Department of Electronics and Telecommunication Engineering
University of Moratuwa

EN2031 - Fundamentals of Computer Organization and Design

Processor Design Report

Team Tech – Tycoons

Name	**Index No.**
Pushpakumara H.M.R.M.	200488E
Tilakarathna U.A	200664P
Ranaviraja R.W.H.M.T.A.	200520X

(a)

I. Required Instructions

ALU operation

1. ADD – Addition
   - ADD R_n , R_m $R_n \leftarrow R_n + R_m$
   - 2 Operands
2. SUB – Subtraction
   - SUB R_n , R_m $R_n \leftarrow R_n - R_m$
   - 2 Operands
3. AND – Bitwise AND
   - AND R_n , R_m $R_n \leftarrow R_n \& R_m$
   - 2 Operands
4. OR – Bitwise OR
   - OR R_n , R_m $R_n \leftarrow R_n \mid R_m$
   - 2 Operands
5. PASS
   - $R_n \leftarrow R_n$

Register Move Instruction

6. MOVE – Move one register to another
   - MOVE R_n, R_m $R_n \leftarrow R_m$

LOAD Instructions

1. LOAD – Indirect Memory load from Data Memory
   - LOAD R_n , R_m $R_n \leftarrow M[R_m]$
2. LOADA – Indirect Memory load with address post modification
   - LOADA R_n , R_m^+ $R_n \leftarrow M[M[R_m]]$
     $R_m \leftarrow R_m + 1$
3. LOADI – 8-bit value extend to 16 bits and store in R_n
   - LOADI R_n , Imm $R_n \leftarrow$ Imm (extended to 16)
4. LOADU – Load upper immediate (upper 8 bits of 16 bits)

- LOADU R_n , Imm $R_n[8:15] \leftarrow$, Imm

5. POP – Remove TOS value to the R_n register and update the stack pointer
   - POP R_n

STORE Instructions

1. STORE – Store data from the R_m register into a R_n Memory location
   - STORE R_n , R_m $M[R_n] \leftarrow R_m$

2. PUSH – Put the value of the R_m to TOS and update the stack pointer.
   - PUSH R_m TOS$\leftarrow R_m$, SP=SP+1

BRANCH Instructions

1. BRANCHU – Branch if unconditional
   - BRANCHU offset PC ← PC + offset
2. BEQ – Branch if equal zero
   - BEQ R_d , offset PC ← PC + offset (if equal zero)
3. BLTE – Branch if less than or equal zero
   - BLTE R_d , offset PC ← PC + offset (if less than or equal zero)
4. BGTE – Brach if greater than or equal zero
   - BGTE R_d , offset PC ← PC + offset (if greater than or equal zero)

Opcode	Operand	Meaning
ADD	R_n , R_m	$R_n \leftarrow R_n + R_m$
SUB	R_n , R_m	$R_n \leftarrow R_n - R_m$
AND	R_n , R_m	$R_n \leftarrow R_n \& R_m$
OR	R_n , R_m	$R_n \leftarrow R_n \mid R_m$
MOVE	R_n , R_m	$R_n \leftarrow R_m$
LOAD	R_n , R_m	$R_n \leftarrow M[R_m]$

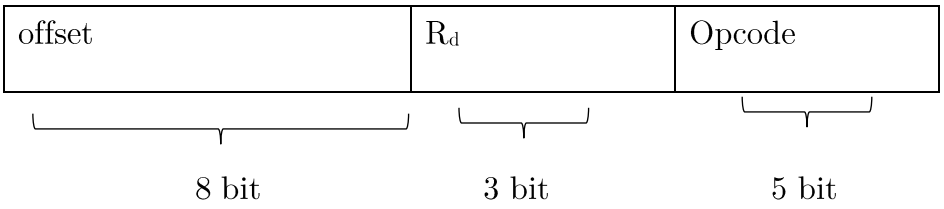
LOADA	R_n , R_m^+	$R_n \leftarrow M[M[R_m]]$ $R_m \leftarrow R_m + 1$
LOADI	R_n , Imm	$R_n \leftarrow$ Imm (extended to 16)
LOADU	R_n , Imm	$R_n[8:15] \leftarrow$, Imm
STORE	R_n , R_m	$M[R_n] \leftarrow R_m$
POP	R_n	$R_n \leftarrow$TOS SP = SP-1
PUSH	R_m	TOS$\leftarrow R_m$ SP=SP+1
BRANCHU	offset	PC ← PC + offset
BEQ	R_d , offset	PC ← PC + offset (if equal zero)
BLTE	R_d , offset	PC ← PC + offset (if less than or equal zero)
BGTE	R_d , offset	PC ← PC + offset (if greater than or equal zero)

II. Instruction format

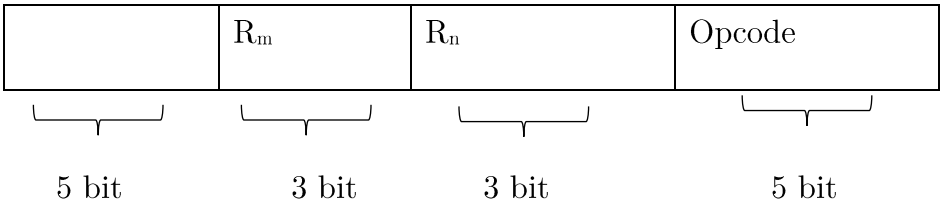
ALU

func	R_m	R_n	Opcode
5 bit	3 bit	3 bit	5 bit

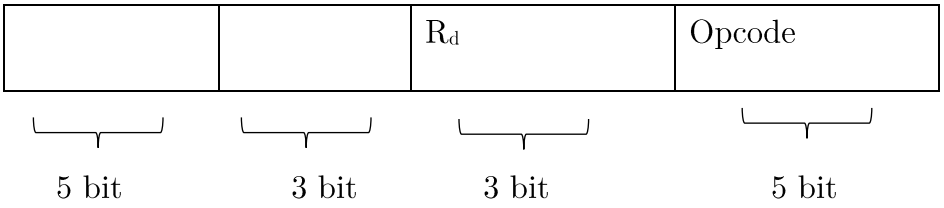
Control



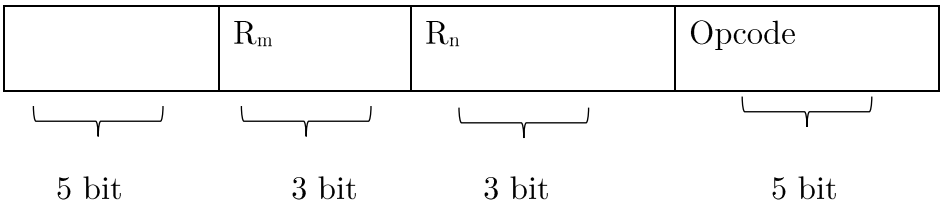
LOAD



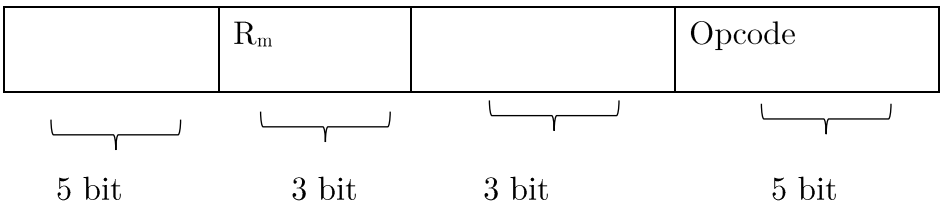
POP



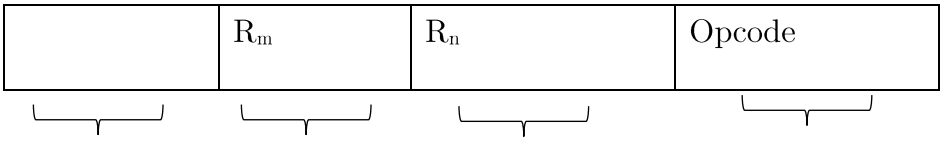
STORE



PUSH

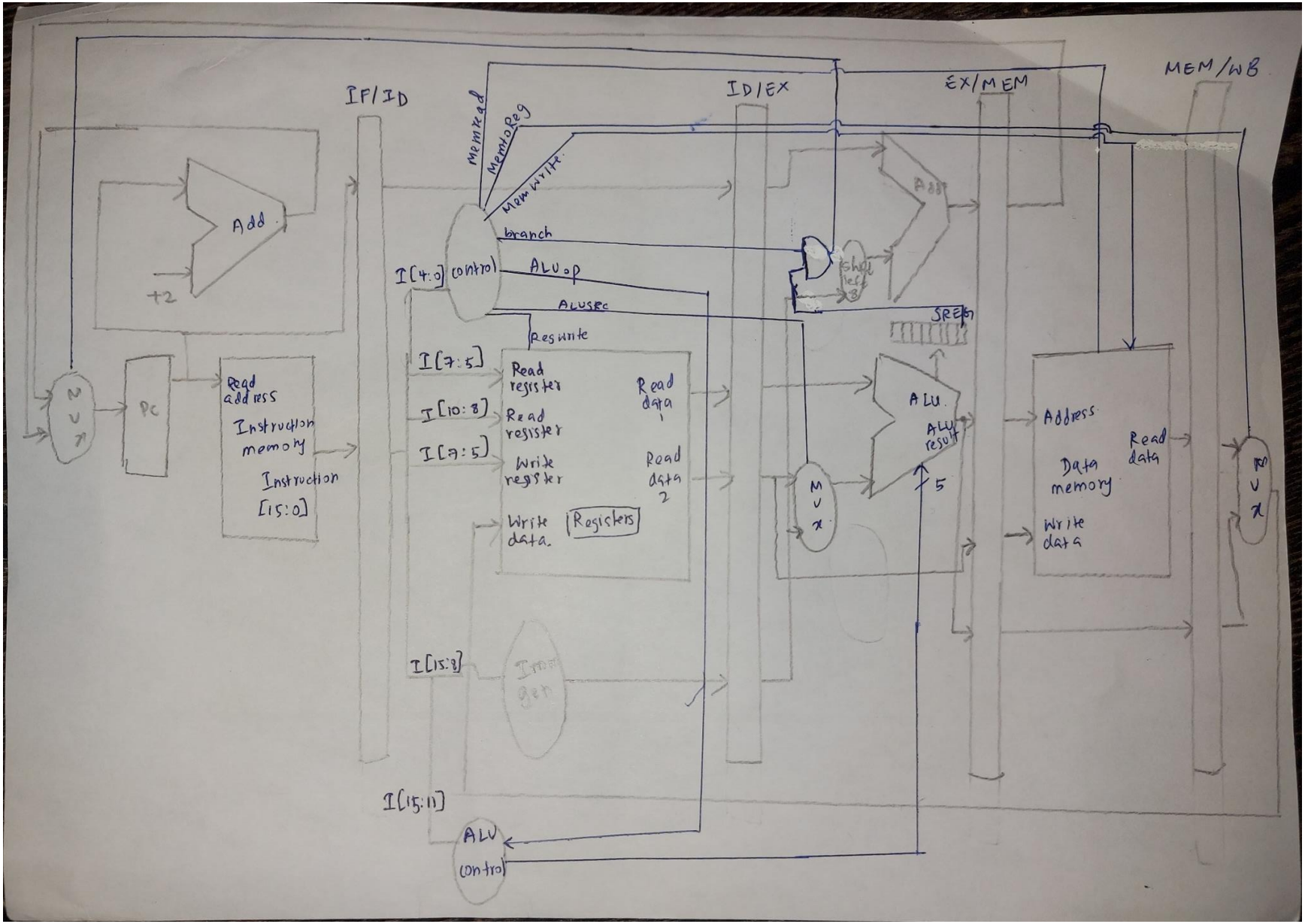


R type



(a)

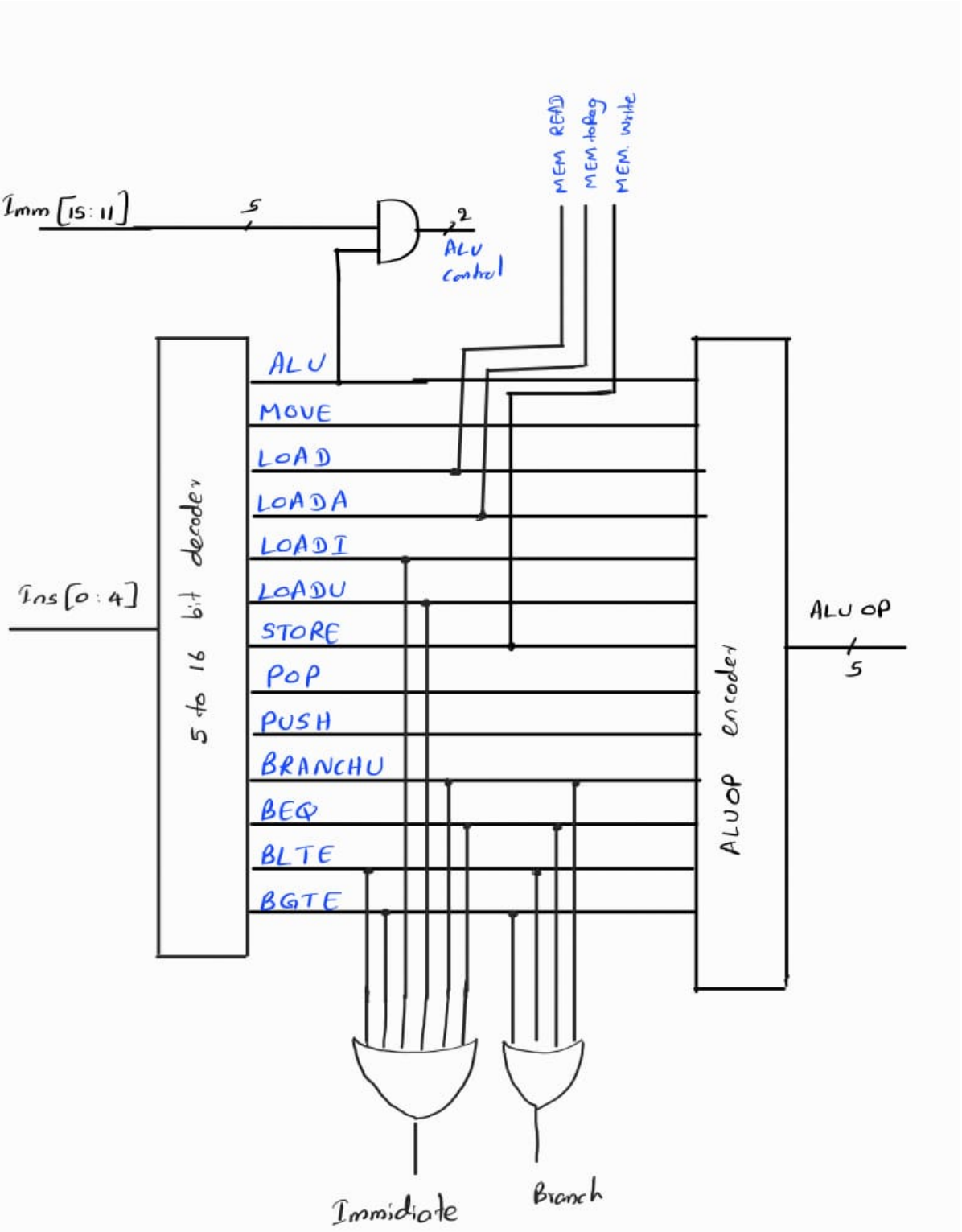
Opcodes	00	01	11	10
000	ALU			POP
001	LOAD	LOADA	LOADI	LOADU
010	STORE	PUSH		
011	MOVE			
100	BRANCHU	BEQ	BLTE	BGTE


IF/ID
ID/EX
EX/MEM
MEM/WB
MemtoReg
MemRead
MemWrite
branch
ALUop
ALUSrc
RegWrite
I[4:0] control
Add
+2
Shift left 2
PC
Read address
Instruction memory
Instruction [15:0]
I[7:5]
I[10:8]
I[7:5]
Read register
Read register
Write register
Write data
Read data 1
Read data 2
Registers
I[15:8]
Imm Gen
I[15:11]
ALU control
ALU
ALU result
5
Address
Data memory
Read data
Write data
M u x

(b)

(c)

I. The chosen design approach for the microarchitecture is hardwired. The reason for choose this is the heard wired designs are faster than micro architecture design approach. Since we design this processor for high-speed industrial application, it is better to use hardwired design approach. Since this has small set of instructions, the hardwired approach is not hard to implement here.


Imm [15:11]
5
2
ALU control
MEM READ
MEM toReg
MEM. write
ALU
MOVE
LOAD
LOADA
LOADI
LOADU
STORE
POP
PUSH
BRANCHU
BEQ
BLTE
BGTE
Ins [0:4]
5 to 16 bit decoder
ALUOP encoder
ALU OP
5
Immidiate
Branch

II.