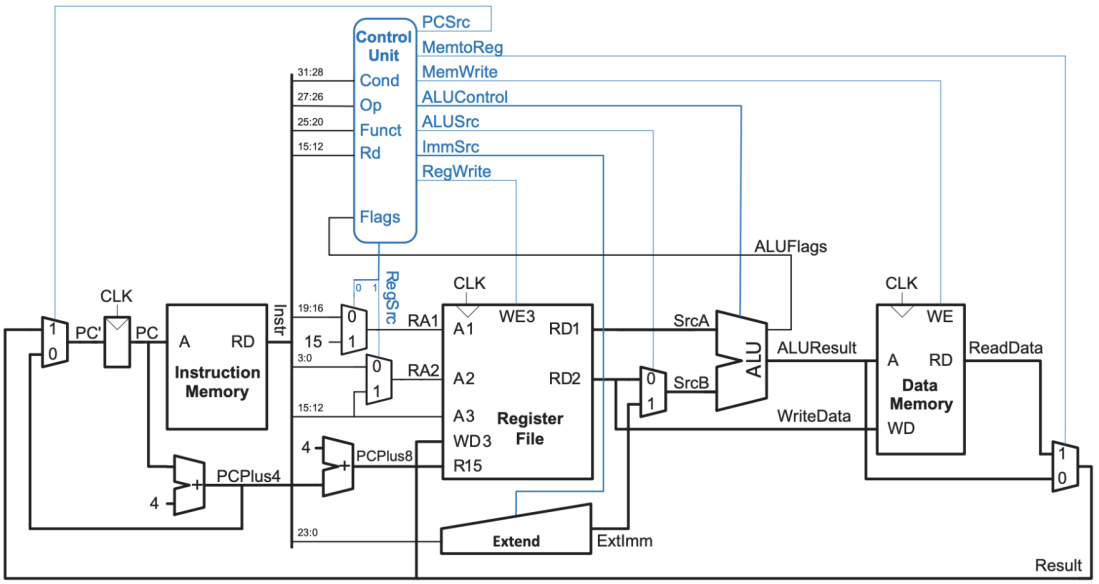
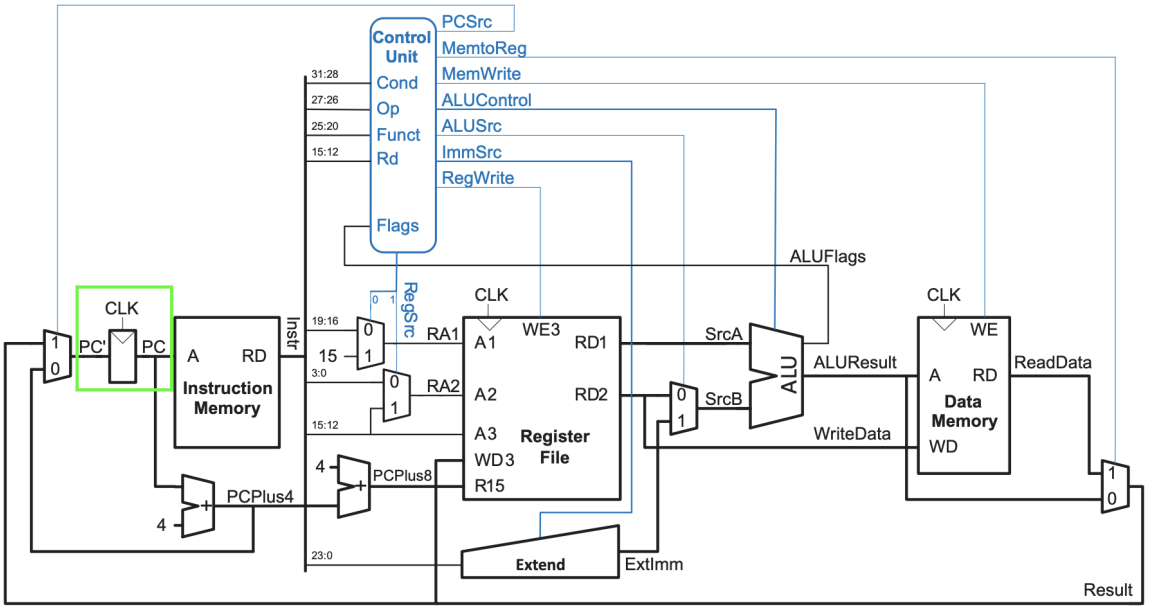
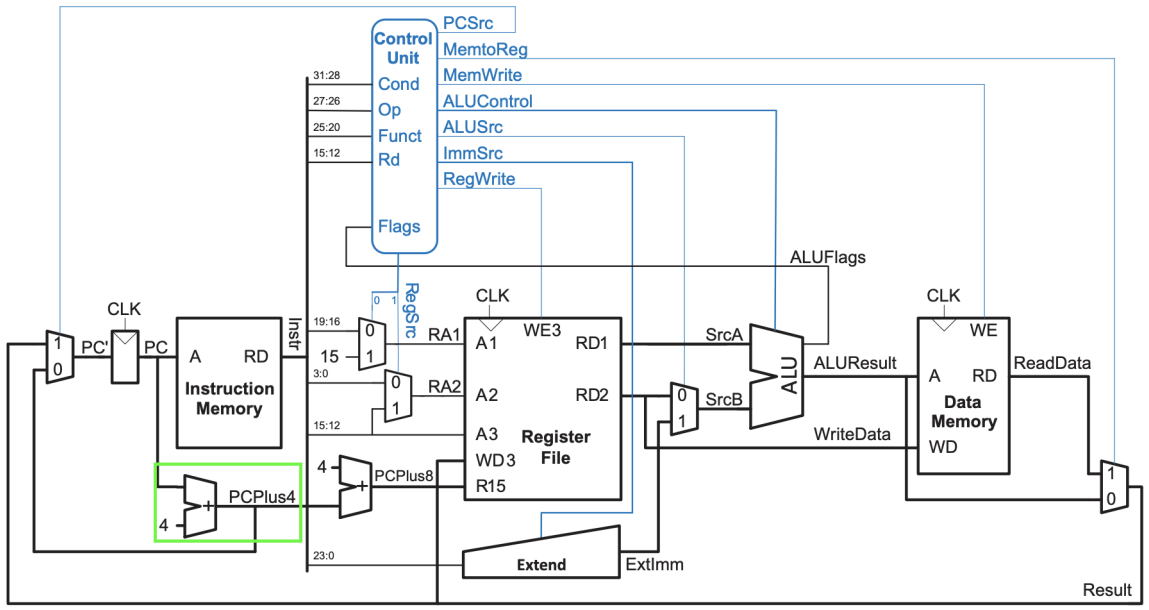



**Figure 7.13** **Complete single-cycle processor**

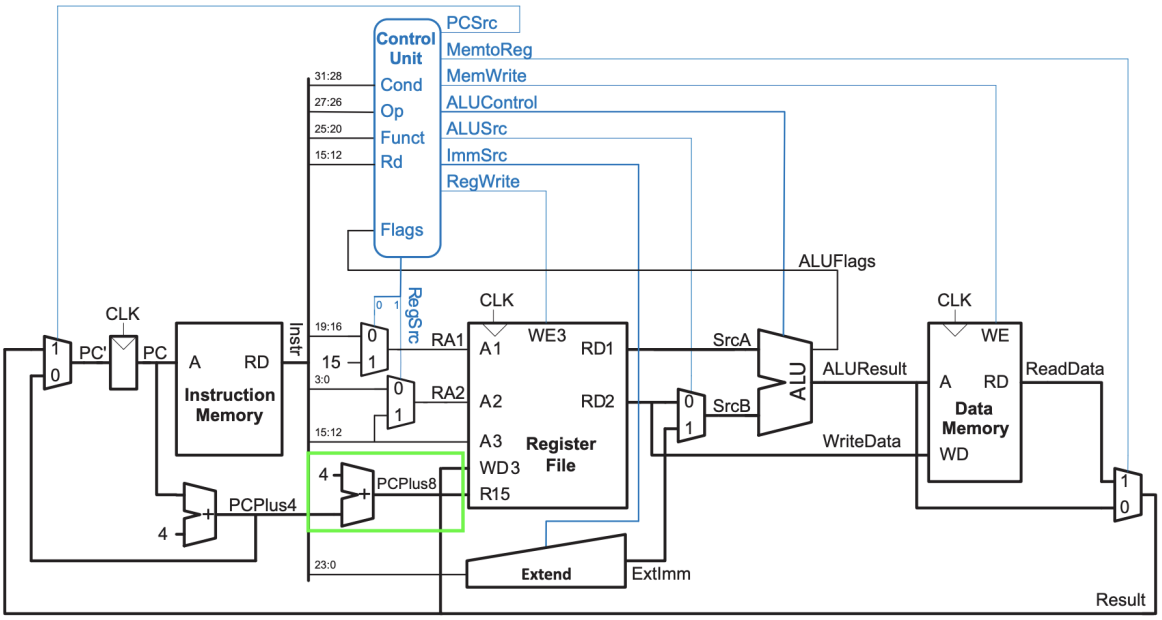



Program Counter

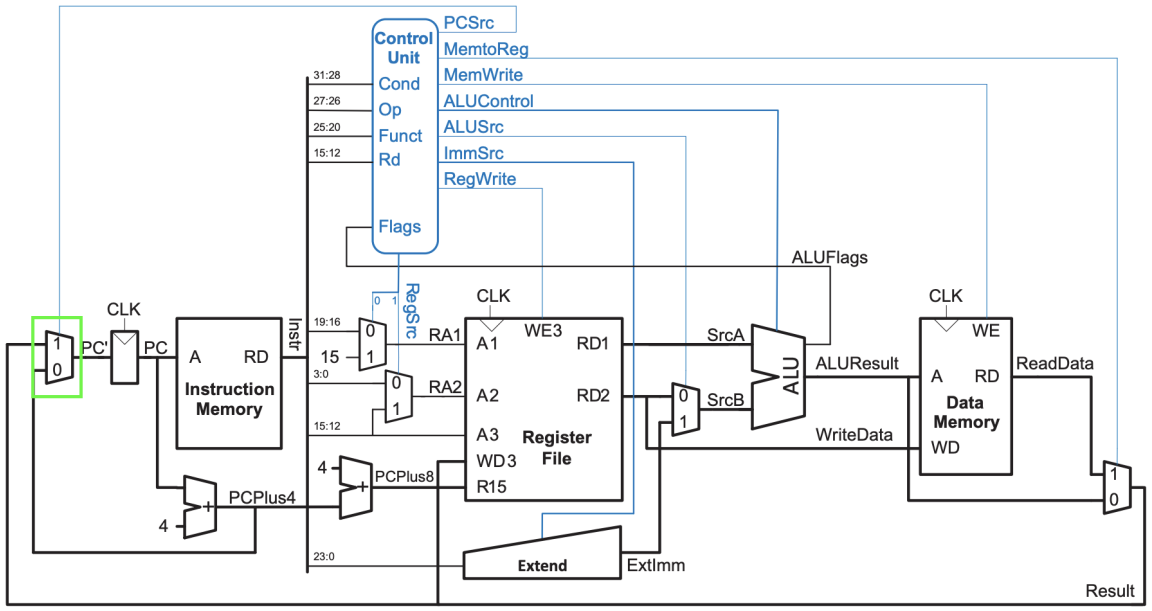
Outputs the current instruction, PC and reads in the next instruction, PC'. On the rising? edge of CLK, PC' becomes PC and a new PC' value is read in.



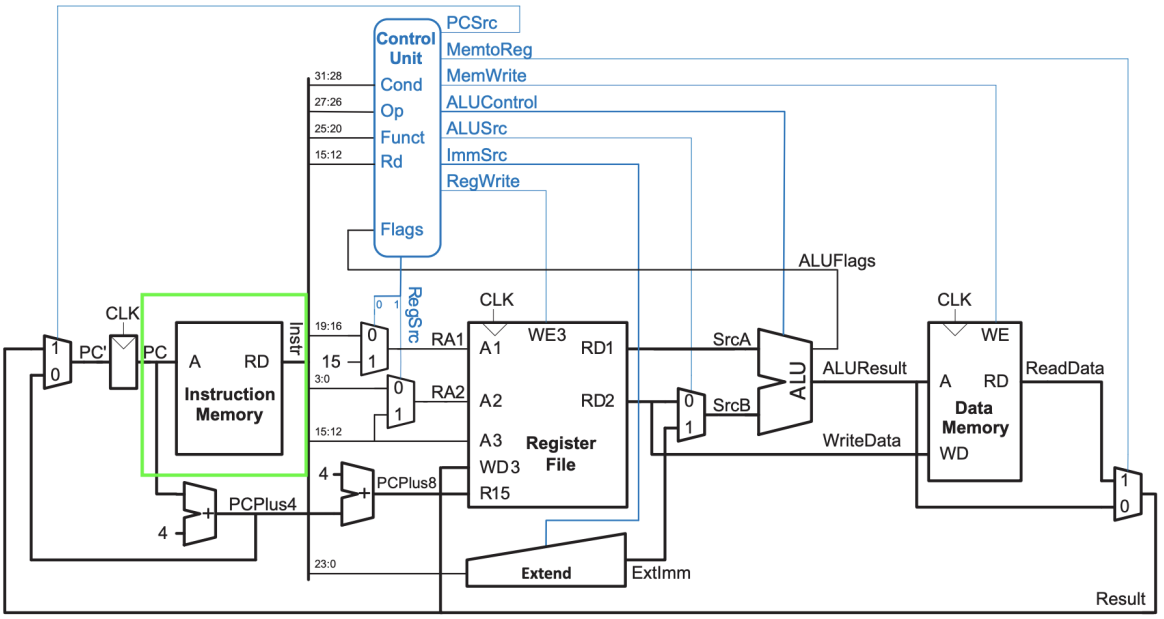

Adder increments PC by 1 word (4 bits) for each clock cycle.
The PCSrc control signal is set to 0 to choose PCPlus4 or 1 to choose ReadData.



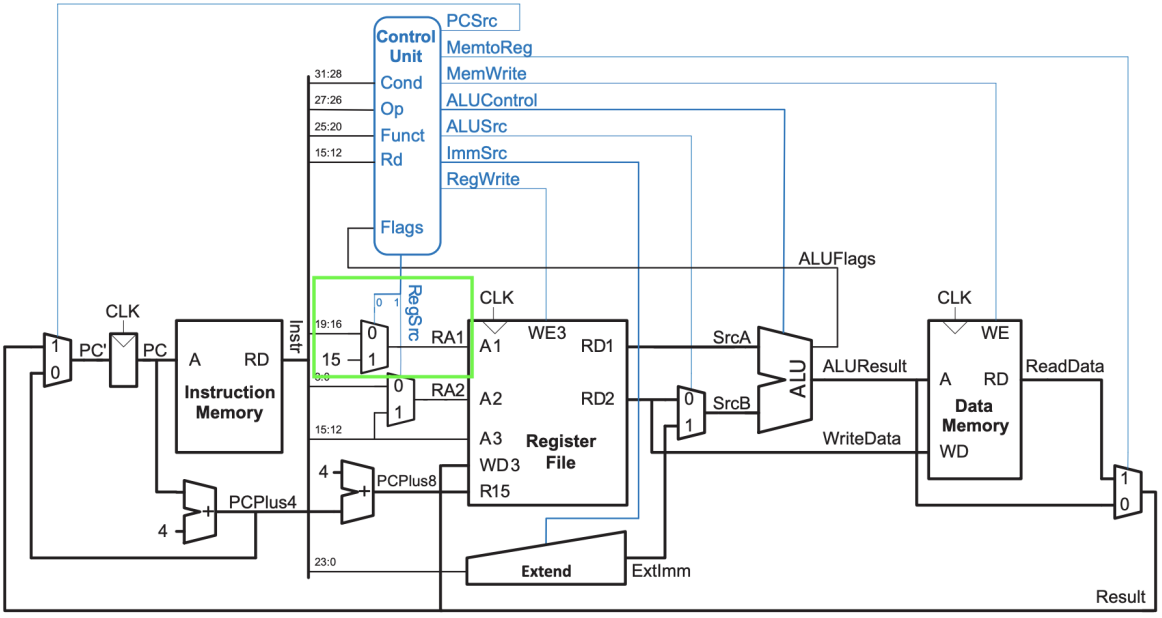

Second Adder for R15. This further increments the PC and passes that value to R15.



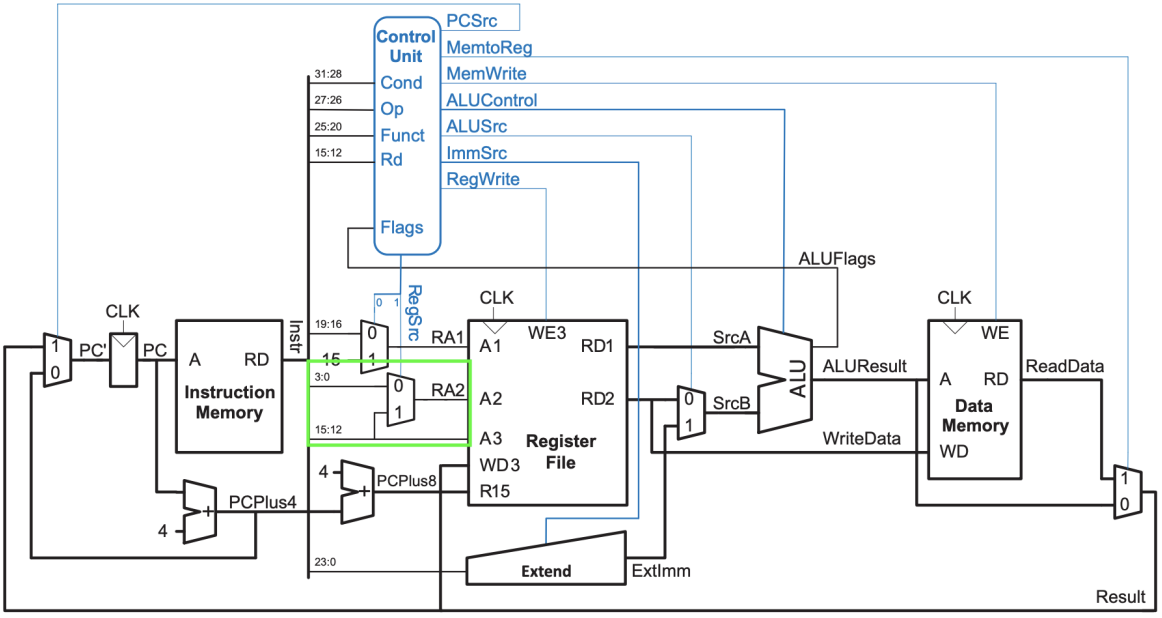

Because changing R15 will change the value of PC, a multiplexer is needed to determine whether R15 is being changed directly or if PC is being changed directly.
If PCSrc is set to 1 the MUX outputs the result from ReadData into the program counter, and if PCSrc is set to 0 it outputs PCPlus4.



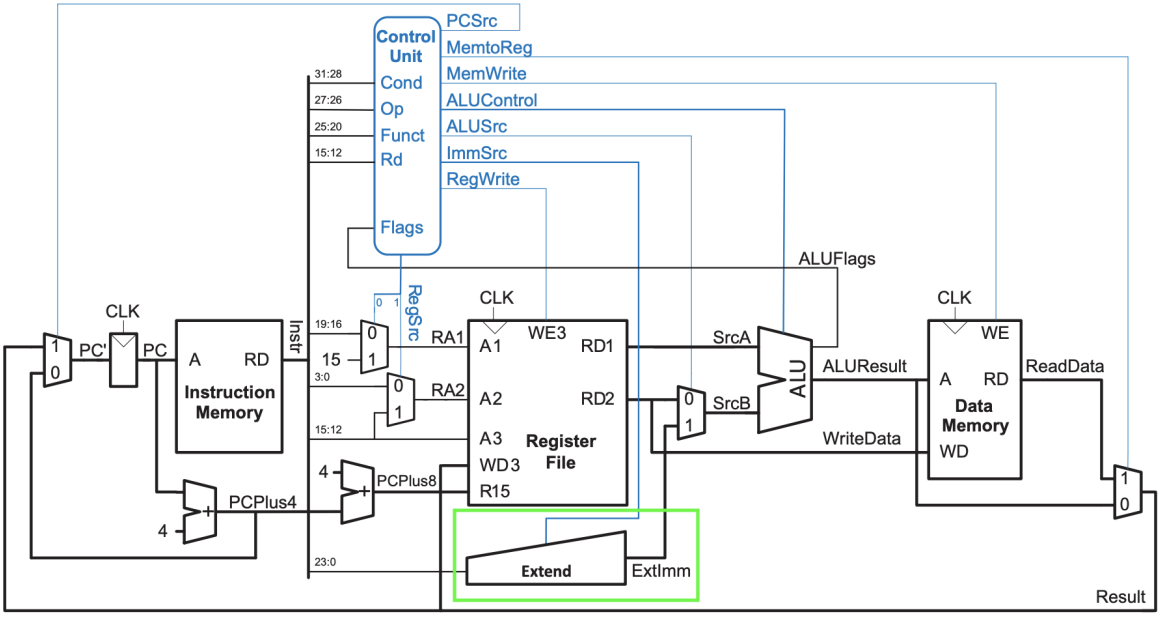

Instruction memory register reads in memory address from PC and outputs the 32 bit instructions from that memory address.




This multiplexer allows R15 to be chosen as input to A1

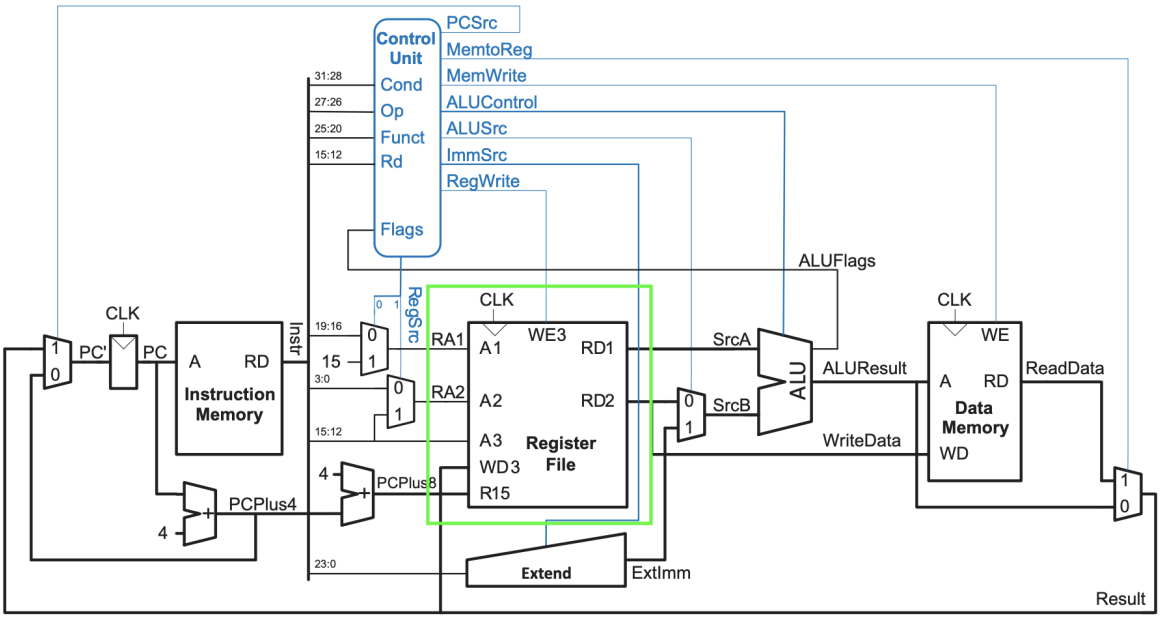



This multiplexer allows the source register to be selected when dealing with register values instead of an immediate.

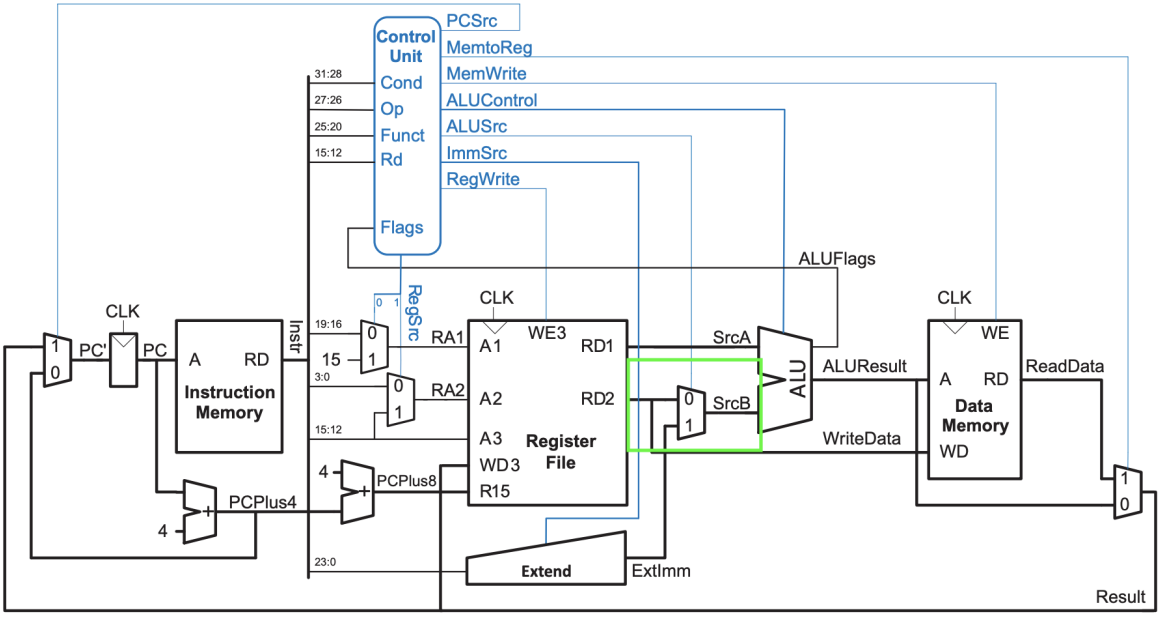



This zero extends the immediate that the LDR bit is offset by. The offset is stored in the immediate field of instruction ($Instr_{11:0}$); it must be offset because it is an unsigned immediate.

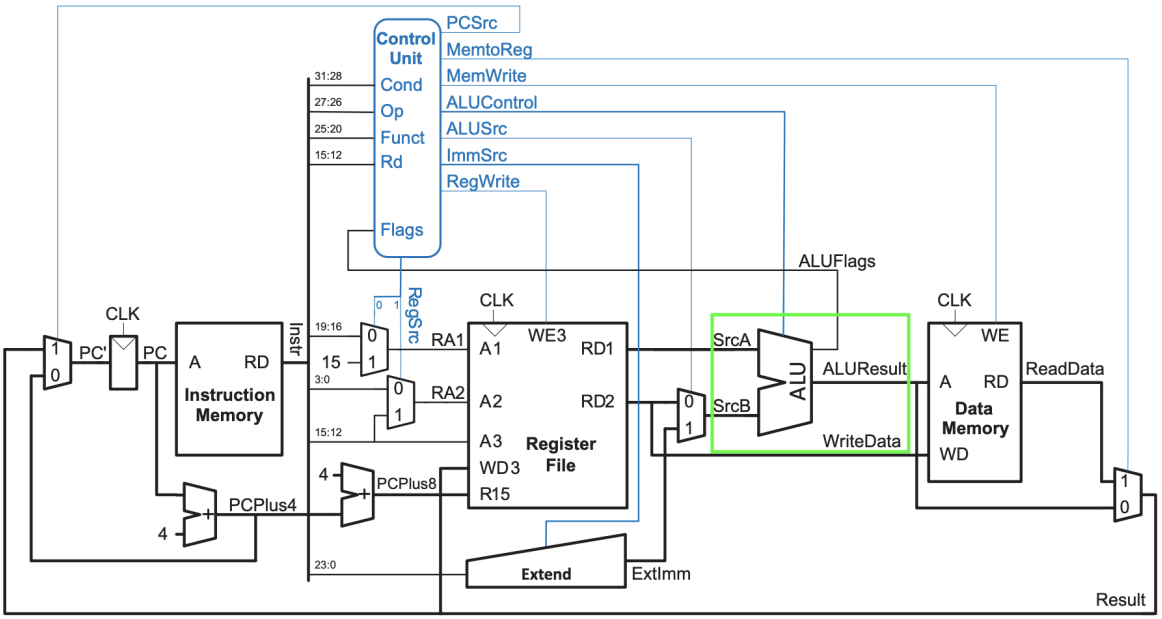
This extend block basically adds zeros to the beginning of the data to make it a 32 bit value.



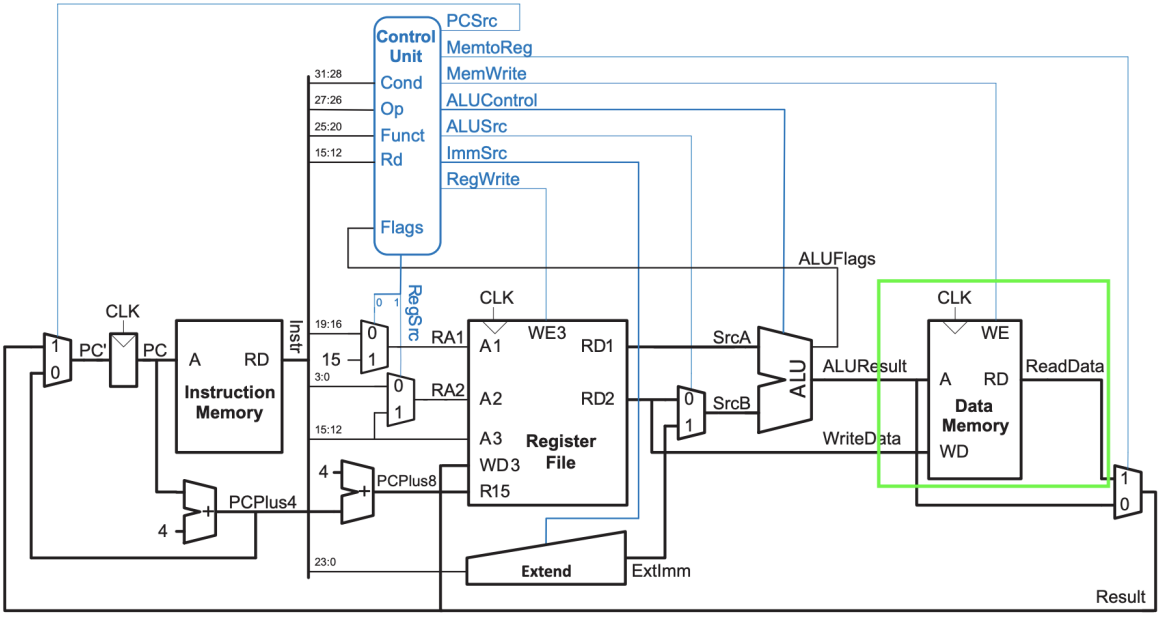

The register file holds all of the registers that store variables. It consists of read ports and write ports.



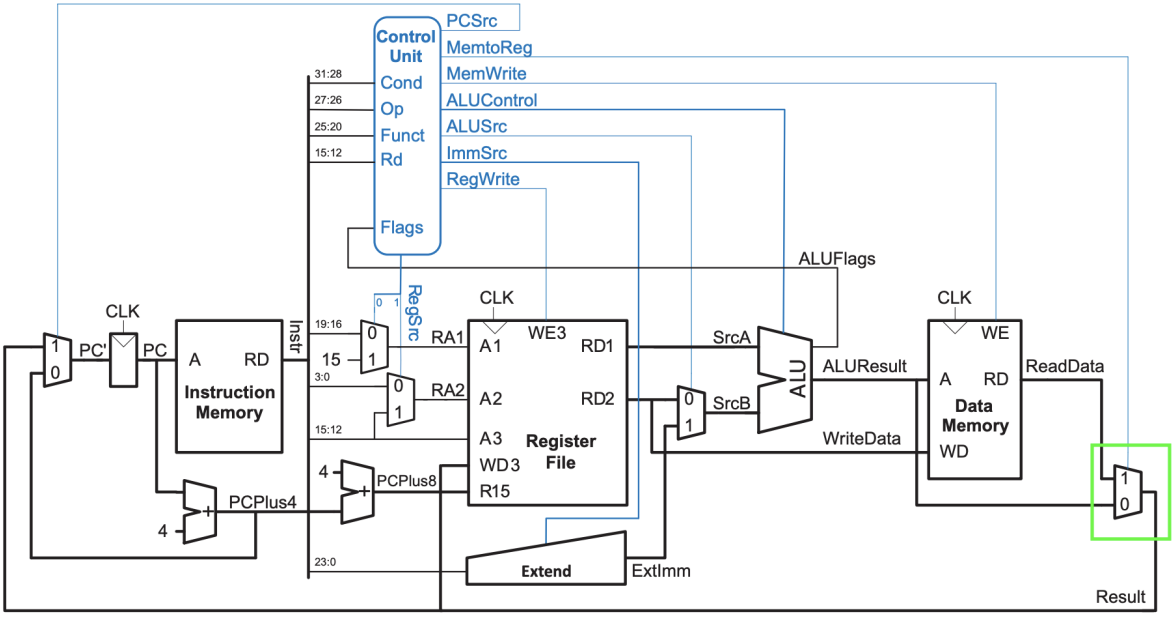

Because data processing instructions that use register addressing receive their second source from Rm, a multiplexer is added to the second input of the ALU. This allows the second source register to be selected.




Arithmetic Logic Unit. This performs instructions on data from SrcA and SrcB. The ALU control bits specify which operation to perform.




The data memory reads in the result of the ALU and the write data and writes it into the destination register at the end of the clock cycle.




This multiplexer chooses between the read data and the ALU result. MemtoReg is 0 for data processing instructions, which chooses the ALU result. It is 1 to choose LDR from ReadData.

When the ImmScr is 0, ExtImm is extended from $\text{Instr}_{7:0}$, and when ImmScr is 1, ExtImm is zero extended from $\text{instr}_{11:0}$

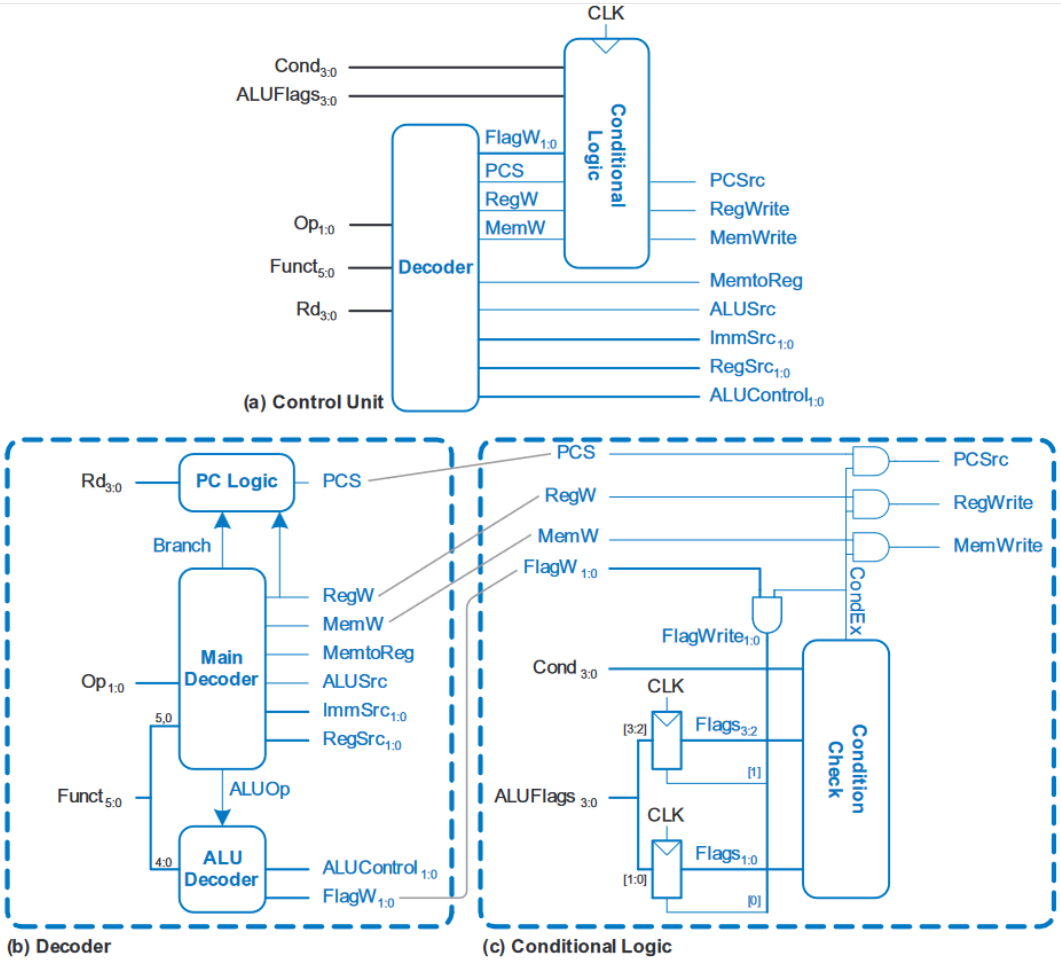



**Figure 7.14** Single-cycle control unit

This is a detailed version of the control unit that is used in this single cycle processor.