

# Адресация

## Обзор главы

# 7

В разделе	Вы найдете	на стр.
7.1	Обзор	7–2
7.2	Виды адресов	7–4

## 7.1 Обзор

### Что такое адресация?

Многие команды FUP работают с одним ли несколькими операндами. Операнд определяет константу или адрес, по которому команда находит переменную, которую она использует для выполнения логической операции. Этот адрес может быть битом, байтом, словом или двойным словом.

Возможными операндами являются, например:

- константа, значение таймера или счетчика или строка символов ASCII
- бит в слове состояния программируемого контроллера
- блок данных и адрес внутри области блока данных

### Непосредственная и прямая адресация

В Вашем распоряжении имеются следующие виды адресации:

- Непосредственная адресация (задание константы в качестве операнда)
- Прямая адресация (задание переменной в качестве операнда)

На рис. 7–1 показан пример непосредственной и прямой адресации.

Функция блока состоит в том, чтобы сравнить два входных параметра (в данном случае два 16-битовых целых числа), чтобы установить, меньше или равен первый вход второму. Константа 50 вводится как входной параметр IN1. Меркерное слово MW200, адрес в памяти, вводится как входной параметр IN2.

Так как в этом примере константа 50 является фактическим значением, с которым IN1 блока будет работать, то 50 является непосредственным адресом блока команды. Так как MW200 указывает на адрес в памяти, по которому находится значение, с которым должен работать IN2 блока, MW200 является прямым адресом. MW200 - это адрес, а не само фактическое значение.

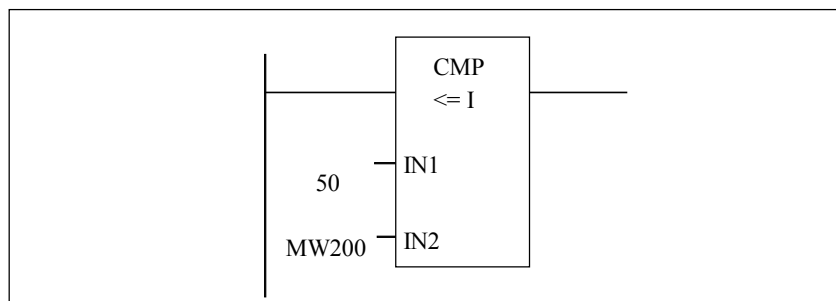


Рис. 7-1. Непосредственная и прямая адресация

Таблица 7–1. Форматы констант для непосредственной адресации, использующей адреса элементарных типов данных

Тип и описание	Размер в битах	Возможные форматы	Диапазон и представление чисел (от минимального до максимального значения)	Пример
BOOL (бит)	1	Булев текст	TRUE/FALSE	TRUE
BYTE (байт)	8	Шестнадцатичное число	от B#16#0 до B#16#FF	B#16#10 byte#16#10
WORD (слово)	16	Двоичное число Шестнадцатичное число BCD Десятичное число без знака	от 2#0 до 2#1111_1111_1111_1111 от W#16#0 до W#16#FFFF от C#0 до C#999 от B#(0,0) до B#(255,255)	2#0001_0000_0000_0000 W#16#1000 word16#1000 C#998 B#(10,20) byte#(10,20)
DWORD (двойное слово)	32	Двоичное число Шестнадцатичное число Десятичное число без знака	от 2#0 до 2#1111_1111_1111_1111_1111_1111_1111_1111 от DW#16#0000_0000 до DW#16#FFFF_FFFF от B#(0,0,0,0) до B#(255,255,255,255)	2#1000_0001_0001_1000_1011_1011_0111_1111_1111_1111_1111_1111 DW#16#00A2_1234 dword#16#00A2_1234 B#(1,14,100,120) byte#(1,14,100,120)
INT (целое число)	16	Десятичное число со знаком	от –32768 до 32767	1
DINT (двойное целое число)	32	Десятичное число со знаком	от L#–2147483648 до L#2147483647	L#1
REAL (число с плавающей точкой)	32	IEEE число с плавающей точкой	Верхняя граница: $\pm 3.402823 \times 10^{38}$ Нижняя граница: $\pm 1.175495 \times 10^{-38}$ (см. также табл. C–5)	1.234567e+13
S5TIME (время в формате SIMATIC)	16	Время S5 шагами по 10 мс (значение по умолчанию)	от S5T#0H_0M_0S_10MS до S5T#2H_46M_30S_0MS и S5T#0H_0M_0S_0MS	S5T#0H_1M_0S_0MS S5TIME#0H_1M_0S_0MS
TIME (время в формате IEC)	32	Время в формате IEC с дискретностью 1 мс, целое число со знаком	от T#–24D_20H_31M_23S_648MS до T#24D_20H_31M_23S_647MS	T#0D_1H_1M_0S_0MS TIME#0D_1H_1M_0S_0MS
DATE (дата в формате IEC)	16	Дата в формате IEC с дискретностью 1 день	от D#1990–1–1 до D#2168–12–31	D#1994–3–15 DATE#1994–3–15
TIME_OF_DAY (время суток)	32	Время суток с дискретностью 1 мс	от TOD#0:0:0.0 до TOD#23:59:59.999	TOD#1:10:3.3 TIME_OF_DAY#1:10:3.3
CHAR (символ)	8	Символ ASCII	'A', 'B' и т.д.	'E'

## 7.2 Виды адресов

### Возможные адреса

в качестве адреса команды FUP может использоваться один из следующих элементов:

- бит, состояние сигнала которого должно быть опрошено
- бит, которому присваивается состояние сигнала цепи логических операций
- бит, которому присваивается результат логической операции (RLO)
- бит, который должен быть установлен или сброшен
- номер счетчика, который должен быть увеличен или уменьшен
- номер таймера, который должен быть использован
- меркер фронта, сохраняющий предыдущее значение RLO
- меркер фронта, сохраняющий предыдущее состояние сигнала другого адреса
- байт, слово или двойное слово, содержащее значение, с которым должен работать элемент и блок FUP
- номер блока данных (DB или DI), который должен быть открыт или создан
- номер подлежащей вызову функции (FC), системной функции (SFC), функционального блока (FB) или системного функционального блока (SFB)
- метка перехода, на которую нужно перейти

### Идентификатор адреса

Переменные, используемые в качестве операндов, состоят из идентификатора адреса и адреса внутри области памяти, указываемой идентификатором адреса. Идентификатор адреса может быть одного из следующих двух основных типов:

- Идентификатор адреса, задающий следующие два объекта данных:
  - область памяти, в которой операция находит значение (объект данных), с которым она выполняет логическую операцию (например, "I" для отображения процесса на входах, см. табл. 6–1).
  - размер значения (объекта данных), с которым команда должна выполнить логическую операцию (напр., B для байта, W для слова и D для двойного слова, см. табл. 6–1).
- Идентификатор адреса, указывающий область памяти, но не размер объекта данных в этой области (например, идентификатор для области T (таймеры), C (счетчики) или DB или DI (блок данных) и номер таймера, счетчика или блока данных, см. табл. 6–1).

### Указатели

Указатель идентифицирует адрес переменной. Указатель содержит адрес вместо значения. При назначении фактического параметра для параметра типа "Pointer" ("Указатель") Вы предоставляете адрес в памяти. В STEP 7 Вы можете ввести указатель или в формате указателя, или просто как адрес (например, M 50.0). Следующий пример иллюстрирует формат указателя для обращения к данным, начинающимся в M 50.0.

R#M50.0

## Работа со словам или двойными словами как объектами данных

Если Вы работаете с командой, идентификатор адреса которой указывает на область памяти Вашего программируемого контроллера и на объект данных, который по размеру является словом или двойным словом, помните, что к адресу в памяти всегда обращаются как к байтовому адресу. Этот байтовый адрес является наименьшим номером байта или номером старшего байта внутри слова или двойного слова. Адрес в команде, показанной на рис. 7-2, например, указывает на четыре последовательных байта в области памяти М, начиная с байта 10 (MB 10) до байта 13 (MB 13).

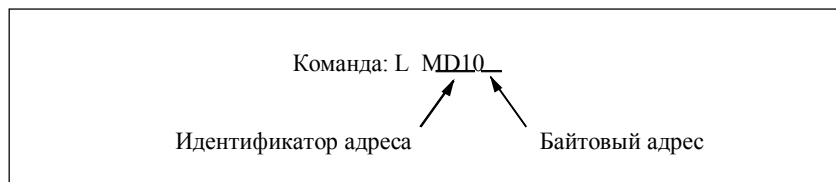


Рис. 7-2. Пример адреса памяти, ссылка на который делается в форме байтового адреса

На рис. 7-3 показаны объекты данных следующих размеров:

- двойное слово: меркерное двойное слово MD10
- слово: меркерное слово MW10, MW11 и MW12
- байт: меркерные байты MB10, MB11, MB12 и MB13

Если Вы используете абсолютные операнды размером в слово или двойное слово, то убедитесь, что Вы избежали таких назначений байтов, при которых они перекрываются.

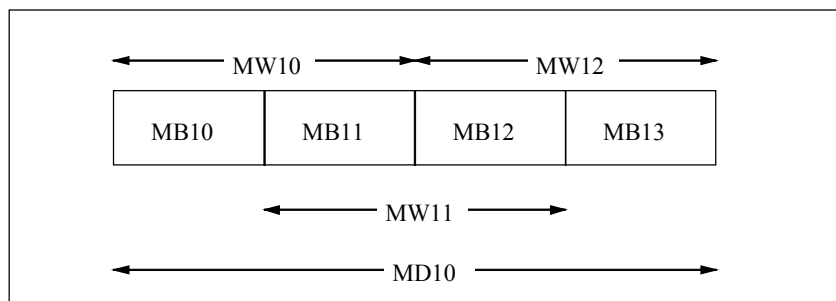


Рис. 7-3. Ссылка на адрес памяти в форме байтового адреса