LC4 Instruction Set Reference v. 2015-02		
Mnemonic	**Semantics**	**Encoding**
NOP	PC = PC + 1	0000 000x xxxx xxxx
BRp <Label>	(P) ? PC = PC + 1 + (sext(IMM9) offset to <Label>)	0000 001i iiii iiii
BRz <Label>	(Z) ? PC = PC + 1 + (sext(IMM9) offset to <Label>)	0000 010i iiii iiii
BRzp <Label>	(Z|P) ? PC = PC + 1 + (sext(IMM9) offset to <Label>)	0000 011i iiii iiii
BRn <Label>	(N) ? PC = PC + 1 + (sext(IMM9) offset to <Label>)	0000 100i iiii iiii
BRnp <Label>	(N | P) ? PC = PC + 1 + (sext(IMM9) offset to <Label>)	0000 101i iiii iiii
BRnz <Label>	(N|Z) ? PC = PC + 1 + (sext(IMM9) offset to <Label>)	0000 110i iiii iiii
BRnzp <Label>	(N|Z|P) ? PC = PC + 1 + (sext(IMM9) offset to <Label>)	0000 111i iiii iiii
ADD Rd Rs Rt	Rd = Rs + Rt	0001 ddds ss00 0ttt
MUL Rd Rs Rt	Rd = Rs * Rt	0001 ddds ss00 1ttt
SUB Rd Rs Rt	Rd = Rs - Rt	0001 ddds ss01 0ttt
DIV Rd Rs Rt	Rd = Rs / Rt	0001 ddds ss01 1ttt
ADD Rd Rs IMM5	Rd = Rs + sext(IMM5)	0001 ddds ss1i iiii
CMP Rs Rt	NZP = sign(Rs - Rt) [1]	0010 sss0 0xxx xttt
CMPU Rs Rt	NZP = sign(uRs - uRt)	0010 sss0 1xxx xttt
CMPI Rs IMM7	NZP = sign(Rs - IMM7)	0010 sss1 0iii iiii
CMPIU Rs UIMM7	NZP = sign(uRs - UIMM7)	0010 sss1 1uuu uuuu
JSRR Rs	R7 = PC + 1; PC = Rs	0100 0xxs ssxx xxxx
JSR <Label>	R7 = PC + 1; PC = (PC & 0x8000) | ((IMM11 offset to <Label>) << 4)	0100 1iii iiii iiii
AND Rd Rs Rt	Rd = Rs & Rt	0101 ddds ss00 0ttt
NOT Rd Rs	Rd = ~Rs	0101 ddds ss00 1xxx
OR Rd Rs Rt	Rd = Rs | Rt	0101 ddds ss01 0ttt
XOR Rd Rs Rt	Rd = Rs ∧ Rt	0101 ddds ss01 1ttt
AND Rd Rs IMM5	Rd = Rs & sext(IMM5)	0101 ddds ss1i iiii
LDR Rd Rs IMM6	Rd = dmem[Rs + sext(IMM6)]	0110 ddds ssii iiii
STR Rt Rs IMM6	dmem[Rs + sext(IMM6)] = Rt	0111 ttts ssii iiii
RTI	PC = R7; PSR [15] = 0	1000 xxxx xxxx xxxx
CONST Rd IMM9	Rd = sext(IMM9)	1001 ddd i iiii iiii
SLL Rd Rs UIMM4	Rd = Rs << UIMM4	1010 ddds ss00 uuuu
SRA Rd Rs UIMM4	Rd = Rs >>> UIMM4	1010 ddds ss01 uuuu
SRL Rd Rs UIMM4	Rd = Rs >> UIMM4	1010 ddds ss10 uuuu
MOD Rd Rs Rt	Rd = Rs % Rt	1010 ddds ss11 xttt
JMPR Rs	PC = Rs	1100 0xxs ssxx xxxx
JMP <Label>	PC = PC + 1 + (sext(IMM11) offset to <Label>)	1100 1iii iiii iiii
HICONST Rd UIMM8	Rd = (Rd & 0xFF) | (UIMM8 << 8) [2]	1101 dddx uuuu uuuu
TRAP UIMM8	R7 = PC + 1; PC = (0x8000 | UIMM8); PSR [15] = 1	1111 xxxx uuuu uuuu
Pseudo-Instructions		
RET	Return to R7	JMPR R7
LEA Rd <Label>	Store address of <Label> in Rd	CONST/HICONST
LC Rd <Label>	Store value of constant <Label> in Rd	CONST/HICONST
Assembler Directives		
.CODE	Current memory section contains instruction code	
.DATA	Current memory section contains data values	
.ADDR UIMM16	Set current memory address value to UIMM16	
.FALIGN	Pad current memory address to next multiple of 16	
.FILL IMM16	Current memory address's value = IMM16	
.STRINGZ "String"	Expands to a .FILL for each character in "String"	
.BLKW UIMM16	Reserve *UIMM16* words of memory from the current address	
<Label> .CONST IMM16	Associate *<Label>* with IMM16	
<Label> .UCONST UIMM16	Associate *<Label>* with UIMM16	

0101: opcode or sub-opcode ddd: destination register sss: source register 1 ttt: source register 2
iii: signed immediate value uuu: unsigned immediate value xxx: "don't care" value

[1] sign(Rs- Rt) results in one of three values: +1, 0, or -1, which set the appropriate bit in the NZP register. sign(uRs- uRt) indicates that Rs and Rt are treated as unsigned values.
[2] In this case the source and destination register are one and the same as HICONST reads and modifies the same register.
[3] The NZP register is updated on any instruction that writes to a register, and on CMPx instructions.