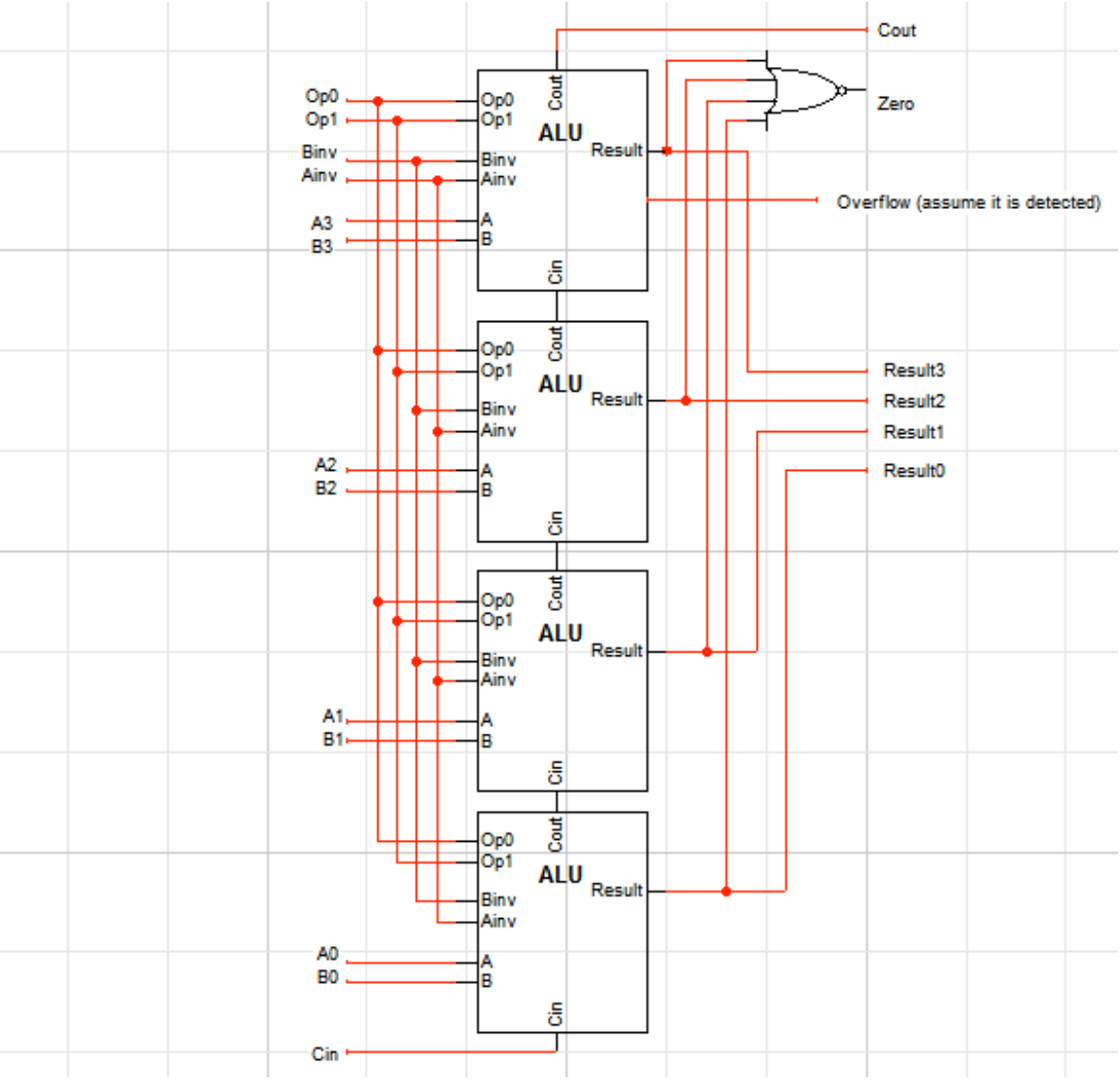
Computer Science 240
ALU and Memory
Assignment for Lab 4

1. Given the following diagram for a 4-bit ALU, with A and B 4-bit hexadecimal values, fill in the table below (assume that there is internal circuitry to detect overflow from an addition operation):



- Use ***4-bit, two's complement***, representation.
- Record **Result** as a ***hexadecimal value***; the other outputs are single bit (0 or 1).
- **Cout** and **overflow** come from adding **A** and **B**, even when the ALU function is a logical operation.

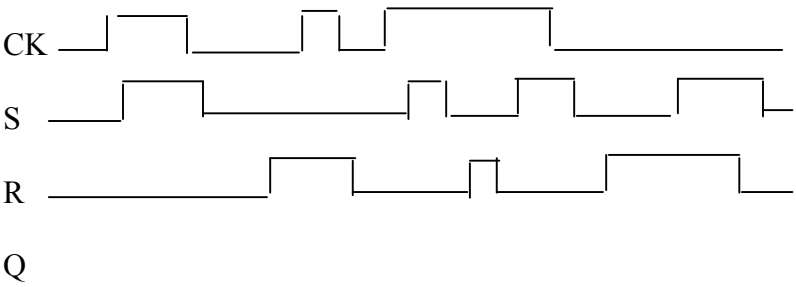
Function	Test Inputs	Result	Cout	Zero	Overflow
A + B	A = 3, B = 7				
A + B	A = F, B =E				
A AND B	A = 1,B = F				
A AND B	A = 7, B = 8				
A OR B	A = 3, B= C				
A NOR B	A = 6, B= 9				
A - B	A = 3, B= 9				
A - B	A= C,B= 7				

2. Give the truth table and draw the circuit diagram for the SR latch:

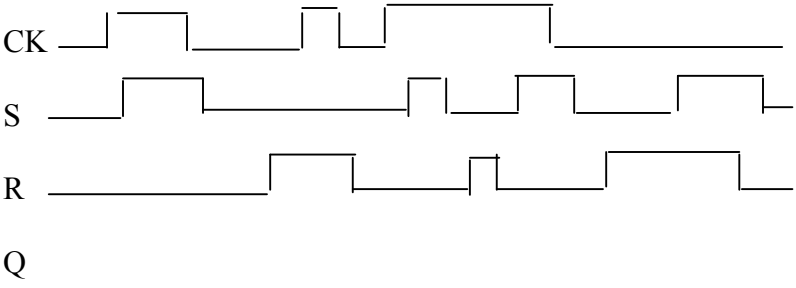
S	R	Q	Q'
0	0		
0	1		
1	0		
1	1		

3. Assume the inputs are S=0 and R=0. Do you know if the output Q is 0 or 1? Explain.

4. Assume you have a clocked SR **latch**. Draw Q , given the following CK, S, and R:



5. Assume you have a clocked SR **flip-flop**, that is activated on the positive edge of the clock. Draw Q given the same CK, S, and R:



6. Explain why the outputs are different for the latch and the flip-flop:

7. Flip-flops are **sequential circuits,** circuits whose output depends not only on the present value of its input signals but also on the value of past inputs. This is in contrast to combinational logic, whose output is a function of only the present input (the other circuits you have studied up to this point, such as basic gates, multiplexers, decoders, and ALU).

Because latches and flip-flops can store a state of 0 or 1, they are used as 1-bit memories, and can be used to create larger, n-bit memories such as registers and RAM. We will use those circuits in our implementation of an instruction set architecture.

Flip-flops are also used for solving problems which can be described by a Finite State Machine. Go to the following link:

http://www.cs.princeton.edu/courses/archive/spr06/cos116/FSM_Tutorial.pdf

which steps you through the process of solving a simple problem of this kind.

We will be doing a similar problem in lab this week. To prepare, read the example carefully, and write a short synopsis: