

Computer Science 210 s1c
Computer Systems 1
2008 Semester 1
Lecture Notes

Lecture 13, 3Apr08:

The LC-3 ISA

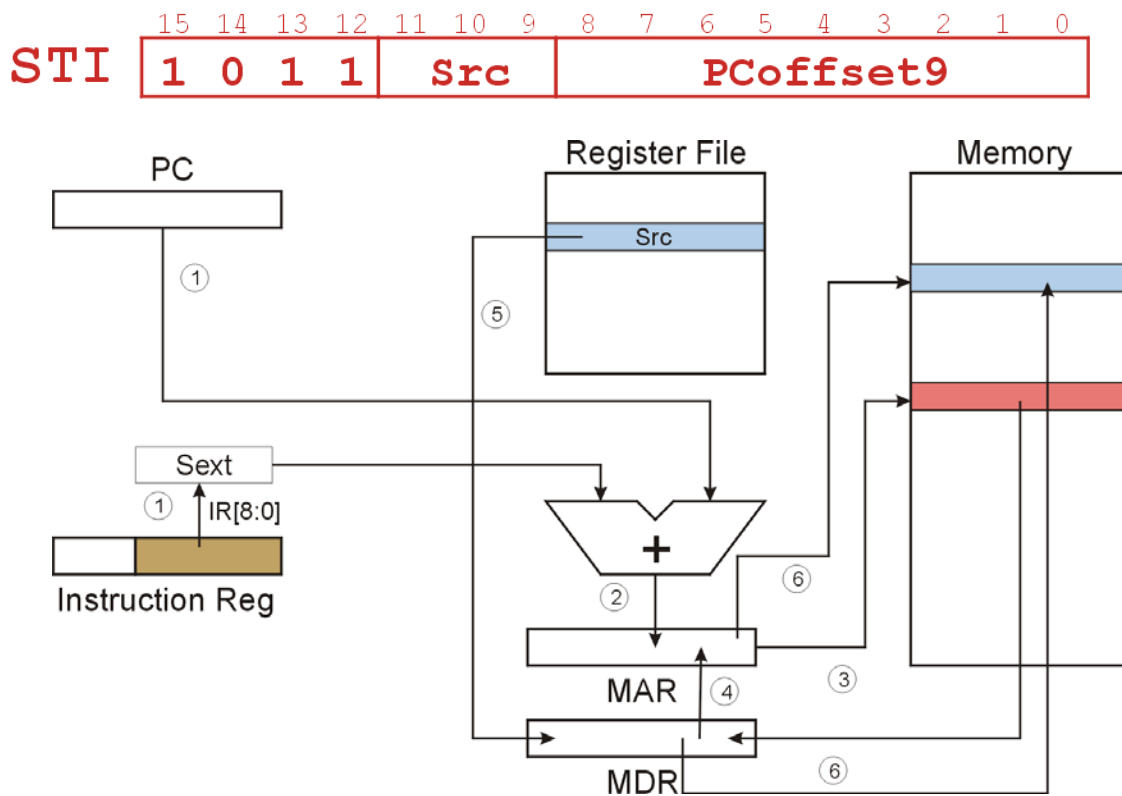
James Goodman



Credits: Slides prepared by Gregory T. Byrd, North Carolina State University

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

STI (Indirect)



Base + Offset Addressing Mode

With PC-relative mode, can only address data within 256 words of the instruction.

- What about the rest of memory?

Solution #2:

- Use a register to generate a full 16-bit address.

4 bits for opcode, 3 for src/dest register,
3 bits for *base* register -- remaining 6 bits are used
as a *signed offset*.

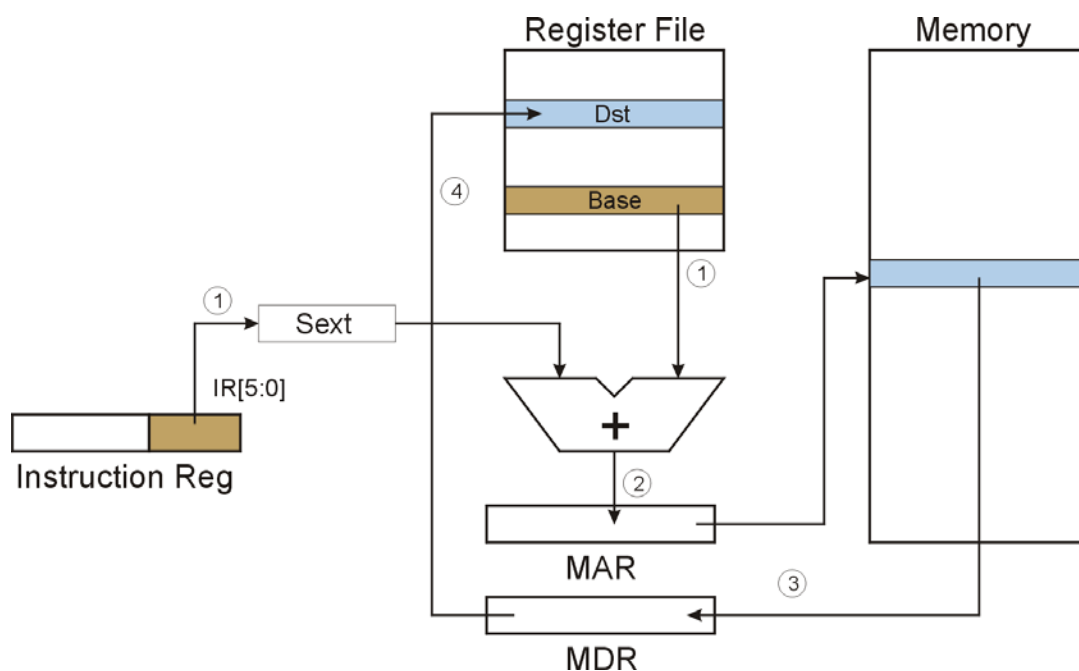
- **Offset is sign-extended before adding to base register.**

3-Apr-08

CS210

231

LDR (Base+Offset)

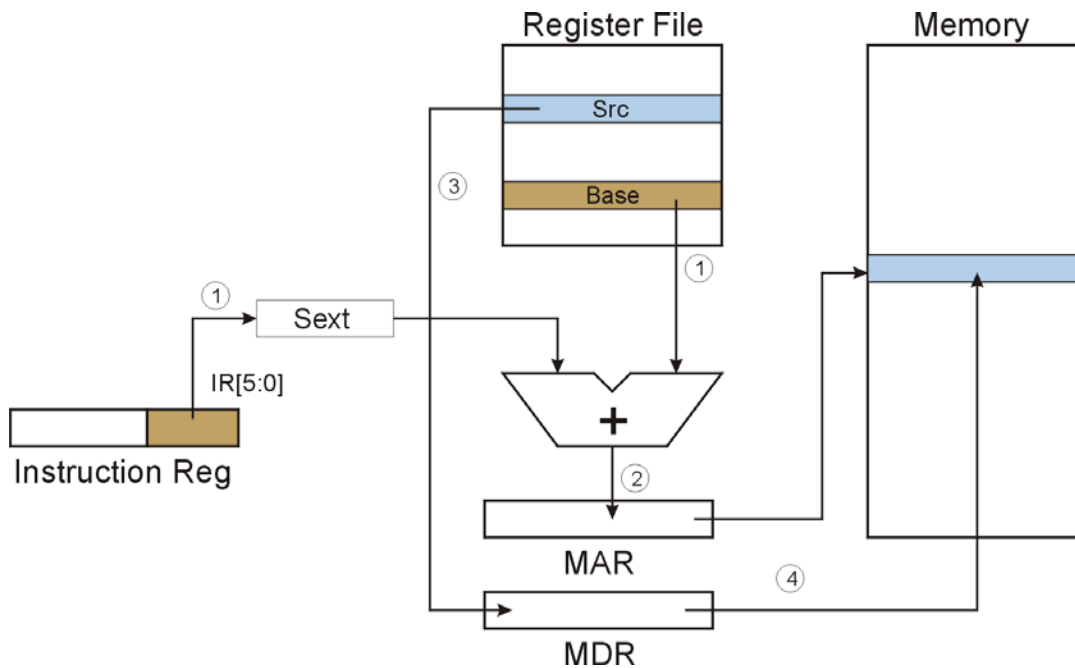


3-Apr-08

CS210

232

STR (Base+Offset)



3-Apr-08

CS210

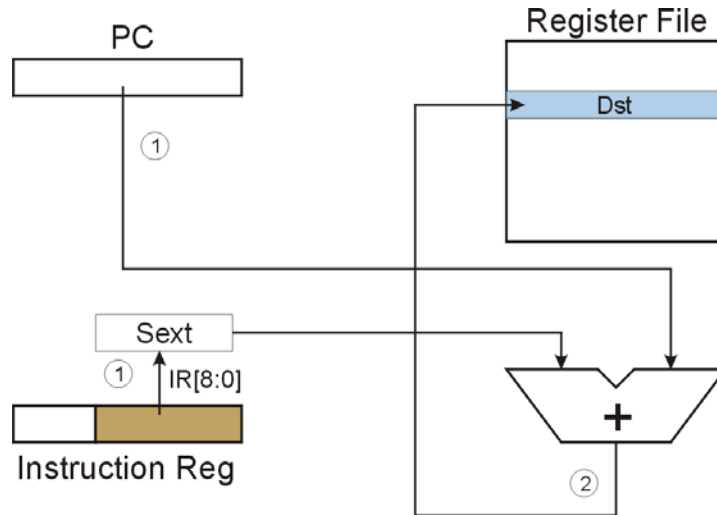
233

Load Effective Address

Computes address like PC-relative (PC plus signed offset) and **stores the result into a register.**

Note: The *address* is stored in the register, not the contents of the memory location.

LEA (Immediate)



3-Apr-08

CS210

235

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

Example

Address	Instruction																Comments
x30F6	1	1	1	0	0	0	1	1	1	1	1	1	1	0	1		$R1 \leftarrow PC - 3 = x30F4$
x30F7	0	0	0	1	0	1	0	0	0	1	1	0	1	1	1	0	$R2 \leftarrow R1 + 14 = x3102$
x30F8	0	0	1	1	0	1	0	1	1	1	1	1	1	0	1	1	$M[PC - 5] \leftarrow R2$ $M[x30F4] \leftarrow x3102$
x30F9	0	1	0	1	0	1	0	0	1	0	1	0	0	0	0	0	$R2 \leftarrow 0$
x30FA	0	0	0	1	0	1	0	0	1	0	1	0	0	1	0	1	$R2 \leftarrow R2 + 5 = 5$
x30FB	0	1	1	1	0	1	0	0	0	1	0	0	1	1	1	0	$M[R1+14] \leftarrow R2$ $M[x3102] \leftarrow 5$
x30FC	1	0	1	0	0	1	1	1	1	1	1	1	0	1	1	1	$R3 \leftarrow M[M[x30F4]]$ $R3 \leftarrow M[x3102]$ $R3 \leftarrow 5$

opcode

3-Apr-08

CS210

236

Control Instructions

Used to alter the sequence of instructions
(by changing the Program Counter)

Conditional Branch

- branch is *taken* if a specified condition is true
 - signed offset is added to PC to yield new PC
- else, the branch is *not taken*
 - PC is not changed, points to the next sequential instruction

Unconditional Branch (or Jump)

- always changes the PC

TRAP

- changes PC to the address of an OS “service routine”
- routine will return control to the next instruction (after TRAP)

Condition Codes

LC-3 has three **condition code** registers:

N -- negative

Z -- zero

P -- positive (greater than zero)

Set by any instruction that writes a value to a register
(ADD, AND, NOT, LD, LDR, LDI, LEA)

Exactly one will be set at all times

- Based on the last instruction that altered a register

Branch Instruction

Branch specifies one or more condition codes.

If the set bit is specified, the branch is taken.

- PC-relative addressing:
target address is made by adding signed offset (IR[8:0]) to current PC.
- Note: PC has already been incremented by FETCH stage.
- Note: Target must be within 256 words of BR instruction.

If the branch is not taken,
the next sequential instruction is executed.

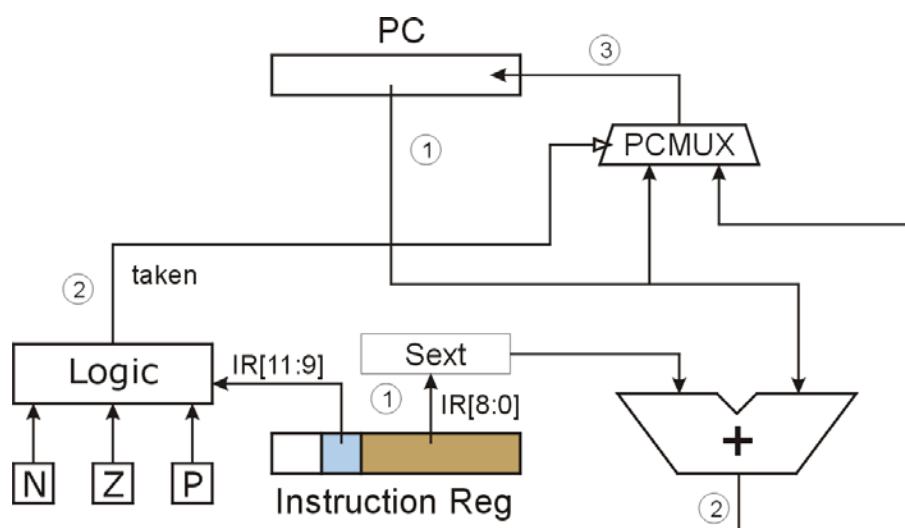
3-Apr-08

CS210

239

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

BR (PC-Relative)



What happens if bits [11:9] are all zero? All one? CS210

240

Using Branch Instructions

Compute sum of 12 integers.

Numbers start at location x3100. Program starts at location x3000.

