Appendix A

The LC-2 ISA

A.1 Overview

The Instruction Set Architecture (ISA) of the LC-2 is defined as follows:

- **Memory address space** 16 bits, corresponding to 2^{16} locations, each consisting of one word (16 bits). Addresses are numbered from 0 (i.e, x0000) to 65,535 (i.e., xFFFF). Addresses are used to identify memory locations and memory-mapped I/O device registers. For convenience, these locations are partitioned into 2^7 pages of 2^9 words each.
- **General purpose registers** Eight 16-bit registers, numbered from 000 to 111.
- **Program counter** A 16-bit register.
- **Bit numbering** Bits of all quantities are numbered, from right to left, starting with bit 0.
- **Instructions** Instructions are 16 bits wide. Bits [15:12] specify the opcode (operation to be performed), bits [11:0] provide further information that is needed to execute the instruction. Section A.3 provides further information on each of the 16 instructions.
- **Condition codes** The load instructions (LD, LDI, LDR, LEA) and the operate instructions (ADD, AND, and NOT) set the three condition codes, depending on whether the result is negative (N = 1, Z = 0, P = 0), zero (N = 0, Z = 1, P = 0), or positive (N = 0, Z = 0, P = 1).
- **Memory mapped I/O** Input and output are handled by standard load/store instructions using memory addresses to designate each I/O device register.

Table A.1 lists each of the relevant device registers, along with the memory address it has been assigned in the LC-2.

Location	I/O Register Name	I/O Register Function
xF3FC	CRT status register	Also known as CRTSR. The ready bit (bit [15]) indicates if the video device is ready to receive another character to print on the screen.
xF3FF	CRT data register	Also known as CRTDR. A character written in the low byte of this register will be displayed on the screen.
xF400	Keyboard status register	Also known as KBSR. The ready bit (bit [15]) indicates if the keyboard has received a new character.
xF401	Keyboard data register	Also known as KBDR. Bits [7:0] contain the last character typed on the keyboard.
xF402	Machine control register	Also known as MCR. Bit [15] is the clock enable bit. When cleared, instruction processing stops.

Table A.1: Device register assignments

A.2 Notation

The notation in Table A.2 will be helpful in understanding the descriptions of the LC-2 instructions (Section A.3).

A.3 The Instruction Set

The 16 LC-2 instructions are summarized in Figure A.1 (page 432). On the following pages, the instructions are described in greater detail. For each instruction, we show the assembly language representation, the actual format of the 16-bit instruction, the operation of the instruction, an English-language description of the operation, and one or more examples of the instruction.

Notation	Meaning
xNumber	The number in hexadecimal notation.
#Number	The number in decimal notation.
A[l:r]	The *field* delimited by bit[l] on the left and bit[r] on the right, from the datum A. For example, if PC contains 0011001100111111, then PC[15:9] is 0011001. PC[2:2] is 1. If l and r are the same bit number, the notation is usually abbreviated PC[2].
A @ B	Concatenation of A and B. For example, if A is 0011 001 and B is 1 1100 l100, A @ B = 0011 0011 1100 1100.
BaseR	Base Register; one of R0..R7, used in conjunction with a six-bit offset to compute Base+offset addresses.
page	The set of 2^9 consecutive memory locations whose addresses share the same high seven address bits.
DR	Destination Register; one of R0..R7, which specifies where the result of an instruction should be written.
imm5	A five-bit immediate value; bits [4:0] of an instruction, when used as a literal (immediate) value. Taken as a 5-bit, 2's complement integer, it is sign-extended to 16 bits before it is used. Range: $-16..15$.
index6	Six-bit immediate value; bits [5:0] of an instruction, when used in a Base+offset instruction. Taken as a six-bit unsigned integer, it is zero-extended to 16 bits before it is used. Range: 0..63.
LABEL	An assembler construct that identifies a location symbolically (i.e., by means of a name, rather than its 16-bit address).
L	Link bit; differentiates JSR from JMP and JSRR from JMPR instructions. If L = 1 (JSR, JSRR), the value of the PC will be saved in R7. If L = 0, the PC is *not* saved in R7.
mem[address]	Denotes the contents of memory at the given address.
PC	Program Counter; 16-bit, processor-internal register which contains the memory address of the *next* instruction to be fetched. For example, during execution of the instruction at address A, the PC contains address A+1.
pgoffset9	Nine bits that differentiate the 2^9 locations on a page. PC[15:9] is concatenated with pgoffset9 to form a 16-bit memory address. Range 0..511.
setcc(X)	Indicates that condition codes N, Z, and P are set based on the value of X. If X is negative, N = 1, Z = 0, P = 0. If X is zero, N = 0, Z = 1, P = 0. If X is positive, N = 0, Z = 0, P = 1.
SEXT(A)	Sign-extend A. The most significant bit of A is replicated as many times as necessary to extend A to 16 bits. For example, if A = 110000, then SEXT(A) = 1111 1111 1111 0000.
SR, SR1, SR2	Source Register; one of R0..R7 which specifies from where an instruction operand is obtained.
trapvect8	Eight-bit trap number used in the TRAP instruction. Range 0–255.
ZEXT(A)	Zero-extend A. Zeroes are appended to the left-most bit of A to extend it to 16 bits. For example, if A = 110000, then ZEXT(A) = 0000 0000 0011 0000.

Table A.2: Notational Conventions

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD^{+}	0001				DR			SR1			0	00		SR2		
ADD^{+}	0001				DR			SR1			1	imm5				
AND^{+}	0101				DR			SR1			0	00		SR2		
AND^{+}	0101				DR			SR1			1	imm5				
BR	0000				n	z	p	pgoffset9								
JSR	0100				L	00		pgoffset9								
JSRR	1100				L	00		BaseR			index6					
LD^{+}	0010				DR			pgoffset9								
LDI^{+}	1010				DR			pgoffset9								
LDR^{+}	0110				DR			BaseR			index6					
LEA^{+}	1110				DR			pgoffset9								
NOT^{+}	1001				DR			SR			111111					
RET	1101				000000000000											
RTI*	1000				000000000000											
ST	0011				SR			pgoffset9								
STI	1011				SR			pgoffset9								
STR	0111				SR			BaseR			index6					
TRAP	1111				0000				trapvect8							

Figure A.1: Formats of the 16 LC-2 instructions. NOTE: + indicates instructions that modify condition codes; $*$ indicates that meaning and use of RTI is beyond the scope of this book.

ADD Addition

Assembler Formats

ADD DR, SR1, SR2
ADD DR, SR1, imm5

Encodings

15–12	11–9	8–6	5	4–3	2–0
0001	DR	SR1	0	00	SR2

15–12	11–9	8–6	5	4–0
0001	DR	SR1	1	imm5

Operation

if (bit[5] == 0)
  DR = SR1 + SR2;
else
  DR = SR1 + SEXT(imm5);
setcc(DR);

Description

If bit [5] is 0, the second-source operand is obtained from SR2. If bit [5] is 1, the second-source operand is obtained by sign-extending the imm5 field to 16 bits. In both cases, the second source operand is added to the contents of SR1, and the result stored in DR. The condition codes are set, based on whether the result is negative, zero, or positive.

Examples

ADD	R2, R3, R4	;	R2 ← R3 + R4
ADD	R2, R3, #7	;	R2 ← R3 + 7

AND Bitwise logical AND

Assembler Formats

AND DR, SR1, SR2
AND DR, SR1, imm5

Encodings

15–12	11–9	8–6	5	4–3	2–0
0101	DR	SR1	0	00	SR2

15–12	11–9	8–6	5	4–0
0101	DR	SR1	1	imm5

Operation

```
if (bit[5] == 0)
  DR = SR1 AND SR2;
else
  DR = SR1 AND SEXT(imm5);
setcc(DR);
```

Description

If bit [5] is 0, the second-source operand is obtained from SR2. If bit [5] is 1, the second-source operand is obtained by sign-extending the imm5 field to 16 bits. In either case, the second-source operand and the contents of SR1 are bitwise ANDed, and the result stored in DR. The condition codes are set, based on whether the binary value produced, taken as a 2's complement integer, is negative, zero, or positive.

Examples

AND	R2, R3, R4	;	R2 ← R3 AND R4
AND	R2, R3, #7	;	R2 ← R3 AND 7

BR Conditional Branch

Assembler Formats

BR	LABEL
BRn	LABEL
BRz	LABEL
BRp	LABEL
BRnz	LABEL
BRnp	LABEL
BRzp	LABEL
BRnzp	LABEL

Encoding

15–12	11	10	9	8–0
0000	n	z	p	pgoffset9

Operation

if ((n AND N) OR (z AND Z) OR (p AND P))
  PC = PC[15:9] @ pgoffset9;

Description

Test the condition codes specified by the state of bits [11:9]. If bit [11] is set, test N; if bit [11] is clear, do not test N. If bit [10] is set, test Z, etc. If any of the condition codes tested is set, branch to the location specified by pgoffset9 on the same page as the branch instruction,

Example

BRzp	LOOP	;	Branch to LOOP if the last result was zero or positive.

JSR — Jump to Subroutine
JMP — Jump

Assembler Formats

JSR LABEL (L = 1)
JMP LABEL (L = 0)

Encoding

15–12	11–8	7–0
1111	0000	trapvect8

Operation

R7 = PC;
PC = mem[ZEXT(trapvect8)];

Description

Load the PC with the contents of the memory location obtained by zero-extending trapvector8 to 16 bits. This is the starting address of the system call specified by trapvector8. Load R7 with the PC, which enables a return to the instruction physically following the TRAP instruction in the original program after the service routine has completed.

Example

```
TRAP    x23    ;   Direct the operating system to execute the
               ;   IN system call.
```

Notes

Memory locations x0020 through x00FF, 192 in all, are available to contain starting addresses for system calls specified by their corresponding trapvectors. This region of memory is called the trap vector table. See Table A.3. Memory locations x0000 through x001F are not part of the trap vector table; therefore, x00 through x1F may not be used as trapvectors.

TRAP Number	**Assembler Name**	**Description**
x20	GETC	Read a single character from the keyboard. The character is not echoed onto the console. Its ASCII code is copied into R0. The high eight bits of R0 are cleared.
x21	OUT	Write a character in R0[7:0] to the console.
x22	PUTS	Write a string pointed to by R0 to the console.
x23	IN	Print a prompt on the screen and read a single character from the keyboard. The character is echoed onto the console, and its ASCII code is copied into R0. The high eight bits of R0 are cleared.
x25	HALT	Halt execution and print a message on the console.

Table A.3: TRAP vector table