Instruction Encoding, Data Path and Control Unit Design

Group - 15

B Pranesh Vijay - 22CS10013

Sai Shree Pradhan - 22CS10089

1. Instruction Format and Encoding

1.1 R - Type Instructions

Opcode	Source Register -1 (rs)	Source Register - 2 (rt)	Destination Register (rd)	Shift Amount (shamt)	Function (funct)
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

Instruction	Usage	Opcode	Funct Code
ADD	ADD rd, rs, rt	000000	000001
SUB	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
NOR	NOR rd, rs, rt	000000	000110
NOT	NOT rd, rs	000000	000111
SL	SL rd, rs, rt	000000	001000
SRL	SRL rd, rs, rt	000000	001001
SRA	SRA rd, rs, rt	000000	001010
SLT	SLT rd, rs, rt	000000	001011
SGT	SGT rd, rs, rt	000000	001100
HAM	HAM rd, rs	000000	001101
INC	INC rd, rs	000000	001110

Instruction	Usage	Opcode	Funct Code
DEC	DEC rd, rs	000000	001111
CMOV	CMOV rd, rs, rt	000000	010000

1.2 I - Type Instructions

Opcode	Source Register (rs)	Destination Register (rt)	Immediate Data (imm)
6 bits	5 bits	5 bits	16 bits

Instruction	Usage	Opcode
ADDI	ADDI rt, rs, imm	000001
SUBI	SUB rt, rs, imm	000010
ANDI	AND rt, rs, imm	000011
ORI	OR rt, rs, imm	000100
XORI	XOR rt, rs, imm	000101
NORI	NOR rt, rs, imm	000110
NOTI	NOT rt, imm	000111
SLI	SL rt, rs, imm	001000
SRLI	SRL rt, rs, imm	001001
SRAI	SRA rt, rs, imm	001010
SLTI	SLTI rt, rs, imm	001011
SGTI	SGTI rt, rs, imm	001100
HAMI	HAM rt, imm	001101
LUI	LUI rt, imm	010001
LD	LD rt, rs, imm	010010
ST	ST rt, rs, imm	010011
BR	BR imm	010100
BMI	BMI rs, imm	010101
BPL	BPL rs, imm	010110
BZ	BZ rs, imm	010111
MOVE	MOVE rt, rs	011000

1.3 Program Control Instructions

Opcode	Immediate Value
6 bits	26 bits

Instruction	Usage	Opcode
HALT	HALT	0011001
NOP	NOP	0011010

2. Register Usage Convention

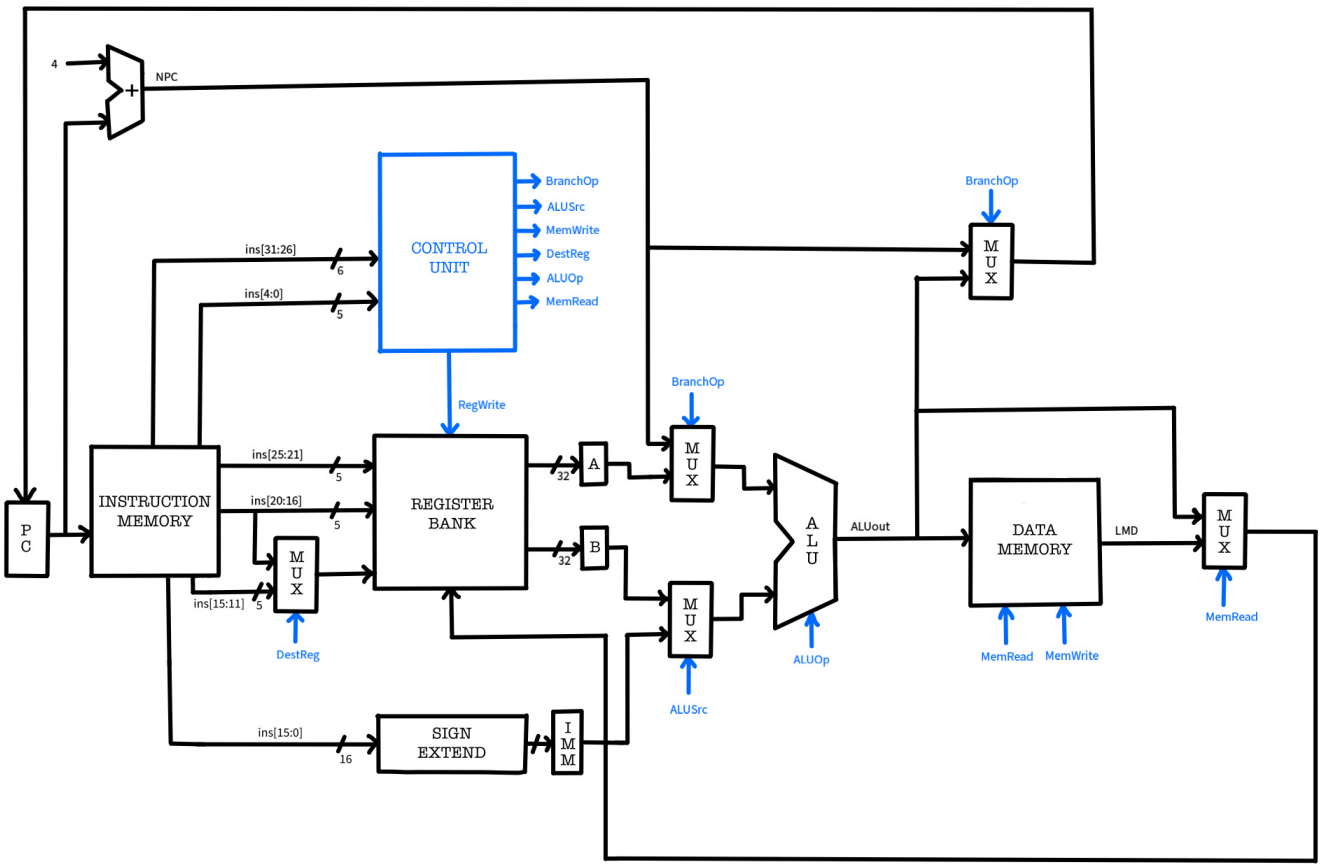
Register	Function	Register Number
$R0	Hardwired to 0	0
$R1 - $R15	Temporary Registers	1 - 15
$SP	Stack Pointer	16
$PC	Program Counter	17

3. Control Signals

1. **BranchOp**: Chooses between the Program Counter (PC) and the address of register A based on the branch instruction type; this input directs the ALU for branching operations.
2. **ALUOp**: Specifies the exact operation that the ALU should perform based on the instruction's function code (e.g., add, subtract).
3. **ALUSrc**: Determines the source of the second operand for the ALU; selects register B for R-type instructions and the immediate value for I-type instructions.
4. **MemRead**: Activates the memory read process to retrieve data from memory during load operations.
5. **MemWrite**: Activates the memory write process to store data in memory during store operations.
6. **RegWrite**: Enables writing the result of an operation back into the register bank, allowing data to be stored in registers.
7. **DestReg**: Specifies which register will receive the output of the operation; uses Rt for I-type instructions and Rd for R-type instructions.

Opcode	BranchOp	ALUOp	ALUSrc	MemRead	MemWrite	RegWrite	DestRe
000000	000	00000	0	0	0	1	1
000001	000	00001	1	0	0	1	0
000010	000	00010	1	0	0	1	0
000011	000	00011	1	0	0	1	0
000100	000	00100	1	0	0	1	0
000101	000	00101	1	0	0	1	0
000110	000	00110	1	0	0	1	0
000111	000	00111	1	0	0	1	0
001000	000	01000	1	0	0	1	0
001001	000	01001	1	0	0	1	0
001010	000	01010	1	0	0	1	0
001011	000	01011	1	0	0	1	0
001100	000	01100	1	0	0	1	0
001101	000	01101	1	0	0	1	0
010001	000	10001	1	0	0	1	0
010010	000	00001	1	1	0	1	0
010011	000	00001	1	0	1	1	0
010100	001	00001	1	0	0	0	x
010101	010	00001	1	0	0	0	x
010110	011	00001	1	0	0	0	x
010111	100	00001	1	0	0	0	x
011000	000	xxxxx	1	0	0	1	0
0011001	000	xxxxx	x	x	x	x	x
0011010	000	xxxxx	x	x	x	x	x

4. Datapath


4
NPC
CONTROL UNIT
BranchOp
ALUSrc
MemWrite
DestReg
ALUOp
MemRead
ins[31:26]
6
ins[4:0]
5
RegWrite
BranchOp
BranchOp
M U X
P C
INSTRUCTION MEMORY
ins[25:21]
5
ins[20:16]
5
ins[15:11]
5
REGISTER BANK
32
A
32
B
M U X
DestReg
M U X
ALUSrc
A L U
ALUout
DATA MEMORY
LMD
MemRead
MemWrite
M U X
MemRead
ins[15:0]
16
SIGN EXTEND
I M M