Design of a 32 Bit Processor

Verilog Assignment

Group 14
Gummadi Vikhyath - 21CS10028
Pasupulety Chethan Krishna Venkat – 21CS30036

1. Instruction format and Encoding

i. 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

R - 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
NOT	NOT Rd,Rs,Rt	000000	000110
SLA	SLA Rd,Rs	000000	000111
SRA	SRA Rd,Rs	000000	001001
SRL	SRL Rd,Rs	000000	001010

ii. I-type instructions:

Opcode	Source Register 1 (Rs)	Destination Register (Rt)	Immediate Value
6 bits	5 bits	5 bits	16 bits

I- Instruction	Usage	Opcode	Funct code
ADDI	ADDI Rs, Rt, Imm	01 0000	X
SUBI	SUBI Rs, Rt, Imm	01 0001	X
ANDI	ANDI Rs, Rt, Imm	01 0010	X
ORI	ORI Rs, Rt, Imm	01 0011	X
XORI	XORI Rs, Rt, Imm	01 0100	X
NOTI	NOTI Rs, Rt. Imm	01 0101	X
SLAI	SLAI Rs, Imm	01 0110	X
SRAI	SRLI Rs, Imm	01 1000	X
SRLI	SRAI Rs, Imm	01 1001	X
MOVE	MOVE Rs,Rt	01 1010	X

iii. Unconditional J-type Instructions:

Opcode	Immediate value
6 bits	26 bits

J type	Usage	Opcode	Funct code
BR	BR Imm	11 0000	X

iv. Conditional j – type Instructions

Opcode	Source Register 1 (Rs)	Immediate Value
6 bits	5 bits	21 bits

J type	Usage	Opcode	Funct code
BMI	BMI Rs, Imm	11 0001	X
BPL	BPL Rs, Imm	11 0010	X
BZ	BZ Rs, Imm	11 0011	X

v. Memory Instructions: encoding similar to I type:

Opcode	Source Register 1 (Rs)	Destination Register (Rt)	Immediate Value
6 bits	5 bits	5 bits	16 bits

Memory instruction	Usage	Opcode	Funct code
LD	LD Rt, Rs, Imm	10 0001	X
ST	ST Rt, Rs, Imm	10 0010	X
LDSP	LDSP SP, Rs, Imm	10 0011	X
STSP	STSP SP, Rs, Imm	10 0100	X

vi. Stack Instructions:

Opcode	Source Register 1 (Rs)	Destination Register (Rt)	Immediate Value
6 bits	5 bits	5 bits	X

Stack Instructions	Usage	Opcode	Funct code
PUSH	PUSH Rs	010011	X
POP	POP Rt	010100	X
CALL	CALL Imm	010101	X

vii. Program Control Instructions:

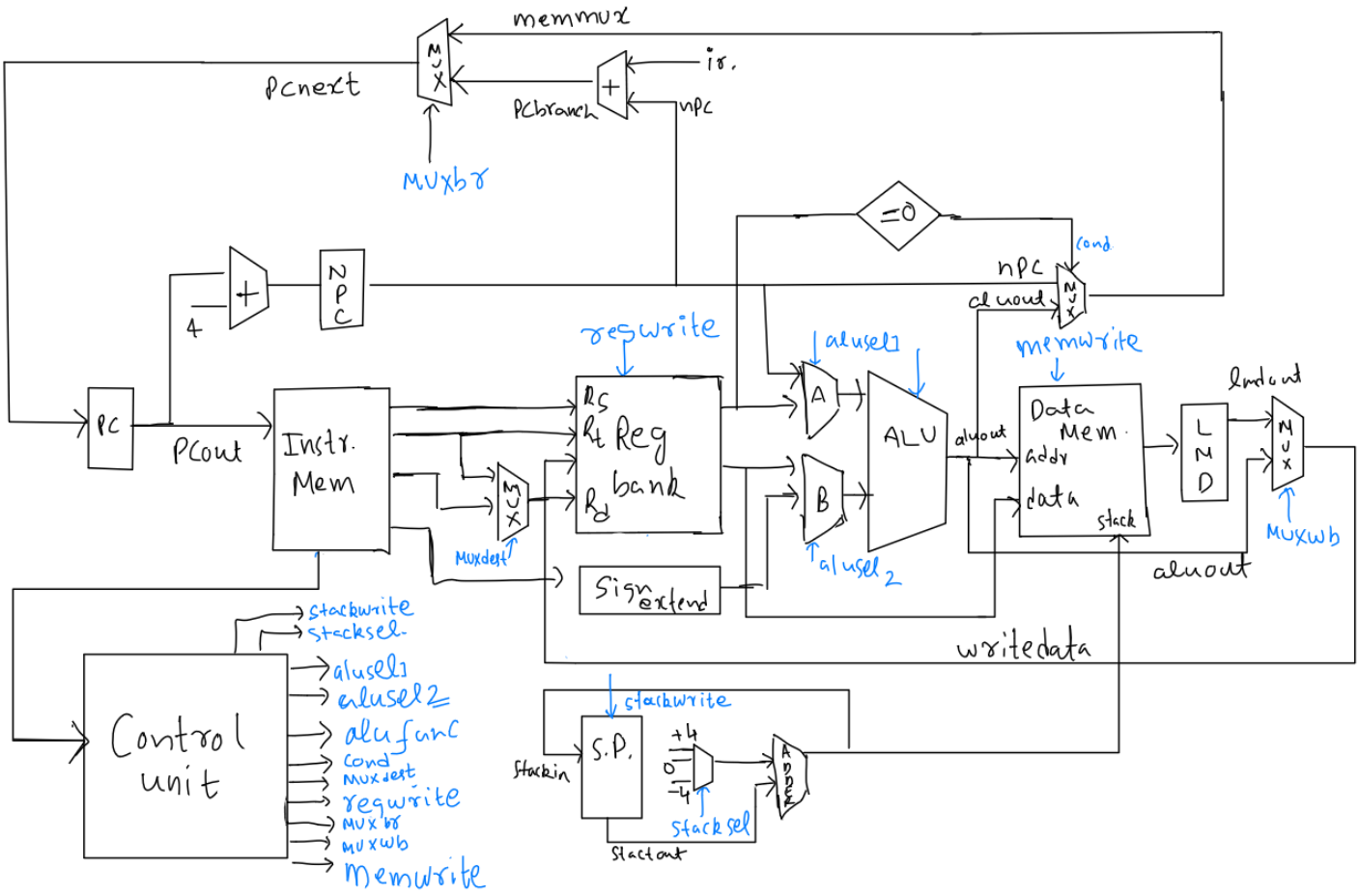
Opcode	-
6 bits	26 bits

Instruction	Usage	Total Instruction
HALT	HALT	11111111111111111111111111111111
NOP	NOP	11111111111111111111111111111110
RET	RET	11111111111111111111111111111101

2. Register Usage Convention

Register	Function	Register Number
$R0	Hardwired to 0	0
$R1-$R15	Temporary registers	1-15
$SP	Stack Pointer	16
$PC	Programme counter	17

3. Datapath


memmux
pcnext
MUXbr
PCbranch
ir.
npc
=0
cond
npc
aluout
4
NPC
regwrite
alusel1
memwrite
lmdout
PC
PCout
Instr. Mem
Rs
Rt Reg
Rd bank
A
B
ALU
aluout
Data Mem
addr
data
stack
LMD
MUXwb
MUXdest
alusel2
Sign extend
aluout
writedata
stackwrite
stacksel
alusel1
alusel2
alu func
cond
MUXdest
regwrite
MUXbr
MUXwb
memwrite
Control unit
stackwrite
Stackin
S.P.
+4
0
-4
ADDER
stacksel
Stactout

4. Control Signals

i. R Type Instructions:

Instr	ALUsel1	ALUsel2	ALUfunc	Cond	WriteReg	MuxDest	WriteDM	MuxWB	MuxBr
ADD	1	0	0000	11	1	0	0	1	0
SUB	1	0	0001	11	1	0	0	1	0
AND	1	0	0010	11	1	0	0	1	0
OR	1	0	0011	11	1	0	0	1	0
XOR	1	0	0100	11	1	0	0	1	0i
NOT	1	0	0101	11	1	0	0	1	0
SLA	1	0	0110	11	1	0	0	1	0
SLL	1	0	0111	11	1	0	0	1	0
SRA	1	0	1000	11	1	0	0	1	0
SRL	1	0	1001	11	1	0	0	1	0

ii. I Type Instructions:

Instr	ALUsel1	ALUsel2	ALUfunc	Cond	WriteReg	MuxDest	WriteDM	MuxWB	MuxBr
ADDI	1	1	0000	11	1	1	0	1	0
SUBI	1	1	0001	11	1	1	0	1	0
ANDI	1	1	0010	11	1	1	0	1	0
ORI	1	1	0011	11	1	1	0	1	0
XOIR	1	1	0100	11	1	1	0	1	0
NOIT	1	1	0101	11	1	1	0	1	0
SLAI	1	1	0110	11	1	1	0	1	0

SLLI	1	1	0111	11	1	1	0	1	0
SRAI	1	1	1000	11	1	1	0	1	0
SRLI	1	1	1001	11	1	1	0	1	0
MOVE	1	1	0000	11	1	1	0	1	0

Instr	ALUsel1	ALUsel2	ALUfunc	Cond	WriteReg	MuxDest	WriteDM	MuxWB	MuxBr
LD	1	1	0000	11	1	1	0	0	0
ST	1	1	0000	11	0	0	1	1	0

Instr	Stacksel	Stackwrite	ALUfunc	Cond	WriteReg	MuxDest	WriteDM	MuxWB	MuxBr
LDsp	1	0	0000	11	1	1	0	0	0
STsp	1	1	0000	11	0	0	1	1	0

iii. J Type Instructions:

Instr	ALUsel1	ALUsel2	ALUfunc	Cond	WriteReg	MuxDest	WriteDM	MuxWB	MuxBr
BR	0	0	0000	00	0	0	0	0	1
MRI	0	1	1010	01	0	0	0	0	0
BPL	0	1	1010	00	0	0	0	0	0
BZ	0	1	1010	10	0	0	0	0	0

5. Control Signals Description

Control Signal	Description
alusel1	It is used to differentiate between the Immediate instructions and the non immediate ones. Ex: ADD and ADDI
alusel2	Performs functions similar to alusel 1, 2 control lines are needed as there are also branch and load/store instructions present.
alufunc	Determines which ALU operation should take place. This is important as all the ALU operations have the same op code, They are differentiated by this particular control line
cond	Determines the true/false of a conditional statement present in a instruction
regwrite	Determines whether for the particular instruction is there a write operation to register which is required
muxdest	It is the control signal for the MUX which determines the destination of the register in the registar bank
memwrite	Determines whether any value should be written in the data memory which is a collection of registers
muxwb	It is the control signal which selects LMDout or ALUout , which of these 2 is to be written in the register bank.
muxbr	Determines whether for the particular instruction we have to branch or not
Stacksel	Determines whether the stack pointer should be selected or not from the register bank
Stackwrite	Select line to increase/decrease the value of the stack pointer during the push/pop operations