Computer Organisation and Architecture Laboratory

# Specifications of Datapath and Control Unit

Group 51

Gayahtri Anant (22CS30026)

Vaasu Singla (22CS10078)

# Instructions

### R-type Instructions

| Opcode | rs | rt | rd | don't care | funct |
|---|---|---|---|---|---|
| 6 bits | 5 bits | 5 bits | 5 bits | 5 bits | 6 bits |

| **Instruction** | **Opcode** | **funct** | **Instruction_no** |
|---|---|---|---|
| ADD | 000000 | 000000 | 0 |
| SUB | 000000 | 000001 | 1 |
| AND | 000000 | 000010 | 2 |
| OR | 000000 | 000011 | 3 |
| XOR | 000000 | 000100 | 4 |
| NOR | 000000 | 000101 | 5 |
| NOT | 000000 | 000110 | 6 |
| SL | 000000 | 000111 | 7 |
| SRL | 000000 | 001000 | 8 |
| SRA | 000000 | 001001 | 9 |
| INC | 000000 | 001010 | 10 |
| DEC | 000000 | 001011 | 11 |
| SLT | 000000 | 001100 | 12 |
| SGT | 000000 | 001101 | 13 |
| HAM | 000000 | 001111 | 15 |
| MOVE | 000000 | 010000 | 16 |
| CMOVE | 000000 | 010001 | 17 |

Note: rt = 0 for MOVE instruction

### I-type instructions

| Opcode | rs | rt | imm1 |
|---|---|---|---|
| 6 bits | 5 bits | 5 bits | 16 bits |

| Instructions | Opcode | Instruction_no |
|---|---|---|
| ADDI | 110000 | 18 |
| SUBI | 110001 | 19 |
| ANDI | 110010 | 20 |
| ORI | 110011 | 21 |
| XORI | 110100 | 22 |
| NORI | 110101 | 23 |
| NOTI | 110110 | 24 |
| SLI | 110111 | 25 |
| SRLI | 111000 | 26 |
| SRAI | 111001 | 27 |
| SLTI | 111100 | 28 |
| SGTI | 111101 | 29 |
| LUI | 111110 | 30 |
| HAMI | 111111 | 31 |
| LD | 100000 | 32 |
| ST | 100001 | 33 |
| BMI | 100010 | 34 |
| BPL | 100011 | 35 |
| BZ | 100100 | 36 |

**J-type instructions**

| Opcode | Imm2 |
|---|---|
| 6 bits | 26 bits |

| Instructions | Opcode | Instruction no |
|---|---|---|

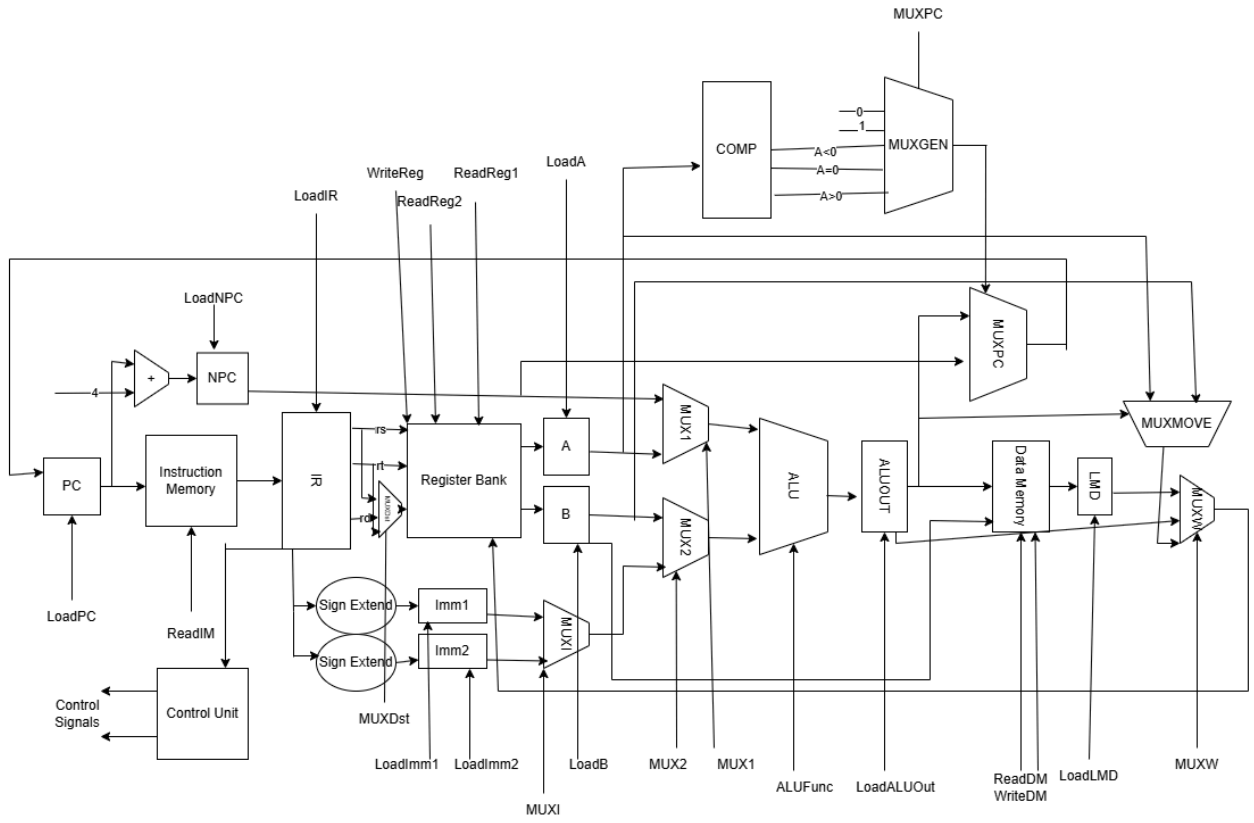
| | | |
|---|---|---|
| BR | 100101 | 37 |

**No-Field instructions**

| Opcode | Unused |
|---|---|
| 6 bits | 26 bits |

| **Instructions** | **Opcode** | **Instruction no** |
|---|---|---|
| NOP | 100110 | 38 |
| HALT | 100111 | 39 |

# Datapath



## Controls

**R-type Instructions**

| Instruction | T1 | T2 | T3 | T4 | T5 |
|---|---|---|---|---|---|
| ADD | ReadIM, LoadIR, LoadNPC | ReadReg1, ReadReg2, Load A, Load B | ALUFunc = ADD, MUX1 = 1, MUX2 = 1, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 0, WriteReg |
| SUB | ReadIM, LoadIR, LoadNPC | ReadReg1, ReadReg2, Load A, Load B | ALUFunc = SUB, MUX1 = 1, MUX2 = 1, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 0, WriteReg |
| AND | ReadIM, LoadIR, LoadNPC | ReadReg1, ReadReg2, Load A, Load B | ALUFunc = AND, MUX1 = 1, MUX2 = 1, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 0, WriteReg |
| OR | ReadIM, LoadIR, LoadNPC | ReadReg1, ReadReg2, Load A, Load B | ALUFunc = OR, MUX1 = 1, MUX2 = 1, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 0, WriteReg |
| XOR | ReadIM, LoadIR, LoadNPC | ReadReg1, ReadReg2, Load A, Load B | ALUFunc = XOR, MUX1 = 1, MUX2 = 1, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 0, WriteReg |
| NOR | ReadIM, LoadIR, LoadNPC | ReadReg1, ReadReg2, Load A, Load B | ALUFunc =NOR, MUX1 = 1, MUX2 = 1,LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 0, WriteReg |
| NOT | ReadIM, LoadIR, LoadNPC | ReadReg1, Load A | ALUFunc = NOT, MUX1 = 1, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 0, WriteReg |
| SL | ReadIM, LoadIR, LoadNPC | ReadReg1, ReadReg2, Load A, Load | ALUFunc = SL, MUX1 = 1, MUX2 = 1, | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 0, WriteReg |

| | | | | | |
|---|---|---|---|---|---|
| | | B | LoadALUOut | | |
| SRL | ReadIM, LoadIR, LoadNPC | ReadReg1, ReadReg2, Load A, Load B | ALUFunc = SRL, MUX1 = 1, MUX2 = 1, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 0, WriteReg |
| SRA | ReadIM, LoadIR, LoadNPC | ReadReg1, ReadReg2, Load A, Load B | ALUFunc = SRA, MUX1 = 1, MUX2 = 1, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 0, WriteReg |
| INC | ReadIM, LoadIR, LoadNPC | ReadReg1, Load A | ALUFunc = INC, MUX1 = 1, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 2, WriteReg |
| DEC | ReadIM, LoadIR, LoadNPC | ReadReg1, Load A, Load B | ALUFunc = ADD, MUX1 = 1, MUX2 = 1, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 2, WriteReg |
| SLT | ReadIM, LoadIR, LoadNPC | ReadReg1, ReadReg2, Load A, Load B | ALUFunc = ADD, MUX1 = 1, MUX2 = 1, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 0, WriteReg |
| SGT | ReadIM, LoadIR, LoadNPC | ReadReg1, ReadReg2, Load A, Load B | ALUFunc = ADD, MUX1 = 1, MUX2 = 1, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 0, WriteReg |
| HAM | ReadIM, LoadIR, LoadNPC | ReadReg1, ReadReg2, Load A, Load B | ALUFunc = ADD, MUX1 = 1, MUX2 = 1, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 0, WriteReg |

| **Instruction** | **T1** | **T2** | **T3** | **T4** | **T5** |
|---|---|---|---|---|---|
| MOVE | ReadIM, LoadIR, LoadNPC | ReadReg1,LoadImm1, Load A, MUXI=0 | ALUFunc = ADD, MUX1 = 1, MUX2 = 0, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 0, WriteReg |

| | | | | | |
|---|---|---|---|---|---|
| CMOV | ReadIM, LoadIR, LoadNPC | ReadReg1, ReadReg2, Load A, LoadB | ALUFunc = SLT, MUX1 = 0, MUX2 = 0, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 2, MUXDst = 0, WriteReg |

**I-type Instructions**

| **Instruction** | **T1** | **T2** | **T3** | **T4** | **T5** |
|---|---|---|---|---|---|
| ADDI | ReadIM, LoadIR, LoadNPC | ReadReg1,LoadImm1, Load A, MUXI = 0 | ALUFunc = ADD, MUX1 = 1, MUX2 = 0, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 1, WriteReg |
| SUBI | ReadIM, LoadIR, LoadNPC | ReadReg1,LoadImm1, Load A, MUXI = 0 | ALUFunc = SUB, MUX1 = 1, MUX2 = 0, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 1, WriteReg |
| ANDI | ReadIM, LoadIR, LoadNPC | ReadReg1,LoadImm1, Load A, MUXI = 0 | ALUFunc = AND, MUX1 = 1, MUX2 = 0, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 1, WriteReg |
| ORI | ReadIM, LoadIR, LoadNPC | ReadReg1,LoadImm1, Load A, MUXI = 0 | ALUFunc = OR, MUX1 = 1, MUX2 = 0, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 1, WriteReg |
| XORI | ReadIM, LoadIR, LoadNPC | ReadReg1,LoadImm1, Load A, MUXI = 0 | ALUFunc = XOR, MUX1 = 1, MUX2 = 0, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 1, WriteReg |
| NORI | ReadIM, LoadIR, LoadNPC | ReadReg1,LoadImm1, Load A, MUXI = 0 | ALUFunc = NOR, MUX1 = 1, MUX2 = 0, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 1, WriteReg |
| NOTI | ReadIM, LoadIR, LoadNPC | LoadImm1, MUXI = 0 | ALUFunc = NOT, MUX1 = 1, MUX2 = 0, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 1, WriteReg |

| | | | | | |
|---|---|---|---|---|---|
| SLI | ReadIM, LoadIR, LoadNPC | ReadReg1,LoadImm1, LoadA, MUXI = 0 | ALUFunc = SL, MUX1 = 1, MUX2 = 0, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 1, WriteReg |
| SRLI | ReadIM, LoadIR, LoadNPC | ReadReg1,LoadImm1, LoadA, MUXI = 0 | ALUFunc = SRL, MUX1 = 1, MUX2 = 0, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 1, WriteReg |
| SRAI | ReadIM, LoadIR, LoadNPC | ReadReg1,LoadImm1, LoadA, MUXI = 0 | ALUFunc = SRA, MUX1 = 1, MUX2 = 0, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 1, WriteReg |
| SLTI | ReadIM, LoadIR, LoadNPC | ReadReg1,LoadImm1, LoadA, MUXI = 0 | ALUFunc = SLT, MUX1 = 1, MUX2 = 0, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 1, WriteReg |
| SGTI | ReadIM, LoadIR, LoadNPC | ReadReg1,LoadImm1, LoadA, MUXI = 0 | ALUFunc = SGT, MUX1 = 1, MUX2 = 0, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 1, WriteReg |
| HAMI | ReadIM, LoadIR, LoadNPC | LoadImm1, MUXI = 0 | ALUFunc = HAM, MUX1 = 1, MUX2 = 0, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 1, WriteReg |
| LUI | ReadIM, LoadIR, LoadNPC | LoadImm1, MUXI = 0 | ALUFunc = LUI, MUX1 = 1, MUX2 = 0, LoadALUOut | LoadPC, MUXPC = 0 | MUXW = 1, MUXDst = 1, WriteReg |

| **Instruction** | **T1** | **T2** | **T3** | **T4** | **T5** |
|---|---|---|---|---|---|
| LD | ReadIM, LoadIR, LoadNPC | ReadReg1, LoadA, LoadImm1, MUXI = 0 | ALUFunc = ADD, MUX1 = 1, MUX2 = 0, LoadALUOut | LoadPC, ReadDM, LoadLMD, MUXD = 0, MUXPC = 0 | MUXW = 0, WriteReg |

| | | | | | |
|---|---|---|---|---|---|
| ST | ReadIM, LoadIR, LoadNPC | ReadReg1, ReadReg2, LoadA, LoadB, LoadImm1, MUXI = 0 | ALUFunc = ADD, MUX1 = 1, MUX2 = 0, LoadALUOut | LoadPC, WriteDM, MUXD = 0, MUXPC = 0 | |
| BMI | ReadIM, LoadIR, LoadNPC | ReadReg1, LoadA, LoadImm1, MUXI = 0 | ALUFunc = ADD, MUX1 = 0, MUX2 = 0, LoadALUOut | LoadPC, MUXPC = 2 | |
| BPL | ReadIM, LoadIR, LoadNPC | ReadReg1, LoadA, LoadImm1, MUXI = 0 | ALUFunc = ADD, MUX1 = 0, MUX2 = 0, LoadALUOut | LoadPC, MUXPC = 4 | |
| BZ | ReadIM, LoadIR, LoadNPC | ReadReg1, LoadA, LoadImm1, MUXI = 0 | ALUFunc = ADD, MUX1 = 0, MUX2 = 0, LoadALUOut | LoadPC, MUXPC = 3 | |

**J-type Instructions**

| Instruction | T1 | T2 | T3 | T4 | T5 |
|---|---|---|---|---|---|
| BR | ReadIM, LoadIR, LoadNPC | LoadImm2, MUXI = 1 | ALUFunc = ADD, MUX1 = 0, MUX2 = 0, LoadALUOut | LoadPC, MUXPC = 1 | |

**No Field type Instructions**

| Instruction | T1 | T2 | T3 | T4 | T5 |
|---|---|---|---|---|---|
| NOP | ReadIM, LoadIR, LoadNPC | | | | |
| HALT | ReadIM, LoadIR, LoadNPC | | | | |

**Notes on Control Signals**

**MUXI:** 0 - Imm1
1 - Imm2

**MUXDst:** 0 - dst = rd
1 - dst = rt
2 - dst = rs

**MUX1:** 0 - NPC
1 - A
**MUX2:** 0 - Imm
1 - B

**MUXD:** 0 - ALUOut
1 - B

**MUXW:** 0 - LMD
1 - ALUOut
2 - Output of MUXMOVE

**MUXPC:** 0 - 0
1 - 1
2 - A < 0
3 - A = 0
4 - A > 0