**Appendix B - Input Samples**

*An Example C program:*

sum = 0;

for( i = 0; i < 3; i++ ) {

sum += i;

}

Suppose that

1. The sizes of *sum* & *i* are words.
2. The address of *sum* is located at 0x00000000 in D memory, while the address of *i* is at 0x00000008 in D memory.
3. PC is initially 0, and $sp is initially 0x400.

*Translate into assembly*:

andi $t0, $0, 0 # sum = $t0 = 0

andi $t1, $0, 0 # i = $t1 = 0

loop: slti $t2, $t1, 3 # $t2 = ( i < 3 )

beq $t2, $0, end # if (i >= 3), go to end

add $t0, $t0, $t1 # sum = sum + i

addi $t1, $t1, 1 # i++

j loop # jump to loop

end: sw $t0, 0($0) # store sum

halt

halt

halt

halt

halt

Then, this program will be provided as the following binary contents. Note that no comments will be in official input files; they are here to help you understand their meanings. Additionally, the binary content represent only hexadecimal value of each word respectively, there are nothing to do with little-endian or big-endian.

*iimage.bin:*

0x00000000 # initial value of PC

0x0000000D # number of words to be loaded into I memory

0x30080000 # contents of I memory begins

0x30090000

0x292A0003

0x11400003

0x01094020

0x21290001

0x08000002

0xAC080000

0xFFFFFFFF

0xFFFFFFFF

0xFFFFFFFF

0xFFFFFFFF

0xFFFFFFFF

*dimage.bin:*

0x00000400 # initial value of $sp

0x00000003 # number of words to be loaded into D memory

0x12345678 # content of D memory begins

0x9ABCDEF0

0x13572468