Q1: (PLD)

(10) Tabulate the PLA programming table for the four Boolean functions listed below. Minimize the number of product terms.

$A(x,y,z) = \Sigma\,(2,3,4,6) = x'y + xz'$

$B(x,y,z) = \Sigma\,(1,3,4,6) = x'z + xz'$

$C(x,y,z) = \Sigma\,(1,2,3) = x'z + x'y$

$D(x,y,z) = \Sigma\,(2,3,4,5) = x'y + xy'$

Product Term	Inputs			Outputs			
	x	y	z	A	B	C	D
x'y	0	1	-	1	-	1	1
xz'	1	-	0	1	1	-	-
x'z	0	~~0~~ -	1	-	1	1	-
xy'	~~0~~ 1	0	-	-	-	-	1

Specify the size of a ROM (number of words and number of bits per word) that will accommodate the truth table for the following combinational circuit components:

(4) i. A magnitude comparator that compares two 8-bit binary numbers and generates 1 if they are equal and 0 if they are not.

2^{16} X 1 ROM

(4) ii. A 16-bit adder–subtractor (including a carry-in/carry-borrow)

2^{33} X 17 ROM

Q2: (Basic Computer)

(a) Memory and register contents of the Basic Computer is as follows (everything is in base 16):

Memory:

011	0000
021	0083
022	7800
083	B8F2
084	7400
8F2	F800

B8F2 = 1011 1000 1111 0010

A937= 1010 1001 0011 0111

^ ________________________________

A832= 1010 1000 0011 0010

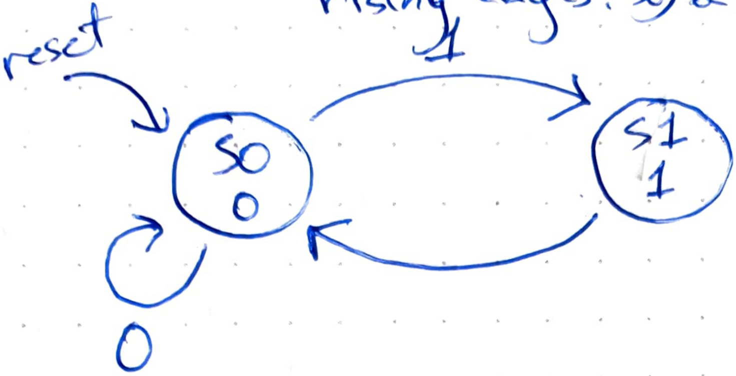
PC: 021 **AC:** A937

Go over the Fetch and execute cycles and determine the content of the following registers at the end of the execute cycle:

PC: 022 **AC:** A832 **IR:** 0083

Ceng232 Final - Meta 2019-2020 Spring

Q3 a) Question clearly indicates changes occur only on rising edges. So a moore machine is designed



b)

Current state	Next State input=0	Next State input=1	Output (input independent)
a	b	e	0
b	e	c	1
c	d	e	0
d	e	a	1
e	b	a	0

a & e are equivalent because their outputs are the same AND their target states are equivalent. Notice how they'd be same if you swapped a & e. Next iteration:

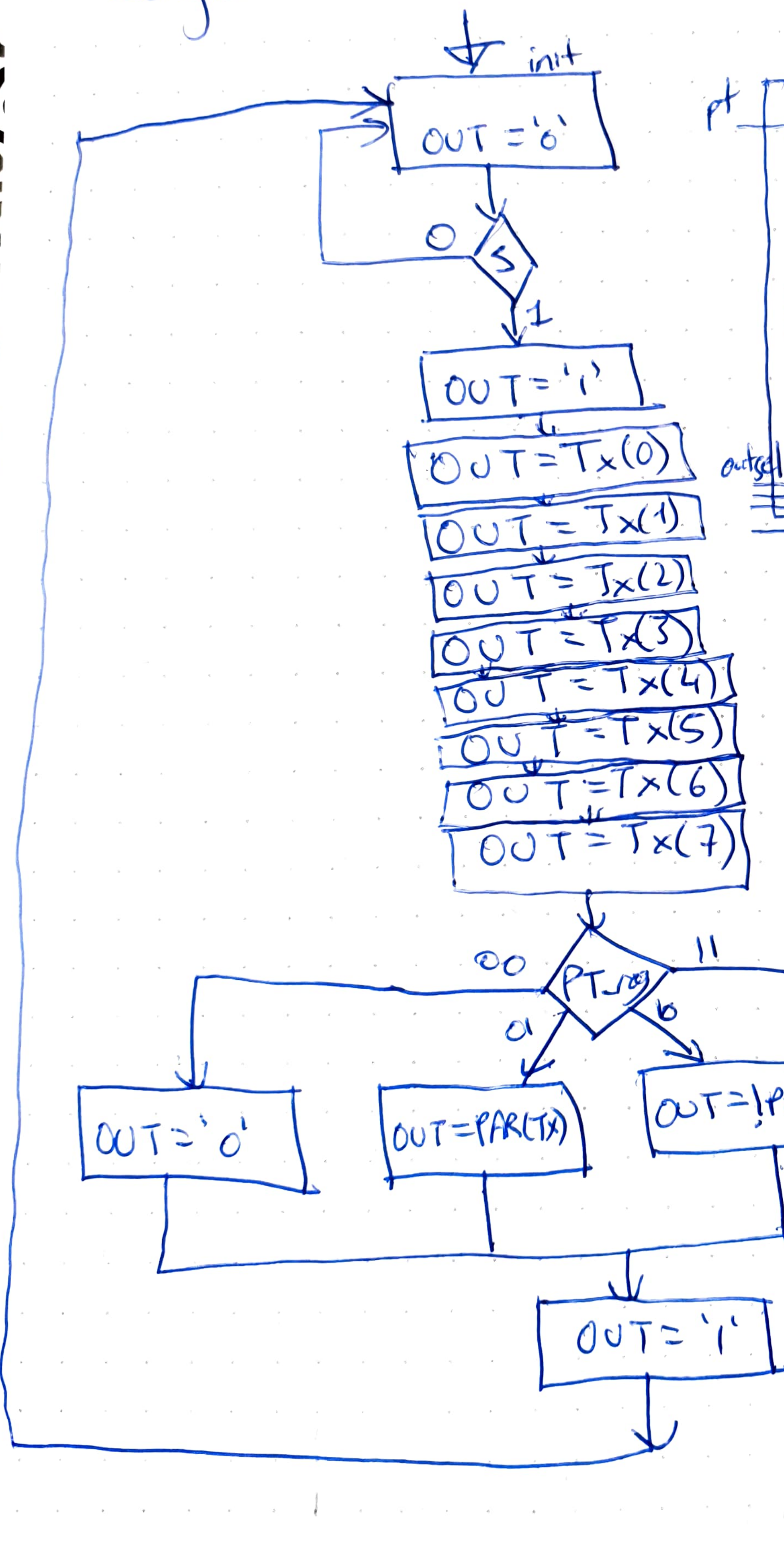
a	b	a	0
b	a	c	1
c	d	a	0
d	a	a	1

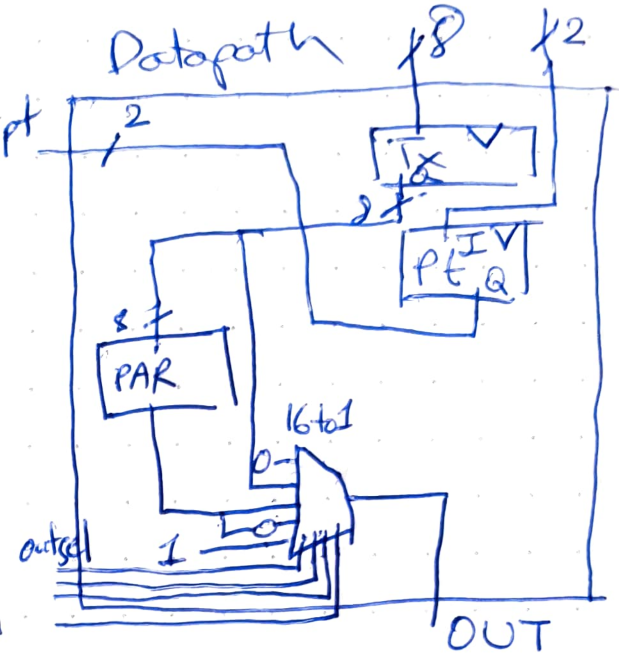
Potentially equivalent states are a & c, b & d. However target states differ. No further redux. Binary encoded state to output table:

00	01	00	0
01	00	10	1
10	11	00	0
11	00	00	1

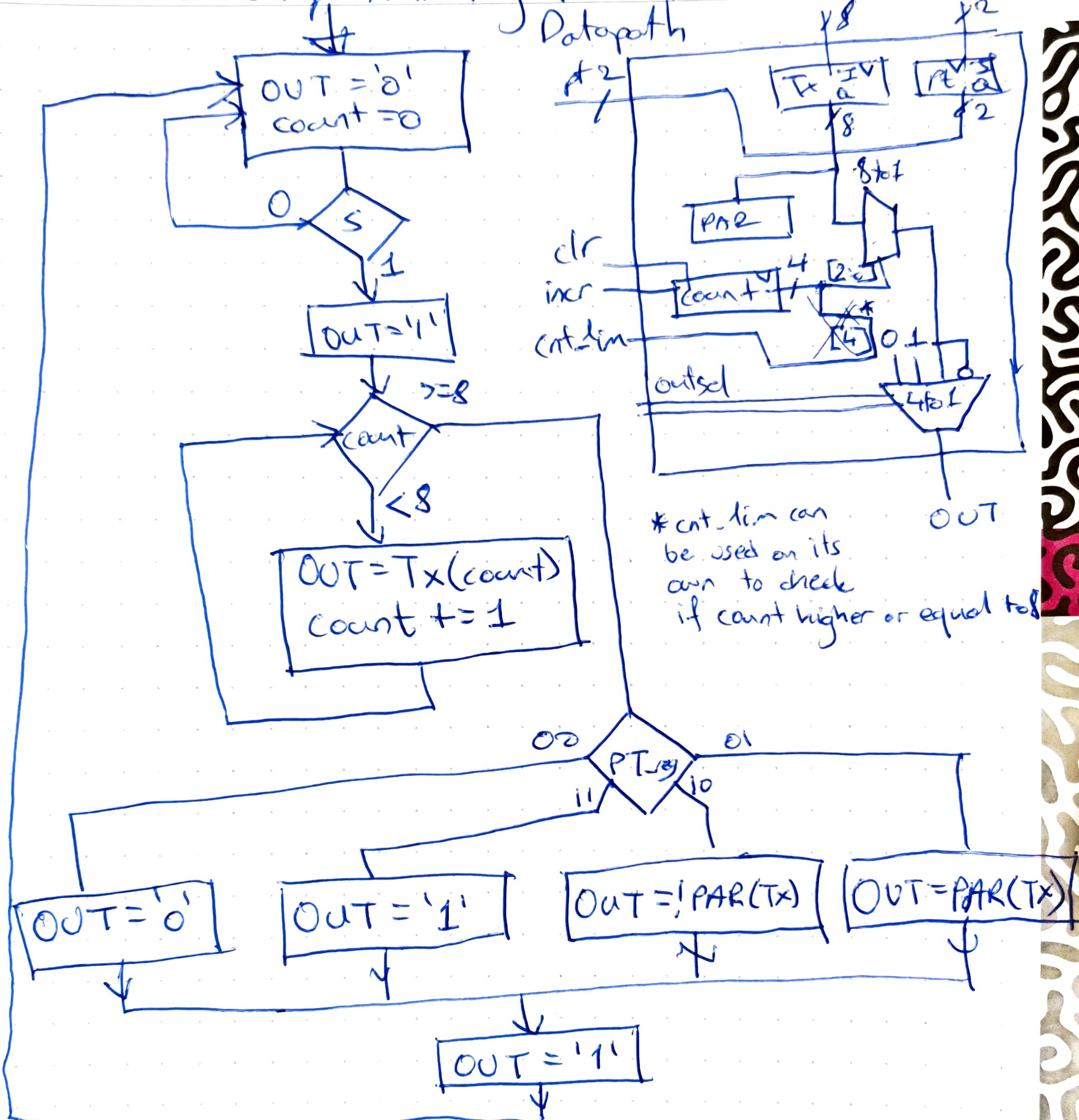
2019-2020 Metu

Ceng232 Final Q4 Soln 1





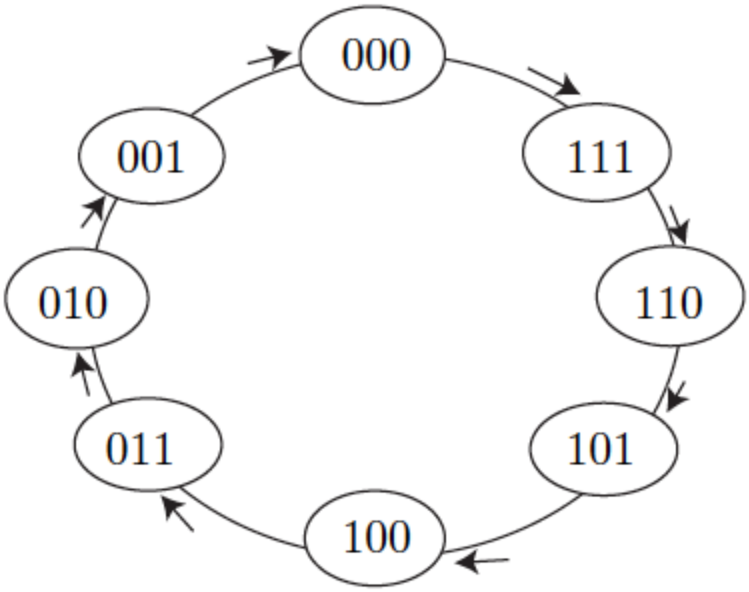
Ceng232 Final Q4 Soln2 (alternative)
(a bit more complex in my opinion)



*cnt_lim can be used on its own to check if count higher or equal to 8

Q4.

a)



b)

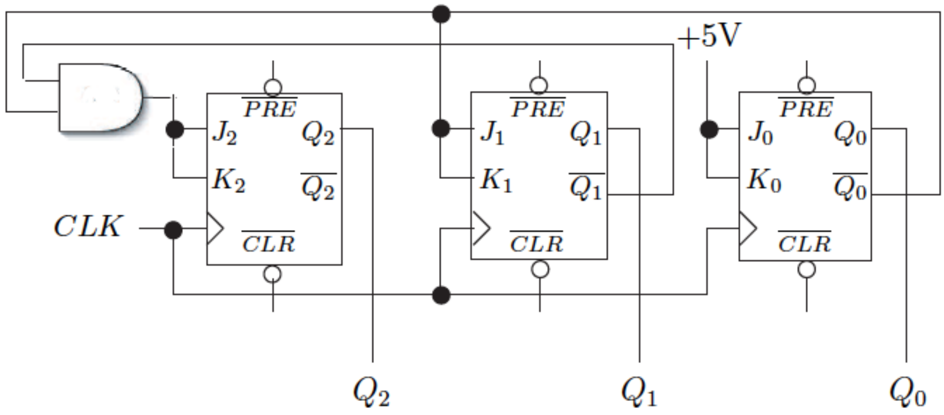
Present State			Next State			Inputs					
Q_2	Q_1	Q_0	Q_2^*	Q_1^*	Q_0^*	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	1	1	1	1	X	1	X	1	X
0	0	1	0	0	0	0	X	0	X	X	1
0	1	0	0	0	1	0	X	X	1	1	X
0	1	1	0	1	0	0	X	X	0	X	1
1	0	0	0	1	1	X	1	1	X	1	X
1	0	1	1	0	0	X	0	0	X	X	1
1	1	0	1	0	1	X	0	X	1	1	X
1	1	1	1	1	0	X	0	X	0	X	1

Note that J_0 and K_0 are either 1 or X (don't care), so they may be set high; $J_0 = K_0 = 1$.

Also note that J_1 and K_1 are 1 or X whenever $Q_0 = 0$. Thus $J_1 = K_1 = \overline{Q_0}$. This could also be determined from a 3-variable K-map of J_1, with variables Q_2, Q_1 and Q_0.

Similarly, note that J_2 and K_2 are both 1 or X whenever both Q_1 and Q_0 are 0. Their equation is thus found as $J_2 = K_2 = \overline{Q_1} \cdot \overline{Q_0}$. Again, this could also be found from a 3-variable K-map of J_2, with variables Q_2, Q_1 and Q_0.

c)



+5V
$\overline{PRE}$
$\overline{PRE}$
$\overline{PRE}$
J_2
Q_2
K_2
$\overline{Q_2}$
J_1
Q_1
K_1
$\overline{Q_1}$
J_0
Q_0
K_0
$\overline{Q_0}$
CLK
$\overline{CLR}$
$\overline{CLR}$
$\overline{CLR}$
Q_2
Q_1
Q_0