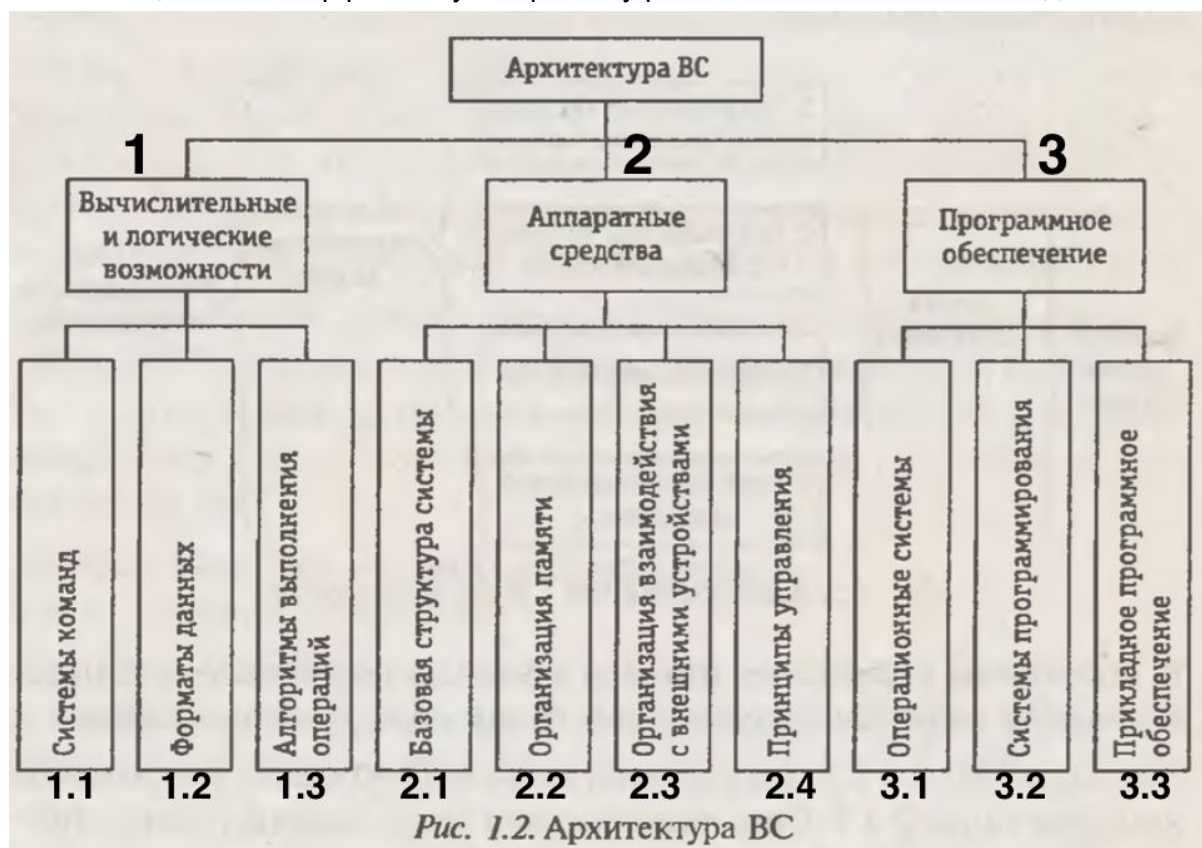


// вопрос про Позиционное/непозиционное кодирование информации (158),  
кодирование с помощью вычетов (186) отсюда ему не нравится -> даже за полностью  
отсюда списанное ставил плохие оценки

**Архитектура как набор взаимодействующих компонент (25).**

**Архитектура компьютера**, характеризующая его логическую организацию, может быть представлена как **множество взаимосвязанных компонент**, включающих, на первый взгляд, элементы различной природы: **программное обеспечение** (software), **аппаратное обеспечение** (hardware), **алгоритмическое обеспечение** (brainware), **специальное фирменное обеспечение** (firmware) — и поддерживающих слаженное функционирование компьютера в форме единого архитектурного ансамбля, позволяющего вести эффективную обработку различных объектов и своих данных.



Рассмотрим структуру системы команд в зависимости от класса решаемых задач (рис. 1.3).

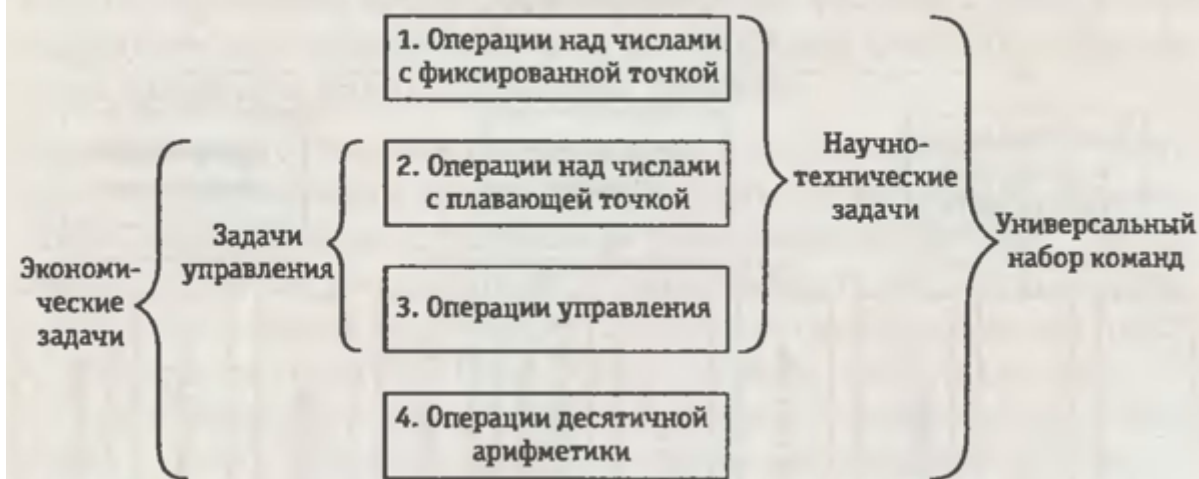


Рис. 1.3. Классификация СК по назначению

**1.2** Современные компьютеры имеют развитую систему форматов. Например, компьютер IBM имеет форматы в 2, 4, 8 и 16 байт.

**1.3** Алгоритмы выполнения операций достаточно полно отражают производительность только однопроцессорных вычислительных систем (ВС).

**2-2.1** Аппаратные средства. Простейшая ВС включает модули 5 типов: центральный процессор (ЦП), основная память (ОП), каналы, контроллеры и внешние устройства;

- ЦП (включает в себя устройство управления (УУ) + арифметико-логическое устройство (АЛУ) + память) управляет работой системы.
- ОП предназначена для хранения команд и данных и обеспечивает адресный доступ к ним от процессора.
- Каналы - спец. устройства, управляющие обменом данных с внешними устройствами.
- Контроллеры ввода-вывода служат для подсоединения внешних устройств (ВНУ) к каналам.
- ВНУ предназначенное для ввода-вывода информации с различных носителей.

**2.2** Память может быть организована как многоуровневая (внутренние регистры процессора, сверхоперативная, оперативная, внешняя) и одноуровневая (виртуальная). Более высокий уровень памяти быстрее и меньше по объему, чем низкий.

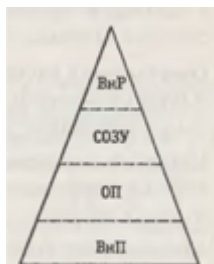


Рис. 1.4. Типы памяти

Найденные или не найденные на соответствующем уровне данные, то есть успешное или неуспешное обращение к уровню памяти, называется

соответственно **попаданием** (hit) или **промахом** (miss), а соответствующее время -- **временем обращения** (hit time или miss penalty).

2.3 Существенное влияние на производительность ВС оказывают каналы ввода-вывода. Мультиплексный канал обеспечивает работу группы медленных устройств, блок-мультиплексный -- группы быстрых устройств, селекторный - занимает информационную магистраль только одним быстродействующим устройством.

2.4 Аппаратные средства защиты памяти служат для управления доступом к различным областям памяти, в соответствии с имеющимися у пользователя полномочиями.

### ***Конвейеризация (64).***

Принцип конвейерной обработки основан на разбиении вычислительного процесса на несколько подпроцессов, каждый из которых выполняется на отдельном устройстве.

**Конвейерная обработка - это технология, при которой несколько команд в процессе обработки выполняются одновременно, при этом каждая команда (или ее этап) выполняется на отдельном функциональном устройстве с последующей передачей результатов обработки от  $i$ -ой команды (или этапа) к  $i+1$ -ой, то есть используется параллелизм инструкций в последовательном потоке команд.**

Выполнение типичной команды в этом случае можно разделить на следующие этапы:

- выборка команды IF (Instruction Fetch): из памяти по адресу, заданному счетчиком команд, извлекается команда;
- декодирование команды ID (Instruction Decode);
- выполнение операции EX (Execute);
- обращение к памяти MEM (Memory);
- запоминание результата WB (Write Back).

Для конвейеризации потока команд разобьем каждую из них на указанные этапы, отведя для выполнения каждого этапа один такт синхронизации, и в каждом такте начинаем выполнение очередной команды.

Работа конвейера представляется в виде временных диаграмм (табл. 2.1), на которых обычно изображаются выполняемые команды, номера тактов и этапы выполнения команд.

Таблица 2.1

Временная диаграмма конвейера

Номер команды	Номер такта								
	1	2	3	4	5	6	7	8	9
I	IF	ID	EX	MEM	WB				
I+1		IF	ID	EX	MEM	WB			
I+2			IF	ID	EX	MEM	WB		
I+3				IF	ID	EX	MEM	WB	
I+4					IF	ID	EX	MEM	WB

Во-первых, не все этапы выполнения команды совпадают по времени, во-вторых, не все команды выполняются до конца (ветвления, циклы и т.д.), а в-третьих, имеется информационная зависимость между операциями, в связи с чем не все устройства ВС работают на полную мощность.

Поэтому такт синхронизации не может быть меньше времени, необходимого для работы наиболее медленной ступени конвейера.

Ситуации, которые препятствуют выполнению очередной команды, называются **конфликтами**.

Три класса конфликтов:

1. **Структурные конфликты**, возникающие в случае, когда аппаратные средства не могут поддерживать все возможные комбинации команд в режиме одновременного выполнения с совмещением (конфликты по ресурсам).
2. **Конфликты по данным**, возникающие в случае, когда выполнение одной команды зависит от результата выполнения предыдущей команды.
3. **Конфликты по управлению**, которые возникают при конвейеризации команд переходов и других команд, изменяющих значение счетчика команд.

*Классификация конфликтов по данным.*

Конфликт по данным возникает при наличии зависимости между командами, расположенными по отношению друг к другу настолько близко, что совмещение операций, происходящее при конвейеризации, может привести к изменению порядка обращения к операндам. В зависимости от порядка операций чтения и записи различают три возможных конфликта по данным.

- RAW (чтение после записи)
- WAR (запись после чтения)
- WAW (запись после записи)

Указанные конфликты могут быть разрешены с помощью достаточно простого аппаратного решения, называемого пересылкой или продвижением данных (data forwarding), обходом (data bypassing), иногда — «закороткой» (short-circuiting).

Существуют *аппаратные методы*, позволяющие изменить порядок выполнения команд так, чтобы минимизировать приостановки конвейера - **методы динамической оптимизации**

Основные средства динамической оптимизации:

- размещение схемы обнаружения конфликтов в возможно более низкой точке конвейера команд
- буферизация команд, ожидающих разрешение конфликта
- организация коммутирующих магистралей

### **Иерархия памяти (о каждом типе памяти) (83)**

Память компьютера может быть организована как **многоуровневая** с различным объемом и временем доступа к ней - **сверхоперативная** (СОЗУ), **оперативная** (ОП), **внешняя** (ВнП) (рис. 3.1), так и **одноуровневая**, виртуальная. Почти всегда виртуальная память есть переупорядоченное подмножество реальной памяти.



Уровни иерархии памяти взаимосвязаны между собой - все данные одного уровня могут присутствовать на более низком уровне.

**Основная память** предназначена для хранения команд и данных и обеспечивает адресный доступ к ним от процессора. Современные устройства памяти работают со скоростью, близкой к скорости работы процессора.

При обращении процессора к памяти сначала производится поиск необходимых данных в СОЗУ. Так как время доступа к этому типу памяти меньше, чем к ОП, и, как правило, большинство необходимых для работы данных уже содержится там, то среднее время доступа к памяти уменьшается.

**Постоянная память** (ПЗУ, ROM — read only memory) предназначена только для чтения ее содержимого. В ней хранятся программы проверки оборудования компьютера, инициирования загрузки операционной системы и выполнения базовых функций по обслуживанию устройств, а также настройки конфигурации компьютера (SETUP). Так как основное содержимое ПЗУ связано с обслуживанием ввода-вывода, часто программы ROM-памяти называют BIOS (Basic Input-Output System).

Небольшая по объему **полупостоянная** память используется для хранения параметров конфигурации компьютера. По технологии изготовления ее часто называют CMOS-памятью (complementary metal-oxide semiconductor}, она имеет низкое энергопотребление.

### ***Позиционное/непозиционное кодирование информации (158), кодирование с помощью вычетов (186)***

*Позиционное (есть зависимость значения числа от позиции цифры в слове)/непозиционное кодирование информации.*

**Десятичные** числа хранятся и обрабатываются в компьютере в **двоично-десятичном** коде. Существует несколько способов кодирования десятичных цифр, использующих знаки двоичного алфавита. Представление знаков одного алфавита средствами (знаками, буквами, словами) другого алфавита называют **кодированием**. Простейший случай кодирования сообщения состоит в его дублировании. Процесс обратного преобразования информации называют **декодированием**. Для представления информации в компьютере преимущественно используется двоичное кодирование, при котором символы вводимой в компьютер информации представляются средствами двоичного алфавита, состоящего из двух знаков, в качестве которых используются символы «0» и «1». Двоичный алфавит по числу входящих в него символов является минимальным, поэтому при двоичном кодировании алфавита, включающего большее число знаков, каждому знаку ставится в соответствие последовательность нескольких двоичных знаков или **двоичное слово**. Такие последовательности называются **кодowymi комбинациями**.

Использование двоичного алфавита обуславливается сравнительной простотой построения электронных элементов с двумя устойчивыми состояниями.

Приведем некоторые определения из теории кодирования.

**Равномерный код** - код, все комбинации которого содержат одинаковое количество символов.

**Неравномерный код** - код с различным количеством символов для различных комбинаций (например, азбука Морзе).

В компьютерах обычно используют равномерные коды.

**Кодовое расстояние** - число позиций, в которых коды не совпадают

*Система кодирования данных с применением вычетов.* Она разработана с целью ускорения арифметических операций, повышения точности и надежности вычислений.



Пусть  $P_1, P_2, \dots, P_n$  — целые числа;  $P_i > 1$ ,  $(P_i, P_j) = 1, i \neq j$ ;  $M = \prod_{i=1}^n P_i$ ;  $x_i$  — наименьшие неотрицательные решения системы сравнений

$$A \equiv x_i \pmod{P_i}, i = 1, 2, \dots, n; A \in [0, M).$$

Тогда кортеж  $(x_1, x_2, \dots, x_n)$  будем называть кодом числа  $A$  в системе кода вычетов (СКВ) при заданных основаниях  $P_1, P_2, \dots, P_n$ . Записывают это обычно так:  $A \sim (x_1, x_2, \dots, x_n)$ . Компоненты  $x_i$  называют разрядами числа  $A$  в СКВ. Своими корнями данная система восходит к так называемой китайской теореме об остатках.

**Теорема.** Пусть числа  $M_s$  и  $M'_s$  определены из условий

$$P_1, P_2, \dots, P_k \equiv M_s P_s, \quad M_s M'_s \equiv 1 \pmod{P_s}$$

и пусть

$$x_0 = M_1 M'_1 b_1 + M_2 M'_2 b_2 + \dots + M_k M'_k b_k.$$

Тогда совокупность значений  $x$ , удовлетворяющих системе сравнения  $x \equiv b_1 \pmod{P_1}, x \equiv b_2 \pmod{P_2}, \dots, x \equiv b_k \pmod{P_k}$ , определяется сравнением  $x \equiv x_0 \pmod{P_1, \dots, P_k}$ .

Конек по АК

2 странич А4

21. ноября

Вопрос:

- 1) Архитектура как набор ~~функций~~ взаимодействующих компонент ~~с 48~~ (25)
- 2) Конвейеризация ~~с 48~~ (64)
- 3) Иерархия памяти (окажется тип памяти) ~~с 83~~
- 4) Позиционное/непозиционное кодирование инструкций ~~с 158~~  
Кодирование с помощью вейтов ~~с 174~~

$$\begin{cases} -2x_1 + 8x_2 - 2x_3 + 10x_4 = 0 \\ 2x_1 + 2x_2 + 3x_3 + 4x_4 = 0 \\ 10x_2 + x_3 + 4x_4 = 0 \\ 2x_1 + 2x_2 + 3x_3 + 4x_4 = 0 \end{cases}$$

$$\begin{cases} 10x_2 + x_3 + \\ 2x_1 - 28x \end{cases}$$

$$\begin{cases} x_3 = 3 \\ x_1 = 2 \end{cases}$$

$$\varphi = 3(1 - 42x) +$$

$$= x_2 -$$

3-10  
14x