

CSE 369 QUIZ 3

Name: _____

UWNetID: _____

Please do not turn the page until 10:30.

Instructions

- This quiz contains 4 pages, including this cover page. You may use the backs of the pages for scratch work.
- Please clearly indicate (box, circle) your final answer.
- 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, and watches.
- You have 30 minutes to complete this quiz.

Advice

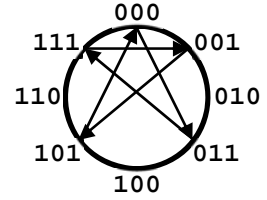
- Read questions carefully before starting. Read *all* questions first and start where you feel the most confident to maximize the use of your time.
- There may be partial credit for incomplete answers; please show your work.
- Relax. You are here to learn.

Question	Points	Score
(1) Counters	12	
(2) Shift Registers	9	
(3) Routing Elements	9	
Total:	30	

Question 1: Counters [12 pts]

Implement a counter that goes through the following state sequence: **000** → **011** → **111** → **001** → **101** → **000** → ... using a *minimal number of 2-input logic gates*.

Fun side note – this counter creates a star on the number wheel:



PS ₂	PS ₁	PS ₀	NS ₂	NS ₁	NS ₀
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

NS ₂	00	01	11	10
0				
1				

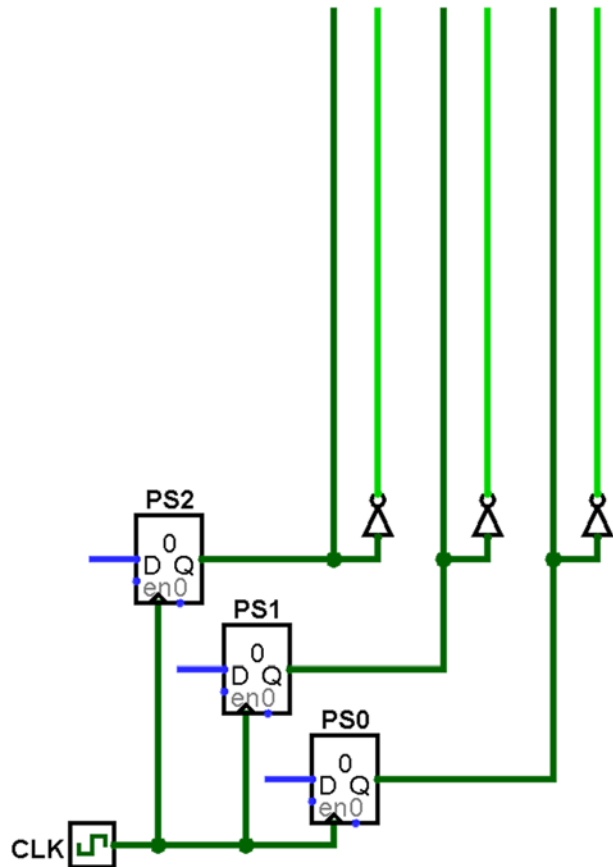
NS ₁	00	01	11	10
0				
1				

NS ₀	00	01	11	10
0				
1				

Wire connection:



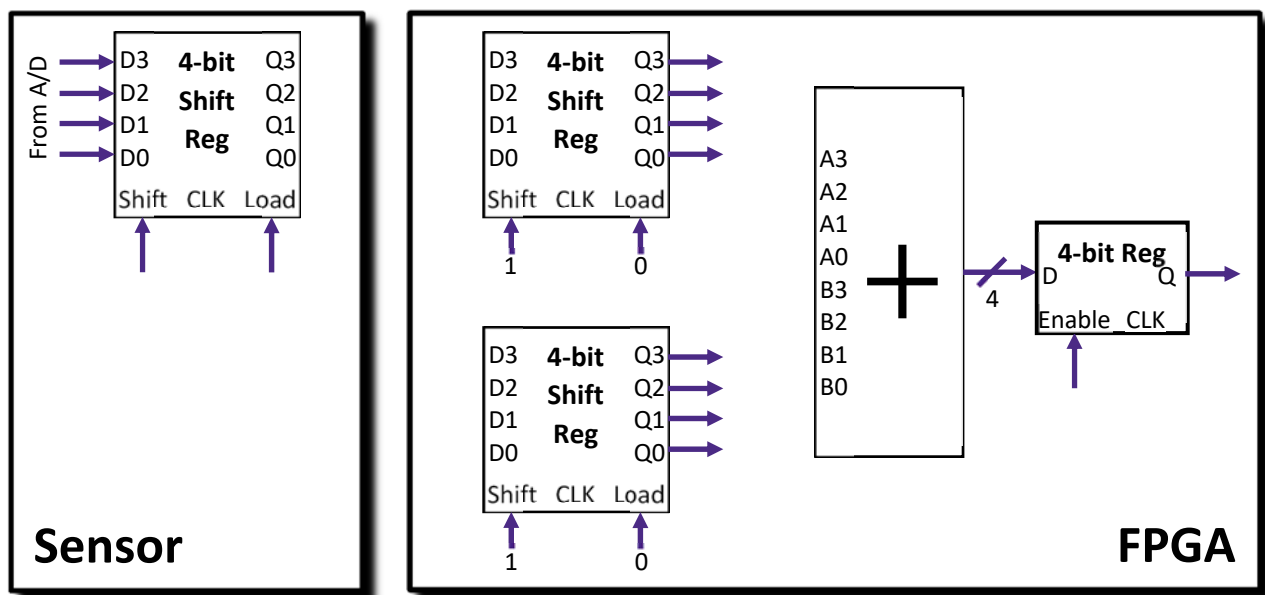
Wire crossing:



Question 2: Shift Registers [9 pts]

We are reading **serialized** data from a sensor that includes a 4-bit analog-to-digital (A/D) converter and a shift register. We sum up **consecutive readings** of sensor data and store the result in an enabled 4-bit register. We only have access to 4-bit shift registers with parallel load capability. *Serial loading is done from the D0 input.*

- (A) Connect the circuit elements shown below to allow for the behavior described above. [3 pt]
- (B) Assuming the first sensor data is available on Cycle 0, fill in the tables at the bottom for the unspecified inputs (Shift and Load of the sensor shift register and Enable to the result register) to produce the behavior described above. [6 pt]



Ungraded (for your use)

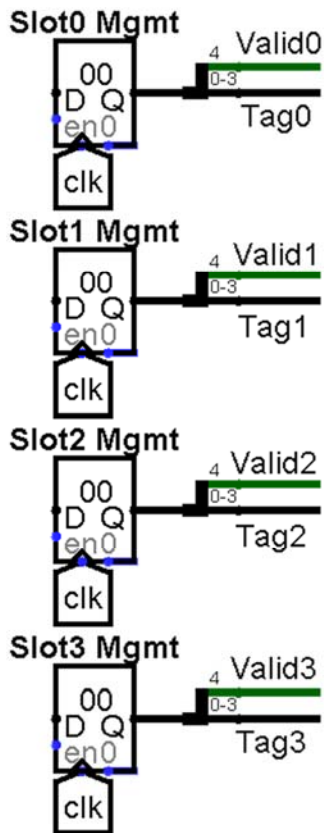
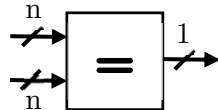
Cycle	Sensor Shift Reg						Result Reg
	Shift	Load	Q0	Q1	Q2	Q3	
0							
1							
2							
3							
4							
5							
6							
7							
8							
9							

Question 3: Routing Elements [9 pts]

A **cache** is a hardware device that stores the most recently accessed blocks of memory for a CPU. Implement the logic to **check for a cache hit** in a small 4-block direct-mapped cache below. *Ignore all other normal cache functionality.* As a reminder, the procedure is as follows:

- Split the requested data's address into Tag, Index, and Offset fields (provided for you)
- The value of the Index bits (2 bits) tells you which cache line to check
- A cache hit occurs if the cache line is **Valid (1 bit)** and the **Tag (4 bits) matches**

We represent the cache line management bits below using registers. Assume you can freely use equality logic blocks:



Hit?

