1. Организация подпрограмм БЭВМ. Команды вызова подпрограммы и возврата

Вызов подпрограммы осуществляет команда **CALL M** – переход к подпрограмме по адресу M и запись адреса возврата в след. ячейку стека. Работает как FILO (стопка книг).

Возврат – команда **RET** – переход по адресу возврата из стека и SP+1.

1. Аргументы и возвращаемые значения подпрограммы. Способы организации передачи аргументов и возвращаемых значений.

Передача аргументов идет через стек. Запомните **главное правило** работы со стеком в подпрограммах: Перед вызовом CALL кладем на стек все нужные аргументы. После возврата надо вернуть SP туда, где он был до вызова CALL (команда POP)

**Положить на стек** **– PUSH**. Записывает в следующую ячейку стека аргумент.

**Снять со стека – POP** – читает по адресу SP аргумент из стека и делает SP+1. Для удобства есть SWAP – обменять вершину стека и AC.

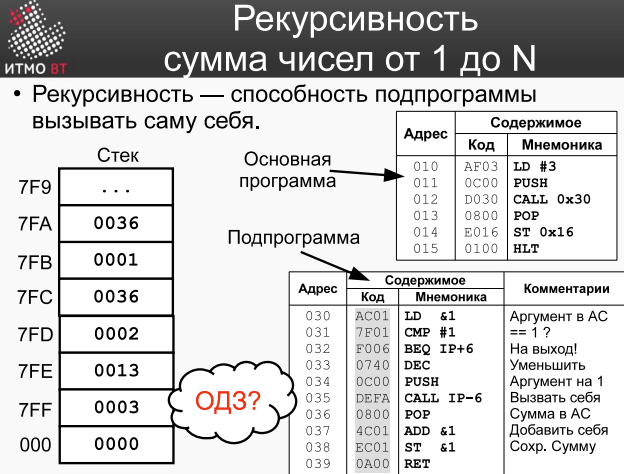
Внутри самой подпрограммы доступ к значениям в стеке осуществляется “**Адресацией со смещением SP**”. При этом, последний аргумент, положенный до вызова CALL имеет смещение &1. Смещение &0 укажет на адрес возврата из подпрограммы.

Следует помнить, что передача параметров возможна не только лишь через стек, но и через регистр общего назначения. В БЭВМ он единственный – АС (юзаем LD ST).

1. Рекурсивный вызов подпрограмм. Организация стека.

Подпрограмма может вызывать сама себя. При этом подпрограмма будет выполнена столько раз, сколько она вызвала сама себя.

Стек будет организован так: Вспомним **главное правило** перед каждым вызовом CALL мы кладем аргументы на стек, а сверху кладем адрес возврата. Повторяем нужное кол-во раз. При завершении подпрограммы – возвращаем SP, где он был до вызова CALL.



1. Описание команд CALL и RET: имя, назначение, тип команды, адресация. Кол-во и название машинных циклов, потактовое выполнение команды, кол-во обращений к памяти.

CALL

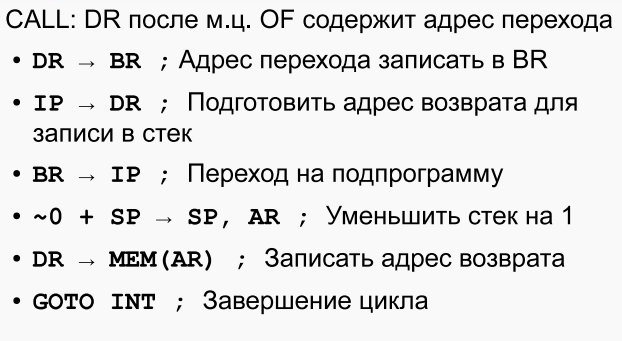
1) Имя – вызов подпрограммы

2) Назначение – переход к подпрограмме с записью адреса возврата в стек

3) Тип – Адресная

4) Циклы: IF AF OF Execution

5)



RET

1) Имя – выход из подпрограммы

2) Назначение – Переход к ячейке после CALL и SP+1

3) Тип – Безадресная

4) Циклы: IF Execution

5)

