

## Оглавление

1.	Введение .....	4
1.1.	Цели курса. Основные разделы курса. ....	4
1.2.	Основные определения. Классификация МПК.....	5
1.3.	Классификация ОМК.....	5
1.4.	Основные архитектуры процессоров ОМК.....	6
1.5.	Классификация микропроцессорных систем.....	6
1.6.	Гарвардская и Фон-Неймовская архитектура памяти контроллера (ОМК) .....	8
1.7.	Общая структура микропроцессорного устройства для систем управления .....	8
1.8.	Структура программного обеспечения МПУ .....	10
2.	Проектирование МПУ на основе периферийных ОМК.....	11
2.1.	Основные особенности периферийных ОМК.....	11
2.2.	Семейства однокристальных PIC-контроллеров .....	11
2.3.	PIC-контроллер PIC16C58 .....	12
2.4.	Внутренняя структура контроллера PIC16C58.....	13
2.5.	Схемы синхронизации PIC16C58.....	13
2.6.	Организация памяти программ контроллера PIC16C58 (ПЗУ).....	14
2.7.	Организация памяти данных (внутреннее ОЗУ контроллера PIC16C58) .....	15
2.8.	Модуль таймера/счетчика контроллера PIC16C58.....	17
2.9.	Сторожевой таймер WDT контроллера PIC16C58 .....	18
2.10.	Система команд контроллера PIC16C58 .....	19
2.10.1.	Команды работы с байтами .....	19
2.10.2.	Команды работы с битами PIC16C58 .....	22
2.10.3.	Команды передачи управления и работы с константами .....	23
2.10.4.	Команды управления режимами работы контроллера PIC16C58 .....	24
2.11.	Влияние команд контроллера на биты-признаки результата (в регистре STATUS) .....	24
2.12.	Разработка простейшего устройства на основе PIC-контроллера. ....	24
3.	Проектирование МПУ на основе универсальных восьмиразрядных ОМК семейства MSC-51 .....	25
3.1.	Основные технические характеристики и структура ОМК K1816BE51 .....	25
3.2.	Назначение основных выводов БИС ОМК K1816BE51 .....	26
3.3.	Организация памяти данных контроллера BE51 .....	28
3.4.	Система команд ОМК K1816BE51 .....	30
3.4.1.	Команды пересылки данных .....	30
3.4.2.	Команды арифметических операций.....	32
3.4.3.	Команды работы с битами.....	33
3.4.4.	Команды передачи управления или переходов.....	34
3.5.	Встроенные таймеры счетчики контроллера K1816BE51 .....	35

3.6.	Встроенный последовательный интерфейс ОМК .....	37
3.7.	Задание скорости передачи по последовательному интерфейсу ...	39
3.8.	Система прерываний ОМК BE51 .....	40
3.9.	Механизм обслуживания прерываний ОМК.....	42
3.10.	Построение расширенных МП систем на основе ОМК BE 51 .....	42
3.11.	Дальнейшее расширение и развитие семейства ОМК MCS-51 .....	44
4.	Принципы организации систем дискретного ввода-вывода в МПС .....	45
4.1.	Общая структура систем дискретного ввода-вывода .....	45
4.2.	Реализация селекторов адреса .....	46
4.3.	Реализация портов ввода-вывода .....	47
4.4.	Программно-аппаратное обеспечение ввода дискретных сигналов. ....	49
4.4.1.	Согласование логических уровней сигналов.....	49
4.4.2.	Схемы устранения дребезга контактных датчиков.....	51
4.4.3.	Программные способы устранения влияния дребезга контактов .....	52
4.4.4.	Аппаратное обеспечение вывода логических сигналов .....	53
4.4.5.	Особенности ввода коротких логических сигналов .....	55
4.5.	Обеспечение взаимодействия микропроцессорных устройств и ЭВМ верхнего уровня.....	56
4.6.	Средства взаимодействия МПУ с оператором .....	58
4.7.	Жидкокристаллические индикаторные панели .....	60
4.8.	Совмещенные контроллеры клавиатуры и индикатора.....	61
5.	Организация подсистем ввода/вывода аналоговых сигналов МПУ.....	64
5.1.	Вывод аналоговых сигналов .....	64
5.2.	Общая структура подсистемы ввода аналоговых сигналов .....	66
5.3.	Основные типы АЦП, используемые в МПУ. Основные характеристики выбора АЦП .....	67
5.4.	Программно-аппаратная реализация время-импульсного АЦП с однократным интегрированием.....	68
5.5.	Время-импульсный АЦП двойного интегрирования .....	69
5.6.	Программно-аппаратная реализация АЦП последовательного счета и последовательных приближений .....	72
5.7.	Понятия о АЦП с дельта-сигма-модуляцией.....	73
5.8.	Устройство выборки хранения (УВХ).....	73
6.	Общие принципы использования 16-разрядных ОМК.....	76
6.1.	Общая характеристика 16-разрядных ОМК.....	76
6.2.	Основные технические характеристики ОМК SAB80C167 .....	77
6.3.	Внутренняя структура ОМК SAB80C167 .....	80
6.4.	Организация памяти ОМК C167 .....	80
6.5.	Системный стек ОМК C167.....	81
6.6.	Способы адресации данных в ОМК C167 .....	82
6.7.	Общая структура процессорного модуля C167 (CPU C167).....	83
6.8.	Особенности системы прерываний ОМК SAB80C167 .....	85
6.9.	Периферийный контроллер событий (PEC).....	86

6.10.	Модули захват-сравнение ОМК С167 .....	87
6.11.	Встроенный начальный загрузчик ПО или ОС ОМК С167.....	88
6.12.	Встроенный модуль ШИМ ОМК С167.....	88
6.13.	Встроенный контроллер промышленной CAN-сети (CAN-модуль) .....	89

## **Тема 1.- 6 часов (у.з.-1) Введение.**

Основные понятия и определения микропроцессорной техники. Понятие об архитектуре микропроцессорной системы. Классификация однокристальных микроконтроллеров (ОМК) и микропроцессоров. RISC-архитектура ОМК. Основные типы МПСУ. Принципы построения и структура технических средств микропроцессорного контроллера (МПК). Основные этапы разработки микропроцессорной системы. Цифровые сигнальные процессоры.

---

# **1. Введение**

## **1.1. Цели курса. Основные разделы курса.**

Обучение разработки аппаратных средств микропроцессорных устройств СУ на основе однокристальных микроконтроллеров (ОМК). Обучение навыкам программирования ОМК на языках низкого уровня.

### **Основные разделы курса:**

1. Основные определения. Классификация микропроцессорных контроллеров (МПК).
2. Разработка микропроцессорных устройств на основе периферийных ОМК (PIC).
3. Разработка микропроцессорных устройств на основе универсальных 8-ми разрядных ОМК (MCS-51, K1816BE51).
4. Особенности использования 16-разрядных ОМК.
5. Особенности построения подсистем ввода/вывода аналоговых и дискретных сигналов.
6. Построение подсистем взаимодействия с оператором и управляющими ЭВМ высших уровней.

### **Литература.**

1. Сташин В.В. , Урусов А.В. и др. Проектирование цифровых устройств на ОМК. Москва, Энергоиздат, 1990г. 300 с.
2. Однокристальные микро-ЭВМ. Справочник. Под редакцией Боборыкина А.В., Москва, Бином ,1994.
3. Щелкунов Н.Н., Дианов А.П. Микропроцессорные средства и системы. Москва, Радиосвязь,1989.
4. Предько М. Руководство по микроконтроллерам. Москва, Постмаркет, 2001.
5. ОМК фирмы Microchip PIC16C5X. Под редакцией Владимирова А.М. Рига, Ормикс, 1996.
6. Федоров Б.Э., Телец В.А. Интегральные схемы ЦАП и АЦП. Москва, Энергоатомиздат,1990г.
7. Угрюмов Е.П. Цифровая схемотехника. Санкт-Петербург, БХВ 2000г.

## 1.2. Основные определения. Классификация МПК

Микропроцессор (МП) - функционально законченный процессор ЭВМ реализованный в виде одной или нескольких БИС и предназначен для обработки цифровой информации по заданным программам.

Микропроцессорный контроллер (МПК) – функционально законченная микро-ЭВМ, предназначенная для целей контроля и управления.

МПК может реализовываться на следующей элементной базе:

- однокристальных микропроцессорах (ОМП);
- секционных (многокристалльных) МП;
- однокристальных микроконтроллерах (ОМК);
- сложных матричных программируемых логических схемах (ПЛИС, PLD, CPLD и др.).

ОМК – функционально законченный МПК, реализованный в виде одной СБИС (сверх-БИС).ОМК включает в состав: процессор, ОЗУ, ПЗУ, порты ввода/вывода для подключения внешних устройств, модули ввода аналогового сигнала АЦП, таймеры, контроллеры прерывания, контроллеры различных интерфейсов и т.д.

Простейший ОМК представляет собой БИС площадью не более  $1\text{ см}^2$  и всего с восемью выводами.

## 1.3. Классификация ОМК

Различают:

- 1) Периферийные (интерфейсные) ОМК предназначен для реализации простейших МП систем управления. Имеют малую производительность и малые габаритные размеры. В частности может использоваться периферийными устройствами ЭВМ (клавиатура, мышь и т.п.)  
К ним относятся: PIC – Micro Chip, VPS – 42 (Intel).
- 2) Универсальные 8–разрядные ОМК предназначены для реализации МП систем малой и средней производительности.  
Имеют простую систему команд и большую номенклатуру встроенных устройств. Основные типы: MSC – 51 (Intel)  
Motorola HC05 – HC012 и др.
- 3) Универсальный 16–разрядный ОМК. Предназначен для реализации систем реального времени средней производительности. Структура и система команд нацелены на скорейшую реакцию на внешние события.  
Наибольшее использование имеют в системах управления электродвигателями (мехатронные системы).
- 4) Специализированные 32–разрядные ОМК реализуют высокопроизводительную ARM архитектуру и предназначены для систем телефонии, передачи информации, телевидения и других , требующие высокоскоростной обработки информации.

К типовым 16–разрядным ОМК относятся: MSC96/196/296 (Intel), C161–C167 (Siemens, Infineon), HC16 Motorola и др.

- 5) Цифровые сигнальные процессоры (DSP – Digital Signal Processor) предназначен для сложной математической обработки измеряемых сигналов в режиме реального времени. Широко используются в телефонии и связи.

Основные отличия DSP: повышенная разрядность обрабатываемых слов (16,32,64 бита) и высокая скорость в формате с плавающей точкой (16 flops). Производители: Texas Instruments (TMS 320 и др.), Analog Device (ADSP 2181 и др.).

## 1.4. Основные архитектуры процессоров ОМК

В современных ОМК применяются следующие архитектуры процессоров :

- RISC – (Reduce Instruction Set Commands ) архитектура с сокращенным набором команд.
- CISC – (Complex Instruction Set Commands) традиционная архитектура с расширенным набором команд.
- ARM – (Advanced RISC - machine) усовершенствованная RISC архитектура.

Главная задача RISC архитектуры обеспечение наивысшей производительности процессора. Её отличительными чертами является:

- малое число команд процессора (несколько десятков);
- каждая команда выполняется за минимальное время (1-2 машинных цикла, такта).
- максимально возможное число регистров общего назначения процессора (несколько тысяч);
- увеличенная разрядность процессора (12,14,16 бит).

Современная RISC архитектура включает, как правило, только последние 3 пункта, т.к. за счет повешенной плотности компоновки БИС стало возможным реализовать большое количество команд.

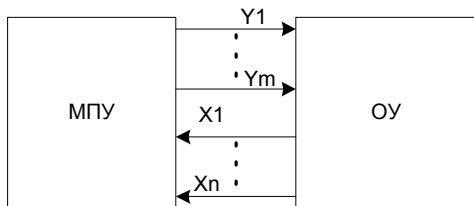
В современных 32–разрядных ОМК используют ARM архитектуру (расширенная RISC архитектура с суперсокращением команд THUMB ).

## 1.5. Классификация микропроцессорных систем

МПС делят на три основных типа:

- программно-логического управления (ПЛУ);
- системы сбора и обработки информации (СОИ);
- системы цифрового автоматического управления (ЦАУ).

*Системы ПЛУ* характеризуются тем, что все измеренные сигналы  $X_i$  и выданные на объект сигналы управления  $Y_i$  имеют логический характер (да/нет, вкл/выкл).



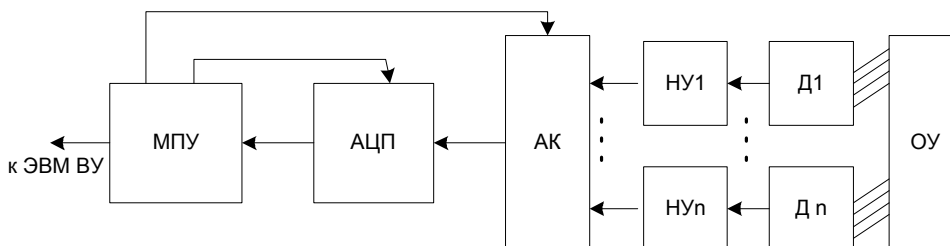
$$Y_i = \{1 \text{ или } 0\}_{i=1,m}; \quad X_i = \{1 \text{ или } 0\}_{i=1,n}$$

Логические сигналы управления  $Y_i$  вычисляются в МПУ программно, как выходные сигналы логических функций или конечных логических аппаратов.

*Системы СОИ* предназначены для выполнения трех основных функций:

- постоянный опрос и измерение сигналов с группы датчиков (датчики давления, температуры, тока и т.д.) находящихся на объекте
- первичная обработка измерений информации (устранение помех, преобразование формата данных и т.д.)
- сохранение блоков измеренной информации в памяти или передача её на ЭВМ верхнего уровня (ЭВМ ВУ)

Общая структура имеет вид:

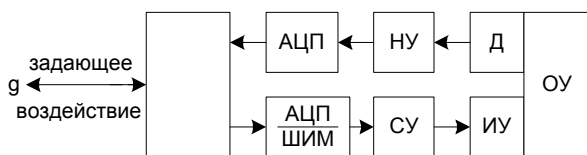


$D_1, \dots, D_n$  – датчики на ОУ.

$HУ_1, \dots, HУ_n$  – нормирующие устройства, преобразующие сигнал с датчиков в требуемый диапазон для измерения АЦП.

АК – аналоговый коммутатор, осуществляет подключение одного из датчиков к АЦП для измерения

*Системы ЦАУ* предназначены для организации с замкнутого контура управления объектом и реализует функции автоматического регулятора, заданного  $Z$ -передаточной функцией или разностным уравнением.



ИУ – исполнительное устройство;

СУ – согласующее устройство, усиливающее сигнал с ЦАП или ШИМ до уровня требуемого ИУ.

ШИМ – широтно-импульсный модулятор, используется для преобразования кода в аналоговый сигнал, путем модуляции ширины импульса.

Важную роль в системах ЦАП играют устройства отчета времени – таймера, они определяют интервал измерения и выдачи управляемых сигналов в системе.

## 1.6. Гарвардская и Фон-Неймовская архитектура памяти контроллера (ОМК)

Главными отличительными чертами Гарвардской архитектуры организации памяти контроллера является:

- реализация в виде различных устройств памяти для программ и памяти для данных.
- использование двух параллельно работающих независимых шин для чтения данных и команд.

*Замечание:* Объем ПД как правило значительно меньше объема ПП.

Основные преимущества Фон-Неймоновской архитектуры:

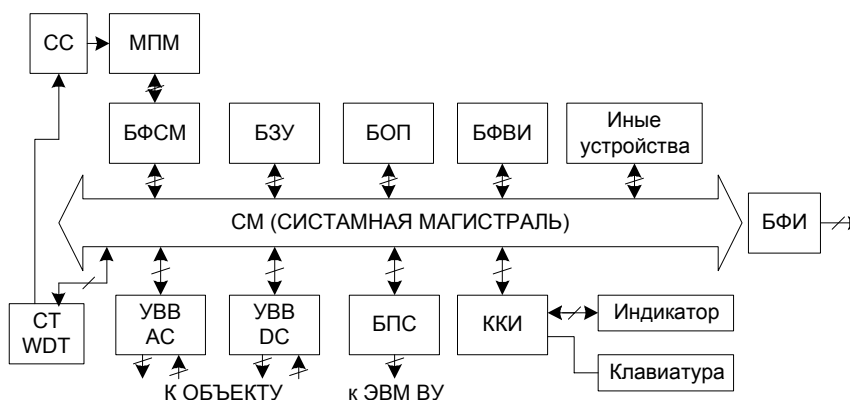
- простота аппаратной реализации
- универсальность исполнения команд

В настоящее время используются обе архитектуры памяти: Гарвардская в несложных восьми - разрядных контроллерах, Фон-Неймоновская в универсальных 16- разрядных и выше.

## 1.7. Общая структура микропроцессорного устройства для систем управления

Она

имеет



следующий вид:

В микропроцессорных устройствах используются магистрально – модульная структура, т.е. к единой системной магистрали (СМ) подключены все основные блоки:

**МПМ**–микропроцессорный модуль реализуется на основе однокристалльного процессора или микроконтроллера.

**СС**–схема синхронизации, обеспечивает генерацию тактовой частоты процессора и сигнала «Сброс».



**БФСМ**–блок формирования системной магистрали, преобразует сигналы управления микропроцессора в сигналы шин системной магистрали.

**БЗУ**–блок запоминающих устройств, включает ОЗУ, ПЗУ, часто энергонезависимую память.

**БОП**–блок обработки прерывания от основных модулей микропроцессорного устройства.

**БФИ**–блок формирования времени интервалов, используется как для подсчета времени, так и для подсчета внешних импульсов; реализуется в виде таймера-счетчика.

**СТ (WDT)**–Watch Dog Timer-сторожевой таймер, предназначен для устранения аварийных закливаний программ микропроцессорного устройства.

**УВВ АС**–устройство ввода/вывода аналоговых сигналов, предназначенных для измерения и генерации напряжений различной амплитуды, как правило, находящейся в диапазоне 0...10 В. При этом как устройство ввода используется АЦП, а как устройство вывода аналогового сигнала ЦАП или ШИМ.

**УВВ ДС**–устройство ввода/вывода дискретных сигналов, предназначенных для измерения и выдачи логических сигналов, как правило, ТТЛ-уровней.

**БПС**–блок последовательной связи, предназначен для приема и передачи информации из МПУ на ЭВМ ВУ, или на другое устройство. Представляет собой либо последовательный интерфейс, либо промышленную сеть.

**ККИ**–контроллер клавиатуры и индикаторов, предназначен для подключения к устройству клавиатуры и линейки семи сегментных или жидкокристаллических индикаторов.

**БФИ**–блок формирования интерфейса, предназначен для усиления сигналов СМ и выдачи их на разъем.

*Определение.* МПУ называется с *открытой архитектурой*, если сигналы СМ выдаются на разъем и могут быть использованы для подключения внешних устройств. Иначе-с *закрытой архитектурой*.

**Иные периферийные устройства**–в качестве них может быть использованы контроллеры различных последовательных и параллельных интерфейсов(для подключения приборов, внешней дисковой памяти, специальных устройств управления и т.д.).

Часто в промышленных МПУ используется накопители на «твердых» дисках. По сути дела это Flash-память, но с файловой организацией, как у диска.

## **Тема 2. - 2 часа (у.з.-2). Системное программное обеспечение (ПО) микропроцессорных систем.**

Состав и структура ПО микропроцессорной системы. Общая структура и основные функции управляющей программы «монитор». Общая характеристика

тестовых процедур основных узлов микропроцессорного контроллера. Библиотеки стандартных прикладных программ.

---

## 1.8. Структура программного обеспечения МПУ

Программное обеспечение можно разбить на три основные части:

- резидентное системное ПО
- резидентное прикладное ПО
- инструментальный кросс системы разработки ПО.

Резидентное системное ПО включает:

- операционные системы реального времени ОСРВ для МПУ (RTX, X11, QHS, LINUX,...). Основная функция ОСРВ - минимизация времени ответа прикладного ПО на внешние запросы, а также упрощение взаимодействия прикладных программ с основными узлами МПУ. В простейшем случае ОС реального времени может заменяться программой–монитором;
- тестовые программные процедуры;
- библиотеки прикладных подпрограмм, предназначены для упрощения написания прикладного программного обеспечения.

*Программа-монитор* предназначена для упрощения взаимодействия с пользователем через клавиатуру или последовательный интерфейс. К основным командам монитора относятся:

- просмотр и модификация ячеек памяти устройства (S-команды);
- запуск программы с введенного адреса (G-команды);
- просмотр и модификация портов ввода/вывода контроллера (I/O-команды);
- пошаговое выполнение программы контроллера (J-команды) и т.д.

*Тестовые процедуры* предназначены для поиска неисправностей и проверке работоспособности основных узлов МПУ.

*Стандартные библиотеки прикладных программ* могут иметь различный состав в зависимости от области применения МПУ.

*Инструментальные кросс системы разработки ПО* представляют собой совокупность программ для инструментальной ЭВМ.

*Замечание.* Основным отличием ОС реального времени от универсальных ОС является то, что их главная цель-минимизация задержки для ответа на внешний запрос. Основная цель универсальных ОС - оптимальное распределение ресурсов ЭВМ при выполнении программ.

**Тема 3.- 10 часов (у.з.-3) Разработка микропроцессорных систем на базе однокристальных микроконтроллеров с RISC-архитектурой .**

Особенности разработки микропроцессорных систем на базе контроллеров с RISC-архитектурой. Программируемые периферийные контроллеры. Основные технические характеристики контроллера серии PIC16C5X. Общая архитектура и обозначение выводов. Организация памяти контроллера семейства PIC16C5X. Регистры ввода-вывода, внутренний таймер и схемы синхронизации. Система команд контроллера семейства PIC16C5X. Особенности контроллеров с AVR-архитектурой. Средства разработки PIC-контроллеров.

---

## **2. Проектирование МПУ на основе периферийных ОМК**

### **2.1. Основные особенности периферийных ОМК**

Предназначены для реализации простейших алгоритмов управления объектами или реализации вспомогательных функций в сложных схемах управления. Отличительными особенностями их является:

- реализуются на основе Гарвардской архитектуры.
- система команд не предназначена для реализации сложных математических вычислений (отсутствует умножение и деление).
- практически исключают аппаратное расширение, т.к. не поддерживают реализацию внешней СМ.
- имеют малые габаритные размеры.
- используется восьмиразрядный процессор обработки данных.

В настоящее время используются следующие типы периферийных контроллеров:

- PIC(Peripheral Interface Controller) фирмы MicroChip,
- I8042(Intel), UPI-42(NEC), K1835BE49 (РОССИЯ),
- K1878BE1(Тесей) на основе RISC-архитектуры и т.д.

### **2.2. Семейства однокристальных PIC-контроллеров**

Выделяют четыре основных семейства восьмиразрядных PIC-контроллеров с RISC-архитектурой:

- 1) PIC16C5\*-основное или базовое семейство контроллеров с двенадцатиразрядными командами. Это имеет простейшую архитектуру.

- 2) PIC16C6\*/7\*/8\*-усовершенствованное семейство средней производительности с четырнадцатиразрядными командами. Данное семейство характеризуется большой номенклатурой встроенных устройств в контроллер(АЦП,ЦАП,ШИМ).
- 3) PIC17C\*\*-высокопроизводительное семейство с шестнадцатиразрядными командами. По своим возможностям приближающиеся к восьмиразрядным универсальным контроллерам.
- 4) PIC18C\*\*-модифицированное высокопроизводительное семейство контроллеров с четырнадцать и шестнадцатиразрядными командами. По сравнению с предыдущими семействами имеет встроенную Flash-память (перепрограммируемое ПЗУ);систему команд адаптированную под компилятор языка C (Си) и линейно адресное пространство (в отличие от страничного, используемого ранее).

*Замечание.* Все перечисленные PIC-контроллеры обладают полной командной совместимостью на уровне ассемблерных мнемоник.

\* –любая цифра

### 2.3. PIC-контроллер PIC16C58

Данный контроллер относится к базовому семейству контроллеров с двенадцатиразрядными командами и построен на основе RISC-архитектуры. Основные технические характеристики:

- 1) Тактовая частота контроллера-20 МГц (быстродействие при этом –5 MIPS). Контроллер является полностью ?????? устройством, т.к. он может работать при любой тактовой частоте от 0 до 20 МГц.
- 2) Объем внутреннего ПЗУ (памяти программ) 2К-двенадцатиразрядных слов (ПЗУ с ультрафиолетовым стиранием, либо однокристалльным программируемым ПЗУ).
- 3) Объем внутреннего ОЗУ (памяти данных) 73-восмиразрядных ячеек или регистра общего назначения.
- 4) Система команд включает 33 однословных команды (т.к. RISC-архитектура). Каждая команда выполняется за один машинный цикл (200нс), исключение составляет лишь одна команда перехода, выполняемая за два машинных цикла.
- 5) Имеется встроенный восьмиразрядный таймер-счетчик (tmv0) с восьмиразрядными предварительным делителем частоты.
- 6) Имеется двухуровневый аппаратный стек. Т.о. вложенность подпрограмм не может превышать 2.

- 7) Контроллер обладает низким электропотреблением: при напряжении питания 2,4~6 В., потребляемый ток 2мА в нормальном режиме и 2мкА в режиме пониженного электропотребления.
- 8) реализуется в виде ИС с восемнадцатью выводами.
- 9) Имеется 12 линий ввода/вывода для подключения к объекту управления, организованных в два порта ввода/вывода:
  - порт А – четырехразрядный;
  - порт В – восьмиразрядный.

## 2.4. Внутренняя структура контроллера PIC16C58

Внутренняя ОЗУ (память данных) контроллера состоит из 73 восьмиразрядных РОН (регистров общего назначения) и 6 регистров специального назначения.

К специальным регистрам относят:

- TMR0-регистр данных таймера счетчика контроллера.
- STATUS-регистр признаков выполненной операции (слова состояния процессора).
- FSR-регистр адреса для косвенной адресации данных.
- PCL-младший байт счетчика PC контроллера.
- ПОРТ А и ПОРТ В- регистр данных портов ввода/вывода.

Все перечисленные специальные регистры относятся к устройствам непосредственно не связанным с ОЗУ. Однако, все эти регистры отображены в адресном пространстве ОЗУ, т.е. для команд они рассматриваются как ячейки ОЗУ с соответствующими адресами.

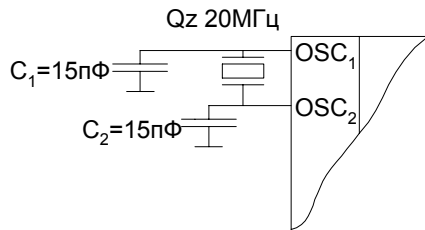
АЛУ контроллера позволяет обрабатывать восьмиразрядные данные и выполнять при этом следующие операции: сложение, вычитание, сдвиг, логические операции и битовые операции.

Если в команде контроллера используется 2 операнда, то один из них обязательно размещается в рабочем регистре (аккумуляторе W). Это основной регистр контроллера, и большинство команд оперирует с ним.

При выполнении команды, признаки полученного результата АЛУ обязательно запишется в регистр STATUS.

## 2.5. Схемы синхронизации PIC16C58.

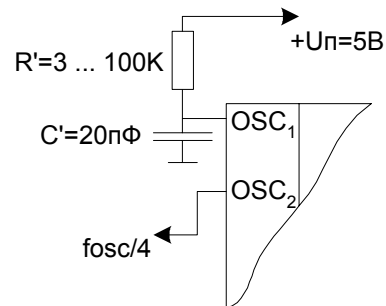
Для генерации тактовой частоты к выводам OSC<sub>1</sub> и OSC<sub>2</sub> контроллера подключают кварцевый резонатор по схеме:



Емкости  $C_1$  и  $C_2$ -запускающие.  
Вместо кварцевого резонатора ( $Qz$ ) возможно использование внешних RC- цепочек:

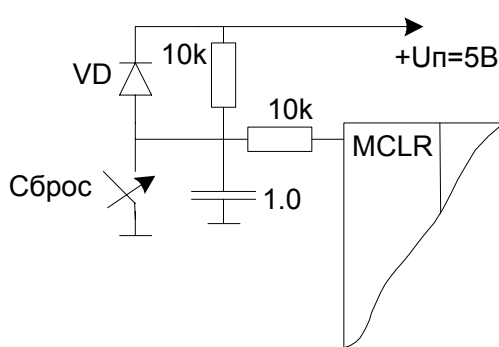
$$R' = 3 \text{ кОм}, f_{osc} = 5 \text{ МГц}$$

$$R' = 100 \text{ кОм}, f_{osc} = 100 \text{ кГц}$$



При использовании RC-цепочек тактовая частота  $f_{osc}$  нестабильна и зависит от изменения напряжения питания  $U_p$ , температуры окружающей среды и старения RC-элементов.

Для начального «Сброса» контроллера используют следующую схему:



Диод  $VD$  для быстрого разряда емкости при включении питания  $U_p$ .

Данная схема формирует сигнал «Сброс» (MCLR) требуемой длительности (10...100мс).

Тактовая частота  $f_{osc}$  внутри контроллера делится на 4, т.о. формируется частота машинных циклов:  $f_{мц} = f_{osc}/4$ .

За один машинный цикл (МЦ) в

контроллере параллельно выполняется две операции:

- выполнение текущей команды.
- Выборка следующей команды из памяти.

## 2.6. Организация памяти программ контроллера PIC16C58 (ПЗУ)

Память программ контроллера предназначена для хранения выполняемых контроллером программ и имеет объем 2к двенадцатиразрядных слов. Весь объем памяти программ разбит на четыре страницы, причем объем каждой страницы- 512 слов.

Команды всегда выполняется из текущей страницы памяти программ. Номер текущей страницы определяется шестым и пятым битом регистра «STATUS».

Эти биты обозначаются PA1 и PA0.

Общая структура памяти программ имеет вид:

Ячейка с адресом 7FFH является стартом, т.к. на нее осуществляется переход при сбросе, при включении питания. В ней располагают команду безусловного перехода GOTO на нужную программу.

## 2.7. Организация памяти данных (внутреннее ОЗУ контроллера PIC16C58)

Внутреннее ОЗУ состоит из :

- 73 регистров 0H (ячеек), организованных в виде четырех банков регистров;
- банка специальных регистров, включающих семь регистров управления.

Общая структура памяти данных имеет вид:

Текущий, используемый в данный момент банк PОН определяется шестым и пятым битом регистра FSR.

Память данных построена т.о., что при выборе любого из банков остается доступным банк специальных регистров. При этом он заменяет отсутствующие ячейки с адресами (20H...2FH, 40...4FH, 60...6FH).

Рассмотрим подробнее управляющие регистры из банка специальных регистров:

**INDF\*** (адрес в ОЗУ 0H) – это регистр данных при косвенной адресации данных.

Физически данный регистр не существует, он используется только совместно с регистром FSR. При обращении к регистру INDF фактически происходит обращение к ячейке ОЗУ, адрес которой находится в регистре FSR.

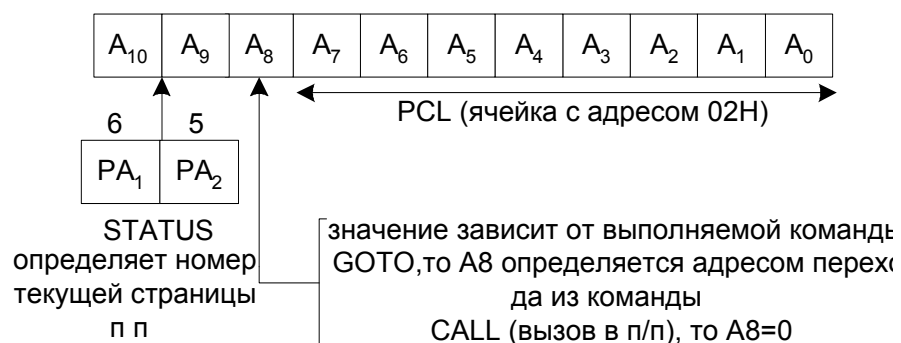
ADDNF INDF,0 ; W ← W+(ячейка с адресом из FSR)

*Пример.*

**TMR0** (адрес в ОЗУ 1H) - регистр данных восьмиразрядного таймера-счетчика контроллера .

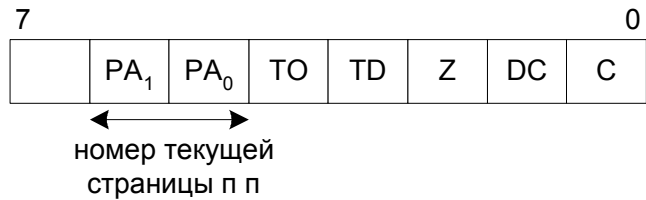
**PCL** (адрес в ОЗУ 2H) - младший байт счетчика команд контроллера PC . Счетчик команд PC одиннадцатиразрядный и используется для адресации команд в памяти программ контроллера. Он позволяет адресовать любую из  $2048 = 2^{11}$  команд, находящихся в памяти программ.

Формат счетчика команд PC:



PCL может изменяться любой командой перехода, либо командой изменения содержимого ячейки памяти .

**STATUS** (адрес в ОЗУ 03Н) - регистр признаков результатов (регистр состояния контроллера) имеет формат:



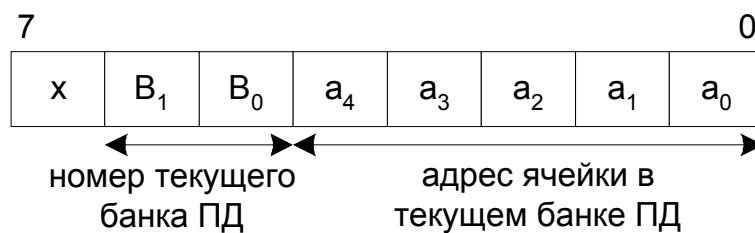
**Z** - признак нулевого результата.

**C** - бит-признак переноса (заем) из старшего разряда.

**DC** - бит-признак дополнительного переноса из одной тетрады байта в другую.

**TO** и **TD** - биты, определяющие источник сигнала «Сброс контроллера» (сторожевой таймер WDT, внешний сигнал сброса), а также режим контроллера, в котором этот сигнал поступил (рабочий режим, режим пониженного потребления) .

**FSR** (адрес в ОЗУ 4Н) - регистр адреса при косвенной адресации операндов (см. INDF\* ). Формат регистра FSR:



*Замечание.* Если разряд a<sub>4</sub> в регистре FSR a<sub>4</sub> = 0, то номер текущего банка B<sub>1</sub>, B<sub>0</sub> не анализируется и всегда выбирается банк специальных регистров.

**Port A, Port B** (адрес в ОЗУ 05Н и06Н) - регистры данных портов ввода/вывода контроллера.

Разрядность регистра PortA- 4 бита (отсутствует 4 старших бита), разрядность PortB - 8 бит.

Для управления портами используются специальные регистры управления: TRIS<sub>A</sub> и TRIS<sub>B</sub> (Transiver / Resiver Instruction Set).

Разрядность регистров TRIS<sub>A</sub> и TRIS<sub>B</sub> совпадает с разрядностью соответствующих портов (Port A и Port B).

Установка бита в регистр TRIS управляет соответствующим битом PortA и Port B.

Установка разряда регистра TRIS в единицу программирует соответствующий разряд порта на ввод информации. Установка в ноль программирует разряд порта на вывод.

Для записи информации в регистр TRIS используются специальные команды:

TRIS 5 ; W → TRIS<sub>A</sub>



TRIS 6 ; W → TRIS<sub>B</sub>

Чтение или запись информации в порт А или В осуществляется теми же командами, что и для записи/чтения ячеек ОЗУ:

MOV WF Port A, W ; Port A → N

MOV F Port A, W ; W → Port A.

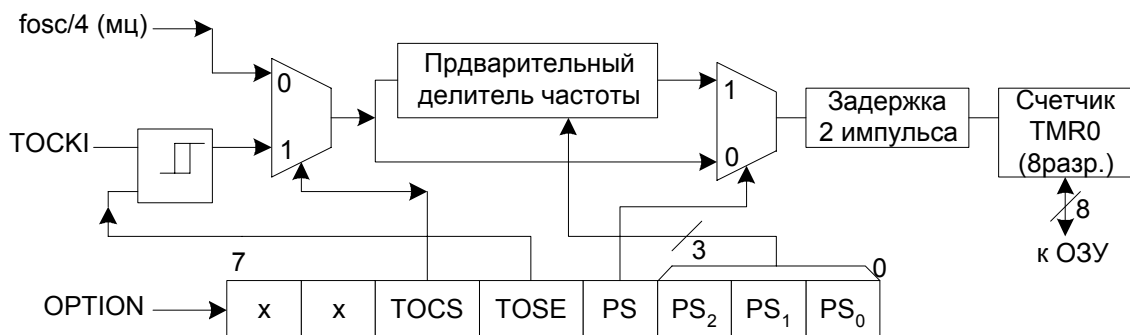
## 2.8. Модуль таймера/счетчика контроллера PIC16C58

Данный модуль предназначен для :

- подсчета числа внешних импульсов, подаваемых на вход контроллера ТОСКІ (режим счетчика);
- подсчета числа машинных циклов контроллера ( $f_{osc}/4$ ) (режим таймера).

*Замечание.* Т.к. частота машинных циклов при использовании кварцевого резонатора высокостабильна, то второй режим используют для подсчета времени.

Структурная схема таймера/счетчика имеет вид:



Основными блоками таймера/счетчика является:

- восьмиразрядный двоичный суммирующий счетчик TMR0;
- восьмиразрядный *предварительный делитель частоты*, который также представляет восьмиразрядный счетчик с программируемым коэффициентом деления частоты .

Основной отличительной чертой счетчика TMR0 от восьмиразрядного предделителя частоты является то, что информацию о числе бит подсчитанных предделителем частоты считать невозможно.

Предделитель работает как программируемый делитель частоты.

При записи информации в TMR0 автоматически обнуляется предделитель частоты.

Поясним назначение бит регистра OPTION, управляющего работой таймера/счетчика и изображенного на рисунке выше:

**TOCS** - бит, определяющий режим таймера (TOCS=0) или счетчика (TOCS=1);

**TOSE** - определяет управляющий фронт по которому осуществляется модификация таймера- счетчика:



**PS** -бит, определяющий назначение предва-рительного делителя частоты ( если PS=0, то предделитель используется со сторожевым таймером WDT, но не с таймером TMR0, с TMR0 когда PS=1).

PS <sub>2</sub>	PS <sub>1</sub>	PS <sub>0</sub>	TMR0	WDT
0	0	0	1:2	1:1
0	0	1	1:4	1:2
0	1	0	1:8	1:4
0	1	1	1:16	1:8
1	0	0	1:32	1:16
1	0	1	1:64	1:32
1	1	0	1:128	1:64
1	1	1	1:256	1:128

PS<sub>2</sub> – PS<sub>0</sub> - биты, определяющие коэффициент деления частоты предделителя в соответствии с таблицей.

Для записи информации в регистр OPTION используется специальная команда:

OPTION; W→OPTION

## 2.9. Сторожевой таймер WDT контроллера PIC16C58

Сторожевой таймер предназначен для устранения аварийных зацикливаний программ контроллера (зависания).

Принцип его работы состоит в следующем, через строго определенный интервал времени ( $\approx 18$ мс) сторожевой таймер генерирует сигналы «сброс контроллера». Нормально функционирующее ПО не должно допускать сброса контроллера, осуществляя постоянное обнуление WDT и тем самым, заставляя его начать отсчет времени заново.

Обнуление WDT осуществляется специальными командами:

CLEEP;

или

CLRWDT.

Если же программа зациклилась, то обнуление WDT не произойдет вовремя, а следовательно произойдет сброс контроллера и перезапуск ПО.

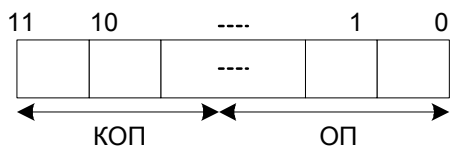
Общая структура WDT имеет вид:

Непосредственно таймер WDT представляет собой RC-генератор встроенный в контроллер и работающий независимо от основной тактовой частоты.

В случае использования предделителя частоты с WDT интервал генерации сигнала «сброс» увеличивается на коэффициент деления предделителя (максимальный интервал может быть  $128 \cdot 8\text{мс} \approx 2,3\text{с}$ ).

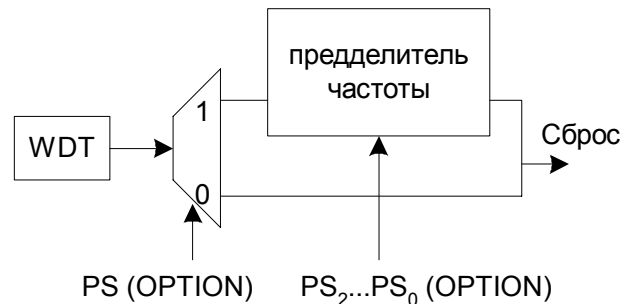
## 2.10. Система команд контроллера PIC16C58

Состоит из 33 двенадцатиразрядных команд. При этом каждая команда имеет поля бит, определяющие код операции (КОП) и используемые командой операнды (ОП):



В качестве операнда в команде контроллера может использоваться:

- любая ячейка внутреннего ОЗУ контроллера (при этом в команде указывается ее адрес);



- непосредственно число (данное), участвующее в операции;
- любой из битов одной из ячеек памяти ОЗУ (при этом указывается адрес ячейки и номер бита).

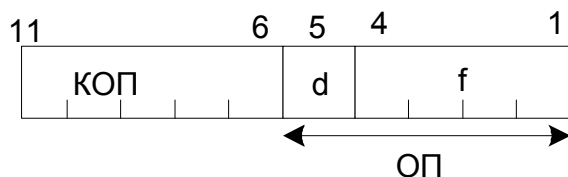
Выделяют три основных группы команд контроллера:

- команды работы с байтами;
- команды работы и битами;
- команды передачи управления и работы с константами.

Группировка команд осуществляется по общности битовых форматов команд.

### 2.10.1. Команды работы с байтами

Битовый формат этих команд имеет вид:



d-бит, определяющий приемник результата операции.

$$d = \begin{cases} 0, & \text{приемник результата W} \\ 1, & f \end{cases}$$

f - поле, определяющие адреса используемого регистра из внутреннего ОЗУ контроллера.

Все команды работы с байтами делят на подгруппы:

- 1) Команды арифметических операций.
- 2) Команды логических операций.
- 3) Команды пересылки данных.

### 1) Команды арифметических операций.

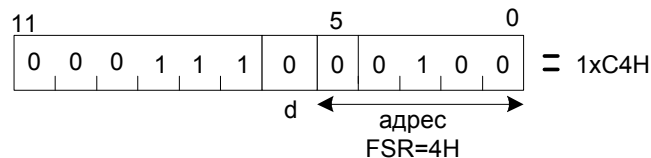
$$\text{a) ADDWF } f, d; f+W \rightarrow \begin{cases} W, d=0 \\ f, d=1 \end{cases}$$

Сложить содержимое регистра F с содержимым рабочего регистра W, результат в W, если d=0 и в f, если d=1.

КОП: 000 111

*Пример.* ADDWF FSR, 0; FSR+W → W.

Для данной команды составим КОП:



$$\text{б) SUBWF } f, d; f-W = \begin{cases} W, d=0 \\ f, d=1 \end{cases} \text{ команда вычитания.}$$

$$\text{в) DECF } f, d; f-1 = \begin{cases} f, d=1 \\ W, d=0 \end{cases} \text{ команда декремента (уменьшения на 1).}$$

$$\text{г) DECFSZ } f, d; f-1 = \begin{cases} f, d=1 \\ W, d=0 \end{cases} \text{ команда спецдекремента.}$$

Уменьшение f на единицу и с попуском следующей команды, если получен нулевой результат.

*Замечание.* Реально процессор при пропуске, вместо выполнения следующей команды, выполнит пустую операцию, если при вычитании получен нулевой результат.

Команда DECFSZ используется для организации цикла.

*Пример.* MOVLW 0x10H; W загрузить 10H

MOVWF COUNT; W перегрузить в яч. ОЗУ с адресом COUNT

L1 .....; метка L1

< команда 1 > }  
< команда 2 > } повторяется (тело цикла)

DECFSZ COUNT, 1; COUNT ← COUNT-1

GOTO L1

Тело цикла будет повторено столько раз, какое значение было изначально загружено в ячейку ОЗУ COUNT.

$$\text{д) INCF } f, d; f+1 \rightarrow \begin{cases} f, d=1 \\ W, d=0 \end{cases} \text{ команда инкремента (увеличения на 1).}$$

Пример. INCF FSR, 1; FSR ← FSR+1  
 INCF 4, 1; FSR ← FSR+1  
 ↑  
 адрес FSR

е) INCFSZ f, d; f+1 → { f, d=1; W, d=0 } ; команда спец. инкремента.

Инкремент, но с пропуском следующей команды, если получен нулевой результат.

Замечание. Нулевой результат возникнет при переполнении регистра f. Эта команда также используется для организации цикла.

**2) Команды логических операций.**

- а) ANDWF f, d – логическое умножение (“И”).
- IORWF f, d – логическое сложение (“ИЛИ”).
- XORWF f, d – логическое сложение по модулю два (исключающее ИЛИ).

$$f \oplus W \rightarrow \begin{cases} W, d = 0 \\ f, d = 1 \end{cases}$$

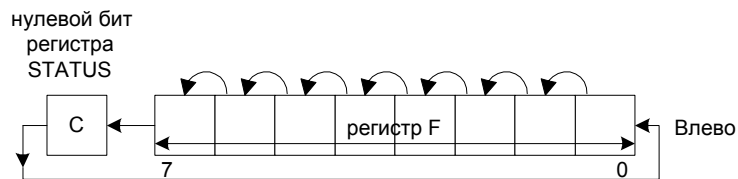
Замечание. Все логические операции выполняются побитно

$$\begin{array}{r} 0010 \quad 1010 \\ \& \\ 0110 \quad 1100 \\ \hline 0010 \quad 1000 \end{array}$$

- б) RLF f, d- } команды логического сдвига: – на 1 бит влево
- RRF f, d- } – на 1 бит вправо

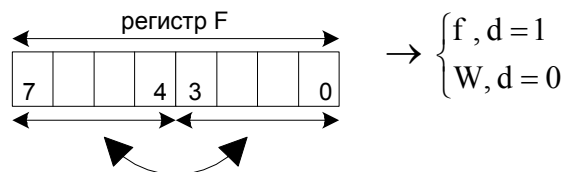
$$\begin{array}{l} \leftrightarrow \\ f \rightarrow \end{array} \begin{cases} f, d = 1 \\ f, d = 0 \end{cases}$$

Схема сдвигов имеет вид:



Для получения сдвига вправо необходимо изменить направление стрелок.

- в) SWAPF f, d ; команда обмена тетрад в регистре f.



- г) CMPI f, d ; команда инверсии бит регистра f.

$$\bar{f} \rightarrow \begin{cases} f, d=1 \\ W, d=0 \end{cases}$$

### 3) Команды пересылки данных.

$$\text{a) MOV } f, d ; f \rightarrow \begin{cases} f, d=1 \\ W, d=0 \end{cases}$$

Переслать значение из регистра  $f$  в рабочий регистр  $W$ , если  $d=0$  и назад в регистр  $f$ , если  $d=1$ .

Пересылка из  $f$  в  $f$  используется для установки признаков результата в регистр STATUS, например, нужно проверить содержимое ячейки ОЗУ.

б) MOVWF  $f$  ;  $W \rightarrow f$  – пересылка значения из  $W$  в  $F$ .

*Замечание.* Во всех перечисленных командах работы с байтами регистра  $f$  выбирается из текущего банка внутреннего ОЗУ, который определяется четвертым и пятым битом регистра FSR.

## 2.10.2. Команды работы с битами PIC16C58

Битовый формат данных команд имеет вид:

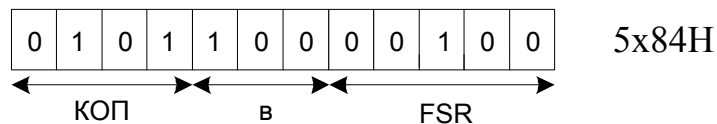
КОП – код операции;

$f$  – регистр, используемый как источник операндов (любая ячейка ОЗУ);

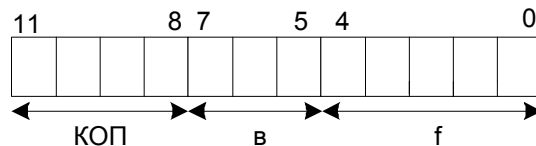
$v$  – номер бита из регистра  $f$ , с которым выполняются действия в команде.

$$\text{a) } \left. \begin{array}{l} \text{BCF } f, v \text{ } \left\{ \begin{array}{l} \text{команды установки – в '0'} \\ \text{бита } \text{№}v \text{ регистра } F \text{ – в '1'} \end{array} \right\} \text{КОП} \begin{array}{l} 0100 \\ 0101 \end{array} \end{array} \right\}$$

*Пример:* BSF FSR, 4 ; установка в единицу четвертого бита FSR (FSR.4=1)



б) BTFSC  $f, v$  ; – команда пропуска следующей команды, если бит  $\text{№}v$  в регистре  $F=0$ .



в) BTFSS  $f, v$  – команда пропуска следующей команды, если бит  $\text{№}v=1$ .

Эти две команды используются для организации условных переходов. При этом в качестве следующей команды используется безусловный переход GOTO, а анализируются биты регистра STATUS.

*Пример.* Переход, если результата равен нулю.

$$\left[ \begin{array}{l} \text{BTFSC STATUS},3 \\ \text{GOTO M1} \end{array} \right]_{Z=0}$$

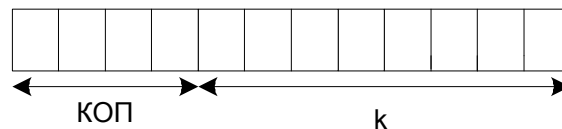
В команде анализируется третий бит регистра STATUS признак нулевого результата Z. При этом команда перехода GOTO будет выполняться (не пропускаться), если анализируемый бит Z=1 (т.е. получен нулевой результат).

$$\left[ \begin{array}{l} \text{BTFSC STATUS},0 \\ \text{GOTO M1} \end{array} \right]$$

команда перехода, если перенос C и т.д.

### 2.10.3. Команды передачи управления и работы с константами

Битовый формат этих команд имеет вид:



k– константа или адрес перехода, используемый в команде исключение составляет лишь одна команда GOTO.

Ее формат имеет вид:

а) MOVLW k ;  $W \leftarrow k$

загрузка в W константы k.

б) GOTO k; безусловный переход на команду с девятиразрядным адресом в k, в текущей странице памяти программ.

в) CALL k; команда вызова подпрограмм по восьмиразрядному адресу k, в текущей странице памяти программ.

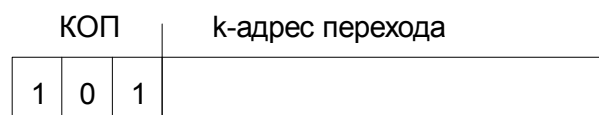
Действия, выполняемые по команде CALL:

– в стек загружается одиннадцатиразрядный адрес команды следующей за командой CALL (адрес возврата).

– Осуществляется переход в текущей странице на адрес k, но при этом девятый бит адреса подпрограммы устанавливается в ноль. PC.8=0.

г) RETLW k– возврат из подпрограммы с загрузкой константы k в рабочий регистр W.

Действия по команде:



– в регистр W загружается k.

– из стека читается одиннадцатиразрядный адрес возврата и осуществляется переход по этому адресу.

Команду RETLW k, удобно использовать для организации таблиц значений памяти программ контроллера.

## 2.10.4. Команды управления режимами работы контроллера PIC16C58

К ним относятся:

**CLRWDT**—команда обнуления сторожевого таймера.

**SLEEP**—команда перевода контроллера в режим пониженного электропотреблением (спящий режим). В этом режиме:

- останавливается работа основного тактового генератора контроллера с частотой  $f_{osc}$ ;
- останавливается работа процессора;
- на всех выводах портов А и В сохраняется текущее значение;
- электропотребление снижается в 1000 раз  $I_{потр} \approx 1 \mu A$ .

Режим используется в случаях, когда контроллер ожидает внешнего события. Для выхода из режима SLEEP необходимо:

- либо сгенерировать внешний сигнал «сброс» (вход MCLR);
- либо сгенерировать «сброс» с помощью сторожевого таймера WDT.

**TRIS К** – команда загрузки управляющего слова в регистры управления порта А и В .

**OPTION** – загрузка управляющего слова из W в регистр управления таймером OPTION .

**NOP** – пустая операция (задержка на один машинный цикл).

## 2.11. Влияние команд контроллера на биты-признаки результата (в регистре STATUS)

- 1) На все признаки результата Z,C,DC влияют только две команды:
  - сложение ADD WF
  - вычитание SUB WF
- 2) влияют на признак переноса C (только на него!) команды сдвигов RLF, RRF.
- 3) Не влияют ни на один из признаков следующие команды: MOV WF, DECFSZ, INCFSZ все битовые команды; команды передачи управления.
- 4) Все оставшиеся команды влияют только на один признак Z (нулевого результата).

*Пример.* Составить программу заполнения константой 0AAH области ОЗУ с адресами: 50H...5FH (второй банк).

## 2.12. Разработка простейшего устройства на основе PIC-контроллера.

Рассмотрим разработку кодового замка на основе PIC- контроллера. Принципиальная схема, которого следующая :



Пусть замок открывается (срабатывает реле РР), если нажата комбинация клавиш 567.

Напишем программу управления кодовым замком:

При удерживании клавиш 5,6,7 замок будет открыт, иначе—закрыт.

**Тема 4.- 14 часов (у.з.-3). Особенности архитектуры универсальных однокристальных 8-разрядных микроконтроллеров (ОМК). Построение МПУ на базе универсальных ОМК.**

Особенности архитектуры и технические характеристики ОМК серии 1816 (MCS-51). Структура, назначение выводов и основные режимы работы ОМК K1816BE51. Организация памяти данных и программ ОМК. Особенности работы портов ввода/вывода ОМК, понятие квазидвухнаправленности. Система команд и основы языка ассемблера ОМК K1816BE51. Способы адресации операндов. Построение микропроцессорного модуля на базе ОМК серии 1816. Особенности подключения внешней памяти программ и данных, а также дополнительных портов ввода/вывода. Расширение ввода-вывода с помощью интерфейсных БИС.

### **3. Проектирование МПУ на основе универсальных восьмиразрядных ОМК семейства MSC–51**

#### **3.1. Основные технические характеристики и структура ОМК K1816BE51**

Контроллер K1816BE51 является функциональным аналогом ОМК I8051(Intel), выполненного по Н-МОП технологии (n-MOS). Существует аналог K1830BE51, который представляет собой тот же контроллер, но выполненный по К-МОП технологии (I80C51).

*Замечание:* ОМК, выполненные по н-МОП технологии обладают большим быстродействием, а по к-МОП минимальным электропотреблением.

Контроллер представляет собой БИС с 40 выводами. Структурная схема контроллера K1816BE51 имеет вид:

Основными функциональными модулями ОМК K1816BE51 являются:

- 1) быстродействующее восьмиразрядное АЛУ с УУ;
- 2) встроенный генератор тактовых импульсов (ГТИ), к входам которого подключен внешний кварцевый резонатор;
- 3) два многофункциональных 16-разрядных таймер-счетчика T0 и T1;
- 4) резидентная память данных РПД (внутреннее ОЗУ) объемом 128 байт;
- 5) резидентная память программ РПП (внутреннее ПЗУ) объемом 4Кх8;

- 6) встроенный контроллер прерываний К Пр., позволяющий одновременно обрабатывать запросы прерывания от восьми источников;
- 7) универсальный адаптер последовательного интерфейса, типа RS 232 (УАПИ);
- 8) четыре 8-разрядных двунаправленных порта ввода/вывода. P0, P1, P2, P3. Направление передачи информации по линиям портов программируется.

*Замечание:* отличие универсальных ОМК от PIC контроллеров в основном состоит в возможности универсальных ОМК расширять номенклатуру устройств путем подключения внешних ИС с использованием внешней системной магистрали.

Например, ОЗУ ОМК может расширяться до 64 кбайт путем подключения внешних ИС, аналогично можно расширять и ПЗУ до 64кбайт.

Быстродействие ОМК K1816BE51 составляет  $0,5 \div 1$  MIPS, при тактовой частоте  $f_{osc} = 3.5 \div 12$  МГц.

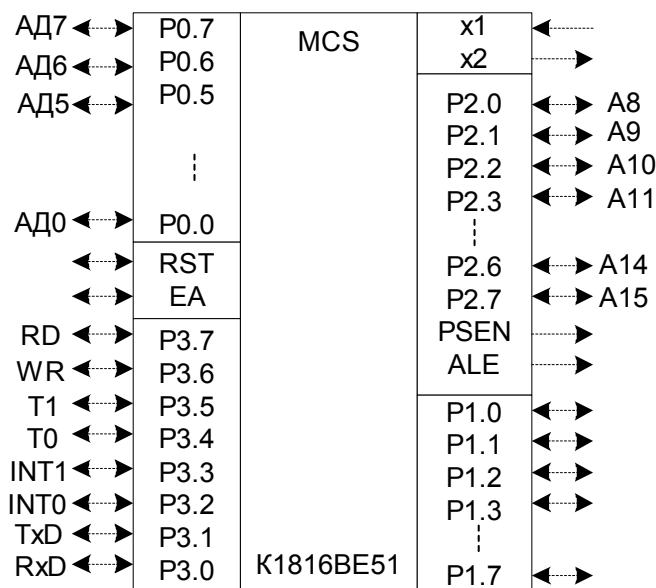
Частота машинных циклов ОМК определяется по формуле:  $f_{mc} = f_{osc} / 12$ , а частота машинных тактов:  $f_{mt} = f_{osc} / 2$ .

Система команд состоит из 111 базовых операций представляемых в виде 1, 2 и 3-байтовых команд. Большинство команд выполняется за 1-2 машинных цикла, исключение составляют лишь две команды: умножения (MUL) и деление (DIV), выполняемые за четыре машинных цикла.

Система команд ориентирована на эффективную обработку как логической информации (имеются битовые операции), так и на выполнение сложных арифметических вычислений (аппаратно реализованы команды MUL и DIV).

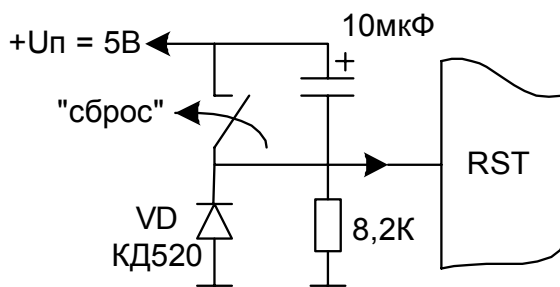
### 3.2. Назначение основных выводов БИС ОМК K1816BE51

На принципиальных схемах ОМК BE-51 изображают в виде:



RST– входной сигнал начальной установки на этот вход должен быть подан сигнал «1» в течении 20мс.

Схема начальной установки по включению питания и по кнопке "сброс" имеет вид:



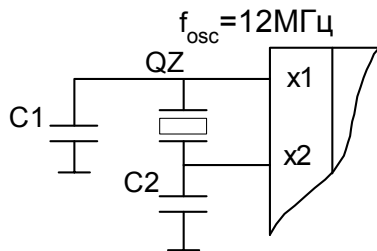
По сигналу «сброс» в ОМК осуществляется:

- перевод всех портов ввода/вывода на ввод информации;
- запрещаются все прерывания контроллера;
- информация во внутреннем и внешнем ОЗУ сохраняется;

–осуществляется переход на программу, начинающую с адреса 0H.

X1 и x2-выводы для подключения, внешнего кварцевого резонатора с частотой  $f_{osc}$  ( $f_{osc}$  может лежать в пределах 3,5÷12 МГц или, если используются аналогичные фирмы Atmel, то 0÷24 МГц).

Схема подключения кварцевого резонатора:



C1 и C2–запускающие емкости.

Контроллер BE51 имеет четыре 8-разрядных порта ввода/вывода P0, P1, P2, P3; большинство из них могут работать в двух режимах:

- в режиме обычного порта ввода/вывода информации;
- в режиме ввода или вывода управляющих сигналов (альтернативный режим).

P0.0...P0.7–выводы двунаправленного 8-разрядного порта P0. В альтернативном режиме эти выводы могут использоваться для выдачи и приема сигнала адреса/данных (АД0...АД7), т.е. эти линии используются для обмена данными Д0...Д7 или передачи адреса А0...А7 во внешние устройства или внешнюю память. Передача адреса при этом по данным линиям подтверждается отрицательным фронтом на выводе ALE.

ALE (Address Latch Enable–разрешение фиксации адреса). Он обязательно генерируется в каждом машинном цикле, поэтому на выходе ALE всегда присутствует сигнал, изменяющийся с частотой машинных циклов  $f_{osc}/12$ .

P2.0...P2.7–линии 8-разрядного квазидвунаправленного порта P2.

*Замечание:* квазидвунаправленным портом называется порт вывода информации, которая при определенном исполнении может применяться и для ввода информации. Например, для ввода информации по линии P2 изначально в эту линию необходимо выдать «1».

В альтернативном режиме P20...P27 используется для выдачи старшего байта адреса А8-А15 во внешнюю память или внешнее устройство.

Передача адреса сопровождается отрицательным фронтом ALE.

P3.0...P3.7—линии 8-разрядного квазидвунаправленного порта P3. В альтернативном режиме эти линии используются для передачи сигналов управления:

–RD, WR-сигнал чтения и записи внешней памяти данных (внешнее ОЗУ). По этим сигналам осуществляется передача данных во внешнюю память через линии P0.0...P0.7;

–T1, T0-входы для подачи внешних импульсов на таймеры счетчики контроллера T1 и T0.

–INT1, INT0-входы запроса прерываний от внешних источников.

–TxD, RxD-выходной сигнал передатчика и входной сигнал приемника встроенного последовательного интерфейса ОМК.

Для перевода любой линии порта P3 в альтернативный режим необходимо в соответствующую линию порта вывести «1».

*Пример:* MOV P3,#3H ;P3=3H (линии TxD и RxD выдаются в младшие разряды порта P3).

–PSEN-выходной сигнал чтения информации из внешней памяти программы (внешнее ПЗУ).

В контроллере BE51 реализовано *гарвардская архитектура памяти*, т.е. память программ (ПП) и память данных (ПД) разделены физически и логически.

Физическое разделение означает, что для обращения к каждой из них используются свои сигналы управления:

–RD, WR-для обращения к памяти данных;

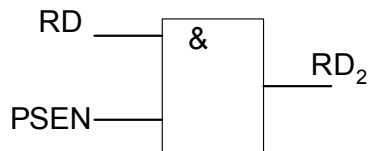
–PSEN-для обращения к памяти программ.

Логическое разделение означает, что для обращения к каждой из них используются разные команды:

–MOV, MOVX-для обращения к ПД;

–MOVC-для обращения к ПП.

В некоторых случаях при отладке программ удобно совмещать в едином устройстве ПП и ПД (в виде единого ОЗУ). В этом случае формируют единый сигнал чтения RD<sub>2</sub>.



В случае такого объединения памяти организуется архитектура Фон-Неймана.

P1.7÷P1.0 - линии 8-разрядного квазидвунаправленного порта ввода/вывода P1.

Линии данного порта не имеют альтернативных функций.

### 3.3. Организация памяти данных контроллера BE51

Память данных (ПД) разделяют на:

- резидентную ПД (РПД)-встроенную, объемом 128 байт, адресами 0÷7FH;
- внешнюю ПД (ВПД), объемом до 64 кбайт, с адресами (0÷0FFFFH).

Для обращения к РПД и ВПД используют различные команды:

- MOVX-пересылка данных ВПД;
- MOV-пересылка данных РПД.

Общая структура РПД имеет вид:

Ячейки с адресами  $0H \div 1FH$  РПД размещают банки регистров общего назначения (РОН). Всего банков четыре, и каждый из них содержит по восемь регистров:  $R_0, R_1, \dots, R_7$ . Текущий банк регистров, с которым работает процессор задается программно.

В адресном пространстве  $20H \div 2FH$  РПД размещается область прямоадресуемых бит, в этой области каждый из 128 битов имеет свой адрес и к каждому из них возможно отдельное обращение с помощью команд.

В области с адресами  $30H \div 7FH$  могут размещаться любые данные пользователя (байты).

Непосредственно РПД в адресном пространстве примыкает область регистра специальных функций (РСФ) с адресами от  $80H \div 0FFH$ .

Вся работа основных узлов ОМК управляется с помощью РСФ. Перечислим основные РСФ:

- 1)  $A^*$  -основной регистр контроллера-аккумулятор(его адрес  $0E0H$ ).

*Замечание:* «\*» означает, что данный РСФ позволяет прямо адресовать свои отдельные биты. Например, для обращения к первому биту аккумулятора в команде записывают  $ACC.1$ .

- 2)  $V^*$  - регистр расширитель аккумулятора может использоваться как обычная ячейка памяти с адресом  $0F0H$ , а также используется для хранения старшего байта результата в операциях умножения и деления.

- 3)  $PSW^*$  (Program Status Word)-слово состояния программы. Его формат:

где

C-признак переноса;

AC-признак дополнительного переноса;

F0-признак пользователя, который устанавливается программно по усмотрению пользователя;

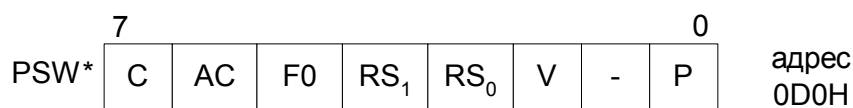
$RS_1, RS_0$ -биты, определяющие номер текущего регистрового банка ('00'-нулевой банк РОН, ..., '11'-третий банк РОН);

V-признак переполнения, т.е. переноса в старший знаковый разряд;

P- признак паритета (нечетности).

- 4) SP - регистр указатель стека в РПД.

Перед записью информации в стек, значение SP увеличивается на единицу ( $SP \leftarrow SP+1$ ), после чтения информации из стека значение



уменьшается на единицу ( $SP \rightarrow SP-1$ ).

Такой режим работы со стеком называется «преинкрементный / постдекрементный».

После сигнала «сброс» SP=07H, т.е. стек располагается в первом регистровом банке.

5) DPTR-16-разрядный регистр указатель данных. В DPTR хранится адрес ячейки при косвенном обращении к внешней ПП или данных.

DPTR состоит из двух 8-разрядных регистров:

–DPH-старший байт;

–DPL-младший байт.

6) P0\*, P1\*, P2\*, P3\*-8-разрядные регистры данных портов ввода/вывода ОМК.

7) IP\*, IE\*-регистр управления приоритетами прерываний и регистр маскирования прерываний (запрещений).

8) TCON\*, TMOD-регистр состояния и регистр управления внутренними таймерами-счетчиками ОМК T1, T0.

9) TH0, TL0 старший и младший байт данных соответственно TH1, TL1 таймера счетчика T0 и T1.

10) SCON\*, SBUF-регистр управления и регистр данных встроенного последовательного интерфейса.

11) PCON-регистр управления потребляемой мощностью ОМК.

### **3.4. Система команд ОМК K1816BE51**

Система команд включает 111 базовых 1, 2, 3-х байтных команд. Первый байт в команде всегда ход операции, а второй и третий либо адреса операндов, либо непосредственно сам операнд.

В системе команд используются следующие виды адресации операнда:

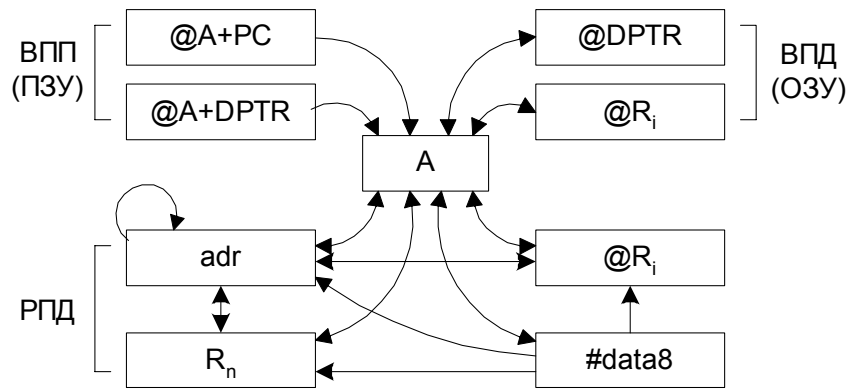
- регистровая, если операнд находится в регистре;
- косвенная-регистровая, если в регистре адрес операнда;
- прямая, если в команде значением задан адрес операнда;
- непосредственная, если в команде задано значение операнда (число);
- неявная, если операнды закодированы в КОП.

Выделяют пять основных групп команд BE51:

- команды пересылки данных;
- команды арифметических операций;
- команды логических операций;
- команды работы с битами;
- команды передачи управления или переходов.

#### **3.4.1. Команды пересылки данных**

Все основные пути пересылки данных в контроллере изобразим в виде графа:



@A+PC-ячейка памяти программ, адрес которой вычисляется как сумма, содержимого счетчика команд PC и аккумулятора (косвенно регистровая адресация);

@A+DPTR- ячейка памяти программ, адрес вычисляется как сумма 16-разрядного регистра указателя DPTR и аккумулятора;

@DPTR-ячейка внешней памяти данных, адрес которого находится в DPTR (адрес 16-разрядный);

@R<sub>i</sub>-ячейка ВПД, 8-разрядный адрес который находится в R<sub>i</sub> (i=0,1);

adr-ячейка, РПД адрес которой прямо указан в команде;

@R<sub>i</sub>-ячейка РПД, 8-разрядный адрес (i=0,1);

R<sub>n</sub>-операнд находится в регистре контроллера, текущего регистрового банка (n=0,7);

#data8- 8-разрядное значение (байт).

Кроме указанных путей имеются еще два:

DPTR ← #data16, C ← bit т.е. в DPTR можно загружать 16-разрядное значение, а в признак переноса C

прямо адресуемый бит.

Можно выделить три типа команд пересылки:

- пересылки данных в РПД;
- пересылки данных в ВПД;
- чтение данных из ВПП.

#### А) Команды пересылки данных в РПД.

**MOV A, src ; A ← src (src=R<sub>n</sub>, @R<sub>i</sub>, adr, #data8)**

**MOV R<sub>n</sub>, src ; R<sub>n</sub> ← src (src=A, adr, #data8)**

**MOV @R<sub>i</sub>, src ; @R<sub>i</sub> ← src РПД (src=A, adr, #data8)**

**MOV adr, src ; adr ← src (src=A, adr, R<sub>n</sub>, @R<sub>i</sub>, #data8)**

*Пример:* 1) переслать содержимое регистра R1 в R2:

MOV R2, R1; такой команды нет

MOV 2, 1 ; переслать ячейку с адресом 1 в 2.

2) MOV @R<sub>0</sub>, A ; из A в ячейку РПД, адрес которой в R<sub>0</sub>.

3) MOV R<sub>3</sub>, #8H ; R<sub>3</sub>=8H.

*Команды работы со стеком:*

**PUSH adr**-загрузить прямоадресуемый операнд в стек;

**POP** adr-прочитать значение прямоадресуемого операнда из стека.

*Пример:* 1) PUSH 1, загрузить в стек значение из РПД с адресом 1Н.

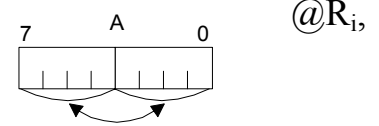
2) загрузить в аккумулятор значение 10Н:

MOV A, #10H

MOV ACC, #10H

**XCH** A, src-команда обмена  $A \leftrightarrow src$  ( $src=R_n$ ,  
adr);

**SWAP** A-обмен тетрад аккумулятора.



### Б) Команды пересылки данных в ВПД.

Если используется 16-разрядная адресация ячеек ВПД, то команды пересылки:

**MOVX** A, @DPTR ;  $A \leftarrow$  из ячейки по адресу DPTR (ПЗУ)

**MOVX** A, @DPTR ; @DPTR  $\leftarrow$  A

Для загрузки регистра DPTR используют команду MOV DPTR, #data16

В случае 8-разрядной адресации ячеек ВПД используют команду:

**MOVX** A, @R<sub>i</sub> ;  $A \leftarrow$  @R<sub>i</sub>

**MOVX** @R<sub>i</sub>, A ; при этом в R<sub>i</sub> 8-разрядный адрес ячейки ВПД.

### В) Команды чтения информации из ВПП (или внутренняя ПП, если EA=1).

**MOVC** A, @A+PC ;  $A \leftarrow$  @A+PC

**MOVC** A, @A+DPTR ;  $A \leftarrow$  @A+DPTR в аккумулятор загружается значение из ячейки ПЗУ, адрес которой определяется как сумма A+PC или A+DPTR.

*Пример:* 1) прочитать содержимое ячейки ПП, расположенной следом за выполненной командой.

MOV A, #0H

MOVC A, @A+PC

2) прочитать из ВПД содержимое ячейки памяти с адресом 2000H.

MOV DPTR, #2000H

MOVX A, @DPTR

## 3.4.2. Команды арифметических операций

А) команды сложения:

**ADD** A, src ;  $A \leftarrow A+src$  ( $src=R_{ni}$ , @R<sub>i</sub>, adr, #data8)

**ADC** A, src ;  $A \leftarrow A+src+C$  (C-признак переноса)

Б) команды вычитания:

**SUBB** A, src ;  $A \leftarrow A-src-C$

В) команды инкремента и декремента:

**INC** src ;  $src \leftarrow src+1$  ( $src=A$ , R<sub>n</sub>, @R<sub>i</sub>, adr)

**DEC** src ;  $src \leftarrow src-1$

**INC** DPTR ; увеличение на единицу 16-разрядного DPTR.



Пример: INC 20H ; увеличение на единицу содержимого ячейки по адресу 20H.

Г) команды умножения и деления:

MUL AB ; (B)·(A) ← A·B , если значения в регистре B ≠ 0 после умножения, то устанавливается признак переполнения V=1.  
 старший байт результата ↑ младший байт результата

DIV AB ; (A)·(B) ← A / B  
 целая часть результата ↑ остаток от деления

Команды работают с целыми числами: DA A- десятичная коррекция аккумулятора после сложения двоично-десятичных чисел.

Команды логических операций.

А) команды логического умножения, сложения и сложения по модулю два.

"И"  $\left[ \begin{array}{l} \text{ANL A, src ; } A \leftarrow A \wedge \text{src (src = R}_n, @R_i, \text{adr, data8)} \\ \text{ANL adr, src ; } \text{adr} \leftarrow \text{adr} \wedge \text{src (src = A, \#data8)} \end{array} \right]$

"ИЛИ"  $\left[ \begin{array}{l} \text{ORL A, src} \\ \text{ORL adr, src} \end{array} \right]$

"искл. ИЛИ"  $\left[ \begin{array}{l} \text{XRL A, src} \\ \text{XRL adr, src} \end{array} \right]$

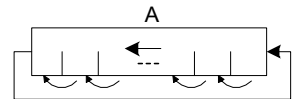
Логические операции выполняются побитно.

Б) команды инверсии, отчистки и сдвига аккумулятора.

CPL A ;  $A \leftarrow \bar{A}$

CLR A ;  $A = 0$

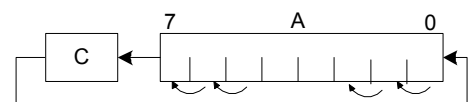
RLA-циклический сдвиг аккумулятора на один бит влево.



RR A- циклический сдвиг аккумулятора на один бит вправо.

RLC A-команда сдвига аккумулятора влево через бит переноса C.

RRC A- команда сдвига аккумулятора вправо через бит переноса C.



### 3.4.3. Команды работы с битами

В данных командах в качестве операндов используется:

- отдельные биты РСФ (помеченные \*);
- биты из области прямоадресуемых бит РПД (20H÷2FH);
- отдельные биты портов ввода/вывода.

В качестве битового аккумулятора данных команд используется признак переноса с контроллера. Таким образом один из битовых операндов и результат выполнения операции как правило сохраняется в признаке переноса C.

А)команды очистки, инверсии и установки бита:

CLR bit ; bit = 0

SETB bit ; bit = 1

CPL bit ;  $\text{bit} = \overline{\text{bit}}$

*Пример:* установка третьего бита аккумулятора:

SETB ACC.3

CPL P0.6 ; инверсия шестого бита порта P0

CLR 20.6H ; отчистка шестого бита ячейки с адресом ??????

Б) команды пересылки бит:

MOV c, bit ;  $c \leftarrow \text{bit}$

MOV bit, c

*Пример:* MOV c, P3.7 ; значение седьмого бита P3 загружается в c.

В) команды битовых логических операций:

ANL c, bit ;  $c \leftarrow c \text{ ”и” } \text{bit} (\vee)$

ANL c, /bit ; подобная команда, но  $c \leftarrow c \text{ ”и” } \overline{\text{bit}}$  с инверсией бита

ORL c, bit ;  $c \leftarrow c \text{ “или” } \text{bit} (\wedge)$

ORL c, /bit ;  $c \leftarrow c \text{ “или” } \overline{\text{bit}}$

*Пример:* найти логическое произведение третьего бита порта P2 с шестым битом порта P3 и результат выдать в нулевой бит порта P0:

MOV c, P2.3 ;  $c \leftarrow P2.3$

ANL c, P3.6 ;  $c \leftarrow c \vee P3.6$

MOV P0.0, c ;  $P0.0 \leftarrow c$ .

### 3.4.4. Команды передачи управления или переходов.

К ним относятся команды условных и безусловных переходов, а также команды вызова возврата из подпрограммы.

Все команды передачи управления делят на три группы:

–команды дальних переходов (long);

–команды абсолютных переходов (absolute);

–команды коротких (относительных) переходов (short).

Команды дальних переходов осуществляется только внутри страниц объемом 2к байта каждая.

Относительные переходы осуществляются на расстоянии  $\pm 128$  байт от текущей команды.

А)команды безусловных переходов и вызова подпрограмм.

LJMP adr16 ; безусловный дальний переход

LCALL adr16 ; дальний вызов подпрограммы

AJMP adr11 ; абсолютный безусловный переход

ACALL adr11 ; абсолютный вызов подпрограммы (adr11–11-разрядный адрес перехода)

SJMP rel ; относительный безусловный переход (rel–смещение относительно текущей команды)

Переход осуществляется по схеме  $PC \leftarrow PC + \text{rel}$  смещение rel может принимать как положительное значения (в этом случае оно записывается в прямом коде), так и отрицательное (в дополнительном коде).

*Замечание:* в случае использования компиляторов с языков ассемблера часто возможно применения обобщенных команд перехода JMP и CALL. В этом случае при трансляции компилятор сам определяет какой тип переходов использовать.

Б)команда относительного безусловного перехода.

JMP @A+DPTR ; безусловный переход по адресу вычисляемому как сумма аккумулятора A и регистра указателя DPTR.

В)команды условных переходов (все они относятся к относительному типу-короткие).

JZ rel ; переход, если результат равен 0

JNZ rel ; переход, если результат не равен 0

JC rel ; переход, если перенос (c=1)

JNC rel ; переход, если нет переноса (c=0)

JB bit, rel ; переход, если значение указанного бита равно 1

JNB bit, rel ; переход, если значение указанного бита равно 0

JBC bit, rel ; переход, если bit =1 с последующей отчисткой бита.

Г)команды организации циклов (относятся к относительным переходам-коротким).

DJNZ R<sub>n</sub>, rel ; R<sub>n</sub>←R<sub>n</sub>-1 (R<sub>n</sub> ≠ 0) уменьшение регистра R<sub>n</sub> на единицу и переход по адресу rel , если R<sub>n</sub> ≠ 0

DJNZ adr, rel ; уменьшение прямоадресуемой ячейки РПД на единицу и переход по адресу rel

CJNZ A,#data, rel ; проверка равенства аккумулятора значению data A=data и переход на rel, если не равно

CJNZ A, adr, rel ; проверка равенства аккумулятора значению из ячейки adr и переход на rel, если не равно

CJNZ R<sub>n</sub>, #data, rel

CJNZ @R<sub>i</sub>, #data, rel ; проверка равенства значения и ячейки, адрес которой в R<sub>i</sub> значению data и переход, если не равно

Г)команды возврата из подпрограмм.

RET ; возврат из подпрограммы с извлечением из стека 16-разрядного адреса возврата.

RETI ; возврат из подпрограммы обработки прерываний в отличии от предыдущей команды осуществляет разблокировку приоритетов прерываний после обработки подпрограммы.

### **3.5. Встроенные таймеры счетчики контроллера K1816BE51**

В контроллере реализованы два 16-разрядных суммирующих таймера счетчика T0 и T1. Они предназначены для подсчета внешних импульсов, подаваемых на входы T0 и T1 (режим счетчика) и для подсчета числа выполненных машинных циклов контроллера (режим таймера).

Оба таймера счетчика управляются с помощью РСФ: TMOD и TCON.

Возможны четыре различных режима работы каждого таймера счетчика (нулевого; первого; второго; третьего-режима).

Нулевой режим работы.

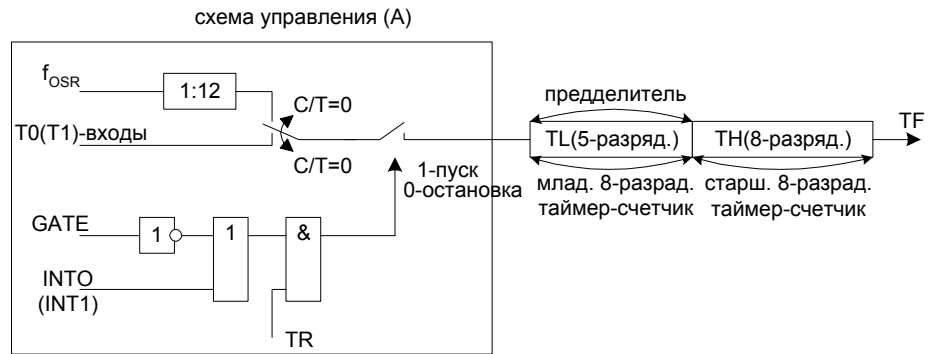
В этот режиме каждый таймер счетчик работает как 8-разрядный двоичный счетчик импульсов с 5-разрядным делителем частоты и схемой управления.

Управление данным режимом осуществляется с помощью TMOD:



GATE–сигнал (значение бита TMOD) управляющий блокировкой работы таймера –счетчика.

Если GATE=1, то работа таймера-счетчика может быть приостановлена внешним сигналом INT=0. Сигнал, INT подается на входы внешних прерываний порта P.3 в альтернативном режиме работы.



Данный режим удобно использовать для подсчета длительности импульсов, подаваемых на вход INT.

C / T–бит задания режима работы таймера-счетчика.

C / T=0–режим таймера.

C / T=1–режим счетчика.

M x.1, M x.0–биты режима работы таймера/счетчика (00–нулевой режим; 01–первый режим; 10–второй режим; 11–третий режим).

Первый режим таймера-счетчика.

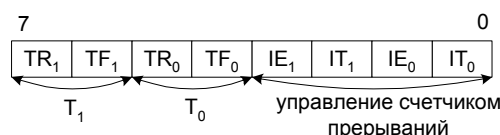
Полностью совпадает с нулевым режимом, отличие- используется не 5-разрядный, а 8-разрядный предделитель частоты TL.

*Замечание:* значение любого таймера/счетчика (T0,T1) может быть прочитано в любой момент времени путем чтения информации из регистра данных таймеров/счетчиков:

–TL0, TH0 для T0;

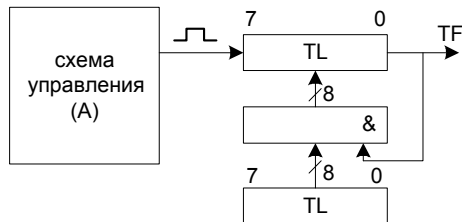
–TL1, TH1 для T1.

*Замечание:* управление сигналом блокировки счета TR, а также чтение сигнала переполнения таймера/счетчика TF осуществляется через регистр специальных функций TCON:



Второй режим таймера/счетчика (автоперезгружаемого генератора частоты).

В этом режиме для подсчета импульсов используется только младший байт TL таймера/счетчика, а старшие байты TH используются для хранения начальных значений счетчика TL. При этом при переполнении младшего байта TL автоматически осуществляется перезагрузка значения из TH в TL и счет возобновляется.



&-схема разрешения перезагрузки.

*Замечание:* изменяя начальное значение в регистре TH пользователь может варьировать период (частоту) следования периодических импульсов на выходе TF. Таким образом реализуется генератор импульсов с программируемой частотой.

Данный режим используют для задания скорости передачи (следования импульсов) по внутреннему последовательному интерфейсу.

Третий режим таймеров/счетчиков.

В этом режиме работает только таймер T0. При этом таймер T0 представляет два независимых 8-разрядных двоичных счетчика:

- TH0-может подстыковать машинные циклы;
- TL0-может подсчитывать машинные циклы или внешние импульсы.

Таким образом используя третий режим в ОМК ВЕ51 можно получить три независимых двоичных счетчика.

### 3.6. Встроенный последовательный интерфейс ОМК

Предназначен для передачи байт по трем выделенным линиям (TxD, RxD и GND) последовательно бит за битом.

Последовательный интерфейс представляет собой универсальный асинхронный приемопередатчик (УАПП).

Возможны четыре режима работы УАПП. Управление и обмен данными с УАПП осуществляется через два регистра специальных функций:

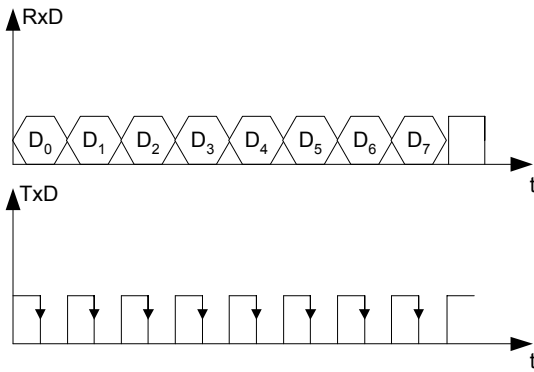
- SBUF-буферного регистра данных;
- SCON-регистра управления интерфейсом.

*Нулевой режим УАПП (сдвигового регистра).*

В этом режиме выход интерфейса RxD (P.3.0) используется как для приема, так и для передачи восьми битовых информационных посылок.

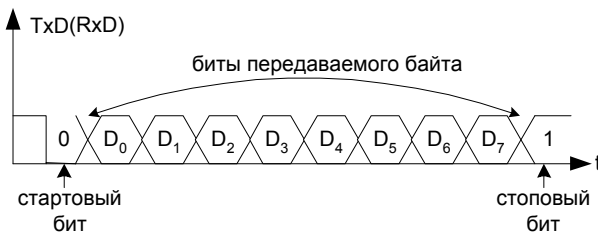
Вывод же TxD (P.3.1) используется для выдачи импульсов синхронизации, сопровождающих каждый передаваемый бит (сдвиговые импульсы).

Скорость передачи в этом режиме постоянна, и определяется частотой следования импульсов  $f_{osc} / 12$



*Первый режим УАПП (старт-стопный десятибитный).*

В этом режиме вывод RxD используется для приема, а TxD для передачи 10-битных информационных сообщений вида:



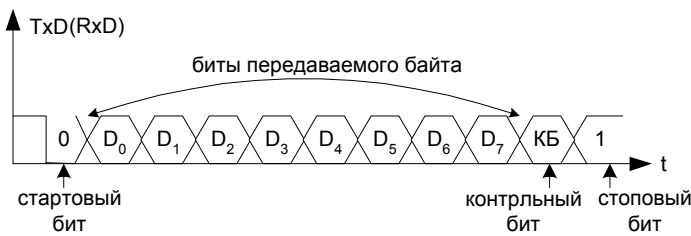
Если по линиям информация не передается, то она находится в состоянии высокого логического уровня «1».

При появлении на линии TxD (RxD) нулевого уровня, он интерпретируется как стартовый бит, после которого начинается отсчет девяти передаваемых бит с заданной скоростью передачи. Следующее появление нулевого уровня интерпретируется как следующий стартовый бит и т.д.

Скорость передачи в этом режиме (частота следования бит) определяется частотой переполнения таймера T1 во втором режиме. (при этом автоматически выхода TF таймера T1 подключаются к УАПП)

*Второй режим УАПП (старт-стопный 11-битный с фиксированной скоростью).*

Работа аналогична предыдущему режиму, только информация сообщения имеет формат:



Значение КБ устанавливается пользователем программно.

Скорость передачи в данном режиме может приниматься только два

фиксированных значения:  $f_{osc} / 64$  и  $f_{osc} / 32$ .

*Третий режим работы УАПП (старт-стопный 11-битный с переменной скоростью передачи).*

Полностью совпадает с предыдущим режимом, но скорость передачи - переменная величина, определяемая частотой переполнения T1 во втором режиме (автоперегружаемый генератор).

Управление режимом работы последнего интерфейса осуществляется с использованием РСФ SCON

7	6	5	4	3	2	1	0
SM <sub>1</sub>	SM <sub>0</sub>	SM <sub>2</sub>	REN	TB <sub>8</sub>	RB <sub>8</sub>	TI	RI

SM<sub>1</sub>, SM<sub>0</sub>–биты, определяющие номер режима УАПП (00–нулевой;...;11–третий).

SM<sub>2</sub>–бит запрещающий прием сообщений с нулевым значением бита КБ.

REN–бит разрешения приема информации по последовательному интерфейсу (REN=0–вся информация не принимается).

TB<sub>8</sub>–бит, определяющий значение контрольного бита передаваемых сообщением.

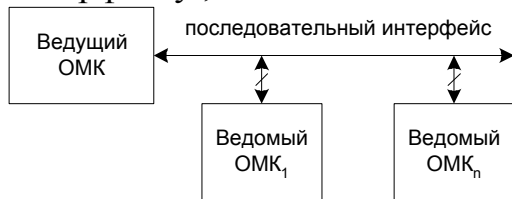
RB<sub>8</sub>–бит, определяющий значение КБ в принятом сообщении.

TI–бит, определяющий, что передатчик пуст и готов к передаче последующих сообщений.

RI– бит, определяющий, что принят очередной байт по последовательному интерфейсу.

*Замечание:* сигналы TI и RI могут генерировать прерывания процессора (если они разрешены).

КБ, а также управляющий бит SM<sub>2</sub> удобно использовать для организации многопроцессорных систем взаимодействующих по последовательному интерфейсу, вида:



Ведущий ОМК управляет приемом и передачей данными с ведомых ОМК.

При этом ведущий передает команды ведомым информационными сообщениями с установленным КБ=1, а следовательно они воспринимаются сразу всеми ведомыми. Обмен же данными осуществляется сообщениями с КБ=0, а следовательно эти данные воспринимаются только одним, выбранным командой, ведомым ОМК.

### 3.7. Задание скорости передачи по последовательному интерфейсу

Во втором режиме работы последовательного интерфейса (УАПП) скорость передачи информации рассчитывается следующим

$$\text{образом: } f_{\text{сдв}} = \left( \frac{2^{\text{SMOD}}}{64} \right) \cdot f_{\text{osc}} \text{ (Гц, Бод)}$$

SMOD – старший седьмой бит РСФ PCON (регистр управления потребляемой мощности).

В первом и третьем режимах работы УАПП (старт-стопные режимы) скорость передачи определяется по

$$\text{формуле: } f_{\text{сдв}} = \left( \frac{2^{\text{SMOD}}}{32} \right) f_{\text{OV}_{\text{TI}}} = \frac{2^{\text{SMOD}}}{32} \cdot \frac{f_{\text{osc}}}{12} [256 - \text{TH}_1]^{-1}$$

где TH<sub>1</sub>–начальное (автоперегружаемое) значение счетчика T<sub>1</sub>, работающего в режиме автоперегружаемого генератора;

$f_{ov_{T_1}}$  -частота переполнения таймера  $T_1$  в режиме автоперегружаемого генератора.

*Замечание:* согласно ГОСТ скорость передачи по последовательному интерфейсу должен выбираться из следующего ряда:  $f_{сдв} - \dots, 1200, 2400, 4800, 9600, \dots$  бит/сек. (Бот).

*Пример:* определить значение автоперегружаемого счетчика  $TH_1$  и бита SMOD для обеспечения скорости передачи по последовательному интерфейсу  $f_{сдв} = 9600$  бит/сек., если  $f_{osc} = 11$  МГц.

$$f_{сдв} = \frac{2^0}{32} \cdot \frac{11 \cdot 10^6}{12} [256 - 253]^{-1} = 9568 \text{ бит/сек.}$$

$TH_1 = 256_{10} = 0FCH$ ;  $SMOD = 0$

Составим программу задания режима работы последовательного интерфейса (инициализация УАПП).

;Модуль инициализации УАПП

CLR TR1 ; TCON TR<sub>1</sub>=0-остановили  $T_1$

MOV TH1, #0FCH ;  $TH_1 \leftarrow 253_{10}$

ANL PCON, #7FH ; SMOD=0

MOV SCON, #11011100B ; задали третий режим работы УАПП(SCON←0DCH)

MOV TMOD, #00100000 ;  $T_1$ -в режим автоперегружаемого генератора

SETB TR1 ; TR<sub>1</sub>=1-запуск  $T_1$

; процедуры приема и передачи байта по УАПП

CIN: JNB RI, CIN ; ожидание приема байта по последовательному интерфейсу RI=1

MOV A, SBUF ; считывание принятого байта в аккумулятор

CLR RI ; SCON.RI=0

COUNT: JNB TI, COUNT ; ожидание готовности передатчика последовательного интерфейса TI=1

MOV SBUF, A ; загрузка передаваемой информации из аккумулятора в последовательный интерфейс SBUF←A

CLR TI ; обнуление TI, TI=0

### 3.8. Система прерываний ОМК ВЕ51

Контроллер позволяет обслуживать прерывания от пяти возможных источников:

- 1) от таймеров  $T_1$  и  $T_0$  при возникновении их переполнения  $TF_1$  и  $TF_0$  (два уровня прерывания).
- 2) Прерывания по внешним сигналам  $INT_1$  (P3.3) поступающим на этот вход и  $INT_0$  (P3.2). Два уровня
- 3) Прерывания от последовательного интерфейса при установке бит RI или TI (один общий уровень для приемника и передатчика).



К каждому из перечисленных источников прерывания могут быть присвоен свой приоритет:

- высший
- низший

Если одновременно возникают несколько запросов от разных источников, то очередность обработки следующая:

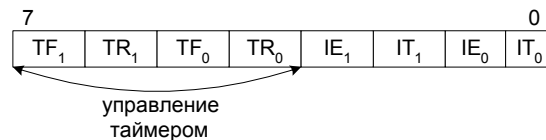
- 1) прерывание  $INT_0$
- 2) прерывание  $T_0$
- 3) прерывание  $INT_1$
- 4) прерывание  $T_1$
- 5) прерывание от УАПП (RI, TI).

Управление работой системы прерывания осуществляется с использованием следующих РСФ:

- TCON (младшая тетрада)
- Регистр приоритетов прерывания  $IP^*$
- Регистр маскирования (запрещения) прерывания  $IE^*$

Регистр TCON управляет запросами, поступающими по входам  $INT_0$  и  $INT_1$ .

TCON



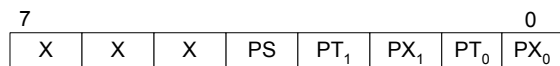
$IE_1, IE_0$ – биты устанавливаются в единицу, если поступил запрос прерывания по входам  $INT_1$  и  $INT_0$  соответственно;

$IT_1, IT_0$ – биты, определяющие признак прерывания на входах  $INT_1$  и  $INT_0$ .

Если  $IT=0$ , то как прерывание будет рассматриваться нулевой уровень сигнала на входе  $INT_1$  или  $INT_0$  соответственно.

Если  $IT=1$ , то как признак прерывание будет рассматриваться будет рассматриваться переход сигнала из нулевого уровня в единичный (прерывание по фронту сигнала).

Регистр приоритетов прерывания  $IP^*$  имеет следующий формат:



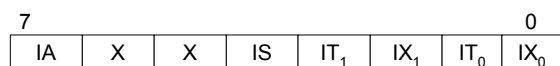
Установка любого из битов этого регистра в «1» означает присвоение одному из источников прерывания высшего приоритета, установка в «0» присвоение низшего приоритета.

PS–соответствует последовательному интерфейсу;

PT<sub>1</sub>, PT<sub>0</sub>–соответствует таймерам  $T_1$  и  $T_0$  соответственно;

PX<sub>1</sub>, PX<sub>0</sub>–соответствует входам  $INT_1, INT_0$  соответственно.

Формат регистра маскирования прерывания  $IE^*$  имеет вид:



Установка IA=0 запрещает все прерывания ОМК.

Установка одного из оставшихся битов «0» означает запрещение одного из источников прерывания.

IS–последовательный интерфейс;

IT<sub>1</sub>, IT<sub>0</sub> – таймеры T<sub>1</sub> и T<sub>0</sub>;

IX<sub>1</sub>, IX<sub>0</sub>–прерывания по входам INT<sub>1</sub>, INT<sub>0</sub>.

### 3.9. Механизм обслуживания прерываний ОМК

1. При поступлении любого из запросов прерывания ОМК анализируется его приоритет, а также установка маски прерывания.

2. Если прерывания возможно, то ОМК на аппаратном уровне генерирует для себя код команды LCALL (вызов п/п).

При выполнении этой команды в ОМК выполняются следующие действия:

–в стеке сохраняется адрес текущей прерванной команды (адрес возврата из п/п);

–Осуществляется переход на п/п обработки прерываний в соответствии с источником прерывания:

INT<sub>0</sub>–адрес п/п 03H;

T<sub>0</sub>–0BH;

INT<sub>1</sub>–13H;

T<sub>1</sub>–1BH;

Последовательный интерфейс (TI, RI)–23H.

По данным адресам различают команды безусловных переходов LJMP непосредственно на п/п обработки прерывания, размещенные в любом месте памяти программ.

Данные команды безусловных переходов LJMP называют векторами прерываний.

3. Любая п/п обработки прерывания должна завершаться командой возврата RETI (а не RET).

*Замечание:* перед выходом из п/п обработки прерывания от последовательного интерфейса обязательно необходимо установить в ноль биты RI и TI регистра SCON (т.е. обнулить запросы прерывания).

### 3.10. Построение расширенных МП систем на основе ОМК ВЕ 51

Если при разработке МП системы необходимо подключить к ОМК ВЕ51 внешнюю память данных (ОЗУ), или внешнюю память программы (ПЗУ), или дополнительное устройство ввода/вывода информации, то ОМК предварительно формирует внешнюю системную магистраль.

Наиболее часто используют трех шинную внешнюю системную магистраль, состоящую из:

–ША (максимально 16-разрядная);

–ШД (8-разрядная);

–ШУ.



### 3.11. Дальнейшее расширение и развитие семейства ОМК MCS-51

В настоящее время выпускается большое количество ОМК, совместимых с семейством MCS-51 (Atmel, Siemens, Philips, Sony,...). Отличие от базового контроллера ВЕ-51 этих совместимых ОМК состоит в следующем:

- имеются дополнительные встроенные устройства: дополнительный таймер T2, сторожевой таймер (WDT); 8–12-разрядный АЦП; встроенный контроллер ШИМ и т.д.
- расширена резидентная память программы (внутреннее ПЗУ) до 16К.
- снижено электропотребление и т.д.
- повышена тактовая частота

Дальнейшим развитием MCS-51 является 8-разрядные контроллеры MCS-151/251 (Intel).

Дальнейшее развитие семейств ОМК 51/151/251 является 16-разрядное семейство ОМК MCS/191/291. Данное семейство характеризуется высокой производительностью и широкой номенклатурой дополнительных устройств для повышения производительности (например, встроенный контроллер событий, модуль захват-сравнение, встроенный контроллер сетей.)

В настоящее время также как развитие семейств 51/151/251 рассматривают 16-разрядные ОМК семейства C167 (Siemens).

---

**Тема 5. - 10 часов (у.з.-3). Обеспечение ввода/вывода дискретных сигналов в МПУ. Системы программно-логического управления.**

Организация ввода информации с группы дискретных датчиков. Согласование уровней сигналов. Виды обмена данными МП с портами ввода/вывода. Методы устранения дребезга контактных датчиков. Подключение клавиатуры к МПУ. Организация вывода информации в МПУ на семисегментные индикаторы в режимах статической и динамической индикации. ЖК-индикаторы. Контроллеры клавиатуры и дисплея. Организация последовательной связи с ЭВМ верхнего уровня по стандартным интерфейсам ИРПС, стык С2. Промышленные сети. Интерфейсы I2C, SPI.

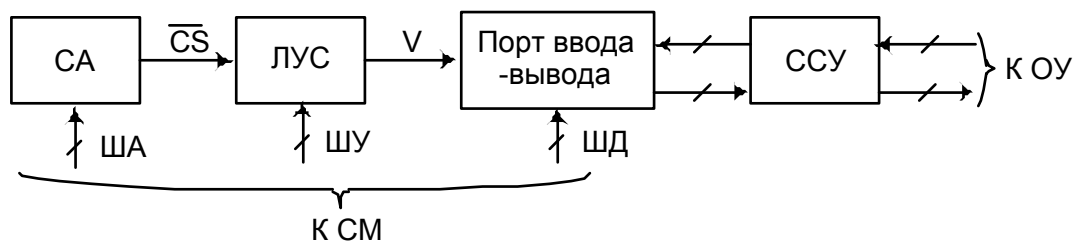
---

## 4. Принципы организации систем дискретного ввода-вывода в МПС

### 4.1. Общая структура систем дискретного ввода-вывода

В данной главе будут рассмотрены принципы построения и подключения к МПС дополнительных внешних устройств дискретного ввода-вывода (УВВ).

Подключение УВВ осуществляется к системной магистрали. Общая структурная схема этих устройств имеет вид:



СА – селектор адреса, осуществляющий обнаружение на шине адреса комбинаций бит, соответствующий адресу данного устройства. При этом СА формирует сигнал выбора устройства  $\overline{CS}$ .

ЛУС – логическая схема управляющих сигналов, которая на основе сигналов ШУ формирует сигналы внутреннего управления УВВ.

Порт ввода-вывода – устройство, осуществляющее непосредственную передачу дискретных сигналов с ОУ на ШД системы или наоборот. Порты реализуются на основе ИС регистров или шинных формирователей.

ССУ – схема согласования уровней. Преобразует уровни логических сигналов с объекта в ТТЛ-уровни, используемые в ОМК. Или наоборот, ТТЛ – в логические уровни, используемые в ОУ.

## 4.2. Реализация селекторов адреса

В соответствии с принципами реализации различают 2 основных вида селекторов адреса (СА):

- с полной дешифрацией;
- с неполной дешифрацией сигналов шины адреса;

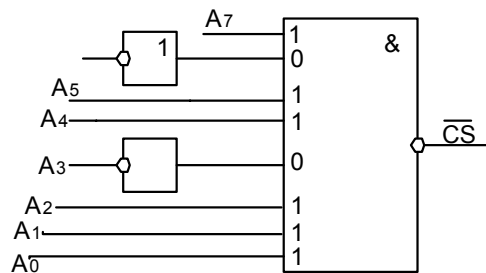
В первых СА для формирования выбора устройства  $\overline{CS}$  использовались все разряды шины адреса.

*Замечание:* В большинстве МПС для адресации УВВ использовалась восьмиразрядная шина адреса (младшие восемь разрядов ША). В СА с неполной дешифрацией для генерации сигнала  $\overline{CS}$  использовались не все, а как правило, только старшие разряды ША.

Простейшим СА с неполной дешифрацией является линейный СА. Для генерации сигнала  $\overline{CS}$  в это СА используется только один разряд ША (например,  $\overline{CS} = \overline{A_7}$ ).

Пример различных типов СА:

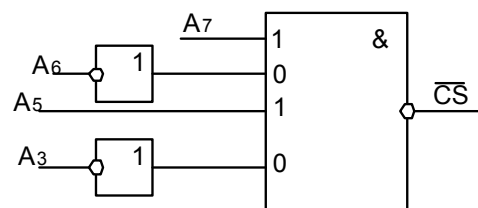
а) с полной дешифрацией:



Адрес устройства ( $\overline{CS}=0$ ):

1011 0111<sub>2</sub>  
       B      7 h

б) с неполной дешифрацией:



Адрес устройства ( $\overline{CS} = 0$ ).

1010 xx xx

нач. A 0 h  
 кон. A F h

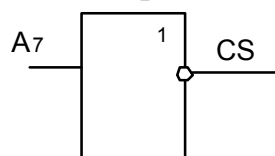
Всего устройство займёт 16 адресов.

в) линейный СА.

Адрес устройства:

1xxx xxxx=80h÷0ffh

127 адресов



Замечание: использование СА с неполной дешифровкой приводит к неэффективному использованию адресного пространства МПС, т.к. в этом случае каждое устройство получает не по одному адресу, а по несколько.

В зависимости от структуры и количества обслуживаемых устройств СА делят на:

- централизованные
- локальные
- распределённые.

Пример реализации централизованного СА на дешифраторе (для восьми устройств).

Пример комбинированного СА на основе дешифратора для 64-х устройств.



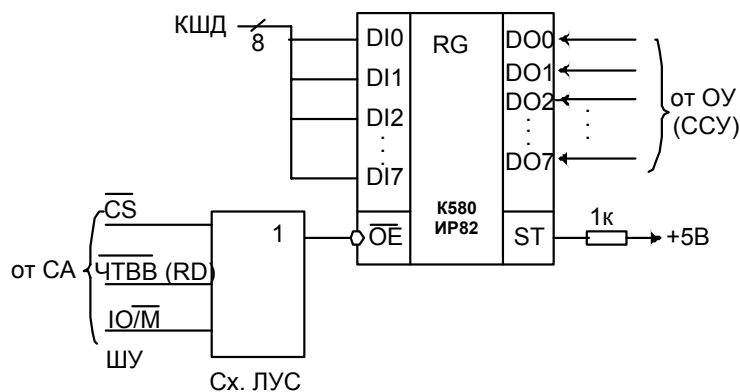
адреса групп:

$\underbrace{10}_{\text{№ группы}}$ 
 $\underbrace{\text{xxx}}_{\text{№ устройства группы}}$

Замечание: В большинстве МПС, как правило реализуют центральные или комбинированные СА на основе ПЗУ или ПЛМ.

### 4.3. Реализация портов ввода-вывода

Простейшие порты ввода логических сигналов реализуются на основе регистров (триггеров). Общая схема порта ввода логических сигналов, подключаемого к системной магистрали контроллера, имеет вид:



$\overline{OE}$  – сигнал разрешения выдачи информации от ОУ на шину данных. Если  $\overline{OE}=1$ , то все линии DI и D0 находятся в третьем состоянии, т.е. отключены от ШД и ОУ.

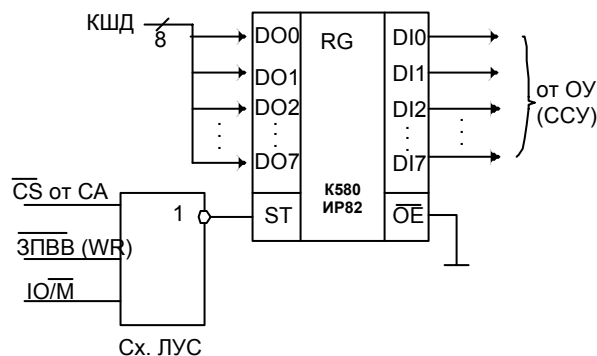
ST – сигнал записи информации в регистр (ST=1 – регистр пропускает через себя информацию без записи).

Заключение: Вместо регистра для реализации порта ввода используются шинные формирователи, инверторы, буферные схемы, но обязательно с третьим состоянием выхода, которое необходимо для отключения от ШД МПС.

Фрагмент программы чтения информации ОМК ВЕ51 из изображённого порта имеет вид (пусть его адрес 40h), т.е. сигнал CS от СА генерируется при значении 40h на шине адреса.

Замечание: т.к. порты ввода/вывода в ОМК ВЕ51 подключаются как дополнительные внешние ячейки ОЗУ, то для обращения к ним используется команда MOVX.

Простейший порт вывода дискретной информации также реализуется на основе регистров.



Запись информации в порт вывода (регистр) осуществляется по сигналу  $\overline{ЗПВВ}=0$ . Если выбран данный порт сигналом  $CS=0$  и  $IO/M=1$ .

Фрагмент программы записи информации в изображённый порт для ВЕ51 (адрес 41h):

Возможно подключение с СМ универсальных портов ввода/вывода, реализованных на основе специализированных БИС. Примером такого порта может служить программируемый периферийный адаптер (ППА) К80ВВ55. Схема подключения ППА к СМ имеет вид:

ППА К580 ВВ55 (Intel 8255) представляет собой 3 8-ми разрядных порта ввода-вывода дискретной информации А, В, С, реализованных в одном корпусе. Каждый 8-ми разрядный порт используется как для вывода так и для ввода информации. Конкретно режим его работы задаётся путём записи специального байта (управляющего слова) в сигнальный внутренний регистр (регистр управляющего слова). Обращение к каждому из портов ППА осуществляется с использованием адресных линий А0 и А1 (00 – порт А, 10 – порт В, 01 – порт С, 11 – РУС).

Адрес портов на изображённой схеме будет принимать значения:



A7	A6	A5	A4	A3	A2	A1	A0		
1	0	1	0	1	0	0	0		=0A9H порт А
1	0	1	0	1	0	0	1		=0A9H– В
					1	0	0		=0AAH–С
					1	1	0		=0ABH –РУС

*Пример.* Написать фрагмент программы программирования режима работы ППА и выдачи байта 0CDH в порт А.

Замечание: Аналогично подключению ППА с системной магистралью осуществляется подключение иных специализированных БИС:

- контроллера клавиатуры и индикаторов (К580 ВД79, I8279)
- контроллер двоичного последовательного интерфейса (К580 ВВ51, I8251)
- три дополнительных 16-ти разрядных таймера (К580 ВИ53, I8253)
- контроллер прямого доступа к памяти ПДП (К580 ВТ54) и т.д.

#### **4.4. Программно-аппаратное обеспечение ввода дискретных сигналов.**

При вводе дискретных логических сигналов в МПС, как правило, решаются три основные задачи:

- 1) согласование логических уровней дискретных сигналов. Все вводимые логические сигналы должны быть преобразованы в ТТЛ уровни;
- 2) устранение дребезга контактных датчиков с которых осуществляется ввод логического сигнала. Задача решается при вводе информации с клавиатуры, переключателей, контактов реле, пускателей и т.д.;
- 3) обеспечение ввода очень коротких по длительности логических сигналов.

##### **4.4.1. Согласование логических уровней сигналов.**

Возможны три случая для диапазона изменения входного логического сигнала:

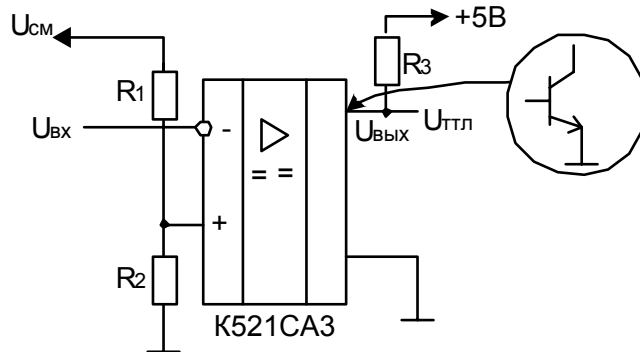
- а) если входной логический сигнал имеет логические уровни:
  - логического нуля  $U_0$ :  $-24В \leq U_0 \leq 0,4$
  - логической единицы  $U_1$ :  $2,4В \leq U_1 \leq 24В$ .

В этом случае для преобразования используются ограничители (диодные или транзисторные).

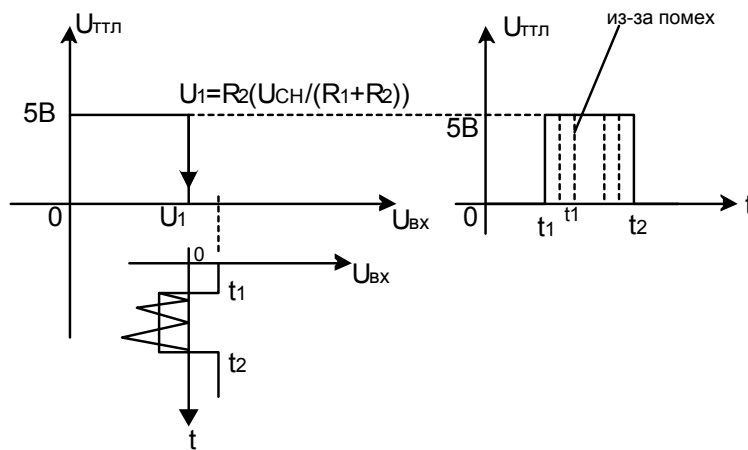
- б)  $U_{ВХ}$ :  $|U_0| < 15В (24В)$ ;  $|U_1| < 15В (24В)$   
 $|U_1 - U_0| < 2В$ .

– малый размах логических уровней.

В этом случае для приведения к ТТЛ – уровням используют схемы на аналоговых компараторах.



Характеристика вход-выход

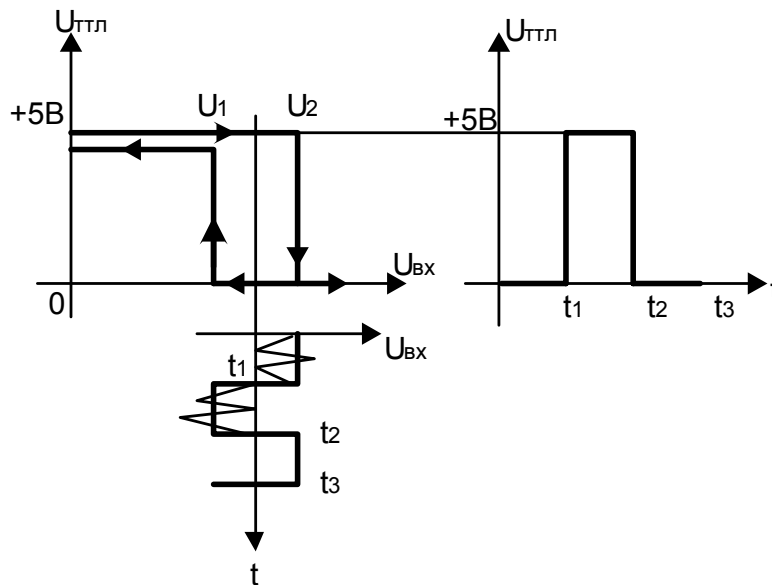


Для устранения чувствительности к помехам используют триггеры Шмитта, реализованные на аналоговых компараторах. В этом случае схема имеет вид

Отрицательным свойством изображённого преобразователя является чувствительность к помехам во входном логическом сигнале, если амплитуда помехи больше чем:  $\frac{|U_1 - U_0|}{2}$ .

Замечание: Подбором сопротивлений  $R_1 \div R_4$  добиваются требуемых значений  $U_1$ ,  $U_2$ , и  $\Delta U$ .

Напряжение  $\Delta U = U_2 - U_1$  определяет гистерезис изображённой схемы. Его введение позволяет значительно повысить помехоустойчивость по сравнению с предыдущей схемой:



Наличие помехи во входном сигнале с амплитудой  $\frac{\Delta U}{2}$  никак не сказывается на выходном сигнале  $U_{ТТЛ}$ .

в) Если входной сигнал  $U_{ВХ}$  или логические уровни:

$$U_{ВХ}: |U_0| > 24В; |U_1| > 24В.$$

В этом случае преобразователь уровней обязательно должен содержать схему гальванической развязки, которая предохранит цепи управления контроллера от перенапряжений. Гальваническая развязка может реализоваться на основе:

- магнитных элементов (магнитных сердечников);
- с использованием оптопар.

Рассмотрим схему гальванической развязки на оптопаре.

Замечание: Источники питания  $U_1$  и  $U_2$  должны быть полностью электрически несвязанными, например, используют две разные обмотки одного трансформатора.

#### 4.4.2. Схемы устранения дребезга контактных датчиков

При коммутации контакта различных контактных устройств (кнопок, переключателей, контактов реле) в первый момент времени возникает переходной процесс. Это явление при коммутации контакта называют дребезгом.

$t$  дребезга лежит в диапазоне от 1 до 30 мкс и зависит от конструкции контакта.

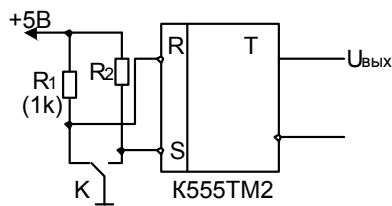
Для устранения влияния дребезга используют два типа способов:

- аппаратные способы
- программные способы.

Аппаратные способы реализуются в виде следующих типов схем:

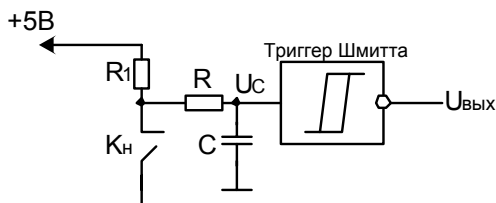
- триггерные схемы
- схемы на основе RC цепочек
- схемы на основе счётчиков или сдвиговых регистров.

Простейшая триггерная схема для устранения дребезга имеет вид:



RS – триггер переключится только по первому нулевому импульсу на входе R, все остальные импульсы вызванные дребезгом далее восприниматься не будут, т.к. триггер уже переключился.

Простейшая RC – цепочка, предназначенная для устранения дребезга:



В настоящее время выпускается большая номенклатура интегральных схем устранения влияния дребезга контактов на основе счётчиков и сдвигающих регистров. Например: ИС СД4490 (6 элементов устранения дребезга).

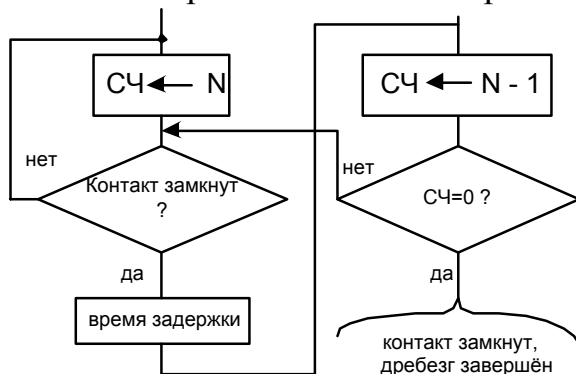
Схема одного из них:

### 4.4.3. Программные способы устранения влияния дребезга контактов

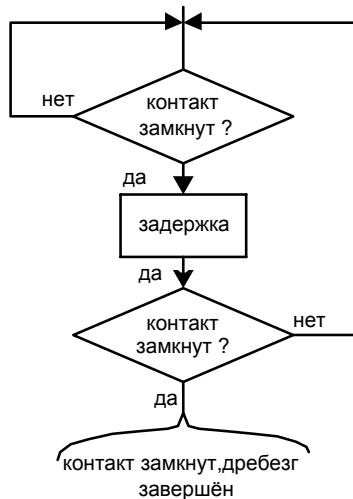
Выделяют два основных способа:

- способ N-кратного считывания;
- способ временной задержки.

Схема алгоритма способа N-кратного считывания имеет вид:



Фрагмент программы метода N-кратного считывания для ОМК ВЕ51:



M1: MOV R0,#N; R0←N  
 M3: JB P1.0,M1; если P1=0, то M1  
       MOV R1,#N1; временная задержка  
 M2: DJNZ R1,M2; R1←R1-1, если R1≠0,  
 то M2  
       DJNZ R0,M3; R0←R0-1, если R0≠0,  
 то M3

.....  
 Время  $t_3$  временной задержки равно:

$$t_3 = (2 \cdot N1 + 2) T_{\text{МЦ}}$$

Величина N выбирается так, чтобы время:

$$t_3 = (4T_{\text{МЦ}} + t_3) \cdot N \approx t_{\text{ДРЕБ}}$$

Схема алгоритма способа временной задержки имеет вид.

Фрагмент программы способа временной задержки для ОМК ВЕ51 имеет вид:

M1: JB P1.0, M1;  
       MOV R0, #N1;  
 M3: MOV R1, #N1;  
 M2: DJNZ R1, M2;  
       DJNZ R0, M3;  
       JB P1.0, M1;

.....  

$$t_3 = [N1 \cdot (2 + 2 \cdot N2) + 2] \cdot T_{\text{МУ}}$$

Время задержки  $t_3$  выбирается:

$$t_3 \geq t_{\text{дрб.}}$$

#### 4.4.4. Аппаратное обеспечение вывода логических сигналов

При выводе логических сигналов из МПУ решают три основные задачи:

- обеспечивают требуемый ток в нагрузке при выдаче логического сигнала (усиления по току);
- преобразование логических сигналов из ТТЛ-уровней напряжения в иные логические уровни;
- осуществляют оптоэлектронную гальваническую развязку;

Для обеспечения требуемого тока в нагрузке могут использоваться следующие устройства:

- интегральные буферные схемы (типа шинных формирователей К580ВА86/87; К589 АП16/24 либо интегральные буферные элементы К555 АП1/АП2) при этом ток в нагрузке обеспечивается от 20 до 100мА;
- с использованием транзисторных эмиттерных повторителей;
- с использованием повторителей на ОУ.

Преобразование ТТЛ-уровней в иные логические уровни напряжения может осуществляться с использованием:

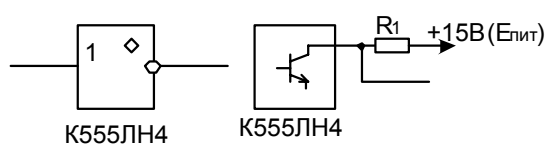
- интегральных преобразователей уровней;
- интегральных схем с открытым коллектором и повышенной нагрузочной способностью;
- схем на дискретных элементах (транзистор).

Существуют следующие типы интегральных преобразователей уровней:

- из ТТЛ в ЭСЛ («0» → -5В, «1» → +5В); [К500 ПУ125 из ТТЛ в ЭСЛ (прямое); К500 ПУ124 из ЭСЛ в ТТЛ (обратное)].
- из ТТЛ в RS («0» → -12В, «1» → +15В); [К170 АП2 (прямое); К170 УП2 (обратное); МАХ 232 (прямое и обратное);
- из ТТЛ в КМОП («0» → 0≈В, «1» → +15В); [К176 ПУ4 (прямое); К564 ПУ6 (обратное)].

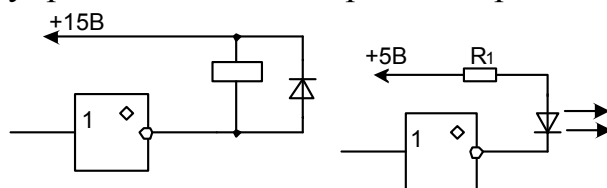
Замечание: При выборе функциональных интегральных схем (АЦП, ЦАП, аналог. коммутаторов и др.) необходимо обращать внимание – какими логическими уровнями управляется данная схема и нужно или нет использовать преобразователи уровней.

Интегральные схемы с открытым коллектором и повышенной нагрузочной способностью удобно использовать для увеличения уровня напряжения логической единицы:



Некоторые интегральные схемы с ОК поддерживают напряжение  $E_{П}$  вплоть до 30В и током нагрузки до 50мА.

Такие ИС удобно использовать для управления индикаторами, требующими повышенное напряжение (газоразрядные и некоторые светодиодные индикаторы). Кроме того ИС с ОК часто используют для управления миниатюрными герконовыми реле:



Если требуется обеспечить нестандартные логические уровни на выходе, то используют схемы на дискретных элементах типа:

Основным элементом преобразования является транзистор VT1, включённый по схеме с общей базой. Если напряжение на эмиттере VT1 соответствует логической единице (ок. +5В), то VT1 открывается и его коллекторным током открывается VT2. При этом на выходе схемы формируется напряжение  $U_{ВЫХ} = -U_2$ .

Если на эмиттере VT1 напряжение соответствует уровню логического нуля, (меньше 0,4В), то VT1 заперт смещением формируемым на диодах VD1 и VD2. В этом случае коллекторный ток не протекает, VT2 закрыт и  $U_{ВЫХ} = +U_1$ .

Замечание: Для повышения КПД изображённые схемы выходную цепь на транзисторе VT2 чаще реализуют на двух транзисторах (как в усилителях мощности). В этом случае будет исключено протекание прямого тока между  $+U_1$  и  $-U_2$ .

Замечание: Вместо транзисторов для преобразования уровней часто используют ОУ.

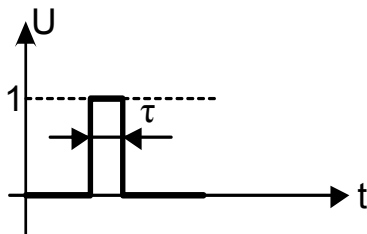
#### 4.4.5. Особенности ввода коротких логических сигналов

Короткими логическим сигналами называют сигналы, продолжительность которых соизмерима с машинными циклами микропроцессора.  $\tau \approx \text{МЦ}$ .

Для ввода в МПС логических сигналов  $\tau \leq T_{\text{МЦ}}$  используют специальные приёмы:

Пример:

M1: JB P1.0, M1; ( $2\text{мц}=12.2\text{мт}$ ).

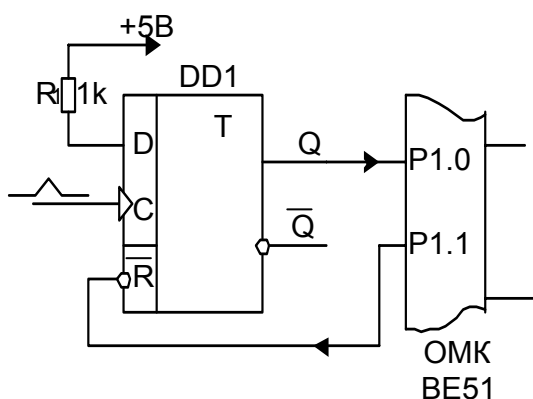


Команда JB осуществляет опрос линии порта P1.0 только в течение одного машинного такта из 24-х, в течение которого эта команда выполняется. Следовательно, появление короткого

импульса  $\tau$  на линии P1.0 ( $\tau \leq T_{\text{МЦ}}$ ) может не совпасть с машинным тактом опроса линии P1.0. Т.о. циклически выполняемая команда JB с большой вероятностью может не обнаружить короткого сигнала.

Аналогичная ситуация и при подаче коротких сигналов  $\tau$  на входы запроса прерывания контроллера ( $\overline{\text{INT1}}$  и  $\overline{\text{INT0}}$  для ОМК BE51) или на счётные входы внутренних таймеров-счётчиков (T1, T0 для BE51). Т.к. опрос перечисленных входов схемно также реализуется в течение одного МТ за МЦ.

Для фиксации подобных коротких импульсов используются внешние триггерные схемы, которые запоминают сигнал.



CLR P1.1; сброс триггера

SETBP1.1;

M1: JB P1.0, M1;

Рассматриваемая последовательность команд и схема обеспечит ввод любого короткого сигнала длительности  $\tau \geq 0,01$  нс.

Замечание: Для обеспечения ввода и подсчёта коротких логических

сигналов, следующих через короткие интервалы времени друг за другом, используют внешние двоичные счётчики на триггерах. Информация с данных счётчиков водится через порты ввода-вывода в ОМК.

## 4.5. Обеспечение взаимодействия микропроцессорных устройств и ЭВМ верхнего уровня

Для обеспечения взаимодействия МПУ как между собой так и с ЭВМ ВУ используют:

- последовательные интерфейсы (типа RS232, ИРПС «токовая петля» и др.);
- промышленные сети, типа FieldBus.

Перечисленные последовательные интерфейсы реализуют радиальную структуру подключения. Это означает, что для подключения к каждому МПУ необходимо реализовать свой последовательный интерфейс:

КПИ – встроенные контроллеры последовательного интерфейса;

ССЛС – схема согласования с ЛС;

ЛС – линия связи;

Подобные соединения устройств называются соединениями типа точка-точка (point-point).

В интерфейсе RS232C информация передаётся побайтно по трём проводным линиям бит за битом. При этом используются две информационные линии (Т×D передаваемых данных, R×D – принимаемых данных) и общий провод – GND (земля).

Каждый передаваемый бит в линии связи интерфейса RS232C кодируется уровнем напряжения «0» – -12В, «1» – +12В. Скорость передачи в RS232C для ЛС типа «витая пара» измеряется в пределах 100 Кбод. Длина линии может достигать 0,5 км.

В интерфейсе ИРПС «токовая петля 20 мА» передача информации осуществляется по четырёхпроводной линии, при этом каждая пара проводов образуют свою токовую петлю:

- R<sub>+</sub>, R<sub>-</sub> – токовая петля для приёма информации;
- T<sub>+</sub>, T<sub>-</sub> – токовая петля для передачи информации.

Каждый передаваемый бит кодируется уровнем тока:

«1» – 18÷20мА

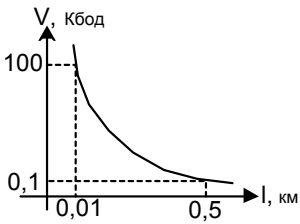
«0» – 0÷3 мА.

Замечание: Использование тока для кодирования информации повышает помехозащищённость передаваемой информации, но как правило, снижает скорость передачи.

Длина линии связи ИРПС «токовая петля» может достигать 10км. при скорости передачи 100б÷19,2 Кбод (1бод=1бит в 1с.).

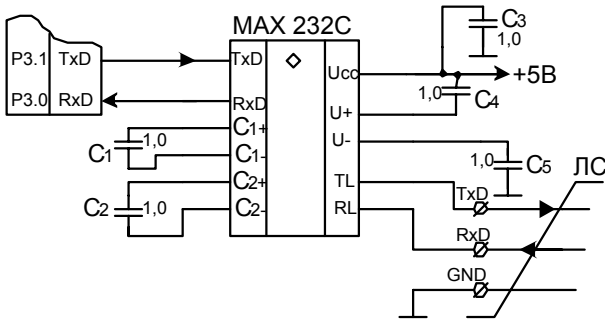


Следует помнить: увеличение длины линии связи требует снижения скорости передачи для обеспечения бесперебойного обмена информацией:

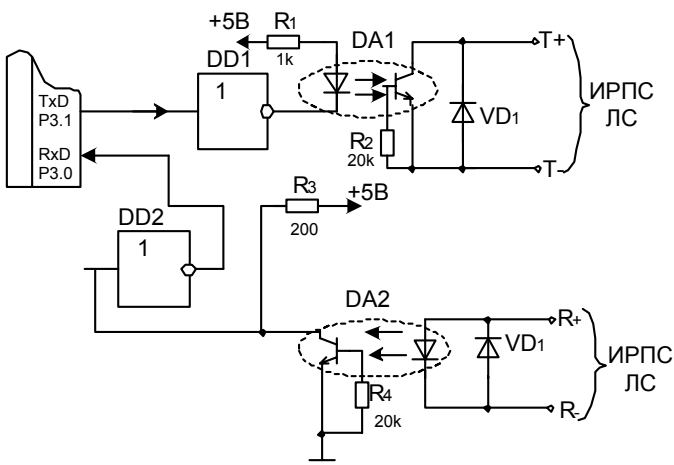


Снижение скорости передачи объясняется увеличением переходных процессов в длинных линиях и помехами.

В качестве схемы сопряжения с линией связи ССЛС в интерфейсе RS232C удобно использовать интегральную схему типа MAX232:



Для сопряжения напряжения интерфейса ИРПС «толовая петля» используют следующую схему:



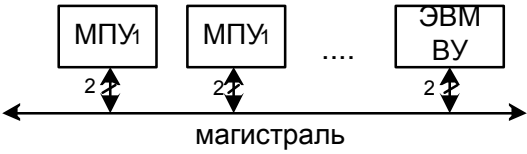
DA1, DA2 – АОТ110А;  
VD1, VD2 – КД522.

Диоды VD1, VD2 служат для защиты транзистора и диода оптопар от возможных обратных напряжений в линии связи. Пример схемы сопряжения источника и приёмника сигналов с ИРПС «токовая петля»:

Интерфейс ИРПС «токовая петля» обязательно подразумевает гальваническую развязку.

Промышленная сеть представляет собой последовательную двухпроводную магистраль, в которой параллельно может находиться несколько микропроцессорных устройств и ЭВМ ВУ:

В настоящее время используются следующие типы промышленных сетей:



– CAN – сеть (Control Area Network) – сеть области управления. Это последовательная магистраль с разрешением

коллизий на основе приоритетов. Скорость передачи 10Мбит/сек., длина ЛС –до нескольких километров.

- ProfiBus (RS-485) – маркерная последовательная магистраль, передача и приём информации в которой каждым МПУ осуществляется в строго выделенные интервалы времени . Скорость передачи может достигать 12 Мбит/сек. Длина ЛС – до неск. км.

В современных 16-ти разрядных ОМК SAB80C167 (Infineon), Z8 (Zilog) реализованы встроенные контроллеры промышленных сетей. Однако возможна и дополнительная реализация сети с использованием внешних контроллеров: SAB80 C91 (CAN–сеть).

Для реализации сопряжения с ЛС используют следующие шинные формирователи:

- RLE6850 (Infineon) – CAN-сеть.
- MAX485 (MAXIM) – Profi Bus.

Замечание: Эти интегральные схемы подобны MAX232, только для промышленных сетей.

В рассмотренных промышленных сетях обязательно подразумевается гальваническая развязка с ЛС.

## 4.6. Средства взаимодействия МПУ с оператором

Для взаимодействия с оператором в МПУ используют:

- индикаторы для отображения информации
- клавиатуры для ввода информации.

Если количество кнопок, подключаемой клавиатуры МПУ>5, то при подключении кнопки организуют в виде матрицы:

Такое включение позволяет экономить число требуемых линий портов ввода-вывода для подключения кнопок:

Пример: 12 кнопок:

- традиционный способ – 12 линий
- матричный –7 линий.

Опрос (сканирование) матричной клавиатуры осуществляют следующим образом:

- 1) На линии порта PS последовательно выдают двоичные коды, содержащие 0 только в одном разряде:

PS <sub>3</sub>	PS <sub>2</sub>	PS <sub>1</sub>	PS <sub>0</sub>
1	1	1	0
1	1	0	1
1	0	1	1
0	1	1	1

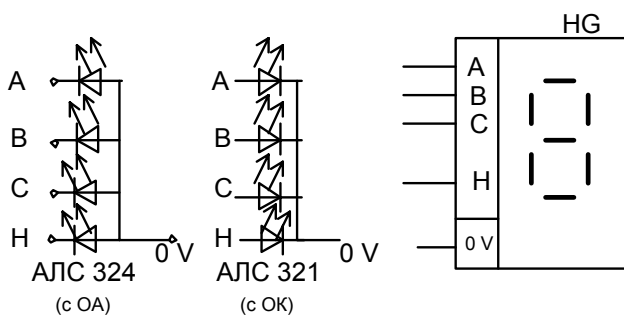
- 2) Переключение двоичных кодов осуществляют через короткие интервалы времени:

20÷40 мкс. После каждого переключения осуществляется считывание порта PO.

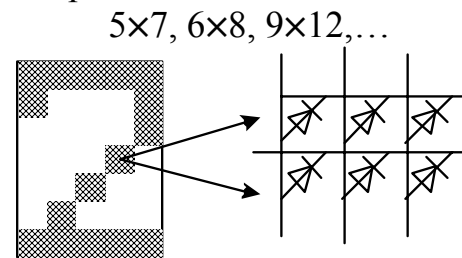
Если в строке, в которой была нажата кнопка, был выдан «0», то нулевой уровень будет передан в соответствующий столбец и введён через порт PO. По информации портов PSi PO можно судить о нажатой кнопке. при вводе информации из порта PO необходимо устранить дребезг контактного датчика. Диоды VD1, ..., VD4 необходимы для защиты линий порта PS от к.з. при одновременном нажатии кнопок в одном столбце.

Для индикации алфавитной и цифровой информации в МПУ, как правило используют семисегментные (матричные), светодиодные, газоразрядные или ж/к индикаторы.

Светодиодный семисегментный индикатор представляет собой совокупность светодиодов, включённых по схеме с общим анодом или катодом:



В матричных индикаторах выводимый символ синтезируется из отдельных мелких сегментов, образованных, как правило, в матрице:

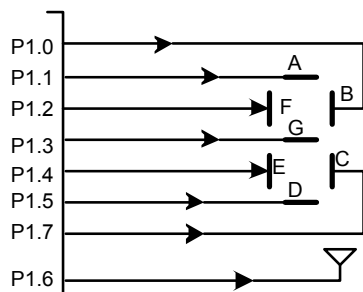


Возможны два основных способа подключения индикатора:

- в режиме статической индикации
- в режиме динамической индикации.

Для простоты рассмотрим указанные способы подключения на примере сегментного индикатора.

В режиме статической индикации для подключения каждого сегмента индикатора используется отдельная линия порта вывода



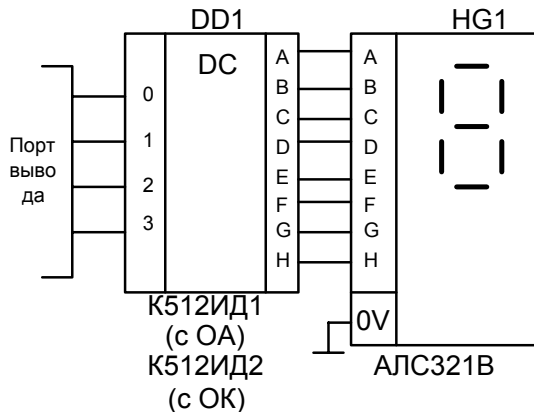
**Определение:** Семисегментным кодом символа называют двоичный код, который при записи его в порт вывода обеспечивает свечение на индикаторе данного символа:

Разряды порта вывода →

P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
C	H	D	E	G	F	A	B

Код двойки: 00111011=3ВН, т.о. семисегментный код двойки – 3ВН.

Для сокращения требуемых линий порта вывода при подключении 7-ми сегментного индикатора используют внешние 7-ми сегментные дешифраторы:



Семисегментный дешифратор DD1 обеспечивает преобразование 16-ричной цифры в её семисегментный код.

Замечание: недостаток использования DD1: вывод на индикацию только HEX цифр.

Для уменьшения требуемых линий портов при подключении сразу нескольких индикаторов

(индикаторная линейка) используют режим динамической индикации.

Схема подключения индикаторов имеет вид:

Пример: для подключения пяти семисегментных индикаторов } 8 линий

- в статическом режиме потребуется  $5 \times 8 = 40$  линий портов;
- в динамическом режиме :  $8 + 5 = 13$  линий.

## 4.7. Жидкокристаллические индикаторные панели

В современном МПУ (пейджеры, сотовые телефоны и др.) часто используют жидкокристаллические индикаторные панели. Они предназначены для вывода как символьной, так и графической информации. При этом графическое изображение синтезируется отдельными точками.

Каждая ЖКИ панель включает:

- матричный ЖК индикатор;
- контроллер вывода информации, реализующий режим динамической индикации.

Наличие контроллера позволяет осуществить вывод информации в динамическом режиме без участия центрального процессора.

Простейшим ЖКИ панелями являются символьные 1-, 2-, 4-х строчные панели с числом символов в строке до 40.

К таким панелям относятся ЖКИ, например, SUNLIKE (Bolymin).

*Замечание:* Основной отличительной особенностью ЖКИ является подключения к системной магистрали МК.

Рассмотрим схему подключения ЖКИ типа CS1609D (Bolymin) 1 строка 16 символов.

$$\text{Адрес индикатора } 1010101_x = \begin{cases} 0AAH \\ 0AAB \end{cases}$$

Поясним основные выводы контроллера DD1:

D7,...,D0–линии предназначенные для записи выводимой на индикацию информации, а также команд индикатора;

ЖКИ панель имеет внутреннее ОЗУ индикатора, в котором и записывается отображаемая информация (как правило в виде ASCII кода)

R/w–вход линия определяющая осуществляется чтение (1) либо запись (0) информации.

RS–вход линия, передающая записывается данные (0) или команда

E–вход сигнал разрешения операции чтения/записи с контроллера.

Перечень основных команд контроллера DD1 индивидуален для каждой индикаторной панели и определяется по инструкции. К этим командам относятся:

–команда записи информации В ОЗУ индикатора;

–команда чтения информации ОЗУ;

–очистка ОЗУ (экрана);

–формирование своих символов в виде точечной матрицы.

## 4.8. Совмещенные контроллеры клавиатуры и индикатора

Разработаны БИС (например K580ВД79, I8279,...) представляют собой устройства одновременно осуществление:

–автоматическое сканирование матричной клавиатуры без участия центрального процессора;

–вывод на индикаторные линейки в режиме динамической индикации без участия центрального процессора требуемой информации.

Такие микросхемы называются контроллеры клавиатуры и индикатора (ККИ).

Контроллер ККИ K580ВД79 (I8279) обеспечивает выполнения следующих функций:

1)автоматическое сканирования матричных клавиш размером 8x8

2)автоматическое устранение дребезга контактных датчиков

3)индикация в автономном режиме информации на линейке из шестнадцати 7-сегментных индикаторов (в динамическом режиме)

Общая схема использования ККИ 580ВД79 имеет вид:

*Замечание:* ККИ K580ВД79 имеет два внутренних ОЗУ:

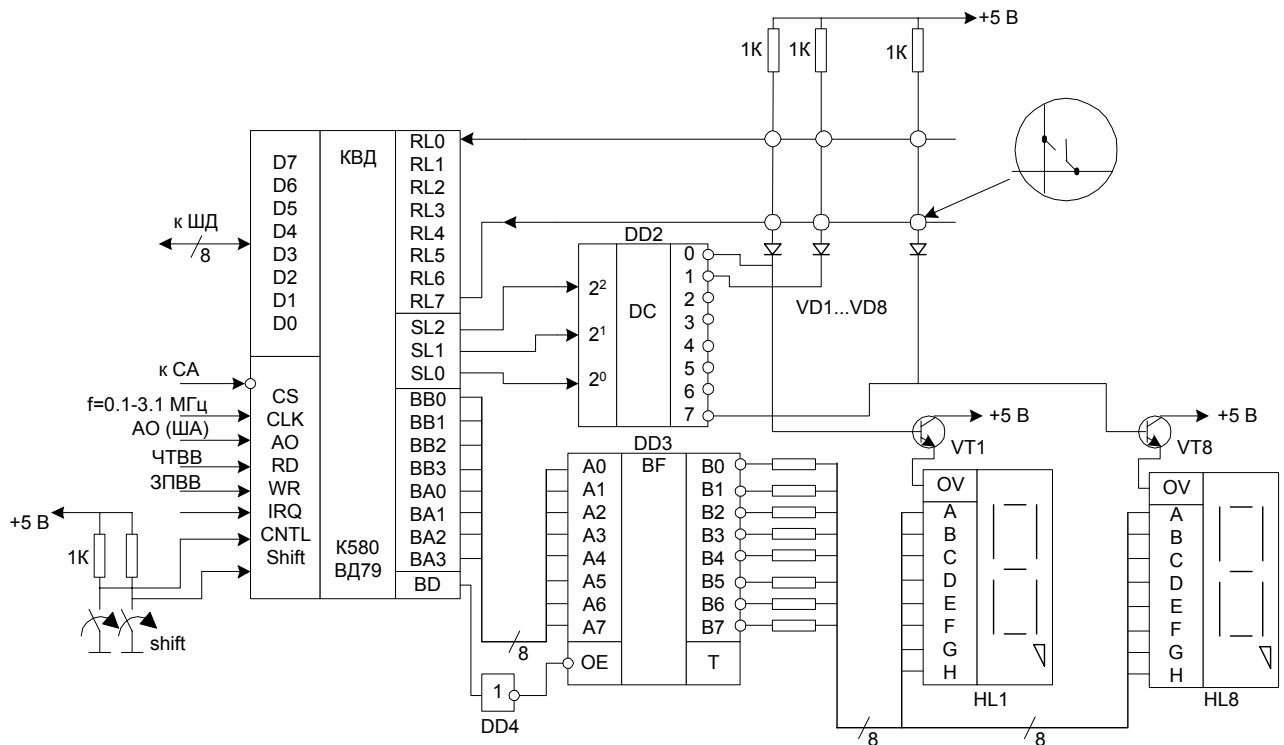
–ОЗУ клавиатуры размером 8байт для хранения кодов нажатых клавиш

–ОЗУ индикаторов размером 16байт для хранения 7-сегментных кодов отражаемых символов

Назначения основных выводов ККИ ВД79:

–D0...D7-линии, используемые для обмена данными и командами с ККИ, подключенных к ШД МПУ

–CS-входной сигнал выбора ККИ для обмена информацией с процессором, подключен к селектору адреса



–CLK-вход для подачи тактовой частоты синхронизации ККИ ( $f=0,1-3,1$  МГц)

–A0-вход адресная линия, определяющая доступ к двум внутренним регистрам ККИ:

1) регистру команд ( $A0=1$ )

2) регистру данных ( $A0=0$ )

*Замечание:* через перечисленные регистры и осуществляется обмен данными и командами ККИ.

–RD, WR-вход сигналы чтения и записи информации ККИ, подключается к сигналам ШУ

–IRQ-вых сигнал запрос прерывания, генерируемый при нажатии клавиш, подключается к входам запроса прерываний МП;

–CNTL, Shift-вход линии для подключения управляющих клавиш ‘CONTR’ и ‘Shift’. В отличие от кнопок матричной клавиатуры не генерирует прерывания;

–RL0...RL7-вход линии чтения состояния строк матричной клавиатуры;

–SL0...SL3-выход линии сканирования клавиатуры и выбора индикатора в линейке при динамической индикации, на них выдается двоичный номер опрашиваемого столбца матричной клавиатуры и индикатора в линейке, на который выдают информацию. Увеличение номера на этих линиях осуществляется с интервалом  $\sim 1 \div 100$  мс.

–BB0...BB3, BA0...BA3-выход линии, на которые выдается 7-сегментный код отображаемого символа. Информация на линиях изменяется синхронно с изменением состояния SL0...SL3.

–BD-выход сигнал гашения всех индикаторов в линейке, генерируется автоматически ККИ при смене отображаемой информации.

**Тема 6. - 18 часов (у.з.-3). Организация ввода-вывода аналоговых сигналов. Системы сбора и обработки информации. Системы цифрового управления.**

Вывод аналоговых сигналов. Простейшие ЦАП, их недостатки и способы устранения. Интегральные ЦАП. Получение биполярного напряжения на выходе ЦАП. Ввод аналоговых сигналов. Структура канала ввода. Устройство выборки-хранения. Алгоритмы аналого-цифрового преобразования. Интегральные АЦП. Устройства выборки хранения. Средства согласования с объектом управления. АЦП с дельта-сигма модуляцией.

## 5. Организация подсистем ввода/вывода аналоговых сигналов МПУ

### 5.1. Вывод аналоговых сигналов

Основным элементом подсистем вывода аналоговых сигналов является устройство преобразования двоичных кодов  $K$  в аналоговые величины ( $U$ ,  $I$ ). В МПУ в качестве таких устройств используются:

- 1) ЦАП;
  - 2) ШИМ с НЧ фильтром на выходе;
- Общая структура ЦАП имеет вид:



$$U_{\text{цап}} = \eta \cdot K, \quad \eta = \text{const.}$$

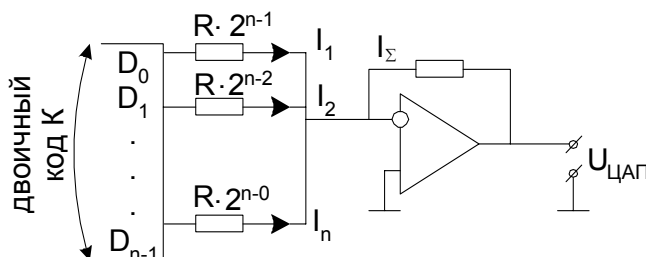
ЦАП для МПУ выбирают по следующим его характеристикам:

- 1) погрешность преобразования (разрешающая способность ЦАП)

$$\delta = \frac{1}{2^n - 1} \cdot 100\%, \quad n\text{-разрядность};$$

- 2) время преобразования - интервал времени между подачей кода на входы ЦАП и получение устойчивого напряжения на выходе;
- 3) Нелинейность преобразования  $D_{\text{нл}}(K) = U_{\text{цап}} - \eta^K$ .

Простейшим ЦАП является ЦАП суммирования токов на основе ОУ:





$$U_{\text{ЦАП}} = -U^{"1"} \left[ D_0 \frac{1}{2^{n-1}} + D_1 \frac{1}{2^{n-2}} + \dots + D_{n-1} \frac{1}{2^0} \right] = -U^{"1"} \frac{K}{2^{n-1}}$$

$D_0 \dots D_{n-1}$  – двоичные разряды кода, подаваемого на ЦАП;

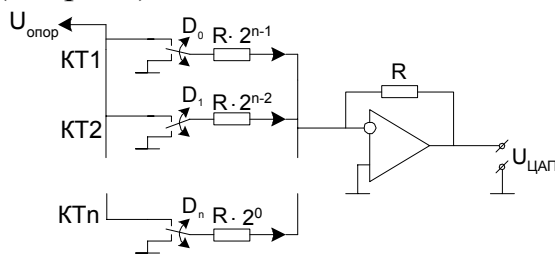
$U^{"1"}$  – напряжение соответствующее уровню «1».

Недостатки данного ЦАП:

– из-за нестабильности уровней логической «1» и «0» ( $U^{"1"}$  и  $U^{"0"}$ ), ЦАП имеет высокую погрешность ( $U^{"1"} = 5 \div 2,4\text{В}$ ;  $U^{"0"} = 0 \div 0,4\text{В}$ );

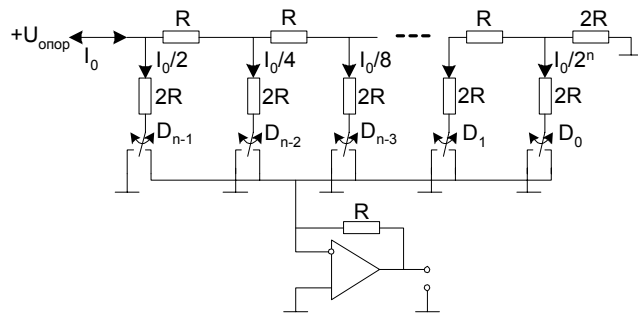
– требуется определенное количество точных резисторов с номиналами отличающихся на 2.

Для устранения первого недостатка логические сигналы  $D_0 \dots D_{n-1}$  используют только для управления ключами, которые коммутируют сопротивление к постоянному высокостабильному источнику напряжения (опорное).



КТ1...КТn – как правило транзисторные ключи на полевых транзисторах.

Для устранения неполадки второго недостатка, (требуем большой номенклатуры резисторов) используют лестничные на резисторах типа “R-2R”.



$$I_0 = \frac{U_{\text{ОПОР}}}{R} = \text{const}$$

$$I_{\Sigma} = \sum_{i=1}^n I_i$$

$$U_{\text{ЦАП}} = -R \left[ D_0 \frac{I_0}{2^{n-1}} + D_1 \frac{I_0}{2^{n-2}} + \dots + D_{n-1} \frac{I_0}{2^1} \right] = -RI_0 \left[ D_0 2^{-(n-1)} + D_1 2^{-(n-2)} + D_{n-1} 2^{-1} \right] =$$

$$= -U_{\text{ОПОР}} \frac{K}{2^{n-1}}$$

где  $K = D_{n-1}; D_{n-2}; \dots$  – целое двоичное число;

Основным свойством схемы “R-2R” является способность делить ток  $I_0$  на 2 при любой коммутации ключей  $D_{n-1}, D_{n-2}, \dots, D_0$ . Осуществляется многократное деление на 2.

*Пример:* Определить двоичное число, которое при выводе ЦАП сформирует на выходе напряжение  $-5,3\text{В}$ , если опорное напряжение равно  $U_{\text{опор}}=+10,24$ , а  $n=11$ .

$$K = -\frac{U_{\text{ЦАП}}}{U_{\text{опор}}} \cdot 2^{n-1} = \frac{-5,3}{-10,24} \cdot 2^{n-1} = 530_n = 212_k = 1000010010_2$$

Современные ЦАП строятся как правило на основе схем “R-2R” с внутренним или внешним суммированием токов на ОУ.

В зависимости от включения данного ОУ и источника опорного напряжения ЦАПы могут работать в режимах:

- однополярного выходного напряжения  $0 \leq U_{\text{ЦАП}} \leq U_{\text{ОПОР}}$ .
- Двуполярного -  $U_{\text{ОПОР}}/2 \leq U_{\text{ЦАП}} \leq U_{\text{ОПОР}}/2$ .

## 5.2. Общая структура подсистемы ввода аналоговых сигналов

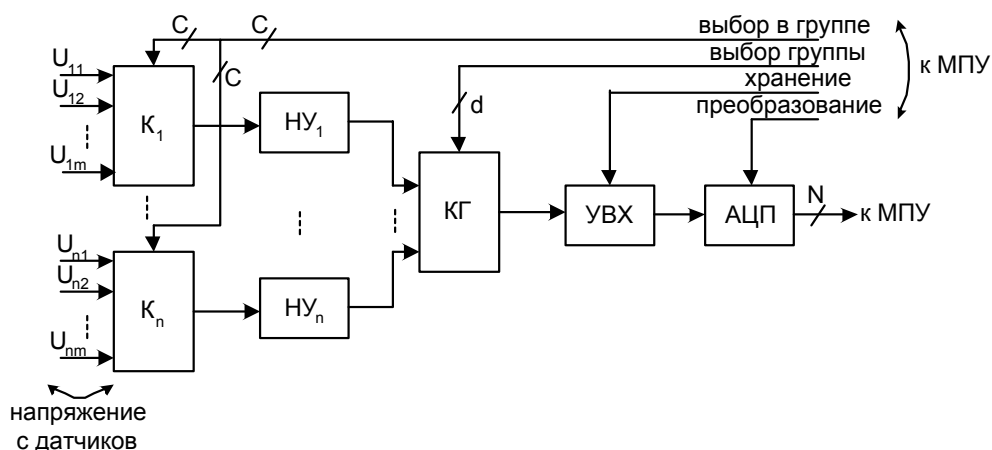
Структура имеет вид:

$U_{11}, \dots, U_{nm}$ —измеряемые сигналы;

$K_1, \dots, K_n$ —аналоговые коммутаторы сигналов в группах; выбор сигнала определяется кодом подаваемым на шину “выбор в группе”;

$HU_1, \dots, HU_n$ —нормирующий усилитель -для преобразования диапазона измерения сигнала к максимальному диапазону измерения сигнала на входе АЦП (УВХ);

КГ—коммутатор групп- для подключения одной из групп;



УВХ—для сохранения сигналов неизменными, во время его АЦ преобразования.

$K_1, \dots, K_n$ ; КГ—реализуется на основе интегрированных схем КМОП и НМОП аналоговых коммутаторов (серии К590, К591, CD4000 и т.д.)

Основными характеристиками при выборе аналоговых коммутаторов является:

- число коммутаторных входов;
- диапазон коммутируемых напряжений;
- максимально коммутируемый ток;
- переходное сопротивление с коммутируемого ключа и др.

Ну как правило реализуется на основе инвертируемых и не инвертируемых усилителей с ООС на основе ОУ.

Коэффициент усиления НУ выбирается исходя из диапазона измеряемого сигнала и диапазона входного сигнала на АЦП:

$$K_{\text{ну}} = \frac{U_{\text{max АЦП}} - U_{\text{min АЦП}}}{U_{\text{max СГ}} - U_{\text{min СГ}}}$$

В случае измерения (слабого изменения) сигналов, ОУ для НУ выбирается по напряжению смещения нуля  $U_{\text{см0}}$ :

$$U_{\text{см0}} \cdot K_{\text{ну}} \leq \Delta_{\text{АЦП}} \quad \Delta_{\text{АЦП}} = \frac{U_{\text{max АЦП}} - U_{\text{min АЦП}}}{2^n - 1}.$$

$\Delta_{\text{АЦП}}$  – измерение входного напряжения АЦП на единицу младшего разряда (максимальная погрешность).

Изображенная схема подсистемы ввода аналоговых сигналов используется при измерении сигналов с сильно различающимися диапазонами измерения. В этом случае сигналы с близким диапазоном измерения формируются в единую группу. Такое объединение позволяет уменьшить число требуемых НУ.

### 5.3. Основные типы АЦП, используемые в МПУ. Основные характеристики выбора АЦП

В настоящее время в МПУ используют следующие типы АЦП:

- АЦП на основе преобразователей “напряжение-частота”. К данному типу относят АЦП с дельта – сигма -модуляцией;
- АЦП “время-импульсные” однократным и двойным интегрированием сигнала, они преобразуют измеряемое напряжение в временной интервал;
- АЦП последовательного счета и последовательных приближений. Они строятся на основе ЦАП;
- АЦП параллельного преобразования, данный тип быстродействующий, но малоразрядный.

Выбор АЦП для МПУ осуществляют на основе следующих характеристик:

- 1) погрешность преобразования  $\Delta = \pm \frac{U_{\text{max АЦП}} - U_{\text{min АЦП}}}{2^n - 1} \cdot \frac{1}{2}$
- 2) время преобразования – интервал времени между подачей сигнала на вход АЦП и появление устойчивого кода на выходе;

- 3) апертурное время – интервал времени, в течении которого невозможно точно сказать к какому из моментов времени относится полученное измерение (погрешность измерения сигнала во времени). Для интегральных АЦП немаловажными являются:
- 4) интерфейс передачи полученного в МПУ:
- 5) наличие встроенного источника опорного напряжения, УВХ.

Рассмотрим подробно, программно -аппаратную реализацию основных типов АЦП.

## 5.4. Программно–аппаратная реализация время–импульсного АЦП с однократным интегрированием

Общая структура имеет вид:

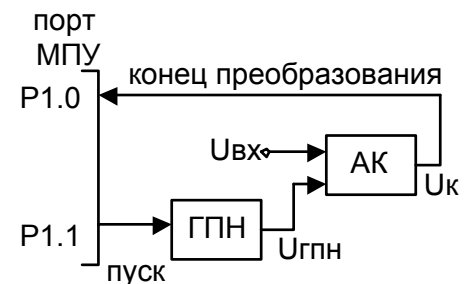
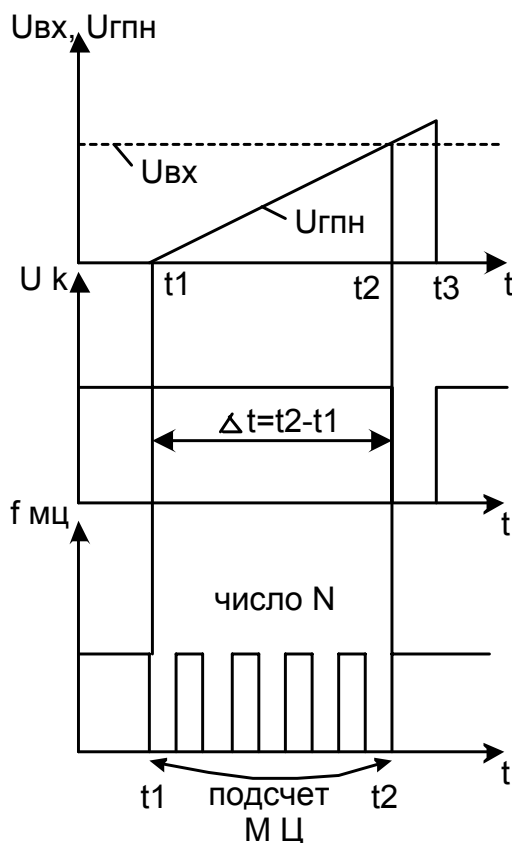
АК –аналоговый компаратор, используется для сравнения  $U_{вх}$  и  $U_{гпн}$

$$U_k = \begin{cases} 1, & U_{вх} > U_{гпн} \\ 0, & U_{вх} \leq U_{гпн} \end{cases}$$

ГПН –генератор пилообразного напряжения.

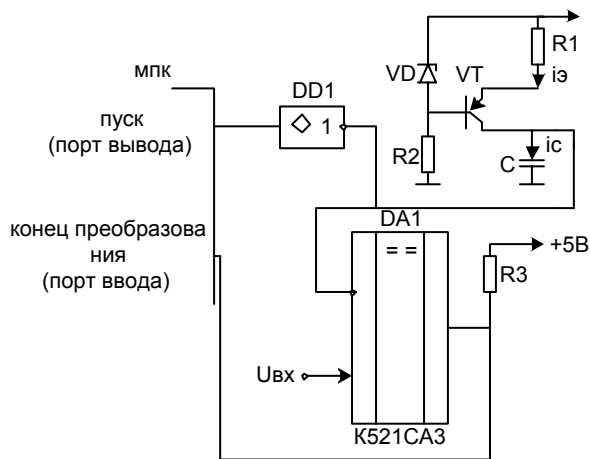
Алгоритм управления АЦП с однократным интегрированием:

- 1) МПУ в момент  $t_1$  формирует сигнал Пуск=0, тем самым заставляя ГПН формировать на выходе линейно –нарастающее напряжение.



- 2) Осуществляется постоянный опрос выхода АК, в случае, когда выход АК  $U_k=0$  ( $U_{вх} = U_{гпн}$ ), формируется второй момент времени  $t_2$ .

- 3) Измеренный интервал времени  $\Delta t$  с помощью внутренних таймеров – счетчиков и будет являться числовым эквивалентом  $N$  входного напряжения  $U_{вх}$ .



На элементах VD, R1, VT, R2 реализован генератор постоянного тока  $i_c = \text{const}$ , которым заряжают конденсатор C:

$$i_c \approx i_{\text{э}} = \frac{U_{VD} - U_{\text{эБVT}}}{R_1} \approx \text{const}$$

Получим выражение, описывающее время – импульсное преобразование АЦП

$$U_c = U_{\text{ГПН}} = \frac{1}{C} \int_{t_2}^{t_1} i_c dt = \frac{i_c}{C} (t_2 - t_1)$$

$$U_{\text{ВХ}} = U_c = \frac{i_c}{C} \Delta t \quad \Delta t = \frac{U_{\text{ВХ}} C}{i_c} \quad (1)$$

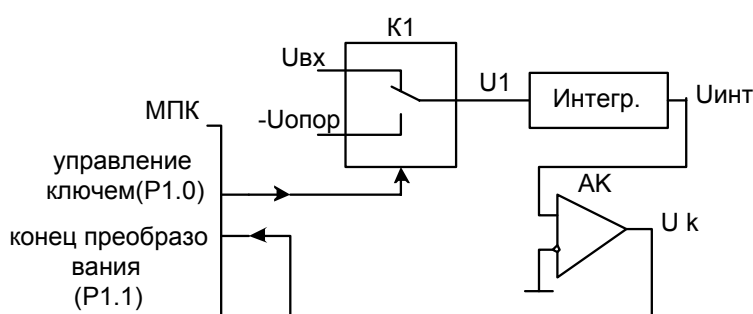
Получили выражение, описывающее время – импульсное преобразование АЦП.

*Замечание:* Т.к. в формулу (1) входят параметры схемы (C,  $i_c$ ), которые могут изменяться от температуры или старения, то можно сказать, что точность преобразования данного АЦП невысока и зависит от изменения этих параметров.

Изображенный АЦП имеет низкую помехозащищенность, так присутствие в  $U_{\text{ВХ}}$  случайной помехи может резко повлиять на точность результата.

## 5.5. Время–импульсный АЦП двойного интегрирования

Обладает более высокой помехоустойчивостью. Общая структурная схема имеет вид (при программно – аппаратной реализации):

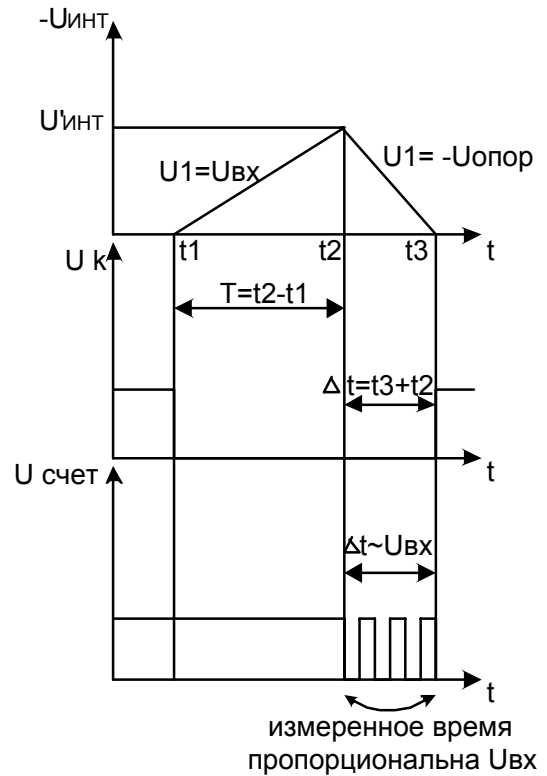


$$U_k = \begin{cases} 0, & U_{\text{инт}} < 0 \\ 1, & U_{\text{инт}} \geq 0 \end{cases}$$

Работу АЦП двойного интегрирования поясним на временной диаграмме.

Алгоритм работы АЦП двойного интегрирования:

- 1) В момент  $t_1$  (когда напряжение на интеграторе  $U_{\text{инт}}=0$ ) на вход интегратора подают  $U_1=U_{\text{вх}}$ . В течении заранее заданного интервала  $T=t_2-t_1$  осуществляется интегрирование  $U_{\text{вх}}$ , при этом напряжение на выходе интегратора достигает  $U'_{\text{инт}}$ .
- 2) В момент  $t_2$  на вход интегратора подключают отрицательное опорное напряжение  $U_1= -U_{\text{опор}}$  и с момента  $t_2$  начинают подсчет времени.
- 3) В момент  $t_3$ , когда  $U_{\text{инт}}=0$ , прекращают подсчет времени и интервал  $\Delta t=t_3-t_2$  считают числовым эквивалентом измеряемого входного напряжения.  $\Delta t$  измеряют, как правило, внутренними таймерами – счетчиками. Момент  $t_3$  определяют по переключению  $U_k$  на выходе компаратора.



*Замечание:* интервал  $T=t_2-t_1$  стараются выбрать кратным периоду основной промышленной частоты (1/50 Гц). В этом случае интегрирование  $U_{\text{вх}}$  на интервале  $T$  автоматически исключит из сигнала помеху, обусловленную промышленной частотой.

Данный АЦП часто используют для измерения напряжения с удаленных датчиков, т.к. он легко исключает из сигнала промышленную частоту.

*Пример:* принципиальная схема программно-аппаратного АЦП двойного интегрирования.

$$P_{1.0} = \left\{ \begin{array}{l} 0, U_1 = U_{\text{вх}} \\ 1, U_{\text{опор}} = -U_{\text{опор}} \end{array} \right\} \quad P_{1.1} = \left\{ \begin{array}{l} 0, U_{\text{инт}} < 0 \\ 1, U_{\text{инт}} \geq 0 \end{array} \right\}$$

Получим основное выражение для АЦП двойного интегрирования:

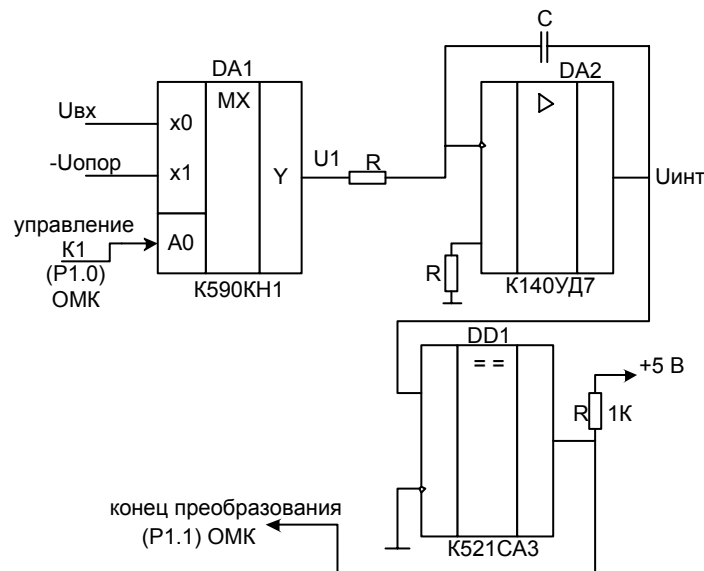
$$U_{\text{инт}}(t_2) = \frac{1}{RC} \int_{t_1}^{t_2} U_{\text{вх}} dt = \frac{U_{\text{вх}}}{RC} (t_2 - t_1) = U'_{\text{инт}}$$

$$U_{\text{инт}}(t_3) = U'_{\text{инт}} - \frac{1}{RC} \int_{t_2}^{t_3} U_{\text{опор}} dt = \frac{U_{\text{вх}}}{RC} T - \frac{U_{\text{опор}}}{RC} (t_3 - t_2) \equiv 0$$

$$\frac{U_{\text{вх}}}{RC} T = \frac{U_{\text{опор}}}{RC} \Delta t \quad \Delta t = \frac{U_{\text{вх}}}{U_{\text{опор}}} T \quad (2)$$

*Замечание:* Т.к. в формулу преобразования (2) для АЦП двойного интегрирования не входят параметры схемы (R, C, и т.д.), то следовательно эти параметры не влияют на точность преобразования. И их изменения от температуры или во времени никак не сказывается на погрешность измерения.

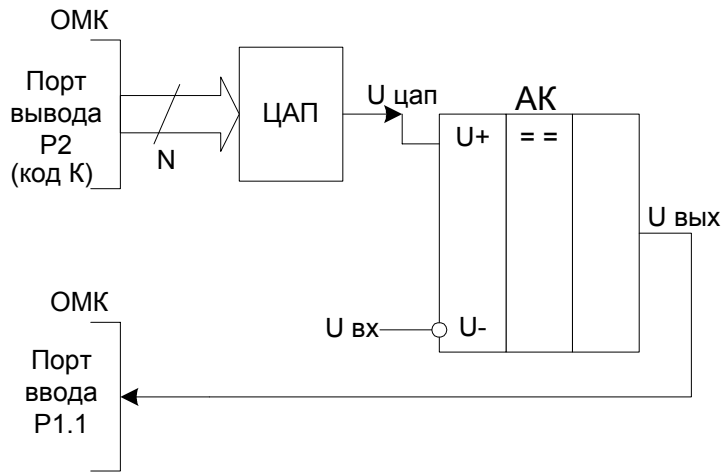
Т.к. это АЦП на компараторе сравнивается интеграл  $U_{\text{вх}}$ , то следовательно АЦП не чувствителен к случайным импульсным помехам.



Пример фрагмента программы управления АЦП двойного интегрирования. Для подсчета временных интервалов  $T=t_2-t_1$ ;  $\Delta t=t_3-t_2$  используют внутренний таймер T0 OMK BE51.

## 5.6. Программно–аппаратная реализация АЦП последовательного счета и последовательных приближений

Данный АЦП реализуется на основе ЦАП по следующей схеме:



АК–аналоговый компаратор.

Работа данного типа АЦП состоит в подборе такого двоичного кода К (порт P2), чтобы напряжение  $U_{\text{ЦАП}} = U_{\text{ВХ}}$ .

Существует два типа алгоритма подбора кода К:

- алгоритм последовательного счета;
- алгоритм последовательных приближений.

Алгоритм последовательного счета:

- 1) На цифровые входы ЦАП подается увеличивающийся на единицу n-разрядный двоичный код, при этом на выходе ЦАП формируется линейно нарастающее напряжение  $U_{\text{ЦАП}}$ .
- 2) При достижении  $U_{\text{ЦАП}}$  уровня входного напряжения  $U_{\text{ВХ}}$  происходит переключение аналогового компаратора (АК) и  $U_k = 0$ .

Двоичный код к выданной в этот момент на ЦАП и является тем числовым эквивалентом.

*Замечание:* недостатками АЦП последовательного счета является:

– большое время преобразования, так при 8-разрядном ЦАП требуется до 255 циклов суммирования К, а при 16-разрядном до  $2^{16} = 65536$ .

– невысокая помехозащищенность.

Для ускорения аналого-цифрового преобразования используют алгоритм последовательных приближений (поразрядного взвешивания).

Основные этапы этого алгоритма:

1) на цифровые входы ЦАП выдают двоичный код К с установленным в «1» только старшим разрядом (100...0). При этом на выходе ЦАП формируется напряжение:

$$U_{\text{ЦАП}} = \frac{U_{\text{ЦАП max}}}{2}, \text{ где } U_{\text{ЦАП}} - \text{напряжение, соответствующее коду}$$

111...1.

2) анализируется состояние АК, если переключение не произошло (т.е.  $U_{\text{ВХ}} < U_{\text{ЦАП}}$ ), то «1» в старшем разряде сохраняется в результате преобразования, иначе не сохраняется.



3) аналогичным образом устанавливаются в «1» следующий разряд ЦАП (более младший) и при этом сохраняются все подобранные «1» старших разрядов. Анализируют состояние АК. В случае его не переключения ( $U_{вх} < U_{цап}$ ) «1» в подбираемом разряде результата оставляют, иначе в разряд результата записывают «0».

4) аналогичным образом осуществляют подбор, все оставшиеся разряды ЦАП вплоть до младшего подобраны таким образом код  $K$  и является двоичным эквивалентом  $U_{вх}$ .

*Замечание:* Число итераций для подбора кода  $K$  по методу последовательных приближений равно разрядности ЦАП  $N$ .

Временная диаграмма работы АЦП последовательных приближений:

*Пример 2.* Фрагмент программы управления АЦП последовательных приближений:

*Замечание:* АЦП последовательного приближения также обладает низкой помехозащищенностью как и АЦП последовательного счета.

## 5.7. Понятия о АЦП с дельта–сигма–модуляцией

Данный тип АЦП подобен время-импульсному АЦП с двойным интегрированием.

Основной принцип работы его состоит в компенсации среднего входного тока АЦП с помощью внутреннего переключаемого источника тока  $I_0 = \text{const}$ . Общая схема АЦП с дельта –сигма -модуляцией имеет вид:

Основной принцип работы состоит в компенсации входного тока  $I_{вх}$  путем подключения источника тока  $I_0$ . Ток  $I_0$  подключается на интервал времени  $\Delta t$  достаточный для разряда с интегратора и компенсации заряжающего тока  $I_{вх}$ .

Число тактовых выходных импульсов  $TI$  подсчитанных за интервал времени  $\Delta t$  подключение  $I_0$  и определяет числовой код  $K$  соответствующий  $U_{вх}$ .

Подсчет же импульсов осуществляют с помощью счетчика, который сбрасывается через строго заданный интервал времени  $T$ .

Преимущества АЦП с дельта –сигма –модуляцией:

–высокая помехозащищенность, т.к. интегрируется измеряемое входное напряжение  $U_{вх}$  ( $I_{вх}$ )

–высокая точность преобразования, позволяющая строить многоразрядный АЦП (16, 20, 24-битные)

–независимость точности преобразования от изменения параметров схемы АЦП ( $R, C, \dots$ )

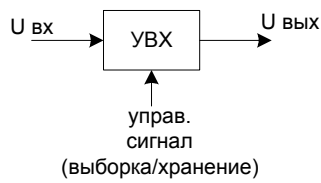
Недостаток:

–невысокая скорость преобразования.

## 5.8. Устройство выборки хранения (УВХ)

УВХ предназначен для кратковременного сохранения измеряемого напряжения неизменным, например, на время его А-Ц преобразования.

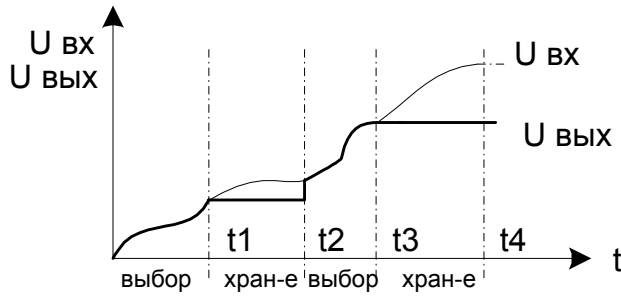
УВХ представляет собой аналоговую ячейку памяти, запоминаящую уровень (значение) измеряемого напряжения.



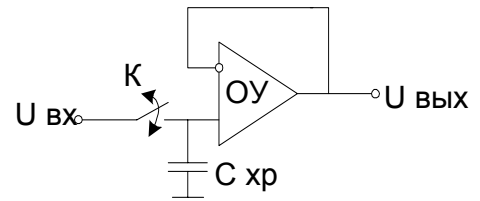
УВХ может работать в двух основных режимах:  
 –выборки, когда входное напряжение без изменения передается на выход УВХ ( $U_{\text{ВЫХ}}=U_{\text{ВХ}}$ );  
 –хранения, когда на выходе УВХ сохраняется напряжение, которое было на входе в момент перехода

в режим хранения.

Покажем диаграмму работы УВХ:



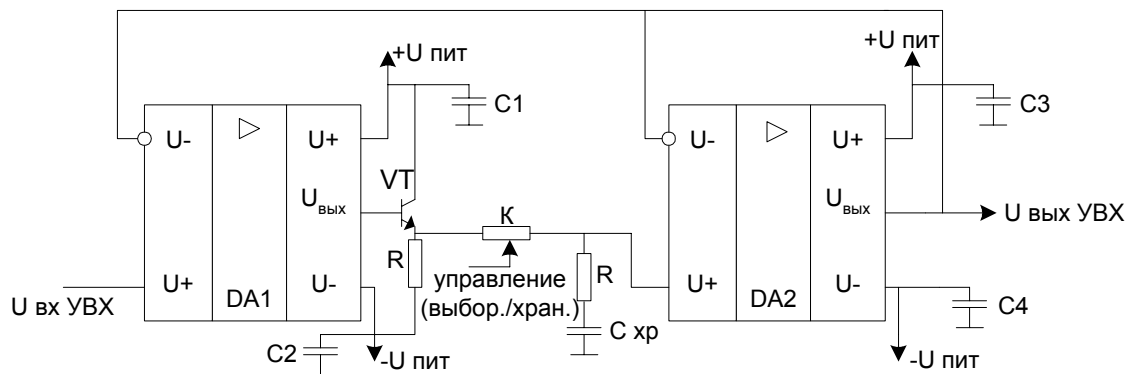
Простейшая реализация УВХ на основе ОУ имеет вид:



Основным элементом УВХ является  $C_{\text{ХР}}$ . Ключ К определяет режим работы УВХ:

- режим выборки -если ключ замкнут;
- режим хранения –если ключ разомкнут.

Рассмотрим принципиальную схему реализации УВХ на дискретных элементах:



DA1 – К140УД7 (УД20);

DA – К140УД8;

VT – КТ814; C1-C4 – 0,1 мкФ.

ОУ DA1 и DA2 охвачены единой ООС, поэтому коэффициент передачи УВХ=1.

Основным элементом хранения является  $C_{\text{ХР}}$ , ее значение выбирается в пределах  $C_{\text{ХР}} \sim 0.01 \div 0.1$  мкФ, что обеспечивает время хранения порядка

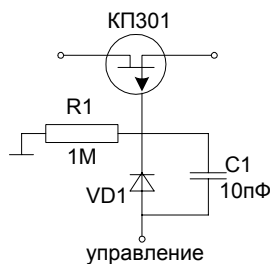
нескольких мс и минимальную задержку при переходе в режим выборки (время на заряд  $C_{ХР}$ ).

Для уменьшения переходных процессов колебательного типа при заряде  $C_{ХР}$  последовательно с ней включается малое балластное сопротивление  $R$  ( $R \sim 10 \div 100 \text{ Ом}$ ).

Для замедления разряда  $C_{ХР}$  ОУ DA2 выбирают с высоким входным сопротивлением (входные цепи на полевых транзисторах)

Емкости  $C1 \div C4$  представляют собой фильтры, исключающие передачу помех по питанию в УВХ.

Для ускорения заряда  $C_{ХР}$  ОУ DA1 снабжен мощным выходным каскадом на транзисторе VT (эмиттерный повторитель). Он обеспечивает достаточный ток для быстрого заряда  $C_{ХР}$ .



Ключ  $K$ , как правило, реализуют на полевых транзисторах по следующей схеме:

VD1—диод используется для запрещения отпириания р-п перехода полевого транзистора в прямом направлении;

$C1$ —ускоряющая емкость при отпириании и запириании транзистора;

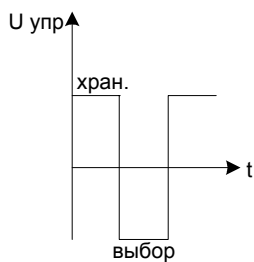
$R1$ — сопротивление обеспечивающее стекание паразитного заряда с затвора транзистора.

*Замечание:* в подавляющий большинстве устройств для реализации УВХ используют стандартные ИС типа:

K1100СК1—однополярный УВХ;

K1100СК2—двуполярный УВХ на дискретных элементах используют только в случаях если необходимо обеспечить продолжительное время хранения (порядка нескольких секунд) с одновременными быстрым зарядом емкости  $C_{ХР}$ .

Современные АЦП часто включают встроенный УВХ .



---

**Тема 7.- 18 часов. Особенности внутренней структуры и применения 16- и 32-разрядных ОМК в МУСУ.**

Общая характеристика 16- и 32-разрядных ОМК. Основные характеристики ОМК SAB80C167. Общая внутренняя структура ОМК SAB80C167. Организация памяти ОМК SAB80C167. Способы адресации данных. Центральный процессорный модуль ОМК SAB80C167. Особенности системы прерываний. Периферийный контроллер событий. Модули захват-сравнение. Встроенный начальный загрузчик. Модуль ШИМ. Контроллер промышленной CAN-сети.

---

## **6. Общие принципы использования 16-разрядных ОМК**

### **6.1. Общая характеристика 16-разрядных ОМК**

Впервые 16-разрядные ОМК были разработаны японскими фирмами: NEC, HITACHI и TOSHIBA в начале 80-х годов.

В 1982 г. Intel разработала первое 16-разрядное семейство ОМК. (MCS-96)

Основные общие черты 16-разрядных ОМК:

- использование высокопроизводительного 16- или 32-разрядного процессорного модуля (быстродействие которого  $10 \div 50$  MIPS)
- использование внутри 16-разрядной системной магистрали (16-разрядная ШД)
- использование широкой номенклатуры специализированных встроенных устройств обеспечивающих значительное ускорение обработки цифровой информации.

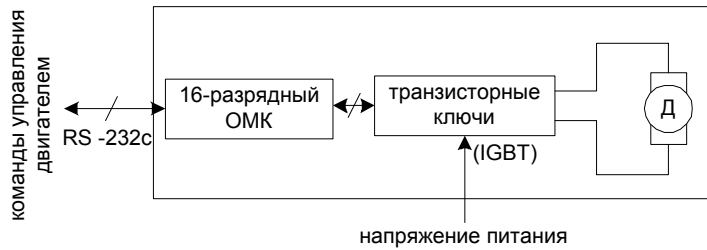
Основные области применения 16-разрядных ОМК:

- распределенные МПС управления станками с ЧПУ, роботами – манипуляторами и технологическими линиями;
- МПС управления транспортными объектами;
- МПС передачи данных, телекоммуникаций и связей.

С появлением 16-разрядных ОМК связывают и появление термина « мехатроника ».

Мехатроника – это конструктивное объединение на принципиально новом уровне механических устройств и их электронных СУ. При таком объединении обеспечивается минимальное число коммутационных связей В СУ, высокая гибкость управления, а следовательно и высокая надежность.

Например, схема мехатронного двигателя:



Подобные электро–механические устройства часто называют интеллектуальными электро–механическими устройствами, а 16-разрядный ОМК используемый в этих устройствах часто называют типа “motor controls”.

В нашей стране наиболее широко распространены следующие типы 16-разрядных ОМК:

- семейство C16xx (Siemens, Infinion), например, SAB80C161÷C167;
- семейство MCS-96, 196, 296 (Intel);
- семейство HC12, HC16 (motorola);

*Замечание:* в данном курсе подробно будем рассматривать наиболее типичного представителя 16-разрядного ОМК SAB80C167.

## 6.2. Основные технические характеристики ОМК SAB80C167

Процессорный модуль –16-разрядный, обеспечивающий быстроедействие 10 MIPS при тактовой частоте 20МГц.

Имеется встроенный сопроцессор операций умножения и деления, а также битовый процессор для ускорения операций с битами.

Время выполнения операций умножения и деления составляет 0,5-1мкс.

- 1) 16-разрядный модуль реализован на основе RISC-архитектуры с сокращенным набором команд.
- 2) Объем адресуемой памяти 16М (реализована 24-разрядная шина адреса).

Память построена по фон-неймановской архитектуре, т.е. в едином адресном размещается как данные, так и программы (не существует раздельной ПП и ПД).

Имеется встроенное ОЗУ –2К и встроенное ПЗУ –32К (типа PROM)

- 3) Реализован плавающий регистровый банк (регистровый файл). Регистровый банк содержит 16 двухбайтных регистров R0...R15. Регистровый банк может размещаться в любом месте внутреннего ОЗУ. Его местоположение задается программно.

Т.о. в ОЗУ может быть реализовано 64 независимых не пересекающихся регистровых банка.

4) Система команд содержит более 300 базовых команд. Все команды контроллера либо двух-, либо четырех- байтные. Большинство команд выполняются за 2 МЦ. Исключение составляют лишь команды переходов умножения и деления. Система команд нацелена на реализацию высокоэффективных вычислительных алгоритмов и имеет несколько модификаций команд умножения и деления.

5) Реализовано программно – настраиваемая внешняя системная магистраль. Программно можно задавать различные режимы работы ША и ШД:

–ШД может быть 8- или 16-разрядной;

–ША/ШД может быть совмещенная (мультиплексированная), либо отдельные шины ША и ШД.

*Замечание:* В системе, реализованной на ОМК С167 может одновременно присутствовать несколько системных магистралей, причем к каждой из магистралей может быть подключены свои устройства.

Имеется встроенный селектор адреса для пяти внешних устройств, генерирующий сигналы выбора CS<sub>0</sub>,...,CS<sub>4</sub>. (при этом используется программно – задаваемая полная 24-разрядная дешифрация адреса)

В контроллере для обращения к внутренней ОЗУ и некоторым встроенным устройствам используется режим X - шины.

*Замечание:* Режим X – шины позволяет внешним устройствам, подключенным к ОМК обращаться через системную магистраль к внутренним элементам (ОЗУ, CAN, ...) так, как будто эти элементы находятся вне контроллера и подключены к внешней системной магистрали.

6) Реализован встроенный 8-канальный контроллер событий (PEC – Periferial Events Controller).

Контроллер событий предназначен для ускоренной обработки прерываний, вызванных обменом данными между внешними устройствами и памятью.

При обработке этих прерываний п/п обработки не вызывается, а весь обмен данными обеспечивается аппаратно контроллером событий за время равное 1 МЦ.

7) Система прерываний контроллера позволяет обслуживать 57 внешних и внутренних источников, причем каждому из них может быть присвоен 1 из 16 уровней приоритета.

8) Реализовано пять 16-разрядных встроенных таймеров счетчика, которые организованы в двух модулях таймера:

–модуль GPT1 (T2, T3, T4)

–GPT2 (T5, T6)

Каждый из модулей представляет собой многофункциональный таймерный блок, таймеры которые могут работать как совместно, так и раздельно. Таймерные модули обеспечивают:

- подсчет времени;
- подсчет импульсов;
- определение длительности импульсов;
- осуществление, как деления, так и умножение частоты и др.

9) Реализованы два 16-разрядных модуля “захват - сравнение” (CAPCOM1 и CAPCOM2)

Данные модули предназначены для высокоточной фиксации или генерации сигнала во времени. Модуль CAPCOM1 работает совместно с двумя таймерами (T0 и T1), а CAPCOM2 – T7 и T8.

10) Реализован встроенный модуль ШИМ (PWM).

Данный модуль предназначен для управления мощными устройствами (двигателя) с использованием внешних транзисторных ключевых схем (не ниже пяти транзисторов, как правило, IGBT -транзисторы)

Например, возможна реализация 3-х фазного управления асинхронным двигателем с различной частотой.

Модуль ШИМ имеет 4 канала для одновременного управления четырьмя транзисторными ключами на основе широтной модуляции. Максимальная частота изменения ШИМ 74 КГц.

11) Реализован встроенный модуль АЦП (10-разрядный) с 16-канальным аналоговым коммутатором и встроенной схемой выборки – хранения. Время преобразования АЦП порядка ~9мкс.

12) Реализованы два встроенных последовательных интерфейса:

–синхро – асинхронный, типа RS232 (ASC0) (подобный ОМК BE51). Максимальная скорость передачи данных 625 Кбод.

–высокоскоростной синхронный последовательный интерфейс (SSC)-подобный интерфейсу SPI и обеспечивающий передачу данных со скоростью 5 Мбод. Данный интерфейс обеспечивает передачу по трем линиям:

- SCLK;
- MTRSR;

–MRST и может использоваться как дополнительная последовательная системная магистраль для подключения ОЗУ, ПЗУ, АЦП.

13) Реализован встроенный контроллер промышленной CAN –сети, обеспечивающий одновременное сопряжение в единую сеть до 100 различных ОМК. Скорость передачи порядка 1Мбод. Все протоколы CAN –сети, а также алгоритм обмена реализуется контроллером на аппаратном уровне.

14) Реализовано 111 линий для организации портов ввода/вывода. Все они сформированы в виде 9 портов P0÷P8.

15) Реализован встроенный модуль начальной загрузки контроллера (Bootstrap Loader). Загрузчик обеспечивает при включении питания загрузку в память ОС или иной управляющей программы. Загрузка осуществляется через последовательный интерфейс ASC0. Т.о. системы построенные на основе ОМК С167 могут работать вообще без ПЗУ, а управляющие программы будут грузиться в ОЗУ и другие модули.

### 6.3. Внутренняя структура ОМК SAB80C167

В состав контроллера С167 входит 9 портов ввода/вывода P0÷P8 каждый из перечисленных портов может работать в двух основных режимах:

- как обычный порт ввода/вывода дискретных сигналов, программируемый пользователем;
- как порт для выдачи управляемых сигналов основных блоков ОМК (альтернативный режим)

В первом случае использования портов их линии могут быть запрограммированы:

- как обычные ТТЛ входы и выходы (при вводе/выводе)
- как выходы с отрицательным коллектором

*Замечание:* в режиме с открытым коллектором могут работать порты P2, P2, P6, P7, P8.

Остановимся подробнее на альтернативных функциях портов P0÷P8.

### 6.4. Организация памяти ОМК С167

Память организована с помощью Фон - Неймановской архитектуры, т.е. в едином адресном пространстве 16 Мбайт размещается:

- модули внутренней и внешней ОЗУ;
- внутренний и внешний модули ПЗУ;
- регистры специальных функций (РСФ) (SFR, ESFR) предназначены для управления всеми встроенными модулями ОМК;
- регистры данных и управления всех подключенных внешних устройств.

Все единое адресное пространство объемом 16 Мбайт разбито на 256 сегментов, объем каждого из них 64 Кб.

Каждый сегмент в свою очередь разбит на 4 страницы, с объемом 16 Кб каждая (1024 стр.). Разбивка памяти на сегменты и страницы вызвана малой разрядностью процессорного модуля (16 бит) по сравнению с 24-разрядной ША.

Построим карту памяти контроллера С167. Она определяет распределение ОЗУ, ПЗУ и регистра устройств по всему адресному пространству:



SFR, ESFR–регистры специальных функций для управления всеми внутренними ОМК.

XRAM–расширение внутреннего ОЗУ , имеется у отдельных контроллеров модели C167.

CAN–регистры управления встроенного контроллера CAN-сети.

Вся внутренняя память ОМК C167 располагается в системном сегменте памяти. При этом:

–внутреннее ПЗУ объемом 32К размещается в ячейках 0'0000÷0'8000;

–внутреннее ОЗУ объемом 32К располагается вместе с управляющими регистрами SFR и ESFR в области с адресами: 0'F000H÷0'FFFFH.

Внутреннее ПЗУ программируемым способом может переключаться с начала ячеек нулевого сегмента в начало ячеек первого сегмента, при этом в нулевом сегменте с начальных ячеек может располагаться внешнее ПЗУ (удобно при отладке программы).

Рассмотрим подробнее организацию внутреннего ОЗУ контроллера. Посмотрим карту ОЗУ:

Внутреннее ОЗУ размещается в третьей странице памяти, при этом назначения областей ОЗУ следующие:

–область с адресами 0'F600H÷0'FA00H используются для расположения плавающего регистрового банка, состоящего из 16 регистров и переменных пользователя;

–область 0'FA00H÷0'FBFEN используется для расположения системного стека, размер которого может программно задаваться (по умолчанию 256 слов).

–область 0'FCE0H÷0'FCFEN отведена для расположения регистров указателей контроллера событий PES.

–область 0'FCFEN÷0'FE00H используется для расположения прямоадресуемых бит, каждый бит в этой области имеет свой адрес и к нему возможно прямое обращение.

*Замечание:* заштрихованные области регистров SFR ESFR также допускают прямую адресацию бит.

## 6.5. Системный стек ОМК C167

Располагается в ОЗУ с 0'FA00H и может иметь размер от 32 до 1024 байт. Размер стека задается программно через регистр специальных функций FSR с именем SYSCON. Помимо указанного управляющего регистра имеется два регистра специальных функций SFR, управляющих стеком:

STKOV–определяет верхнюю границу стека (в нем хранится адрес последней ячейки стека);

STKUN–определяет начало стека (в нем хранится адрес последней ячейки стека).

При выходе стека за границы, указанные в STKOV и STKUN генерируется прерывания (ошибка стека). Выход за границу STKOV – переполнение стека, за STKUN – антипереполнение.

Запись и чтение информации в стек контролируется регистром SP (Stack Pointer). Этот регистр всегда содержит адрес текущей вершины стека.

При записи информации в стек он растет от старших адресов в ОЗУ к младшим. В стек возможна запись только слов.

При записи информации типа «слово», оно располагается в памяти следующим образом:

- младший байт слова всегда записывается по четному адресу N;
- старший байт слова записывается по нечетному адресу N+1.

*Замечание:* все команды C167 всегда располагается в памяти только по четным адресам (иначе будет генерироваться ошибка).

Для обращения к данным, хранящимся в памяти в ОМК C167 может использоваться различные способы адресации.

## 6.6. Способы адресации данных в ОМК C167

ОМК C167 использует три основных способа адресации данных:

–короткая адресация, при этом указывается 8-разрядный (короткий) адрес в команде;

–длинная адресация, если в команде указывается 16-разрядный адрес ячейки или УВВ.

–косвенная адресация, если 16-разрядная ячейка памяти или УВВ содержится в данном из регистров процессора.

Короткая адресация используется для обращения к регистрам общего назначения (РОН) R0,...,R15; к регистрам ESFR и SFR; а также к ячейкам, находящимся в области прямоадресуемых бит.

Короткий адрес (КА) представляет собой 8-разрядное смещение относительно определенного базового 24-разрядного адреса (БА). Адрес ячейки 24-разрядный вычисляется по короткому адресу по следующей формуле:

$$\Phi A = BA + \Delta \cdot KA, \quad \Delta = \begin{cases} 1, & \text{если адресуется байт} \\ 0, & \text{если адресуется слово} \end{cases}$$

В случае КА к РОН предыдущая формула приобретет вид:  $\Phi A = CP + \Delta \cdot KA$

CP –контекстный регистр, в нем всегда хранится 24-разрядный адрес начала плавающего банка ОЗУ. (CP=0'F600H÷0'FA00H)

При короткой адресации с регистром SFR формула определения фактического адреса имеет вид:  $\Phi A = 0'FE00H + \Delta \cdot KA$

KA –8-разрядный адрес конкретного регистра SFR (00H÷0FFH)/

В случае короткой адресации к расширенным управляющим регистрам ESFR формула приобретает вид:  $\Phi A = 0'F000H + \Delta \cdot KA$

*Пример:* коротких адресаций:

MOV R0, R1 ; R0←R1

MOV SP, #2100H ; SP←2100H

Иначе MOV 0E7H, #2100H ; SP←2100H

Длинная адресация используется для обращения к ячейке памяти и регистру, находящемуся в любом месте адресного пространства 16Мбайт.

При длинной адресации используется 16-разрядный адрес ячейки памяти или регистра, а фактически 24-разрядный адрес, вычисляется специальным способом.

Выделяют два типа длинной адресации:

*Пример:* длинной адресации с использованием EXTS:

EXTS #10H, 1 ; устанавливает текущий сегмент с номером 10H, на следующую команду

MOV R1, 4225H

*Замечание:* в случае сегментной адресации ФА (24-разряд.) вычисляется просто, он равен:

Seg'(16-разрядный адрес), т.е. перед 16-разрядным адресом достаточно дописать номер сегмента, в рассмотренном примере: ФА=104225H (24-разрядный)

Косвенная адресация также использует 16-разрядный адрес ячейки памяти и регистра, но он указывается не числом в команде, а регистр, в котором он находится.

## 6.7. Общая структура процессорного модуля C167 (CPU C167)

Ядро процессорного модуля включает:

- 16-разрядное АЛУ;
- аппаратный модуль умножения/деления (MUL/DIV) со своими регистрами: MDC, MDH, MDL;
- аппаратный сдвигатель –для ускоренной реализации операции сдвига;
- битовый процессор для скоростной обработки бит;
- четырёх этапный конвейер команд, позволяющий одновременно осуществлять выборку до четырех команд из памяти и обработку АЛУ.

Процессорный модуль также включает банк РОН R0,..., R15, однако аппаратно банк располагается во внутреннем ОЗУ в виде плавающего регистрового банка.

Поясним назначение основных управляющих регистров CPU:

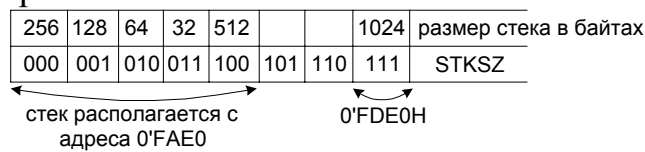
- SP, STKOV, STKUN –регистры управления стеком;
- CSP (SFR:04H)-(Code Segment Pointer) указатель текущего сегмента выполняемой программы;
- IP (SFR:... (недоступен, только косвенно))-(Instruction Pointer) указатель текущей выполняемой команды в сегменте CSP; в IP всегда содержится адрес(смещения) текущей выполняемой команды (0÷65535).

*Замечание:* полный физический 24-разрядный адрес текущей выполняемой команды всегда равен:

CSP : IP



–STKSZ – поле бит, определяющее размер стека ОМК реализованного во внутреннем ОЗУ.



–ROMS1 –бит, переключающий внутреннее ПЗУ из нулевого сегмента в первый, это делается, чтобы с начала нулевого адреса расположить внешнее ПЗУ при отладке. Если ROMS1=1, то начальный адрес внутреннего ПЗУ меняется: 0'0000H→1'0000H.

–SGMDIS –бит, определяет формат адреса возврата, который будет сохраняться в стеке при вызове п/п (в том числе п/п прерываний)

Если SGMDIS=0 в стеке сохраняется полный 24-разрядный адрес→CSP:IP

Если SGMDIS=1 в стеке сохраняется только IP;

–ROMEN –разрешение работы с внешним ПЗУ. При сбросе контроллера бит ROMEN дублирует состояние вывода EA контроллера;

–BYTDIS –запрещение передачи по 16-разрядной ШД D15...D0 отдельных байт. Если BYTDIS=0, то на разряд P3.12 ОМК выдается сигнал VHE (разрешение передачи по ШД старшего байта).

## 6.8. Особенности системы прерываний ОМК SAB80C167

Архитектура C167 поддерживает несколько механизмов обработки прерываний:

- нормальная обработка прерываний.
- обработка прерывания с использованием периферийного контроллера событий (PEC-Peripheral Events Controller).
- прерывания-ловушки (trap), генерируется в ответ на ошибку возникшую при выполнении программы или по линии NMI

– ускоренная обработка внешних прерываний:

С каждым источником прерывания связаны:

–регистр управления прерывания (xxIC)

–вектор прерывания (IV)

Регистр управления прерыванием (IC) содержит:

–биты определения уровня приоритета (0...15) ILVL

–бит запроса прерывания (xxIR)

–бит разрешения/запрещения прерывания (xxIE)

В ОМК реализуется векторная система прерываний.

С каждым вектором прерывания (IV) связана область в нулевом сегменте памяти, состоящие из четырех байт: Адрес области = 4·IV

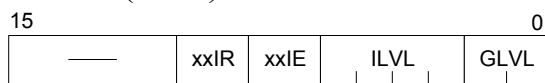
При возникновении прерывания осуществляется переход на эту область, где хранится команда безусловного перехода на п/п обработки.

*Замечание:* Со всеми прерываниями-ловушками связан один вектор прерывания (ОАН), для определения возникшей ошибки используется специальный регистр флагов-ловушек (TFR – Trap Flag Register)

В ОМК имеется 8 входов быстрой обработки прерываний (P2.8÷P2.15). Если обычные входы запросов прерываний опрашиваются каждые 400 нс (20 МГц), то входы быстрой обработки каждые 50 нс (20 МГц). Эти внешние входы могут программно задавать признак прерываний. Управляются они через специальный регистр EXICON (0E0H).

Прерывания – ловушки (типа trap) генерируется ОМК при возникновении ошибочной ситуации или поступление запроса прерывания на вход NMI (немаскируемое прерывание).

С каждым источником прерывания связан регистр управления прерыванием (xxIC). Все они имеют одинаковую структуру:



xxIR–бит запроса прерывания (1-если этот источник выставил запрос)

xxIE–бит разрешения/запрещения данного прерывания (0-запрещено)

ILVL–уровень приоритета прерывания данного источника (0-низший; 15-высший)

GLVL–групповой уровень приоритета (0-низший; 3-высший) анализируется при одновременном (групповом) поступлении запросов с одинаковыми уровнями ILVL.

Биты регистра xxIC устанавливаются программно пользователем.

При поступлении запроса от устройства ОМК читает его уровень приоритета ILVL и сравнивает с приоритетом текущей программы (поле ILVL в PSW). Если приоритет поступившего прерывания выше оно обрабатывается, иначе – игнорируется.

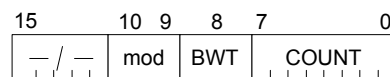
*Замечание:* всем прерываниям типа trap соответствует один и тот же вектор прерывания (ОАН), т.е. при их возникновении вызывается одна п/п обработки и для определения типа возникшей ошибки ОМК используется регистр TFR (trap Flag Register).

## 6.9. Периферийный контроллер событий (PES)

PES имеет восемь независимых каналов обработки прерываний. Каждый из каналов позволяет обрабатывать свой запрос прерывания на аппаратном уровне.

При обработке прерываний PES осуществляет передачу 1байта (слово) между двумя устройствами или памятью. Каждому каналу контроллера событий PES соответствует по два регистра: SRCPx (x=0...7) – это регистр для хранения адреса устройства источника информации; DSTPx – регистр для хранения адреса устройства приемника информации.

Кроме того каждый канал PES имеет свой регистр управления PESCx (x=0...7). Его формат:



COUNT –счетчик реализованных обменов данными (битами, словами) по каналу PEC.

Счетчик COUNT вычитаемый и при его обнулении генерируется обычное прерывание для обработки полученного массива байт (слов).

BWT–определяет тип передаваемой информации (байт или слов);

mod–определяет режимы модификации адресов в регистре SRCPx и DSTPx после каждого обмена. Mod=00 – адреса не модифицируются; mod=01 – после каждого обмена SRCPx+1(2) – увеличивается на 1 или 2 (режим используется при передачи массива из памяти); mod=10 – после каждого обмена DSTPx+1(2). Используется для сохранения массива информации в памяти; mod=11 – не используется.

*Замечание:* PEC удобно использовать:

–при чтении данных из АЦП В память;

–при реализации обмена данными между памятью и последовательными интерфейсами.

## 6.10. Модули захват-сравнение ОМК С167

В ОМК реализованы два модуля захват-сравнение (CAPCOM – CAPTURE/COMPARE): CAPCOM1 и CAPCOM2. Они позволяют реализовать до 32 каналов захват-сравнение, каждый из которых может работать с 4 таймерами: T0, T1, T7, T8.

Каждый канал захват-сравнение связан со своим выводом ОМК ССxIO (x=0,...,31).

Для реализации выводов ССxIO используются выводы портов P1, P2, P7, P8 в альтернативном режиме работы.

Каждый из каналов захват-сравнение может работать в двух основных режимах:

–режим захвата.

–режим сравнения.

Каналы CAPCOM наиболее часто используют:

–для генерации сигналов широтно-импульсной модуляции (ШИМ) высокоточные во времени;

–для генерации различных конфигураций логических сигналов;

–для точной фиксации во времени внешних событий.

–для формирования до 32 программных таймер - счетчиков.

Разрешающая способность (точность) CAPCOM ~400нс (20МГц)

Общая структура модулей CAPCOM1 (2) имеет вид:

Каждый модуль CAOCOM содержит два 16-разрядных таймера:

–CAPCOM1: T0/T1

–CAPCOM2: T7/T8

Таймеры в свою очередь управляются регистрами управления T01CON, T78CON и каждый имеет регистр автоперезагрузки T0REL,...,T8REL. Синхронизация работы счетчиков – таймеров может осуществляться, как от внешних источников, так и от частоты синхронизации ОМК (с предварительным делением ее с помощью таймера T6)

## 6.11. Встроенный начальный загрузчик ПО или ОС ОМК С167

Встроенный загрузчик, позволяет загрузить в память ОМК по сигналу сброс, через последовательный интерфейс программу (таблицу кодов).

Загрузчик передает данные получаемые из интерфейса ASCO во внутреннем ОЗУ ОМК. После передачи осуществляется переход на загружаемую программу.

*Замечание:* Если полученная программа может осуществлять прием данных из последовательного интерфейса и перезаписывает их во внешнее ОЗУ. Т.о. в системах с ОМК С167 можно работать вообще без ПЗУ.

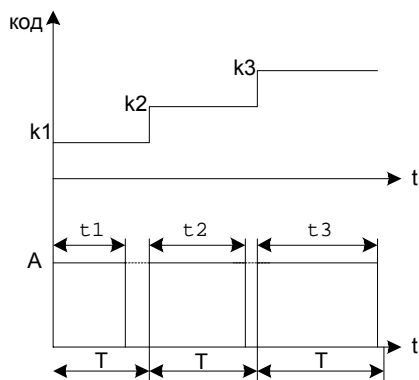
*Замечание:* Этот механизм загрузки широко применяют в отладочных системах, а также для записи информации во FLASH –память или для загрузки ОС в ОМК.

## 6.12. Встроенный модуль ШИМ ОМК С167

В настоящее время во многих СУ мощными устройствами (двигателя, электромагнитные исполнительные устройства, печи) находит распространение принципа кода в аналоговую величину посредством ШИМ.

*Замечание:* Преобразование кода в величину напряжения (тока) с помощью АЦП, реализует АИМ.

ШИМ – широтно-импульсная модуляция (Pulse Width Modulation)– PWM, отличается от АИМ, тем что преобразование не в величину напряжения, а в длительность импульса.



$$\tau_1 = \alpha \cdot K_1$$

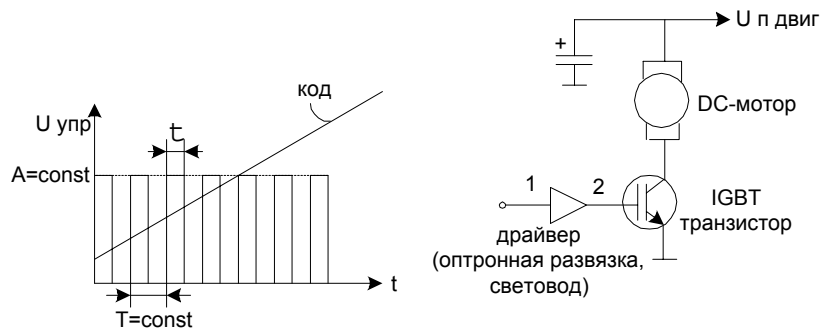
$$\tau_2 = \alpha \cdot K_2$$

$$\tau_3 = \alpha \cdot K_3$$

$T = \text{const}$  – период следования импульсов.

В простейшем случае схема Управления двигателем постоянного тока с помощью ШИМ:





В более сложных случаях управления АС –двигателями, вентильными или индукторными используют мостовые схемы включения транзисторов ( тиристоров ), инверторы напряжения.

Генерация импульсов управления транзисторами осуществляется с помощью модулей ШИМ ОМК.

ШИМ ОМК позволяет генерировать четыре независимые последовательности импульсов (4-канала ШИМ). Размещающая способность изменения  $\tau$ :  $\Delta\tau=50\text{нс}$ .

Частота следования импульсов ШИМ: от 2,4 Гц до 10(5) МГц.

Существует четыре способа формирования ШИМ:

- генерация стандартного ШИМ;
- генерация симметричного ШИМ;
- генерация пакетов импульсов;
- режим одновибратора

## 6.13. Встроенный контроллер промышленной CAN-сети (CAN-модуль)

Контроллер CAN-сети позволяет связать посредством шины (витая пара, коаксиальный кабель, световод) ряд станций (контроллеров, ЭВМ). Протокол взаимодействия станций по шине реализуется при этом на аппаратном уровне.

CAN-модуль позволяет, автоматически от центрального процессора, осуществлять прием и передачу CAN-сообщений в соответствии со стандартом CAN V2.0B.

Передаваемые сообщения делятся на :

- стандартные, с 11-разрядным идентификатором
- расширенные, с 29- разрядным идентификатором

Каждое сообщение содержит 8 байт данных. Одновременно CAN-модуль может передавать и принимать до 15 сообщений (15 каналов связи)

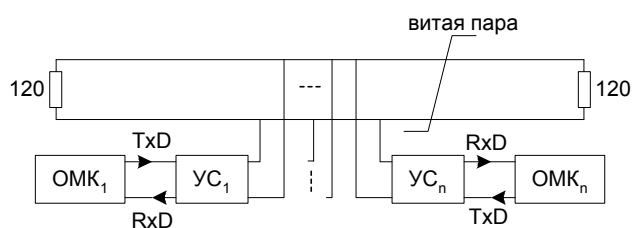
При приеме/передаче сообщений CAN-модуль обеспечивает:

- скорость передачи до 1 Мб/с
- несколько средств контроля ошибок, в том числе циклическое кодирование с исправлением до 5 ошибок в сообщении.

–при обнаружении большого числа ошибок в принятом сообщении автоматически CAN-модуль формирует повторный запрос. Это

обеспечивается аппаратно, пока сообщение не будет принято верно, либо не переполнится счетчик ошибок (128).

Схема подключения ОМК С167 с CAN-сетью.



УС1,...,УСn-устройства согласования с сетью (например можно использовать драйвер интерфейса RS-485, MAX485)