COMP2611 Spring 2016 Homework #1

(Due Monday Mar 7 5PM)

Note:

- **This is an individual assignment; all the work must be your own.**
- **Submit a pdf file via CASS. Name the pdf file as <*your_stu_id*>.*pdf* (without the brackets)**
- **Use Logisim to draw all digital circuits, and paste the image into your homework.**

Question 1: Data representation, conversion between bases

a) Conversion between decimal, binary and hexadecimal. Assume we're dealing with 16 bits signed numbers.

146 $_{(10)}$
F0D1 $_{(16)}$

b) Do the same for the same values for 32bits signed numbers.

c) Do the same for 32 bits unsigned numbers.

d) Find the IEEE754 single-precision floating-point representation for the following decimal numbers. Briefly show your steps.

i. 3.75

ii. -313.3125

e) What decimal numbers are represented by the following IEEE754 single-precision floating-point representations? Write your answer in decimal (scientific notation is fine). Briefly show your steps.

i. 0 00000000 11100000000000000000000

ii. 1 10000001 00011000000000000000000

Question 2: Boolean algebra and Combinational logic

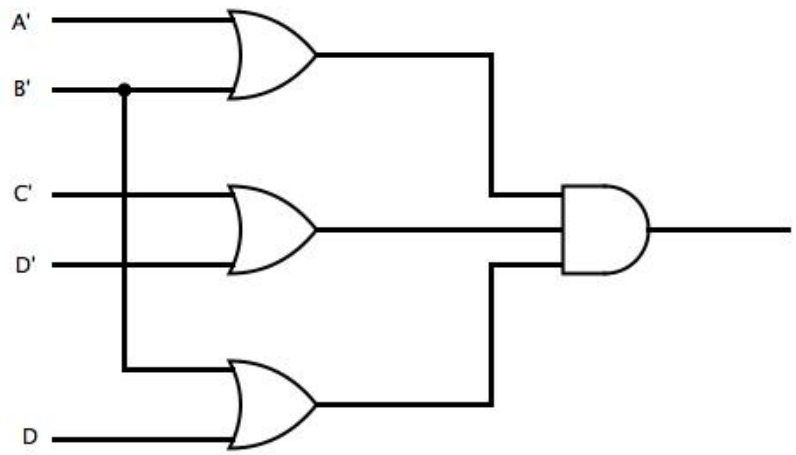
a) Simplify the following logic equation using the laws of Boolean algebra only.

$$\overline{ABC + \bar{A}B}$$

b) Simplify the following logic equation using K-map.

$$\bar{A}BC + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + A\bar{B}C + ABC$$

c) Given the following logic circuit, give the corresponding logic equation in product of sums form then in sum of products form, and finally, using K-map, simplify this latter. (Note that the prime notation represents inversion)



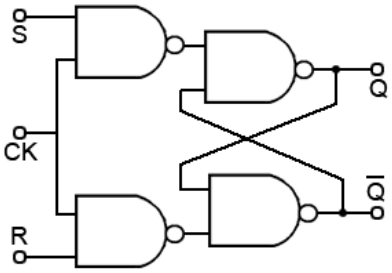
Question 3: Sequential logic

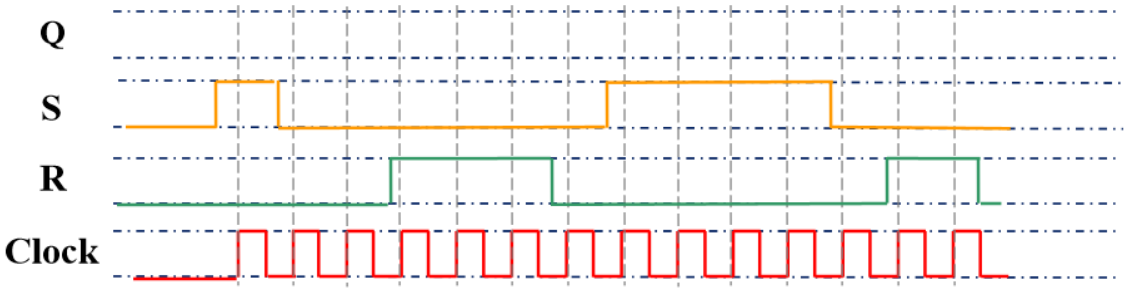
During the lecture, we explained how to build an unclocked SR-latch from a pair of cross-coupled `NOR` gates. Starting from that circuit show how to:

a) Construct an SR-latch with one `AND` gate, one `OR` gate, and as many `NOT` gates as needed, and inputs `S` and `R` are active if deasserted (i.e., Set when `S`=0, Reset when `R`=0). Write down the truth table and draw your circuit.

b) Construct an SR-latch with one AND gate, one OR gate and as many NOT gates as needed, and inputs `S` and `R` are active if asserted (that is, Set when `S`=1, Reset when `R`=1). Write down the truth table and draw your circuit.

Now consider the following clocked SR-latch built with NAND gates. Draw the output `Q` versus `clock`, `Set`, and `Reset` inputs, assuming inputs `Set` and `Reset` are active in the high state (i.e., when asserted).


S
CK
R
Q
Q̄


Q
S
R
Clock