

32-bit Processor Design and Instruction Format

Group 8

Bratin Mondal (21CS10016)

Somya Kumar (21CS30050)

Submitted on: October 11th, 2023

ISA Design:

The instruction set architecture we propose has four types of Instructions. The details of the instruction type and their encoding are given below

A - Type:

This type of instruction takes as input 3 register numbers of 5 bits each. 6 bit Opcode for all of them is 0 and a 10 bit function code is provided :

31	26	25	21	20	16	15	11	10	0
Opcode		rs (Register Source)			rt (Register Operand)		rd (Register Destination)		Function

The opcode and function code for different types of operations are shown below :

Operation	Opcode	Function	Instruction
ADD	000000	0000000001	add \$rd, \$rs, \$rt
SUB	000000	0000000010	sub \$rd, \$rs, \$rt
AND	000000	0000000100	and \$rd, \$rs, \$rt

C - Type:

This type of instruction takes as input 6 bit Opcode and a 26 bit immediate value is provided :

31 26	25	0
Opcode	Immediate	

The opcode for different types of operations are shown below :

Operation	Opcode	Instruction
BR	010011	br imm
CALL	010100	call imm

D - Type:

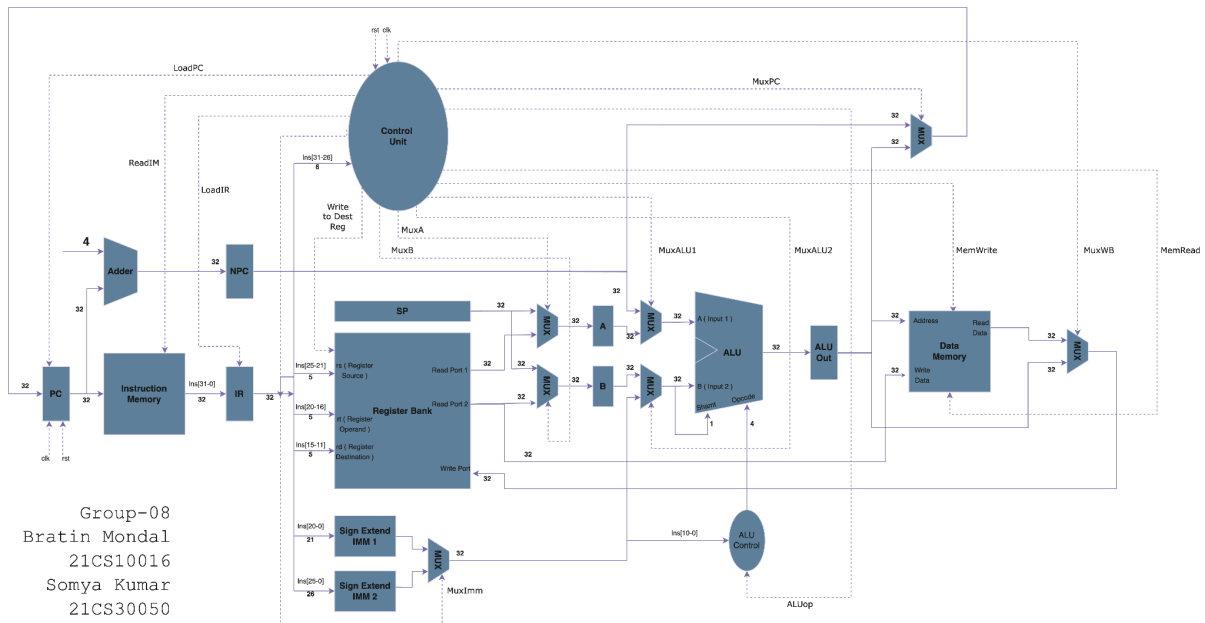
This type of instruction takes as input 6 bit Opcode and rest 26 bit is ignored :

31 26	25	0
Opcode	Don't Care	

The opcode for different types of operations are shown below :

Operation	Opcode	Instruction
RET	010101	ret
HALT	010110	halt
NOP	010111	nop

Schematic Diagram:



Control Signals

1. LoadPC: Control signal to load data into PC from data bus
2. ReadIM: Control signal to read instruction from instruction memory
3. LoadIR: Control signal to load instruction into instruction register
4. {MuxIM, MuxA, MuxB, MuxALU1, MuxALU2, MuxWB, MuxPC}: Control signal to select one of the two multiplexer input
5. ALUop: Control signal to decode function code and send the corresponding opcode to ALU
6. MemWrite: Control signal to write data to memory
7. MemRead: Control signal to read data from memory
8. WriteToDestReg: Control signal to write data into destination register

[G - Drive Link](#)

[Published Link](#)