ITU
Computer Engineering Department
Prof. Dr. Emre HARMANCI
Asoc. Prof. Dr. Feza BUZLUCA

BLG 322E COMPUTER ARCHITECTURE
HOMEWORK 2 SOLUTIONS

a-)

```
1                .data
2   A:           .word   44      ! First multiplier
3   B:           .word   03      ! Second multiplier
4   R:           .word   900     ! Result address
5
6                .text
7   start:       sethi   A       %r1
8                sethi   B       %r2
9                sethi   0       %r5
10  loop:        and     %r1     01      %r3     ! Check if the first multiplier is odd.
11               be      skip    ! If the first multiplier is even, skip the adding operation.
12               add     %r5     %r2     %r5     ! Else add the second multiplier to the result.
13  skip:        sll     %r2     1       %r2     ! Multiply the second multiplier by two
14               srl     %r1     1       %r1     ! Divide the first multiplier by two
15               bne     loop    ! If the first multiplier is not zero, repeat the process
16               sethi   R       %r4
17               st      %r5     [%r4]   ! Else, store the results in the memory cell at address 900.
18  end:         ta      0
```

When using pipelining, there can be two pipeline hazards. First hazard is at line 12. Addition operation is done not depending upon the branch condition. The other hazard is at line 16. The program stores R value into R4 for each iteration.

b-)

```
1                .data
2   A:           .word   44
3   B:           .word   03
4   R:           .word   900
5
6                .text
7   start:       sethi   A       %r1
8                sethi   B       %r2
9                sethi   0       %r5
10  loop:        and     %r1     01      %r3
11               be      skip
```

```
12          nop
13          add     %r5     %r2     %r5
14  skip:   sll     %r2     1       %r2
15          srl     %r1     1       %r1
16          bne     loop
17          nop
18          sethi   R       %r4
19          st      %r5     [%r4]
20  end:    ta      0
```

c-)

```
1           .data
2   A:      .word   44
3   B:      .word   03
4   R:      .word   900
5
6           .text
7   start:  sethi   A       %r1
8           sethi   B       %r2
9           sethi   0       %r5
10  loop:   and     %r1     01      %r3
11          be      skip
12          srl     %r1     1       %r1     ! M1 should always be divided by 2 in each iteration.
    We can insert this instruction after the conditional branch.
13          add     %r5     %r2     %r5     ! The program adds the second multiplier to the result
    only if the branch is not taken.
14  skip:   bne     loop    ! This branch is used for controlling if the multiplication is complete.
    The multiplication is complete if the first or second multiplier is zero.
15          sll     %r2     1       %r2     ! M2 should always be divided by 2 in each iteration.
16          sethi   R       %r4
17          st      %r5     [%r4]
18  end:    ta      0
```