

SYSC2310 Lab 5 Report

1.0 EXERCISE 1: DESIGN A BINARY COUNTER

1.1 Description of Circuit's Operation

The circuit designed in Exercise 1 has 1 input, the clock, and 3 outputs, which are the signals representing a 3-bit binary number, as shown in Figure 1. The circuit cycles from 0 to 6 and back to 0. An inner view of the circuit is shown in Figure 2. The circuit includes 3 D-flip flops, because a total of 7 states are required to represent each of the numbers in the cycle.

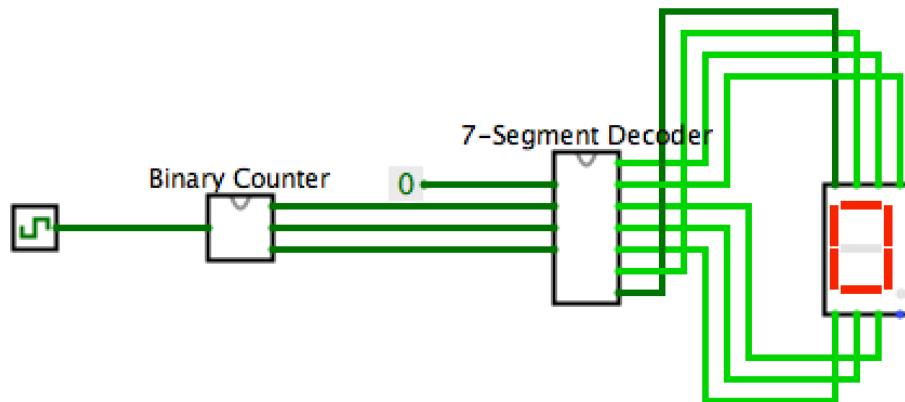


Figure 1. External view of binary counter circuit

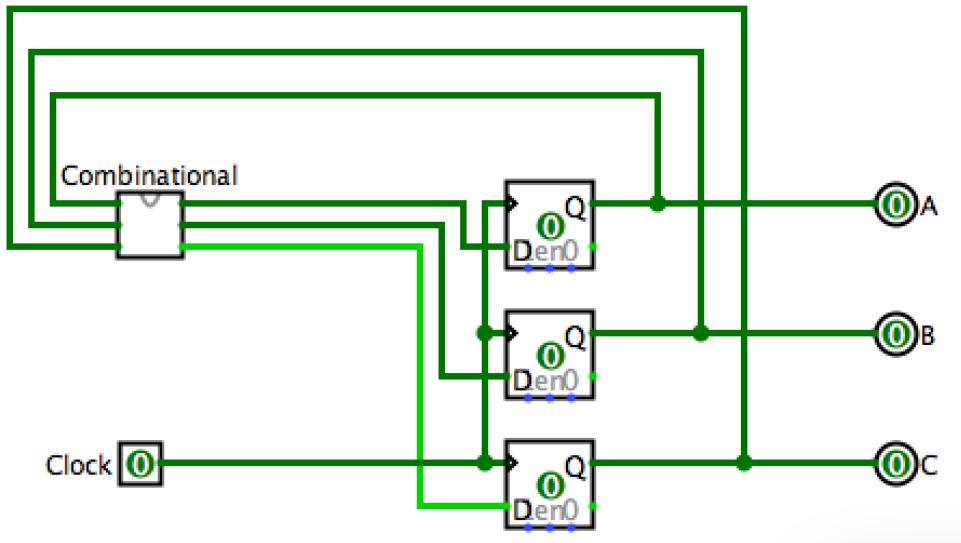


Figure 2. Internal view of binary counter circuit

The combinational circuit controls the shift to the next state. The combinational circuit is shown in Figure 3 and its subcircuits for each output are shown in Figure 4, 5, and 6.

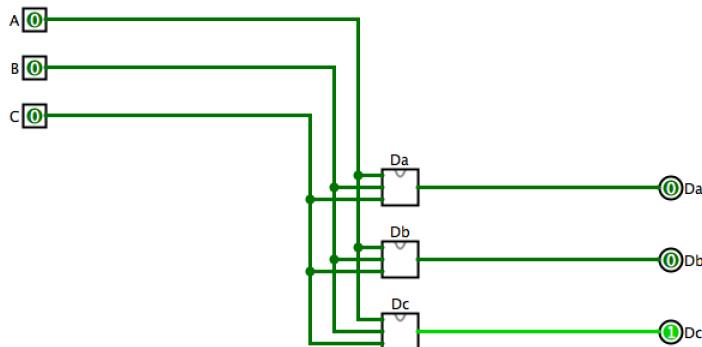


Figure 3. Main combinational circuit diagram.

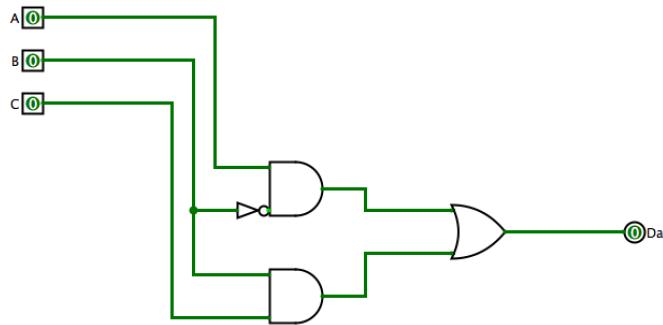


Figure 4. Circuit diagram for flip-flop A input

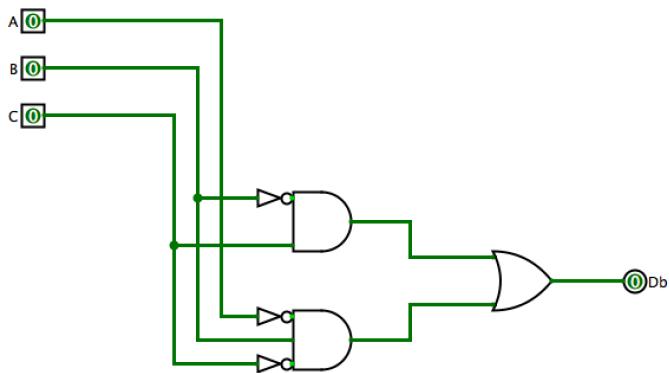


Figure 5. Circuit diagram for flip flop B input

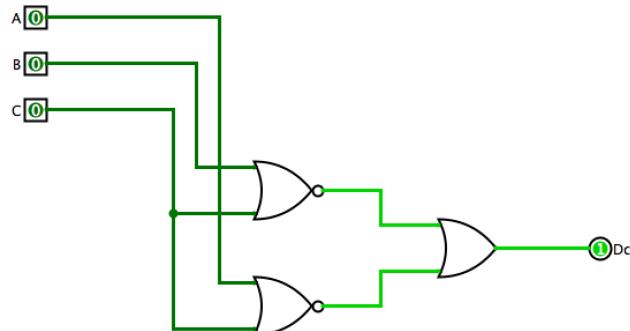


Figure 6. Circuit diagram for flip flop C input

1.2 Verification of Circuit Operation

The circuit's operation was verified using a 7-segment display, as shown in Figure 1.

The operation of the counter during the invalid state of 7 was verified by manually presetting the value of the flip-flops to 111, shown in Figure 2.

1.3 Comments

There are two types of counters: ripple and synchronous counters. The binary counter circuit made in this exercise is classified under the synchronous counter group because all of the flip flops have a common clock. The circuit could also have been designed as a ripple counter, in which all of the flip flops are connected in cascade.

2.0 EXERCISE 2: DESIGN A BINARY SEQUENCE DETECTOR

2.1 Description of Circuit's Operation

The circuit in Exercise 2 is a binary sequence detector that detects the binary sequence "1101", shown in Figure 7. The circuit has input x , which is where the stream of bits comes from, and outputs at y , which signals whether the sequence "1101" has been detected. The circuit is comprised of a next state combinational circuit, a state register, and an output combinational circuit, resembling a Moore finite state machine.

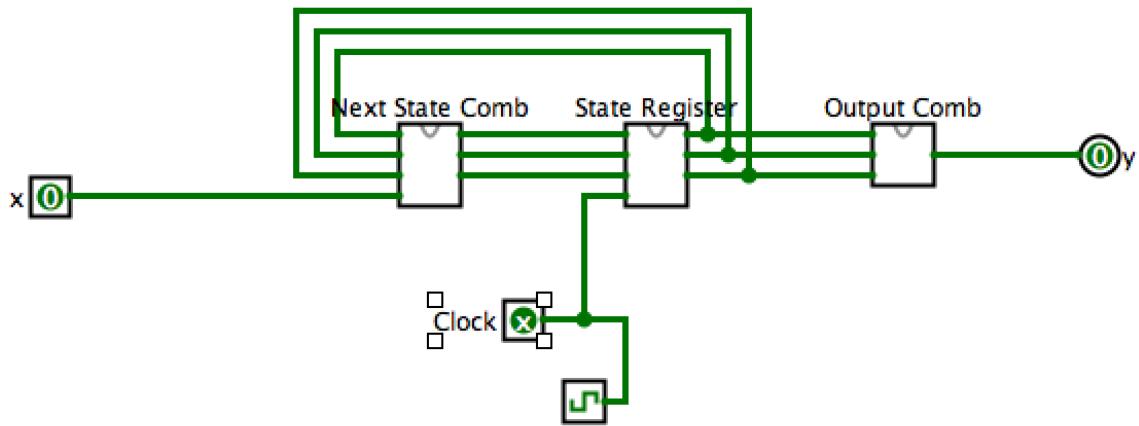


Figure 7. Circuit diagram of binary sequence detector

The next state combinational circuit controls the input to the state register, shown in Figure 8, with its subcircuits for each individual input shown in Figure 9, 10, and 11. The state register is comprised of 3 flip flops, shown in Figure 12. The output combinational circuit, which outputs high when "1101" is detected, is shown in Figure 13.

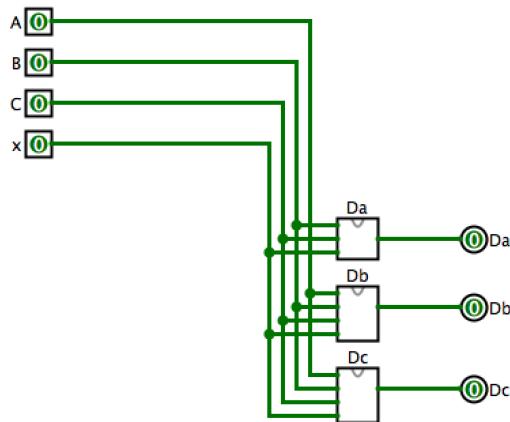


Figure 8. Next state combinational circuit diagram

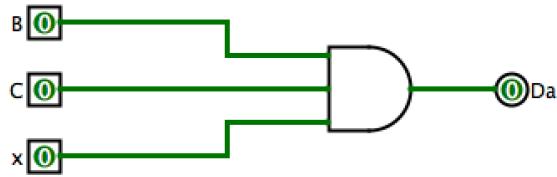


Figure 9. Circuit diagram for input at flip flop A

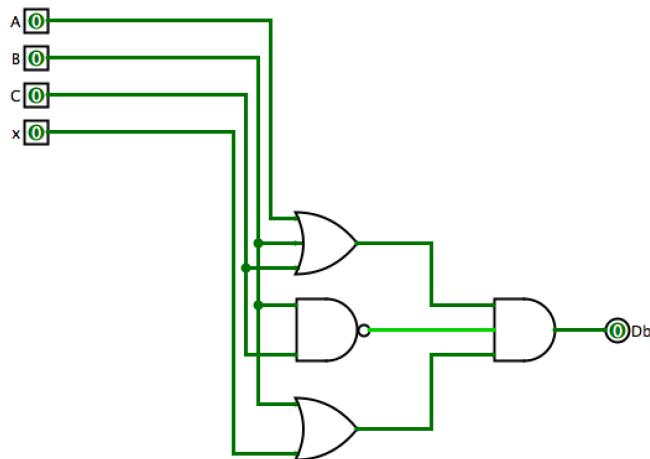


Figure 10. Circuit diagram for input at flip flop B

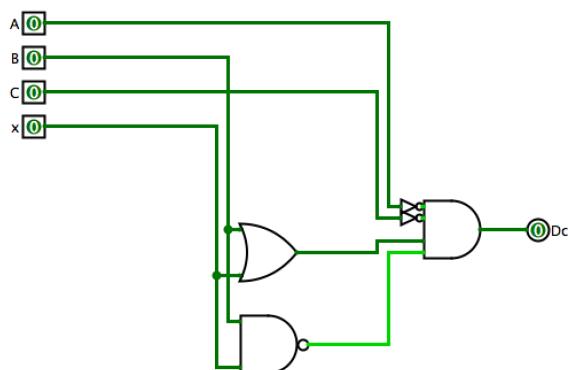


Figure 11. Circuit diagram for input at flip flop C

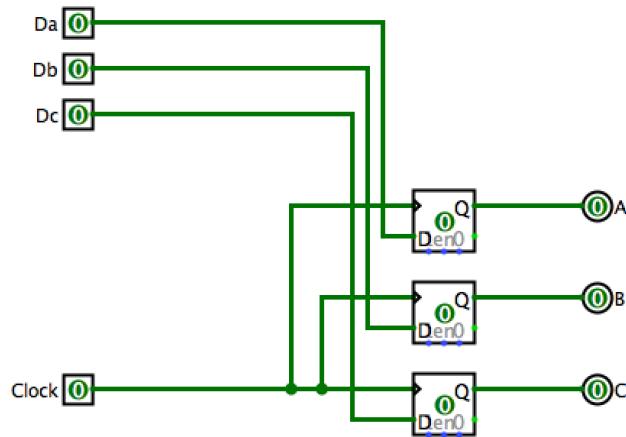


Figure 12. Circuit diagram for state register

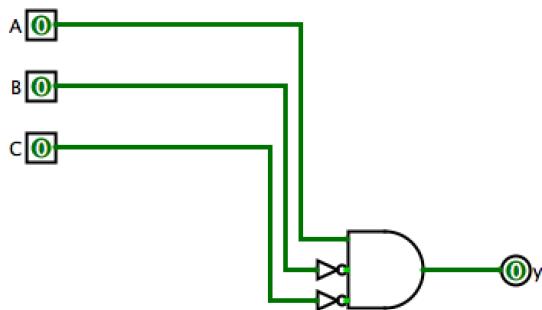


Figure 13. Circuit diagram for output combinational circuit

2.2 Verification of Circuit Operation

The verification of the circuit's operation was verified using Logisim's logging function, shown in Figure 14, by checking the values of x, the current states of the flip flops, the next state, and the output.

x	Da	Db	Dc	A	B	C	y
0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0
1	0	1	0	0	0	1	0
1	0	1	0	0	1	0	0
0	0	1	1	0	1	0	0
0	0	0	0	0	1	1	0
1	1	0	0	0	1	1	0
1	0	1	0	1	0	0	1
1	0	1	0	0	1	0	0
0	0	1	1	0	1	0	0
0	0	0	0	0	1	1	0
1	1	0	0	0	1	1	0

Close Window

Figure. 14. Logging file used to verify circuit in Exercise 2

2.3 Comments

It is possible to create this sequence detector using a shift register. It is also possible to make the shift register detect any 4-bit sequence. Figure 15 shows an example of a circuit that detects any sequence according to the value inputted in target. A stream of bits is inputted into the shift register comprised of D flip-flops and the state of the flip flops is directly compared to the 4-bit target.

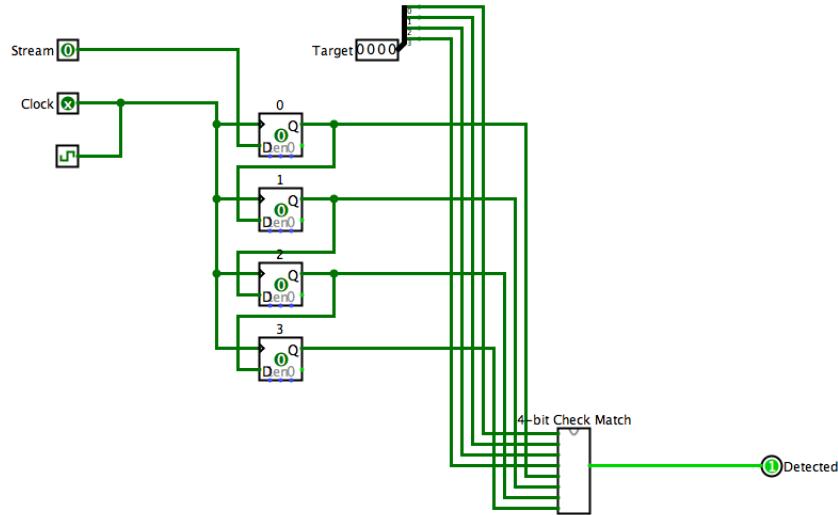


Figure 15. Circuit diagram for sequence detector using shift register

3.0 EXERCISE 3: DESIGN USING A JK FLIP-FLOP

3.1 Description of Circuit's Operation

The circuit works exactly like the one in Exercise 2, however the only difference is that JK flip flops are used instead of D flip flops. It has the same format, with an input combinational circuit, a state register, and an output combinational, as well as input x and output y, shown in Figure 16, however there are more bits going from the input combinational and the state register because the JK flip flops have both J and K inputs, instead of just a single D input.

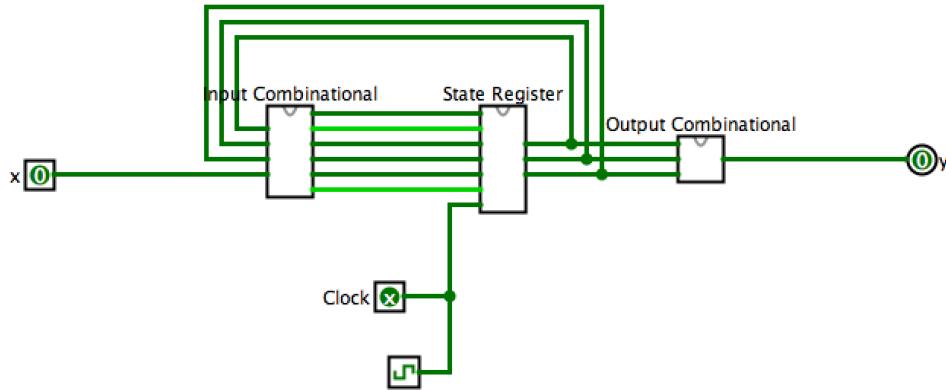


Figure 16. Sequence detector circuit diagram using JK flip flops

The input combinational circuit is shown in Figure 17. The state register is shown in Figure 18. The output combinational circuit is the same as the one for Exercise 2, shown in Figure 13.

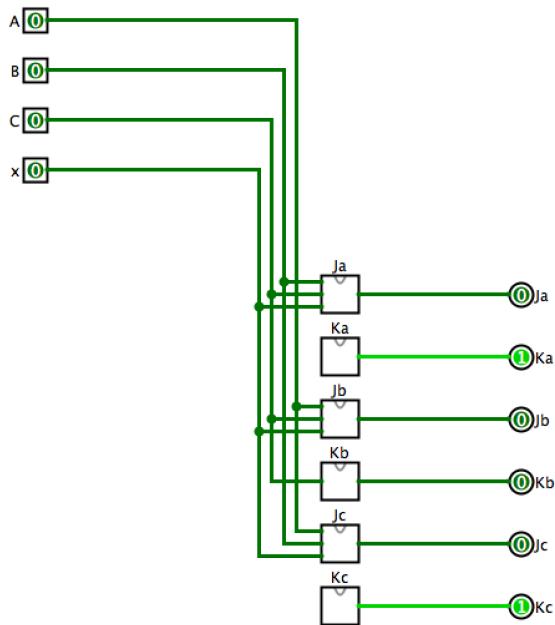


Figure 17. Combinational input circuit for Exercise 3 circuit

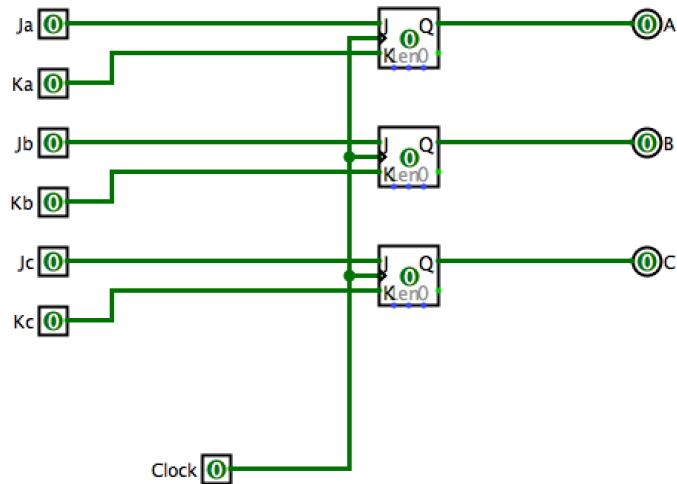


Figure 18. State register circuit diagram

3.2 Verification of Circuit's Operation

The circuit's operation was verified using Logisim's logging function, verifying the current and next states, and the input and output, shown in Figure 19.

Logisim: Log Sequence Detector of Lab5_Ex3

x	A	B	C	Ja	Ka	Jb	Kb	Jc	Kc	y	
0	0	0	0	0	1	0	0	0	0	1	0
1	0	0	0	0	1	0	0	1	1	0	
1	0	0	1	0	1	1	1	1	1	0	
0	0	0	1	0	1	0	1	0	1	0	
1	0	0	1	0	1	1	1	1	1	0	
1	0	1	0	0	1	0	0	0	1	0	
0	0	1	0	0	1	0	0	1	1	0	
1	0	1	0	0	1	0	0	0	1	0	
0	0	1	0	0	1	0	0	1	1	0	
1	0	1	0	0	1	0	0	0	1	0	
0	0	1	0	0	1	0	0	1	1	0	
0	0	1	1	0	1	0	1	1	1	0	
1	0	1	1	1	1	1	1	0	1	0	
0	0	1	1	0	1	0	1	1	1	0	
0	0	0	0	1	0	0	0	0	1	0	

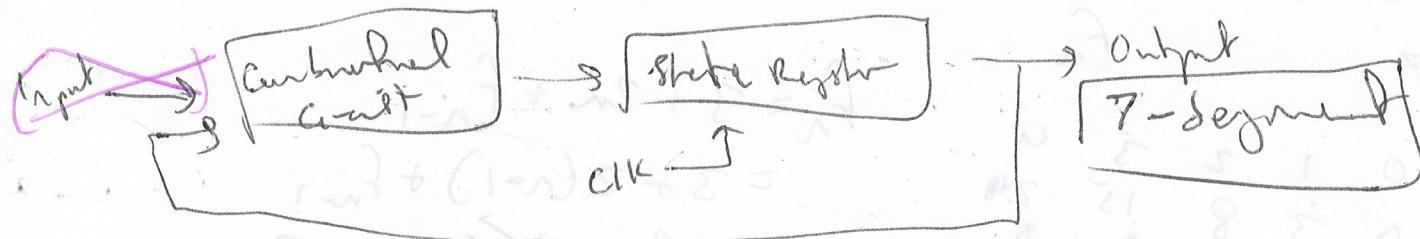
Close Window

Figure 19. Logisim logging file used to verify circuit operation

SYSL2310 - LAB 5: SEQUENTIAL LOGIC CIRCUITS

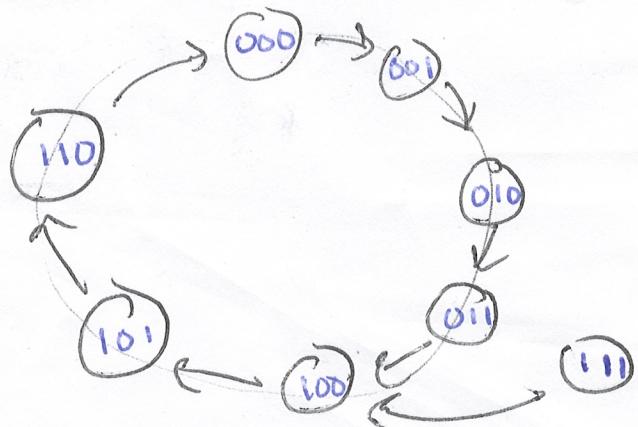
Esercizio 1: scrivere un programma

Name: Nani Zakariah
I number:
ST#: (01143497)



State table:

State Diagram:



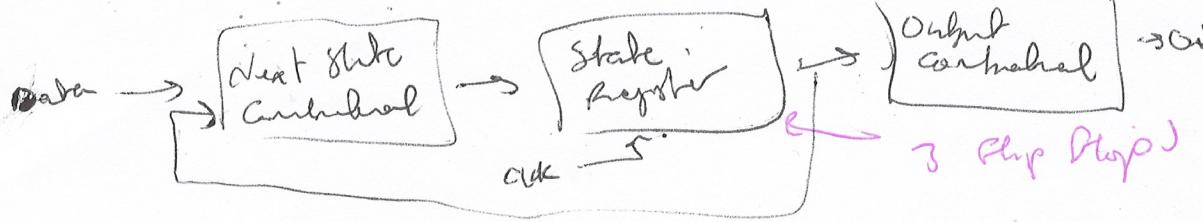
$$D_C = B^T C^T + A^T C^T$$

$$D_B = B^T C + A^T B C^T$$

Output C:

A	BC	00	01	11	10
0	-	-	-	-	C
1	i	-	-	X	-

Exercise 2: Design a Binary Sequence Detector



States:

State 000: done
 001: 1 detected
 010: 11 detected
 011: 110 detected
 100: 1101 detected

selected 1101

Present State	Input	next state			Output		
A	B	C	X	A'	B'	C'	y
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0
0	0	1	1	0	1	0	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	0	0
0	1	1	0	0	0	0	0
0	1	1	1	1	0	0	1
1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0
1	0	1	0	X	X	X	0
1	0	1	1	X	X	X	0
1	1	0	0	X	X	X	0
1	1	0	1	X	X	X	0
1	1	1	0	X	X	X	0
1	1	1	1	X	X	X	0

Don't care
or
specify ???

Output A': pos

AB	00	01	11	10
Cx	00	00	00	00
AB	00	01	11	10
00	00	00	00	00
01	00	01	10	10
11	X	X	X	X
10	00	00	X	X

$$\begin{aligned}
 D_A &= (\bar{C})(\bar{x})(B) \\
 &= B \bar{C} \bar{x}
 \end{aligned}$$

Output B': pos

AB	00	01	11	10
Cx	00	01	11	10
AB	00	01	11	10
00	00	00	00	00
01	11	11	00	00
11	X	X	X	X
10	00	00	1X	X

$$D_B = (A + B + C)(B' + C')(x + B)$$

Exercise 2: Design a binary sequence detector (CONT)

Output C': P_{0,5}

A'B'C'	00 01 11 10
00 0 1 0 0	
01 1 0 0 0	
11 X X X X	
10 0 0 X X	

$$D_C = (A')(C')(B + X)(B' + X')$$

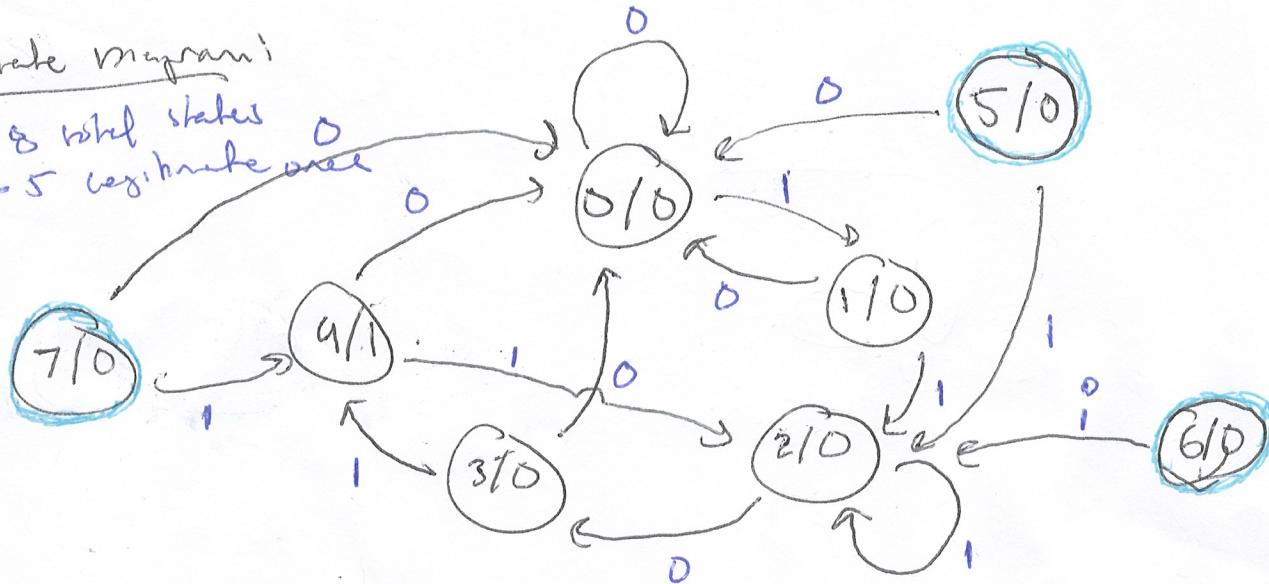
Output y: SOR

A'B'C'	00 01 11 10	A'B'C'	00 01 11 10
00 0 1 0 0		0 1 1 1	
01 1 0 0 0		1 1 1 1	
11 X X X X		X X X X	
10 0 0 X X		X X X X	

$y = A'B'C$

State diagram:

- 8 total states
- 5 legitimate ones



→ Note: the highlighted states should be impossible to enter and the only time they ever could is if the flip flops were manually adjusted to any of these states at startup.

Exercise 3: Design using a JK Flip Flop

JK Flip Flop Excitation Table:

$Q(t)$	$Q(t+1)$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Output

Present State A B C	Input X	next state A' B' C'	Flip Flop Inputs				Y
			J_A	K_A	J_B	K_B	
0 0 0	0	0 0 0	0	X	0	X	0 X
0 0 0	1	0 0 1	0	X	0	X	1 X
0 0 1	0	0 0 0	0	X	0	X	0
0 0 1	1	0 1 0	0	X	1	X	1 X
0 1 0	0	0 1 1	0	X	X	0	1 X
0 1 0	1	0 1 0	0	X	X	0	0 X
0 1 1	0	0 0 0	0	X	X	1	X 1
0 1 1	1	1 0 0	1	X	X	1	X 1
1 0 0	0	0 0 0	X	1	0	X	0 X
1 0 0	1	0 1 0	X	1	1	X	0 X
1 0 1	0	X X X	X	X	X	X	X X
1 0 1	1	X X X	X	X	X	X	X X
1 1 0	0	X X X	X	X	X	X	X X
1 1 0	1	X X X	X	X	X	X	X X
1 1 1	0	X X X	X	X	XX	XX	XX
1 1 1	1	X X X	X	X	XX	XX	XX

Output J_A :

Cx	00	01	11	10
AB	00	-	-	-
AB	01	-	-	-
AB	11	X	X	X X
AB	10	X	X X	X

$$J_A = BCx$$

Output K_A :

Cx	00	01	11	10
AB	00	X	X	X
AB	01	X	X	X
AB	11	X	X	X
AB	10	L	L	X

$$K_A = 1$$

Output J_B :

Cx	00	01	11	10	
AB	00	0	0	1	0
AB	01	X	X	X	X
AB	11	X	X	X	X
AB	10	0	1	X	X

$$J_B = Ax + Cx$$

Exercise 3 (contd) :

Output K_B:

AB\X	00	01	11	10
00	X	X	X	X
01	0	0	1	1
11	X	X	X	X
10	X	X	X	X

$$K_B = C$$

Output J_C:

AB\X	00	01	11	10
00	0	0	1	1
01	0	0	X	X
11	X	X	X	X
10	0	0	X	X

$$J_C = (B + X)(B' + X')(A')$$

Output K_C:

AB\X	00	01	11	10
00	X	X	1	1
01	X	X	1	1
11	X	X	X	X
10	X	X	X	X

$$K_C = 1$$

Output Y:

AB\BC	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

$$y = A B' C'$$

COUNTING THE GATES:

D-Rippl Flops:

- 9 Gates used

JK Clip Flips:

- 9 Gates Used

Same number of gates used for both.