Name Kristen Chui

First Last

Student ID # 104289172

University of California

Los Angeles

Computer Science Department

CSM51A/EEM16 Midterm Exam #2

Winter Quarter 2015

February 23rd 2015

This is a closed book exam. Absolutely nothing is permitted except pen, pencil and eraser to write your solutions. Any academic dishonesty will be prosecuted to the full extent permissible by university regulations.

Time allowed 100 minutes.

Problem (possible points)	Points
1 (20)	18
2 (20)	20
3 (20)	17
4 (20)	17
5 (20)	12
Total (100)	84

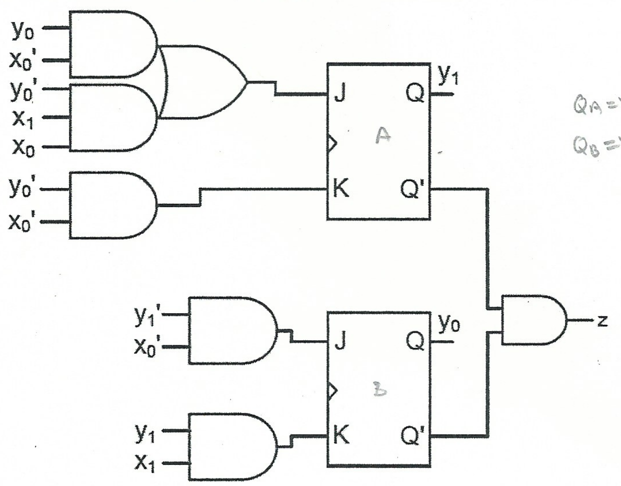
38

46

18

Problem 1 (20 points)

Obtain a high-level description (state transition table) of the network shown in the figure below. The system has two input bits x_1 and x_0, with a single output bit z.



$Q_A = Y_1$

$Q_B = Y_0$

$Q(t+1) = Q_A K' + Q_A' J$

$J_A = X_0' Y_0 + X_0 X_1 Y_0'$ $\qquad$ $J_B = X_0' Y_1'$ $\qquad$ $Z = Q_B' + Q_A'$

$K_A = X_0' Y_0'$ $\qquad$ $K_B = X_1 Y_1$

$Q_A(t+1) = Q_A (X_0' Y_0')' + Q_A' (X_0' Y_0 + X_0 X_1 Y_0')$

$= Q_A (X_0 + Y_0) + Q_A' (X_0' Y_0 + X_0 X_1 Y_0')$

$= Q_A X_0 + Q_A Y_0 + Q_A' X_0' Y_0 + Q_A' X_0 X_1 Y_0'$

$= Q_A X_0 + Y_0 (Q_A + Q_A' X_0') + Q_A' X_0 X_1 Y_0'$

$= Q_A X_0 + Q_A Y_0 + X_0' Y_0 + Q_A' X_0 X_1 Y_0'$

$Q_B(t+1) = Q_B (X_1 Y_1)' + Q_B' (X_0' X_1')$

$= Q_B (X_1' + Y_1') + Q_B' X_0' X_1'$

$= Q_B X_1' + Q_B Y_1' + Q_B' X_0' X_1'$

$= X_1' (Q_B + Q_B' X_0') + Q_B Y_1'$

$= Q_B X_1' + X_1' X_0' + Q_B Y_1'$

$Q_A(t+1) = X_0 Y_1 + Y_1 Y_0 + X_0' Y_0 + Y_1' X_0 X_1 Y_0'$

$Q_B(t+1) = Y_0 X_1' + X_1' X_0' + Y_0 Y_1'$

$Z = Y_1' Y_0'$

-2

	$Y_1 Y_0$ PS	$X_1 X_0$ 00	01	10	11
S_1	0 0	01,1	00,1	00,1	10,1
S_2	0 1	01,0	01,0	11,0	01,0
S_3	1 0	00,0	10,0	00,0	10,0
S_4	1 1	11,0	11,0	10,0	10,0

NS, z

binary

PS	name
00	S_1
01	S_2
10	S_3
11	S_4

input	
00	a
01	b
10	c
11	d

output	
0	N
1	Y

PS	a	b	c	d
S_1	S_2,Y	S_1,Y	S_1,Y	S_3,Y
S_2	S_2,N	S_2,N	S_4,N	S_2,N
S_3	S_1,N	S_3,N	S_1,N	S_3,N
S_4	S_4,N	S_4,N	S_3,N	S_3 N

NS, z

<high level state transition table

20

Problem 2 (20 points)

Consider the state transition table as shown below. Add 3 new states and their transitions to the table, so that the new table will have 5 states after minimization.

	INPUT	
PS	x=0	x=1
A	B,0	C,0
B	B,0	D,0
C	B,0	A,0
D	C,0	E,1
E	E,1	F,1
F	F,1	E,1

Current table

1eq: {A,B,C} 1
{D} 2
{E,F} 3

2eq:

PS	x=0	x=1
A	1	1
B	1	2
C	1	1

PS	x=0	x=1
E	3	3
F	3	3

{A,C} 1 {B} 2 {D} 3 {E,F} 4

3eq:

PS	x=0	x=1
A	2	1
C	2	1

PS	x=0	x=1
E	4	4
F	4	4

minimized table ⇒ {A,C}, {B} {D} {E,F} = 4 states

To make 5 states after minimization by adding 3 new states:

minimized = {A,C}, {B}, {D} {E,F}, {G,H,I}

PS	x=0	x=1
A	B,0	C,0
B	B,0	D,0
C	B,0	A,0
D	C,0	E,1
E	E,1	F,1
F	F,1	E,1
G	H,1	G,0
H	G,1	G,0
I	I,1	I,0

NS, Z

1eq: 0,0 {A,B,C} 1
0,1 {D} 2
1,1 {E,F} 3
1,0 {G,H,I} 4

2eq:

PS	x=0	x=1
A	1	1
B	1	2
C	1	1

PS	x=0	x=1
E	3	3
F	3	3

	x=0	x=1
G	4	4
H	4	4
I	4	4

→ {A,C} 1, {B} 2 {D} 3 {EF} 4, 5 {G,H,I}

3 eq:

PS	x=0	x=1
A	2	1
C	2	1

PS	x=0	x=1
E	4	4
F	4	4

PS	x=0	x=1
G	5	5
H	5	5
I	5	5

17

Problem 3 (20 points)

Using RD flip-flops as defined below, design a system as described below. Use only multiplexers to implement your combinational logic.

Input set: ~~{a, b, c}~~ {a, b, c, d}

Output: 1, if x(t-n, t) = a[b|c]*d+a

0, otherwise

Note: * denotes a character can appear 0 to infinite number of times.

+ denotes a character can appear 1 to infinite number of times.

b|c denotes b or c.

For example, given abcbdda, the output should be 1.

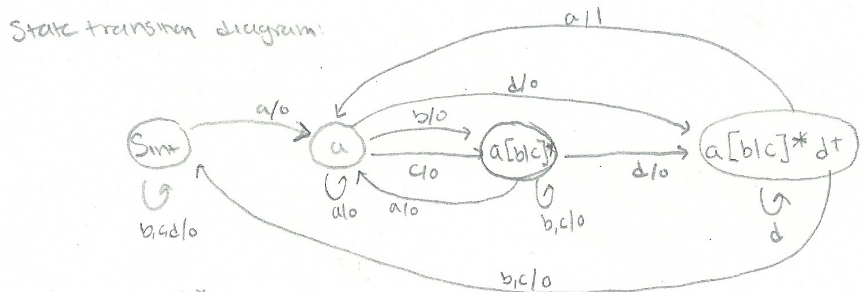
	RD			
PS, Q(t)	00	01	10	11
0	1	0	0	1
1	1	0	1	0
	NS, Q(t+1)			

PS	RD 00	01	10	11
0	1	0	0	1
1	1	0	1	0
	NS			

	RD
0→0	01, 10
0→1	00/11
1→0	−1
1→1	−0

+5

State transition diagram:

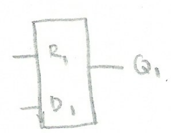


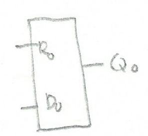
PS
Sint = 00
a = 01
a[b|c]* = 10
a[b|c]* d+ = 11

$x_1 x_0$
a = 00
b = 01
c = 10
d = 11

	$Q_1 Q_0$ PS	a 00	b 01	c 10	d 11
Sint	00	01,0	00,0	00,0	00,0
a	01	01,0	10,0	10,0	11,0
a[b|c]*	10	00,0	10,0	10,0	11,0
a[b|c]*d+	11	01,1	00,0	00,0	11,0
		NS, Z			

) = +3





R_1

Q_1Q_0 \ x_1x_0	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	−	−	−	−
10	−	−	−	−

R_0

0	0	0	0
−	−	−	−
−	−	−	−
0	0	0	0

use
0→1 : 00
0→0 : 01

+5

Z

0	0	0	0
0	0	0	0
1	0	0	0
0	0	0	0

$Z = Q_1 Q_0' x_1' x_0'$

use
10 when 0→0
&
11 when 0→1

D_1

0	0	0	0
0	1	1	1
1	1	1	0
1	0	0	0

D_0

0	1	1	1
0	1	0	1
0	1	0	1
1	1	0	1

$R_1 = \cancel{Q_1'} = 1$

$D_1 = Q_1 x_1' x_0' + Q_0 x_0 + Q_1' Q_0 x_1$

$R_0 = 0$

$D_0 = x_0 x_1' + x_1 x_0' + Q_1' Q_0' x_0 + Q_1 Q_0' x_1'$

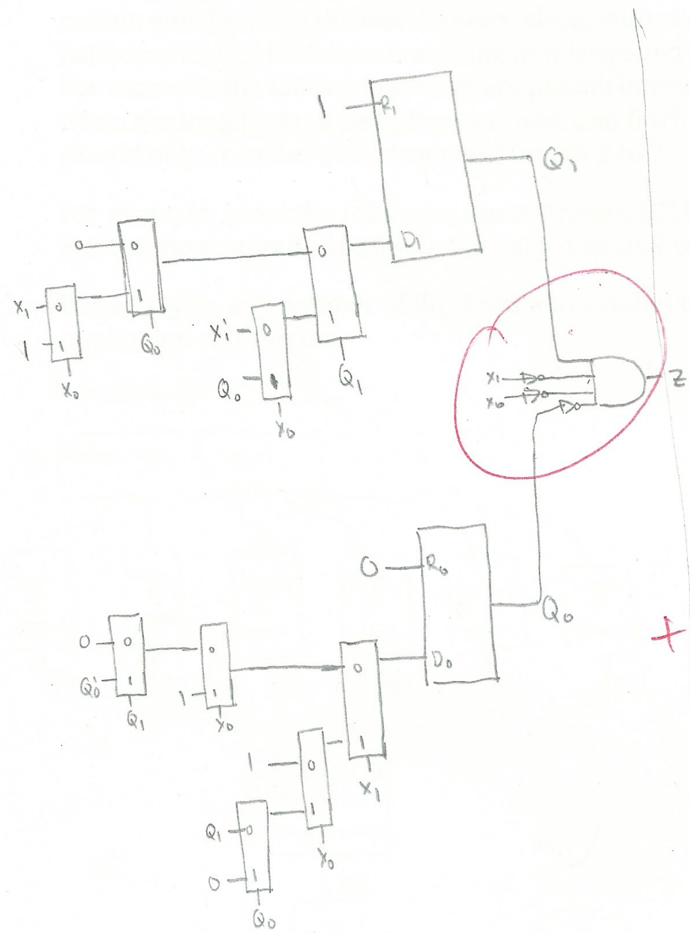
Problem 3) Extra Page

$R_1 = 1$

$D_1 = Q_1 x_1' x_0' + Q_0 x_0 + Q_1' Q_0 x_1$

$R_0 = 0$

$D_0 = x_0 x_1' + x_1 x_0' + Q_1' Q_0' x_0 + Q_1 Q_0' x_1'$



$D_1 = Q_1 x_1' x_0' + Q_0 x_0 (Q_1 + Q_1') + Q_1' Q_0 x_1$

$Q_1 (x_1' x_0' + Q_0 x_0) + Q_1' (Q_0 x_0 + Q_0 x_1)$

$\downarrow \qquad\qquad \downarrow$

$x_0 (Q_0) + x_0' (x_1') \qquad Q_0 (x_0 + x_1) + Q_0' (0)$

$\downarrow$

$x_0 (1 + x_0' (x_1)$

$D_0 = x_0 x_1' + x_1 x_0' + Q_1' Q_0' x_0 + Q_1 Q_0' x_1'$

$x_0 x_1' + x_1 x_0' + Q_1' Q_0' x_0 (x_1 + x_1') + Q_1 Q_0' x_1'$

$x_1 (x_0' + Q_1' Q_0' x_0) + x_1' (x_0 + Q_1' Q_0' x_0 + Q_1 Q_0')$

$\downarrow \qquad\qquad \downarrow$

$x_0 (Q_1' Q_0') + x_0' (1) \qquad x_0 (1 + Q_1' Q_0' x_0 + Q_1 Q_0')$

$\downarrow \qquad\qquad + x_0' (Q_1 Q_0')$

$Q_0 (0) + Q_0' (Q_1) \qquad \downarrow$

$Q_1 (Q_0') + Q_1' (0)$

+4

17

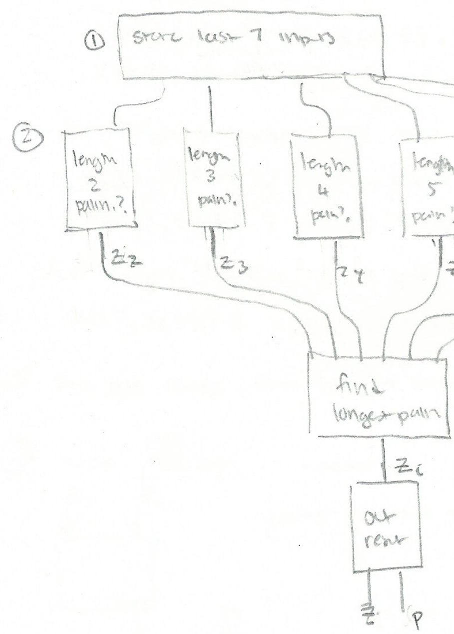
Problem 4 (20 points)

Given an input stream of 0s and 1s, design a system that outputs the length, **Z**, of the largest palindrome found in the last 7 inputs, along with the parity, **P**, of the length of that palindrome. A palindrome is a string that is spelled the same forwards as it is backwards. For example, the following strings are palindromes: 10101, 11, 1001, 0000. **P** is equal to 1 when the length of the palindrome is odd, and 0 when its length is even. Your system should only consider palindromes of length 2 to 7.

For example, given the following input stream, 1010101, the output should be Z=7 and P=1. For the input stream, 1010000, the output should be Z=4 and P=0.

Use any type, any number of flip-flops and combinational gates of your choosing to implement this system.

high level implementation



Z = length of longest palindrome

P = parity of palindrome

+5

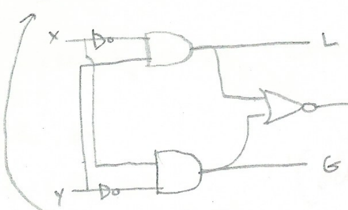
x(t-6) x(t-5) x(t-4) x(t-3) x(t-2) x(t-1) x(t)

Z_{2-7} : output length in binary (3 bits) if palindrome

else output 0

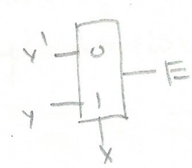
$x'y = L \quad (x'y + xy')' = E \quad xy' = G$

How to implement 1 bit comparator?

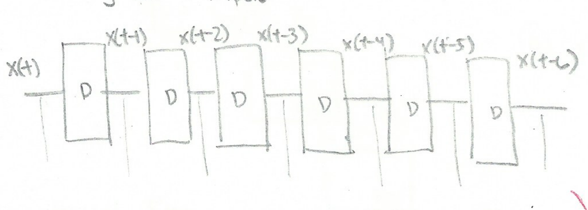


$E = (x+y')(x'+y)$

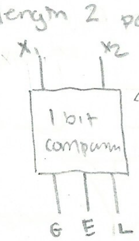
$= xy + x'y'$



① storing last 7 inputs

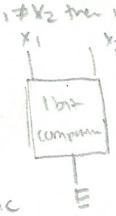


② length 2 palin?



If $X_1 = X_2$ then palindrome (E=1)

If $X_1 \neq X_2$ then not palindrome (G,L=1) E=0



if E=1 → Z_2=010

else Z_2=000

X_1 & X_2 must cycle through 6 times using these as inputs:

- X(t), X(t+1)
- X(t+1), X(t+2)
- X(t-2), X(t-3)
- X(t-3), X(t-4)
- X(t-4), X(t-5)
- X(t-5), X(t-6)

t   t-1   t-2   t-3   t-4   t-5   t-6

Problem 4) Extra Page

② cont for length 3, use 1 bit comparator to compare first and last digits of length 3 string need to cycle 4 times,

inputs
- $x(t)$, $x(t-2)$ · $x(t-2)$, $x(t-4)$ · $x(t-4)$, $x(t-6)$
- $x(t-1)$, $x(t-3)$ · $x(t-3)$, $x(t-5)$

If $E=1$ $Z_3 = 011$ otherwise $Z_3 = 000$

- for length 4: use 2 1 bit comparators with inputs
  - $x(t)$, $x(t-3)$ & $x(t-1)$, $x(t-2)$ · $x(t-2)$, $x(t-5)$ & $x(t-3)$, $x(t-4)$
  - $x(t-1)$, $x(t-4)$ & $x(t-2)$, $x(t-3)$ · $x(t-3)$, $x(t-6)$ & $x(t-4)$, $x(t-5)$

If $E=1$, $Z_4 = 100$ otherwise $Z_4 = 000$

- for length 5: use 2 1 bit comparator w/ inputs:
  - $x(t)$, $x(t-4)$ & $x(t-1)$, $x(t-3)$
  - $x(t-1)$, $x(t-5)$ & $x(t-2)$, $x(t-4)$
  - $x(t-2)$, $x(t-6)$ & $x(t-3)$, $x(t-5)$

If $E=1$ $Z_5 = 101$ otherwise $Z_5 = 000$

- for length 6: use 3 1 bit comparators w/ inputs:
  - $x(t)$, $x(t-5)$ & $x(t-1)$, $x(t-4)$ & $x(t-2)$, $x(t-3)$
  - $x(t-1)$, $x(t-6)$ & $x(t-2)$, $x(t-5)$ & $x(t-3)$, $x(t-4)$

If $E=1$ $Z_6 = 110$ otherwise $Z_6 = 000$

+5

- for length 7: use 3 1 bit comparators w/ inputs:
  - $x(t)$, $x(t-6)$ & $x(t-1)$, $x(t-5)$ & $x(t-2)$, $x(t-4)$

If $E=1$ $Z_7 = 111$ otherwise $Z_7 = 000$

* In all cases for length comparator, if ever output 1, constantly output 1 afterwards

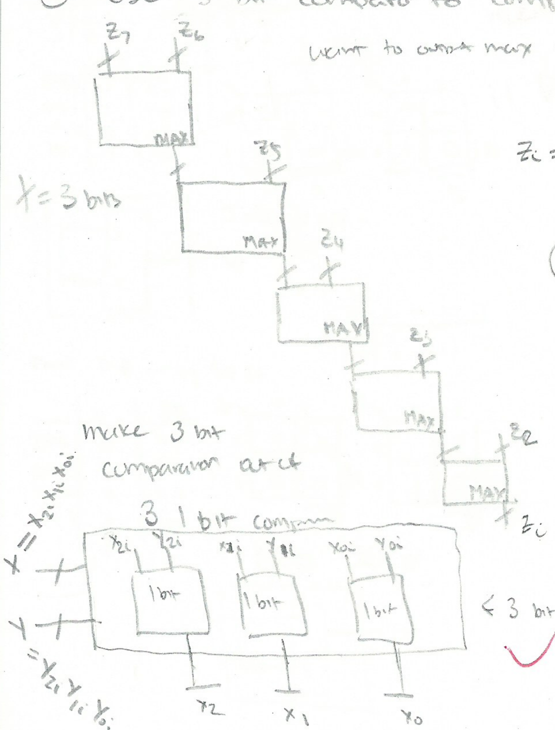
+2

③ Use 3 bit comparator to compare $Z_2, Z_3, Z_4, Z_5, Z_6, Z_7$

want to output max: start by comparing max(Z_7, Z_6)
then compare max$(\ \ ), Z_5)$

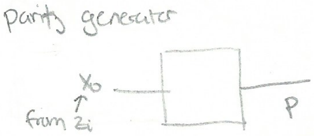
(didn't finish the implementation)

Z_i = 3 bit # of longest length of palindrome $Z_i = \underline{\ \ }_{x_2}\ \underline{\ \ }_{x_1}\ \underline{\ \ }_{x_0}$



← 3 bit comparator

④ parity generator



If $X_0 = 0 \rightarrow$ even $\Rightarrow P = 0$
$X_0 = 1 \rightarrow$ odd $\Rightarrow P = 1$

final output:
$Z = Z_i(x_2 x_1 x_0)$ & $P = x_0$

+5

Problem 5 (20 points)

12

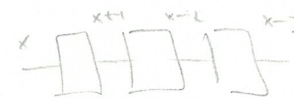
Using at most 1 JK flip-flop, at most 1 SR flip-flop, and at most 8 D flip-flops, design a system as specified below. You may use any gates to implement your combinational logic.

Input set: {0,1}

Output: 1, if x(t, t-3)=11-0, x(t-4, t-7)=1-10 or 0-0-, x(t-8, t-11)=1001

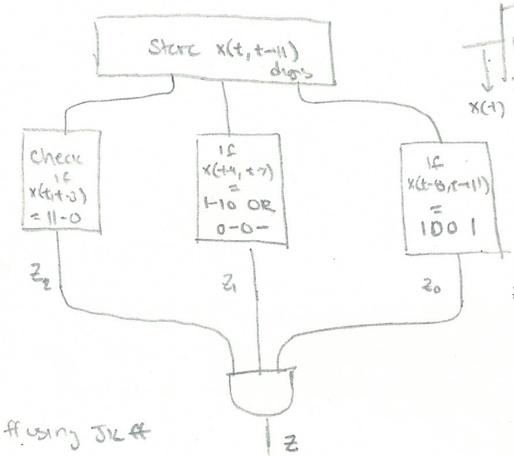
0, otherwise

don't care about x(t-5)

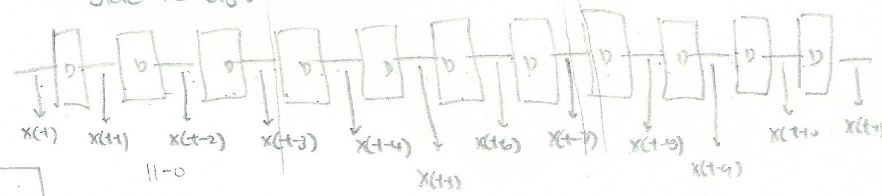


For example, for the given input sequence x(t, t-11)=110010101001, output is 1. For the input sequence x(t, t-11)=101001011001, output is 0.

high level design



Store 12 digits

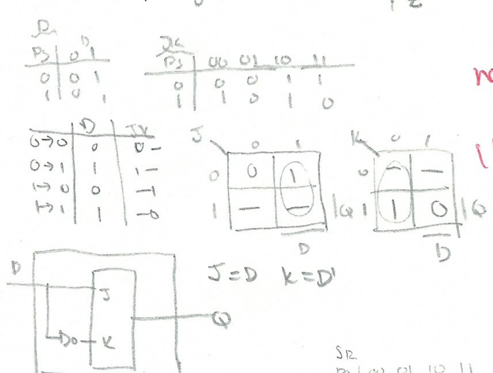


.need 11 D flip flops but only given 8

$Z_2, Z_1, Z_0 = 1$ if recognize pattern, else = 0

To store 4 values needs 3 flip flops

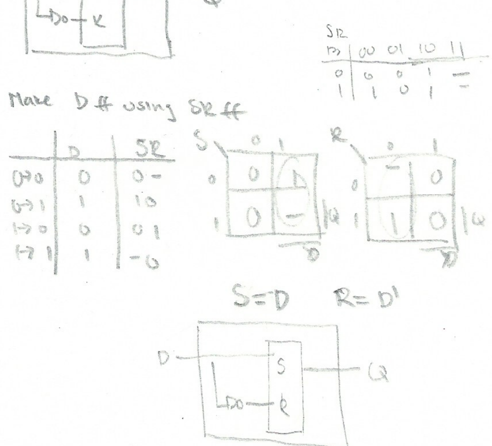
Make D ff using JK ff



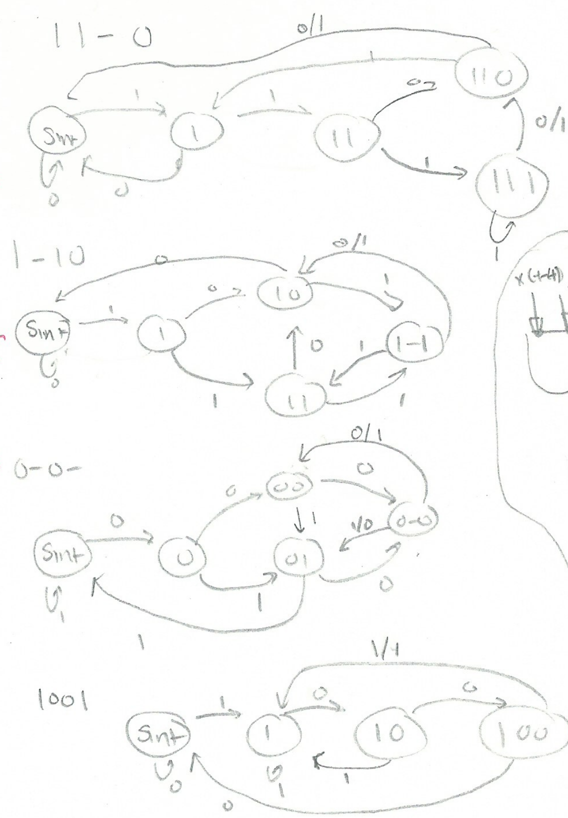
J = D K = D'

need 11 D-FFs

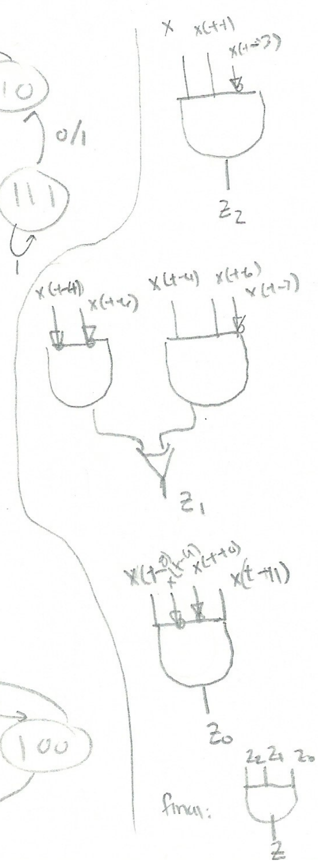
Make D ff using SR ff



S = D R = D'



-8



final: