

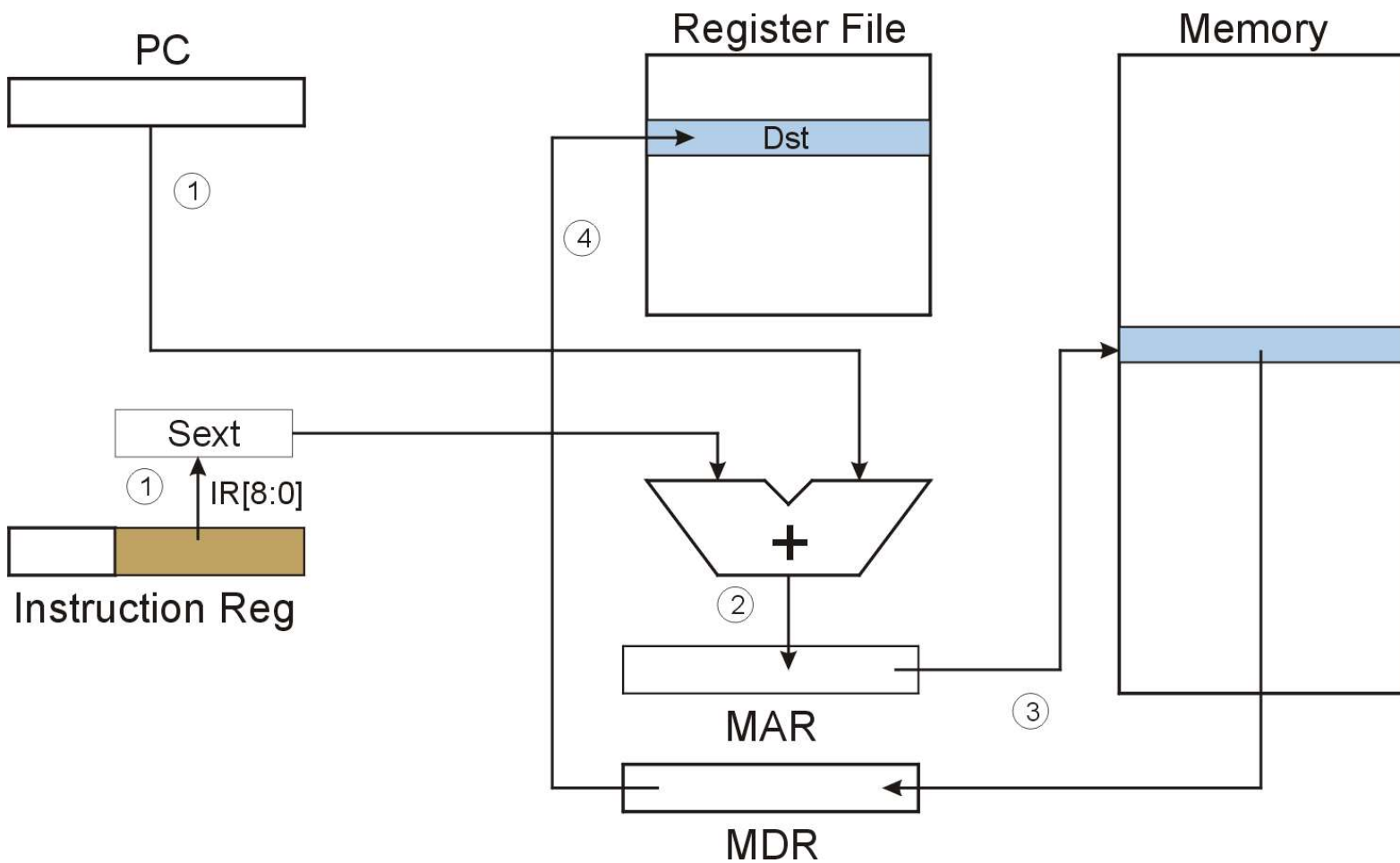
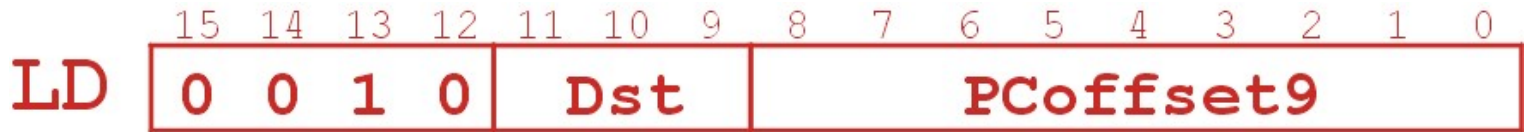
LC3 Instruction Review

Load Instructions

- **LD**
 - Load a value at the address calculated by adding the current PC and a signed offset found in the lowest nine bits of the instruction
 - PC has been incremented to the address of the next instruction
- **LDI**
 - Similar to LD except one additional indirection
 - Value generated by LD is taken as an address and then value at that address is the final value stored in the register
 - Double indirection
 - Ex: argv
 - Can reach anywhere in memory

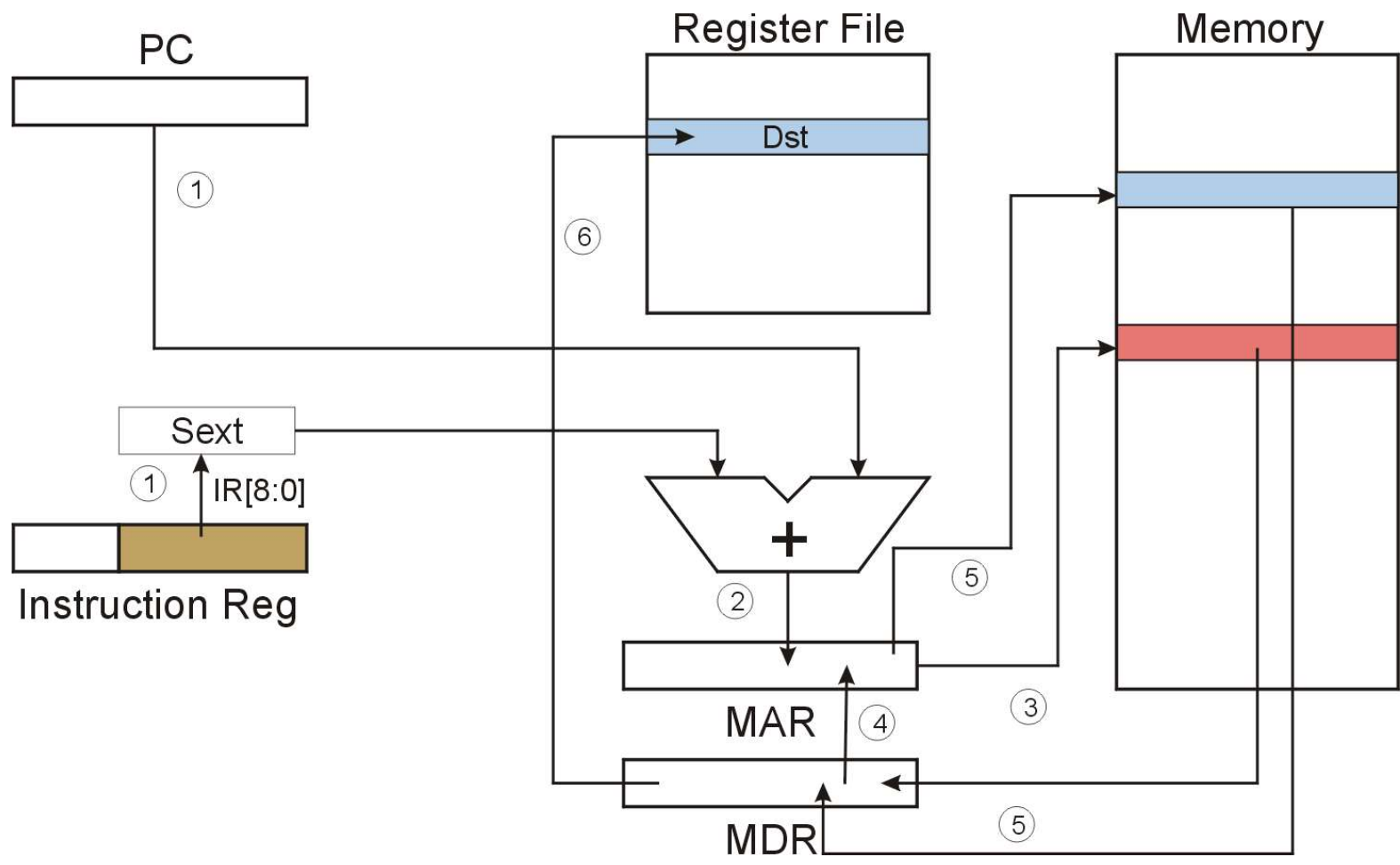
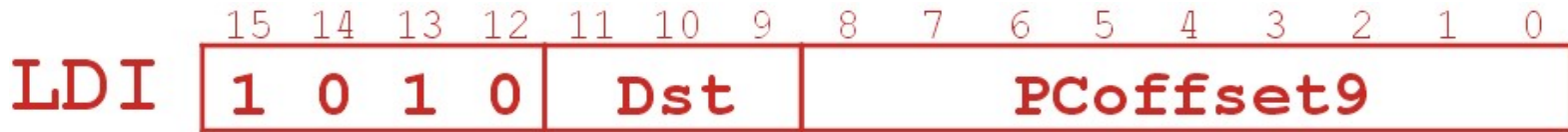
Assembly Ex:
LD R1, Label1

LD (PC-Relative)



LDI (Indirect)

Assembly Ex:
LDI R4, Adr



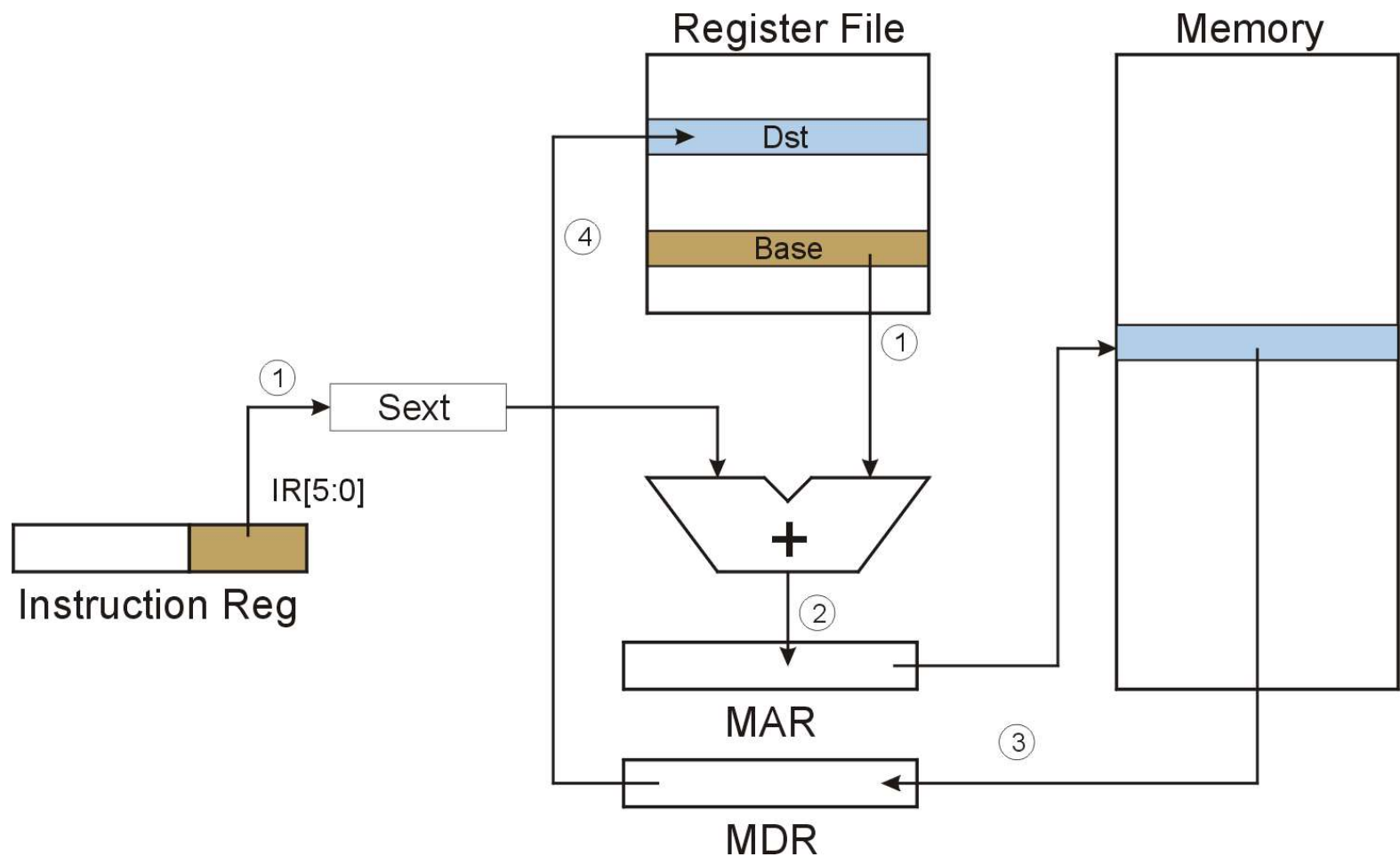
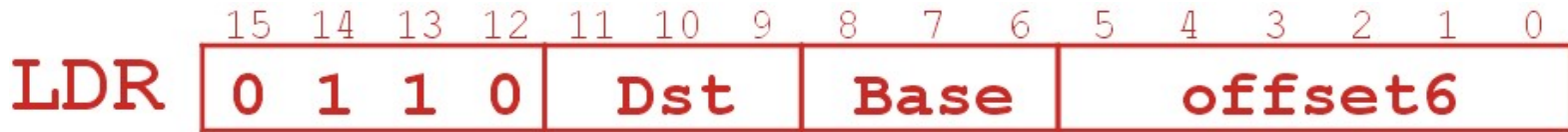
Load Instructions

- **LDR**
 - **Already have an address in a register**
 - **Can load from anywhere in memory**
 - **Single indirection**
 - **Access elements in an array**
 - **Least significant 6 bits used as a signed offset**

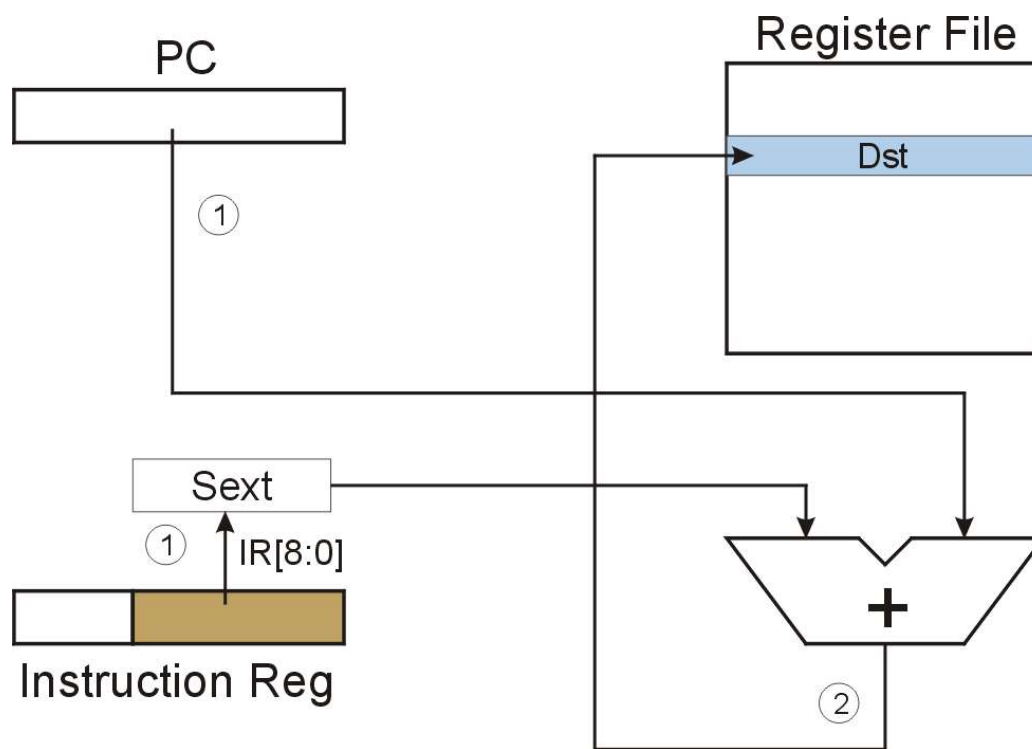
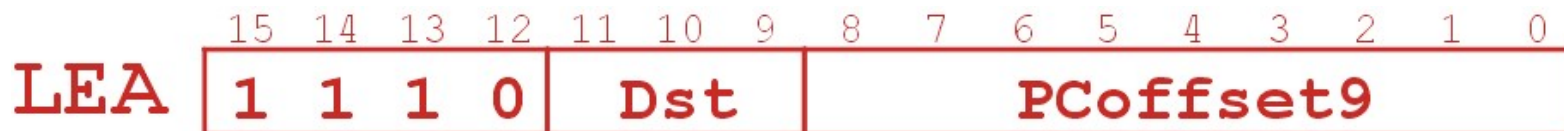
- **LEA**
 - **Load an **address** into a register**
 - **Other loads are values**
 - **Address produced by PC and signed value in least significant nine bits of instruction**
 - **PC has been incremented**
 - **Put base address of array in a register so elements of the array can be accessed by LDR**

LDR (Base+Offset)

Assembly Ex:
LDR R4, R1, #1



LEA (Immediate)



Assembly Ex:
LEA R1, Lab1

Condition Codes

Negative, Zero, Positive

- **nzp**

Set by the following instructions

- **ADD, AND, NOT**
- **LD, LDR, LDI**
- **LEA**

Set by any instruction that writes a destination register

Only one value set at any one time

- **n or z or p**