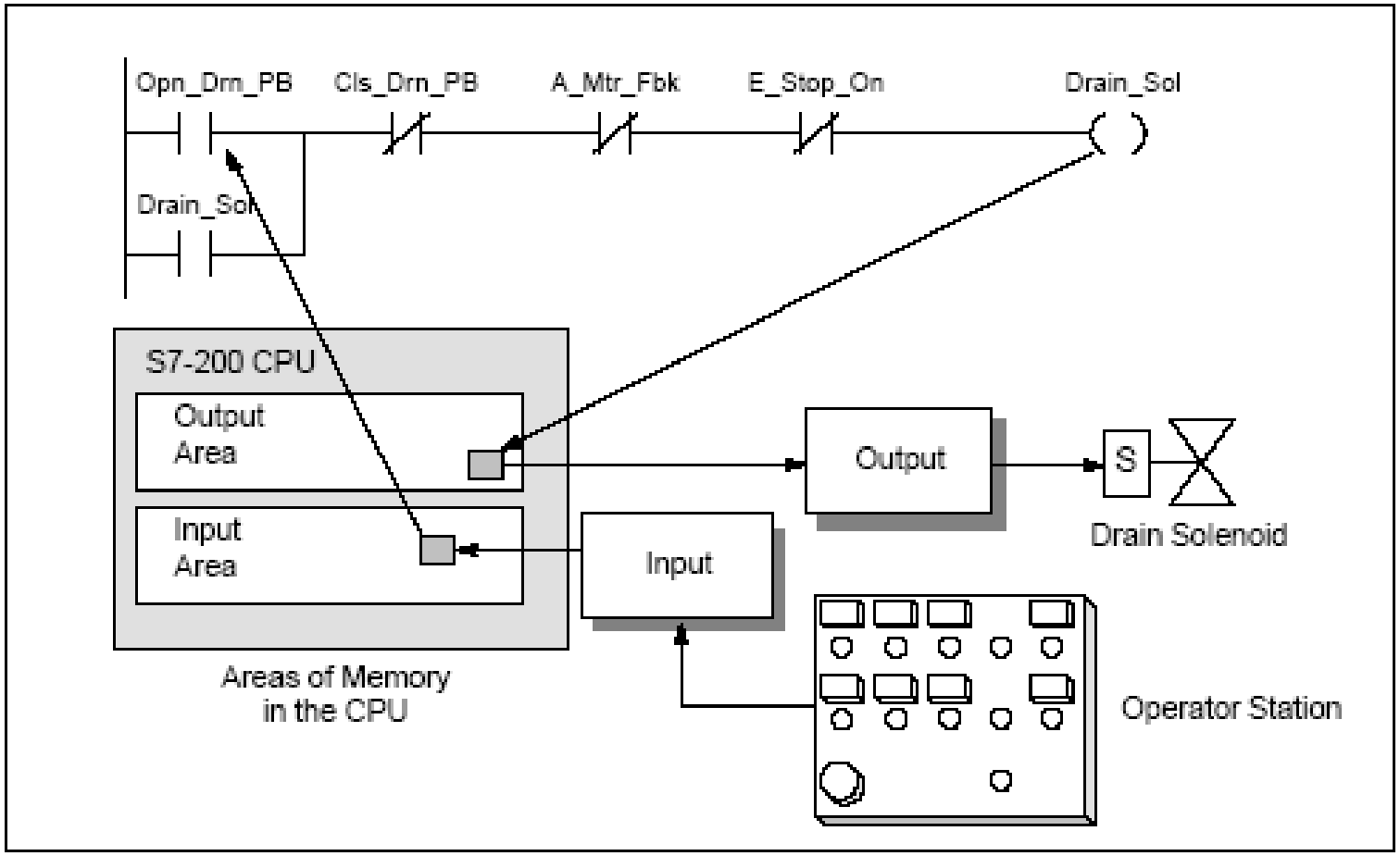
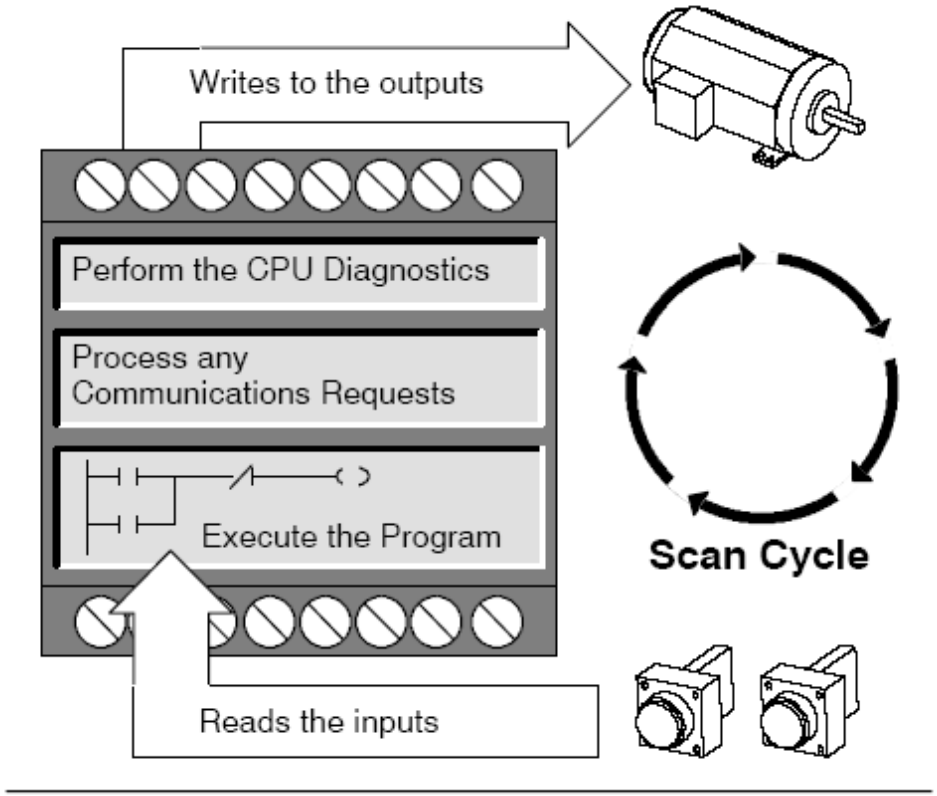
Povezava fizičnih vhodov in izhodov z elementi programa



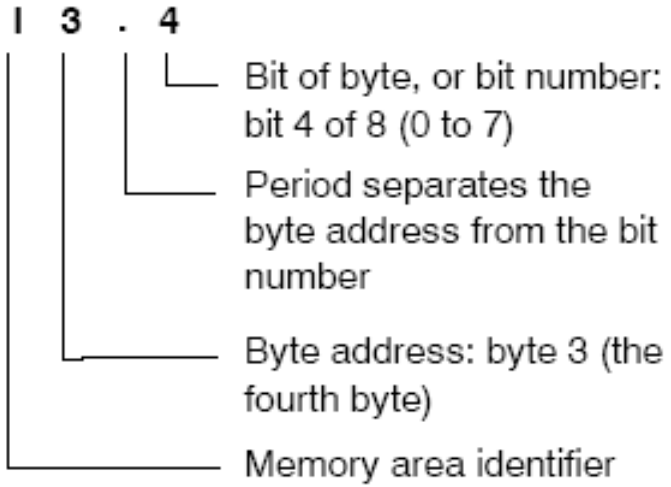
Programski cikel

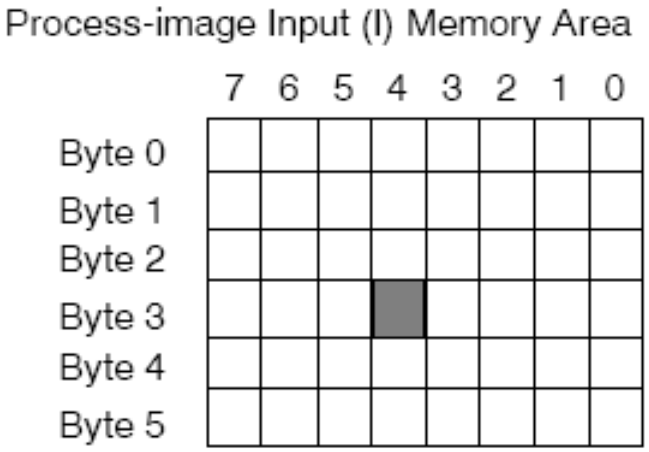
- Reading the inputs: The S7-200 copies the state of the physical inputs to the process-image input register.
- Executing the control logic in the program: The S7-200 executes the instructions of the program and stores the values in the various memory areas.
- Processing any communications requests: The S7-200 performs any tasks required for communications.
- Executing the CPU self-test diagnostics: The S7-200 ensures that the firmware, the program memory, and any expansion modules are working properly.
- Writing to the outputs: The values stored in the process-image output register are written to the physical outputs.



Naslavljanje podatkov

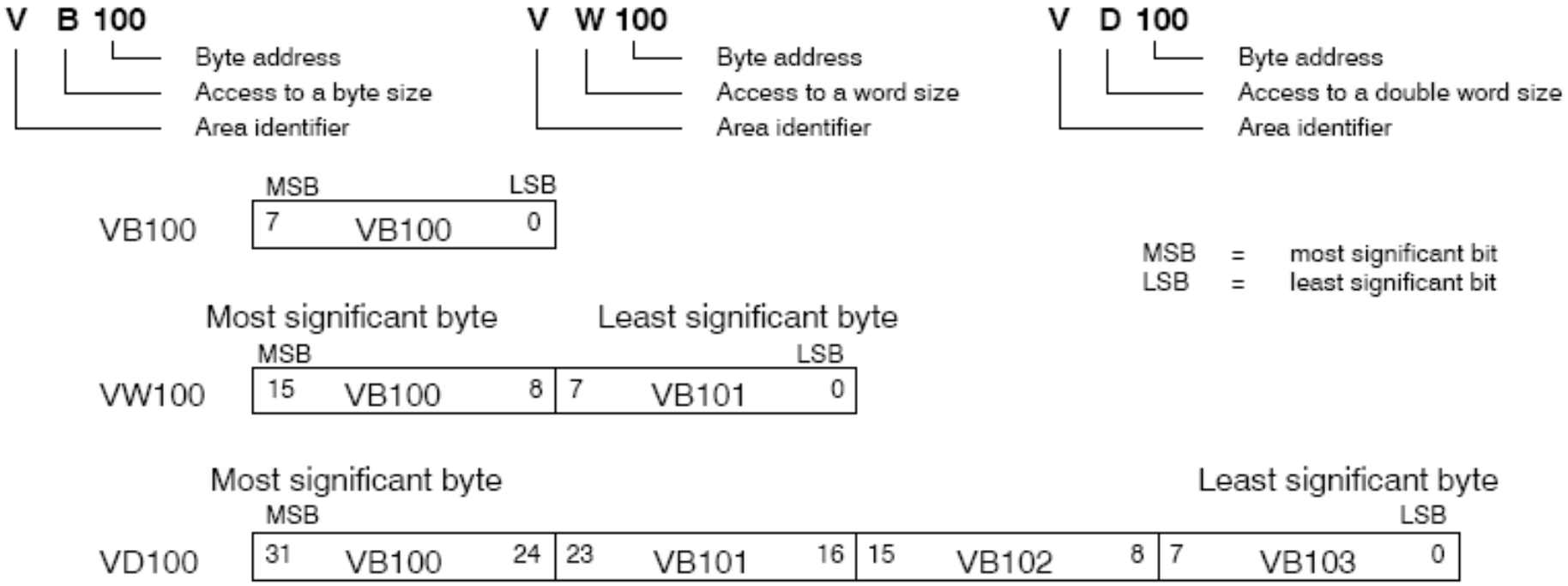
Primer: byte.bit dostopanje do podatka





Naslavljanje podatkov

Primerjava dostopa do istega naslova na način Byte, Word ali Double-Word



Podatkovna področja

Process-Image Input Register: I

The S7-200 samples the physical input points at the beginning of each scan cycle and writes these values to the process-image input register. You can access the process-image input register in bits, bytes, words, or double words:

Bit:	I*[byte address].[bit address]*	I0.1
Byte, Word, or Double Word:	I*[size][starting byte address]*	IB4

Podatkovna področja

Process-Image Output Register: Q

At the end of the scan cycle, the S7-200 copies the values stored in the process-image output register to the physical output points. You can access the process-image output register in bits, bytes, words, or double words:

Bit:	Q*[byte address].[bit address]*	Q1.1
Byte, Word, or Double Word:	Q*[size][starting byte address]*	QB5

Podatkovna področja

Variable Memory Area: V

You can use V memory to store intermediate results of operations being performed by the control logic in your program. You can also use V memory to store other data pertaining to your process or task. You can access the V memory area in bits, bytes, words, or double words:

Bit:	V*[byte address].[bit address]*	V10.2
Byte, Word, or Double Word:	V*[size][starting byte address]*	VW100

Podatkovna področja

Bit Memory Area: M

You can use the bit memory area (M memory) as control relays to store the intermediate status of an operation or other control information. You can access the bit memory area in bits, bytes, words, or double words:

Bit:	M*[byte address].[bit address]*	M26.7
Byte, Word, or Double Word:	M*[size][starting byte address]*	MD20

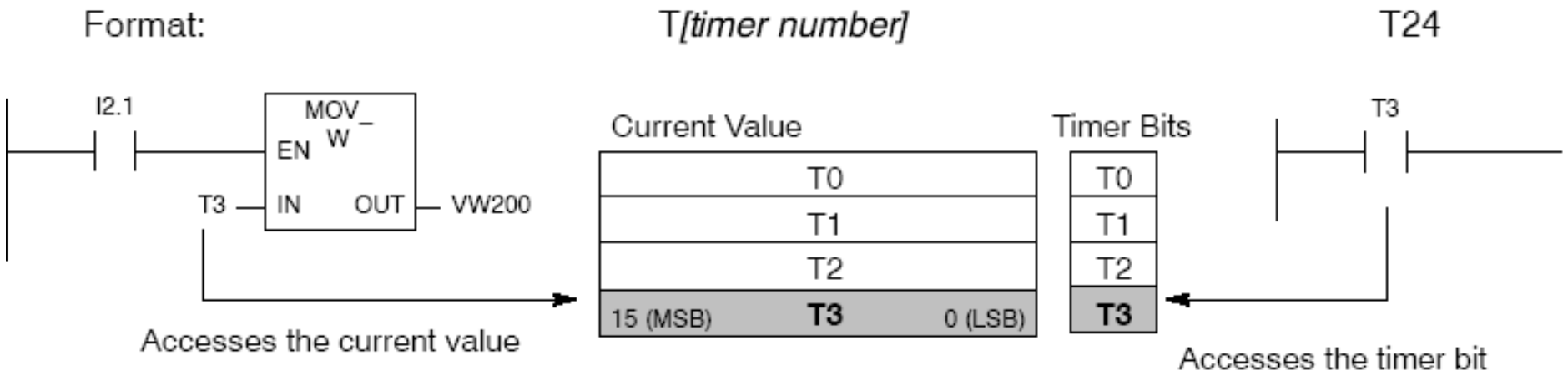
Podatkovna področja

Timer Memory Area: T

The S7-200 provides timers that count increments of time in resolutions (time-base increments) of 1 ms, 10 ms, or 100 ms. Two variables are associated with a timer:

- Current value: this 16-bit signed integer stores the amount of time counted by the timer.
- Timer bit: this bit is set or cleared as a result of comparing the current and the preset value. The preset value is entered as part of the timer instruction.

You access both of these variables by using the timer address (T + timer number). Access to either the timer bit or the current value is dependent on the instruction used: instructions with bit operands access the timer bit, while instructions with word operands access the current value. As shown in Figure 4-5, the Normally Open Contact instruction accesses the timer bit, while the Move Word instruction accesses the current value of the timer.



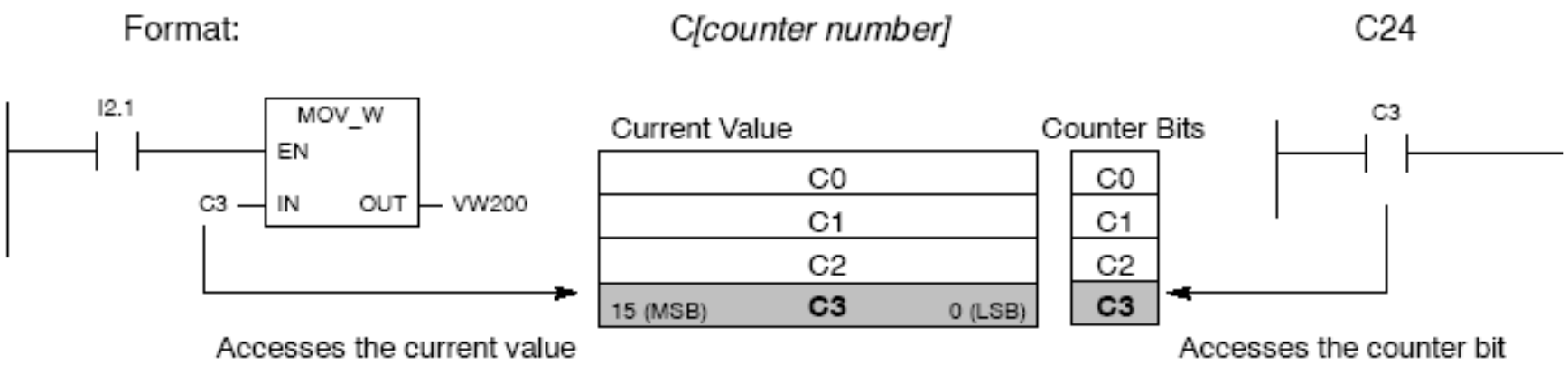
Podatkovna področja

Counter Memory Area: C

The S7-200 provides three types of counters that count each low-to-high transition event on the counter input(s): one type counts up only, one type counts down only, and one type counts both up and down. Two variables are associated with a counter:

- Current value: this 16-bit signed integer stores the accumulated count.
- Counter bit: this bit is set or cleared as a result of comparing the current and the preset value. The preset value is entered as part of the counter instruction.

You access both of these variables by using the counter address (C + counter number). Access to either the counter bit or the current value is dependent on the instruction used: instructions with bit operands access the counter bit, while instructions with word operands access the current value. As shown in Figure 4-6, the Normally Open Contact instruction accesses the counter bit, while the Move Word instruction accesses the current value of the counter.



Podatkovna področja

High-Speed Counters: HC

The high-speed counters count high-speed events independent of the CPU scan. High-speed counters have a signed, 32-bit integer counting value (or current value). To access the count value for the high-speed counter, you specify the address of the high-speed counter, using the memory type (HC) and the counter number (such as HC0). The current value of the high-speed counter is a read-only value and can be addressed only as a double word (32 bits).

Format:	HC*[high-speed counter number]*	HC1

Podatkovna področja

Special Memory: SM

The SM bits provide a means for communicating information between the CPU and your program. You can use these bits to select and control some of the special functions of the S7-200 CPU, such as: a bit that turns on for the first scan cycle, a bit that toggles at a fixed rate, or a bit that shows the status of math or operational instructions. (For more information about the SM bits, see Appendix D.) You can access the SM bits as bits, bytes, words, or double words:

Bit:	SM*[byte address].[bit address]*	SM0.1
Byte, Word, or Double Word:	SM*[size][starting byte address]*	SMB86

Podatkovna področja

SMB0: Status Bits

As described in Table D-1, SMB0 contains eight status bits that are updated by the S7-200 at the end of each scan cycle.

Table D-1 Special Memory Byte SMB0 (SM0.0 to SM0.7)

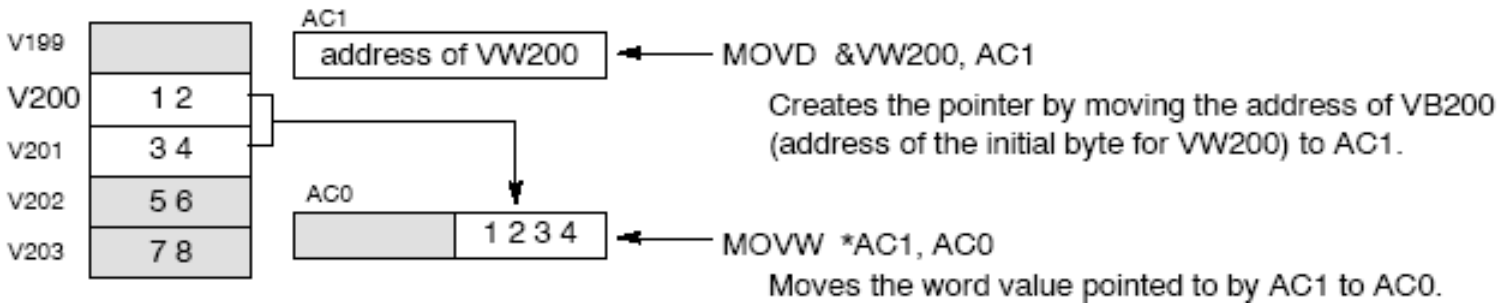
SM Bits	Description (Read Only)
SM0.0	This bit is always on.
SM0.1	This bit is on for the first scan cycle. One use is to call an initialization subroutine.
SM0.2	This bit is turned on for one scan cycle if retentive data was lost. This bit can be used as either an error memory bit or as a mechanism to invoke a special startup sequence.
SM0.3	This bit is turned on for one scan cycle when RUN mode is entered from a power-up condition. This bit can be used to provide machine warm-up time before starting an operation.
SM0.4	This bit provides a clock pulse that is on for 30 seconds and off for 30 seconds, for a duty cycle time of 1 minute. It provides an easy-to-use delay, or a 1-minute clock pulse.
SM0.5	This bit provides a clock pulse that is on for 0.5 seconds and then off for 0.5 seconds, for a duty cycle time of 1 second. It provides an easy-to-use delay or a 1-second clock pulse.
SM0.6	This bit is a scan cycle clock which is on for one scan cycle and then off for the next scan cycle. This bit can be used as a scan counter input.
SM0.7	This bit reflects the position of the Mode switch (off is TERM position, and on is RUN position). If you use this bit to enable Freeport mode when the switch is in the RUN position, normal communications with the programming device can be enabled by switching to the TERM position.

Podatkovna področja

Indirektno naslavljanje

To indirectly access the data in a memory address, you create a pointer to that location by entering an ampersand (&) and the memory location to be addressed. The input operand of the instruction must be preceded with an ampersand (&) to signify that the address of a memory location, instead of its contents, is to be moved into the location identified in the output operand of the instruction (the pointer).

Entering an asterisk (*) in front of an operand for an instruction specifies that the operand is a pointer. As shown in Figure 4-11, entering *AC1 specifies that AC1 is a pointer to the word-length value being referenced by the Move Word (MOVW) instruction. In this example, the values stored in both VB200 and VB201 are moved to accumulator AC0.



Načini programiranja

- LAD (contact plan)
- FBD (functional plan)
- STL (instruction list)
- additional high level languages and graphical sequence languages.

Načini programiranja

STL (Structured Text Language)

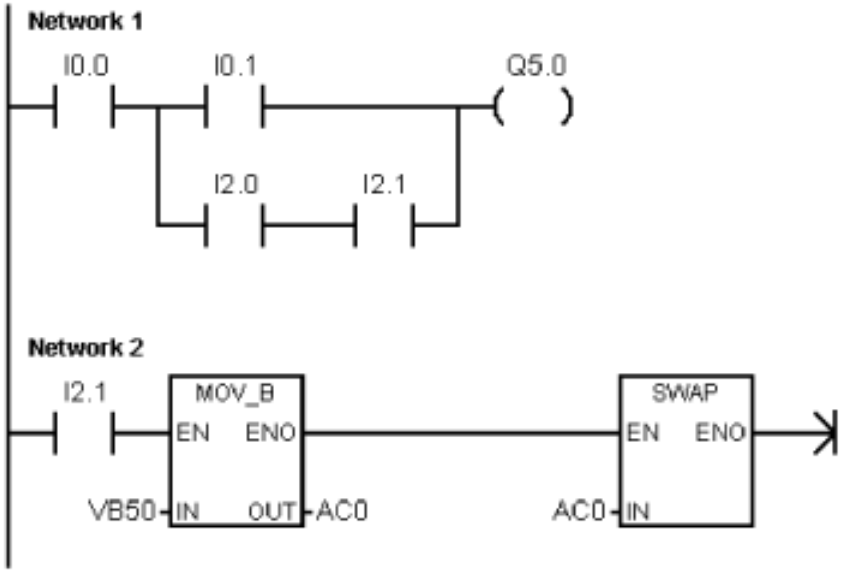
- Za izkušene programerje
- V nekaterih primerih omogoča rešitev problemov, ki jih drugi načini ne omogočajo
- Včasih ni možnosti prehoda STL->LAD ali STL->FBD
- Omejenost na proizvajalca krmilnika

```
LD      I0.0        //Read one input
A       I0.1        //AND with another input
=       Q1.0        //Write value to output 1
```

Načini programiranja

LAD (Ladder Diagram)

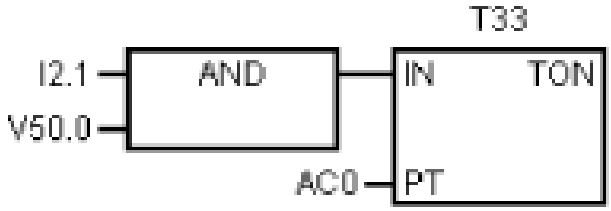
- Izhaja iz časov elektromehanskih krmilnikov
- Primeren za začetnike
- Korenine elektromehanskih krmilnikov so nekonsistentne (npr. matematične operacije)
- Grafična prezentacija je enostavna za razumevanje



Načini programiranja

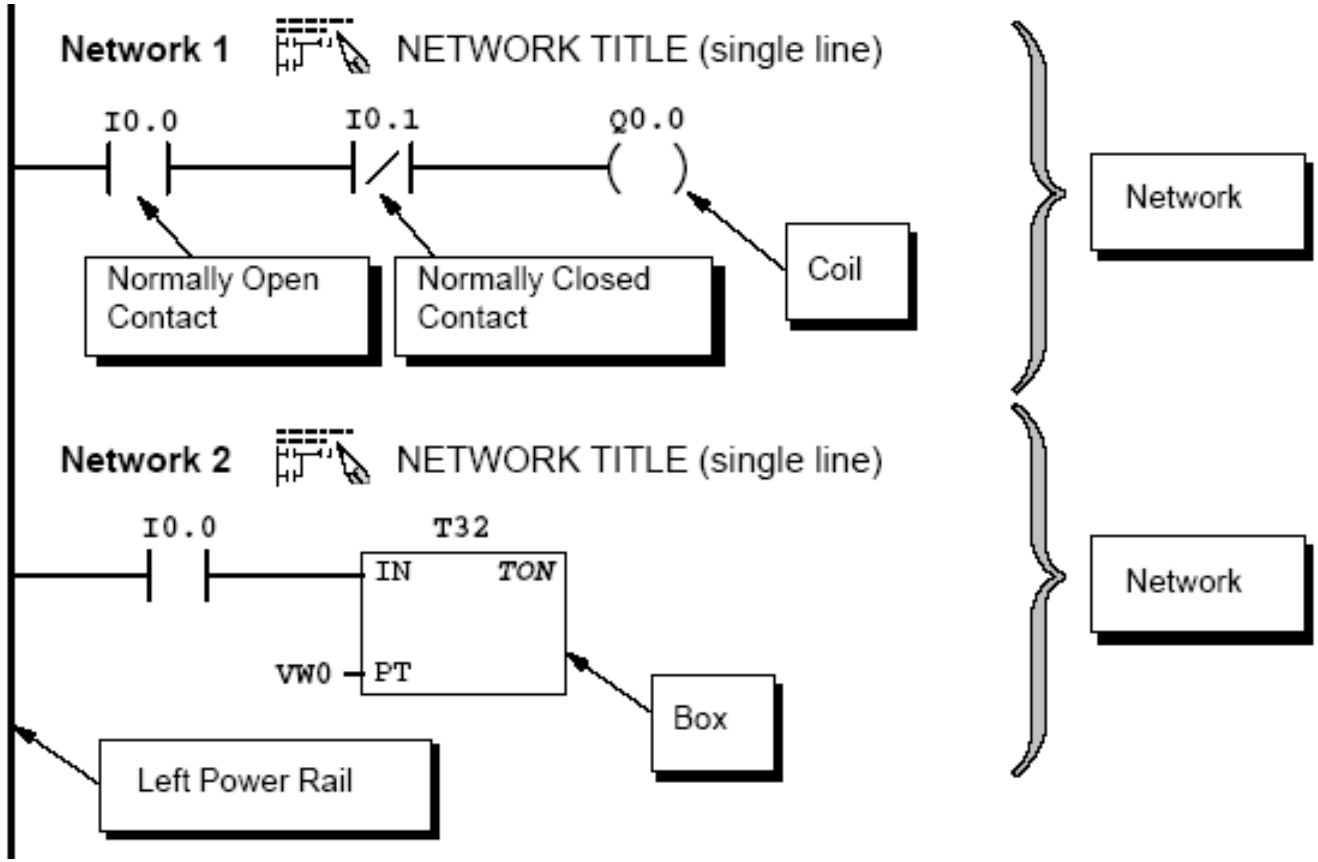
FBD (Function Block Diagram)

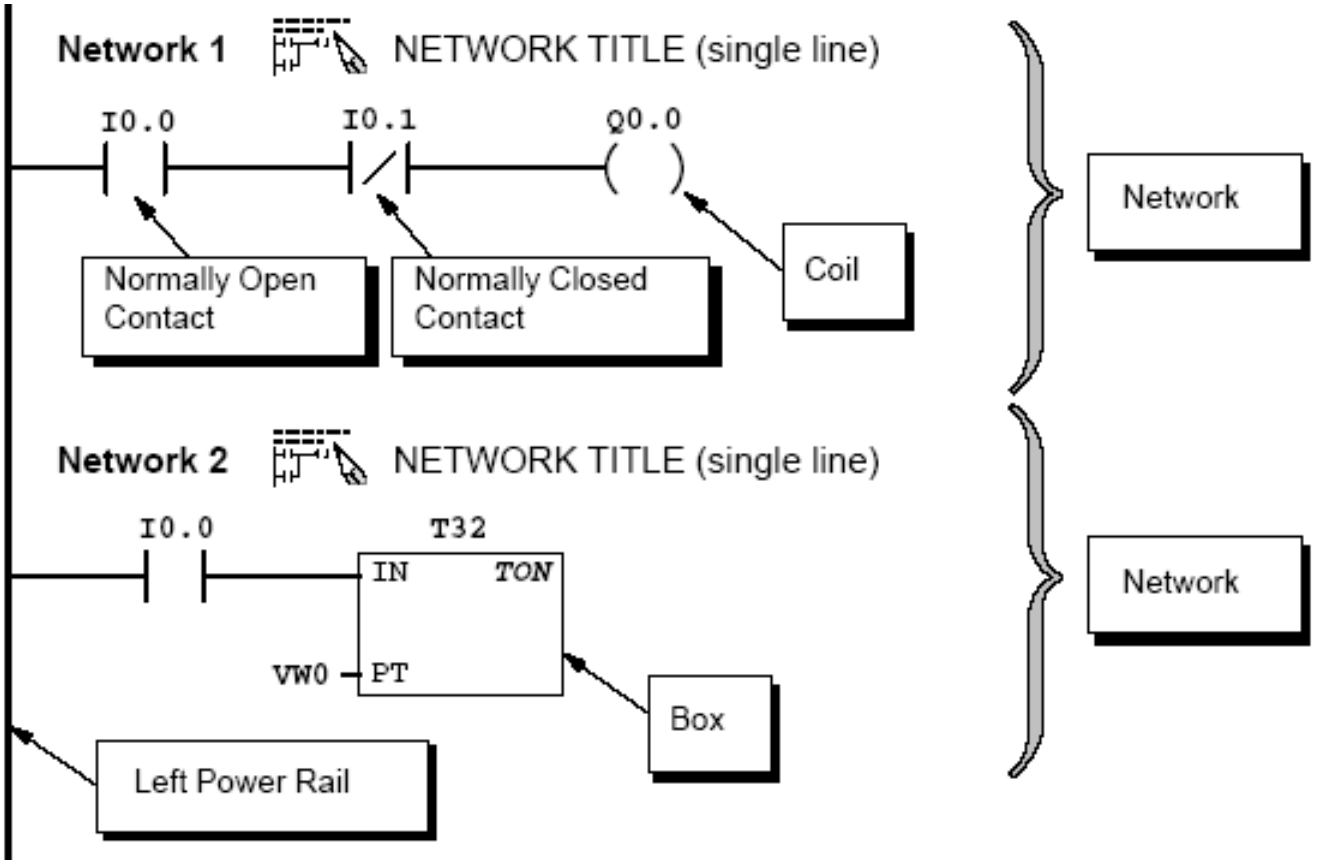
- Enostavno spremljanje programskega toka



Načini programiranja

Osnovni elementi
(Contacts, Coils, Boxes, Networks)

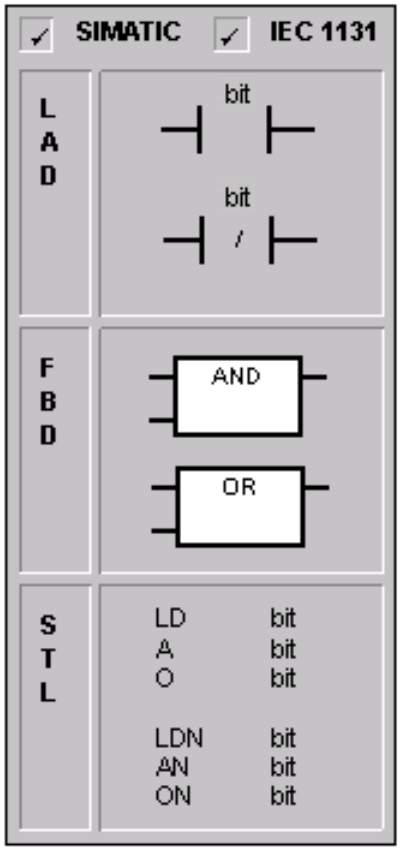




- Contacts: each of these elements represents a switch through which power can flow when a switch is closed.
- Coils: each of these elements represents a relay that is energized by power flowing to that relay.
- Boxes: each of these elements represents a function that is executed when power flows to the box.
- Networks: each of these elements forms a complete circuit. Power flows from the left power rail through the closed contacts to energize the coils or boxes.

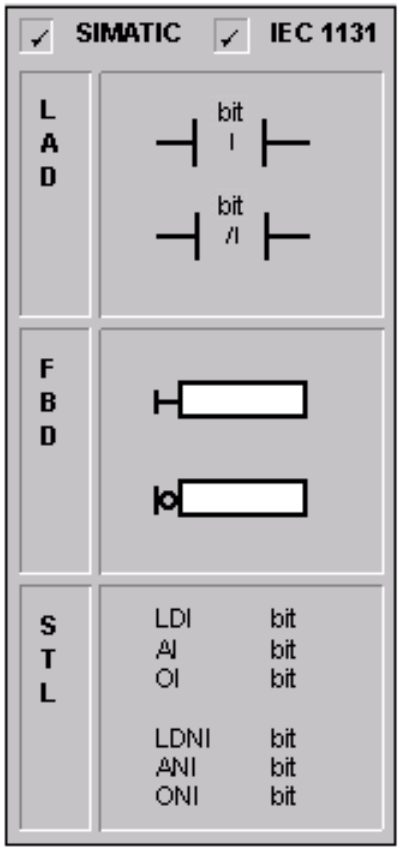
Contacts

- Standardni kontakt vhod
- Standardni kontakt vhod negiran



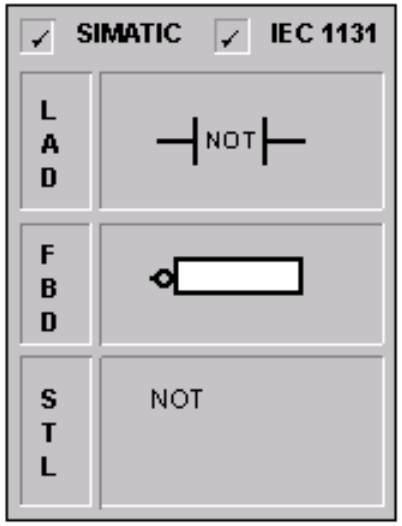
Contacts

- Standardni kontakt vhod – direktni dostop
- Standardni kontakt vhod negiran – direktni dostop



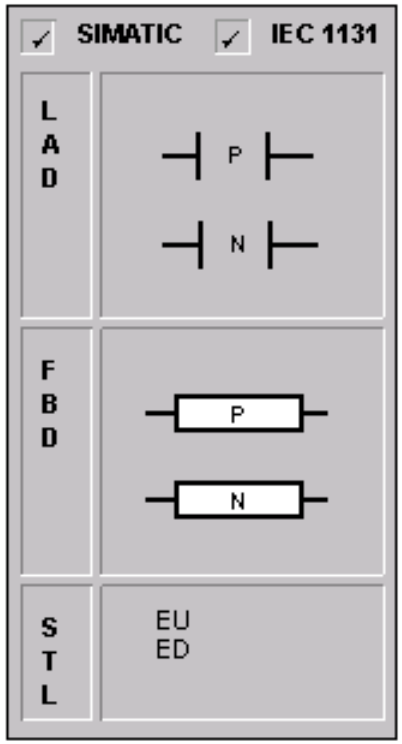
Contacts

- Negacija



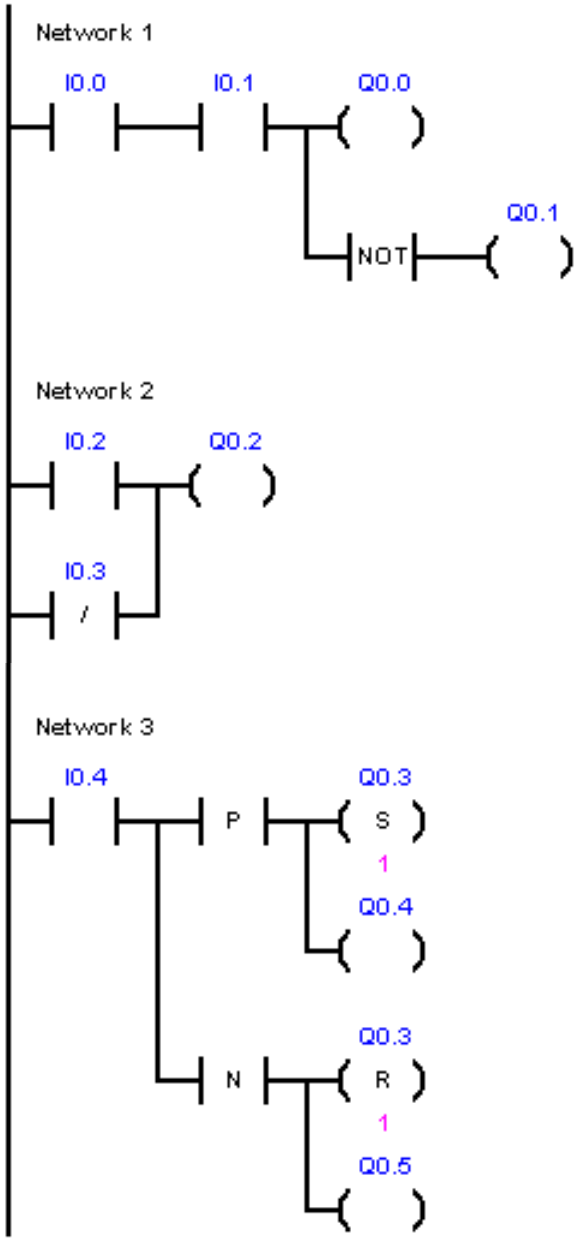
Contacts

- Impulz pri prehodu iz stanja 0 v 1
- Impulz pri prehodu iz stanja 1 v 0



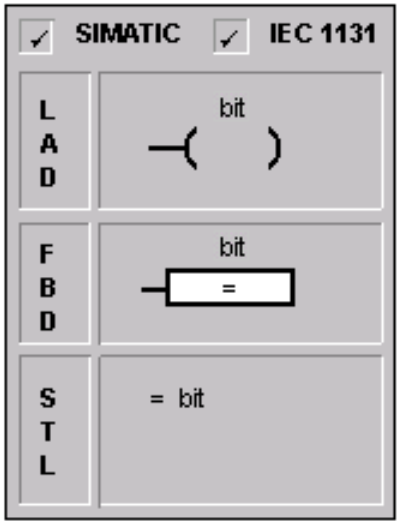
Contacts

Primeri



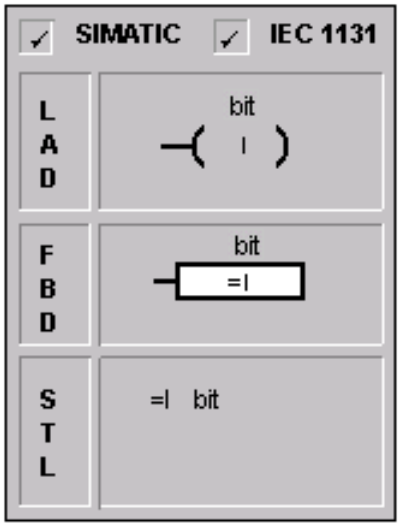
Coils

- Standardni kontakt izhod



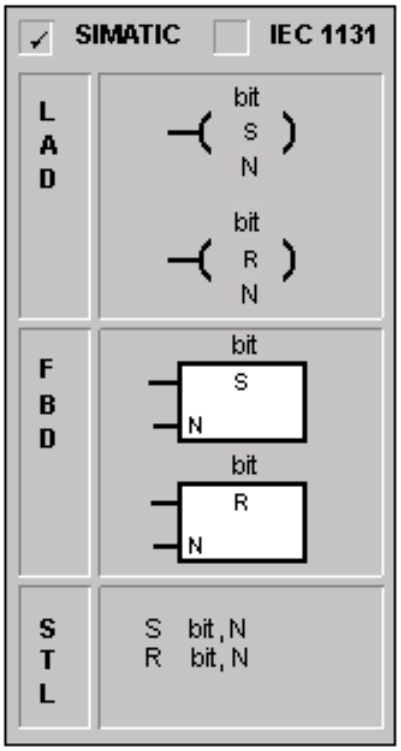
Coils

- Standardni kontakt izhod-direktni



Coils

- Postavitev vrednosti na 1 (SET)
- Postavitev vrednosti na 0 (RESET)

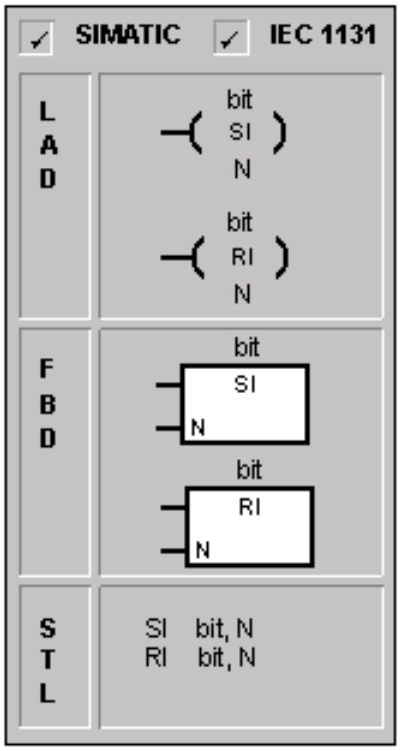


The **Set (S)** and **Reset (R)** instructions set (turn on) or reset (turn off) the specified number of points (N), starting at the specified address (Bit). You can set or reset from 1 to 255 points.

If the Reset instruction specifies either a timer bit (T) or counter bit (C), the instruction resets the timer or counter bit and clears the current value of the timer or counter.

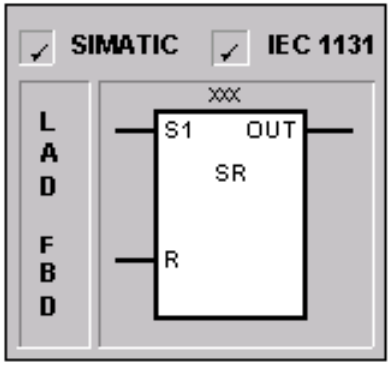
Coils

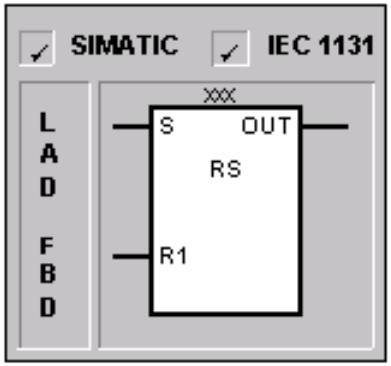
- Direktna postavitev vrednosti na 1 (SET)
- Direktna postavitev vrednosti na 0 (RESET)



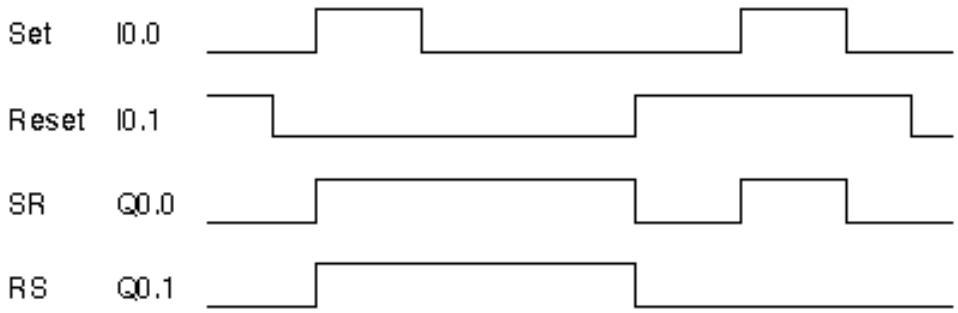
Coils

- Dominantno bistabilno stanje SR
- Dominantno bistabilno stanje RS



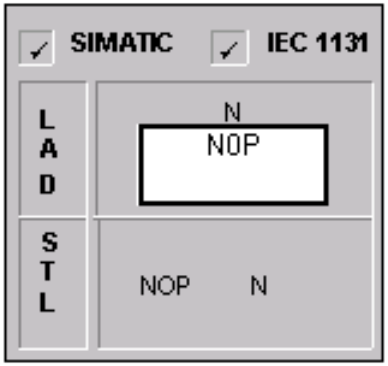


Timing Diagram



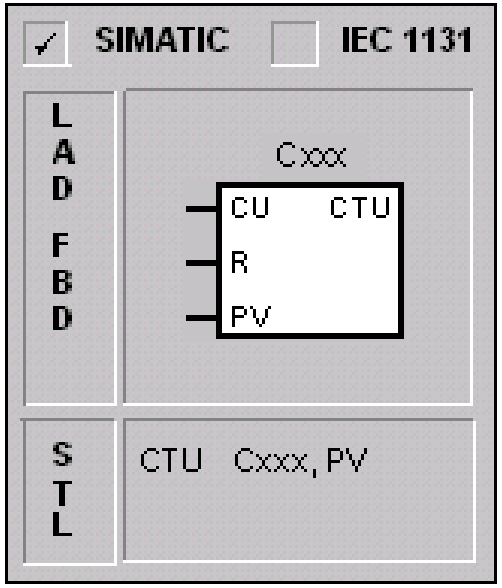
Coils

- No operation



Števci

- Števec gor (CTU)



The **Count Up (CTU)** instruction counts up from the current value each time the count-up input CU makes the transition from off to on. When the current value (Cxxx) is greater than or equal to the Preset Value (PV), the counter bit (Cxxx) turns on. The counter is reset when the Reset (R) input turns on, or when the Reset instruction is executed. The counter stops counting when it reaches the maximum value (32,767).

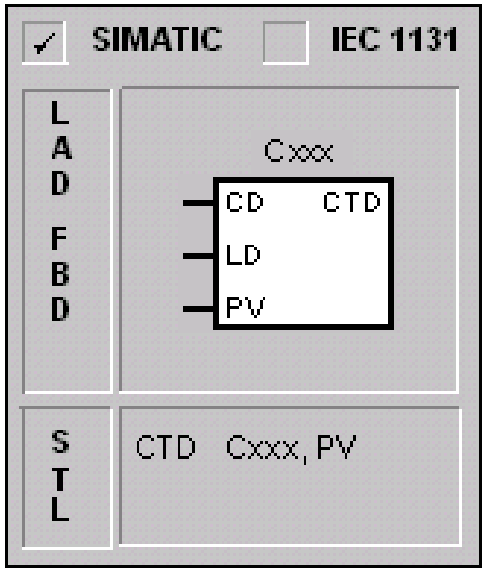
Counter ranges: Cxxx=C0 through C255

In STL, the CTU Reset input is the top of the stack value, while the Count Up input is the value loaded in the second stack location.

Note:
Since there is one current value for each counter, do not assign the same counter number to more than one counter. (Up Counters, Up/Down Counters, and Down Counters with the same number access the same current value.)

Števci

- Števec gor (CTD)



The **Count Down (CTD)** instruction counts down from the current value of that counter each time the count down input CD makes the transition from off to on. When the current value Cxxx is equal to zero, the counter bit (Cxxx) turns on. The counter resets the counter bit (Cxxx) and loads the current value with the preset value (PV) when the load input (LD) turns on. The Down Counter stops counting when it reaches zero, and the counter bit Cxxx turns on.

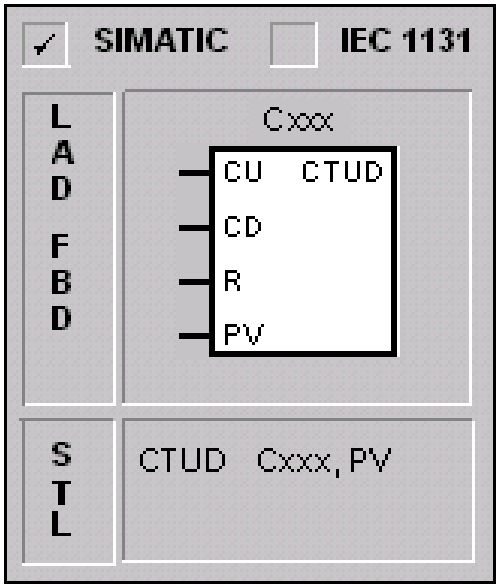
Counter ranges: Cxxx=C0 through C255

In STL, the CTD Load input is the top of the stack value, while the Count Down input is the value loaded in the second stack location.

Note:
Since there is one current value for each counter, do not assign the same counter number to more than one counter. (Up Counters, Up/Down Counters, and Down Counters with the same number access the same current value.)

Števci

- Števec gor/dol (CTUD)



The **Count Up/Down (CTUD)** instruction counts up each time the count-up input CU makes the transition from off to on, and counts down each time the count-down input CD makes the transition from off to on. The current value Cxx of the counter maintains the current count. The preset value PV is compared to the current value each time the counter instruction is executed.

Upon reaching maximum value (32,767), the next rising edge at the count-UP input causes the current count to wrap around to the minimum value (-32,768). On reaching the minimum value (-32,768), the next rising edge at the count-DOWN input causes the current count to wrap around to the maximum value (32,767).

When the current value Cxx is greater than or equal to the preset value PV, the counter bit Cxx turns on. Otherwise, the counter bit turns off. The counter is reset when the Reset (R) input turns on, or when the Reset instruction is executed. The CTUD counter stops counting when it reaches PV.

Counter ranges: Cxxx=C0 through C255