DIGITAL ELECTRONICS II
Revision Examples 2011

Exam Format

3 compulsory questions: Q1 has 5 parts worth 8% each, Q2,3 are worth 30%.
Q1 is the same standard as in previous years, Q2,3 are similar to Q2,3,4 in previous years but a little bit easier.

Revision Lectures
Three revision lectures will be given on the topics listed below. The example problems are all taken from past papers and are intended to cover the full range of the course. It should not be assumed that this year's questions will be on these precise topics. Several exam papers from previous years are available at http://www.ee.ic.ac.uk/hp/staff/dmb/courses/dig2/dig2.htm. Note that papers prior to 2008 contain questions on integrating converters and those prior to 2003 contain questions on CMOS gate circuitry – these topics are no longer in the syllabus.

Fri 6 May 3:00	1. State Machines (EEE 1994 Q1) 2. Adder Circuit Propagation Delays (ISE 1996 Q3)
Wed 11 May 9:00	3. Timing (ISE 1996 Q4) 4. Analog/Digital converter (EEE 2004 Q1)
Fri 20 May 9:00	5. Memory interfacing (EEE 1996 Q3) 6. State Machine analysis (EEE 1999 Q1)

1. *Figure 1* shows the state diagram of a synchronous state machine having a single input, IN, and a single output, OUT. The state is represented by a two-bit number S0:1 whose value is indicated within each state circle along with the value of OUT. Transitions from a state to itself have not been shown.

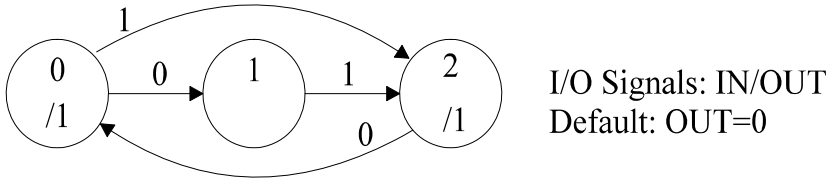


Figure 1

(a) If the next state is represented by a two bit number NS0:1, derive Boolean expressions for NS1, NS0 and OUT in terms of S1, S0 and IN. You should ensure that the state machine will ultimately follow the correct sequence regardless of its initial state. [7]

[2]

[6]

[5]

(b) Using a 2-bit D-type register and as few logic gates as possible draw a circuit diagram for the state machine. The register operates on the rising edge of the signal CLOCK.

(c) Complete the timing diagram shown in *Figure 2* by showing the state during each clock cycle and the waveform of OUT. The circuit is initially in state 2.

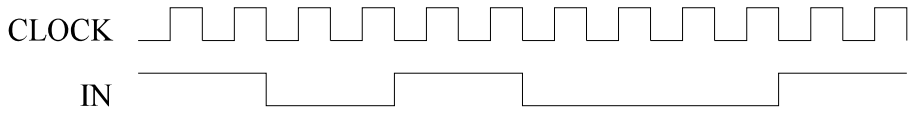


Figure 2

----------Solution ----------

(a) We can draw Karnaugh maps for NS1, NS0 and OUT:

	NS1	IN 0	IN 1
S1,S0	00	0	1
	01	0	1
	11	X	X
	10	0	1

NS0	IN 0	IN 1
00	1	0
01	1	0
11	X	X
10	0	0

OUT	IN 0	IN 1
00	1	1
01	0	0
11	X	X
10	1	1

From this we get:

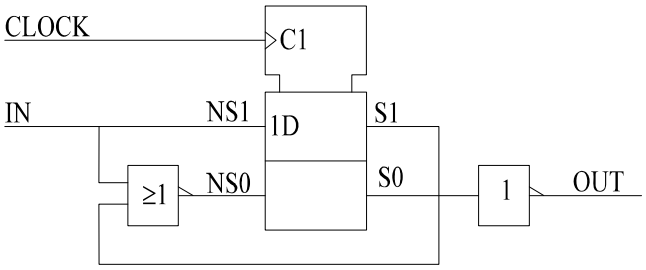
$NS1 = IN$

$NS0 = \overline{S1} \cdot \overline{IN} = \overline{S1 + IN}$

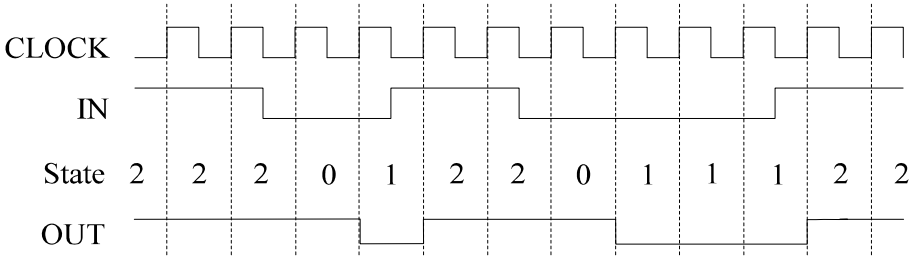
$OUT = \overline{S0}$

State 3 branches to either state 0 (IN=0) or state 2 (IN=1) so there is no chance of getting trapped.

(b)



(c)



2. *Figure 3* shows the symbol and worst case propagation delays (in units of 1 gate delay) for an *m*-bit full adder. P*i*, Q*i* and S*j* denote any of the P, Q or S bits respectively.

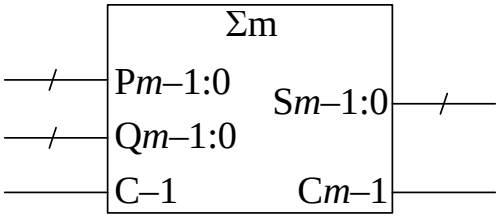
The circuit of *Figure 4* uses two full adders together with two multiplexers. The C–1 inputs of the two full adders are connected to logic 0 and logic 1 levels respectively. In the question below, the phrases *simple adder block* and *complex adder block* refers to the circuits of *Figure 3* and *Figure 4* respectively.

a) Determine the worst case delays from each of C–1 and P*i* to each of C*m*–1 and S*j* for the *complex adder block*. Each multiplexer has a propagation delay of 3 gate delays from its select input (labelled G1) and 2 gate delays from the other inputs. [7]

b) Show that the *complex adder block* is functionally equivalent to the *simple adder block*. Explain the relative advantages of the two implementations. [4]

c) A 16-bit adder is constructed by cascading four 4-bit full adder blocks. A *simple adder block* is used for the least significant 4 bits and three *complex adder blocks* are used for the 12 most significant bits. Determine the longest propagation delay path for the 16-bit adder. [6]

d) Determine the number of bits to which each of the complex adder blocks in part (c) can be increased without increasing the length of the longest propagation delay path. Give the total size of the resultant adder. [3]



Worst-case delays

C–1 → *Sj*	*m*+3
C–1 → C*m*–1	*m*
P*i* → S*j*	*m*+3
P*i* → C*m*–1	*m*

Figure 3

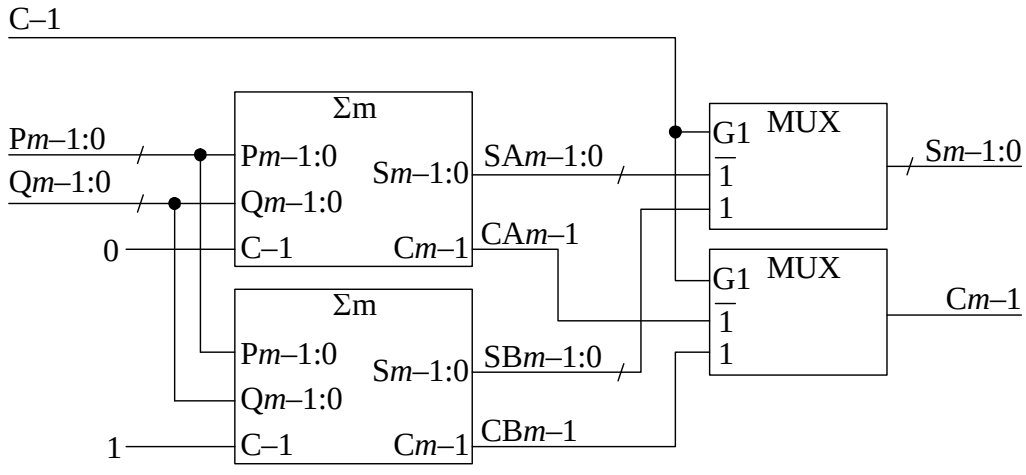


Figure 4

----------Solution ----------

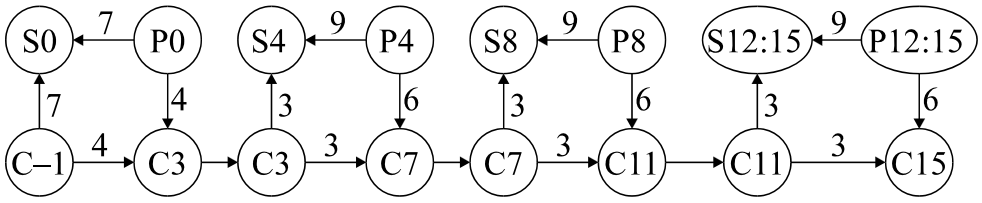
a)

C–1 → S*j*	3
C–1 → C*m*–1	3
P*i* → SA*j*, SB*j* → S*j*	(*m*+3) + (2) = *m*+5
P*i* → CA, CB → C*m*–1	(m) + (2) = *m*+2

b) The two simple adder blocks are identical except that the C–1 input is 0 for the upper one and 1 for the lower one. The multiplexers then select the outputs from the appropriate block according to the actual value of C–1.

The complex adder block uses over twice as much circuitry as the simple adder block. Its main advantage is the reduction (for m>3) in the carry path delay. The delays from Pi to Si within the block have actually been increased.

c) The diagram shows the worst-case delay between various inputs and outputs. In all cases *m* = 4.



The longest path is P0→C3→C7→C11→S15 = 4+3+3+3 = 13

[Note that using a complex block for the 4 least significant bits would lengthen this path by 2 since P0 → C3 would increase to 6]

d) We have the following delays:

P4→ C7→C11→S15 = 12

P8→C11→S15 = 9

P12→S15 = 9

We can increase the size of each block to bring these numbers up to 13: make the second block 5 bits and the third and fourth blocks 8 bits. This gives a 25-bit adder with a total delay of 13 gate delays.

3. *Figure 5* shows a circuit comprising two microprocessors and two line-drivers. The microprocessors communicate via a cable that is 300 cm long through which both the clock and data signals travel. The propagation speed of the cable is 15 cm/ns. The propagation delay of the line-drivers may vary between 10 and 15 ns.

As indicated in *Figure 5*, the Z output of each microprocessor changes shortly after the *falling* edge of its clock input while data at the D input is clocked in on the *rising* edge. The timing specifications for these signals are:

t_{zp} = 70 ns	Propagation delay of Z output
t_{zh} = 10 ns	Hold time of Z output
t_{ds} = 20 ns	Setup time of D input
t_{dh} = 30 ns	Hold time of D input

a) The signal CLOCK is a symmetrical squarewave of constant frequency. Determine f_{max}, the clock frequency up to which the circuit will allow microprocessor B to transmit data reliably to microprocessor A. [10]

b) A supply of registers is available having a propagation delay of 10 ns. The registers are available in two types which respond to rising and falling edges respectively at their clock inputs. The data inputs of the registers have a setup time of 5 ns and a hold time of 5 ns relative to the active clock edge.

Determine the highest value of f_{max} that can be achieved by inserting registers at either or both of points X and Y in the figure. You should indicate the clock polarity of each register that you use and its position in the circuit.

State the additional delay in clock cycles that your circuit modification has inserted between the Z output of microprocessor B and the D input of microprocessor A. [10]

[**Note:** You should not assume that all the information given in this question is required in order to answer it]

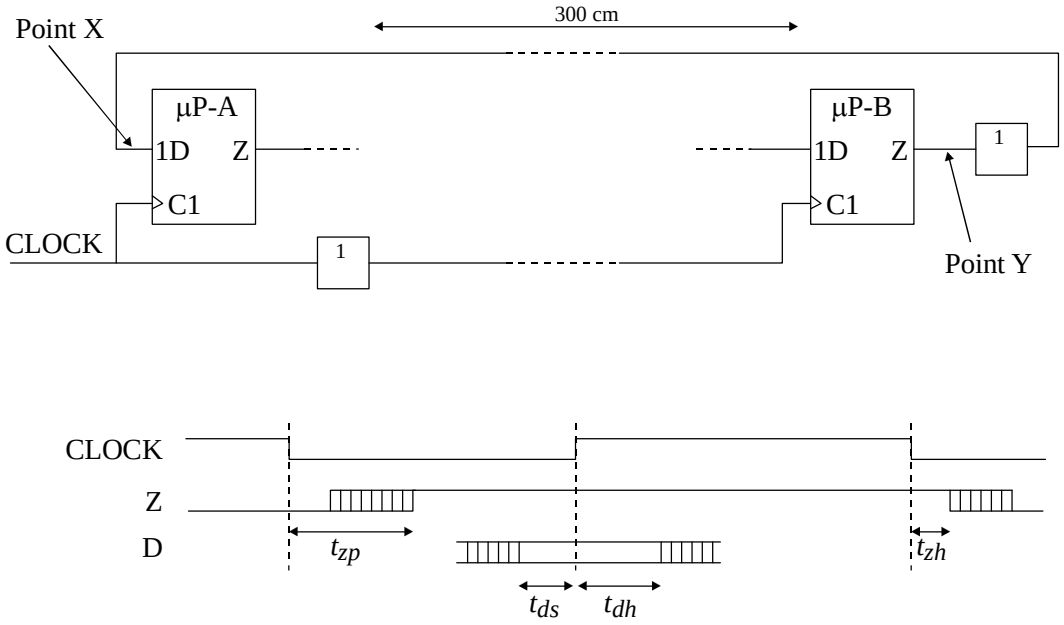


Figure 5

----------Solution ----------

We choose the driver delay to be t_D = 10 or t_D = 15 according to whether it is on the right or left of an inequality respectively. The cable delay is t_c = 20 ns. Define the clock period to be T.

a) $t_D + t_c + t_{zp} + t_D + t_c < ½T - t_{ds} \Rightarrow 15 + 20 + 70 + 15 + 20 < ½T - 20 \Rightarrow T > 320$

Hence the max frequency is 1/320ns = 3.125 MHz

b) A register inserted at point X must respond to a falling clock edge because the data will otherwise change during the hold period t_{dh}.

A register inserted at point Y can respond to either edge, but a falling edge will allow more time to encompass t_{zp}+ cable delay.

We therefore put a falling-edge-clocked register at both points X and Y. This inserts an extra 2 clock cycles delay.

For the registers we have $t_s = t_h$ = 5 and t_p = 10. This gives the following constraints:

i) µP-B to Register-Y: $t_{zp} < T - t_s \Rightarrow \quad T > 75$

ii) Register-Y to Register-X: $t_D + t_c + t_p + t_D + t_c < T - t_s \quad \Rightarrow \quad T > 85$

iii) Register-X to µP-A: $t_p < ½T - t_{ds} \quad \Rightarrow \quad T > 60$

Hence the max frequency is 1/85ns = 11.8 MHz

4. *Figure 6* shows the circuit for a successive approximation Analog-to-Digital converter having an input voltage *V* and a twos-complement signed 8-bit output *X7:0* in the range −128 to +127. One LSB of the converter is equal to 0.2 V and input voltages in the range ±0.1 V are converted to the value 0. [3]

   (a) If the input voltage *V* equals –10.45 V, determine the value of the output *X7:0*. [11]

   (b) Give the sequence of values taken at *X7:0* during the process of converting an input voltage of –10.45 V and give the corresponding voltages at *W*. [3]

   (c) The aperture of the sampling switch is 50 ns and the propagation delays of the D/A converter and comparator are 100 ns and 60 ns respectively. Neglecting any propagation delays in the control logic, calculate the minimum time to convert an input voltage. You may assume that *X7:0* is set to the correct initial value before the conversion process begins. [3]

   (d) If the comparator input current lies in the range ±2 μA, calculate the minimum value of *C* to ensure that the capacitor voltage changes by no more than ½ LSB during the conversion time found in part (c).

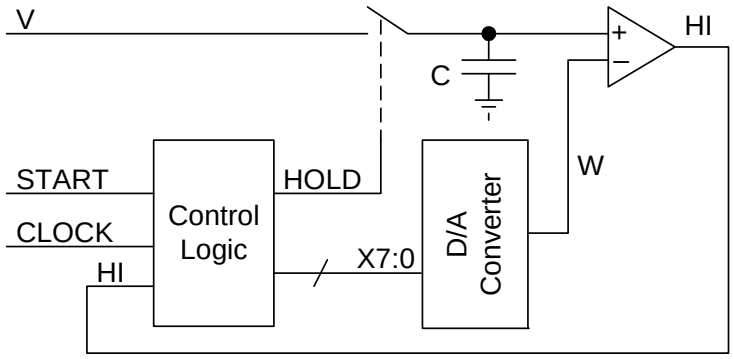


Figure 6

----------Solution ----------

(a) $x = \text{round}(V / 0.2) = \text{round}(-52.25) = -52 = 11001100$

(b) The voltage W must be the lower threshold of the corresponding input interval and is therefore $w = 0.2x - 0.1$. The sequence is as follows:

Clock	Step	X	X (binary)	W
0	128	0	0000 0000	-0.1
1	64	-64	1100 0000	-12.9
2	32	-32	1110 0000	-6.5
3	16	-48	1101 0000	-9.7
4	8	-56	1100 1000	-11.3
5	4	-52	1100 1100	-10.5
6	2	-50	1100 1110	-10.1
7	1	-51	1100 1101	-10.3
8	0.5	-52	1100 1100	-10.5

(c) If we assume that X7:0 is set to 0 before the conversion starts, then we don't have to wait for the 100 ns the first time around. The minimum conversion time is therefore $50 + 60 + 7 \times 160 = 1.23\ \mu s$.

(d) $C = \dfrac{I \times \Delta t}{\Delta V} = \dfrac{2\ \mu A \times (1.23 - 0.05)\ \mu s}{0.1V} = 23.6\ pF$

5. The circuit of *Figure 7* forms part of a logic analyser in which the eight lines DATA7:0 are sampled repetitively and their values stored in a memory. The circuit comprises a static RAM memory, a counter, an 8-bit register and two flipflops. The propagation delays of the counter, register and flipflops may vary between 5 and 10 ns and may have different values for rising and falling output transitions. The flipflops have setup and hold times of 5 ns and 0 ns respectively relative to the clock rising edge. The timing requirements for a memory write cycle are shown in the figure.

a) Draw a timing diagram showing the waveforms of CLOCK, CNT, $\overline{\text{WRITE}}$, A12:0 and D7:0. The signal CLOCK is a symmetrical squarewave of constant frequency. [7]

b) For each of the timing constraints shown in the figure, determine the corresponding restriction on the period of CLOCK. [10]

c) Hence determine the maximum CLOCK frequency for reliable circuit operation. [3]

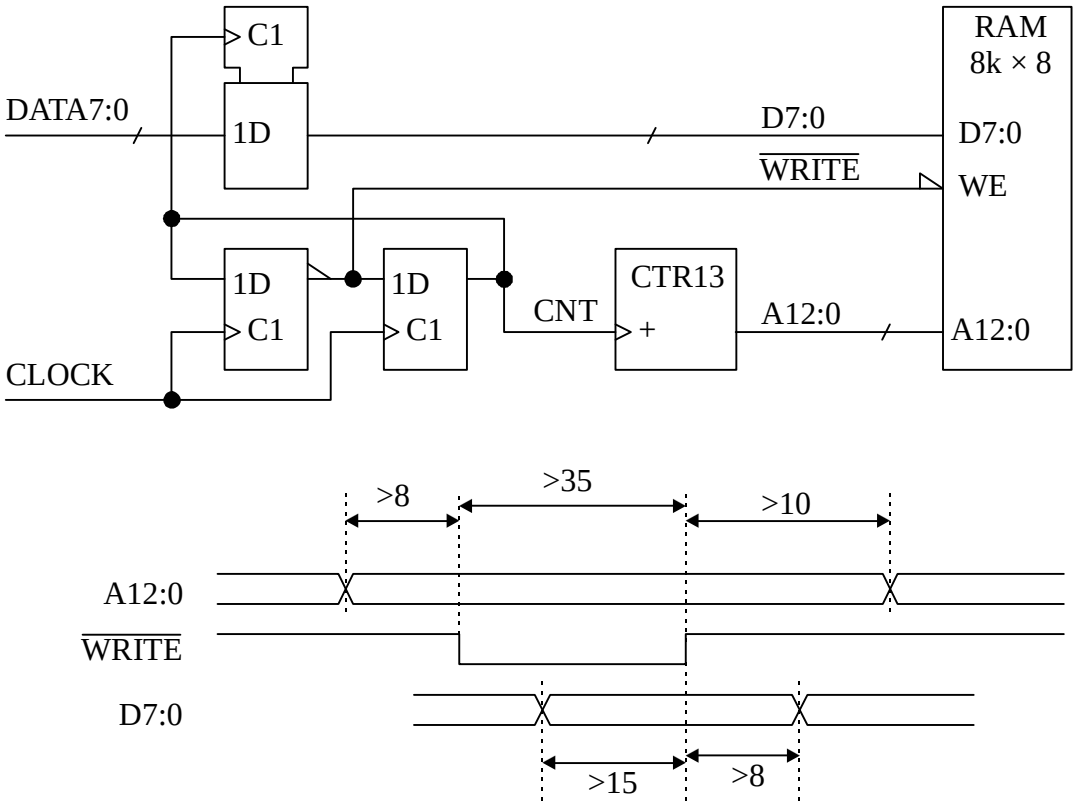
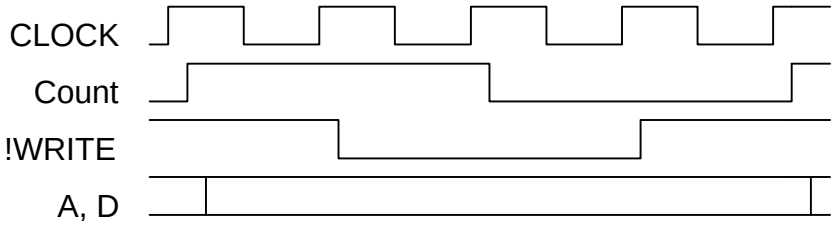


Figure 7

----------Solution ----------

a) The waveforms of the various signals are drawn below with exaggerated propagation delays:



b) We need to check each of the timing requirements in the diagram. In the expressions below, we use T for the clock period and t for the propagation delay of a counter/register/flipflop. All the t values are then replaced by 10 or 5 according to whether they are on the left or right respectively of an < sign.

Addr setup:	$2t+8 < T+t$	$\Rightarrow$	$T > 20 + 8 – 5 = 23$
Write pulse:	$t + 35 < 2T + t$	$\Rightarrow$	$T > (10 + 35 – 5)/2 = 20$
Addr hold:	$t+10 < T+2t$	$\Rightarrow$	$T > 10 + 10 – 10 = 10$
Data setup:	$2t+15 < 3T+t$	$\Rightarrow$	$T > (20 + 15 – 5)/3 = 10$
Data hold:	$t+8 < T+2t$	$\Rightarrow$	$T > 10 + 8 – 10 = 8$

c) The address setup is therefore the limiting factor and the maximum clock frequency is 43.4 MHz.

6. *Figure 1* shows the circuit of a synchronous state machine having a 2-bit input P0:1 and two outputs, X and Y. The logic block implements the following Boolean equations:

$$X = P1 + P0 + (S0 \oplus S1) \qquad D1 = P1 \oplus S0$$
$$Y = P1 \bullet P0 \bullet (S0 \oplus S1) \qquad D0 = P0 \oplus S1$$

(a) Construct the state table for the circuit. [6]

(b) Draw a state diagram for the circuit in which the state is represented by the decimal value of the 2-bit number S1:0. You should simplify your diagram by using appropriate default values for the output signals. [8]

(c) Complete the timing diagram of *Figure 2* by showing the waveforms of the two output signals and the sequence of states followed by the circuit. The circuit is initially in state 0 as shown in the diagram. [6]

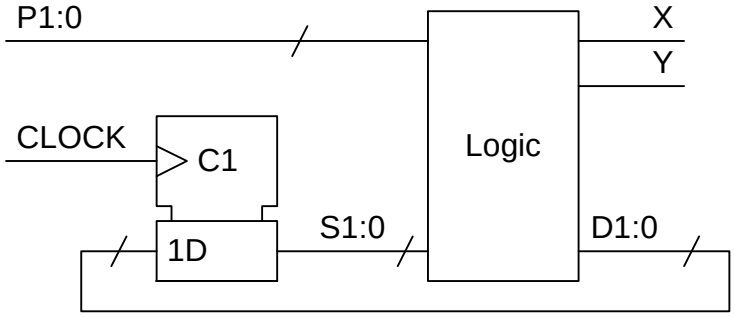


Figure 1

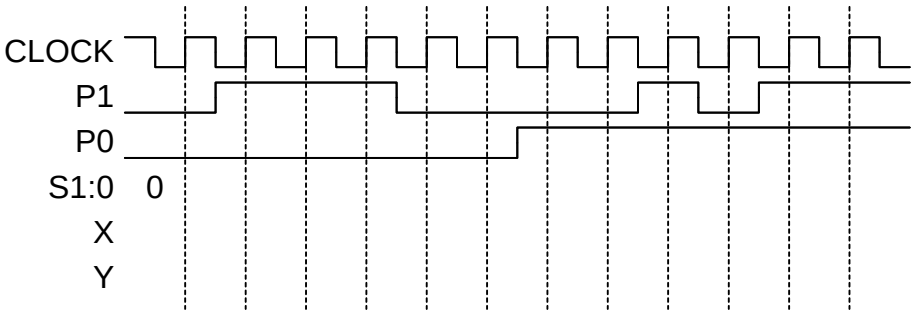


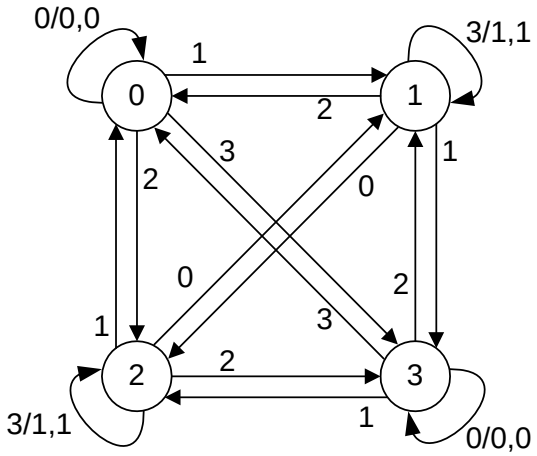
Figure 2

----------Solution ----------

(a) From the Boolean equations we get the following state table:

D1,D0/X,Y	P1,P0			
S1,S0	00	01	11	10
00	00/00	01/10	11/10	10/10
01	10/10	11/10	01/11	00/10
11	11/00	10/10	00/10	01/10
10	01/10	00/10	10/11	11/10

(b) From the state table, we can construct the state diagram. Transitions are labelled with the value of P1:0.



I/O Signals: P1:0/X,Y

Defaults: X=1, Y=0

(c)