

## Design of the MIPS Processor (contd)

First, revisit the datapath for **add, sub, lw, sw**.  
We will augment it to accommodate the **beq**  
and **j** instructions.

### Execution of branch instructions

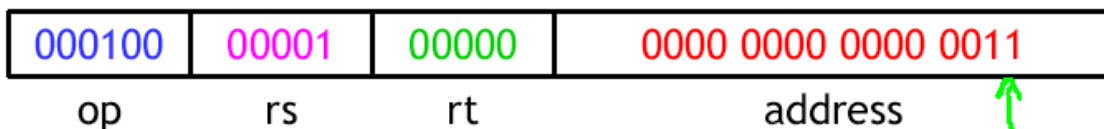
**beq \$at, \$zero, L**

add \$v1, \$v0, \$zero

add \$v1, \$v1, \$v1

j somewhere

**L:** add \$v1, \$v0, \$v0

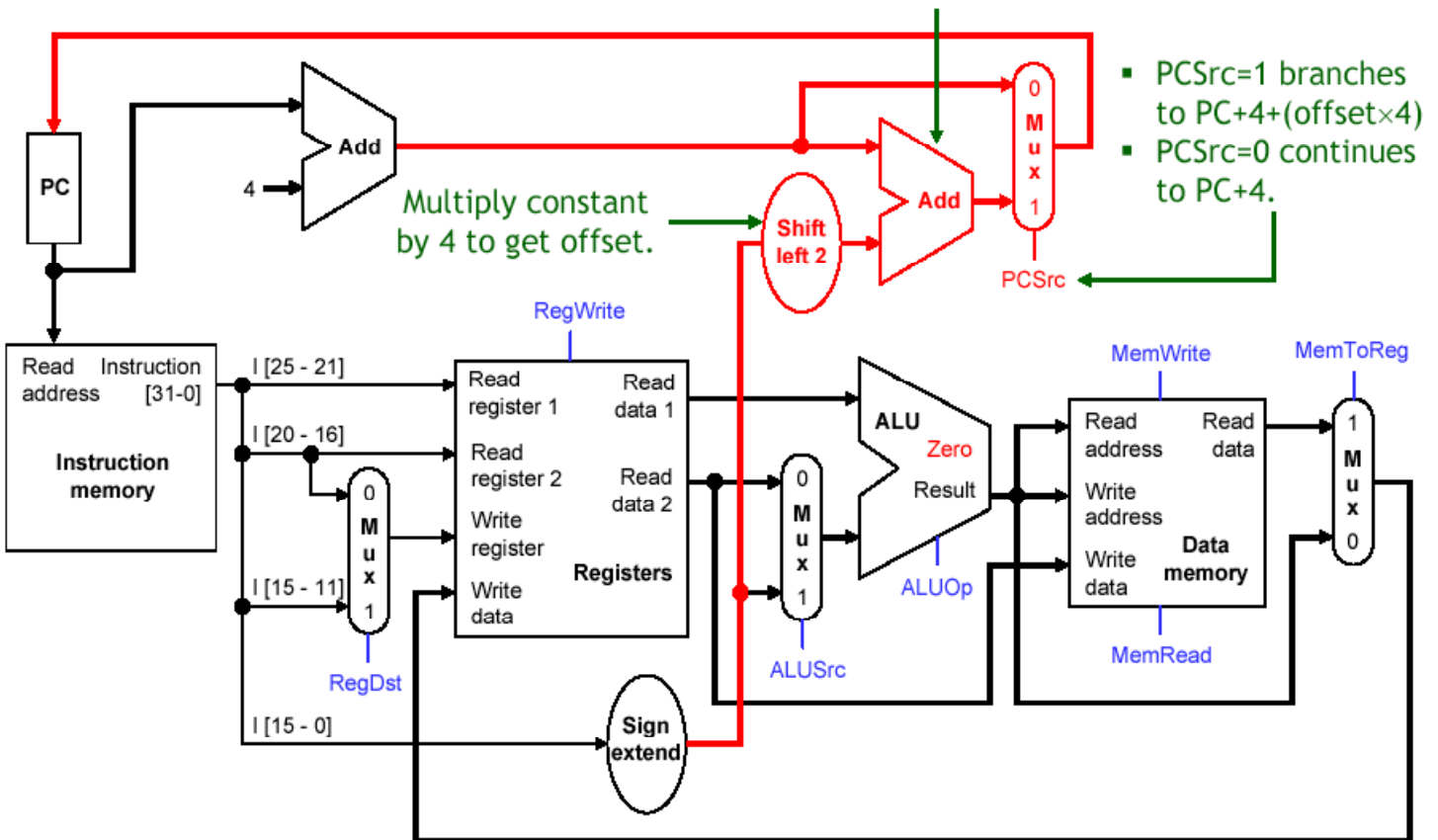


Offset=  $3 \times 4 = 12$

The offset must be added to the next PC to  
generate the target address for branch.

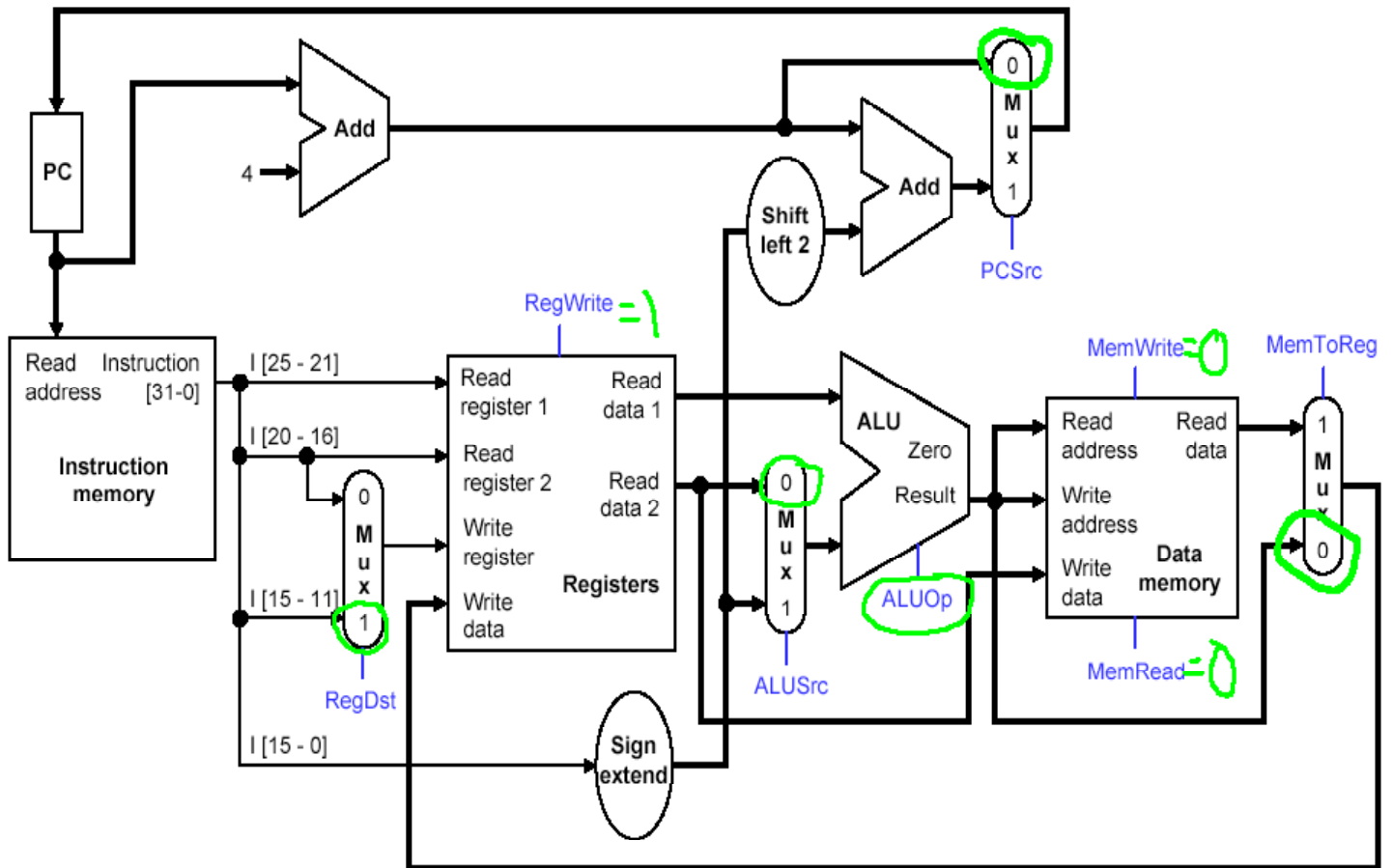
# The modified version of MIPS

We need a second adder, since the ALU is already doing subtraction for the beq.



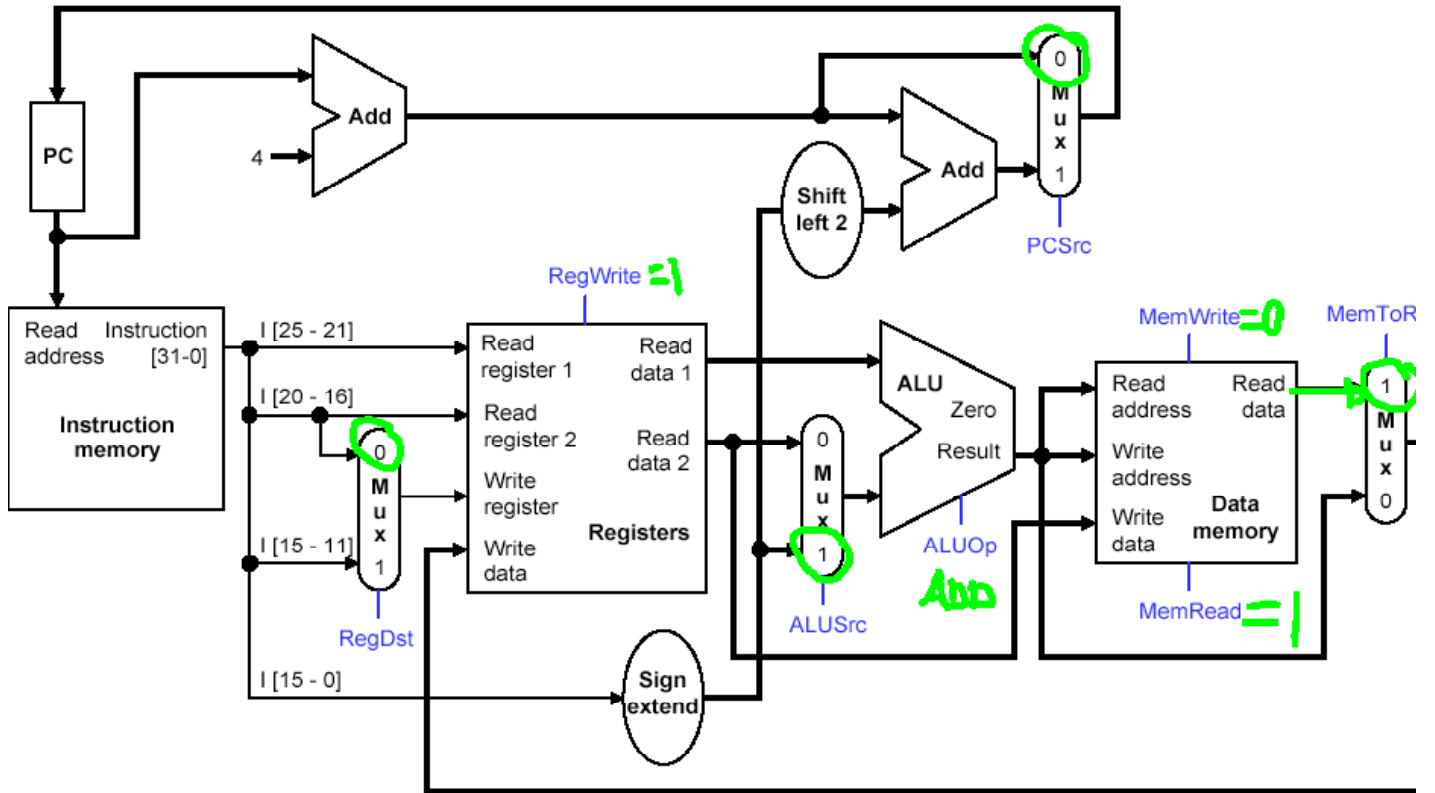
The final datapath for single cycle MIPS. Find out which paths the signal follow for **lw**, **sw**, **add** and **beq** instructions

# Executing R-type instructions



The **ALUOp** will be determined by the value of the **opcode** field and the **function** field of the instruction word

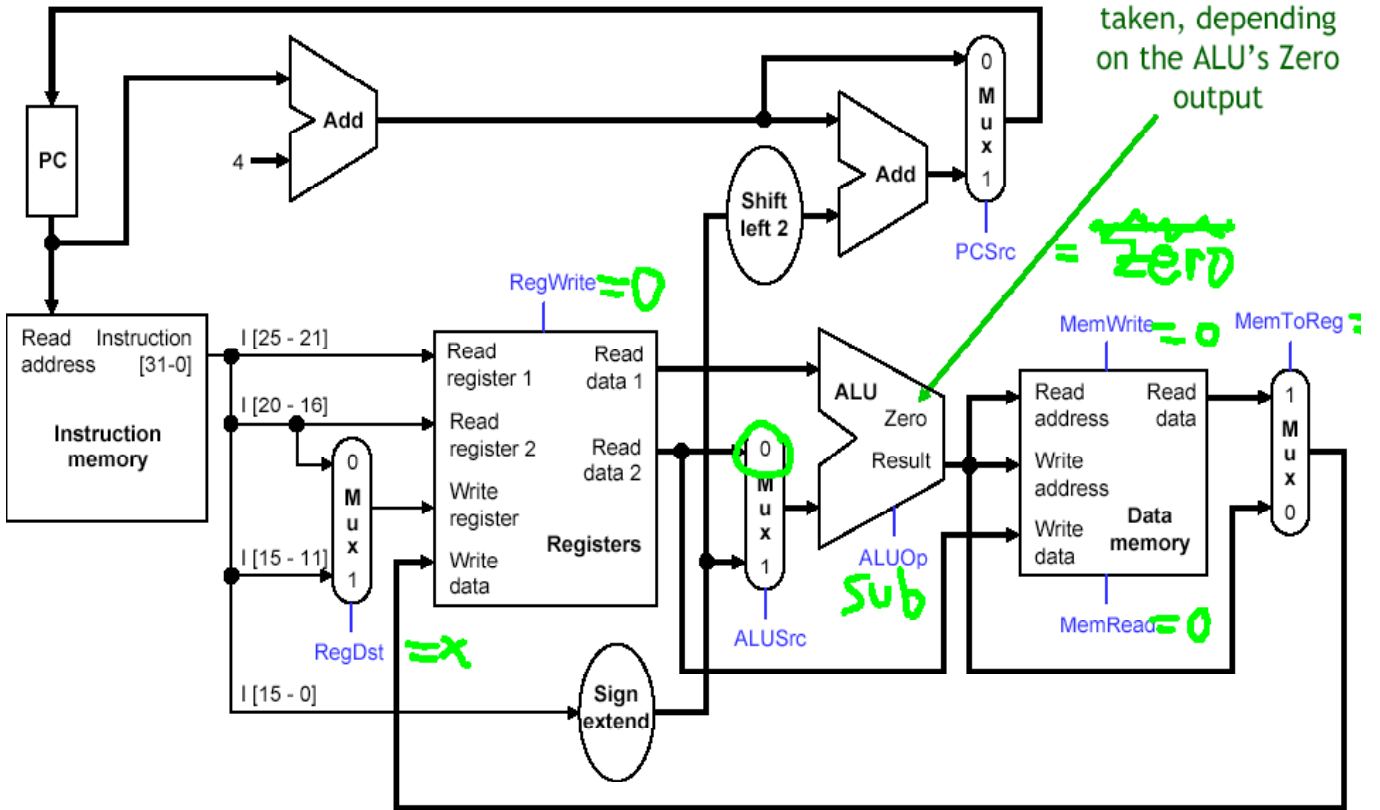
# Executing LW instruction



# Executing beq instruction

The branch may

or may not be taken, depending on the ALU's Zero output



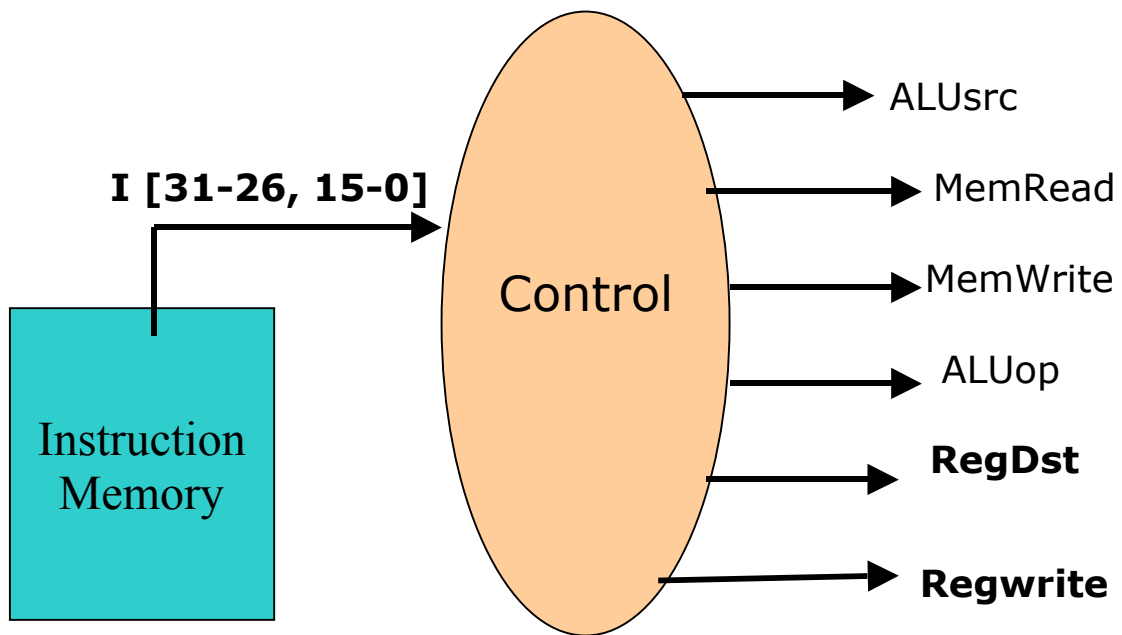
## Control signal table

This table summarizes what control signals are needed to execute an instruction. The set of control signals vary from one instruction to another.

Operation	RegDst	RegWrite	ALUSrc	ALUOp	MemWrite	MemRead	MemToReg
add	1	1	0	010	0	0	0
sub	1	1	0	110	0	0	0
and	1	1	0	000	0	0	0
or	1	1	0	001	0	0	0
slt	1	1	0	111	0	0	0
lw	0	1	1	010	0	1	1
sw	X	0	1	010	1	0	X
beq	X	0	0	110	0	0	X

How to implement the control unit?

# The Control Unit



**All control signals are not shown here**

## 1-cycle implementation is not used

Why? Because the length of the clock cycle will **always** be determined by the slowest operation (**lw, sw**) even when the data memory is not used. Practical implementations use **multiple cycles per instruction**, which fixes some shortcomings of the 1-cycle implementation.

Faster instructions are not held back by the slower instructions

The clock cycle time can be decreased.

Eventually simplifies the implementation of **pipelining**, the universal speed-up technique

This requires some changes in the datapath