

5. Equipment Required

Power supply, 7400 Quad 2-Input NAND Gate, 7402 Quad 2-Input NOR Gate, 7486 Quad 2-input XOR, Breadboard, Digital Circuit Evaluator, Wiring Kit

6. Debugging (or What to Try When Things Aren't Working)

There are several things/procedures you should use to debugging circuits when things are not working correctly. These include (but are not limited to):

- Draw a copy of the circuit on paper and label all PINs to aid in debugging.
- Check that all component pins are correctly inserted in the breadboard (sometimes they get bent underneath a component).
- Make sure that components are not "misaligned" in the breadboard (e.g., off by one row).
- Try a different section in the breadboard (in case there is a bad internal connection).
- Measure key points in the circuit for proper voltage/waveform (i.e., divide-and-conquer).
- Make sure the power and input signals are turned on.

7. Lab Procedure

7.1 Full Adder

4

i. Construct a full-adder circuit on your breadboard similar to Figure 3-2 but using only NAND and XOR gates (from Prelab 4.2).

ii. You should include a picture of your breadboard with the completed circuit in your lab report.

iii. Fill in the appropriate Truth Table and comment if the circuit is working as expected.

REQUIRED: Demonstrate the Digital Circuit Evaluator to me to prove that your full-adder circuit is working. Make sure that I record that you successfully demonstrated the full-adder circuit functioning.

7.2 Parity Generator

6

i. Construct the five-bit "parity" generator circuit shown in Figure 7-1 on your breadboard.

ii. You should include a picture of your breadboard with the completed circuit in your lab report.

iii. Fill in the appropriate Truth Table and comment if the circuit is working as expected.

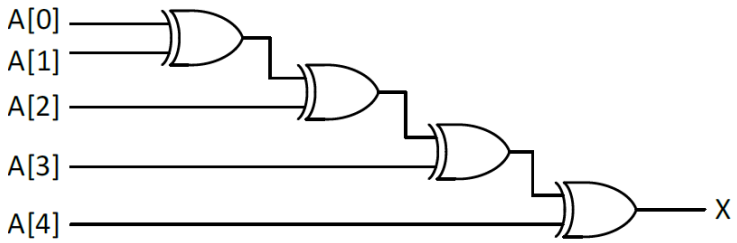

Figure 7-1: Parity Generator

iv. For the given parity generator, there is always a parity checker at the receiver end to detect errors in the received codes. Now, what will be the expression for the accompanying parity checker circuit? Explain it considering the output that the checker yields for a different combination of inputs.

7.3 NAND Equivalent Circuit

4

i. Construct the NAND-only version of the circuit that implements the function

$f(x_1, x_2, x_3) = \Sigma m(1, 3, 4, 6, 7)$

ii. Write minimum-cost SOP expression for the function.

iii. Show schematic (can use 3 input-NAND gate)

iv. Include a picture of your breadboard with the completed circuit in your lab report.

v. Comment if the circuit is working as expected or not.

7.4 NOR Equivalent Circuit

4

i. Construct the NOR-only version of the circuit that implements the function
$f(x_1, x_2, x_3) = \Sigma m(1, 3, 4, 6, 7)$

ii. Write minimum-cost POS expression for the function.

vi. Show schematic ((can use 3 input-NOR gate)).

iii. Include a picture of your breadboard with the completed circuit in your lab report.

iv. Comment if the circuit is working as expected or not.

Note: No need to include pin-out sheets for this lab as well. I want you to include the PINs on the schematic diagrams rather. Provide the functional diagram/pinning information of ICs in the deliverables or Appendix.