Homework #4 v20200229

(Deadline: 11:59PM PDT, Thursday March 12, 2020)

Name (Last, First): Arora, Gurbir

Student Id #: 105178554

INSTRUCTIONS

This homework is to be done individually. You may use any tools or refer to published papers or books, but may not seek help from any other person or consult solutions to prior exams or homeworks from this or other courses (including those outside UCLA). You're allowed to make use of tools such as Logisim, WolframAlpha (which has terrific support for boolean logic) etc.

You must submit all sheets in this file based on the procedure below. Because of the grading methodology, it is much easier if you print the document and answer your questions in the space provided in this problem set. It can be even easier if you answer in electronic form and then download the PDF. Answers written on sheets other that the provided space will not be looked at or graded. Please write clearly and neatly - if we cannot easily decipher what you have written, you will get zero credit

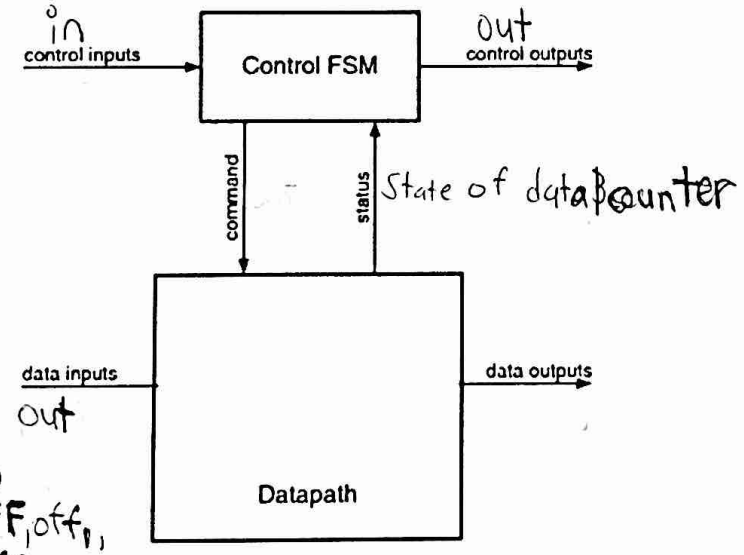
SUBMISSION PROCEDURE: You need to submit your solution online at Gradescope (https://gradescope.com/). Please see the following guide from Gradescope for submitting homework. You'd need to upload a PDF and mark where each question is answered.
http://gradescope-static-assets.s3-us-west-2.amazonaws.com/help/submitting_hw_guide.pdf

Problem #1

Design a light flasher

- Inputs: in (1-bit wide)
- Outputs: out (1-bit wide)
- Operations:
  - When in = 1, FSM goes through 5 sequences: On-Off-On-Off-On
  - Each On sequence: out = 1 for 6 cycles long
  - Each Off sequence: out = 0 for 4 cycles long
  - After the 5 sequences, FSM goes back to an OFF state to wait for new input. During the OFF state, out = 0 (1 cycle – the OFF state isn't a sequence of outputs and checks for updates at each cycle).
- Notes:
  - in need not be asserted (= 1) while the 5 sequences are outputted. Once the FSM starts the sequences, it will continue the sequences until completed regardless of in.
  - Assume all the logic has access to a system clock (clk) to define the cycle length.

(a) Design the control logic;
(b) Design the datapath logic;
(c) Use the following diagram as a template to write down (either over the diagram or separately) what are the control inputs, control outputs, command, status, data inputs and data outputs of your design.

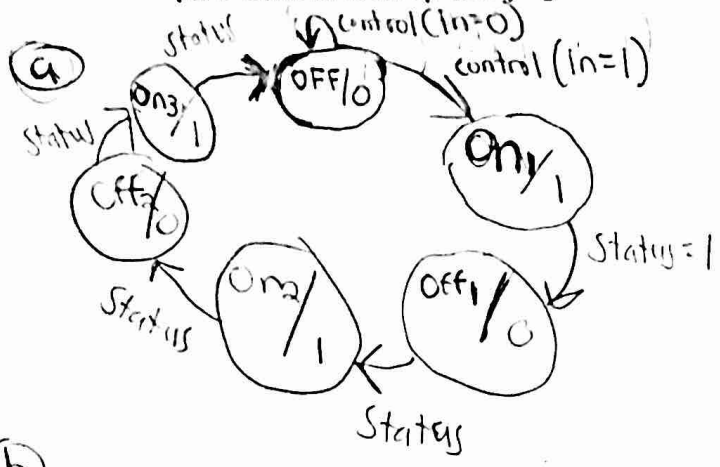


data$_1$ = 0
if OFF, off$_1$, off$_2$
data$_1$ = 1
if on$_1$, on$_2$, on$_3$

form: (state/out)

(IMPORTANT: Keep this page in submission even if left unused)

(a)



Control FCM

6-States

3 inputs - control
4 cycle
6 cycle

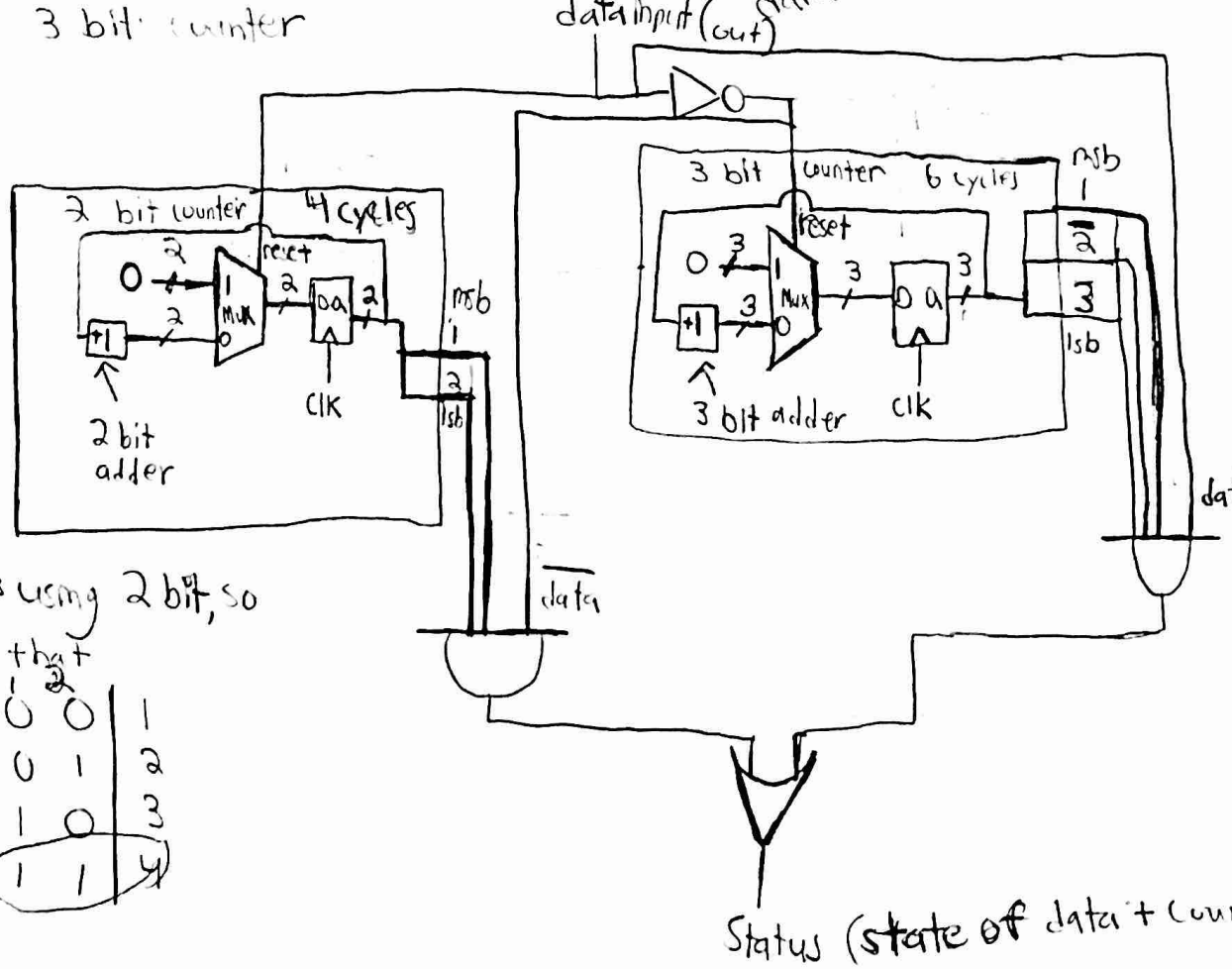
outputs - data input

(b)

Datapath

2 bit counter
3 bit counter

data=0 in OFF, off1, off2
data=1 in on1, on2, on3



using 2 bit, so that

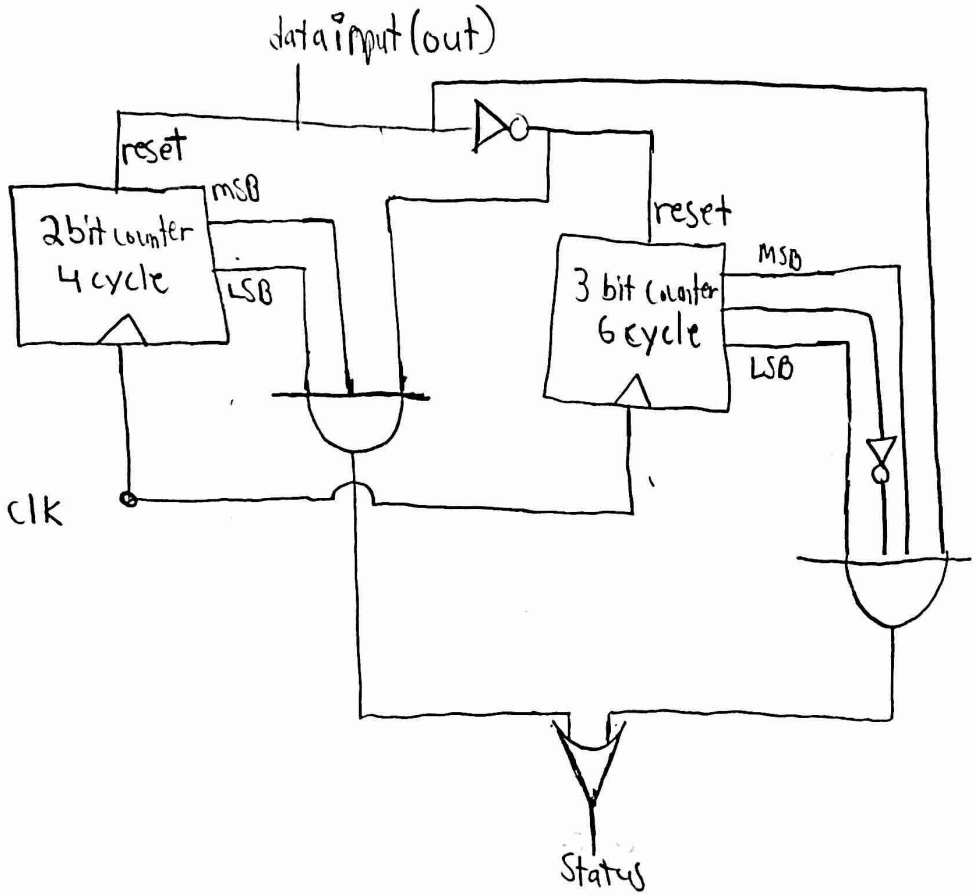
1	2	
0	0	1
0	1	2
1	0	3
1	1	4

using 3 bit so that

1	2	3	
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

Status (state of data + counter)

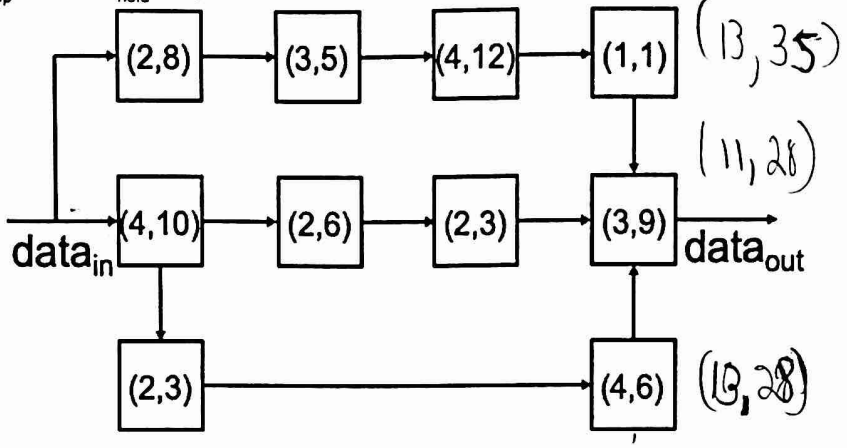
(IMPORTANT: Keep this page in submission even if left unused)



(IMPORTANT: Keep this page in submission even if left unused)

Problem #2

The following shows blocks of logic interconnected. Each block is atomic (cannot be further divided). The delays of each block are represented by the notation (contamination, propagation). If $data_{in}$ are from registers and $data_{out}$ are stored by registers. The registers have $t_{dCLK\text{-}Q} = 3$, $t_{setup} = 1$ and $t_{hold} = 1$.



(a) If this logic block is represented as a single block, how would you represent its contamination and propagation delay?

(b) What is the minimum cycle time of this logic?

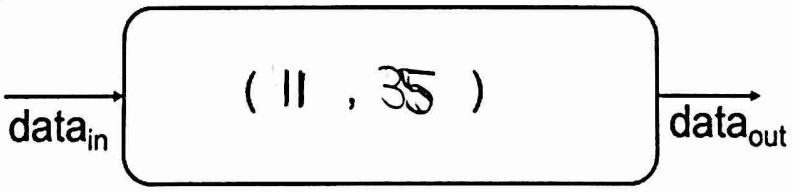
(c) If this logic block is now pipelined into 2 stages, where would you insert the registers to balance the delay of the stages? Show each register in the diagram with a vertical line through the arrows. In the new diagram representing all the logic on each side of the registers as a single block (one block before the registers and one after), indicate the contamination and propagation delay of each of these blocks.

(d) What is the minimum cycle time of the design in (c)?

Answer all parts of the problem in the space provided below.

(13,35)
(11,28)
(13,28)

(a) Figure provided

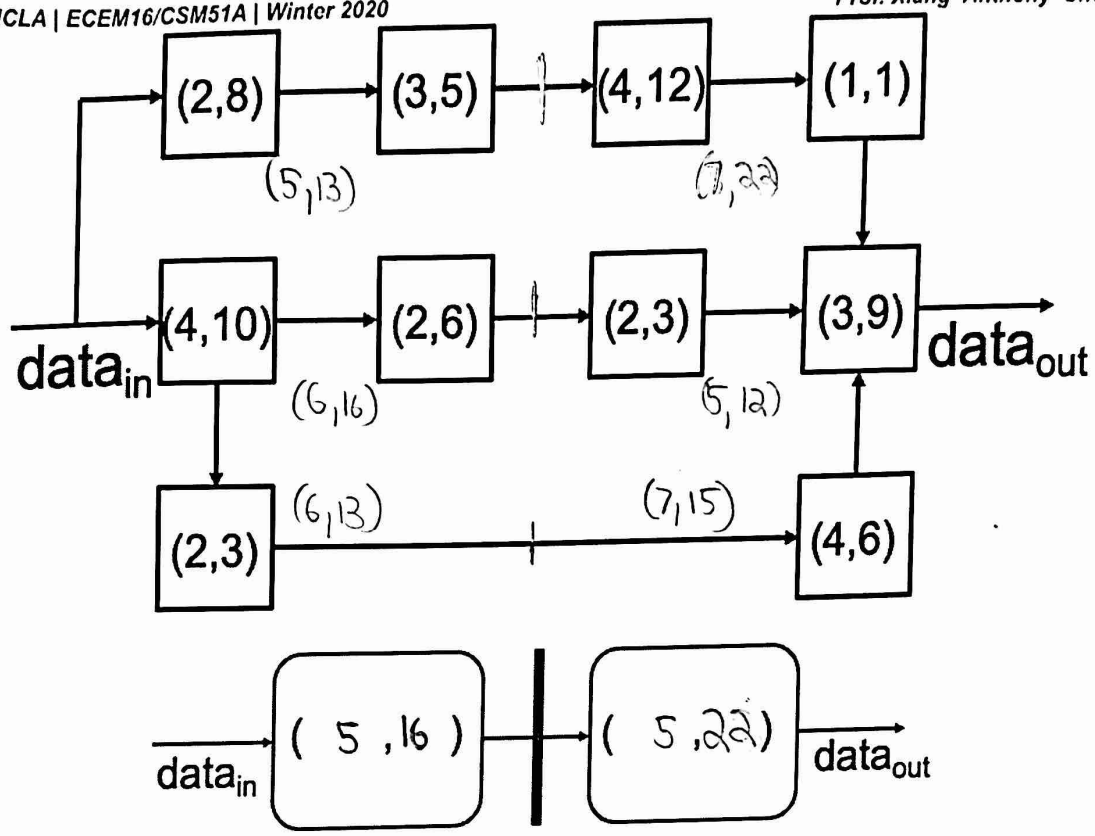


(b) Cycle time: $t_{cy} \geq t_{dCQ} + t_{dMax} + t_s$

$t_{cy} \geq 3 + 35 + 1$

$t_{cy} \geq 39$

(c) Figure provided for the answer.



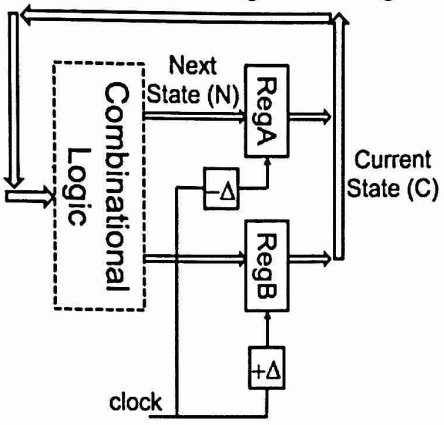
(d) Cycle time = $t_{cy} \geq t_{dca} + t_{dMax} + t_s$

$t_{cy} \geq 3 + 22 + 1$

$t_{cy} \geq 26$

Problem #3

The following shows a conceptual state machine. The registers have the following characteristics: $(t_{cCLK\text{-}Q}, t_{dCLK\text{-}Q}) = (1,5)$, $t_{setup} = 2$ and $t_{hold} = -1$.
(Note: the hold time is NOT a mistake. Hold time can be negative, meaning the input to the register can be changed even before the clock edge and still get registered properly.)



(a) If the combinational logic has delay of $(t_{cCN}, t_{dCN}) = (2,10)$, and clock skew of $\Delta=0$, what is the minimum cycle time?

(b) With the same delay as in (a), is there a hold time violation? (Show your work)

(c) What is the maximum amount of clock skew, Δ, such that there would be a hold time violation? (Show your work)

(d) Assuming the $\Delta = 5$, what is the new minimum cycle time?

a) $t_{cy} \geq t_{dCLKQ} + t_{dmax} + t_s$
$t_{cy} \geq 5 + 10 + 2 = 17$

b) $t_h \leq t_{ccq} + t_{cmn}$
$t_h \leq 1 + 2$
$t_h \leq 3$
$-1 \leq 3$, so no hold time violation

c) $t_h \leq t_{cca} + t_{cmn} - t_K$
$t_h \leq 1 + 2 - t_K$
$-1 \leq 3 - t_K$
$t_K \leq 4$ $\Delta_{max} = 4$, s.t. $\exists$ a hold time violation

(IMPORTANT: Keep this page in submission even if left unused)

d) $\Delta = 5$, new minimum cycle time?

$$t_{cy} \geq t_{dCQ} + t_{dmux} + t_S + t_K$$

$$t_{cy} \geq 5 + 10 + 2 + 5$$

$$t_{cy} \geq 22$$