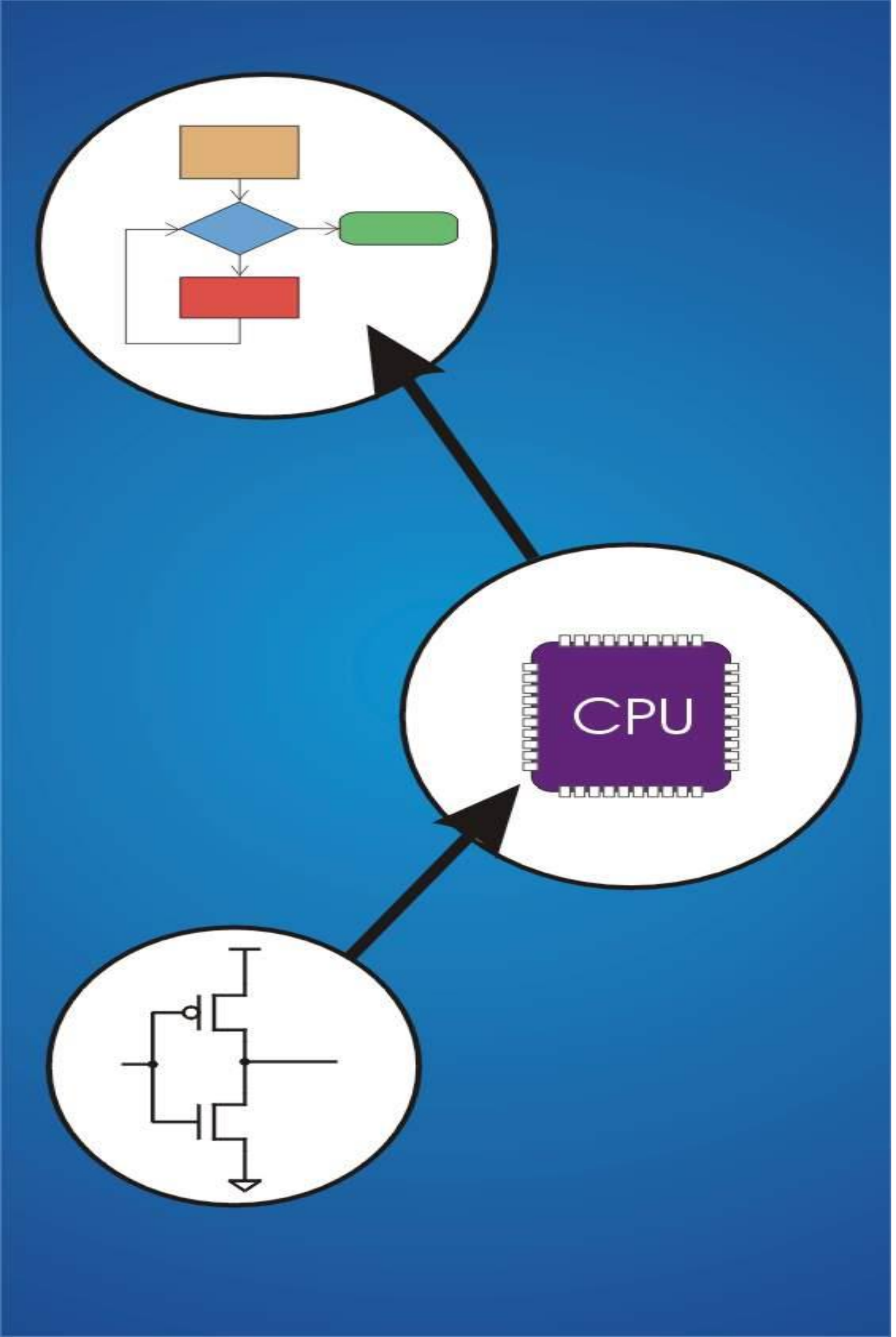
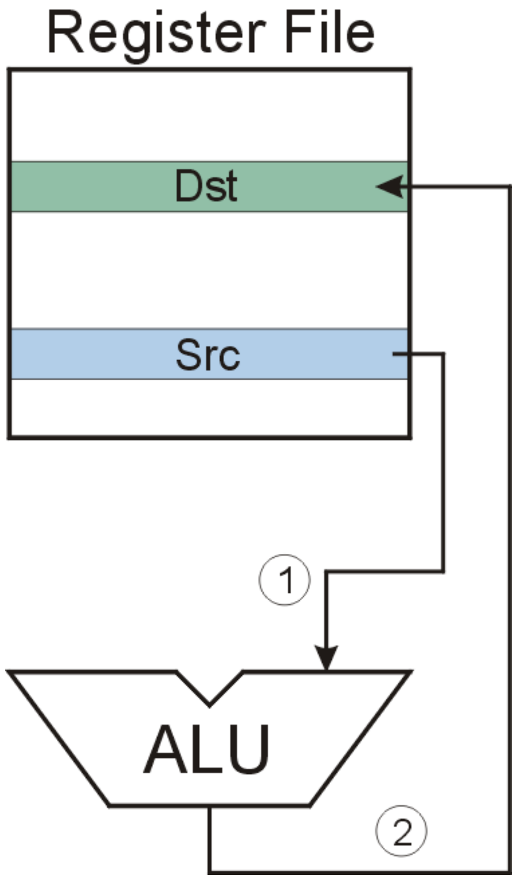


LC3 Instruction Diagrams

NOT (Register)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT	1	0	0	1	Dst			Src			1	1	1	1	1	1



Assembly Ex:
NOT R3, R2

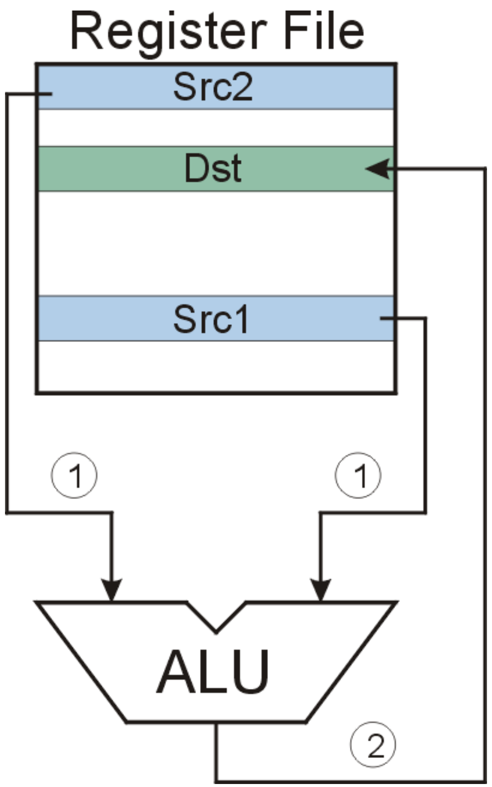
Note: Src and Dst could be the <u>same</u> register.

ADD/AND (Register)

this zero means “register mode”

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD	0	0	0	1	Dst			Src1			0	0	0	Src2		

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AND	0	1	0	1	Dst			Src1			0	0	0	Src2		



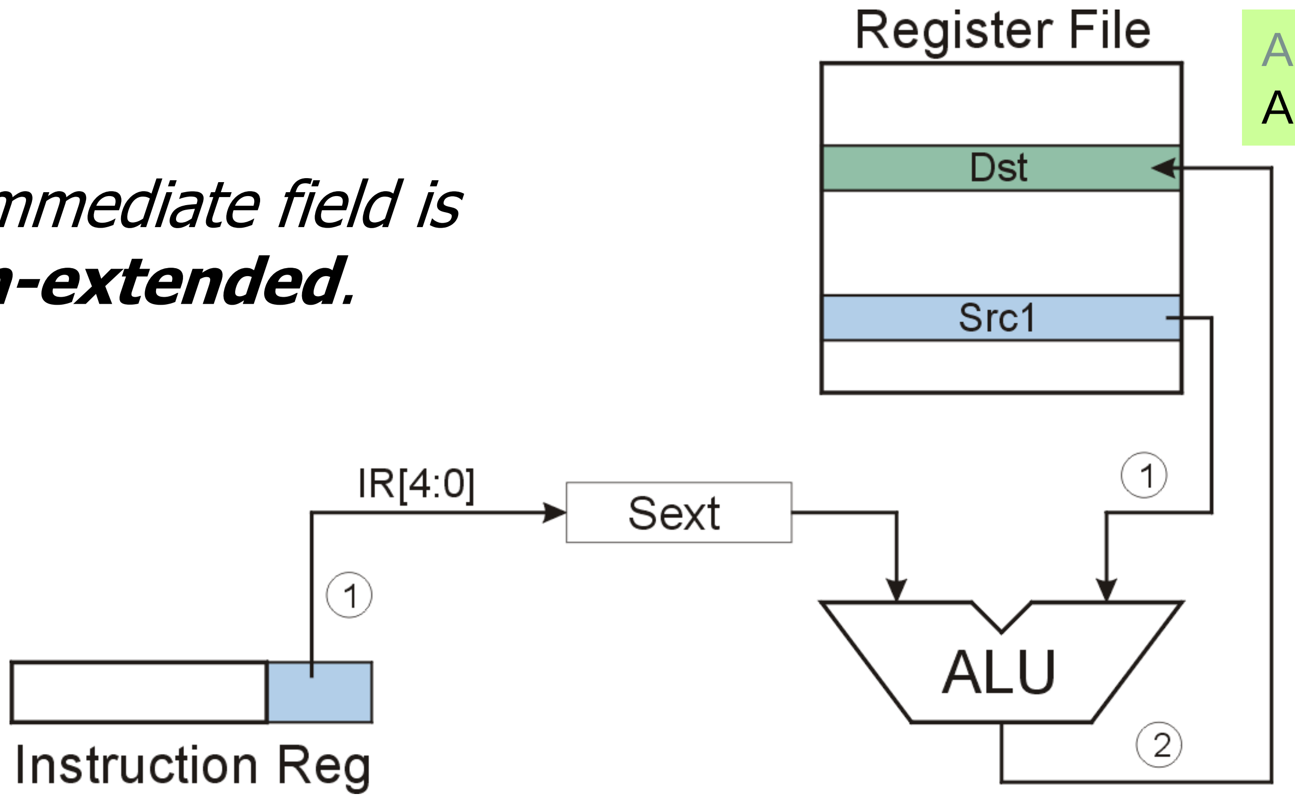
Assembly Ex:
Add R3, R1, R3

ADD/AND (Immediate)

this one means “immediate mode”

	15	14	13	12	11 10 9	8 7 6	5	4 3 2 1 0
ADD	0	0	0	1	Dst	Src1	1	Imm5
AND	0	1	0	1	Dst	Src1	1	Imm5

Note: Immediate field is ***sign-extended****.*

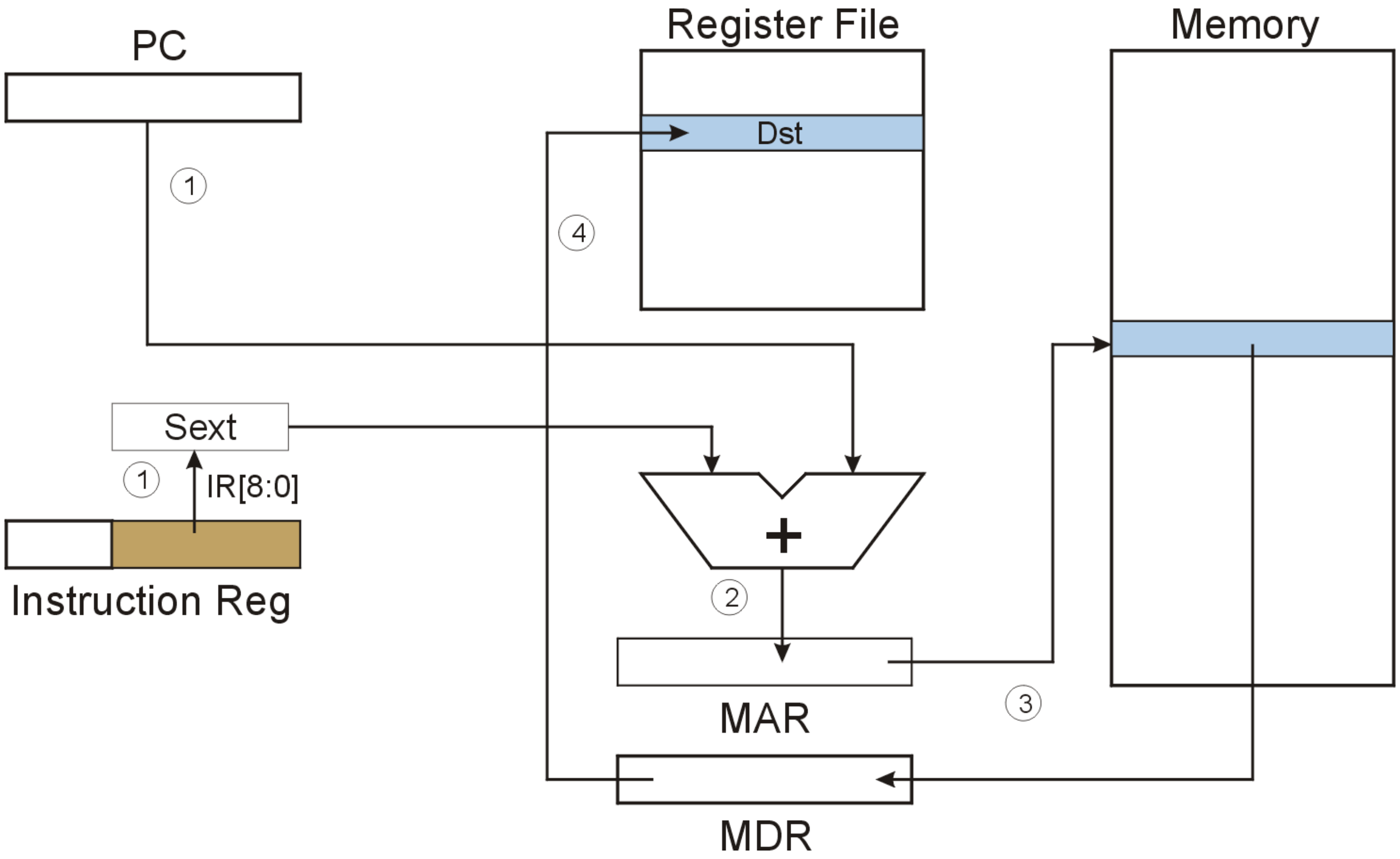


Assembly Ex:
Add R3, R3, #1

LD (PC-Relative)

Assembly Ex:
LD R1, Label1

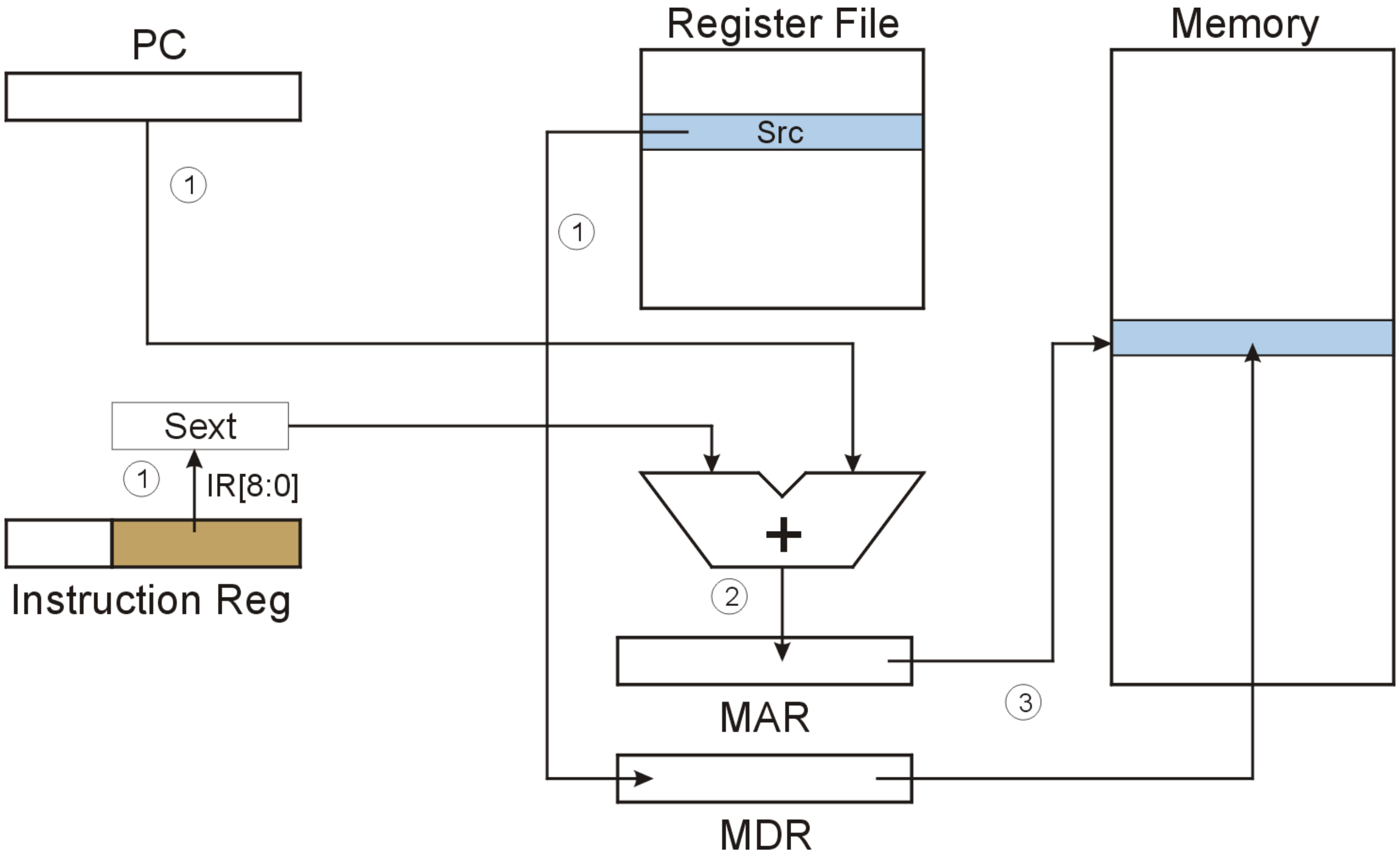
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LD	0	0	1	0	Dst			PCoffset9								



ST (PC-Relative)

Assembly Ex:
ST R1, Label2

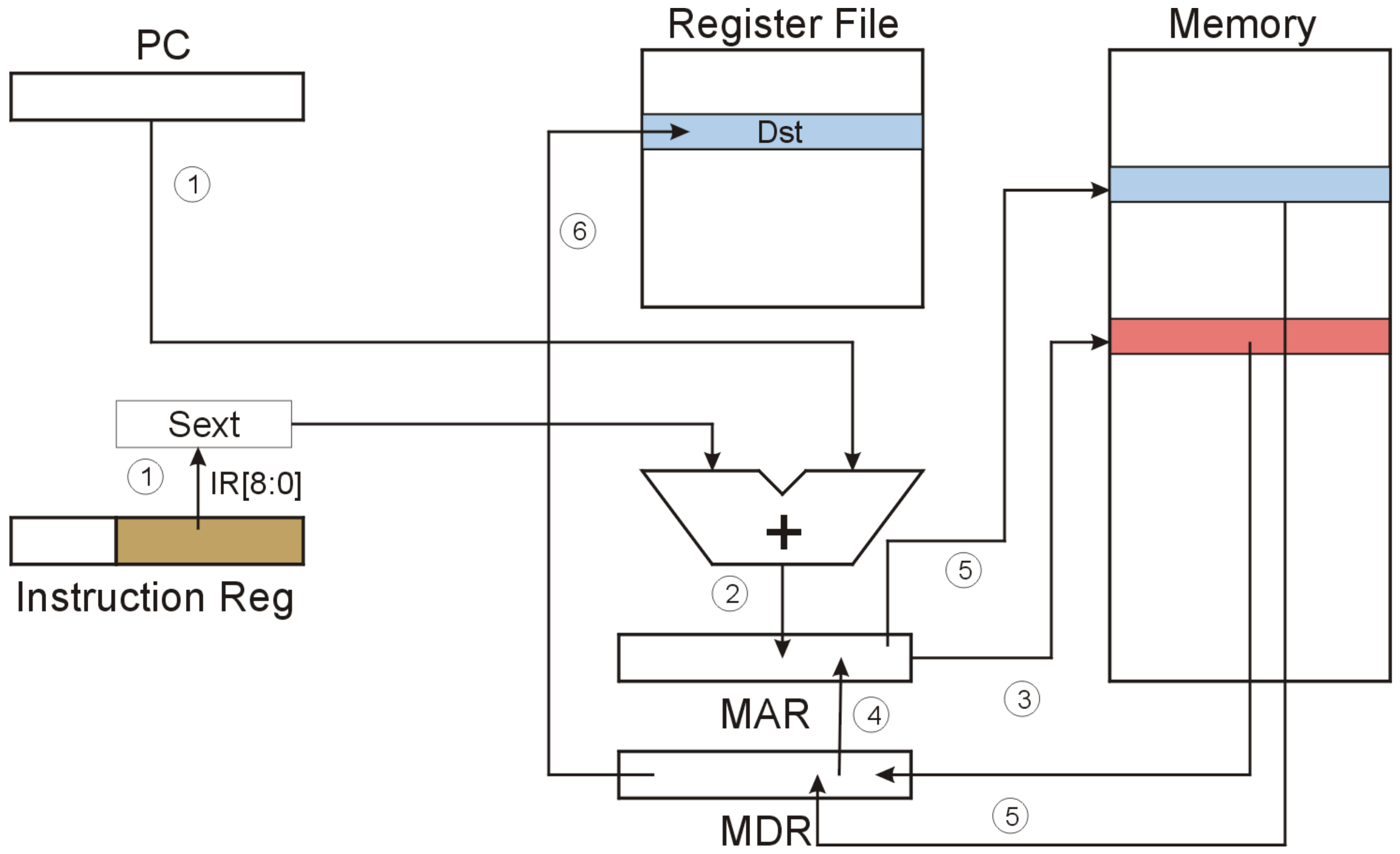
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST	0	0	1	1	Src			PCoffset9								



LDI (Indirect)

Assembly Ex:
LDI R4, Adr

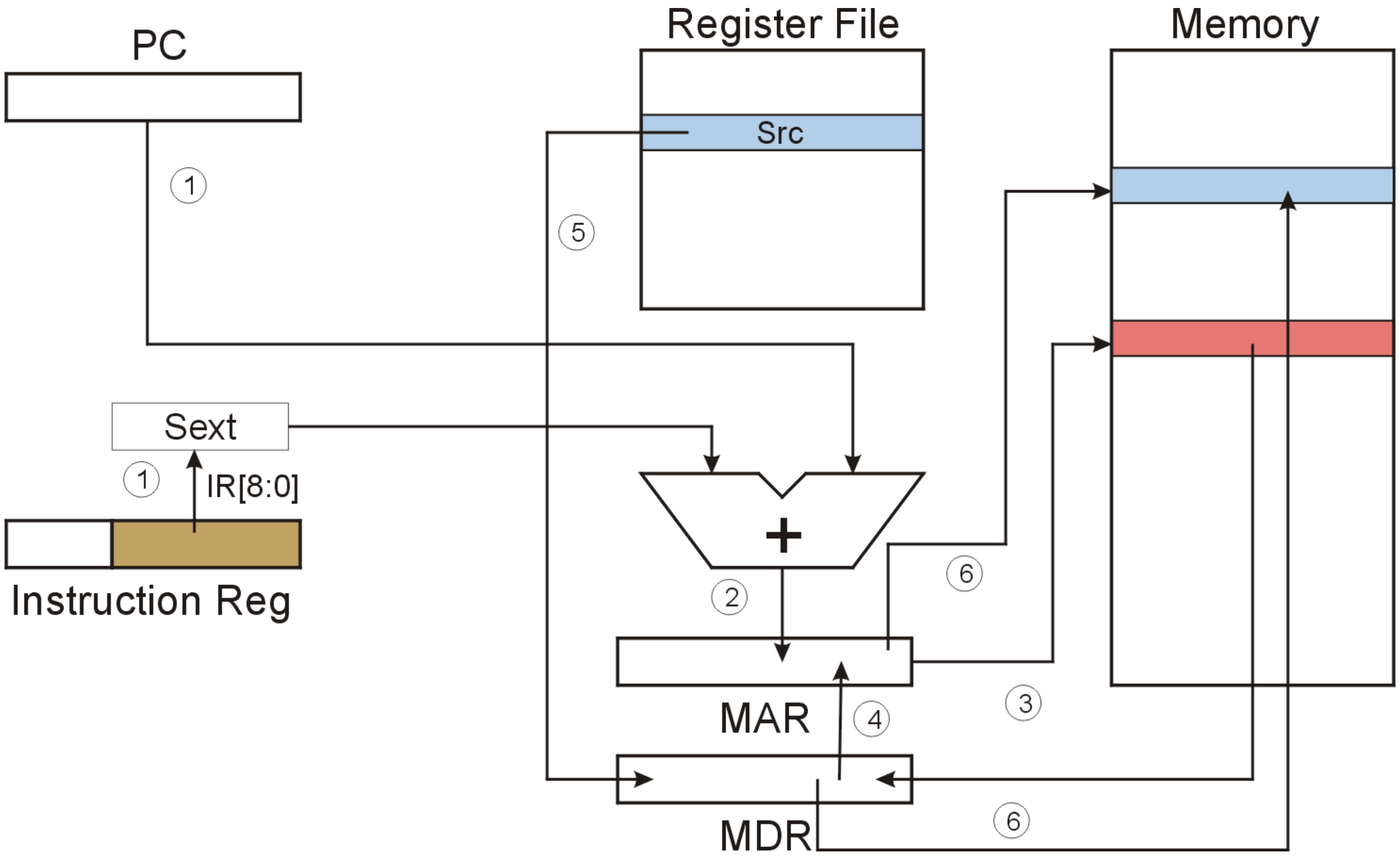
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LDI	1	0	1	0	Dst			PCoffset9								



STI (Indirect)

Assembly Ex:
STI R4, Adr

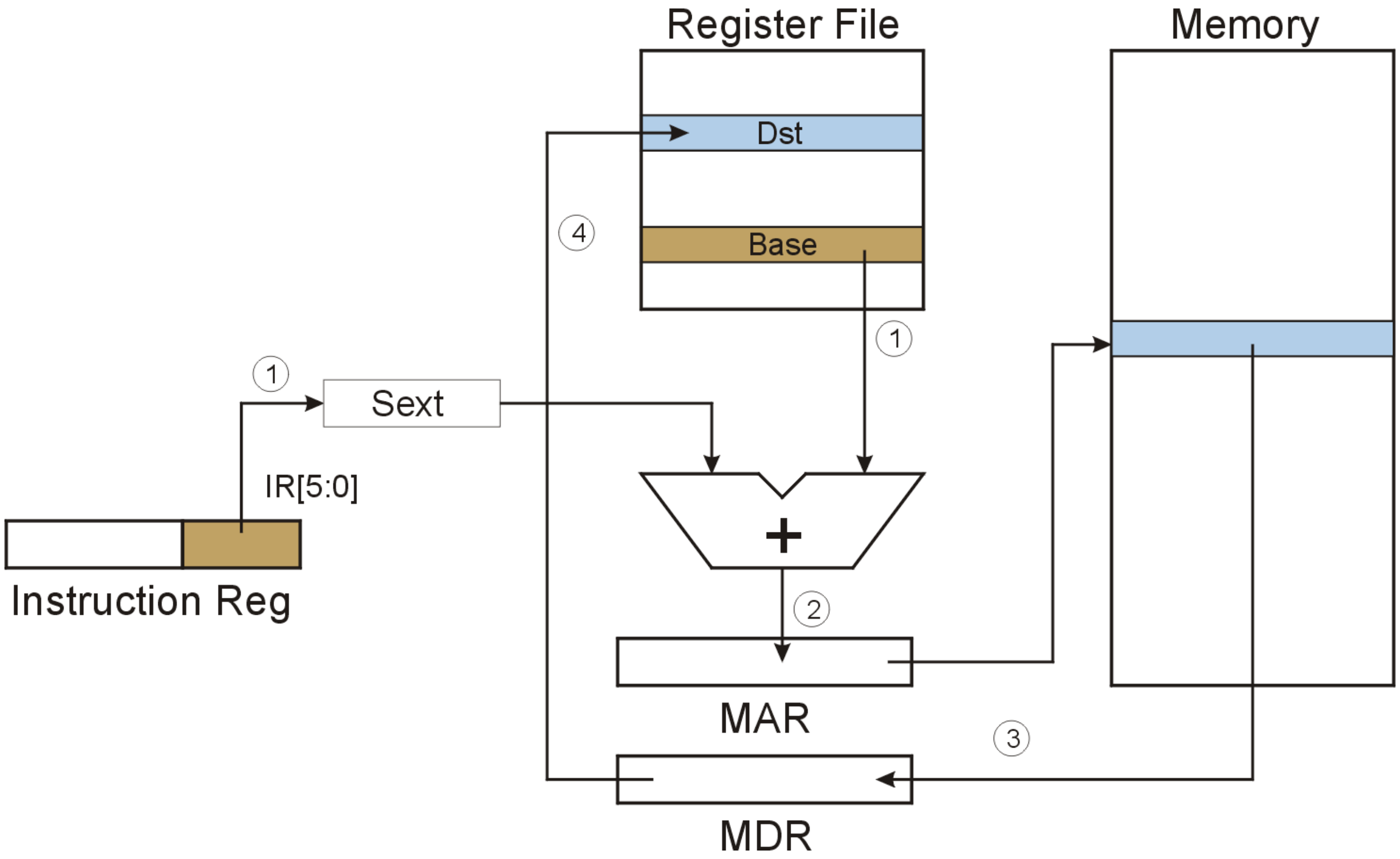
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STI	1	0	1	1	Src			PCoffset9								



LDR (Base+Offset)

Assembly Ex:
LDR R4, R1, #1

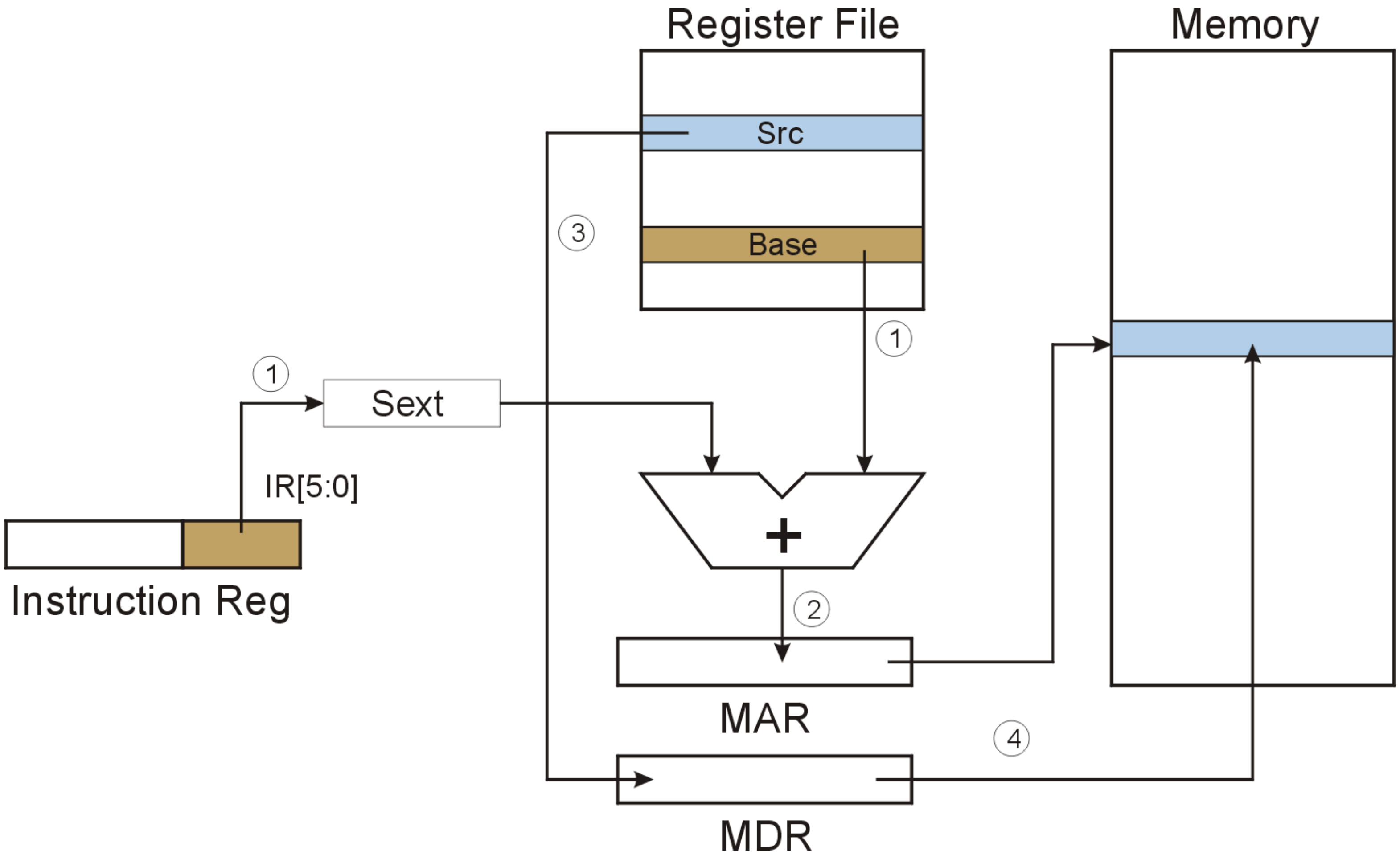
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LDR	0	1	1	0	Dst			Base			offset6					



STR (Base+Offset)

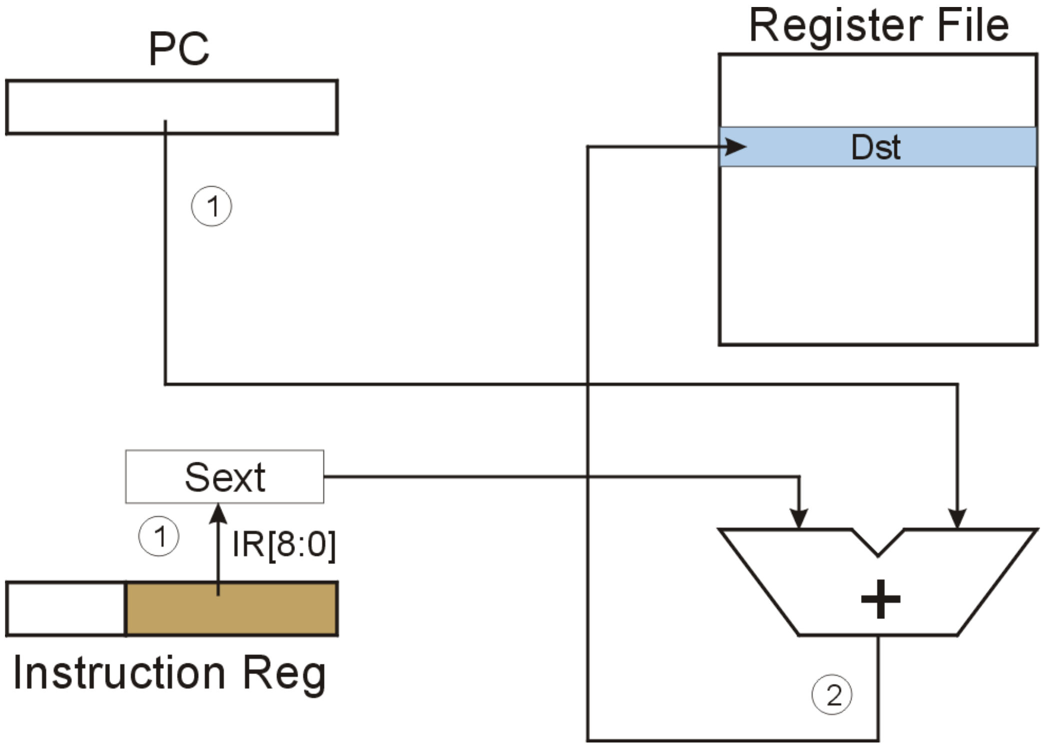
Assembly Ex:
STR R4, R1, #1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STR	0	1	1	1	Src			Base			offset6					



LEA (Immediate)

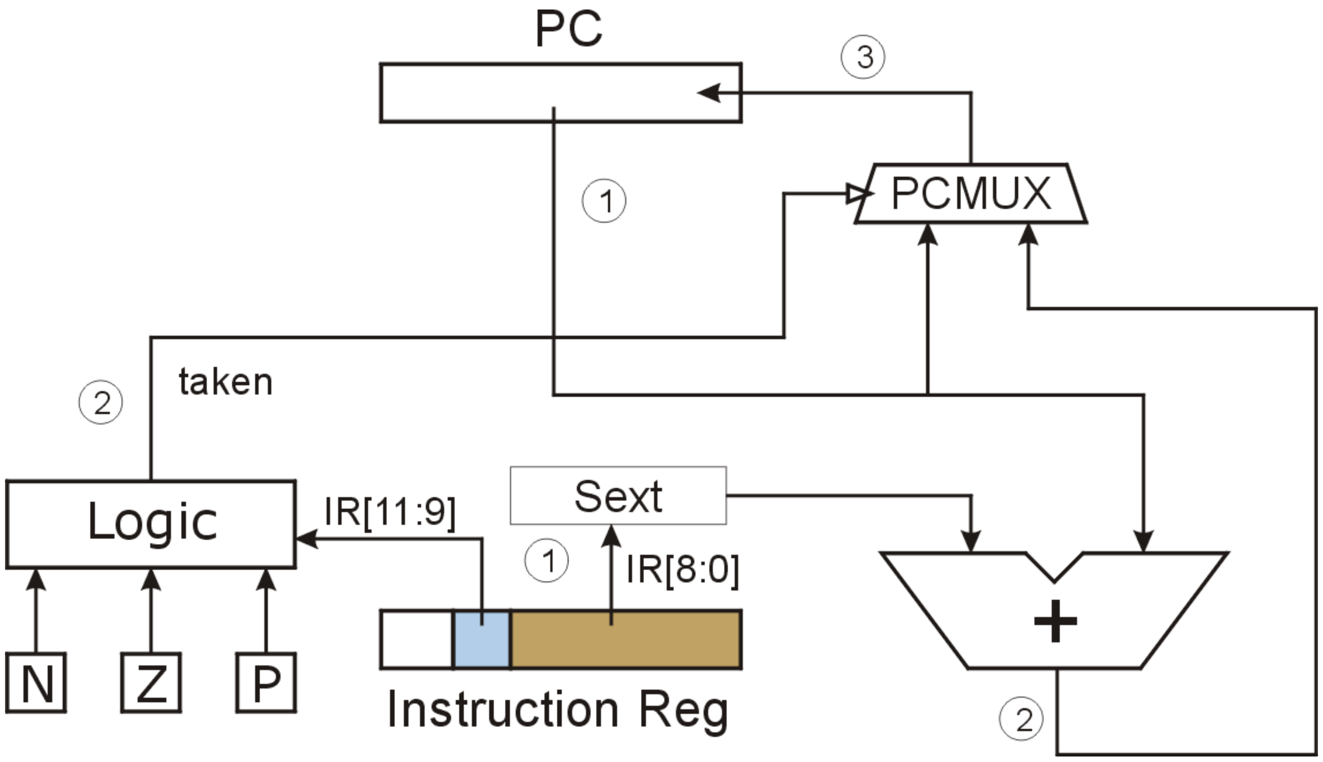
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LEA	1	1	1	0	Dst			PCoffset9								



Assembly Ex:
LEA R1, Lab1

BR (PC-Relative)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR	0	0	0	0	n	z	p	PCoffset9								



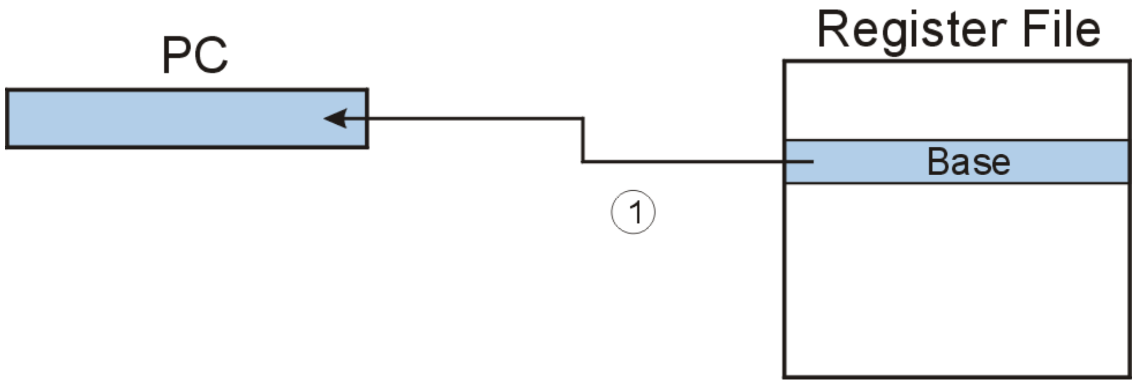
What happens if bits [11:9] are all zero? All one?

JMP (Register)

Jump is an unconditional branch -- *always* taken.

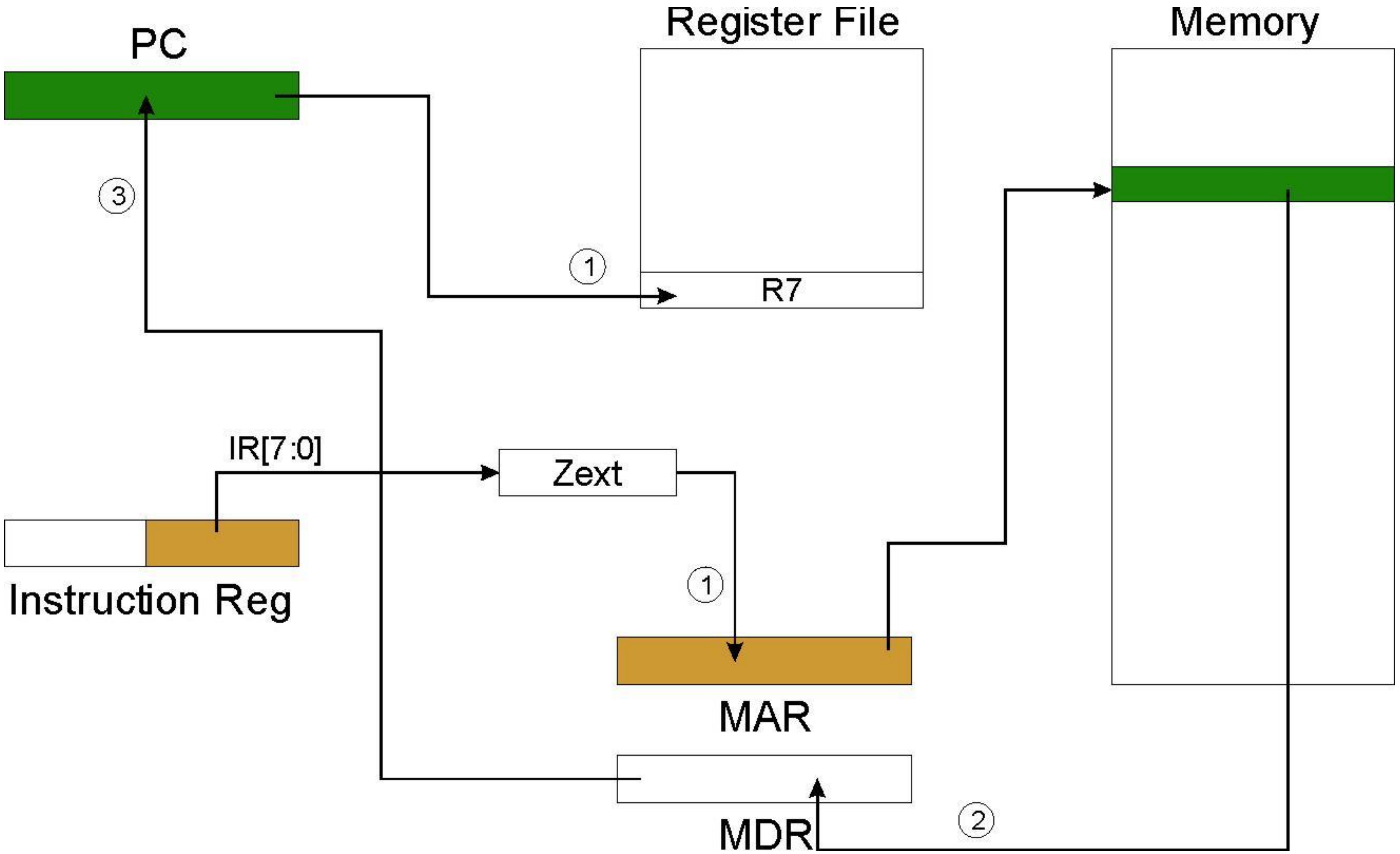
- **Target address is the contents of a register.**
- **Allows any target address.**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JMP	1	1	0	0	0	0	0	Base			0	0	0	0	0	0



TRAP

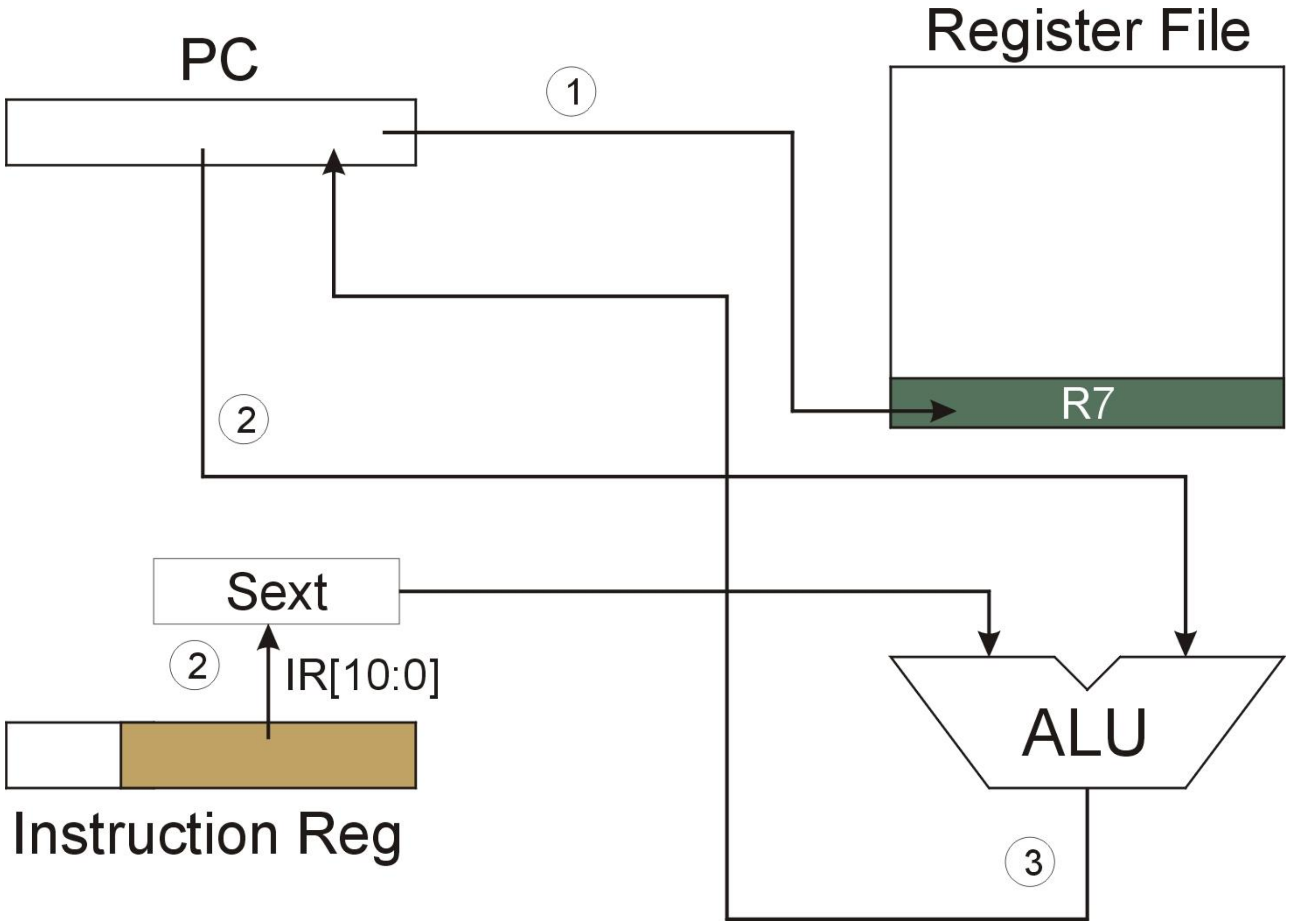
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRAP	1	1	1	1	0	0	0	0	trapvect8							



NOTE: PC has already been incremented
during instruction fetch stage.

JSR

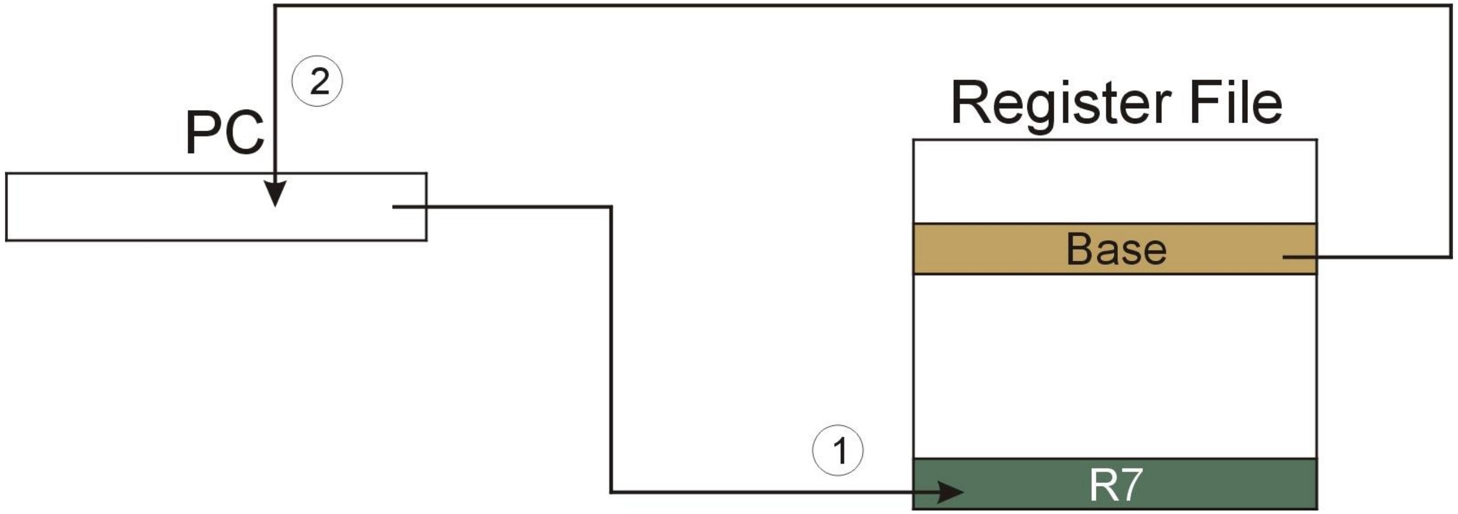
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JSR	0	1	0	0	1	PCoffset11										



NOTE: PC has already been incremented during instruction fetch stage.

JSRR

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JSRR	0	1	0	0	0	0	0	Base			0	0	0	0	0	0



NOTE: PC has already been incremented
during instruction fetch stage.