

B. Eng. C.S.E. Digital Circuits Combinational and Sequential Systems

If an output is to be generated strictly on the basis of the present input condition, then we have what is commonly referred to as a combinational system. On the other hand, if the output is dependent on a sequence of input conditions, then we have what is called a sequential system. The majority of systems contain a mixture of both of these.

The Sequential or Finite State Machine (FSM)

There are two types of FSM

- (i) asynchronous
- (ii) synchronous

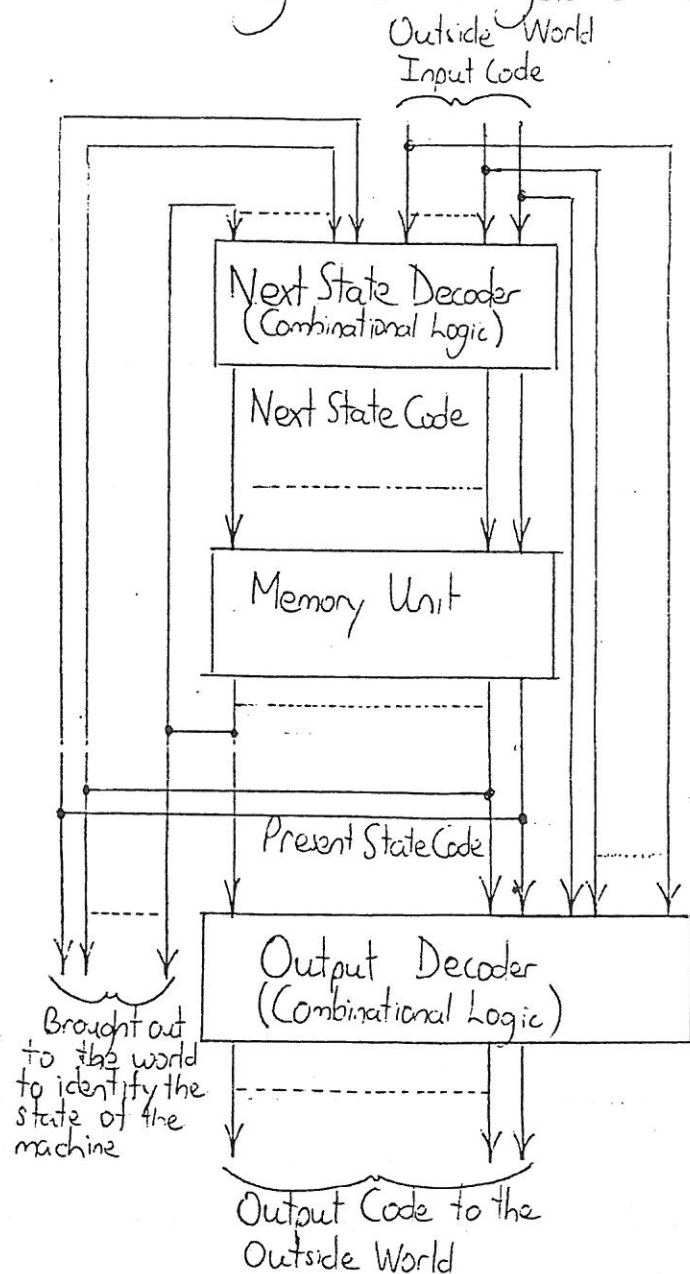
The uses of asynchronous circuits are more limited than synchronous circuits. These two types are formally defined as follows:

Asynchronous sequential circuits are basically combinational logic with direct feedback and are cycled by transitions of each input and strictly use the propagation delay of the next state decoder for memory.

Synchronous circuits use hardware devices called flip/flops for memory and are cycled by a special single input waveform commonly called the SYSTEM CLOCK.

Fundamentals of Sequential Machine Operation.

The block diagram is the general model for a FSM



Sequential machines were classified by Huffman and Moore when they introduced the STATE of a machine as a method of characterizing the memory of these circuits. The output code of the memory at an appropriately chosen time was called the STATE of the machine. This 'appropriate time' is when the machine is not in a transient condition i.e. changing states. Sequential circuit design is performed using a STATE DIAGRAM.

The function of the NEXT STATE DECODER is to decode the inputs from the outside world and the PRESENT STATE of the machine (stored by the memory) and to generate as its output a code called the NEXT STATE CODE. This next state code will become the PRESENT STATE CODE when the memory loads and stores it. This process is defined as a CHANGE OF STATE.

The combinational logic called the output decoder has the function of decoding the present state of the machine and the PRESENT INPUT CONDITIONS in order to generate the desired control outputs to the outside world.

So in conclusion a FSM steps through states (memory output codes) in a sequential manner. The STATE CODES of the machine are generated by the NEXT STATE DECODER that converts the OUTSIDE WORLD INPUTS combined with the PRESENT STATE CODE into bit patterns called the NEXT STATE CODES of the machine.

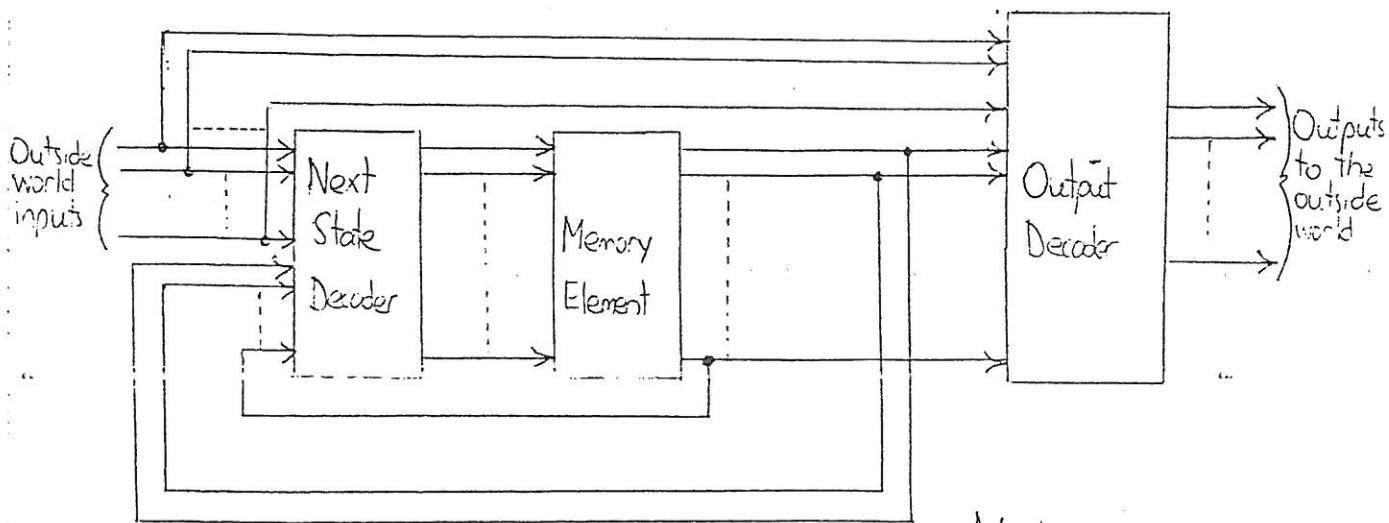
Classification of Finite State Machines

There are three classes of machine worth mentioning:

(i) Class A machine (Mealy machine)

The outputs, to the outside world, of a Mealy machine are a function of two sets of variables:

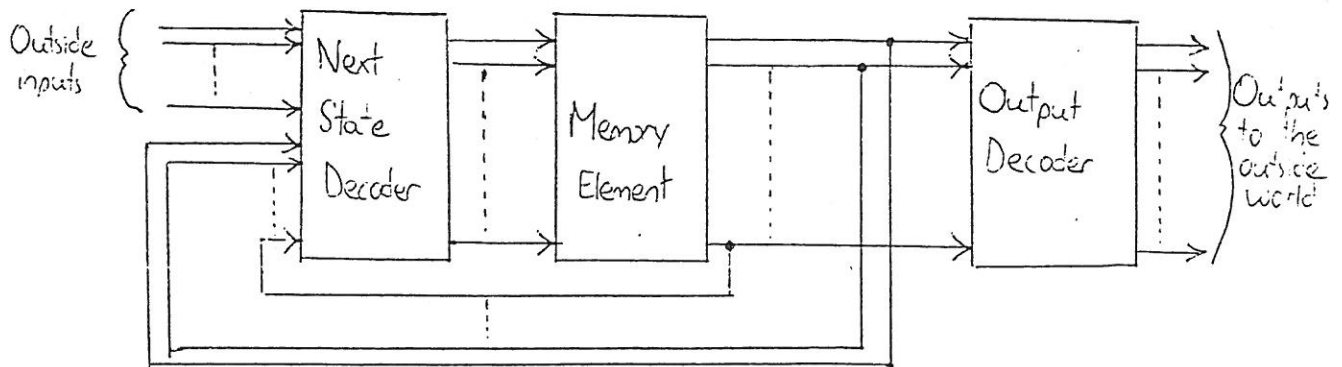
- (a) the present input conditions
- (b) the present state of the machine



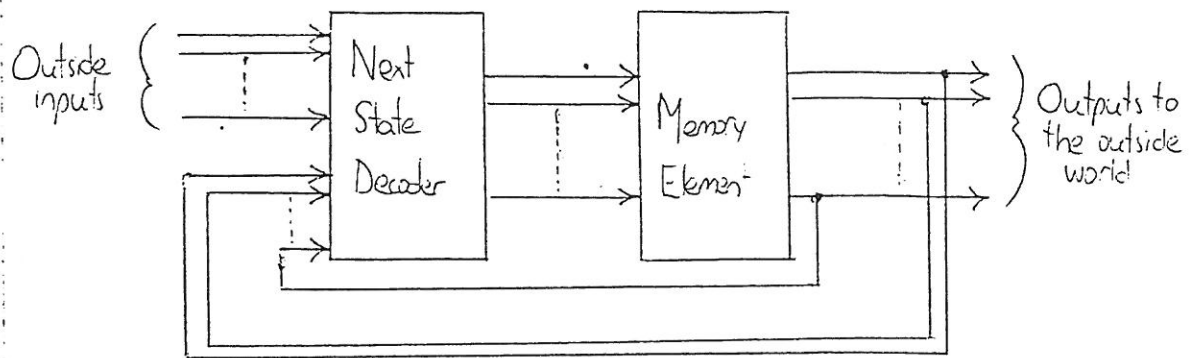
Mealy or Class A machine

(ii) Class B and C machines (Moore machine)

The basic distinction of a Moore machine is that its output is strictly a function of the state of the machine.



Class B machine, or Moore machine



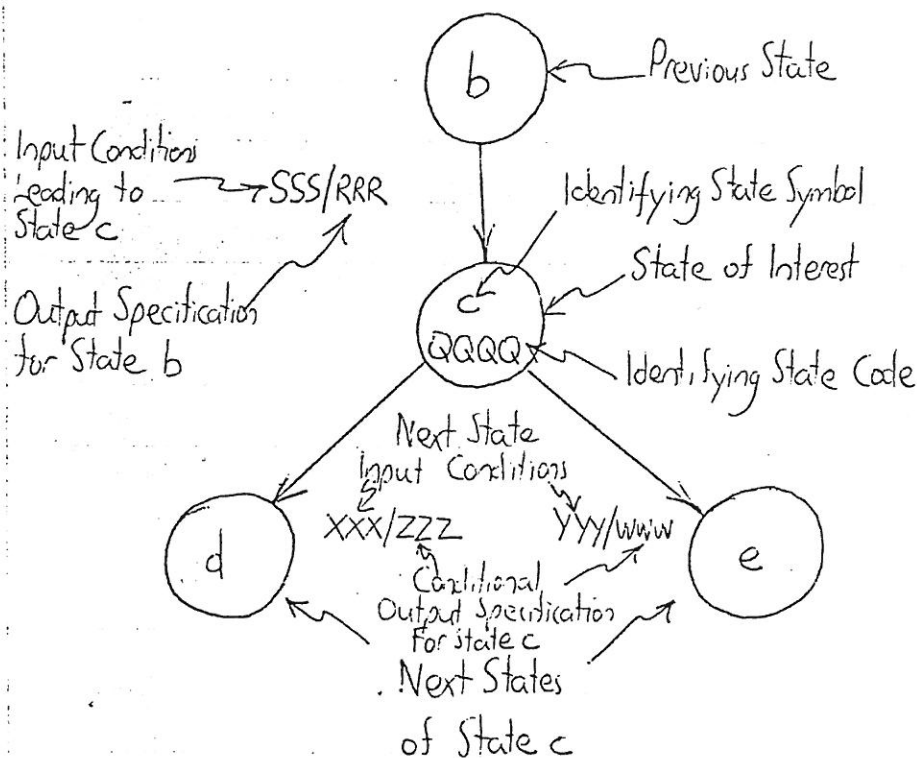
Class C machine, or Moore machine without an output decoder

Both the Mealy and Moore machines are widely used and it is possible to derive machines that are mixtures of both; in other words, some outputs are conditional on both the inputs and the state of the machine, where others are dependent only on the state of the machine.

The State Diagram

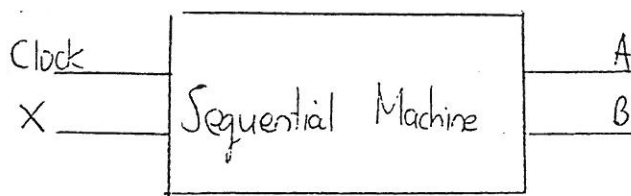
The STATE DIAGRAM is to sequential design what the TRUTH-TABLE is to combinational design. An example of a state diagram is shown below illustrating the state of interest and 5 important and required pieces of information related to each state of the machine. The diagram is a series of circles connected by straight lines with arrowheads. Each circle represents a state of the machine and the lines are graphical indications of state changes. In the diagram five important and descriptive pieces of information for each state are given and labeled as follows:

- (1) some STATE identifying symbol or code;
- (2) the previous STATE of the STATE OF INTEREST;
- (3) the INPUT conditions leading to the STATE OF INTEREST;
- (4) the OUTPUT specification for the STATE OF INTEREST;
- (5) the NEXT STATES and NEXT STATE branching conditions for the STATE OF INTEREST.

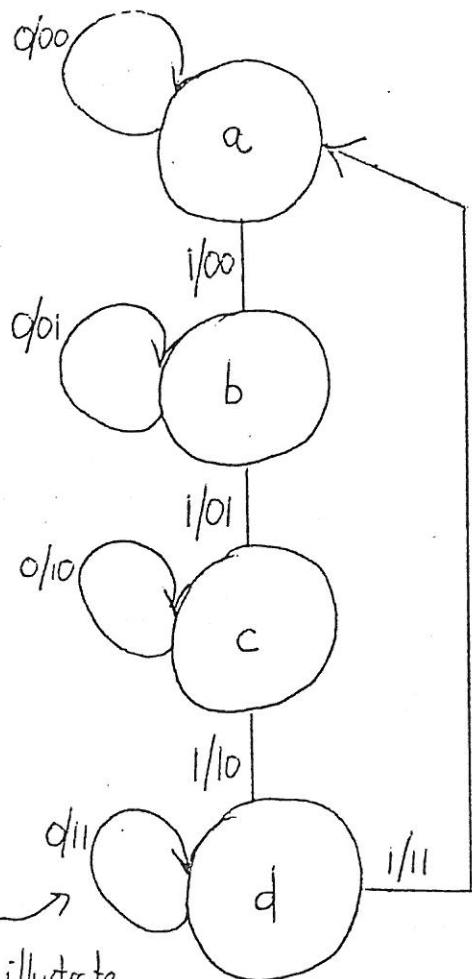


Worked Example

Consider the following specification describing the sequential operation of a machine. The machine has a single control input X and a clock, and two outputs A and B . On consecutive rising edges of the clock the code on A and B changes from 00 to 01 to 10 to 11 and then repeats itself if X is asserted; if at any time X is de-asserted this machine is supposed to hold its present state.



The state diagram for this machine is shown below:



Symbology format: X/AB

Note: the clock is usually considered implied

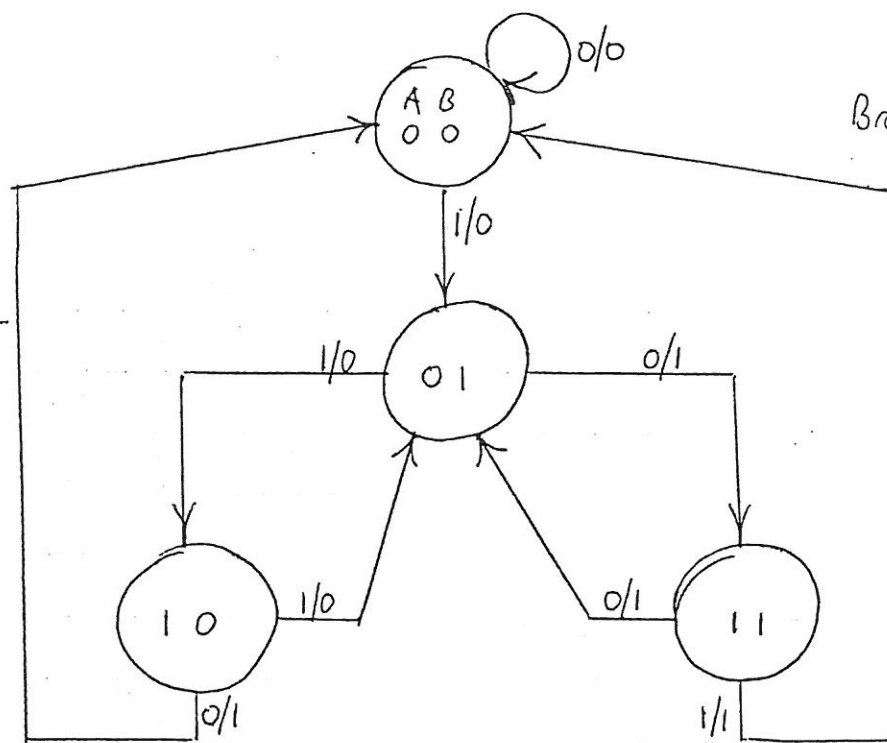
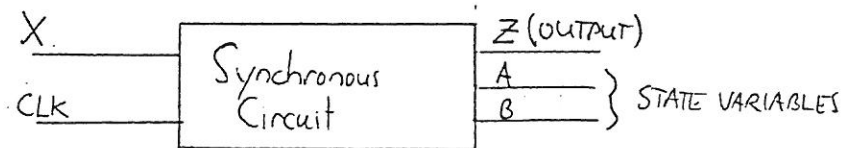
These loops illustrate a HOLD in this state

The descriptive features of the state diagram make it an invaluable aid in the following three major areas:

- (i) the definition of sequential systems;
- (ii) the analysis of sequential circuits;
- (iii) the design of sequential circuits.

(i) Definition of sequential systems

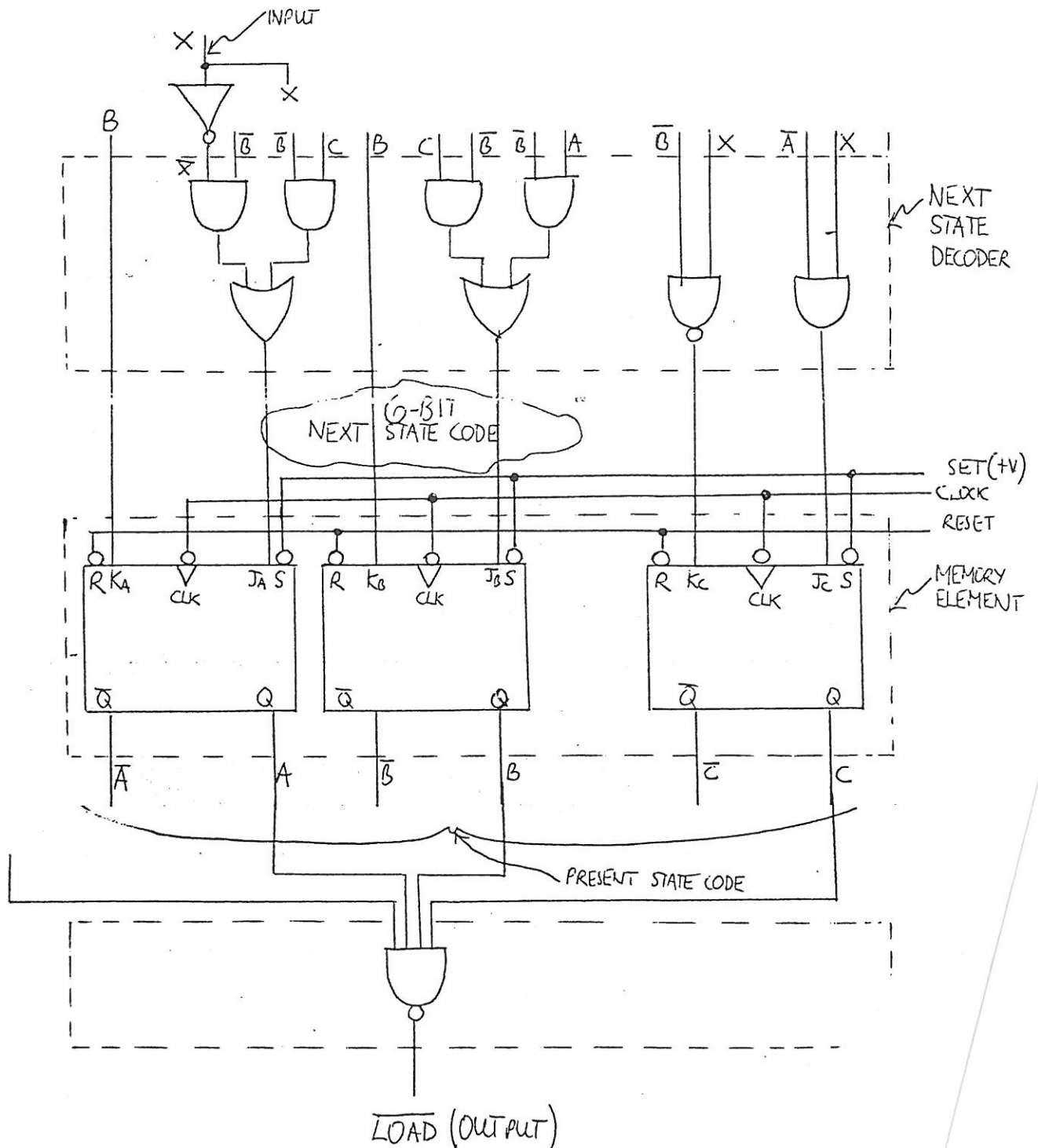
Example



Branch Format: X/Z

(ii) Analysis of Synchronous Sequential Circuits

Step 1: Identify the functional blocks of the sequential machine, that is, NEXT STATE DECODER, MEMORY ELEMENT and OUTPUT DECODER (Note: the circuit below is a classic machine).



Step 2: Write the Boolean expression for each of the outputs of the NEXT STATE DECODER. This six-bit CODE makes up the inputs for the J-K Flip-Flops. Once these are known, they can be coupled with the J-K flip-flop characteristics to determine the TRUE NEXT STATE code.

$$J_A = \bar{B}.C + \bar{B}.\bar{X}$$

$$K_A = B$$

$$J_B = A.\bar{B} + \bar{B}.C$$

$$K_B = B$$

$$J_C = \bar{A}.C$$

$$K_C = \overline{X.B} = \bar{X} + \bar{B}$$

and the output:

$$\overline{LOAD} = \overline{A.B.C.X}$$

Step 3: Plot the map for each of the NEXT STATE variables. These maps serve as a useful tool for tabulating the NEXT STATE code.

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

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

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

$$J_A = \bar{B}.C + \bar{B}.\bar{X}$$

$$K_A = B$$

$$J_B = A.\bar{B} + \bar{B}.C$$

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

$$K_B = B$$

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

$$J_C = \bar{A}.C$$

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

$$K_C = B + \bar{X}$$

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

OUTPUT MAP

$$LOAD = ABCX$$

Step 4: Using the entries in these maps, plot the values for the flip flop inputs ($J_A, K_A, J_B, K_B, J_C, K_C$) in a PRESENT and NEXT STATE TABLE.

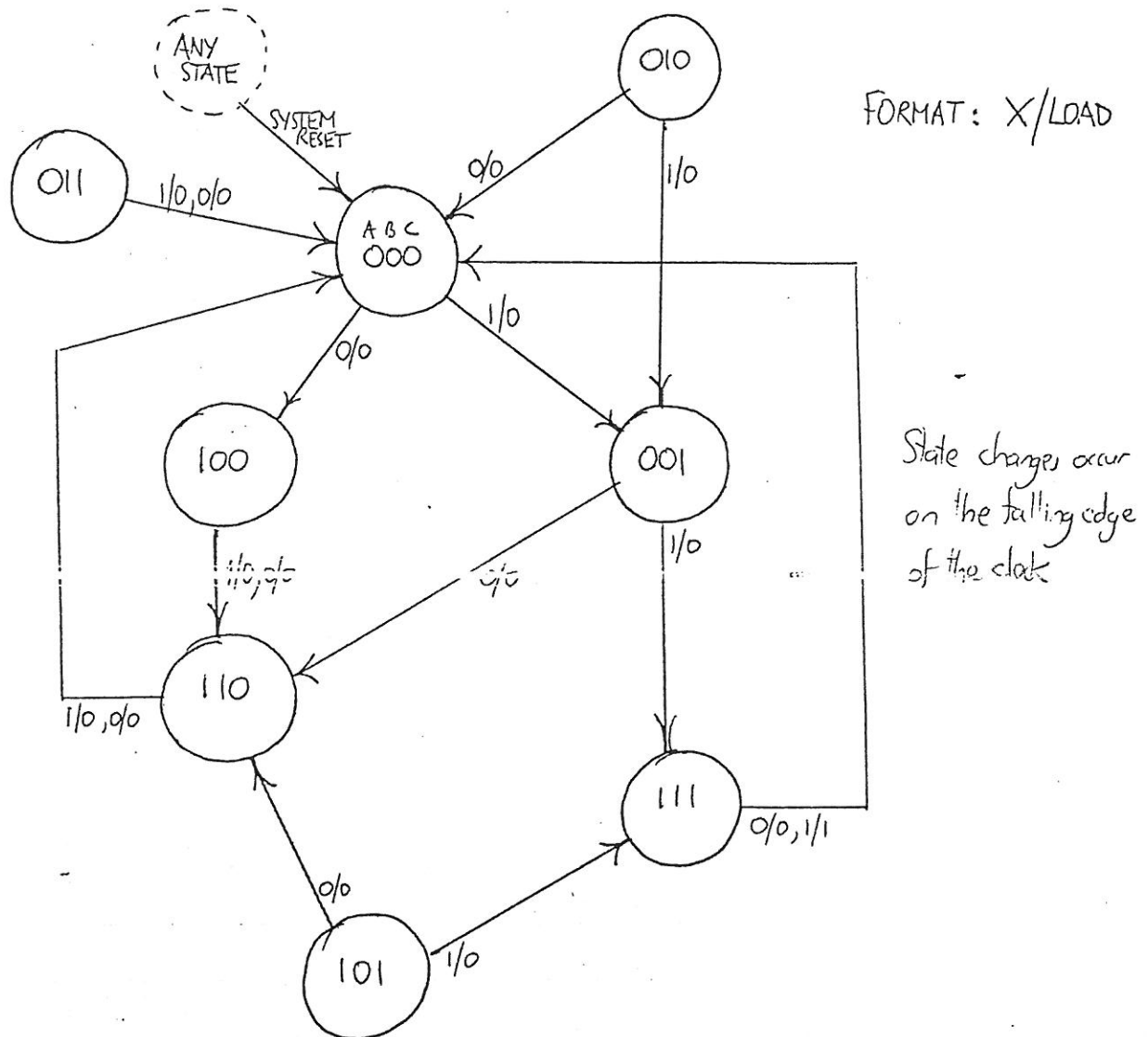
	PRESENT STATE			INPUTS	NEXT STATE CODE						OUTPUT	TRUE NEXT STATE		
	A	B	C		J_A	K_A	J_B	K_B	J_C	K_C		A	B	C
STATE 0	0	0	0	0	1	0	0	0	0	1	0	1	0	0
	0	0	0	1	0	0	0	0	1	0	0	0	0	1
STATE 1	0	0	1	0	1	0	1	0	0	1	0	1	1	0
	0	0	1	1	1	0	1	0	1	0	0	1	1	1
STATE 2	0	1	0	0	0	1	0	1	0	1	0	0	0	0
	0	1	0	1	0	1	0	1	1	1	0	0	0	1
	0	1	1	0	0	1	0	1	0	1	0	0	0	0
	0	1	1	1	0	1	0	1	1	1	0	0	0	0
	1	0	0	0	1	0	1	0	0	1	0	1	1	0
	1	0	0	1	0	0	1	0	0	0	0	1	1	0
	1	0	1	0	1	0	1	0	0	1	0	1	1	0
	1	0	1	1	1	0	1	0	0	0	0	1	1	1
	1	1	0	0	0	1	0	1	0	1	0	0	0	0
	1	1	0	1	0	1	0	1	0	1	0	0	0	0
STATE 7	1	1	1	0	0	1	0	1	0	1	0	0	0	0
	1	1	1	1	0	1	0	1	0	1	1	0	0	0

Step 5: Using the entries for the NEXT STATE CODE in the table and a Flip flop next state table for the flip flop, determine the TRUE NEXT STATE for each flip flop one row at a time (see above).

J	K	ACTION	NEXT STATE
0	0	Donothing	Unchanged
0	1	Reset	0
1	0	Set	1
1	1	Toggle	Change State (0→1, 1→0)

NEXT STATE TABLE
FOR J-K FLIPFLOP

Step 6: Using the PRESENT STATE, INPUT, and NEXT STATE data plotted in the PRESENT and NEXT STATE TABLE, develop a state diagram to graphically illustrate the circuits state to state behaviour.



States 000, 001, 100, 110 and 111 are the sequential states and states 010, 011, and 101 are conditions which can never be reached unless NOISE or a POWER CONDITION forces the circuit into these states. However, these states are not hang states (i.e. no input condition causes a transition from that state), because if the circuit happens to be jammed to one of these states by some abnormal condition such as noise or power transients, it will not stay there for more than one clock cycle. This is a good feature to

incorporate into most designs because unwanted states are undesirable and unwanted states that hang the machine must be avoided. Designs should be analyzed to ensure hang states are not possible.

For further analysis the next state tables for the D, T and RS flip flops are shown:

D flip flop:

Input D	Action	Next State
0	Reset	0
1	Set	1

T (toggle) flip flop:

Input T	Action	Next State
0	Do Nothing	Unchanged
1	Toggle	Change State

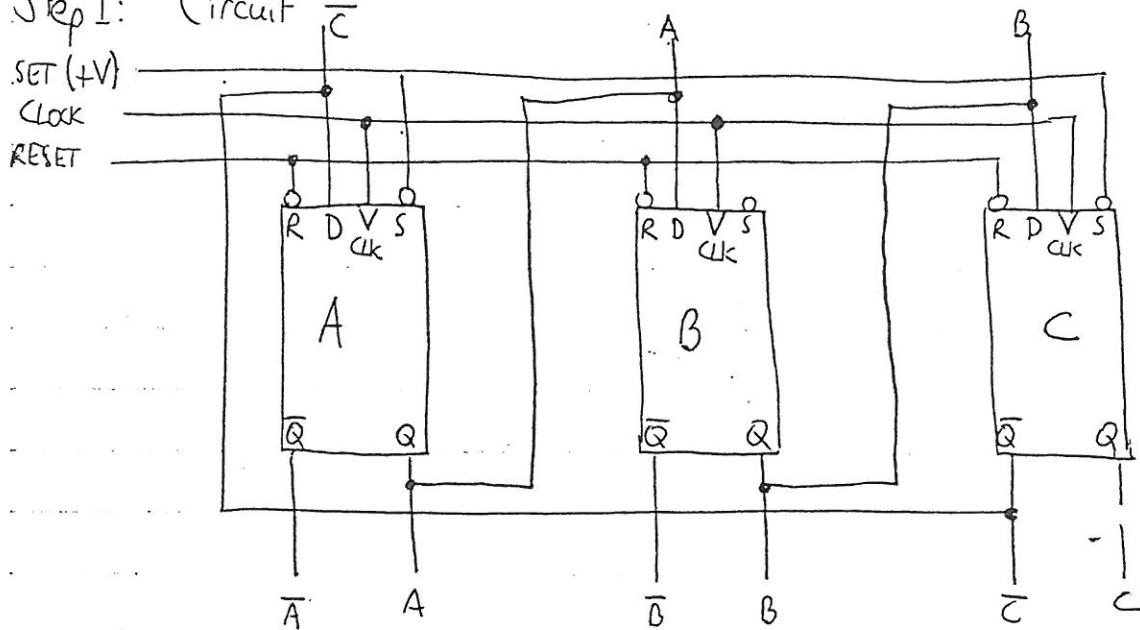
SR flip flop:

Input S R	Action	Next State
0 0	Do Nothing	Unchanged
0 1	Reset	0
1 0	Set	1
1 1	Avoid!	?

Worked Example 1:

To analyse the circuit below.

Step 1: Circuit



Step 2: FF equations

$$D_A = \bar{C}, D_B = A, D_C = B$$

Step 3: Next State Maps.

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

D_A

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

D_B

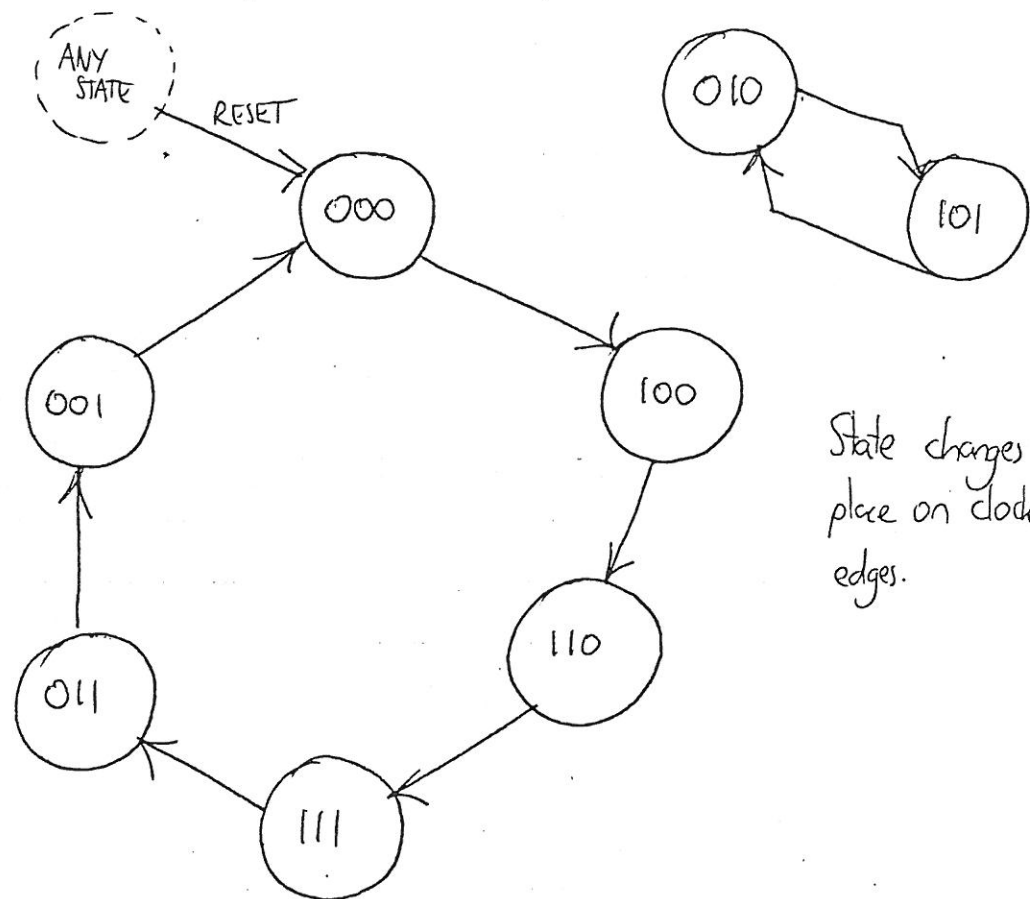
C \ AB				
	00	01	11	10
0	0	1	1	0
1	0	1	1	0

D_C

Steps 4/5: Present and Next State

Present State			Inputs ϕ	Next State Codes			Outputs ϕ	True Next State		
A	B	C		D_A	D_B	D_C		A	B	C
0	0	0	-	1	0	0	-	1	0	0
0	0	1	-	0	0	0	-	0	0	0
0	1	0	-	1	0	1	-	1	0	1
0	1	1	-	0	0	1	-	0	0	1
1	0	0	-	1	1	0	-	1	1	0
1	0	1	-	0	1	0	-	0	1	0
1	1	0	-	1	1	1	-	1	1	1
1	1	1	-	0	1	1	-	0	1	1

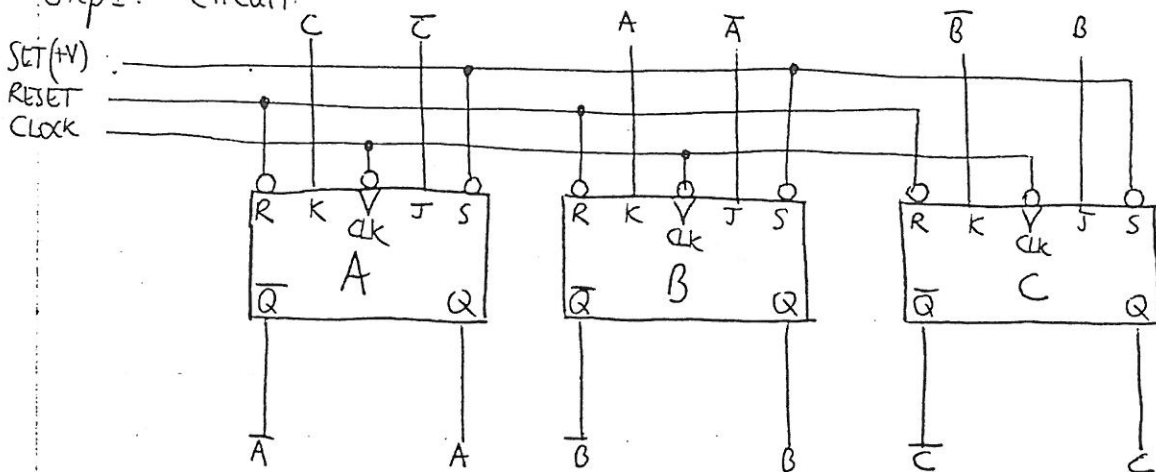
Step 6: This circuit has 2 states (010 and 101) isolated from the main sequence. This is typical for a 'twisted ring counter' which is the circuit type we have analysed.



Worked Example 2:

Analyse the twisted-type ring counter, below:

Step 1: Circuit



Step 2: Flip Flop equations.

$$J_A = \bar{C}, \quad J_B = \bar{A}, \quad J_C = B$$

$$K_A = C, \quad K_B = A, \quad K_C = \bar{B}$$

Step 3: Next State Maps.

$\bar{C}$ \ AB	00	01	11	10
0	1	1	1	1
1	0	0	0	0

J_A

$\bar{C}$ \ AB	00	01	11	10
0	0	0	0	0
1	1	1	1	1

K_A

$\bar{C}$ \ AB	00	01	11	10
0	1	1	0	0
1	1	1	0	0

J_B

$\bar{C}$ \ AB	00	01	11	10
0	0	0	1	1
1	0	0	1	1

K_B

$\bar{C}$ \ AB	00	01	11	10
0	0	1	1	0
1	0	1	1	0

J_C

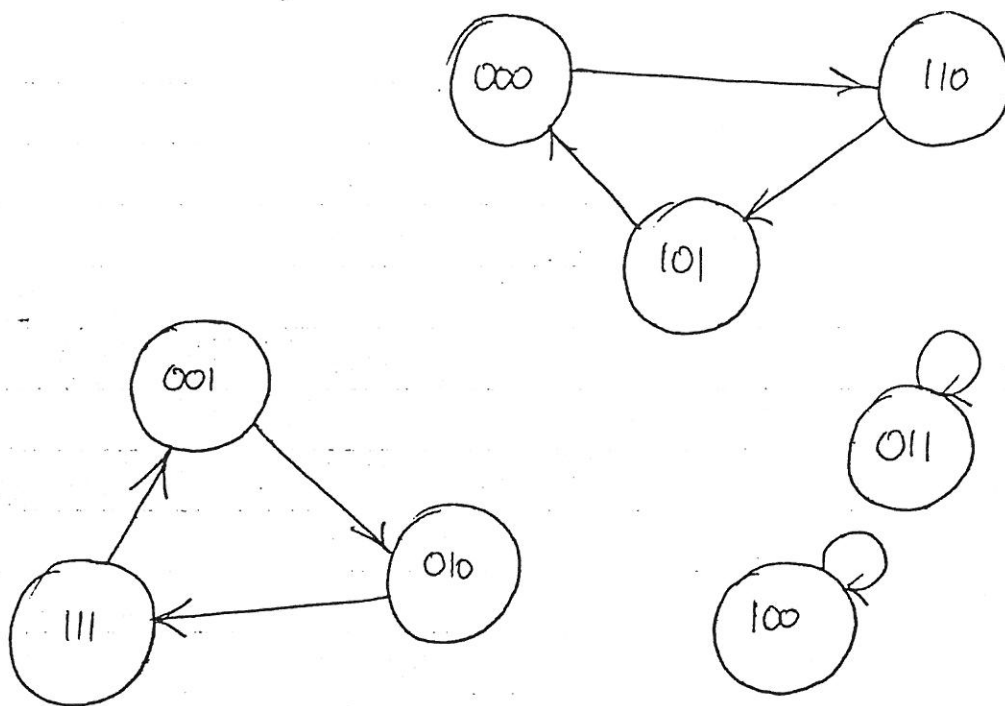
$\bar{C}$ \ AB	00	01	11	10
0	1	0	0	1
1	1	0	0	1

K_C

Step 4/5: Present and Next State Table

Present Input			Input ϕ	Next State Code						Output ϕ	True Next State		
A	B	C		J_A	K_A	J_B	K_B	J_C	K_C		A	B	C
0	0	0	-	1	0	1	0	0	1	-	1	1	0
0	0	1	-	0	1	1	0	0	1	-	0	1	0
0	1	0	-	1	0	1	0	1	0	-	1	1	1
0	1	1	-	0	1	1	0	1	0	-	0	1	1
1	0	0	-	1	0	0	1	0	1	-	1	0	0
1	0	1	-	0	1	0	1	0	1	-	0	0	0
1	1	0	-	1	0	0	1	1	0	-	1	0	1
1	1	1	-	0	1	0	1	1	0	-	0	0	1

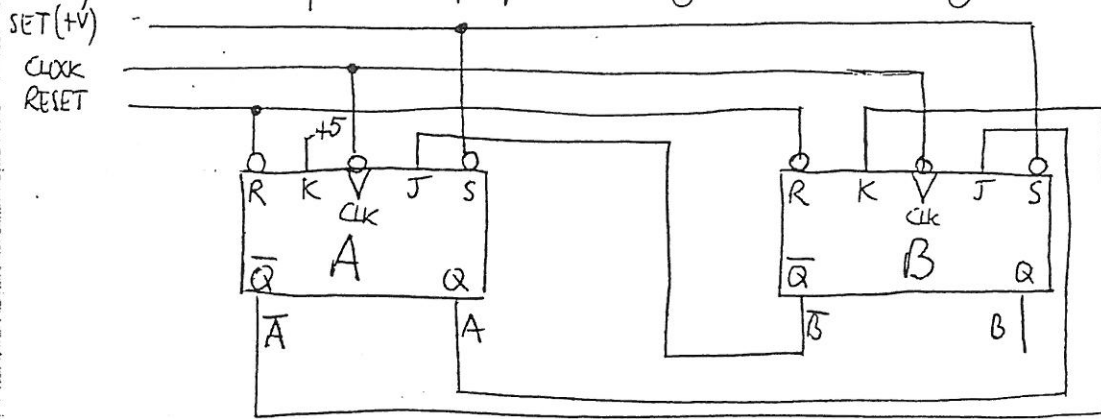
Step 6: State Diagram.



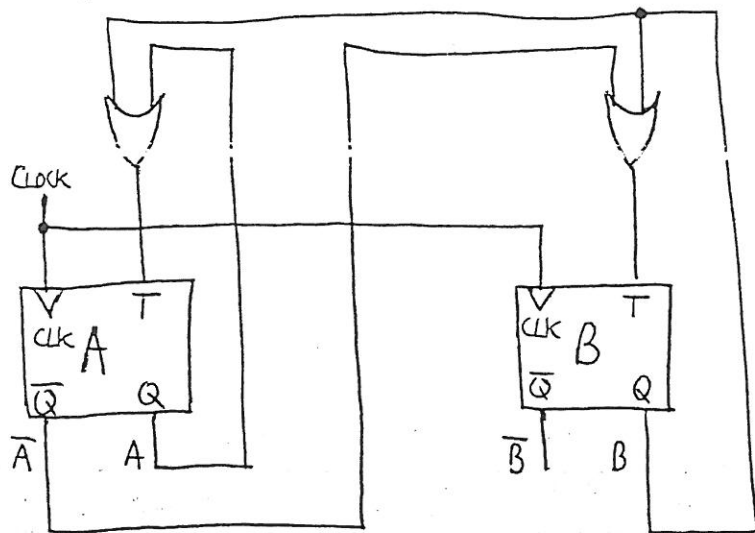
This circuit has major problems!

Problems

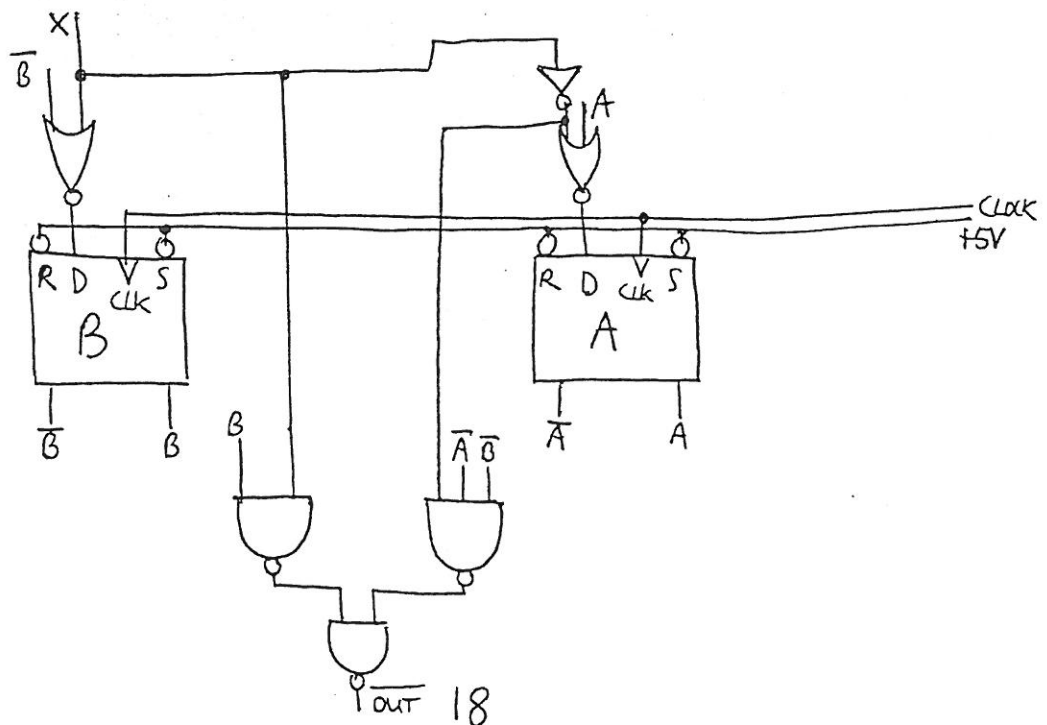
- 1) Carry out an analysis on the following circuits. Note any questionable operational properties and give the state diagrams.



2)



3)



Worked Example 2

Step 1: Receive Circuit Specification. You are to design a special circuit that will produce an output pulse if a sampled data input is tested three times and found to be at a relatively high voltage level an odd number of times, and will then return to an initial condition and start the testing process over again.

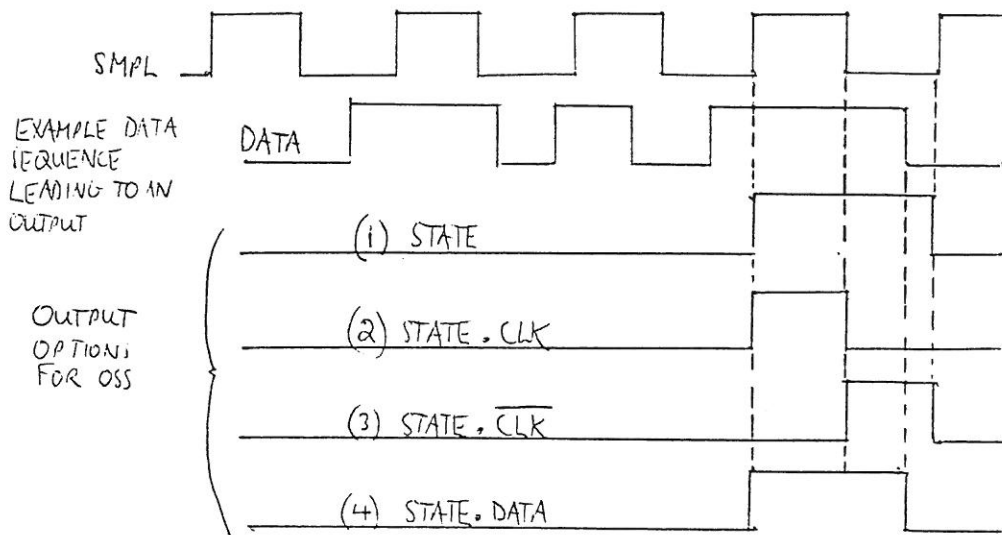
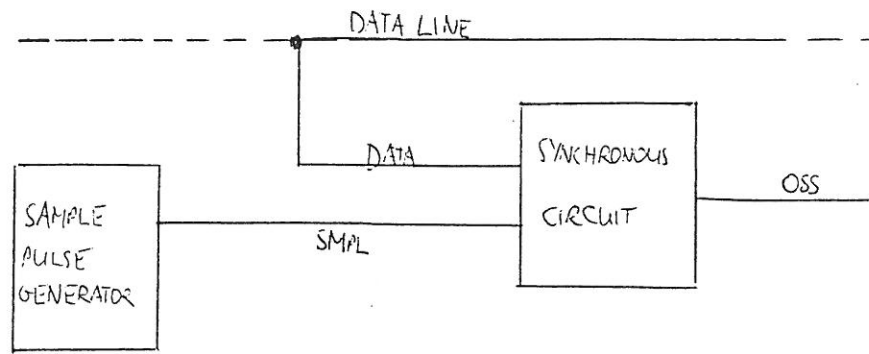
Step 2: Study of Specifications: According to these specifications our design is to be sequential and it is apparently meant to sample some data line asynchronously with respect to the information on that line. Further, it is apparently meant to sample this data three times, generate an output if necessary, and then start over again, repeating this operation indefinitely. However, this should all be done in some synchronous fashion with respect to the machine which is to be designed.

Step 3: Make a Circuit Model. A block diagram and output specification timing diagram are shown below.

Input Definitions: DATA is defined as the input from the data line and is

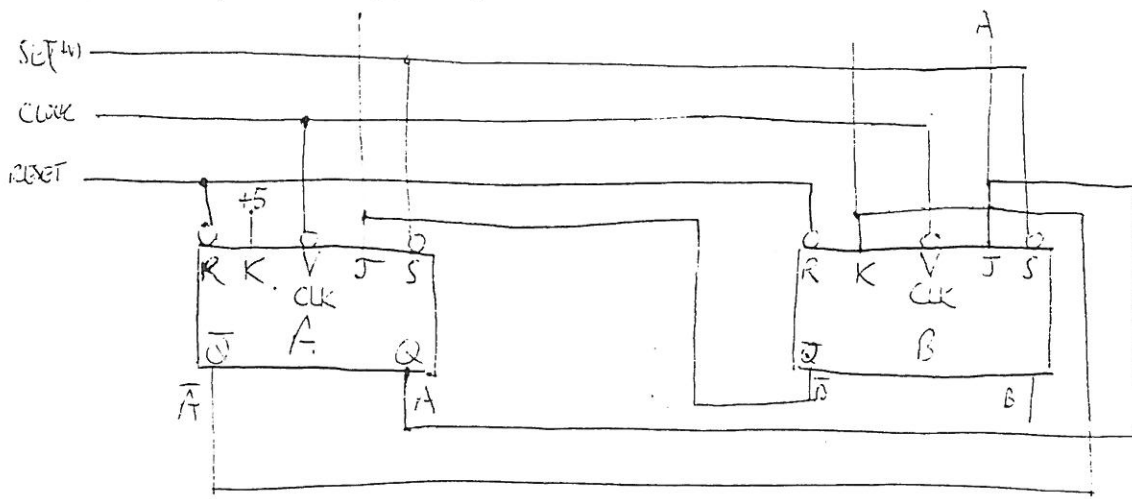
SAMPLE is defined as the sampling input which paces the sequential circuit through the sequence. It can also be thought of as the system clock, with the sampling done on the rising edge.

Output Definitions: OSS (odd sample strobe) is defined as the output from this circuit which is to be generated after three samples have been taken and an odd number of high samples have been found. The waveform shows some of the possible options for specifying the output.



The choice of output option affects the formatting of the state diagram.

Model Answer: Question 1



$$K_A = 1 \quad J_A = \bar{B} \quad K_B = \bar{A} \quad J_B = A$$

A	B	0	1
0	0	1	1
0	1	1	1
1	0	1	1
1	1	1	1

A	B	0	1
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

A	B	0	1
0	0	1	0
0	1	1	0
1	0	1	0
1	1	1	0

A	B	0	1
0	0	0	1
0	1	0	1
1	0	0	1
1	1	0	1

Present State

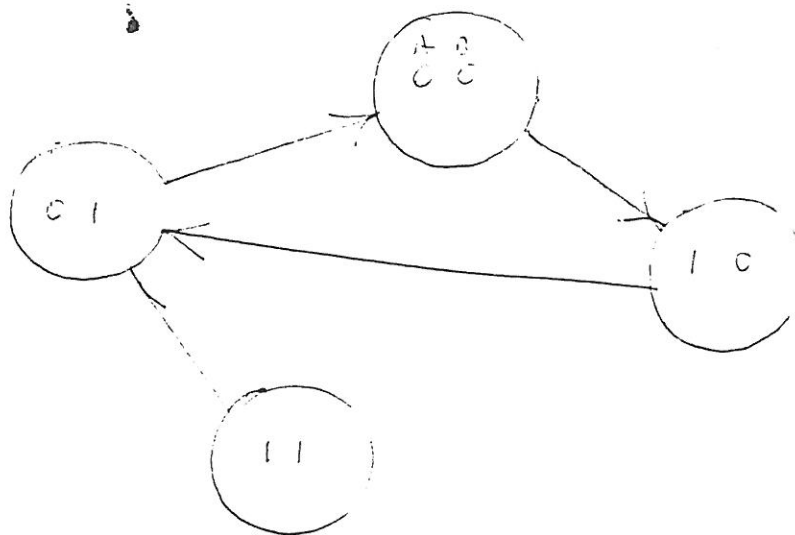
A	B
0	0
0	1
1	0
1	1

Next State Codes

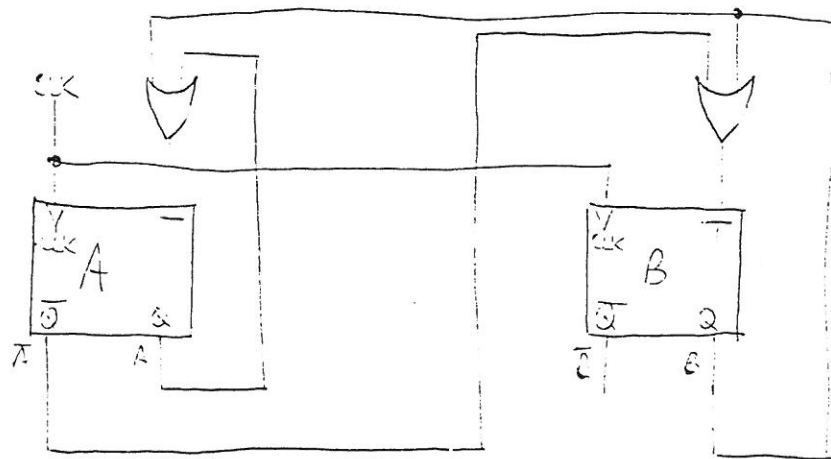
J_A	K_A	J_B	K_B
1	1	0	1
0	1	0	1
1	1	1	0
0	1	1	0

True Next State

A	B
1	0
0	0
0	1
0	1



Model Answer Problem 2.



$$T_A = A + B$$

$$T_B = \bar{A} + B$$

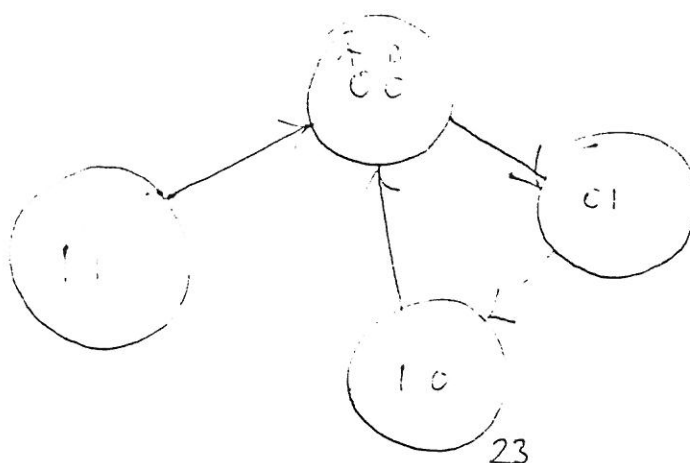
A	0	1
0	0	1
1	1	1

T_A

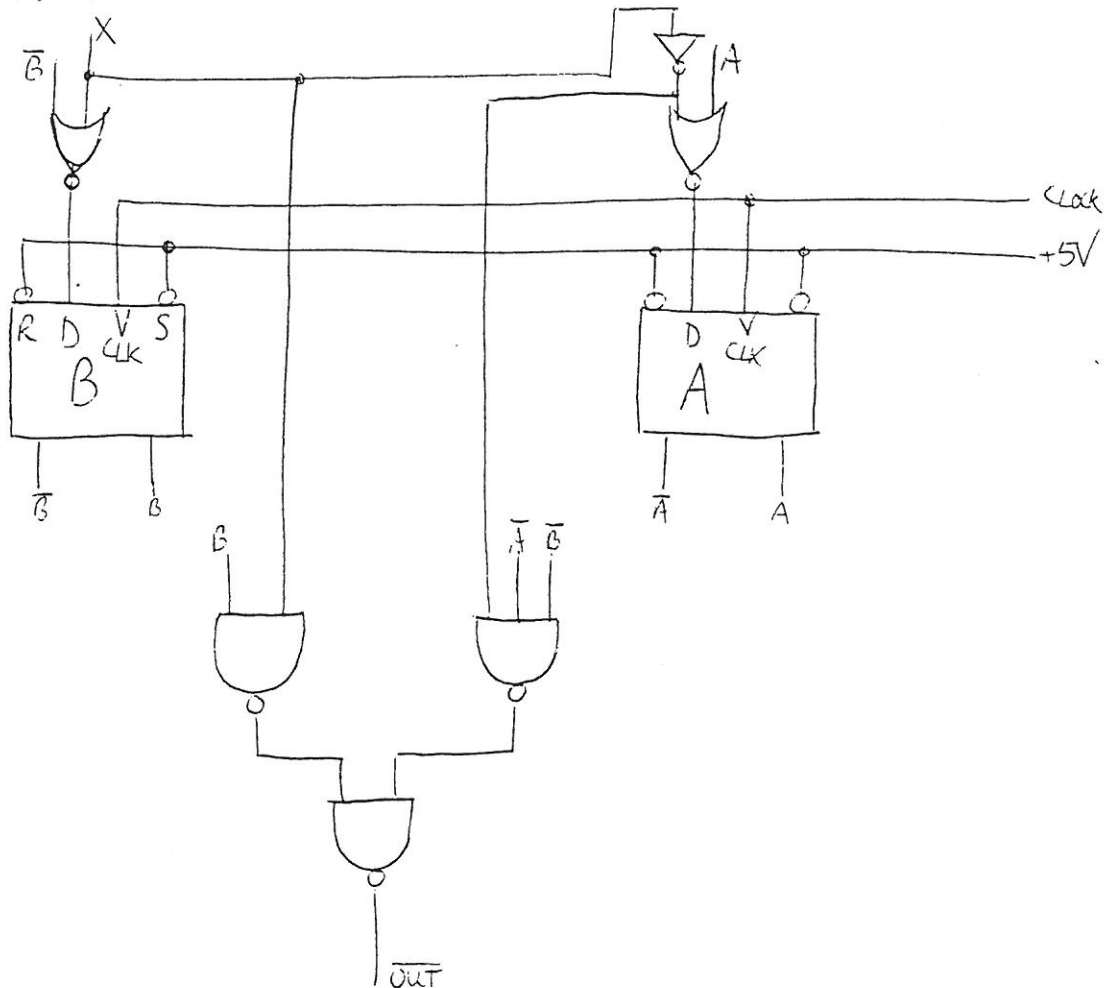
A	0	1
0	1	0
1	1	1

T_B

Present State	Next State Code	True Next State
A	T _A	A
0	0	0
0	1	0
1	0	0
1	1	0



Model Answer: Problem 3



$$D_B = \overline{B + X} = B.X$$

$$D_A = \overline{A + X} = \overline{A}.X$$

$$\overline{OUT} = (\overline{B.X}).(\overline{A.B.X})$$

$$OUT = (\overline{B.X}).(\overline{A.B.X})$$

$$= (\overline{B} + \overline{X}).(\overline{A} + \overline{B} + \overline{X})$$

$$= \overline{B}.A + \overline{B}.B + \overline{B}.X + \overline{X}.A + \overline{X}.B + \overline{X}.X$$

$$= \overline{B}.A + \overline{B}.X + \overline{X}.A + \overline{X}.B$$

A \ B	B			
	00	01	11	10
0	0	1	1	0
1	0	0	0	0

D_B

A \ B	B			
	00	01	11	10
0	0	0	0	0
1	1	1	0	0

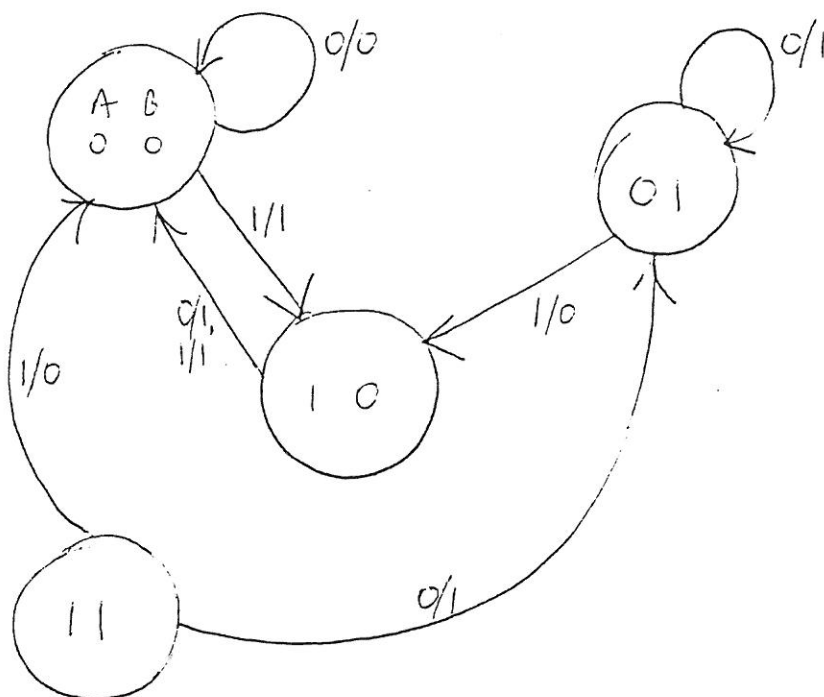
D_A

A \ B	B			
	00	01	11	10
0	0	1	1	1
1	1	0	0	1

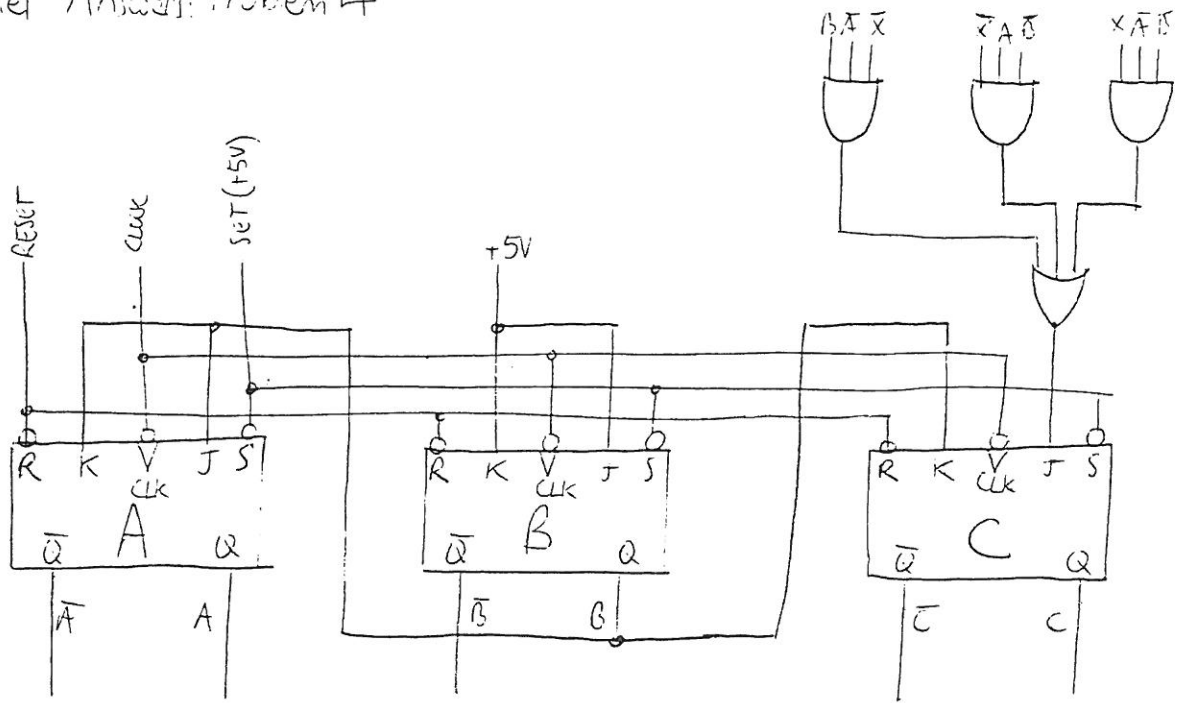
OUT

	PRESENT STATE		INPUTS	NEXT STATE CODE		OUTPUT	TRUE NEXT STATE	
	A	B		D _A	D _B		A	B
STATE 0	0	0	0	0	0	0	0	0
	0	0	1	0	0	1	1	0
STATE 1	0	0	0	0	1	1	0	1
	0	1	1	0	0	0	1	0
STATE 2	1	0	0	0	0	1	0	0
	1	0	1	0	0	1	0	0
STATE 3	1	1	0	0	1	1	0	1
	1	1	1	0	0	0	0	0

FORMAT: X/OUT



Model Answers: Problem 4



$$K_A = B; J_A = G, K_G = I; \overline{F}_D = 1; K_C = B$$

$$J_c = B \cdot \bar{A} \cdot \bar{X} + \bar{X} \cdot A \cdot \bar{B} + X \cdot \bar{A} \cdot \bar{B}$$

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

$$k_A = J_A = k_c = \beta$$

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

$$T_c$$

	Present State			Inputs	Next State Codes						True Next State		
	A	B	C		T _A	K _A	T _B	K _B	T _C	K _C	A	B	C
State 0	0	0	0	0	0	0	1	1	0	0	0	1	0
	0	0	0	1	0	0	1	1	1	0	0	1	1
1	0	0	1	0	0	0	1	1	0	0	0	1	0
	0	0	1	1	0	0	1	1	1	0	0	1	1
2	0	1	0	0	1	1	1	1	1	1	1	0	1
	0	1	0	1	1	1	1	1	0	1	1	0	0
3	0	1	1	0	1	1	1	1	1	1	1	0	1
	0	1	1	1	1	1	1	1	0	1	1	0	0
4	1	0	0	0	0	0	1	1	1	0	1	1	1
	1	0	0	1	0	0	1	1	0	0	1	1	1
5	1	0	1	0	0	0	1	1	1	0	1	1	1
	1	0	1	1	0	0	1	1	0	0	1	1	1
6	1	1	0	0	1	1	1	1	0	1	0	0	0
	1	1	0	1	1	1	1	1	0	1	0	0	0
7	1	1	1	0	1	1	1	1	0	1	0	0	0
	1	1	1	1	1	1	1	1	0	1	0	0	0

FORMAT X

