

[===== Лекция № 01: «Системы счисления» =====]

Системы счисления:

2-я использует цифры: 0,1 - проще для ЭВМ

10-я использует цифры: 0,1,2,3,4,5,6,7,8,9 - проще для человека

16-я использует цифры: 0,1,2,3,4,5,6,7,8,9, A, B, C, D, E, F - для удобства общения ЭВМ и человека

2-я:	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
10-я:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16-я:	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Перевод чисел в разные системы счисления:

1) $???_{10} \rightarrow ???_n$ - деление столбиком (делим, затем выписываем остатки в обратном порядке)

$125_{10} \rightarrow ???_7 = 236_7$ $125_{10} \rightarrow ???_{16} = 7D_{16}$ (13 – смотрим таблицу – цифра D)

$$\begin{array}{r} 125 \overline{) 7} \\ \underline{119} 7 \\ 6 14 2 \\ \underline{3} \end{array}$$

$$\begin{array}{r} 125 \overline{) 16} \\ \underline{112} 7 \\ 13 \end{array}$$

Деление происходит до тех пор, пока не получится результат, меньший n - основание системы счисления. Если остаток получился >10 – смотрим нашу таблицу.

2) $???_n \rightarrow ???_{10}$

7 6 5 4 3 2 1 0

$00100101_2 \rightarrow ??_{10}$ — нумеруем разряды по старшинству (с 0-го по 7-й)

$$0 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 2^5 + 2^2 + 2^1 = 32 + 4 + 1 = 37_{10}$$

1 0
 $7D_{16} \rightarrow ??_{10}$

$$7 \cdot 16^1 + D \cdot 16^0 = 7 \cdot 16 + 13 \cdot 1 = 112 + 13 = 125_{10} \text{ — верно, проверьте на калькуляторе}$$

3) $???_n \rightarrow ???_m$



Из произвольной в произвольную - за 2 шага, через промежуточную 10-ю!

4) $????_2 \leftrightarrow ?_{16}$ - быстрый перевод (используется, что $16 = 2^4$)

Каждому шестнадцатеричному разряду строго соответствует четыре двоичных (и наоборот)!

$101100_2 \rightarrow _ _ 101100_2 \rightarrow 00101100_2 \rightarrow 2C_{16}$ — разбивка на четверки, дополняем старшие нули и замена по таблице

$6F5_{16} \rightarrow 011011110101_2 \rightarrow 11011110101_2$ — замена по таблице (1 цифра = 1 четверка бит) и убираем старшие нули

Простейшие арифметические операции в 2-й системе:

1) Сложение: $5_{10} + 7_{10} = 12_{10}$

$$5_{10} \rightarrow 00000101_2$$

$$7_{10} \rightarrow +00000111_2$$

$$\underline{00001100_2}$$

<<< — переносы в старший разряд

2) Умножение:

При сдвиге двоичного числа на 1 разряд влево, оно удваивается.

При сдвиге двоичного числа на n разрядов влево, оно умножается на 2^n .

Вправо - происходит целочисленное деление (с отбрасыванием остатка).

$$7_{10} * 5_{10} = 111_2 * 101_2 = 111_2 * (2^2 + 2^0) = 11100_2 + 111_2 = 00100011_2 = 2^5 + 2^1 + 2^0 = 32 + 2 + 1 = 35_{10}$$

3) Вычитание: $7_{10} - 5_{10} = 7_{10} + (-5_{\text{дополнительный код}}) = 2_{10}$

$$5_{10} \rightarrow 00000101_2 \rightarrow |$$

$$11111010_2 \leftarrow | \quad \text{— инверсия}$$

$$+ \quad 1_2$$

$$\begin{array}{r} \text{-5}_{\text{доп}} \quad 11111011_2 \\ 7_{10} \rightarrow + 00000111_2 \\ \hline 1 \quad 00000010_2 = 2_{10} \end{array}$$

Возникшее переполнение (9-й разряд) — отбрасывается!

$$5_{10} - 7_{10} = 5_{10} + (-7_{\text{дополнительный код}}) = -2_{10}$$

$$7_{10} \rightarrow 00000111_2 \rightarrow |$$

$$11111000_2 \leftarrow | \quad \text{— инверсия}$$

$$+ \quad 1_2$$

$$\begin{array}{r} \text{-7}_{\text{доп}} \quad 11111001_2 \\ 5_{10} \rightarrow + 00000101_2 \\ \hline \end{array}$$

$$\begin{array}{r} \text{-2}_{\text{доп}} \quad 11111110_2 \rightarrow | \\ 00000001_2 \leftarrow | \quad \text{— инверсия} \\ + \quad 1_2 \\ \hline \end{array}$$

00000010₂ — верно, совпадает

В машинном представлении чисел старший разряд (№ 7) - знаковый, показывает знак числа ("0" - плюс, "1" - минус). Если результат операции - отрицательный, значит - он записан в дополнительном коде (для проверки надо преобразовать обратно).

[== Лекция № 02: «Введение в низкоуровневое программирование» ==]

Язык Assembler – представляет собой набор элементарных команд, выполняемых процессором напрямую. С одной стороны, это гарантирует максимальную скорость работы программ, с другой – для реализации нетривиальных задач требуются значительные усилия. Кроме того, необходимо учитывать, что в разных процессорах используются разные наборы команд (которые и кодируются по-разному). Присутствует проблема переносимости ПО.

Рассмотрим краткую историю МП-техники на примере семейства Intel:

8086 - Одним из первых был процессор 8086. Он был 16 битным, мог работать с 1 Мб ОЗУ.

80186 - Промежуточный вариант, который быстро был вытеснен следующим представителем.

80286 - Позволял работать в 2 режимах:

- реальный режим (загружается по умолчанию)
- защищенный режим (доступна вся память – снимаются все ограничения, пользователь сам отвечает за любые свои действия)

80386 - Данная серия процессоров была полностью 32-битовой, поддерживала 3 режима работы:

- реальный режим (то, что мы будем изучать)
- защищенный режим 2 (по сравнению с предыдущ. добавлена таблица новых команд)
- виртуальный режим 386 процессора (впервые появляется мультизадачность)

Отметим, что далее повышаются характеристики процессоров: быстродействие, разрядность; добавляются новые функции. Мы остановимся именно на реальном режиме 386 процессора, поскольку он корректно воспринимается всеми последующими представителями данной серии.

Адресация памяти:

Вся память в ЭВМ представляет собой аналог одномерного массива – длинная лента, состоящая из элементарных/битовых ячеек (значения 0 или 1). 8 бит называют байтом, а 2 байта – машинным словом. Поскольку емкость памяти современных запоминающих устройств огромна, память поделена для удобства на небольшие порции – сегменты. Каждый сегмент является независимым от других, и может быть использован для загрузки программы при ее выполнении или для хранения каких-либо данных. В общем случае, если информация не уместится в одном сегменте, ей могут быть выделены другие. Но нам и одного сегмента хватит с избытком!

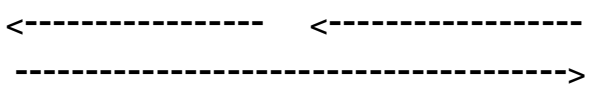
1 байт – самая мелкая ячейка памяти, которую можно адресовать с помощью стандартной системы адресации. При этом, адрес состоит из 2 частей: номер сегмента и смещение (т.е. на сколько надо внутри отступить от начала этого сегмента). Формат адреса записывается в виде восьми 16-ричных цифр, и хорошо знаком вам по сообщениям «Критическая ошибка Windows».

Пример: Адрес = Сегмент + Смещение = SEG : SME = 0012 : 4F7B

На самом деле, ЭВМ рассчитывает абсолютный 20-разрядный адрес любой ячейки по формуле $A = 2^4 * SEG + SME$, но нас это пока не будет интересовать.

Машинное представление информации в ячейке памяти:

Машинное слово:  - старшие байты идут правее младших

 старшинство разрядов в байтах
-----> старшинство байт в машинном слове

Кроме того, существуют специальные ячейки памяти, соединенные с процессором напрямую – **регистры**. У них нет адреса (находятся вне ОЗУ), обращаться к ним можно только по именам. Все операции с ними производятся в тысячи раз быстрее, чем с обычными ячейками.

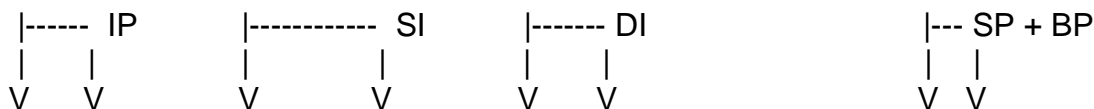
В ЭВМ 8086 - 8186 было 14 регистров. Они делятся на служебные и общего назначения. Служебные регистры: сегментные (указывают на начало сегментов в памяти) и указатели (указывают текущее положение внутри сегментов – смещение от начала). См. рисунок:

CS

DS

ES

SS



... память ...][сегмент кода][сегмент данных][экстракод – доп. сегмент][с т е к][... память ...

Сегментные регистры:

CS - сегмент кода (область памяти, куда загружен код программы)

DS - сегмент данных (область памяти, куда программа обращается за данными)

ES - экстракодовый сегмент (дополнительный сегмент, если мало двух предыдущих)

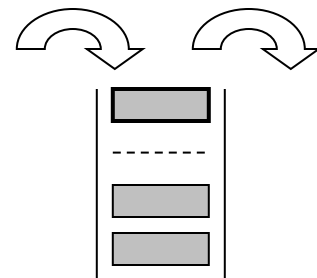
SS - сегмент стека (временное хранение промежуточных результатов – освобождаем регистры)

Стек – особая логическая структура, осуществляющая хранение данных по принципу **”последним пришел – первым ушел”** (аналогия – стакан).

В каждый момент времени доступен только один – верхний элемент. При добавлении нового все остальные как бы сдвигаются, и он становится верхним. При извлечении верхнего элемента – обратная картина: все сдвигаются обратно, верхним становится бывший второй элемент.

При работе со стеком могут произойти 2 ошибки: 1 – попытка извлечь элемент из пустого стека, 2 – попытка добавить элемент в полный стек (переполнение). Отслеживание таких ситуаций - задача пользователя.

Емкость стека – 64 Кб, размер хранимых элементов – 2 байта (16 бит).



Регистры - указатели:

IP - (в паре с **CS**) указывает на ту инструкцию, которая должна выполняться следующей, после ее выполнения автоматически сместится на длину инструкции (изменяется с шагом +n)

SP - указатель стека, (в паре с **SS**) указывает на верхний элемент стека (изменяется с шагом ± 2)

BP - указатель базы, также используется при работе со стеком

DI - Destination Index register - (в паре с **ES**) индекс приемника – использ. в строковых операциях

SI - Source Index register - (в паре с **DS**) индекс источника - используется в строковых операциях

Регистры общего назначения (доступны для решения задач пользователя):

AX (Accumulator register - аккумулятор) - обычно для хранения каких-либо промежуточных данных (используется по умолчанию при вызове прерываний и команд)

BX (Base register - база) - иногда для хранения базового адреса некоторого объекта в памяти

CX (Count register - счетчик) - в командах, производящих некоторые повторяющиеся действия (используется по умолчанию оператором цикла как параметр)

DX (Data register - регистр данных) - как и **AX**, хранит некоторые промежуточные данные (используется по умолчанию при вызове прерываний и команд)

В общем случае, эти регистры могут быть задействованы пользователем и не по назначению! Кроме того, для удобства, каждый из них разделен пополам: старшие 8 + младшие 8 разрядов. Соответственно, их имена: **AH, AL; BH, BL; CH, CL; DH, DL** (H – high - старшие, L – low - младшие). В командах языка можно обращаться как к отдельным половинкам, так и целым регистром!

Флаговый регистр **FLAGS** (задействовано 9 бит – каждый отвечает за свое событие):

Флаг (комп.) – ячейка памяти логического типа (1 бит), принимающая 1 из 2 значений: 0 или 1.

OF (Over Flow - флаг переполнения): равен 1, если в результате выполнения операции произошло переполнение в знаковый разряд. ПРИМЕР: $7FFF + 1 = 8000$ (было > 0 , стало < 0).

DF (Direction Flag - флаг направления): пока не нужен

IF (Interrupt Flag - флаг разрешения прерывания): пока не нужен

TF (Trace Flag - флаг трассировки): пока не нужен

SF (Sign Flag - флаг знака): равен 1, если результат операции < 0 (отриц.). ПРИМЕР: $01 - 02 = FF$

ZF (Zero Flag - флаг нуля): равен 1, если результат операции равен нулю. ПРИМЕР: $1A - 1A = 00$

AF (Additional carry Flag - флаг дополн. переноса): пока не нужен (в двоично-десятичном коде)

PF (Parity Flag - флаг четности): пока не нужен

CF (Carry Flag - флагом переноса): равен 1, если в результате выполнения операции результат вышел за пределы разрядной сетки (17-й разряд). ПРИМЕР: $FFFF + 1 = 1\ 0000$

[===== Лекция № 03: «Команды языка Assembler» =====]

Поскольку исполняемый код программы является последовательностью целых чисел, мало понятных для пользователя, мы будем использовать специально разработанную программу-дебаггер, позволяющую писать команды в текстовом виде (**мнемокод**), и затем преобразующую их в числовое представление (и наоборот).

Все команды Assembler-а делятся на целочисленные и нецелочисленные (с плавающей точкой). Целочисленные – выполняются процессором, их мы и будем изучать. Нецелочисленные – математическим сопроцессором, за рамками курса.

Целочисленные команды:

Поскольку Assembler отличается от других языков тем, что все его команды элементарны, и для выполнения любого действия требуется написать в программе целый блок команд, основную часть программы составляют комментарии. Иначе вы уже через 5 минут забудете, что вы имели ввиду раньше. Комментарий начинается с символа ; и продолжается до конца строки.

1) Пересылка данных:

MOV AX, BX ; AX = BX - присвоение значения (после ; идет комментарий до конца строки)
; AX - приемник (первым всегда имя переменной или регистра - туда сохр. результат)
; BX - источник (может быть как имя, так и значение - число)

MOV AH, 11 ; AH = 11 – присвоить число 11 в десятичной системе
MOV AL, 11h ; AL = 11₁₆ – присвоить число 11 в шестнадцатеричной системе (=17₁₀)
MOV BX, 002Fh ; BX = 2F₁₆ - необходимо следить за совпадением разрядности операндов
MOV CX, 1111000011110000b ; CX = F0F0₁₆ – пример присвоения двоичных чисел

XCHG AX, BX ; меняет значения местами (оба должны быть одинаковой разрядности)

PUSH AX ; помещает AX в стек (**PUSH AL** – нельзя, почему? ...*элемент стека - 2 байта*)
POP AX ; извлекает слово из стека и помещает его в AX

Примечание: если требуется записать 16-ричное число, начинающееся с буквы (F4CD), то будет сообщение об ошибке. Дебаггер подумает, что это имя переменной и не найдет такого. Поэтому надо поставить старший ноль (0F4CD), тогда все сработает как надо.

2) Арифметические операции (двоичная и десятичная арифметики):

ADD AX, BX ; AX = AX + BX - сложение (результат всегда в первом операнде - приемнике)
INC AX ; AX = AX + 1 - инкремент

SUB AX, BX ; AX = AX - BX - вычитание (результат всегда в первом операнде - приемнике)
DEC AX ; AX = AX - 1 - декремент

NEG DX ; переводит в дополнительный код и обратно (смена знака числа)

3) Логические операции (манипуляции с битами):

NOT AX ; инверсия (поразрядное отрицание); зная ее мы можем написать замену
; команде **NEG AX = NOT AX + INC AX** (но не будем, т.к. NEG - выгоднее)

MOV AX, 0097h ; присвоим в AX начальное значение 0000 0000 1001 0111₂
AND AX, 00F0h ; поразрядное наложение на AX маски 0000 0000 1111 0000₂

0000 0000 1001 0111 – значение в AX (в командах нам будет удобнее задавать 16-ричные числа)
0000 0000 1111 0000 – маска (так что не забываем правила быстрого перевода $X_2 \leftrightarrow Y_{16}$)

0000 0000 1001 0000 (в AX обнуляются биты, где в маске стоят нули, прочие - без изменений)

MOV AX, 0097h ; присвоим в AX начальное значение 0000 0000 1001 0111₂
OR AX, 00F0h ; поразрядное дополнение AX маской 0000 0000 1111 0000₂

0000 0000 1001 0111 – значение в AX (из прошлого примера)
0000 0000 1111 0000 – маска (из прошлого примера)

0000 0000 1111 0111 (в AX единицы в тех битах, где в маске стоят единицы, прочие - без изменений)

MOV AX, 0097h ; присвоим в AX начальное значение 0000 0000 1001 0111₂
XOR AX, 00F0h ; шифрование AX по маске 0000 0000 1111 0000₂

0000 0000 1001 0111 – значение в AX
0000 0000 1111 0000 – маска (чтобы расшифровать число, нужно наложить ту же маску еще раз)

0000 0000 0110 0111 (в AX инвертируются биты, где в маске стоят единицы, прочие - без изменений)

SHL AX, 1 ; сдвиг AX на 1 разряд влево (умножение на 2, старшие могут "потеряться")
SHR AX, 3 ; сдвиг AX на 3 разряда вправо (деление нацело на 8, младшие - отбрасываются)

TEST AX, 0103h ; проверяет те разряды AX, которые указаны в маске 0000 0001 0000 0011
; ZF = 0 если в AX есть хоть одна единица среди указанных, ZF = 1 если ни одной

CMP AX, BX ; анализирует разность (AX - BX), устанавливает флаги ZF и SF
; флаг знаков: SF = 0 если AX-BX ≥ 0, SF = 1 если AX-BX < 0 (0 – "+", 1 – "-")
; флаг значений: ZF = 0 если AX ≠ BX, ZF = 1 если AX = BX (разность = или ≠ 0)

Примечание: команды **TEST** и **CMP** только меняют флаги, значения операндов остаются прежними!

Пример взаимозаменяемости команд: задача - обнулить AX

1. **MOV AX, 0000h**
2. **SUB AX, AX**
3. **AND AX, 0000h**
4. **XOR AX, AX** ; наиболее быстрый способ (без лишней пересылки данных)
5. **SHR AX, 16**

4) Передача управления (безусловный и условный переход):

Мы привыкли, что в языках высокого уровня использование операторов перехода является дурным тоном. В Assembler-е же любое разветвление алгоритма (отклонение от линейности) реализуется только через подобные операторы!

МЕТКА1:
JMP МЕТКА1 ; безусловный переход к указанной метке (вверх или вниз)

Переход по условию:

JS МЕТКА ; переход происходит, если SF = 1 (буква **S** в имени команды – указывает флаг)
JNS МЕТКА ; переход происходит, если SF = 0 (буква **N** в имени команды – как бы отрицание)
JZ МЕТКА ; переход происходит, если ZF = 1 (буква **Z** в имени команды – указывает флаг)
JNZ МЕТКА ; переход происходит, если ZF = 0 (буква **N** в имени команды – как бы отрицание)

Цикл с параметром (CX – используется по умолчанию в качестве счетчика):

MOV CX, 5 ; установим количество повторов цикла
МЕТКА2:
... тело цикла ... ; блок команд, которые надо повторить в цикле (регистр CX - занят)
.....
LOOP МЕТКА2 ; если CX ≠ 0, то уменьшает CX на 1 и возвращается вверх к метке

5) Цепочечные (выполнение действий над заданной цепочкой данных): пропускаем

6) Управление состоянием ЦП:

INT <номер> ; вызов прерывания (когда обращаемся к какой-либо стандартной функции)

Пример:

; [=== Стандартное завершение программы ===]

MOV AH, 4Ch ; номер функции из библиотеки данного прерывания, которую мы хотим вызвать
; **4C** - передача управления родительской задаче (т.е. операционной системе)

MOV AL, 00h ; передаваемый в функцию параметр (выход с кодом **0**, т.е. без ошибки)
; можно было сразу присвоить **MOV AX, 4C00h**, но в две строки вам нагляднее

INT 21h ; вызов прерывания DOS – выполняется указанная функция (завершение работы)

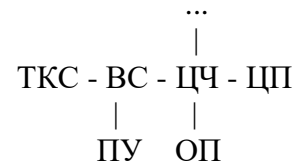
Если в конце программы не поставить такой блок команд, то дебаггер не узнает, что программа завершена, и продолжит считывать из памяти и выполнять всякий "мусор"!

Как видите, команда **INT** всегда обращается к регистру **AX**, чтобы узнать исходные данные вызова – номер библиотечной функции, передаваемый параметр. Кроме того, некоторые функции будут обращаться за данными к регистру **DX** и совсем редко – к **BX**, но об этом позже.

Лекция № 4. Введение.

Цифровые ЭВМ и ВС традиционной архитектуры.

ВС - любой набор компьютерного оборудования, которая может содержать центральную часть, а также периферию.



ЦЧ - центральная часть

ЦП - центральный процессор

ПУ - периферийные (внешние) устройства

ТКС - телекоммуникационная система

... - потенциал расширения ВС (возможность подключения доп.устройств)

Система - многомашинная многопроцессорная структура.

1941-44 - точка отсчета для истории ВС: в Гарвардском университете Эйкен Говард Хэтэуэй разработал и построил ЭВМ "Mark 1". Ее производительность была 1 операция умножения за 3 секунды!

Особенность машины: программа отсутствовала.

1943-46 - Джон Преспер Эккерт и Джон Уильям Мочли разработали компьютер "ENIAC" - первая практически полноценная машина.

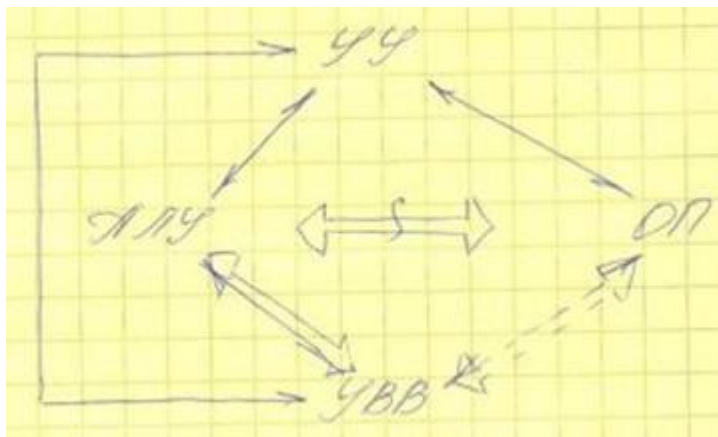
Джон фон Нейман - присоединился к проекту в сентябре 1944 года в качестве научного консультанта.

На основе анализа недостатков ЭНИАКа внес существенные предложения по созданию новой более совершенной машины - EDVAC. В процессе дискуссий о данном проекте появился сводный документ рекомендаций, получивший название «Архитектура фон Неймана».

Основные принципы фон Неймана:

1. Использование двоичной системы счисления в вычислительных машинах. Преимущество перед десятичной системой счисления заключается в том, что устройства можно делать достаточно простыми, арифметические и логические операции в двоичной системе счисления также выполняются достаточно просто.
2. Программное управление ЭВМ. Работа ЭВМ контролируется программой, состоящей из набора команд. Команды выполняются последовательно друг за другом. Созданием машины с хранимой в памяти программой было положено начало тому, что мы сегодня называем программированием.
3. Память компьютера используется не только для хранения данных, но и программ. При этом и команды программы и данные кодируются в двоичной системе счисления, т.е. их способ записи одинаков. Поэтому в определенных ситуациях над командами можно выполнять те же действия, что и над данными.
4. Ячейки памяти ЭВМ имеют адреса, которые последовательно пронумерованы. В любой момент можно обратиться к любой ячейке памяти по ее адресу. Этот принцип открыл возможность использовать переменные в программировании.
5. Возможность условного перехода в процессе выполнения программы. Не смотря на то, что команды выполняются последовательно, в программах можно реализовать возможность перехода к любому участку кода.

Общая структура ЭВМ (классическая/традиционная архитектура фон Неймана):



УУ - устройство управления

АЛУ - арифметико-логическое устройство

ОП - основная (оперативная) память

УВВ - устройство ввода-вывода

-> - связи по управлению

=> - связи по данным

= => - первоначально такой связи не было, но потом УВВ стали напрямую общаться с ОП

~ - "узкое" место архитектуры фон Неймана, нехватка пропускной способности канала ПАМЯТЬ - АЛУ

Базовый цикл работы ЭВМ:

- 1) формирование адреса инструкции
- 2) извлечение инструкции из основной памяти
- 3) извлечение из памяти данных (если это необходимо)
- 4) выполнение операции (что выполнять)
- 5) запись результата выполненной операции в память
- 6) возврат к пункту 1, если не произошло остановок

Многоуровневая организация ЭВМ:

- 1) Архитектура - общие принципы построения ЭВМ без вникания в детали конкретной реализации.
- 2) Организация:

7 [концептуальный]
6 [ЯПВУ] - язык программирования высокого уровня
5 [язык ассемблера]
4,5 [ОС] - операционная система
{ 4 [МЯ] - машинный язык
В { 3 [[М-программный]] - микропрограммный (может отсутствовать)
С { 2 [схемотехнический]
{ 1 [физический]

1. Физический уровень - описание физических/электрических процессов.
2. Схемотехнический - логическое описание элементов и узлов ЭВМ.
3. Микропрограммный - набор простейших действий для выполнения низкоуровневой инструкции
4. Машинного языка - набор операций или команд, которые могут быть выполнены.
- 4,5. Операционной системы - промежуточный уровень, посредник между ПО и ВС.
5. Язык ассемблера - язык, который позволяет делать практически все с использованием машинного языка.
6. Язык программирования высокого уровня - уровень разработки сложных программных комплексов.
7. Концептуальный - уровень представления задачи.

Эниак: https://computer-museum.ru/frgnhist/universal_p.htm

В феврале 1946 года Джон Преспер Экерт и Джон Уильям Мочли готовились впервые представить миру электронный вычислитель. Их ENIAC — Electronic Numerical Integrator and Computer — мог решать за одну секунду 5000 задач на сложение, гораздо больше, чем любое известное тогда устройство. Многие историки признают, что другие компьютеры появились раньше: Z3 в Германии, Colossus в Англии и Atanasoff-Berry Computer в Университете штата Айова. Но ENIAC сделал нечто более важное: он разжег воображение ученых и предпринимателей. Компьютер UNIVAC (конечно же с декоративными лампочками), построенный компанией, которую впоследствии основали Экерт и Мочли, предсказал результаты президентских выборов 1952 года; другой фигурировал в рекламе, пропагандируя еще одно научное достижение. Английская вычислительная машина Colossus приобрела известность в военных кругах. Но после Второй Мировой войны она была демонтирована и десятилетия оставалась засекреченной. Среди других компьютеров, которые выполняли такие же практические функции, ENIAC был в техническом отношении белой вороной. Он опирался на десятичную систему счисления, а не на двоичную из единиц и нулей, как практически все последующие компьютеры, даже те, что разработали Экерт и Мочли. ENIAC не хранил программы. И не использовал условные переходы — краеугольный камень современного программирования.

1) Первое поколение.

Использовались вакуумные полупроводниковые базы - электронные лампы.

Особенности:

Только 1 вариант использования - монопольный доступ (как с отдельным персональным компьютером).

Основная характеристика - надежность. Из нее вытекает метод использования и прочее. Отказы возникали каждые 5-6 часов непрерывной работы.

По той же причине ее конструкция не могла быть усовершенствована. Что касается архитектуры - классика фон Неймана.

2) Второе поколение оформилось с появлением твердотельных полупроводников - транзисторов (удалось минимизировать оборудование).

Произошло снижение потребляемой электроэнергии, на порядки возросла отказоустойчивость (надежность).

Основным методом использования стала пакетная обработка.

Бывает 2 типа пакетной обработки:

1. Элементарная - в ЭВМ вводится не одно отдельное задание (как в 1-м поколении), а задания объединяются в пакет. Пакет ставился в устройство ввода, и задания считывались одно за другим. Пакет копировался на магнитную ленту, и затем выполнялось.

ЭВМ работала с различной скоростью.

Необходимо было совместить во времени работу устройств ввода-вывода и процессора. Эта проблема была решена посредством введения децентрализации внешнего обмена (1-я мысль).

2-я мысль - мультипрограммный режим - одновременно в процессоре обрабатывается не одно, а несколько заданий.

Эти две мысли и лежат в идее пакетной обработки.

Режим работы ЭВМ при наличии децентрализации внешнего обмена и мультипрограммного режима:

1. Составляется элементарный пакет, а затем начинается ввод первого задания.

2. Идет обработка первого задания и одновременно вводится второе.

3. Далее, процессор выдает ответ по первому заданию, обрабатывает второе и вводит третье.

4. Аналогично для следующих

Процесс отладки программы:

Пользователь общался с компьютером через посредника (вычислительный центр) - оператора ЭВМ.

Пользователь писал текст программы и отдавал оператору, который наносил на перфокарту исходные данные. Затем делался тренировочный экземпляр (это все была подготовка данных), который далее вносится в пакет для выполнения. Затем возможен процесс отладки в связи с ошибками в программе.

Особенности архитектуры ЭВМ 2 поколения:

1. Начался переход к иерархической организации памяти.
2. Появились аппаратные средства децентрализации внешнего обмена (ввода-вывода).
3. Появление системы прерываний (устройство информирует о возникновении события, происходит обработка полученного прерывания, затем - возвращение к выполняемой программе).
4. Появление оборудования преобразования адресов (индексный регистр).

Недостатки прошлой системы адресации:

1. Расход памяти на хранение масок
2. Снижение производительности (выполнение операций в цикле)

Был введен индексный регистр, который загружался инкрементными значениями. В каждом цикле на аппаратном уровне происходит модификация адресного регистра, что приводит к:

1. Существенное увеличение производительности
2. Стало можно создавать защищенные области программы, к которым нельзя обращаться с целью их модификации (актуально в мультипрограммных режимах).
3. Возможно хранить программы в ПЗУ (постоянное запоминающее устройство).
5. Появление таймера (возникновение потребности в учете времени выполнения программ).

Представитель семейства суперкомпьютеров: БЭСМ-6 (> 1.000.000 операций в секунду).

3) Третье поколение.

Изменение элементной базы - интегральные микросхемы. Изменения в тепловыделении, быстродействии, габаритах.

Новый метод доступа - режим разделения времени:

доступ к ЭВМ получают одновременно несколько пользователей (от 1 до нескольких сотен). Доступ осуществляется через устройство ввода-вывода - терминал (как минимум 2 устройства: дисплей и клавиатура, без вычислительной "начинки"). Все терминалы подключены к ЭВМ, так что у пользователя появляется иллюзия монопольного владения всеми ресурсами машины.

Пользователь лишен доступа к самому оборудованию, в дисплейном классе ему выделено рабочее место. На выполнение программы выделяется квант времени (от 0.01 до 0.1 доли секунды). Если программа не успевает выполниться за один квант времени, она приостанавливается, и ресурсы передаются следующему пользователю (режим вытесняемой многозадачности).

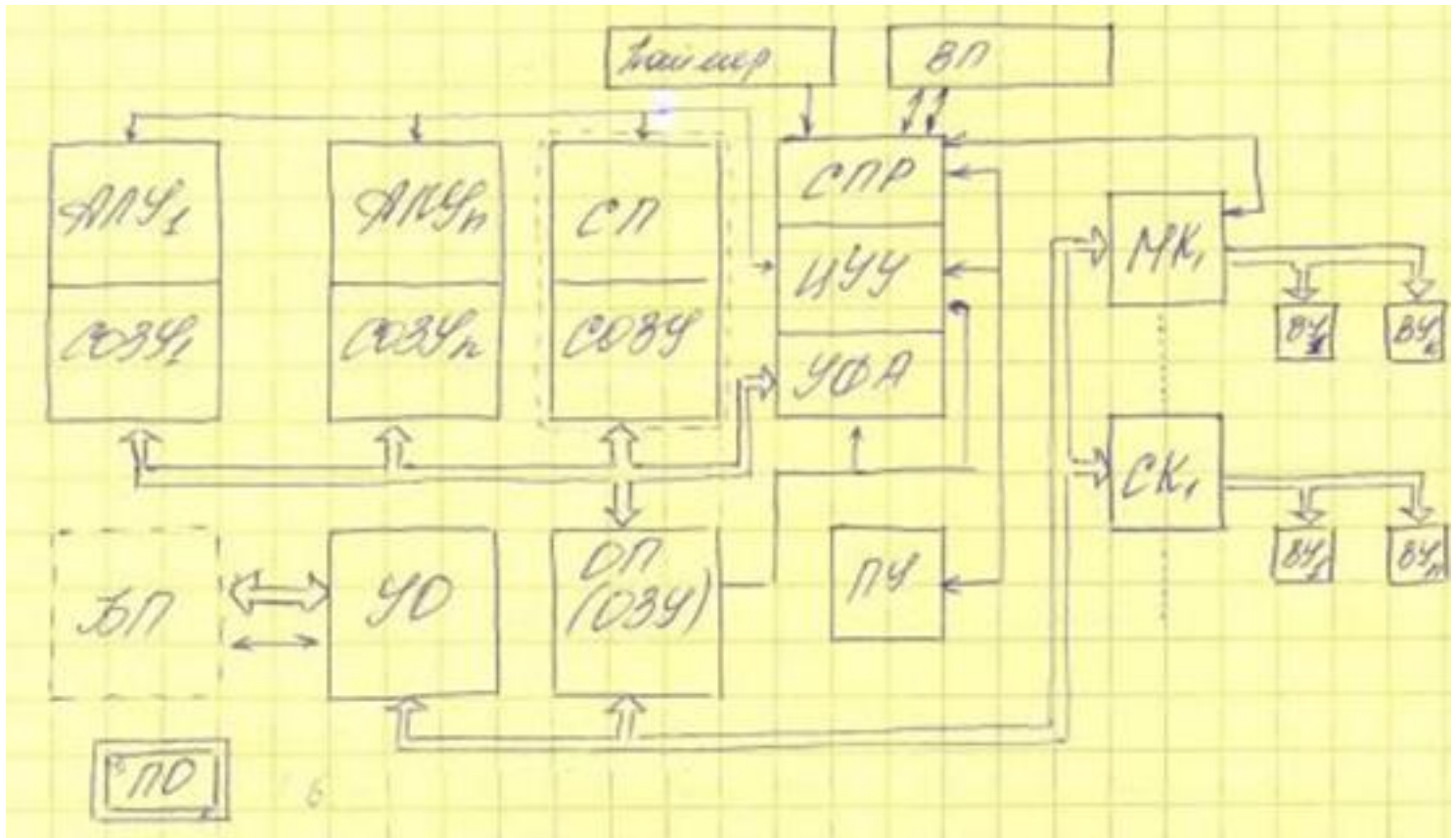
Основные архитектурные особенности ЭВМ 3-го поколения:

1. Усложнение центрального процессора (появляется несколько функциональных узлов), получило развитие оборудование формирования адресов, на которое стали возлагаться дополнительные функции: перемещение в памяти программ и данных, защита областей памяти.
2. Дальнейшее развитие принципа децентрализации внешнего обмена, появление специальных процессоров ввода-вывода (каналы).
3. Развитие системы прерываний (добавились прерывания от таймера).
4. Усложнение устройства управления, появляются микропрограммные реализации устройств управления.
5. Усложнение иерархической организации памяти, появление большой памяти, занимающей промежуточное место между основной и внешней.

6. Значительное развитие и расширение типов периферийного оборудования (появление телекоммуникационных устройств, посредством которых можно было подключать терминалы на значительном расстоянии).

7. Увеличение роли программного обеспечения.

SuperVisor - специальное устройство, выделяет каждому терминалу определенную область и периодически проверяет их. Для каждой задачи выделяется квант времени с использованием таймера (в отличие от пакета). Если задача не успела завершиться, поступает сигнал прерывания от таймера, по которому SuperVisor передает управление следующей задаче, выделив новый квант времени. Квант не всегда одинаков: может иметь место приоритет терминала.



АЛУ - арифметико-логическое устройство

СОЗУ - сверхоперативное запоминающее устройство

БП - большая память

УО - устройство обмена

ОП - основная память (ОЗУ)

ПО - программное обеспечение

ПУ - пульт управления

ЦУУ - центральное устройство управления

СПр - система прерываний

УФа - устройство формирования адресов

МК1 - мультиплексный канал

СК1 - селекторный канал

ВУ - внешнее устройство

СП - специализированный процессор

ВП - виртуальная память (может быть реализована как физически, так и логически; реализуется за счет части внешней памяти)

4) Четвертое поколение – появление персональных компьютеров и микроЭВМ.

Компьютеры стали доступны рядовым пользователям (ПК – персональные компьютеры).

Сфера применения расширилась, теперь их можно было применять даже в быту.

Элементной базой продолжили оставаться интегральные схемы (появились ИС большой БИС и сверхбольшой плотности СБИС).

Позднее компьютеры стали массово объединяться в сети, что позднее привело к созданию глобальной сети Интернет.

5) Пятое поколение – предполагается максимальное сближение ЭВМ и человека.

Элементной базой предполагается использование СБИС.

ЭВМ должен обладать мышлением, сходным с мышлением человека: функции искусственного интеллекта, распознавание речи и образов...

Общение должно проходить на естественном (человеческом) языке.

Работа с базами знаний (БЗ) вместо баз данных (БД).

Знание – это данные + правила их порождения (например, доказательство теорем из аксиом).

В языках программирования переход от алгоритмических (Fortran, C, Pascal) к декларативным (Lisp, Prolog).

Самое масштабное исследование систем 5-го поколения проводилось в Японии и длилось целое десятилетие (1982-1993)!

Сводная характеристика поколений ЭВМ

Характеристика	Поколения			
	<u>Первое</u>	<u>Второе</u>	<u>Третье</u>	<u>Четвертое</u>
Годы применения	1946-1960	1950-1964	1964-1970	1970-1990-е
Основной элемент	Электронная лампа	Транзистор	Интегральная схема	Большая интегральная схема
Количество ЭВМ в мире, шт	Сотни	Тысячи	Сотни тысяч	Десятки миллионов
Размеры	Очень большие (ENIAC, UNIVAC, EDSAC)	Значительно меньшие	Миникомпьютеры	Микрокомпьютеры
Быстродействие	1 (условно)	10	1 000	100 000
Носитель информации	Перфорированная лента	Магнитный диск, м. лента	Диск	Гибкий диск

Технологический процесс в электронной промышленности

Технологический процесс полупроводникового производства — технологический процесс изготовления полупроводниковых (п/п) изделий и материалов; состоит из последовательности технологических (обработка, сборка) и контрольных операций, часть производственного процесса производства п/п изделий (транзисторов, диодов и т. п.).

При производстве п/п интегральных микросхем применяется фотолитография и литографическое оборудование. Разрешающая способность (в мкм и нм) этого оборудования (т. н. проектные нормы) и определяет название применяемого конкретного технологического процесса.

Совершенствование технологии и пропорциональное уменьшение размеров п/п структур способствуют улучшению характеристик (размеры, энергопотребление, рабочие частоты, стоимость) полупроводниковых приборов (микросхем, процессоров, микроконтроллеров и т. д.). Особую значимость это имеет для процессорных ядер, в аспектах потребления электроэнергии и повышения производительности, поэтому ниже указаны процессоры (ядра) массового производства на данном техпроцессе.

Технологии производства полупроводниковой продукции с субмикронными размерами элементов основана на чрезвычайно широком круге сложных физико-химических процессов: получение тонких плёнок термическим и ионно-плазменным распылением в вакууме, механическая обработка пластин производится по 14-му классу чистоты с отклонением от плоскостности не более 1 мкм, широко применяется ультразвук и лазерное излучение, используются отжиг в кислороде и водороде, рабочие температуры при плавлении металлов достигают более 1500°C, при этом диффузионные печи поддерживают температуру с точностью 0,5°C, широко применяются опасные химические элементы и соединения (например, белый фосфор).

Всё это обуславливает особые требования к производственной гигиене, так называемую «электронную гигиену», ведь в рабочей зоне обработки полупроводниковых пластин или на операциях сборки кристалла не должно быть более пяти пылинок размером 0,5 мкм в 1 л воздуха. Поэтому в чистых комнатах на фабриках по производству подобных изделий все работники обязаны носить специальные комбинезоны. В рекламных материалах Intel спецодежда работников получила название bunny suit («костюм кролика»).

Техпроцессы 1970-х — 1980-х

Ранние техпроцессы, до стандартизации NTRS (National Technology Roadmap for Semiconductors) и ITRS, обозначались «xx мкм» (xx микрон), где xx сперва обозначало техническое разрешение литографического оборудования, затем стало обозначать длину затвора транзистора, полушаг линий металла (half pitch) и ширину линий металла. В 1970-х существовало несколько техпроцессов, в частности 10, 8, 6, 4, 3, 2 мкм; в среднем, каждые три года происходило уменьшение шага с коэффициентом 0,7

3 мкм

3 мкм — техпроцесс, соответствующий уровню технологии, достигнутому в 1975 году Zilog (Z80) и в 1979 году Intel (Intel 8086). Соответствует линейному разрешению литографического оборудования, примерно равному 3 мкм.

1,5 мкм

1,5 мкм — техпроцесс, соответствующий уровню технологии, достигнутому Intel в 1982 году. Соответствует линейному разрешению литографического оборудования, примерно 1,5 мкм.

Intel 80286

0,8 мкм

0,8 мкм — техпроцесс, соответствующий уровню технологии, достигнутому в конце 1980-х — начале 1990-х годов компаниями Intel и IBM.

Intel 80486 (1989 год)

MicroSPARC I (1992 год)

Первые Intel P5 Pentium на частотах 60 и 66 МГц (1993 год)

0,6-0,5 мкм

Техпроцесс, достигнутый производственными мощностями компаниями Intel и IBM в 1994—1995.

80486DX4 CPU (1994 год)

IBM/Motorola PowerPC 601, первый чип архитектуры PowerPC

Intel Pentium на частотах 75, 90 и 100 МГц

МЦСТ-R100 (1998 г., 0,5 мкм, 50 МГц)

Техпроцессы после середины 1990-х

Обозначения для техпроцессов, внедренных, начиная с середины 1990-х, были стандартизированы NTRS и ITRS и стали называться «Technology Node» или «Cycle». Реальные размеры затворов транзисторов логических схем стали несколько меньше, чем обозначено в названии техпроцессов 350 нм — 45 нм, благодаря внедрению технологий resist-pattern-thinning и resist ashing. С этих пор коммерческие названия техпроцессов перестали соответствовать длине затвора.

С переходом на следующий техпроцесс ITRS площадь, занимаемая стандартной ячейкой 1 бита памяти SRAM, в среднем уменьшалась вдвое. В период с 1995 по 2008 года такое удвоение плотности транзисторов происходило в среднем каждые 2 года.

350 нм (0,35 мкм)

350 нм — техпроцесс, соответствующий уровню технологии, достигнутому в 1997 году ведущими компаниями-производителями микросхем, такими как Intel, IBM, и TSMC. Соответствует линейному разрешению литографического оборудования, примерно равному 0,35 мкм.

Intel Pentium MMX (P55)

Intel Pentium Pro

Pentium II (Klamath)

МЦСТ-R150 (2001 г., 150 МГц)

250 нм

250 нм — техпроцесс, соответствующий уровню технологии, достигнутому в 1998 году ведущими компаниями-производителями микросхем. Соответствует линейному разрешению литографического оборудования, примерно равному 0,25 мкм.

Используется до 6 слоёв металла, минимальное количество литографических масок 22.

Pentium II (Deschutes)

Pentium III (Katmai)

180 нм

180 нм — техпроцесс, соответствующий уровню технологии, достигнутому в 1999 году ведущими компаниями-производителями микросхем. Соответствует удвоению плотности размещения по отношению к предыдущему техпроцессу 0,25 мкм.

Содержит до 6-7 слоёв металла. Минимальное количество литографических масок около 22.

AMD Athlon XP (Palomino)

Intel Pentium III (Coppermine)

130 нм

130 нм — техпроцесс, соответствующий уровню технологии, достигнутому в 2001 году ведущими компаниями-производителями микросхем. В соответствии с моделями ITRS, соответствует удвоению плотности размещения элементов по отношению к прошлому техпроцессу 0,18 мкм.

Intel Pentium III Tualatin — июнь 2001

Intel Celeron Tualatin-256 — октябрь 2001

Intel Pentium M Banias — март 2003

Intel Pentium 4 Northwood — январь 2002

Intel Celeron Northwood-128 — сентябрь 2002

Intel Xeon Prestonia и Gallatin — февраль 2002

AMD Athlon XP Thoroughbred, Thorton и Barton

AMD Athlon MP Thoroughbred — август 2002

AMD Athlon XP-M Thoroughbred, Barton и Dublin

AMD Duron Applebred — август 2003

AMD K7 Sempron Thoroughbred-B, Thorton и Barton — июль 2004

AMD K8 Sempron Paris — июль 2004

AMD Athlon 64 Clawhammer и Newcastle — сентябрь 2003

AMD Opteron Sledgehammer — июнь 2003

МЦСТ Эльбрус 2000 (1891BM4Я) — июль 2008

МЦСТ-R500S (1891BM3) — 2008, 500 МГц

Техпроцессы менее 100 нм

Для обозначения более тонких техпроцессов разные технологические альянсы могут следовать различным рекомендациям (Foundry/IDM). В частности, TSMC использует обозначения 40 нм, 28 нм и 20 нм для техпроцессов, сходных по плотности с процессами Intel 45 нм, 32 нм и 22 нм соответственно.

90 нм

90 нм — техпроцесс, соответствующий уровню полупроводниковой технологии, которая была достигнута к 2002—2003 годам. В соответствии с моделями ITRS, соответствует удвоению плотности размещения элементов по отношению к предыдущему техпроцессу 0,13 мкм.

Технологический процесс с проектной нормой 90 нм часто используется с технологиями напряженного кремния, медных соединений с меньшим сопротивлением, чем у ранее применяемого алюминия, а также с новыми диэлектрическими материалами с низкой диэлектрической проницаемостью.

Intel Pentium 4 (Prescott)

МЦСТ-4R (готовится к выпуску, 4 ядра, 1 ГГц)

AMD Turion 64 X2 (мобильный)

Эльбрус-S — 2010

65 нм

65 нм — техпроцесс, соответствующий уровню технологии, достигнутому к 2004 году ведущими компаниями-производителями микросхем. В соответствии с моделями ITRS, соответствует удвоению плотности размещения элементов по отношению к предыдущему техпроцессу 90 нм.

Intel Pentium 4 (Cedar Mill) — 2006-01-16

Intel Pentium D 900-series — 2006-01-16

Intel Celeron D (Cedar Mill cores) — 2006-05-28

Intel Celeron M

Intel Core — 2006-01-05

Intel Core 2 — 2006-07-27

Intel Core 2 Duo

Intel Core 2 Quad

Intel Xeon — 2006-03-14

AMD Athlon 64 — 2007-02-20

AMD Phenom X3, X4

AMD Turion 64 X2 (мобильный)

AMD Turion 64 X2 Ultra (мобильный)

STI Cell — PlayStation 3 — 2007-11-17

Microsoft Xbox 360 «Falcon» CPU — 2007-09

Microsoft Xbox 360 «Opus» CPU — 2008

Microsoft Xbox 360 «Jasper» CPU — 2008-10

Microsoft Xbox 360 «Jasper» GPU — 2008-10

Sun UltraSPARC T2 — 2007-10

TI OMAP 3 — 2008-02

VIA Nano — 2008-05

Loongson — 2009

Эльбрус-4С — 2014

45 нм / 40 нм

45 нм и 40 нм — техпроцесс, соответствующий уровню технологии, достигнутому к 2006—2007 годам ведущими компаниями-производителями микросхем. В соответствии с моделями ITRS, соответствует удвоению плотности размещения элементов по отношению к предыдущему техпроцессу 65 нм.

Для микроэлектронной промышленности стал революционным, так как это был первый техпроцесс, использующий технологию high-k/metal gate (HfSiON/TaN в технологии компании Intel), для замены физически себя исчерпавших SiO₂/poly-Si.

Intel Core 2 Duo

Intel Core 2 Quad

AMD Phenom II X2, X3, X4, X6

AMD Athlon II X2, X3, X4

Fujitsu SPARC64 VIIIfx

XCGPU (APU от GlobalFoundries, с 2010)

32 нм / 28 нм

32 нм — техпроцесс, соответствующий уровню технологии, достигнутому к 2009—2010 годам ведущими компаниями-производителями микросхем. В соответствии с моделями ITRS, соответствует удвоению плотности размещения элементов по отношению к предыдущему техпроцессу 45 нм.

Осенью 2009 компания Intel находилась на этапе перехода к этому новому техпроцессу. С начала 2011 начали производиться процессоры по данному техпроцессу.

В третьем квартале 2010 года на новых мощностях расположенной на Тайване фабрики Fab 12 компании TSMC начался серийный выпуск продукции по технологии, получившей маркетинговое обозначение «28-нанометров» (не является обозначением, рекомендуемым ITRS).

Intel Sandy Bridge

Intel Saltwell

AMD Bulldozer

AMD Piledriver (англ.)русск. (второе поколение Bulldozer)

APU от AMD: Llano и Trinity (второе поколение AMD APU)

Многоядерные процессоры Snapdragon фирмы Qualcomm.

Мобильные процессоры Apple A7, изготавливаемые Samsung.

AMD Steamroller (третье поколение Bulldozer - 2014)

Baikal-T1 — 2015

Эльбрус-8С (восьмиядерный процессор серверного класса с архитектурой «Эльбрус» - 2015)

В мае 2011 по технологии 28 нм фирмой Altera была выпущена самая большая в мире микросхема, состоящая из 3,9 млрд транзисторов.

22 нм / 20 нм

22 нм — техпроцесс, соответствующий уровню технологии, достигнутому к 2009—2012 гг. ведущими компаниями — производителями микросхем. Соответствует удвоению плотности размещения элементов по отношению к предыдущему техпроцессу 32 нм.

22-нм элементы формируются путём фотолитографии, в которой маска экспонируется светом с длиной волны 193 нм.

В 2008 году, на ежегодной выставке высоких технологий International Electron Devices Meeting в Сан-Франциско технологический альянс компаний IBM, AMD и Toshiba продемонстрировал ячейку памяти SRAM, выполненную по 22-нм техпроцессу из транзисторов типа FinFET, которые, в свою очередь, выполняются по прогрессивной технологии high-k/metal gate (затворы транзистора изготавливаются не из кремния, а из гафния), площадью всего 0,128 мкм² (0,58×0,22 мкм).

Также о разработке ячейки памяти типа SRAM площадью 0,1 мкм², созданной по техпроцессу 22 нм, объявили IBM и AMD

Первые работоспособные тестовые образцы регулярных структур (SRAM) представлены публике компанией Intel в 2009 году. 22-нм тестовые микросхемы представляют собой память SRAM и логические модули. SRAM-ячейки размером 0,108 и 0,092 мкм² функционируют в составе массивов по 364 млн бит. Ячейка площадью 0,108 мкм² оптимизирована для работы в низковольтной среде, а ячейка площадью 0,092 мкм² является самой миниатюрной из известных сегодня ячеек SRAM.

По такой технологии производятся (с начала 2012 года):

Intel Ivy Bridge / Ivy Bridge-E

Intel Haswell (последователь Ivy Bridge, со встроенным GPU).

Intel Bay Trail-M (мобильные Pentium и Celeron на микроархитектуре Silvermont; сентябрь 2013)

14 нм / 16 нм FinFET

По состоянию на май 2014, компания Samsung продолжала разработки техпроцессов 14 нм LPE/LPP; а выпускать процессоры для Apple планирует в 2015 году.

По состоянию на сентябрь 2014, TSMC продолжала разработку 16-нм техпроцесса на транзисторах с вертикально расположенным затвором (Fin Field Effect Transistor, FinFET) и планировала начать 16-нм производство в 1 квартале 2015 года.

Согласно экстенсивной стратегии фирмы Intel, уменьшение техпроцесса до 14 нм изначально ожидалось через год после представления чипа Haswell (2013); процессоры на новом техпроцессе будут использовать архитектуру с названием Broadwell. Для критических слоёв техпроцесса 14 нм Intel потребовалось применение масок с технологией Inverse Lithography (ILT) и SMO (Source Mask Optimization) процессоры Celeron N3000, N3050, N3150 и Pentium N3700 (Braswell) — начало продаж апрель 2015

Coffee Lake — десктопные процессоры от Intel (24 сентября 2017)

AMD Ryzen — десктопные процессоры от AMD (2017)

Мобильные процессоры Apple A10

Компания МЦСТ к 2020 году планирует разработать 14-нм процессор Эльбрус-32С.

В апреле 2018 года AMD представила процессоры Zen+ на улучшенном 14-нм техпроцессе, условно обозначенном как "12 нм":

Ryzen 5 2600 и 2600X

Ryzen 7 2700 и 2700X

10 нм

Тайваньский производитель United Microelectronics Corporation (UMC) сообщил, что присоединится к технологическому альянсу IBM для участия в разработке 10-нм CMOS-техпроцесса.

В 2011 году публиковалась информация о планах Intel по внедрению 10-нм техпроцесса к 2018 году, в октябре 2017 Intel сообщил о планах начать производство до конца 2017 года, но в итоге, после выпуска крайне ограниченной партией 10-нм мобильного процессора Intel Core i3-8121U в 2018, массовое производство процессоров Intel по 10-нм техпроцессу началось только в 2019 году для мобильных устройств и в 2020 для десктопных.

Пробный выпуск продукции по нормам 10 нм намечался компанией TSMC на 2015 год, а серийный — на 2016.

В начале 2017 года выпуск 10 нм составлял около 1 % от продукции TSMC

Samsung запустил 10-нм производство в 2017 году

Apple A11 Bionic — 64-битный шестиядерный процессор для iPhone 8 (2017).

Cannon Lake — первое поколение нескольких моделей 10-нм мобильных процессоров Intel с отключенным графическим ядром.

Ice Lake — второе поколение 10-нм процессоров Intel.

Snapdragon 835.

Snapdragon 845.

7 нм

В 2018 году на фабриках TSMC началось производство мобильных процессоров Apple A12, Kirin 980 а также Snapdragon 855. Производство 7-нм процессоров на архитектуре x86 задерживается, первые образцы на данной архитектуре появляются не раньше 2019 года. Согласно интернет-изданию Russian Tom's Hardware Guide, с помощью первого поколения 7-нм техпроцесса TSMC может разместить 66 миллионов транзисторов на квадратном миллиметре, в то же время с помощью 10-нм техпроцесса Intel может разместить на аналогичной площади 100 миллионов транзисторов. Согласно изданию Hardwareluxx, при 7-нм техпроцессе Intel планирует разместить 242 миллиона транзисторов на квадратном миллиметре.

Apple A12X (с 2018)

Zen 2 (микроархитектура) от AMD

серверные процессоры Epyc; ожидаются в 2019 году

десктопные процессоры Matisse; в продаже с начала июля 2019 года

графический ускоритель Vega от AMD

RDNA (микроархитектура) от AMD

Snapdragon 855

Snapdragon 865

Exynos 990

HiSilicon Kirin 980

Переход на второе поколение 7-нм техпроцесса у TSMC состоялся в 2019 году. Первым массовым продуктом, произведённым по этому техпроцессу, стал Apple A13. Также на этом техпроцессе в 2020 будут производиться десктопные процессоры Ryzen 4-го поколения на архитектуре Zen 3.

6 нм

16 апреля 2019 года компания TSMC анонсировала освоение технологического процесса 6-нм в рисковом производстве, что позволяет повысить плотность упаковки элементов микросхем на 18 %, данный техпроцесс является более дешевой альтернативной техпроцессу 5 нм, он позволяет легко масштабировать топологии, разработанные для 7 нм.

5 нм

В первой половине 2019 года компания TSMC начала рисковое производство чипов по 5-нм техпроцессу. Переход на эту технологию позволяет повысить плотность упаковки электронных компонентов на 80 % и повысить быстродействие на 15 %.

Samsung в марте 2017 презентовал дорожную карту по выпуску процессоров по 7- и 5-нм технологиям. В ходе презентации вице-президент Samsung по технологии Хо-Кью Кан отметил, что многие производители столкнулись с проблемой при разработке технологий меньше 10 нм. Однако Samsung справилась с задачей, ключом к которой стало использование полевого транзистора с «кольцевым» затвором (GAAFET). Эти транзисторы позволят компании продолжить уменьшать элементы до размера 7 и 5 нм. Для изготовления пластин компания применит технологию экстремальной ультрафиолетовой литографии (EUV).

3 нм

Исследовательский центр imec (Бельгия) и компания Cadence Design Systems создали технологию и в начале 2018 года выпустили первые пробные образцы микропроцессоров по технологии 3 нм.

Samsung к 2021 году намерена начать производство 3-нанометровой продукции с использованием технологии GAAFET.

Таким образом, ведущие мировые производители ведут интенсивные научные исследования с целью дальнейшего увеличения плотности интегральных схем (ИС). К настоящему моменту масштаб величин приближается к критической величине, когда габаритные размеры кристаллов микроэлементов приближаются к молекулярному уровню. При этом, становится все сложнее управлять классическими электрическими процессами, и все большую погрешность вносят квантовые эффекты отдельных молекул и атомов, для компенсации которых приходится также разрабатывать свои технологические решения.

Классификация устройств памяти

Запоминающее устройство (ЗУ) – носитель информации, предназначенный для записи и хранения данных. В основе работы ЗУ может лежать любой физический эффект, обеспечивающий приведение системы к двум или более устойчивым состояниям.

Классификация ЗУ

По устойчивости записи и возможности перезаписи ЗУ делятся на:

- Постоянные ЗУ (ПЗУ), содержание которых не может быть изменено конечным пользователем (например DVD-ROM). ПЗУ в рабочем режиме допускает только считывание информации.
- Записываемые ЗУ, в которые конечный пользователь может записать информацию только один раз (например DVD-R).
- Многократно перезаписываемые ЗУ (например DVD-RW).
- Оперативные ЗУ (ОЗУ) – обеспечивают режим записи, хранения и считывания информации в процессе ее обработки.

По типу доступа ЗУ делятся на:

- ЗУ с последовательным доступом (например, магнитные ленты).
- ЗУ с произвольным доступом (RAM – Random Access Memory, например, оперативная память).
- ЗУ с прямым доступом (например, HDD – жесткие магнитные диски).
- ЗУ с ассоциативным доступом (специальные ЗУ для повышения производительности БД).

По геометрическому исполнению:

- Дисковые (магнитные диски, оптические, магнитооптические).
- Ленточные (перфоленты, магнитные ленты).
- Барабанные (магнитные барабаны).
- Карточные (перфокарты, магнитные карты, флеш-карты).
- Печатные платы (карты DRAM – Dynamic Random Access Memory – Динамическая Память с Произвольным Доступом).

По физическому принципу:

- Перфорационные (перфокарты, перфоленты).
- С магнитной записью (ферритовые сердечники, магнитные ленты, магнитные диски, магнитные карты).
- Оптические (CD, DVD, HD-DVD, Blue-ray Disc).
- Использующие эффекты в полупроводниках (флеш-память).

По форме записанной информации:

- Аналоговые.
- Цифровые.

По организации доступа к запоминающему массиву носителю:

- С неподвижным носителем.
- С движущимся (вращающимся, перематывающимся) носителем.

Устройство USB-флеш-накопителя

USB-флеш-накопитель — запоминающее устройство, использующее в качестве носителя флеш-память, и подключаемое к компьютеру или иному считывающему устройству по интерфейсу USB, пришедшее на замену флоппи-дискам. Флэш-накопители USB обычно являются съёмными и перезаписываемыми, и физически намного меньше, чем оптический диск. Большинство весит менее 30 грамм. USB-накопители часто используются для тех же целей, для которых когда-то использовались гибкие диски или компакт-диски; то есть для хранения, резервного копирования данных и передачи компьютерных файлов. Они меньше, быстрее, имеют гораздо большую ёмкость и более прочны и надёжны, потому что у них нет движущихся частей. Кроме того, они невосприимчивы к магнитным полям (в отличие от флоппи-дисков) и не подвергаются воздействию поверхностных царапин (в отличие от компакт-дисков). Основное назначение USB-накопителей — хранение, перенос и обмен данными, резервное копирование, загрузка операционных систем и др.

Устройство

Основные компоненты флешки:

- USB-интерфейс (чаще USB 2.0 или 3.0 Стандарт-A, иногда microUSB) — обеспечивает физическое соединение с компьютером;
- Контроллер — небольшой микроконтроллер со встроенными ROM и RAM;
- NAND-чип флеш-памяти — хранит информацию;
- Осциллятор — генерирует синхронизирующий сигнал (12 MHz) для шины USB;
- На большинстве флешек повсеместно используются файловые системы семейства FAT.

В зависимости от размера накопителя применяются файловые системы FAT16, FAT32 или exFAT. Для флешек размером 64ГБ и более используются NTFS или exFAT.

Преимущества и недостатки

Преимущества

- Малый вес, бесшумность работы и портативность.
- Универсальность: современные компьютеры, телевизоры, DVD- и медиа-проигрыватели имеют USB-порты.
- Низкое энергопотребление (благодаря отсутствию механических систем, в отличие от CD, DVD, BD и жёстких дисков)
- Работоспособность в широком диапазоне температур.
- Более устойчивы к механическим воздействиям (вибрации и ударам), а также к воздействию магнитных полей по сравнению с жёсткими дисками.
- Не подвержены воздействию царапин и пыли, которые были проблемой для оптических носителей и дискет.

- Способны сравнительно длительно хранить данные в автономном режиме (не требуя питания), от единиц до 10 лет. В худшем случае (дешевый накопитель с большим износом) хранение данных обеспечивается в течение 3—6 месяцев.

Недостатки

- Ограниченное число циклов записи-стирания перед выходом из строя. Чипы памяти, сделанные по технологии MLC (большинство), чаще всего выдерживают не более 5000 циклов перезаписи. Кроме этого ограничен ресурс USB-коннектора — около 1500 подключений
- Скорость записи и чтения ограничены пропускной способностью USB, что особенно сильно проявляется для USB 2.0 (не более 35 МБ/с)
- В отличие от компакт-дисков, имеют недостатки, свойственные любой электронике:
 - чувствительны к электростатическому разряду — обычное явление в быту, особенно зимой;
 - чувствительны к радиации.
- Несимметричность интерфейса при симметрично выглядящем разъёме, отчего подключить сразу получается не всегда. Недостаток многих разъёмов, проявившийся для USB вообще, а для флешек особенно — из-за частого подключения-отключения. Похожий недостаток у ключей, устранённый симметричной бородкой. Однако данная проблема в будущем будет устранена благодаря симметричному разъёму USB Type-C.

История флеш-технологии

Предшественниками технологии флеш-памяти можно считать ультрафиолетово-стираемые постоянные запоминающие устройства (EPROM) и электрически стираемые ПЗУ (EEPROM). Эти приборы также имели матрицу транзисторов с плавающим затвором, в которых инжекция электронов в плавающий затвор («запись») осуществлялась созданием большой напряжённости электрического поля в тонком диэлектрике. Однако площадь разводки компонентов в матрице резко увеличивалась, если требовалось создать поле обратной напряжённости для снятия электронов с плавающего затвора («стирания»). Поэтому и возникло два класса устройств: в одном случае жертвовали цепями стирания, получая память высокой плотности с однократной записью, а в другом случае делали полнофункциональное устройство с гораздо меньшей ёмкостью.

Соответственно усилия инженеров были направлены на решение проблемы плотности компоновки цепей стирания. Они увенчались успехом изобретением инженера компании Toshiba Фудзиро Масуокой в 1984 году. Название «флеш» было придумано также в Toshiba — Сёдзи Ариидзуми, процесс стирания содержимого памяти которому напомнил фотовспышку (англ. flash). Масуока представил свою разработку в 1984 году на конференции IEEE International Electron Devices Meeting (IEDM), проходившей в Сан-Франциско.

В 1988 году Intel выпустила первый коммерческий флеш-чип NOR-типа.

NAND-тип флеш-памяти был анонсирован Toshiba в 1989 году на International Solid-State Circuits Conference.

NOR и NAND тип

Флеш-память различается методом соединения ячеек в массив.

Конструкция NOR использует классическую двумерную матрицу проводников, в которой на пересечении строк и столбцов установлено по одной ячейке. При этом проводник строк

подключался к стоку транзистора, а столбцов — ко второму затвору. Исток подключался к общей для всех подложке.

Конструкция NAND — трёхмерный массив. В основе та же самая матрица, что и в NOR, но вместо одного транзистора в каждом пересечении устанавливается столбец из последовательно включенных ячеек. В такой конструкции получается много затворных цепей в одном пересечении. Плотность компоновки можно резко увеличить (ведь к одной ячейке в столбце подходит только один проводник затвора), однако алгоритм доступа к ячейкам для чтения и записи заметно усложняется. Также в каждой линии установлено два МОП-транзистора. Управляющий транзистор разрядной линии (англ. Bitline select transistor), расположенный между столбцом ячеек и разрядной линией. И управляющий транзистор заземления, расположенный перед землёй (англ. Ground select transistor).

Технология NOR позволяет получить быстрый доступ индивидуально к каждой ячейке, однако площадь ячейки велика. Наоборот, NAND имеют малую площадь ячейки, но относительно длительный доступ сразу к большой группе ячеек. Соответственно, различается область применения: NOR используется как непосредственная память программ микропроцессоров и для хранения небольших вспомогательных данных.

Названия NOR и NAND произошли по ассоциации схемы включения ячеек в массив со схемотехникой микросхем КМОП-логики — NOR- и NAND-элементов.

NAND чаще всего применяется для USB-флеш-накопителей, карт памяти, SSD, NOR — во встраиваемых системах.

Существовали и другие варианты объединения ячеек в массив, но они не прижились.

Иерархическая структура памяти ЭВМ

Начиная со 2-го поколения ЭВМ начался переход к иерархической структуре памяти. Запоминающие устройства (ЗУ) разных типов обладали весьма разными характеристиками (объем, быстродействие, стоимость в пересчете на единицу хранимой информации). Использование самого лучшего типа памяти было бы экономически невыгодно, т.к. существенно удорожало ЭВМ. Поэтому разработчики пошли по пути комбинирования памяти разных типов, оптимизируя ее использование – на критически важные звенья вычислительного процесса пускали наиболее совершенную и дорогую память, на менее значимые – обычную. В итоге все это эволюционировало в следующую иерархическую структуру:



Рисунок из доклада о микропроцессорах (про память начинается со слайда 9):

<https://presentacii.ru/presentation/mikroprocessory>

Регистры (регистровая память)

Самая быстрая память – соединена напрямую с процессором (или иным вычислительным узлом), нет системы адресации – доступ только по именам регистров (замыкается нужная цепь на схеме), строится на триггерах (<https://ru.wikipedia.org/wiki/Триггер>) – быстрых логических устройствах, но и самая дорогая – каждый регистр по сути является самостоятельным ЗУ.

Наращивание объема данной памяти невозможно без изменения архитектуры процессора.

Относится к энергозависимой памяти, т.к. теряет хранимую информацию при отключении питания.

Кэш-память (СОЗУ – сверхоперативное ЗУ)

Это также персональная память процессора или какого-либо вычислительного узла. Строится на регистровых структурах. Для доступа к ячейкам используется система адресации, что немного замедляет ее работу. По быстродействию уступает регистрам на 1-2 порядка, но быстрее ОЗУ примерно на 1 порядок.

Используется для ускорения массовой обработки данных в оперативной памяти (например, массив обрабатывается в цикле, его ячейки расположены в памяти подряд, часто оказывается выгоднее по времени загрузить страницу памяти из ОЗУ в СОЗУ, обработать там, а потом вернуть результаты обратно).

Также энергозависима, расширение объема теоретически возможно, используется для разгрузки оперативной памяти.

Часто бывает многоуровневой (например L1,L2,L3).

Основная память (оперативная память, ОП, ОЗУ)

Когда-то была единственной памятью в ЭВМ. Сейчас остается основной – в нее загружаются программы с жесткого диска при их запуске на выполнение.

Энергозависима, строится на конденсаторах. Каждый конденсатор хранит определенный заряд (поддерживается положительная или отрицательная разность потенциалов). Поскольку это устройство физическое, то в нем течет «ток утечки» - разность потенциалов постепенно стремится к нулю. Для данной памяти необходима система регенерации – своеобразная «подзарядка» всех конденсаторов с высокой периодичностью.

Есть система адресации, наращивание объема происходит довольно легко путем подключения новых носителей в свободные слоты на системной плате ЭВМ.

Жесткий диск (винчестер, HDD)

Энергонезависимая память для длительного хранения информации, весьма дешевая. Строится на магнитных кристаллах (диполях), которые расположены в материале дорожки носителя, и под воздействием магнитной дорожки могут менять свое положение, разворачиваясь «плюсом» или «минусом» вверх. После контакта с магнитной головкой могут длительное время сохранять свое текущее положение.

Уступают по быстродействию ОЗУ на 2-3 порядка. Если бы программы не загружались в ОЗУ, а запускались прямо на жестком диске, то они работали бы в ~1000 раз медленнее.

Внешние накопители (CD, DVD, дискеты, флешки и др.)

Данные носители являются сменными, подключаются к ЭВМ лишь эпизодически. Используются либо для хранения огромных архивов данных, которые не требуются регулярно, либо для мобильной транспортировки некоторой информации пользователя между разными ЭВМ.

Наиболее дешевая память в пересчете на единицу информации, и наиболее медленная.

В CD/DVD-носителях используется всего 1 дорожка – носитель вращается в приводе, лазерная считывающая головка перемещается вдоль дорожки, пока не найдет нужную ячейку. Это последовательный принцип доступа, пришедший по наследству от катушечных ЗУ.

В случае двухслойного DVD дорожка скручивается в спираль, один слой закручен по часовой стрелке, второй – против. Доходя до точки смены слоев меняется частота лазерного луча, и он теперь проникает в более глубокий слой, отражаясь от его подложки. Таким образом для пользователя все происходит бесшовно – диск не меняет скорости и направления вращения, задержка на смену слоя незаметна глазу.

В целом, в иерархии памяти прослеживаются следующие закономерности:

- От основания к вершине пирамиды повышается быстродействие и стоимость хранения единицы информации – тут оптимизируется вычислительная составляющая.
- От вершины к основанию пирамиды падает стоимость хранения единицы информации, повышается объем носителя – тут оптимизируется хранение.

Система заботится о том, чтобы все процессы в ней протекали по возможности оптимально, поэтому регулирует за взаимодействием смежных уровней памяти между собой, обменом информации между ними.