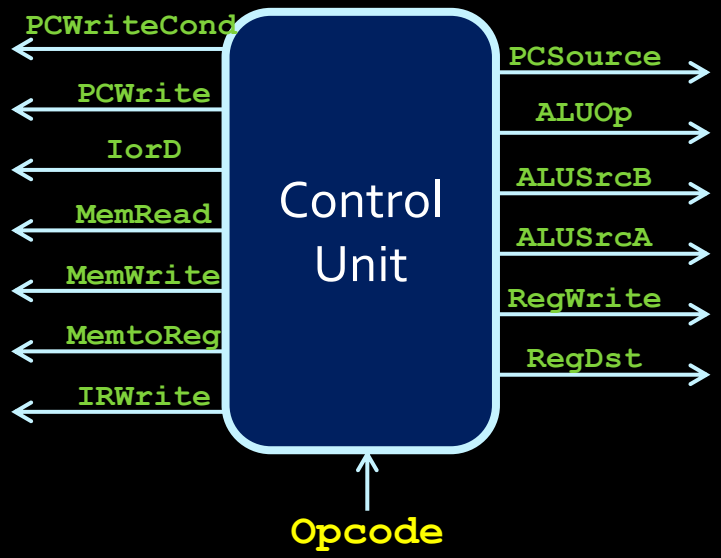
Week 7, part F: Datapath control

Controlling the Datapth

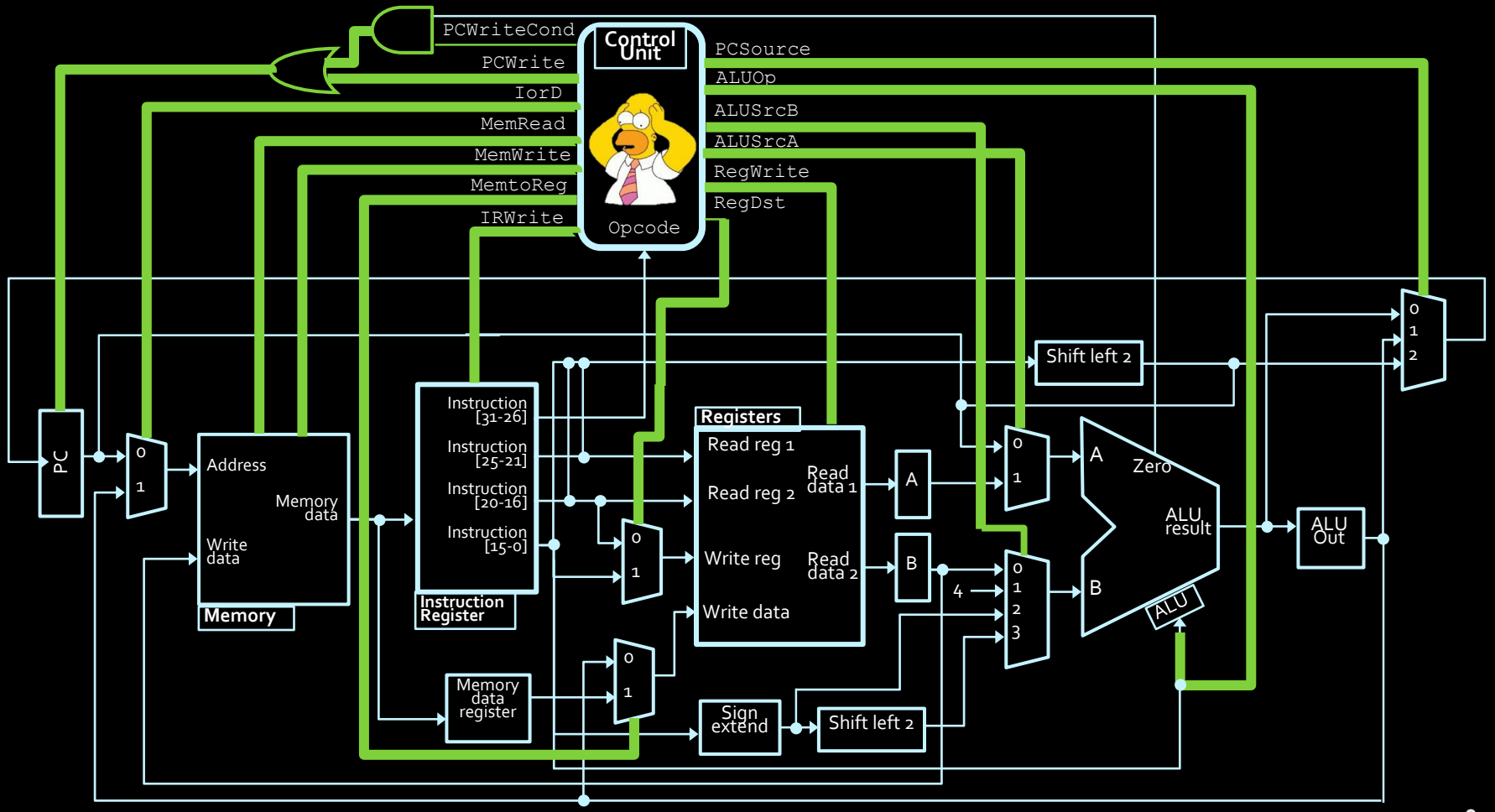
- Instructions are executed by turning various parts of the datapath on and off, to direct the flow of data from the correct source to the correct destination.
- The control unit needs to turn on these various components at the correct times.

Control unit

- The control unit takes in the opcode from the current instruction, and sends signals to the rest of the processor.
- Within the control unit is a finite state machine that can take multiple clock cycles for a single instruction.
  - The control unit sends out different signals on each clock cycle, to make the overall operation happen.

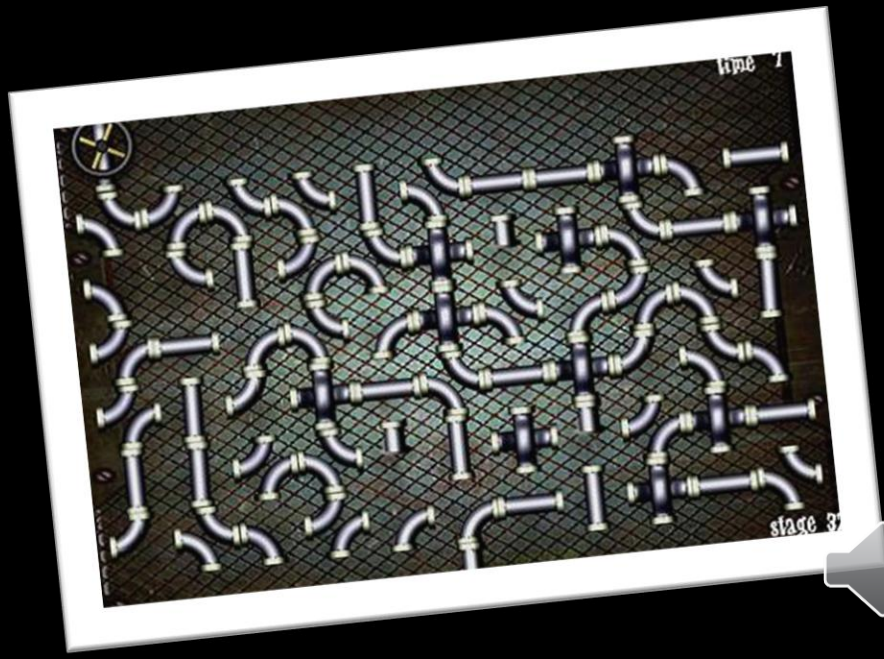


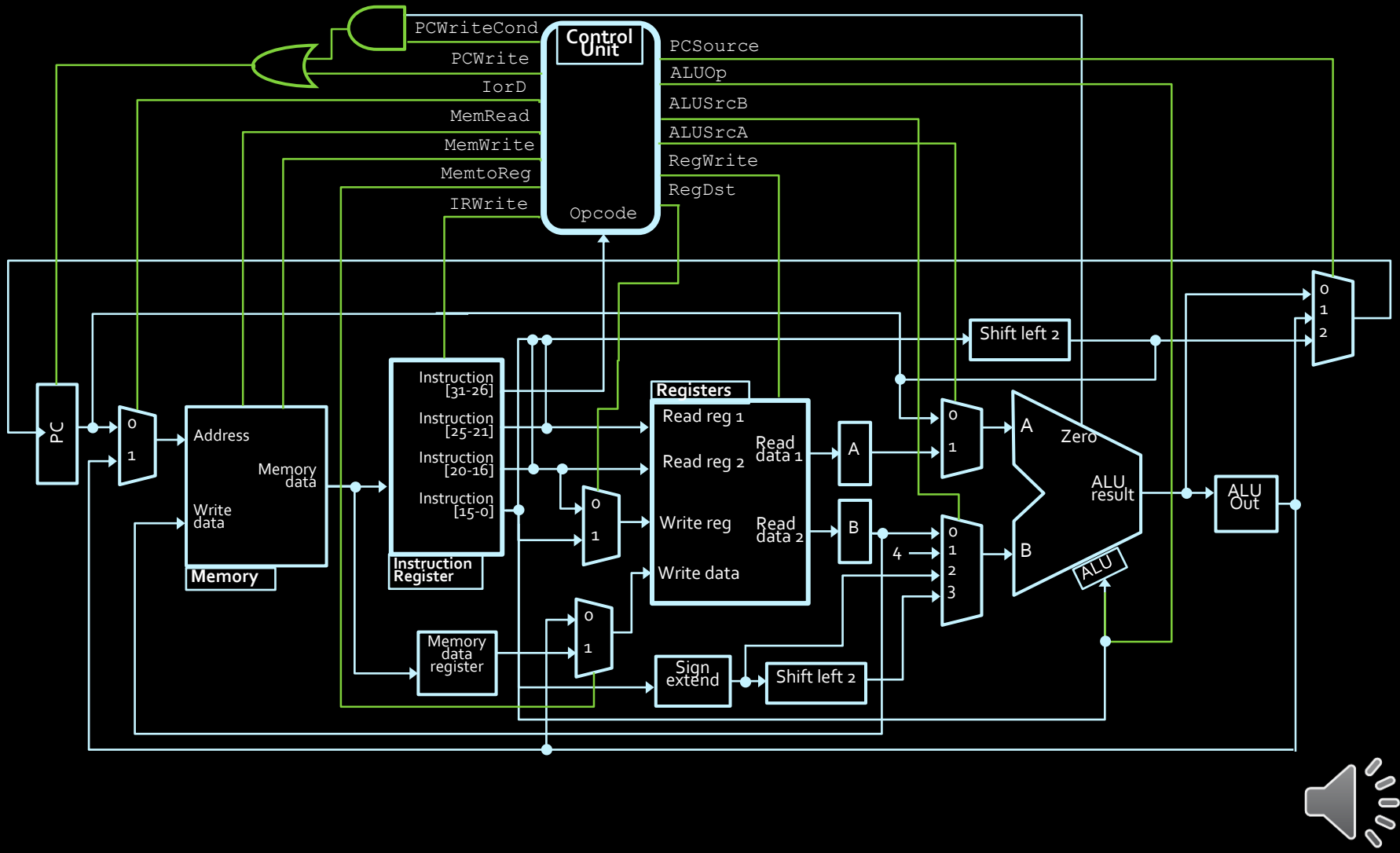
- The control unit sends signals (green lines) to various processor components to enact all possible operations.



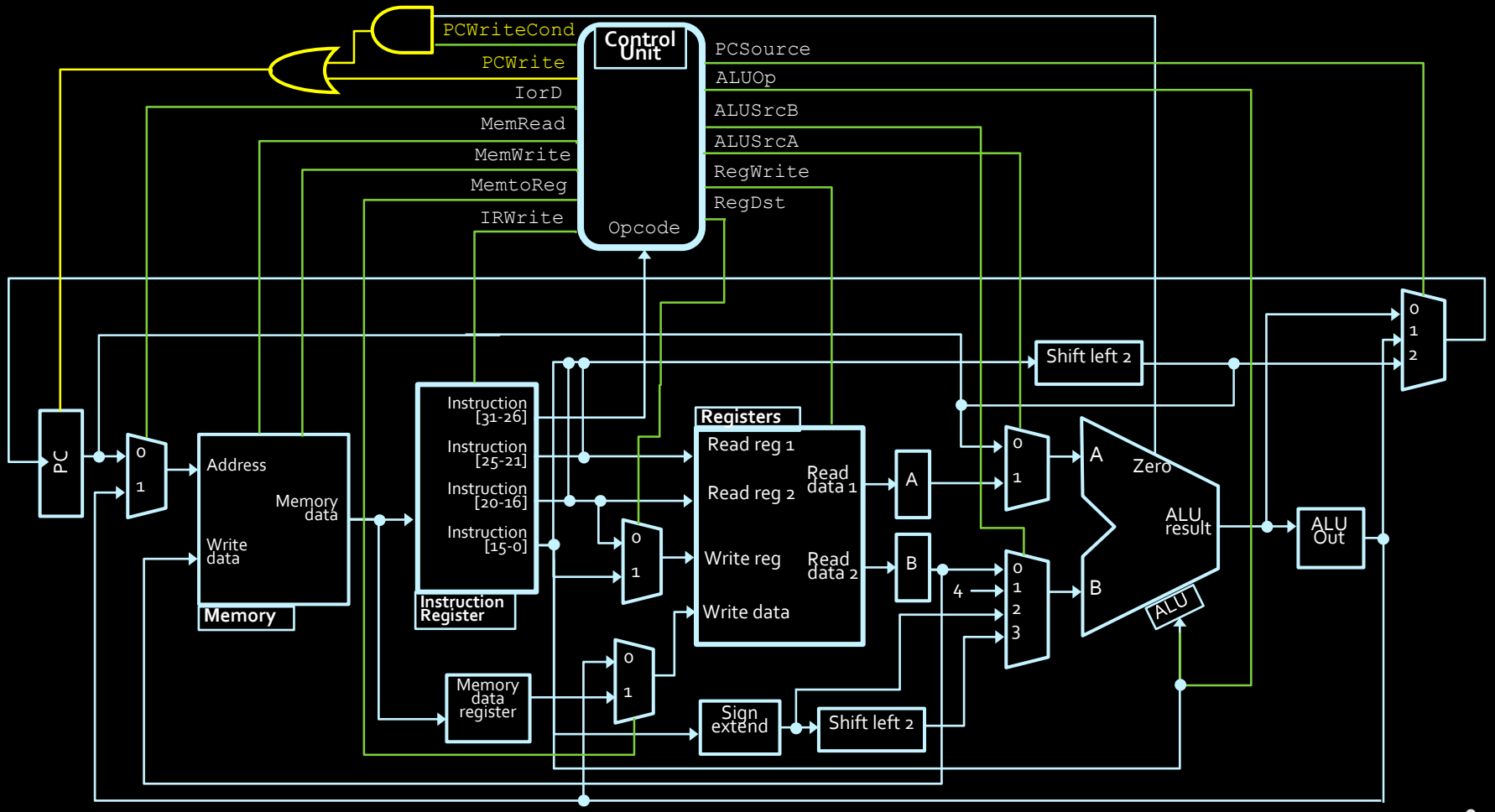
Signals → instructions

- A certain combination of signals will make data flow from some source to some destination.
- Just need to figure out what signals produce what behaviour.

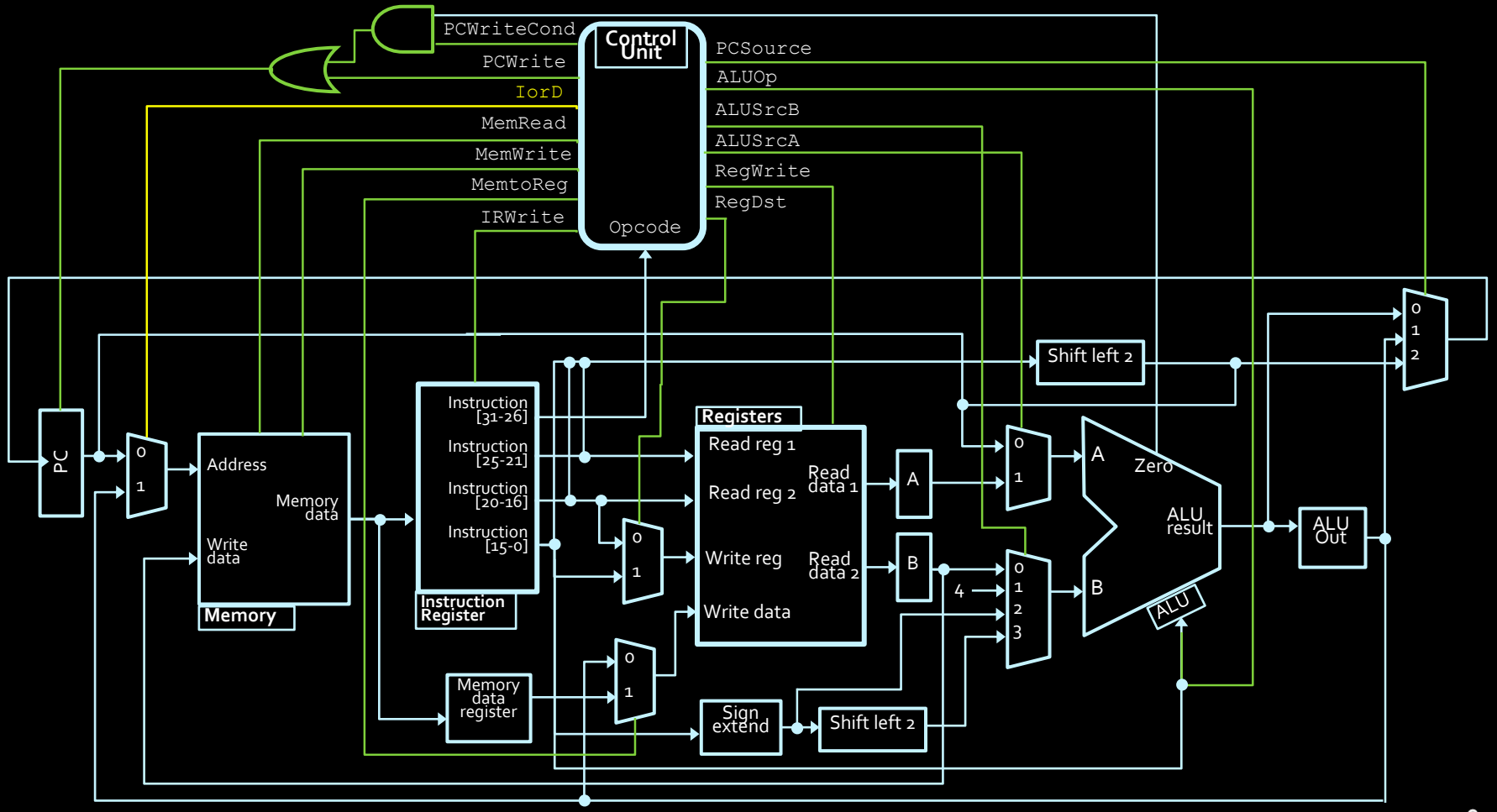



PCWriteCond
PCWrite
IorD
MemRead
MemWrite
MemtoReg
IRWrite
Control Unit
Opcode
PCSource
ALUOp
ALUSrcB
ALUSrcA
RegWrite
RegDst
PC
0
1
Address
Memory data
Write data
Memory
Instruction [31-26]
Instruction [25-21]
Instruction [20-16]
Instruction [15-0]
Instruction Register
Memory data register
Registers
Read reg 1
Read reg 2
Write reg
Write data
Read data 1
Read data 2
A
B
4
Shift left 2
Sign extend
Shift left 2
A
B
Zero
ALU result
ALU
ALU Out
0
1
2
3

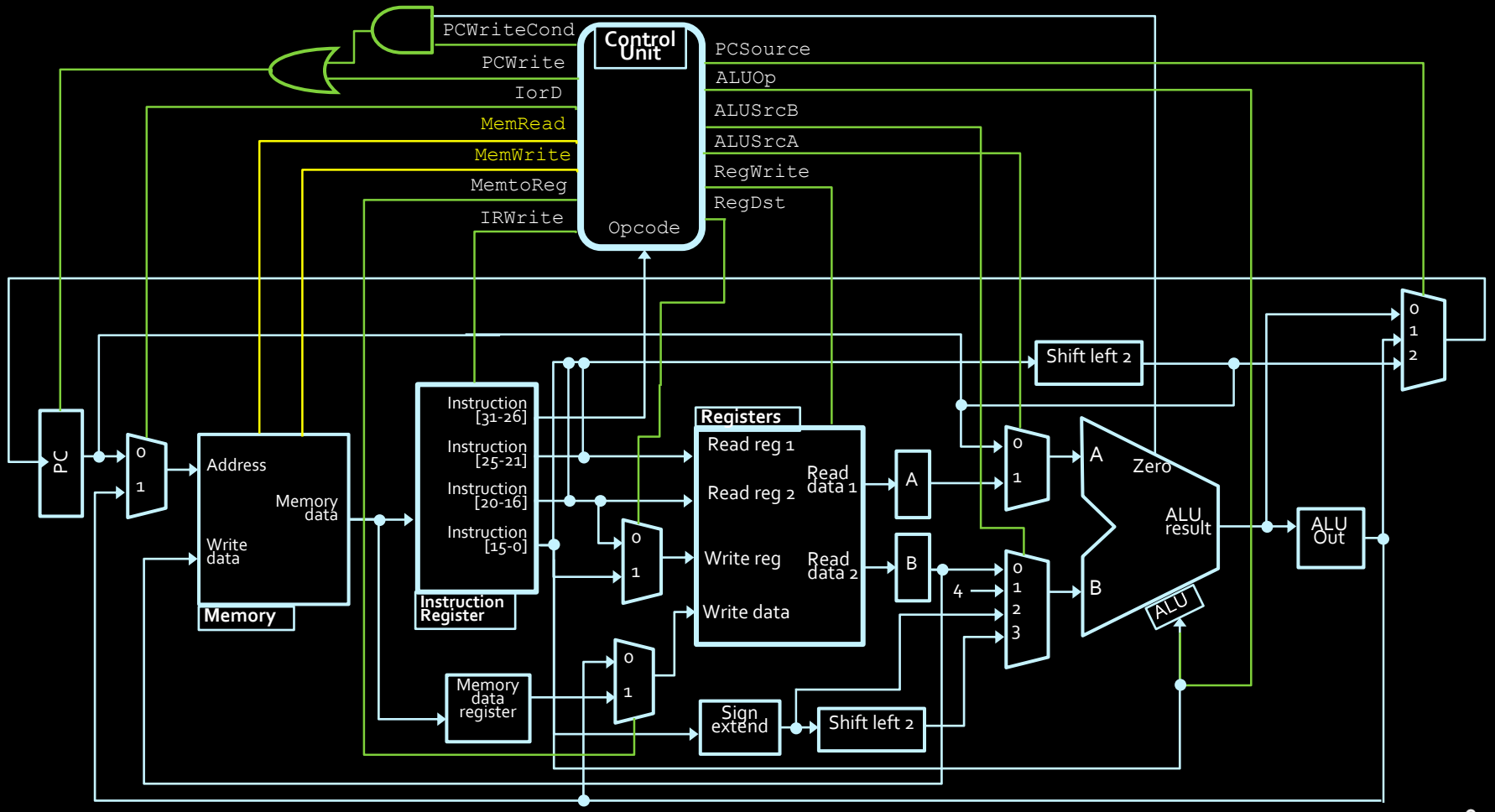
- PCWrite: Write the ALU output to the PC.
- PCWriteCond: Write the ALU output to the PC, only if the Zero condition has been met.



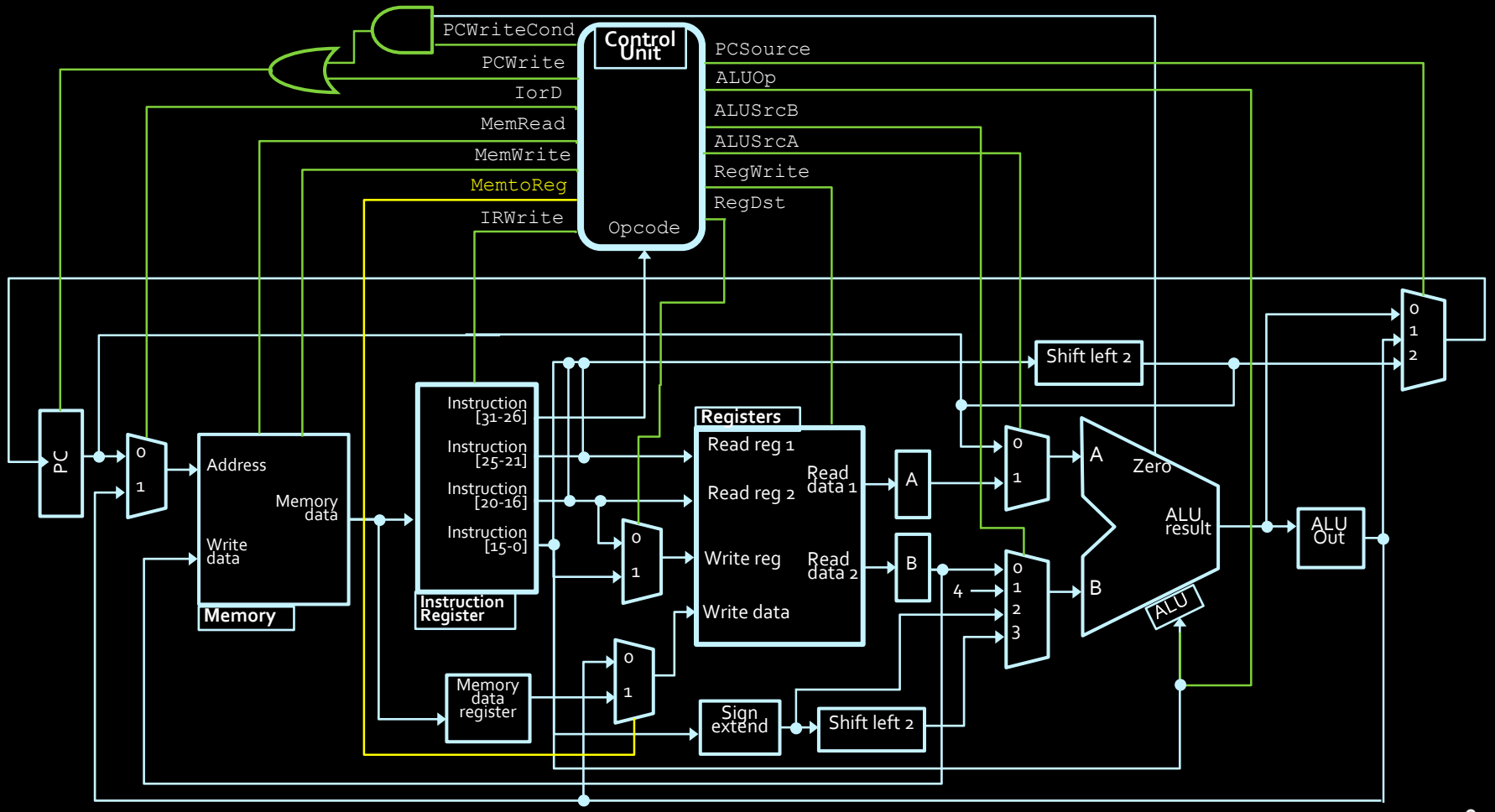
- IorD: For memory access; short for "Instruction or Data". Signals whether the memory address is being provided by the PC (for instructions) or an ALU operation (for data).



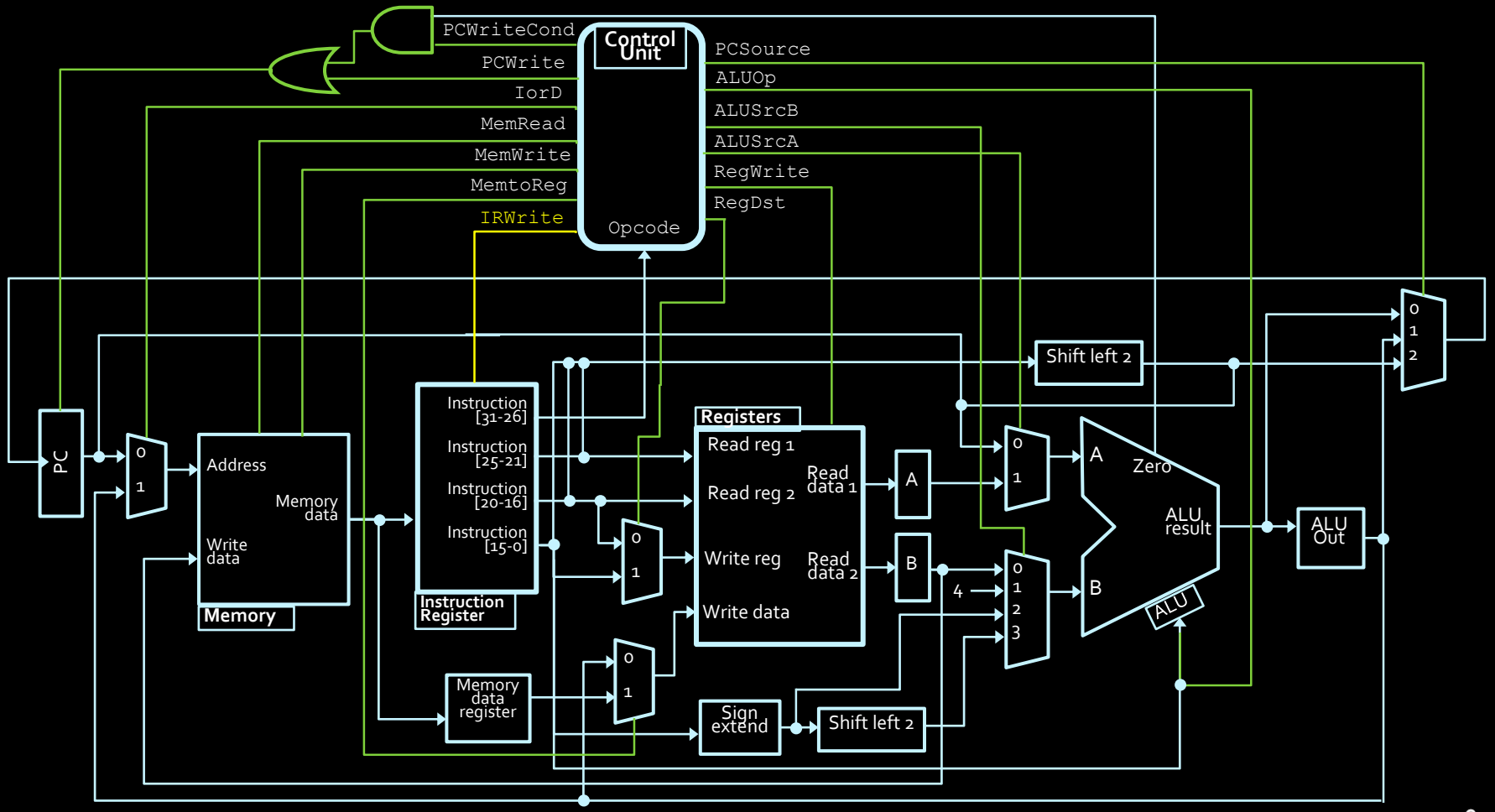
- MemRead: The processor is reading from memory.
- MemWrite: The processor is writing to memory.



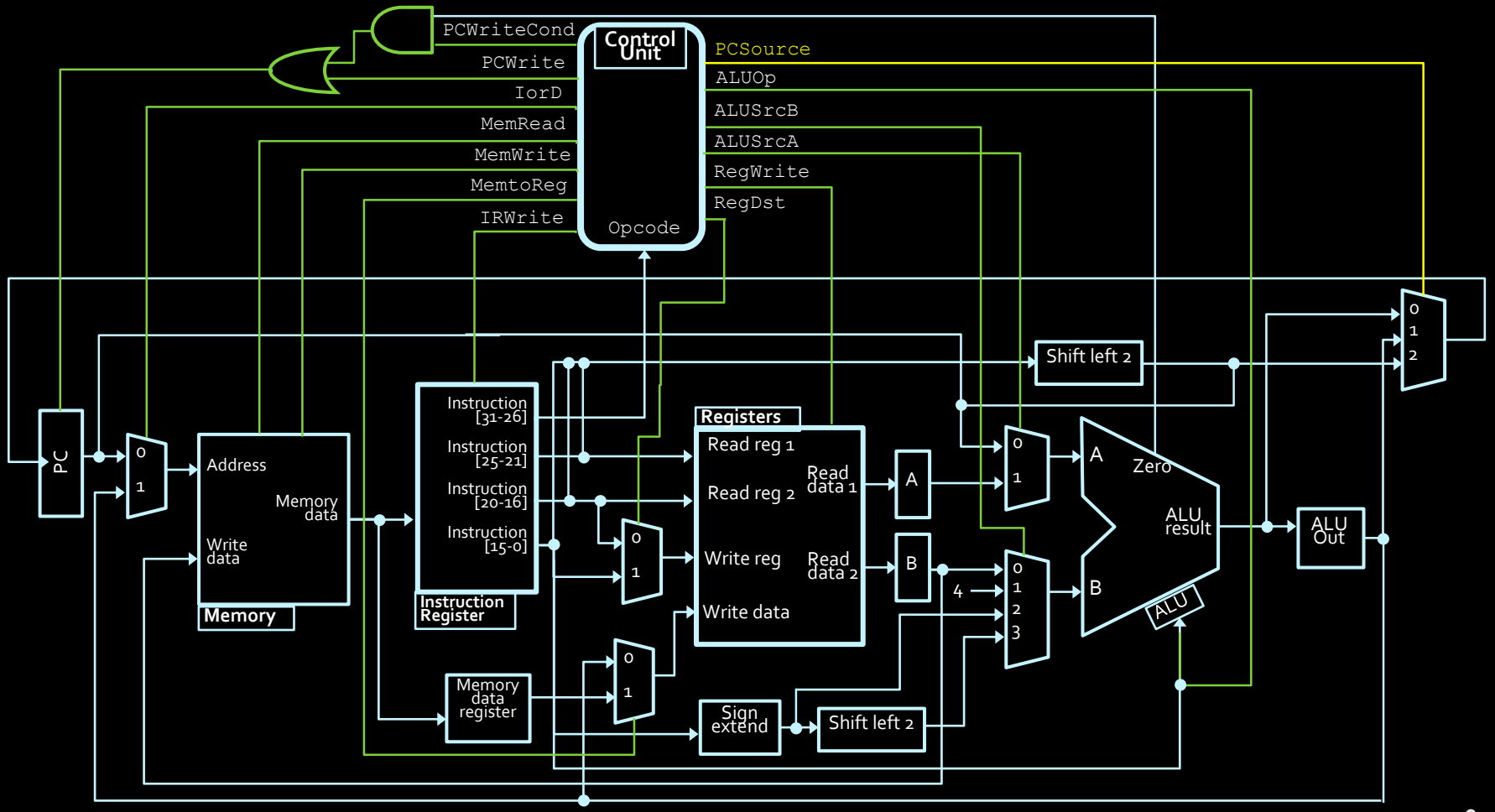
- MemToReg: The register file is receiving data from memory, not from the ALU output.



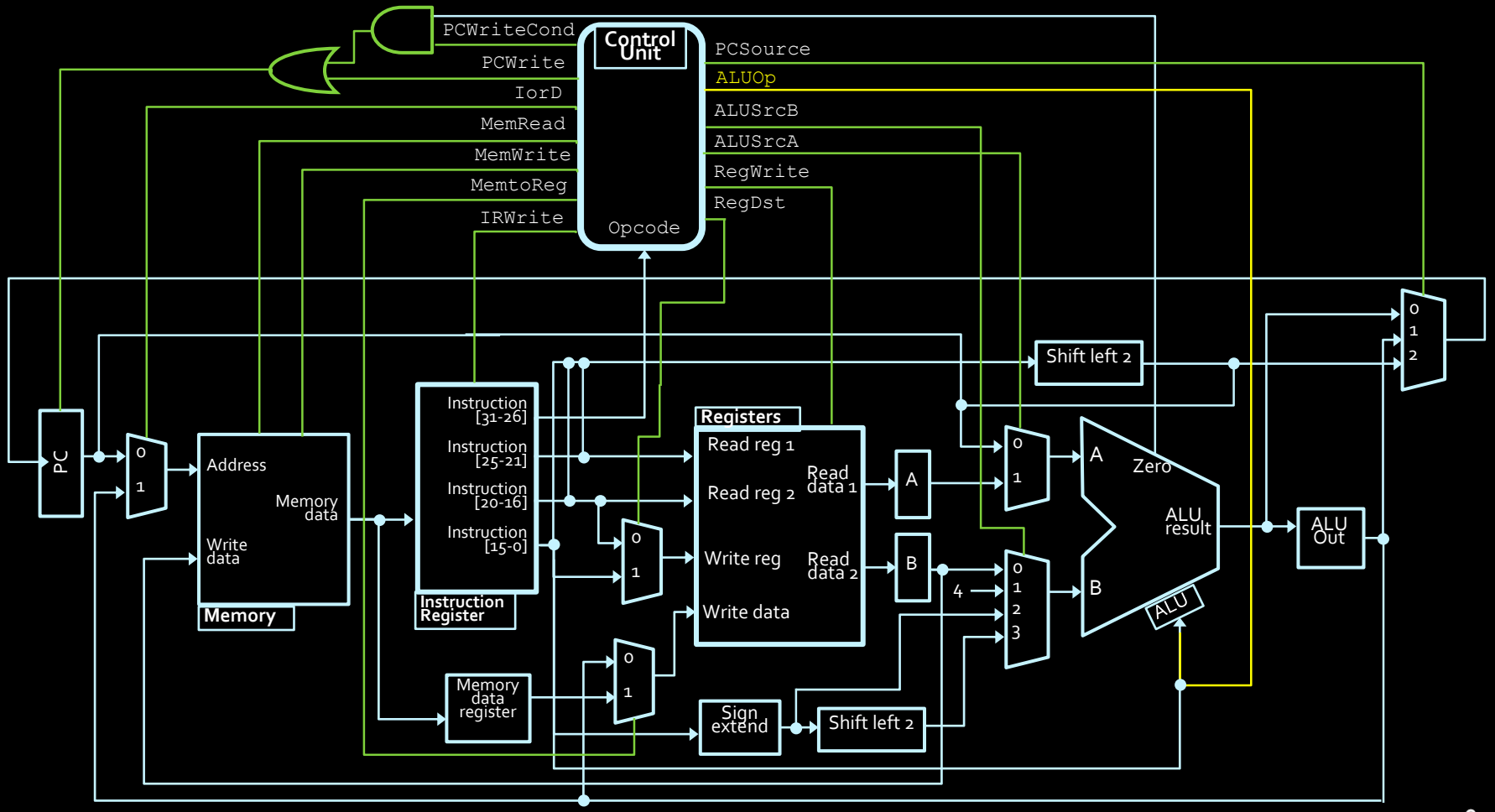
- IRWrite: The instruction register is being filled with a new instruction from memory.



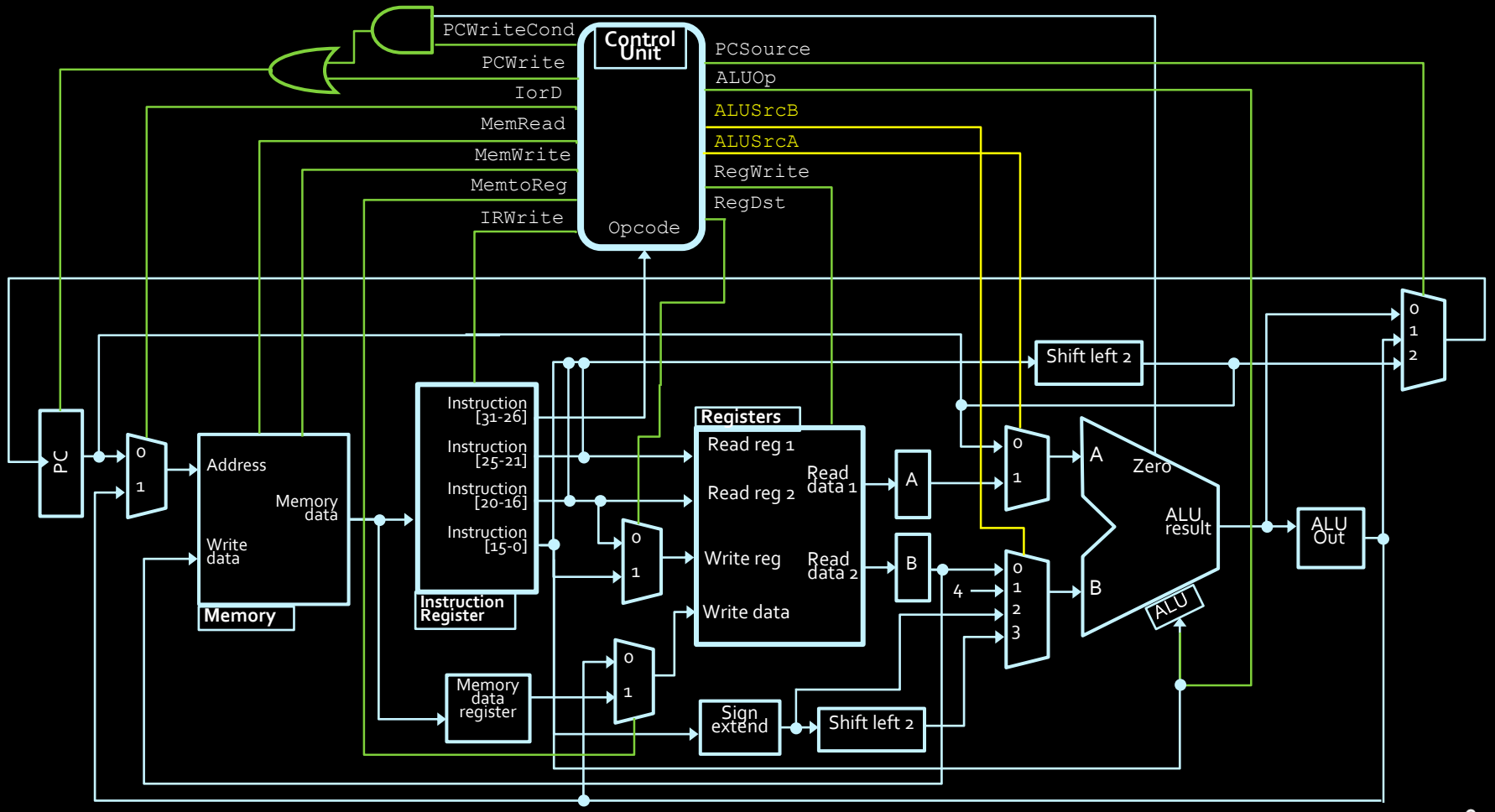
- PCSource: Signals whether the value of the PC resulting from a jump, or from an ALU operation.



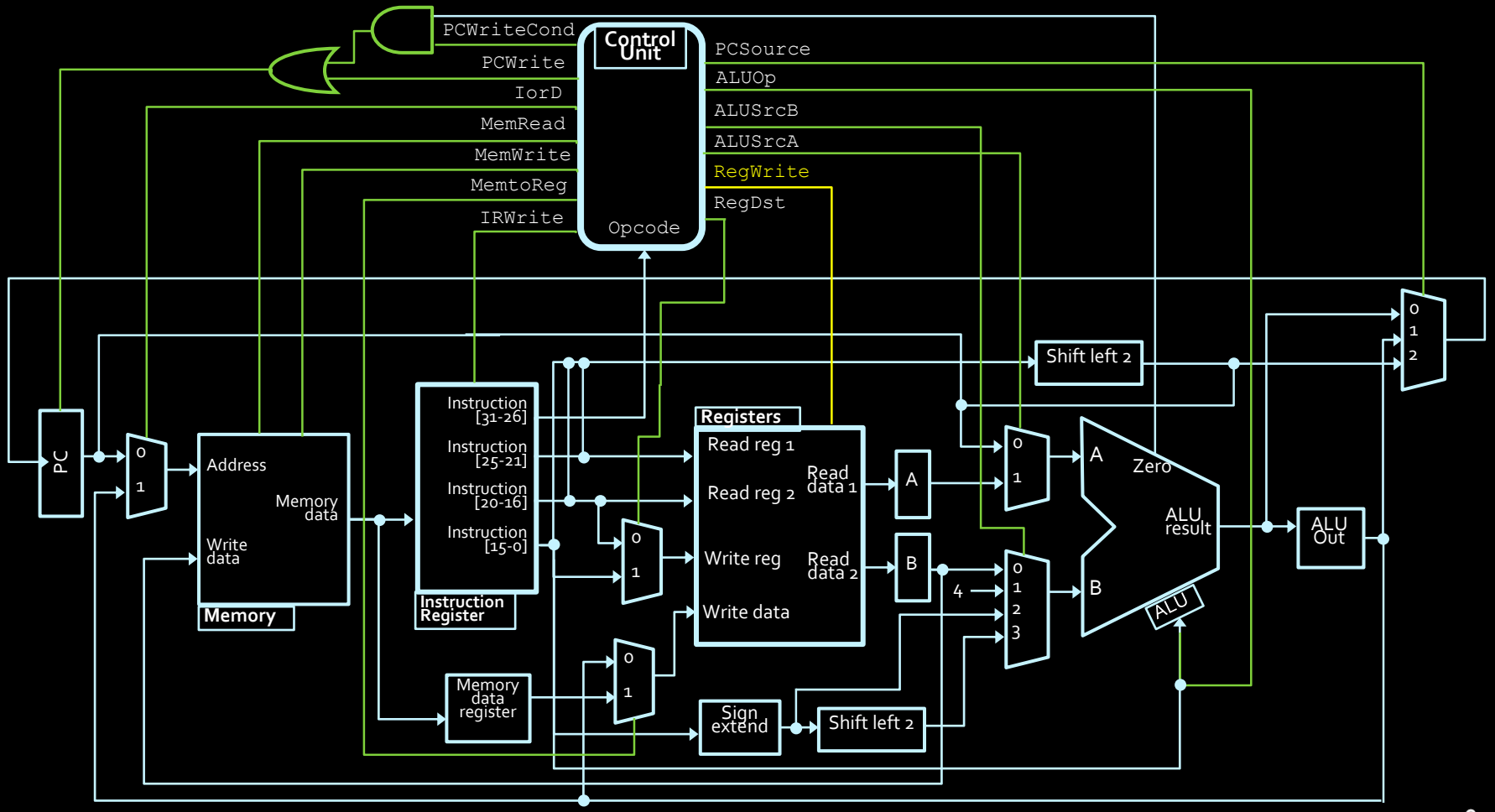
- ALUOp (multi-bit): Signals the execution of an ALU operation.



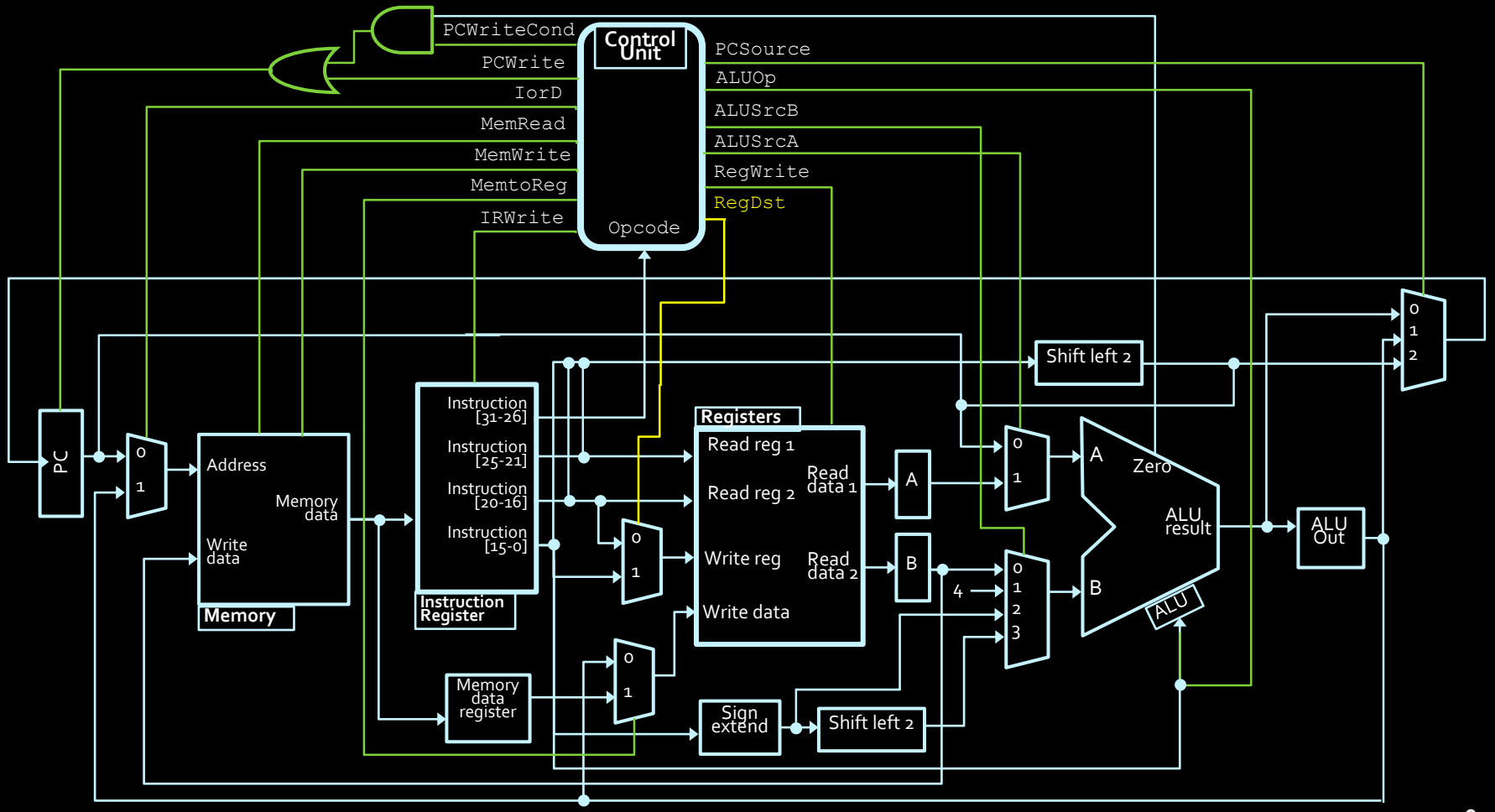
- ALUSrcA: Input A into the ALU is coming from the PC (0) or the register file (1).
- ALUSrcB (2 bits): Input B into the ALU is coming from the register file (0), a constant value of 4 (1), the instruction register (2), or the shifted instruction register (3).



- RegWrite: The processor is writing to the register file.



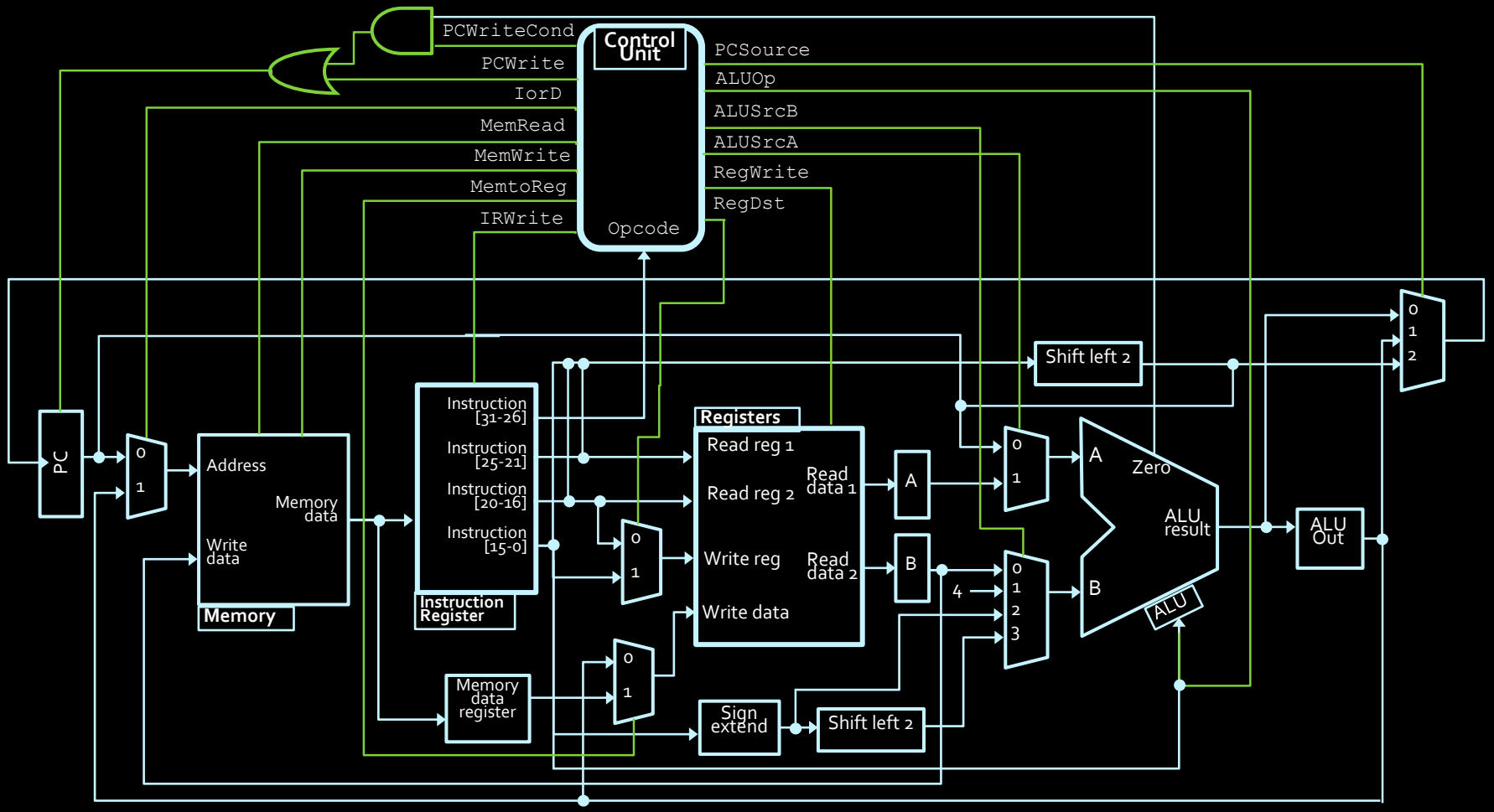
- `RegDst`: Which part of the instruction is providing the destination address for a register write (`rt` versus `rd`).



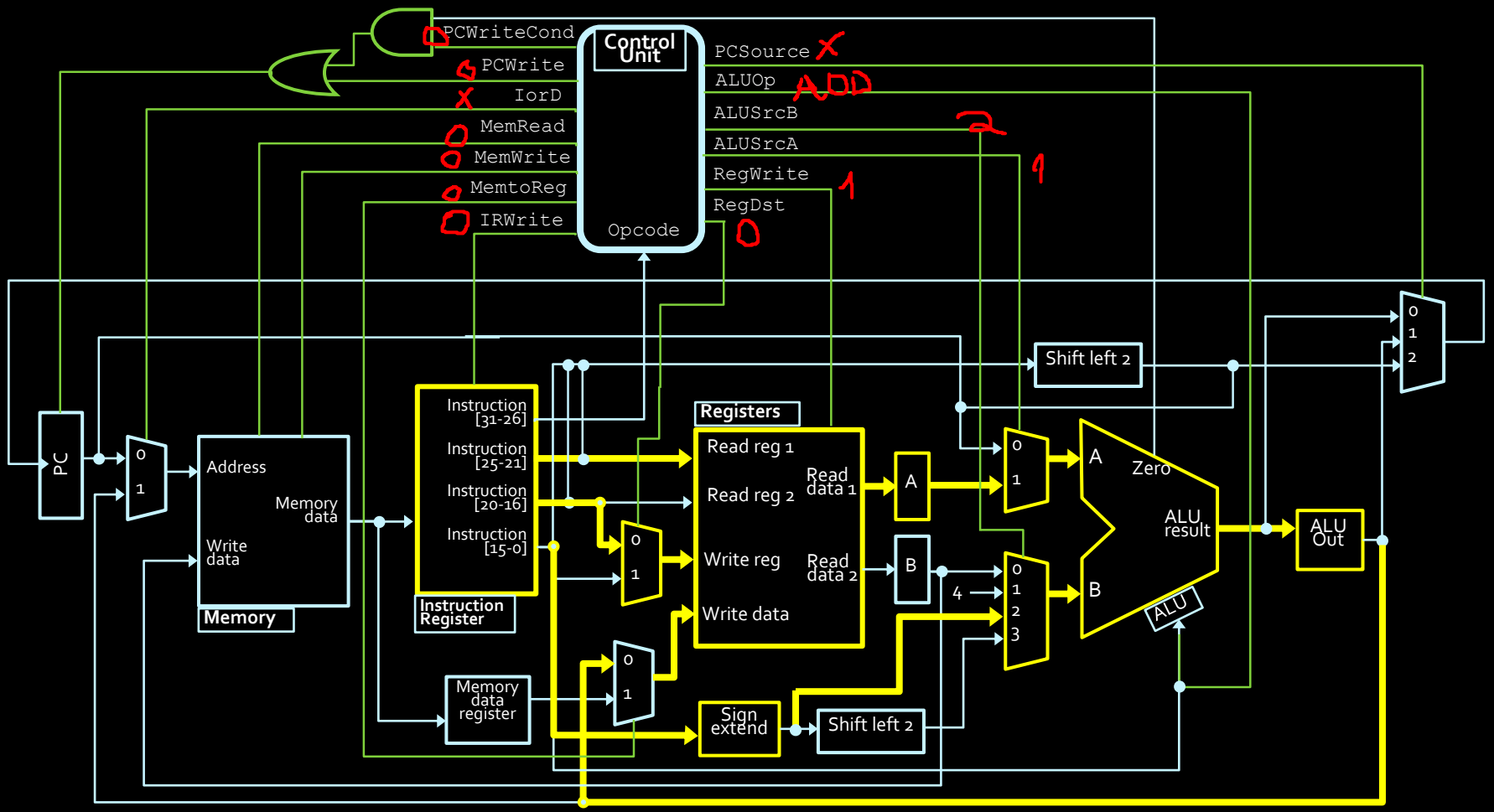
Example instruction

- **addi $15, $8, 42**
  - PCWrite = ?
  - PCWriteCond = ?
  - IorD = ?
  - MemWrite = ?
  - MemRead = ?
  - MemToReg = ?
  - IRWrite = ?
  - PCSource = ?
  - ALUOp = ?
  - ALUSrcA = ?
  - ALUSrcB = ?
  - RegWrite = ?
  - RegDst = ?

addi $15, $8, 42


PCWriteCond
PCWrite
IorD
MemRead
MemWrite
MemtoReg
IRWrite
Control Unit
Opcode
PCSource
ALUOp
ALUSrcB
ALUSrcA
RegWrite
RegDst
PC
Address
Memory data
Write data
Memory
Instruction [31-26]
Instruction [25-21]
Instruction [20-16]
Instruction [15-0]
Instruction Register
Memory data register
Registers
Read reg 1
Read reg 2
Write reg
Write data
Read data 1
Read data 2
A
B
4
Shift left 2
Sign extend
Shift left 2
Zero
ALU result
ALU
ALU Out

addi $15, $8, 42


PCWriteCond
PCWrite
IorD
MemRead
MemWrite
MemtoReg
IRWrite
Control Unit
Opcode
PCSource
ALUOp
ALUSrcB
ALUSrcA
RegWrite
RegDst
PC
Address
Memory data
Write data
Memory
Instruction [31-26]
Instruction [25-21]
Instruction [20-16]
Instruction [15-0]
Instruction Register
Memory data register
Registers
Read reg 1
Read reg 2
Write reg
Write data
Read data 1
Read data 2
A
B
4
Shift left 2
Sign extend
Shift left 2
Zero
ALU result
ALU
ALU Out

Example instruction

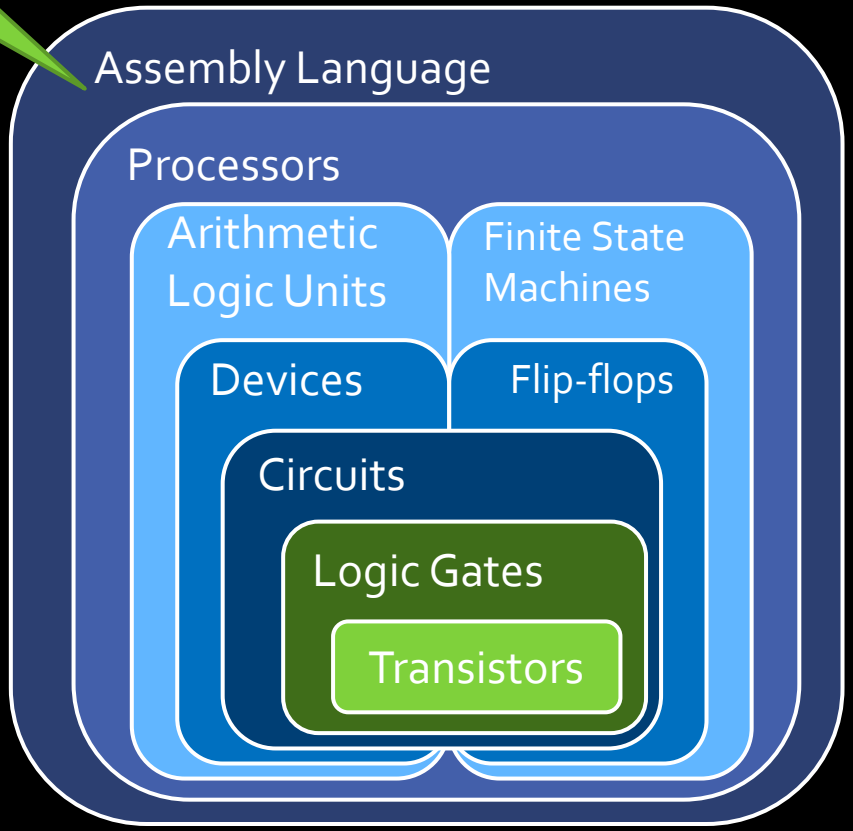
- **addi $15, $8, 42**
  - PCWrite = 0
  - PCWriteCond = 0
  - IorD = X
  - MemWrite = 0
  - MemRead = 0
  - MemToReg = 0
  - IRWrite = 0
  - PCSource = X
  - ALUOp = 001 (add)
  - ALUSrcA = 1
  - ALUSrcB = 10
  - RegWrite = 1
  - RegDst = 0

Example instruction

- `addi $15, $8, 42`
- `addi $t7, $t0, 42`

This is a line of **assembly language**

Started from the bottom now we're here



Tale of a program

- Programmer writes code
- Compile code into machine code instructions
- Save instructions in an executable file
- Run the executable file
- OS will load file into memory and set the PC
- CPU:
  - Loads instructions into instruction register
  - Control unit reads opcode and decodes it
  - Signals setup the datapath
  - Billions of transistors turning on/off…
  - Trillions of electrons start flowing…
- Resulting in…

Hello, World