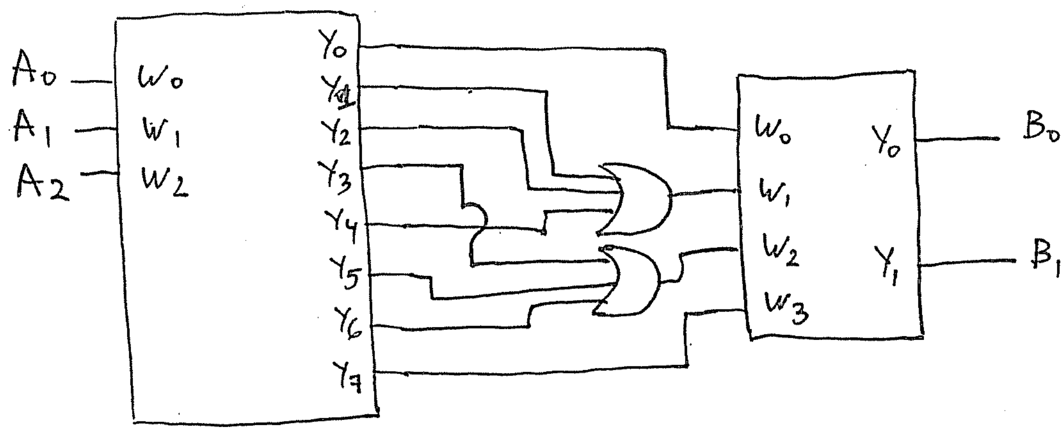
CprE 281: Digital Logic
Midterm 2: Monday November 3, 2014

Student Name: **Student ID Number:**

Lab Section:	**Mon 9-12(N)**	**Mon 12-3(P)**	**Mon 5-8(R)**	**Tue 2-5(M)**	**Wed 8-11(J)**
(circle one)	**Thur 2-5(L)**	**Thur 5-8(K)**	**Fri 11-2(G)**	**Fri 5-8(T)**	

1. **True/False Questions (10 x 1p each = 10p)**
   (a) I forgot to write down my name and student ID number. TRUE / FALSE
   (b) The inputs of a code converter are one-hot encoded. TRUE / FALSE
   (c) A T flip-flop can be implemented with an XOR gate and a D latch. TRUE / FALSE
   (d) When J=K=1 a J-K Flip-Flop is equivalent to a T Flip-Flop. TRUE / FALSE
   (e) Any Boolean function can be implemented using only 2-to-4 decoders. TRUE / FALSE
   (f) Any Boolean function can be implemented using only OR and NOT gates. TRUE / FALSE
   (g) In 2's complement notation 0110 > 1011 TRUE / FALSE
   (h) In 1's complement notation 0000 < 1111 TRUE / FALSE
   (i) The total delay through a half-adder is 2 gate delays. TRUE / FALSE
   (j) In a carry-lookahead adder all carry signals are generated after 3 gate delays. TRUE / FALSE

2. **Function Implementation with Decoder and Encoder (10p)**
Draw a circuit that accepts a 3-bit input number A (A2, A1, A0) and outputs a 2-bit number B (B1, B0) that is equal to the number of 1's that appear in the input A. For example, A=101 contains two 1's, so B=10. Also, A=111 contains three 1's, so B=11. You are allowed to use only one 3-8 decoder, two OR gates, and one 4-2 encoder. Label all inputs, pins and outputs.



3.Binary Addition and Subtraction (5 x 3p each = 15p)

Convert the following integers into binary numbers and perform the addition or subtraction using 2's complement if necessary. Write your answers and all intermediary steps to the right of each problem. Use 5-bit numbers for all problems and indicate if any bits need to be ignored.

```
  (+5)        00101                  00101
-                                 +
  (+2)      - 00010  =>2's           11110
------      -------               ---------
  +3                  ignore -> (1) 00011
```

```
  (+8)    +   01000                   01000
+                                  +
  (-1)      (-00001)  =>2's           11111
------      --------               ---------
  +7                   ignore -> (1) 00111
```

```
  (-7)      (-00111) =>2's      11001               11001
-         -                   -                   +
  (-2)      (-00010) =>2's      11110 =>2's         00010
-------     --------          -------             -------
  -5                                                11011
```

```
  (-15)     (-01111)  =>2's       10001
+                             +
  (+14)   +  01110                01110
-------     -------           -------
  -1                              11111
```

```
  (- 3)     (-00011) =>2's        11101
+                              +
  (-12)   + (-01100) =>2's        10100
-------     --------          ---------
  -15                      (1) 10001
                          ignore
```

4. Number Conversions (4 x 5p each = 20p)

(a) Convert $3FA00000_{16}$ (a 32-bit float stored in IEEE 754 format) to decimal:

$0\,|\,\underbrace{0111\,1111\,1}_{127}\,|\,010\ 0000\ 0000\ 0000\ 0000\ 0000$

2^{-2}

$$(-1)^0 \times 2^{127-127} \times \left(1 + \frac{1}{4}\right) = 2^0 \times \frac{5}{4} = 1.25$$

(b) Convert the following 32-bit float number (in IEEE 754 format) to decimal

$1\,|\,\underbrace{1\,0\,0\,0\,0\,1\,1\,0}_{128+4+2=134}\,|\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0$

negative

$$(-1)^1 \times 2^{134-127} \times \left(1 + \frac{1}{4} + \frac{1}{8}\right) = -2^7 \times \frac{11}{8} = -\cancel{2^3}\times 2^4 \times \frac{11}{\cancel{2^3}} = -16 \times 11$$

$$= -176$$

(c) Write down the 32-bit floating point representation for the real number 42

$42 / 32 = 1.3125$

```
  42
- 32
 100
- 96
  40
- 32
  80
- 64
 160
-160
   0
```

$2^5 = 32$

$5 + 127 = 132$

$2^{-1} = 0.5$

$2^{-2} = 0.25$

$2^{-3} = 0.125$

$2^{-4} = 0.0625$

$+ = 0.3125$

$0\,|\,10000100\,|\,01010\underbrace{0\ldots0}_{19 \text{ zeros}}$

(d) Write down the 32-bit floating point representation for the real number -9

$-9 / 8 = 1.125$

```
  9
- 8
 10
- 8
 20
-16
 40
-40
  0
```

$0.125 = 2^{-3}$

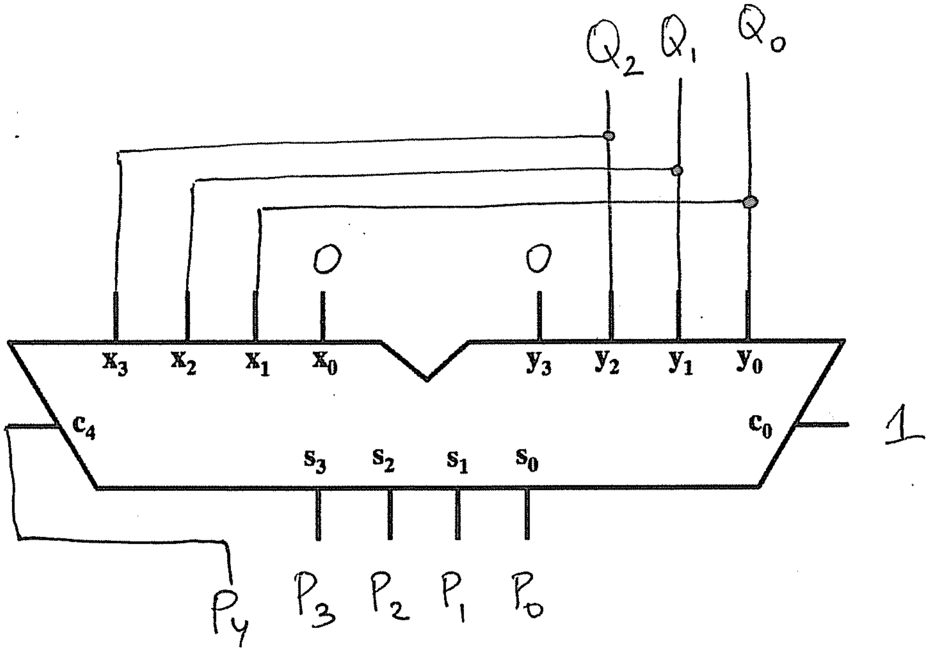
$2^3 = 8$

$3 + 127 = 130$

$1\,|\,1000\,0010\,|\,001\underbrace{00\ldots0}_{20 \text{ zeros}}$

5. Chip Implementation (7p + 3p = 10p)

a) Draw a circuit that uses the 4-bit adder shown below and any other basic logic gates that you think you might need to compute the equation P=3*Q +1. Assume that you have a 3-bit input Q (Q2, Q1, Q0) and a 5-bit output P(P4, P3, P2, P1, P0).
Clearly label all inputs and outputs. **(7p)**



b) Explain your solution. **(3p)**

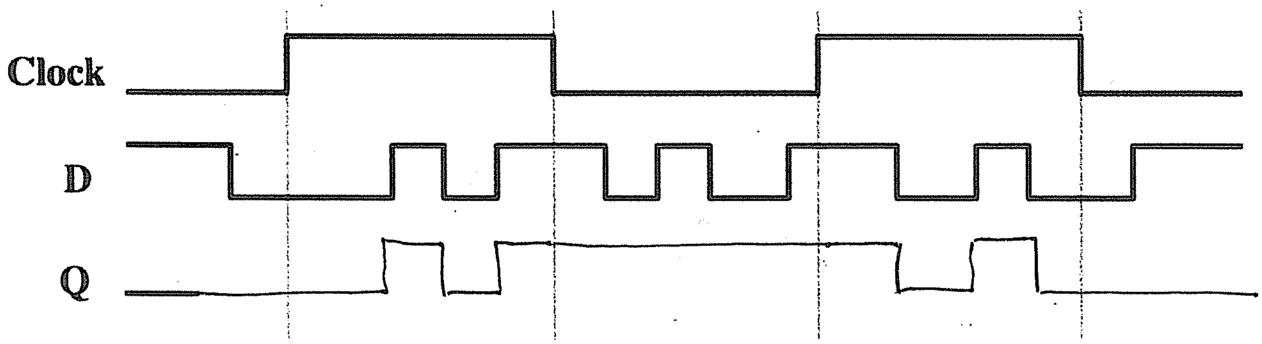
The equation can be rewritten as:

$$P = 2Q + Q + 1.$$

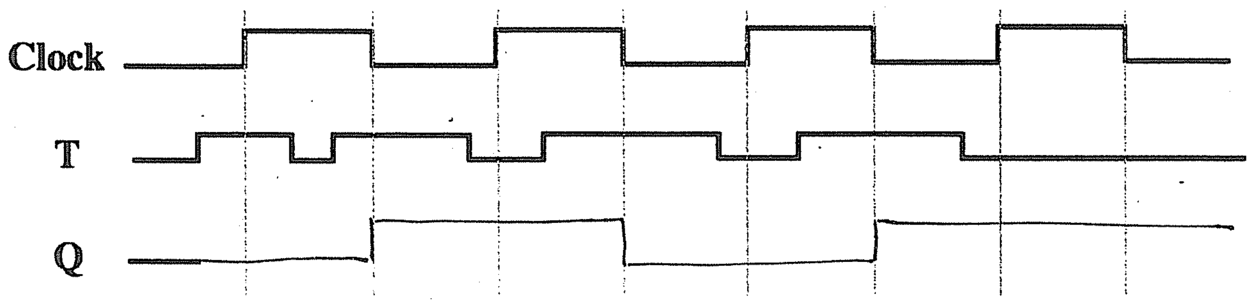
The multiplication by 2 in 2Q can be performed implicitly by simply shifting the wires to the left and setting $X_0 = 0$. Line Y_3 should be set to 0 as well since this is a 4-bit adder. Finally, the 1 in the equation can be added implicitly through the carry-in line. The carry-out is the most significant bit of the result. This solution uses no additional logic gates.

6. **Timing Diagrams (3 x 5p = 15p)**
Assume that Q is initially zero for all sub-problems.

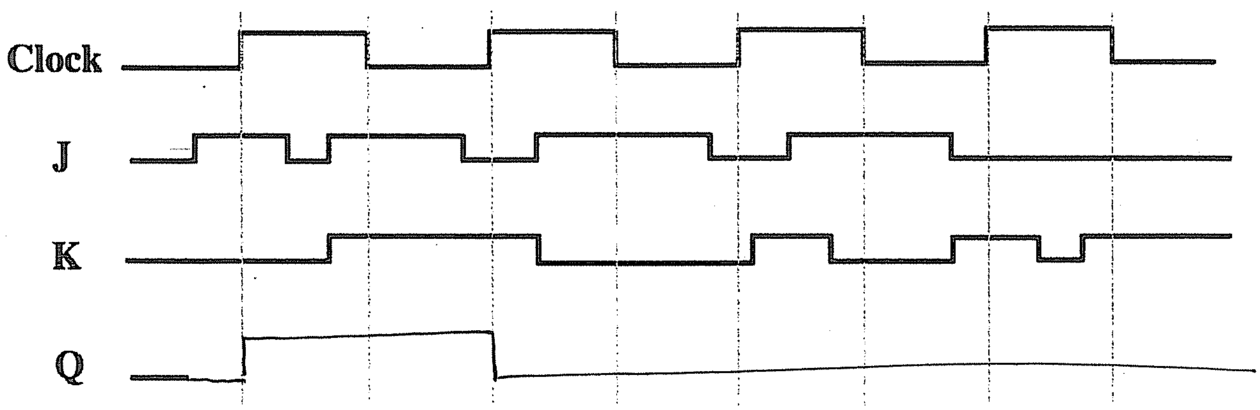
a) Complete this timing diagram for a gated D latch. (5pt)



b) Complete this timing diagram a negative edge-triggered T flip-flop. (5pt)



c) Complete the timing diagram for a positive edge-triggered J-K flip-flop. (5pt)



7. **Multiplexers (5p + 10p = 15p)**

a) **Draw the truth table for the function** $f = \overline{x}\,y + \overline{x}\,z + x\,\overline{y}\,\overline{z}$ **(5p)**

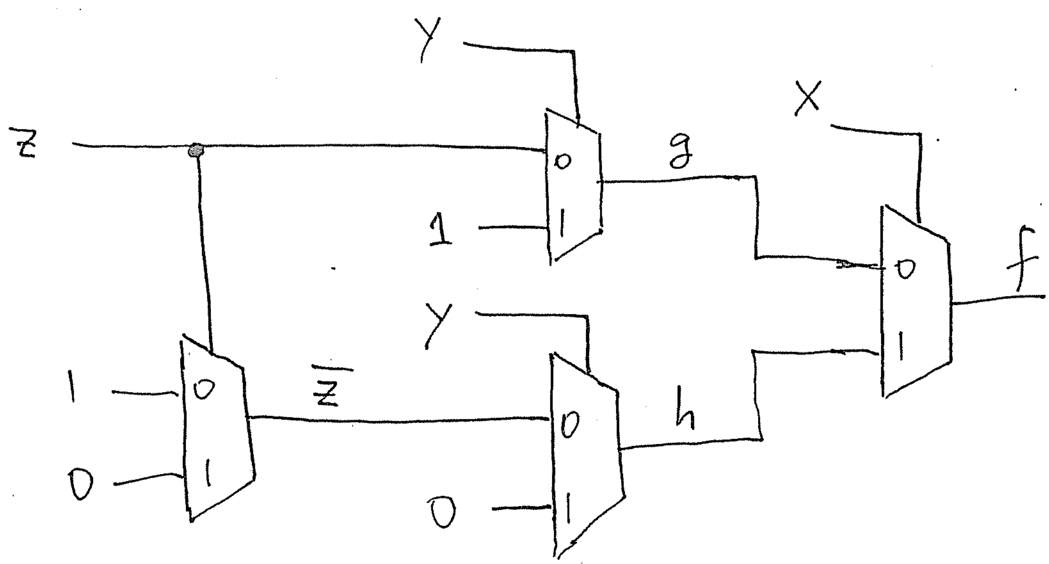
X	Y	Z	$\overline{X}$	$\overline{Y}$	$\overline{Z}$	$\overline{X}Y$	$+\overline{X}Z$	$+X\overline{Y}\overline{Z}$	f	
0	0	0	1	1	1	0	0	0	0	$g = OR(Y,Z)$
0	0	1	1	1	0	0	1	0	1	
0	1	0	1	0	1	1	0	0	1	
0	1	1	1	0	0	1	1	0	1	
1	0	0	0	1	1	0	0	1	1	$h = NAND(Y,Z)$
1	0	1	0	1	0	0	0	0	0	
1	1	0	0	0	1	0	0	0	0	
1	1	1	0	0	0	0	0	0	0	

b) **Implement this function using only 2-to-1 multiplexers and no other logic gates. The inputs to your circuits can be only the signals X, Y, and Z in their non-inverted form. (10p)**

Y	Z	g	
0	0	0	Z
0	1	1	
1	0	1	1
1	1	1	

Y	Z	h	
0	0	1	$\overline{Z}$
0	1	0	
1	0	0	0
1	1	0	

Z	$\overline{Z}$
0	1
1	0

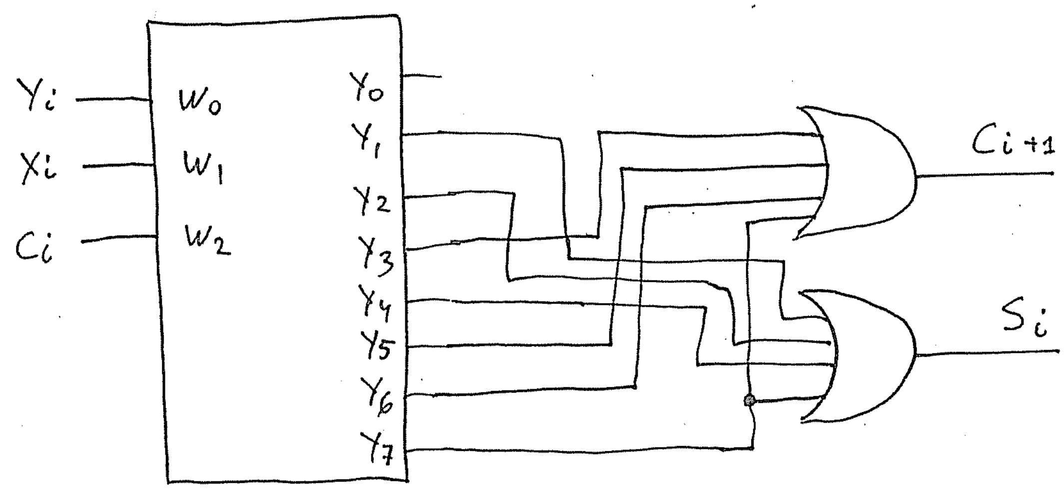


8. Full-Adder with 3-to-8 Decoder (10p)

a) Draw the truth table for a full-adder. **(2p)**

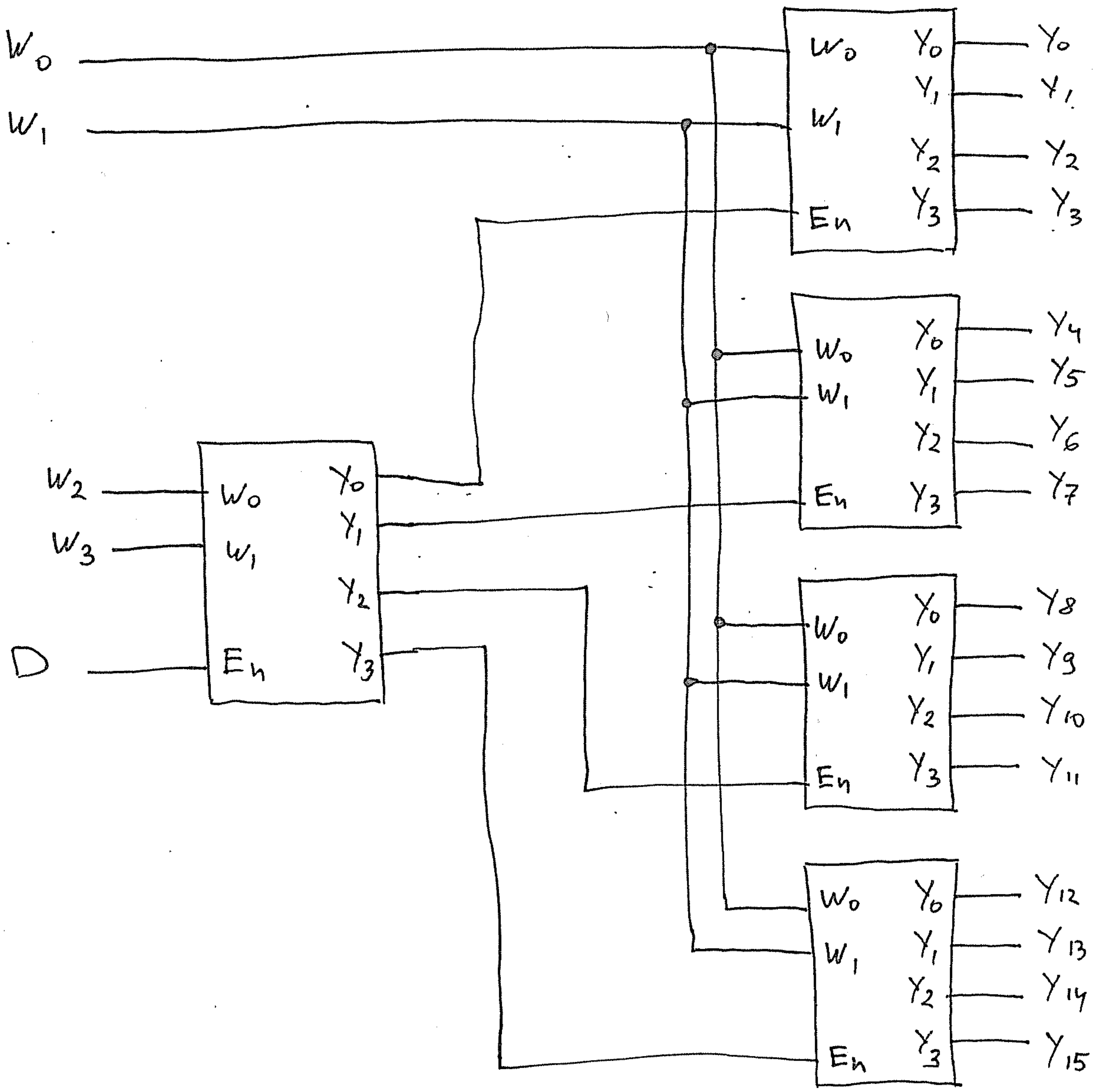
C_i	X_i	Y_i	C_{i+1}	S_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

b) Implement a full-adder circuit using one 3-to-8 decoder and a minimal number of basic logic gates. Clearly label all inputs, pins, and outputs of your circuit. **(8p)**



9. Demultiplexer (10p)

Implement a 1-to-16 demultiplexer using only 2-to-4 decoders with enable inputs and no other logic gates. Clearly label all inputs, pins, and outputs of your circuit.



The enable input of the top-level decoder is used as the input data line.

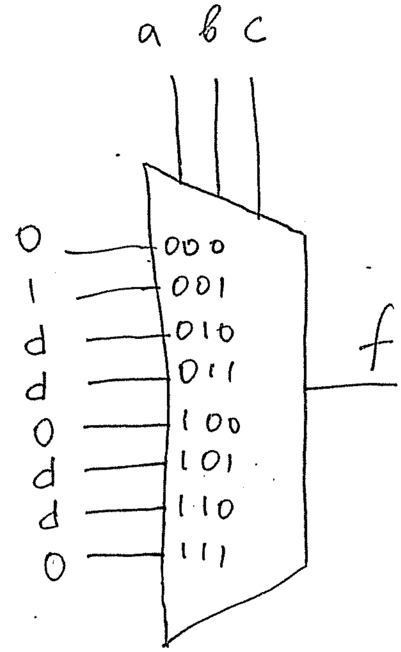
10. Prime Numbers (3 x 5p = 15p) [Note that part c) is on the next page]
Design a circuit that tests if a 4-bit binary number is a prime number in the decimal system. Assume that the input is represented by the signals a, b, c, and d (where a is the most significant bit and d is the least significant bit) and the output is called f. Hint: 0 and 1 are not prime.

a) Draw the truth table for the function f(a,b,c,d). **(5p)**

b) Implement the function f using one 8-to-1 multiplexer and a minimal number of extra AND, OR, and NOT gates. Clearly label all inputs, pins, and outputs. **(5p)**

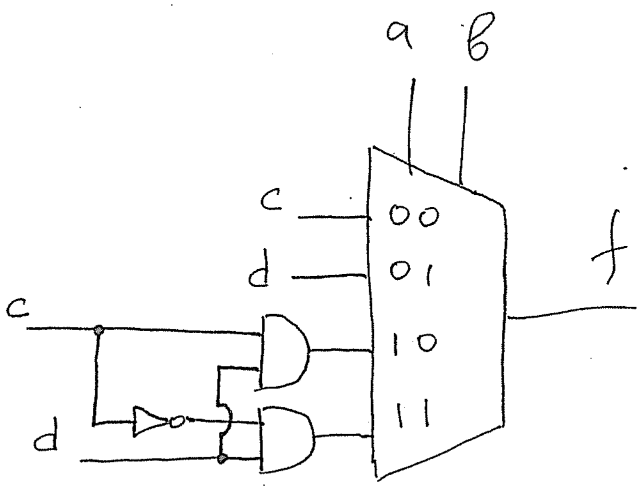
a)

#	a	b	c	d	f	
0	0	0	0	0	0	0
1	0	0	0	1	0	
2	0	0	1	0	1	1
3	0	0	1	1	1	
4	0	1	0	0	0	d
5	0	1	0	1	1	
6	0	1	1	0	0	d
7	0	1	1	1	1	
8	1	0	0	0	0	0
9	1	0	0	1	0	
10	1	0	1	0	0	d
11	1	0	1	1	1	
12	1	1	0	0	0	d
13	1	1	0	1	1	
14	1	1	1	0	0	0
15	1	1	1	1	0	



c) **Draw the truth table for the function f again on this page. Now, implement f using one 4-to-1 multiplexer and a minimal number of extra AND, OR, and NOT gates. Clearly label all inputs, pins, and outputs.** **(5p)**

#	a	b	c	d	f	
0	0	0	0	0	0	c
1	0	0	0	1	0	
2	0	0	1	0	1	
3	0	0	1	1	1	
4	0	1	0	0	0	d
5	0	1	0	1	1	
6	0	1	1	0	0	
7	0	1	1	1	1	
8	1	0	0	0	0	c • d
9	1	0	0	1	0	
10	1	0	1	0	0	
11	1	0	1	1	1	
12	1	1	0	0	0	$\bar{c} \cdot d$
13	1	1	0	1	1	
14	1	1	1	0	0	
15	1	1	1	1	0	



Question	Max	Score
1. True/False	10	
2. Decoder and Encoder	10	
3. Addition/Subtraction	15	
4. Number Conversions	20	
5. Chip Implementation	10	
6. Timing Diagrams	15	
7. Multiplexers	15	
8. Full-Adder with Decoder	10	
9. Demultiplexer	10	
10. Prime Numbers	15	
TOTAL:	130	