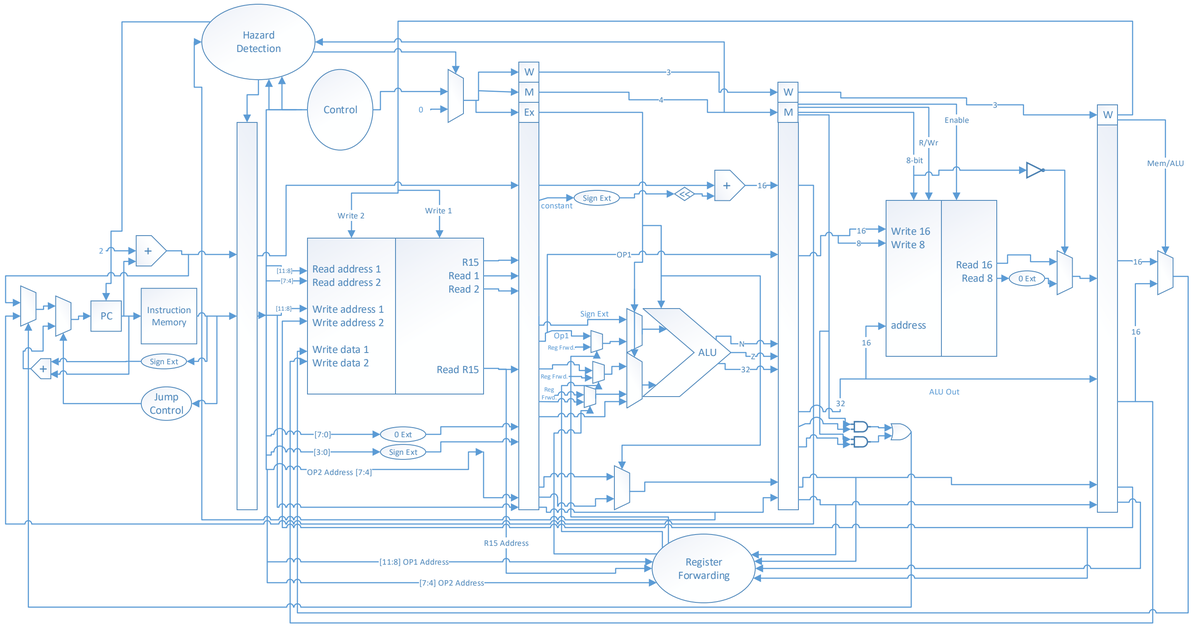
CSC/CpE 142

Term project

Phase 1

Andrew Robertson & Ehsan Halterman
(50% contribution from each)


Hazard Detection
Control
W
M
Ex
0
3
4
W
M
3
Enable
R/Wr
8-bit
W
Mem/ALU
Sign Ext
constant
<<
+
16
Write 2
Write 1
2
+
PC
Instruction Memory
Sign Ext
Jump Control
[11:8]
[7:4]
[11:8]
Read address 1
Read address 2
Write address 1
Write address 2
Write data 1
Write data 2
R15
Read 1
Read 2
Read R15
OP1
Sign Ext
Op1
Reg Fwd
Reg Fwd
Reg Fwd
ALU
N
Z
32
16
8
Write 16
Write 8
address
Read 16
Read 8
0 Ext
16
16
ALU Out
32
[7:0]
[3:0]
OP2 Address [7:4]
0 Ext
Sign Ext
R15 Address
[11:8] OP1 Address
[7:4] OP2 Address
Register Forwarding

ALU:
Contains 2 inputs, one 3-bit control line, and one output with the N and Z flags. The output is 32 bits with both inputs being 16 bits. The functions of the ALU are listed below.

Buffers:
Buffers will hold data, addresses, and control being used by the pipeline in each stage. The control consists of Write, Memory, and Execute with respect to the relative stage. Buffer 1 has one control line for a total of 33 bits. Buffer 2 contains the Write, Read, and Execute control lines with a total of 114 bits. Buffer 3 contains the Write and Memory control lines for a total of 93 bits, and Buffer 4 has the Write control line for a total of 71 bits.

Hazard detect:
Contains 4 inputs and 3 outputs which are used to stop the flow of data in the case a hazard is detected. This is done by zeroing out the buffer, disabling PC, and disabling the IF/D buffer for that cycle.

Reg forward:
7 inputs and 3 outputs. Monitors OP1, OP2, and R15 addresses for the current instruction vs their addresses in the memory and write back stages specifically to avoid data dependencies. All seven inputs are addresses, the outputs are control lines for MUXes to the ALU that select what version of that operand to use.

Path control:
One input one output. Input is the combination of opcode and function bits, output is the interpreted control bits for EX, MEM, and WB. These bits then later control MUX, ALU, and enable selections.

Jump control:
Tapped directly off of the opcode section of our instruction. Checks for equality with the opcode for jump and outputs assert if true.

Instruction memory:
One input, one output. The input is for an address provided by PC, the output is for the instruction.

Reg block:
Has six inputs and four outputs with two control lines.Two of the inputs are for reading operand addresses, two are for reading write addresses, and the remaining two are for writing data. Three of the outputs are for reading data out of R15, operand 1, and operand 2, the final output is for reading R15s address. Both control lines are for enables, one for writing to the register provided by operand 1 and the other for operand 2.

Data memory:

Has 3 inputs, 2 outputs, and three control lines. One control line is to toggle between eight bit and sixteen bit modes. The next line is to toggle between read and write modes, and the final line is to enable or disable the device entirely.

Instruction	Jump control	W			M				EX						
add	1	1	1	0	1	x	x	1	1	0	0	0	0	1	x
sub	1	1	1	0	1	x	x	1	1	0	0	1	0	1	x
mul	1	1	0	0	1	x	x	1	1	0	1	0	0	1	1
div	1	1	0	0	1	x	x	1	1	0	1	1	0	1	1
mv	1	1	1	0	1	x	x	1	1	1	1	1	0	1	x
swp	1	1	0	0	1	x	x	1	1	1	0	0	0	1	0
andi	1	1	1	0	1	x	x	1	1	1	0	1	1	0	x
or	1	1	1	0	1	x	x	1	1	1	1	0	1	0	x
lbu	1	0	1	0	1	0	1	0	0	0	0	0	0	1	x
sb	1	1	1	1	1	0	0	0	0	0	0	0	0	1	x
lw	1	0	1	0	1	1	1	0	0	0	0	0	0	1	x
sw	1	1	1	1	1	1	0	0	0	0	0	0	0	1	x
blt	1	1	1	1	0	x	x	1	1	0	0	1	0	0	x
bgt	1	1	1	1	0	x	x	1	1	0	0	1	0	0	x
beq	1	1	1	1	0	x	x	1	1	0	0	1	0	0	x
jmp	0	1	1	1	1	x	x	1	x	x	x	x	x	x	x
halt	1	1	1	1	1	x	x	1	x	x	x	x	x	x	x
	(Active low)	Memoralu	Write2 (activelow)	Write1 (activelow)	Branch (activelow)	Eightbit (activelow)	R/~WR (activelow)	Memenable (activelow)	signext	ALU			Secondin		WBop2

	ALU		
add	0	0	0
sub	0	0	1
mul	0	1	0
div	0	1	1
swp	1	0	0
and	1	0	1
or	1	1	0
OP2	1	1	1

Hazard detect:

If lw destination address = source address of next instruction

Deassert control, pc, IF buffer

Else

Normal operation

Reg forward:

		MUX A
If D/EX OP1 addr matches (& being written)	EX/M OP1 addr	001
	M/WB OP1 addr	010
	EX/M OP2 addr	011
	M/WB OP2 addr	100
	EX/M R15 addr	101
	M/WB R15 addr	110
	O.W.	000

MUX B

If D/EX OP2 addr matches (& being written)	EX/M OP1 addr	001
	M/WB OP1 addr	010
	EX/M OP2 addr	011
	M/WB OP2 addr	100
	EX/M R15 addr	101
	M/WB R15 addr	110
	O.W.	000

MUX C

If D/EX R15 addr matches (& being written)	EX/M OP1 addr	001
	M/WB OP1 addr	010
	EX/M OP2 addr	011
	M/WB OP2 addr	100
	EX/M R15 addr	101
	M/WB R15 addr	110
	O.W.	000

2	x	8’→16’ zero extend	Ehsan
2	x	8’ →16’ sign extend	Ehsan
1	x	12’→16’ zero extend	Ehsan
1	x	16’→32’ ALU	Drew
3	x	16’ Adders	Ehsan
4	x	buffers	Ehsan
1	x	hazard detect	Drew
1	x	reg forward	Drew
1	x	path control	Ehsan
1	x	jump control	Ehsan
1	x	PC	Ehsan
1	x	instruction memory	Drew
1	x	reg block	Drew
1	x	data memory	Drew
7	x	2:1 MUX	Ehsan
4	x	4:1	Drew
1	x	left shifter	Ehsan