LAB 10 // Hardware

THREE-OUT-OF-FOUR DETECTION

INSTRUCTIONS and RUBRIC

CSE 2301 – Fall 2024

INTRODUCTION

Prerequisites

This lab will rely on your understanding of how to determine whether a sequence has occurred from a string of binary bits. You need to have a solid understanding of sequence detection and be proficient with manipulating registers.

Objectives & Background

Many communication systems use start and stop codes to delineate information, so detecting these codes is very important. The theory behind this bit counting relies on storing states in a series of flip flops to tell the circuit what has been detected so far. For example, if you are looking to find a `010` string then there must be a state that recognizes the `01` pattern and proceeds appropriately when subsequently given a `0` or `1`.

We will not be giving you much instruction for this lab. Work independently and with attention to detail; your hardware implementation will be reasonably complex. Make sure that you have a strong grasp of D flip-flops and registers, state diagrams, and sequence detection before starting.

YOUR TASK:

101/010 Sequence Detector

Make a sequence detector **(MOORE MACHINE)** that identifies "`010`" and/or "`101`" (they can be overlapping and multiple sequences possible) using D flip-flops. When identified, a solitary LED should light up ***after*** the rising edge of the clock. Note that just waiting for the sequence to appear in a serially inputted register is ***not an acceptable*** solution.

1. First, determine how many states are required. Recall that n D flip-flops are needed to account for between 2^{n-1} and 2^n states.
2. Draw a state diagram indicated the input (A) and the output (B, sequence detection) at each transition. The form of the annotation for each transition should be A/B. All states should have two transitions *from* them.
3. Create a transition table including all states. The independent terms will be the initial state of the register and the input (A), and the dependent terms will be the "next" state of the register and the output (B).
4. Draw appropriate Karnaugh Maps to minimize the input functions to each D flip-flop (change of state based on current state an input).

5. Build the circuit in LogicWorks and test for precision.
6. Implement your design in hardware. We recommend having one LED for the output and [as many LEDs as you need] to read out the current state for debugging purposes. Remember to write down what each light is associated with.
7. Check results with the state diagram and then demo.

SCORING

This is a straightforward lab with a large hardware implementation. Take your time and work deliberately, and be sure that you're paying attention to the guidelines for the *nature* of your solution, *how* it should be displayed, and *when* the sequence should be detected.

Demo Requirements [5 pts]:

- ✓ Demo your hardware implementation. If, when given both sequences, both are detected in the appropriate manner, you will receive full credit. If one fails, you will receive partial credit, but at least one of the sequences must work.

Report Requirements [2.8 pts]:

- ✓ [0.8] ***Theory***:
  - ○ [0.8] What is the difference between sequential and combinational logic? Your answer should be detailed and precise. Pay close attention to the definitions.
- ✓ [1.4] ***Deliverables***:
  - ○ [0.4] Include a tidy state diagram. Label each state with its ID (D flip-flop values) and the sequence that it represents.
  - ○ [0.6] Include the transition table. This table should include all possible states, not just those that are used (use don't care states for unused rows). If you have three registers (as you should), then you will have columns Q2, Q1, Q0, A (input), Q2+, Q1+, Q0+, and B (output).
  - ○ [0.4] Completed Karnaugh maps and reduced equations. Deductions will be made if these are not optimally reduced.
- ✓ [0.6] ***Discussion section***. Should conform to standard lab report guidelines.

Practice Questions [2.2 pts]:

- ✓ [1.4] ***Question 1:***
  - ○ Using JK flip flops, design a counter that counts from DCBA=0000 sequentially to DCBA=1011 and then returns to 0000. Complete the table below.

	current				next											
	D	C	B	A	D	C	B	A	JD	KD	JC	KC	JB	KB	JA	KA
0	0	0	0	0	0	0	0	1								
1	0	0	0	1												
2	0	0	1	0												
3	0	0	1	1												
4	0	1	0	0												
5	0	1	0	1												
6	0	1	1	0												
7	0	1	1	1												
8	1	0	0	0												
9	1	0	0	1												
10	1	0	1	0												
11	1	0	1	1												
12	1	1	0	0												
13	1	1	0	1												
14	1	1	1	0												
15	1	1	1	1												

- ✓ [0.8] ***Question 2:***
  - ○ Draw the state transition diagram for the transition table given below (0.4 pts).
  - ○ We are asked to build this counter using D flip-flops. Use a K-map to find the logic expressions for D0 (which defines how Q0+ is determined) (0.4 pts).

Current			Next		
Q2	Q1	Q0	Q2+	Q1+	Q0+
0	0	0	1	1	1
0	0	1	1	0	1
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	0	1	1