
UIT2201: CS & the IT Revolution Tutorial Set 10 (Fall 2016)

**(D-Problems discussed on Friday, 21-Oct-2016)
(Q-Problems due on Tuesday, 25-Nov-2016)**

Discussion Problems: -- Prepare (individually) for tutorial discussion.

T10-D0: [IMPORTANT READINGS]

Read and UNDERSTAND Section 4.5 (pp. 179-182) of [SG3] on control circuits (namely, multiplexors and decoder circuits).

THEN, do T10-D4, D5.

T10-D1: [Logic Design, [From Quiz, Spring 2008, Q2]

Click here -- | [Fig-T10-D1-2008-Sp-Quiz.pdf](#) |

T10-D2: [Design of the 3-bit-Majority Circuit]

Build a 3-input "*majority*" circuit. This is a circuit that has 3 inputs A, B, C and one output M. The value of M is 1 *if and only if two or more* of its inputs are 1; otherwise, the the value of M is 0. (For example, when A, B, C are 0, 1, 1, respectively, then M (the output) is 1. And when A, B, C are 0, 1, 0, respectively, then M (the output) is 0.)

(a) First draw out the truth table for the function M (with inputs A, B, C).

(b) Apply the "sum of products" algorithms to get the logic function of M.

(c) Simplify the logic function as much as you can and minimize the #gates used.

(d) Draw the logic circuit that implements M based on your answer for (c) above. How many 2-input AND/OR gates did you used in all? (We usually don't count the number of NOT gates.)

(Note: This circuit is used in **fault-tolerant computing** -- see Prob 19 of Ch.4. of [SG3] for details.)

T10-D3: [Design of the 1-ADD circuit]

The 1-ADD circuit is used in the design of the Adder circuit. The truth table of the two outputs are given in the lecture notes. Using the "sum of products" algorithm and simplification to get a good design of the 1-ADD circuit (minimizing the total number of OR/AND gates). Show the optimized logic function and the circuit diagram for the 1-ADD circuit. How many OR and AND gates does your design use?

T10-D4: (Multiplexor Circuits) [2- and 4-input MUXes]

(a) [2-input MUX] Read up "design of multiplexor circuit" in Sec 4.5 of [SG3] & lecture notes. Explain the working of a 2-to-1 multiplexor circuit with input D0 and D1. Explain the role of the "selector" S0.

(b) [4-input MUX] (Prob. 23 of Ch.4 of [SG3].)

Design a *4-input multiplexor circuit* (or a 4-to-1 MUX). Use the design of the 2-to-1 MUX as a guide. This multiplexor has 4 input lines C0, C1, C2, C3 and requires two selector lines S1, S0.

(For simplicity, you can make use of 3-input OR/AND gates in your circuits. Think of them as *higher-order complex gates* that are decomposed into two 2-input OR/AND gates.)

T10-D5: (Decoder Circuits)

Read up the design of decoder circuits in Ch4.5 of [SG3] & the lecture notes.

- (a) Explain the working of a 2-to-4 decoder circuit given in lecture notes (Figure 4.29 of [SG3]).
 (b) What does a 1-to-2 decoder circuit look like?

Problems to be Handed in for Grading by the Deadline:

(Note: Please submit *hard copy* to me. Not just soft copy via email.)

T10-Q1: (10 points) (Simple Logic Circuit Design)

(a) (5 points) (Modified from Problem 18 in p.185 (Ch. 4) of [SG3]).

Using the "sum of products" circuit construction algorithm of Section 4.4.2 (also given in the lecture notes), design a circuit using only AND, OR and NOT gates to implement the following truth table.

a	b	Output
0	0	1
0	1	0
1	0	1
1	1	1

(b) (5 points) (Making a "7-input" AND gate with many 2-input AND gates)

Explore different ways to design/build a "7-input" AND gate from standard 2-input AND gates.

Assume that each standard 2-input AND gate has a time delay of d (namely, it takes time d seconds for input signal to propagate to the output signal).

(i) How many 2-input AND gates are needed?

(ii) The different designs (circuits) for the 7-input AND gates will have different delays (time taken to produce the final output). Give a design for the 7-input AND gate that has the *maximum delay*. What is this maximum delay (in terms of d)?

(iii) Give a design for the 7-input AND gate that has the *minimum delay*. What is this minimum delay (in terms of d)?

T10-Q2: (10 points) (Logic Design) [modified from previous MidTerm]

Consider the following logic formula: $F = X*Y + (\sim X)*Z + X*(\sim Y)*Z$

where we use "*" to denote logical **AND** function, "+" to denote logical **OR** function, "~" to denote logical **NOT** function.

(a) Give the truth table for F .

(b) Using the truth table, give a logical formula for F as a sum of products.

(c) Simplify the formula for F as much as you can.

(For this step, you may want to explore the solution space a bit to find good simplifications.)

(d) Give a logic circuit for F using your answer for (c) above.

(Use only 2-input gates in your circuit.)

T10-Q3: (10 points) [8-input multiplexor circuit]

Extend T10-D1 to design an 8-input MUX. It has 8 input data lines $C_0, C_1, C_2, C_3, C_4, C_5, C_6, C_7$ and has three selector lines S_2, S_1, S_0 . (For simplicity, you can make use of k -input OR/AND gates.) State how many 2-input AND/OR gates you use. (Convert k -input AND/OR gates to 2-input ones.)

[*Helpful Pointer:* In these circuit diagrams, it is useful to draw all the input signals (both X and $\sim X$) as vertical lines on the left side of the circuit diagrams -- as shown in Figures 4.29 or 4.26. This makes your circuit neater and easier to "debug" or spot mistakes.]

T10-Q4: (10 points) [3-to-8 Decoder Circuit]

Using T10-D2 as a guide, design and draw the circuit diagram for a 3-to-8 decoder circuit. Label your output lines $C_0, C_1, C_2, \dots, C_7$ and your input lines (*decoder*) lines s_2, s_1, s_0 is such a way that if the binary number represented by $(S_2 S_1 S_0)_2$ is k , then the line C_k is selected.

State how many 2-input AND/OR gates you use.

A-Problems: OPTIONAL Challenging Problems for Further Exploration**A8-2016: (Circuit Optimization) Challenge Work 1 in p185 of [SG3].**

The full adder circuit of Figure 4.27 (pp.176-177) is correct, but its design is not optimized -- it uses too many gates, and so too many transistors -- 2,208 transistors. Use the boolean logic laws (also called transformation rules) covered in lectures to improve the design and minimize the number of transistors used. How many gates (transistors) are used in your *optimized* design?

A9-2016: (Faster ADD circuits)

The n -bit ADD circuit given in Ch-4.4.3 of [SG] is called a *ripple* carry adder since the carry bit ripples through the individual 1-bit ADD circuits, one step at a time from the 0^{th} bit to the $(n-1)^{\text{st}}$ bit. It therefore takes a total of n time steps to complete the addition, and for large n (say $n=32$ or 64), this makes for a rather slow n -bit ADD circuit.

Design a faster n -bit ADD circuit that takes only $\lg n$ time steps.

(**Note:** You may have to use more gates to achieve this -- trade off area for speed.)

UIT2201: CS & IT Revolution; (Fall 2016); A/P Leong HW