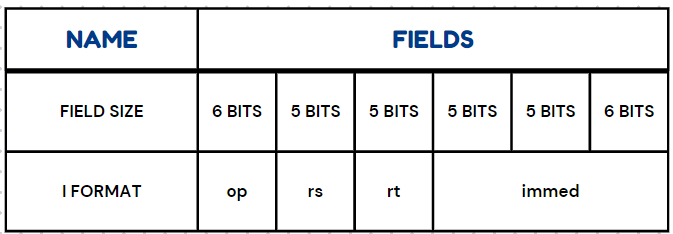
### **ADDING LOAD AND STORE INSTRUCTION**



op - used to identify the type of instruction.

rs - source register.

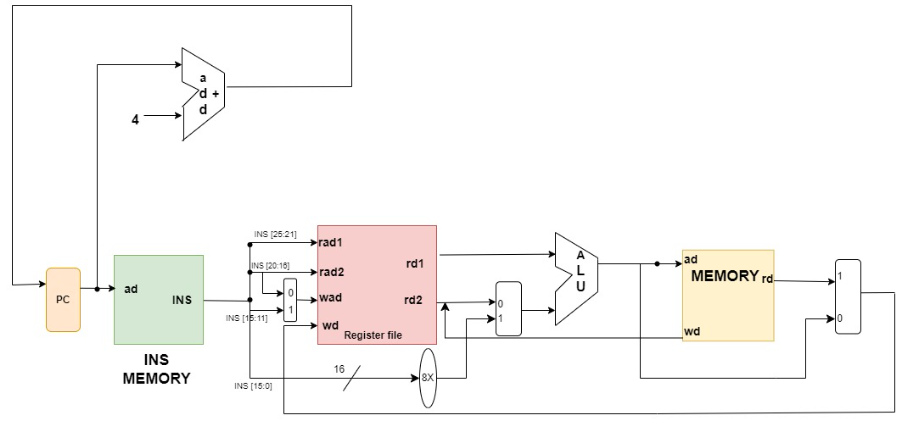
rt - either source or destination register( depends on instruction )

Immed - zero-extended if it is a logical operation.Else, it is sign extended.

**Load Instruction**

In a load instruction, data is fetched from memory using a calculated address and stored in a specified register.

**lw $rt, immed($rs)**

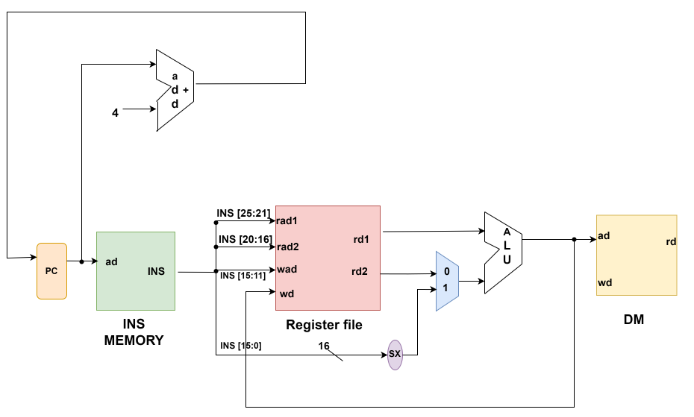
****

To implement a load word instruction, data from memory is written back to the register file. This requires adding a multiplexer to select the data source: ALU for arithmetic/logical instructions (control=0) or memory for load instructions (control=1). Another multiplexer selects the correct write address: bits 11-15 for arithmetic/logical instructions or bits 16-20 for load instructions. The control signal determines the appropriate input for each instruction type.

**Store Instruction**

In a store instruction, data from a specified register is stored in memory at a calculated address.

**sw $rt, immed($rs)**

****

The ALU in the processor receives inputs from the register file and immediate values. For load and store instructions, a multiplexer selects between the register file output (for R-class instructions) and the sign-extended immediate value for address calculation.

Multiplexers choose appropriate input paths based on the instruction type: immediate value for load/store, and register file output for add/subtract. The controller manages these multiplexers to ensure correct data path selection.

For store operations, data from the register file's specified register (rt) is connected to the memory's data input.