Homework 1 - ECE Review

Problem 1

Perform the following arithmetic using **6 bit 2's complement** numbers. Provide the answer in binary and hexadecimal form.

```
  30
 - 8
----
```

```
   011110            011110
 - 001000   →      + 111000
 --------          --------
                   [010110₂]

                   [16₁₆]
```

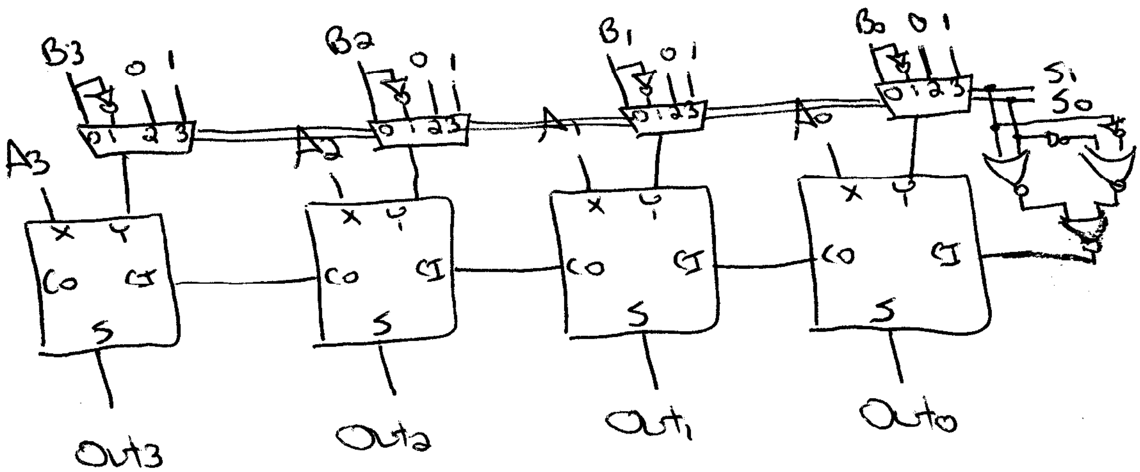
```
     011110          011110
OR - 001000  →     + 110111
     ------        + 000001
                   --------
                   1|010110₂
```

Problem 2

Design a 4-bit arithmetic block that performs the following functions. You may only use muxes, NOR gates, and 4 full adders. Assme all inputs are 4-bit binary numbers in 2's complement form.

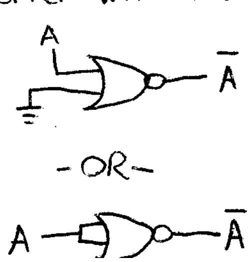
S_1 S_0	function	CI
0 0	A+B	0
0 1	A-B	1
1 0	A+1	1
1 1	A-1	0

$$C_I = (S_1+S_0)(\bar{S_1}+\bar{S_0}) = \overline{\overline{(S_1+S_0)}+\overline{(\bar{S_1}+\bar{S_0})}}$$

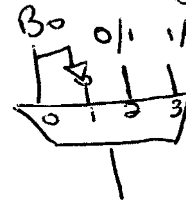


To get an inverter with NOR:

A B	A+B	$\overline{A+B}$
0 0	0	1
0 1	1	0
1 0	1	0
1 1	1	0



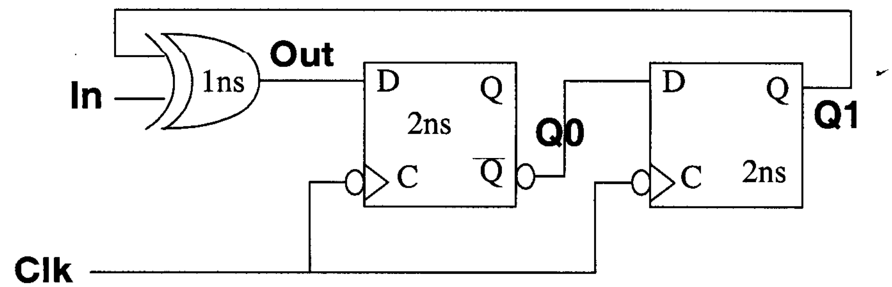
alternatively:

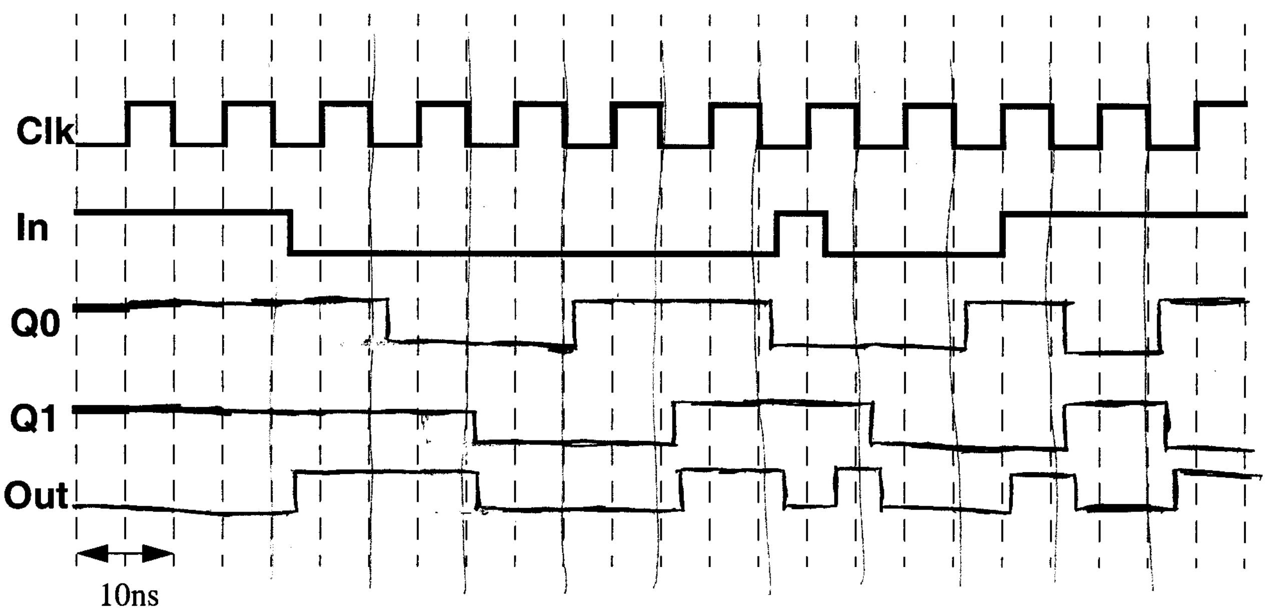


CI
0
1
1/0
0/1

Problem 3

Complete the timing diagram for the following circuit. Assume that the flip-flops have a 2ns delay each and that the XOR gate has a 1ns delay. Both flip-flops are negative edge triggered. Q0 and Q1 are initially high. Assume there are no set-up or hold violations.





Problem 4

An electronic lock is controlled by a keypad that has three buttons labelled **A**, **B**, and **R**. The keypad is restricted such that, at most, one key can be pressed at a time, and that each keypress generates an input signal that lasts for only one clock cycle. A state machine takes the keypresses as input, and its output is either lock (**L**) or unlock (**U**). If the correct sequence of keys is typed and followed by an **R**, the state machine generates an unlock (**U**) signal and then resets. At any other time the output should be locked (**L**). If an incorrect sequence of keys is pressed the lock becomes inactive until the reset (**R**) key is pressed, at which point the lock resets (with no unlock signal).

(a) Create state diagram for a MOORE machine that will test for the correct key sequence is BBA. Label your inputs A, B, R, and O (for no key pressed). Label your output L or U. Label your states with words such as "reset" and "inactive," or with partial key sequences such as "BBA." Make sure that all possible transistions exist.

(b) How many flip-flops are necessary to implement this state machine if the outputs are one-hot encoded? Explain your answer. need 1 F.F. for each state
⇒ need 6 F.F.

(c) How many flip-flops are necessary to implement this state machine if the states are binary encoded? Explain your answer. need 6 states, $2^3=8$ ⇒ need 3 flip-flops
000, 001, 010, 011, 100, 101 } 6 states
↳ $\lceil \log_2 6 \rceil$

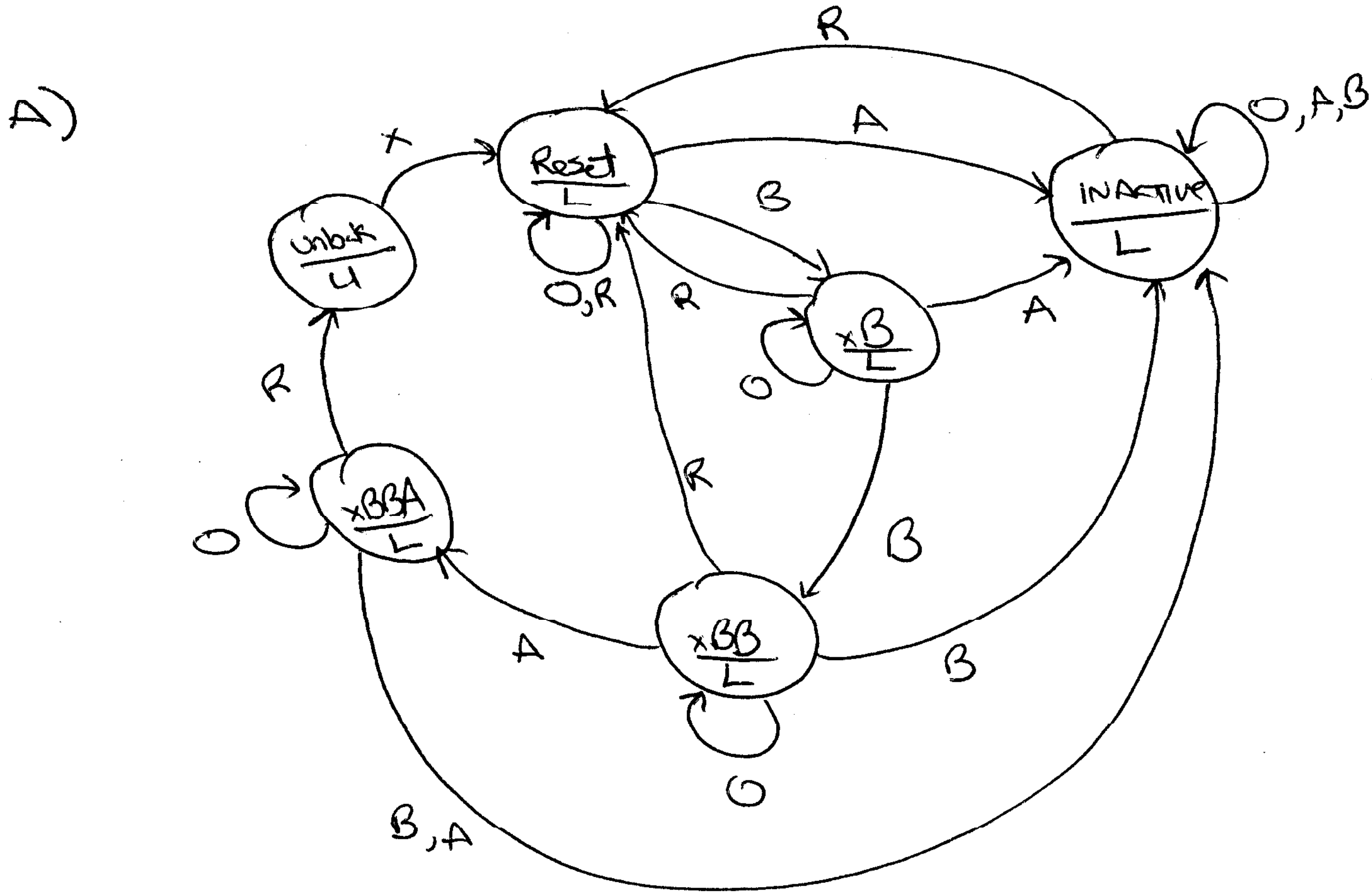
(d) Show how the state diagram would change for a Meely implementation?
See next page

(e) How many flip-flops are necessary to implement the Meely state machine if the states are one-hot encoded? Explain your answer. need 5 F.F. because have 5 states

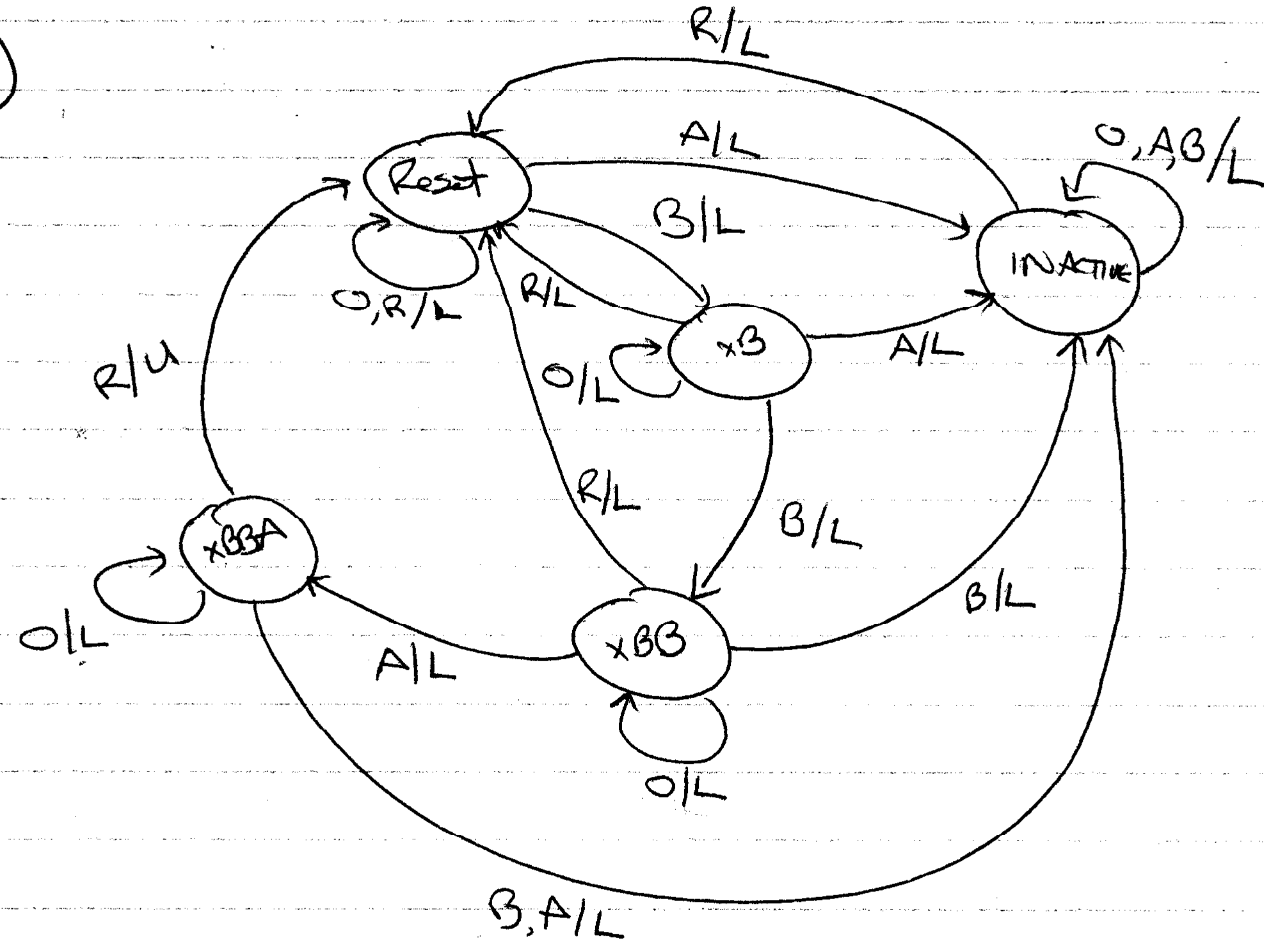
(f) How many flip-flops are necessary to implement the Meely state machine if the states are binary encoded? Explain your answer. still need 3 flip-flops to encode 5 states: 000, 001, 010, 011, 100

(g) Under what circumstances would a designer choose the Moore implementation? The Meely implementation? see next page

A)



D)



G) moore implementation is always necessary if you are concerned about the output changing if the inputs changes (i.e., output change not tied to the clock)
⇒ if going to another system that is clocked, not necessary
Mealy implementation if # of F.F. is a serious concern and you are using one-hot encoding.