Name:_____

Homework #3 Computer Organization
Due: Feb. 13, 2015 (Friday) by 4 PM

1. Draw the circuit (using AND, OR, and NOT gates) to implement an 16-input to 1-output multiplexer (MUX). Your MUX should have 4 control wires (c_3, c_2, c_1, c_0) to select which input is switched to the single output. (You can use ". . ." to avoid drawing the whole MUX, but show enough to demonstrate your understanding of MUXs)

2. Recall that the sum-of-products (SOP) Boolean formula for the carry-out (c_{i+1}) of a n-bit adder was :

1-bit adder: $c_{i+1} = x_i y_i + x_i c_i + y_i c_i$.

2-bit adder: $c_{i+1} = x_i y_i + x_i (x_{i-1} y_{i-1} + x_{i-1} c_{i-1} + y_{i-1} c_{i-1}) + y_i (x_{i-1} y_{i-1} + x_{i-1} c_{i-1} + y_{i-1} c_{i-1})$
$= x_i y_i + x_i x_{i-1} y_{i-1} + x_i x_{i-1} c_{i-1} + x_i y_{i-1} c_{i-1} + y_i x_{i-1} y_{i-1} + y_i x_{i-1} c_{i-1} + y_i y_{i-1} c_{i-1}$

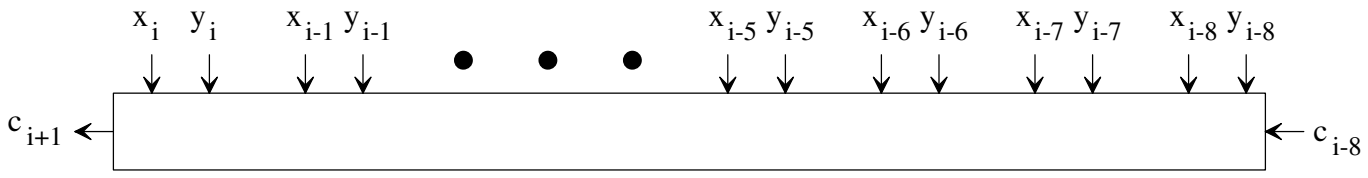
3-bit adder: $c_{i+1} = x_i y_i + x_i$ (7 product terms of 2-bit adder) $+ y_i$ (7 product terms of 2-bit adder)
$= x_i y_i + x_i x_{i-1} y_{i-1} +$ ... (12 product terms omitted) $+ y_i y_{i-1} y_{i-2} c_{i-2}$

4-bit adder: $c_{i+1} = x_i y_i + x_i$ (15 product terms of 3-bit adder) $+ y_i$ (15 product terms of 3-bit adder)
$= x_i y_i + x_i x_{i-1} y_{i-1} +$ ... (28 product terms omitted) $+ y_i y_{i-1} y_{i-2} y_{i-3} c_{i-3}$

a) Complete the following table showing the gate delays for different types of adders **assuming a 9-input limit into any gate.**

Type of Adder	# of product terms in SOP expression to be OR'ed	Most # of inputs in any of the product terms	# gate delays due to product/AND terms	# gate delays due to sum/OR	Gate Delay per Adder	Gate Delays for 32-bit ripple adder using adders of this type
1-bit	3	2	1	1	2	32 x 2 = 64
2-bit	7	3	1	1	2	16 x 2 = 32
3-bit	15	4	1	2	3	10 x 3 + 2 = 32
4-bit	31	5	1	2	3	8 x 3 = 24
5-bit	63	6				
6-bit	127	7				
7-bit	255	8				
8-bit	511	9				
9-bit	1,023	10				

b) Consider the SOP Boolean formula for the carry-out (c_{i+1}) of a 9-bit adder:



Give an example of each of the following:

i) a product term with only two terms

ii) a product term with 10 terms

Name:_____

3. Suppose we have a register file with the following specifications (see Memory Supplement in Lecture 8 on the eLearning system and in your CopyWorks course packet):

- 16 registers numbered from 0 to 15
- each register has 24-bits (a strange amount I know)
- one write port
- two read ports

a) What how many bits(/"wires") would be need for each of the following?

- data to be written for a write port
- specifying the register number of a read port
- specifying the register number of a write port
- output read from a read port

b) How many write enable wires would be needed for the whole register file?

c) How many decoders would be needed in the implementation of the whole register file? Explain how you arrived at that number and specify the type of decoders (i.e., # of inputs and # of outputs for each decoder)

d) How many MUXs would be needed in the implementation of the whole register file? Explain how you arrived at that number and specify the type of MUXs (i.e., # of inputs and # of outputs for each MUX)

4. Complete the below diagram of a 4-bit register so that it is able to perform the following operations:

- parallel read/output of all bits (just look at the Q values)

Control the MUXs using the following codes:

00_2 - parallel write/load/input of all bits

01_2 - circular shift left one bit position (values shifted out of most-significant bit is shifted into the least-significant bit)

10_2 - arithmetic shift right (sign-extend the most-significant bit)

11_2 - logical shift right **two** bit positions (each bit shifted out of least-significant bit is lost and a "0" is shifted into the most-significant bit)

Note: For each D-flip flop, the output of a MUX is used as the D-input as shown below.