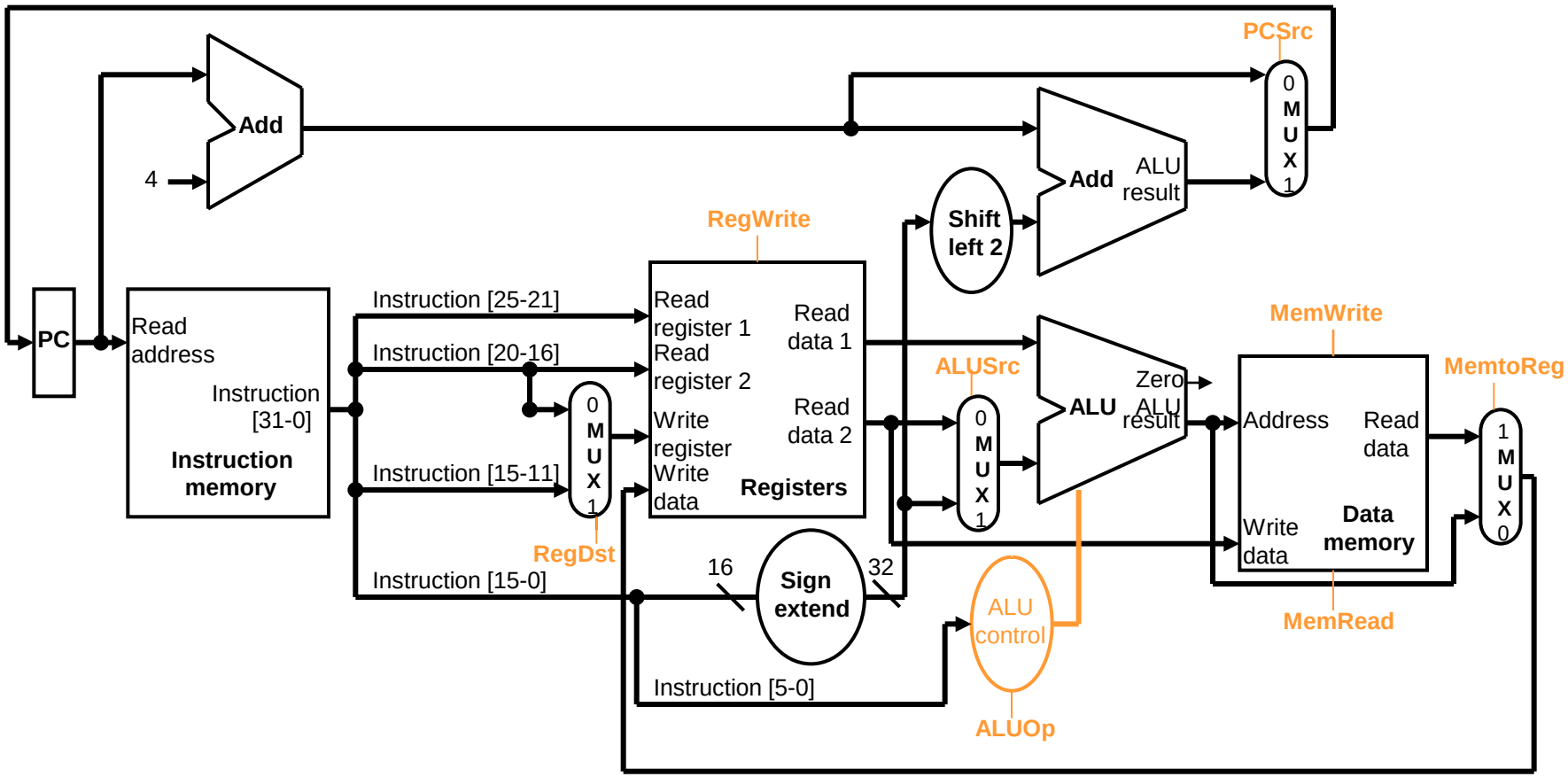
Computer Architecture

Single-Cycle Implementation

Single-Cycle Datapath



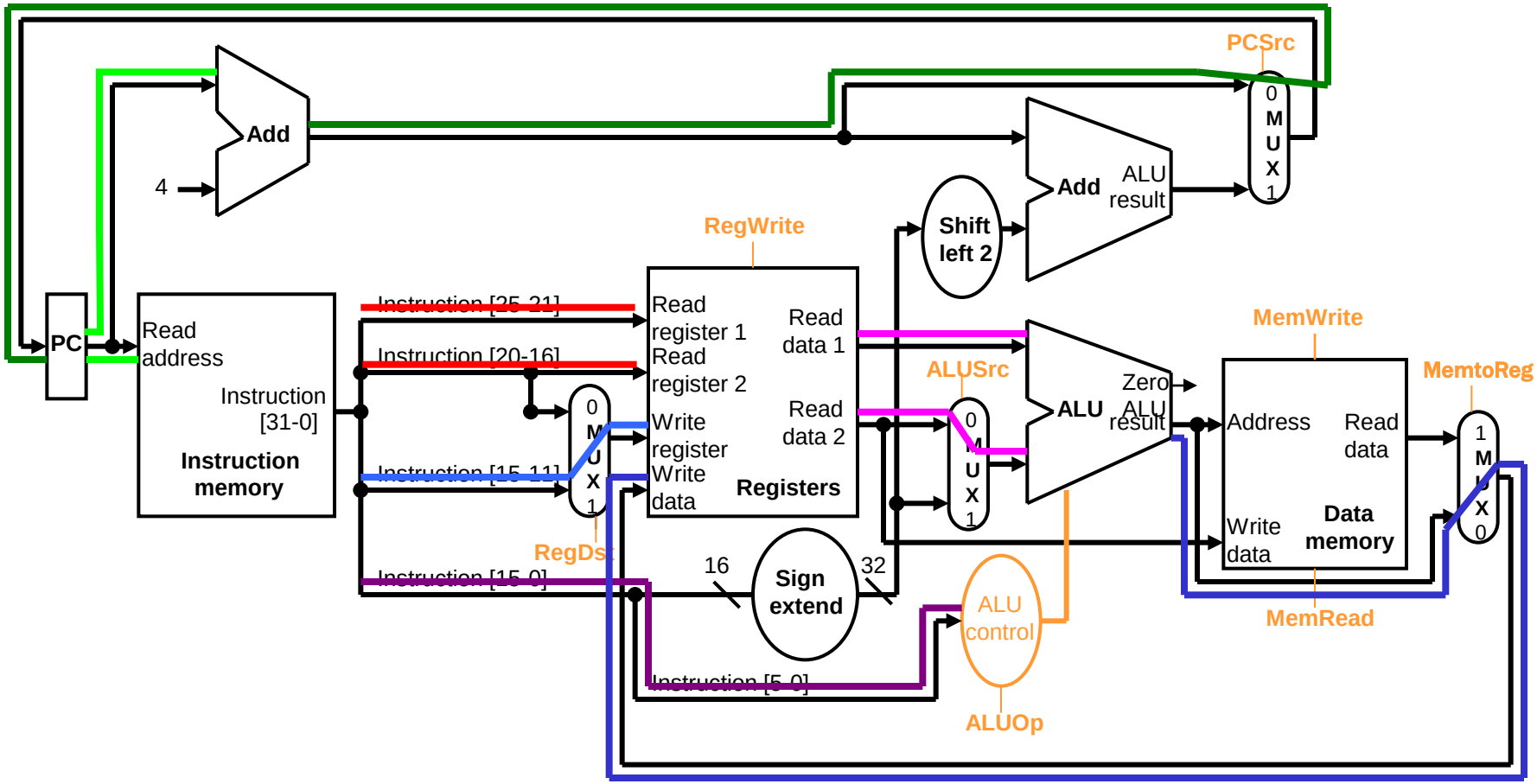
- This datapath supports the following instructions:
  - add, sub, and, or, slt, lw, sw, beq

Single-Cycle Control

RegDst	Select destination register
RegWrite	Specify if the destination register is written
ALUSrc	Select whether source is register or immediate
ALUOp	Specify operation for ALU
MemWrite	Specify whether memory is to be written
MemRead	Specify whether memory is to be read
MemtoReg	Select whether memory or ALU output is used
PCSrc	Select whether next PC or computed address is used

R-format Instruction Dataflow

- For add, sub, and, or, slt instructions



R-format Instruction Control

- Control signal summary

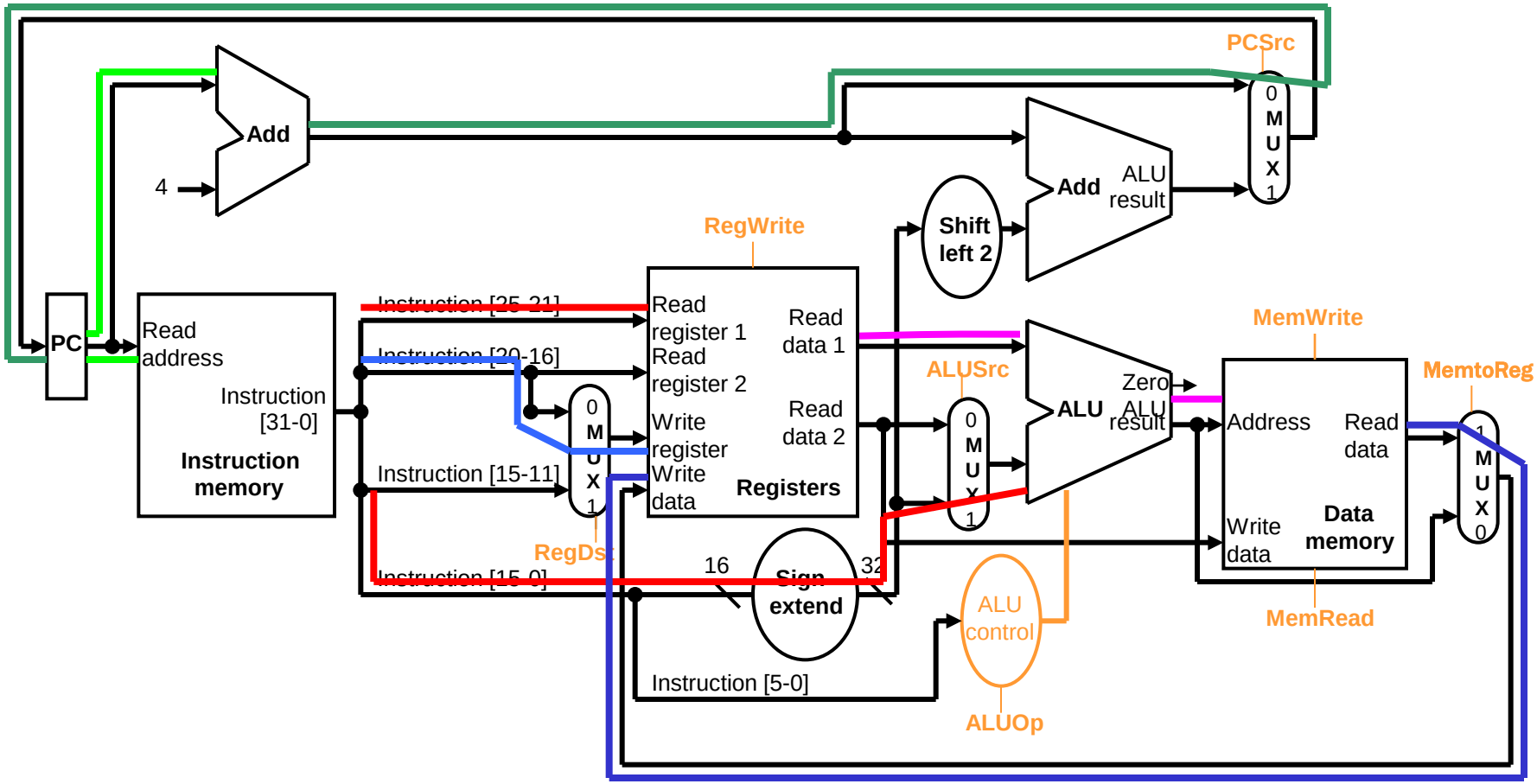
RegDst	1 to select Rd
RegWrite	1 to enable writing Rd
ALUSrc	0 to select Rt value from register file
ALUOp	*Dependent on operation* (see below)
MemWrite	0 to disable writing memory
MemRead	0 to disable reading memory
MemtoReg	0 to select ALU output to register
PCSrc	0 to select next PC

- ALUOp

add	OP (add)	and	OP (and)
sub	OP (sub)	or	OP (or)
slt	OP (slt)		

I-format Load Instruction Dataflow

- For lw instruction



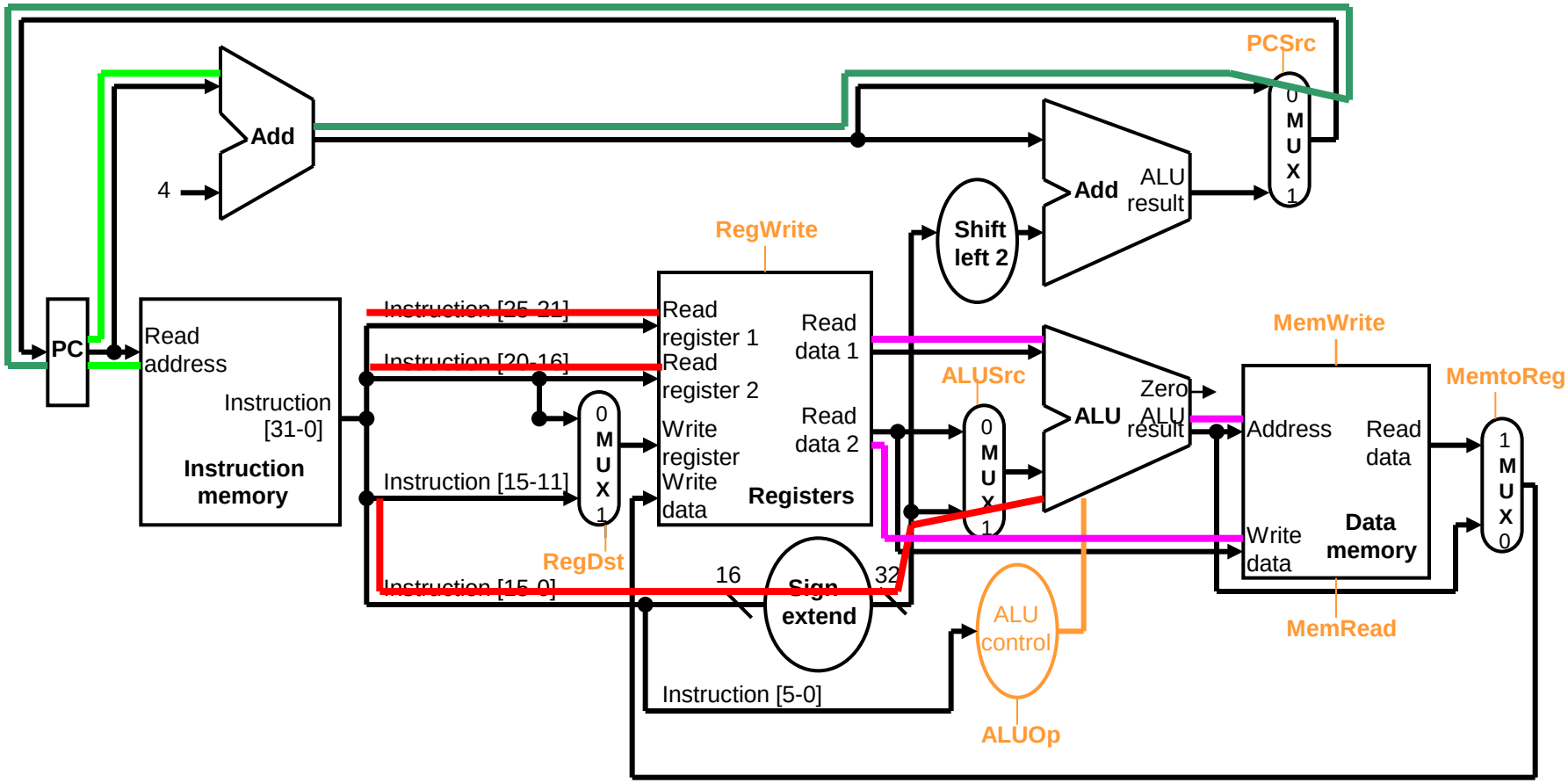
I-format Load Instruction Control

- Control signal summary

RegDst	0 to select Rt
RegWrite	1 to enable writing Rt
ALUSrc	1 to select immediate field value from instruction
ALUOp	add
MemWrite	0 to disable writing memory
MemRead	1 to enable reading memory
MemtoReg	1 to select memory output to register
PCSrc	0 to select next PC

I-format Store Instruction Dataflow

- For sw instruction



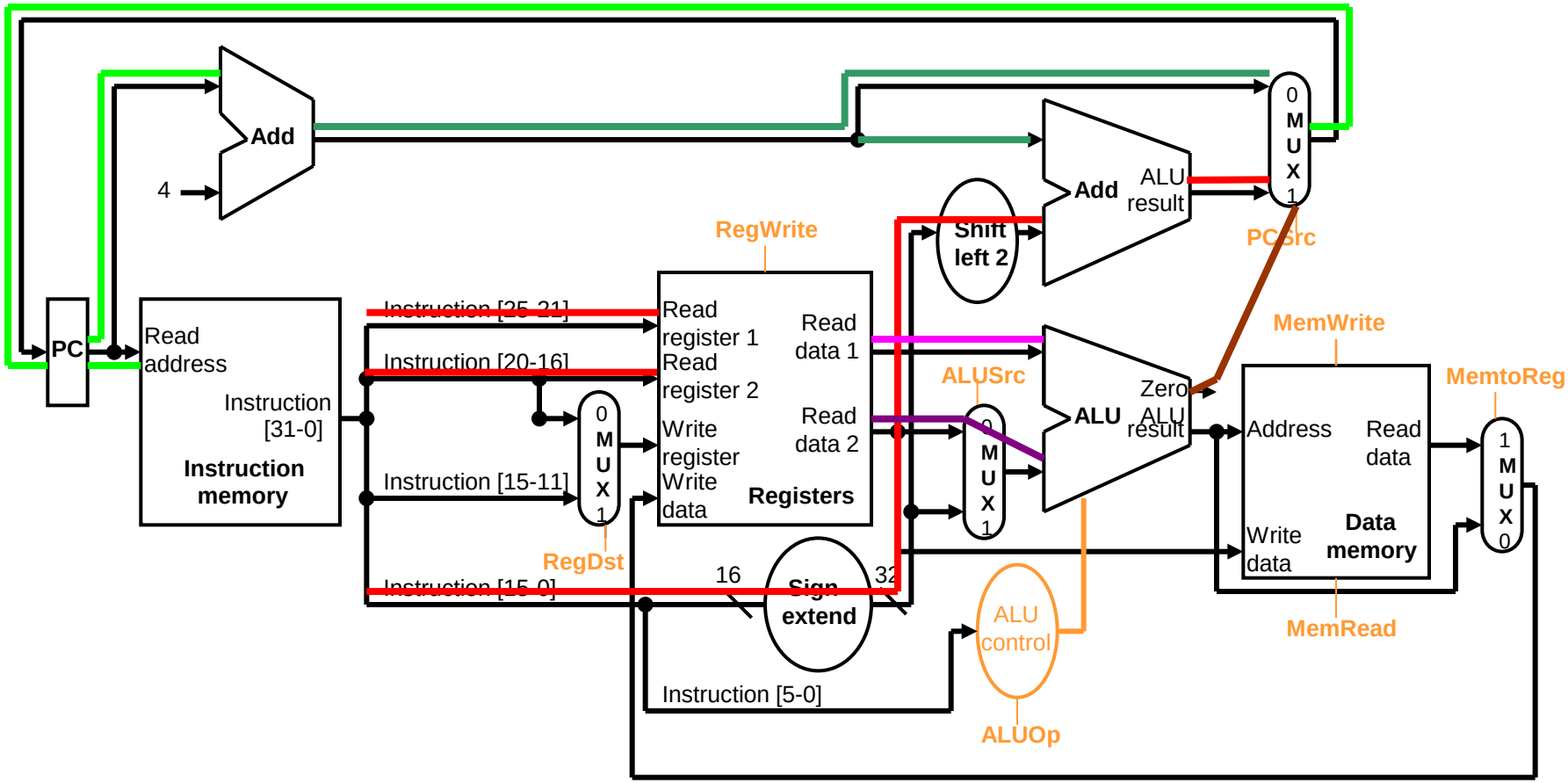
I-format Store Instruction Control

- Control signal summary

RegDst	x (don’t care)
RegWrite	0 to disable writing a register
ALUSrc	1 to select Rt value from register file
ALUOp	add
MemWrite	1 to enable writing memory
MemRead	0 to disable reading memory
MemtoReg	x (don’t care)
PCSrc	0 to select next PC

I-format Branch Instruction Dataflow

- For beq instruction



I-format Branch Instruction Control

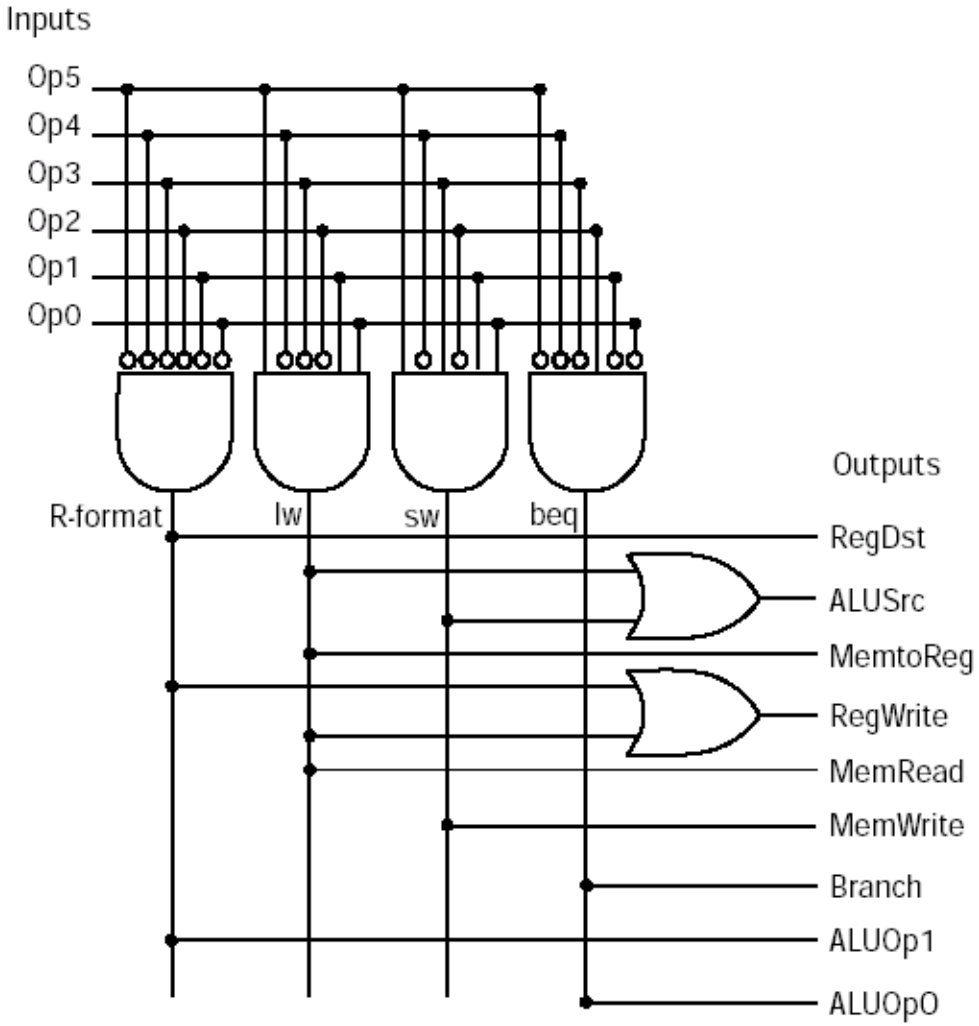
- Control signal summary

RegDst	x (don’t care)
RegWrite	0 to disable writing a register
ALUSrc	0 to select Rt value from register file
ALUOp	sub
MemWrite	0 to disable writing memory
MemRead	0 to disable reading memory
MemtoReg	x (don’t care)
PCSrc	zero

Single-Cycle Control Signals Summary

Signal	R-fmt	I-fmt (lw)	I-fmt (sw)	I-fmt (beq)
RegDst	1	0	x	x
RegWrite	1	1	0	0
ALUSrc	0	1	1	0
ALUOp	OP	add	add	sub
MemWrite	0	0	1	0
MemRead	0	1	0	0
MemtoReg	0	1	x	x
PCSrc	0	0	0	zero

More Details on Control Signal Generation



Review and Projection

- Single-Cycle implementation is easy
  - Control is based solely on the operation (and results!)
    - dictates ALU operation
    - controls multiplexor selection
    - enables/disables storage elements
  - Processor signals (control and data) stabilize and then any state (register and/or memory) change takes place when the clock cycle ends
- Multi-Cycle implementation explained in the next class
  - Instruction processing takes multiple steps, one step per cycle
  - Within a given clock cycle, signals stabilize and their local state change takes place when the clock cycle ends