Jan. 8, 2012

COMPUTER ENGINEERING DEPARTMENT

COE 561

Digital System Design and Synthesis

Final Exam

(Open Book Exam)

First Semester (111)

Time: 7:00-10:00 PM

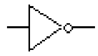
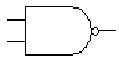
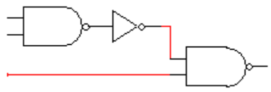
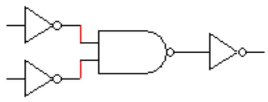
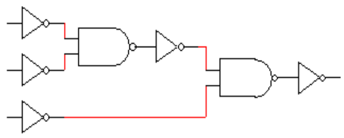
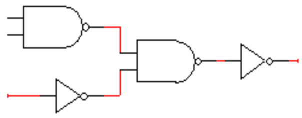
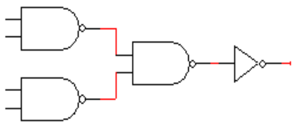
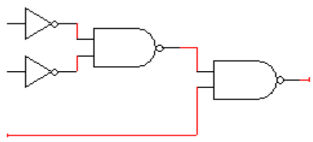
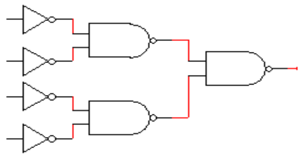
Student Name : _KEY____

Student ID.   : ____

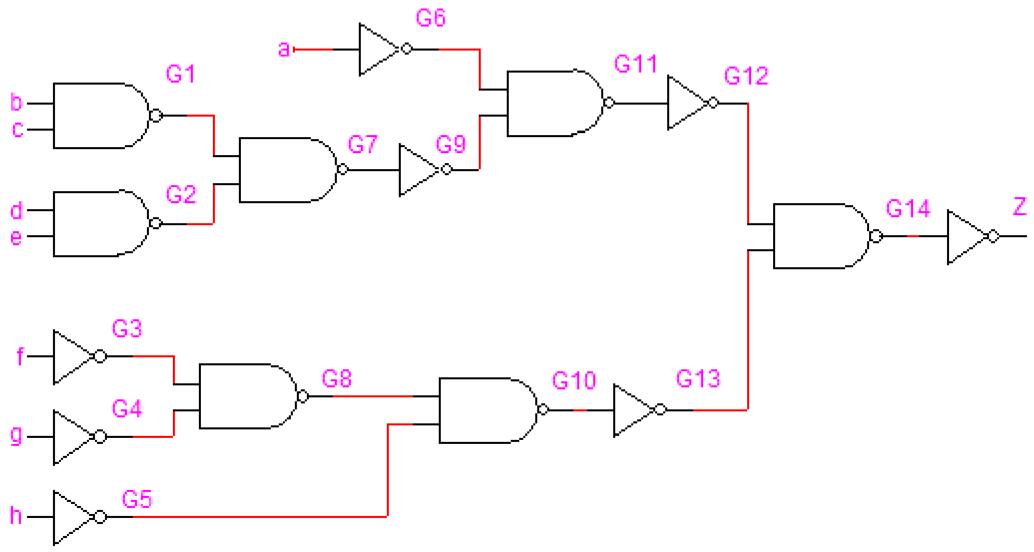
Question	Max Points	Score
Q1	20	
Q2	22	
Q3	21	
Q4	12	
Q5	25	
Total	100	

[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}* 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) Can you obtain a better mapping than the one obtained in (i). If the answer is yes, show the better solution and explain how it is obtained.

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

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

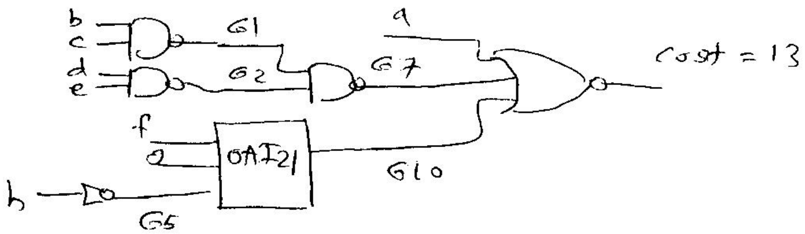
(i)

vertex	gate	cost
G1	Nand2 (b, c)	2
G2	Nand2 (d, e)	2
G3	INV (f)	1
G4	INV (g)	1
G5	INV (h)	1
G6	INV (a)	1
G7	Nand2 (G1, G2)	2 + 2 + 2 = 6
G8	Nand2 (G3, G4)	2 + 1 + 1 = 4

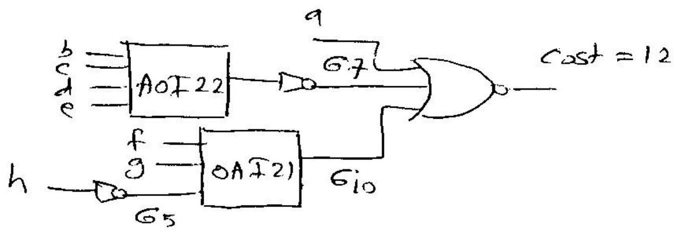
vertex	gate	cost
G9	INV(G7)	1 + 6 = 7
	AOI22(b, c, d, e)	4
G10	Nand2 (G8, G5)	2 + 4 + 1 = 7
	OAI21 (f, g, G5)	3 + 1 = 4
G11	Nand2 (G6, G9)	2 + 1 + 4 = 7
	Nand3 (G6, G1, G2)	3 + 1 + 2 + 2 = 8
G12	INV (G11)	1 + 7 = 8
	NOR2 (a, G7)	2 + 6 = 8
G13	INV (G10)	1 + 4 = 5
	AOI21 (G3, G4, h)	3 + 1 + 1 = 5
G14	Nand2 (G12, G13)	2 + 8 + 5 = 15
	Nand3 (G6, G9, G13)	3 + 1 + 4 + 5 = 13
	Nand3 (G12, G8, G5)	3 + 8 + 4 + 1 = 16
Z	INV (G14)	1 + 13 = 14
	NOR2 (G11, G10)	2 + 7 + 4 = 13
	NOR3 (G10, a, G7)	3 + 4 + 6 = 13

Thus, there are several solutions with minimum area cost of 13.

One minimal area solution is :



(ii) Yes, a better solution with a cost of 12 is as follows:



This solution is obtained by adding a cascade of inverters at the output of G7.

(iii) $C_1 = \{ \overset{B}{(e)}, \overset{U}{(f)}, \overset{U}{(g)} \}$

$C_2 = \{ \overset{B}{(a,b)}, \overset{U}{(c,d)} \}$

Thus, number of ROBDD's needed is $2! = 2$

[22 Points]

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

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

(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) Compute the prime compatibility classes along with their implied state pairs.

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

(i) Compatibility Table:

	S0	S1	S2	S3
S1	S3, S4			
S2	S0, S1	S3, S4 S0, S1		
S3	S2, S3 S0, S2	~~S2, S4~~ ~~S0, S2~~	S1, S2	
S4	S2, S3 S0, S1	X	X	S1, S2

Thus, the incompatible states are:

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

The compatible states with their implied pairs are:

$(S_0, S_1) \Leftarrow (S_3, S_4)$

$(S_0, S_2) \Leftarrow (S_0, S_1)$

$(S_0, S_3) \Leftarrow (S_2, S_3)$ and (S_0, S_2)

$(S_0, S_4) \Leftarrow (S_2, S_3)$ and (S_0, S_1)

$(S_1, S_2) \Leftarrow (S_3, S_4)$ and (S_0, S_1)

$(S_2, S_3) \Leftarrow (S_1, S_2)$

$(S_3, S_4) \Leftarrow (S_1, S_2)$

(ii) <u>Maximal Compatible Classes:</u>

From the incompatible state pairs, we have:

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

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

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

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

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

$(S_0, S_3, S_4) \Leftarrow (S_0, S_2), (S_2, S_3), (S_0, S_1), (S_1, S_2)$

$(S_0, S_2, S_3) \Leftarrow (S_0, S_1), (S_1, S_2)$

$(S_0, S_1, S_2) \Leftarrow (S_3, S_4)$

(iii) Prime Compatibility Classes

In addition to the maximal compatible classes, we have the following prime classes:

$(S_0, S_3) \Leftarrow (S_2, S_3), (S_0, S_2)$

$(S_0, S_4) \Leftarrow (S_2, S_3), (S_0, S_1)$

$(S_3, S_4) \Leftarrow (S_1, S_2)$

$(S_0, S_2) \Leftarrow (S_0, S_1)$

$(S_2, S_3) \Leftarrow (S_1, S_2)$

(S_0)
(S_1)
(S_2)
(S_3)
(S_4)

(iv) The following minimum cover can be used which satisfies the closure:

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

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

P.S.	Next State, Z $x = 0$	$x = 1$
$S_{0,1,2}$	$S_{3,4}$, 0	$S_{0,1,2}$, –
$S_{3,4}$	$S_{0,1,2}$, 1	$S_{0,1,2}$, –

[21 Points]

(Q3) Consider the given FSM which has 4 states, one input and one output, represented by the following state table:

Product	Input	Present State	Next State	Output
P1	0	S1	S2	0
P2	1	S1	S2	0
P3	0	S2	S2	0
P4	1	S2	S3	0
P5	0	S3	S4	0
P6	1	S3	S3	0
P7	0	S4	S4	0
P8	1	S4	S1	1

(i) Assuming the following constraints: S3 covers S2, and that the code of S4 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
–	S1, S2	S2	0
1	S2, S3	S3	0
1	S4	S1	1

(ii) Compute all the seed dichotomies and construct their compatibility graph. Find a minimum cover for the seed dichotomies. Based on the found cover, derive an encoding satisfying the given constraints with minimal bit length.

(i) P1 and P3 can be merged using implicant merging into the following row:

r1 0 S1, S2 S2 0

P4 and P6 can be merged using implicant merging into the following row:

r2 1 S2, S3 S3 0

Since in P4, we have 1 S2 S3 0, and since S3 covers S2, we can add S2 to P2 resulting in:

r3 1 S1, S2 S2 0

Next, r1 and r3 can be merged using implicant merging into the following row:

r4 — S1, S2 S2 0

P5 and P7 can be removed since S4 is covered by all states i.e. its code will be all 0's and since the output is 0.

This results in the given reduced table composed of r4, r2 and P8.

(ii) Seed Dichotomies

S1a: (S1, S2), (S3) ✗

S1b: (S3), (S1, S2) ✓

S2a: (S1, S2), (S4) ✓

S2b: (S4), (S1, S2) ✗

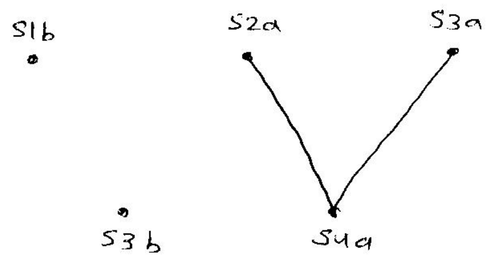
S3a: (S2, S3), (S1) ✓

S3b: (S1), (S2, S3) ✓

S4a: (S2, S3), (S4) ✓

S4b: (S4), (S2, S3) ✗

Seed dichotomy Compatibility Graph



Based on the compatibility graph, a minimum cover of 3 prime dichotomies is needed as follows:

P1: s1b: (S3), (S1, S2)

P2: (S1, S2, S3), (S4)

P3: (S2, S3), (S1, S4)

Thus, a minimum of 3 bits are needed as follows:

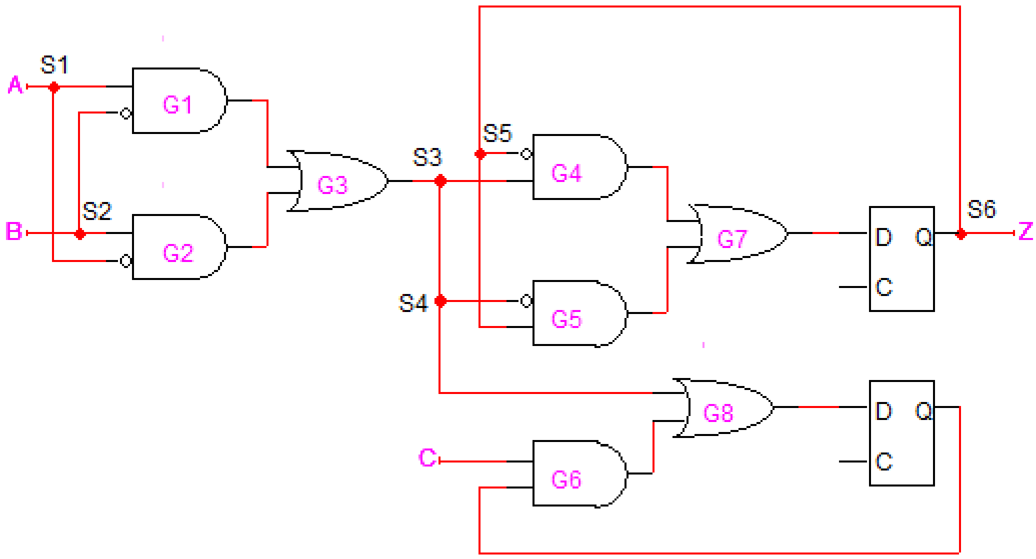
	P1	P2	P3
S1	0	1	0
S2	0	1	1
S3	1	1	1
S4	0	0	0

Note that for P1 we have to assign S4 to 0 to satisfy the covering constraints.

Note that all the encoding and covering constraints are satisfied by the derived encoding.

[12 Points]

(Q4) Consider the sequential circuit given below having 3 inputs {A, B, C} and one output {Z}. Assume that the delay of all given gates is 2 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.

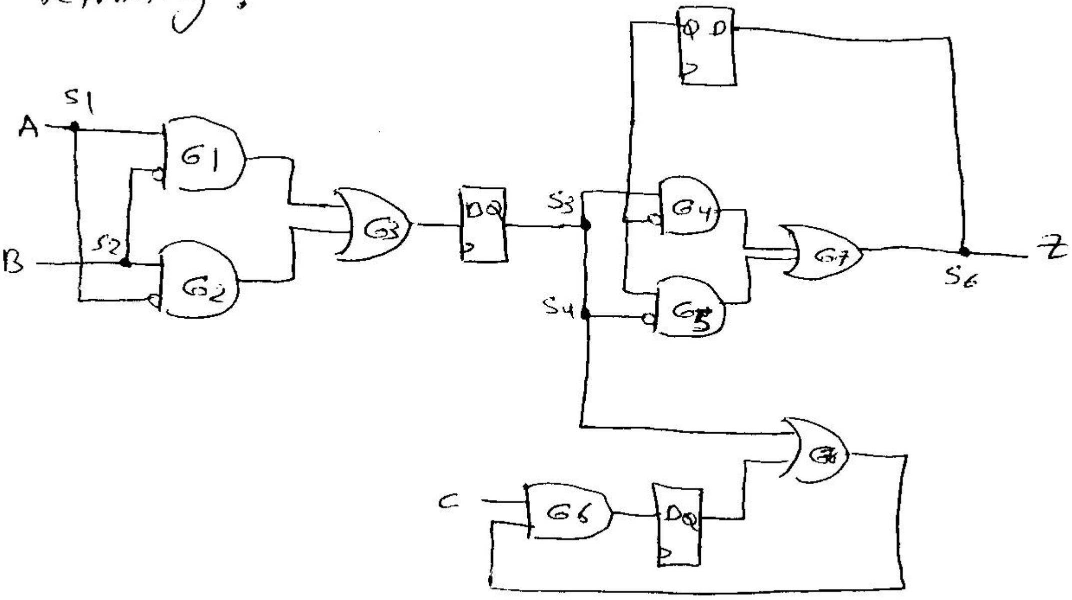
(i) The maximum propagation delay is 8
There are 8 critical paths from either A or B through either gates G1 or G2, through G3, then through either gates G4 or G5, through G7.

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

- retime G7 by +1
- retime G4 by +1
- retime G5 by +1
- retime S5 by +1

- retime G8 by +1
- retime S4 by +1
- retime S3 by +1

This results in the following circuit after retiming:



The maximum propagation delay in the resulting circuit is 4. The number of flip-flops has increased from 2 to 3.

[25 Points]

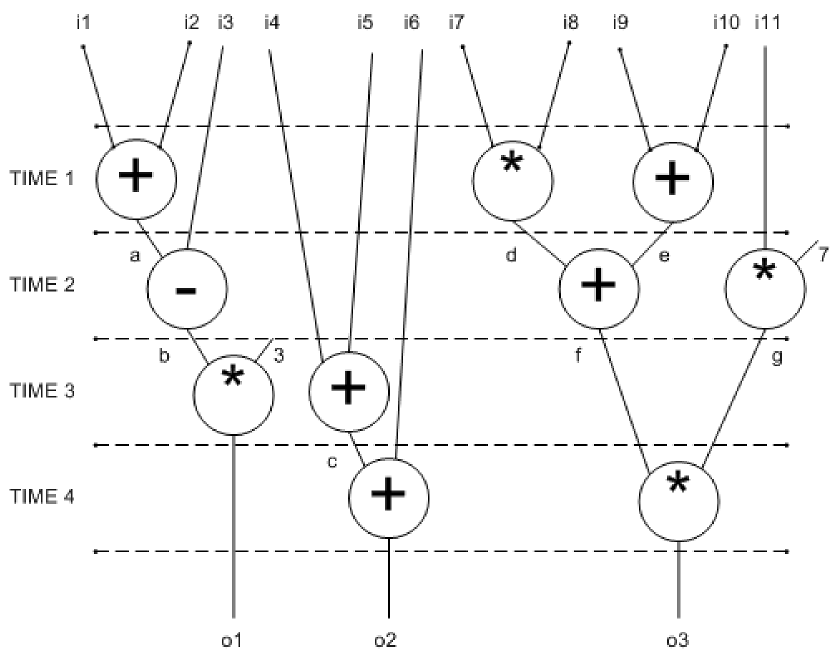
(Q5) Consider the network given below with inputs *{i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, i11}* and outputs *{o1, o2, o3}*. Assume that the delay of both the Adder and the Multiplier fit within one clock cycle and that the input values will be available to the circuit for only one clock cycle. Also assume that both addition and subtraction operations will be performed by the Adder.

a = i1 + i2;	b = a – i3;	c = i4 + i5;	d = i7 * i8;	e = i9 + i10;
f = d + e;	g = i11 * 7;	o1= b * 3;	o2 = c + i6;	o3= g * f;

(i) 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. Show the details of the algorithm step by step and the resulting scheduled sequencing graph.

(ii) Using **List Scheduling** for minimum resource usage algorithm LIST_R, schedule the sequencing graph under the latency constraint of **5 clock cycles** minimizing the number of resources required. Show the details of the algorithm step by step and the resulting scheduled sequencing graph.

(iii) Consider the scheduled sequencing graph below:



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) List – L:

$l = 1$:

$U_{1,add} = \{a, c, e\}$. We need to schedule either a or e since they have longer path to sink. Let's schedule a.

$U_{1,mul} = \{d, g\}$. We schedule d since it has longer path to sink.

$l = 2$:

$U_{2,add} = \{b, c, e\}$. We schedule e since it has longer path to sink.

$U_{2,mul} = \{g\}$

$l = 3$:

$U_{3,add} = \{b, c, f\}$. Any of them can be scheduled as they have the same path to sink. Let's schedule b.

$U_{3,mul} = \{\}$

$l = 4$:

$U_{4,add} = \{c, f\}$. Any can be scheduled. Let's schedule c.

$U_{4,mul} = \{o1\}$

$l = 5$:

$U_{5,add} = \{o2, f\}$. Schedule f since it has longer path to sink.

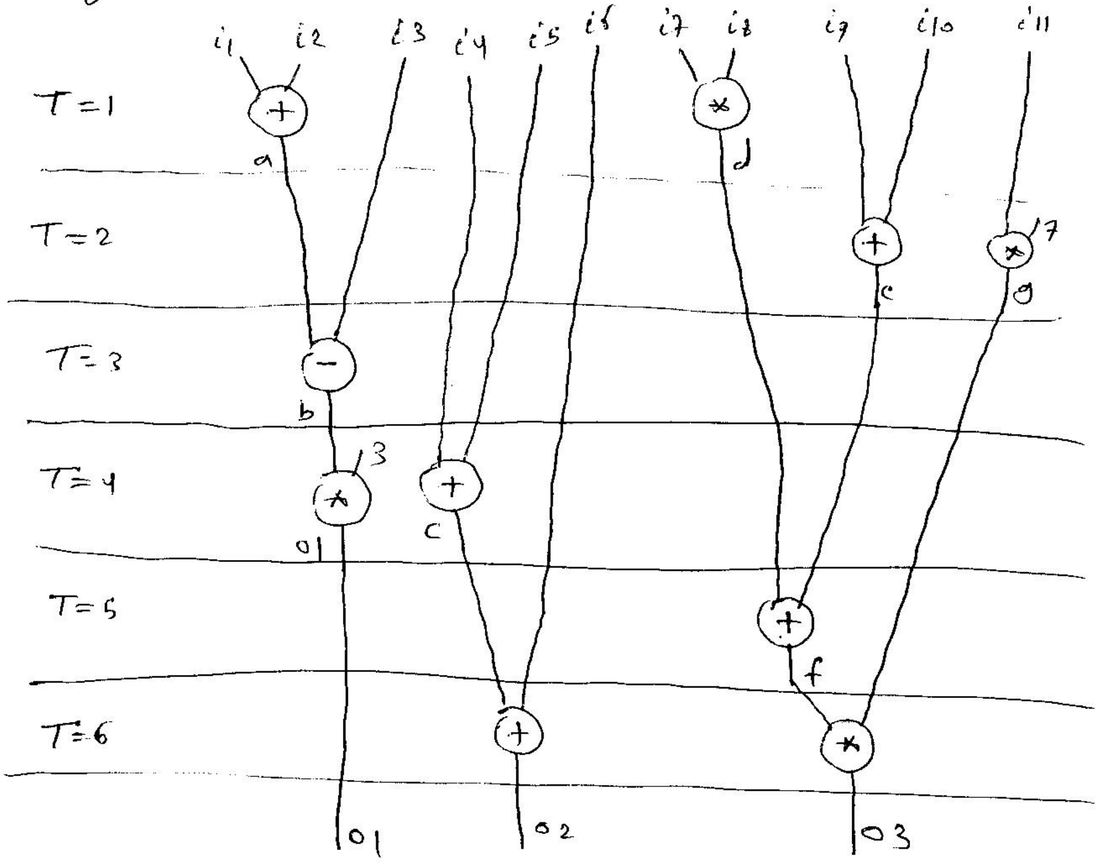
$U_{5,mul} = \{\}$

$l = 6$:

$U_{6,add} = \{o2\}$

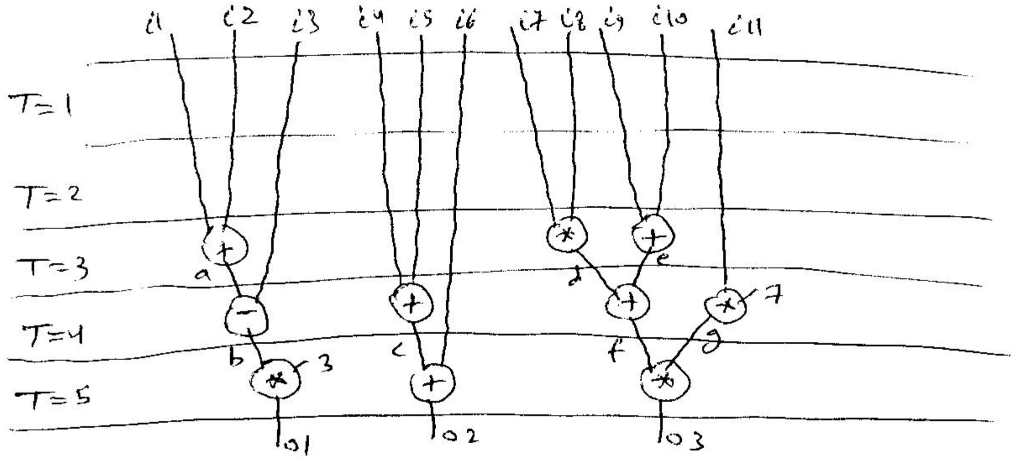
$U_{6,mul} = \{o3\}$

Thus, the minimum required schedule is 6 clock cycles as follows:



(ii) List-R:

$\lambda = 5$, $a = [1, 1]$.
We need to compute the ALAP schedule with $\lambda = 5$ as follows:



$\underline{l=1}$:

$U_{1,add} = \{a, c, e\}$. None of the operations has 0 slack. We schedule one operation with the lowest slack. Schedule a.

$U_{1,mul} = \{d, g\}$. None of the operations has 0 slack. We schedule one operation with the lowes slack. We schedule d.

$\underline{l=2}$:

$U_{2,add} = \{b, c, e\}$. None of the operations has 0 slack. We schedule e since it has lower slack

$U_{2,mul} = \{g\}$

$\underline{l=3}$:

$U_{3,add} = \{b, c, f\}$. None of the operations has 0 slack. Any can be scheduled as they have the same slack. Let's schedule b.

$U_{3,mul} = \{\}$

$\underline{l=4}$:

$U_{4,add} = \{c, f\}$. Both of them have a slack of 0 and need to be scheduled. Thus, $a = [2, 1]$.

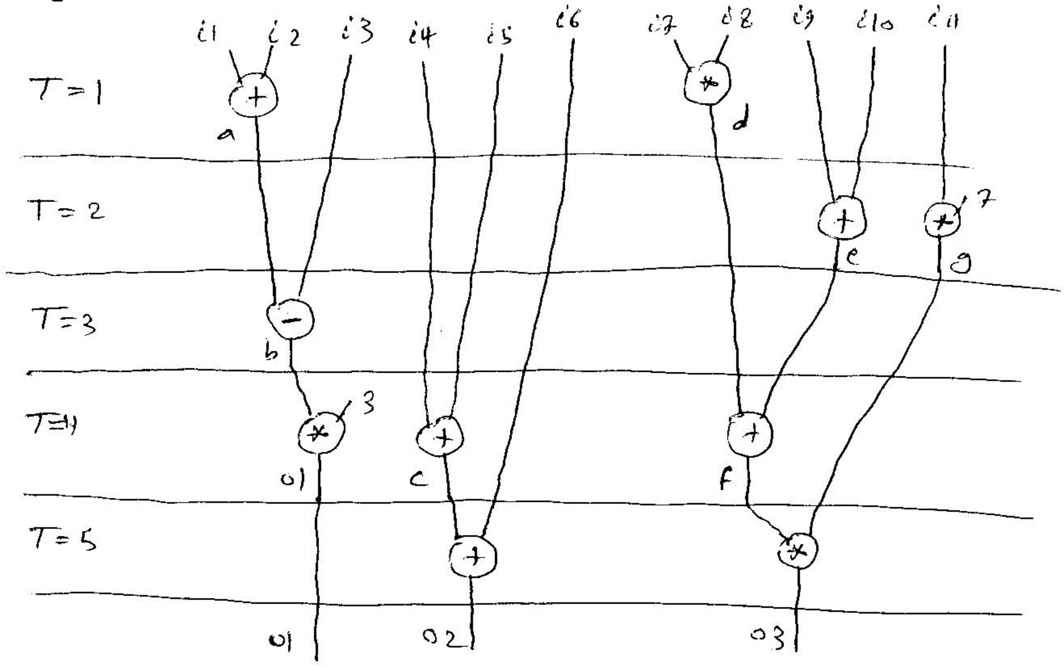
$U_{4,mul} = \{o1\}$

$\underline{l=5}$:

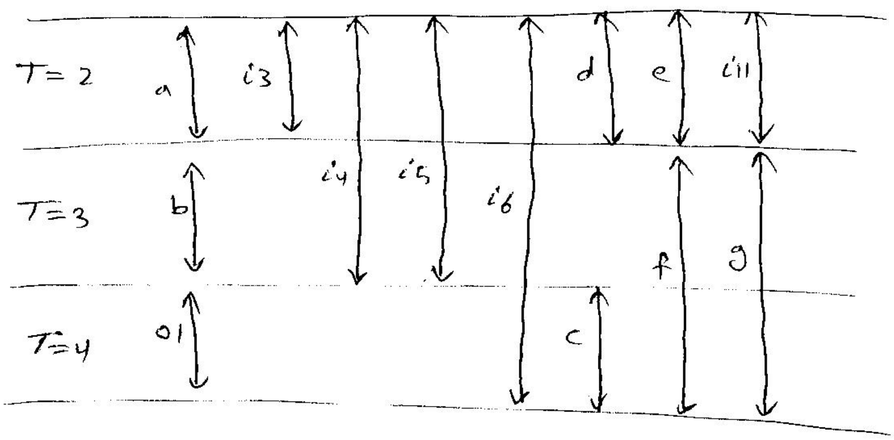
$U_{5,add} = \{o2\}$

$U_{5,mul} = \{o3\}$

Thus, with 5 clock cycles, we need a minimum of 2 adders and 1 multiplier as shown below:



(iii) a. Life time of variables



b. Based on the lifetime of all variables, it is obvious that we need 8 registers to store all variables.
We will assign variables to registers as follows to minimize area:

R1: (a, b, c) R2: (e, f) R3: (d, g)
R4: (i3, o1) R5: i4 R6: i5
R7: i6 R8: i11

c. Data Path: