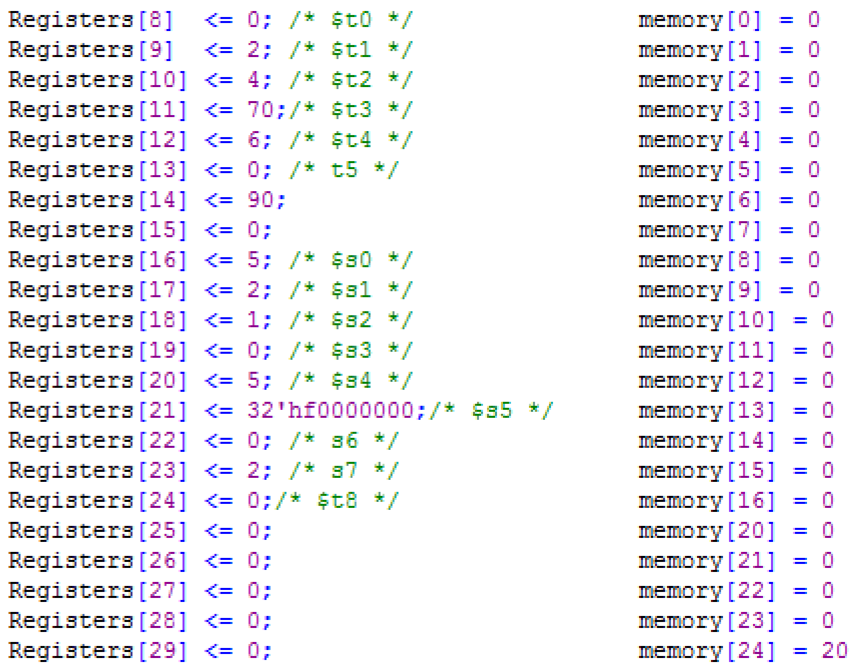
Test Cases

1 - Normal Test Cases :

1) Assembly Format	Binary Format
add $s0,$s1,$s2	00000010001100101000000000100000
sub $s3,$s4,$s5	00000010100101011001100000100010
ori $s1,$s1,10	00110110001100010000000000001010
addi $s2,$s2,10	00100010010100100000000000001010
sll $s4,$s4,2	00000000000101001010000010000000
sra $s5,$s5,2	00000000000101011010100010000011
slt $s6,$s7,$s0	00000010111100001011000000101010
andi $t0,$t1,10	00110001001010000000000000001010
lw $t1,20($t2)	10001101010010010000000000010100
sw $t2,0($t4)	10101101100010100000000000000000
lui $t5,100	00111100000011010000000001100100

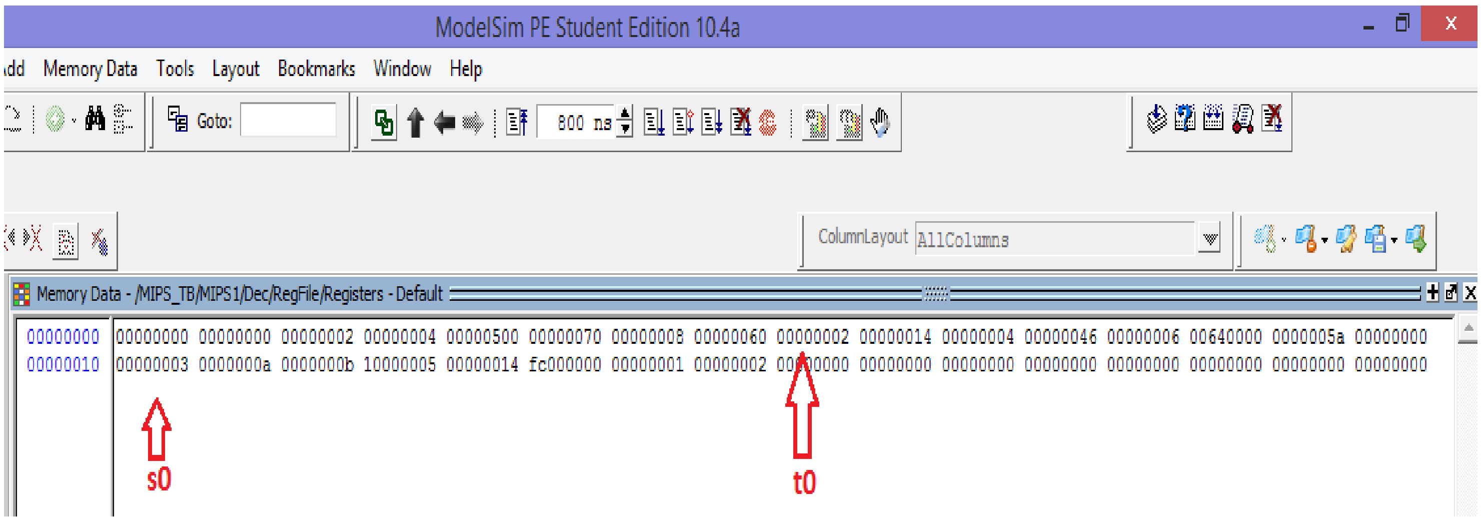


```
Registers[8]  <= 0; /* $t0 */              memory[0] = 0
Registers[9]  <= 2; /* $t1 */              memory[1] = 0
Registers[10] <= 4; /* $t2 */              memory[2] = 0
Registers[11] <= 70;/* $t3 */              memory[3] = 0
Registers[12] <= 6; /* $t4 */              memory[4] = 0
Registers[13] <= 0; /* t5 */               memory[5] = 0
Registers[14] <= 90;                       memory[6] = 0
Registers[15] <= 0;                        memory[7] = 0
Registers[16] <= 5; /* $s0 */              memory[8] = 0
Registers[17] <= 2; /* $s1 */              memory[9] = 0
Registers[18] <= 1; /* $s2 */              memory[10] = 0
Registers[19] <= 0; /* $s3 */              memory[11] = 0
Registers[20] <= 5; /* $s4 */              memory[12] = 0
Registers[21] <= 32'hf0000000;/* $s5 */    memory[13] = 0
Registers[22] <= 0; /* s6 */               memory[14] = 0
Registers[23] <= 2; /* s7 */               memory[15] = 0
Registers[24] <= 0;/* $t8 */               memory[16] = 0
Registers[25] <= 0;                        memory[20] = 0
Registers[26] <= 0;                        memory[21] = 0
Registers[27] <= 0;                        memory[22] = 0
Registers[28] <= 0;                        memory[23] = 0
Registers[29] <= 0;                        memory[24] = 20
```

Expected outputs :

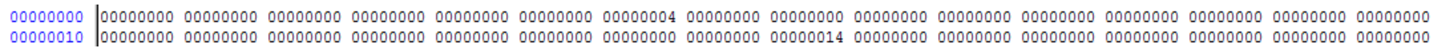
```
s0 = 3
s3 = -35
s1 = 10
s2 = 11
s4 = 20
s5 = 0xfc
s6 = 1
t0 = 2
t1 = 20
memory[6] = 4
t5 = 0x0064
```

Actual outputs :


ModelSim PE Student Edition 10.4a
dd Memory Data Tools Layout Bookmarks Window Help
Goto:
800 ns
ColumnLayout AllColumns
Memory Data - /MIPS_TB/MIPS1/Dec/RegFile/Registers - Default
s0
t0

```
00000000 00000000 00000000 00000002 00000004 00000500 00000070 00000008 00000060 00000002 00000014 00000004 00000046 00000006 00640000 0000005a 00000000
00000010 00000003 0000000a 0000000b 10000005 00000014 fc000000 00000001 00000002 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

Memory :



```
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000004 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000010 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000014 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

Testing Branch and Jump :

2) Assembly Format

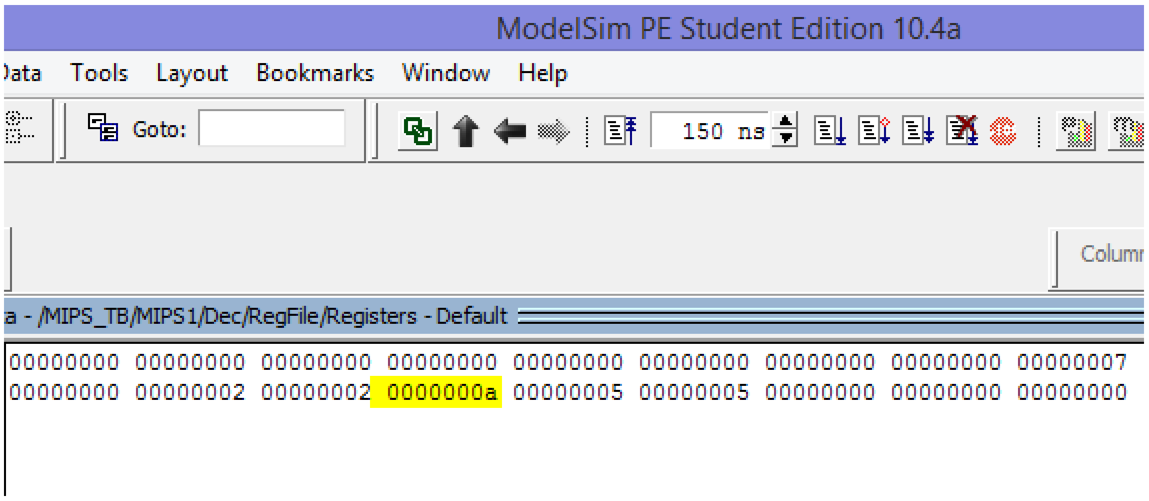
Assembly Format	Binary Format
InstrucionMem[2] = beq $s1,$s2,label	00010010001100100000000000001010
InstrucionMem[3] = add $s3,$s4,$t0	00000010100010001001100000100000
InstrucionMem[14] = label : add $s3,$s4,$s5	00000010100101011001100000100000

```
Registers[8]  <= 7; /* $t0 */
Registers[9]  <= 0; /* $t1 */
Registers[10] <= 0; /* $t2 */
Registers[11] <= 0;/* $t3 */
Registers[12] <= 0; /* $t4 */
Registers[13] <= 0; /* t5 */
Registers[14] <= 0;
Registers[15] <= 0;
Registers[16] <= 0; /* $s0 */
Registers[17] <= 2; /* $s1 */
Registers[18] <= 2; /* $s2 */
Registers[19] <= 0; /* $s3 */
Registers[20] <= 5; /* $s4 */
Registers[21] <= 5;/* $s5 */
Registers[22] <= 0; /* s6 */
Registers[23] <= 0; /* s7 */
Registers[24] <= 0;/* $t8 */
```

Expected outputs :

s3 = 10

Actual outputs :

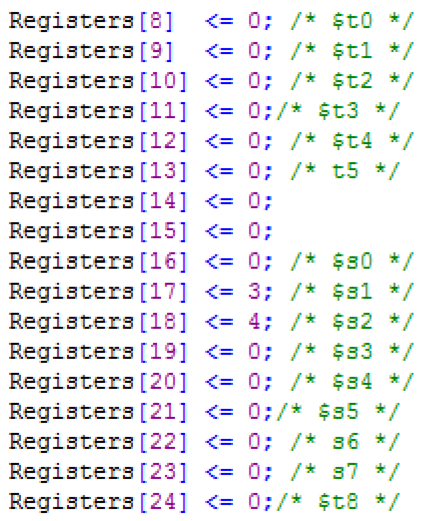

ModelSim PE Student Edition 10.4a
Data Tools Layout Bookmarks Window Help
Goto:
150 ns
Column

a - /MIPS_TB/MIPS1/Dec/RegFile/Registers - Default

```
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000007
00000000 00000002 00000002 0000000a 00000005 00000005 00000000 00000000 00000000
```

3) Assembly Format

Assembly Format	Binary Format
InstrucionMem[0] = bnq $s1,$s2,label	00010110001100100000000000001101
InstrucionMem[1] = add $s3,$s1,$s2	00000010001100101001100000100000
InstrucionMem[15] = label : addi $s3,$s3,10	00100010011100110000000000001010

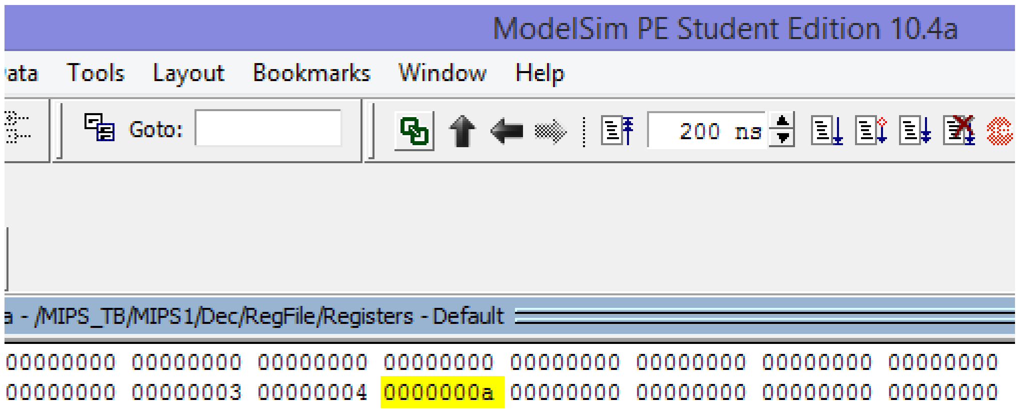


```
Registers[8]  <= 0; /* $t0 */
Registers[9]  <= 0; /* $t1 */
Registers[10] <= 0; /* $t2 */
Registers[11] <= 0;/* $t3 */
Registers[12] <= 0; /* $t4 */
Registers[13] <= 0; /* t5 */
Registers[14] <= 0;
Registers[15] <= 0;
Registers[16] <= 0; /* $s0 */
Registers[17] <= 3; /* $s1 */
Registers[18] <= 4; /* $s2 */
Registers[19] <= 0; /* $s3 */
Registers[20] <= 0; /* $s4 */
Registers[21] <= 0;/* $s5 */
Registers[22] <= 0; /* s6 */
Registers[23] <= 0; /* s7 */
Registers[24] <= 0;/* $t8 */
```

Expected outputs :

s3 = 10

Actual outputs :

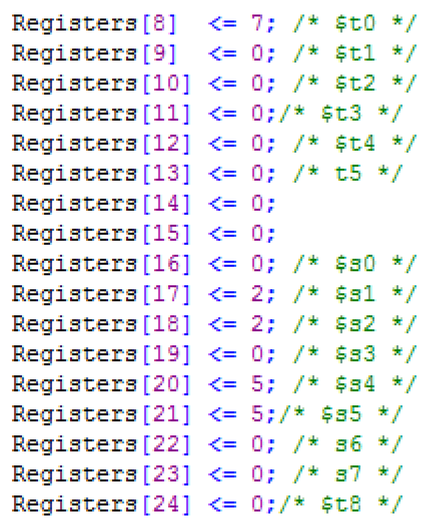

ModelSim PE Student Edition 10.4a
ata Tools Layout Bookmarks Window Help
Goto:
200 ns

a - /MIPS_TB/MIPS1/Dec/RegFile/Registers - Default

```
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000003 00000004 0000000a 00000000 00000000 00000000 00000000
```

4) Assembly Format **Binary Format**

InstrucionMem[1] = j label	00001000000000000000000000010011
InstrucionMem[2] = add $s3,$s4,$t0	00000010100010001001100000100000
InstrucionMem[21] = label : add $s0,$s1,$s2	00000010001100101000000000100000

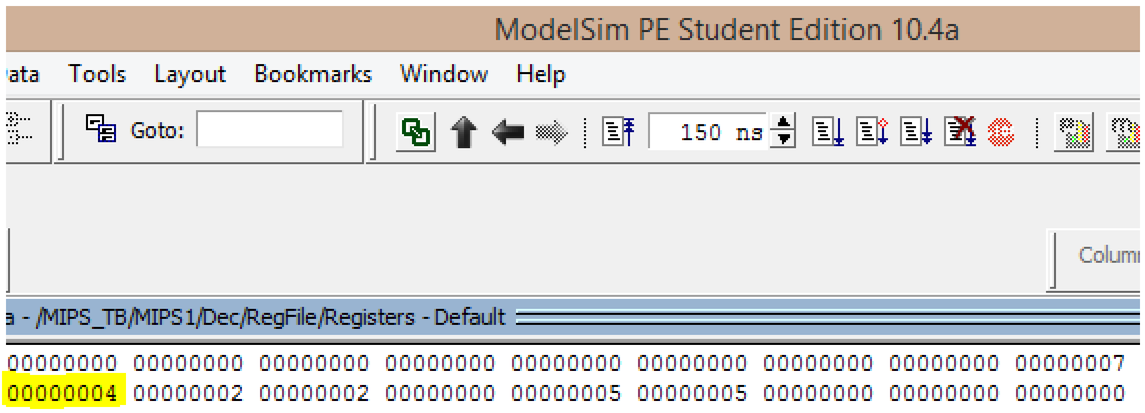


```
Registers[8]  <= 7; /* $t0 */
Registers[9]  <= 0; /* $t1 */
Registers[10] <= 0; /* $t2 */
Registers[11] <= 0;/* $t3 */
Registers[12] <= 0; /* $t4 */
Registers[13] <= 0; /* t5 */
Registers[14] <= 0;
Registers[15] <= 0;
Registers[16] <= 0; /* $s0 */
Registers[17] <= 2; /* $s1 */
Registers[18] <= 2; /* $s2 */
Registers[19] <= 0; /* $s3 */
Registers[20] <= 5; /* $s4 */
Registers[21] <= 5;/* $s5 */
Registers[22] <= 0; /* s6 */
Registers[23] <= 0; /* s7 */
Registers[24] <= 0;/* $t8 */
```

Expected outputs :

s0 = 4, s3 = 0

Actual outputs :



ModelSim PE Student Edition 10.4a
ata Tools Layout Bookmarks Window Help
Goto:
150 ns
Colum

a - /MIPS_TB/MIPS1/Dec/RegFile/Registers - Default

```
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000007
00000004 00000002 00000002 00000000 00000005 00000005 00000000 00000000 00000000
```

5) Assembly Format

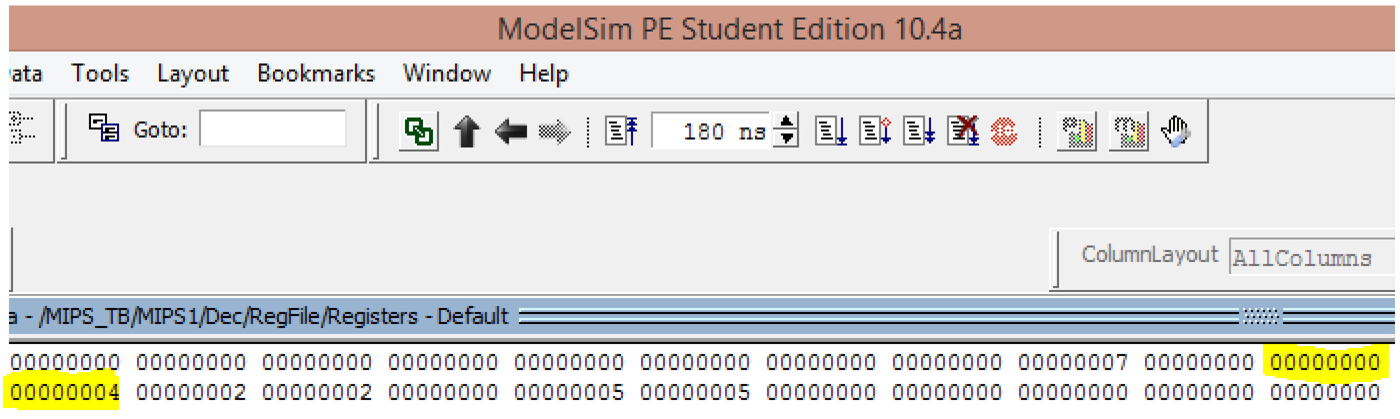
Assembly Format	Binary Format
InstrucionMem[1] = jr $t5	00000001101000000000000000001000
InstrucionMem[2] = add $t2,$s1,$s2	00000010001100100101000000100000
InstrucionMem[19] = add $s0,$s1,$s2	00000010001100101000000000100000

```
Registers[8]  <= 7; /* $t0 */
Registers[9]  <= 0; /* $t1 */
Registers[10] <= 0; /* $t2 */
Registers[11] <= 0;/* $t3 */
Registers[12] <= 0; /* $t4 */
Registers[13] <= 76; /* t5 */
Registers[14] <= 0;
Registers[15] <= 0;
Registers[16] <= 0; /* $s0 */
Registers[17] <= 2; /* $s1 */
Registers[18] <= 2; /* $s2 */
Registers[19] <= 0; /* $s3 */
Registers[20] <= 5; /* $s4 */
Registers[21] <= 5;/* $s5 */
Registers[22] <= 0; /* s6 */
Registers[23] <= 0; /* s7 */
Registers[24] <= 0;/* $t8 */
Registers[25] <= 0;
Registers[26] <= 0;
Registers[27] <= 0;
Registers[28] <= 0;
```

Expected outputs :

s0 = 4, t2 = 0

Actual outputs :


ModelSim PE Student Edition 10.4a
ata Tools Layout Bookmarks Window Help
Goto:
180 ns
ColumnLayout AllColumns

a - /MIPS_TB/MIPS1/Dec/RegFile/Registers - Default

```
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000007 00000000 00000000
00000004 00000002 00000002 00000000 00000005 00000005 00000000 00000000 00000000 00000000 00000000
```

6) Assembly Format

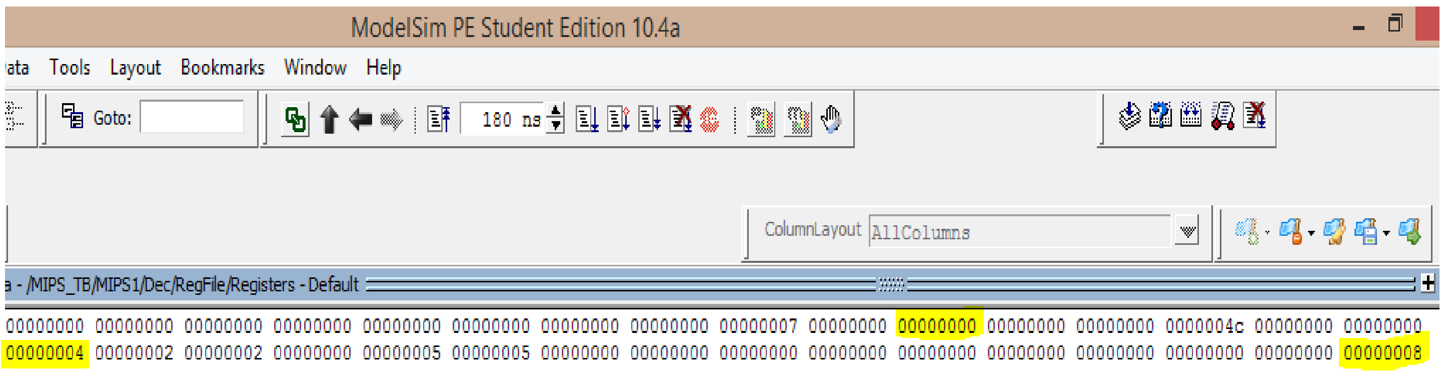
Assembly Format	Binary Format
InstrucionMem[1] = jal L	00001100000000000000000000010011
InstrucionMem[2] = add $t2,$s1,$s2	00000010001100100101000000100000
InstrucionMem[21] = L: add $s0,$s1,$s2	00000010001100101000000000100000

```
Registers[8]  <= 7; /* $t0 */
Registers[9]  <= 0; /* $t1 */
Registers[10] <= 0; /* $t2 */
Registers[11] <= 0;/* $t3 */
Registers[12] <= 0; /* $t4 */
Registers[13] <= 76; /* t5 */
Registers[14] <= 0;
Registers[15] <= 0;
Registers[16] <= 0; /* $s0 */
Registers[17] <= 2; /* $s1 */
Registers[18] <= 2; /* $s2 */
Registers[19] <= 0; /* $s3 */
Registers[20] <= 5; /* $s4 */
Registers[21] <= 5;/* $s5 */
Registers[22] <= 0; /* s6 */
Registers[23] <= 0; /* s7 */
Registers[24] <= 0;/* $t8 */
Registers[25] <= 0;
Registers[26] <= 0;
Registers[27] <= 0;
Registers[28] <= 0;
```

Expected outputs :

s0 = 4, t2 = 0, ra = 8

Actual outputs :


ModelSim PE Student Edition 10.4a
ata Tools Layout Bookmarks Window Help
Goto:
180 ns
ColumnLayout AllColumns

a - /MIPS_TB/MIPS1/Dec/RegFile/Registers - Default

```
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000007 00000000 00000000 00000000 00000000 0000004c 00000000 00000000
00000004 00000002 00000002 00000000 00000005 00000005 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000008
```

2 - Test Cases with Data Hazards :

1) Assembly Format

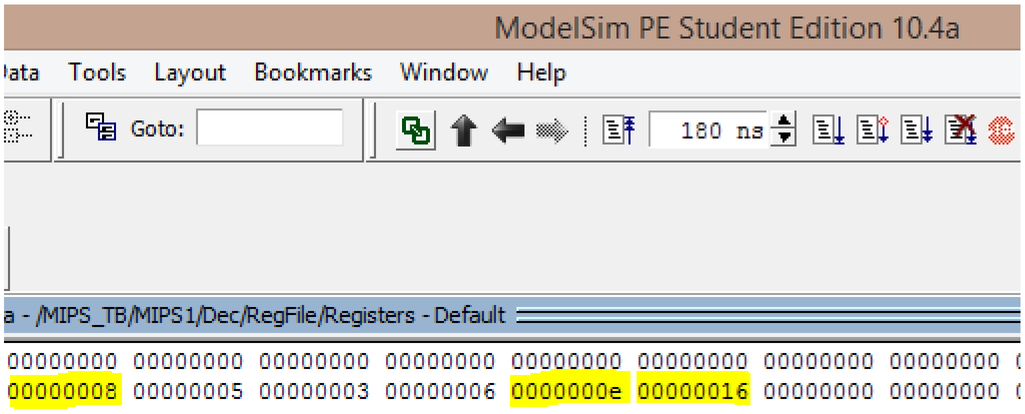
Assembly Format	Binary Format
InstrucionMem[1] = add $s0,$s2,$s1	00000010010100011000000000100000
InstrucionMem[2] = add $s4,$s0,$s3	00000010000100111010000000100000
InstrucionMem[3] = add $s5,$s4,$s0	00000010100100001010100000100000

```
Registers[16] <= 0; /* $s0 */
Registers[17] <= 5; /* $s1 */
Registers[18] <= 3; /* $s2 */
Registers[19] <= 6; /* $s3 */
Registers[20] <= 0; /* $s4 */
Registers[21] <= 0;/* $s5 */
Registers[22] <= 0; /* s6 */
Registers[23] <= 0; /* s7 */
Registers[24] <= 0;/* $t8 */
```

Expected outputs :

s0 = 8, s4 = 14, s5 = 24

Actual outputs :


ModelSim PE Student Edition 10.4a
ata Tools Layout Bookmarks Window Help
Goto:
180 ns
a - /MIPS_TB/MIPS1/Dec/RegFile/Registers - Default

```
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000008 00000005 00000003 00000006 0000000e 00000016 00000000 00000000
```

2) Assembly Format **Binary Format**

InstrucionMem[1] = lw $s0,0($s1)	10001110001100000000000000000000
InstrucionMem[2] = add $s4,$s0,$s3	00000010001100100101000000100000
InstrucionMem[3] = add $s5,$s4,$s0	00000010001100101000000000100000

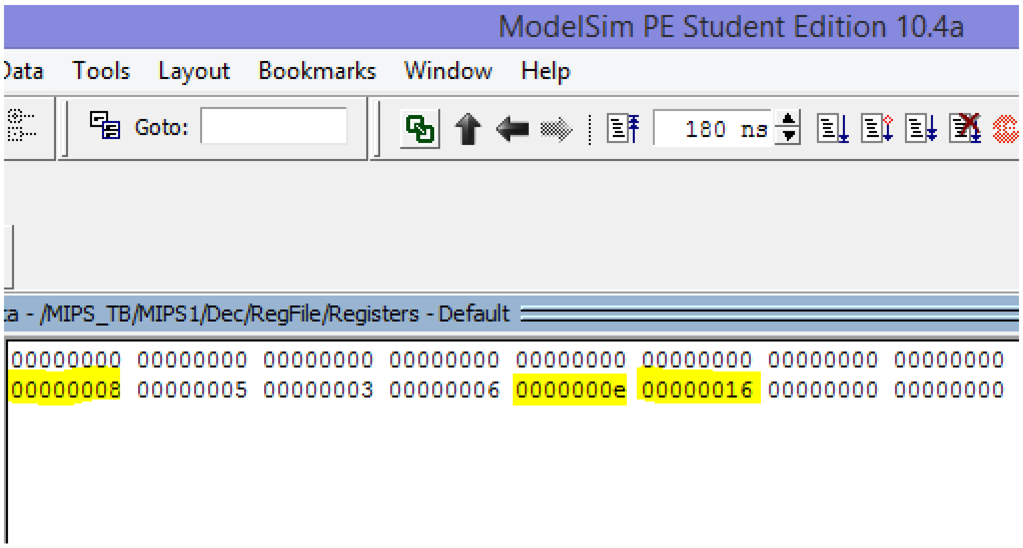
```
Registers[8]  <= 0; /* $t0 */
Registers[9]  <= 0; /* $t1 */
Registers[10] <= 0; /* $t2 */
Registers[11] <= 0;/* $t3 */
Registers[12] <= 0; /* $t4 */
Registers[13] <= 0; /* t5 */
Registers[14] <= 0;
Registers[15] <= 0;
Registers[16] <= 0; /* $s0 */
Registers[17] <= 5; /* $s1 */
Registers[18] <= 3; /* $s2 */
Registers[19] <= 6; /* $s3 */
Registers[20] <= 0; /* $s4 */
Registers[21] <= 0;/* $s5 */
Registers[22] <= 0; /* s6 */
Registers[23] <= 0; /* s7 */
Registers[24] <= 0;/* $t8 */
Registers[25] <= 0;
Registers[26] <= 0;
Registers[27] <= 0;
Registers[28] <= 0;
```

```
memory[0] = 0
memory[1] = 0
memory[2] = 0
memory[3] = 0
memory[4] = 0
memory[5] = 8
memory[6] = 0
memory[7] = 0
memory[8] = 0
memory[9] = 0
memory[10] = 0
memory[11] = 0
memory[12] = 0
memory[13] = 0
memory[14] = 0
memory[15] = 0
memory[16] = 0
memory[20] = 0
memory[21] = 0
memory[22] = 0
memory[23] = 0
```

Expected outputs :

s0 = 8, s4 = 14, s5 = 24

Actual outputs :


ModelSim PE Student Edition 10.4a
Data Tools Layout Bookmarks Window Help
Goto:
180 ns

a - /MIPS_TB/MIPS1/Dec/RegFile/Registers - Default

```
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000008 00000005 00000003 00000006 0000000e 00000016 00000000 00000000
```

3) Assembly Format

Assembly Format	Binary Format
InstrucionMem[1] = lw $s0,0($s1)	10001110001100000000000000000000
InstrucionMem[2] = sw $s2,0($s0)	10101110000100100000000000000000
InstrucionMem[3] = lw $t0,0($t1)	10001101001010000000000000000000

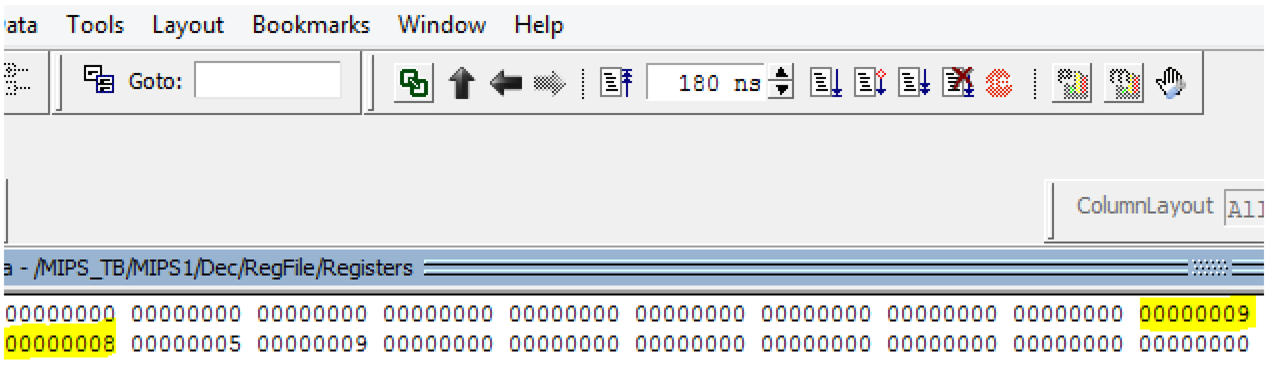
```
Registers[8]  <= 0; /* $t0 */
Registers[9]  <= 9; /* $t1 */
Registers[10] <= 0; /* $t2 */
Registers[11] <= 0;/* $t3 */
Registers[12] <= 0; /* $t4 */
Registers[13] <= 0; /* t5 */
Registers[14] <= 0;
Registers[15] <= 0;
Registers[16] <= 0; /* $s0 */
Registers[17] <= 5; /* $s1 */
Registers[18] <= 9; /* $s2 */
Registers[19] <= 0; /* $s3 */
Registers[20] <= 0; /* $s4 */
Registers[21] <= 0;/* $s5 */
Registers[22] <= 0; /* s6 */
Registers[23] <= 0; /* s7 */
Registers[24] <= 0;/* $t8 */
```

```
memory[0] = 0
memory[1] = 0
memory[2] = 0
memory[3] = 0
memory[4] = 0
memory[5] = 8
memory[6] = 0
memory[7] = 0
memory[8] = 0
memory[9] = 0
memory[10] = 0
memory[11] = 0
memory[12] = 0
memory[13] = 0
memory[14] = 0
memory[15] = 0
memory[16] = 0
```

Expected outputs :

s0 = 8, t0 = 9, memory[8]=9

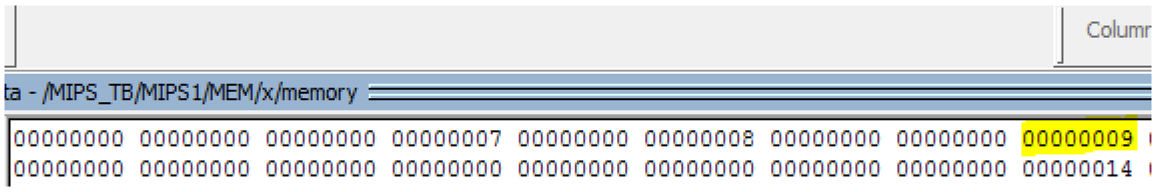
Actual outputs :



a - /MIPS_TB/MIPS1/Dec/RegFile/Registers

```
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000009
00000008 00000005 00000009 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

Memory:



ta - /MIPS_TB/MIPS1/MEM/x/memory

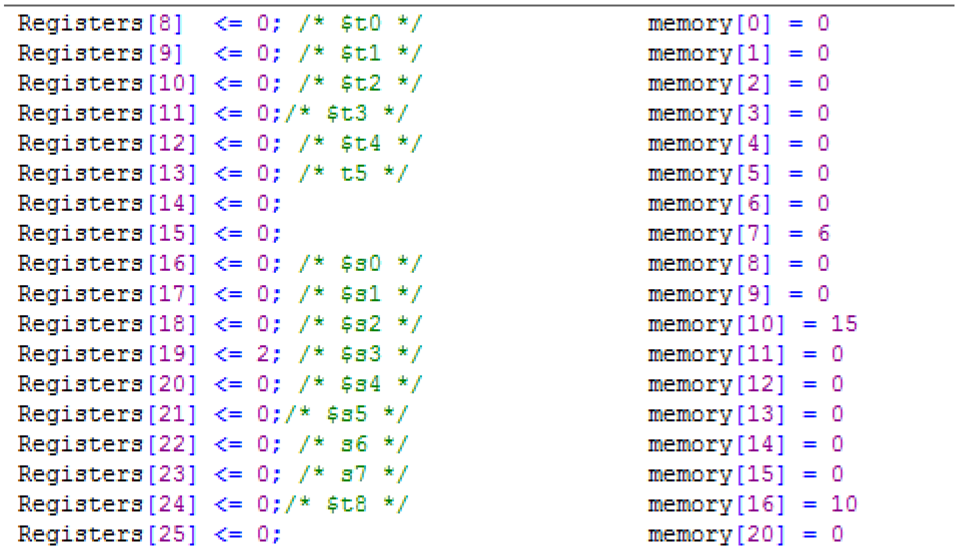
```
00000000 00000000 00000000 00000007 00000000 00000008 00000000 00000000 00000009
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000014
```

4) <u>Assembly Format</u> **<u>Binary Format</u>**

InstrucionMem[1] = lw $s1, 5($s3) 10001110011100010000000000000101

InstrucionMem[2] = lw $t0, 10($s1) 10001110001010000000000000001010

InstrucionMem[3] = lw $s2, 0($t0)) 10001101000100100000000000000000

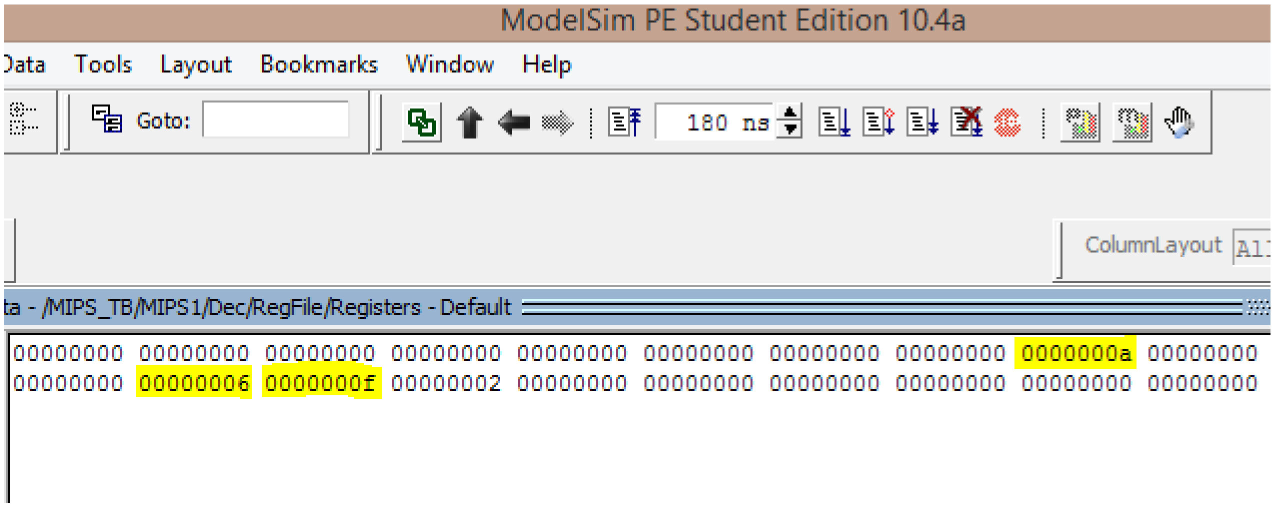


```
Registers[8]  <= 0; /* $t0 */           memory[0] = 0
Registers[9]  <= 0; /* $t1 */           memory[1] = 0
Registers[10] <= 0; /* $t2 */           memory[2] = 0
Registers[11] <= 0;/* $t3 */            memory[3] = 0
Registers[12] <= 0; /* $t4 */           memory[4] = 0
Registers[13] <= 0; /* t5 */            memory[5] = 0
Registers[14] <= 0;                     memory[6] = 0
Registers[15] <= 0;                     memory[7] = 6
Registers[16] <= 0; /* $s0 */           memory[8] = 0
Registers[17] <= 0; /* $s1 */           memory[9] = 0
Registers[18] <= 0; /* $s2 */           memory[10] = 15
Registers[19] <= 2; /* $s3 */           memory[11] = 0
Registers[20] <= 0; /* $s4 */           memory[12] = 0
Registers[21] <= 0;/* $s5 */            memory[13] = 0
Registers[22] <= 0; /* s6 */            memory[14] = 0
Registers[23] <= 0; /* s7 */            memory[15] = 0
Registers[24] <= 0;/* $t8 */            memory[16] = 10
Registers[25] <= 0;                     memory[20] = 0
```

Expected outputs :

s1 = 6, t0 = 10, s2=15

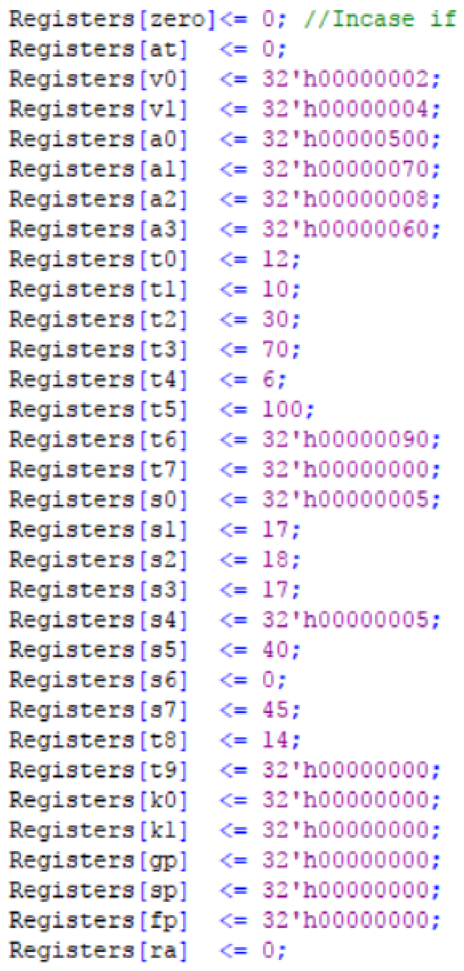
Actual outputs :



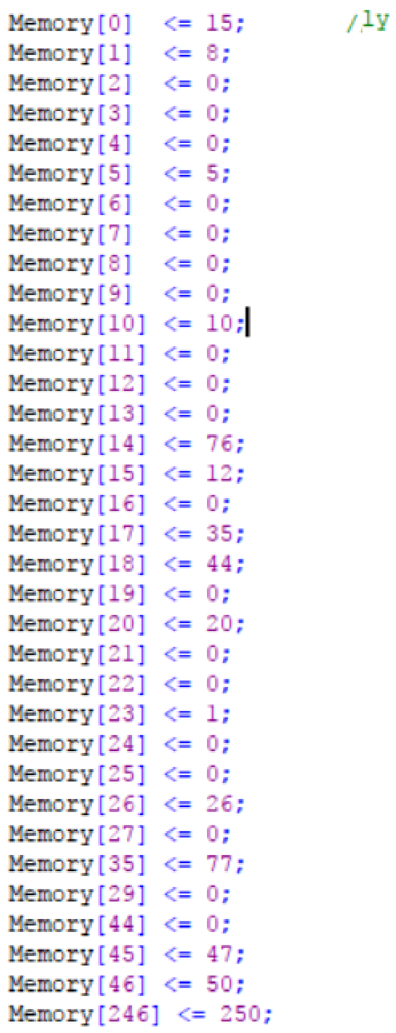
ModelSim PE Student Edition 10.4a
Data Tools Layout Bookmarks Window Help
Goto:
180 ns
ColumnLayout
ta - /MIPS_TB/MIPS1/Dec/RegFile/Registers - Default

```
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 0000000a 00000000
00000000 00000006 0000000f 00000002 00000000 00000000 00000000 00000000 00000000 00000000
```

5) **Assembly Format**	**Binary Format**
add $s1, $s2, $s3	00000010010100111000100000100000
add $s1, $s1, $s4	00000010001101001000100000100000
sub $s5, $s5, $s1	00000010101100011010100000100010



```
Registers[zero]<= 0; //Incase if
Registers[at]   <= 0;
Registers[v0]   <= 32'h00000002;
Registers[v1]   <= 32'h00000004;
Registers[a0]   <= 32'h00000500;
Registers[a1]   <= 32'h00000070;
Registers[a2]   <= 32'h00000008;
Registers[a3]   <= 32'h00000060;
Registers[t0]   <= 12;
Registers[t1]   <= 10;
Registers[t2]   <= 30;
Registers[t3]   <= 70;
Registers[t4]   <= 6;
Registers[t5]   <= 100;
Registers[t6]   <= 32'h00000090;
Registers[t7]   <= 32'h00000000;
Registers[s0]   <= 32'h00000005;
Registers[s1]   <= 17;
Registers[s2]   <= 18;
Registers[s3]   <= 17;
Registers[s4]   <= 32'h00000005;
Registers[s5]   <= 40;
Registers[s6]   <= 0;
Registers[s7]   <= 45;
Registers[t8]   <= 14;
Registers[t9]   <= 32'h00000000;
Registers[k0]   <= 32'h00000000;
Registers[k1]   <= 32'h00000000;
Registers[gp]   <= 32'h00000000;
Registers[sp]   <= 32'h00000000;
Registers[fp]   <= 32'h00000000;
Registers[ra]   <= 0;
```



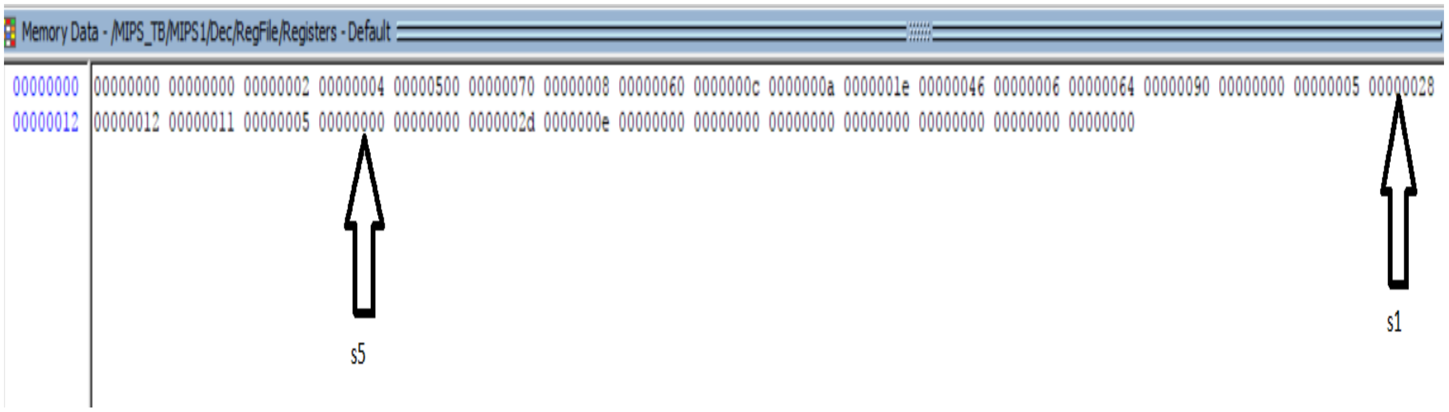
```
Memory[0]   <= 15;        /ly
Memory[1]   <= 8;
Memory[2]   <= 0;
Memory[3]   <= 0;
Memory[4]   <= 0;
Memory[5]   <= 5;
Memory[6]   <= 0;
Memory[7]   <= 0;
Memory[8]   <= 0;
Memory[9]   <= 0;
Memory[10]  <= 10;
Memory[11]  <= 0;
Memory[12]  <= 0;
Memory[13]  <= 0;
Memory[14]  <= 76;
Memory[15]  <= 12;
Memory[16]  <= 0;
Memory[17]  <= 35;
Memory[18]  <= 44;
Memory[19]  <= 0;
Memory[20]  <= 20;
Memory[21]  <= 0;
Memory[22]  <= 0;
Memory[23]  <= 1;
Memory[24]  <= 0;
Memory[25]  <= 0;
Memory[26]  <= 26;
Memory[27]  <= 0;
Memory[35]  <= 77;
Memory[29]  <= 0;
Memory[44]  <= 0;
Memory[45]  <= 47;
Memory[46]  <= 50;
Memory[246]  <= 250;
```

Expected outputs :

S1=40

S5=0

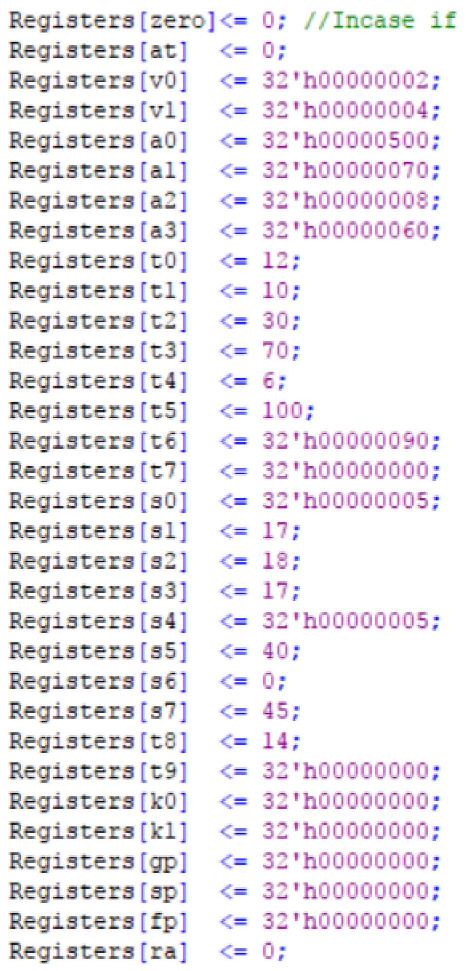
Actual outputs :


Memory Data - /MIPS_TB/MIPS1/Dec/RegFile/Registers - Default

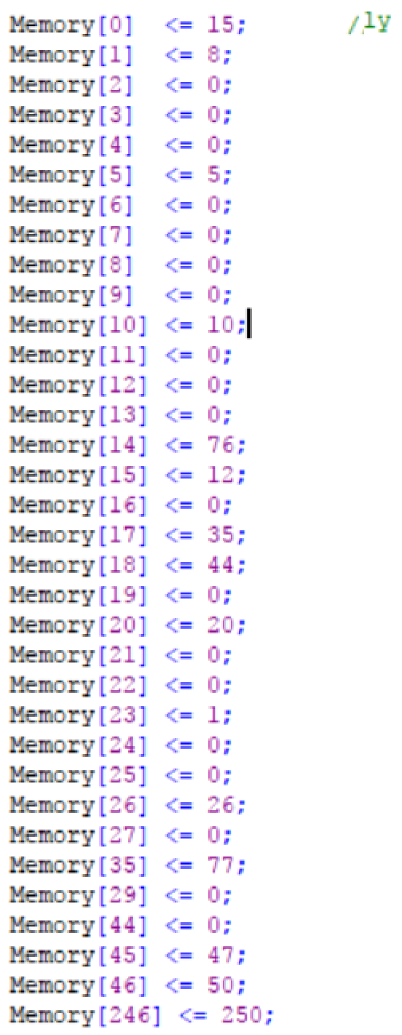
```
00000000 00000000 00000000 00000002 00000004 00000500 00000070 00000008 00000060 0000000c 0000000a 0000001e 00000046 00000006 00000064 00000090 00000000 00000005 00000028
00000012 00000012 00000011 00000005 00000000 00000000 0000002d 0000000e 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

6) <u>Assembly Format</u> **<u>Binary Format</u>**

lw $s1,0($s2)	10001110010100010000000000000000
sw $s1,0($s4)	10101110100100010000000000000000



```
Registers[zero]<= 0; //Incase if
Registers[at]   <= 0;
Registers[v0]   <= 32'h00000002;
Registers[v1]   <= 32'h00000004;
Registers[a0]   <= 32'h00000500;
Registers[a1]   <= 32'h00000070;
Registers[a2]   <= 32'h00000008;
Registers[a3]   <= 32'h00000060;
Registers[t0]   <= 12;
Registers[t1]   <= 10;
Registers[t2]   <= 30;
Registers[t3]   <= 70;
Registers[t4]   <= 6;
Registers[t5]   <= 100;
Registers[t6]   <= 32'h00000090;
Registers[t7]   <= 32'h00000000;
Registers[s0]   <= 32'h00000005;
Registers[s1]   <= 17;
Registers[s2]   <= 18;
Registers[s3]   <= 17;
Registers[s4]   <= 32'h00000005;
Registers[s5]   <= 40;
Registers[s6]   <= 0;
Registers[s7]   <= 45;
Registers[t8]   <= 14;
Registers[t9]   <= 32'h00000000;
Registers[k0]   <= 32'h00000000;
Registers[k1]   <= 32'h00000000;
Registers[gp]   <= 32'h00000000;
Registers[sp]   <= 32'h00000000;
Registers[fp]   <= 32'h00000000;
Registers[ra]   <= 0;
```



```
Memory[0]   <= 15;        /ly
Memory[1]   <= 8;
Memory[2]   <= 0;
Memory[3]   <= 0;
Memory[4]   <= 0;
Memory[5]   <= 5;
Memory[6]   <= 0;
Memory[7]   <= 0;
Memory[8]   <= 0;
Memory[9]   <= 0;
Memory[10]  <= 10;
Memory[11]  <= 0;
Memory[12]  <= 0;
Memory[13]  <= 0;
Memory[14]  <= 76;
Memory[15]  <= 12;
Memory[16]  <= 0;
Memory[17]  <= 35;
Memory[18]  <= 44;
Memory[19]  <= 0;
Memory[20]  <= 20;
Memory[21]  <= 0;
Memory[22]  <= 0;
Memory[23]  <= 1;
Memory[24]  <= 0;
Memory[25]  <= 0;
Memory[26]  <= 26;
Memory[27]  <= 0;
Memory[35]  <= 77;
Memory[29]  <= 0;
Memory[44]  <= 0;
Memory[45]  <= 47;
Memory[46]  <= 50;
Memory[246]  <= 250;
```

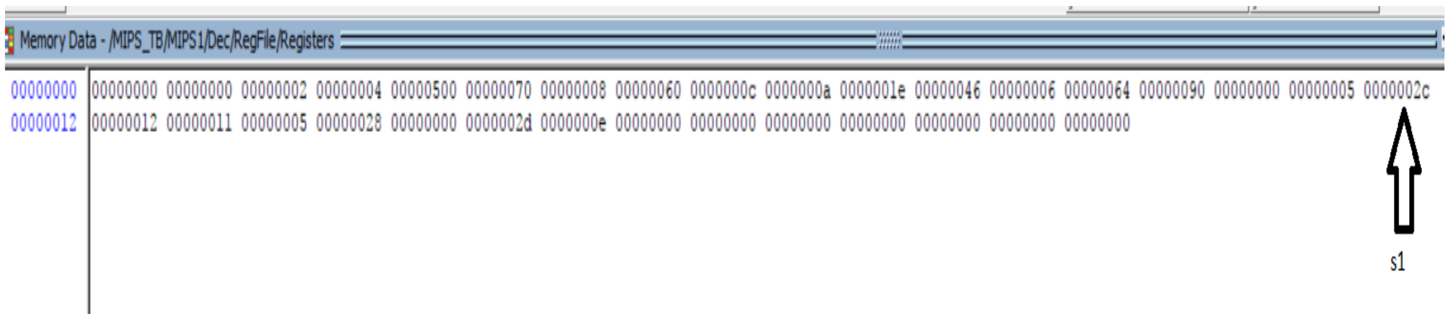
Expected outputs :

S1=44

Memory[5]=44

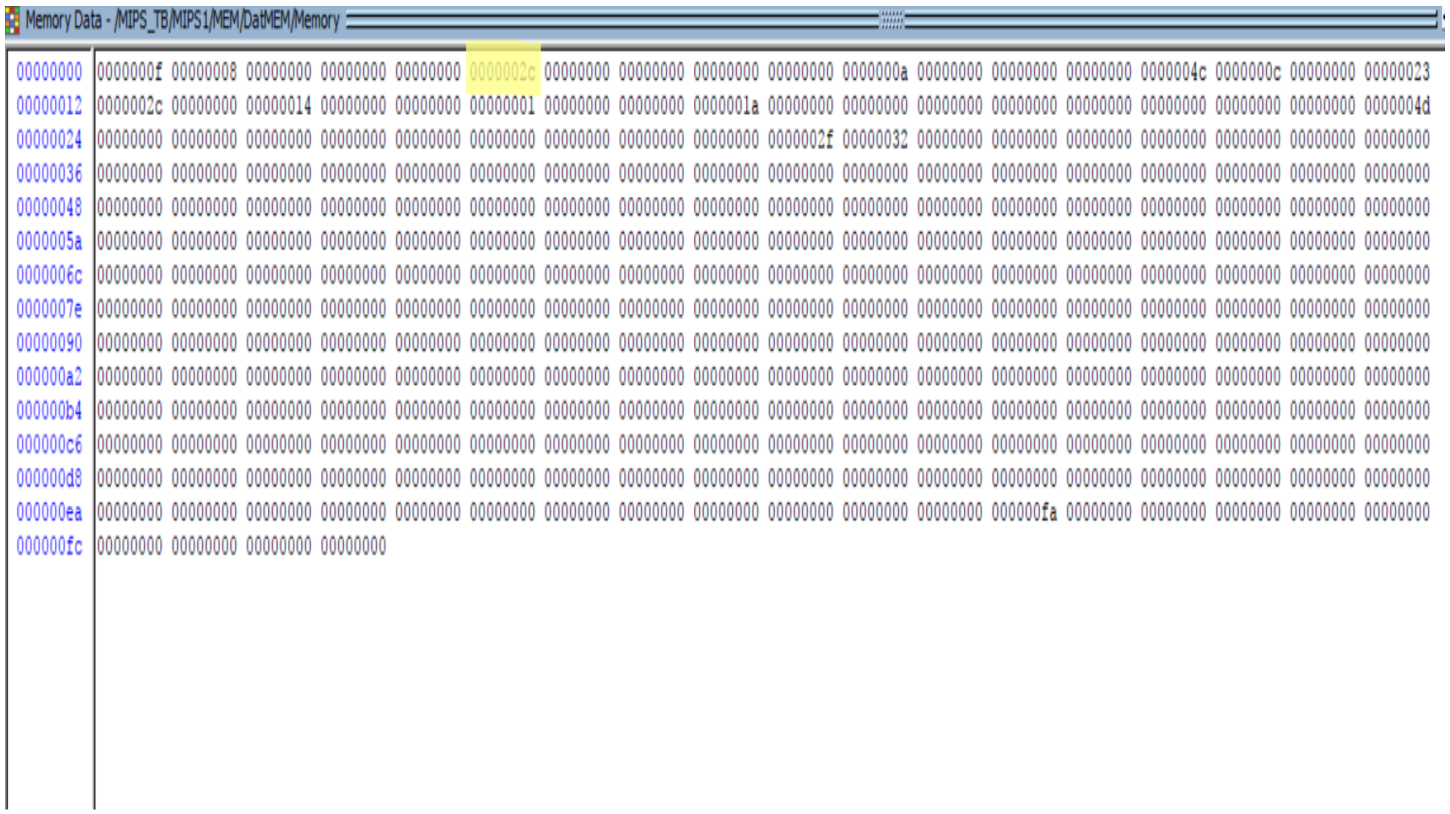
Actual outputs :

Registers:



```
Memory Data - /MIPS_TB/MIPS1/Dec/RegFile/Registers
00000000 00000000 00000000 00000002 00000004 00000500 00000070 00000008 00000060 0000000c 0000000a 0000001e 00000046 00000006 00000064 00000090 00000000 00000005 0000002c
00000012 00000012 00000011 00000005 00000028 00000000 0000002d 0000000e 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

Memory:

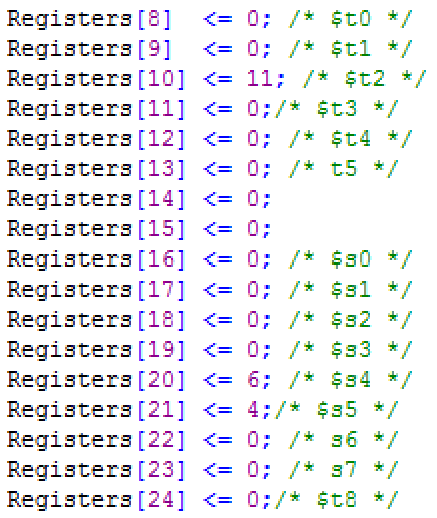


```
Memory Data - /MIPS_TB/MIPS1/MEM/DatMEM/Memory
00000000 0000000f 00000008 00000000 00000000 00000000 0000002c 00000000 00000000 00000000 00000000 0000000a 00000000 00000000 00000000 0000004c 0000000c 00000000 00000023
00000012 0000002c 00000000 00000014 00000000 00000000 00000001 00000000 00000000 0000001a 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 0000004d
00000024 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 0000002f 00000032 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000036 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000048 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0000005a 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0000006c 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0000007e 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000090 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
000000a2 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
000000b4 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
000000c6 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
000000d8 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
000000ea 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 000000fa 00000000 00000000 00000000 00000000 00000000
000000fc 00000000 00000000 00000000 00000000
```

3 - Test Cases with Control Hazards :

1) Assembly Format

Assembly Format	Binary Format
InstrucionMem[0] = add $s2,$s4,$s5	00000010100101011001000000100000
InstrucionMem[1] = add $s1,$s2,$s6	00000010010101101000100000100000
InstrucionMem[2] = beq $s1,$s2,Label	00010010001100100000000000001010
InstrucionMem[3] = add $t0,$s1,$t2	00000001001010100100000000100000
Label : InstrucionMem[13] = add $t0,$t0,$t2	00000001000010100100000000100000

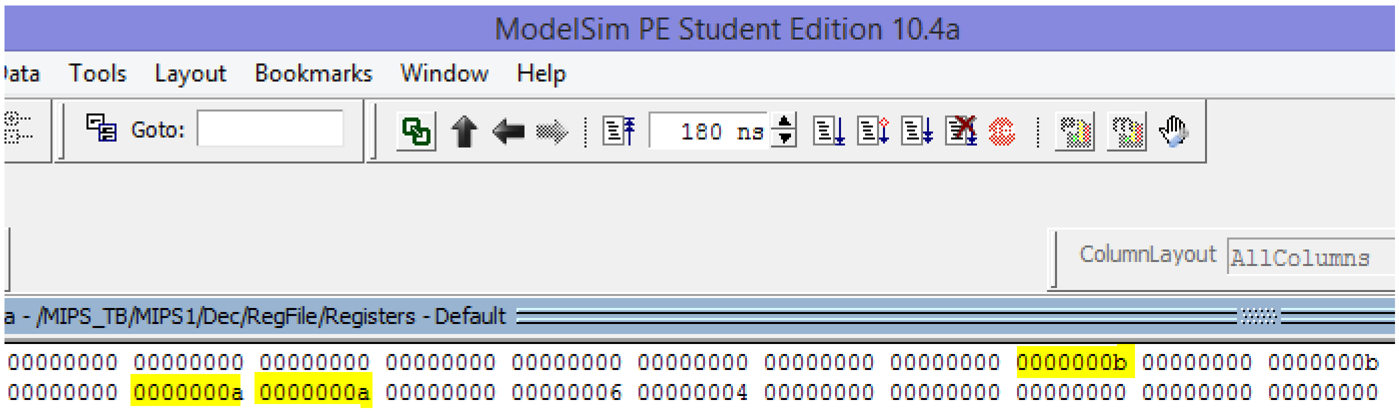


```
Registers[8]  <= 0; /* $t0 */
Registers[9]  <= 0; /* $t1 */
Registers[10] <= 11; /* $t2 */
Registers[11] <= 0;/* $t3 */
Registers[12] <= 0; /* $t4 */
Registers[13] <= 0; /* t5 */
Registers[14] <= 0;
Registers[15] <= 0;
Registers[16] <= 0; /* $s0 */
Registers[17] <= 0; /* $s1 */
Registers[18] <= 0; /* $s2 */
Registers[19] <= 0; /* $s3 */
Registers[20] <= 6; /* $s4 */
Registers[21] <= 4;/* $s5 */
Registers[22] <= 0; /* s6 */
Registers[23] <= 0; /* s7 */
Registers[24] <= 0;/* $t8 */
```

Expected outputs :

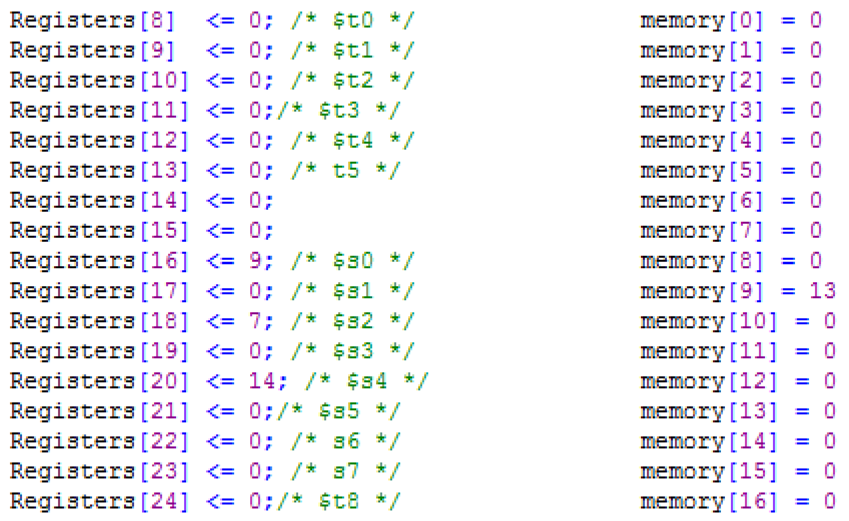
s1 = 10, s2 = 10, t0=11

Actual outputs :


ModelSim PE Student Edition 10.4a
ata Tools Layout Bookmarks Window Help
Goto:
180 ns
ColumnLayout AllColumns
a - /MIPS_TB/MIPS1/Dec/RegFile/Registers - Default

```
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 0000000b 00000000 0000000b
00000000 0000000a 0000000a 00000000 00000006 00000004 00000000 00000000 00000000 00000000 00000000
```

2) Assembly Format	**Binary Format**
InstrucionMem[0] = lw $t1, 0($s0)	10001110000010010000000000000000
InstrucionMem[1] = add $t2,$t1,$s3	00000001001100110101000000100000
InstrucionMem[2] = beq $t1,$t2,label	00010001001010100000000000010000
InstrucionMem[3] = add $s1,$s3,$s4	00000010011101001000100000100000
Label : InstrucionMem[19] = add $s1,$s2,$s3	00000010010100111000100000100000

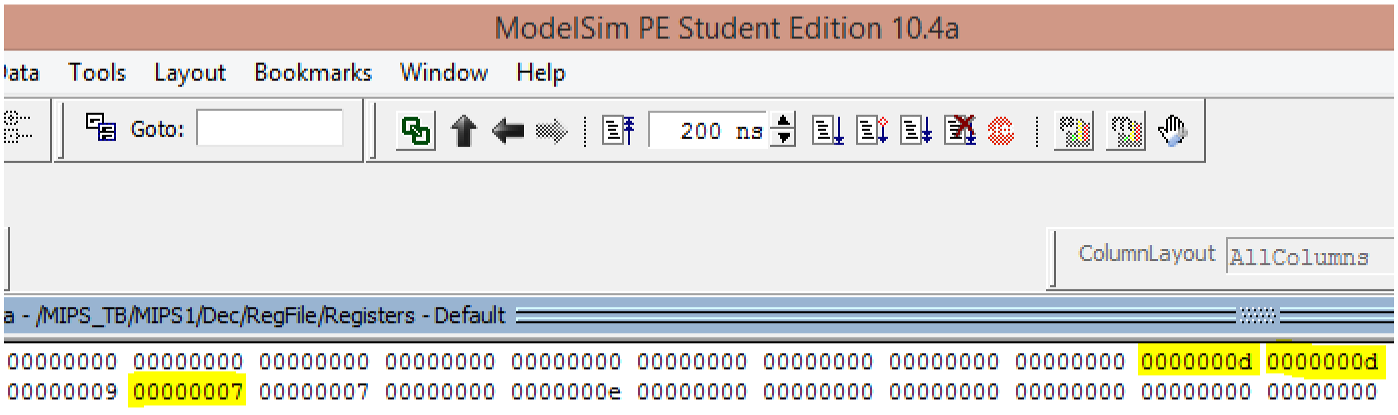


```
Registers[8]  <= 0; /* $t0 */          memory[0]  = 0
Registers[9]  <= 0; /* $t1 */          memory[1]  = 0
Registers[10] <= 0; /* $t2 */          memory[2]  = 0
Registers[11] <= 0;/* $t3 */           memory[3]  = 0
Registers[12] <= 0; /* $t4 */          memory[4]  = 0
Registers[13] <= 0; /* t5 */           memory[5]  = 0
Registers[14] <= 0;                    memory[6]  = 0
Registers[15] <= 0;                    memory[7]  = 0
Registers[16] <= 9; /* $s0 */          memory[8]  = 0
Registers[17] <= 0; /* $s1 */          memory[9]  = 13
Registers[18] <= 7; /* $s2 */          memory[10]  = 0
Registers[19] <= 0; /* $s3 */          memory[11]  = 0
Registers[20] <= 14; /* $s4 */         memory[12]  = 0
Registers[21] <= 0;/* $s5 */           memory[13]  = 0
Registers[22] <= 0; /* s6 */           memory[14]  = 0
Registers[23] <= 0; /* s7 */           memory[15]  = 0
Registers[24] <= 0;/* $t8 */           memory[16]  = 0
```

Expected outputs :

t1 = 13, t2 = 13, s1=7

Actual outputs :


ModelSim PE Student Edition 10.4a
)ata Tools Layout Bookmarks Window Help
Goto:
200 ns
ColumnLayout AllColumns

a - /MIPS_TB/MIPS1/Dec/RegFile/Registers - Default

```
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 0000000d 0000000d
00000009 00000007 00000007 00000000 0000000e 00000000 00000000 00000000 00000000 00000000 00000000
```

3) Assembly Format — Binary Format

Assembly Format	Binary Format
InstrucionMem[0] = lw $t1, 0($s0)	10001110000010010000000000000000
InstrucionMem[1] = lw $t2, 0($s1)	10001110001010100000000000000000
InstrucionMem[2] = beq $t1,$t2,label	00010001001010100000000000001010
InstrucionMem[3] = add $s2,$s3,$s3	00000010011100111000100000100000
Label : InstrucionMem[13] = add $s1,$t1,$t2	00000001001010101000100000100000

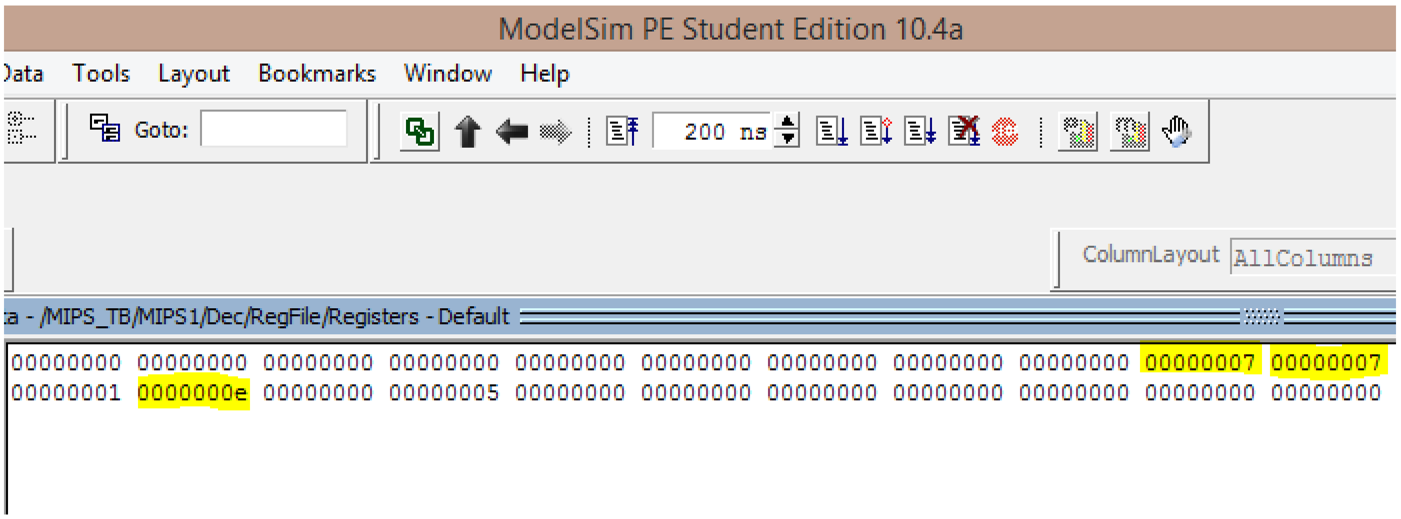
```
Registers[8]  <= 0; /* $t0 */
Registers[9]  <= 0; /* $t1 */
Registers[10] <= 0; /* $t2 */
Registers[11] <= 0;/* $t3 */
Registers[12] <= 0; /* $t4 */
Registers[13] <= 0; /* t5 */
Registers[14] <= 0;
Registers[15] <= 0;
Registers[16] <= 1; /* $s0 */
Registers[17] <= 2; /* $s1 */
Registers[18] <= 0; /* $s2 */
Registers[19] <= 5; /* $s3 */
Registers[20] <= 0; /* $s4 */
Registers[21] <= 0;/* $s5 */
Registers[22] <= 0; /* s6 */
Registers[23] <= 0; /* s7 */
Registers[24] <= 0;/* $t8 */
Registers[25] <= 0;
```

```
memory[0] = 7
memory[1] = 7
memory[2] = 0
memory[3] = 0
memory[4] = 0
memory[5] = 0
memory[6] = 0
memory[7] = 0
memory[8] = 0
memory[9] = 0
memory[10] = 0
memory[11] = 0
memory[12] = 0
memory[13] = 0
memory[14] = 0
memory[15] = 0
memory[16] = 0
memory[20] = 0
```

Expected outputs :

t1 = 7, t2 = 7, s1=14

Actual outputs :


ModelSim PE Student Edition 10.4a
Data Tools Layout Bookmarks Window Help
Goto:
200 ns
ColumnLayout AllColumns
a - /MIPS_TB/MIPS1/Dec/RegFile/Registers - Default

```
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000007 00000007
00000001 0000000e 00000000 00000005 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

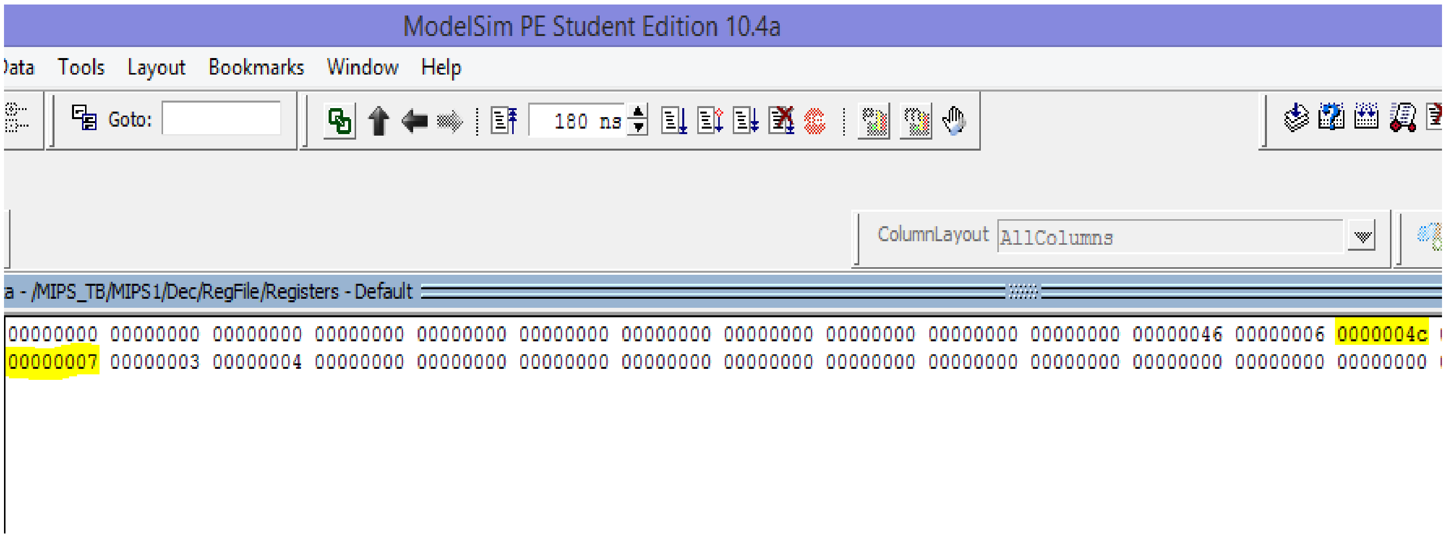
4) Assembly Format	**Binary Format**
InstrucionMem[1] = add $t5,$t3,$t4	00000001011011000110100000100000
InstrucionMem[2] = jr $t5	00000001101000000000000000001000
InstrucionMem[3] = add $t2,$s1,$s2	00000010001100100101000000100000
InstrucionMem[20] = add $s0,$s1,$s2	00000010001100101000000000100000

```
Registers[8]  <= 0; /* $t0 */
Registers[9]  <= 0; /* $t1 */
Registers[10] <= 0; /* $t2 */
Registers[11] <= 70;/* $t3 */
Registers[12] <= 6; /* $t4 */
Registers[13] <= 0; /* t5 */
Registers[14] <= 0;
Registers[15] <= 0;
Registers[16] <= 0; /* $s0 */
Registers[17] <= 3; /* $s1 */
Registers[18] <= 4; /* $s2 */
Registers[19] <= 0; /* $s3 */
Registers[20] <= 0; /* $s4 */
Registers[21] <= 0;/* $s5 */
Registers[22] <= 0; /* s6 */
Registers[23] <= 0; /* s7 */
Registers[24] <= 0;/* $t8 */
```

Expected outputs :

t5 = 76, s0=7

Actual outputs :

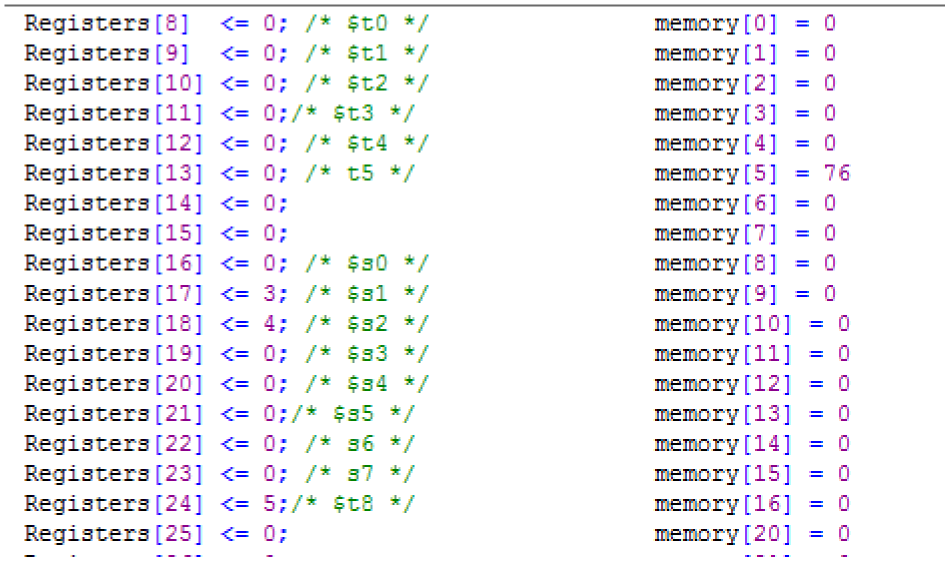

ModelSim PE Student Edition 10.4a
Data Tools Layout Bookmarks Window Help
Goto:
180 ns
ColumnLayout AllColumns

a - /MIPS_TB/MIPS1/Dec/RegFile/Registers - Default

```
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000046 00000006 0000004c
00000007 00000003 00000004 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

5) Assembly Format **Binary Format**

InstrucionMem[1] = lw $t5,0($t8)	10001111000011010000000000000000
InstrucionMem[2] = jr $t5	00000001101000000000000000001000
InstrucionMem[3] = add $t2,$s1,$s2	00000010001100100101000000100000
InstrucionMem[20] = add $s0,$s1,$s2	00000010001100101000000000100000

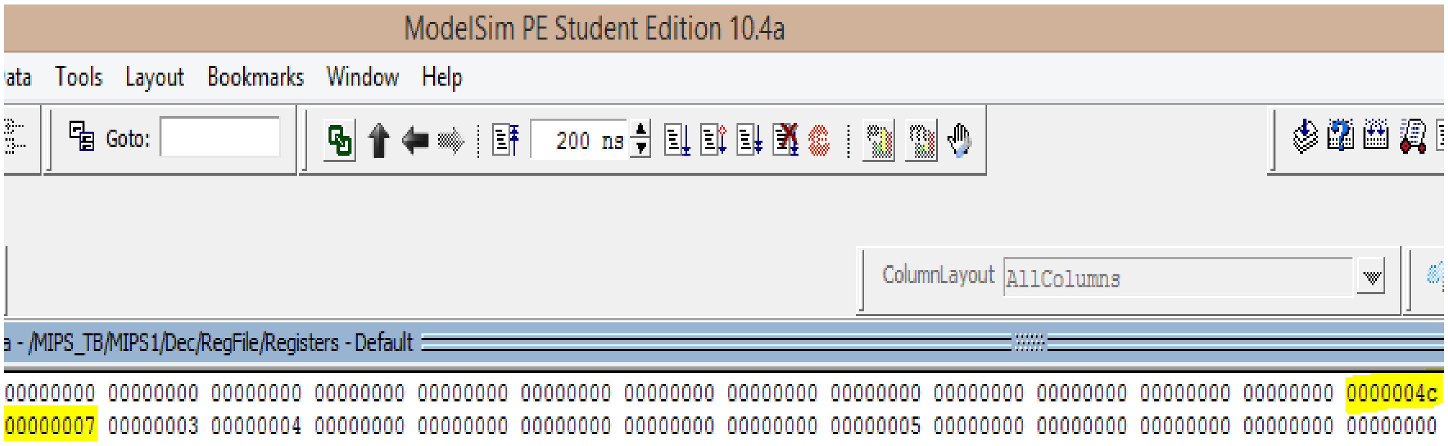


```
Registers[8]  <= 0; /* $t0 */          memory[0] = 0
Registers[9]  <= 0; /* $t1 */          memory[1] = 0
Registers[10] <= 0; /* $t2 */          memory[2] = 0
Registers[11] <= 0;/* $t3 */           memory[3] = 0
Registers[12] <= 0; /* $t4 */          memory[4] = 0
Registers[13] <= 0; /* t5 */           memory[5] = 76
Registers[14] <= 0;                    memory[6] = 0
Registers[15] <= 0;                    memory[7] = 0
Registers[16] <= 0; /* $s0 */          memory[8] = 0
Registers[17] <= 3; /* $s1 */          memory[9] = 0
Registers[18] <= 4; /* $s2 */          memory[10] = 0
Registers[19] <= 0; /* $s3 */          memory[11] = 0
Registers[20] <= 0; /* $s4 */          memory[12] = 0
Registers[21] <= 0;/* $s5 */           memory[13] = 0
Registers[22] <= 0; /* s6 */           memory[14] = 0
Registers[23] <= 0; /* s7 */           memory[15] = 0
Registers[24] <= 5;/* $t8 */           memory[16] = 0
Registers[25] <= 0;                    memory[20] = 0
```

Expected outputs :

t5 = 76, s0=7

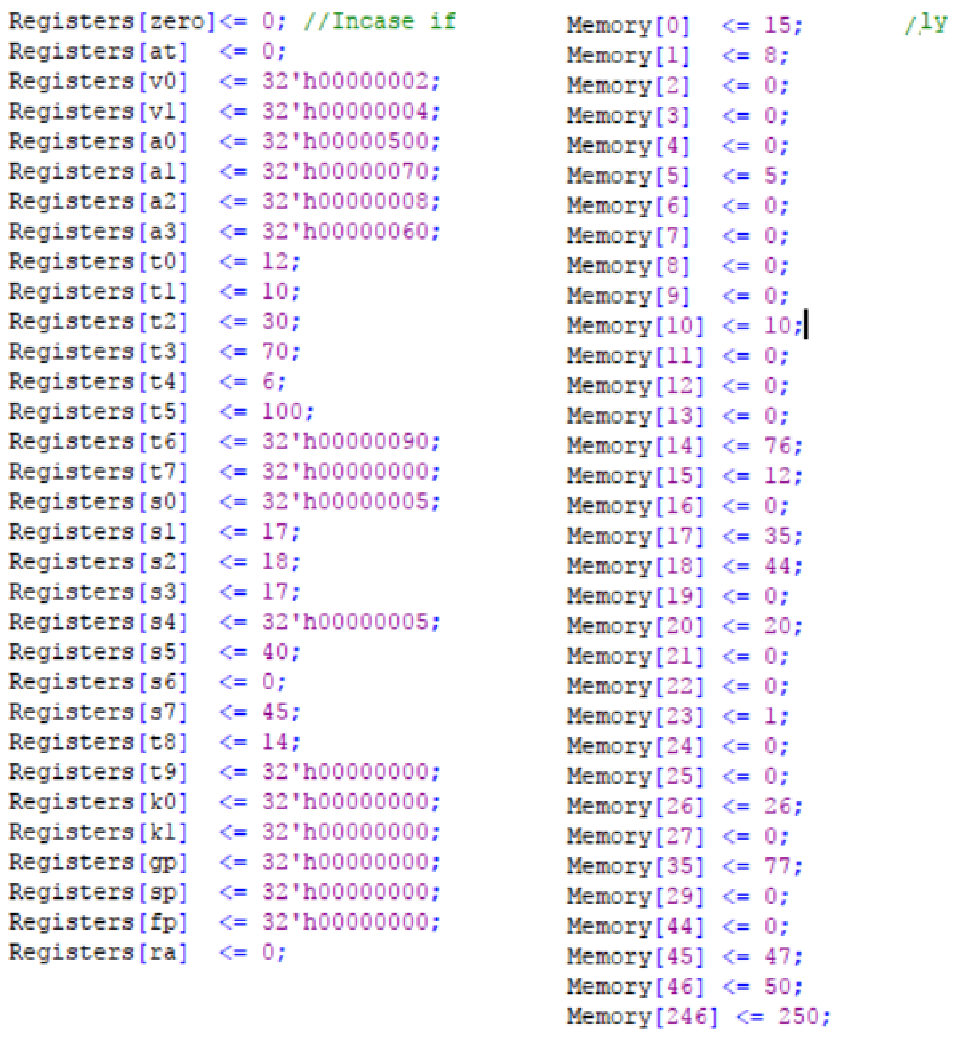
Actual outputs :


ModelSim PE Student Edition 10.4a
ata Tools Layout Bookmarks Window Help
Goto:
200 ns
ColumnLayout AllColumns
a - /MIPS_TB/MIPS1/Dec/RegFile/Registers - Default

```
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 0000004c
00000007 00000003 00000004 00000000 00000000 00000000 00000000 00000000 00000005 00000000 00000000 00000000 00000000 00000000
```

1 – General Test with (or,ori,xor) :

1) Assembly Format	**Binary Format**
xor $s0,$s1,$s2	00000010001100101000000000100110
ori $t1,$t2,15	00110101010010010000000000001111
or $t3,$t4,$t5	00000001100011010101100000100101
lw $s3,4($t8)	10001111000100110000000000000100
and $s6,$s3,$s4	00000010011101001011000000100100
add $s6,$zero,$s6	00000000000101101011000000100000



```
Registers[zero]<= 0; //Incase if
Registers[at]  <= 0;
Registers[v0]  <= 32'h00000002;
Registers[v1]  <= 32'h00000004;
Registers[a0]  <= 32'h00000500;
Registers[a1]  <= 32'h00000070;
Registers[a2]  <= 32'h00000008;
Registers[a3]  <= 32'h00000060;
Registers[t0]  <= 12;
Registers[t1]  <= 10;
Registers[t2]  <= 30;
Registers[t3]  <= 70;
Registers[t4]  <= 6;
Registers[t5]  <= 100;
Registers[t6]  <= 32'h00000090;
Registers[t7]  <= 32'h00000000;
Registers[s0]  <= 32'h00000005;
Registers[s1]  <= 17;
Registers[s2]  <= 18;
Registers[s3]  <= 17;
Registers[s4]  <= 32'h00000005;
Registers[s5]  <= 40;
Registers[s6]  <= 0;
Registers[s7]  <= 45;
Registers[t8]  <= 14;
Registers[t9]  <= 32'h00000000;
Registers[k0]  <= 32'h00000000;
Registers[k1]  <= 32'h00000000;
Registers[gp]  <= 32'h00000000;
Registers[sp]  <= 32'h00000000;
Registers[fp]  <= 32'h00000000;
Registers[ra]  <= 0;
```

```
Memory[0]   <= 15;        /ly
Memory[1]   <= 8;
Memory[2]   <= 0;
Memory[3]   <= 0;
Memory[4]   <= 0;
Memory[5]   <= 5;
Memory[6]   <= 0;
Memory[7]   <= 0;
Memory[8]   <= 0;
Memory[9]   <= 0;
Memory[10]  <= 10;
Memory[11]  <= 0;
Memory[12]  <= 0;
Memory[13]  <= 0;
Memory[14]  <= 76;
Memory[15]  <= 12;
Memory[16]  <= 0;
Memory[17]  <= 35;
Memory[18]  <= 44;
Memory[19]  <= 0;
Memory[20]  <= 20;
Memory[21]  <= 0;
Memory[22]  <= 0;
Memory[23]  <= 1;
Memory[24]  <= 0;
Memory[25]  <= 0;
Memory[26]  <= 26;
Memory[27]  <= 0;
Memory[35]  <= 77;
Memory[29]  <= 0;
Memory[44]  <= 0;
Memory[45]  <= 47;
Memory[46]  <= 50;
Memory[246] <= 250;
```

Expected outputs :

S0=3

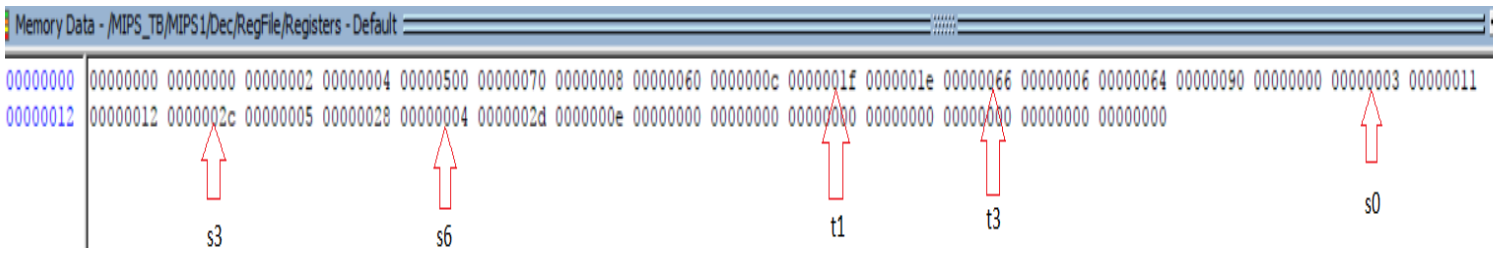
T1=31

T3=102

S3=44

S6=4

Actual outputs :

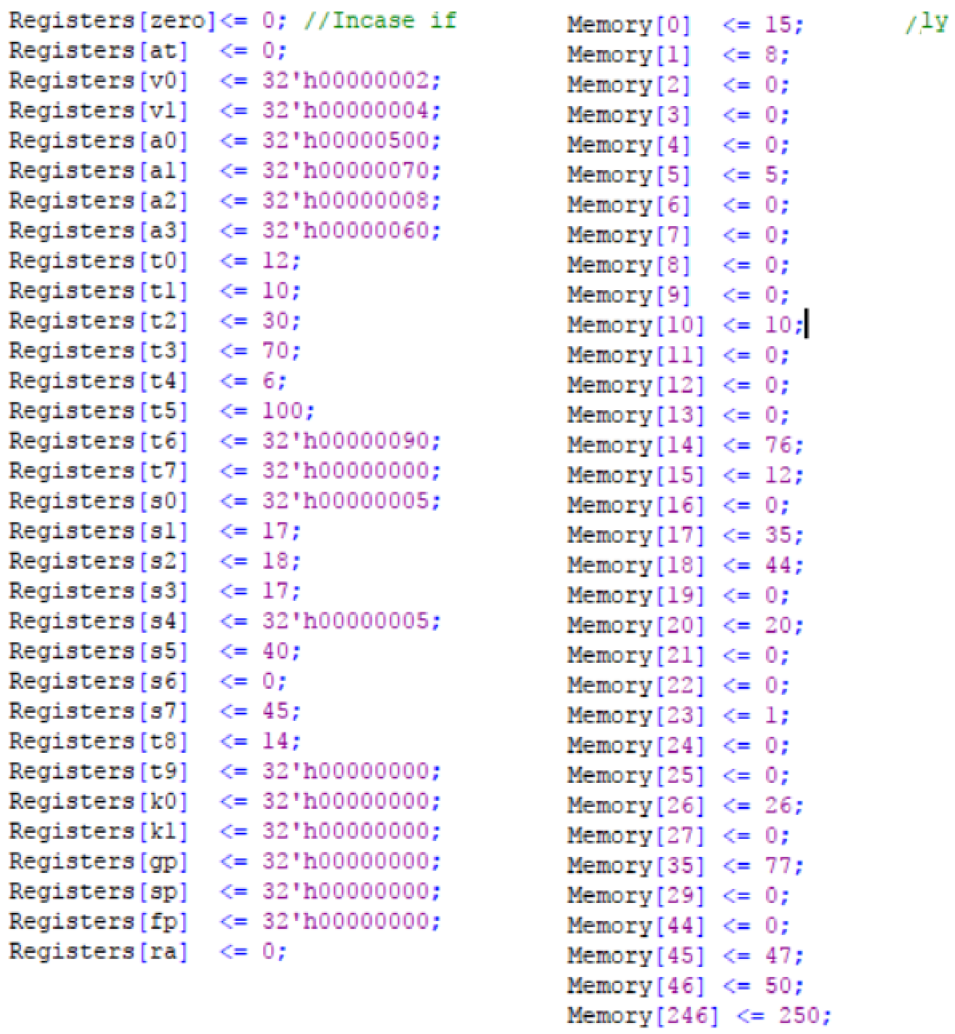


Memory Data - /MIPS_TB/MIPS1/Dec/RegFile/Registers - Default

```
00000000 00000000 00000000 00000002 00000004 00000500 00000070 00000008 00000060 0000000c 0000001f 0000001e 00000066 00000006 00000064 00000090 00000000 00000003 00000011
00000012 00000012 0000002c 00000005 00000028 00000004 0000002d 0000000e 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

4 – General Test with (or,ori,xor) :

1) Assembly Format	Binary Format
add $s1, $s2, $s3	00000010010100111000100000100000
add $s1, $s1, $s4	00000010001101001000100000100000
sub $s5, $s5, $s1	00000010101100011010100000100010



```
Registers[zero]<= 0; //Incase if
Registers[at]   <= 0;
Registers[v0]   <= 32'h00000002;
Registers[v1]   <= 32'h00000004;
Registers[a0]   <= 32'h00000500;
Registers[a1]   <= 32'h00000070;
Registers[a2]   <= 32'h00000008;
Registers[a3]   <= 32'h00000060;
Registers[t0]   <= 12;
Registers[t1]   <= 10;
Registers[t2]   <= 30;
Registers[t3]   <= 70;
Registers[t4]   <= 6;
Registers[t5]   <= 100;
Registers[t6]   <= 32'h00000090;
Registers[t7]   <= 32'h00000000;
Registers[s0]   <= 32'h00000005;
Registers[s1]   <= 17;
Registers[s2]   <= 18;
Registers[s3]   <= 17;
Registers[s4]   <= 32'h00000005;
Registers[s5]   <= 40;
Registers[s6]   <= 0;
Registers[s7]   <= 45;
Registers[t8]   <= 14;
Registers[t9]   <= 32'h00000000;
Registers[k0]   <= 32'h00000000;
Registers[k1]   <= 32'h00000000;
Registers[gp]   <= 32'h00000000;
Registers[sp]   <= 32'h00000000;
Registers[fp]   <= 32'h00000000;
Registers[ra]   <= 0;
```

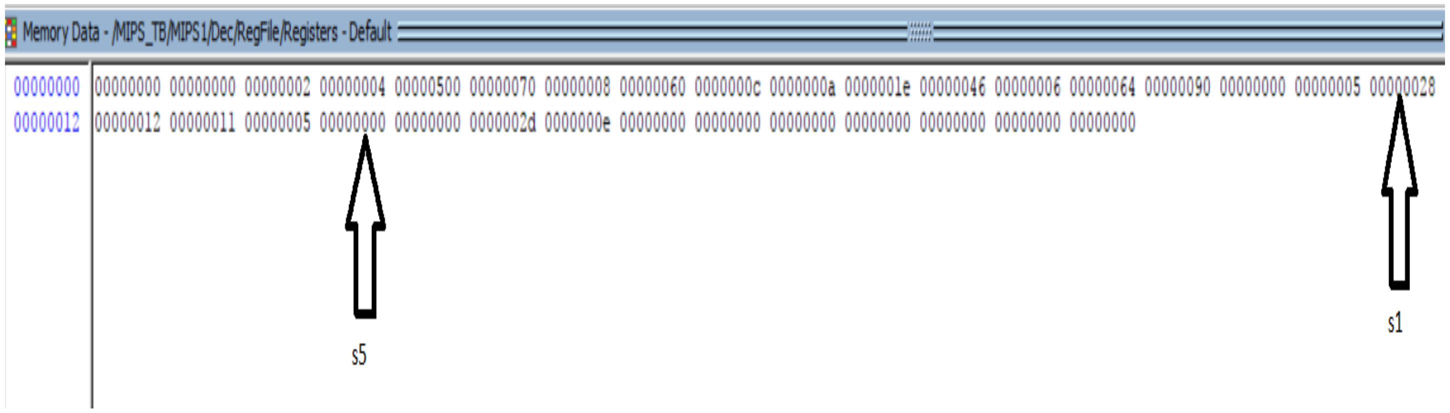
```
Memory[0]   <= 15;        /ly
Memory[1]   <= 8;
Memory[2]   <= 0;
Memory[3]   <= 0;
Memory[4]   <= 0;
Memory[5]   <= 5;
Memory[6]   <= 0;
Memory[7]   <= 0;
Memory[8]   <= 0;
Memory[9]   <= 0;
Memory[10]  <= 10;
Memory[11]  <= 0;
Memory[12]  <= 0;
Memory[13]  <= 0;
Memory[14]  <= 76;
Memory[15]  <= 12;
Memory[16]  <= 0;
Memory[17]  <= 35;
Memory[18]  <= 44;
Memory[19]  <= 0;
Memory[20]  <= 20;
Memory[21]  <= 0;
Memory[22]  <= 0;
Memory[23]  <= 1;
Memory[24]  <= 0;
Memory[25]  <= 0;
Memory[26]  <= 26;
Memory[27]  <= 0;
Memory[35]  <= 77;
Memory[29]  <= 0;
Memory[44]  <= 0;
Memory[45]  <= 47;
Memory[46]  <= 50;
Memory[246] <= 250;
```

Expected outputs :

S1=40

S5=0

Actual outputs :



Memory Data - /MIPS_TB/MIPS1/Dec/RegFile/Registers - Default

```
00000000 00000000 00000000 00000002 00000004 00000500 00000070 00000008 00000060 0000000c 0000000a 0000001e 00000046 00000006 00000064 00000090 00000000 00000005 00000028
00000012 00000012 00000011 00000005 00000000 00000000 0000002d 0000000e 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```