University of Washington – Computer Science & Engineering
Spring 2026 Instructor: Naomi Alterman 2026-06-02

CSE 369 QUIZ 3

Name: Suleiman Solutionizi

Student ID Number: 369369

Please do not turn the page until 1:50.

Instructions

- This quiz contains **8** pages, including this cover page.
- If you use any scratch paper, write your name and student ID number at the top and turn it in with your exam.
- Clearly box and indicate your final answers
- The quiz is closed book and closed notes.
- Please silence and put away all cell phones and other mobile or noise-making devices.
- Remove all hats, headphones, smart glasses, watches, and other digital wearables.
- You have 60 (+10) minutes to complete this quiz.

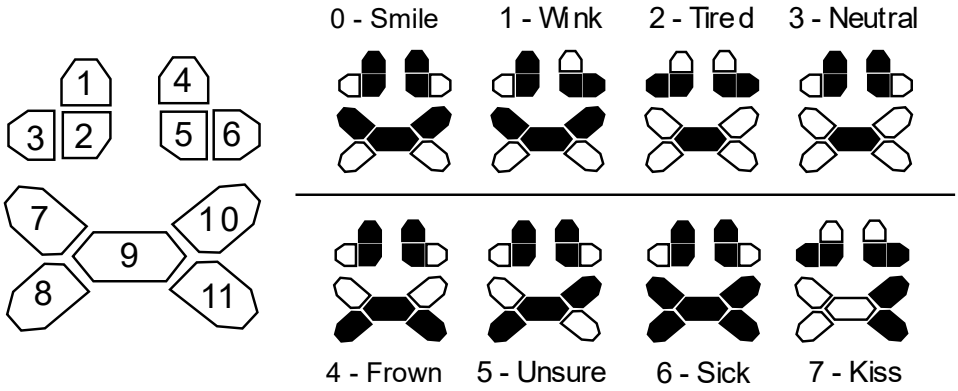
Advice

- Read questions carefully before starting. Read *all* questions first and start where you feel the most confident to maximize the use of your time.
- You may get partial credit for incomplete answers; show your work.
- Relax. If you've been practicing, you got this. If you haven't, you'll learn something now.

Question	Points
(1) Decoders	16
(2) Memory and Counters	23
(3) Shift Registers	19
Total:	**58**

Question 1: Decoders [16 pt]

We're building an electronic emoji keychain using **decoders** and a specially-designed **11-segment display**:

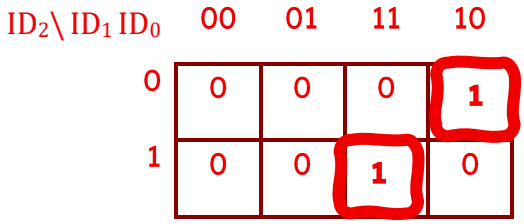


The input to the circuit is the 3-bit binary ID number for the emoji to display (listed above next to each emoji name). The output is an 11-bit bus called "S." Each bit of S drives one of the 11 display segments (numbered above-left). '1' means the segment should be filled in/black, and '0' means the segment should be empty/white.

(A) Complete the truth table to the right. [8 pt]

(B) In the space below, find the minimal two-level logical expression for S_3. For full credit, show your work. [8 pt]

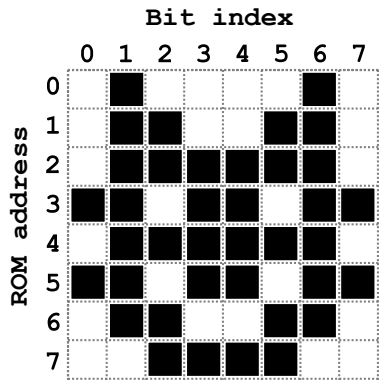
ID_2	ID_1	ID_0	S_3	S_4
0	0	0	**0**	**1**
0	0	1	**0**	**0**
0	1	0	**1**	**0**
0	1	1	**0**	**1**
1	0	0	**0**	**1**
1	0	1	**0**	**1**
1	1	0	**0**	**1**
1	1	1	**1**	**0**



$$S_3 = \overline{ID_2}\ ID_1\ \overline{ID_0} + ID_2 ID_1 ID_0$$

(could simplify to "$ID_1\ (ID_2\ \mathbf{xnor}\ ID_0)$" but not required for full credit)

Question 2: Memory and Counters [23 pt]

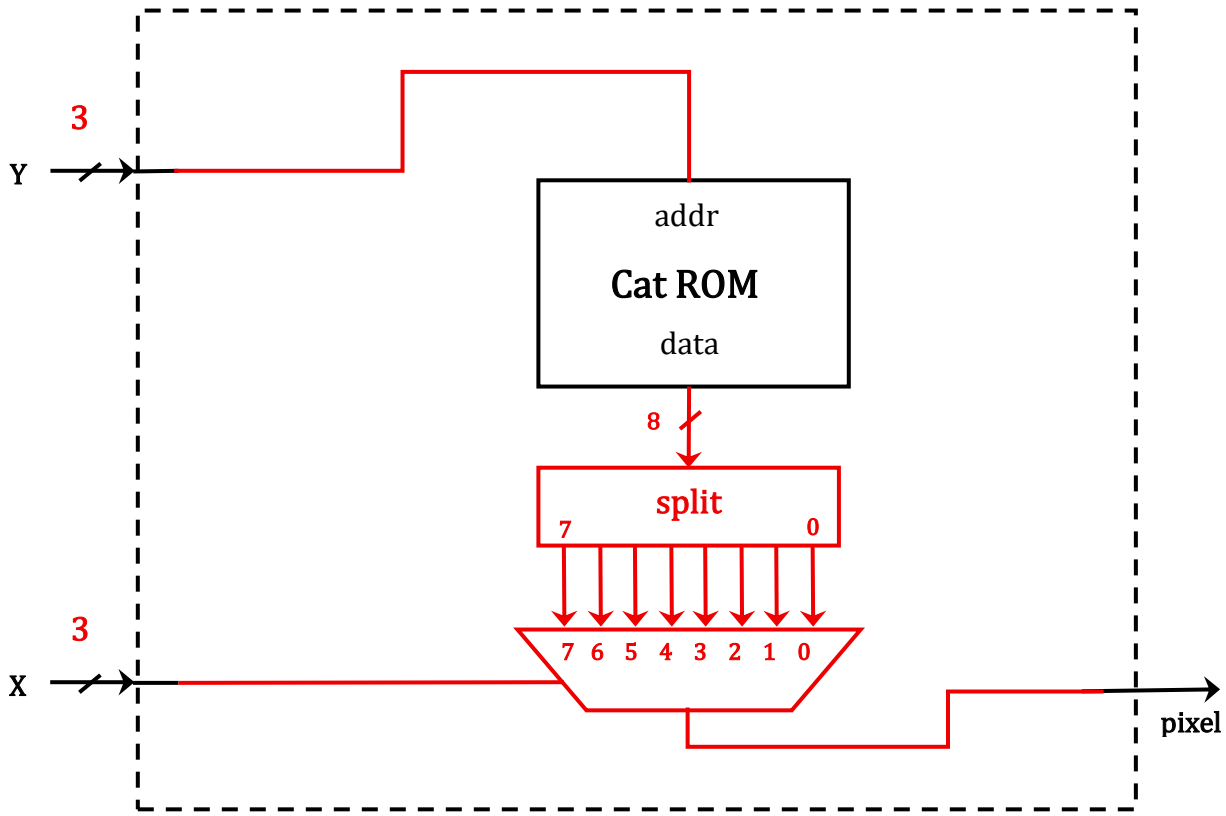


Ahahahahahaa yes, finally, it's time to build a circuit that draws this cartoon of Danni The Cat on an 8x8 LED screen. Yessss.

The picture is stored in a ROM with 8 8-bit words, where **each word represents a horizontal row of pixels**. Black squares in the diagram are "0"s and mean the LED should be off. White squares are "1"s and mean the LED should be on.

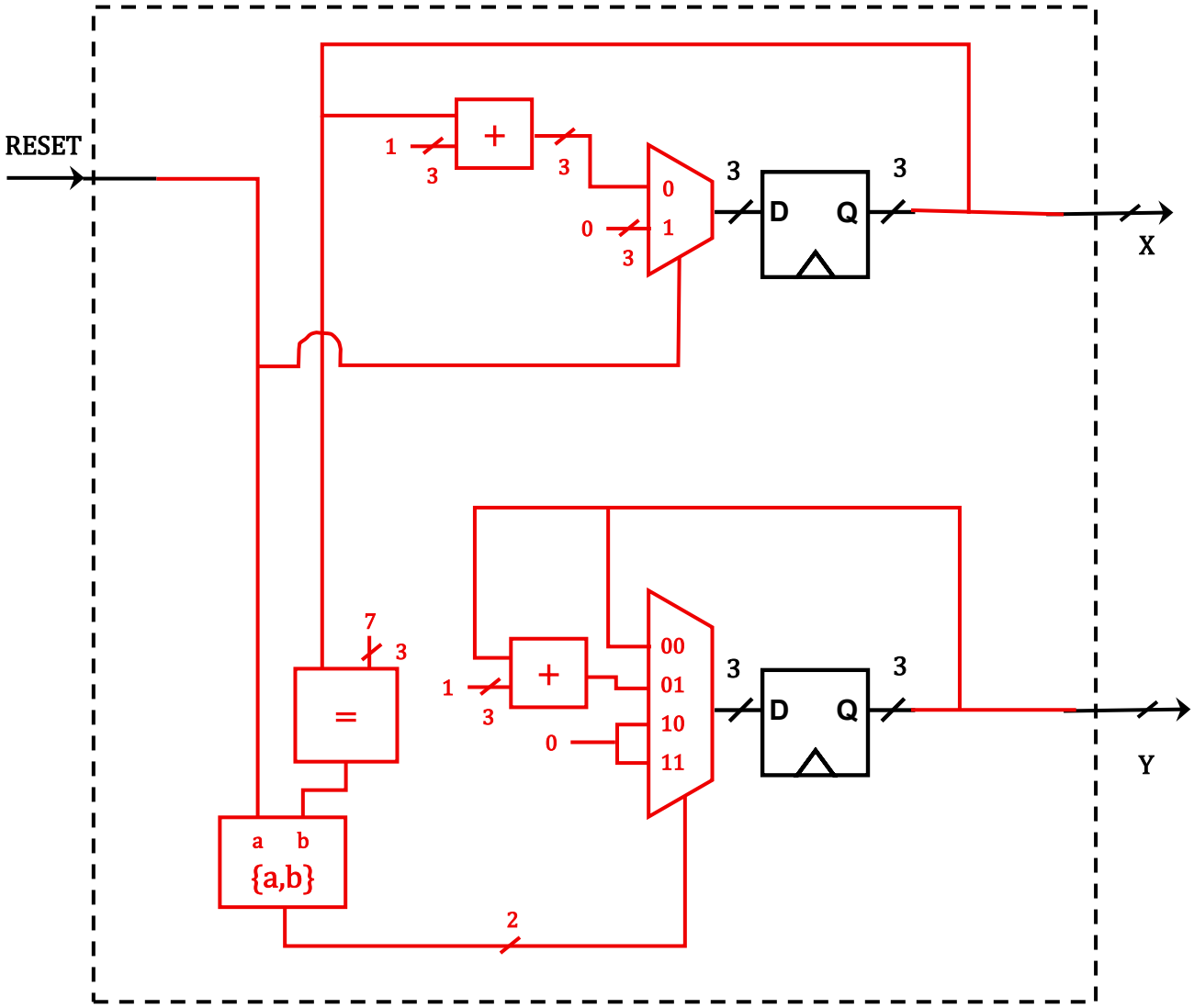
A) Complete the block diagram for a module called "PixelLookup", which takes an 3-bit "X" input and an 3-bit "Y" input and uses the ROM to give a 1-bit "pixel" output, which indicates whether the pixel at the specified (X, Y) coordinate should be on or not. The coordinate (0,0) is the top left of the display and LEDs are active-high.

This block diagram should be **entirely combinational**. You may use any of the blocks in the design library attached to the back of this exam (**hint:** take a look at the signal splitter). Make sure to thoroughly label the directionality, bit widths, and names of your signals. [8 pt]



B) Complete the block diagram for a module called CoordinateCounter, which generates the X and Y signals that get fed into the PixelLookup module (elsewhere, not in this diagram). The counter resets to output (0,0). Then, every cycle, X should increase by 1. Every **eight** cycles Y should increase by 1. When either number increases beyond 7, just let it overflow back to 0. So for example, the first several cycles of (X,Y) output after reset should be: (0,0), (1,0), (2,0), (3,0), (4,0), (5,0), (6,0), (7,0), (0,1), (1,1), (2,1), (3,1), …

Assume that all flip-flops are connected to the same external clock signal (you don't have to draw it). You may use any of the blocks in the design library attached to the back of this exam. Make sure to thoroughly label the directionality, bit widths, and names of your signals. [10 pt]



C) How many cycles would we need to run our counter module after reset to thoroughly test it? Explain how you arrived at this number. [4 pt]

The X counter overflows after it counts past 7 (8 cycles)
The Y counter counts up one every time X overflows and overflows after it counts past 7 (so, 8*8 cycles)
We want to confirm both X and Y overflow to (0,0)
So we'll want to run it for **65 cycles** to confirm correct behavior.

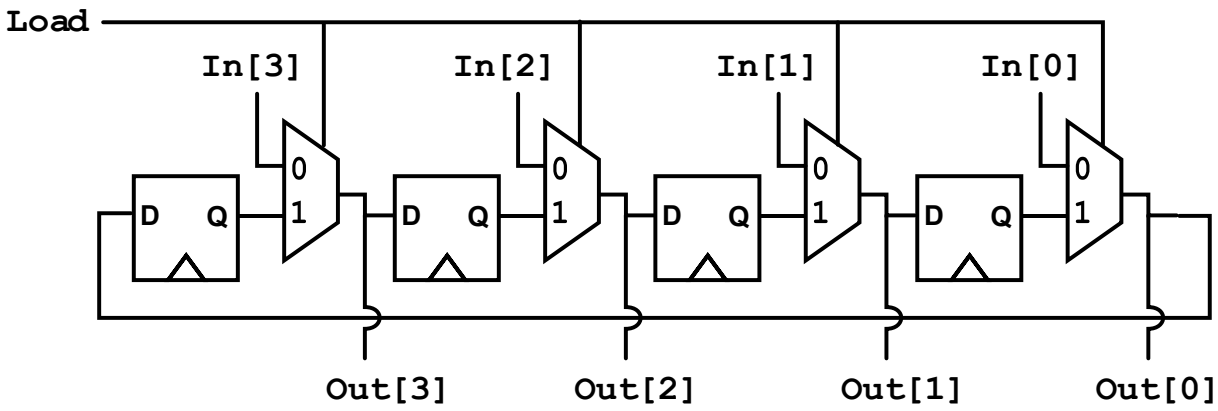
Depending on your interpretation of the question this answer might be 64. With a good explanation either number is an acceptable answer.

D) Danni is very pleased by this machine. She wants three hundred of them. How much money do we sell them to her for? (the only wrong answer is not writing anything down) [1 pt]

$0, Danni deserves every good thing in the whole world and we should be proud to give it to her

Question 3: Shift registers [19 pt]

Consider the following shift register:



a) Complete the table below to show how the output value of the shift register would evolve over time if it was loaded with the value "1011" before-hand. [8 pt]

Cycle #	Out[3]	Out[2]	Out[1]	Out[0]
1	1	0	1	1
2	**1**	**1**	**0**	**1**
3	**1**	**1**	**1**	**0**
4	**0**	**1**	**1**	**1**
5	**1**	**0**	**1**	**1**
6	**1**	**1**	**0**	**1**
7	**1**	**1**	**1**	**0**
8	**0**	**1**	**1**	**1**

b) The "`Load`" signal is intended to signal that the shift register should load in the four-bit `In` signal instead of just permuting its previous state. As shown in the block diagram, is `Load` active-high or active-low? [1 pt]

Active low

Let's see if it violates any timing constraints. Assume $t_{C2Q} = 2$, $t_{Mux} = 3$, $t_{Setup} = 4$, $t_{Hold} = 6$, $t_{Period} = 18$. Assume signals entering and leaving the diagram (eg, the bits of In and Out) have no timing constraints attached to them.

Is the shift register violating setup time constraints? Show your work for full credit. [5 pt]

$$t_{Period} - t_{Setup} \geq t_{C2Q} + t_{Mux}$$

$$18 - 4 \geq 2 + 3$$

$$14 \geq 5$$

No, it meets setup time constraints!

c) Is the shift register violating hold time constraints? Show your work for full credit. [5 pt]

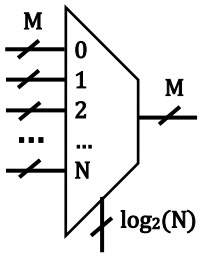
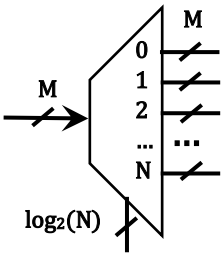
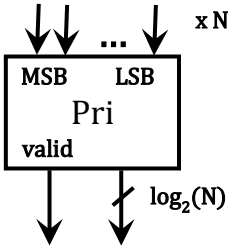
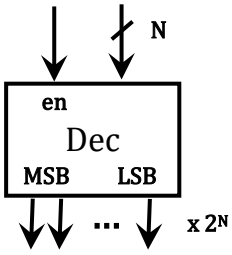
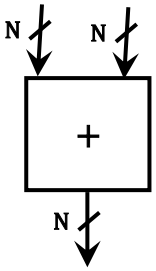
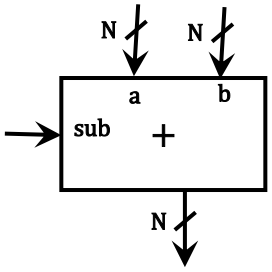
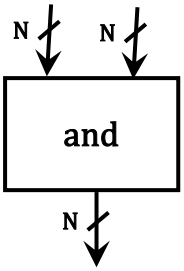
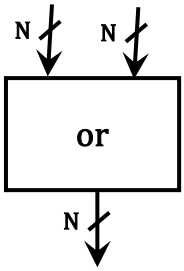
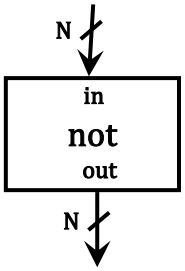
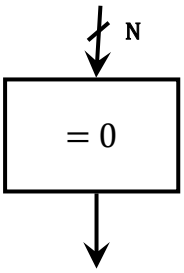
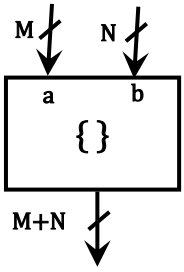
$$t_{Hold} \leq t_{C2Q} + t_{Mux}$$

$$6 \leq 2 + 3$$

$$6 \leq 5$$

Yes, it violates hold time constraints!

Design Library

M-bit N:1 Mux 	M-bit 1:N Demux 	N-bit Priority Encoder 
N-bit Binary to One-Hot Decoder w/ Enable 	N-bit Adder 	N-bit Adder/Subtractor (a+b / a-b) 
N-bit Bit-wise AND 	N-bit Bit-wise OR 	N-bit Bit-wise NOT 
Zero Comparator 	Concatenate (glue signals together) 	Split (bus to individual bits)