COA Project Part 1 : ISA Design

Team Members
1. Ansh Jain 19ucs059
2. Gaurav Niranjan 19ucs054
3. Neeraj Kumar Singhal 19ucs029
4. Aditya Raj 19ucs025

Instruction length - 16 bits (2 words)
Number of registers in Register File = 8
Main memory = 256 bytes
Word addressable memory
Size of address bus = 8 bits
Size of data bus = 16 bits
Capabilities :

i) Data Movement: LOAD and STORE by using a 2 step process ie, first copying the value from source main memory location into a register, and then copy it from the register to the destination main memory location. We use this 2 step process because the Instruction length of 16 bits cannot hold 2 addresses of main memory at the same time.

ii) Logical Operations: AND, OR, XOR, NAND, NOR on two input values and NOT on a single input value.

iii) Arithmetic Operations: Addition, Subtraction, Multiplication and Division.

iv) Unconditional Branch

v) Conditional Branch: Along with the ability to compare two values and store this result in status flags(equal to zero, less than zero and greater than zero).

Total Opcodes to support these capabilities = 16.

Instruction:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

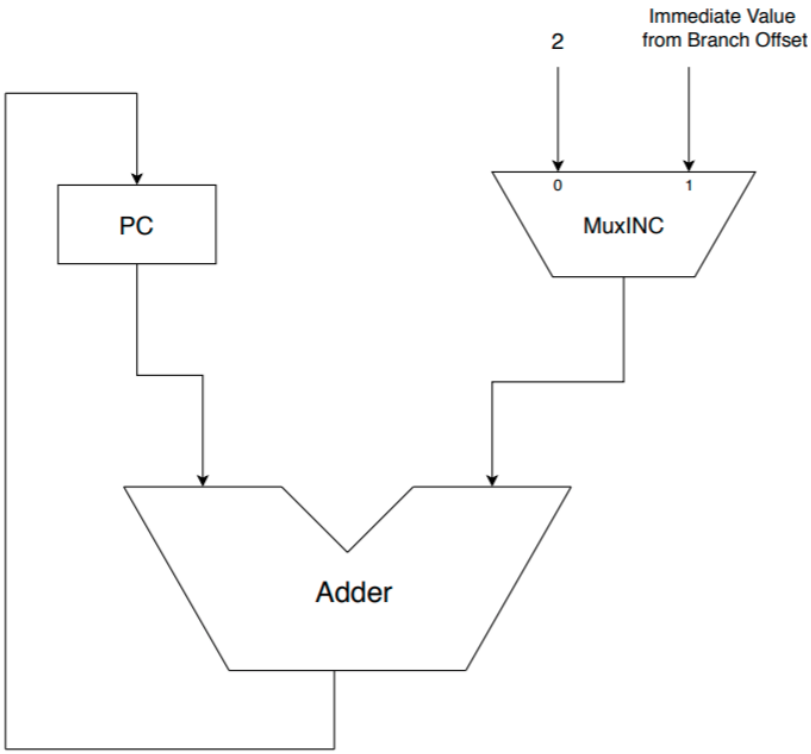
0 - 3 -> Opcode
4 -> 0: Register address in 5 - 8
1:Immediate value in 5 - 8
5 - 7 -> Immediate value/Address of register
8 - 15 -> Memory Address

Instruction	Opcode	FORMAT	DESCRIPTION
LOAD	0000	LDM Rx (MemAdd) Rx: 3 bits MemAdd: 8 bits	Load the value at given memory address in Rx
STORE	0001	STR Rx (MemAdd) or STR ImmdVal(MemAdd) Rx/ImmdValue: 3 bit MemAdd: 8 bits	Store value in register or immediate value to Memory location
AND	0010	AND Rx (MemAdd) Rx:3 bits MemAdd: 8 bits	(Rx and Memory Address) stored in Memory address
OR	0011	OR Rx (MemAdd) Rx:3 bits MemAdd: 8 bits	(Rx OR Memory Address) stored in Memory address
XOR	0100	XOR Rx (MemAdd) Rx:3 bits MemAdd: 8 bits	(Rx XOR Memory Address) stored in Memory address
NAND	0101	NAND Rx (MemAdd) Rx:3 bits MemAdd: 8 bits	(Rx NAND Memory Address) stored in Memory address
NOR	0110	NOR Rx (MemAdd) Rx:3 bits MemAdd: 8 bits	(Rx NOR Memory Address) stored in Memory address
NOT	0111	NOT Rx (MemAdd) Rx:3 bits MemAdd: 8 bits	(Not RX) stored in Memory address
ADD	1000	ADD Rx (MemAdd) Rx:3 bits MemAdd: 8 bits	(Rx + value at memory Address) stored at memory address
SUB	1001	SUB Rx (MemAdd) Rx:3 bits MemAdd: 8 bits	(value at memory Address - Rx) stored at memory address
MUL	1010	MUL Rx (MemAdd) Rx:3 bits MemAdd: 8 bits	(Rx * value at memory Address) stored at memory address

DIV	1011	DIV Rx (MemAdd) Rx:3 bits MemAdd: 8 bits	(value at memory Address / Rx) stored at memory address
CMP	1100	CMP Rx (MemAdd) Or CMP ImmdVal(MemAdd) Rx/ImmdValue: 3 bit MemAdd: 8 bits	Compare Value at memory address and Rx Flag = 0 :values equal Flag =1 Values not equal
JMP	1101	JMP(Offset) Offset: 12 bits	Jump
JEQ	1110	JEQ(Offset) Offset: 12 bits	Jump if flag set 0
JNE	1111	JNE(Offset) Offset: 12 bits	Jump if flag set 1

Flow of Instructions:

- Since Instruction length is 2 words and memory is word addressable so we increment the value of PC by 2. PC <- PC + 2.
- In case of unconditional branch: PC <- X where X is the branch target address.


Immediate Value
from Branch Offset
2
0
1
PC
MuxINC
Adder