

Number: Name :

Q1	Q2	Q3	Q4	Q5	Q6	Total

Q1) [20pts] Consider the following MIPS code segment. Note: values of a0 and a1 are passed from the calling function and the result is returned in v0.

```

f1 :
    lw $t0, 0($a0)
    addi $t1, $0, 1
loop:
    bge $t1, $a1, exit
    mul $t2, $t1, 4
    add $t2, $t2, $a0
    lw $t2, 0($t2)
    ble $t2, $t0, next
    add $t0, $t2, $0

next:
    addi $t1, $t1, 1
    j loop
exit:
    add $v0, $t0, $0
    jr $ra

```

a) Translate the function f1 into a C code [use variable names as register names].

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

b) Describe briefly what the function f1 perform?

.....

Q2) [6pts] For multi-Cycle CPU, decide which of the following is true or false:

- a) Faster instructions are not held back by slower ones.
- b) We don't have to duplicate any hardware units.
- c) The cycle time is limited by the slowest functional unit.

Q3) [4pts] For a multi-cycle processor, consider the following code segment:

```
lw    $t2, 0($t3)
lw    $t3, 4($t3)
beq   $t2, $t3, Label
add   $t5, $t2, $t3
sw    $t5, 8($t3)
```

Label: ...

- a) What is going on during the 8th cycle of execution?
- b) In what cycle is the branch target address is calculated?

Q4) [20pts] Assume that the following MIPS program is run on a 500MHz processor, with a clock cycle time of 2ns. The number of clocks per instruction is shown in the table. Let \$a1=5, calculate the total CPU time required for executing this function.

```
func:lw $t0,0($a0)
      addi $t1,$0,1
loop:bge $t1,$a1,exit
      mul $t2,$t1,4
      add $t2,$t2,$a0
      lw $t2,0($t2)
      add $t0,$t2,$0
      addi $t1,$t1,1
      j loop
exit:add $v0,$t0,$0
      jr $ra
```

Instruction type	Clock per instruction
add/addi	4
Mul	10
Load	5
branch, jump	3

The total number of cycles is

The total CPU time is

Q6) [25pts] Assume that it is required to add the following instruction

MemIndAdd rt,offset(rs)

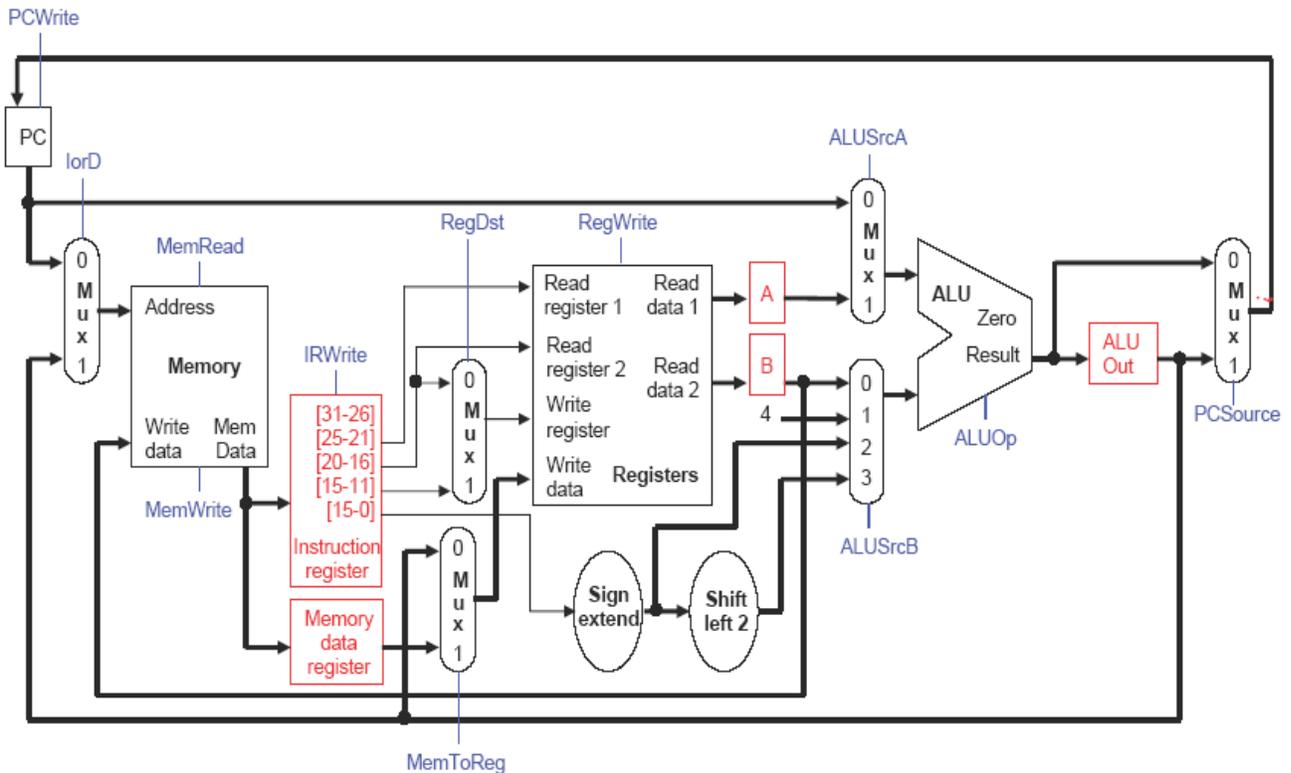
to the multicycle data-path shown below. This instruction employs the following operations:

tmp=memory[offset+rs]

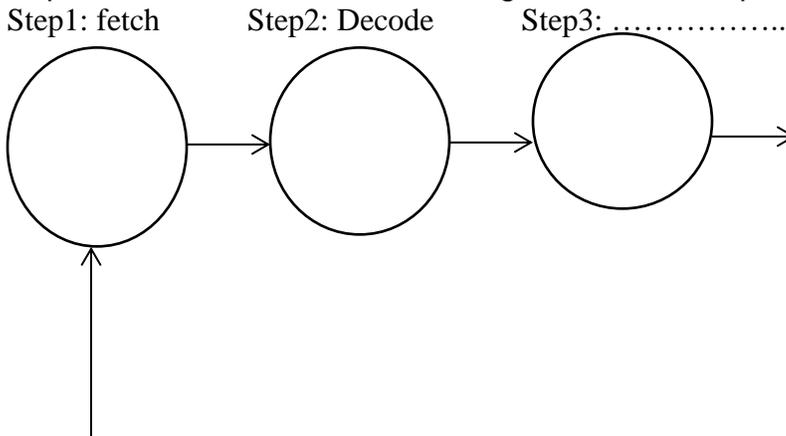
tmp=memory[tmp]

rt=rt+tmp

a) **[10pts]** Add **clearly** any necessary data-paths and justify the need for the modifications, if any.



b) **[10pts]** Provide the finite state diagram for executing this instruction. Specify the required control lines values starting from the 3rd step.



c) **[5pts]** how many cycles required for executing this instruction.