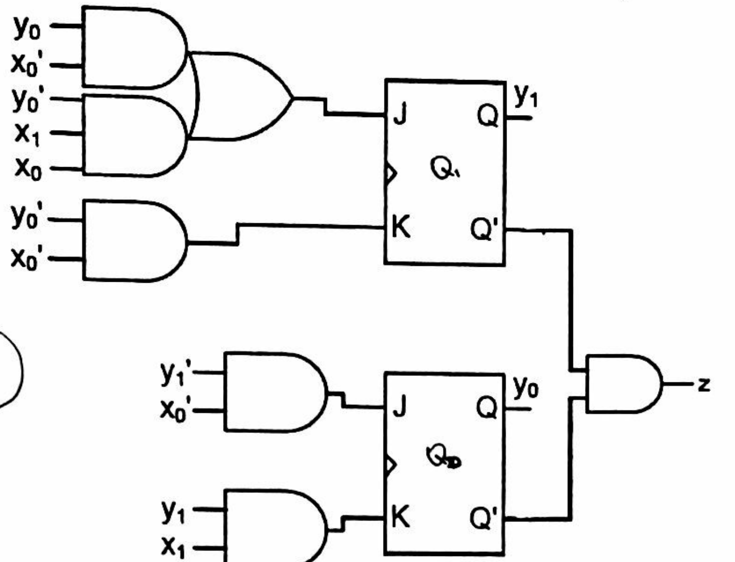
8

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.

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



$Q_1 = y_1(y_0' x_0')' + x_1(y_0 x_0' + y_0' x_1 x_0)$

$= y_1(y_0 + x_0) + x_1 y_0 x_0' + y_0' x_1 x_0$

$Q_0 = y_0(y_1 y_0)' + x_0(y_1' x_0')$

$= y_0(y_1' + y_0') + 0$

$= y_0 y_1'$

PS $y_1 y_0$	x_1x_0 00	01	10	11
0 0				
0 1				
1 0				
1 1				
	NS, Z			

$Q_1 = (y_1)(y_0' x_0')' + y_1'(y_0 x_0' + y_0' x_1 x_0)$

$= y_1(y_0 + x_0) + y_1' y_0 x_0' + y_1' y_0' x_1 x_0$

$Q_0 = y_0(y_1 y_0)' + y_0'(y_1' x_0')$

$= y_0(y_1' + y_0') + y_0' y_1' x_0'$

$= y_0 y_1' + y_0' y_1' x_0'$

$= y_1'(y_0 + y_0' x_0')$

$= y_1'(y_0 + x_0')$

+5

$Q_1 = y_1 y_0 + y_1 x_0 + y_1' y_0 x_0' + y_1' y_0' x_1 x_0$

$Q_1 = x_0(y_1 + y_1' y_0' x_1) + y_0(y_1 + y_1' x_0')$

$Q_1 = x_0(y_1 + y_0' x_1) + y_0(y_1 + x_0')$

$z = y_1' y_0' = (y_1 + y_0)'$

PS $y_1 y_0$	x_1x_0 00	01	10	11
0 0	01, 1			
0 1	11, 0			
1 0	00, 0			
1 1				
	NS, Z			

... complete table using expressions of Q_1, Q_0, and Z

switch table and name inputs to i, j, k, l respectively
name states to a, b, c, d respectively

PS	i	j	k	l
a	b, 1			
b	d, 0			
c	a, 0			
d				
	NS, output			

... complete the table using

+3

$y_1 y_0$
00 → a
01 → b
10 → c
11 → d

$x_1 x_0$
00

Problem 2 (20 points)

20

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
G	F,1	B,1
H	F,1	B,1
I	F,1	B,1

k=1 {A,B,C} {D} {E,F}

k=2 {A,C} {B} {D} {E,F}

k=3 same!

add GHI so that they are k=3 distinguishable with {E,F}

k=2 {A,C} {B} {D} {E,F,G,H,I}

checking: k=1 {A,B,C} {D} {E,F,G,H,I}

k=2 {A,C} {B} {D} {E,F,G,H,I}

k=3 {A,C} {B} {D} {E,F} {G,H,I}

A B D E G

after minimization:

	x=0	x=1
A	B,0	A,0
B	B,0	D,0
D	A,0	E,1
E	E,1	E,1
G	E,1	B,1

20

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.

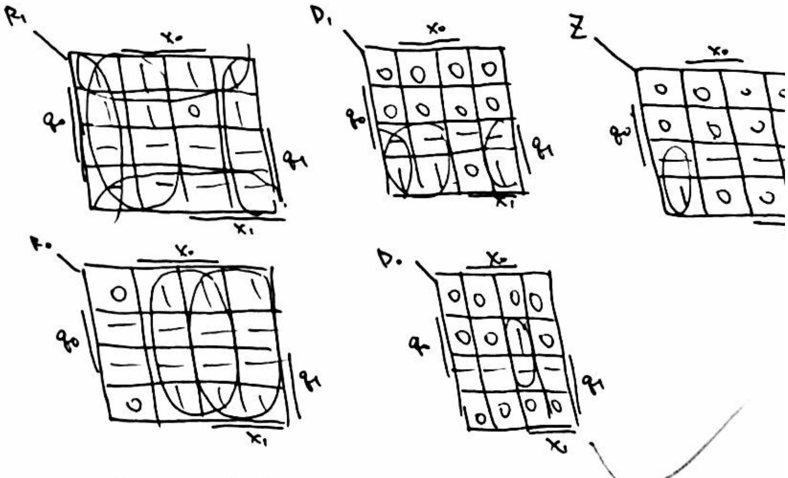
For example, given abcbdda, the output should be 1.

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

RD excitation table

	RD
0→0	10 or 01
0→1	00 or 11
1→0	−1
1→1	−0

q_1q_0	PS	a = 00	b = 01	c = 10	d = 11
		Input X_1X_0			
S_{00}	S_{init}	S_{01}, 0	S_{00}, 0	S_{00}, 0	S_{00}, 0
S_{01}	Sa [b|c]*	S_{01}, 0	S_{01}, 0	S_{01}, 0	S_{10}, 0
S_{10}	Sa [b|c]* d+	S_{01}, 1	S_{00}, 0	S_{00}, 0	S_{10}, 0
S_{11}		—	—	—	—



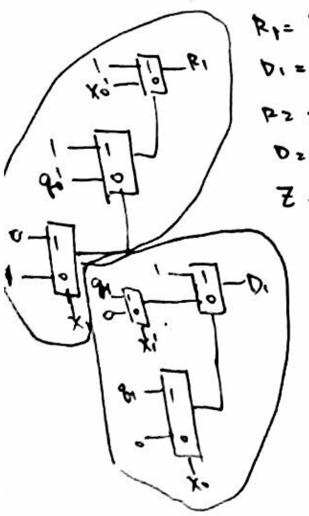
$R_1 = X_1' + q_0' + X_0'$

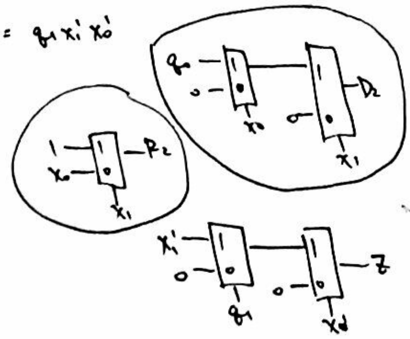
$D_1 = X_1' q_1 + q_1 X_0$

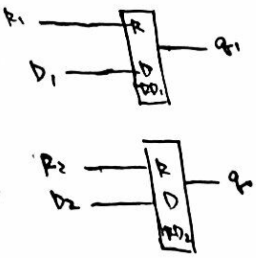
$R_2 = X_0 + X_1$

$D_2 = q_0 X_1 X_0$

$Z = q_1 X_1' X_0'$







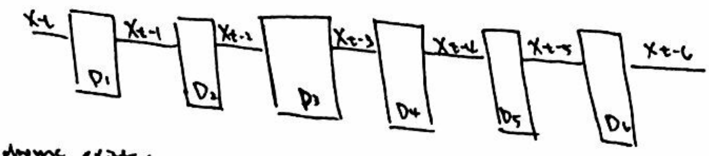
Problem 4 (20 points)

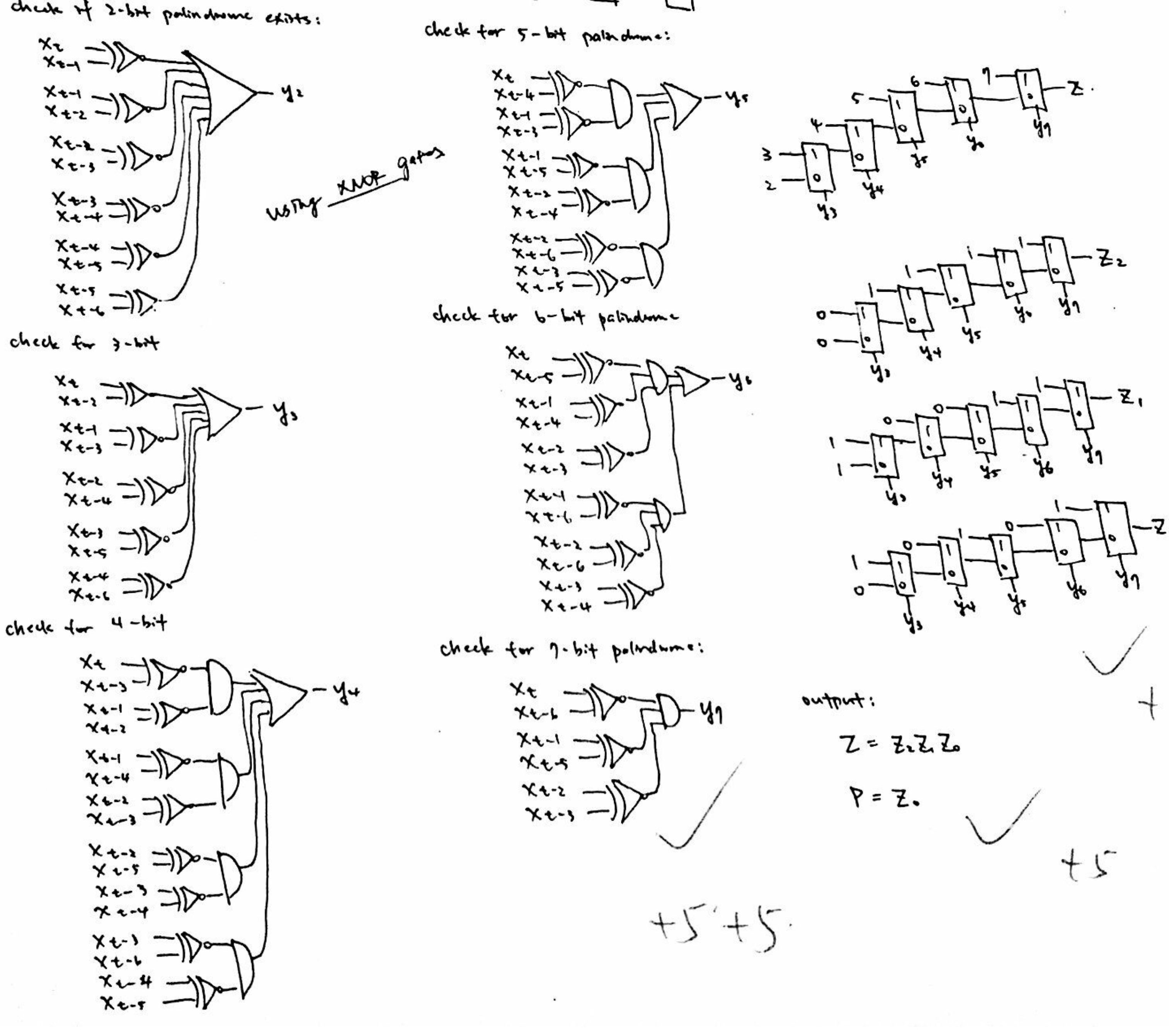
20

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.





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

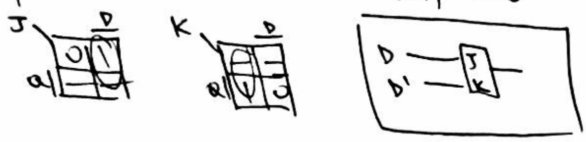
For example, for the given input sequence x(t, t-11)=110010101001, output is 1. For the input sequence x(t, t-11)=101001011001, output is 0.

Implement D ff using JK

PS	D=0	D=1
0	0	1
1	0	1

PS	JK=00	01	10	11
0	0	0	1	1
1	1	0	1	0

	JK
0→0	0-
0→1	1-
1→0	-1
1→1	-0

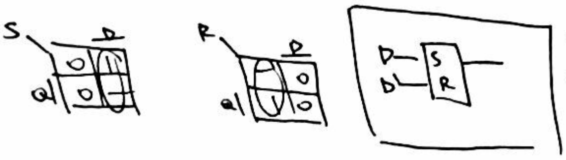


D-ff using JK

Implement D ff using SR

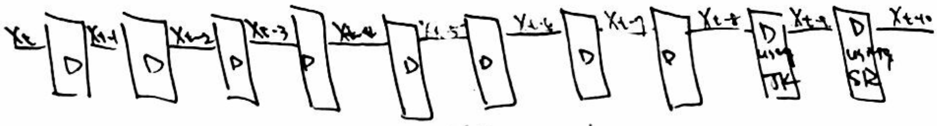
	SR
0→0	0-
0→1	10
1→0	01
1→1	-0

PS	SR=00	01	10	11
0	0	0	1	-
1	1	0	1	-

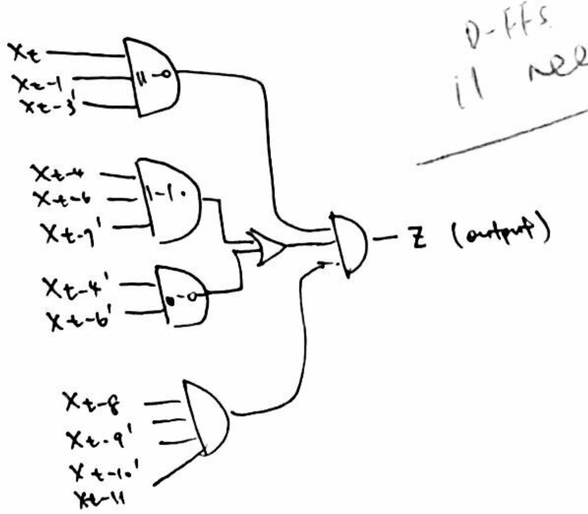


D-ff using SR

Now, combined with the above 2 D-ff, with 8 regular D-ff,



D-FFs 11 needed



To get X_{t-11}, use Q

-8