CS 506

HOMEWORK ASSIGNMENT 4

Chapter 2 (Morgan Kauffman book)

**2.1** [5] <§2.2> For the following C statement, what is the corresponding MIPS

assembly code? Assume that the variables f, g, h, and i are given and could be

considered 32-bit integers as declared in a C program. Use a minimal number of

MIPS assembly instructions.

f = g + (h − 5);

# assuming f, g, h, i are $t0, $t1, $t2 and $t3

# f = (g - h) + (i - 4)

sub $t4, $t1, $t2 # $t4 = g - h

addi $t5, $t3, -4 # $t5 = i - 4

add $t0, $t4, $t5 # f = (g - h) + (i - 4

**2.2** [5] <§2.2> For the following MIPS assembly instructions above, what is a

corresponding C statement?

add f, g, h

add f, i, f

The C statement is:

In C adding of two variable and store in third variable is written as:

x= y+z;

and corresponding MIPS instruction is :

add x , y, z means x is the sum of y and z

So according to the given instructions :

Statement 1: add f, g, h

C statement is :  f= g+ h

Statement 2 : add f, i, f

C statement is : f= i+ f

put the value of f from the statement 1 at right side

f= i + g + h

C statement is f= i + g + h

**2.3** [5] <§§2.2, 2.3> For the following C statement, what is the corresponding

MIPS assembly code? Assume that the variables f, g, h, i, and j are assigned to

registers $s0, $s1, $s2, $s3, and $s4, respectively. Assume that the base address

of the arrays A and B are in registers $s6 and $s7, respectively.

B[8] = A[i−j];

#i is in $s3  
#j is in $s4

#get value of i-j  
sub $t0,$s3,$s4

#get address of i-j  
mul $t0,$t0,4

#get address of A[i-j]  
add $t0,$t0,$s6

#load value of A[i-j]  
lw $t0,($t0)

#save value of A[i-j] at location B[12]  
sw $t0,48($s7)

**2.8** [5] <§2.4> Translate 0xabcdef12 into decimal. (Show the work and apply any method you like)

The digits of a hexadecimal number are

1,2,3,4,5,6,7,8,9,a,b,c,d,e,f

where a=10,b=11,c=12,d=13,e=14,f=15

=10\*16^7+11\*16^6+12\*16^5+13\*16^4+14\*16^3+13\*16^2+1\*16^1+2\*16^0

**2.9** [5] <§§2.2, 2.3> Translate the following C code to MIPS. Assume that the

variables f, g, h, i, and j are assigned to registers $s0, $s1, $s2, $s3, and $s4,

respectively. Assume that the base address of the arrays A and B are in registers $s6

and $s7, respectively. Assume that the elements of the arrays A and B are 4-byte

words:

B[8] = A[i] + A[j];

sll $t0, $s3, 2      # $t0 = 4\* i as its 4 byte word

sll $t1, $s4, 2       # $t1 = 4\*j, as its 4 byte word

add $t0, $t0, $s6     # A address is in s6 so Aij]

add $t1, $t1, $s6 # A address is in s6 so A[j]

lw $t0, 0($t0)     # Load A[i] into , $t0 = A[i]

lw $t1, 0($t1). # Load A[j] into , $t1= A[j]

add $t0, $t1, $t0 # $t0=A[i]+A[j]

addi $t1, $s7, 32   # 4\*8 is 32 as its 4 bytes word i.e and s7 has B, so t1 has address of B[8]

sw $t0, 0($t1) # B[8]=A[i]+A[j]