

## **Раздел 2. От первых ЭВМ к суперЭВМ и начало эпохи программирования .**

### **2.1. Основы конструкции первых ЭВМ и принцип программного управления ЭВМ.**

Почему важно рассмотреть конструкцию первых ЭВМ!?

Это принципиально важно знать постольку на их идейной основе построены все современные ЭВМ и суперЭВМ.

Основными элементами первых ЭВМ были:

- - арифметическое и логическое устройство;
- - оперативная память;
- - долговременная память;
- - каналы, связывающие все компоненты в единое целое;
- - периферийные устройства (ввод и вывод данных.)

Остановимся подробнее на каждом из этих устройств, т.к. это принципиально для последующего.

# Арифметическое и логическое устройство

Сегодня это устройство принято называть – **процессором**.

Его назначение – выполнение арифметических и логических операций.

История развития процессоров сама по себе весьма увлекательна и представляет яркий пример совершенствования технологий с середины XX века по начало века XXI. Первые процессоры имели производительность до тысяч операций в секунду и были построены на лампах и диодах. Сегодняшние микропроцессоры развиваются в очень широком диапазоне направлений - от специальных до массовых, которые имеют, как правило, многоядерный характер, т.е. сам процессор, - состоит из системы подпроцессоров. Например, широко используемый в суперЭВМ процессор - *Six-Core AMD Opteron™ (Istanbul)* является 6 ядерным. Но, об этом далее.



# Оперативная память или **ОЗУ** (Оперативное Запоминающее Устройство)

Её ключевое назначение обмен данными с процессором.

В первых компьютерах использовались различные физические принципы для реализации оперативной памяти (мы об этом говорили). При этом их емкость была весьма малой – от нескольких десятков у ЭВМ МЭСМ (1951 г.) - ОЗУ 94 16-разрядных (битных) слов, до 4096 (4 Килобайт) 45-разрядных слов М-20 (1958 г.).

Принципиально важно понимать, что начиная с первых компьютеров, ОЗУ всегда имели и имеют до сего дня (за редким исключением специальных систем) важнейшее назначение - хранения команд – инструкций и данных для решения задачи. При этом всю совокупность команд – инструкций стали называть **программой** для данной задачи.

Напомним, что как физическое устройство каждый бит может иметь только два состояния, одно из которых отождествляется с хранением 0 другое с хранением 1. При этом основной, элементарной составляющей, оперативной памяти является байт (byte), состоящий из восьми битов, - наименьших возможных хранителей информации (рис 2).

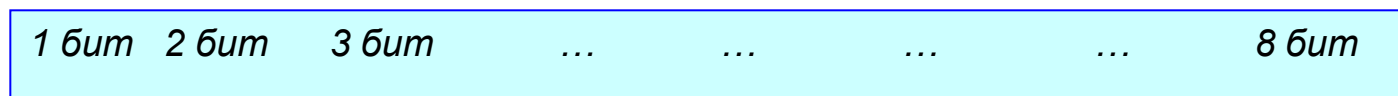


Рис. 2

Бит и байт хранители очень маленьких объёмов информации. Более крупными являются 1К (Килобайт)=1024 байтам, 1М (Мегабайт)=1024К, 1Гбайт (Гигабайт)=1024М, и, наконец, 1Тбайт (Терабайт)=1024Г. Чтобы Вы почувствовали, что же такое, например, 1Мегабайт, мы возьмем книгу объёмом примерно в 1000 страниц с достаточно плотным шрифтом, - вот это и есть примерный эквивалент одного мегабайта информации. Для хранения данных, как правило, используют несколько подряд расположенных байтов, в частности, **слово** - отвечает четырем последовательным байтам, полуслово – двум, **двойное слово** – восьми. Одно слово позволяет хранить восьмизначное число.

## *Долговременная память.*

Начиная с самых первых машин, долговременную память выполняли в виде специальных накопителей – магнитных барабанов и магнитных дисков. Эта технология сохраняется и до сего дня, конечно, достигнув очень большого совершенства. В Ваших домашних компьютерах дисковые накопители имеют объемы сотни Гбайт.

Нам же важно понимать, что долговременная память это хранилище не оперативной информации, то есть для подавляющего числа ЭВМ данные в долговременную память и из нее поступают только из ОЗУ.

## *Каналы, связывающие все компоненты ЭВМ в единое целое.*

По своему назначению каналы являлись предтечей того, что сегодня называют «материнской платой» у персональных компьютеров.

## *Периферийные устройства (ввод и вывод данных).*

Думается здесь все понятно. Но не нужно думать, что всегда все было, так как сегодня. Эпоха перфокарт и быстродействующих печатающих устройств, которые выводили на бумажную ленту 1 (!) колонку из 12 цифр была всего лишь 40 лет тому назад. Дисплеи же для машин ЕС ЭВМ были «прорывом» всего лишь 20 лет тому назад.

# Теперь мы можем привести функциональную схему первых ЭВМ

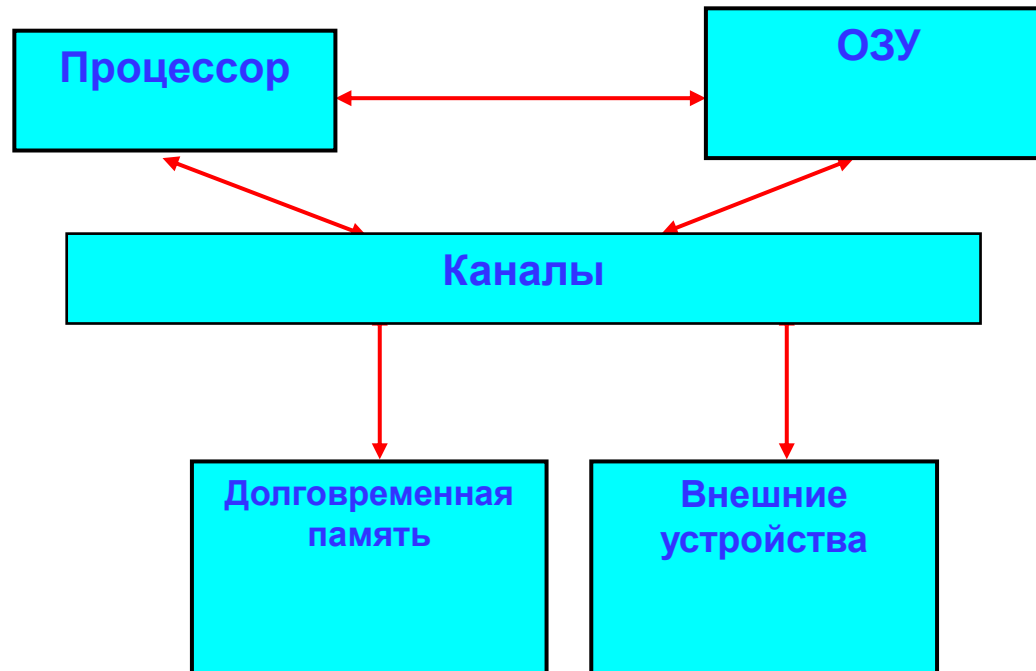


Рис. 3. Функциональная схема первых ЭВМ.

Все множество связующих магистралей ЭВМ обозначено **красными линиями**, причем стрелки указывают направление передачи информации.

Конечно, схема, приведенная на рис 3 довольно общая, и не отражает большого числа деталей, характеризующих сложное устройство ЭВМ. Тем, не менее, она дает нам возможность понять и проиллюстрировать важнейший принцип – **принцип программного управления ЭВМ**. Его суть в следующем:

одновременно компьютер выполняет команду за командой, или используя принятую сегодня терминологию команды идут одним «поток» (***Single Instruction***) и используют один поток данных (***Single Data***). При этом и команды и данные расположены в одном месте компьютера – в оперативной памяти ЭВМ.

**2.2. Начало эпохи программирования- программирование на языке ЭВМ (в кодах) и появление языка *Fortran* – начало эпохи языков программирования и алгоритмических языков. Последовательный характер программирования.**

Операционные системы *Windows – 2007* или другие *Windows*, с которыми Вы знакомы неперемный атрибут каждого персонального компьютера. Эти, очень мощные по возможностям и очень большие по объёму, занимающие сотни мегабайт, программы, можно назвать программами – оболочками.

Однако, в той луковице, которой для пытливого глаза представляет программное обеспечение Вашего персональный компьютер, они являются самым внешним «слоем». Под этим «слоем», как правило, имеется ещё несколько «слоёв-уровней» и самый первый из них это уровень команд микропроцессора. Именно из этих команд и состоит в конечном итоге каждая из программ выполняемых на компьютере. В одной программе может быть несколько тысяч или даже миллионов команд.

Сейчас же нам важно проследить обратную тенденцию, а именно как десятилетиями лучшие математики и программисты мира «наращивали» слои упомянутой «луковицы».

Или, говоря современным языком, посмотрим, как развивалось программирование от уровня команд-инструкций первых ЭВМ, до уровня современных мощных системных и прикладных программ для суперЭВМ. Для этого мы вновь должны вернуться на более чем 50 лет назад, обратившись к функциональной схеме первых ЭВМ приведенной на рис. 3.

Итак, в конце предыдущего пункта мы назвали **принцип программного управления ЭВМ** важнейшим.

Переходя к его технической сути, покажем, что это действительно так. Поясним это на примере машины БЭСМ-2М, на которой мне удалось работать в 1969-1971 годах. ОЗУ этой ЭВМ состояло из 2048 слова, каждое размером в 41 разряд (бит). При этом, все слова, которые именовались тогда ячейками, имели свою нумерацию от 0000 до 2047, которые называли адресами ячеек.



Выше мы также говорили о том, что в оперативной памяти хранятся как данные задачи (числа необходимые для вычислений), так и команды-инструкции, реализующие процесс вычислений, то есть **программа хода вычислений**. Отсюда и пошло наименование **программа!**

То есть процессор руководствуется в своих действиях, инструкциями – командами, «записанными» в ОЗУ. В этом и состоит **принцип программного управления ЭВМ**.

Но здесь не все так просто. Упомянутая машина БЭСМ-2М относилась к числу так называемых 3-х адресных ЭВМ. Это значило, что в процессе выполнения одной команды процессор обращался к трем ячейкам оперативной памяти машины. Например, большинство команда имела такой вид:

Адрес	КОП	A1	A2	A3
-------	-----	----	----	----

где

- Адрес – номер той ячейки ОЗУ где записана данная команда;
- КОП – номер выполняемой операции, например 001 – сложение, 002 – вычитание и т.д.;
- A1, A2 и A3 – адреса (номера) тех ячеек с которыми «работает» данная команда.

## Например, команда

1023	001	0242	0351	0101
------	-----	------	------	------

записанная в ячейке ОЗУ 1023 означает сложение чисел из ячеек 0242 и 0351 и размещение итоговой суммы в ячейке с номером 0101.

Здесь приведен пример 3-х адресной машиной, какой была БЭСМ -2М. Были и 2 – х и 1 – о адресные машины:

КОП	A1	A2	A3
-----	----	----	----

КОП	A1	A2
-----	----	----

КОП	A1
-----	----

Изображенные схемы команд отвечают (сверху вниз) трех, двух и одноадресным типам команд. Сегодня, как правило компьютеры имеют команды переменной адресности, то есть в процессоре используются все типы команд приведенные на схеме.

# А теперь задумаемся над вопросом. Какого труда стоит написать программу на таком внутреннем языке команд компьютера?

Авторы первых программ выписывали на бланках тысячи команд в виде чисел, отвечающих как командам, так и адресам оперативной памяти. Представьте себе среднюю программу размером в 1000 трехадресных команд! Это же

$$1000 \times (4 + 3 + 12) = 19\,000 \text{ знаков!}$$

Даже самый опытный и аккуратный человек допустит ошибки, но ведь еще нужно переписать (закодировать) это на перфокарты. Кстати попробуйте переписать из любой таблицы подряд двадцать восемнадцатизначных чисел! И потом проверьте с друзьями ...

Понятно что такое программирование

- трудоемко,
- ненаглядно,
- порождает ошибки.

А ведь именно с этого и начиналось программирование!

## Язык программирования **Fortran** – начало эпохи языков программирования и алгоритмических языков

На первых этапах внедрение ЭВМ тормозилось значительными трудностями в написании программ «в кодах», т.е. на языке команд ЭВМ, Трудности и проблемы с программированием в кодах заставили ученых и инженеров задуматься над тем, - как облегчить процесс программирования? Как переложить всю рутинную работу на «плечи» ЭВМ?

Схема по которой здесь шла мысль исследователей такова:

- любая математическая проблема (а только такие и рассматривались на первых этапах внедрения ЭВМ) может быть хорошо формализована;
- мы знаем все команды наших компьютеров;
- а почему бы не построить такую программу, которая на основе нашей формально записанной задачи строила бы программу на языке команд ЭВМ?

Но предварительно нужно записать на каком то близком к обычному математическому языку нашу задачу.

Такие языки, получившие наименование языков программирования начали создавать в самом начале 50 – годов. Здесь были и интересные отечественные разработки, но наибольшее распространение получил язык **FORTTRAN** (Formula Translation). Названные выше программы, - программы – переводчики стали называть трансляторами (в дальнейшем компиляторами, когда к ним добавились еще ряд функций).

Создание языка **FORTRAN**, породило «эпоху» проблемно ориентированных языков:

**ALGOL 60, ALGOL 68** – алгоритмические языки, построенные на основе математической лингвистики были ориентированы в основном на очень широкий круг вычислительных задач;

**PASCAL** – язык программирования, ориентированный на широкий круг задач;

**COBOL** – язык который был создан для работы с большими массивами экономических данных;

**LISP** - язык созданный для работы с большими списками, и т.д.

Любопытно, что сам «прародитель» **FORTRAN** претерпел массу модификаций от **FORTRAN** к **FORTRAN 4** от **FORTRAN 77** к **FORTRAN 90** и, наконец к параллельной версии **FORTRAN**, об этом немного далее.

Сегодня многие пользуются языкам **C++**, впитавшим в себя все сильные стороны многих, языков, начиная с **FORTRAN**.

## Естественно возникает вопрос, какое отношение все сказанное имеет к суперкомпьютерам?

Косвенно, конечно, понятно, что имеет! И с точки зрения структуры ЭВМ, и с точки зрения программирования. Но все-таки какое?

**Первое**, что здесь необходимо отметить, это важность знания основ структуры первых ЭВМ, на которых строились и первые суперкомпьютеры.

**Второе**, это, безусловно, принцип программного управления, который в первых компьютерах был **последовательным**. То есть команды выполнялись одна за другой (это, конечно, не отменяет того, что имелись команды переходов от одной к другой, так сказать скачком). Но при этом в первых компьютерах всегда решалась шаг за шагом (т.е. команда за командой) **только одна задача** (вспомним - команды идут одним «потокком» (***Single Instruction***) и используют один поток данных (***Single Data***)! Т.о., используя сегодняшнюю терминологию первые ЭВМ были *однозадачными*.

В современных суперкомпьютерах реализуются уже другие подходы и идеи.

## 2.3. Понятие операционной системы

Однако, мы рассмотрим еще ряд вопросов без которых невозможно понимание принципов применения и понимания работы современных суперкомпьютеров. Важнейшим из них является вопрос об **операционных системах**, который мы уже упоминали в начале данного параграфа, проводя параллель современной ЭВМ с луковицей. **Операционные системы** (ОС) *Windows – 2007* или другие *Windows*, с которыми Вы знакомы это управляющие работой нашего компьютера программы (программные системы).

Что они делают? Или каково их назначение?!

Когда мы пишем программы для решения наших задач, мы должны заботиться и о текущих проблемах, возникающих при их выполнении. Это проблемы ввода и вывода как исходных, так и итоговых данных, проблемы выделения при необходимости дополнительной оперативной памяти или памяти на дисках и так далее. Команды, выполняющие все эти по своей сути сугубо вспомогательные операции, могут занимать не одну сотню команд процессора и являются для нас рутинными. Операционные системы уровня *Windows* избавляет нас от этой рутины, выполняя многочисленные вспомогательные операции в процессе работы компьютера.

Но это все РУТИНА!

Гораздо более важным было «научить» ОС управлять компьютером более эффективно, например, что бы она могла одновременно обеспечивать прохождение через него нескольких задач!

Например, одна задача «считается» другая «вводит» свои исходные данные, третья, закончившаяся «выводит» результаты и т.д. Но при этом все они находятся в одной оперативной памяти (так было на первых порах).

Первой широко распространенной подобной системой была OS – 360, разработанная фирмой IBM (это 70-е годы прошлого века). В ней, например, были модули с такими характерными названиями – планировщик задач и диспетчер...

Современные ОС кардинально отличаются от своих прародителей. Таких отличий весьма и весьма много, но мы назовем одно важнейшее отличие – **многозадачность**. То есть сегодня практически все ОС позволяют на одном процессоре «решать» несколько задач одновременно. Как это реализуется мы увидим далее.

Конечно, все современные ОС - это системы, построенные с учетом кардинально изменившейся архитектуры современных компьютеров в сравнении с первыми ЭВМ. Об этом далее поговорим поподробнее.

Заметим, что здесь мы впервые столкнулись с некоторыми началами **параллельной** работы ЭВМ, где **многозадачность** есть простейшее проявление параллелизма в работе компьютера.



## 2.4. Типы структуры современных ЭВМ. От ЭВМ к СуперЭВМ.

К тому что представляет собой современный компьютер и суперкомпьютер мы будем подходить с разных сторон на протяжении всего нашего курса. При этом укажем, что важнейший принцип реализуемый в современных ЭВМ, - **это параллельность их архитектуры.**

Сейчас же мы остановимся на существующих сегодня наиболее распространенных архитектурах (структурах) компьютеров и перейдем от них к суперкомпьютерам.

Практически все современные компьютеры, из интересующей нас сферы инженерных и естественнонаучных исследований, подразделяются на два класса в зависимости от организации работы с оперативной памятью.

**Системы с разделяемой памятью**, у которых имеется **одна общая память** и все **процессоры** (подчеркнем, что их может быть несколько в одной вычислительной системе) **имеют одинаковый доступ к данным и командам**, хранящимся в этой памяти. Это - **UMA (Uniform Memory Access)** системы. На этой основе строятся так называемые векторные параллельные процессоры **PVP (Parallel Vector Processor – работающие с векторами)** и симметричные мультипроцессоры **SMP (Symmetric MultiProcessor)**, симметрия здесь подразумевает одинаковость условий доступа к памяти всех процессоров.



Второй тип это -

**системы с распределенной памятью**, у которых каждый процессор имеет свою собственную оперативную память, к которой у других процессоров доступа к ней нет.



В системах **с распределенной памятью** объединяющая **коммуникационная сеть** есть важнейший элемент вычислительной системы.

Если в **системах с разделяемой памятью** все задачи «живут» в рамках одной общей оперативной памяти, то в системах **с распределенной памятью** объединяющим началом для задач служит именно **коммуникационная сеть** позволяющая «собрать» все данные нашей задаче в требуемом месте.

Указанные два типа организации памяти могут быть реализованы в различных архитектурах компьютеров и служат основой архитектуры большинства современных суперЭВМ.

Рассмотрим различные классификации современных компьютеров, указывая там, где это существенно, способ организации оперативной памяти.

Исторически наиболее ранней является классификация *М. Флинна* (1966).

Построенная им классификация достаточно условна и основана на понятии *потока*, под которым, как мы уже отмечали, понимается *последовательность команд* или *данных*, обрабатываемых процессором (процессорами).

На основе числа потоков команд и потоков данных выделяют следующие четыре класса архитектур:

**SISD** (Single Instruction stream/Single Data stream) – один поток команд и один поток данных;

**SIMD** (Single Instruction stream/Multiple Data stream) – один поток команд и множество потоков данных;

**MISD** (Multiple Instruction stream/Single Data stream) – множество потоков команд и один поток данных;

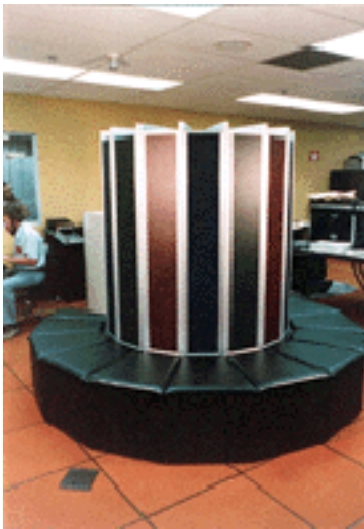
**MIMD** (Multiple Instruction stream/Multiple Data stream) – множество потоков команд и множество потоков данных.

С **SISD** мы уже встречались, говоря о первых ЭВМ, именно этот тип архитектуры и был в них реализован.

Технология **MISD** определена скорее по формальному признаку (для завершенности классификации архитектур), т. к. такие системы практически не разрабатываются.

Отметим, что технология **SIMD** широко очень использовалась при разработке наиболее мощных машин в 70 – 80 годы.

К их числу относились машины CDC 6600 и CDC 7600 компании Control Data Corporation машина ILLIAC IV (1974) и CRAY (1976), «гремевшие» как наиболее мощные в 70 – х годах прошлого века.



**CRAY**  
**(1976 г)**

Пиковая производительность **160 миллионов операций в секунду**, оперативная память до 1Мслова (слово - 64 разряда). Главное новшество – векторные команды, работающих с целыми массивами независимых данных и позволяющие эффективно использовать конвейерные устройства, т.е. по сути, работу с массивами.