CSE 3038 – Computer Organization **Spring 2022**
Prof. Haluk Topcuoglu

HW2 - Key

1. 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 $s2, $s3, $s4, $s5, and $s6, respectively. Assume that the base address of the arrays A and B are in registers $s0 and $s1, respectively.

B[i+3] = A[i+4*j];

Answer:

```
sll $t0, $s6, 2              # $t0 = 4*j
add $t0, $s5, $t0            # $t0 = 4*j + i
sll $t0, $t0, 2              # $t0 = t0*4
add $t0, $t0, $s0            # $t0 = &A[i+4*j]
lw $t1, 0($t0)               # $t1 = A[i+4*j]
addi $t2,$s5,3               # $t2 = i+3
sll $t2,$t2,2                # $t2 = t2*4
add $t2,$t2,$s1              # $t2 = &B[i+3]
sw $t1, 0($t2)               # B[i+3] = $t1
```

2. Show how the value 0xcabd1f2e would be arranged in memory of a little-endian and a big-endian machine. Assume the data is stored starting at address 0.

Answer:

Little Endian

Address	Data
0	2e
1	1f
2	bd
3	ca

Big Endian

Address	Data
0	ca
1	bd
2	1f
3	2e

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[i+j+1] = A[i+j-2] + A[i-j+1];

Answer:

```
sub $t0, $s3, $s4            # $t0 = i - j
add $t1, $s3, $s4            # $t1 = i + j
```

```
addi $t0,$t0,1              # $t0 = i – j+1
addi $t2,$t1,1              # $t2 = i + j + 1
addi $t1,$t1,-2             # $t1 = i + j-2
sll $t0, $t0, 2             # $t0 = (i-j+1)*4
sll $t1, $t1, 2             # $t1 = (i+j-2)*4
add $t3, $s6, $t0           # $t3 = &A[i-j+1]
add $t4, $s6, $t1           # $t4 = &A[i+j-2]
lw $t5, 0($t3)              # $t5 = A[i-j+1]
lw $t6, 0($t4)              # $t6 = A[i+j-2]
add $t7, $t5, $t6           # $t7 = A[i+j-2] + A[i-j+1]
sll $t2,$t2,2               # $t2 = (i+j+1)*4
add $t2,$t2,$s7             # $t2 = &B[i+j+1]
sw $t7, 0($t2)              # B[i+j+1] = $t7
```

4. Provide the type, assembly language instruction, and binary representation of instruction described by the following MIPS fields:
op=0, rs=5, rt=8, rd=20, shamt=0, funct=36.

Answer:
Its R type instruction
Op=0 funct=36 ➔ and

and $20, $5, $8 or
and $s4, $a1, $t0

5. For the following C statement, write a minimal sequence of MIPS assembly instructions that does the identical operation. Assume $t1 = A, $t2 = B, and $s1 is the base address of C.

```
A = C[0] << 8;
```

Answer:

```
lw $t3, 0($s1)
sll $t1, $t3, 8
```

6. Translate the following C code to MIPS assembly code. Use a minimum number of instructions. Assume that the values of a, b, i, and j are in registers $s0, $s1, $t0, and $t1, respectively. Also, assume that register $s2 holds the base address of the array D.

```
for(i=0; i<a; i++)
        for(j=0; j<b; j++)
                D[2*i] = i + j-5;
```

Answer:

```
addi $t0, $0, 0
beq $0, $0, TEST1
LOOP1: addi $t1, $0, 0
```

```
beq $0, $0, TEST2
LOOP2: add $t3, $t0, $t1
sub $t3,$t3,5
sll $t2, $t0, 3
add $t2, $t2, $s2
sw $t3, ($t2)
addi $t1, $t1, 1
TEST2: slt $t2, $t1, $s1
bne $t2, $0, LOOP2
addi $t0, $t0, 1
TEST1: slt $t2, $t0, $s0
bne $t2, $0, LOOP1
```

7. How many MIPS instructions does it take to implement the C code given in the previous question (Question 6)? If the variables a and b are initialized to 10 and 1 and all elements of D are initially 0, what is the total number of MIPS instructions that is executed to complete the loop?

Answer:

14 instructions to implement and 144 instructions executed

CSE 3038 – Computer Organization **Spring 2022**
Prof. Haluk Topcuoglu

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1 addi $t0, $0, 0	*																															
2 beq $0, $0, TEST1		*																														
3 LOOP1: addi $t1, $0, 0					*														*													
4 beq $0, $0, TEST2						*														*												
5 LOOP2: add $t3, $t0, $t1									*														*									
6 sll $t2, $t1, 4										*														*								
7 add $t2, $t2, $s2											*														*							
8 sw $t3, ($t2)												*														*						
9 addi $t1, $t1, 1													*														*					
10 TEST2: slt $t2, $t1, $s1							*							*							*							*				
11 bne $t2, $0, LOOP2								*							*							*							*			
12 addi $t0, $t0, 1																*														*		
13 TEST1: slt $t2, $t0, $s0			*														*														*	
14 bne $t2, $0, LOOP1				*														*														*

Inner loop will execute ones since b is 1

Single outer loop takes 14 instructions, 10x14 = 140

Remaining instructions are instruction 1,2,13,14 outside of the loop pattern. Instruction 1 and 2 will be executed at the beginning of the program and instruction 13 and 14 after the last loop iteration.

Total instruction execution:140+4=144