Tutorial Sheet ODD SEM 2020
Computer Organization and Architecture (15B11CI313)

5th Sem CSE
Jaypee Institute of Information Technology, Noida

Instruction for Tutorials

1. **Tutorial has to be submitted (Turnin) within three days of tutorial class conducted.**
2. **Graded tutorial sheet will be returned within week of conduct of tutorial class by the concerned teacher.**
3. **All tutorial submission has to be handwritten with your name, enrollment and batch compulsory on it.**
4. **End of the semester COA project will be evaluated in tutorial class**
5. **Attendances are compulsory for all tutorial classes**

ODD

Tutorial-10

Tutorial on MIPS

Q.1 Answer the following questions with proper justification

a) For R-Type instructions, you need to increase the size of rs, rt and rd bit field to 7 bits. There are more instructions, you need to expand opcode and (or) funct fields to encode them. Justify with valid reason that why your new format is enough for the increased number of instructions.

b) If the instruction set of the MIPS processor is modified, the instruction format must also be changed. For each of the suggested changes, show the size of the bit_fields of an R-type format instruction. What is the total number of bits needed for each instruction?

(i) 8 Registers
(ii) 10 bit immediate constants
(iii) 128 Registers
(iv) All arithmetic instructions can use base addressing mode with the last argument (Example: add $a0, $a2, 0[$t1])

Why could the suggested change decrease or increase the size of a MIPS assembly program? Justify your answer with valid reason.

Q2. Assume that the code below is run on a machine with a 2 GHz clock that requires the following number of cycles for each instruction: In the worst case, how many seconds will it take to execute this code?

Instruction	**no of cycles**
add,addi,sll	4
lw,	5
bne	3

```
#Code
sll $a2, $a2, 2
sll $a3, $a3, 2
add $v0, $zero, $zero
add $t0, $zero, $zero
outer:    add $t4, $a0, $t0
lw $t4, 0($t4)
add $t1, $zero, $zero
inner:    add $t3, $a1, $t1
lw $t3, 0($t3)
bne $t3, $t4, skip
addi $v0, $v0, 1
skip:     addi$ t1, $t1, 4
bne $t1, $a3, inner
addi $t0, $t0, 4
bne $t0, $a2, outer
```

Q3. Show the single MIPS instruction or minimal sequence of instructions for this C statement:

b = 25 | a;
Assume that a corresponds to register $t0 and b corresponds to register $t1.

Q4. Given your understanding of PC-relative addressing, explain why an assembler might have problems directly implementing the branch instruction in the following code sequence:
here: blt $s0, $s2, there
…
there add $s0, $s0, $s0

Q5: Write a MIPS assembly language **subroutine** to compute GCD of two numbers [hint: int gcd(int x, int y) { if (y == 0) return x; else return gcd(y, x % y); }]. Use stack pointer and recursive call to implement this.

Hint: // $sp is the stack pointer, Use Recursive code below.

```
 push $ra
 addi $sp, $sp,    4
 pop $ra
 lw $ra, 0($sp)
 sw $ra, 0($sp)
 addi $sp, $sp, 4
// jal gcd
//jr $ra.
```

Q6. Implement register indirect conditional branches (beqr and bner) as pseudo-instructions. Give a proposal for adding them to the ISA (i.e., describe how they could be encoded in the I-Type, R-Type, or J-Type format, and propose an opcode for them. Give a (brief) argument for or against their inclusion.

Q7. We wish to add the instruction jalr (jump and link register) to the single-cycle datapath discussed in the lecture. Add any necessary datapath and draw the result datapath.

The jump and link register instruction is described below:

jalr rd, rs # rd = pc + 4 , pc = rs

$op^6 = 0$	rs^5	0	rd^5	0	$func^6 = 9$

Q8: We want to compare the performance of a single-cycle CPU design with a multicycle CPU. Suppose we add the multiply and divide instructions. The operation times are as follows:
Instruction memory access time = 190 ps, Data memory access time = 190 ps
Register file read access time = 150 ps, Register file write access = 150 ps ,
ALU delay for basic instructions = 190 ps, ALU delay for multiply or divide = 550 ps
Ignore the other delays in the multiplexers, control unit, sign-extension, etc.
Assume the following instruction mix: 30% ALU, 15% multiply & divide, 20% load, 10% store, 15% branch, and 10% jump.

a) What is the total delay for each instruction class and the clock cycle for the single cycle CPU design.
b) Assume we fix the clock cycle to 200 ps for a multi-cycle CPU, what is the CPI for each instruction class and the speedup over a fixed-length clock cycle?