North South University

Department of Electrical & Computer Engineering

ISA DESIGN

Course: CSE332.3

Faculty Initials: TnR

Semester: Spring'20

Group Number- 09

Group Members:

Jannatul Ferdaous	1731733642
Mahir Absar	1812777642
Deboraj Saha Dipta	1520537642
Md. Sadman-Al-Islam	1610503042

1. Our ISA will contain 3 operands (r_s,r_t,r_d)

2. The operands will be of both types.
    - Memory based
    - Register based

3. The operations we will be performing –

- Arithmetic Operation
- Logic Operation
- Shifting Operation
- Compare Operation
- Load/Store Operation
- Unconditional Operation

We want to perform these operations because we are assigning 4 bits to our opcode which means we can perform 2^4 or 16 instructions.

4. We will have total 3 formats.

R-type:

Opcode	r_s	r_t	r_d
5 bits	3 bits	3 bits	3 bits

I-type:

Opcode	r_s	r_t	Shamt/Immediate
5 bits	3 bits	3 bits	3 bits

T-type:

Opcode	Address
5 bits	9 bits

5. **Instructions List**

Type	Opcode	Name	Format Type	Example	Meaning
Arithmetic	00000	ADD	R	Add $to, $t1,t2	$to = $t1 + $t2
	00001	Sub	R	Sub $to, $t1, $t2	$to = $t1 - $t2
	00010	ADDi	I	Addi $to, $t1, [const]	$t0 = $t1 + [const]
Logic	00011	AND	R	And $to, $t1, $t2	$to = $t1&&$t2
	00100	OR	R	Or $to, $t1, $t2	$to = $t1&&$t2
	00101	NOR	R	Nor $t1,$t2,t3	$t1 = ! ($t2 |$t3)
	00110	ANDi	I	Andi $to, $t1, [1/0]	$to = $t1&&[1/0]
	00111	Ori	I	Ori $to, $t1, [1/0]	$to = $t1&&[1/0]
Memory Based	01000	Load	I	Lw $to, [offset],$t1	$to = MEM[$t1 + offset]
	01001	Store	I	Sw $to,[offset],$t1	MEM[$t1 + offset] = $t0
Conditional	01010	Beq	I	Beq $t1, $t2, L1	If ($t1 = $t2) then go to the next label
	01011	Bne	I	Bne $t1,$t2,L1	If ($t1 = $t2) then go to the L1 label
Shifting	01100	Sll	I	Sll $t1, $t2, offset	$t1 = $t2 <<offset
	01101	Srl	I	Srl $t1, $t2, offset	$t1 = $t2 >> offset
Compare	01110	Slt	R	Slt $t1, $t1, $t2	If $t1 < $t2, $t1 =1
Compare	01111	Slti	I	Slti $t1,$t2,[const]	If $t2 <[const], $t1 = 1
Unconditional	10000	Jump	T	J L1	Jumps to the location

6. **List of registers**

-

Number	Register Name	Value of Register	Purpose
0	$t0	000	Temporary register
1	$t1	001	Temporary register
2	$t2	010	Temporary register
3	$t3	011	Temporary register
4	$t4	100	Temporary register
5	$t5	101	Temporary register
6	$t6	110	Temporary register
7	$t7	111	Temporary register

7. **Addressing Modes**

Similar to MIPS addressing modes, our design will follow the addressing modes given below –

1. Register Addressing.
2. Immediate Addressing.
3. Base Addressing.
4. PC – relative addressing.
5. Pseudodirect addressing.

Limitations

1. We have assigned total 16 registers but only 4 of them will be used by the destination (r_d). We may have to overwrite the contents stored in the registers more frequently as our instruction is limited to 14bits.
2. For shift operations we went for I-type format and we can shift the value by only 4 bits which means we cannot multiply or divide more than 2^4 or 16.