Ch2 questions and answers

2.6 assume f,g,h,I,j assigned to s0,s1,s2,s3 and s4. Array B base in s7

a- f=-g+h+B[1];

2.6.1 What is MIPS assembly

Sub s0, s2, s1; f= h-g

Lw t0, 4(s7) ; t0= B[1]

Add s0, s0, t0; f=h-g+B[1]

2.6.2 How many instructions

3

2.6.3 how many registers used

5

Translate to C the following assuming f,g,h,I,j assigned to s0,s1,s2,s3 and s4. Array A base in s6

b-Addi s6, s6, -20

Add s6, s6, s1

Lw s0, 8(s6)

2.6.4 f= A[(g-20 +8)/4]

2.6.5 if g=s1=20 , then s0=f=A[(20-20+8)/4]= A[2]=300

2.6.6 I type lw s0 , 8(s6) ; lw=35, s6-=22, s0=16, imm=8= 100011, 10110,10000,0000000000001000

R type add s6,s6,s1= add=0, s6=22, s1=17,s6=22,shmt=0, func=32

2.12- assume MIPS uses 8 registers, 10 bit immediate

2.12.1 size of R type

Opcode=6, 3xlog2(8), shmt=5, funct=6 total=26 bit

2.12.2 I type= opcode=6, 2xlog2(8), imm=10 total=22 bit

2.12.3- why change reduces program size?

Because size of instructions is smaller

Why change increase program size

Because you need more instructions to load, store from memory due to small number of registers (register spills)

2.12.5 what is the following instruction given it is in hex=

b-0x8D090012

inst in binary= 1000 1101 0000 1001 0000 0000 0001 0010

opcode=100011= LW, 01000 rs= t0, rt=01001=t1, offset=18; lw t1, 18(t0)

2.18. Assume the following C:

```
b- while(a<10){
            D[a] = b + a;
           A +=1;
                       }
```

Assume s0=a, s1=b, s2 address of D[0]

```
Loop: slt t0, s0, 10;   is a<10
          beq t0, $0, Exit
          add t1, s0, s1;            t0 =a + b
          sll s3, s0, 2;                 4a to point to D[a]
          add s3, s3, s2;               address of D[a]
            sw t1, 0(s3);                  D[a]= a+b
            addi s0, s0, 1 ;               a=a+1
           j Loop;
Exit
```

2.18.3 because a !< 10, it will Exit

2 instructions executed first and second,, total number of instructions=8

2.18.4 assume following MIPS :

```
a-          addi t1, $0, 100;                       i=t1=100
     Loop: lw  s1, 0(s0)                            s1=Mem[s0/4]
               add s2, s2, s1;                       s2=result=result + Mem[s0/4]
                addi s0, s0,4;                     s0 = s0 + 4 to inc Mem array by 1
```

```
        subi t1, t1, 1 ;                  i=i-1
         bne t1, $0, Loop;                is I=0 if not go to Loop
```

2.18.4 total number of MIPS executed 1 + 100X5= 501

```
2.18.5 for(i=100; i!=0; i--){
    result= result + Mem[s0];
     s0=s0+1;
                          }
```

2.24. assume t1 has address 0x1000 0000 and t2 has address 0x1000 0010

```
a- lb t0, 0(t1);           t0= content of LSB at 0x1000 000 address
   sw t0, 0(t2)            memory of location     0x1000 0010 has LSB
```

2.24.1- If address of t2 has 0x FFFF FFFF after executing a it will have

0000 0012

2.24.2 it will have 0x 00000080

2.27.3.a assume

```
      Lui s0, 100;           load register upper 16 bit by 100 or 0x0064
      Ori s0, s0, 40          or content of s0 with 40 or 0x0028
```

Result in s0= 0x00640028 32 bit imm load

2.30. how to implement move t1, t2 in MIPS

```
    Add t1, t2, $0
```