Jan. 30, 2010

COMPUTER ENGINEERING DEPARTMENT

COE 561

Digital System Design and Synthesis

Final Exam

(Open Book Exam)

First Semester (091)

Time: 7:00-10:00 PM

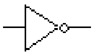
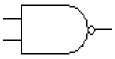
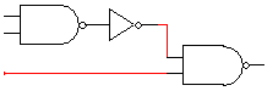
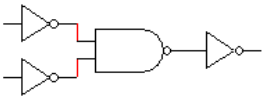
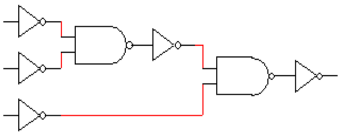
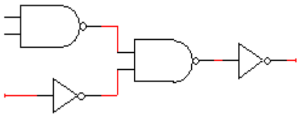
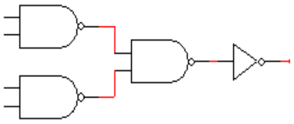
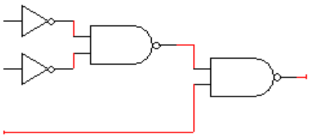
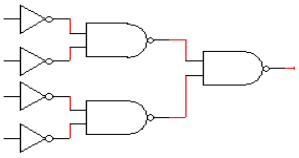
Student Name : ___KEY____

Student ID.   : ____

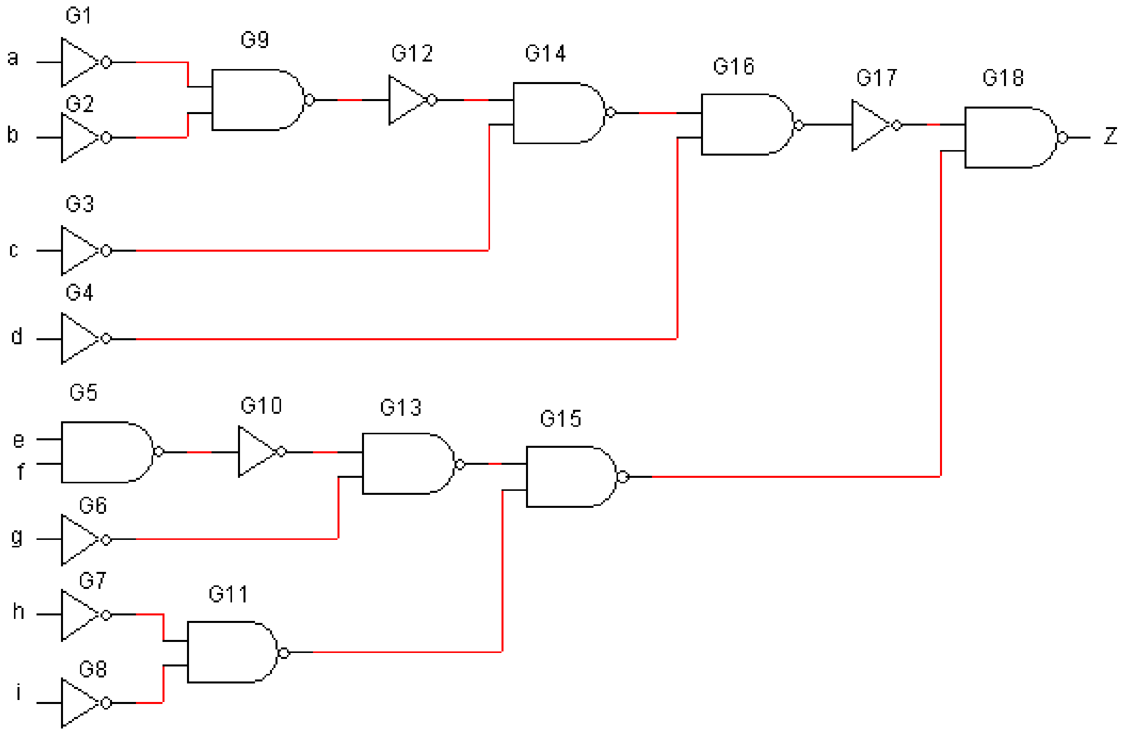
Question	Max Points	Score
Q1	20	
Q2	20	
Q3	23	
Q4	12	
Q5	25	
Total	100	

[20 Points]

(Q1) Consider a technology library containing the following cells:

Cell	Area Cost	Gate
INV(x1) = x1’	1	
NAND2(x1, x2) = (x1 x2)’	2	
NAND3(x1, x2, x3) = (x1 x2 x3)’	3	
NOR2(x1, x2) = (x1 + x2)’	2	
NOR3(x1, x2, x3) = (x1 + x2 + x3)’	3	
AOI21(x1, x2, x3) = ((x1 x2) + x3)’	3	
AOI22(x1, x2, x3, x4) = ((x1 x2) + (x3 x4))’	4	
OAI21(x1, x2, x3) = ((x1+x2) x3)’	3	
OAI22(x1, x2, x3, x4) = ((x1+x2) (x3+x4))’	4	

(i) Consider the circuit given below with inputs *{a, b, c, d, e, f, g, h, i}* and output {*Z*}. Using the dynamic programming approach and **Structural Matching**, **map** the circuit using the given library into the **minimum area** cost solution.



(ii) Assuming **Boolean Matching**, determine the number of ROBDD's that need to be stored in the cell library for the following cell. Justify your answer.

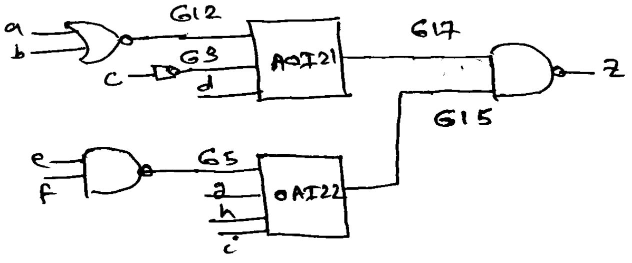
Y = a b c + d e + f g + f' g'

(i)

vertex	gate	cost
G1	INV(a)	1
G2	INV (b)	1
G3	INV(c)	1
G4	INV(d)	1
G5	Nand 2 (e,f)	2
G6	INV(g)	1
G7	INV(h)	1
G8	INV(i)	1
G9	Nand2 (G1,G2)	2+1+1=4
G10	INV (G5)	1+2=3
G11	Nand2 (G7,G8)	2+1+1=4
G12	INV (G9) NOR2 (a,b)	1+4=5 $\underline{\underline{2}}$
G13	Nand2 (G10, G6) Nand3 (e,f, G6)	2+3+1=6 3+1= $\underline{\underline{4}}$
G14	Nand2 (G12, G3) Nand3 (G1, G2, G3)	2+2+1= $\underline{\underline{5}}$ 3+1+1+1=6
G15	Nand2 (G13, G11) OAI21 (G5, g, G11) OAI21 (G13, h, i) OAI22 (G5, g, h, i)	2+4+4=10 3+2+4=9 3+4=7 4+2= $\underline{\underline{6}}$

vertex	gate	cost
G16	Nand2 (G14, G15) OAI21 (G9, c, G4)	$2 + 5 + 6 = 13$ $3 + 4 + 1 = \underline{\underline{8}}$
G17	INV (G16) AOI21 (G12, G3, d)	$1 + 8 = 9$ $3 + 2 + 1 = \underline{\underline{6}}$
G18	Nand2 (G17, G15) Nand3 (G14, G3, G15)	$2 + 6 + 6 = \underline{\underline{14}}$ $3 + 5 + 1 + 6 = 15$

Thus, the minimum area cost solution is 14 as shown below:



(ii) $c2 = \{ (d,e), (f,g) \}$

$c3 = \{ (a,b,c) \}$

Number of ROBDD's required is 1 since (d,e) are unate while (f,g) are binate.

[20 Points]

(Q2) Consider the incompletely-specified FSM that has 5 states, one input (X) and one output (Z), represented by the following state table:

Present State	Next State, Z	
	X=0	**X=1**
S0	S0, 0	S1, 1
S1	S0, 1	S3, 0
S2	S4, 0	S3, 1
S3	S2, 1	S3, –
S4	S2, –	S1, 1

(i) Determine the incompatible states and the compatible states along with their implied pairs.

(ii) Compute the maximal compatible classes along with their implied state pairs.

(iii) Reduce the state table into the minimum number of states and show the reduced state table.

(i) Compatibility Table :

S1	☒			
S2	(S0, S4) (S1, S3)	☒		
S3	☒	(S0, S2)	☒	
S4	(S0, S2)	☒	(S1, S3)	(S1, S3)
	S0	S1	S2	S3

The incompatible states are :

(S0, S1) , (S0, S3), (S1, S2) , (S1, S4) , (S2, S3)

The compatible states with their implied pairs:

(S0 , S2) ⇐ (S0, S4) and (S1, S3)

(S1 , S3) ⇐ (S0, S2)

(S2, S4) ⇐ (S1, S3)

(S3, S4) ⇐ (S1, S3)

(S0, S4) ⇐ (S0, S2)

(ii) Maximal compatible classes:

From the incompatible state pairs we have:

$$(\overline{S_0}+\overline{S_1})(\overline{S_0}+\overline{S_3})(\overline{S_1}+\overline{S_2})(\overline{S_1}+\overline{S_4})(\overline{S_2}+\overline{S_3})$$

$$=(\overline{S_0}+\overline{S_1}\,\overline{S_3})(\overline{S_1}+\overline{S_2}\,\overline{S_4})(\overline{S_2}+\overline{S_3})$$

$$=(\overline{S_0}\,\overline{S_1}+\overline{S_0}\,\overline{S_2}\,\overline{S_4}+\overline{S_1}\,\overline{S_3})(\overline{S_2}+\overline{S_3})$$

$$=\overline{S_0}\,\overline{S_1}\,\overline{S_2}+\overline{S_0}\,\overline{S_1}\,\overline{S_3}+\overline{S_0}\,\overline{S_2}\,\overline{S_4}+\overline{S_1}\,\overline{S_2}\,\overline{S_3}+\overline{S_1}\,\overline{S_3}$$

$$=\overline{S_0}\,\overline{S_1}\,\overline{S_2}+\overline{S_0}\,\overline{S_2}\,\overline{S_4}+\overline{S_1}\,\overline{S_3}$$

Thus, the maximal compatible classes with their implied pairs are:

$(S3, S4) \Leftarrow (S1, S3)$
$(S1, S3) \Leftarrow (S0, S2)$
$(S0, S2, S4) \Leftarrow (S1, S3)$

(iii) The following cover can be used which satisfies the closure property:
$\{(S0, S2, S4), (S1, S3)\}$

Thus, the state machine can be reduced to two states as follows:

P.S.	N.S., Z	
	x = 0	x = 1
S024	S024, 0	S13, 1
S13	S024, 1	S13, 0

[23 Points]

(Q3) Consider the incompletely-specified FSM that has 5 states, two inputs and one output, represented by the following state table:

Product	Input	Present State	Next State	Output
P1	00	S0	S0	0
P2	01	S0	S0	0
P3	10	S0	S3	0
P4	11	S0	S3	0
P5	00	S1	S0	0
P6	01	S1	S4	0
P7	10	S1	S1	0
P8	11	S1	S1	0
P9	00	S2	S3	0
P10	01	S2	S3	0
P11	10	S2	S1	0
P12	11	S2	S4	0
P13	00	S3	S3	0
P14	01	S3	S3	0
P15	10	S3	S3	0
P16	11	S3	S2	1
P17	00	S4	S4	0
P18	01	S4	S0	0
P19	10	S4	S4	0
P20	11	S4	S1	0

(i) Assuming the following constraints, S4 covers S0, S4 covers S1 and that the code of S3 is covered by all other state codes, the state table can be reduced into the table given below. Using implicant merging and covering relations show step by step how you can obtain the reduced state stable given below.

Input	Present State	Next State	Output
0–	S0, S1, S4	S0	0
1–	S1, S2, S4	S1	0
–0	S4	S4	0
01	S1	S4	0
11	S2	S4	0
11	S3	S2	1

(ii) Compute all the prime dichotomies. Find a state encoding satisfying the given constraints with minimal bit length.

(i) P1 and P5 can be merged using implicant merging into the following row:

r1 00 S0, S1 S0 0

P2 and P18 can be meged using implicant merging into the following row:

r2 01 S0, S4 S0 0

Since in P17, 00 s4 s4 0, and s4 covers
s0, we can add s4 to r1 resulting in:

r3 00 s0, s1, s4 s0 0

Since in P6, 01 s1 s4 0, and s4 covers
s0, we can add s1 to r2 resulting in:

r4 01 s0, s1, s4 s0 0

Next, r3 and r4 can be merged into the follow.
row:

r5 0– s0, s1, s4 s0 0

P7 and P11 can be merged into the following
row:

r6 10 s1, s2 s1 0

P8 and P20 can be merged into the following
row:

r7 11 s1, s4 s1 0

Since in P19, 10 s4 s4 0, and s4 covers
s1, we can add s4 to r6 resulting in:

r8 10 s1, s2, s4 s1 0

Since in P12, 11 s2 s4 0, and s4 covers
s1, we can add s2 to r7 resulting in:

r9 11 s1, s2, s4 s1 0

Next, r8 and r9 can be merged into the
following row:

r10 1– s1, s2, s4 s1 0

P17 and P18 can be merged into the
following row:

r11 –0 s4 s4 0

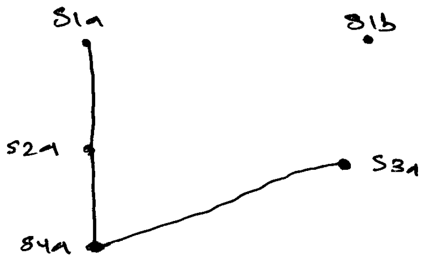
The rows P3, P4, P9, P10, P13, P14 and P15 can all be removed since S3 is covered by all other states and will be assigned the all 0's code.
This results in the given reduced table.

(ii)

Seed Dichotomies

S1a : (S0, S1, S4), (S2)   S2a : (S0, S1, S4), (S3)
S1b : (S2), (S0, S1, S4)   S2b : (S3), (S0, S1, S4) X

S3a : (S1, S2, S4), (S0)   S4a : (S1, S2, S4), (S3)
S3b : (S0), (S1, S2, S4) X   S4b : (S3), (S1, S2, S4) X



Prime dichotomies are:

P1 : (S0, S1, S4), (S2, S3)
P2 : (S0, S1, S2, S4), (S3)
P3 : (S1, S2, S4), (S0, S3)

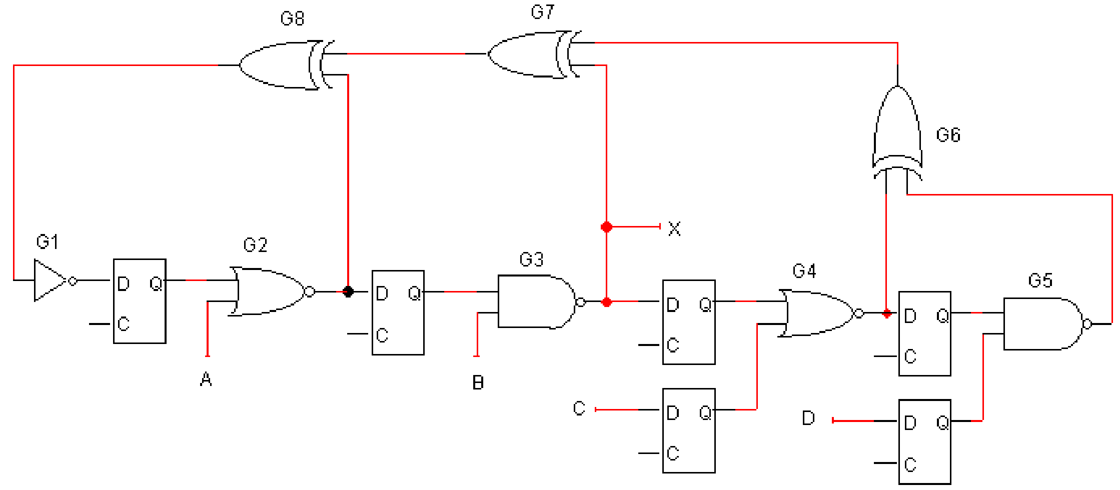
The prime dichotomies P1 and P3 cover all the seed dichotomies and are selected.

Thus, we obtain the following encoding:

State	code
S0	1 0 0
S1	1 1 0
S2	0 1 0
S3	0 0 0
S4	1 1 1

[12 Points]

(Q4) Consider the sequential circuit given below having 4 inputs {A, B, C, D} and one output {X}. Assume that the delay of an inverter is 1 unit delay, the delay of a 2-input NAND gate is 2 unit delays, the delay of a 2-input NOR gate is 2 unit delays and the delay of a 2-input XOR gate is 3 unit delays.



(i) Determine the critical path of this circuit and the maximum propagation delay.

(ii) Using only the **Retiming** transformation, reduce the critical path of this circuit with the minimum number of flip-flops possible. Determine the maximum propagation delay after retiming.

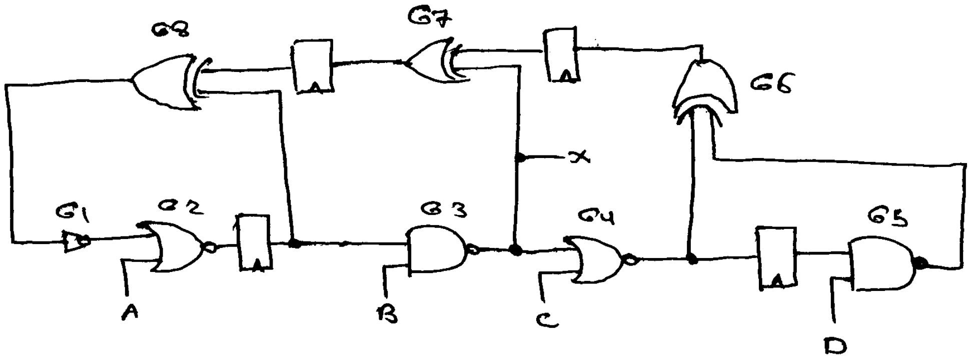
(i) The maximum propagation delay is 12 and there are two critical paths as follows;

{G5, G6, G7, G8, G1}, {G4, G6, G7, G8, G1}

(ii) We can apply the following retiming transformations to reduce the critical path:

- retime G5 by −1
- retime G4 by −1
- retime the stem on fanout of G4 by −1
- retime G6 by −1
- retime G1 by +1
- retime G8 by +1
- retime the stem on fanout of G2 by +1

This results in the following circuit after retiming:



The maximum propagation delay in the resulting circuit is 7 across the path {G3, G4, G5}. The number of registers has also been reduced from 6 to 4 registers.

[25 Points]

(Q5) Consider the network given below with inputs *{a, b, c, d}* and output *{y}*:

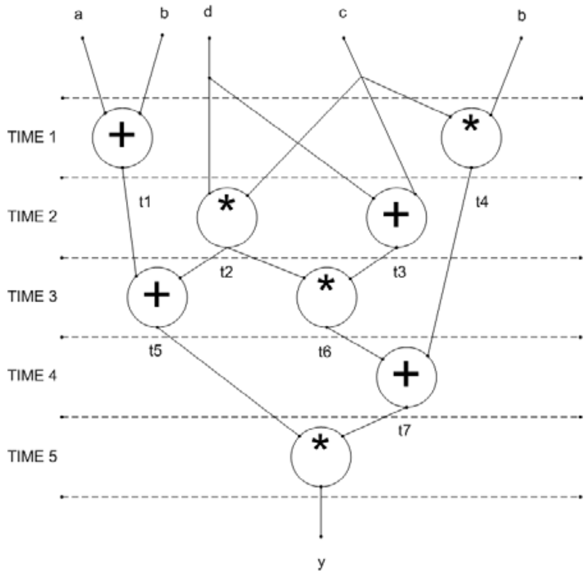
[1]=a+b;	[2]=c*d;	[3]=c+d;	[4]=b*c;
[5]=[1]+[2];	[6]=[2]*[3];	[7]=[4]+[6];	y=[5]*[7];

(i) Draw the **sequencing graph** for the above network.

(ii) Assuming that the delay of an Adder fits within one clock cycle and the delay of a Multiplier fits within two clock cycles, show the **ASAP** and **ALAP** scheduling of the sequencing graph. Compute the **mobility** of each operation.

(iii) Using **List Scheduling** algorithm LIST_L, schedule the sequencing graph into the **minimum number of cycles** under the resource constraints of one Adder and one multiplier. Assume that the delay of an Adder fits within one clock cycle and the delay of a Multiplier fits within two clock cycles. Show the details of the algorithm step by step and the resulting scheduled sequencing graph.

(iv) Consider the scheduled sequencing graph below assuming that the delay of an Adder fits within one clock cycle and the delay of a Multiplier fits within one clock cycle. Assume that the input values will be available to the circuit for only one clock cycle.

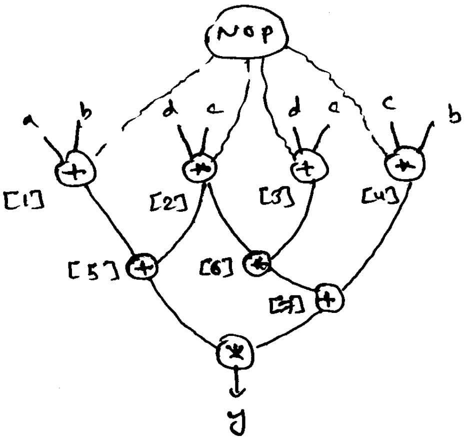


a. Show the life-time of all variables.

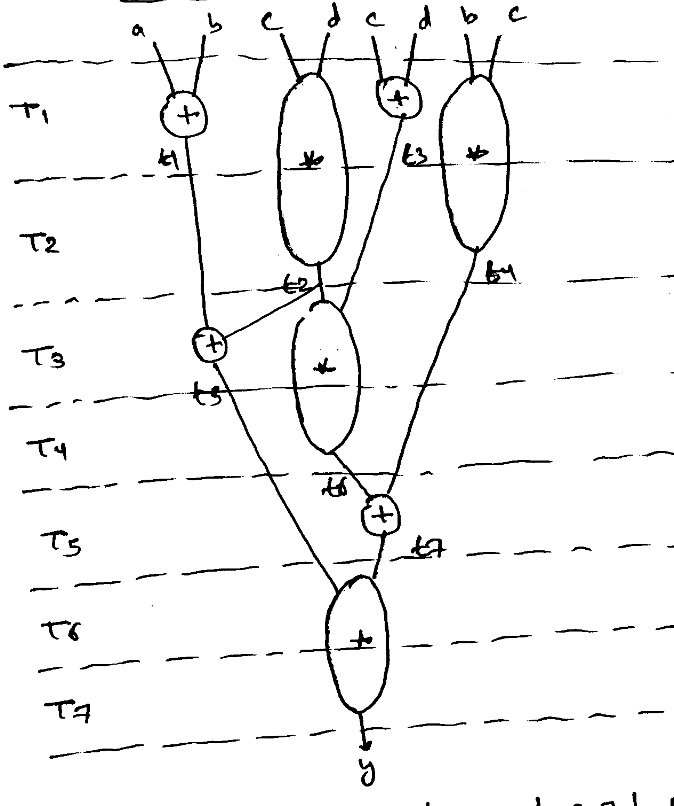
b. Determine the minimum number of registers that are required to store all the variables. Show the mapping of variables to registers. Select a mapping that **minimizes the number of multiplexers and interconnect area** as much as possible.

c. Draw the **data-path** implementing the scheduled sequencing graph based on the variable-register mapping that you obtained in (b).

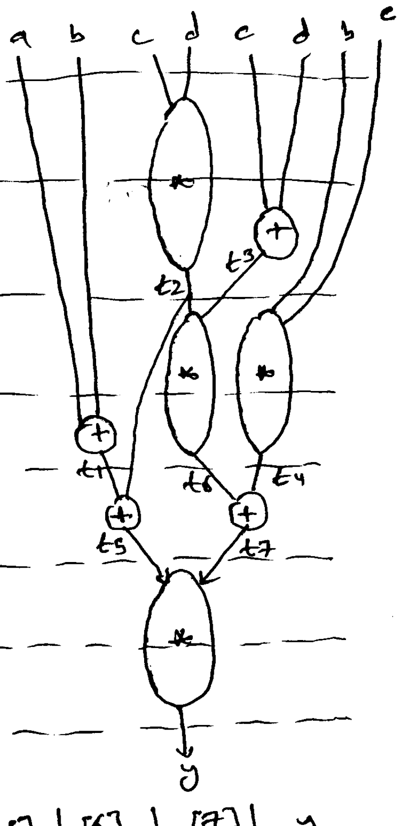
(i) Sequencing Graph:



(ii) ASAP Scheduling



ALAP scheduling



Operation	[1]	[2]	[3]	[4]	[5]	[6]	[7]	y
Mobility	3	0	1	2	2	0	0	0

(iii) List Scheduling

$l=1$

$U_{1,add} = \{ [1], [3] \}$

we schedule [3] since it has lower slack

$U_{1,mul} = \{ [2], [4] \}$

we schedule [2] since it has lower slack

$l=2$

$U_{2,add} = \{ [1] \}$

we schedule [1].

$U_{2,mul} = [4]$

since $T_{2,mul} = \{[2]\}$

[4] is not scheduled.

$l=3$

$U_{3,add} = \{ [5] \}$

we schedule [5].

$U_{3,mul} = \{ [4], [6] \}$

we schedule [6] since it has lower slack

$l=4$

$U_{4,add} = \{ \}$

$U_{4,mul} = \{ [4] \}$

since $T_{4,mul} = \{ [6] \}$

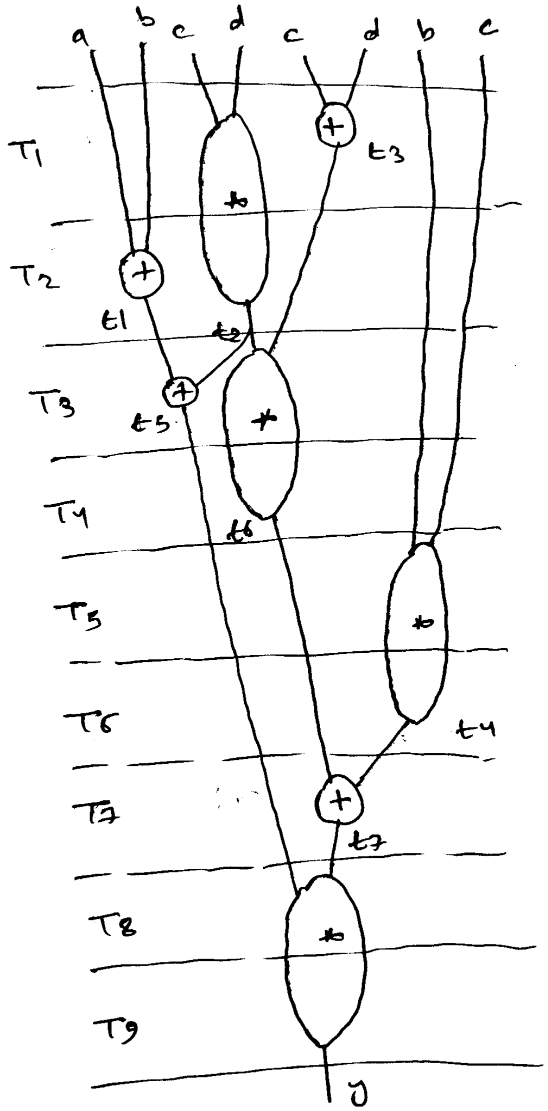
[4] is not scheduled.

$l=5$

$U_{5,add} = \{ \}$

$U_{5,mul} = \{ [4] \}$

[4] is scheduled.



$l = 6$

$U_{6,add} = \{\ \}$
$U_{6,mul} = \{\ \}$

$l = 7$

$U_{7,add} = \{[7]\}$
[7] is scheduled
$U_{7,mul} = \{\ \}$

$l = 8$

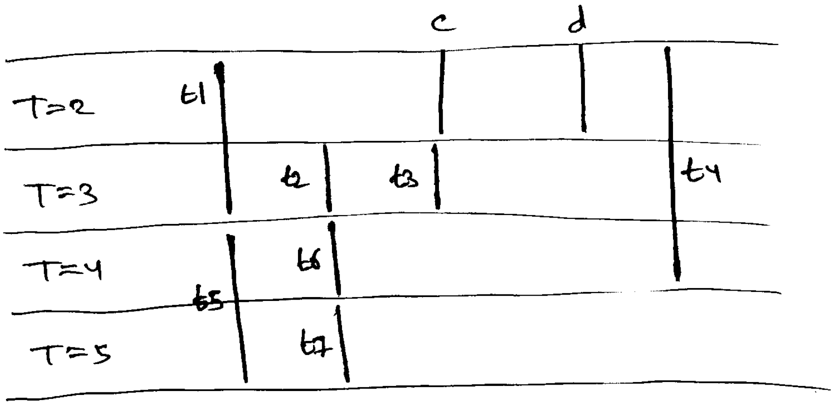
$U_{8,add} = \{\ \}$
$U_{8,mul} = \{4\}$
4 is scheduled

$l = 9$

$U_{9,add} = \{\ \}$
$U_{9,mul} = \{\ \}$

Thus, with one adder and one multiplier the minimum schedule is 9 clock cycles.

(iv) a. Life time of variables :



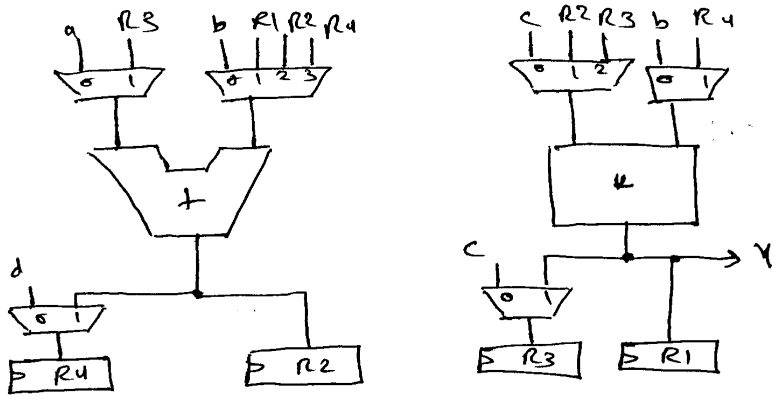
b. Based on the lifetime of all variables, it is obvious that 4 registers are needed to store all the variables.

Adder	Mult.
$a + b = t_1$	$c * b = t_4$
$d + c = t_3$	$c * d = t_2$
$t_1 + t_2 = t_5$	$t_2 * t_3 = t_6$
$t_4 + t_6 = t_7$	$t_5 * t_7 = y$

To minimize the number of mux inputs and wiring, we make the following assignment of variables to registers:

R1 : $\{t_4\}$ R2 : $\{t_1, t_5\}$ R3 : $\{c, t_2, t_6\}$ R4 : $\{d, t_3, t_7\}$

c. Data Path


a
R3
b
R1
R2
R4
0
1
0
1
2
3
+
d
0
1
R4
R2
c
R2
R3
b
R4
0
1
2
0
1
*
c
Y
0
1
R3
R1