Lab 3C – Supporting R Type (ALU Instructions)

We want to extend Lab 3B to support R Type or ALU instructions. R Type instructions are of the form ADD X1, X2, X3. This means Register X2 and Register X3 must be added and the result must be written in Register X1. In Lab 3B, we obtained the results of some ALU operations but they were not *put back to the Register File*. We have to close the loop. We have to take the ALU results and put them into the Register File. The book shows this operation by a feedback path in say, Fig.4.1 (p.258) where the ALU result is taken back to Register File by a path on the top and gets into Register File as Data.

64

64

2

ALUOp

116

Opcode field

Read register Address 1

Read Data 1

data1

64

5

Read Register Address 2

Zero

A

5

5

ALU result

Read Data 2

Data2

Write register Address

B

4

Write Data

ALU operation

64

clock

RegWrite

The format for an R Type instruction in machine language is given in p. 274, Fig.4.14 top most entry. If ADD X1, X2, X3 is the instruction in Assembly language, we find that the bits are arranged as follows:– From left, 11 bits of Opcode, then 5 bits for pointing the second source register X3 (Rm), 6 bits to indicate the shift amount (we will not implement shift instructions and so we can ignore them), 5 bits for pointing the first source register X2 (Rn) and 5 bits to point to the destination register X1 (Rd).

destreg

Src reg1

shamt

Src reg2

Opcode

5 (Rn)

5 (Rm)

6

5 (Rd)

11

5

64

Read register Address1

Rn

Zero

A

ALU result

Read Register Address 2

Rm

5

Rd

Write register (Address)

5

B

2

ALUOp

4

116

Opcode field

Write Data

ALU operation

64

clock

RegWrite

In the figure above, the two fields of the source registers are going to the Register File as two Read Addresses. The 11 bits of Opcode field with 2 bits of ALUOp are decoded to give us the 4 bit ALU operation control input.

Modify Lab 3 B so that the ALU results reach the Register File again and get written. You may write the 4 results in Registers 1 to 4 (clock must be raised for each write). You can display the results by giving different Read Addresses and checking that the data appear at the data outputs of the Register File. A timing diagram will be very helpful. You may read out say registers 1 and 2 after they are written and 3 and 4 afterwards.

mc, s17