



# *Классификация машинных команд*

Учитель информатики МБОУ «СОШ № 4  
МО «Ахтубинский район» Предвечная А. Н.

# *Немного истории...*

Когда-то ассемблер был языком, без знания которого нельзя было заставить компьютер сделать что-либо полезное.

Постепенно появлялись более удобные средства общения с компьютером. Но, в отличие от других языков, ассемблер не умирал, более того он не мог сделать этого в принципе. Почему?

Если коротко, то язык ассемблера — это символическое представление машинного языка. Все процессы в машине на самом низком, аппаратном уровне приводятся в действие только командами машинного языка.

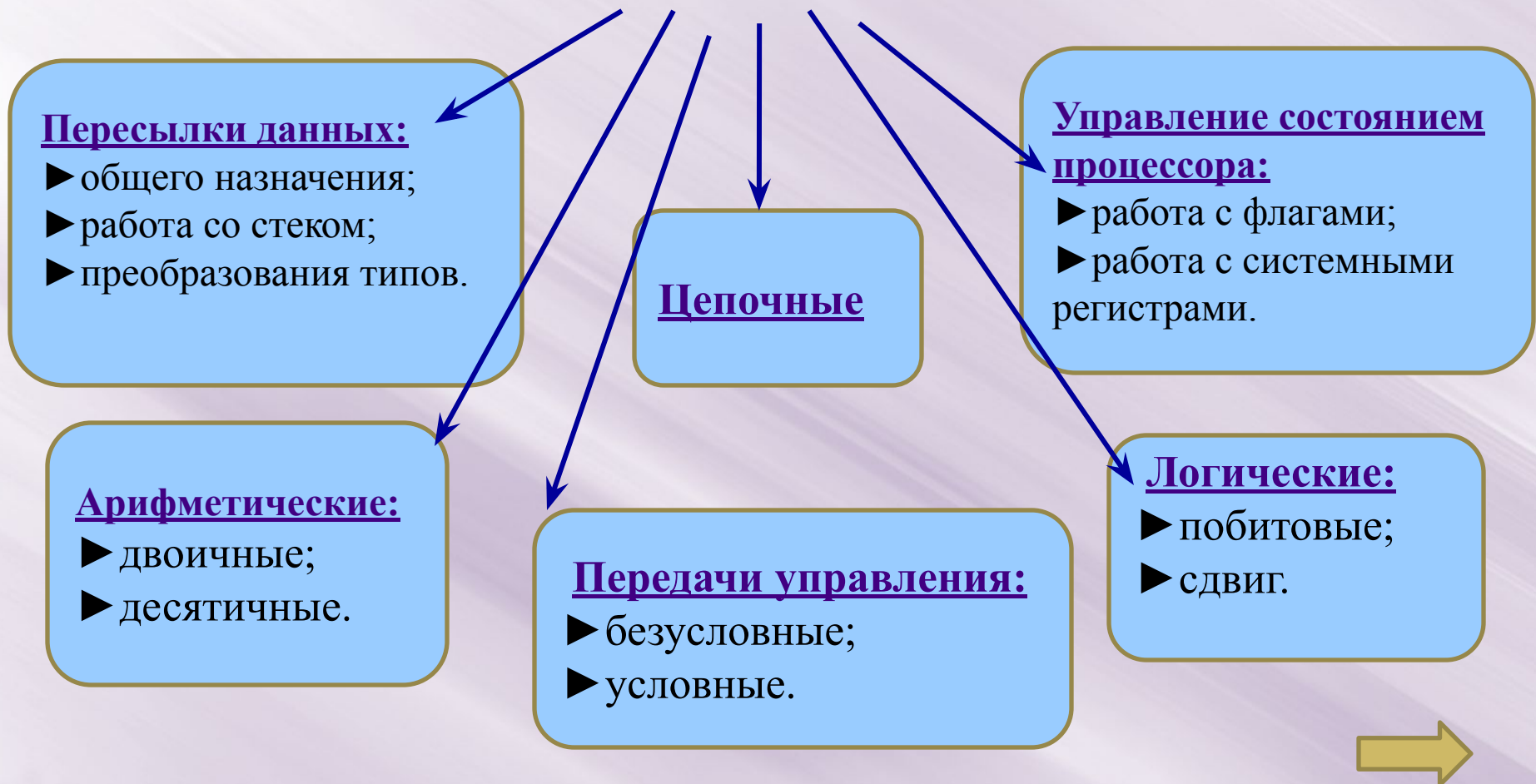
Так как язык ассемблера для компьютера “родной”, то и самая эффективная программа может быть написана только на нем.



```
0400 2073FE JSR #FE73      s
0403 A200   LDX #0         "D
0405 B00004 LDR #480,X    =Dx
0408 F006   BEQ #410     p/
040A 2075FE JSR #FE75    u"
040D E8     INX         h
040E D0F5   BRE #405     Pu
0410 00     BRK         D
0411 B9     X=#480
0430 48     'H         H
0481 45     'E         E
0482 4C     'L         L
0483 4C     'L         L
0484 4F     'O         O
0485 00     #0         D
0486 67     'I
```

**Машинная команда представляет собой закодированное по определенным правилам указание процессору на выполнение некоторой операции.**

## **Классификация машинных команд**



# Команды пересылки данных

## Команды пересылки данных общего назначения

**mov** - основная команда пересылки данных. Она реализует самые разнообразные варианты пересылки.

Командой **mov** нельзя осуществить пересылку из одной области памяти в другую. Если такая необходимость возникает, то нужно использовать в качестве промежуточного буфера любой доступный в данный момент регистр общего назначения.

## Команды работы со стеком

Это набор специализированных команд, ориентированных на организацию гибкой и эффективной работы со стеком.

**Стек** — это область памяти, специально выделяемая для временного хранения данных программы.

Для работы со стеком предназначены три регистра:

**ss** — сегментный регистр стека;

**sp/esp** — регистр указателя стека;

**bp/ebp** — регистр указателя базы кадра стека.



# АРИФМЕТИЧЕСКИЕ КОМАНДЫ

Микропроцессор может выполнять целочисленные операции и операции с плавающей точкой. Для этого в его архитектуре есть два отдельных блока:

- **устройство для выполнения целочисленных операций;**
- **устройство с плавающей точкой.**

Каждое из этих устройств имеет свою систему команд. В принципе, целочисленное устройство может взять на себя многие функции устройства с плавающей точкой, но это потребует больших вычислительных затрат.

**!** Для большинства задач, использующих язык ассемблера, достаточно целочисленной арифметики.



# Логические команды

В основе логических преобразований лежат правила формальной логики.

Формальная логика работает на уровне утверждений истинно и ложно. Для микропроцессора это, как правило, означает 1 и 0 соответственно.

Для компьютера язык нулей и единиц является родным, но минимальной единицей данных, с которой работают машинные команды, является байт. Однако, на системном уровне часто необходимо иметь возможность работать на предельно низком уровне — на уровне **бит**.

Все команды сдвига перемещают биты в поле операнда влево или вправо в зависимости от кода операции.

Количество сдвигаемых разрядов — **счетчик\_сдвигов** — располагается на месте второго операнда и может задаваться двумя способами:

- статически, что предполагает задание фиксированного значения с помощью непосредственного операнда;
- динамически, что означает занесение значения счетчика сдвигов в регистр `cl` перед выполнением команды сдвига.

Все команды сдвига устанавливают флаг переноса `cf`.

По принципу действия команды сдвига можно разделить на два типа:

- команды линейного сдвига;
- команды циклического сдвига.

# Команды передачи управления

Регистр **ecx/cx** имеет определенное функциональное назначение — он выполняет роль счетчика в командах управления циклами и при работе с цепочками символов.

Синтаксис этой команды условного перехода таков:

**jcxz** метка\_перехода (Jump if cx is Zero) — переход, если **cx** ноль;

**jecxz** метка\_перехода (Jump Equal ecx Zero) — переход, если **ecx** ноль.

Эти команды очень удобно использовать при организации цикла и при работе с цепочками символов.

Нужно отметить ограничение, свойственное команде **jcxz/jecxz**.

В отличие от других команд условной передачи управления, команда **jcxz/jecxz** может адресовать только короткие переходы — на  $-128$  байт или на  $+127$  байт от следующей за ней команды.



# ЦЕПОЧЕЧНЫЕ КОМАНДЫ

**Цепочка** - непрерывная последовательность байт, слов или двойных слов, обрабатываемая как единое целое. Основное отличие цепочек от массивов состоит в способе доступа к элементам: для массивов - произвольный доступ, для цепочек - только последовательный (от начала цепочки к концу или от конца к началу).

**Цепочечные команды** - команды для обработки цепочек.

Особенностью всех цепочечных команд является автоматическое продвижение к следующему элементу цепочки.

## Адресация операндов

цепочка источник - *ds:si*

цепочка приёмник - *es:di*

## Направление обработки

от начала к концу

**df = 0**; *si* и *di* автоматически увеличиваются

команда **cld** (clear direction flag) сбрасывает флаг **df**

от конца к началу

**df = 1**; *si* и *di* автоматически уменьшаются

команда **std** (set direction flag) устанавливает флаг **df**





# Команды управления состоянием процессора:

Для контроля над работой процессора используются различные **регистры**. В большинстве машин эти регистры в основном не доступны пользователю. Некоторые из них могут быть доступны для машинных команд, исполняемых в так называемом режиме управления или режиме операционной системы. Конечно, у разных типов машин организация регистров отличается; для их классификации также используется различная терминология.

**Флаги** используются, как правило, в командах проверки условий.

**0 (CF)** - флаг переноса при вычислениях или операциях сдвига

**2(PF)** - флаг приоритета

**4(AF)** - флаг дополнительного переноса

**6(ZF)** - флаг нулевого результата

**7(SF)** - флаг знака числа.


**8(TF)** - флаг трассировки. Полезен при отладке программ.

**9(IF)** - флаг разрешения прерываний

**10(DF)** - флаг направления (в циклических операциях)

**11(OF)** - флаг переполнения при вычислениях.





**Т. о., система машинных команд - важнейшая часть архитектуры компьютера, определяющая возможности его программирования.**

**Для работы процессора достаточно программы в двоичных кодах, но такое прямое программирование на практике не используется. Язык ассемблера - символический аналог машинного языка.**

**Преобразование команд ассемблера в соответствующие машинные команды производит программа-транслятор - ассемблер. Дальнейшая интерпретация машинных команд в конкретные сигналы электронных схем осуществляется с помощью блока микропрограммного управления, входящего в состав процессора.**



## Литература

1. Юров В., Хорошенко С. Assembler: учебный курс. – СПб: Питер Ком, 1999.
2. [http://mf.grsu.by/Kafedry/prikl\\_mat/academic\\_process/045/lec\\_37.doc](http://mf.grsu.by/Kafedry/prikl_mat/academic_process/045/lec_37.doc)