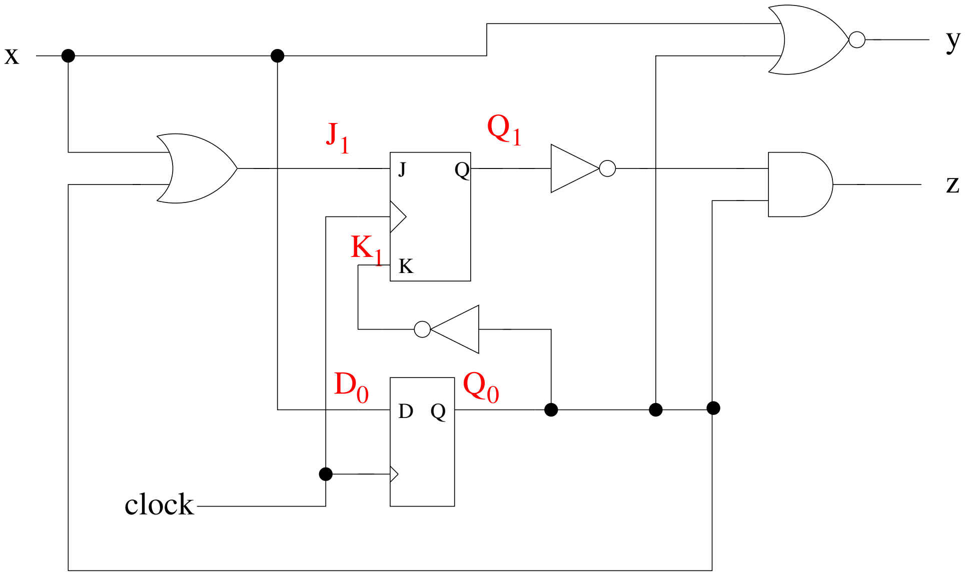
EE 231

Homework 12

Due November 22, 2010

1. Consider the circuit below:



(a) Is this a Mealy machine or a Moore machine? Explain.

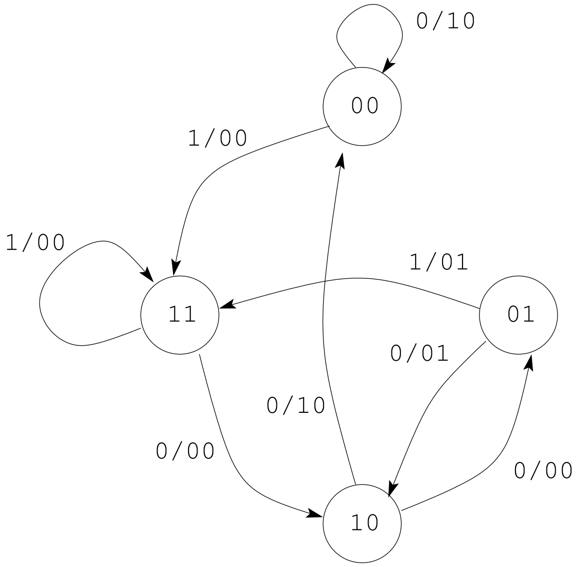
This is a Mealy machine – output y depends only both the present state and the present input.

(b) Find the state transition table for the circuit.

The equations are: $J_1 = x + Q_0$, $K_1 = Q_0'$, $D_0 = x$, $y = (x + Q_0)'$, $z = Q_1'Q_0$.

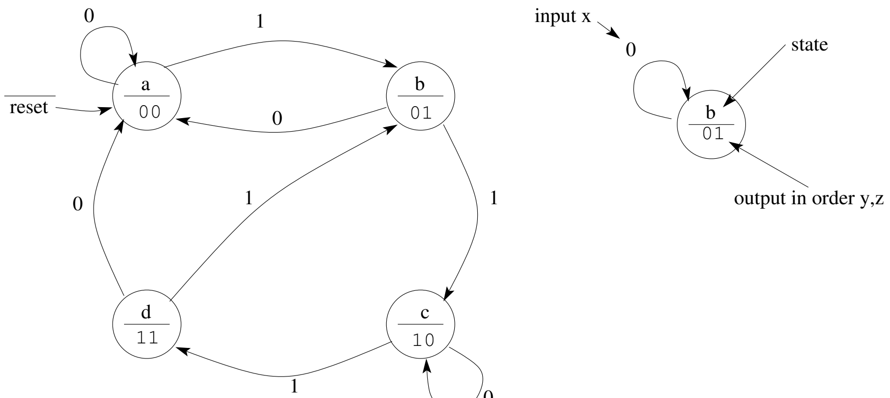
Q_1	Q_0	x	J_1	K_1	Q_1	Q_0	y	z
0	0	0	0	1	0	0	1	0
0	0	1	1	1	1	1	0	0
0	1	0	1	0	1	0	0	1
0	1	1	1	0	1	1	0	1
1	0	0	0	1	0	0	1	0
1	0	1	1	1	0	1	0	0
1	1	0	1	0	1	0	0	0
1	1	1	1	0	1	1	0	0

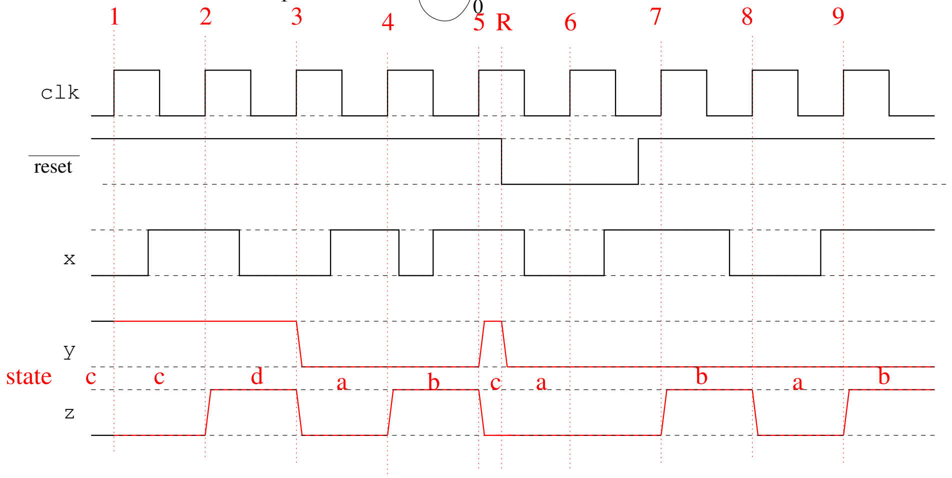
(c) Draw the state transition diagram for the circuit.



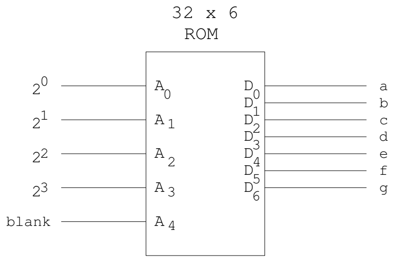
2. Consider the state tranition diagram below for a state machine with one input x (plus *clock* and $\overline{reset}$) and two outputs y and z. Complete the timing diagram for the outputs y and z. You do not have to find the state transition table. You should be able to go directly from the state diagram to the timing diagram.

   Before the first clock pulse, the outputs are 10, so the system is in state c. On the first clock pulse, x is low so the system stays is state c (10). On clock pulse 2, the system is in state c with x high, so the system goes to state d (11). On the third clock pulse, the system is in state d with x low so the system goes to state a (00). On the 4th clock pulse the system is in state a with x high so the system goes to state b (01). When reset goes low at R, the system goes to the reset state a (00). On the 6th clock pulse, reset is still low so the system stays in state a. On the 7th clock pulse, the system is in state a with x high so the system goes to state b (01). On the 8th clock pulse the system is in state b with x low, so the system goes to state a (00). On the eigth clock pulse the system is in state a with x high so the sytem goes to state b (01).





3. The 32 x 7 ROM, as shown below, is to be programmed as a 7-segment decoder with blanking. It is used to drive a seven-segment LED where the inputs are active low (a 0 turns on the segment, a 1 turns off the segment). When the `blank` input is high, all the segments `a` through `g` should be high. When the `blank` input is low, the 4-bit hex value should be displayed on the seven-segment display. For example, when the input is 00010, the display should be active, and should display a 2, so segments `a`, `b`, `d`, `e` and `g` should be low, and segments `c` and `f` should be high. Specify the truth table for the ROM.



Addr	A_4	A_3	A_2	A_1	A_0	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hex	Char
00	0	0	0	0	0	1	0	0	0	0	0	0	40	0
01	0	0	0	0	1	1	1	1	1	0	0	1	79	1
02	0	0	0	1	0	0	1	0	0	1	0	0	24	2
03	0	0	0	1	1	0	1	1	0	0	0	0	60	3
04	0	0	1	0	0	0	1	0	1	1	0	0	2C	4
05	0	0	1	0	1	0	0	0	1	0	0	1	05	5
06	0	0	1	1	0	0	0	0	0	0	0	1	01	6
07	0	0	1	1	1	1	0	1	1	1	0	0	5C	7
08	0	1	0	0	0	0	0	0	0	0	0	0	00	8
09	0	1	0	0	1	0	0	0	1	0	0	0	08	9
0A	0	1	0	1	0	0	0	0	0	1	0	0	04	A
0B	0	1	0	1	1	0	1	0	0	0	0	1	21	b
0C	0	1	1	0	0	0	1	1	0	0	1	1	33	c
0D	0	1	1	0	1	0	1	1	0	0	0	0	30	d
0E	0	1	1	1	0	0	0	0	0	0	1	1	03	E
0F	0	1	1	1	1	0	0	0	0	1	1	1	07	F
10	1	0	0	0	0	1	1	1	1	1	1	1	7F	Blank
11	1	0	0	0	1	1	1	1	1	1	1	1	7F	Blank
12	1	0	0	1	0	1	1	1	1	1	1	1	7F	Blank
13	1	0	0	1	1	1	1	1	1	1	1	1	7F	Blank
14	1	0	1	0	0	1	1	1	1	1	1	1	7F	Blank
15	1	0	1	0	1	1	1	1	1	1	1	1	7F	Blank
16	1	0	1	1	0	1	1	1	1	1	1	1	7F	Blank
17	1	0	1	1	1	1	1	1	1	1	1	1	7F	Blank
18	1	1	0	0	0	1	1	1	1	1	1	1	7F	Blank
19	1	1	0	0	1	1	1	1	1	1	1	1	7F	Blank
1A	1	1	0	1	0	1	1	1	1	1	1	1	7F	Blank
1B	1	1	0	1	1	1	1	1	1	1	1	1	7F	Blank
1C	1	1	1	0	0	1	1	1	1	1	1	1	7F	Blank
1D	1	1	1	0	1	1	1	1	1	1	1	1	7F	Blank
1E	1	1	1	1	0	1	1	1	1	1	1	1	7F	Blank
1F	1	1	1	1	1	1	1	1	1	1	1	1	7F	Blank

4. Specify the size of a ROM (number of words and number of bits per word) that will accomodate the truth table for the following combinational circuit components:

   (a) a binary multiplier the multiplies two 6-bit binary words.

   When you multiply two 6-bit numbers, you can get a 12-bit number. Need 12 address lines (for the first and second 6-bit numbers), for $2^{12} = 4,096$ words; each word should be 12 bits to hold the 12-bit result.

   (b) a greatest common divisor generator, which finds the greatest common divisior of two eight-bit numbers.

   The GCD could be as large as 8 bits. Need 16 address lines (for the first and second 8-bit numbers), for $2^{16} = 65,536$ words; each word should be 8 bits to hold the 8-bit result.

   (c) a quadruple two-to-one line multiplexer with common select and enable inputs.

   Each mux will have two inputs and one output, so four muxes will have $4 \times 2 = 8$ inputs and $4 \times 1 = 4$ outputs. Also need one common select input and one common enable input, for a total of $8 + 1 + 1 = 10$ input (address) lines. Need $2^{10} = 1,024$ words; each word should be 4 bits.

5. Derive the PAL programming table for a 4-bit decimal-to-gray-code converter. The Gray code table is on Page 22 of the text. The input should be the 4-bit decimal number, and the output should be the Gray code for that number. Draw the corresponding fuses on the attached PAL figure.

   The truth table for the Hex to Gray converter is:

I_3	I_2	I_1	I_0	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

The equations are: $O_3 = I_3$, $O_2 = I_3' I_2 + I_3 I_2'$, $O_1 = I_2 I_1' + I_2' I_1$, and $O_0 = I_1' I_0 + I_0 I_1'$.

O_3 comes directly from I_3 so there is no need to run it through the PAL. Here are the fuses which should remain:

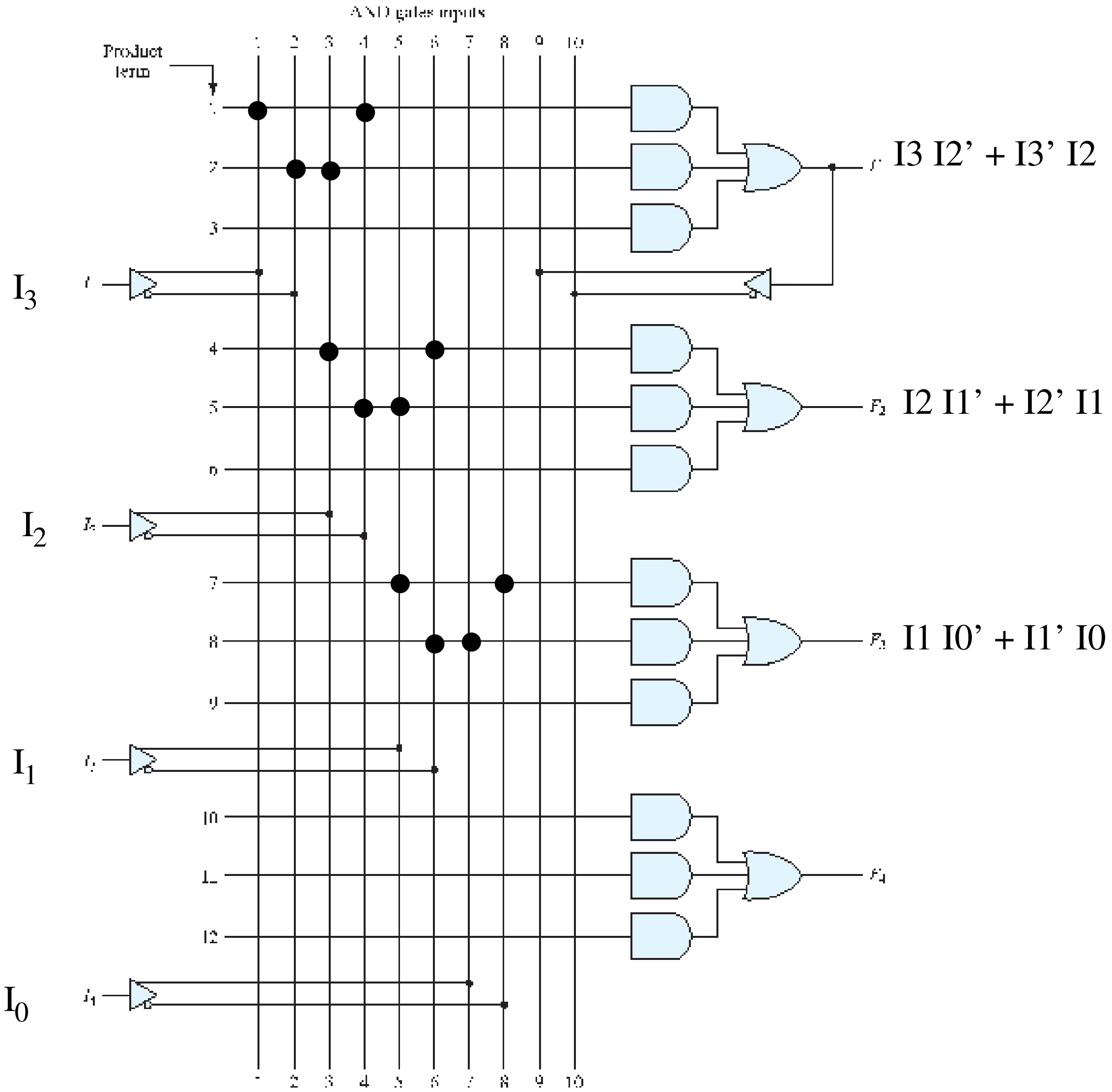


Figure Number 07.16
Mano/Ciletti
Digital Design 4e