

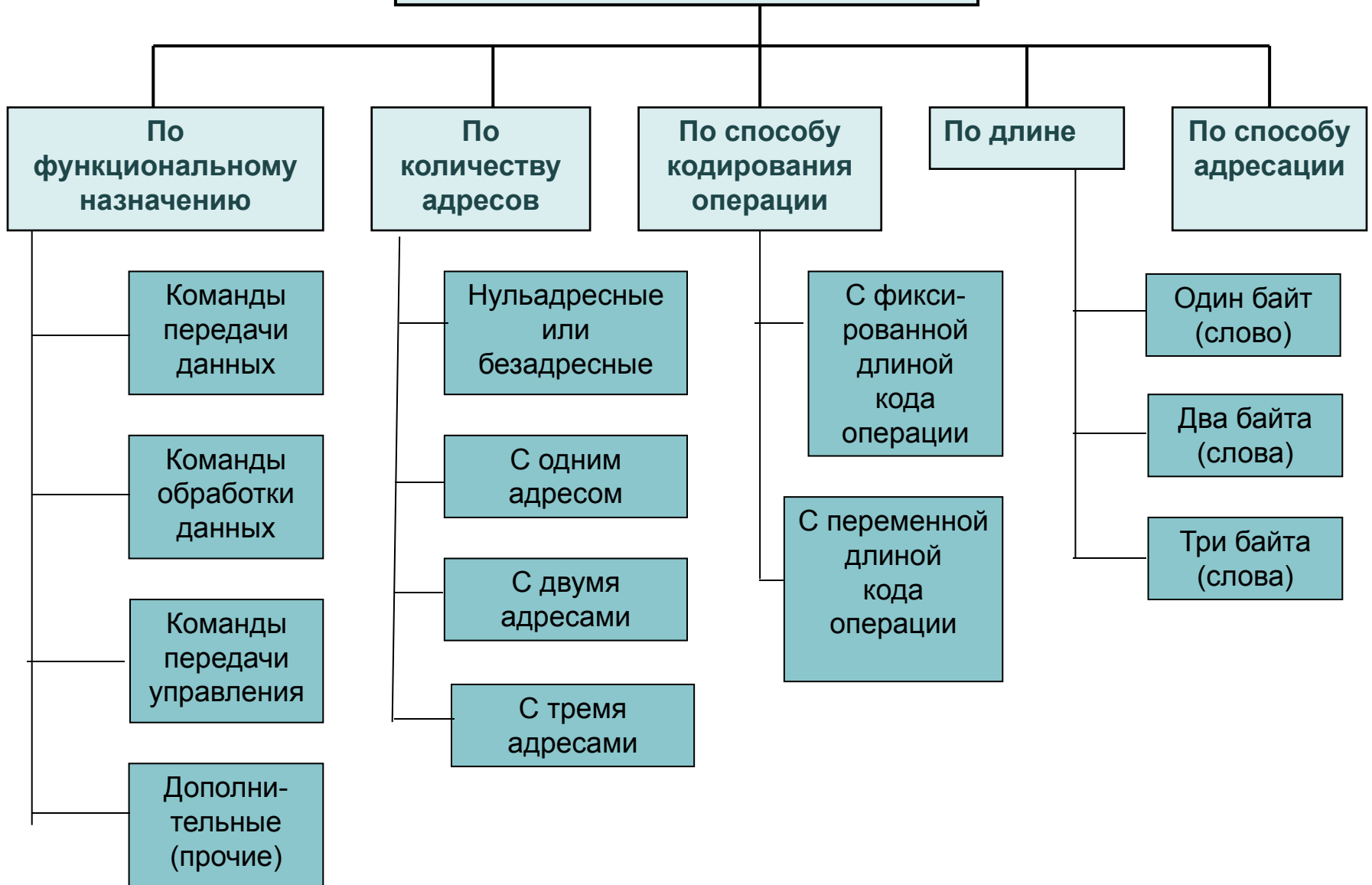
ОСНОВНЫЕ КОМАНДЫ ЭВМ

- Классификация команд по различным признакам
- Структура команд ЭВМ
- Команды передачи данных
- Команды обработки данных
- Команды передачи управления
- Команды для работы с подпрограммами. Стеки.
- Прочие команды ЭВМ.

Система команд ЭВМ

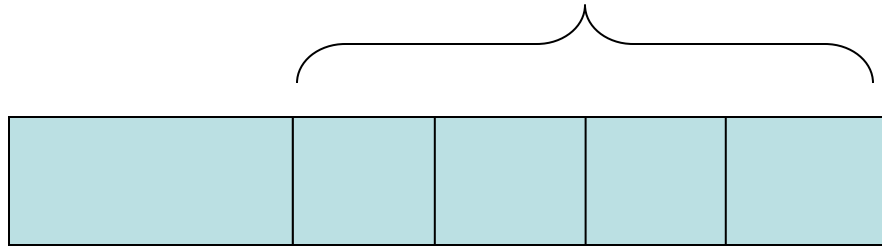
- Все разнообразие решаемых на ЭВМ задач реализуется с помощью небольшого набора очень простых команд.
- Система команд у типичной ЭВМ включает в себя всего 60-150 базовых команд. Все команды в основном служат для выполнения очень простых действий, таких, как прочитать, запомнить, сложить, сдвинуть, сравнить и т.д.
- Интеллектуальность ЭВМ достигается за счет того, что ЭВМ способна выполнять программы, состоящие из большого числа таких простых действий с огромной, не достижимой для человека скоростью.
- При описании системы команд ЭВМ обычно принято классифицировать команды по следующим признакам.

Классификация команд ЭВМ



Структура команды ЭВМ

Адресная часть (коды адресов ячеек памяти)



Операционная часть (код операции)

Код операции	a1	a2	a3
--------------	----	----	----

- Трехадресная команда

a1, a2 – адреса ячеек (регистров), где находятся числа, участвующие в операции (операнды)

a3 – адрес ячейки оперативной памяти, куда нужно поместить результат

Код операции	a1	a2
--------------	----	----

- Двухадресная команда

Результат записывается в ячейку a2

Код операции	a1
--------------	----

- Одноадресная команда

a1 – адрес ячейки, где хранится число участвующее в операции или адрес ячейки, где записывается результат

Код операции

- Нуль адресная команда

Все операнды в регистре ЦП

Команды передачи данных

- Данная группа команд включает в себя подгруппы команд передачи кодов между регистрами внутри процессора, из регистров процессора в память, из памяти в регистры процессора, из одних ячеек памяти в другие и передачи данных между процессором и портами внешних устройств.
- Отдельную подгруппу составляют команды работы со стеком. Они позволяют включить данные в стек для временного хранения и извлекать данные из стека при необходимости их использования.

Команды обработки данных

- Данную группу команд с точки зрения выполняемых над данными операций можно подразделить на **арифметические** (сложить, вычесть, умножить, сравнить), **логические** (операции И, ИЛИ, НЕ и т.д.) и команды **сдвига**.
- Команды этого типа могут иметь один или два операнда. Операнды могут храниться в регистрах центрального процессора, в памяти или в самой команде.

- Результат операции формируется в регистре-приемнике или в специализированном регистре-аккумуляторе.
- Команды данной группы формируют также признаки результатов, устанавливаемые в регистре флагов процессора: перенос из старшего разряда, переполнение, нулевой результат и др.

Подробнее о команде сравнения

- Обычно для сравнения двух чисел процессор выполняет операцию вычитания. По результату вычитания устанавливаются флаги во флаговом регистре. Очевидно, что если сравниваемые числа равны, результат вычитания будет нулевым и в регистре установится флаг нулевого результата.
- Если первое из сравниваемых чисел больше — результат вычитания будет отрицательным и установится флаг отрицательного результата.
- Результат вычитания не сохраняется в памяти, поскольку по состоянию флагового регистра можно судить о результатах сравнения чисел.

Команды передачи управления

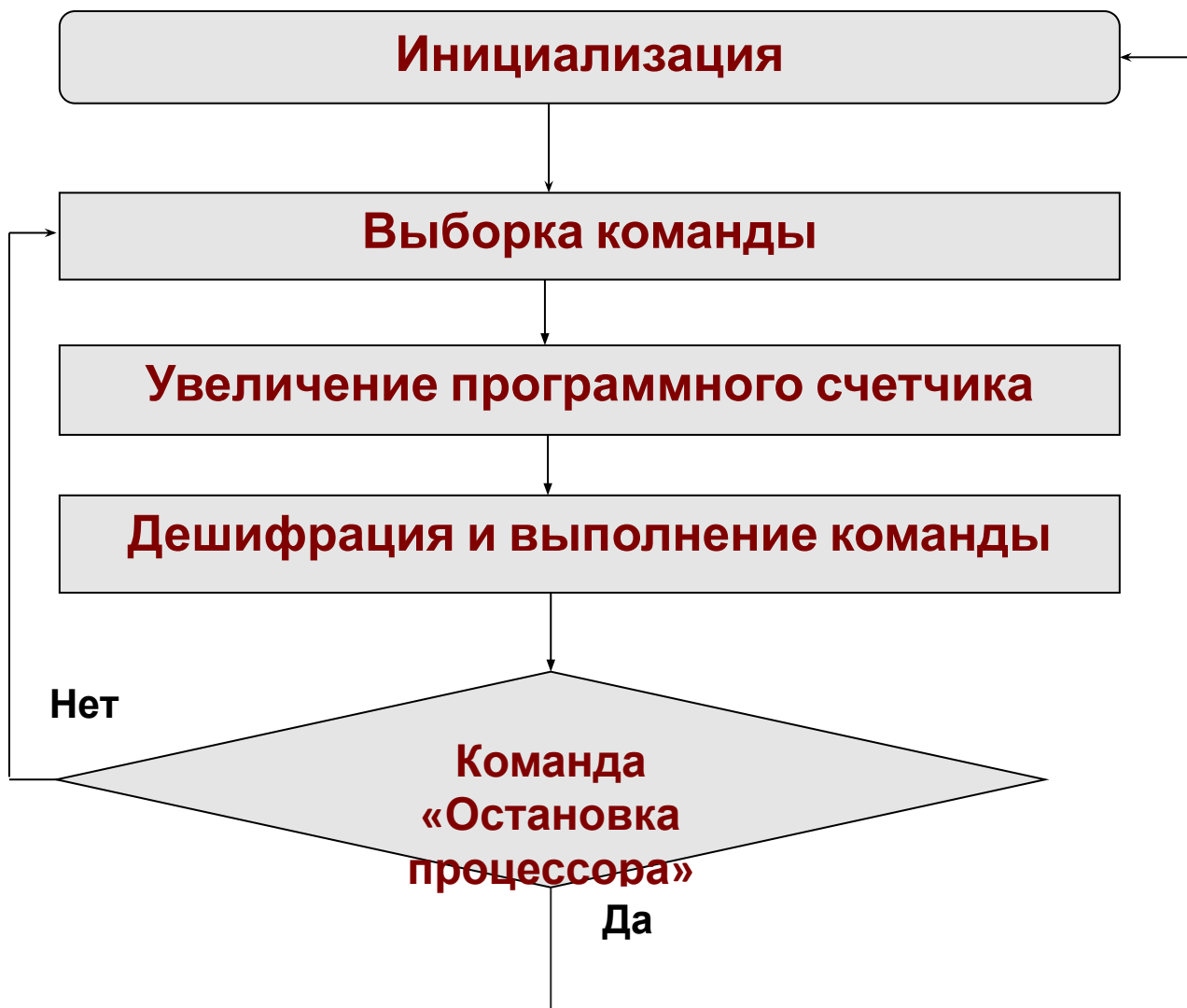
- Они имеют важное значение, так как используются для изменения естественного порядка следования команд и организации циклических участков в программах.
- Простейшей командой передачи управления является команда **безусловного** перехода **JMP <адрес>**, которая загружает адрес перехода, указанный в команде, в программный счетчик.
- Команды **условного** перехода проверяют указанное в команде условие и модифицируют программный счетчик, если условие истинно.

Пример команды условного перехода

- Оператор **IF (A>B) then go to L** некоторого языка высокого уровня может быть реализован двумя командами ЭВМ:
 - СРАВНИТЬ A и B
 - ПЕРЕЙТИ ЕСЛИ БОЛЬШЕ К АДРЕСУ L
- Если $A > B$, то результат вычитания будет положителен и соответственно флаг знака во флаговом регистре не установится.
- Вторая команда (условный переход) проверяет состояние флага знака и, если он не установлен, модифицирует **программный счетчик** так, чтобы его значение указывало на **адрес L**.

Организация подпрограмм

- В программировании широко используется такой прием, как организация подпрограмм. Подпрограмма описывается один раз, а вызываться может из различных мест программы неоднократно. Подпрограмма в процессе своей работы может вызвать другую.
- После того как подпрограмма закончила свою работу, управление должно быть передано на команду, следующую в памяти сразу за командой обращения к этой подпрограмме.
- Адрес команды, на которую управление передается после окончания работы подпрограммы, называется **адресом возврата**.



Где надо хранить адрес возврата?

- Для того, чтобы начать выполнять подпрограмму, в программный счетчик необходимо загрузить адрес первой команды подпрограммы. Для осуществления возврата из подпрограммы необходимо запомнить в каком-то месте адрес возврата. Можно, например, сохранить адрес возврата в одном из регистров процессора. Такой способ сохранения адреса возврата очень прост и легко реализуется.
- Однако часто встречаются подпрограммы, которые вызывают другие подпрограммы. Пусть основная программа вызвала подпрограмму А. Она в свою очередь обратилась к подпрограмме В. Если адрес возврата для подпрограммы А хранится в регистре процессора, то куда размещать адрес возврата при вызове подпрограммы В?

Понятие стека

- Большинство ЭВМ используют **аппаратно** поддерживаемую структуру данных, называемую стеком. **Стек** — это структура данных, организованная по принципу: последним вошел — первым вышел, т.е. последние записанные в стек данные извлекаются из него первыми.
- В переводе с англ. **stack** — стопка. Аналогом стека может служить стопка тарелок. Положить тарелку в стопку можно только сверху, извлечь без проблем опять-таки только верхнюю тарелку.

Организация стека

1. В ЭВМ для организации стека выделяется область оперативной памяти, а для ее адресации и доступа к стеку используется регистр— указатель стека.
2. **Регистр -указатель стека** хранит адрес ячейки памяти, содержащей последнее помещенное в стек значение.
3. При записи числа в стек указатель стека модифицируется так, чтобы он указывал на следующую свободную ячейку ОП, и в нее записываются данные.

4. При извлечении из стека данные считываются из той ячейки ОП, на которую показывает указатель, затем указатель стека модифицируется так, чтобы указывать на предпоследнее сохраненное в стеке значение.
5. Обычно стеки растут в сторону уменьшения адресов, т.е. при записи числа указатель стека уменьшается, при извлечении числа из стека — увеличивается.

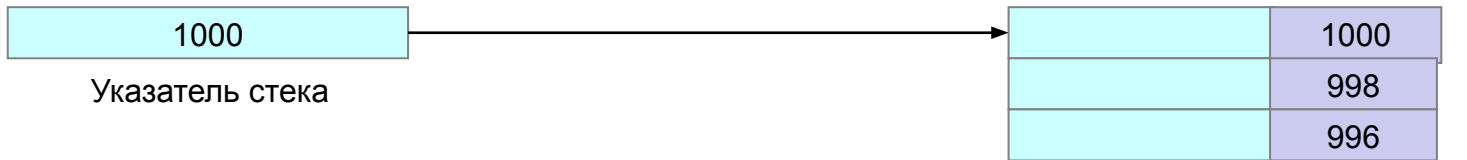
Команды для работы с подпрограммами. Стеки

Стек = ячейки ОП + регистр - указатель стека (АЛУ ЦП)

Регистр - указатель стека хранит адрес ячейки ОП, в которой содержится последний помещенный в стек адрес возврата

Структура данных стека на примере

А) Начальное состояние: стек пустой



Б) В стек записаны два адреса возврата: 1234 и 5678



В) Из стека извлечен один адрес, последний



Работа команды вызова подпрограмм **CALL <адрес>**

1. Когда процессор считывает из памяти команду **CALL <адрес>**, программный счетчик увеличивается и показывает на команду, следующую за командой **CALL**. То есть программный счетчик теперь содержит адрес возврата, с которого должно продолжиться выполнение основной программы после окончания работы подпрограммы.
2. При выполнении обращения к подпрограмме процессор сохраняет содержимое программного счетчика в стеке, точнее, в его ячейках ОП.
3. Далее в программный счетчик загружается *адрес* команды, с которого начинается подпрограмма. Процессор приступает к выполнению подпрограммы.

Работа команды возврата **RETURN**

1. Для возврата из подпрограммы в основную программу служат *команды возврата* **RETURN**.
2. Команда возврата из подпрограммы извлекает из стека сохраненный в нем адрес возврата помещают его в программный счетчик.
3. Процессор приступает к выполнению основной программы.
4. Если имели место несколько вложенных вызовов подпрограмм, то возврат произойдет по адресу возврата, сохраненному после последнего вызова, (так как для хранения адресов возврата используется стек и последний сохраненный адрес возврата будет вызван первым).

Прочие команды ЭВМ

- В ЭВМ могут быть дополнительные (специальные) команды.
- К их числу можно отнести команды остановки центрального процессора, сброса внешних устройств, установки или сброса отдельных признаков и т.д.
- Итак, на этой лекции были кратко рассмотрены базовые команды, используемые в типичных ЭВМ, и действия реализуемые этими командами.