**Опера́нд** (англ. **operand**) в языках программирования ― аргумент операции; данные, которые обрабатываются командой; грамматическая конструкция, обозначающая выражение, задающее значение аргумента операции.

[**Операнд**](https://dic.academic.ru/dic.nsf/fin_enc/26315) — константа, переменная, функция, выражение и другой объект языка программирования, над  которым производятся операции

Способ **адресации** — **это** способ формирования исполнительного **адреса** **операнда** по адресному коду команды. Исполнительным **адресом** **операнда** (Амп)называется двоичный код номера ячейки памяти, служащей источником или приемником **операнда**. Адресный код команды (АК)— **это** двоичный код в адресном поле команды, из которого необходимо сформировать исполнительный **адрес** **операнда**

Большая часть команд микропроцессора работает с кодами данных (операндами). Одни команды требуют входных операндов (одного или двух), другие выдают выходные операнды (чаще один операнд). Входные операнды называются операндами-источниками, а выходные - операндами-приемниками. Коды операндов (входные и выходные) могут находиться во внутренних регистрах процессора. Они могут располагаться в системной памяти (самый распространенный вариант) или в устройствах ввода/вывода. Определение места положения операндов производится кодом команды. Существуют разные методы, с помощью которых код команды может определить, откуда брать входной операнд и куда помещать выходной операнд. Эффективность выбранных методов адресации определяет эффективность работы всего микропроцессора в целом.

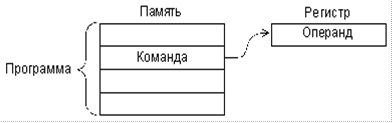
**Количество методов адресации** в различных микропроцессорах может быть от 4 до 16. Рассмотрим несколько методов адресации операндов, используемых в настоящее время в большинстве микропроцессоров.

***Непосредственная адресация*** (рис. 1) предполагает, что операнд (входной) находится в памяти непосредственно за кодом команды. Операнд - константа, которую надо куда-то переслать, к чему-то прибавить и т.д. Например, команда может состоять в том, чтобы прибавить число 6 к содержимому какого-то внутреннего регистра процессора. Число 6 будет располагаться в памяти, внутри программы в адресе, следующем за кодом данной команды сложения.

|  |  |
| --- | --- |
| http://ok-t.ru/studopedia/baza16/1366604574476.files/image002.gif   Рис.1. Непосредственная адресация | http://ok-t.ru/studopedia/baza16/1366604574476.files/image004.gif Рис. 2. Прямая адресация |

***Прямая* (абсолютная) *адресация*** (рис. 2) предполагает, что операнд (входной или выходной) находится в памяти по адресу, код которого находится внутри программы сразу же за кодом команды. Например, необходимо очистить (сделать нулевым) содержимое ячейки памяти с адресом 0В00. Код адреса 0В00 будет располагаться в памяти, внутри программы в следующем адресе за кодом данной команды очистки.

*Регистровая адресация* (рис. 3) предполагает, что операнд (входной или выходной) находится во внутреннем регистре процессора. Например, команда может состоять в том, чтобы переслать число из нулевого регистра в первый. Номера обоих регистров (0 и 1) будут определяться кодом команды пересылки.

  
Рис. 3. Регистровая адресация

***Косвенно-регистровая* (она же косвенная) *адресация***предполагает, что во внутреннем регистре процессора находится не сам операнд, а его адрес в памяти (рис. 4) Например, команда может состоять в том, чтобы очистить ячейку памяти с адресом, находящимся в нулевом регистре. Номер этого регистра (0) будет определяться кодом команды очистки

