LC-3 ISA Organized Version (and how to recite them all)

By TA Ziyuan Chen on April 28, 2022 – It helps boost your speed in the exams!

Opcode ↓ + →	00	01	10	11
00	BR	ADD	LD	ST
01	JSR	AND	LDR	STR
10	RTI	NOT	LDI	STA
11	JMP	–	LEA	TRAP

General Rules:

1. The opcode's last two digits decide its *general function*, and the first two digits decide its *specific method*.
2. **Register-based (0)** and **PC-based / *imm*ediate-number-based (1)** address modes appear in pairs.
3. Memory R/W modes: EA/Effective Address (no memory access), Default (once), I/Indirect (twice).
4. The SEXT (sign extension) function is assumed for all offsets and is omitted in the expressions.

	Method	*Dest*	*Data*		
ADD	0001	DR	SR1	000	SR2
ADD	0001	DR	SR1	1	imm5
AND	0101	DR	SR1	000	SR2
AND	0101	DR	SR1	1	imm5
NOT	1001	DR	SR	111111	

01 – Operations: write *Data* to *Dest* with *Method*

`ADD DR, SR1, SR2`	DR ← SR1 + SR2, **Setcc**
`ADD DR, SR1, imm5`	DR ← SR1 + imm5, **Setcc**
`AND DR, SR1, SR2`	DR ← SR1 & SR2, **Setcc**
`AND DR, SR1, imm5`	DR ← SR1 & imm5, **Setcc**
`NOT DR, SR`	DR ← ~ SR, **Setcc**

	Method	*Dest*	*Src*	
LEA	1110	DR	PCoffset9	
LD	0010	DR	PCoffset9	
LDI	1010	DR	PCoffset9	
LDR	0110	DR	BaseR	offset6

10 – Memory Read: load *Src* to *Dest* with *Method*

`LEA DR, PCoffset9`	DR ← PC + PCoffset9, **Setcc**
`LD DR, PCoffset9`	DR ← M[PC + PCoffset9], **Setcc**
`LDI DR, PCoffset9`	DR ← M[M[PC + PCoffset9]], **Setcc**
`LDR DR, BaseR, offset6`	DR ← M[BaseR + offset6], **Setcc**

	Method	*Src*	*Dest*	
ST	0011	SR	PCoffset9	
STI	1011	SR	PCoffset9	
STR	0111	SR	BaseR	offset6

11 – Memory Write: store *Src* to *Dest* with *Method*

`ST DR, PCoffset9`	M[PC + PCoffset9] ← SR
`STI DR, PCoffset9`	M[M[PC + PCoffset9]] ← SR
`STR DR, BaseR, offset6`	M[BaseR + offset6] ← SR

	Function	*Args*			
BR	0000	n	z	p	PCoffset9
JMP*	1100	000		BaseR	000000
JSR	0100	1	PCoffset11		
JSRR	0100	000		BaseR	000000
RTI	1000	0000 0000 0000			
TRAP	1111	0000	trapvect8		

* Also known as RET if BaseR = 111.

00 – Control Flow: perform *Function* with *Args*

`BR{nzp} PCoffset9`	PC ← PC + PCoffset9 (if `Nn+Zz+Pp`)
`JMP BaseR`	PC ← BaseR
`JSR PCoffset11`	R7 ← PC, PC ← PC + PCoffset11
`JSRR BaseR`	R7 ← PC, PC ← BaseR
`RTI`	*No need to understand for now.*
`TRAP trapvect8`	R7 ← PC, PC ← M[trapvect8]