## Homework 4 - Due at Lecture on Monday, March 5

*Primary contact for this homework: Kamlesh Prakash [kprakash2@wisc.edu]*
You must do this homework in groups of **two**. Please hand in ONE copy of the homework that lists the **section number**, **full** names (as they appear in Learn@UW) and **UW ID** numbers of all students. You must **staple** all pages of your homework together to receive full credit.

### Problem 1 (6 points)

A logic circuit has two 2-bit unsigned binary numbers X[1:0] and Y[1:0] as the inputs and it has two 1-bit outputs. One of the outputs is EQUAL and the other is XGTY. The EQUAL output is true when X[1:0]=Y[1:0] and the XGTY output is true when X[1:0] > Y[1:0].

**a)** Write a truth table for these two functions.

**b)** Determine the needed logic equations.

**c)** Draw the Gate level circuit for EQUAL using AND, OR and NOT gates.

**a) Truth Table**

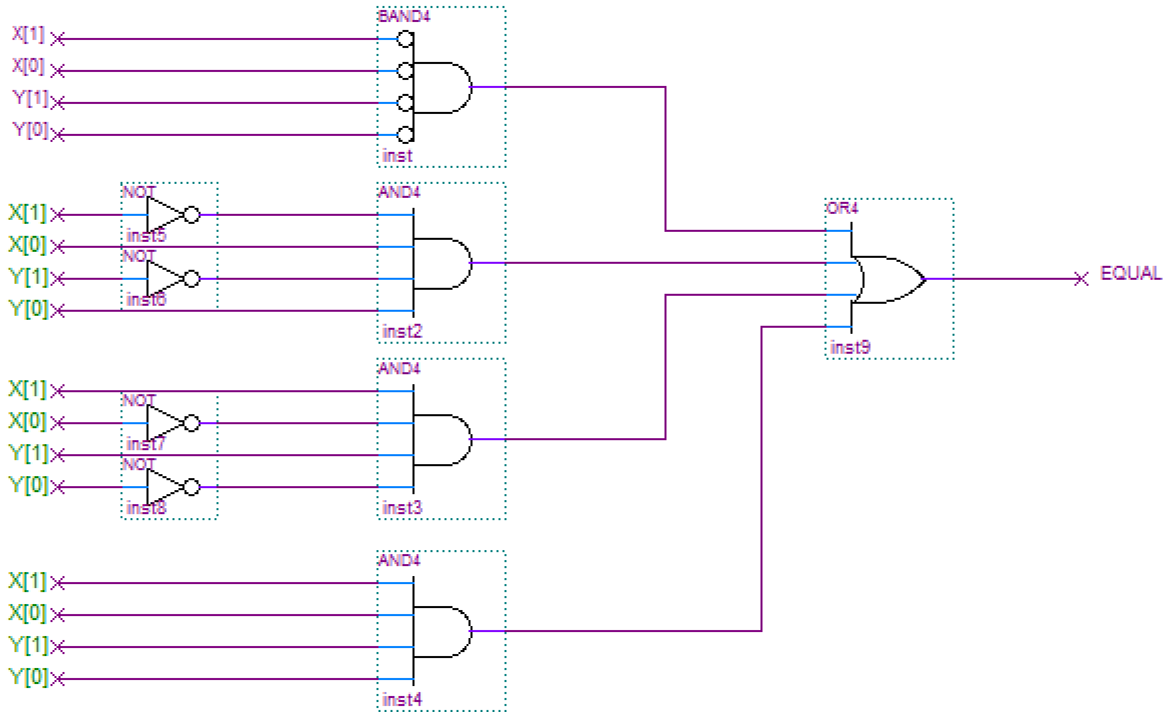
| X[1] | X[0] | Y[1] | Y[0] | EQUAL | XGTY |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 |

**b)**

**EQUAL =** $\overline{X[1]}\,\overline{X[0]}\,\overline{Y[1]}\,\overline{Y[0]} + \overline{X[1]}X[0]\overline{Y[1]}Y[0] + X[1]\overline{X[0]}Y[1]\overline{Y[0]} + X[1]X[0]Y[1]Y[0]$

**XGTY =** $\overline{X[1]}X[0]\overline{Y[1]}\,\overline{Y[0]} + X[1]\overline{X[0]}\,\overline{Y[1]}\,\overline{Y[0]} + X[1]\overline{X[0]}\,\overline{Y[1]}Y[0] + X[1]X[0]\overline{Y[1]}\,\overline{Y[0]} + X[1]X[0]\overline{Y[1]}Y[0] + X[1]X[0]Y[1]\overline{Y[0]}$

## C) Gate level circuit for EQUAL:




**Problem 2 (4 points)**

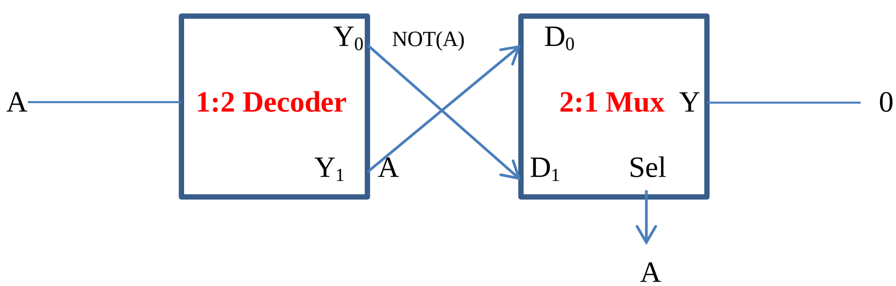
**a)** Using only one 1-to-2 decoder and one 2-to-1 multiplexer, draw a circuit that always outputs a 0.

**b)** Using only one 1-to-2 decoder and one 2-to-1 multiplexer, draw a circuit that always outputs a 1.

**Note: Inputs cannot be set to constant values (0, 1) and no other gates should be used. Use decoders and mux as blocks.**

**Hint:** Input to the 1-to-2 decoder is some variable "A" and the outputs of the decoder are fed as inputs to the 2-to-1 mux.

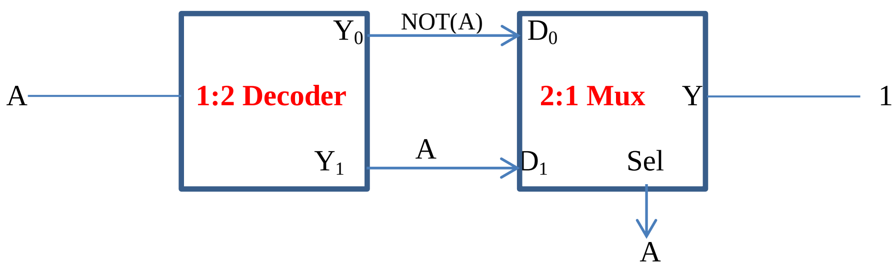
**a)**




**If A=0 => $D_0$ is selected => Y = A = 0**

**If A=1 => $D_0$ is selected => Y = NOT(A) = 0**

**b)**




**If A=0 => $D_0$ is selected => Y = NOT(A) = 1**
**If A=1 => $D_1$ is selected => Y = A = 1**

**Problem 3 (3 points)**
Given that a certain machine has a clock cycle period of 0.5ns and takes 2 cycles to execute an instruction, find the following**:**

**a)** Clock Frequency
Clock Frequency = 1 / clock cycle period = 1 / 0.5ns = 2GHz

**b)** Instructions per second
Inst. Per sec = 1/ (CPI* clock cycle time) = $10^9$ Inst/sec

**c)** Suppose we have a program that has 500 instructions. How long will it take the program to run?
Program time = Inst. per program / (Inst. per sec) = 500ns

**Problem 4 (4 points)**
Suppose a 64-bit instruction takes the following format:

| | | | |
|---|---|---|---|
| OPCODE | SR | DR | IMM |

There are 126 opcodes and 32 registers.

a. What is the minimum number of bits required to represent an OPCODE?
126 Opcodes < 128 = 2^7 7 bits opcode

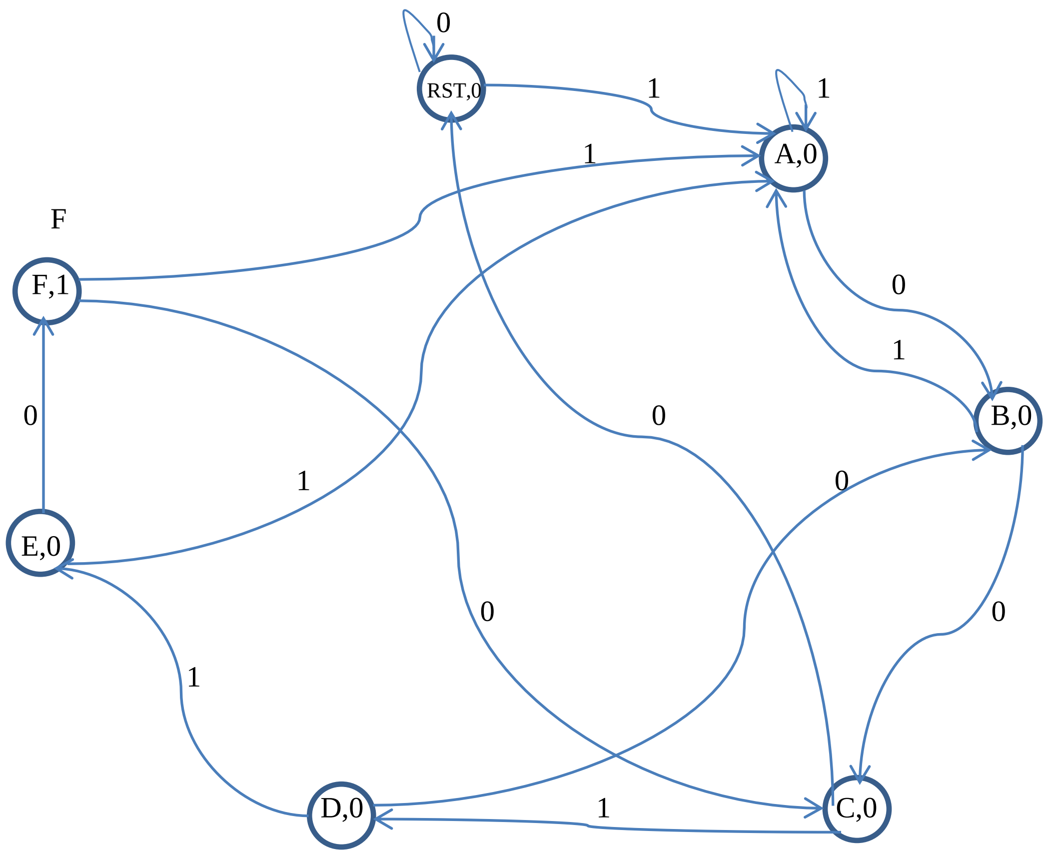
b. What is the minimum number of bits required to represent a register?
32 registers = 2^5 5 bits

c. What is the maximum number of bits that can be used to represent the immediate field (IMM)?
64 - 7 - 5 - 5 = 47 bits IMM

d. If the immediate (IMM) uses two's complement representation, what is its maximum range of values?
IMM ranges from -(2^46) to (2^46) -1

**Problem 5 (6 points)**

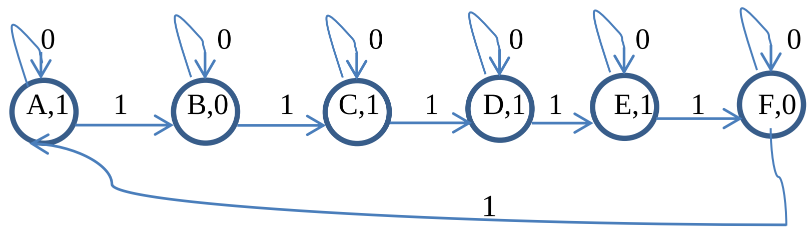
a. Draw a state diagram for a finite state machine that outputs 1 when it recognizes the pattern "100110". For instance, if we have an input of "1001100110" we should get an output of "0000010001". (This means that for the last 6 bits whenever it sees the pattern it outputs 1).




b. How many flip-flops (storage elements) will be needed to implement the finite state machine designed in your answer to part a?

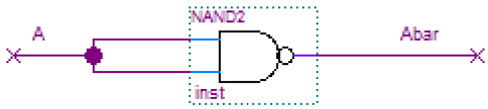
There are 7 states, hence we require 3 flip flops.

**Problem 6 (4 points)**
Draw a state machine that should output a 1'b1 if the number of 1's that have appeared on the input (including the current input bit) is a multiple of 2 **or** a multiple of 3.

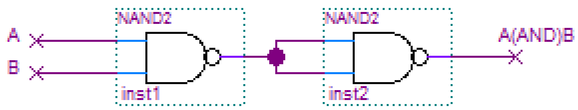



**Problem 7 (3 points)**
Prove that a NAND gate, by itself, is logically complete. (Hint: Construct a logic circuit that performs the AND function, a logic circuit that perform the OR function and a logic circuit that perform the NOT function. Use only NAND gates in these three logic circuits.)

**NOT using NAND:**




**AND using NAND:**




**OR using NAND:**