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 2	1	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	000000001	add \$rd, \$rs, \$rt
SUB	000000	000000010	sub \$rd, \$rs, \$rt
AND	000000	000000100	and \$rd, \$rs, \$rt

OR	000000	000001000	or \$rd, \$rs, \$rt
XOR	000000	0000010000	xor \$rd, \$rs, \$rt
NOT	000000	0000100000	not \$rd, \$rs
SLA	000000	0001000000	sla \$rd, \$rs, \$rt
SRA	000000	0010000000	sra \$rd, \$rs, \$rt
SRL	000000	010000000	srl \$rd, \$rs, \$rt
MOVE	000000	100000000	move \$rd, \$rs

B - Type:

This type of instruction takes as input register(s) and immediate value. We further subdivide it in 2 category:

B1: (1 Register)

This type of instruction takes as input 1 register number of 5 bits each. 6 bit Opcode for all of them is specified and a 21 bit immediate value is provided

31 26	25 21	20 0
Opcode	rs (Register Source)	Immediate

The opcode for different types of operations are shown below :

Operation	Opcode	Instruction
ADDI	000001	addi \$rs, imm
SUBI	000010	subi \$rs, imm
ANDI	000011	andi \$rs, imm
ORI	000100	ori \$rs, imm
XORI	000101	xori \$rs, imm

NOTI	000110	noti \$rs, imm
SLAI	000111	slai \$rs, imm
SRAI	001000	srai \$rs, imm
SRLI	001001	srli \$rs, imm
ВМІ	001010	bmi \$rs, imm
BPL	001011	bpl \$rs, imm
BZ	001100	bz \$rs, imm
LDSP	001101	ldsp sp, imm(\$rs)
STSP	001110	stsp sp, imm(\$rs)
PUSH	001111	push \$rs
POP	010000	pop \$rs

B2: (2 Register)

This type of instruction takes as input 2 register numbers of 5 bits each. 6 bit Opcode for all of them is specified and a 16 bit immediate value is provided

31 26	25 21	20 16	15	0
Opcode	rs (Register Source)	rt (Register Operand)	Immediate	

The opcode for different types of operations are shown below :

Operation	Opcode	Instruction
LD	010001	ld \$rs,imm(\$rt)
ST	010010	st \$rs,imm(\$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	ż	Immediate	

The opcode for different types of operations are shown below :

Operation	Opcode	Instruction
BR	010011	br imm
CALL	010100	call imm

D - Type:

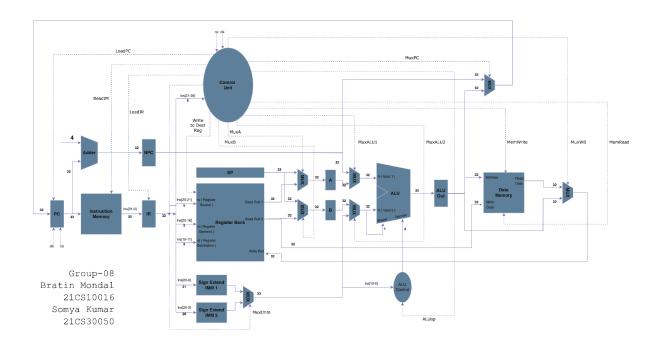
This type of instruction takes as input 6 bit Opcode and rest 26 bit is ignored :

31 26	6	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	пор

Schematic Diagram:



Control Signals

- 1. LoadPC: Control signal to load data into PC form 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, 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