

1. Complete the following table of equivalent values.

Binary	Octal	Decimal	Hexadecimal
1011.0011	13.14	11.1875	B.3
11101.11111101	35.77	29.99	1D.FD
11011.010011	33.23	$27 \frac{19}{64}$	1B.4C

2. Calculate the following

- a) $(11001)_2$ plus $(101)_2$ = 11110
- b) 11010_2 minus 10101_2 using 2's complement representation = 000101
- c) 1101_2 times 1001_2 = 1110101

3. Complete the following table of equivalent values. Use binary numbers with a sign bit and 7 bits for the value

Decimal	Unsigned	Signed Magnitude	Signed 2's Complement
12	0001100	0001100	0001100
-12	--	1001100	1110100
45	0101101	0101101	0101101
-34	--	1100010	1011110

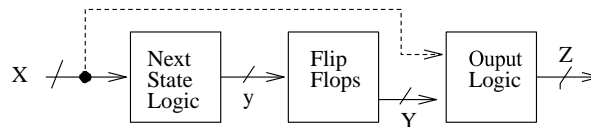
4. (a) Explain the difference between a **Moore** machine and a **Mealy** machine.

Sol The outputs in a Moore machine depend only on the present state. The outputs in a Mealy machine depend on both the present state and the present input.

(b) What is the same about both kinds of state machines?

Sol Both have present state dependent on past inputs.

(c) Draw a block diagram indicating the structure of a general state machine. Indicate on the diagram where one can find the **present state** and **nextstate**.



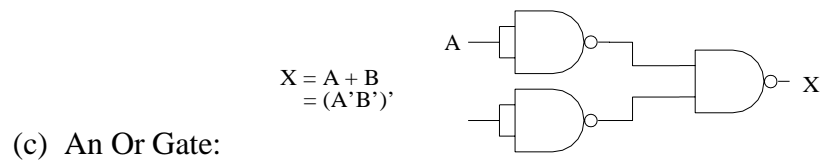
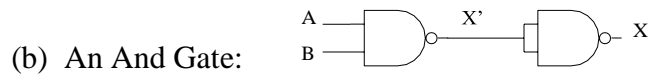
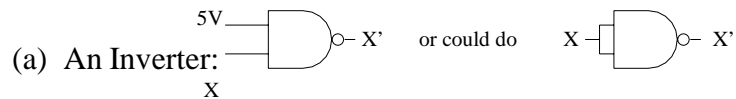
5. Give a truth table and a standard sum of products expression that describes

$$F = A \oplus B \oplus C$$

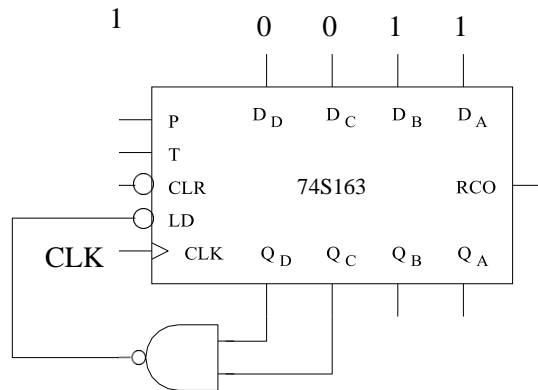
A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$F = AB'C' + A'B'C + A'BC' + ABC$$

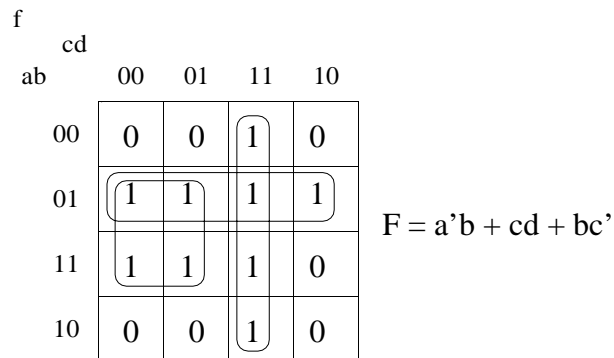
6. Indicate how a NAND gate can be used to implement:



7. Using the 74ALS163 counter shown below and logic gates design a counter that counts in the sequence 3,4, 5, 6, 7, 8, 9, 10, 11, 12, 3, ... Connect all unused inputs. The counter may cycle through several unwanted states before settling into the final count sequence. Q_d is the MSB of the counter output.

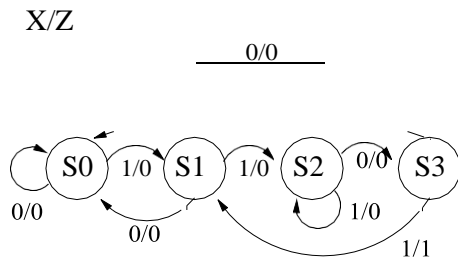


8. Find a minimum sum of products expression for $F = abc' + bc'd' + cd + a'b$

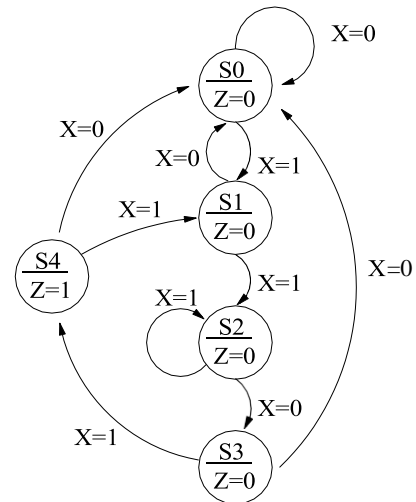


9. Create a state diagram for a sequence detector that outputs a 1 when it detects the final bit in the serial data stream 1101. Assume overlapping is allowed.

Mealy Machine



Moore Machine



10. Determine the D flip-flop equations for the system represented with in the state- transition table below. Assign states: $S_0 = 00$, $S_1 = 01$, $S_2 = 10$ and $S_3 = 11$.

Present AB	Present S	Next State		Output Z
		X=0	X=1	
00	S ₀	S ₁	S ₂	0
01	S ₁	S ₁	S ₂	1
10	S ₂	S ₂	S ₃	1
11	S ₃	S ₃	S ₀	0

Da

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

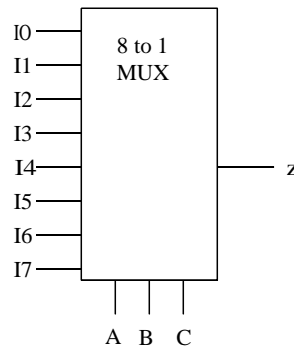
$$D_a = X'A + XA' + \begin{cases} AB' \\ XB' \end{cases}$$

Db

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

$$D_b = XAB' + X'A' + X'B$$

11. Give the output expression for the 8-to-1 MUX shown below.



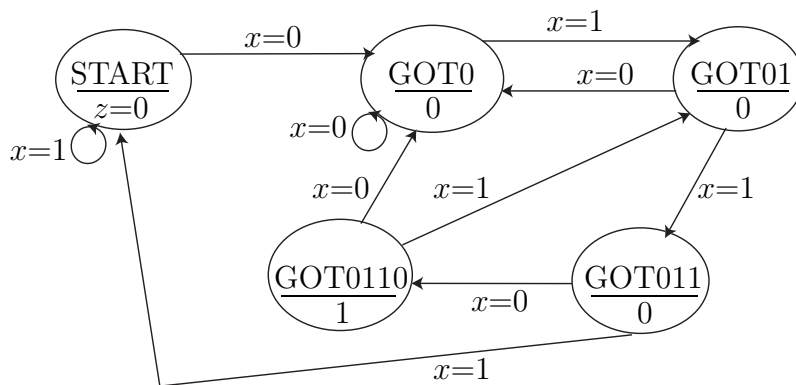
$$Z = A'B'C'I_0 + A'B'CI_1 + A'BC'I_2 + A'BCI_3 + AB'C'I_4 + AB'CI_5 + ABC'I_6 + ABCI_7$$

12. Finite State Machines: Sequence Recognizer

You want to build a finite state machine that will recognize the sequence $x = 0110$ and output the sequence $z = 0001$ as this sequence occurs. In other words, output $z = 0$ when first receiving $x = 0$. Then output $z = 0$ if the next bit of $x = 1$; output $z = 0$ again if the following bit of $x = 1$. Finally, if the last (fourth) bit of $x = 0$, output $z = 1$. More simply, output $z = 0$ until the sequence $x = 0110$ is received, at which time output $z = 1$.

1(a). First, draw and label the transitions in the state bubble diagram below. The states are already labeled (but state bit values have not been assigned). Allow overlap of sequences. Build as a Moore machine. Include the input bits of x and output bits of z .

Solution:



Note: This sequence recognizer allows for overlap of sequences, and thus outputs $z = 1$ whenever the sequence $x = 0110$ is recognized with overlap of sequences. If we do not allow for overlap, then the transitions from state GOT0110 will go to state START for both $x = 0$ and $x = 1$. Without overlap, whenever the sequence $x = 0110$ is input, the output will be $z = 0001$.

12(b). Write the state table from the state bubble diagram. Fill in the table given below. Use the following state assignments: START=000, GOT0=001, GOT01=010, GOT011=011 and GOT0110=100. All unused states should go to the START state and output $z = 0$. Assume you will build the circuit with J-K flip-flops.

Solution: (next page)

Present State			Input	Output	Next State			Flip-Flop Inputs					
Q_2	Q_1	Q_0	x	z	Q_2^*	Q_1^*	Q_0^*	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	0	0	1	0	X	0	X	1	X
0	0	0	1	0	0	0	0	0	X	0	X	0	X
0	0	1	0	0	0	0	1	0	X	0	X	X	0
0	0	1	1	0	0	1	0	0	X	1	X	X	1
0	1	0	0	0	0	0	1	0	X	X	1	1	X
0	1	0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	0	0	1	0	0	1	X	X	1	X	1
0	1	1	1	0	0	0	0	0	X	X	1	X	1
1	0	0	0	1	0	0	1	X	1	0	X	1	X
1	0	0	1	1	0	1	0	X	1	1	X	0	X
1	0	1	0	0	0	0	0	X	1	0	X	X	1
1	0	1	1	0	0	0	0	X	1	0	X	X	1
1	1	0	0	0	0	0	0	X	1	X	1	0	X
1	1	0	1	0	0	0	0	X	1	X	1	0	X
1	1	1	0	0	0	0	0	X	1	X	1	X	1
1	1	1	1	0	0	0	0	X	1	X	1	X	1

12(c). Find the minimized logic equations for output z and flip-flop inputs J_2 , K_2 , J_1 , K_1 and J_0 , K_0 ; use K-maps if needed.

Solution:

Output z :

Output z can be determined directly from the truth table, as $z = 1$ only when $Q_2 = 1$, $Q_1 = 0$ and $Q_0 = 0$. Therefore, $z = Q_2 \overline{Q_1} \overline{Q_0}$.

You may check with a K-map if you prefer.

$$z = Q_2 \overline{Q_1} \overline{Q_0}$$

K-map for J_2 :

$Q_2 \backslash Q_1$	Q_0	00	01	11	10
00	0	0	0	0	0
01	0	0	0	1	
11	X	X	X	X	
10	X	X	X	X	

From the K-map above, we obtain the following expression for J_2 :

$$\boxed{J_2 = Q_1 Q_0 \bar{x}}$$

K₂:

Note that K_2 is either X or 1 for every entry of the truth table. Therefore, we may set

$$K_2 = 1.$$

$$\boxed{K_2 = 1}$$

K-map for J_1 :

$Q_2 \backslash Q_1$	$Q_0 x$	00	01	11	10
00	0	0	1	0	
01	X	X	X	X	
11	X	X	X	X	
10	0	1	0	0	

From the K-map above with two groupings, we obtain the following expression for J_1 :

$$J_1 = \bar{Q}_2 Q_0 x + Q_2 \bar{Q}_0 x$$

This expression may be further simplified by recognizing that $\bar{Q}_2 Q_0 + Q_2 \bar{Q}_0 = Q_2 \oplus Q_0$;

$$\boxed{J_1 = x(Q_2 \oplus Q_0)}$$

K-map for K_1 :

$Q_2 \backslash Q_1$	$Q_0 x$	00	01	11	10
00	X	X	X	X	
01	1	0	1	1	
11	1	1	1	1	
10	X	X	X	X	

From the K-map above with three groupings, we obtain the following expression for K_1 :

$$\boxed{K_1 = Q_0 + Q_2 + \bar{x}}$$

K-map for J_0 :

$Q_2 \backslash Q_1$	$Q_0 \bar{x}$	00	01	11	10
00		1	0	X	X
01		1	1	X	X
11		0	0	X	X
10		1	0	X	X

From the K-map above with two groupings, we obtain the following expression for J_0 :

$$J_0 = \overline{Q_1} \bar{x} + \overline{Q_2} Q_1$$

K-map for K_0 :

$Q_2 \backslash Q_1$	$Q_0 \bar{x}$	00	01	11	10
00		X	X	1	0
01		X	X	1	1
11		X	X	1	1
10		X	X	1	1

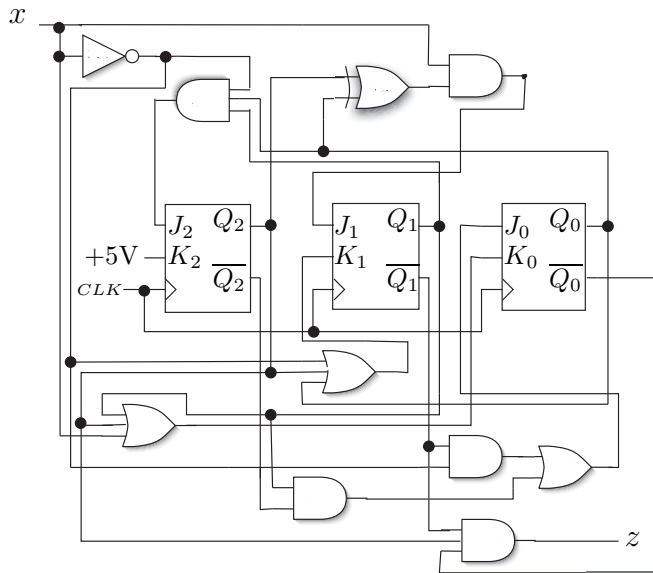
From the K-map above with three groupings, we obtain the following expression for K_0 :

$$K_0 = Q_1 + Q_2 + x$$

12(d). Draw the corresponding circuit diagram with J-K flip-flops. Label all inputs and outputs.

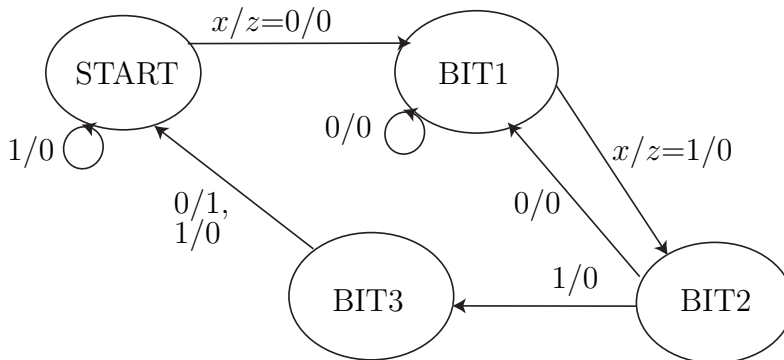
Solution: (next page)

Note: The black dots in the figure indicate a connection between wires. If there is no black dot, there is no connection between wires that cross.



13. State Diagram of Mealy Machine Redraw the state diagram using a Mealy machine design. Be sure to label the transitions and bubbles. You may name your states whatever you like. Again allow overlap of sequences. How many states and flip-flops do you need for the Mealy design?

Solution:

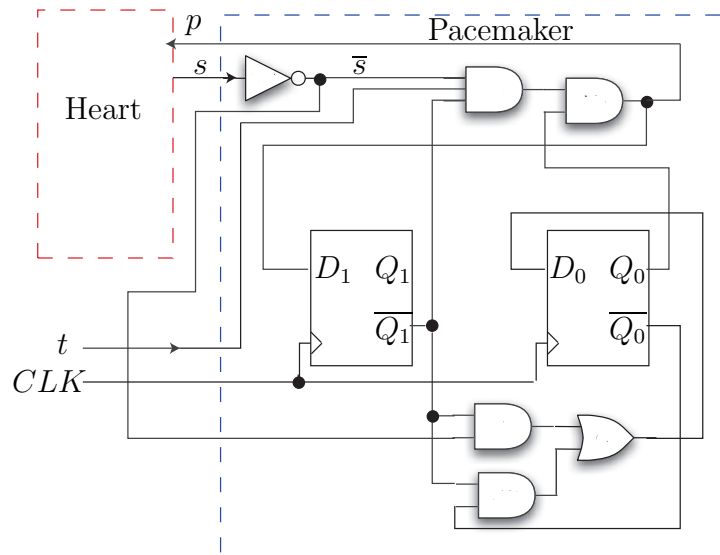


Only four states are required for the Mealy machine design. Two state bits are thus needed to determine all the states. This means that only two flip-flops are required to implement the Mealy machine design of the sequence recognizer.

14. Analysis of FSM Circuit Design

Below is a simplified block diagram of a heart pacemaker. The output p from the pacemaker pulses high if the heart does not contract within a certain time. The input s indicates whether the heart has contracted ($s = 1$) or not ($s = 0$). The input t comes from a timer,

which counts for the expected time between contractions (approximately 1 second). When $t = 1$, the timer has counted up to 1 second, and the heart should have contracted. If the heart has not contracted within that time, the pacemaker sends a pulse $p = 1$. (There is also a timer reset control coming from the pacemaker, which we ignore for this problem.) The inputs to the pacemaker circuit are s (from the heart) and t (from a timer - not shown). The output of the pacemaker circuit is p , which goes to the heart. Arrows indicate the inputs and output.



Note: The black dots in the figure indicate a connection between wires. If there is no black dot, there is no connection between wires that cross.

14(a). Is this a Mealy or a Moore machine? Why?

Solution:

This is a Mealy machine. The output p depends on the input s , as well as the sequential logic output states Q_1 and Q_0 and the input t from a timer clocked synchronously with FF1 and FF0. This means that the output p can change at times other than on a rising clock edge, if the input s changes. The input s is not clocked, and is not constrained to change only on a rising clock edge.

14(b). Write the equations for output p and the flip-flop inputs D_1 and D_0 from the block diagram.

Solution:

Output p :

$$p = \overline{Q_1} Q_0 \overline{s} t$$

D input for Flip-flop 1, D_1 :

$$D_1 = p = \overline{Q_1} Q_0 \overline{s} t$$

D input for Flip-flop 0, D_0 :

$$D_0 = \overline{Q_1} \overline{Q_0} + \overline{Q_1} \overline{s}$$

14(c). Fill in the state table below, using the equations obtained in part **3(b)**.

Solution:

Present State		Inputs		Output	Next State	
Q_1	Q_0	s	t	p	Q_1^*	Q_0^*
0	0	0	0	0	0	1
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	0	0	1
0	1	0	1	1	1	1
0	1	1	0	0	0	0
0	1	1	1	0	0	0
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	0	0	0
1	1	0	0	0	0	0
1	1	0	1	0	0	0
1	1	1	0	0	0	0
1	1	1	1	0	0	0