

# Computer Organization and Architecture Laboratory

## Instruction Set Architecture Assignment

Sumit Kumar (22CS30056)

Aviral Singh (22CS30015)

Group 79

### 1. Format of Instruction and Encoding

#### 1.1 R type instructions:

Opcode	Source 1	Source 2	Destination	Shift Amount	Function code
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

Instruction	Usage	Opcode	Funct code
Add	add rd,rs,rt	000000	000001
Subtract	sub rd,rs,rt	000000	000010
And	and rd,rs,rt	000000	000011
Or	or rd,rs,rt	000000	000100
Xor	xor rd,rs,rt	000000	000101
Not	not rd,rs,rt	000000	000110
Shift left	sl rd,rs	000000	000111
Shift right logical	srl rd,rs	000000	001000
Shift right arithmetic	sra rd,rs	000000	001001
Increment	inc rd,rs	000000	001010
Decrement	dec rd,rs	000000	001011
Shift left Than	slt rd,rs	000000	001100
Shift Greater Than	sgt rd,rs	000000	001101
Hamming	ham rd,rs	000000	001111

## 1.2 I type instructions:

Opcode	Source	Destination	Immediate Value
6 bits	5 bits	5 bits	16 bits

Instruction	Usage	Opcode
Addl	addi rs, imm	000001
Subtractl	subi rs, imm	000010
Andl	andi rs, imm	000011
Orl	ori rs, imm	000100
Xorl	xori rs, imm	000101
Notl	noti rs, imm	000110
Shift Left Immediate	sli rs, imm	000111
Shift Right Logical Immediate	srlr rs, imm	001000
Shift Right Arithmetic Immediate	srai rs, imm	001001
Branch	br imm	001010
Branch if Minus	bmi rs, imm	001011
Branch if Plus	bpl rs, imm	001100
Branch if equal to zero	bz rs, imm	001101
Load	Ld rd, rs, imm	001110
Store	st rd, rs, imm	001111
Increment Immediate	inci rd,imm	010000
Decrement Immediate	deci rd,imm	010001
Load Upper Immediate	lui rd,imm	010001
Hamming Immediate	hamm rd,imm	010011

### 1.3 Miscellaneous Instructions:

Opcode	Don't care
6 bits	26 bits

Instruction	Usage	Opcode
Halt	halt	111110
No operation	nop	111111

## 2. Register Usage Convention

Register	Function	Register Number
\$zero	Hardwired to 0	0
\$r1-\$r15	Temporary registers	1-15
\$rt	Return Pointer	16
\$pc	Programme counter	17

### 3. Control Unit Design

Instr	opcode	func	Reg dest	Reg write	Mem Read	Mem Write	Mem To Reg	ALU src	ALU op	Branch	Jump Addr	Addr Sel	Mod SP	Inc PC
Add	000000	000001	1	1	0	0	2	0	0	0	X	X	0	1
Subtract	000000	000010	1	1	0	0	2	0	1	0	X	X	0	1
And	000000	000011	1	1	0	0	2	0	5	0	X	X	0	1
Or	000000	000100	1	1	0	0	2	0	7	0	X	X	0	1
Xor	000000	000101	1	1	0	0	2	0	8	0	X	X	0	1
Not	000000	000110	1	1	0	0	2	0	6	0	X	X	0	1
Shift left	000000	000111	1	1	0	0	2	0	3	0	X	X	0	1
Shift right logical	000000	001000	1	1	0	0	2	0	4	0	X	X	0	1
Shift right arithmetic	000000	001001	1	1	0	0	2	0	2	0	X	X	0	1
Move	000000	001010	1	1	0	0	2	0	9	0	X	X	0	1
Return	000000	001101	X	0	1	1	2	X	9	0	1	0	1	1
Addl	000001	X	0	1	0	0	2	1	0	0	X	X	0	1
Subtractl	000010	X	0	1	0	0	2	1	1	0	X	X	0	1
Andl	000011	X	0	1	0	0	2	1	5	0	X	X	0	1

Orl	000100	X	0	1	0	0	2	1	7	0	X	X	0	1
Xorl	000101	X	0	1	0	0	2	1	8	0	X	X	0	1

Notl	000110	X	0	1	0	0	2	1	6	0	X	X	0	1
Shift Left Immediat e	000111	X	0	1	0	0	2	1	3	0	X	X	0	1
Shift Right Logical Immediat e	001000	X	0	1	0	0	2	1	4	0	X	X	0	1
Shift Right Arithmeti c Immediat e	001001	X	0	1	0	0	2	1	2	0	X	X	0	1
Branch	001010	X	X	0	0	0	X	1	0	1	0	X	0	1
Branch if Minus	001011	X	X	0	0	0	X	1	0	1	0	X	0	1
Branch if Plus	001100	X	X	0	0	0	X	1	0	1	0	X	0	1
Branch if equal to zero	001101	X	X	0	0	0	X	1	0	1	0	X	0	1
Load	001110	X	0	1	1	0	1	1	0	0	X	1	0	1
Store	001111	X	X	0	0	1	X	1	0	0	X	1	0	1
Halt	111110	X	X	0	0	0	X	X	10	0	X	X	0	0
No Operatio n	111111	X	X	0	0	0	X	X	10	0	X	X	0	1

## Description of Control Signals:

- RegDst: Chooses the destination register address from either *rs* or *rt*.
- RegWrite: Enables writing to the register file.
- MemRead: Initiates a memory read operation.
- MemWrite: Initiates a memory write operation.
- MemToReg: Selects the value to write to the register.
- ALUsrc: Determines the second input for the ALU; *0* for R-type and *1* for I-type instructions.
- ALUop: Specifies the operation to be performed by the ALU.
- Branch: Indicates if the instruction is a branch instruction.
- JumpAddr: Specifies the type of address to use for jumps: *1* for stack pointer, *2* for a register value, and *0* for an immediate value.
- AddrSel: Chooses between *ALUout* and the stack pointer value for the memory address.
- Mod SP: Controls how the stack pointer is modified: *0* for no change, *1* for an increase by 4, and *2* for a decrease by 4.
- Inc PC: Decides if the program counter (PC) will be incremented: *1* for incremented, *0* for not incremented.

## DataPath

# Data Path

