

**COMPUTER ORGANIZATION AND
ARCHITECTURE LABORATORY
ASSIGNMENT - 7**

SCHEMATIC DIAGRAM OF DATA PATH AND DESIGN OF CONTROL UNIT

GROUP 22

Aritra Maji (22CS30011)

Srinjoy Das (22CS30054)

INSTRUCTIONS

R-type (17 instructions)

Opcode	Rs	Rt	Rd		funct
5 bits	4 bits	4 bits	4 bits		5 bits

INSTRUCTION	OPCODE	FUNCTION
ADD	0000	00000
SUB	0000	00001
AND	0000	00010
OR	0000	00011
XOR	0000	00100
NOR	0000	00101
NOT	0000	00110
SL	0000	00111
SRL	0000	01000
SRA	0000	01001
INC	0000	01010
DEC	0000	01011
SLT	0000	01100
SGT	0000	01101
HAM	0000	01110
MOVE	0000	01111
CMOV	0000	10000

I-type (19 instructions)

Opcode	Rs	Rt	immediate
5 bits	4 bits	4 bits	19 bits

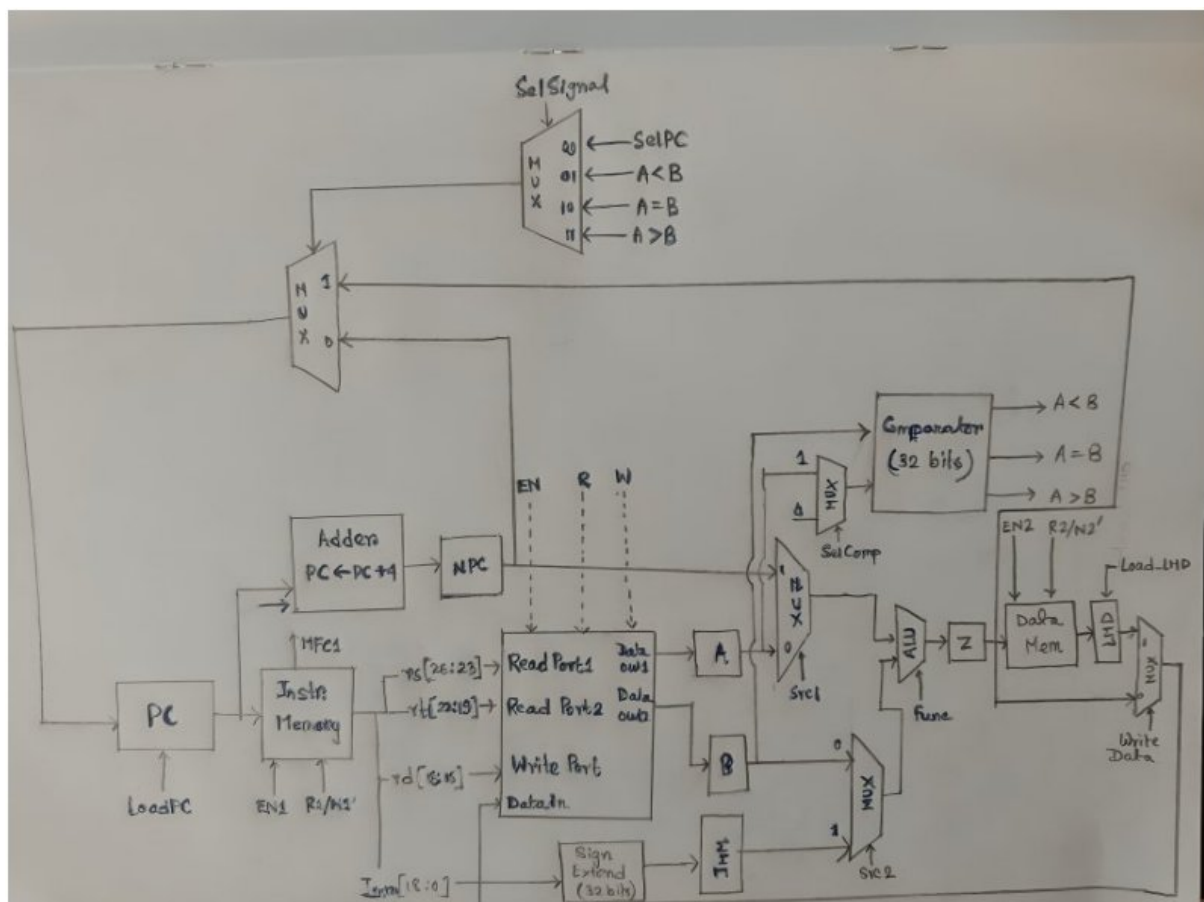
INSTRUCTION	OPCODE
ADDI	00001
SUBI	00010
ANDI	00011
ORI	00100
XORI	00101
NORI	00110
NOTI	00111
SLI	01000
SRLI	01001
SRAI	01010

SLTI	01011
SGTI	01100
LUI	01101
LD	01110
ST	01111
BR	10000
BML	10001
BPL	10010
BZ	10011
RET	10100
HAMI	10101

J-type (2 instructions)

Opcode	immediate
5 bits	27 bits

Opcode	immediate
HALT	10110
NOP	10111



Datapath Diagram

Control Signals

- LoadPC – Load signal of PC(+ve edge triggered FF)
- EN1,R1/W1',MFC1 – Control signals for Instruction Memory(Enable,Read,Memory Function complete)
- LoadIR – Load signal of IR(+ve edge triggered FF)
- EN, R, W – Control Signals for Register Bank
- WritePort – MUX signal to choose destination register among rt, rd
- Src1, Src2 – MUX signals to choose inputs to ALU
- func– ALU function select signal
- SelectComp– MUX signal to select between 0 and R[rt] as input to Comparator(For Branch instruction choose 0 for CMOV choose R[rt])
- EN2,R2/W2',MFC2– Control signals for Data Memory(Enable,Read,Memory Function complete)
- LoadLMD– Load signal of LMD(+ve edge triggered FF)
- WriteData– MUX signal to select between Data Loaded from memory and ALU output
- SelPC, SelSignal– MUX signals to select the final PC value will be NPC or ALU output

CONTROL PATH:

For every instruction, there is a common step to fetch instructions that occurs before any other step:

T1 : EN1, L1, LoadIR, WMFC1

The subsequent steps for all the instructions are :

- BZ:
 - T2 : Src1, Src2, func = add, SelComp=0, SelSignal=10, LoadPC
- BMI :
 - T2 : Src1, Src2, func = add, SelComp=0, SelSignal=01, LoadPC
- BPL :
 - T2 : Src1, Src2, func = add, SelComp=0, SelSignal=11, LoadPC
- BR :
 - T2 : Src1, Src2, func = add, SelSignal = 00, selPC, LoadPC
- ST :
 - T2 : EN, R, Src1=0, Src2=1, func = add, EN2, R2=0, WMFC2
 - T3 : SelSignal = 00, SelPC = 0, LoadPC
- LD :
 - T2 : EN, R, W, WritePort = 1, Src1=0, Src2=1, func = add, EN2, R2, LoadLMD, WMFC2
 - T3 : WriteData = 1, WritePort = 1, SelSignal = 00, SelPC = 0, LoadPC
- For f = one of { ADD, SUB, OR, AND, NOR, XOR, SL, SRL, SRA, SGT, SLT, NOT, INC, DEC, MOVE, HAM }
 - T2 : EN, R, W, WritePort = 0, Src1=0, Src2=0, func = f, WriteData=0, SelSignal = 00, SelPC = 0
- For g = one of { ADDI, SUBI, ORI, ANDI, NORI, XORI, SLI, SRLI, SRAI, SGTI, SLTI, NOTI, HAMI, LUI}
 - T2 : EN, R, W, WritePort = 1, Src1=0, Src2=1, func = g, WriteData=0, SelSignal=00, SelPC=0

- CMOV :
 - T2 : EN, R, W, Src1=0, Src2=0, (A<B)?func=PassB, WriteData=0, WritePort=0, SelSignal=00, SelPC=0, LoadPC
- HALT :
 - T2 : Wait for external interruption signal
- NOP :
 - Apply RESET to step decoder
- RET :
 - T3 : EN, R, Src1 = 0, Func=func(PassA), SelSignal=00, SelPC=1, LoadPC=1