SIEMENS

Load and Transfer Instructions

Nguyen Duc Hoang

SIEMENS

Load and Transfer Instructions

- The Load (L) and Transfer (T) instructions enable you to program an interchange of information between input or output modules and memory areas, or between memory areas.
- The CPU executes these instructions in each scan cycle as unconditional instructions, that is, they are not affected by the result of logic operation of a statement.

SIEMENS

Load and Transfer Instructions

L	Load
L STW	Load Status Word into ACCU 1
LAR1 AR2	Load Address Register 1 from Address Register 2
LAR1 <D>	Load Address Register 1 with Double Integer (32-Bit Pointer)
LAR1	Load Address Register 1 from ACCU 1
LAR2 <D>	Load Address Register 2 with Double Integer (32-Bit Pointer)
LAR2	Load Address Register 2 from ACCU 1

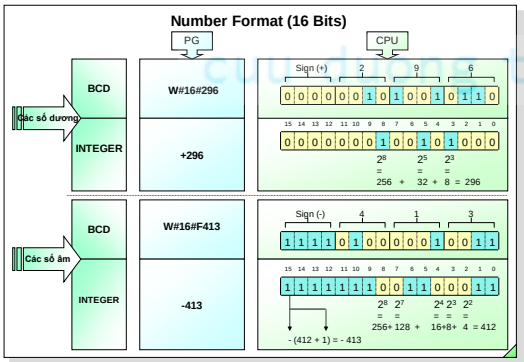
SIEMENS

Load and Transfer Instructions

T	Transfer
T STW	Transfer ACCU 1 into Status Word
TAR1 AR2	Transfer Address Register 1 to Address Register 2
TAR1 <D>	Transfer Address Register 1 to Destination (32-Bit Pointer)
TAR2 <D>	Transfer Address Register 2 to Destination (32-Bit Pointer)
TAR1	Transfer Address Register 1 to ACCU 1
TAR2	Transfer Address Register 2 to ACCU 1
CAR	Exchange Address Register 1 with Address Register 2

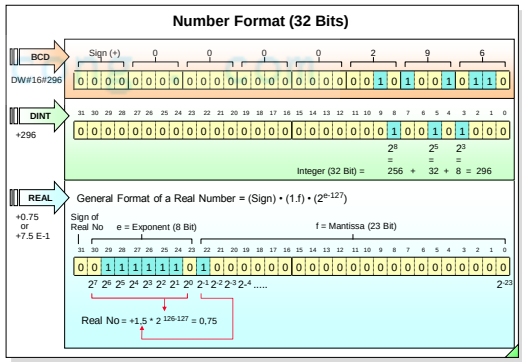
SIEMENS

Number Format (16 Bits)



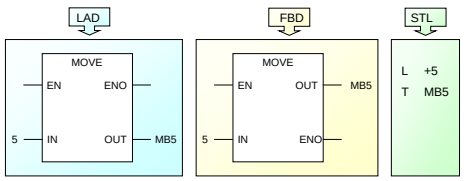
SIEMENS

Number Format (32 Bits)



SIEMENS

Load and Transfer Instructions (1)

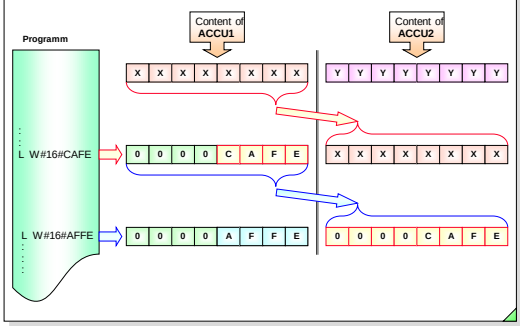


Examples of Load

L +5	//	16-bit constant (Integer)
L L#523123	//	32-bit constant (Double Integer)
L B#16#EF	//	byte in hexadecimal form.
L 2#0010 0110 1110 0011	//	16-bit binary value
L 3.14	//	32-bit constant (Real)

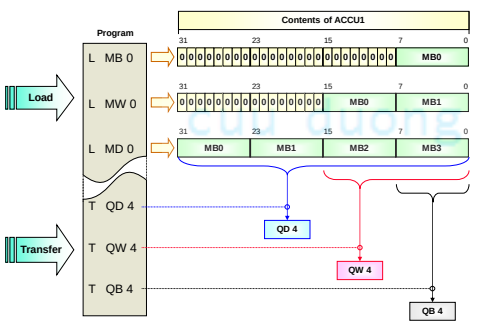
SIEMENS

Load and Transfer Instructions (2)



SIEMENS

Load and Transfer Instructions (3)



SIEMENS

Structure of a Statement

- Statement group 1

Instruction alone

- Statement group 2

Instruction + address

SIEMENS

Addressing

- Immediate Addressing
- Direct Addressing
- Memory Indirect Addressing
- Address Registers
- Area-Internal Register Indirect Addressing
- Area-Crossing Register Indirect Addressing

SIEMENS

Immediate Addressing

Example	Description
SET	Set the RLO to 1.
OW W#16#A320	Or Word.
L 27	Load the integer value 27 into accumulator 1.
L 'ABCD'	Load the ASCII characters ABCD into accumulator 1.
L B#(100,12)	Load the two bytes 100 and 12 into accumulator 1.
L C#0100	Load the BCD value 0000 into accumulator 1.

SIEMENS

Direct Addressing

Example	Description
A I 0.0	Perform an AND logic operation on input bit I 0.0.
S L 20.0	Set the local data bit L 20.0.
= M 115.4	Assign the RLO to memory bit M 115.4
L IB0	Load input byte IB0 into accumulator 1.
L MW64	Load memory word MW64 into accumulator 1.
T DBD12	Transfer the contents from accumulator 1 into data double word DBD12.

SIEMENS

Memory Indirect Addressing

Example	Description
A I [MD2] or A I [anna]	Perform an And logic operation on the input bit whose exact location is in memory double word MD2 or in the location designated by "anna" in the symbol table, as a reference to MD2.
= DIX [DBD2]	Assign the RLO bit to the instance data bit whose exact location is in data double word DBD2.
OPN DB [LW2]	Open the data block whose data block number is located in local word LW2.
O Q [LD3] or O Q [boxcar]	Perform an Or logic operation on the output bit that is located in a local data double word LD3 or in a local TEMP variable designated as "boxcar."

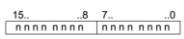
SIEMENS

Pointer Format

- Word Pointer Format
- Double Word Pointer Format

SIEMENS

Word Pointer Format for Memory Indirect Addressing



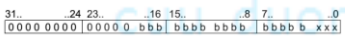
Bits 0 to 15 (nnnn nnnn nnnn nnnn): number (range 0 to 65,535) of a timer (T), counter (C), data block (DB), function (FC), or function block (FB)

STL	Explanation
L +5	Load the value 5 as an integer into accumulator 1.
T MW2	Transfer the contents of accumulator 1 to memory word MW2.
OPN DB[MW2]	Open data block 5.

STL	Explanation
OPN DB10	Open data block DB10.
L +20	Load the value 20 as an integer into accumulator 1.
T DBW10	Transfer the contents of accumulator 1 to data word DBW10
A T[DBW10]	Check the signal state of timer T 20.

SIEMENS

Double Word Pointer Format for Memory Indirect Addressing



Bits 3 to 18 (bbbb bbbb bbbb bbbb): byte number (range 0 to 65,535) of the addressed byte

Bits 0 to 2 (xxx): bit number (range 0 to 7) of the addressed bit

STL	Explanation
L P#8.7	Load 2#0000 0000 0000 0000 0000 0000 0100 0111 (binary value) into accumulator 1.
T MD2	Store the exact location 8.7 in memory double word MD2.
A I [MD2]	The controller checks input bit I 8.7 and assigns its signal
= Q [MD2]	state to output bit Q 8.7.

STL	Explanation
L P#8.0	Load 2#0000 0000 0000 0000 0000 0000 0100 0000 (binary value) into accumulator 1.
T MD2	Store the exact location 8 in memory double word MD2.
L IB [MD2]	The controller loads input byte IB8 and transfers the contents
T MW [MD2]	to memory word MW8. The exact location 8 comes from memory double word MD2.

SIEMENS

Address Registers

- Address registers 1 and 2 (AR1 and AR2) : 32-bit registers that accept an area-internal or area-crossing pointer for commands that use register-indirect addressing.
- Pointers are used in register-indirect addressing:
  - Area-internal: used for area-internal access to bits, bytes, words, and double words in memory areas P, I, Q, M, DBX, DIX, and L
  - Area-crossing: used for area-crossing access to bits, bytes, words, and double words in memory areas P, I, Q, M, DBX, DIX, and L

SIEMENS

Area-Internal Register Indirect Addressing

- two-part address
  - An address identifier (ex: “LD” for “local data double word”)
  - An address register and a pointer to specify byte and bit. The byte and bit indicate an offset, which, when added to the contents of the register, indicate the memory location of the value that the instruction is to process.

SIEMENS

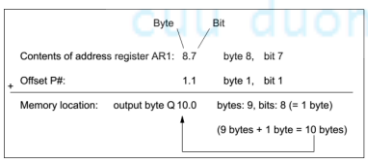
Area-Internal Register Indirect Addressing

Example	Description
A I [AR1, P#4.3]	Perform an And logic operation on the input bit whose memory location is calculated by the contents of address register AR1 plus 4 bytes, plus 3 bits.
= DIX [AR2, P#0.0]	Assign the RLO bit to the instance data bit whose memory location is in address register AR2.
L IB [AR1, P#100.0]	Load the input byte whose memory location is calculated by the contents of address register AR1 plus 100 bytes into accumulator 1.
T LD [AR2, P#56.0]	Transfer the contents of accumulator 1 into local data double word LD whose memory location is calculated by the contents of address register AR2 plus 56 bytes. With reference to addressing local data, please read the Warning below.

SIEMENS

Area-Internal Register Indirect Addressing

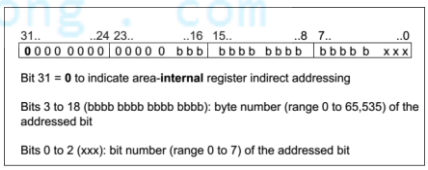
- Calculating the Memory Location of the Address ?
- Ex: = Q [AR1, P#1.1]



SIEMENS

Area-Internal Register Indirect Addressing

- Area-internal register indirect addressing has only one possible pointer format: double word.



SIEMENS

Area-Internal Register Indirect Addressing

STL	Explanation
L P#8.7	Load a double word pointer to bit address location 8.7 into accumulator 1.
LAR1	Store a double word pointer to bit address location 8.7 in address register AR1.
A I [AR1, P#0.0]	The CPU adds the offset (P#0.0) to the contents of address register AR1 (8.7) and uses this address as the location of an And bit logic instruction. The contents of AR1 remain unchanged.
= Q [AR1, P#1.1]	The CPU assigns the result of the And bit logic operation (RLO) to an address (Q 10.0). The CPU calculates this address by adding the contents of address register AR1 (8.7) and the offset (P#1.1).

SIEMENS

Area-Internal Register Indirect Addressing

STL	Explanation
L P#8.0	Load a double word pointer to bit address location 8.0 into accumulator 1.
LAR2	Store a double word pointer to bit address location 8.0 in address register AR2.
L IB [AR2, P#2.0]	The CPU loads input byte IB10 into accumulator 1.
T MW [AR2, P#200.0]	The CPU transfers the contents of accumulator 1 to memory word MW208. The location 208 comes from 8 (AR2) plus 200 (offset), which is 208.

SIEMENS

Area-Crossing Register Indirect Addressing

- **two-part address**
  - **An address identifier that indicates the size of a data object (ex:"B" for "byte,"). The memory area is indicated in bits 24, 25, and 26 of the address register.**
  - **An address register and a pointer that indicate an offset which, when added to the contents of the address register, indicates the memory location of the value that is to be processed by the instruction. The pointer is indicated by P#byte.bit.**

SIEMENS

Area-Crossing Register Indirect Addressing

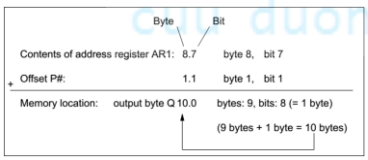
- **Area Identification for Area-Crossing Register Indirect Addressing**

Area Identification (Memory Area)	Binary Contents of Bits 26, 25, and 24
P (I/O, external inputs and outputs)	000
I (process-image input)	001
Q (process-image output)	010
M (bit memory)	011
DBX (data block)	100
DIX (data block)	101
(previous local data, that is, the local data of the previous incompleted block)	111

SIEMENS

Area-Crossing Register Indirect Addressing

- **Calculating the Memory Location of the Address ?**
- **Ex: = [AR1, P#1.1]**



SIEMENS

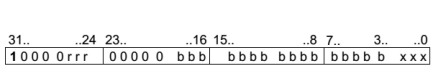
Area-Crossing Register Indirect Addressing

Example	Description
A [AR1, P#4.3]	Perform an And logic operation on the bit whose memory location is calculated by the contents of address register AR1, plus 4 bytes plus 3 bits. The memory area of the bit is indicated in bits 24, 25, and 26 of address register AR1.
= [AR2, P#0.0]	Assign the RLO bit to the bit whose memory location is in address register AR2. The memory area of the bit is indicated in bits 24, 25, and 26 of address register AR2.
L B [AR1, P#100.0]	Load into accumulator 1 the byte whose memory location is calculated in address register AR1 plus 100 bytes. The memory area of the byte is indicated in bits 24, 25, and 26 of address register AR1.
T D [AR2, P#56.0]	Transfer the contents of accumulator 1 into the double word whose memory location is calculated by the contents of address register AR2 plus 56 bytes. The memory area of the double word is indicated in bits 24, 25, and 26 of address register AR2.

SIEMENS

Area-Crossing Register Indirect Addressing

- **Area-crossing register indirect addressing has only one possible pointer format: double word.**



Bit 31 = **1** to indicate area-**crossing** register indirect addressing

Bits 24, 25, and 26 (rrr): area identification (memory area, see Table 3-6)

Bits 3 to 18 (bbbb bbbb bbbb bbbb): byte number (range 0 to 65,535) of the addressed bit

Bits 0 to 2 (xxx): bit number (range 0 to 7) of the addressed bit

SIEMENS

Area-Crossing Register Indirect Addressing

STL	Explanation
L P#I 8.7	Load a double word pointer to bit address location I 8.7 into accumulator 1.
LAR1	Store a double word pointer to bit address location I 8.7 in address register AR1.
L P#Q 8.7	Load a double word pointer to bit address location Q 8.7 into accumulator 1.
LAR2	Store a double word pointer to bit address location Q 8.7 in address register AR2.
A [AR1, P#0.0]	The CPU adds the contents of address register AR1 (P#I 8.7) and the offset (P#0.0) and uses the address pointed to by the result (I 8.7) as the address of an And bit logic instruction. The contents of AR1 remain unchanged.
= [AR2, P#1.1]	The CPU assigns the result of the And bit logic operation (RLO) to an address (Q 10.0). The CPU calculates this address by adding the contents of address register AR2 (P#Q 8.7) and the offset (P#1.1) and dereferencing the pointer. The contents of AR2 remain unchanged.

SIEMENS

Area-Crossing Register Indirect Addressing

STL	Explanation
L P#I 8.0	Load a double word pointer to bit address location I 8.0 into accumulator 1.
LAR2	Store a double word pointer to bit address location I 8.0 in address register AR2.
L P#M 8.0	Load a double word pointer to bit address location M 8.0 into accumulator 1.
LAR1	Store a double word pointer to bit address location M 8.0 in address register AR1.
L B [AR2, P#2.0]	The CPU loads input byte IB10.
T W [AR1, P#200.0]	The CPU transfers the contents to memory word MW208.
	Input byte 10 comes from 8 (AR2) plus 2 (offset). Memory word 208 comes from 8 (AR1) plus 200 (offset), which is 208.

SIEMENS

Examples of how to calculate the pointer

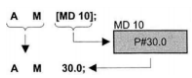
- LAR1 P#8.2
- A I [AR1,P#10.2]
- Result: Input 18.4 is addressed

- L MD 0 Random pointer, e.g. P#10.5
- LAR1
- A I [AR1,P#10.7]
- Result: Input 21.4 is addressed

SIEMENS

Examples of how to calculate the pointer

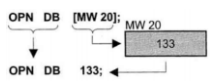
```
L   P#30.0;        //Load pointer into accumulator 1
T   MD 10;         //save in memory doubleword
L   MW [MD 10];    //assign digital addresses
...
A   M [MD 10];     //assign binary addresses
```



SIEMENS

Examples of how to calculate the pointer

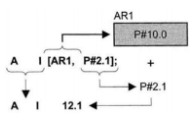
```
L   133;           //Load pointer into accumulator 1
T   MW 20;         //save in memory word
OPN DB [MW 20];    //Open global DB
...
SP  T [MW 20];     //Start timer
```



SIEMENS

Examples of how to calculate the pointer

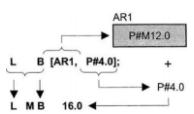
```
LAR1 P#10.0;       //Load pointer into address register 1
...
L    MW [AR1,P#4.0];//assign digital addresses
...
A    I [AR1,P#2.1]; //assign binary addresses
```



SIEMENS

Examples of how to calculate the pointer

```
LAR1 P#M12.0;      //Load pointer into address register 1
...
L    B [AR1,P#4.0]; //assign digital addresses
...
=    [AR1,P#0.7];  //assign binary addresses
```



SIEMENS

Comparison of Indirect Addressing Types

Memory-Indirect		Register-Indirect Area-Internal		Register-Indirect Area-Crossing	
L	P#4.7	LAR1	P#4.7	LAR1	P#Q4.7
T	MD 24				
S	Q [MD 24]	S	Q [AR1,P#0.0]	S	[AR1,P#0.0]

SIEMENS

Examples

- LAR2 P#20.0 ;
- L P#24.0 ;
- LAR1;
- LAR1 MD120 ;
- LAR1 AR2;

- //Load AR2 with P#20. 0
- //Load AR1 with <Accum1>
- //Load AR1 with <MD120>
- //Load AR1 with <AR2>

SIEMENS

Examples

- TAR2 MD140 ;
- TAR1;
- TAR1 AR2;

- //Transfer <AR2 > to MD140
- //Transfer <AR1 > to Accum1
- //Transfer <AR1 > to AR2