

АВИАЦИОННАЯ СЛУЖБА РОССИИ  
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ГРАЖДАНСКОЙ АВИАЦИИ

Кафедра "Радиотехнические устройства".

А.Л. Горбунов

**ТЕКСТЫ ЛЕКЦИЙ**  
по дисциплине  
"Микропроцессорные устройства РЭС"  
для студентов III - V курсов сп. 201300 всех форм обучения

Москва-1997

Данные тексты лекций по дисциплине "Микропроцессорные устройства РЭС" издаются в соответствии с учебным планом для студентов III - V курсов сп. 201300 всех форм обучения.

Рассмотрены и одобрены на заседаниях кафедры РТУ / /99 г. и редакционно-издательского Совета ФАСК / /99 г.

Рецензенты:

к.т.н., доц. каф. РТУ МГТУ ГА Ю.П. Сафоненков.

к.т.н., доц. каф. РПУ МИРЭА Г.В. Куликов

Используемые сокращения:

АЛУ - арифметико-логическое устройство

ВУ - внешнее устройство

БИС - большая интегральная схема

БПФ - быстрое преобразование Фурье

ЗУ - запоминающее устройство

ИМС - интегральная микросхема

КО - канал обмена

МП - микропроцессор

РЭС - радиоэлектронная система

ЦОС - цифровая обработка сигналов

ЦФ - цифровой фильтр

DSP - (Digital Signal Processor) - процессор ЦОС

Цель и задачи дисциплины:

Цель преподавания дисциплины - изучение теории, принципов построения, методов разработки и применения микропроцессорной (МП) техники в РЭС.

Необходимый комплекс знаний и умений:

Иметь представление о назначении, составе, технических характеристиках, областях применения используемых и перспективных отечественных и зарубежных МП-комплектов включая процессоры цифровой обработки сигналов;

Знать основные алгоритмы обработки данных, применяемые при решении задач в РЭС при помощи МП; основы и особенности эксплуатации изучаемых технических средств; государственные стандарты, нормы ЕСКД применяемые при разработке, производстве и эксплуатации МП-техники;

Уметь проводить сравнительный технико-экономический анализ средств МП-техники; выбирать вычислительные алгоритмы, адекватные реализуемой задаче; разрабатывать, отлаживать и испытывать аппаратные средства и программное обеспечение МП-систем; оценивать эффективность принимаемых технических решений.

Иметь опыт практического программирования на уровне системы команд микропроцессора.

## 1. Общие сведения о микропроцессорной вычислительной технике.

ГОСТ 17021-88: Микропроцессор - устройство, состоящее из одной или нескольких БИС, включающее в себя все необходимые средства для обработки и управления данными и рассчитанное на совместную работу с устройствами памяти и ввода-вывода информации.

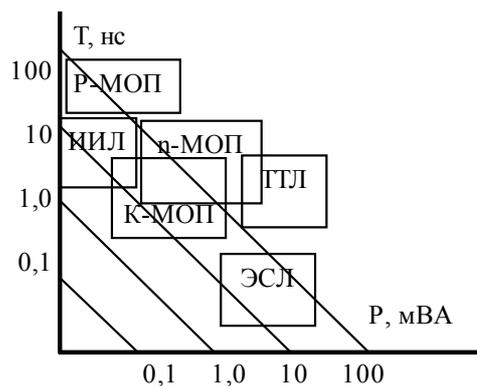
Появление МП как специального класса электронных цифровых приборов в начале 70-х обусловлено обозначившимся к этому времени противоречием между быстро развивавшимися потребностями в цифровых управляющих устройствах рынка товаров для конечного потребителя с одной стороны и тогдашними возможностями производства такого рода электроники с другой. К тому времени цифровая техника в достаточной мере проявила свои потребительские качества и, как следствие, сформировался устойчивый спрос на различные цифровые контроллеры. Однако процесс их разработки и производства оставался весьма времяемким и дорогостоящим. Очевидное решение проблемы - разработка прибора, обладающего универсальными возможностями и допускающего настройку на специфику выполняемой задачи, который можно было бы выпускать массовым тиражом и за счет этого добиться сравнительно низкой себестоимости. В 1971 году небольшая молодая компания Intel предложила такое устройство - 4-разрядную микросхему i4004 - первая БИС, которую с полным основанием можно отнести к классу МП.

Продукт продемонстрировал востребованность потребителем, и положил начало лавинообразному процессу перехода аппаратных решений вычислительной техники на МП-платформы. Эволюция МП-техники с момента появления первого МП хорошо иллюстрируется генеалогией продуктов Intel:

| Поколение | Год выпуска | Индекс БИС | Разрядность | Частота МГц |
|-----------|-------------|------------|-------------|-------------|
| I         | 1971        | i4004      | 4           | 1           |
|           | 1971        | i8008      | 8           | 1           |
| II        | 1974        | i8080      | 8           | 2,5         |
|           | 1976        | i8085      | 8           | 2,5         |
| III       | 1978        | i8086      | 16          | 4,7         |
|           | 1979        | i8088      | 16          | 4,7         |
|           | 1983        | i80186     | 16          | 8           |
|           | 1983        | i80286     | 16          | 12          |
| IV        | 1987        | i80386     | 32          | 33          |
|           | 1989        | i80486     | 32          | 33-100      |
| V         | 1993        | Pentium    | 64          | 66-200      |
| VI        | 1998        | Pentium II | 64          | 200-450     |

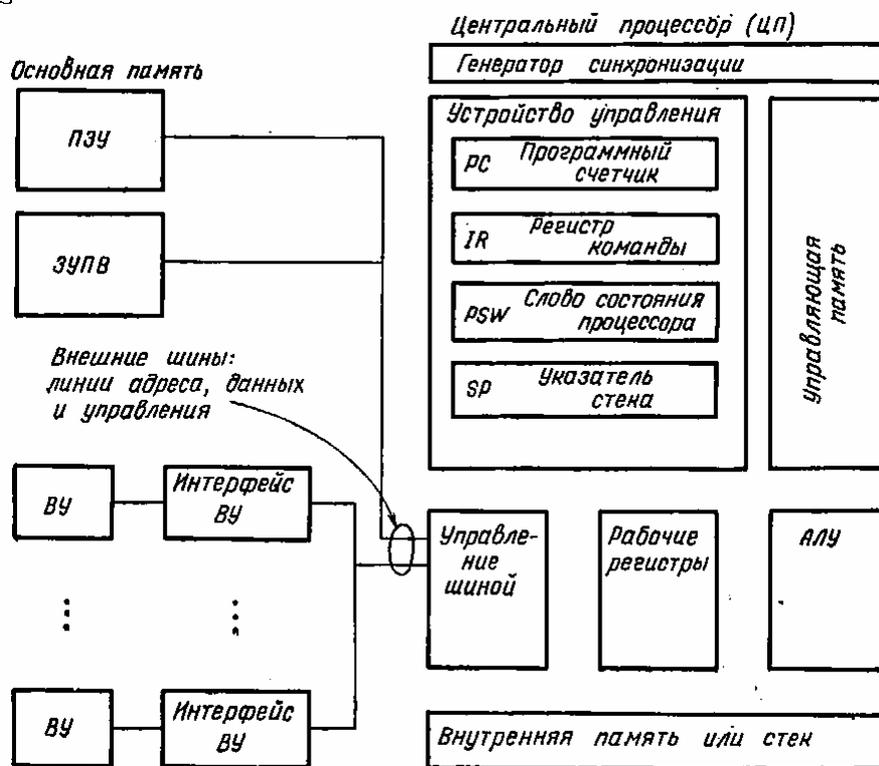
МП-решения в области РЭС в основном связаны с реализациями задач ЦОС, при этом как правило первым шагом является спектральное преобразование с целью получения цифрового энергетического спектра сигнала. В этом смысле специфика настоящего курса напрямую корреспондируется с главным сегодняшним вектором движения информационных технологий. Этот вектор связан с телекоммуникационным транспортом через глобальные сети мультимедиа-продуктов и доминантой здесь являются алгоритмы и программно-аппаратная поддержка процессов сжатия данных, а эти инструменты в настоящее время во многом базируются на спектральных преобразованиях. Означенное обстоятельство делает курс «Микропроцессорные устройства РЭС» весьма актуальным не только в ракурсе специализации, но и в целом в общем горизонте современной техники.

Классификация МП может осуществлена с использованием различных классификационных признаков. Среди них наиболее общими являются тип формирования обрабатывающего ядра (однокристалльные МП, секционные МП, заказные МП); прикладная специализация (универсальные МП, DSP, процессоры для параллельных вычислителей, микроконтроллеры и т.д.); технология изготовления, которая в значительной мере определяет быстродействие и потребляемую мощность.



## 2. Архитектура МП.

Архитектура МП - общая конфигурация его компонент, главные их характеристики и способы соединения.



Поскольку внутренняя архитектура МП определяется его местом в составе вычислительного устройства, рассмотрим обобщенную схему компьютера и определим функции его частей.

*Устройство управления* -

предназначено для выборки команд из ячеек внешней памяти, их интерпретации и выполнения.

Обычно содержит следующие регистры:

- PC - программный счетчик
- IR - регистр команд
- PSW - регистр признаков
- SP - указатель стека

*Управляющая память* -

содержит последовательность микрокоманд, образующих команду. Может быть доступна или недоступна пользователю.

*Рабочие регистры* -

ЗУ небольшого объема, предназначенное для хранения адресов и данных. В том случае, когда и адреса и данные хранятся в одних и тех же рабочих регистрах, эти регистры называются РОНами (Регистры Общего Назначения).

*АЛУ* -

арифметико-логическое устройство, выполняет арифметические и логические операции над содержимым РОНов, ячеек памяти и РС.

*Управление шиной* -

координирует взаимодействие центрального процессора с ВУ

*Внутренняя память* -

память небольшого объема внутри центрального процессора. Основное отличие от РОНов - способ адресации.

*Генератор синхронизации* -

формирует согласованные во времени последовательности импульсов для координации всех действий компьютера.

*Оперативная память (ОЗУ, ЗУПВ)* -

предназначена для команд и данных, используемых в текущих операциях, обычно быстродействующая, энергозависимая и сравнительно небольшого объема.

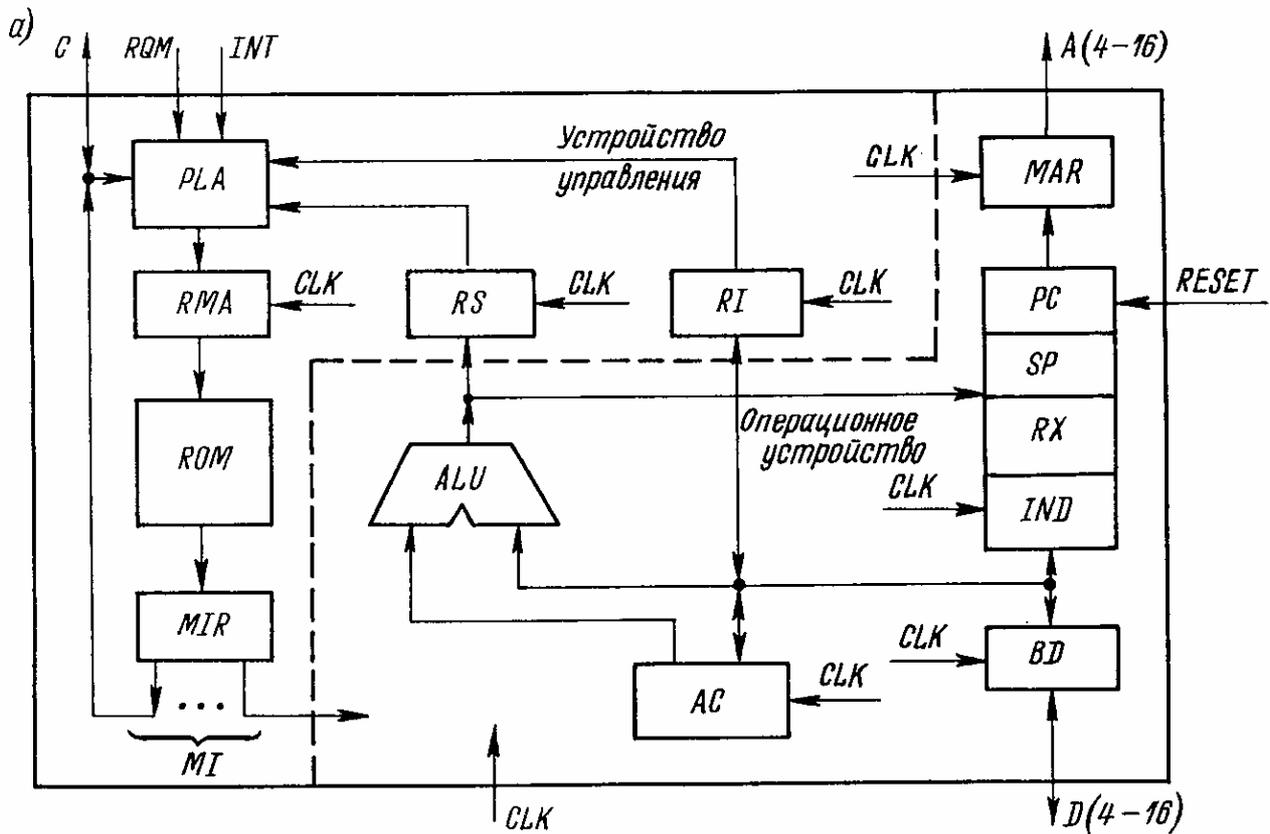
*Постоянная память (ПЗУ)* -

обычно память, допускающая только считывание или запись/считывание с относительно невысоким быстродействием, энергонезависимая. Используется для хранения общеупотребительных программ и данных.

*Интерфейсы* -

устройства сопряжения ВУ с каналом обмена компьютера

Аппаратные средства компьютера, объединенные понятием **ЦЕНТРАЛЬНЫЙ ПРОЦЕССОР**, обычно составляют содержимое МП-БИС. Для однокристалльных МП это действительно одна БИС, для секционированных - несколько. Обобщенная структура однокристалльного МП:



На схеме:

- MAR - регистр адреса памяти
- PC - счетчик адресов команд
- SP - указатель стека
- Rx - РОНы
- IND - индексный регистр
- BD - буфер данных
- AC - регистр-аккумулятор
- ALU - арифметико-логическое устройство
- RS - регистр состояний
- RMA - регистр микрокоманд
- ROM - память микрокоманд
- MIR - регистр адреса микрокоманды
- PLA - схема образования адресов микрокоманд
- RI - регистр команд

### Цикл работы МП

Адрес выполняемой команды генерирует счетчик команд PC. Адрес команды фиксируется в MAR и выдается через шину адреса A во внешнюю память. Содержимое проадресованной ячейки памяти - выполняемая команда - через BD попадает в МП и фиксируется в RI. Далее происходит ее дешифрация посредством PLA, результатом которой является последовательность адресов микрокоманд, соответствующих данной команде, которая после фиксации в RMA воздействует на ROM. Извлеченные из ROM микрокоманды фиксируются в MIR и обрабатываются прочими узлами МП.

Проиллюстрируем работу однокристалльного на примере микропрограммы команды сложения содержимого регистра-аккумулятора и содержимого ячейки памяти:

| Такт | Операция в микропроцессоре     | Комментарий  |
|------|--------------------------------|--|
| T1   | (PC):=(PC)+1<br>(MAR):=(PC)    | Инкремент счетчика команд                                    |
| T2   | (RI)=(D)                       | Считывание команды из памяти                                 |
| T3   | (MAR):=(IND)+(RI) <sub>a</sub> | Образование адреса операнда                                  |
| T4   | (AC):=(AC)+(D)<br>(RS):=f(AC)  | Считывание операнда и суммирование, установка признаков в RS |

### 3. Элементы программирования МП

Команда - единица информации, указывающая компьютеру, какую операцию необходимо выполнить. Поименованный набор команд составляет программу.

Здесь и далее примеры команд и программ приводятся на языке ассемблера МП i8080 (отечественный аналог - МП 580-й серии). Будучи давно (по меркам компьютерной индустрии) устаревшей моделью этот микропроцессор в смысле основных архитектурных принципов во многом является базовым решением для сегодняшних микропроцессоров универсального характера, поэтому практическая неактуальность данного МП не является препятствием к использованию его структуры и системы команд для получения основных понятий о функционировании МП.

Рассмотрим пример фрагмента программы для МП i8080, осуществляющей сложение 2-х чисел:

```
LDA    NUM1
MOV    B,A
LDA    NUM2
ADD    B
JZ     TASK
```

Первая команда передает первое слагаемое из ячейки памяти с адресом NUM1 в аккумулятор, вторая переносит его в регистр B, третья загружает в регистр второе слагаемое. Четвертая производит сложение, оставляя сумму в аккумуляторе, пятая заставляет процессор выполнять команду из ячейки памяти с адресом TASK в том случае, если сумма равна нулю.

Пример позволяет сделать несколько выводов: большинство команд связано с передачей информации из одной части системы в другую; некоторые команды служат только для этого или определяют местонахождение какой-либо информации. Поэтому очень важным для вычислительного устройства, каковым является МП, наличие надежных и гибких способов идентификации местонахождения информации, называемых режимами адресации. К числу основных режимов адресации относятся следующие:

*Прямая адресация* - адрес определяется как часть команды.

*Регистровая адресация* - операнд (число, над которым осуществляется операция) находится в регистре, а адрес регистра является частью команды.

*Косвенная адресация* - в ячейке, адрес которой является частью команды, находится адрес операнда.

*Базовая адресация* - адрес образуется сложением содержимого ячейки памяти или регистра с определенным числом, называемым смещением.

*Инкрементация/декрементация* - обращение к смежным или равномерно распределенным адресам.

*Относительная адресация* - адрес равен сумме числа и текущего содержимого РС.

*Непосредственная адресация* - операнд содержится непосредственно в команде.

Форматом команды называется ее конструкция со спецификацией частей. Команда состоит из кода операции (КОП), определяющего функцию команды, и/или параметров, уточняющих эту функцию.

Общий формат команды на языке ассемблера на примере команды MOV:

| <u>Метка:</u> | <u>Операция</u> | <u>Опреанды</u> | <u>;Комментарий</u>                         |
|---------------|-----------------|-----------------|---|
|               | <b>MOV</b>      | <b>В,А</b>      | <b>Пересылка содержимого регистра А в В</b> |

Метка - символическое наименование ячейки, содержащей команду. Комментарий - неисполняемая часть команды - пояснение к ней. Метка и комментарий являются необязательными частями команды.

### 3.1. Типовые программные процедуры.

#### Передача информации

Пусть необходимо передать три байта информации из ячеек с адресами 0080-0082 в ячейки 0120-0122 (здесь и далее в примерах программ числа шестнадцатиричные). Предположим, что ассемблер (программа-транслятор записей на языке ассемблера) ассоциирует адрес 0080 с символическим наименованием SRC, а адрес 0120 - с DES. Если программа пересылки начинается в ячейке 0050, то ее запись будет выглядеть следующим образом:

| Адрес | Машинный код | Ассемблерный код | Комментарий             |
|-------|--------------|------------------|-------------------------|
| 50    | 21           | LXI H, SRC       | ; Загрузка адреса SRC   |
| 51    | 80           |                  | ; в пару HL             |
| 52    | 00           |                  |                         |
| 53    | 46           | MOV B, M         | ; Загрузка B из SRC     |
| 54    | 23           | INX H            | ; Инкремент адреса      |
| 55    | 4E           | MOV C, M         | ; Загрузка C из SRC + 1 |
| 56    | 23           | INX H            | ; Инкремент адреса      |
| 57    | 56           | MOV D, M         | ; Загрузка D из SRC + 2 |
| 58    | 21           | LXI H, DES       | ; Загрузка адреса DES   |
| 59    | 20           |                  | ; в пару HL             |
| 5A    | 01           |                  |                         |
| 5B    | 70           | MOV M, B         | ; Запись (B) в DES      |
| 5C    | 23           | INX H            | ; Инкремент адреса      |
| 5D    | 71           | MOV M, C         | ; Запись (C) в DES + 1  |
| 5E    | 23           | INX H            | ; Инкремент адреса      |
| 5F    | 72           | MOV M, D         | ; Запись (D) в DES + 2  |
| 60    | *            | *                |                         |

#### Арифметически процедуры

Пусть необходимо просуммировать содержимое 0110 и 0111, сумму вычесть из содержимого 0120, к разности прибавить 6 и результат запомнить в 0070. Предположим, что адресам 0070, 0110 и -120 присвоены символически наименования ANS, NOS и TOTL, а программа начинается в ячейке 0030, тогда ее запись будет выглядеть следующим образом:

| Адрес | Машинный код | Ассемблерный код | Комментарий                 |
|-------|--------------|------------------|-----------------------------|
| 30    | 21           | LXI H, NOS       | ; Загрузка адреса NOS в HL  |
| 31    | 10           |                  |                             |
| 32    | 01           |                  |                             |
| 33    | 7E           | MOV A, M         | ; Загрузка (NOS) в A        |
| 34    | 23           | INX H            | ; Инкремент адреса          |
| 35    | 86           | ADD M            | ; Прибавление (NOS + 1) к A |
| 36    | 47           | MOV B, A         | ; Передача (A) в B          |
| 37    | 3A           | LDA TOTL         | ; Загрузка (TOTL) в A       |
| 38    | 20           |                  |                             |
| 39    | 01           |                  |                             |
| 3A    | 90           | SUB B            | ; Вычитание (B) из (A)      |
| 3B    | C6           | ADI 6            | ; Прибавление 6 к A         |
| 3C    | 06           | STA ANS          | ; Запоминание в ANS         |
| 3D    | 32           |                  |                             |
| 3E    | 70           |                  |                             |
| 3F    | 00           |                  |                             |

#### Логические процедуры

Большинство применений логических процедур связано с селективными установками, сбросом, изменениями и проверками отдельных бит в байте. Эти действия осуществляются с помощью логических операций над нужным байтом и вторым байтом, называемом маской:

| Адрес | Машинный код | Ассемблерный код |          | Комментарий              |
|-------|--------------|------------------|----------|--------------------------|
| 10    | 21           | LXI              | H, MASKS | ; Загрузка адреса        |
| 11    | 30           |                  |          | ; масок в HL             |
| 12    | 00           |                  |          |                          |
| 13    | 3A           | LDA              | OTBL     | ; Загрузка (CTRL) в A    |
| 14    | 40           |                  |          |                          |
| 15    | 00           |                  |          |                          |
| 16    | B6           | ORA              | M        | ; Установка бит 2 и 3    |
| 17    | 23           | INX              | H        | ; Инкремент до MASKS + 1 |
| 18    | A6           | ANA              | M        | ; Сброс бита 6           |
| 19    | 23           | INX              | H        | ; Инкремент до MASKS + 2 |
| 1A    | AE           | XRA              | M        | ; Изменение бита 5       |
| 1B    | 32           | STA              | CTRL     | ; Запоминание в CTRL     |
| 1C    | 40           |                  |          |                          |
| 1D    | 00           |                  |          |                          |
| 1E    | A6           | ANA              | M        | ; Проверка бита 5        |
| 1F    | *            |                  |          |                          |

### Переходы

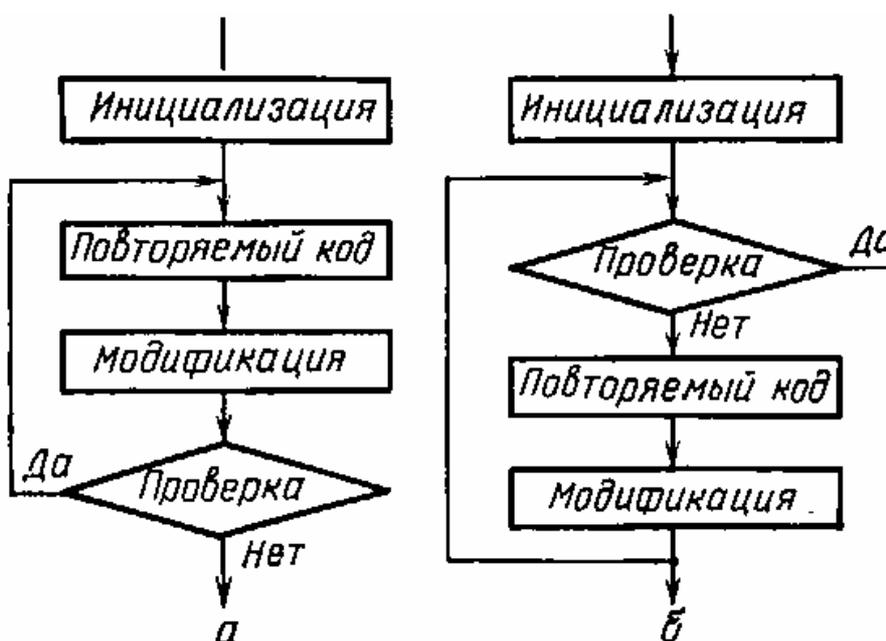
Обычно команды выбираются из последовательных ячеек памяти. Однако при этом возможно выполнение только таких программ, которые не допускают какого-либо ветвления в зависимости от складывающейся вычислительной ситуации. Для автоматического управления ходом выполнения программы используются команды условного и безусловного переходов. Безусловные переходы заставляют программу совершить переход вне зависимости от условий работы МП, условные - если имеет место определенная комбинация условий, которые фиксируются в регистре признаков. Пример команды условного перехода:

JNZ 0040

Данная команда проверяет состояние бита Z регистра признаков и заставляет МП выполнить команду из ячейки 0049 в том случае, если не обнаружит в бите Z логической единицы. Содержимое бита Z регистра признаков зависит от результатов выполнения очередной команды: если результат равен нулю, то в бите Z устанавливается логическая единица

### Организация циклов

Процедура повторяющегося выполнения одного и того же набора команд называется циклом. Цикл - один из основных программных объектов, реализующий возможность компьютерной техники выполнять огромные объемы рутинной вычислительной работы.



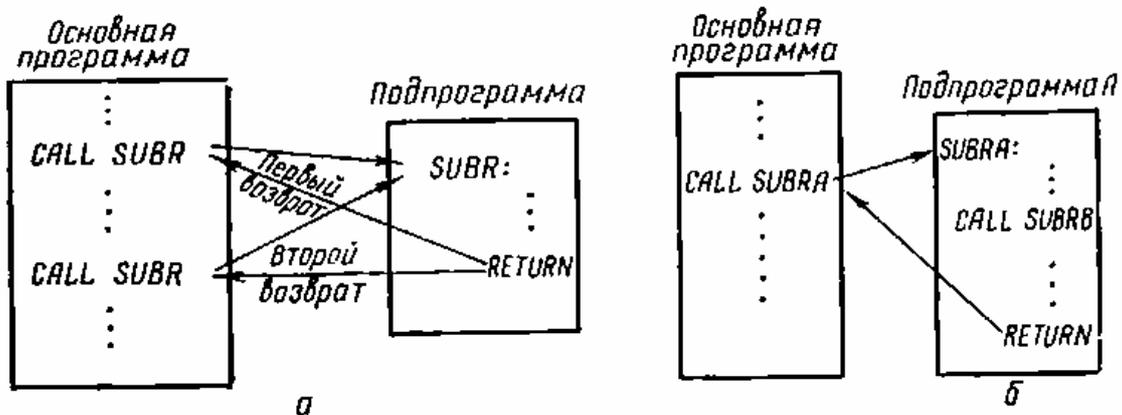
Пусть необходимо передать  $50_{16}$  данных из области с начальным адресом (символическое обозначение SRC) в область с начальным адресом 0040 (символическое обозначение DES). Тогда программа для цикла будет выглядеть следующим образом:

|    |    |       |      |          |                   |
|----|----|-------|------|----------|-------------------|
| 40 | 2E |       | MVI  | L, BSIZE | } Инициализация   |
| 41 | 42 |       |      |          |                   |
| 42 | 11 |       | LXI  | D, SRC   |                   |
| 43 | 00 |       |      |          |                   |
| 44 | 02 |       |      |          |                   |
| 45 | 01 |       | LXI  | B, DES   | } Повторяемый код |
| 46 | 00 |       |      |          |                   |
| 47 | 04 | LOOP: | LDAX | D        |                   |
| 48 | 1A |       | STAX | B        |                   |
| 49 | 02 |       |      |          |                   |
| 4A | 13 |       | INX  | D        | } Модификация     |
| 4B | 03 |       | INX  | B        |                   |
| 4C | 2D |       | DCR  | L        |                   |
| 4D | C2 |       | JNZ  | LOOP     |                   |
| 4E | 48 |       |      |          | } Проверка        |
| 4F | 00 |       |      |          |                   |

### Подпрограммы

В случае, когда один и тот же сегмент программы необходимо использовать в произвольных ее местах (к примеру вычисление синуса), этот сегмент удобно оформить в виде автономной программы.

Организованный таким образом автономный программный сегмент называется подпрограммой, переход к ней называется вызовом подпрограммы.



При вызове подпрограммы требуется ответить на несколько вопросов:

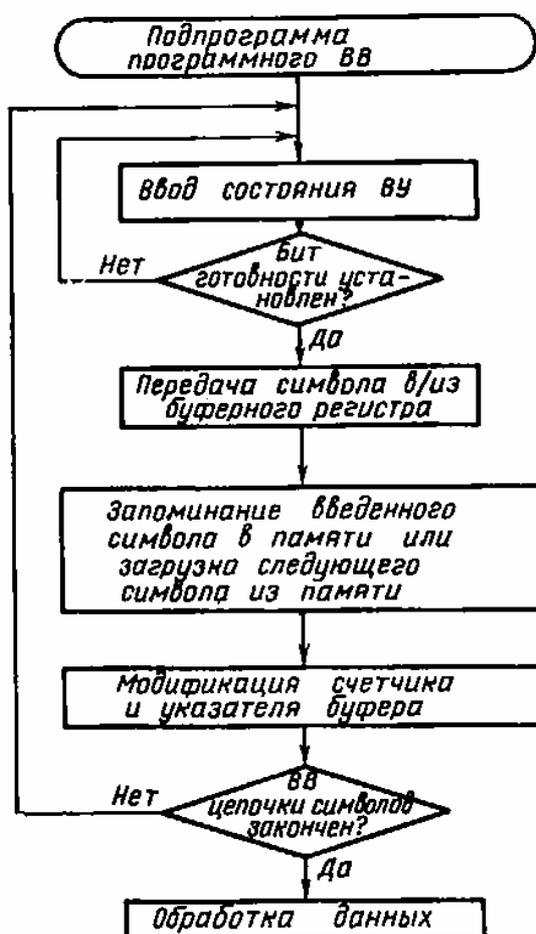
1. Где запомнить текущее содержимое РС, чтобы подпрограмма знала, куда возвращаться? - Автоматически запоминается в стеке при использовании команды вызова подпрограммы CALL
2. К какому адресу перейти? - Задается как параметр команды CALL
3. Как подпрограмма найдет адрес возврата? - С помощью команды RETURN

4. Как и где запомнить текущее содержимое РОНов, чтобы оно не разрушалось подпрограммой? - Команда CALL автоматически отправляет содержимое РОНов в стек.
5. Как сообщить подпрограмме адрес преобразуемых данных? - Параметры CALL.

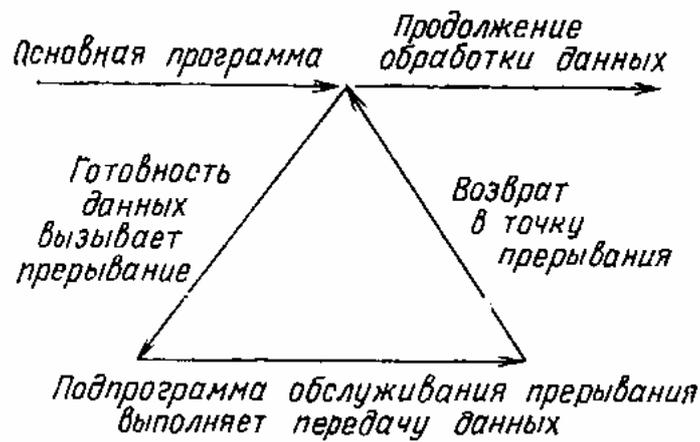
### Программирование ввода/вывода

Передача данных между МП и ВУ называется вводом/выводом (ВВ). Для организации ВВ используются 3 основных способа:

*Программный ВВ* - самый простой способ, требует минимума дополнительной логики, выполняется полностью под управление МП. В ВУ выделяются 2 регистра - регистр готовности и буферный регистр. МП обращается сначала к регистру готовности и обнаружив там признак наличия таковой начинает обмен через буферный регистр:



*ВВ по прерываниям* - основное достоинство программного ВВ - его простота, но при этом МП тратит время на ожидании готовности (доля времени полезной работы может составлять менее 1%). Работу МП можно разделить во времени между 2-мя независимыми программами - программа заднего плана обработки данных и программа ВВ переднего плана. При этом вторая программ производит прерывание первой:



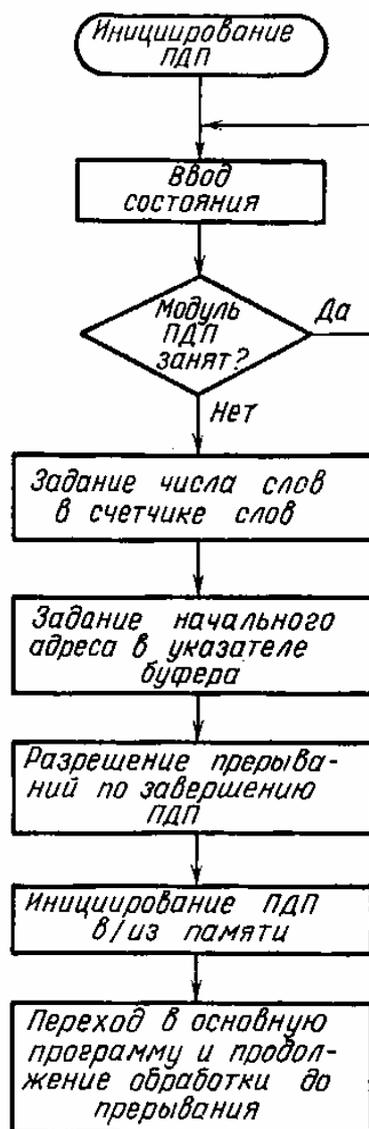
Когда в МП i8080 поступает запрос на прерывание - т.е. сигнал о готовности ВУ к ВВ, выполняются следующие действия:

- завершается выполнение текущей команды
- обнуляется бит INTA слова состояния процессора

- выполняется команда RST N

где N - целое число в диапазоне 0..7. Эта команда отсылает содержимое PC в стек и передает управление команде с адресом 8N, который называется вектором прерывания. Содержимое ячеек 8N..8N+7 определяет адрес, к которому переходит программа для нахождения нужной программы обслуживания прерывания.

*Прямой доступ к памяти* - в быстродействующих ВУ возникает необходимость передачи больших объемов данных за короткий интервал времени. Это достигается путем непосредственной передачи между ВУ и ОЗУ минуя МП (прямой доступ к памяти - ПДП). При этом процессом передачи управляет специальный модуль ПДП без участия МП:



### 3.2. Общие вопросы программирования МП

#### Эффективность программ

При проектировании программ следует в первую очередь учитывать три основных фактора:

- Пространство памяти*, необходимое для хранения и работы команды;
- Число команд* в программе;
- Время выполнения* программы и наиболее важных ее частей.

Одновременная оптимизация по всем критериям невозможна, поскольку они взаимно противоречивы. Объем памяти критичен для небольших систем. Уменьшение числа команд минимизирует время, требуемое для отладки программы. Время выполнения программы становится преобладающим критерием для приложений, работающих в реальном масштабе времени.

## Документирование программ

ГОСТ19.101-77 предусматривает следующие компоненты комплекта документации на программу:

*Техническое задание* - определяет назначение программы, требования к ней, сроки разработки, виды испытаний.

*Формуляр* - основные характеристики программы, комплектность.

*Текст программы* - запись программы с комментариями.

*Описание программы* - сведения о логической структуре и функционировании программы.

*Пояснительная записка* - схема и описание алгоритма и функционирования программы, обоснование принятых решений.

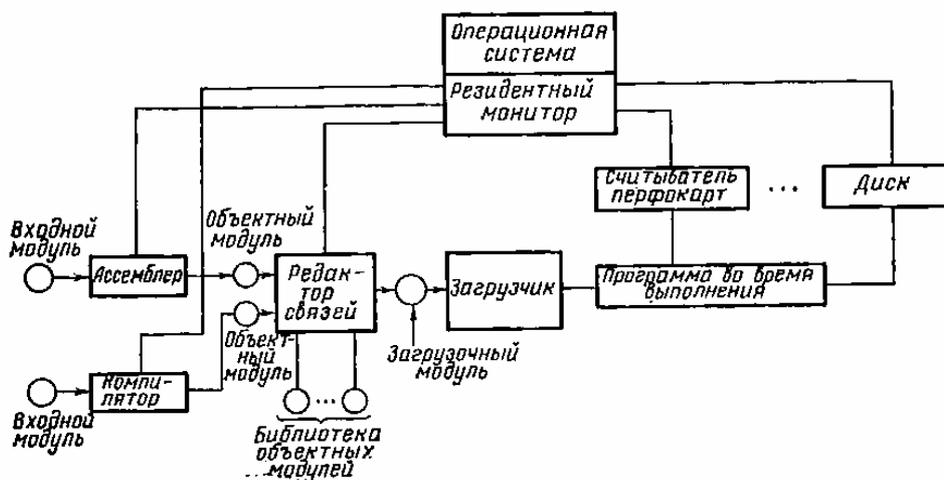
*Руководство программиста* - сведения об установке, настройке и применении программы.

*Руководство пользователя* - сведения об эксплуатации программы.

## Создание и выполнение программы

При программировании на языке машинных кодов - последовательностей логических нулей и единиц, непосредственно обрабатываемых МП - программист самостоятельно решает где и как размещаются команды и данные. Такой подход позволяет создавать наиболее эффективные программные коды, но практически возможен только в случае очень простых программ.

При программировании на языке ассемблера (или языке более высокого уровня) процедура создания и выполнения программы становится более сложной, но программист в ней почти не участвует благодаря наличию специальных сервисных программ.



## **4. Аппаратные решения для МП широкого применения**

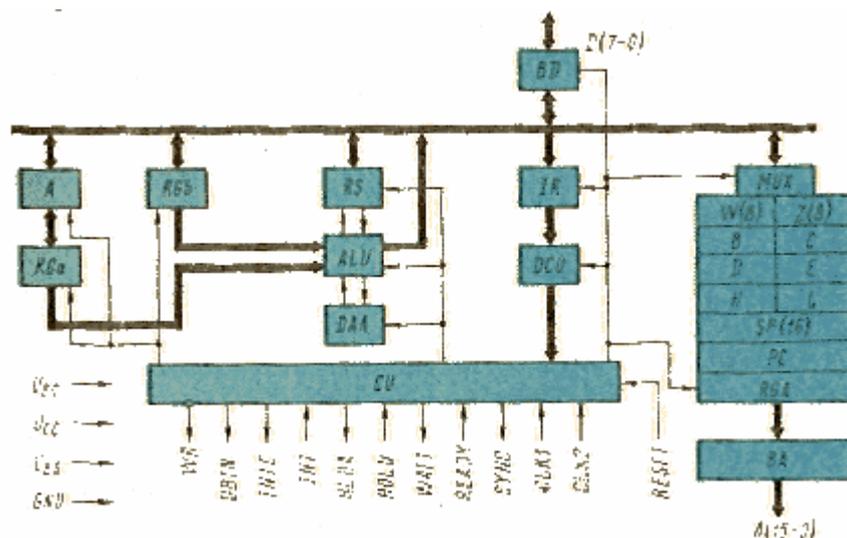
В качестве образцов типовых МП универсального характера рассмотрим архитектуру МП i8080 и МП семейства Pentium. Такой выбор обусловлен следующими соображениями. МП i8080 в свое время стал базовой платформой для разработчиков МП-техники, а технические решения, принятые при создании этого МП, и в настоящее время во многом являются основой архитектуры процессоров, ориентированных на широкий спектр приложений. В то же время структура i8080 или его аналога 580ИК80 сравнительно проста и поэтому удобна при изучении

структуры МП. Устройства же семейства Pentium хорошо отражают современный уровень развития МП и являются наиболее массовой аппаратной платформой компьютерной техники в настоящее время.

#### 4.1. МП i8080

МП i8080 выполнен n-МОП технологии, согласуется по напряжениям логических уровне с микросхемами ТТЛ. Максимальная тактовая частота 2,5 МГц, потребляемая мощность <1,25 Вт, напряжения питания 12В, +5В, -5В.

##### Структура МП i8080



- A - аккумулятор
- Rga, RGb - регистры временного хранения операндов
- RS - регистр признаков
- ALU - арифметико-логическое устройство
- DAA - двоично-десятичный корректор
- IR - регистр команд
- DCU - дешифратор команд
- CU - устройство управления
- BD - буфер данных
- MUX - мультиплексор
- W, Z - вспомогательные регистры
- B, C, D, E, H, L - РОНЫ
- SP - регистр указатель стека
- PC - программный счетчик
- RGA - регистр адреса
- BA - буфер адреса

МП имеет 16-разрядную однонаправленную трехстабильную шину адреса А, 8-разрядную трехстабильную двунаправленную шину данных D, 6 входных и 6 выходных сигналов управления. Обеспечивает адресацию 64К байт внешней памяти, 256 ВУ ввода и 256 ВУ вывода.

## Слово состояния программы

Для управления процессом выполнения программы используется двух байтное слово состояния программы, первый байт которого представляет содержимое регистра-аккумулятора, а второй - содержимое регистра признаков, в котором фиксируются признаки результата выполнения любой операции в АЛУ:

| Бит регистра признаков | Обозначение | Содержимое                                     |
|------------------------|-------------|--|
| 0                      | СУ          | 1, если есть перенос из старшего разряда       |
| 2                      | Р           | 1, если результат содержит четное число единиц |
| 4                      | АС          | 1, если есть перенос из 3-го разряда           |
| 6                      | Z           | 1, если результат = 0                          |
| 7                      | S           | 1, если результат < 0                          |

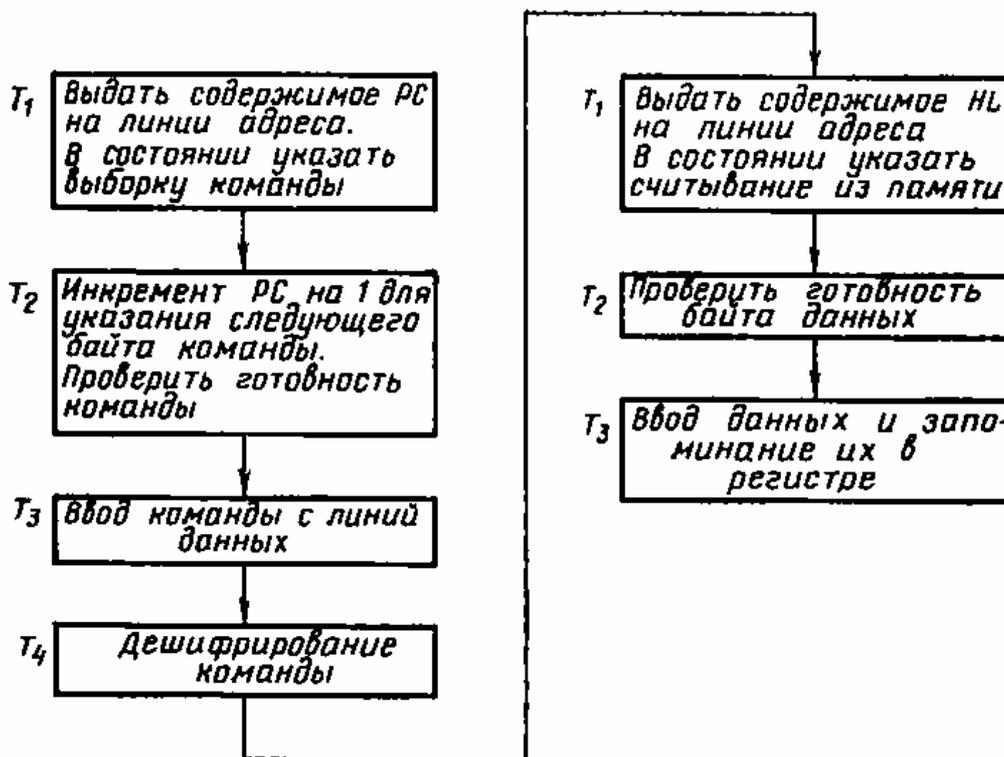
## Слово состояния процессора

Для формирования управляющих сигналов при обращении к ЗУ или ВУ, а также для организации различных режимов работы МП-системы используется однобайтное слово состояния процессора, которое выдается на шину D в такте T2 каждого машинного цикла

| Разряд шины D | Обозначение | Назначение                |
|---------------|-------------|---------------------------|
| D0            | INTA        | Подтверждение прерывания  |
| D1            | WO          | Запись                    |
| D2            | STACK       | На шине A - содержимое SP |
| D3            | HLTA        | Подтверждение останова    |
| D4            | OUT         | На шине A код ВУ вывода   |
| D5            | MI          | Прием 1-го байта команды  |
| D6            | INP         | На шине A код ВУ ввода    |
| D7            | MEMR        | Чтение из ЗУ              |

## Синхронизация работы

Время выполнения команды (которая может быть 1, 2 и 3-байтной) зависит от типа команды и составляет от 1 до 5 машинных циклов. Длительность машинного цикла - от 3 до 5 машинных тактов (T1..T5). Длительность машинного такта равна периоду тактовой частоты генератора синхронизации. На иллюстрации приведена последовательность смены состояний процессора при выполнении команды MOV r,M - запись в регистр содержимого ячейки памяти.



#### 4.2. Способы повышения производительности МП

Основными направлениями разработок, ориентированных на повышение производительности МП, которые велись с момента появления i8080, являются:

*Увеличение частоты и разрядности* - сегодня серийно выпускаемые МП имеют тактовую частоту до 500 МГц и разрядность 64-128 бит

*Конвейеризация этапов отработки команд* - организация структуры МП в виде конвейера, звенья которого обрабатывают отдельные этапы выполнения команды, позволяет совмещать во времени выполнение нескольких последовательных команд.

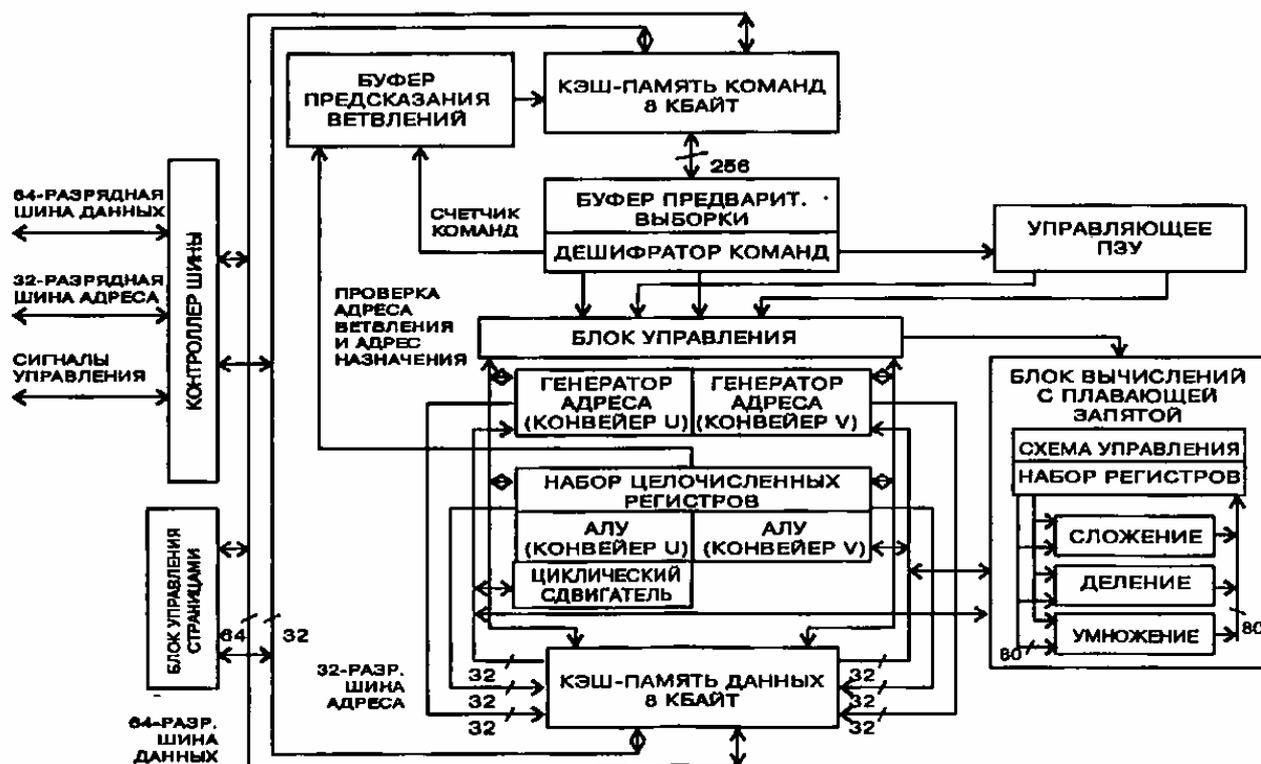
*Использование кэш-памяти* - быстродействующей памяти небольшого объема непосредственно на кристалле МП, содержащей команды и данные, которые обрабатываются в текущий момент. Позволяет значительно уменьшить количество медленных операций обращения к внешней памяти.

*Использование сопроцессоров* - вычислителей, работающих параллельно с обрабатывающим ядром МП и ускоренно выполняющих специальные операции, например арифметические процедуры над числами, представленными в формате с плавающей запятой.

*Использование RISC-процессоров* - МП с уменьшенным набором команд. В уменьшенный набор входят наиболее часто выполняемые команды, отработка которых организована аппаратно, что дает заметный выигрыш в быстродействии.

### 4.3. МП семейства Pentium

Все способы повышения производительности МП, приведенные в предыдущем разделе, реализованы в архитектуре процессоров семейства Pentium. Кроме того данные МП содержат ряд специфических решений, улучшающих их параметры. Pentium - суперскалярный RISC-процессор, изготавливаемый по 0,8-микронной технологии BiCMOS, полностью совместимый со старшими моделями x86.



Основу МП составляют два пятиступенчатых конвейера, позволяющие выполнять две целочисленные команды (а в некоторых случаях - и две операции с плавающей запятой) за один такт синхронизации. Конвейеры не идентичны: один из них (u-pipe) может выполнять любые операции, как целочисленные, так и с плавающей запятой, тогда как второй (v-pipe) реализует лишь часть целочисленных команд и одну команду с плавающей запятой.

Сопроцессор для работы с данными в формате с плавающей запятой, размещенный на основном кристалле МП, использует семиступенчатый конвейер с аппаратной реализацией основных операций - сложения, умножения и деления.

Ускорить вычисления за счет уменьшения частоты обращения к внешнему ОЗУ помогают два блока кэш-памяти - для команд и данных. Двух канальная множественно-ассоциативная кэш-память данных является двухпортовой, что обеспечивает одновременную работу с обоими конвейерами.

Новым решением является система предсказания ветвления, которая ощутимо повышает скорость выполнения циклов

## Процессора Pentium III

Значительный рост производительности процессоров Pentium III (последняя проанонсированная генерация МП семейства Pentium на момент подготовки настоящего КОНСПЕКТА ЛЕКЦИЙ) по сравнению с предшествующими поколениями процессорных устройств на основе архитектуры Intel объясняется сочетанием возможностей микроархитектуры P6, новых SIMD-инструкций, технологии MMX и функции серийного номера процессора. Результатом стала не только возросшая производительность существующих приложений, но и обеспечение ее запаса для программ будущего, полностью реализующих возможности, заложенные в процессор Pentium III.

### *Основные характеристики*

Тактовая частота 500 и 450 МГц. При изготовлении используется 0.25-микронный технологический процесс, позволяющий повысить тактовую частоту ядра и уменьшить потребляемую мощность. Включает новые SIMD-инструкции - 70 новых команд, улучшающих работу с приложениями трехмерной графики, потокового аудио, видео и распознавания речи. Также включает технологию Intel MMX. Процессор Pentium III с тактовой частотой 500 МГц более чем на 93% превышает быстродействие процессора Pentium® II с тактовой частотой 450 МГц при работе с трехмерной графикой (по результатам эталонного теста 3D WinBench\* 99-3D Lighting and Transformation), и на 42% - при работе с приложениями мультимедиа (по результатам эталонного теста MultimediaMark\* 99). Архитектура Двойной независимой шины (DIB) увеличивает пропускную способность и производительность по сравнению с процессорами с единственной шиной данных. Процессор Pentium III использует набор микросхем Intel® 440BX AGPset. Содержит функцию серийного номера процессора, первый компонент системы обеспечения безопасности ПК, предлагаемой корпорацией Intel. Неблокируемая кэш-память первого уровня емкостью 32 Кб (16Кб/16Кб) и унифицированная неблокируемая кэш-память второго уровня емкостью 512 Кб обеспечивают ускоренный доступ к часто используемым данным. Процессор Pentium III поддерживает кэширование памяти с объемом адресного пространства 4 Гб. Позволяет создавать масштабируемые системы с двумя процессорами и физической памятью объемом до 64 Гб.

### *Технология динамического исполнения микроархитектуры P6:*

Предсказание множественных ветвлений прогнозирует исполнение программ по нескольким ветвям. Анализ потока данных оптимизирует и реорганизует последовательность исполнения команд на основе используемых в них данных. Спекулятивное исполнение команд - это исполнение команд на основе оптимизированной последовательности, которая обеспечивает постоянную загрузку исполняющих блоков процессора, что ведет к повышению общей производительности.

### *Потоковые SIMD-расширения для Internet:*

Потоковые SIMD-расширения Internet представляют собой 70 новых команд, среди которых команды для вычислений с плавающей запятой (метод SIMD), дополнительные команды для целочисленных вычислений (метод SIMD) и команды управления кэшированием. Внедрение потоковых SIMD-расширений позволяет существенно увеличить скорость работы мощных приложений обработки изображений, трехмерной графики, видео, аудио, а также систем распознавания речи.

### *Технология Intel® MMX™:*

Горбунов А.И. Конспект лекций по дисциплине «Микропроцессорные устройства РЭС»

Технология MMX, по сути является набором 57 команд общего назначения для целочисленных операций, легко применимых к широкому спектру мультимедийных и коммуникационных приложений.

#### *Серийный номер процессора Intel®:*

Серийный номер процессора - это первый из предлагаемых корпорацией компонентов системы безопасности для ПК. Являясь уникальным идентификатором данного процессора, он может быть использован и для идентификации компьютера или пользователя в сети или со стороны прикладных программ. Серийный номер процессора будет использоваться в следующих типах приложений, для которых надежная идентификация пользователя или системы является особенно необходимой:

Приложения с повышенным уровнем защиты - управляемый доступ к службам Internet, обмен электронными документами. Управляющие приложения - управление ресурсами, удаленная загрузка и конфигурирование систем. Управление информационными ресурсами - службы технической поддержки, защита резервируемых данных.

#### *Другие особенности процессора Pentium® III:*

Разработанный Intel корпус S.E.C.2. облегчает массовое производство процессоров, безопасность при использовании, а также единый форм-фактор высокопроизводительных процессоров будущего. Высокопроизводительная архитектура двойной независимой шины (системная шина и шина кэш-памяти), создает резерв пропускной способности, производительности и расширяемости для систем и технологий будущего. Системная шина поддерживает возможность множественной обработки запросов, расширяя полосу пропускания. Кроме того, она пригодна для построения недорогой, двунаправленной, симметричной многопроцессорной системы (до двух процессоров), обеспечивающей значительный скачок производительности при работе с многозадачными операционными системами и многопоточными приложениями. Встроенная неблокируемая кэш-память второго уровня емкостью 512 Кб повышает производительность путем сокращения среднего времени доступа к памяти и обеспечения быстрого повторного обращения к командам и данным. Производительность увеличивается и благодаря наличию выделенной 64-разрядной шины кэш-памяти. Частота кэш-памяти второго уровня пропорциональна частоте ядра. Кроме того, в процессор встроена кэш-память первого уровня общей емкостью 32 Кб: 16 Кб для команд и столько же для обработки данных. Процессоры Pentium III с тактовой частотой 450 и 500 МГц поддерживает кэширование до 4 Гб адресуемой памяти. Шина кэш-памяти второго уровня имеет поддержку кода исправления ошибок (ECC), что важно при работе с приложениями, требующими интенсивной обработки и надежности данных. Конвейерный блок вычислений с плавающей запятой (FPU) поддерживает 32 и 64-разрядные форматы в соответствии со спецификацией IEEE 754, а также 80-разрядный формат. Повышенная целостность и надежность данных обеспечивается наличием сигналов адресации/запроса и ответа системной шины с контролем четности и механизмом повторных запросов.

#### *Встроенные средства самотестирования и контроля производительности процессора Pentium III:*

Встроенный механизм самотестирования (Built-in Self Test — BIST) обеспечивает постоянный контроль зависаний и сбоев в микрокоде и больших логических матрицах, а также тестирование кэш команд и кэш данных, буферов TLB и сегментов памяти ROM. Механизм стандартного порта доступа к тестированию и периферийному сканированию

IEEE 1149.1 дает возможность осуществлять проверку каналов связи между процессором Pentium III и системой через стандартный интерфейс. Встроенные счетные устройства следят за показателями производительности и ведут подсчет событий. Встроенный в кристалл диод может следить за температурой кристалла. Датчик температур, расположенный на системной плате, контролирует температуру кристалла процессора Pentium III, знание которой необходимо для управления температурным режимом.

## 5. Цифровые сигнальные процессоры

Цифровые сигнальные процессоры (Digital Signal Processor - DSP) представляют собой специализированные процессоры с RISC-архитектурой, предназначенные для решения задач ЦОС.

ЦОС охватывает широкий спектр применений. К ним относятся цифровая фильтрация, кодирование и декодирование информации, распознавание речи, обработка изображений, спектральный анализ, управление системами. В РЭС типовыми задачами ЦОС являются реализации различных видов ЦФ, перенос спектра сигнала из одной частотной области в другую, вычисление энергетического спектра, корреляционных функций. Во большинстве случаев эти задачи решаются с применением алгоритма быстрого преобразования Фурье (БПФ).

### 5.1. Процедура БПФ

БПФ - специальная версия дискретного Фурье - преобразования (ДПФ), позволяющего представить сигналы в виде суперпозиции гармонических функций. ДПФ, описывается парой соотношений (прямым и обратным):

$$G(f_k) = 1/N \cdot \sum [X(t_n)] \cdot \exp(-j2\pi k n/N) \quad (1)$$

$$X(t_n) = \sum [G(f_k)] \cdot \exp(j2\pi k n/N) \quad (2)$$

где  $t_n = n \cdot \Delta t$ ,  $f_k = k \cdot \Delta f = k \cdot 1/T$  выполняет блочный анализ временного сигнала.

Блок состоит из  $N$  отсчетов входного процесса, что соответствует длине реализации  $T$ . При этом в ДПФ подразумевается, что анализируемый блок является одним из периодов бесконечной периодической функции. Соответственно, получаемый частотный спектр (т.е. зависимость амплитуд и фаз гармонических составляющих сигнала от частоты) будет линейчатым, т.е. состоящим из отдельно расположенных вдоль частотной оси с интервалом линий, определяющих различные гармонические компоненты временного сигнала.

Если временная функция содержит только действительные значения, что является случаем, часто встречающимся на практике, то спектр сигнала является сопряженной линейчатой функцией, т.е. компоненты положительной и отрицательной областей частот имеют одинаковые амплитуды, но противоположные фазы. Иначе говоря, только  $N/2$  отсчетов в частотной области являются независимыми, а именно те, что расположены на отрезке  $(0, f_k)$ .

В общем случае при вычислении ДПФ от комплексной входной последовательности необходимо производить вычисление всех  $N$  спектральных компонент, так как в данном случае спектр не обладает свойством комплексно-сопряженной симметрии.

Еще одна особенность ДПФ связана с представлением анализируемого сигнала как одного из интервалов бесконечной периодической функции. Если начальные и конечные значения сигнала на периоде различны, то в моменты времени  $0, T, 2T, \dots$  происходит так называемый скачок функции, который может привести к существенному искажению спектра. Для устранения негативного влияния на спектр данной разрывности и получения лучшей избирательности необходимо использовать сглаживание с помощью одного из видов весовой функции (окна).

Возможность использования ДПФ при вычислении цифровых сверток вытекает из фундаментального свойства преобразования Фурье, гласящего, что Фурье-образ произведения двух временных сигналов эквивалент свертке преобразований от каждого из сомножителей и, наоборот, свертка сигналов во временной области соответствует умножению их Фурье-отображений и в частотной области. Поэтому возможно вычисление циклической (периодической) свертки косвенным путем - посредством выполнения обратного преобразования Фурье от произведения ДПФ исходных периодических временных последовательностей. Такое решение имеет существенное практическое значение в том случае, когда преобразования могут быть выполнены быстрее, чем сама свертка.

Эффективными процедурами вычисления ДПФ являются алгоритмы быстрого преобразования Фурье (БПФ), начало разработки которых было положено исследованиями Кули и Тьюки. Общий принцип построения алгоритмов БПФ заключается в разложении процесса нахождения ДПФ на вычисление преобразований последовательностей меньшего размера. Действительно, как видно из выражения (1), суть ДПФ заключается в получении суммы произведений вида

$$X(k) * W^{-nk} \quad (3)$$

где  $W^{-nk} = \exp(-j2\pi/N)$

Весовая функция является периодической с интервалом  $N$ . Для заданного  $k$  при изменении  $n$  от нуля до  $N-1$  произведение (3) также меняется периодически, причем число этих периодов равно  $k$ . Более того, даже в пределах одного периода могут образовываться комплексно-сопряженные пары. Используя данные свойства, можно существенно уменьшить необходимое для вычислений число умножений. Для этого входная последовательность  $X(t_n)$  разбивается на две вспомогательные последовательности -  $Y(t_n)$  и  $Z(t_n)$  - так, что в  $Y(t_n)$  попадают отсчеты только с четными номерами, а в  $Z(t_n)$  - с нечетными, т.е.  $Y(t_n) = X(t_{2n})$ ,  $Z(t_n) = X(t_{2n+1})$ .

Тогда выражение (1) записывается в виде:

$$G(f_k) = 0,5 * (Y(f_k) + Z(f_k) * W^k) \quad (4)$$

Вычислительные затраты на реализацию  $N$  - точечного ДПФ в соответствии с выражением (4) составляют по  $(N/2)$  - точечных ДПФ  $Y(f_k)$  и  $Z(f_k)$  и дополнительно  $N$  операций на их

соединение, тогда как прямое вычисление  $G(f_k)$  по (1) требует  $N^2$  операций, что почти в два раза больше.

В выражении (4) индекс  $k$  изменяется от 0 до  $N-1$ . Однако  $Y(f_k)$  и  $Z(f_k)$  имеют период  $N/2$  и вычисляются только в диапазоне от 0 до  $N/2-1$ . Используя свойство периодичности функции  $W^k$  можно найти искомое соотношение и для  $N/2 < k < N-1$ . Окончательно получаем :

$$\begin{aligned} G(f_k) &= 0,5 * (Y(f_k) + Z(f_k) * W^k) && \text{при } 0 < k < N/2 - 1 \\ G(f_k) &= 0,5 * (Y(f_{k-N/2}) - Z(f_{k-N/2}) * W^k) && \text{при } N/2 < k < N - 1 \end{aligned} \quad (5)$$

Каждое из двух  $N/2$ -точечных преобразований ( если число их членов четно) может быть найдено повторением указанного приема, т.е. последовательности  $Y(t_n)$  и  $Z(t_n)$  снова разбиваются на две последовательности длиной  $N/4$ , причем для каждой их пары подбираются соответствующие весовые коэффициенты. Если является степенью числа 2, то последовательное прореживание входных значений может быть продолжено до получения двухточечных преобразований.

Данный алгоритм получил название БПФ с прореживанием по времени. Базовая операция его, называемая "бабочкой" (баттерфляй), состоит в получении выходных чисел  $P$  и  $Q$  из входных  $A$  и  $B$  в соответствии с выражением

$$P = A + W B; \quad Q = A - W B. \quad (6)$$

Очевидно, что значения, получаемые в результате выполнения базовой операции, могут быть хранимы в запоминающем устройстве на месте исходных операндов, что существенно сокращает требуемый объем запоминающих устройств. Алгоритм БПФ, в котором реализуется такая переадресация данных, получил название алгоритма с замещением.

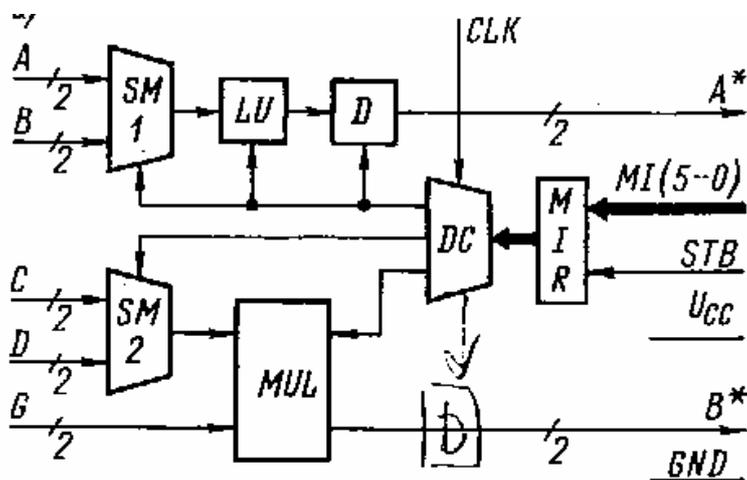
Второй распространенной формой алгоритма БПФ является метод прореживания по частоте, базовая операция БПФ с прореживанием при этом имеет вид:

$$P = A + B; \quad Q = (A - B)W \quad (7) .$$

## 5.2. Программируемый процессор цифровой обработки сигналов 1815ВФ3.

Данный МП является хорошим образцом специализированного DSP. Микросхема предназначена для построения процессоров быстрого преобразования Фурье и различных цифровых фильтров многопроцессорных систем высокой производительности. Она осуществляет обработку последовательных знакоразрядных двоичных кодов цифра за цифрой, начиная со старших разрядов с фиксированной запятой. На частоте 10 МГц БИС обеспечивает обработку базовой операции "бабочка" под 24-разрядными операндами за 2,5 мкс. Разрядность данных для сумматоров - произвольная, для умножителя - не менее 10 бит; точность выполнения операций в сумматорах - абсолютная, в умножителе - не менее 24 бит; производительность - до 1,2 млн арифметических операций в секунду. При организации процессоров БПФ и различных фильтров, как правило, используется несколько микропроцессоров, входы и выходы которых коммутируются согласно алгоритму преобразования данных. БИС выполнена по ИИЛ-технологии, уровни логических сигналов согласуются с ТТЛ-схемами.

В состав БИС входят: 2-входные сумматоры с регистрами данных на входах (SM1, SM2); устройство выполнения логических операций (LU); схема программируемой задержки информации (D); множительное устройство (MUL); дешифратор микрокоманд (DC); регистр микрокоманд (MIR), задающий операцию микропроцессора.



Описание выводов БИС приведено в таблице. Для приема и выдачи информации используются парафазные линии.

#### Описание выводов БИС K1815BF3

| обозначение вывода | номер кон-такта         | назначение вывода   |
|--------------------|-------------------------|---|
| A(1,0)             | 7;6                     | Парафазная входная шина операнда A                        |
| B(1,0)             | 5;4                     | То же B   |
| C(1,0)             | 11;10                   | -"- C   |
| D(1,0)             | 8;9                     | -"- D   |
| G(1,0)             | 3;2                     | -"- G   |
| A*(1,0)            | 23;22                   | Парафазная выходная шина результата A*                    |
| B*(1,0)            | 15;14                   | Парафазная выходная шина результата B*                    |
| CLK                | 13                      | Вход синхронизации приема данных и исполнение микрокоманд |
| STB                | 1                       | Вход строба записи микрокоманд в MIR                      |
| MI(0-5)            | 20;21<br>17;18<br>16;19 | 6-разрядная параллельная шина микрокоманд                 |
| U <sub>cc</sub>    | 24                      | Напряжение питания (+1,2+0,2 В)                           |

Кодировка данных на всех информационных входах и выходах БИС производится в инверсном коде: единица - низкий потенциал, ноль - высокий.

Информация в знакоразрядной системе записывается с использованием отрицательных и положительных двоичных цифр  $+1=1$ ,  $-1=1\bar{}$ .

Перевод чисел из обычной двоичной системы счисления в знакоразрядную осуществляется по правилу:

при  $X > 0$   $X_{zr} = X_2$  (например,  $0,101=0,101$ );

при  $X < 0$   $X_{zr} = X_2^-$ , где  $X_2^- = X_2$  с переносом знака на все единицы (например:  $-0,0101 = 0,01\bar{0}1$ ).

При передаче информации по парафазным шинам положительная часть числа ( $X_{zr}$ ) передается по положительным шинам ( $A1, B1, \dots, G1$ ), отрицательная ( $X_{zr}^-$ ) - по отрицательным ( $A0, B0, \dots, G0$ ). Запятая, используемая в качестве маркера для запуска множительного устройства, передается по обеим шинам сразу ( $X_{zr}, X_{zr}^-$ ).

Обратный перевод осуществляется сложением положительной и отрицательной частей знакоразрядного числа. Например, для числа  $X_{zr} = 0,01\bar{1}1\bar{1}$  в обычной записи  $X = X_{zr} + X_{zr}^- = 0,00101 + (-0,01010) = -0,00101$ .

Ниже приведен пример, иллюстрирующий код знакоразрядного числа с запятой (маркером), передаваемый по парафазным шинам:

|         |   |   |   |   |   |   |   |   |   |
|---------|---|---|---|---|---|---|---|---|---|
| Такты   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |   |
| Число X | , | 1 | 1 | 1 | 0 | 1 | 1 | 1 |   |
| Шина X1 |   | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| Шина X0 |   | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

Сумматор SM1 выполняет операцию  $A+B$  или пропускает на вход LU операнда  $A$  и  $B$ . Узел LU позволяет получить из суммы либо модуль ( $|A+B|$ ), либо выделить максимальное из чисел ( $\max\{A, B\}$ ). Управляемый блок задержки (или устройство масштабирования) пропускает результат операции на выход транзитом или с задержкой в 6 или 12 тактов. Сам сумматор вносит задержку в 6 тактов, таким образом, результат операции появляется на входе  $A^*$  с задержкой в 6, 12 или 18 тактов.

Сумматор SM2 выполняет операцию  $C+D$ , результат которой умножается на операнд  $G$  в блоке MUL. Информация на входе  $B^*$  всегда задерживается относительно входа на 12 тактов. Для устранения возможных переполнений результат может быть сдвинут в сторону младших разрядов на 1, 2 или 6 бит. Операция масштабирования выполняется под управлением микрокоманды в SM1 и SM2.

Микропроцессор выполнен по ИИЛ-технологии. Входной ток 1 мА, нагрузочная способность выхода  $I_o$  15 мА при  $U_o$  0,4 В. При использовании коллекторных резисторов ( $R_k$ ), находящихся снаружи БИС, возможно сопряжение микропроцессора со стандартными ТТЛ-ИЛ и КМОП-схемами.

Управление работой БИС осуществляется 6-разрядным кодом, заносимым в MIR отрицательным фронтом сигнала STB. При записи микрокоманд использованы следующие обозначения: 6tc, 12tc - задержка выдачи результата на 6, 12 и 18 периодов тактовой частоты CLK; /2, /4 и /8- деление при масштабировании, выполняемое везде умножением результата на

2 путем сдвига запятой на 1,2 и 3 бита в сторону старших разрядов; RESET - операция сброса, блокирующая выходы микропроцессора за 18 ts и приводящая его в начальное состояние путем сброса в нуль распределителя умножителя.

Исполнение операций микропроцессором может быть проиллюстрировано на микрокомандах (A+B), (C+D)G при различных задержках (6,12 и 18 тактов, а также с использованием и без использования масштабирования). Для чисел A=0,11; B=0,11; C=0, 1; D=0,1; G=0,1 при наличии масштабирования имеем:

=0,1 с задержкой в 6,12 или 18 tc

=0,1 с задержкой в 12 tc

при отсутствии масштабирования имеем:

=0,01 с задержкой в  $6-1 = 5$  tc;

=0,001 с задержкой в  $2(6-1) = 10$  tc;

=0,0001 с задержкой в  $3(6-1) = 15$  tc;

=0,001 с задержкой в  $2(6-1) = 10$  tc.

=0,0001 с задержкой в  $3(6-1) = 15$  tc;

Основные электрические параметры БИС К1815ВФ3:

Напряжение питания инжектора  $U_{gc}, В \dots \dots \dots +1,2 \pm 0,2$

Номинальный ток питания  $I_{gc}, мА \dots \dots \dots 300$

Напряжение источника питания  $U_{cc}, В \dots \dots \dots 3-5$

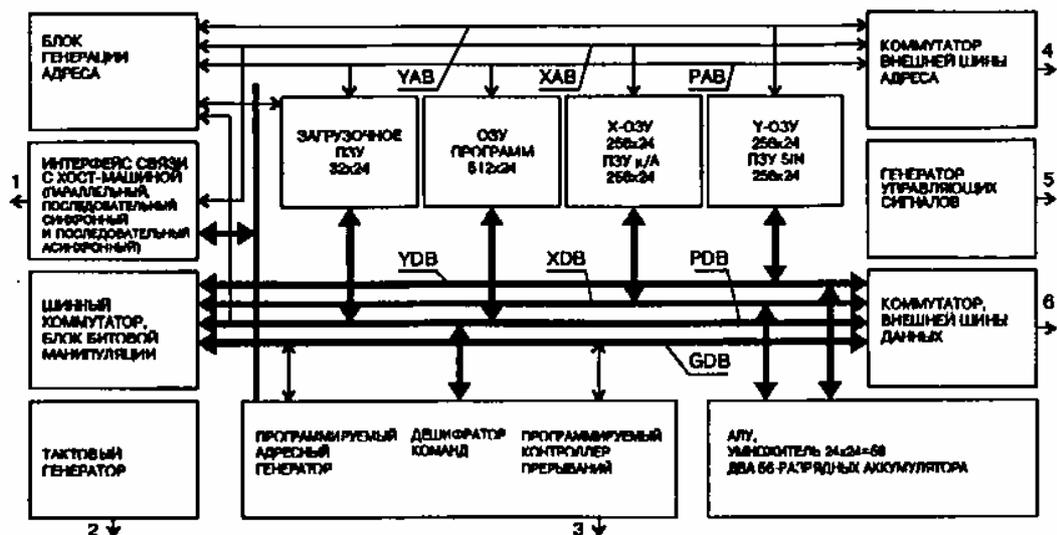
Расчет резистора в цепи питания БИС производится по формуле:  $R_g = (U_{cc} - U_{gc})/I_{gc}$  например, для  $U_{cc} = 5 В$   $R_g$  составляет около 12,70 м.). Расчет резисторов в цепи коллекторов производится исходя из схем подключения выводов. Номиналы коллекторных резисторов обычно выбираются от 300 до 1000 Ом.

### 5.3 DSP

Отличительной особенностью ЦОС является очень большой объем вычислений (обычно типа суммирования парных произведений комплексных чисел), осуществляемый в реальном или близком к реальному масштабе времени. В основу DSP обычно положены следующие принципы:

- использование гарвардской архитектуры;
- сокращение длительности командного цикла;
- применение конвейеризации;
- применение аппаратного умножителя;
- включение в систему команд специальных команд ЦОС.

Типовой архитектурой DSP обладает БИС DS56001 фирмы Motorola:



Этот DSP способен выполнять 10,25 млн. операций в секунду. Разрядность шины данных составляет 24 бита, что обеспечивает обработку сигнала с динамическим диапазоном 144 дБ, а два 56-разрядных аккумулятора обеспечивают точные промежуточные вычисления над сигналами, изменяющимися в диапазоне 336 дБ. Используется модифицированная гарвардская архитектура, включающая два ОЗУ и две шины данных, что создает большие удобства при действиях над комплексными числами. Большинство инструкций позволяют одновременно с выполнением команды производить две пересылки данных и модификацию адресных указателей.

## **Рекомендуемая литература:**

### **Основная литература**

Калабеков Б.А. Микропроцессоры и их применение в системах передачи и обработки сигналов. М.:Радио и связь, 1988.-368 с.

Микропроцессоры в радиотехнических системах./Под ред. Ю.М.Казаринова. М.:Радио и связь, 1982.-280 с.

Самофалов К.Г. Основы теории многоуровневых конвейерных вычислительных систем. М.:Радио и связь, 1989.-286 с.

Плотников В.Н. и др. Цифровые анализаторы спектра. М.: Радио и связь, 1990.-184 с.

Мячев А.А. и др. Интерфейсы систем обработки данных. М.:Радио и связь, 1995.-272 с.

### **Учебно-методическая литература**

#### *Для лабораторных работ*

Горбунов А.Л. Методические указания к выполнению лабораторных работ по курсу "Микропроцессорные устройства РЭС". М.:РИО МИИГА, 1990.

#### *Для курсовой работы*

Горбунов А.Л. Методические указания к выполнению курсовой работы по дисциплине "Микропроцессорные устройства РЭС". М.:РИО МИИГА, 1992.

#### *Для практических занятий*

Горбунов А.Л. Методические указания к изучению дисциплины "Микропроцессорные устройства РЭС". М.:РИО МИИГА, 1992.

### **Дополнительная литература**

Белоус В.Ф. и др. Микропроцессорный комплект для ЦОС 1815. М.:Радио и связь, 1992.

Фрир Дж. Построение вычислительных систем на базе перспективных микропроцессоров. М.:Мир, 1991. - 423 с.

Ушкар М.Н. Микропроцессорные устройства в РЭА. М.:Радио и связь, 1988. - 157 с.