Processor Datapath

Simple subset:

- addu addiu subu
- or ori
- lw sw
- beq
- j

Storage Elements

For the MIPS, storage elements specified by the ISA:

- memory (instructions and data)
- 32 32-bit registers
- program counter

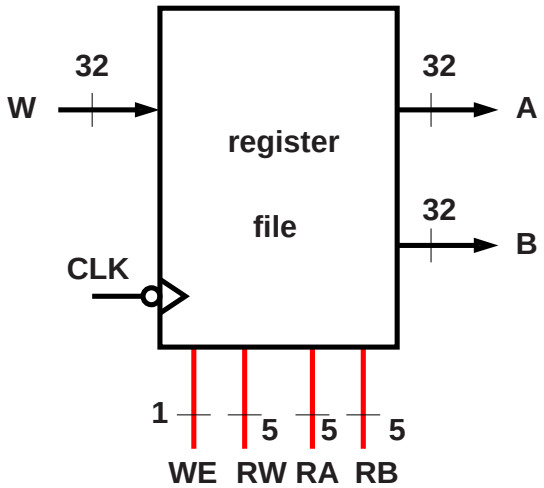
Register 0 is always zero.

We've already seen how to design a single register with a write enable.

Register File

The MIPS register file contains:

- 32 32-bit registers (register 0 special)
- One write bus + write enable
- Two read buses
- Register selection inputs



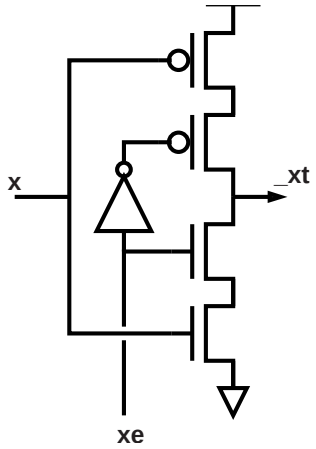
CSL

Register File: Timing

- Reads are combinational
- Writes occur on the negative edge of the clock if WE is high

Implementation:

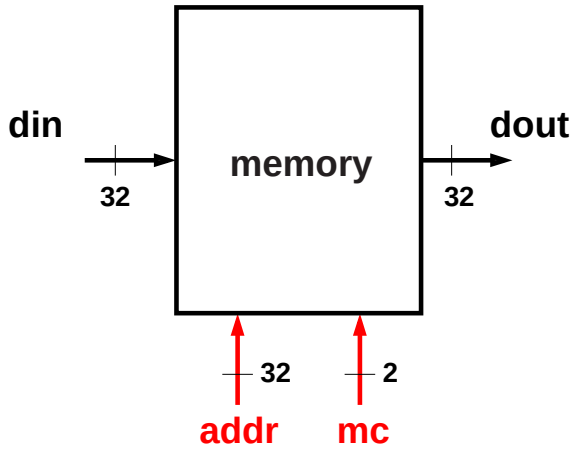
- Standard solution: use mux/demux
- Tri-state outputs



CSL

Memory

- One address bus (32-bit)
- Two data buses (32-bit)
- Two-bit memory control input
- Memory interface doesn't wait for the clock



0 0 => read
0 1 => write byte
1 0 => write halfword
1 1 => write word

Memory

Warning: do not set the control input bits high unless the address and data input is no longer changing.

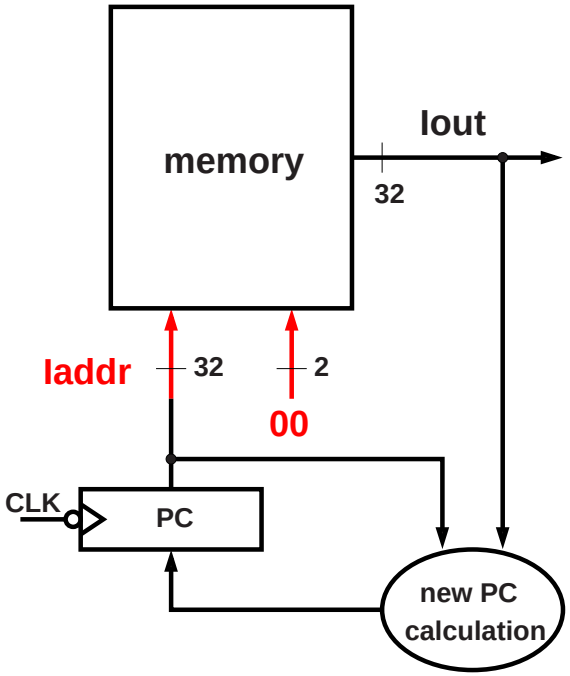
Otherwise, one might modify the wrong address!

Simple solution: only set the `mc` input at the negative edge of the clock.

Translate Specification

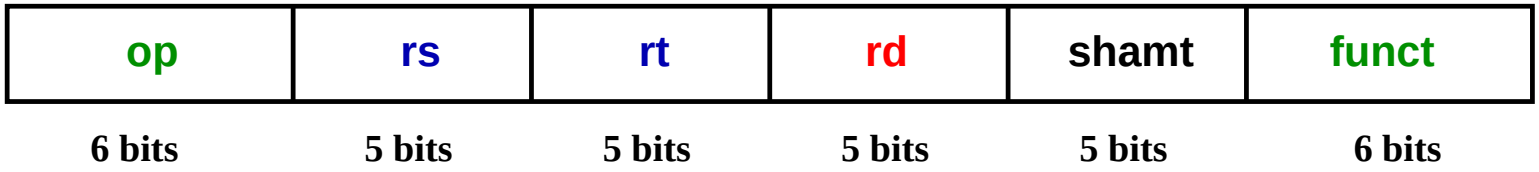
Instruction execution:

- read instruction from memory
- execute instruction
- update PC



CSL

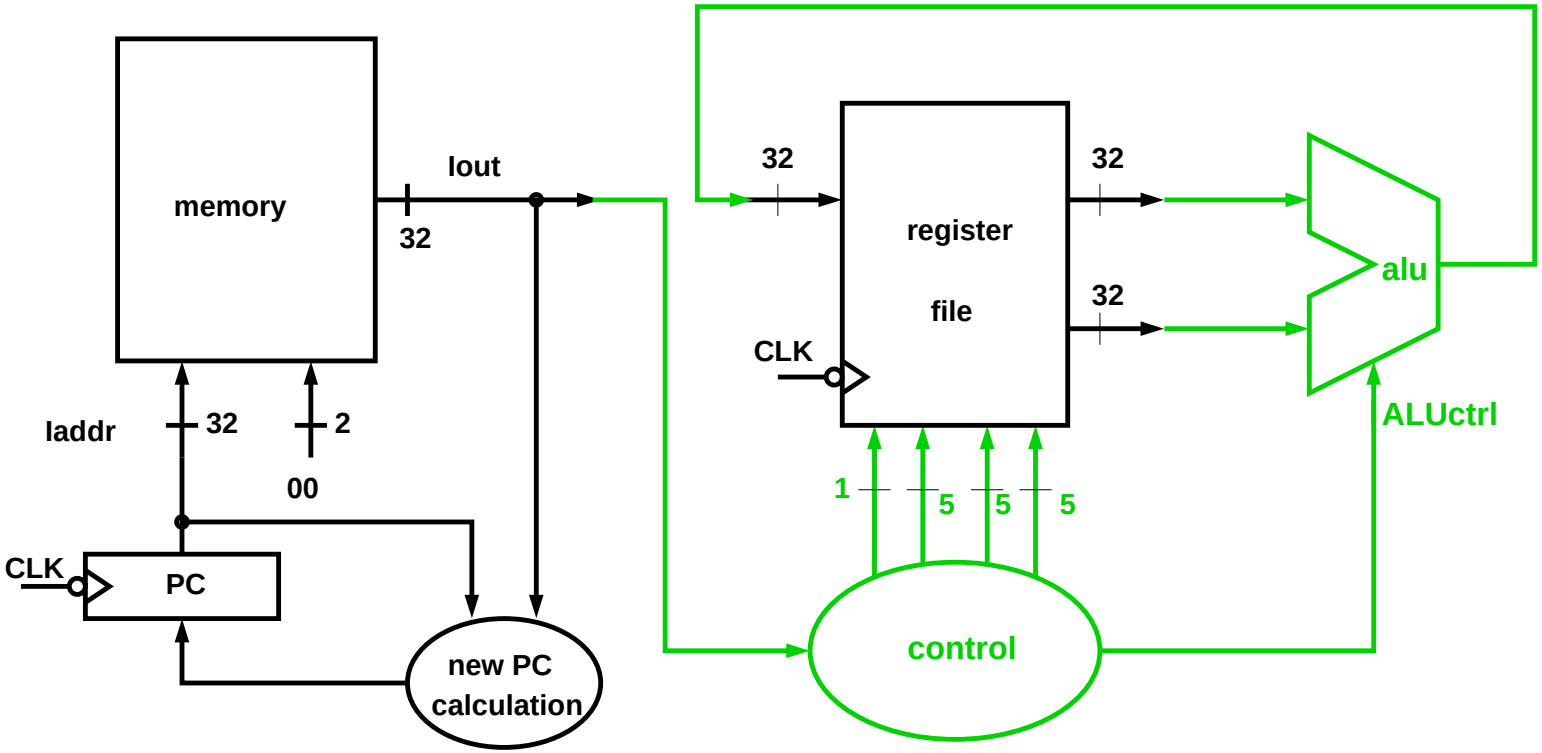
Arithmetic Operations



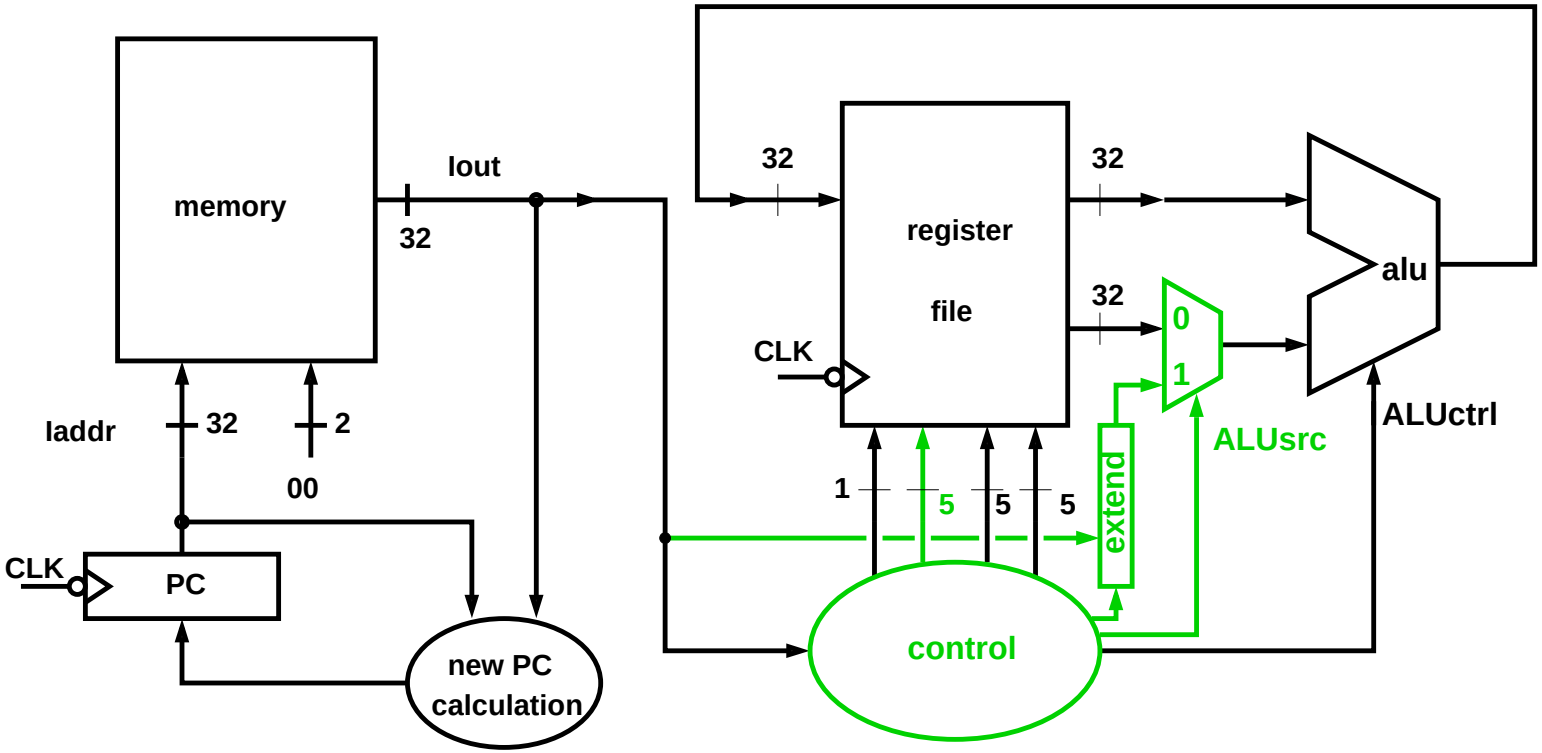
Implemented as:

```
PC <- PC + 4;
if (op == 0 && func == 0x21)
  R[rd] <- R[rs] + R[rt];
else if (op == 0 && func == 0x25)
  R[rd] <- R[rs] | R[rt];
else if (op == 0 && func == 0x23)
  R[rd] <- R[rs] - R[rt];
```

Datapath: Arithmetic Ops

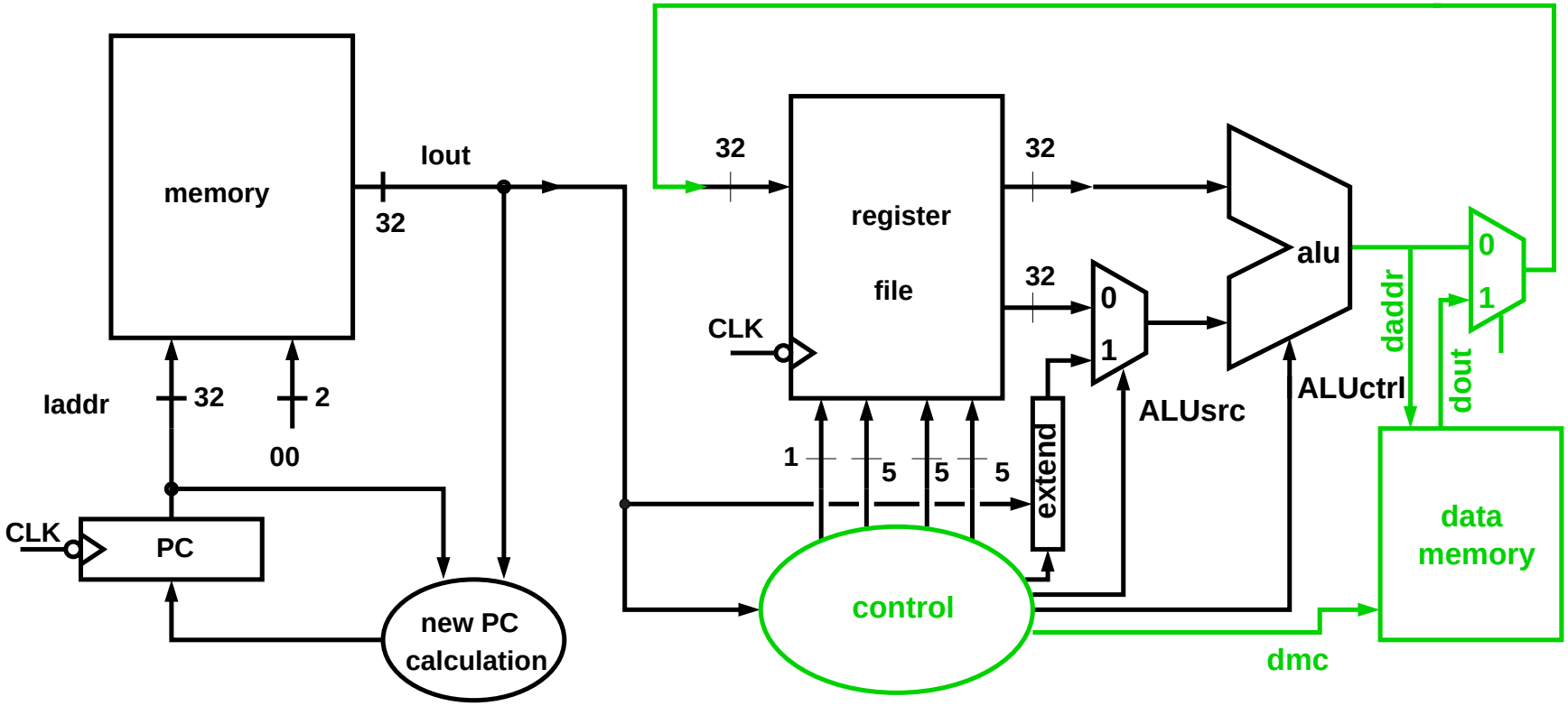


Datapath: Logical Ops

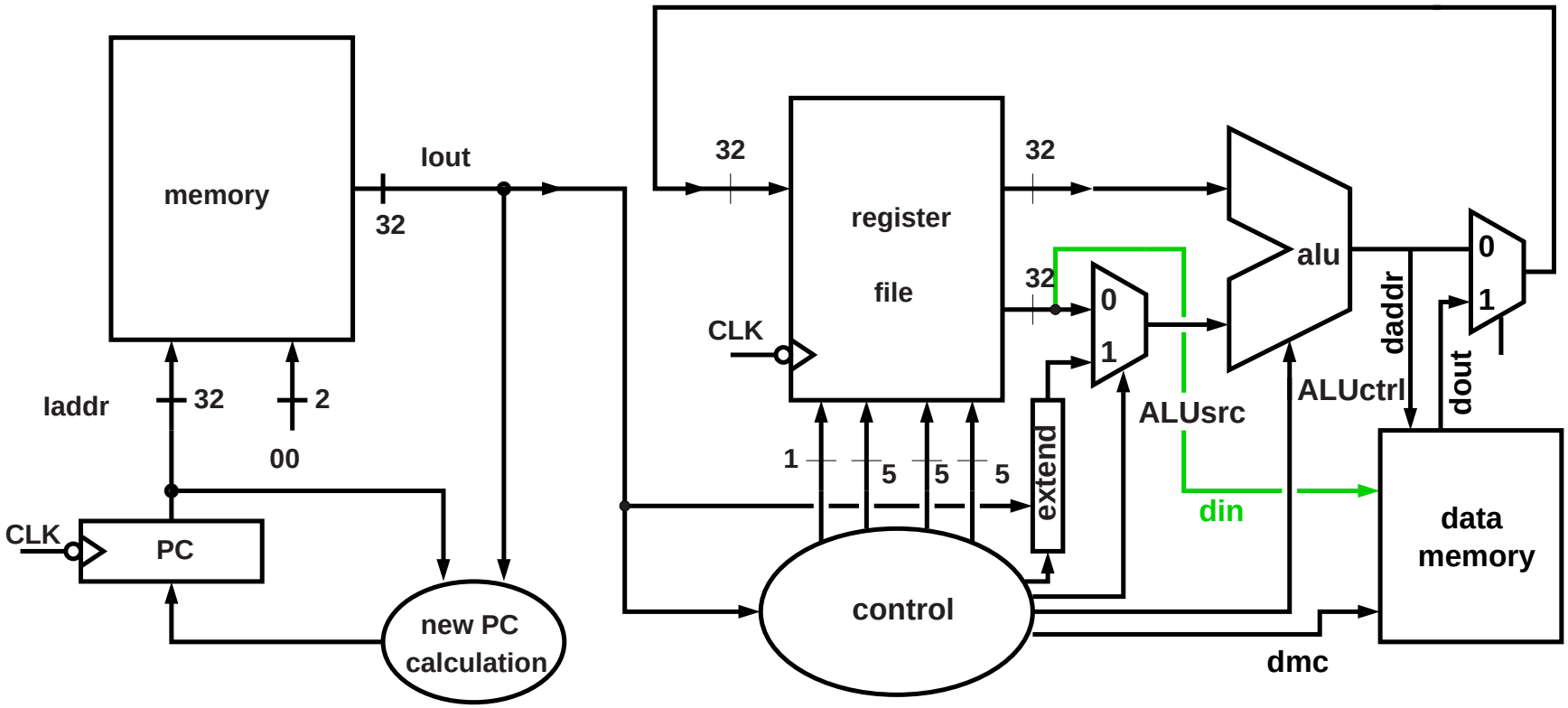


RW control input of register file changed.

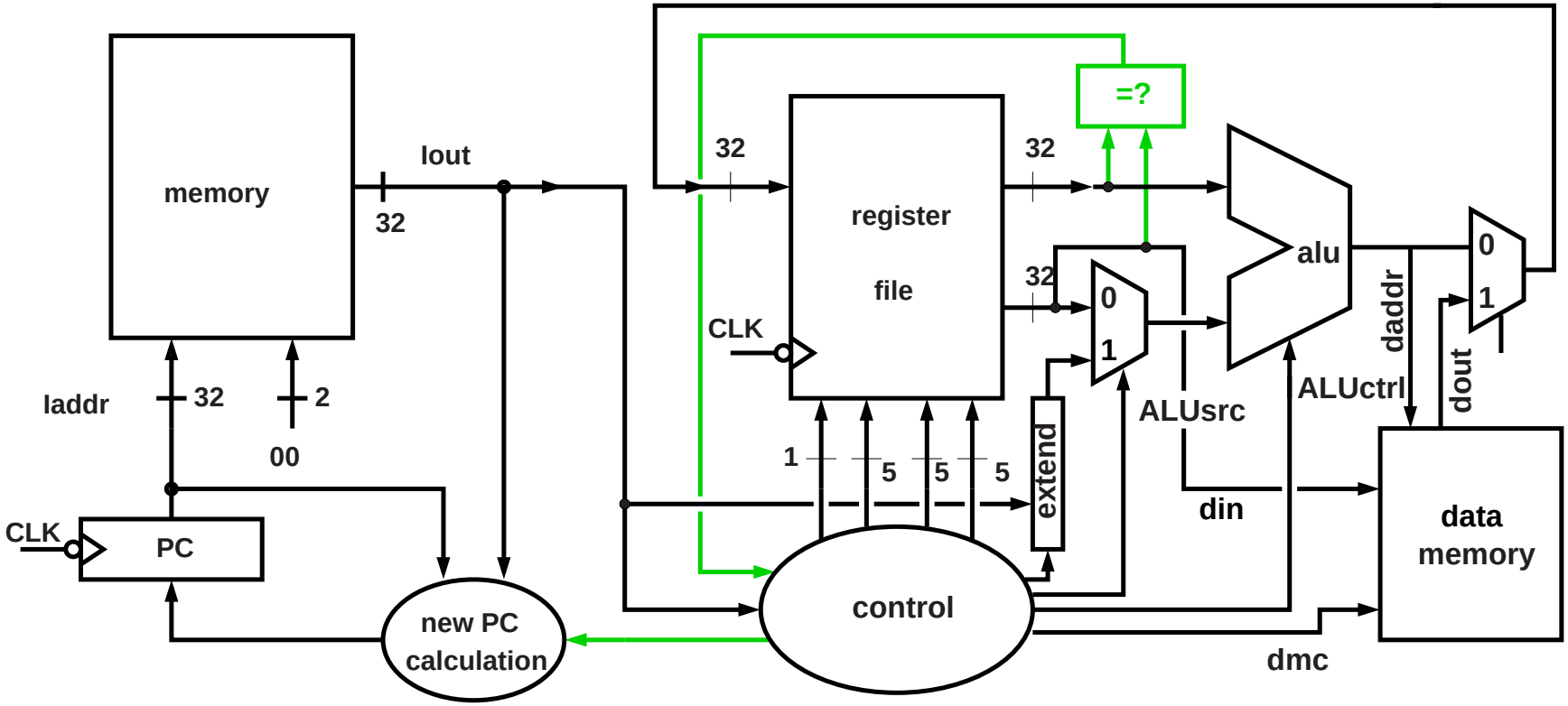
Datapath: Load



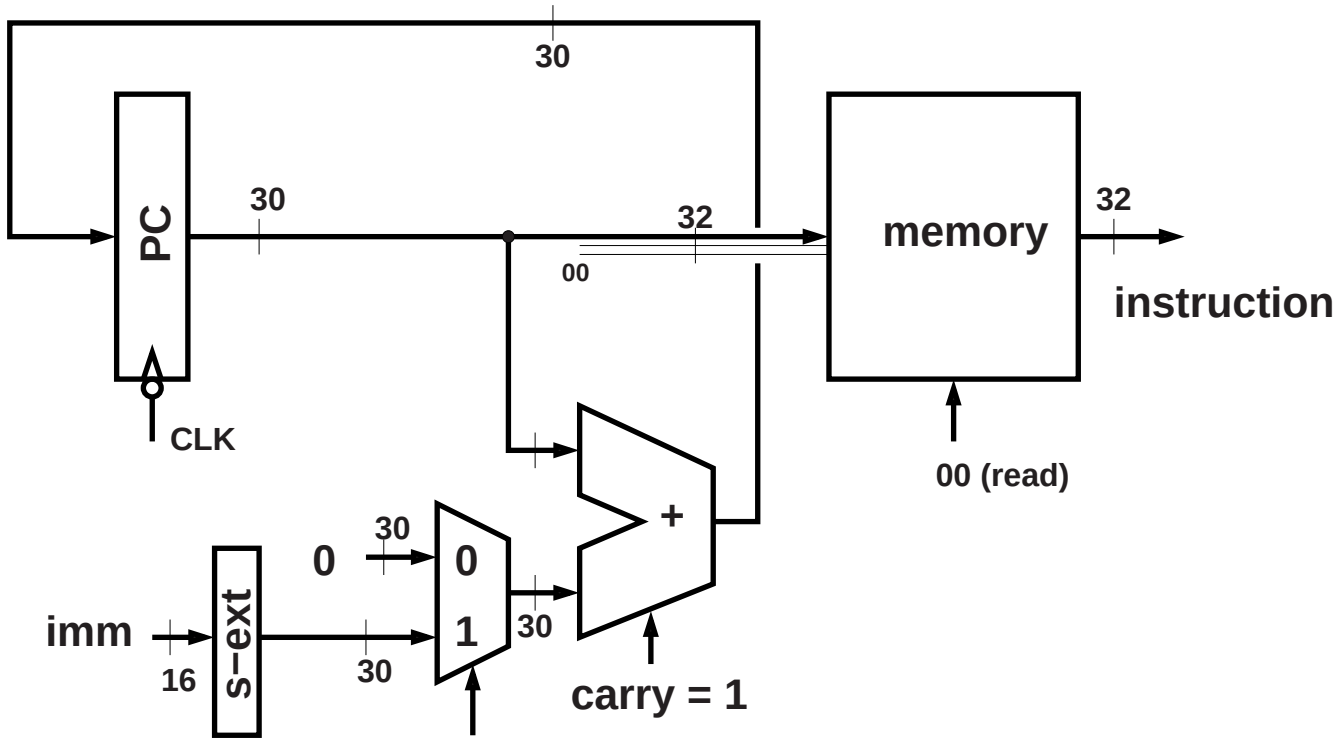
Datapath: Store



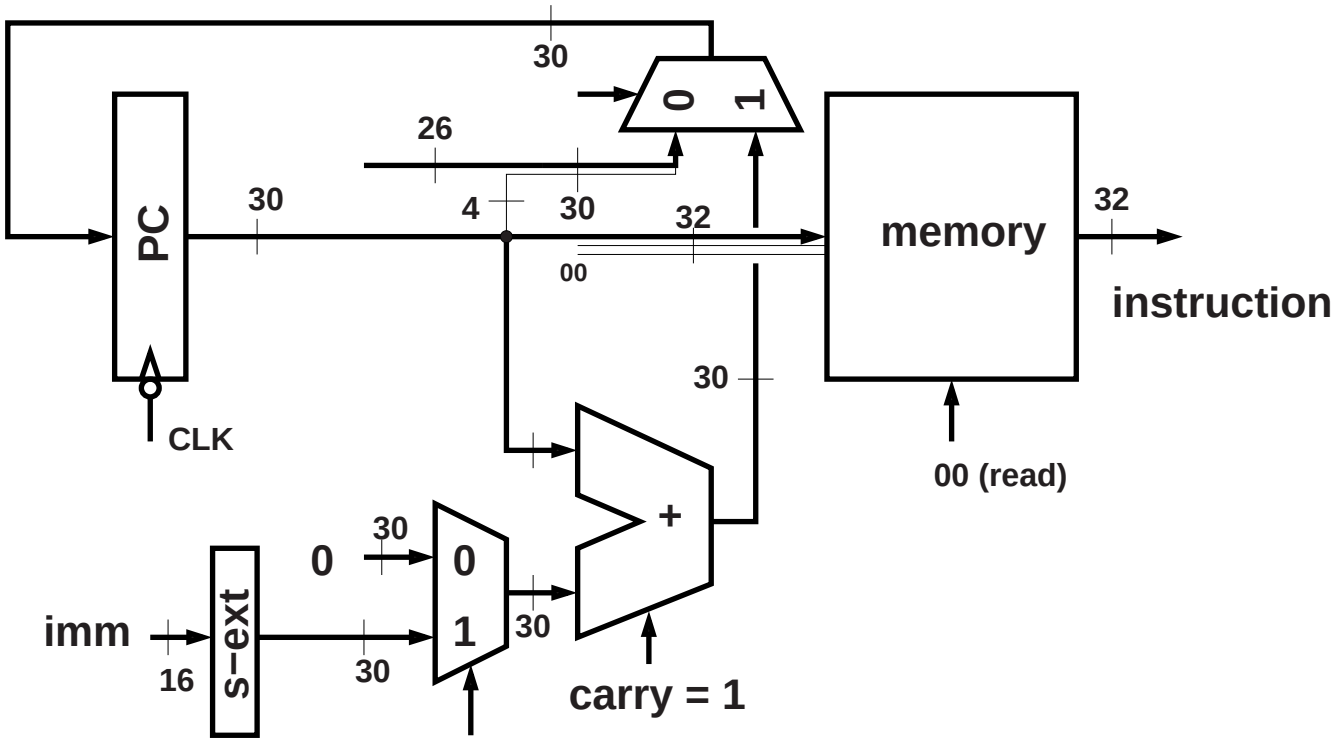
Datapath: Branch



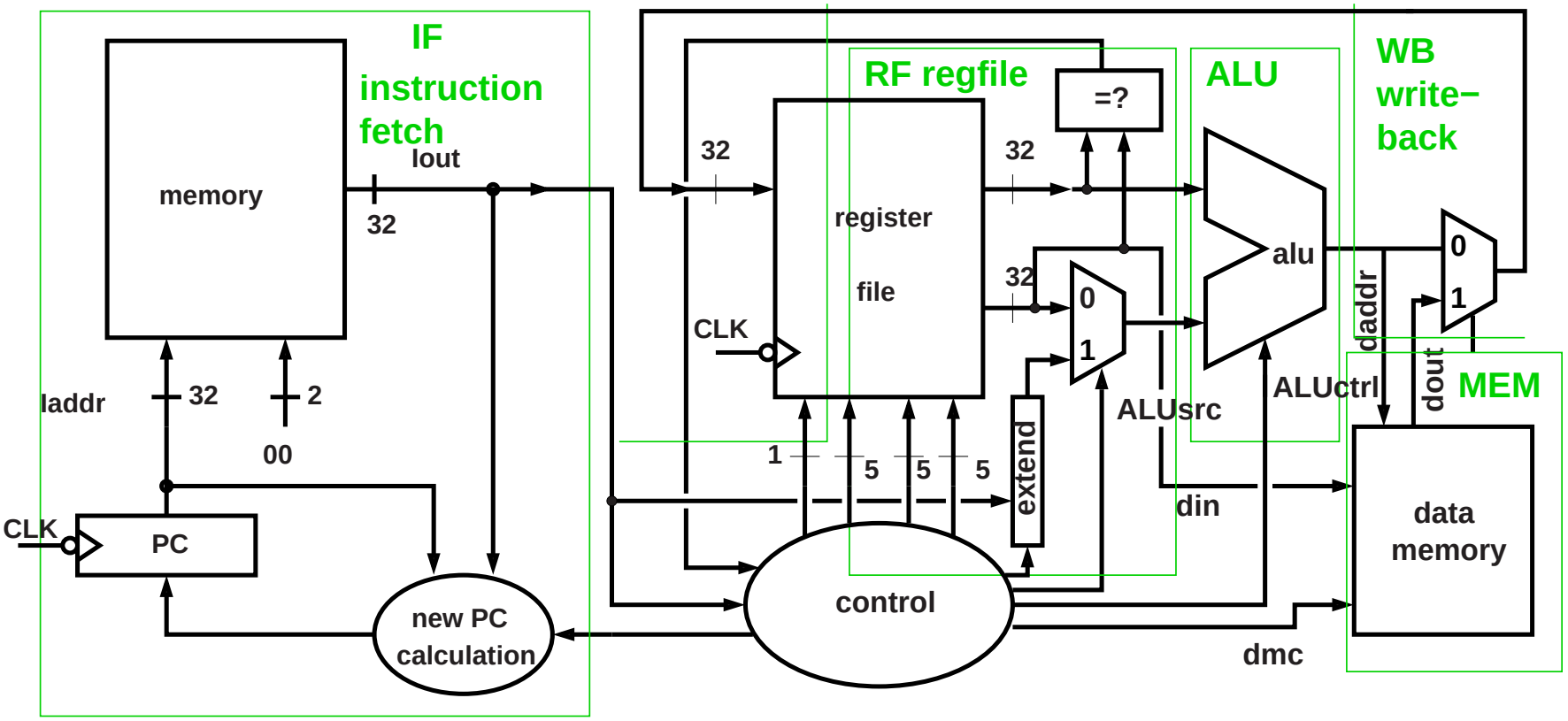
PC Update


30
PC
30
32
00
memory
32
instruction
CLK
00 (read)
+
30
0
0
1
30
imm
16
s-ext
30
carry = 1

PC Update


30
0
1
26
PC
30
4
30
32
memory
32
00
instruction
30
CLK
00 (read)
+
30
0
0
1
imm
16
s-ext
30
30
carry = 1

Datapath


IF
instruction fetch
memory
Iout
32
laddr
32
2
00
CLK
PC
new PC calculation
RF regfile
register file
32
32
CLK
=?
32
0
1
1
5
5
5
extend
control
ALU
alu
ALUsrc
ALUctrl
din
WB write-back
daddr
dout
0
1
MEM
data memory
dmc