SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

南方科技大学

DIGITAL DESIGN

ASSIGNMENT REPORT

ASSIGNMENT ID : 4

Student Name: 杨钰城

Student ID: 12112323

PART 1: DIGITAL DESIGN THEORY

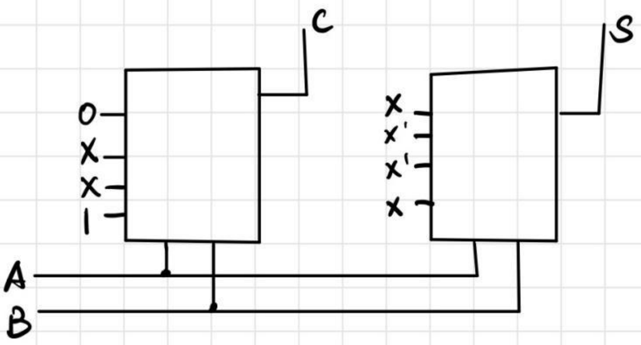
Provide your answers here:

Assignment 4

- (10 points) Implement a full adder with two 4 × 1 multiplexers. Draw the logic diagram.

We use one multiplexers to express Carry "C" and another expresses output "S"

input variables:			output	
A	B	X	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



- (10 points) Design a sequence generator to generate the sequence 110101. Draw the logic diagram.

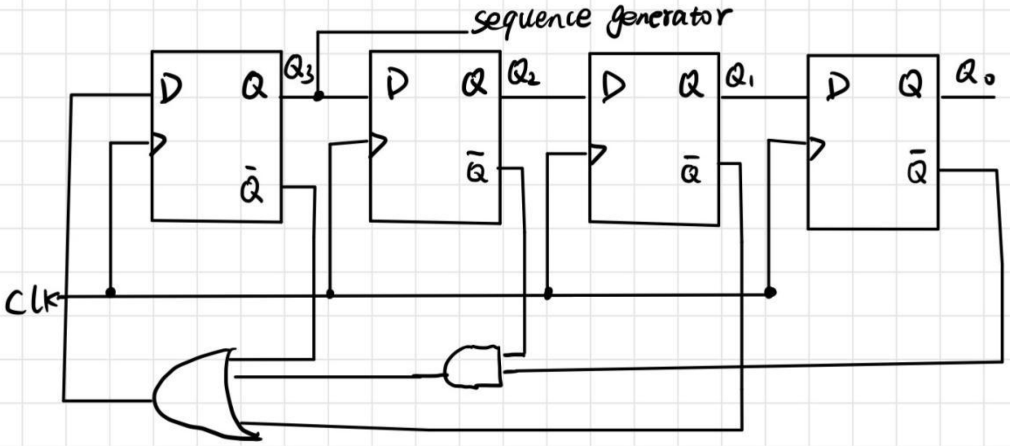
$2^3-1\geq 6$ we need at least three states but 101 apear twice so we need four states.

CLK	Q_3	Q_2	Q_1	Q_0	Z
↑	1	1	0	1	1
↑	1	0	1	0	1
↑	0	1	0	1	1
↑	1	0	1	1	0
↑	0	1	1	1	1
↑	1	1	1	0	0

use K-map

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	x	x	x	x
01	x	1	1	x
11	x	1	x	0
10	x	x	0	1

$$Z = Q_1' + Q_3' + Q_2'Q_0'$$



- (10 points) Design a combinational circuit that generates the 9's complement of a BCD digit. Invalid codes are don't-care conditions. Draw the logic diagram.

input

output

a	b	c	d	A	B	C	D
0	0	0	0	1	0	0	1
0	0	0	1	1	0	0	0
0	0	1	0	0	1	1	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	0	0
0	1	1	0	0	0	1	1
0	1	1	1	0	0	1	0
1	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0

$A = \Sigma(0,1)$

$B = \Sigma(2,3,4,5)$

$C = \Sigma(2,3,6,7)$

$D = \Sigma(0,2,4,6,8)$

$A = a'b'c'$

$B = bc' + b'c = b \oplus c$

$C = c$

$D = d'$

B:

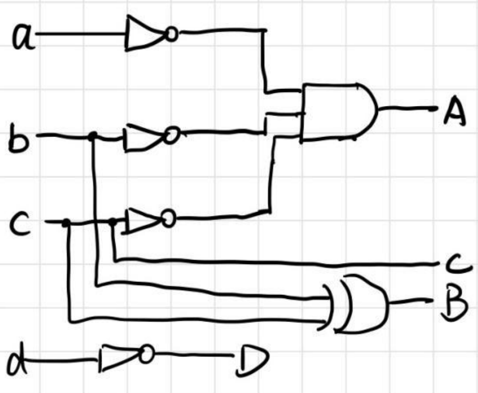
ab\cd	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	X	X	X	X
10	0	0	X	X

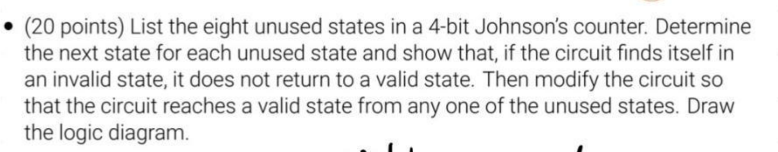
C:

ab\cd	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	x	x	x	x
10	0	0	x	x

D:

ab\cd	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	x	x	x	x
10	1	0	x	x





- (20 points) List the eight unused states in a 4-bit Johnson's counter. Determine the next state for each unused state and show that, if the circuit finds itself in an invalid state, it does not return to a valid state. Then modify the circuit so that the circuit reaches a valid state from any one of the unused states. Draw the logic diagram.

eight unused states

clk	Q_A	Q_B	Q_C	Q_D
X	0	1	0	0
↑	1	0	1	0
↑	1	1	0	1
↑	0	1	1	0
↑	1	0	1	1
↑	0	1	0	1
↑	0	0	1	0
↑	1	0	0	1

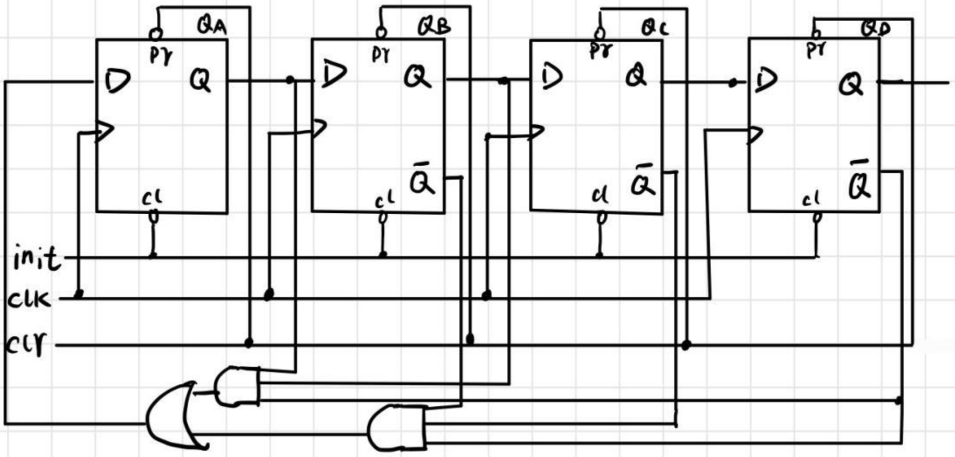
all the states are unused states and they will repeat in the unused states and can't change to the valid states.

clk	Q_A	Q_B	Q_C	Q_D	Z	next state
X	0	1	0	0	0	1010
↑	1	0	1	0	0	1101
↑	1	1	0	1	0	0110
↑	0	1	1	0	0	1011
↑	1	0	1	1	0	0101
↑	0	1	0	1	0	0010
↑	0	0	1	0	0	1001
↑	1	0	0	1	0	0100
↑	0	0	0	0	1	
↑	1	0	0	0	1	
↑	1	1	0	0	1	
↑	1	1	1	0	1	
↑	1	1	1	1	0	
↑	0	1	1	1	0	
↑	0	0	1	1	0	
↑	0	0	0	1	0	

$Z=\Sigma(0,8,9,12,14)$

QAQB \ QCQD	00	01	11	10
00	1	0	0	0
01	0	0	0	0
11	1	0	0	1
10	1	0	0	0

$z = Q_B' Q_C' Q_D' + Q_A Q_B Q_D'$



- (20 points) Design a synchronous counter that has the following sequence: 0010, 0110, 1001, 1000, 1100, 1101, and repeat. From the undesired states the counter must always go to 0010 on the next clock pulse. Draw the logic diagram.

	Prerious state				next state			
	QA	QB	QC	QD	QA	QB	QC	QD
0	0	0	0	0	0	0	1	0
1	0	0	0	1	0	0	1	0
2	0	0	1	0	0	1	1	0
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	0	1	0
5	0	1	0	1	0	0	1	0

6	0	1	1	0	1	0	0	1
7	0	1	1	1	0	0	1	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	0	0	0
10	1	0	1	0	0	0	1	0
11	1	0	1	1	0	0	1	0
12	1	1	0	0	1	1	0	1
13	1	1	0	1	0	0	1	0
14	1	1	1	0	0	0	1	0
15	1	1	1	1	0	0	1	0

$\overline{Q_A} = \Sigma(6, 8, 12)$

$\overline{Q_B} = \Sigma(2, 8, 12)$

$\overline{Q_C} = \Sigma(0, 1, 2, 3, 4, 5, 7, 8, 10, 11, 13, 14, 15)$

$\overline{Q_D} = \Sigma(6, 12)$

$\overline{QA}$

$Q_AQ_B \backslash Q_CQ_D$	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	1	0	0	0
10	1	1	0	0

$QA = Q_AQ_C'Q_D' + Q_AQ_B'Q_C'$
$+ Q_A'Q_BQ_CQ_D'$

$\overline{QB}$

$Q_AQ_B \backslash Q_CQ_D$	00	01	11	10
00	0	0	0	1
01	0	0	0	0
11	1	0	0	0
10	1	0	0	0

$QB = Q_AQ_C'Q_D' + Q_A'Q_B'Q_CQ_D'$

$\overline{Qc}$

$Q_AQ_B \backslash Q_CQ_D$	00	01	11	10
00	1	1	1	1
01	1	1	1	0
11	0	1	1	1
10	0	0	1	1

$Qc = Q_BQ_D + Q_A'Q_B' + Q_A'Q_C' + Q_AQ_C$

$\overline{QD}$

$Q_AQ_B \backslash Q_CQ_D$	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	1	0	0	0
10	0	0	0	0

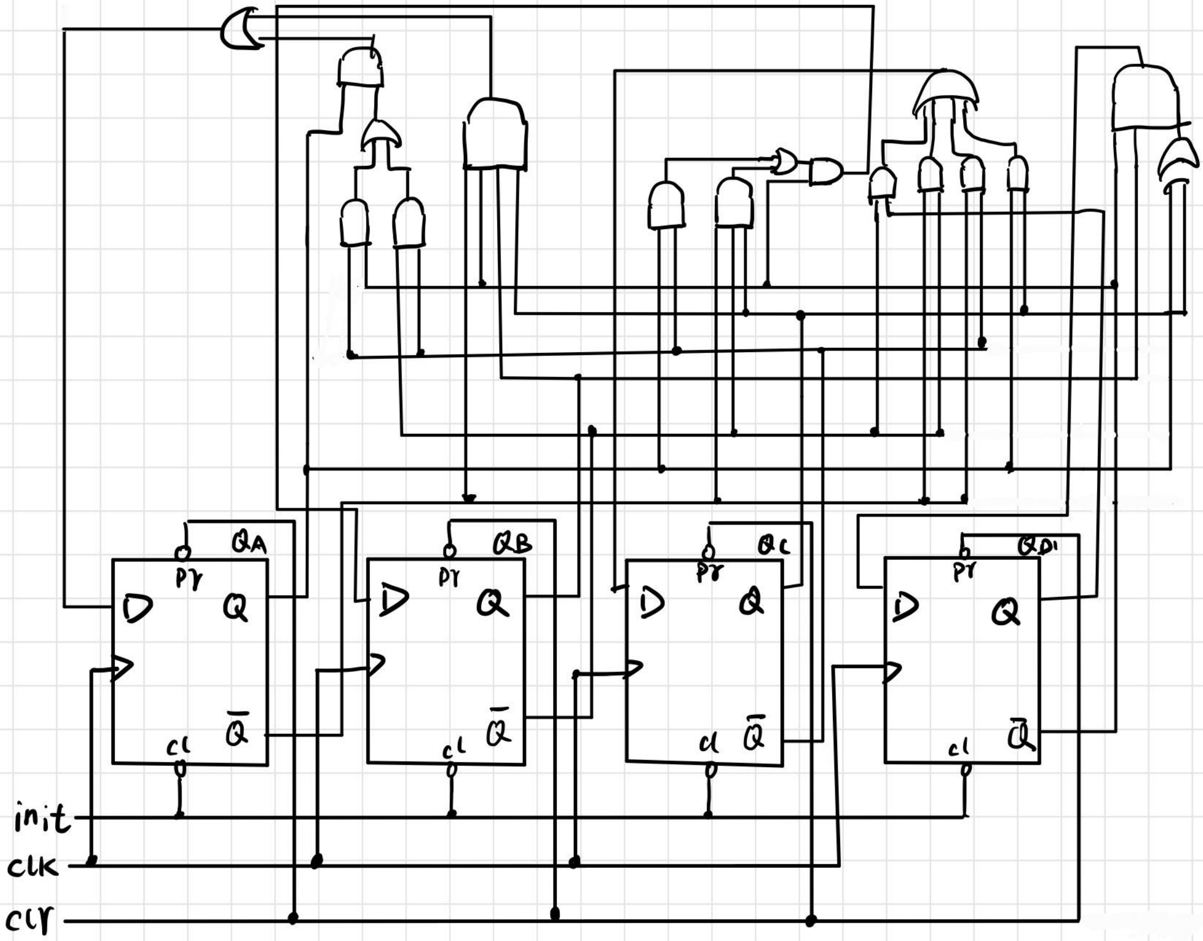
$\overline{QD} = Q_A'Q_BQ_CQ_D' + Q_AQ_BQ_C'Q_D'$

$Q_A = Q_A Q_C' Q_D' + Q_A Q_B' Q_C' + Q_A' Q_B Q_C Q_D'$

$Q_B = Q_A Q_C' Q_D' + Q_A' Q_B' Q_C Q_D' = (Q_A Q_C' + Q_A' Q_B' Q_C) Q_D'$

$Q_C = Q_B Q_D + Q_A' Q_B' + Q_A' Q_C' + Q_A Q_C$

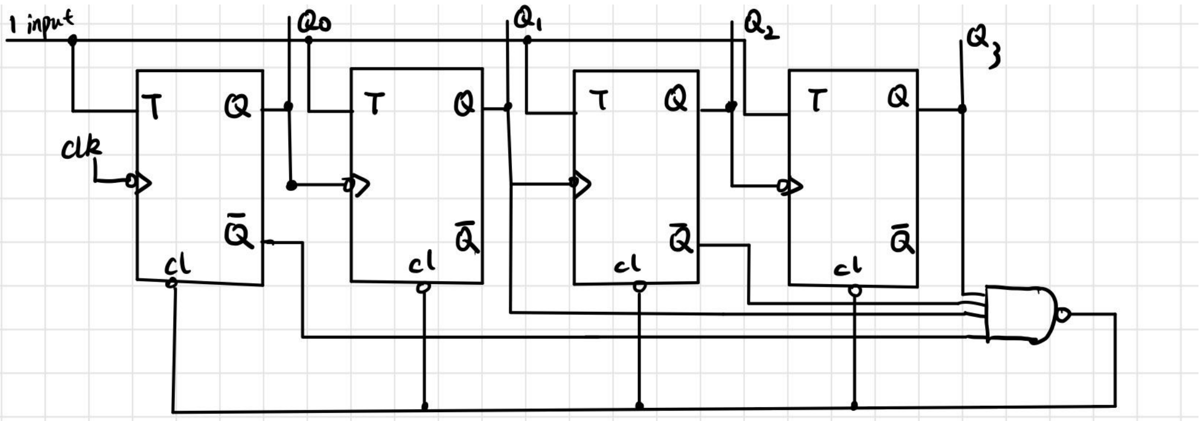
$Q_D = Q_A' Q_B Q_C Q_D' + Q_A Q_B Q_C' Q_D' = (Q_A \oplus Q_C) Q_B Q_D'$



- (15 points) Implement a BCD ripple counter using a four bit binary ripple counter with asynchronous clear and external NAND gates. Draw the logic diagram.

$2^{4-1} < 9 < 2^4$ use four TFF

when $Q_3 Q_2 Q_1 Q_0 = 1010$ the Counter should be rested



- (15 points) Determine the input equations for a BCD counter that uses
  1. only JK flip-flops, and
  2. only D flip-flops.

  Compare these two designs and the one given in the lecture that uses only T flip-flops. Determine which one is the most efficient and why.

JK F-F :

present				next				F-F input							
A_3	A_2	A_1	A_0	A_3	A_2	A_1	A_0	J_{A3}	K_{A3}	J_{A2}	K_{A2}	J_{A1}	K_{A1}	J_{A0}	K_{A0}
0	0	0	0	0	0	0	1	0	X	0	X	0	X	1	X
0	0	0	1	0	0	1	0	0	X	0	X	1	X	X	1
0	0	1	0	0	0	1	1	0	X	0	X	X	0	1	X
0	0	1	1	0	1	0	0	0	X	1	X	X	1	X	1
0	1	0	0	0	1	0	1	0	X	X	0	0	X	1	X
0	1	0	1	0	1	1	0	0	X	X	0	1	X	X	1
0	1	1	0	0	1	1	1	0	X	X	0	X	0	1	X
0	1	1	1	1	0	0	0	1	X	X	1	X	1	X	1
1	0	0	0	1	0	0	1	X	0	0	X	0	X	1	X
1	0	0	1	0	0	0	0	X	1	0	X	0	X	X	1

A_3A_2 \ A_1A_0	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	x	x	x	x
10	x	x	x	x

J_{A_3}

A_3A_2 \ A_1A_0	00	01	11	10
00	x	x	x	x
01	x	x	x	x
11	x	x	x	x
10	0	1	x	x

K_{A_3}

A_3A_2 \ A_1A_0	00	01	11	10
00	0	0	1	0
01	x	x	x	x
11	x	x	x	x
10	0	0	x	x

J_{A_2}

A_3A_2 \ A_1A_0	00	01	11	10
00	x	x	x	x
01	0	0	1	0
11	x	x	x	x
10	x	x	x	x

K_{A_2}

A_3A_2 \ A_1A_0	00	01	11	10
00	0	1	x	x
01	0	1	x	x
11	x	x	x	x
10	0	0	x	x

J_{A_1}

A_3A_2 \ A_1A_0	00	01	11	10
00	x	x	1	0
01	x	x	1	0
11	x	x	x	x
10	x	x	x	x

K_{A_1}

A_3A_2 \ A_1A_0	00	01	11	10
00	1	x	x	1
01	1	x	x	1
11	x	x	x	x
10	1	x	x	x

J_{A_0}

A_3A_2 \ A_1A_0	00	01	11	10
00	x	1	1	x
01	x	1	1	x
11	x	x	x	x
10	x	1	x	x

K_{A_0}

$J_{A3} = A_2A_1A_0$
$K_{A_3} = A_0$

$J_{A_2} = A_1A_0$
$K_{A_2} = A_1A_0$

$J_{A_1} = A_3'A_0$
$K_{A_1} = A_0$

$J_{A_0} = 1$
$K_{A_0} = 1$

D-FF

present				next				F-F input			
A_3	A_2	A_1	A_0	A_3	A_2	A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	1	0	0	1	1
0	0	1	1	0	1	0	0	0	1	0	0
0	1	0	0	0	1	0	1	0	1	0	1
0	1	0	1	0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	1	0	1	1	1
0	1	1	1	1	0	0	0	1	0	0	0
1	0	0	0	1	0	0	1	1	0	0	1
1	0	0	1	0	0	0	0	0	0	0	0

A_3A_2 \ A_1A_0	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	x	x	x	x
10	1	0	x	x

D_{A3}

$DA_3 = A_2A_1A_0 + A_3A_0'$

A_3A_2 \ A_1A_0	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	x	x	x	x
10	0	0	x	x

D_{A2}

$DA_2 = A_2A_1' + A_2A_0' + A_2'A_1A_0$

A_3A_2 \ A_1A_0	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	x	x	x	x
10	0	0	x	x

D_{A1}

$D_{A1} = A_3'A_1'A_0 + A_1A_0'$

A_3A_2 \ A_1A_0	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	x	x	x	x
10	1	0	x	x

$D_{A0} = A_0'$

because J-k F-F use less gates, so it takes up less space, has a clearer logic and is easier to maintain