COMPUTER ORGANIZATION (IS F242)

LECT 16: LC 3 ARCHITECTURE

Data Movement Instructions

Moves information between

- General Purpose Registers (GPR) & Memory
- General Purpose Registers (GPR) & IO Devices

Load → Moving information from memory to registers

Store → Moving information from registers to memory

LC-3 can not move data from memory to memory

Load and Store

- Do not affect the information in the source location
- Overwrite the contents in the destination

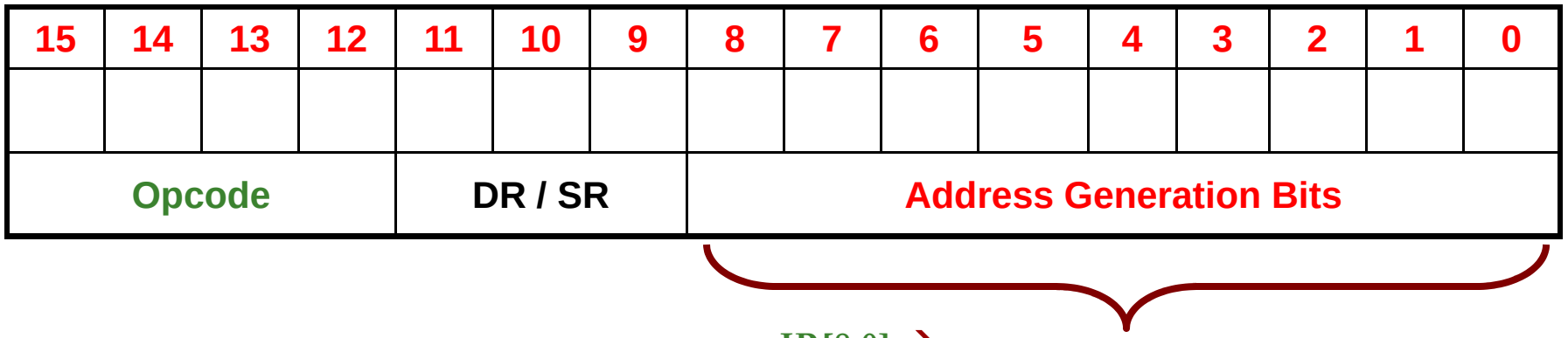
7 data movement instructions with LC 3

- LD, LDR, LDI, LEA, ST, STR, STI

Format of Load / Store Instructions

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Opcode				DR / SR			Address Generation Bits								

Format of Load / Store Instructions



IR[8:0] →

Used to generate 16 bit address in the memory.

Four ways to interpret these 9 bits, collectively called as *addressing modes*.

Data Movement Instructions

- **Load -- read data from memory to register**
  - **LD: PC-relative mode**
  - **LDR: base+offset mode**
  - **LDI: indirect mode**
- **Store -- write data from register to memory**
  - **ST: PC-relative mode**
  - **STR: base+offset mode**
  - **STI: indirect mode**
- **Load effective address -- compute address, save in register**
  - **LEA: immediate mode**
  - **does not access memory**

PC Relative Mode – Direct Addressing

Want to specify address directly in the instruction

- But an address is 16 bits, and so is an instruction!
- After subtracting 4 bits for opcode
  and 3 bits for register, we have 9 bits available for address (IR[8:0]).

Solution:

- Use the 9 bits as a *signed offset* from the current PC.

9 bits → $-256 \leq \text{offset} \leq +255$

Can form any address X, such that → $PC - 256 \leq X \leq PC + 255$

(from the current instruction -255 to +256)

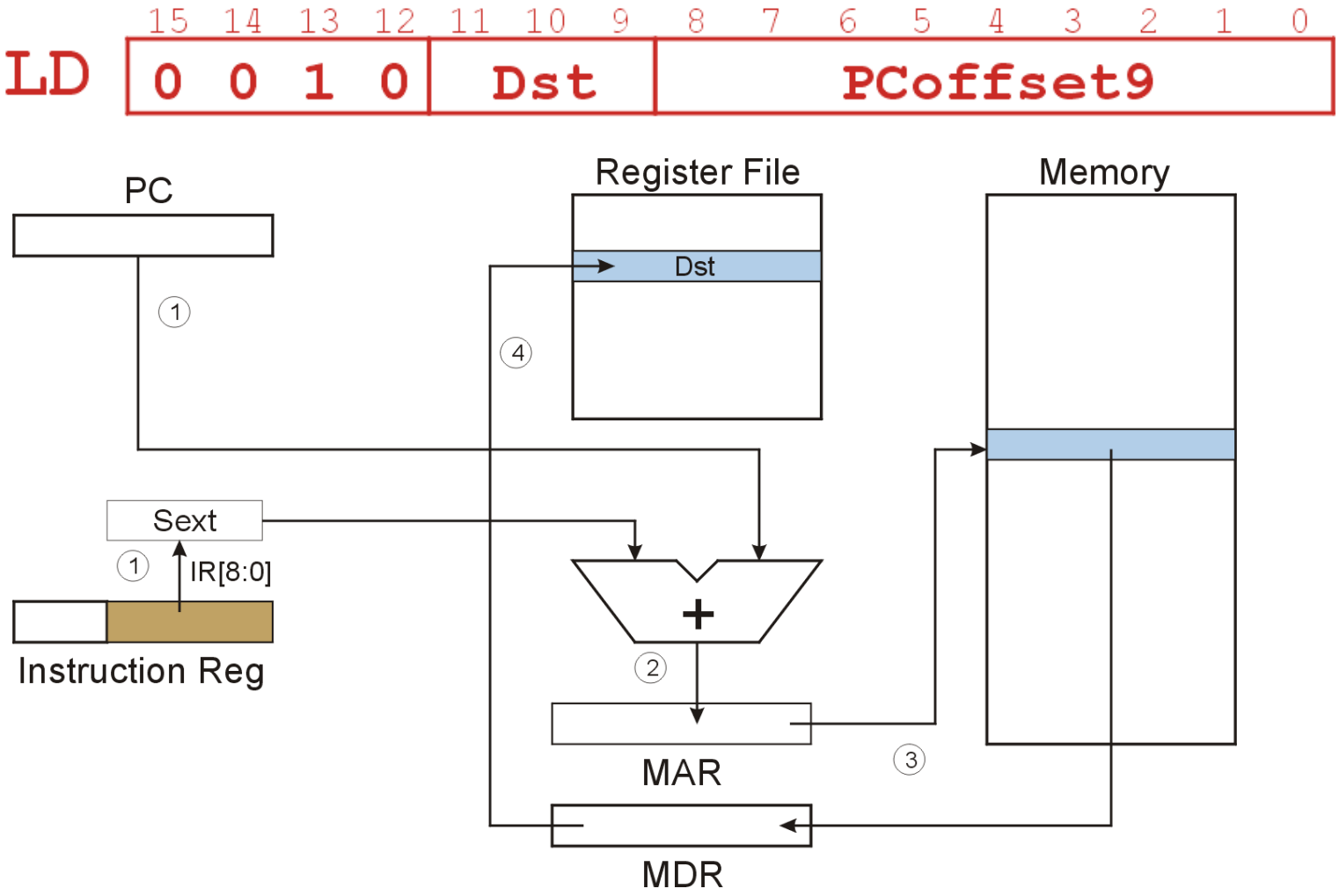
Effective Address = PC + SEXT(IR[8:0])

Remember that PC is incremented as part of the FETCH phase;

- Hence, the incremented PC is used in calculating the address

- Operand location must be within 256 locations of the instruction

PC Relative Mode (Load LD Instruction)



What does the following instruction do ?

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	1	0	1	1	0	1	0	1	1	1	1
Opcode LD				Destination Register R2			PC Offset (9 bits) x1AF								

Assume this LD instruction is in the location x2345. ➔ While this instruction is in execution, PC will have x2346.

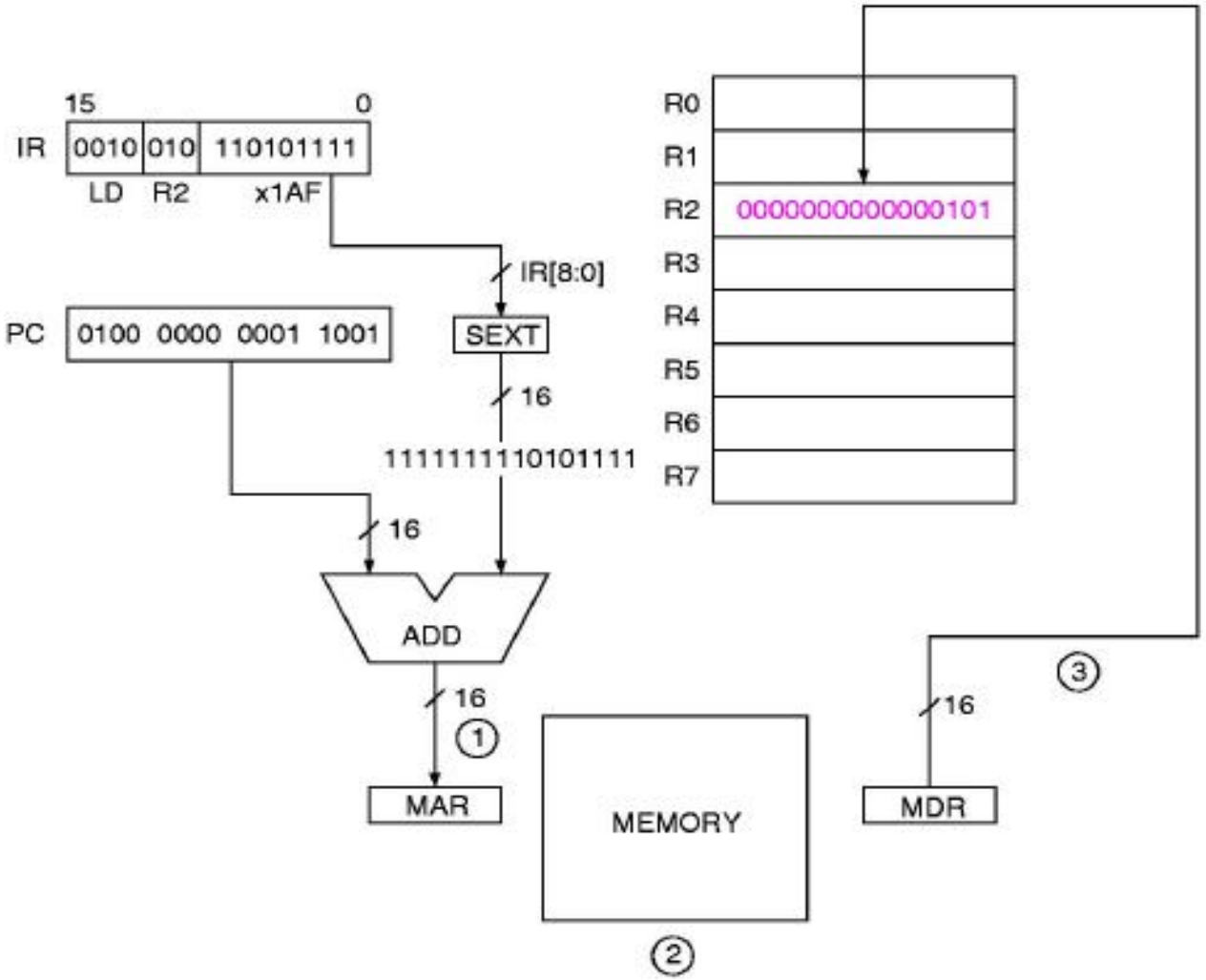
Sign Extended offset is xFFAF

The address for load instruction is now xFFAF + x 2346

The content of memory location xFFAF + x 2346 will be loaded into R2

LD data path

LD R2, x1AF



ST (PC-Relative)