

LC-3 Instruction Set (Entered by Mark D. Hill on 03/14/2007; last update 03/15/2007)

PC': incremented PC. setcc(): set condition codes N, Z, and P. mem[A]:memory contents at address A.
 SEXT(immediate): sign-extend immediate to 16 bits. ZEXT(immediate): zero-extend immediate to 16 bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	0	1		DR		SR1		0	0	0		SR2			ADD DR, SR1, SR2 ; Addition
DR ← SR1 + SR2 also setcc()																
0	0	0	1		DR		SR1		1				imm5			ADD DR, SR1, imm5 ; Addition with Immediate
DR ← SR1 + SEXT(imm5) also setcc()																
0	1	0	1		DR		SR1		0	0	0		SR2			AND DR, SR1, SR2 ; Bit-wise AND
DR ← SR1 AND SR2 also setcc()																
0	1	0	1		DR		SR1		1				imm5			AND DR, SR1, imm5 ; Bit-wise AND with Immediate
DR ← SR1 AND SEXT(imm5) also setcc()																
0	0	0	0		n	z	p									BRx, label (where x = {n,z,p,zp,np,nz,nzp}) ; Branch
GO ← ((n and N) OR (z AND Z) OR (p AND P)) if (GO is true) then PC ← PC' + SEXT(PCOffset9)																
1	1	0	0		0	0	0		BaseR		0	0	0	0	0	JMP BaseR ; Jump
PC ← BaseR																
0	1	0	0		1											JSR label ; Jump to Subroutine
R7 ← PC', PC ← PC' + SEXT(PCOffset11)																
0	1	0	0		0	0	0		BaseR		0	0	0	0	0	JSRR BaseR ; Jump to Subroutine in Register
temp ← PC', PC ← BaseR, R7 ← temp																
0	0	1	0		DR											LD DR, label ; Load PC-Relative
DR ← mem[PC' + SEXT(PCOffset9)] also setcc()																
1	0	1	0		DR											LDI DR, label ; Load Indirect
DR ← mem[mem[PC' + SEXT(PCOffset9)]] also setcc()																
0	1	1	0		DR		BaseR									LDR DR, BaseR, offset6 ; Load Base+Offset
DR ← mem[BaseR + SEXT(offset6)] also setcc()																
1	1	1	0		DR											LEA, DR, label ; Load Effective Address
DR ← PC' + SEXT(PCOffset9) also setcc()																
1	0	0	1		DR		SR		1	1	1	1	1	1	1	NOT DR, SR ; Bit-wise Complement
DR ← NOT(SR) also setcc()																
1	1	0	0		0	0	0		1	1	1		0	0	0	RET ; Return from Subroutine
PC ← R7																
1	0	0	0		0	0	0		0	0	0	0	0	0	0	RTI ; Return from Interrupt
See textbook (2 nd Ed. page 537).																
0	0	1	1		SR											ST SR, label ; Store PC-Relative
mem[PC' + SEXT(PCOffset9)] ← SR																
1	0	1	1		SR											STI, SR, label ; Store Indirect
mem[mem[PC' + SEXT(PCOffset9)]] ← SR																
0	1	1	1		SR		BaseR									STR SR, BaseR, offset6 ; Store Base+Offset
mem[BaseR + SEXT(offset6)] ← SR																
1	1	1	1		0	0	0									TRAP ; System Call
R7 ← PC', PC ← mem[ZEXT(trapvect8)]																
1	1	0	1													; Unused Opcode
Initiate illegal opcode exception																

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