SIC Instruction Set
Information and Description

Command	Effect	Description
ADD m	A ← (A) + (m..m+2)	Add the 3-byte value at memory address **m** to the value in register **A**. Note: The **sum** is stored in register **A**.
AND m	A ← (A) & (m..m+2)	Perform a **logical AND** operation on the value in register **A** using the 3-byte value at memory address **m**. Note: The **result** is stored in register **A**.
COMP m	A : (m..m+2)	Compare the 3-byte value at memory address **m** to the value in register **A**. Notes: The **result** is stored in register **SW**. The **result** is **<** if the value in register **A** is < the value at memory address **m**. The **result** is **=** if the value in register **A** is = to the value at memory address **m**. The **result** is **>** if the value in register **A** is > the value at memory address **m**.
DIV m	A ← (A) / (m..m+2)	Divide the value in register **A** by the 3-byte value at memory address **m**. Note: The **quotient** is stored in register **A**.
J m	PC ← m	Program execution jumps to the address stored at memory address **m**. Note: This is an **unconditional** jump.
JEQ m	PC ← m	Program execution jumps to the address stored at memory address **m** if the **result** of the most recent **COMP** command was **=**.
JGT m	PC ← m	Program execution jumps to the address stored at memory address **m** if the **result** of the most recent **COMP** command was **>**.
JLT m	PC ← m	Program execution jumps to the address stored at memory address **m** if the **result** of the most recent **COMP** command was **<**.
JSUB m	L ← (PC); PC ← m	Program execution jumps to the address stored at memory address **m**. Notes: This is an **unconditional** jump. The **current address** in register **PC** is stored in register **L**.
LDA m	A ← (m..m+2)	Load the 3-byte value at memory address **m** into register **A**.
LDCH m	A ← (m)	Load the **1-byte** value at memory address **m** into register **A**.
LDL m	L ← (m..m+2)	Load the 3-byte value at memory address **m** into register **L**.
LDX m	X ← (m..m+2)	Load the 3-byte value at memory address **m** into register **X**.
MUL m	A ← (A) * (m..m+2)	Multiply the 3-byte value at memory address **m** by the value in register **A**. Note: The **product** is stored in register **A**.
OR m	A ← (A) | (m..m+2)	Perform a **logical OR** operation on the value in register **A** using the 3-byte value at memory address **m**. Note: The **result** is stored in register **A**.
RD m	A ← data	Read a **1-byte** value from the device at memory address **m** into register **A**.
RSUB	PC ← (L)	Program execution jumps to the address stored in register L. Note: The **current address** in register **L** is stored in register **PC**.
STA m	m..m+2 ← (A)	Store the value in register **A** into the 3-byte memory address **m**.
STCH m	m ← (A)	Store the **1-byte** value in register **A** in memory address **m**.
STL m	m..m+2 ← (L)	Store the value in register **L** into the 3-byte memory address **m**.
STSW m	m..m+2 ← (SW)	Store the value in register **SW** into the 3-byte memory address **m**.
STX m	m..m+2 ← (X)	Store the value in register **X** into the 3-byte memory address **m**.
SUB m	A ← (A) - (m..m+2)	Subtract the 3-byte value at memory address **m** from the value in register **A**. Note: The **difference** is stored in register **A**.
TD m	Test if device (m) is ready	Tests if the device at memory address **m** is ready for use. Notes: The **result** is stored in register **SW**. The **result** is **<** if the device is ready. The **result** is **>** if the device is not ready.
TIX m	X ← (X) + 1; (X) : (m..m+2)	Adds **1** to the value in register **X** and compares the 3-byte value at memory address **m** to the value in register **X**. Note: The **result** is stored in register **SW**.
WD m	(m) ← (A)	Write a **1-byte** value from register **A** to the device at memory address **m**.

Register Description

Register	Description
A - Accumulator	Stores results of **arithmetic** and **logical** operations. Used for manipulating **characters** in memory and devices. Notes: **Primary** register used by SIC programs. Register can be directly accessed and manipulated.
L - Linkage	Stores PC address when JSUB command performed. Stored address used to return to next instruction after JSUB command when RSUB command performed. Notes: Value of this register can be set using the **LDL** or **JSUB** commands. Value of this register can be read by the **RSUB** or **STL** commands. Register can be directly accessed and manipulated.
X - Index	Stores value used to iterate through a collection of data. Can be used to access memory reserved by RESB and RESW directives. Notes: Value of this register can be set by the **LDX** or **TIX** commands. Value of this register can be read by the **STX** command. Register can be directly accessed and manipulated.
PC - Program Count	Stores the address of the next instruction to perform. Notes: Register normally updated automatically by the processor. Value of this register can be set by the **JSUB** and **RSUB** commands. Register **cannot** be directly accessed or manipulated.
SW - Status Word	Stores several flags related to the status of the program. The **CC** flag stores the result of comparison operations. Notes: Value of this register is set by the **COMP**, **TD** or **TIX** commands. Value of this register is read by the **JEQ**, **JGT** or **JLT** commands. Register **cannot** be directly accessed or manipulated.