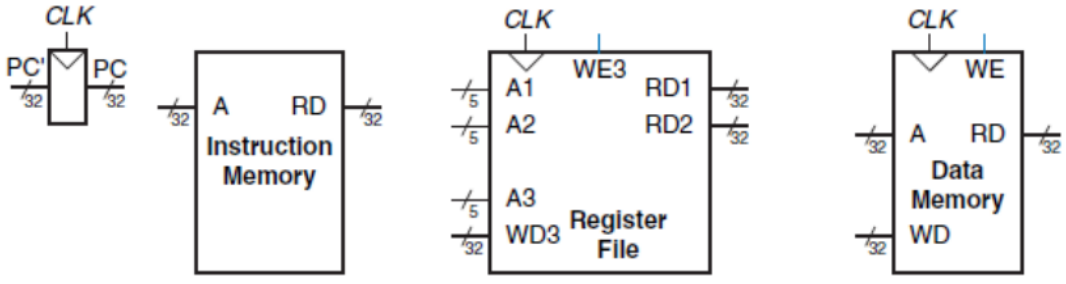
How instructions are Executed

- Fetch
- Decode
- Evaluate address
- fetch
- Execute
- Store results

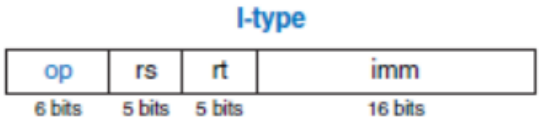
Basic hardware components that implement the cycle

- PC counter (instruction pointer)
- Instruction file
- Control unit
- Register File
- ALU
- Data memory

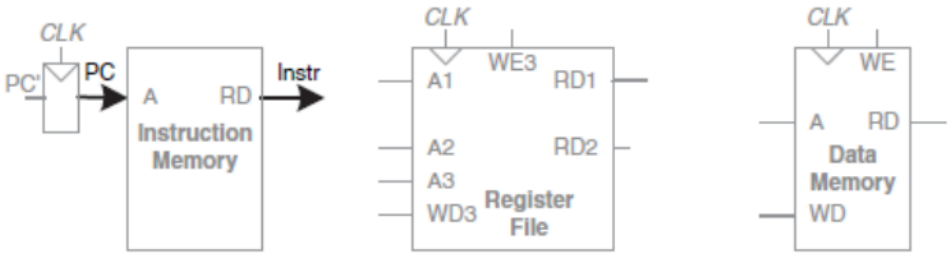
LW instruction implementation



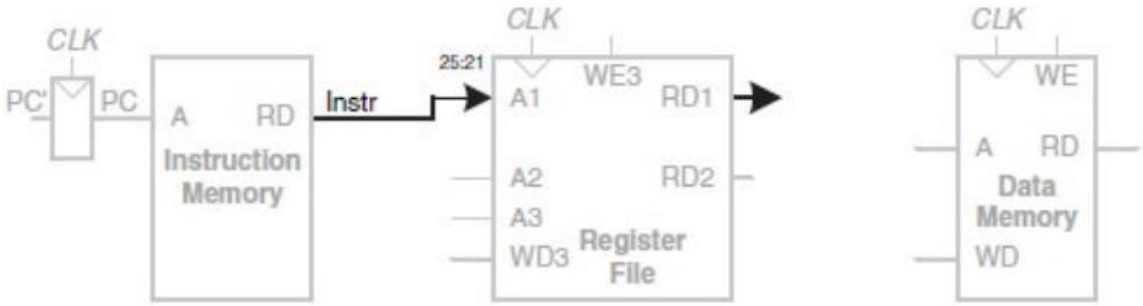
1- first step in a design, identify the type of instruction, and write the field variables as shown below.



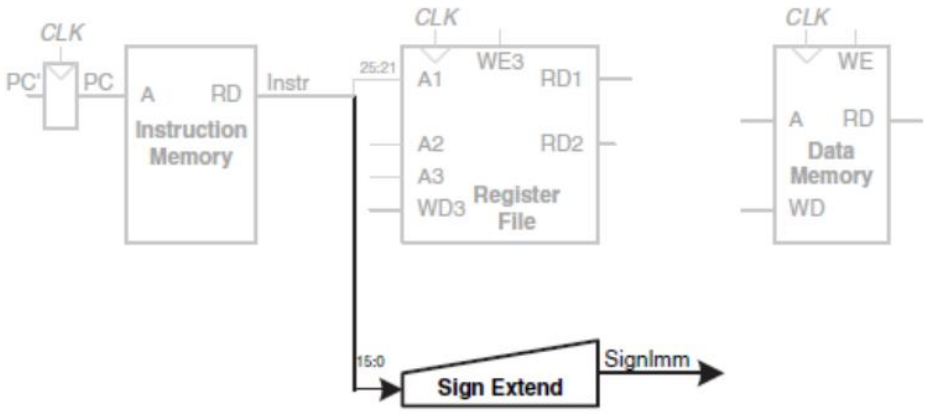
2- The program counter (PC) register contains the address of the instruction to execute. The first step is to read this instruction from instruction memory. The instruction memory reads out, or fetches, the 32-bit instruction.



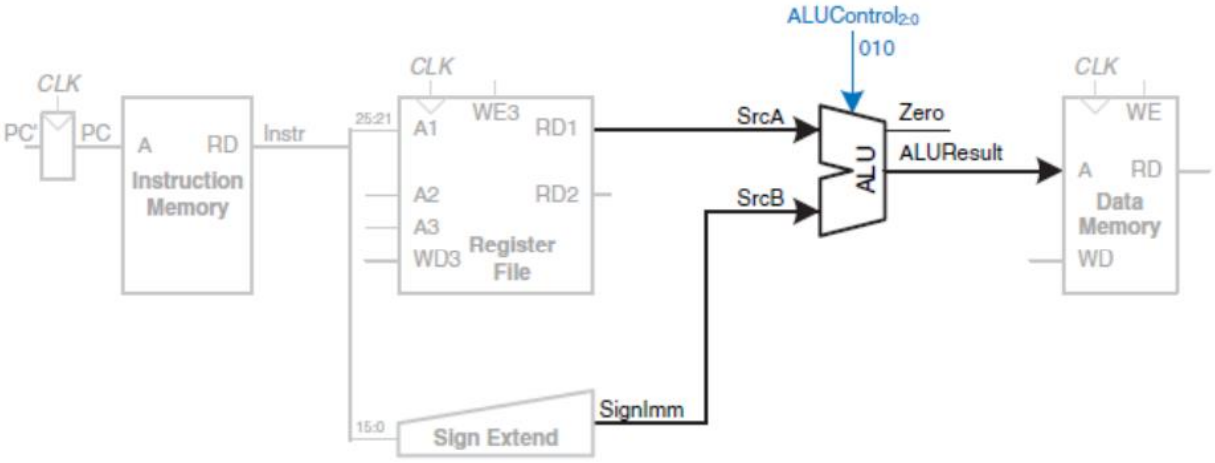
3- Read the source register containing the base address. This register is specified in the rs field of the instruction, Instr25:21. These bits of the instruction are connected to the address input of one of the register file read ports, A1. The register file reads the register value onto RD1.



4- For the immediate value, The lw instruction also requires an offset. The offset is stored in the immediate field of the instruction, Instr15:0. Because the 16-bit immediate might be either positive or negative, it must be signextended to 32 bits The 32-bit sign-extended value is called SignImm.

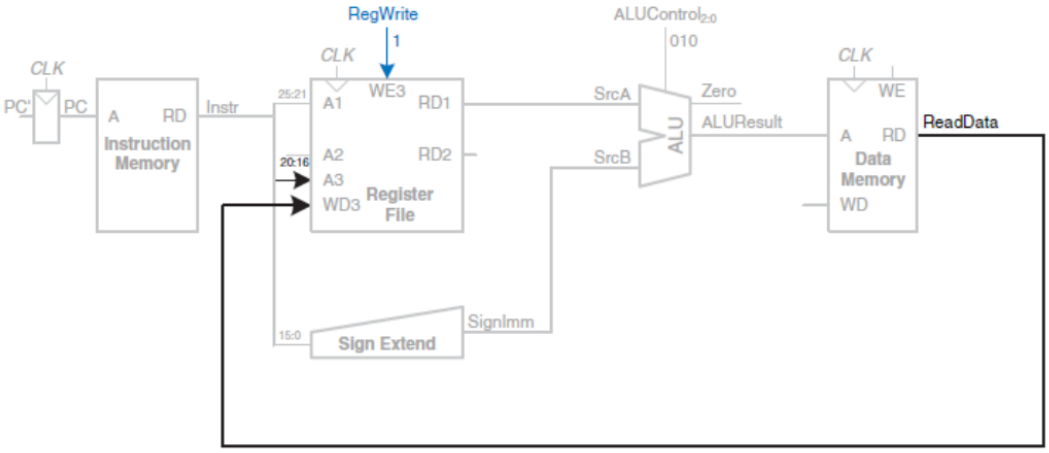


5- The processor must add the base address to the offset to find the address to read from memory .The ALU receives two operands, SrcA and SrcB. SrcA comes from the register file, and SrcB comes from the signextended immediate. The ALU can perform many operations, The 3-bit ALUControl signal specifies the operation. The ALU generates a 32-bit ALUResult and a Zero flag, that indicates whether ALUResult == 0. For a lw instruction, the ALUControl signal should be set to 010 to add the base address and offset. ALUResult is sent to the data memory as the address for the load instruction

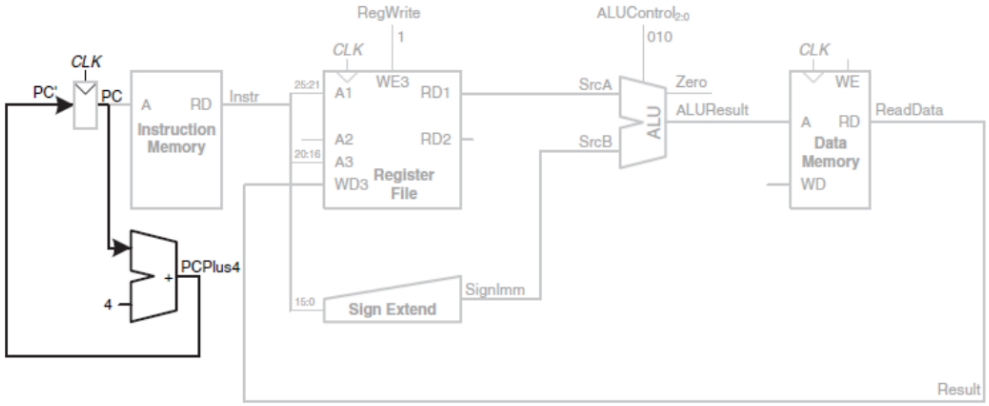


6- The data is read from the data memory onto the ReadData bus, then written back to the destination register in the register file at the end of the cycle Port 3 of the register file is the write

port. The destination register for the lw instruction is specified in the rt field, Instr20:16, which is connected to the port 3 address input, A3, of the register file. The ReadData bus is connected to the port 3 write data input, WD3, of the register file. A control signal called RegWrite is connected to the port 3 write enable input, WE3, and is asserted during a lw instruction so that the data value is written into the register file. The write takes place on the rising edge of the clock at the end of the cycle.

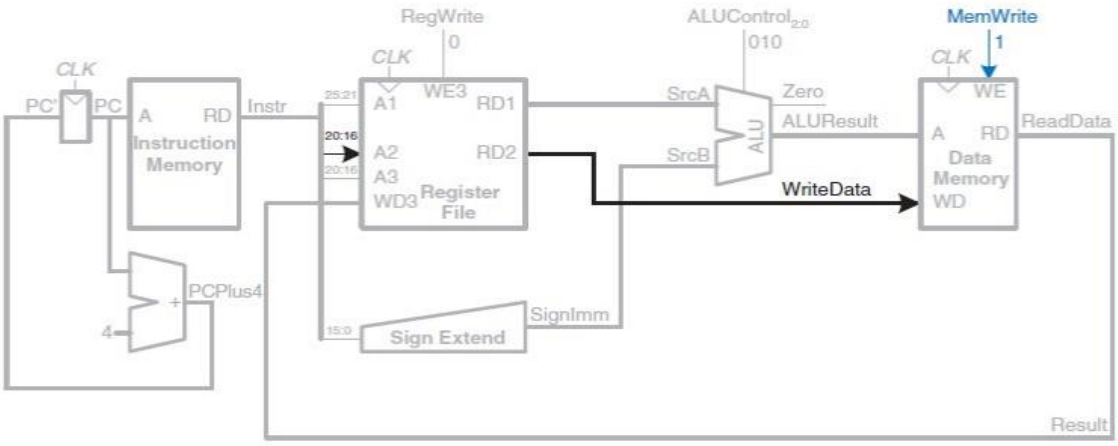


7- While the instruction is being executed, the processor must compute the address of the next instruction, PC′. Because instructions are 32 bits = 4 bytes, the next instruction is at PC + uses another adder to increment the PC by 4. The new address is written into the program counter on the next rising edge of the clock. This completes the datapath for the lw instruction.



SW instruction implementation

The sw instruction also reads a second register from the register file and writes it to the data memory. shows the new connections for this function. The register is specified in the rt field, Instr20:16. These bits of the instruction are connected to the second register file read port,A2. The register value is read onto the RD2 port. It is connected to the write data port of the data memory. The write enable port of the data memory, WE, is controlled by MemWrite. For a sw instruction, MemWrite = 1, to write the data to memory; ALUControl = 010, to add the base address and offset; and RegWrite = 0, because nothing should be written to the register file. Note that data is still read from the address given to the data memory, but that this ReadData is ignored because RegWrite = 0.



Modifying the Processor to implement R-Type instructions (add, sub …etc)

R-Type Format

R-type

op	rs	rt	rd	shamt	funct
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

1- Instructions read two registers from the register file.

2- The ALU performs an operation on these two registers, the ALU always received its SrcB operand from the sign-extended immediate (SignImm)

3- Add a multiplexer to choose SrcB from either the register file RD2 port or SignImm. The multiplexer is controlled by a new signal, ALUSrc. ALUSrc is 0 for R-type instructions to choose SrcB from the register file; it is 1 for w and sw to choose SignImm

4- The register file always got its write data from the data memory. However, R-type instructions write the ALUResult to the register file. Therefore, we add another multiplexer to choose between ReadData and ALUResult

5- the register to write was specified by the rt field of the instruction, Instr20:16, the register is specified by the rd field, Instr15:11, so we add a third multiplexer to choose WriteReg from the appropriate field of the instruction