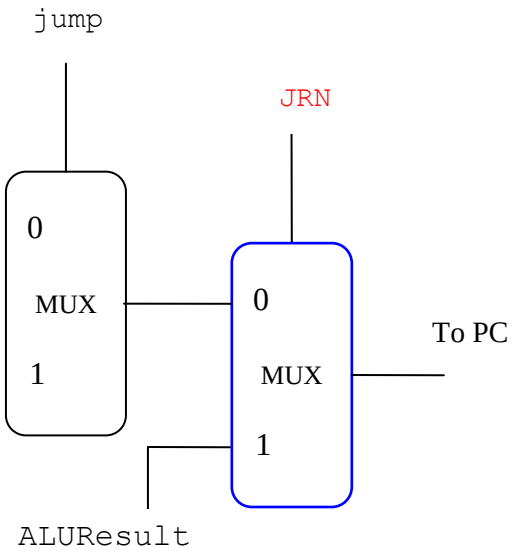
Practice Problem Set – Single Cycle Datapath

(**Solution**)

1. **Solution**

The datapath has already support for addition of registers identified by **`rs`** and **`rt`**. However, for the transfer of control to the address (**`rs`** + **`rt`**), we need to make the following **change**.

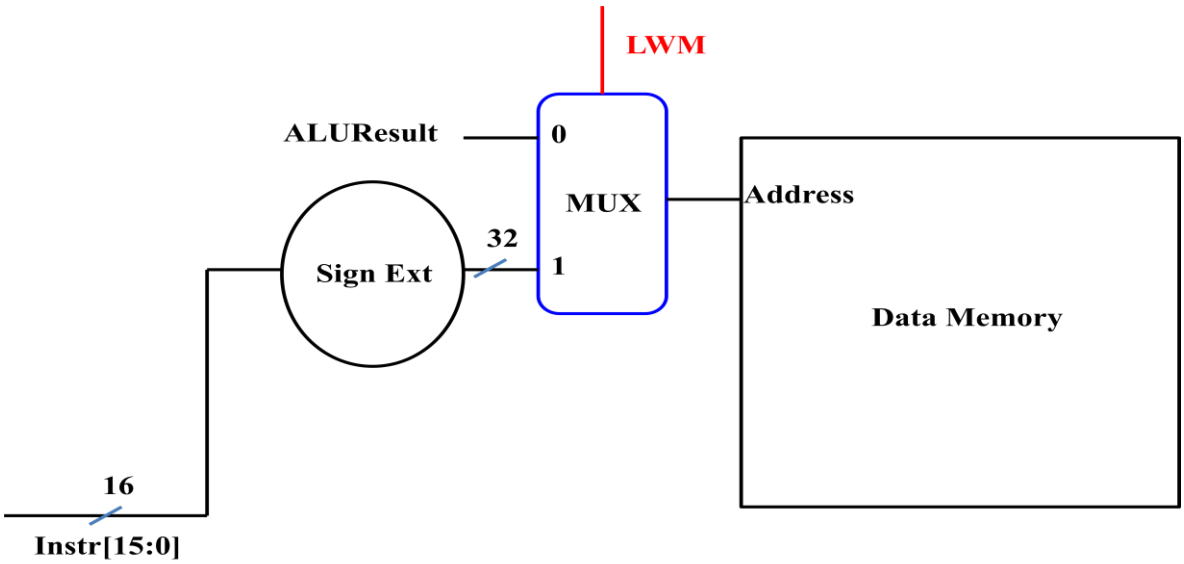


Control Signals

RegWrite	RegDst	MemtoReg	MemWrite	MemRead	ALUSrc	ALUOP	Branch	Jump	JRN
0	x	x	0	0	0	00	0	0	1

2. **Solution**

The sign extension of **`address`** field is already supported in the datapath. Retaining **`lw`**/**`sw`** instructions, while adding support for this new instruction requires following **change** in the datapath.



The datapath already supports contents of data memory to be loaded in the register **`rt`**.

Control Signals

RegWrite	RegDst	MemtoReg	MemWrite	MemRead	ALUSrc	ALUOP	Branch	Jump	LWM
1	0	1	0	1	x	xx	0	0	1

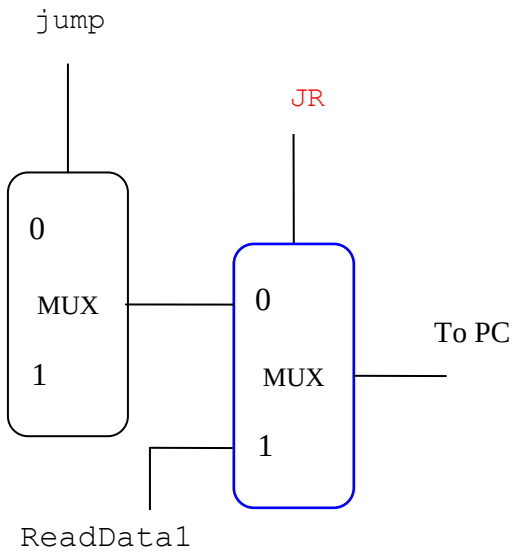
3. **Solution**

No change is warranted in the datapath.

Control Signals

RegWrite	RegDst	MemtoReg	MemWrite	MemRead	ALUSrc	ALUOP	Branch	Jump
1	1	1	0	1	0	10	0	0

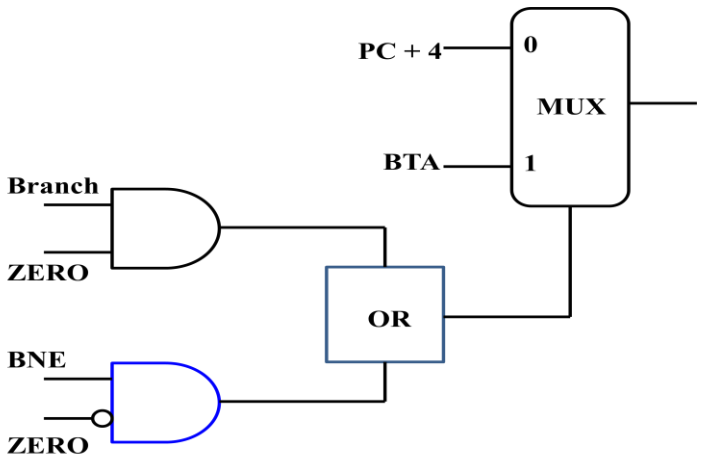
4. **Solution**



Control Signals

RegWrite	RegDst	MemtoReg	MemWrite	MemRead	ALUSrc	ALUOP	Branch	Jump	JR
0	x	x	0	0	x	xx	0	0	1

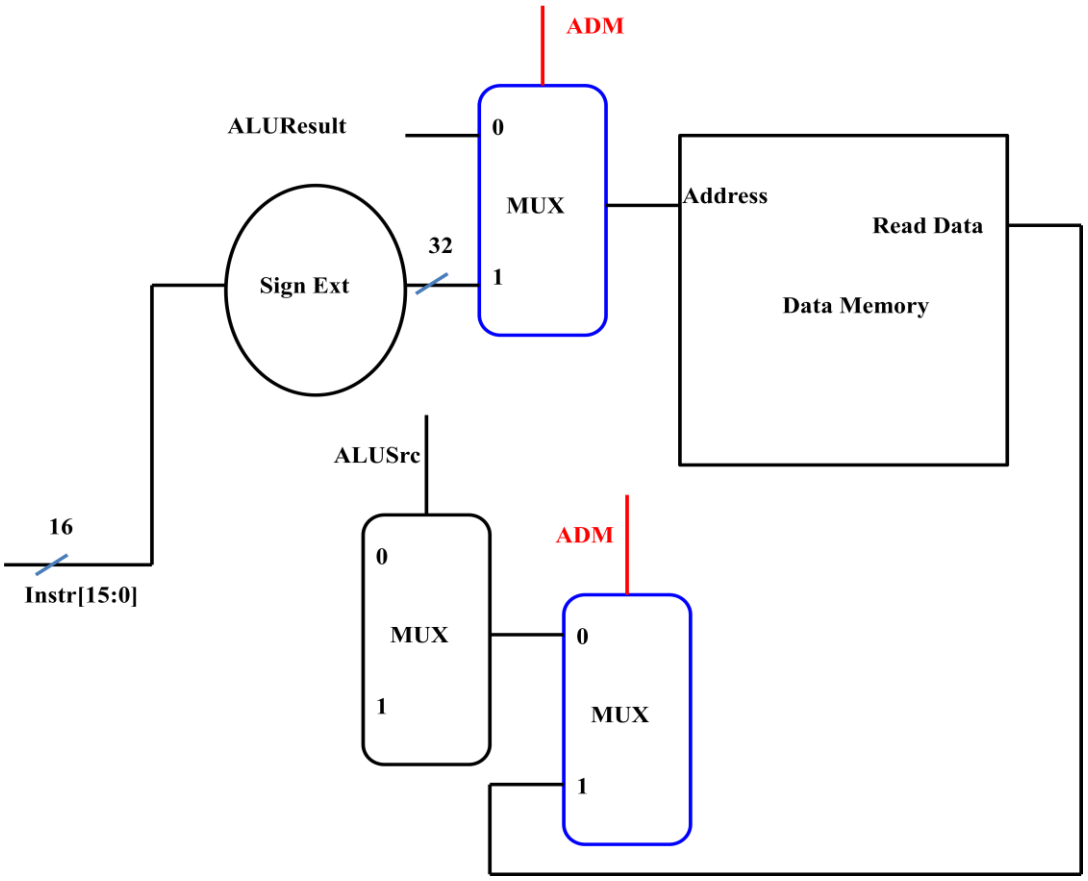
5. **Solution**



Control Signals

RegWrite	RegDst	MemtoReg	MemWrite	MemRead	ALUSrc	ALUOP	Branch	Jump	BNE
0	x	x	0	0	0	01	0	0	1

6. **Solution**

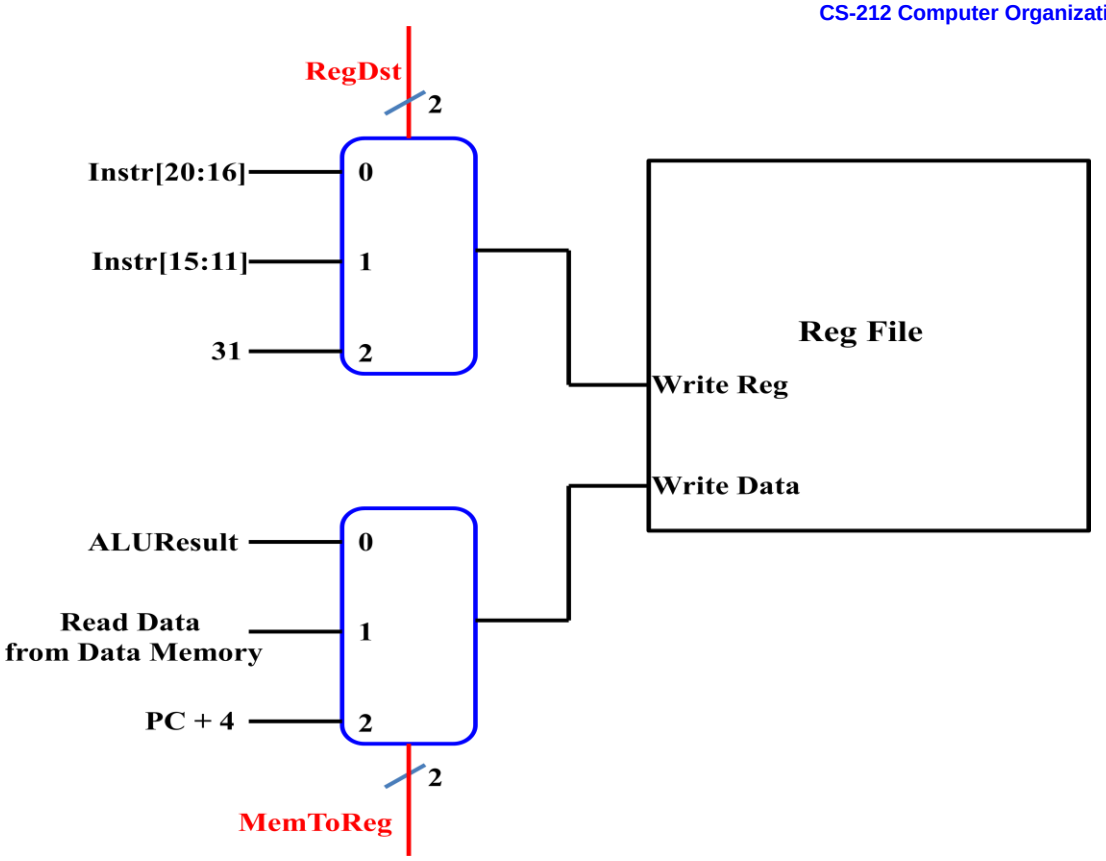


Control Signals

RegWrite	RegDst	MemtoReg	MemWrite	MemRead	ALUSrc	ALUOP	Branch	Jump	ADM
1	0	0	0	1	x	00	0	0	1

7. **Solution**

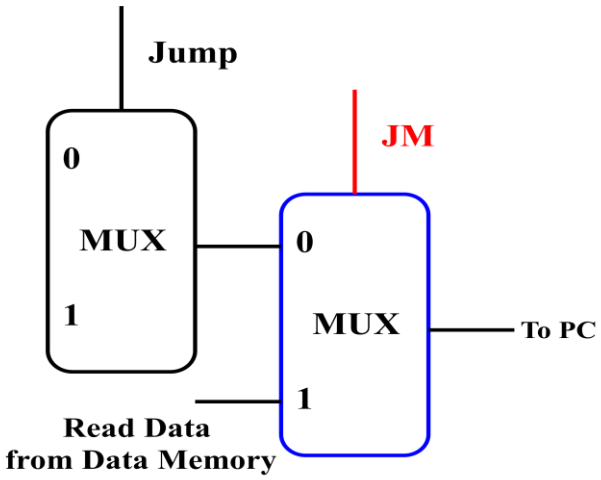
The **`jal`** instruction performs two actions: (1) saves return address (address of the sequential instruction following the **`jal`** instruction) in register **`$ra`** (its identifier is 31) and (2) transferring control to the address of called procedure or function (this address is computed by the hardware in the same way as it does for the **`j`** instruction). The second action is already supported in the datapath. However, for action (1), following **changes** are warranted.



Control Signals

RegWrite	RegDst	MemtoReg	MemWrite	MemRead	ALUSrc	ALUOP	Branch	Jump
1	10	10	0	0	x	xx	0	1

8. **Solution**



Control Signals

RegWrite	RegDst	MemtoReg	MemWrite	MemRead	ALUSrc	ALUOP	Branch	Jump	JM
0	x	x	0	1	1	00	0	0	1