**Устройство БЭВМ.**

Команды эвм, которые используются в лабораторной работе №2 делятся на 2 типа(всего их 4): адресные и безадресные. Адресные команды указывают на какую-то определенную ячейку памяти и несут в себе код операции. Всего есть 3 типа адресных команд, но нас интересуют команды с прямой абсолютной адресацией, которые мы и применяем. 

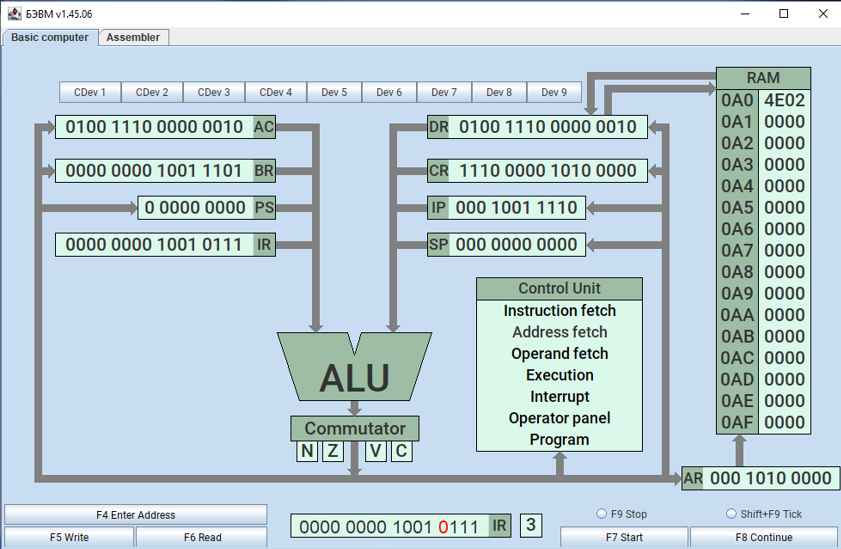
Безадресные команды состоят из команды операции равной 0+расширения кода операции.





Запустив в командной строке **java -Dmode=decoder -jar bcomp-ng.jar** мы увидим как БЭВМ выполняет те или иные команды(в том числе команды с пульта управления, такие как enter address, write, read). По сути, чтобы выполнить лабораторную работу мы должны запустить БЭВМ, ввести первый адрес из варианта, нажать enter address и далее остаётся только вводить значения ячеек по порядку. Когда мы ввели все значения, нужно опять ввести адрес, но уже той команды, с которой начинается программа(помечено +) и запустить её(не забыв нажать на нужные галочки).

Что же за непонятные буквы изображены на следующем изображении?



У коммутатора стоят флаги **N, Z, V, C**. По сути это арифметический регистр флагов.

**N или Sign Flag** - флаг знака, который хранит бит признака отрицательного числа.

**Z или Zero Flag** - флаг нуля, устанавливается при нулевом значении всех битов буферного регистра.

**V или Overflow Flag** - флаг переполнения, устанавливается в командах сложения и вычитания, если результат не помещается в формате, при этом и операнды и результат интерпретируются как знаковые числа.

**C или Carry Flag** - флаг переноса, в котором фиксируется перенос из старшего разряда при сложении и заём в старший разряд при вычитании.

Далее, рассмотрим регистры, в которых хранятся пока непонятные нам единицы и нули.

Наименования и назначения регистров/функциональных блоков в БЭВМ:

Память состоит из 2048 ячеек, каждая ячейка занимает 16 разрядов. Для обращения к памяти существует два регистра:

**AR** (Address Register) – 11-разрядный регистр адреса, в который нужно поместить адрес прежде, чем обратиться к памяти;

**DR**(Data Register) – 16-разрядный регистр, предназначенный для чтения или записи данных в\из ячеек памяти. Чтение данных и запись данных реализуется по шинам, которые подключаются к ячейке памяти.

**IP**(Instruction Pointer) – 11-разрядный регистр, который хранит в себе адрес следующей исполняемой команды.

**ALU**(Arithmetic-n-Logic Unit) – Арифметико-логическое устройство может выполнять несколько операций: сложение, логическое умножение, инверсия и прибавление единицы.

**BR**(Buffer Register) – 16-разрядный регистр, который используется для организации промежуточного хранения данных во время работы.

**CR**(Command Register) – используется для хранения кода команды и декодирования операций.

**AC**(ACcumulator). БЭВМ относится к ЭВМ, которые аккумуляторного типа, где все вычисления с данными производятся через этот регистр.

**SP**(Stack Pointer) – 11-разрядный регистр, указывающий всегда на вершину стека – особого участка памяти, который предназначен для хранения адресов возвратов и параметров подпрограмм и прерываний.

**IR**(Input Register) – находится в составе пульта оператора ЭВМ и предназначен для ввода адреса программы, кодов программы, данных запуска программы на выполнение и управления режимами работы БЭВМ.

**PS**(Program State) – хранит биты, управляющие работой БЭВМ(работа, прерывание и пр.) и признаки результата.

По сути Дата регистр и Адресный регистры служат для работы с памятью. В первый мы выгружаем содержимое ячейки памяти, к которой мы обратились с помощью адресного регистра до этого.

**Как работает команда ввод адреса?**

1. По сути, мы через регистр ввода(IR, да-да та самая строчка, где мы вводим все числа) передаём по цепочке ALU - Commutator-IP(счётчик команд) адрес следующей ячейки, к которой мы будем обращаться.

**Как работает команда записать?**

1. Когда мы передали через IR в счетчик команд адрес, в который мы будем записывать какое-то значение, значение IP проходит через ALU - Commutator->AR;
2. Далее, берётся значение, которое мы заранее указали в IR, проходит через ALU - Commutator -> DR;
3. На предпоследнем шаге выполняется сразу несколько действий одновременно:
4. DR -> RAM и AR -> RAM;
5. IP -ALU- Commutator(значение+ 1) -> IP;

4) PS(регистр с битами работы) передаёт БЭВМ, что нужно остановить выполнение и вернуться к вводу из пульта;

**Как работает команда Start?**

1. Команда просто обнуляет все флаги, DR, CR, SP, AC, BR, AR;
2. Возвращает БЭВМ в состояние приёма команд;

**Как работает команда чтения?**

1. Значение IP передаётся в AR как и в командах ранее;
2. Значение AR передаётся в RAM, и из ячейки под номером, указанным в AR, выгружается значение в DR. Одновременно с этим значение IP проходит круг IP - ALU(увеличивает значение на 1) - Commutator -> IP;
3. PS(регистр с битами работы) передаёт БЭВМ, что нужно остановить выполнение и вернуться к вводу из пульта;

*Также, стоит рассмотреть машинной команды в МПУ - цикла(Микропрограммное устройство управления - простейший компьютер, программа которого состоит из микроопераций).*

*Цикл команды разбит логически на 5 циклов:*

• **Цикл выборки команды** – осуществляет загрузку исполняемой команды в регистр команды и частичное её декодирование.

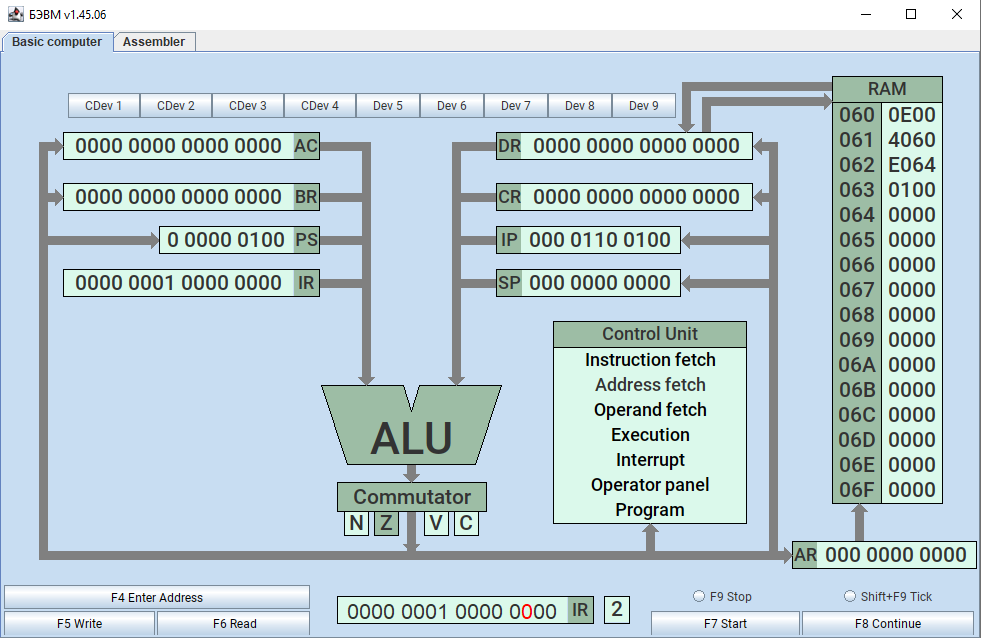
• **Цикл выборки адреса** – обрабатывает адресные команды и выборки адреса операнда с учётом режимов адресации.

• **Цикл выборки операнда** – для тех команд, где это необходимо, размещает в DR второй операнд команды. Первым операндом является аккумулятор.

• **Цикл исполнения** – производит исполнение команды.

• **Цикл прерывания** – цикл выполняется в том случае, если разрешены прерывания и устройство ввода-вывода готово к обмену.

Рассмотрим пару команд и их выполнение.

Возьмём ячейку 060, в которой у нас будет лежать какое-то значение. Программа пусть начинается с ячейки 061, а результат пусть записывается в ячейку 064(063 останавливает программу)

Мы исполняем функцию A+AC и записываем результат в ячейку.

1. Регистр битов работы передаёт через ALU - Commutator - Управляющему блоку команду, что БЭВМ начинаем выполнять выборку команды.
2. Значение ячейки из счётчика IP передаётся через ALU - Commutator -> AR и одновременно с этим значение IP проходит в BR.
3. Далее, AR передаёт в RAM номер ячейки, к которой мы обращаемся командой, и RAM выгружает в DR значение это ячейки. Одновременно с этим заканчивается круг BR - ALU - Commutator - IP, этот “круг”(или точнее “цикл”) увеличил IP на 1.
4. Выгруженное из памяти в DR значение ячейки передаётся в DR - ALU - Commutator -> CR(Command Register), где частично декодируется команда.Всё ещё идёт цикл выборки команды.
5. Значение CR передаётся в управляющий блок.
6. Начинается цикл выборки операнда.
7. Значение из DR передаётся в AR.
8. Значение из RAM выгружается в DR из адреса AR.
9. CR переходит в CR - ALU - Commutator - Control Unit, цикл меняется на цикл исполнения.
10. Значения AC и DR складываются и передаются в AC + DR - ALU - Commutator - AC.
11. PS передаёт в управляющий блок, что пора перейти к циклу прерывания.
12. ...

Это был лишь кусок программы, который прошёлся по ячейке 061, осуществляющей сложение.