

University of Western Ontario, Computer Science Department
CS3350B, Computer Organization

Assignment 2

Due: February 14, 2020

General Instructions: This assignment consists of 3 pages, 4 exercises, and is marked out of 100. For any question involving calculations you must provide your workings. You may collaborate with other students in the class in the sense of general strategies to solve the problems. But each assignment and the answers within are to be solely individual work and completed independently. Any plagiarism found will be taken seriously and may result in a mark of 0 on this assignment, removal from the course, or more serious consequences.

Submission Instructions: The answers to this assignment are to be submitted on OWL as a **single PDF file**. Ideally, the answers are to be typed. At the very least, clearly *scanned* copies (no photographs) of hand-written work. If the person correcting your assignment is unable to easily read or interpret your answer then it may be marked as incorrect without the possibility of remarking.

Useful Facts:

Logism is a good tool for drawing circuits. <http://www.cburch.com/logisim/>

$$1 \text{ GHz} = 1 \times 10^9 \text{ Hz}$$

$$1 \text{ byte} = 8 \text{ bits}$$

$$1 \text{ Kbyte (KB)} = 1024 \text{ bytes}$$

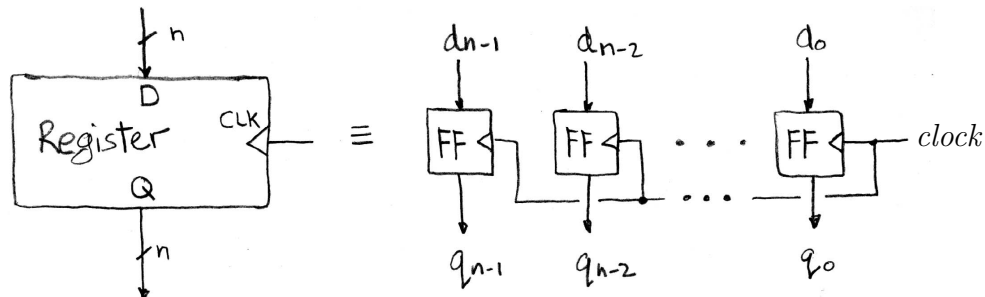
Exercise 1.

(a) [10 Marks] Re-write the following Boolean expression using only NAND.

$$ab + \bar{c}$$

(b) [10 Marks] Draw a circuit, using only NAND gates of arity 2, which implements a NAND gate with an arity of 3. That is, the expression $\overline{(ABC)}$.

Exercise 2. [10 Marks] Draw a 4-bit Serial In, Serial Out register using SR flip-flops. For example, the below diagram represents a Parallel In, Parallel Out n -bit register using D flip-flops.



Exercise 3. [30 Marks] Simplify the following Boolean expressions completely into an expression of reduced sum of products (*not* disjunctive normal form; DNF implies every variable must appear in every product, that is not the case here). For each simplification step, explain which Boolean law is applied to achieve that step. For full marks, do not apply multiple laws per step without explicitly stating all laws.

(a) $xy\bar{z} + xy + (\bar{x}z + y)z$

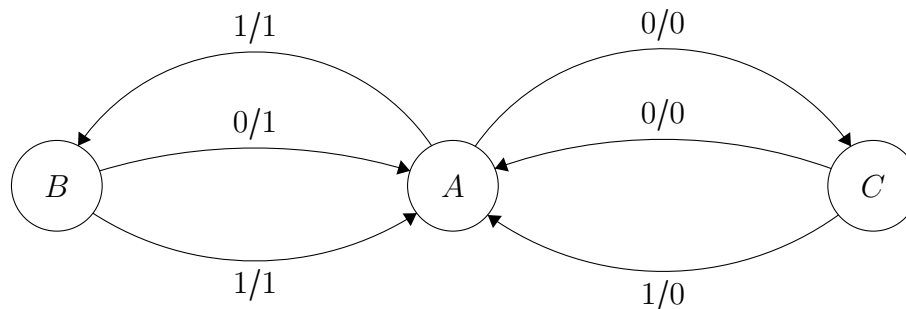
(b) $\bar{x}yz + x\bar{y}z + xy\bar{z} + \bar{x}y\bar{z} + x\bar{y}\bar{z} + \bar{x}\bar{y}z.$

(c) Give a Boolean circuit, using only AND, OR, and NOT gates, implementing the simplified expression from part (b).

Exercise 4. A bit-stream converter is a simple circuit that reads a sequence of bits, one at a time, and outputs another sequence of bits depending on its specification. Consider the following FSM diagram, it implements a *bit-stream converter* with the specification:

- $00 \rightarrow 00$
- $01 \rightarrow 00$
- $10 \rightarrow 11$
- $11 \rightarrow 11$

For example, the 2nd line implies that when the two-bit sequence “01” is read “00” is output.



(a) [12 Marks] Create a truth table which encodes this FSM. (Hint: encode each state A,B,C using some unique binary number).

(b) [8 Marks] For each output in your truth table from part **(a)**, give its disjunctive normal form. (Hint: handle each bit of the binary number encoding the state separately).

(c) [20 Marks] Draw the *combinational logic circuit* implementing this FSM. That is, you do *not* need to show the registers or feedback loop, only the circuit encoding the logic. Use logic gates with an arity of at most 2. (Hint: this circuit should have 3 inputs and 3 outputs).