

А.М. БАКШАЕВ

***ПРАКТИКУМ
ПО ИЗУЧЕНИЮ ПЕРИФЕРИЙНЫХ
БИС***

УЧЕБНОЕ ПОСОБИЕ

Киров, 2005

Министерство науки и образования
Российской Федерации
Вятский государственный университет

А.М. БАКШАЕВ

**ПРАКТИКУМ
ПО ИЗУЧЕНИЮ ПЕРИФЕРИЙНЫХ
БИС**

УЧЕБНОЕ ПОСОБИЕ

Утверждено
Ученым советом ВятГУ
в качестве учебного
пособия

Киров 2005

ББК 32.973.26-04

УДК 681.322-181.48

Бакшаев А. М. Практикум по изучению периферийных БИС: Учебное пособие - Киров: ВятГУ, 2005.- 150 с.

Приведены сведения об организации системы ввода-вывода микроЭВМ, составе периферийных БИС микропроцессорных комплектов и принципах построения микропроцессорных систем. Представлено описание структур, алгоритмов работы и методики программирования наиболее часто используемых периферийных БИС микропроцессорных комплектов серий К1804, К580 и К1810: блока векторных приоритетных прерываний К1804ВН1, программируемого контроллера прерываний К1810ВН59А, программируемого интервального таймера К580ВИ54 и программируемого контроллера клавиатуры и индикации К580ВВ79. Даны области и примеры применения БИС в составе микропроцессорной системы.

Пособие предназначено для студентов специальности 220100 "Вычислительные машины, комплексы, системы и сети".

Работа подготовлена на кафедре электронных вычислительных машин Вятского государственного университета.

Рис. 57, табл. 17, библиогр. 13 назв.

Рецензенты: Кафедра информатики Вятского государственного гуманитарного университета, г. Киров;

Научно-исследовательский институт средств вычислительной техники, г. Киров;

Чарушин Н.А. - к.т.н., директор государственного предприятия лаборатории автоматики и вычислительной техники.

© Бакшаев А.М., 2005

Содержание:

1. Организация подсистемы ввода-вывода	4
1.1 Организация программно-управляемого обмена	4
1.2 Организация подсистемы прерываний	9
1.3 Периферийные БИС	11
2 Обработка прерываний на основе МПК K1804	14
2.1 Блок векторных приоритетных прерываний K1804BH1	14
2.2 Система микрокоманд БВПП K1804BH1	22
2.3 Применение БВПП BH1 в составе МПК K1804	26
2.4 Структура процессора с системой прерываний	34
3 Обработка прерываний в МПК K580 и K1810	37
3.1 Программируемый контроллер прерываний K1810BH59A	37
3.2 Программирование ПКП	45
3.2.1 Режим фиксированных приоритетов	52
3.2.2 Циклический сдвиг приоритетов	55
3.2.3 Режим специального маскирования	59
3.2.4 Режим последовательного опроса	60
3.2.5 Режим чтения регистров ПКП	62
3.3 Каскадное включение БИС ПКП	63
3.4 Влияние помех на входах IRQ в режиме UP=1 (IRQ по уровню)	68
4 Организация временных интервалов в микроЭВМ	70
4.1 Программируемый таймер K580BI54	70
4.2 Программирование таймера и режимы работы	73
4.3. Режимы работы таймера	75
4.4 Применение ПТ BI54 для организации общесистемных средств счета времени	84
5 Организация ввода и отображения данных в МПС	88
5.1. Программируемый контроллер клавиатуры и индикации K580BV79	92
5.2 Программирование и режимы работы БИС ПККИ K580BV79	100
5.3 Работа ПККИ K580BV79	111
5.4 Применение ПККИ K580BV79	122
6 Проектирование микропроцессорных систем на основе однокристальных микропроцессоров	131
6.1 Базовые конфигурации микропроцессора VM86	131
6.2 Цикл шины центрального процессора VM86	136
Литература	140

1. Организация подсистемы ввода-вывода

1.1 Организация программно-управляемого обмена

Подсистема ввода-вывода (ВВ) является одной из основных компонент любой вычислительной системы и обеспечивает обмен информацией между центральным процессором (ЦП), оперативной памятью (ОП) и периферийными устройствами (ПУ), обеспечивая взаимосвязь ЭВМ с “внешней средой”.

Основной проблемой при организации ВВ является большое разнообразие типов устройств ввода-вывода и их основных характеристик, которые могут различаться:

- ◆ по скорости передачи информации;
- ◆ по формату данных;
- ◆ по структуре управляющих сигналов, задающих алгоритм обмена и обеспечивающих работу ПУ;
- ◆ по уровням электрических сигналов.

С точки зрения программиста, работающего на командном уровне, систему ВВ можно представить в виде набора ПУ и ряда команд ВВ, обеспечивающих доступ к этим ПУ. Пространство ввода-вывода обычно организовано в виде набора регистров или n -разрядных ячеек - портов, каждый из которых может быть адресован независимо от других.

Процессор поддерживает два способа адресации устройств ВВ:

- ◆ с отображением на отдельное адресное пространство ВВ с помощью команд ВВ IN и OUT;
- ◆ с отображением через пространство памяти, при этом портами ВВ считаются адреса памяти и для доступа к портам применяется система команд общего назначения, что позволяет “расширить” систему команд для операций ВВ без введения дополнительных команд (допускает адреса до 4 Гбайт физической памяти).

а) ввод-вывод с использованием команд IN и OUT:



IN AX, 40h ; MOV DX, 400h ;
 IN AX, DX ;

OUT 40h, AX ; MOV DX, 400h ;
 OUT DX, AX ;

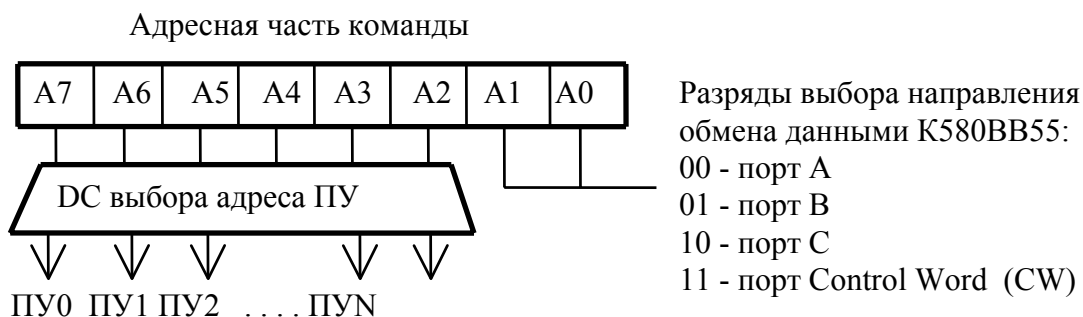
б) ввод-вывод с использованием адресного пространства микроЭВМ:



MOV AX, Addr ; MOV Addr, AX ;

Рисунок 1.1 - Распределение памяти микроЭВМ при различных способах адресации к периферийным устройствам

а) ввод-вывод с использованием команд IN и OUT:



б) ввод-вывод с использованием адресного пространства микроЭВМ:

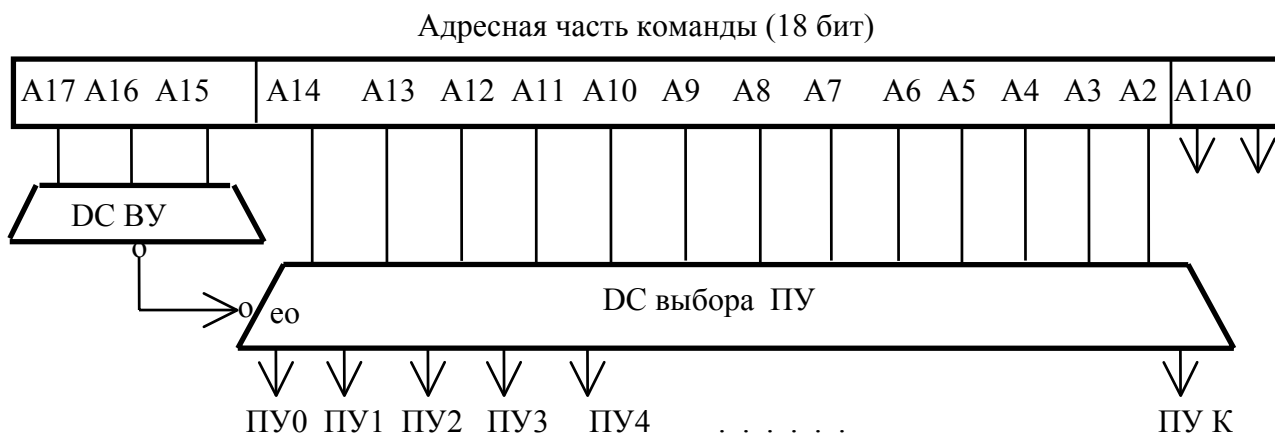


Рисунок 1.2 - Форматы адресной части команд ВВ для выбора номера ПУ

Системы с отображением на адресное пространство ВВ имеют в свою очередь два способа адресации:

- ◆ прямая адресация, которая допускает адреса в диапазоне от 0 до 255;
- ◆ регистровая косвенная адресация, выполняется с привлечением регистра DX, и допускает адреса портов ВВ в диапазоне от 0 до 65535.

На рисунке 1.1 показано распределение памяти при использовании команд IN и OUT (а) и общего адресного пространства микроЭВМ (б), где последние 8 Кбайт оперативной памяти (ОП) отводятся под адреса портов ВВ, а на рисунке 1.2 показаны схемы дешифрации адресной части команды для определения адресов портов ВВ для рассмотренных способов адресации.

Между микропроцессором (МПр) и ПУ происходит обмен информацией двух типов: служебной (управляющей) и собственно данных. К служебной информации относятся команды управления ВВ (управляющие слова (CW)) и слово состояния ПУ (PSW), информирующее о его текущем состоянии. Как правило, для связи с ПУ отводится несколько портов, например, порт ввода, порт вывода, порт служебной информации.

Различные ПУ могут иметь свои форматы данных и способы обмена, однако существуют общие правила обмена информацией между конкретными ПУ и МПр, называемые протоколом обмена. Совокупность этих правил составляет основу для написания драйвера ПУ, представляющего набор программ, организующих взаимодействие с ПУ.

В микропроцессорных системах (МПС) обмен информацией с ПУ может выполняться по одному из следующих методов организации системных пересылок:

- ◆ программно-управляемый синхронный или асинхронный обмен без прерывания (метод программного опроса);
- ◆ программный асинхронный обмен с прерыванием;
- ◆ ввод-вывод с прямым доступом к памяти.

Программно-управляемый ВВ без прерывания является простейшим способом обмена, так как требует минимальных аппаратурных затрат, хотя и сопровождается потерями процессорного времени и используется при подключении низ-

коскоростных ПУ (датчики температуры, индикаторы, механические переключатели и т.п.); можно считать, что ПУ всегда готово к обмену с МПр. При подключении указанных ПУ к МПС можно использовать метод простого синхронного ВВ, когда интервалы между последовательными операциями ВВ меньше, чем время реакции ПУ. При этом частота повторения операций ВВ определяется программой, под управлением которой работает МПС. Аппаратная часть интерфейса состоит только из информационного порта данных ввода или вывода.

Второй вариант называется асинхронным программно-управляемым без прерывания, или методом опроса состояния ПУ, и требует наличия, кроме линий данных, по которым передаются собственно данные дополнительных, по которым передаются сигналы управления от МПр к ПУ и сигналы состояния от ПУ к МПр. В качестве сигналов состояния может использоваться слово состояния ПУ, считываемое по шине данных в микропроцессор.

Данный метод применяется, если МПр работает не в реальном времени обмена с ПУ (с высокоскоростными ПУ). Таким образом, аппаратное обеспечение для методов асинхронного обмена требует, кроме шины данных (порта данных), наличия еще двух портов: порта управления для выдачи от МПр к ПУ управляющих сигналов типа “готовность приемника”, “запрос на ввод” и других и порта состояния ПУ для приема от ПУ в МПр сигналов состояния ПУ типа “готовность источника”, “строб сопровождения данных” и других. Таким образом, процедура обмена сводится к предварительной проверке готовности ПУ к передаче данных в МПр при вводе или готовности ПУ к приему новых данных при выводе. В противном случае может быть обеспечен неверный ввод или вывод данных.

При асинхронном программно-управляемом ВВ без прерывания с занятием цикла ВВ (рисунок 1.3) МПр запрашивает состояние ПУ и входит в цикл ожидания готовности ПУ, тратя на это процессорное время. После завершения операции ВВ ПУ снимает сигнал готовности и выставляет заново только при новой готовности ПУ к обмену, для чего после приема данных МПр (ПУ) информирует ПУ (МПр) о приеме данных путем подачи ответного сигнала подтверждения приема ASK (рисунок 1.4).

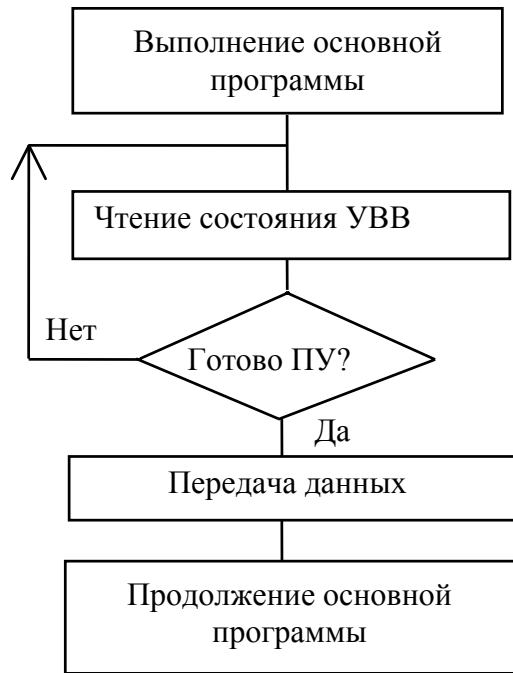


Рисунок 1.3 - Программно-управляемая асинхронная передача данных

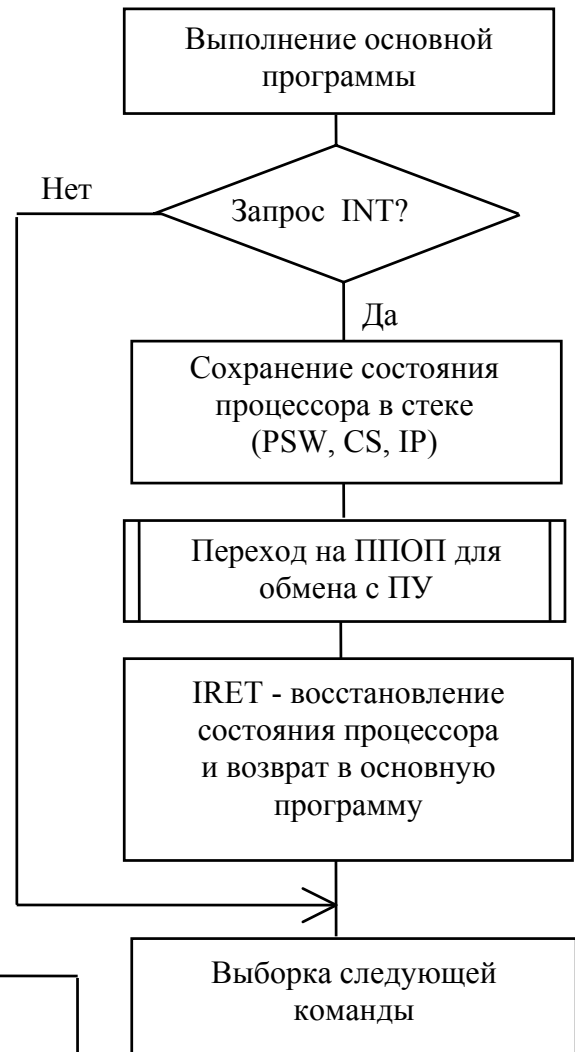


Рисунок 1.5 - Процедура передачи данных по прерыванию

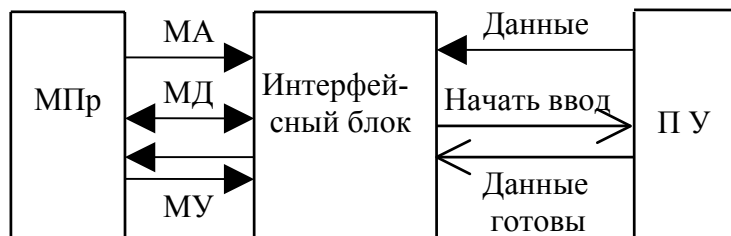


Рисунок 1.4 - Схема асинхронного обмена с УВВ

Предотвратить потери времени в циклах ожидания и существенно ускорить процесс взаимодействия МПр с ПУ, особенно при одновременной работе с несколькими ПУ, позволяет использование системы прерывания. Алгоритм обмена по прерываниям представлен на рисунке 1.5. Данный метод заключается в следующем. Если ПУ готово к обмену, то оно по магистрали управления (МУ) через аппаратуру прерываний выдает в МПр сигнал запроса на прерывание INT. При этом МПр приостанавливает выполнение основной программы, сохраняет в стеке номер следующей невыполненной команды (адрес возврата), слово состояния

процессора и некоторую другую информацию в зависимости от архитектуры МПр и передает управление подпрограмме обработки прерывания, которая реализует протокол обмена с данным ПУ (ввод или вывод).

Таким образом, МПр работает только с ПУ, уже готовыми к обмену, не тратя время на программный опрос готовности ПУ. Организация ВВ в режиме прерываний является наиболее эффективным средством ВВ и обычно используется при подключении ПУ к высокопроизводительным микроЭВМ, а также в МПС реального времени.

В случае использования высокоскоростных ПУ (гибкие и жесткие магнитные диски) часто используется метод прямого доступа к памяти. Данный метод обеспечивает минимальное время обработки запроса на прерывание и максимальную скорость обмена данными, при этом передача данных между ПУ и ОП выполняется без участия процессора. Обмен выполняется под управлением контроллера прямого доступа, выполненного в виде отдельного модуля системы.

1.2 Организация подсистемы прерываний

При организации подсистемы обмена между модулями без прерывания велико время ожидания обмена для других модулей. Кроме того, низка эффективность работы процессора, так как ему необходимо постоянно выполнять последовательный опрос (полинг) других модулей с целью выявления запросов на обмен от ПУ и других модулей. Последовательный опрос реализуется либо с помощью встроенных в БИС средств запросов на обмен, либо программно-аппаратным способом, путем последовательного опроса состояния всех устройств. Например, периферийная БИС контроллера прерываний K1810BH59A имеет один из режимов работы, при котором легко реализуется опрос состояния всех устройств, подключенных к ней.

В системах реального времени для повышения эффективности работы средств поддержки обмена необходимо иметь средства, обеспечивающие прерывание текущего процесса (программы) и переход к обслуживанию запросов на обмен или обработку специфических ситуаций. Средства, реализующие обмен по прерыванию, называются подсистемой прерывания. Все прерывания можно разделить на:

- ◆ **внутренние;**
- ◆ **программные;**
- ◆ **внешние.**

Прерывания, происходящие в результате событий, локализованных внутри модуля системы, например в МПр при делении на ноль, обрабатываются с помощью средств **внутренних прерываний**, в функции которых обычно входит распознавание причины прерывания (вектора прерывания), сохранение текущего состояния процессора (PSW, PC и т.п.) и загрузка в программный счетчик начального адреса подпрограммы, обрабатывающей данную ситуацию.

Программные прерывания инициируются специальными командами, при выполнении которых в стеке запоминается состояние текущего процесса: содержимое регистров общего назначения и специальных регистров, включая слово состояния программы и программного счетчика, после чего обеспечивается загрузка в программный счетчик номера команды подпрограммы обработки прерывания, который содержится в формате команды программного прерывания в виде адреса или номера вектора. Возврат из программного прерывания выполняется с помощью специальных команд или программных средств.

Функциями средств, обслуживающих **внешние прерывания** (аппаратуры прерывания), являются:

- ◆ фиксация запросов на прерывание от внешних источников;
- ◆ определение номера приоритетного незамаскированного запроса для обслуживания;
- ◆ запоминание в стеке состояния текущего процесса (PSW, PC и т.д.);
- ◆ передача управления подпрограмме обслуживания (обработки) данного запроса (ППОП);
- ◆ возврат - восстановление состояния прерванного процесса (программы) и передача ему управления.

В зависимости от способа реализации каждой из перечисленных функций подсистемы прерывания могут классифицироваться:

- ◆ **с маскированием входов** запросов на прерывание;
- ◆ **без маскирования;**

- ◆ **бесприоритетные** (запросы на прерывание обслуживаются в порядке поступления);
- ◆ **приоритетные** (обслуживание запросов происходит в соответствии с назначенными приоритетами, которые могут быть фиксированными или изменяемыми);
- ◆ **одноуровневые** (без вложенности);
- ◆ **многоуровневые**, допускающие вложение ППОП в соответствии с назначенным приоритетом;
- ◆ **динамически маскируемые**, при которых допускается обслуживание запроса на прерывание от источника с меньшим приоритетом (специального маскирования);
- ◆ **безвекторные**, при которых передача управления осуществляется по фиксированному адресу независимо от источника прерывания;
- ◆ **векторные** (запрос от каждого устройства обслуживается своей ППОП, для которой служит вектор точек входа (начальный адрес ППОП)).

В микропроцессорном комплекте (МПК) серии K1810 имеется специальная БИС ВН59А, с помощью которой можно путем программной настройки создавать подсистемы прерываний следующих видов: с маскированием входов; приоритетные (с фиксированными и циклически изменяемыми приоритетами); многоуровневые (до 64 уровней); векторные (до 64 векторов) и динамически маскируемые.

1.3 Периферийные БИС

В состав МПС входит широкий набор ПУ, предназначенных для обмена информацией с МПр. В качестве примеров ПУ можно привести дисплеи, внешние запоминающие устройства (НЖМД, НГМД), каналы передачи информации, измерительные приборы и т.д. Связь МПр с устройствами подобного рода осуществляется промежуточными преобразователями, которые совместно с относящимися к ним портами ВВ получили название периферийных контроллеров.

Унификация интерфейсов ПУ привела к созданию ряда стандартных контроллеров, ориентированных на применение конкретного интерфейса ввода-вывода и микропроцессорного комплекта (МПК). В их функции, кроме ввода-вывода, входит:

- ◆ формирование периферийного интерфейса (заданного набора и разрядности передаваемых данных, адреса и состава управляющих сигналов);
- ◆ физическое управление ПУ в соответствии с командами ввода-вывода от микропроцессора;
- ◆ перенос протокола обмена с МПр на более высокий уровень интеллектуальности;
- ◆ электрическое согласование сигналов.

Передача функций физического управления ПУ его контроллеру освобождает МПр от трудоемких и непроизводительных операций управления ПУ на низшем уровне. При этом функции МПр сводятся к общему управлению контроллеров ПУ с помощью высокоуровневых команд обмена (IN, OUT) и слов состояния, что упрощает программное обеспечение, уменьшает загрузку МПр и повышает общую производительность МПС.

Благодаря успехам электронной промышленности в настоящее время разработано множество однокристалльных периферийных контроллеров, а также ряд вспомогательных схем к ним, которые называются периферийными БИС. В таблице 1.1 приведен неполный перечень БИС фирмы Intel, программно совместимых с различными моделями МПр данного семейства. Эти БИС не только упрощают разработку программного обеспечения и повышают производительность МПС, но и облегчают проектирование аппаратуры, повышают компактность и надежность систем и в большинстве своем являются программируемыми на несколько режимов работы, что обеспечивает многофункциональность их использования. Ряд БИС выполняют ограниченный набор функций в составе МПС.

В любом случае эти устройства решают узкоспециализированные задачи, такие как организация параллельного или последовательного интерфейса, распределение интервалов времени, управление вводом с клавиатуры и выводом на индикацию или управления ЭЛТ и другие. Однако остается еще много нерешенных проблем, для чего в состав МПК вводятся БИС, представляющие собой однокристалльные микроЭВМ с системным интерфейсом для работы с ПУ (микроконтроллеры), либо сопроцессоры ПУ, представляющие возможности подключения различных типов ПУ.

Контрольные вопросы для самопроверки

1. Какие методы обращения к ПУ из программы вы знаете?
2. При использовании каких методов обращения к ПУ количество периферийных устройств может превышать 35000?
3. Какие виды прерываний входят в понятие подсистемы прерывания?
4. Перечислите типы прерываний, характеризующие любую систему прерываний.
5. Что такое динамически маскируемые запросы на прерывание?
6. Какие способы обмена информацией (ввода-вывода) вы знаете?
7. Как реализуется асинхронный обмен между МПр и УВВ и в каких случаях он применяется?
8. Как реализуется обмен между МПр и УВВ по прерываниям и в каких случаях он применяется?

Таблица 1.1 - Функциональный состав БИС МПК

Наименование БИС (назначение)	Обозначение (аналог)	Задержки нс.	Разряд- ность	Колич. вывод.
Центральный процессорный элемент параллельной обработки данных (центральный процессор)	K580BM80 K580BM80A 8080 8080A	400-2000	8	40
Центральный микропроцессорный элемент	K1810BM86 8086	400	16	40
Программируемый последовательный интерфейс связи	K580BB51 8251, 8251A	320-1350	8 <-> 1	28
Программируемый параллельный интерфейс связи	K580BB55 8255, 8255A	850	8	40
Программируемый таймер	K580BI54 8254	380	8, 16	24
Программируемый контроллер ПДП	K580BT57 8257	320-400	8, 16	40
Программируемый контроллер прерываний	K1810BH59A 8259A	850	8	28
Программируемый контроллер ЭЛТ	K580BG75 8275	480	8	40
Программируемый контроллер клавиатуры и индикации	K580BB79 8279	500	8	40

Окончание таблицы 1.1

Генератор тактовых импульсов	K1810ГФ84А 8284А	T=125	-	20
Контроллер системной шины	K1810ВГ88 8288	50	-	20
Арбитр шины	K1810ВБ89 8289	100	-	20
Буферные регистры	K580ИР82,83 8282, 8283		8	20
Шинные формирователи	K580ВА86,87 8286, 8287		8	20

2 Обработка прерываний на основе МПК K1804

2.1 Блок векторных приоритетных прерываний K1804ВН1

Микросхема K1804ВН1 входит в состав МПК K1804, выполненного на базе ТТЛШ-технологии, предназначенного для построения быстродействующих контроллеров, процессоров мини- и микроЭВМ и других средств вычислительной техники.

БИС K1804ВН1 представляет собой 8-разрядную микропрограммируемую наращиваемую схему векторного приоритетного прерывания (БВПП), предназначенную для выполнения приоритетной обработки запросов прерываний, поступающих от различных источников, в составе устройств обработки прерываний центральных процессоров микроЭВМ.

БВПП обеспечивает:

- ◆ прием и фиксацию сигналов прерывания (\sim IRQ), поступающих от устройств ЭВМ;
- ◆ выдачу вектора прерываний в соответствии с приоритетом запроса;
- ◆ маскирование поступающих запросов;
- ◆ запрет и разрешение выработки сигнала запроса на прерывание в МПР (\sim RQINR);
- ◆ установку и автоматическое изменение состояния приоритетов обработанных запросов.

Условное графическое обозначение БВПП и структурная схема приведены на рисунках 2.1 и 2.2, а назначение выводов - в таблице 2.1.

Схема включает 8-разрядный регистр прерываний, состоящий из двух каскадов: регистра защелки (RZ), реализованного на асинхронных RS-триггерах, и синхронного регистра хранения запросов прерывания (RP), реализованного на D-триггерах, на входы которого поступают запросы с выходов RZ (рисунок 2.3).

Прием запросов со входов $\sim\text{IRQ7-0}$ осуществляется в двух режимах:

♦ режим 1: фиксация одновременно поступающих запросов либо добавления вновь поступивших запросов к еще необработанным запросам в каждом такте при сигнале $\text{COMO}=0$. При этом RZ фиксирует все поступившие запросы на входы $\sim\text{IRQ7-IRQ0}$ в течение такта, а по заднему фронту сигнала синхронизации T содержимое RZ переписывается в RP;

♦ режим 2: при $\text{COMO}=1$ RZ выполняет функции инвертора, а в RP в каждом такте фиксируются только те запросы, которые поступили на входы $\sim\text{IRQ}$ к моменту прихода заднего фронта сигнала синхронизации T. При этом в каждом такте информация в RP обновляется в соответствии только с вновь поступившими запросами $\sim\text{IRQ7-IRQ0}$.

8-разрядный регистр маски (RM) позволяет маскировать отдельные запросы IRQ или полностью запрещать обработку прерываний. RM имеет двунаправленные выходы MK7-MK0 , предназначенные для загрузки и чтения регистра маски, для чего в систему микрокоманд БВПП включены микрокоманды (МК) установки и очистки всех разрядов RM и отдельных его разрядов.

Маскирование поступивших запросов выполняется на комбинационной схеме устройства маскирования (UM), выполненного на элементах 2И и реализующего функцию $\text{UM}=\text{RP} \& \sim\text{RM}$, т.е. функцию сброса запросов $\sim\text{IRQ}$ в тех разрядах RP, в которых записаны коды “1” регистра маски.

Буферные схемы маски (BM) выполнены по схеме с тремя состояниями и предназначены для выдачи содержимого RM на выходы (входы) магистрали MK7-MK0 .

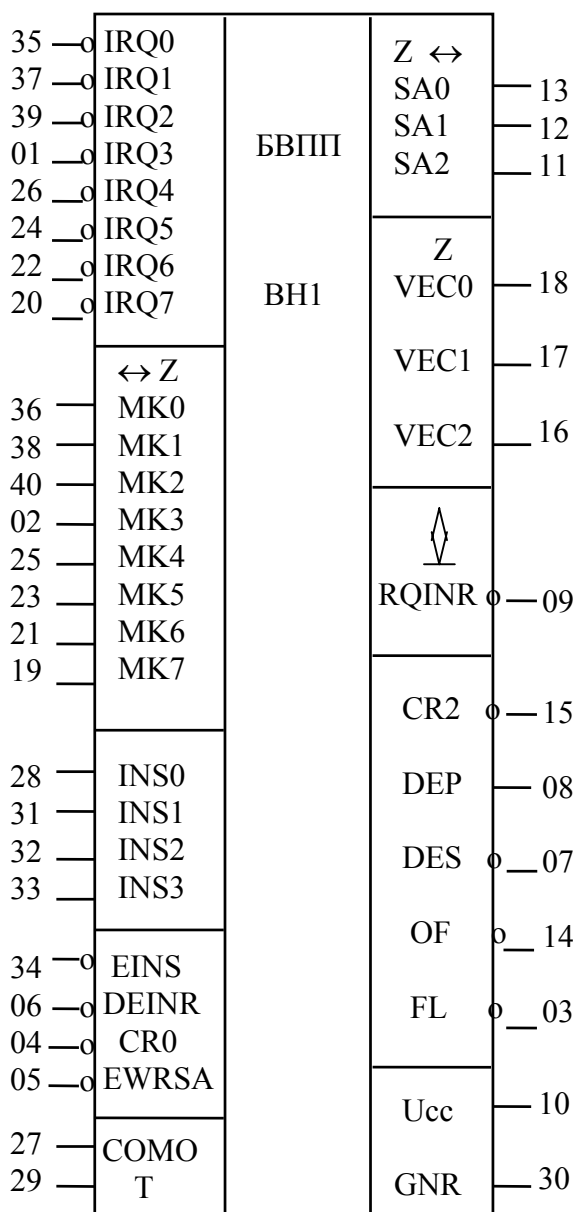


Таблица 2.1 - Назначение выводов БВПП ВН1

Обозначение	Функциональное назначение
~IRQ7-IRQ0 MK7-MK0	Входы запросов прерывания Двунаправленные выходы маски
~DEINR	Вход сигнала запрета на прерывание
~CR0	Вход сигнала переноса из предыдущей группы
~EW RSA	Вход сигнала разрешения записи состояния
~EINS	Вход сигнала разрешения выполнения микрокоманды
INS3-INS0	Входы кода выполняемой микрокоманды
COMO	Вход сигнала управления режимом
T	Тактовый сигнал
SA2-SA0	Двунаправленные выходы регистра состояния
VEC2-VEC0	Выходы шины вектора прерывания
~RQINR	Выход сигнала запроса на прерывание
~CR2	Выход сигнала переноса в следующую группу
DEP	Выход сигнала параллельный запрет
~DES	Выход сигнала последовательный запрет
~OF	Выход сигнала переполнения
~FL	Выход сигнала флага

Рисунок 2.1 - Условное графическое обозначение блока векторных прерываний K1804BH1

Детектор прерываний (DP) обнаруживает любой незамаскированный запрос прерывания на выходах устройства маскирования и реализован на 8-входовой схеме ИЛИ, а сигнал “0” на его выходе свидетельствует об отсутствии запросов на выходах UM.

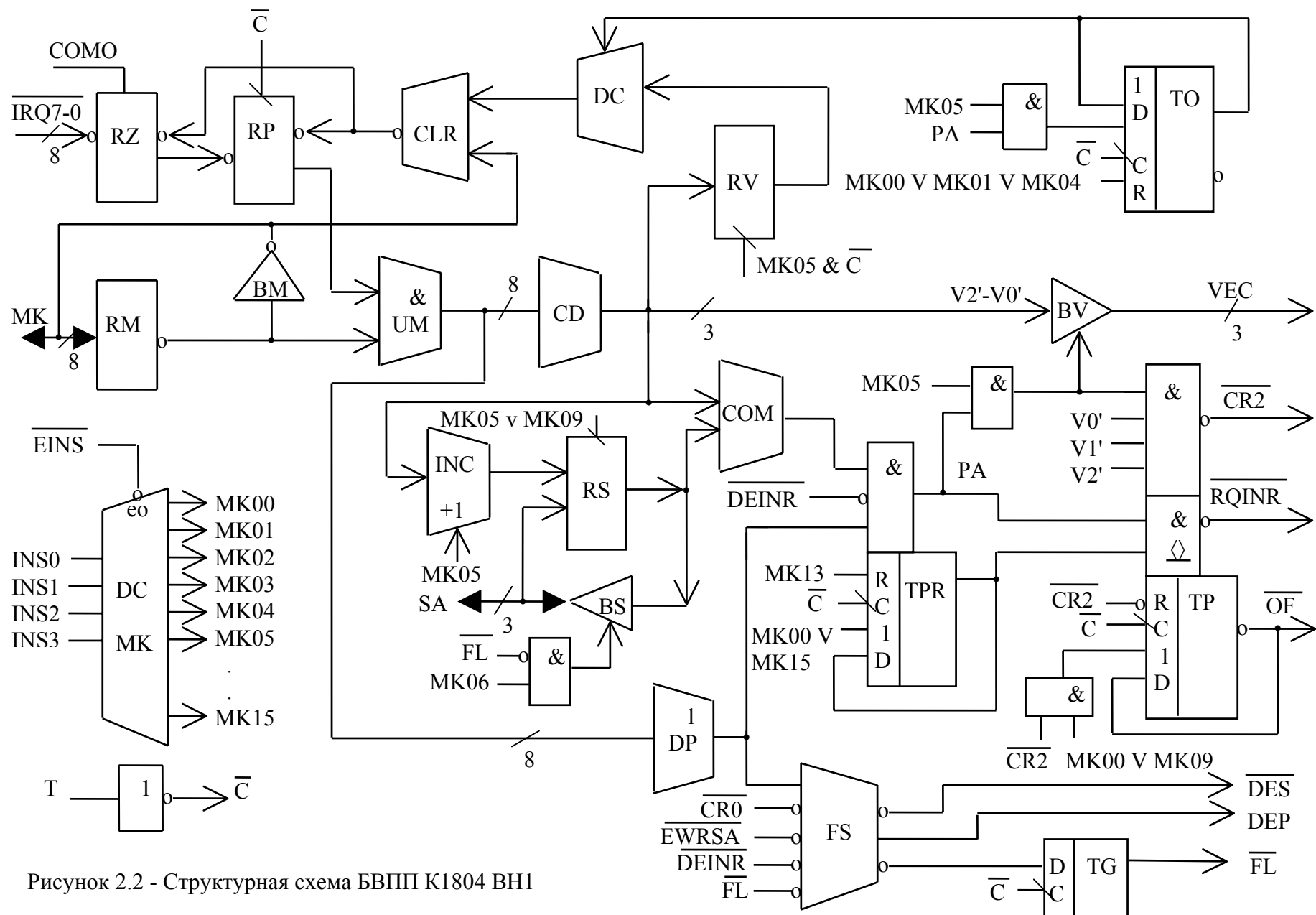


Рисунок 2.2 - Структурная схема БВПП К1804 ВН1

Рисунок 2.3 - Функциональная схема одного бита регистра прерываний

Шифратор приоритета (CD) преобразует 8-разрядный код с выходов UM в двоичный код вектора прерывания, указывающего номер незамаскированного запроса прерывания с высшим приоритетом ~IRQ7-IRQ0.

Буферные схемы вектора (BV) выдают значение вектора прерывания на выходы VEC2-VEC0 при выполнении микрокоманды “Чтение вектора” МК05 при нулевом уровне сигнала T, а в RV загружается код считанного вектора. При выполнении остальных микрокоманд буферные схемы BV переводятся в третье состояние.

Трехразрядный регистр вектора (RV) служит для записи и хранения двоичного кода вектора прерывания, получаемого с выхода шифратора приоритетов CD, при выполнении МК “Чтение вектора”, который в дальнейшем используется для очистки разряда регистров RZ и RP принятого на обработку запроса.

Схема очистки регистров прерывания (CLR) позволяет обеспечить несколько режимов обработки прерываний:

- ◆ очистка всех поступивших запросов прерывания в RZ и RP;
- ◆ очистка разрядов RZ и RP сигналами с шины маски МК7-МК0;
- ◆ очистка разрядов RZ и RP сигналами с выходов регистра маски RM;
- ◆ очистка разряда RZ и RP под управлением регистра вектора RV (рисунок 2.3, где C7-C0 - выходы схемы очистки).

Последний режим применяется для сброса разряда считанного вектора запроса прерывания, принятого на обработку в МПр с шины VEC, по МК04. При этом очистка RZ и RP разрешена только, если триггер разрешения очистки (ТО) находится в состоянии “1”. Установка ТО происходит при выполнении МК05

“Чтение вектора”. Такая схема позволяет обеспечить защиту от потери принятого запроса $\sim \text{IRQ}_i$ вследствие помехи или ошибок пользователя в режиме программного опроса аппаратуры прерываний.

Для обеспечения гибкости обработки прерываний часто вводят понятие порога приоритета запросов прерывания. При введении порога обрабатываются только те запросы, приоритет которых выше указанного порога, что равнозначно маскированию всех приоритетов ниже указанного порога.

Порог приоритетов может быть определен микропрограммой (программой), т.е. установлен путем загрузки в регистр порога прерывания кода, ниже которого все запросы игнорируются, либо установлен автоматически аппаратными средствами на уровне запроса, на единицу превышающем приоритет прерывания, которое обслуживается в данный момент, что предотвращает поступление многократных запросов от одного источника и обеспечивает вложенность обработки запросов прерывания по их приоритету.

Для реализации этого метода в структуру БВПП К1804ВН1 введены регистр состояния (RS) и узел приращения (инкремента (INC)) RS, значение которого сравнивается с приоритетом обрабатываемого запроса с помощью схемы сравнения (COM).

Таким образом, регистр состояния RS определяет самый низкий приоритет, при котором обработка запроса будет разрешена. Двухнаправленные выходы SA2-SA0 служат для загрузки и чтения регистра состояния. Буферные схемы регистра состояния (BS) выполнены на три состояния и предназначены для выдачи содержимого RS на выходы SA2-SA0 по МК “Чтение RS”, при выполнении остальных МК выходы SA2-SA0 находятся в третьем состоянии, а по МК “Загрузка RS” осуществляется запись в RS кода с шины SA2-SA0, если на входе БВПП $\sim \text{EWRSA}$ установлен низкий уровень сигнала ($\sim \text{EWRSA} = 0$).

Во время выполнения МК “Чтение вектора” схема инкремента INC наращивает на “1” текущее значение вектора прерывания с выхода шифратора приоритета CD и результат загружается в RS, что обеспечивает установку порога приоритета в RS.

Схема сравнения (COM) производит сравнение обрабатываемого запроса (выход CD) и значения RS и вырабатывает сигнал “вектор прерывания больше или равен содержимому RS”, который блокирует выработку сигнала запроса на прерывание $\sim RQINR$ в МПр ($COM=0$, т.е. $VEC < RS$).

Реализация процедуры установки порога прерывания позволяет также организовать вложение подпрограмм обработки прерываний, т.е. когда поступление запроса с более высоким приоритетом во время обработки текущего запроса может прервать выполнение подпрограмм обработки прерываний (ППОП). Реализовать вложенные прерывания можно, сохраняя содержимое RS в стеке перед обработкой каждого прерывания и восстанавливая его по окончании обработки прерывания. Остальные процедуры сохранения состояния системы являются стандартными при реализации вложенности выполнения ППОП и зависят от архитектуры процессора.

Схема формирования запроса содержит:

- ◆ триггер разрешения запроса на прерывание (TRP);
- ◆ триггер переполнения (TP);
- ◆ триггер разрешения младшей группы (триггер готовности TG);
- ◆ комбинационные схемы для формирования выходных сигналов $\sim RQINR$, DEP, $\sim DES$, $\sim CR2$ и внутреннего осведомительного сигнала PA.

Выход сигнала запроса на прерывание $\sim RQINR$ выполнен с открытым коллектором и вырабатывается, если входные сигналы $\sim IRQ$ приняты и прошли приоритетную обработку.

Сигнал PA указывает на возможность выработки сигнала запроса на прерывание $\sim RQINR$ независимо от состояния триггера разрешения прерывания TRP, если номер текущего запроса не замаскирован и имеет более высокий приоритет, чем запрос, находящийся в обработке (номер текущего запроса больше или равен содержимому регистра состояния RS), и если не вырабатывается сигнал аппаратного запрета прерывания $\sim DEINR$ (рисунок 2.2).

Триггер разрешения прерывания (TRP) устанавливается в “1” при выполнении МК “Общий сброс” и “Разрешение запроса прерывания” и обнуляется при выполнении МК “Запрет запроса прерывания”. Остальные МК не вызывают изменения TRP.

Триггер переполнения регистра состояния (ТР) устанавливается в состояние “0” после считывания вектора самого высокого приоритета (вектор 7). Сигнал $\sim OF$ с его выхода используется для запрещения прерывания и указывает на переполнение RS. Он сохраняет низкий уровень до выполнения МК “Общий сброс” или “Загрузка RS”.

Триггер разрешения младшей группы (триггер готовности TG) используется при каскадном включении нескольких БИС БВПП для увеличения числа входов запросов прерываний $\sim IRQ$. В такой системе в любой момент времени только в одной БИС K1804BH1 этот сигнал имеет низкий уровень, указывающий на группу из восьми прерываний, в которую входит прерывание с наивысшим воспринимаемым приоритетом, и служит для образования старших разрядов регистра состояния RS при каскадном включении БИС БВПП при выполнении МК “Чтение RS”. Выходом TG является вывод $\sim FL$ микросхемы.

Загрузка TG осуществляется со входа $\sim EWRSA$ при выполнении МК “Загрузка RS” или со входа $\sim CR0$ при выполнении МК “Общий сброс”, а установка TG осуществляется при выполнении МК “Чтение вектора”.

Выходной сигнал переноса в следующую группу $\sim CR2$ формируется при выполнении МК “Чтение вектора”, когда считывается вектор прерывания с наивысшим приоритетом в данной группе (вектор 7). Сигнал используется при каскадном включении БИС и должен быть соединен со входом $\sim CR0$ более старшей группы для передачи информации о состоянии БИС БВПП.

При подаче сигнала $\sim DEINR=1$ запрещается формирование сигнала $\sim RQINR$ от данной микросхемы, а также вырабатывается выходной сигнал “Последовательный запрет” ($\sim DES$).

Выходные сигналы $\sim DES$ и DEP (сигналы последовательного и параллельного запрета соответственно) используются только при последовательном или параллельном каскадном включении БИС БВПП. При этом сигнал DEP обычно используется для формирования старших разрядов номера вектора прерывания, так как высокий уровень сигнала на его выходе указывает на БИС БВПП с наивысшим принятым запросом прерывания.

Ниже приведены булевы функции, реализуемые комбинационными схемами формирования выходных и осведомительных сигналов:

$$\begin{aligned}
 PA &= COM \& DP \& \sim DEINR; & \sim RQINR &= \sim(PA \& TRP); \\
 DEP &= (FL \vee (COM \& DP)); & \sim DES &= DEP \& \sim DEINR; \\
 PRV &= MK05 \& PA; & \sim CR2 &= \sim(PRV \& 7).
 \end{aligned}$$

2.2. Система микрокоманд БВПП К1804ВН1

Система микрокоманд БВПП ВН1 приведена в таблице 2.2.

Микрокоманда “Общий сброс”. При включении питания все разряды регистра микрокоманд (РМК) устройства управления (УУ) процессора устанавливаются в состояние “0”, а в следующем такте в БВПП автоматически выполняется МК “Общий сброс”. Выполнение любой микрокоманды в БВПП разрешается только при низком уровне на входе сигнала “Разрешение выполнения микрокоманды” $\sim EINS$, при $\sim EINS=1$ выполнение МК игнорируется. По МК00 “Общий сброс” производится обнуление RZ, RP, RM, RS и триггера разрешения очистки регистра прерывания ТО. Шины VEC2-VEC0, МК7-МК0 и SA2-SA0 находятся в третьем состоянии, при этом в триггер разрешения младшей группы TG будет загружена информация со входа $\sim CR0$, выход $\sim CR2$ будет иметь высокий уровень, а триггер разрешения прерывания TRP устанавливается в состояние “1”, что позволяет воспринимать запрос с любым приоритетом в RP с выработкой сигнала запроса на прерывание $\sim RQINR$.

Выполнение МК01 “Очистка регистра прерываний” осуществляет обнуление всех разрядов RZ, RP и ТО. Состояние остальных узлов не изменяется.

С помощью МК02 “Очистка регистра прерываний сигналами с шины маски” выполняет обнуление тех разрядов RZ и RP, которым соответствует код “1” на выводах шины маски МК7-МК0.

Таблица 2.2 - Система микрокоманд БВПП К1804ВН1

№ МК	INS3-0	Содержание микрокоманды
МК00	0000	Общий сброс
МК01	0001	Очистка регистра прерывания
МК02	0010	Очистка регистра прерывания сигналами с шины маски
МК03	0011	Очистка регистра прерывания под управлением регистра маски
МК04	0100	Очистка регистра прерывания под управлением регистра вектора
МК05	0101	Чтение вектора
МК06	0110	Чтение регистра состояния
МК07	0111	Чтение регистра маски
МК08	1000	Установка регистра маски
МК09	1001	Загрузка регистра состояния
МК10	1010	Поразрядная очистка регистра маски
МК11	1011	Поразрядная установка регистра маски
МК12	1100	Очистка регистра маски
МК 13	1101	Запрет запроса прерывания
МК14	1110	Загрузка регистра маски
МК15	1111	Разрешение запроса прерывания

МК03 “Очистка регистра прерываний под управлением регистра маски” выполняется аналогично предыдущей МК, только очистка разрядов RZ и RP производится в соответствии с кодами “1” в разрядах регистра маски RM. При этом на выводах МК7-МК0 появляется значение кода из RM.

При выполнении **МК04 “Очистка регистра прерываний под управлением регистра вектора”** происходит сброс разряда RZ и RP, соответствующего последнему прочитанному вектору, если ТО=1. Установка ТО в “1” производится при выполнении МК05 “Чтение вектора”, если был выработан сигнал приоритетного запроса РА, а его обнуление - при выполнении МК “Общий сброс”, “Очистка регистра прерываний” и “Очистка регистра прерываний под управлением регистра вектора” (МК00, МК01 и МК04).

Микрокоманда **МК05 “Чтение вектора”** служит для выдачи значения вектора на выходы VEC2-VEC0, соответствующего запросу с наивысшим приоритетом из поступивших и незамаскированных со входов $\sim\text{IRQ7-IRQ0}$. При выполнении остальных микрокоманд выходы VEC2-VEC0 находятся в третьем состоянии. По этой микрокоманде также в RV загружается значение считанного вектора, в RS загружается значение вектора, увеличенное на “1”, ТО устанавливается в состояние “1”, если имеется более приоритетный запрос на прерывание ($\text{PA}=1$). Если сигнал $\text{PA}=0$, то шина VEC остается в третьем состоянии (рисунок 2.2). Выполнение МК04 “Очистка регистра прерываний под управлением регистра вектора” после МК05 “Чтение вектора” позволяет после считывания вектора для перехода на ППОП выполнить сброс обработанного запроса в регистрах RZ и RP. МК04 должна выполняться только в микропрограмме входа в ППОП, иначе при вложенности обработки запросов IRQ номер сбрасываемого бита в RZ и RP будет потерян.

По МК06 “Чтение регистра состояния” производится считывание и выдача на шину SA2-SA0 кода из RS при значении триггера готовности $\text{TG}=0$.

МК07 “Чтение регистра маски” производит выдачу на шину МК7-МК0 значения RM, а при выполнении МК08 “Установка регистра маски” производится запись во все разряды RM кода “1”, в результате все прерывания будут запрещены (замаскированы).

По МК09 “Загрузка регистра состояния” выполняется запись в RS кода со входов SA2-SA0, если на входе $\sim\text{EWRSA}$ установлен код “0”. Иначе регистр состояния обнуляется. При этом код со входа $\sim\text{EWRSA}$ загружается в триггер готовности TG.

МК10 “Поразрядная очистка регистра маски” производит обнуление тех разрядов регистра маски, которым соответствует код “1” на выводах шины маски МК7-МК0, а по МК11 “Поразрядная установка регистра маски” - установка в “1” тех разрядов RM, которым соответствует код “1” на выводах шины маски МК7-МК0, остальные разряды RM остаются без изменения.

По МК12 “Очистка регистра маски” выполняется обнуление всех разрядов RM, в результате все прерывания будут размаскированы (разрешены) и их реализация будет зависеть от регистра состояния RS и триггера разрешения пре-

рывания TPR. Микрокоманда **МК14 “Загрузка регистра маски”** производит запись в RM информации с выводов шины МК7-МК0.

Обработка всех запросов на прерывание IRQ может блокироваться по **МК13 “Запрещение запроса прерывания”**, по которой производится обнуление триггера разрешения прерывания ($TPR:=0$), а по **МК15 “Разрешение запроса прерывания”** производится его установка ($TRP:=1$), в результате чего выход запроса на прерывание $\sim RQINR$ будет разрешен.

Все преобразования информации, выполняемые в БВПП по каждой МК, приведены в таблице 2.3.

Система микрокоманд БВПП позволяет разработчику СВТ создавать гибкую систему команд процессора, относящуюся к аппаратуре прерываний, а программисту на основе такой системы команд реализовать различные дисциплины обслуживания запросов на прерывание. То есть на основе базовой дисциплины фиксированных приоритетов можно реализовать режим специального маскирования, обработку запросов без вложенности, программного опроса аппаратуры прерываний и другие, а также иметь команды, позволяющие значительно сократить длину ППОП, а следовательно, и уменьшить время реакции системы на прерывание. Заметим также, что для данной БИС характерно размещение триггера разрешения прерываний непосредственно в составе аппаратуры прерываний, поэтому команды запрещения и разрешения прерываний рекомендуется разрабатывать на основе МК13 и МК15, а не путем маскирования (размаскирования) запросов через регистр маски (МК08, МК12), что потребует дополнительного сохранения в стеке старого значения регистра маски RM.

Таблица 2.3 - Преобразования, выполняемые в БВПП

Код МК	Содержание микрокоманды	Преобразования, выполняемые в узлах, регистрах и триггерах БВПП
МК00 0000	Общая очистка	$RZ:= RP:= RM:= RS:= 0, TO:= 0, TRP:= 1, TG:= \sim CR0, TP:= \sim CR2, VEC=SA= MK= Z$
МК01 0001	Очистка регистра прерывания	$RZ := 0, RP := 0, RS := 0, TO := 0, VEC = SA = MK = Z$

МК02 0010	Очистка RP и RZ сигналами с шины маски	$RZ := RZ \& \sim MK, RP := RP \& \sim MK,$ $VEC = SA = RM = Z,$
МК03 0011	Очистка RP и RZ под управлением Rg маски	$RZ := RZ \& \sim RM, RP := RP \& \sim RM,$ $VEC = SA = Z, MK := RM$
МК04 0100	Очистка RP и RZ под управлением Rg вектора	$RZ = RP := (RZ \& \sim RMV) \& TO,$ где RMV - унитарный код от RV, $VEC = SA = RM = Z$
МК05 0101	Чтение вектора	$RV := CD, TO := PA, TP := \sim CR2,$ $RS := (CD + 1) \& PA,$ т.е. $RS := CD + 1$ при $PA=1$, $RS := 0$ при $PA=0$, $SA = MK = Z,$ $TG := (DP \& \sim CR0) \vee CR2 \vee DEINR,$ $VEC = CD,$ если $PA=1$, иначе $VEC = Z$
МК06 0110	Чтение регистра состояния	Если $\sim TG = 0$, то $SA = RS$, иначе $SA = Z,$ $VEC = MK = Z$
МК07 0111	Чтение регистра маски	$MK = RM, SA = VEC = Z$
МК08 1000	Установка регистра маски	$RM := 11111111, VEC = SA = MK = Z$
МК09 1001	Загрузка регистра состояний	$TP := CR2 = 1, TG := \sim EWRSA,$ $RS := SA,$ если $\sim EWRSA = 0$, иначе $RS := 0, VEC = MK = Z$
МК10 1010	Поразрядная очистка регистра маски	$RM := RM \& \sim MK, VEC = SA = Z$
МК11 1011	Поразрядная установка регистра маски	$RM := RM \vee MK, VEC = SA = Z$
МК12 1100	Очистка регистра маски	$RM := 0, VEC = SA = MK = Z$
МК13 1101	Запрет запроса прерывания	$TRP := 0, VEC = SA = MK = Z$
МК14 1110	Загрузка регистра маски	$RM := MK, VEC = SA = Z$
МК15 1111	Разрешение запроса прерывания	$TRP := 1, VEC = SA = MK = Z$

Z - выходы находятся в третьем состоянии

2.3 Применение БВПП ВН1 в составе МПК К1804

Одна БИС ВН1 может быть использована для обработки до восьми входных сигналов прерывания $\sim IRQ$. Для увеличения числа входных сигналов прерывания несколько БИС БВПП могут быть соединены по схеме каскадного включения с последовательным или параллельным переносом между БИС.

В режиме последовательного каскадного включения выход сигнала последовательного запрета $\sim\text{DES}$ БИС с более высоким приоритетом последовательно соединяется со входом “Запрещение прерываний” $\sim\text{DEINR}$ БИС с более низким приоритетом, который запрещает прерывания по запросам с относительно более низкими приоритетами (т.е. выход $\sim\text{RQINR}$ этих БИС будет находиться в состоянии ”выключено”), а выход переноса в старшую группу $\sim\text{CR2}$ соединен со входом переноса из младшей группы $\sim\text{CR0}$.

При параллельном каскадном включении БИС БВПП для организации переносов между БИС используется схема ускоренного группового переноса K1804BP1 [4, 6, 9], при этом появление сигнала “Параллельный запрет” DEP на выходе любой БИС БВПП обеспечивает одновременное запрещение запросов на прерывание на выходах $\sim\text{RQINR}$ во всех группах с более низкими приоритетами. Для подключения схемы ускоренного переноса выходы DEP всех БИС БВПП подаются на одноименные входы распространения переноса $\sim\text{Pi}$, выходы переноса CX , CY , CZ соединены со входами сигнала “Запрещение прерывания” $\sim\text{DEINR}$ БИС с более низкими приоритетами, а на все входы генерации переноса $\sim\text{Gi}$ подается напряжение питания +5 В.

В составе микроЭВМ БВПП занимает важное место и служит для выработки сигнала прерывания $\sim\text{RQINR}$ и номера вектора для вызова ППОП в соответствии с приоритетом поступившего запроса.

На рисунке 2.4 приведена структурная схема включения 8-уровневого БВПП в структуру УУ микроЭВМ.

Входы $\sim\text{CR0}$ и $\sim\text{EWRSA}$ должны быть заземлены, чтобы выходной сигнал младшей разрешенной группы $\sim\text{FL}$ при выполнении МК “Общий сброс” или “Загрузка регистра состояния” приобрел низкий уровень.

Выход триггера переполнения регистра состояния $\sim\text{OF}$ соединяется со входом $\sim\text{DEINR}$ “Запрещение прерывания”, чтобы запретить прерывания после считывания вектора с наивысшим приоритетом (так как иначе при считывания вектора 7 в RS загружается код “0” и любой запрос, даже с самым низким приоритетом может прервать выполнение ППОП с самым высоким приоритетом).

Выходные сигналы \overline{DEP} , $\sim DES$, $\sim CR2$, $\sim FL$ используются только при каскадном включении БИС.

Сигнал выбора режима $COMO$ заземлен, поэтому в RP будут фиксироваться все запросы $\sim IRQ$ любой длительности и поступившие в БВПП в любой момент времени между тактами T (RZ работает в режиме асинхронного RS -триггера). При подключении входа $COMO$ к источнику питания ($COMO=1$) в RP в каждом такте будут фиксироваться только те запросы $\sim IRQ$, которые поступили на момент прихода отрицательного фронта тактового сигнала T , и в каждом такте содержимое RP обновляется. Поэтому первый режим ($COMO=0$) используется, как правило, для фиксации и обработки внутренних прерываний, а второй режим ($COMO=1$) для обработки прерываний от внешних источников, которые выставляют сигнал запроса $\sim IRQ$ через линию интерфейса, а снимают этот сигнал только когда запрос будет принят на обработку путем выдачи ответного сигнала “Разрешение прерывания”.

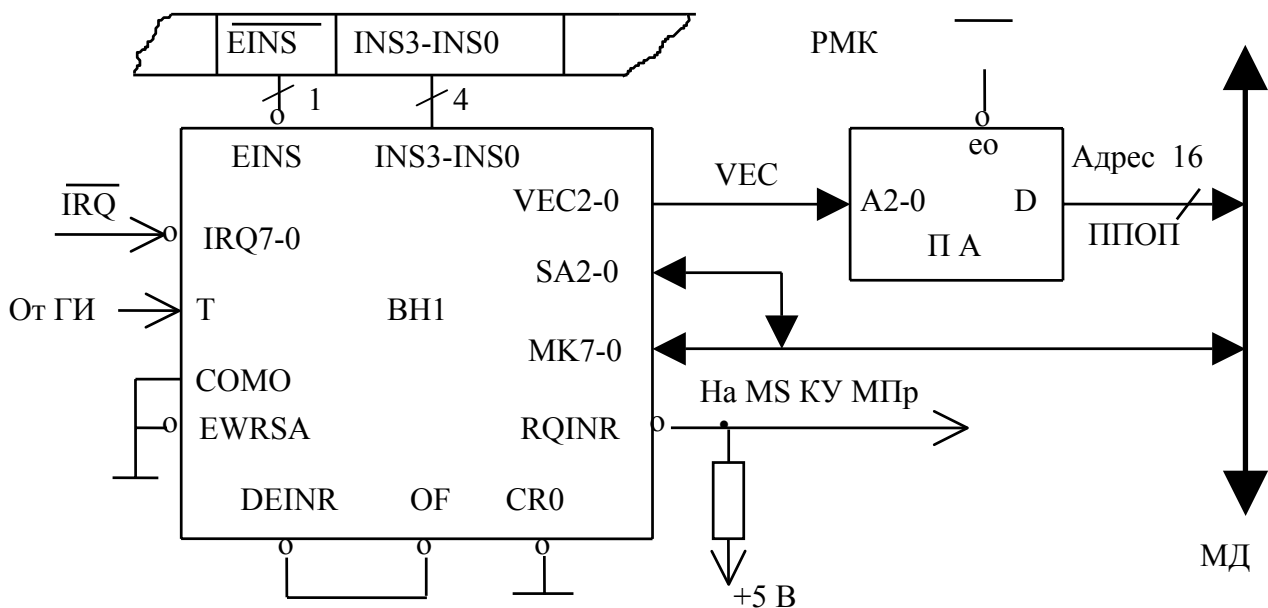


Рисунок 2.4 - Структурная схема включения 8-уровневого БВПП в составе микроЭВМ при обработке прерываний на командном уровне

Управление работой БИС БВПП производится из регистра микрокоманд PMK УУ, для чего в формат микрокоманды введены два поля: одноразрядное $\sim EINS$ и 4-разрядное поле кода микрокоманды INS . Так как выполнение MK в БВПП разрешается только при $\sim EINS=0$, то поле $INS3-0$ в PMK может использо-

ваться другими устройствами микроЭВМ, например, операционным устройством (ОУ) для задания констант от 0 до 15 либо в качестве поля INS может использоваться поле адреса A3-A0 РЗУ операционного устройства, при этом на этапе разработки микропрограмм необходимо учитывать, что при выполнении МК в БВПП ($\sim EINS=0$) поле A3-A0 нельзя использовать в качестве источника операнда для ОУ.

На рисунке 2.4 двунаправленные шины маски МК7-МК0 и регистра состояний SA2-SA0 соединены с шиной данных процессора, что позволяет производить загрузку по шине данных в RM и RS данных либо из ОУ, либо ОЗУ, а также считывать состояние RM и RS в процессор с помощью соответствующих микрокоманд либо команд процессора (т.е. включения в систему команд дополнительных команд типа “Чтение маски”, “Загрузка маски”, Чтение RS”, “Запись в RS” и других). Это позволяет помещать информацию о состоянии БВПП и маски при необходимости в стек микропрограммными или программными средствами при обработке вложенных прерываний.

По МК05 “Чтение вектора” из преобразователя адреса (ПА), реализованного на ПЗУ или ПЛМ при выполнении в УУ на основе БМУ ВУ4 МК “Условного перехода по вектору” (CJV) [2, 4, 6, 9], вырабатывается сигнал $\sim VE$, разрешающий считывание из ПЗУ, и считанный начальный адрес ППОП через шину данных (или шину адреса) в этом же такте записывается в программный счетчик. В зависимости от архитектуры процессора функции ПА можно возложить на ОП, в которой размещается таблица векторов прерываний IDT, а номер вектора сдвинутый на один или два разряда будет являться адресом таблицы IDT, в которой хранится начальный адрес ППОП (программный счетчик IP) и другая информация (например, содержимое кодового сегмента CS для данной ППОП).

На рисунке 2.5 показана структура аппаратуры прерываний с 24-уровневыми запросами на входе. Принцип функционирования блока аналогичен предыдущей схеме. Применяется каскадное включение БИС с последовательным переносом:

♦ выходной сигнал “Параллельный запрет” DEP БИС БВПП с высшим приоритетом используется для формирования старшего разряда номера вектора. При использовании трех и более БИС ВН1 выходы DEP всех БИС необходимо подавать на шифратор для преобразования унарного кода номера БИС БВПП, выставившей сигнал приоритетного запроса $\sim RQINR$, в двоичный код (старшие разряды номера вектора или адреса ПА);

♦ управляемый шифратор K1804BP3 (расширитель схемы приоритетных прерываний) [4, 6, 9] управляется полями $\sim EINS$ и INS из РМК по входам $\sim EZ5$ - $EZ1$ и разрешает выдачу информации на выходы Z только по МК “Чтение регистра состояния”. Таким образом, сигналы $Z0$ и $Z1$ служат для формирования старших разрядов регистра состояния SA4-SA3 при чтении регистра состояний БВПП для сохранения в стеке, а для выбора БИС БВПП при выполнении МК “Загрузка регистра состояния” необходим дешифратор (для данного примера на два входа), выходы которого служат для формирования сигнала разрешения записи с шины SA2-SA0 в одну из выбранных БИС БВПП;

♦ остальные входы и выходы БИС соединяются по правилам организации последовательного переноса по ранее рассмотренному алгоритму.

На рисунке 2.6 приведена структурная схема УУ с блоком прерываний, работающим на микрокомандном уровне обработки прерываний. Здесь считанный из ПА адрес микрокоманды начала подмикропрограммы (ПМП) обработки прерывания подается на входы данных D11-D0 блока управления последовательностью микрокоманд (БУПМ) K1804BY4. Обработка прерываний на микрокомандном уровне используется в основном в системах повышенной “живучести”, а в качестве источников прерываний выступают сигналы ошибок от схем контроля.

Заметим, что организация процедуры обработки прерываний на микрокомандном уровне без введения дополнительных микрокоманд (для анализа сигнала $\sim RQINR$) выполняется не после каждой МК, а только для безусловных микрокоманд (например, CONT или при $\sim CCE=1$), так как в одном такте невозможно организовать проверку двух условий перехода (сигналу $\sim RQINR$ и условию перехода) по входу $\sim CE$ в БУПМ BY4. В результате микропрограммы с системой преры-

ваний на микрокомандном уровне будут состоять только из условных микрокоманд.

С точки зрения структурной организации системы прерываний на микрокомандном уровне в структуру УУ процессора необходимо ввести дополнительный быстродействующий стек (на схеме не показан), в котором необходимо сохранять содержимое внутренних регистров БУПМ ВУ4 и ОУ (регистр/счетчик адреса, счетчик микрокоманд, регистр признаков ОУ, все программно недоступные регистры, значения которых могут быть изменены при выполнении подмикропрограммы обработки прерывания, и регистр состояний БВПП RS при организации вложенности обработки запросов).

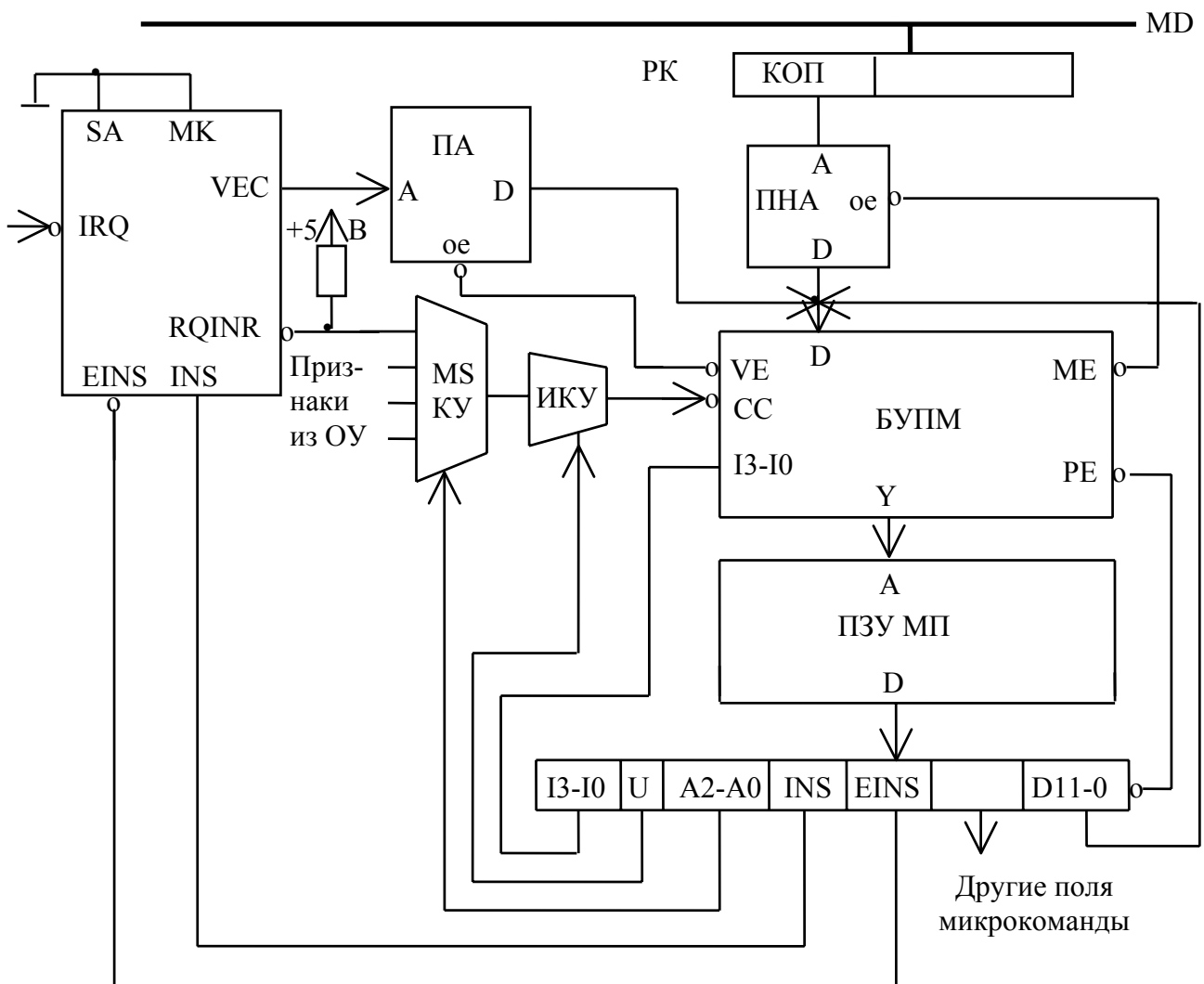


Рисунок 2.6 - Структурная схема БВПП на микрокомандном уровне обработки прерываний без вложенности

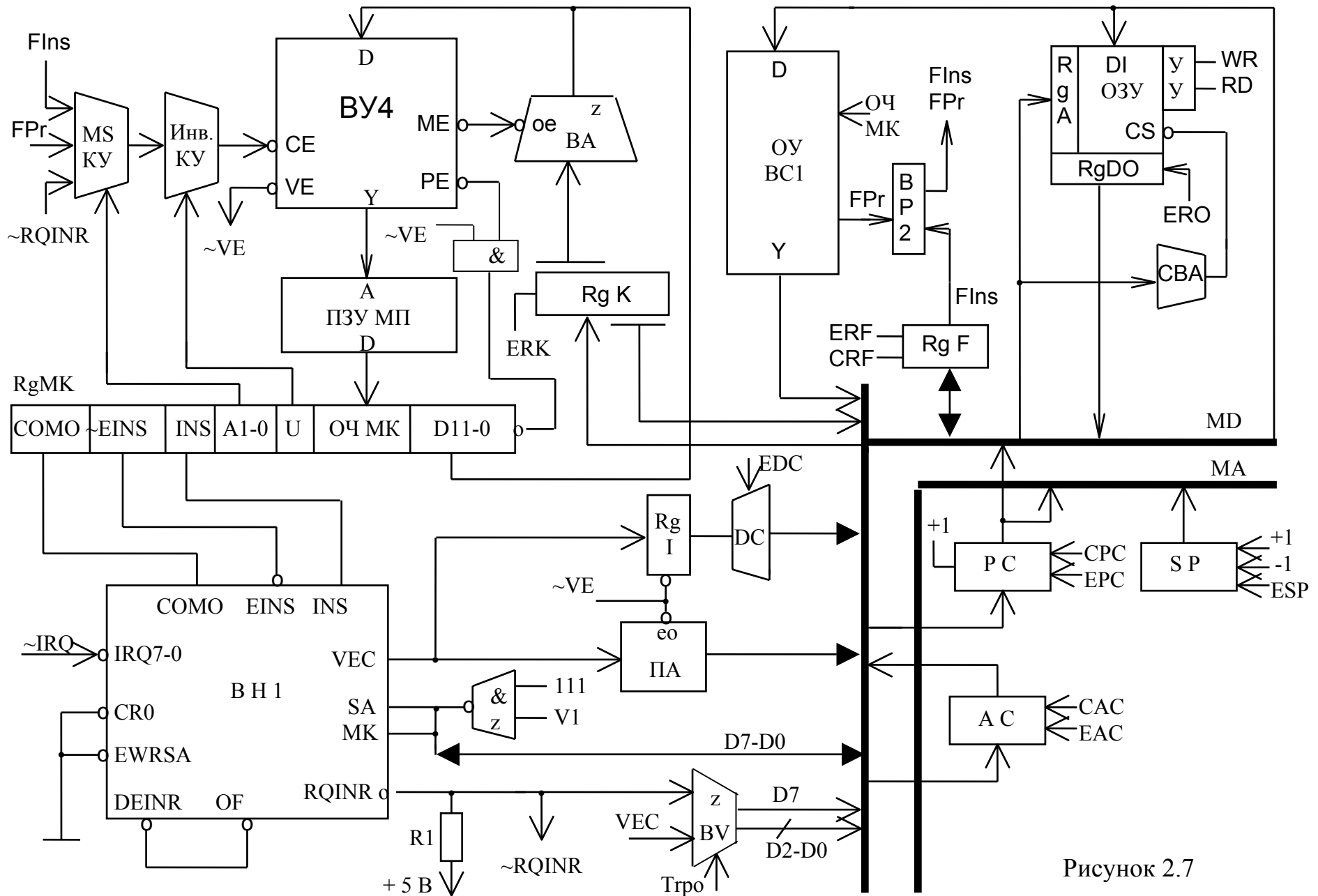


Рисунок 2.7

Обычно обработка прерываний на микрокомандном уровне реализуется без вложенности запросов на прерывание, так как требует сохранения в стеке достаточно больших объемов данных и включает небольшое число источников прерываний, которые не требуют приоритетной обработки.

Следует обратить внимание, что выходы ПА при организации системы прерываний на командном уровне связаны с МД или МА микропроцессора, а на микрокомандном уровне с внутренней шиной адреса БУПМ ВУ4.

2.4 Структура процессора с системой прерываний

На рисунке 2.7 приведена структурная схема процессора на основе МПК К1804 с системой прерываний на командном уровне.

Структурная схема лабораторной установки включает ОЗУ для хранения программ, данных и организации стека, указатель стека SP, программный счетчик PC, регистр слова состояния процессора RgF и регистр команд RgK. Для обработки данных и вычисления исполнительных адресов (Аисп) в структуру процессора входит операционное устройство (ОУ) ВС1 со схемой управления состоянием и сдвига ВР2 для формирования признаков FPr и флагов регистра состояний процессора FIns. Т.к. обработка прерываний не требует выполнения преобразований в ОУ, то его работа не моделируется, а простейшие преобразования по передаче данных выполняются через АС, который для наглядности вынесен из состава регистров общего назначения ВС1 в отдельный узел.

Управление работой процессора осуществляется со стороны УУ на основе блока микропрограммного управления (БМУ), который имеет классическую структуру [2, 4, 6, 9] и включает блок управления последовательностью микрокоманд (БУПМ) на основе ВУ4, ПЗУ микропрограмм, регистр микрокоманд RgМК, мультиплексор и инвертор кода условий (MSKY и Инв.КУ), а также буферную схему с выходом на 3 состояния для развязки по монтажному "ИЛИ" выходов поля D11 - D0 RgМК и кода операций из RgK.

Двунаправленные шины маски МК и регистра состояний SA соединены с шиной данных процессора, что позволяет производить загрузку с шины данных микроЭВМ в регистр маски RM или в регистр состояний RS данных либо из АС,

либо ОЗУ, а также считывать состояние RM и RS с помощью соответствующих микрокоманд либо команд процессора.

Целью лабораторной работы является получение навыков в составлении микропрограмм для организации системы прерываний для различных дисциплин обслуживания:

- ◆ фиксированных приоритетов с вложенностью обработки запросов IRQ;
- ◆ специального маскирования, когда любой из вновь поступивших запросов IRQ_i может прервать обработку текущей ППОП, кроме замаскированных запросов и находящихся в обработке (или как модификация дисциплины, когда вновь поступивший запрос может прерывать выполнение и своей собственной ППОП);
- ◆ обработка прерываний без вложенности, когда ни один из запросов даже с более высоким приоритетом не может прервать выполнение текущей ППОП;
- ◆ программного опроса, когда выход сигнала прерывания $\sim RQINR$ может опрашиваться только из программы пользователя и пользователь сам принимает решение о необходимости обработки данного запроса и выполняет все необходимые действия по сохранению информации в стеке и переходу на ППОП. При этом в систему команд процессора необходимо ввести дополнительную команду “Чтение слова состояния БВПП”, по которой с выходов буферной схемы BV в АС фиксируется состояние аппаратуры прерывания на текущий момент времени в формате:

\overline{RQINR}	X	X	X	X	V	E	C
--------------------	---	---	---	---	---	---	---

где в старшем бите байта фиксируется состояние выхода сигнала запроса на прерывание $\sim RQINR$ (D7), а в трех младших разрядах (D2-D0) номер вектора прерывания с наивысшим текущим приоритетом. То есть, если бит $D7 = \sim RQINR = 0$, то значит три младших разряда указывают на номер вектора, вызвавшего выработку сигнала прерывания, иначе ($D7 = 1$) запросов на прерывание нет и можно продолжать выполнение основной программы.

Для реализации алгоритма входа в прерывание в режиме специального маскирования необходимо запрещать обработку прерываний, находящихся в обработке. Для этой цели в структуру входит дополнительный регистр RgI для запоминания номера вектора, принятого на обработку при выполнении МК05 "Чтение вектора", а в дальнейшем декодированный код вектора (с помощью DC) присоединяется к содержимому RM БВПП по МК11 и сохраняется в стеке для очистки RM при возврате из прерывания по МК10.

Для реализации режима программного опроса (ПО) необходимо выход сигнала запроса на прерывание $\sim RQINR$ БВПП отключить от MSKY и считывать в АС по шине данных. При этом на магистрали данных (MD) в режимах аппаратных прерываний в бите D7 при обмене данными дополнительно формируется значение сигнала $\sim RQINR$ по монтажному "ИЛИ". Для его отключения в состав ЛУ введена буферная схема с выходами на 3 состояния, которая отключает выход сигнала $\sim RQINR$ и выходы VEC от MD в режимах аппаратных прерываний. Для реализации режима программного опроса необходимо предварительно установить триггер программного опроса в "1", а выходы БВПП шины VEC дополнительно выводятся на MD также через буферную схему на три состояния BV.

Для формирования константы нуля при загрузке в регистр состояния RS БВПП на входах шины SA введена схема "И" с выходами на 3 состояния, на выходе которой формируется код "0" при выполнении МК09 "Загрузка в регистр состояний" БВПП, если магистраль данных находится в третьем состоянии.

Контрольные вопросы для самопроверки

1. Какую информацию необходимо сохранять в стеке при обслуживании запросов на прерывание на микрокомандном и командном уровнях?
2. В какие моменты времени возможен анализ запроса на прерывание $\sim RQINR$ на микрокомандном уровне?
3. Для каких целей в БВПП могут использоваться микрокоманды установки и сброса регистров маски и состояния, разрешение и запрещение прерываний при обслуживании запросов на командном уровне?
4. Для каких целей в состав БВПП включен регистр состояний?

5. В какой области памяти размещаются таблица векторов прерываний и ППОП? Перечислите варианты формирования начальных адресов ППОП.
6. Как на основе БВПП ВН1 организовать обработку прерываний без вложенности с минимальным временем реакции на прерывание и с динамически изменяемыми приоритетами?
7. Как реализуется режим программного опроса на основе БВПП ВН1? Перечислите основные этапы выполнения команды "Чтение слова состояния БВПП".

3 Обработка прерываний в МПК K580 и K1810

3.1 Программируемый контроллер прерываний K1810ВН59А

Программируемый контроллер прерываний (ПКП) K580ВН59 и K1810ВН59А реализуют векторную 8-уровневую систему прерываний и позволяют расширить число обслуживаемых запросов до 64 путем каскадного включения нескольких БИС контроллеров (один ведущий ПКП и до 8 ведомых).

ПКП K1810ВН59А является модификацией контроллера K580ВН59 и выполняет ряд дополнительных функций, главная из которых - обслуживание прерываний для МкПр ВМ86/88 и более старших моделей микропроцессоров, а также в ПКП устранены некоторые недостатки, присущие ПКП ВН59. ПКП ВН59А является программно настраиваемым на работу как с МПр ВМ80, так и с ВМ86.

БИС ПКП ВН59А может работать в нескольких режимах (дисциплинах обслуживания приоритетов), которые устанавливаются программным путем, причем приоритеты могут быть изменены в процессе выполнения программ.

ПКП выполняет следующие функции:

- ◆ фиксацию запросов на прерывание от 8 внешних источников;
- ◆ программное маскирование поступивших запросов;
- ◆ выделение приоритетного запроса по отношению к приоритету обслуживаемого прерывания с целью выявления возможности прерывания;
- ◆ присвоение фиксированных или циклически изменяемых приоритетов входам контроллера, на которые поступают запросы;

- ♦ программный опрос внешних устройств для определения, нуждается ли устройство в обмене;
- ♦ формирование номера вектора прерывания, а на его основе:

1. для микропроцессора K580BM80 выдачу кода операции команды CALL (перехода на подпрограмму) и 16-разрядного адреса таблицы переходов для этой подпрограммы;

2. для микропроцессора K1810BM86 и более старших моделей выдачу адреса (вектора) таблицы IDT векторов прерываний.

На рисунке 3.1 приведено условное графическое обозначение контроллера, в таблице 3.1 - назначение выводов БИС, а на рисунке 3.2 показаны структурная схема ПКП а) и схема блока обработки приоритетов б).

Таблица 3.1 - Назначение выводов БИС BH59A

18 —	IRQ0	PIC BH59A	↔ Z	11	Обозначение	Назначение
19 —	IRQ1		D0	10	~CS	Вход выбора микросхемы
20 —	IRQ2		D1	09	~WR	Сигнал записи с шины данных
21 —	IRQ3		D2	08	~RD	Сигнал чтения на шину данных
22 —	IRQ4		D3	07	D7-D0	Двунаправленная шина данных
23 —	IRQ5		D4	06	CAS2-CAS0	Двунаправленная шина каскадирования
24 —	IRQ6		D5	05	~SP/~EN	Выбор ведомой микросхемы/ Разрешение данных
25 —	IRQ7		D6	04	INT	Выход сигнала прерывания
02 —	WR		↔ Z	12	IRQ7-IRQ0	Входы запросов прерывания
03 —	RD		CAS0	13	~INTA	Сигнал подтверждения прерывания
27 —	A0		CAS1	15	A0	Адресный вход порта BB
26 —	INTA		CAS2	17		
01 —	CS		INT			
16 —	SP/EN					

Рисунок 3.1 - Условное графическое обозначение БИС ПКП K1810BH59A

В состав контроллера входят: 8-разрядный буфер шины данных, схема управления чтением/записью (входы ~CS, ~RD, ~WR, A0), регистры управляющих слов, схема управления (вход ~INTA, выход INT), схема каскадирования (входы/выходы ~EN/~SP и CAS2-CAS0), регистр запросов на прерывание IRR (входы IRQ7-IRQ0), схема обработки приоритетов, регистр обслуживания запросов ISR, регистр маскирования запросов на прерывание IMR.

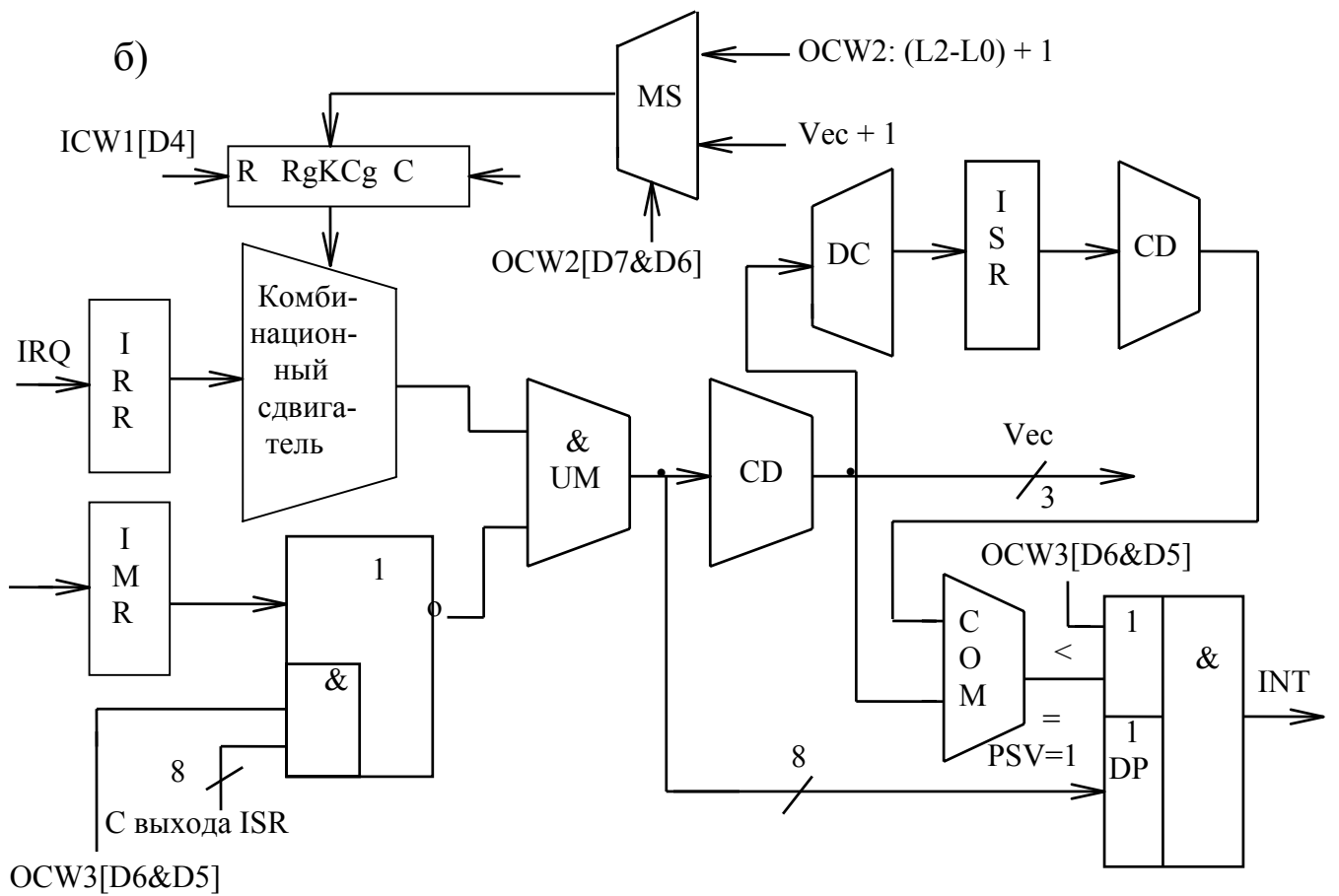
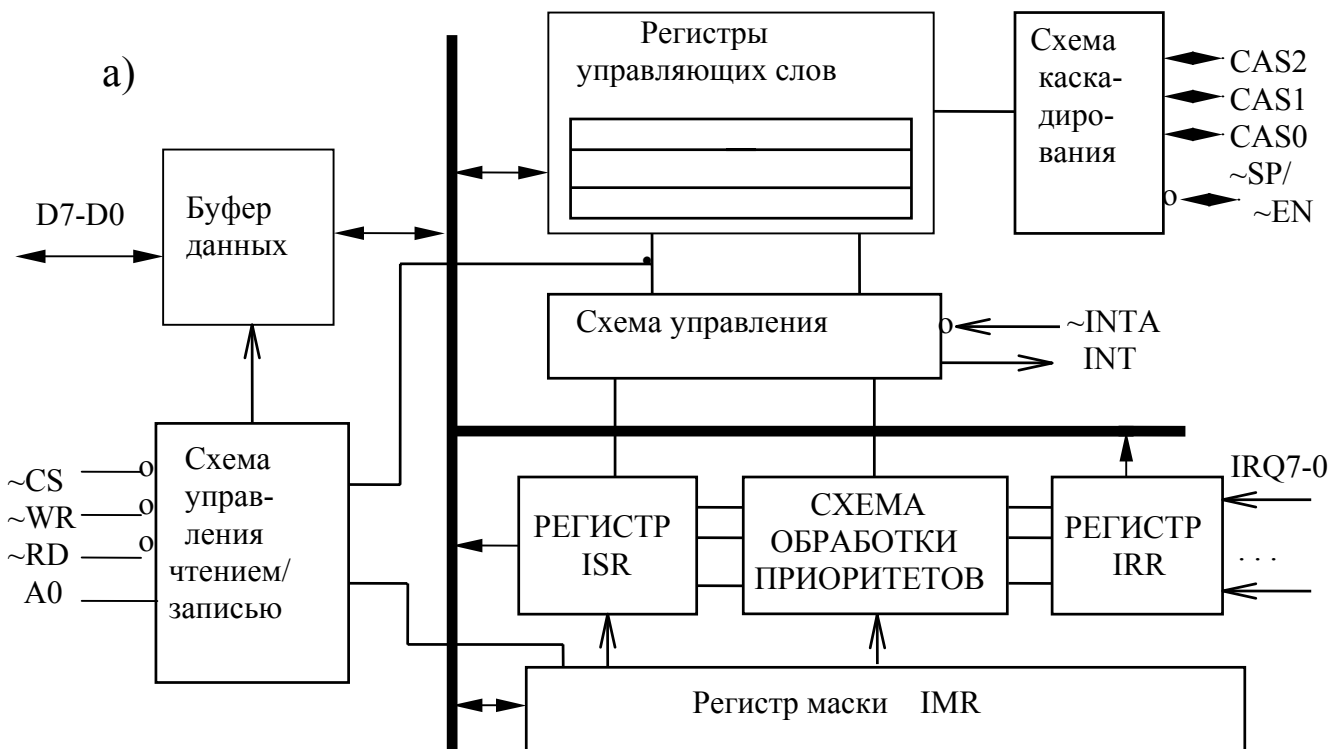


Рисунок 3.2 - Структурная схема ПКП а) и функциональная схема блока обработки приоритетов ПКП б)

Буфер шины данных, представляющий собой двунаправленный 8-разрядный буфер с входами/выходами на три состояния, используется для записи в схему управляющих слов и выдачи информации о состоянии схемы, содержимого внутренних регистров и команды CALL или адреса вектора прерывания.

Схема управления чтением/записью управляет режимами записи управляющих слов в ПКП и считывания из схемы информации на магистраль данных D7-D0. Регистры управляющих слов предназначены для хранения кодов команд при программировании (ICW и OCW), которые устанавливаются в исходное состояние БИС ПКП, режим обработки прерываний (дисциплину обслуживания) и фиксированный статус уровней приоритета.

Схема каскадирования используется при каскадном включении нескольких БИС ВН59А. По линиям CAS2-CAS0 ведущая БИС передает код, а ведомые схемы по этим же цепям принимают и сравнивают этот код с собственным номером.

Схема управления предназначена для выдачи сигнала прерывания INT после поступления одного или нескольких сигналов запроса IRQ и приема двух или трех сигналов “Подтверждения прерывания” ~INTA из МПр для выдачи адреса вектора прерывания таблицы IDT или трехбайтовой команды CALL.

Регистр запроса прерываний IRR предназначен для приема и фиксации поступивших запросов IRQ. Регистр маски IMR служит для записи в него маски прерываний, позволяющей маскировать определенные разряды IRR, давая возможность сигналам с более низким приоритетом прерывать работу МПр.

Схема обработки приоритетов с учетом значения кода в IMR и запрограммированным режимом работы выделяет самый приоритетный запрос и записывает его в регистр обслуживания прерывания ISR (регистр состояния). Таким образом, регистр ISR хранит только номера запросов IRQ, находящихся в обработке в текущий момент времени в случае вложенных прерываний. Сброс бита ISR выполняется текущим словом управления OCW2 по окончании обслуживания запроса, а бита IRR - автоматически.

Запросы, поступающие на входы IRQ7-IRQ0, фиксируются в регистре запросов IRR в двух режимах: по переднему фронту сигнала IRQ и по уровню сигнала IRQ (с динамическим и статическим входом записи).

ПКП может находиться в трех основных состояниях: программирования (настройки или инициализации), приема (фиксации) и обслуживания запросов на прерывание.

В режиме программирования осуществляется загрузка управляющих слов двух видов (при $\sim CS=0$ и $\sim WR=0$): инициализации (Initialization Command Word) ICW и операционных (Operation Command Word) OCW по командам OUT по портам ввода-вывода из МПр. ПКП может быть запрограммирован на следующие базовые режимы обслуживания запросов:

1. Режим полностью установленных приоритетов прерывания (приоритеты внешних устройств фиксированы).
2. Циклического сдвига приоритетов:
 - ◆ циклического сдвига приоритетов в режиме А;
 - ◆ циклического сдвига приоритетов в режиме В.
3. Специального маскирования приоритетов.
4. Последовательного опроса (программного опроса приоритетов).

После поступления запросов IRQ и выделения приоритетного запроса схема управления ПКП формирует сигнал запроса на прерывание INT, который поступает на вход INT микропроцессора, и если в микропроцессоре разрешены прерывания, то он в режиме обслуживания запросов (при $\sim CS=1$) отвечает контроллеру сигналом подтверждения прерывания, поступающим на вход $\sim INTA$. Принцип работы ПКП зависит от типа микропроцессора. Для МПр K580BM80 по первому сигналу $\sim INTA1$ на шину данных из ПКП выдается код операции команды CALL (11001101) перехода на подпрограмму (рисунок 3.3а), а по двум остальным сигналам $\sim INTA2$ и $\sim INTA3$ выдает второй и третий байты команды CALL, содержащие адрес подпрограммы обработки прерывания, т.е. источники прерываний различаются по начальным адресам подпрограмм. Кроме того, по $\sim INTA3$ сбрасывается в “0” разряд принятого на обработку запроса в IRR и устанавливается в “1” соответствующий разряд регистра обслуживания ISR, который идентифицирует, какой из запросов на прерывание обслуживается (принят на обработку).

Для МПр K1810BM86 в ответ на поступление сигнала IRQ по его фронту информация фиксируется в IRR и с некоторой задержкой ПКП выставляет сигнал запроса INT (рисунок 3.3б). В ответ МкПр генерирует два сигнала подтверждения

прерывания $\sim\text{INTA}$. В ответ на первый $\sim\text{INTA1}$ по фронту сигнала ПКП устанавливает в "1" соответствующий выделенному приоритетному запросу бит в ISR регистре и сбрасывает в "0" этот же бит в регистре IRR, а в случае каскадного включения по срезу $\sim\text{INTA1}$ выдает код ведомого ПКП на линии CAS. Во время $\sim\text{INTA2}$ ведущий или ведомый ПКП выдает на шину данных 8-разрядный адрес вектора прерывания по срезу сигнала в формате:

D7	D6	D5	D4	D3	D2	D1	D0
A15	A14	A13	A12	A11	Номер вектора		

. 0 0 ,

который используется для вычисления адреса входа в таблицу векторов прерываний IDT путем его сдвига в микропроцессоре на два разряда влево, а по фронту второго $\sim\text{INTA2}$ снимает сигнал запроса INT и переводит выходы CAS ПКП в третье состояние.

Потенциальный вход IRQ UP=1 требует удержания "1" на входах IRQ до окончания первого $\sim\text{INTA}$ и установку в "0" на входах IRQ к моменту разрешения прерывания, иначе произойдет повторный захват (рисунок 3.3в) этого же запроса на прерывание, так как первый сигнал $\sim\text{INTA}$ по срезу сигнала защелкивает информацию со входов IRQ в IRR, а по фронту - сбрасывает в IRR бит принятого к обработке запроса IRQ и устанавливает в "1" соответствующий бит в ISR регистре, а второй сигнал $\sim\text{INTA}$ вновь защелкнет входной сигнал IRQ в регистре IRR.

Временная диаграмма ложного повторного срабатывания по входу IRQ приведена на рисунке 3.4.

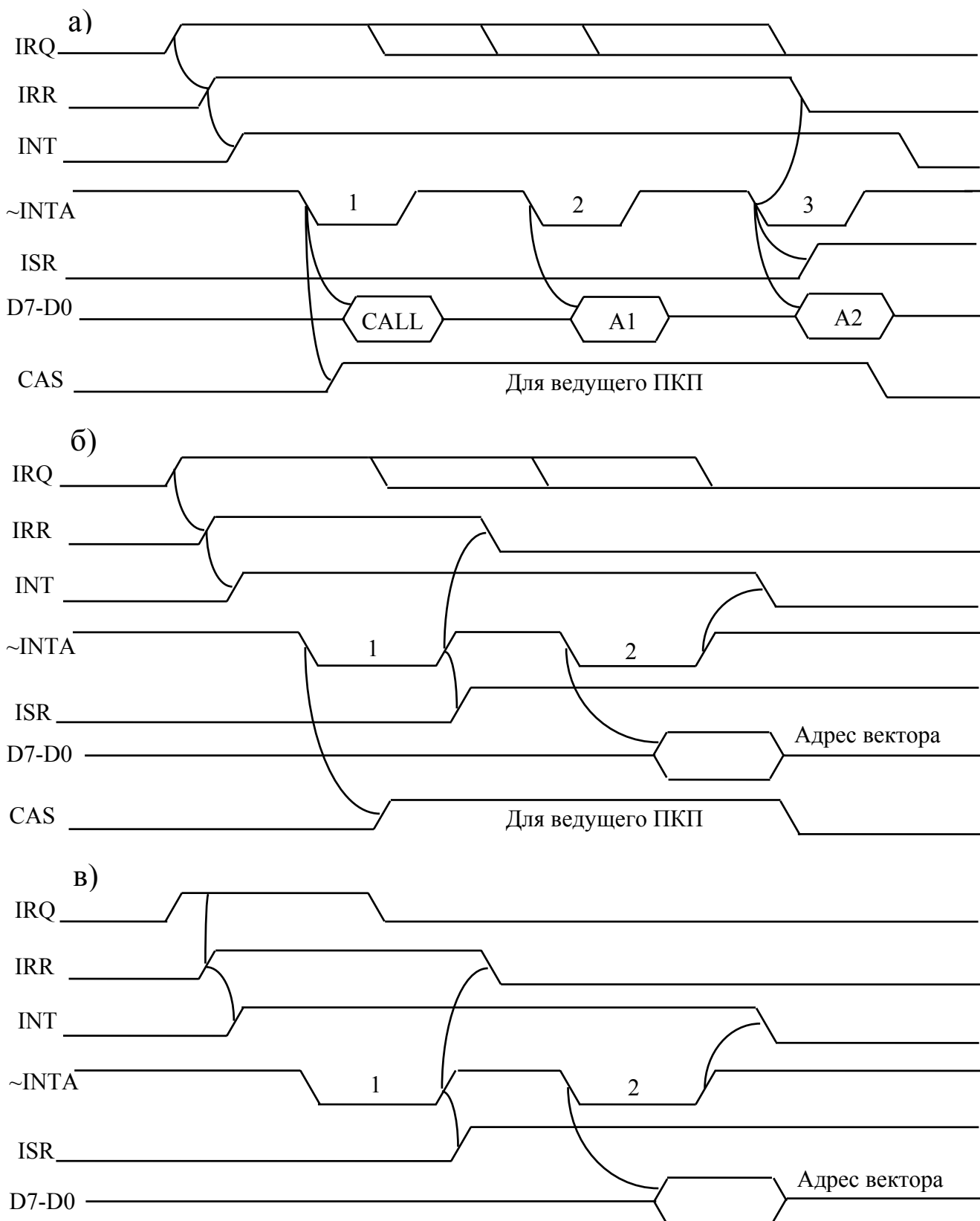


Рисунок 3.3 - Временные диаграммы работы ПКП в режиме обслуживания:

а) для Мпр серии К580 при UP=0, б) для МПр серии К1810 при UP=0,

в) для МПр серии К1810 при UP=1

ряд регистра обслуживания ISR, фиксирующий окончание процедуры обработки прерывания.

а) Старший байт адреса

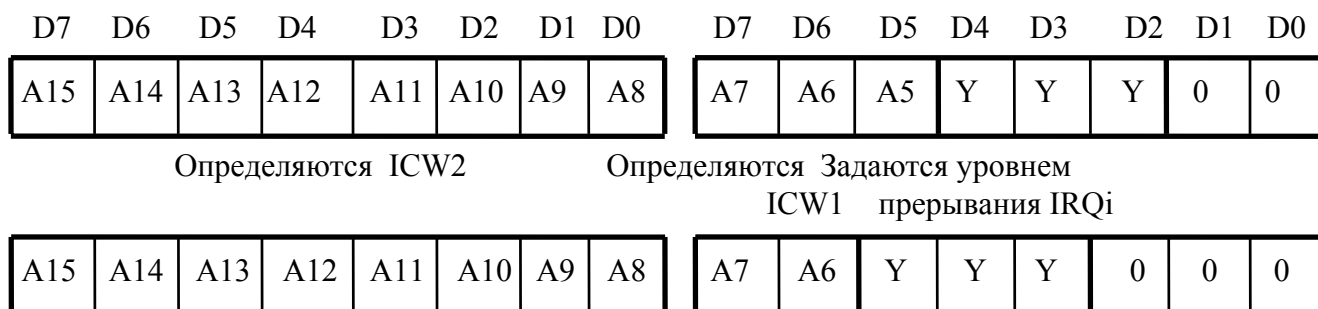
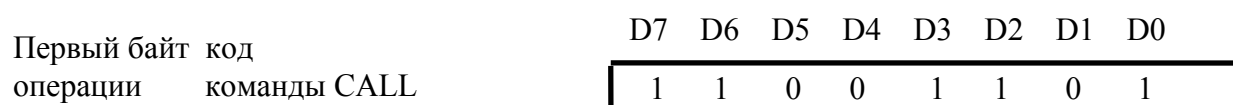


Рисунок 3.5 - Форматы начальных адресов ППОП при различном шаге вектора



Второй байт: младшие разряды начального адреса ППОП A1

Входы запросов	Интервал адресов вектора = 4 D7 D6 D5 D4 D3 D2 D1 D0								Входы запросов	Интервал адресов вектора = 8 D7 D6 D5 D4 D3 D2 D1 D0							
IRQ7	A7	A6	A5	1	1	1	0	0	IRQ7	A7	A6	1	1	1	0	0	0
IRQ6	A7	A6	A5	1	1	0	0	0	IRQ6	A7	A6	1	1	0	0	0	0
IRQ5	A7	A6	A5	1	0	1	0	0	IRQ5	A7	A6	1	0	1	0	0	0
IRQ4	A7	A6	A5	1	0	0	0	0	IRQ4	A7	A6	1	0	0	0	0	0
IRQ3	A7	A6	A5	0	1	1	0	0	IRQ3	A7	A6	0	1	1	0	0	0
IRQ2	A7	A6	A5	0	1	0	0	0	IRQ2	A7	A6	0	1	0	0	0	0
IRQ1	A7	A6	A5	0	0	1	0	0	IRQ1	A7	A6	0	0	1	0	0	0
IRQ0	A7	A6	A5	0	0	0	0	0	IRQ0	A7	A6	0	0	0	0	0	0

Третий байт - старшие разряды адреса вектора прерывания A2

A15	A14	A13	A12	A11	A10	A9	A8
-----	-----	-----	-----	-----	-----	----	----

Рисунок 3.6 - Формат команды CALL, выдаваемой ПКП

3.2 Программирование ПКП

Для программиста контроллер представляет собой два порта ввода-вывода, адреса которых отличаются в разряде A0. Установка ПКП в исходное состояние и установка режима обслуживания прерываний осуществляется с помощью двух типов команд, записываемых в ПКП: управляющих слов инициализации ICW1-

ICW4 и слов управления OCW1-OCW3, которые загружаются после окончания инициализации и служат для:

- ◆ загрузки маски запросов на прерывание;
- ◆ установки статуса уровней приоритета;
- ◆ фиксации конца обработки прерывания;
- ◆ установки режима специального маскирования или программного опроса, а также назначения источника для чтения в микропроцессор.

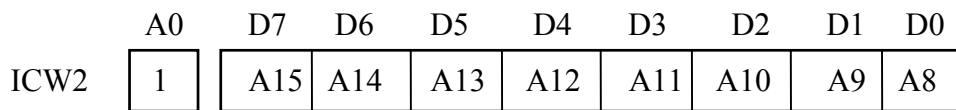
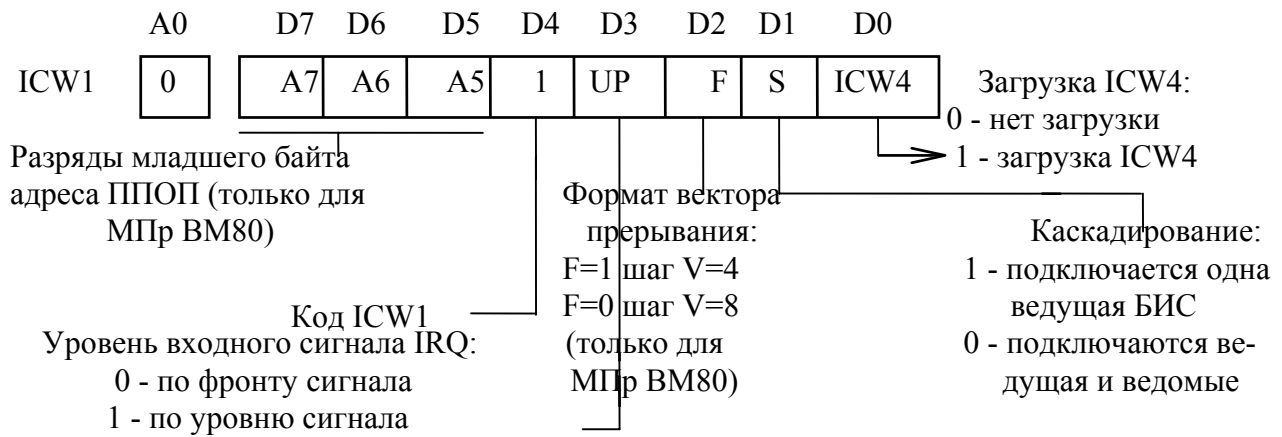
При каскадном включении нескольких БИС ПКП каждая БИС программируется независимо друг от друга, причем эти БИС могут работать в различных режимах обслуживания прерываний.

Прежде всего при $A0=0$ по команде OUT из МП_р загружается управляющее слово ICW1 (рисунок 3.7), где $D4=1$ является кодом операции данного управляющего слова. По команде ICW1 ПКП выполняет следующие действия:

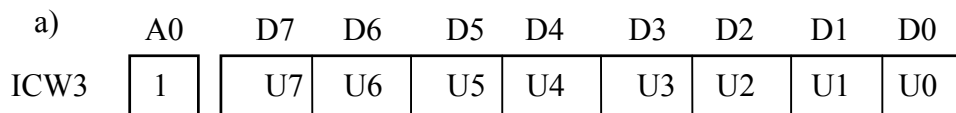
- ◆ устанавливает в исходное состояние схему на режим фиксированных приоритетов путем сброса всех внутренних регистров ПКП: IRR, ISR, IMR, управляющих слов ICW2, ICW3, ICW4, OCW2, OCW3;
- ◆ входу IRQ₇ присваивается уровень приоритета 7 (низший уровень);
- ◆ в формате управляющего слова OCW3 снимается режим специального маскирования и источником чтения назначается регистр IRR.

Команда ICW1 имеет 16 модификаций, что связано с указанием в ней дополнительных признаков:

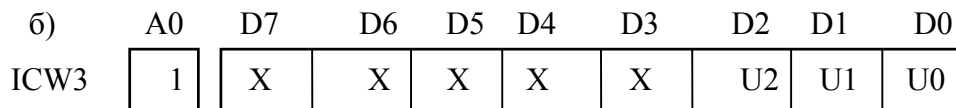
- ◆ в разряде D0 указывается признак необходимости загрузки управляющего слова ICW4. Если $D0=0$, то ICW4 в процессе инициализации ПКП не загружается, иначе последним должно быть загружено ICW4;
- ◆ в бите D1 - признака числа ПКП (S) в аппаратуре прерывания. Если $S=1$, то подключается одна (только ведущая - автономный режим работы) БИС ПКП, иначе в систему входит несколько БИС, включенных каскадно (одна ведущая, остальные ведомые);



Старший байт адреса подпрограммы обработки прерывания (для МП_р VM80) или адрес таблицы IDT (для МП_р VM86)



Для ведущего "1" в разряд U_i ставится в том случае, если к соответствующему входу IRQ_i ведущего подключается выход INT ведомого



Для ведомого U2-U0 указывает приоритет ведомого (номер входа IRQ_i ведущего, к которому подключается выход INT ведомого)

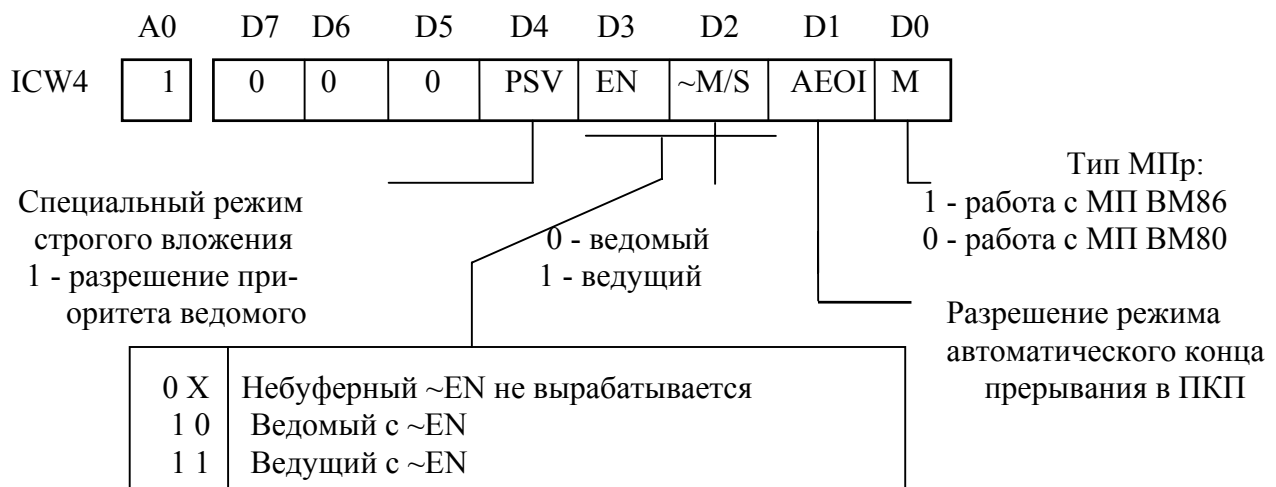


Рисунок 3.7 - Форматы управляющих слов инициализации ПКП

- ◆ в бите D2 указывается признак формирования адреса (шаг адресов векторов ППОП). Это поле используется только при работе с МПр K580BM80. Если D2=1, то разность между адресами векторов равна 4 и из формата ICW1 для формирования младшего байта адреса команды CALL используются биты A7-A5, иначе шаг векторов равен 8, а для формирования младшего байта адреса ППОП используются биты A7-A6 ICW1;
- ◆ значение бита D3 определяет признак типа воспринимаемых запросов IRQ7-IRQ0 (поле UP): если D3=0 - IRQ воспринимается фронтом сигнала из "0" в "1" для внутренних прерываний, если D3=1 - IRQ воспринимается уровнем "1" для внешних источников прерываний. Т.е. для UP=0 регистр IRR работает в режиме асинхронных RS-входов по фронту, а при UP=1 как регистр "защелка" по уровню входного сигнала;
- ◆ бит D4 для ICW1 всегда равен "1" и, как было отмечено, является кодом операции данного управляющего слова, а в битах D7-D5 указывается значение старших разрядов младшего байта адреса ППОП, которые используются только при работе с микропроцессором K580BM80.

Следующей за командой ICW1 при A0=1 всегда загружается команда ICW2 (рисунок 3.7), в которой указываются старший байт A15-A8 адреса перехода на ППОП по команде CALL для МПр K580BM80, а для МПр K1810BM86 старшие разряды A15-A11 адреса таблицы векторов прерываний IDT.

Если в ICW1 бит D1=0, то следующей за ICW2 при A0=1 ПКП воспринимается как команда ICW3 (только для каскадного включения БИС ПКП), которая имеет две модификации (рисунок 3.7). В ведущую БИС ПКП загружается ICW3 формата а), а во все ведомые - формата б). В разрядах D7-D0 ICW3 формата а) "1" ставится в том случае, если к соответствующему входу IRQ_i ведущего ПКП подключается выход INT ведомого. Нулевое состояние бит означает, что к соответствующему входу IRQ подключено периферийное устройство (источник прерывания) либо этот вход не используется.

В разрядах D2-D0 ведомой БИС команды ICW3 формата б) указывается номер входа запроса IRQ_i ведущего ПКП, к которому подключен выход INT данной ведомой микросхемы. Например, если выход INT ведомой БИС подключен ко

входу IRQ5 ведущей, то в разрядах D2-D0 команды ICW3 ведомой указывается код XXXXX101, а в ведущую БИС загружается команда ICW3 формата 00100000. Таким образом, для ведущего ПКП код команды ICW3 используется для сообщения ведомой БИС идентификационного номера запроса IRQ, вызвавшего сигнал запроса прерывания INT, который выдается на шину CAS2-CAS0, а ведомые ПКП принимают этот номер и сравнивают со своим собственным (кодом команды ICW3 формата б)) и по сигналам \sim INTA2, \sim INTA3 для МПр серии K580 или по \sim INTA2 для МПр серии K1810 на шину данных выдается адрес ППОП или адрес таблицы векторов IDT той БИС ПКП, которой принадлежит сигнал IRQ_i, вызвавший выработку запроса на прерывание INT в микропроцессор.

Если в ICW1 бит D1=1, то загрузка команды ICW3 не производится.

При A0=1 следующей в ПКП будет восприниматься команда ICW4, если в ICW1 бит D0=1, указывающий на необходимость загрузки данного управляющего слова. В формате команды ICW4 бит D0 используется для указания, с каким типом микропроцессора работает ПКП: если D0=0, то используется МПр серии K580, а если D0=1, то МПр серии K1810.

Также в этой команде устанавливается признак автоматического конца прерывания (бит D1=1) AEOI (только для ведущего ПКП), при котором сброс бита в ISR регистре сигнала запроса на прерывание IRQ выполняется автоматически без загрузки управляющего слова OCW2 после третьего или второго сигнала \sim INTA в зависимости от типа МПр.

Для МПр серии K1810 в бите D2 команды ICW4 указывается назначение БИС ПКП: при D2=1 БИС является ведущей, а при D2=0 - ведомой. Также в этой команде для МПр K1810BM86 указывается признак разрешения выдачи из ПКП данных \sim EN (бит D3=1), при установке которого вывод \sim EN/ \sim SP становится выходом. Если в команде ICW4 поле \sim EN=1, то на нем формируется сигнал низкого уровня каждый раз, когда на шину данных из ПКП выдается информация (чтение слова состояния, IMR, IRR, ISR, адрес вектора таблицы IDT). Выходной сигнал \sim EN ведущего ПКП подается в контроллер шины МПр ВГ88 для переключения шинного формирователя ВА86. Для МПр BM86 выработка сигнала \sim EN является обязательной, иначе выдаваемый вектор по второму сигналу \sim INTA не будет

с МД принят в микропроцессор, т.к. шинный формирователь контроллера шины будет находиться в третьем состоянии.

В ПКП ВН59А устранен недостаток нарушения режима вложенности обработки запросов при каскадном включении от ведомых ПКП. Для установки режима строгого упорядочения приоритетов при каскадном включении в команду ICW4 введен бит D4, при установке которого в "1" в ведущем ПКП снимается требование строгого упорядочения приоритетов и разрешается прием новых запросов со входов IRQ, только что принятого к обслуживанию (или еще называют разрешение приоритета ведомого), если в ведущем ПКП бит AEОI=0. Пример использования режима строгого упорядочения приоритетов при каскадном включении будет рассмотрен ниже.

Если в команде ICW1 бит D0=0, то команда ICW4 не загружается и ПКП переходит в состояние обслуживания запросов на прерывание - в режим фиксированных приоритетов, при котором приоритеты запросов IRQ7-IRQ0 располагаются в порядке от 0 (высший) для IRQ0 до 7 (низший) для IRQ7. Дальнейшее управление схемой, выбор и изменение алгоритма (дисциплины) обслуживания запросов осуществляется с помощью управляющих слов OCW1-OCW3. На рисунке 3.8 приведен алгоритм последовательности загрузки управляющих слов при инициализации ПКП.

Заметим, что в формате команд ICW2, ICW3 и ICW4 нет разрядов кода операции SW и все они загружаются при значении порта ввода-вывода A0=1. Поэтому ПКП воспринимает эти команды в строгом соответствии с последовательностью их загрузки и нарушение этой последовательности может привести к неверной интерпретации этих команд при инициализации. Например, если в команде ICW1 бит D0=0, а D1=1, то загрузка команд ICW3 и ICW4 будет интерпретирована ПКП как значение маски прерываний и эта маска будет автоматически загружена в IMR (сначала ICW3, а затем ICW4).

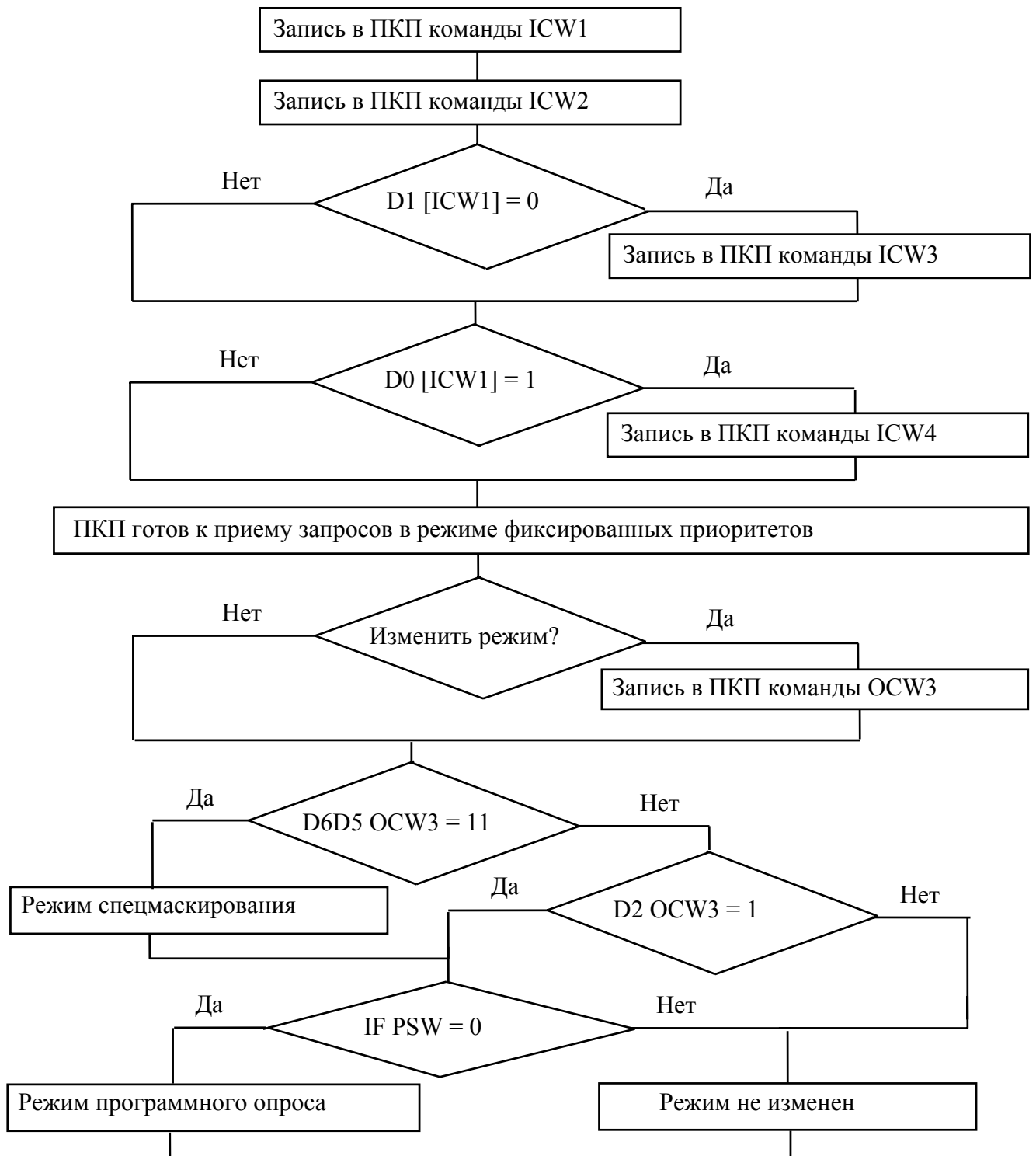


Рисунок 3.8 - Последовательность записи управляющих слов при инициализации и задании режимов работы ПКП

После инициализации некоторые из входов запросов на прерывание IRQ могут быть замаскированы. Для этого по адресу A0=1 в ПКП записывается OCW1 - байт с единицами в разрядах, соответствующих входам, по которым запросы не должны обслуживаться (рисунок 3.9).

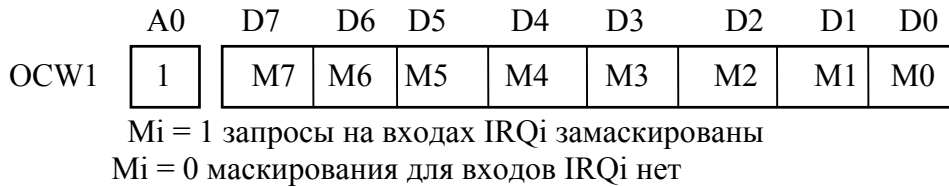


Рисунок 3.9 - Формат слова OCW1 для записи в регистр маски

Рассмотрим режимы работы ПКП.

3.2.1 Режим фиксированных приоритетов

В данном режиме приоритеты входов запросов на прерывание IRQ7-IRQ0 являются неизменными. Если одновременно поступают несколько запросов IRQ, то обслуживается запрос с наивысшим приоритетом (с меньшим номером). Если во время выполнения ППОП по одному из входов IRQ поступит запрос с более высоким приоритетом, то ПКП прерывает выполнение ППОП и переходит к выполнению ППОП с более высоким приоритетом.

Например, одновременно на входы IRQ3 и IRQ6 поступили запросы, которые устанавливают в “1” соответствующие разряды IRR. ПКП формирует сигнал запроса на прерывание INT, а МПр K1810BM86 отвечает выдачей двух сигналов “Подтверждения прерывания” $\sim INTA$. По $\sim INTA1$ в ПКП в регистре ISR устанавливается в “1” бит запроса с наивысшим поступившим приоритетом (бит 3), а в IRR этот же бит сбрасывается. По $\sim INTA2$ ПКП выдает на шину данных адрес таблицы векторов прерываний IDT (в дальнейшем для краткости будем называть его вектором прерывания), сопровождаемый выдачей сигнала $\sim EN$, и снимает сигнал запроса INT. МПр принимает вектор прерывания и выполняет следующую последовательность действий:

- ◆ сохраняет в стеке значение текущего слова состояния процессора SR, регистра кодового сегмента CS, программного счетчика IP и сбрасывает

бит разрешения прерываний в слове состояния SR (запрещает прерывания);

- ◆ выполняет сдвиг адреса вектора прерывания на два разряда влево, и по этому адресу из таблицы векторов прерываний IDT считываются новые значения кодового сегмента CS и программного счетчика IP и загружаются в соответствующие регистры процессора. Далее выполняется переход на ППОП.

В тексте ППОП для организации обработки вложенных прерываний программист должен разрешить обработку прерываний командой STI (установить флаг разрешения прерываний в слове состояния SR), а предпоследней командой OCW2 выполнить сброс в регистре ISR ПКП бита запроса с наивысшим приоритетом, находящегося в обработке. Последней командой ППОП должна быть команда IRET - возврата из прерывания, по которой в процессоре восстанавливается из стека информация, сохраненная при обращении к ППОП (CS, IP, SR).

В МПр ВМ80 по сигналу \sim INTA3 выполняется установка ISR-бита и сброс IRR-бита, а ПКП выдает в МПр три байта команды CALL, при выполнении которой в стеке сохраняется только значение программного счетчика IP, а сохранение слова состояния процессора возлагается на программиста. ППОП заканчивается командой RET - возврата из подпрограммы.

Таким образом, регистр обслуживания отображает текущее состояние аппаратуры прерываний, т.е. указывает на то, какие из запросов обслуживаются в данный момент времени. В нашем примере при обслуживании запроса IRQ3 запросы, поступившие на входы IRQ3-IRQ7, обслуживаться не будут, а если во время выполнения ППОП3 поступит запрос с более высоким приоритетом, например IRQ1, то он вновь вызовет формирование сигнала запроса на прерывание INT и МПр вновь ответит выработкой двух ответных сигналов “Подтверждения прерывания” \sim INTA и в соответствующем разряде регистра ISR установится “1”. При этом запрещаются прерывания с уровнем приоритета от первого до седьмого. Таким образом, запрос IRQ1 прервет выполнение ППОП3 и начинается выполнение ППОП1. После завершения выполнения ППОП1 по команде OCW2 разряд D1 регистра обслуживания ISR сбрасывается и по команде IRET выполняется возврат в прерванную ППОП3, т.е. реализуется процедура вложенности обработки преры-

ваний. Только после завершения выполнения ППОПЗ в ISR-регистре будет сброшен соответствующий бит и разрешена обработка запроса IRQ6.

Форматы OCW2 приведены на рисунке 3.10 и в таблице 3.2. Для режима фиксированных приоритетов выполняется неспецифическая команда EOI (D7-D5=001). При каскадном включении в ведомых БИС команда EOI выполняется дважды для данного ведомого и ведущего ПКП, а для запросов IRQ ведущего ПКП - только для ведущей БИС.

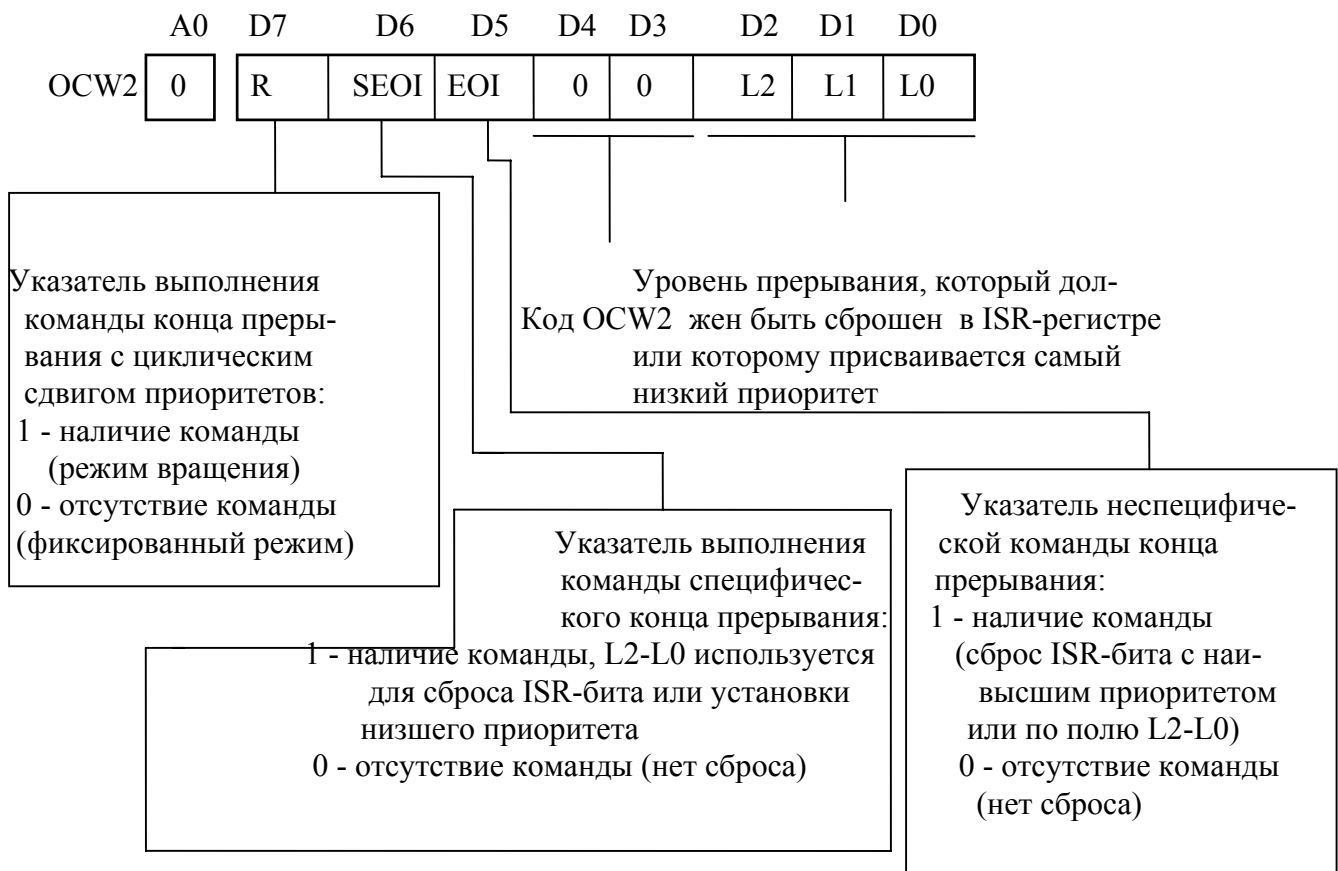


Рисунок 3.10 - Формат OCW2 для задания типа команды конца прерывания

После начальной инициализации ПКП по умолчанию высший уровень приоритета присваивается запросу с номером IRQ0, а низший - IRQ7, называемым дном приоритетного кольца. Положение дна можно программно изменять путем загрузки в ПКП управляющего слова OCW2 "Установки приоритета" формата R-SEOI (рисунок 3.10, таблица 3.2), что позволяет реализовать режим фиксированных приоритетов, при котором низший приоритет может иметь любой из восьми входов IRQ7-IRQ0.

Таблица 3.2 - Форматы команд OCW2

Команда	Содержимое бит OCW2 D7 D6 D5 D4 D3 D2 D1 D0	Описание команды
Неспецифическая (EOI)	0 0 1 0 0 0 0 0	Установка в “0” наивысшего по приоритету бита ISR регистра, установленного в “1”.
Специфическая (SEOI)	0 1 1 0 0 L2 L1 L0	Установка в “0” разряда ISR регистра, указанного в поле L2-L0.
С циклическим сдвигом (R-EOI) (режим А)	1 0 1 0 0 0 0 0	Установка в “0” наивысшего по приоритету бита ISR регистра, установленного в “1”, и присвоение ему низшего приоритета
Со специфическим циклическим сдвигом (режим В) (R-SEOI-EOI)	1 1 1 0 0 L2 L1 L0	Установка в “0” наивысшего по приоритету бита ISR регистра, установленного в “1”, и обозначение низшего приоритета по числу в поле L2-L0.
Установка приоритета (R-SEOI)	1 1 0 0 0 L2 L1 L0	Установка низшего приоритета по числу в поле L2-L0.
Установка режима циклического сдвига по типу А	1 0 0 0 0 0 0 0	Разрешение режима циклического сдвига при AEOI=1 (автоматический сброс ISR-бита)
Отмена режима циклического сдвига по типу А	0 0 0 0 0 0 0 0	Отмена режима циклического сдвига при AEOI=1 (автоматический сброс ISR-бита)

3.2.2 Циклический сдвиг приоритетов

В отличие от предыдущего режима приоритеты могут быть динамически изменены. В более общем случае команда OCW2 служит для установки в “0” произвольных разрядов регистра обслуживания ISR и циклического сдвига приоритетов с присвоением минимального значения приоритета любому из восьми возможных уровней.

Предварительно рассмотрим возможные варианты задания OCW2 (рисунок 3.10):

1. Сброс ISR-бита без изменения dna приоритетного кольца:

- ◆ неспецифическая команда конца прерывания EOI (D7-D5=001, бит OCW2 EOI=1) - выполняется сброс ISR-бита с наивысшим приоритетом (используется для режима фиксированных приоритетов);
- ◆ специфическая команда конца прерывания SEOI (D7-D5=011, биты OCW2 SEOI=1 и EOI=1) - выполняется сброс ISR-бита, указанного в поле L2-L0 OCW2 (используется для режима специального маскирования и может использоваться для режима фиксированных приоритетов). Разряды L2-L0 явно указывают на сбрасываемый разряд регистра ISR независимо от приоритета обслуживаемого запроса IRQ;

2. Сброс ISR-бита с изменением dna приоритетного кольца:

- ◆ с циклическим сдвигом приоритетов R-EOI (D7-D5=101, биты OCW2 R=1 и EOI=1 - циклический сдвиг по типу A) - выполняется сброс ISR-бита с наивысшим приоритетом и присвоение ему низшего приоритета (используется для циклического изменения приоритетов для одностипных внешних устройств). После завершения выполнения текущей ППОП каждый раз динамически изменяется dna приоритетного кольца в соответствии с номером запроса на прерывание IRQ, завершившего обслуживание;
- ◆ со специфическим циклическим сдвигом приоритетов R-SEOI-EOI (D7-D5=111, биты OCW2 R=1, SEOI=1 и EOI=1 - циклический сдвиг по типу B) - выполняется сброс ISR-бита с наивысшим приоритетом и присвоение низшего приоритета по числу в поле L2-L0 (используется для циклического изменения приоритетов). После завершения выполнения текущей ППОП каждый раз изменяется dna приоритетного кольца, причем значение низшего приоритета назначает непосредственно программист по какому-либо алгоритму;
- ◆ установка приоритета R-SEOI (D7-D5=110, биты OCW2 R=1 и SEOI=1) - выполняется только изменение dna приоритетного кольца (используется для изменения приоритетов запросов IRQ для режима фиксированных приоритетов, т.е. приоритеты фиксированы, но относительно нового значения dna приоритетного кольца).

3. Дополнительные команды для организации циклических сдвигов с автоматическим концом прерываний.

В ПКП ВН59А имеется возможность автоматического выполнения команды конца прерывания EOI или автоматического сброса ISR-бита. В данном режиме неспецифическая команда (EOI) формата OCW2 выполняется автоматически сразу же по окончании последнего сигнала $\sim\text{INTA}$ по его фронту, т.е. для ВМ86 после второго, а ВМ80 после третьего сигнала $\sim\text{INTA}$. При этом в подпрограмме обработки прерывания (ППОП) нет необходимости выполнять команду OCW2 формата EOI. Этот режим применяется, если не требуется строгого упорядочения приоритетов в рамках одного ПКП, т.е. нет приоритетной обработки (разновидность режима специального маскирования, но когда запрос, находящийся в обработке, может прерывать также и сам себя) или не выполняется вложенность обработки прерываний (т.е. в ППОП триггер разрешения прерывания не устанавливается в "1"). Если бит AEOI установлен в ведомом ПКП, то нет необходимости в ППОП ведомой БИС выполнять команду OCW2 сброса ISR-бита ведомого, а только ведущего ПКП.

Чтобы добиться автоматического циклического сдвига при установленном бите AEOI в ПКП предусмотрен специальный триггер, который устанавливается загрузкой слова OCW2 с "1" в бите D7, а снимается также загрузкой OCW2 с "0" в бите D7 (таблица 3.2).

При установке режима автоматического циклического сдвига в ПКП после второго (третьего для ВМ80) сигнала $\sim\text{INTA}$ выполняется автоматический сброс ISR-бита с наивысшим приоритетом и присвоение ему низшего уровня (циклический сдвиг по типу А).

Рассмотренные команды позволяют строить системы прерываний с разнообразными (в том числе динамически изменяемыми) структурами приоритетов.

Рассмотрим пример организации циклического сдвига приоритетов. В ППОП перед командой IRET выполняется загрузка команды OCW2 циклического сдвига по типу A:

MOV AL, 0A0h ; OCW2= 10100000b, AL:=OCW2, по команде OUT
OUT 30h, AL ; будет выполнен сброс ISR-бита с наивысшим при-
IRET ; оритетом и устанавливается дно приоритетного
; кольца, равное номеру сброшенного бита.

Например, одновременно поступили запросы IRQ5 и IRQ2. После обработки IRQ2 этому входу будет присвоен низший приоритет, а после обработки IRQ5 низший приоритет будет иметь вход IRQ5.

Дно приоритетного кольца						Высший приоритет		
7	6	5	4	3	2	1	0	
2	1	0	7	6	5	4	3	- после обработки IRQ2
5	4	3	2	1	0	7	6	- после обработки IRQ5

В ППОП перед командой IRET выполняется загрузка команды OCW2 циклического сдвига по типу B:

MOV AL, 0Eyh ; OCW2= 11100YYYb, AL:=OCW2, по команде OUT
OUT 30h, AL ; будет выполнен сброс ISR-бита с наивысшим при-
IRET ; оритетом и устанавливается дно приоритетного
; кольца, равное номеру в разрядах YYY.

Например. одновременно поступили запросы IRQ5 и IRQ2. В ППОП2 загружается OCW2 формата 0E6h, а в ППОП5 - формата 0E3h. После обработки IRQ2 входу IRQ6 будет присвоен низший приоритет, а после обработки IRQ5 низший приоритет будет иметь вход IRQ3.

Дно приоритетного кольца						Высший приоритет		
7	6	5	4	3	2	1	0	
6	5	4	3	2	1	0	7	- после обработки IRQ2
3	2	1	0	7	6	5	4	- после обработки IRQ5

Режимы циклического сдвига приоритетов позволяют исключить недостаток режима фиксированных приоритетов, когда обработка запросов с более низким приоритетом откладывается на неопределенное время из-за частого поступления запросов с более высоким приоритетом. Особенно это важно, когда к системе прерываний подключены однотипные ПУ, имеющие одинаковый приоритет.

При использовании режимов циклического сдвига по типу A и B необходимо учитывать, что при изменении дна приоритетного кольца при использовании

маски приоритетов IMR_i разряд IRQ_i может в дальнейшем оказаться незамаскированным, а другой вход IRQ_j окажется замаскированным.

В данном случае возможно применение режима специального маскирования, который устанавливается загрузкой команды OCW3.

3.2.3 Режим специального маскирования

Режим специального маскирования служит для того, чтобы разрешить прерывания от источников с более низким приоритетом, чем у запросов, находящихся в обработке в текущий момент времени. В данном режиме каждый бит в регистре обслуживания ISR запрещает только собственный уровень, но разрешает все остальные (как с более высоким, так и с более низким приоритетом), т.е. в общем случае регистр маски IMR совместно с регистром обслуживания ISR выполняют функцию маскирования поступающих запросов IRQ. В этом режиме запросы, поступившие на входы с более низкими приоритетами, обслуживаются до тех пор, пока не будет отменен режим специального маскирования. Формат команды OCW3 приведен на рисунке 3.11, который определяет режимы специального маскирования и последовательного опроса (считываемых прерываний), а также возможности чтения и выдачи в микропроцессор состояния регистра запросов прерываний IRR и регистра обслуживания ISR.

Режим специального маскирования назначается динамически после загрузки команды OCW3 и начала обслуживания запроса. Для перехода в режим специального маскирования необходимо выполнить следующие команды:

```
MOV AL, 6Ah    ; OCW3=01101010b, AC:=OCW3
OUT 30h, AL    ; A0=0, запись OCW3 в ПКП с адресом 30h.
Выйти из режима можно по командам:
```

```
MOV AL, 4Ah    ; OCW3=01001010b, AC:=OCW3
OUT 30h, AL    ; A0=0, запись OCW3 в ПКП с адресом 30h.
```

В ППОП при установленном режиме специального маскирования сброс ISR-бита при возврате из прерывания выполняется по команде специфического конца прерывания OCW2 формата SEOI (рисунок 3.10), в которой номер сбрасываемого ISR-бита указывается непосредственно в команде (поле L2-L0). Использование других форматов команд OCW2, например, неспецифического конца прерывания (сброс ISR-бита с наивысшим приоритетом) приведет к сбросу ISR-бита чужой ППОП. Например, если выполнение ППОП3 было прервано запросом с бо-

лее низким приоритетом IRQ5, то при возврате из ППОП5 в ISR-регистре будет сброшен третий бит, а не пятый.

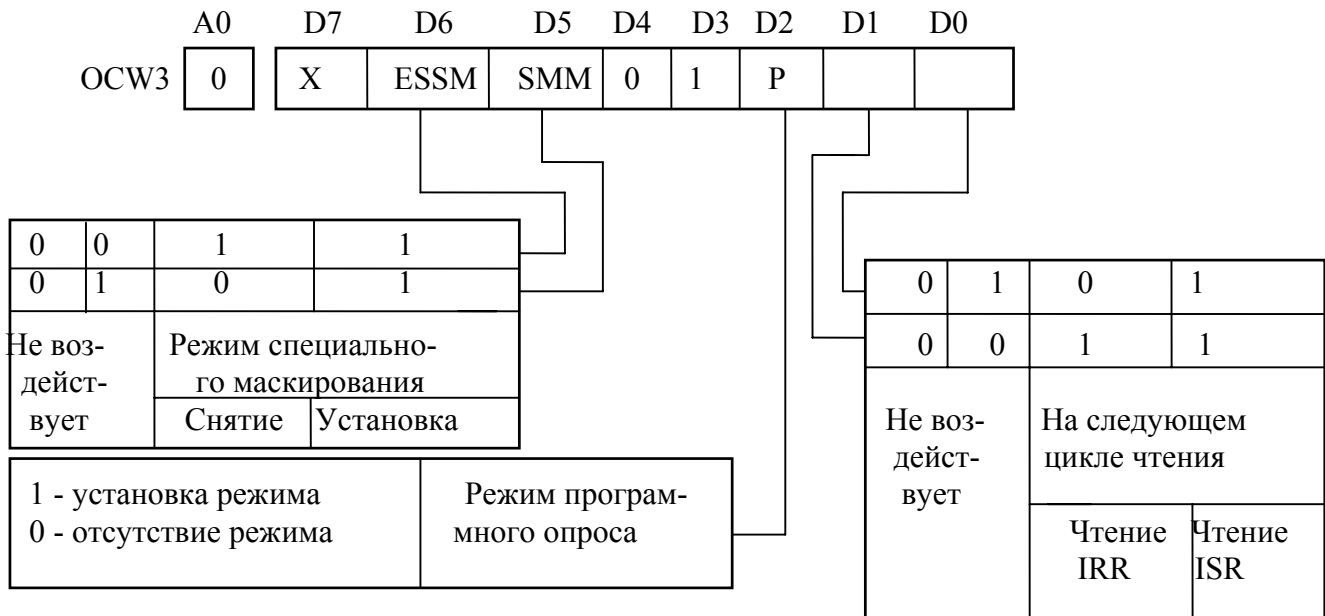


Рисунок 3.11 - Формат управляющего слова OCW3 для изменения режима работы

Следует заметить, что во время начальной установки ПКП (при загрузке команды ICW1) режим специального маскирования автоматически аннулируется.

3.2.4 Режим последовательного опроса

Режим программного опроса используется для определения источника прерывания непосредственно программистом путем последовательного опроса источников запросов на прерывание IRQ многократной загрузкой управляющего слова OCW3 и чтением в МПр слова состояния ПКП. Для реализации этого режима предварительно прерывания в МПр должны быть запрещены (в регистре слова состояния процессора флаг разрешения прерываний IF должен быть сброшен). Тогда в ответ на поступление сигнала прерывания INT микропроцессор игнорирует выдачу сигналов "Подтверждения прерывания" \sim INTA, а функции опроса состояния аппаратуры прерывания возлагаются на программиста.

Следовательно, для перехода в режим последовательного опроса необходимо при инициализации или перед установкой режима программного опроса запретить прерывания в МПр, установить режим программного опроса загрузкой команды OCW3, а затем прочитать в МПр при A0=0 слово состояния ПКП в формате, представленном на рисунке 3.12. Младшие разряды считанного байта слова состояния ПКП указывают номер внешнего устройства (вектора прерывания) с наивысшим приоритетом, а единица в старшем разряде байта указывает на наличие запроса на прерывание INT от этого устройства. После считывания байта слова состояния ПКП в МПр в БИС ПКП автоматически снимается режим программного опроса. Обслуживание запроса заключается в программном декодировании информации, получаемой от ПКП, и выполнении соответствующей подпрограммы, реализующей обмен между МПр и выбранным устройством. При этом все функции по сохранению необходимой информации в стеке и формированию адреса ППОП возлагаются на программиста.



Рисунок 3.12 - Формат слова состояния ПКП для режима программного опроса

Режим последовательного опроса обычно применяется в тех случаях, когда существует единая программа обработки прерывания для нескольких запросов IRQ, а также для расширения числа уровней IRQ (более 64).

```

СТI                               ; IF:=0 запрещение прерываний в микропроцессоре
. . . . .
MOV AL, 01001110b                ; OCW3=01001110b, AC:=OCW3
OUT 30h, AL                      ; A0=0, запись в ПКП OCW3, бит P=D2=1
. . . . .
IN AL, 30h                       ; чтение в МПр слова состояния ПКП,
                                ; OCW3:=01001010b - снятие режима ПО.
  
```

Далее выполняется анализ бита D7 в AL и если D7=1, то производится обращение к ППОП по команде CALL.

3.2.5 Режим чтения регистров ПКП

Входящие в состав ПКП регистры IMR, ISR и IRR могут быть прочитаны в микропроцессор. Регистр маски IMR может быть прочитан в любой момент при A0=1 по команде IN без загрузки OCW3. Содержимое регистров запросов IRR и обслуживания ISR также может быть прочитано при A0=0, но предварительно в формате команды OCW3 в младших разрядах D1-D0 необходимо указать источник чтения (рисунок 3.11), так как в любой момент времени непосредственно доступен только один из регистров IRR или ISR. Если после загрузки команды OCW3 другие управляющие команды ICW не загружались, то содержимое регистров можно многократно читать по команде IN.

MOV AL, 4Bh ; OCW=01001011b, AL:=OCW3 для чтения ISR

OUT 30h, AL ; A0=0, запись OCW3 в ПКП

IN AL, 30h ; A0=0, чтение содержимого ISR в AL

Следует заметить, что при A0=0 в МПР по команде IN могут быть прочитаны: слово состояния ПКП в режиме программного опроса и содержимое регистров IRR и ISR. При этом, если в команде OCW3 установлен режим программного опроса, то первым будет прочитано слово состояния ПКП и сброшен режим программного опроса, а по последующим командам IN при A0=0 в МПР будет прочитано содержимое регистров IRR или ISR, код которого был загружен последней командой OCW3 в битах D1-D0. Загрузка OCW3 с битами D1-D0=00 или 01 не изменяет их значения.

STI ; IF:=0 запрещение прерываний в микропроцессоре

MOV AL, 01001110b ; OCW3=01001110b, AL:=OCW3

OUT 30h, AL ; A0=0, запись в ПКП OCW3, бит P=D2=1

IN AL, 30h ; Чтение в МПР слова состояния ПКП,

; OCW3:=01001010b - снятие режима ПО

IN AL, 30h ; Чтение в МПР содержимого регистра IRR ПКП

При загрузке ICW1 по умолчанию в OCW3 устанавливается режим чтения содержимого IRR.

3.3 Каскадное включение БИС ПКП

Одна БИС ПКП предназначена для обработки до восьми входных сигналов запроса IRQ. На рисунке 3.13 приведена структурная схема подключения 8-уровневого ПКП к магистрали микропроцессора. В адресном пространстве портов ввода-вывода одна БИС ПКП представлена двумя адресами (портами) A7-A1=constant, A0=0 или 1. Сигналы \sim RDIO, \sim WRIO вырабатываются контроллером шины МПр при выполнении команд IN и OUT, сигнал \sim INTA также вырабатывается контроллером шины в ответ на поступление от ПКП сигнала запроса на прерывание INT.

Каскадное соединение нескольких БИС ПКП позволяет расширить число входных уровней до 64 (один ведущий и 8 ведомых). На рисунке 3.14 приведена схема каскадного включения для 22 входных сигналов IRQ для МПр серии K1810.

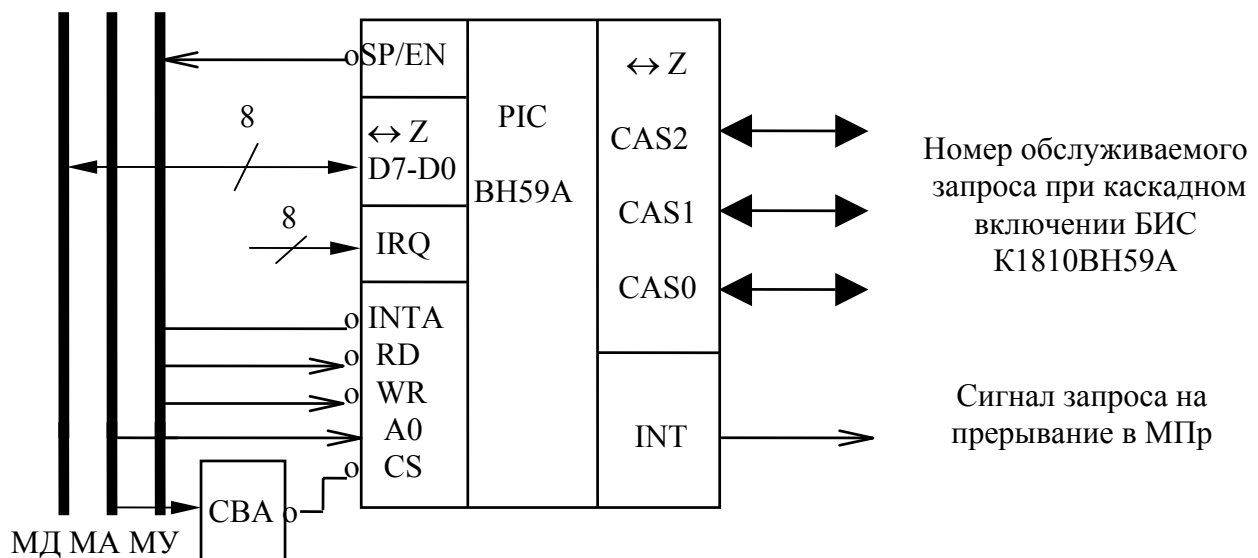


Рисунок 3.13 - Структурная схема подключения ПКП к микропроцессору

При начальной установке ПКП ($S=0$ в ICW1) обязательно в каждую БИС ПКП должно быть загружено управляющее слово ICW3: для ведущего ПКП код "1" содержат те разряды ICW3, к входам которых подключены выходы INT ведомых БИС; для ведомых БИС в разрядах D2-D0 ICW3 указывается номер микро-схемы, совпадающий с номером IRQ ведущей БИС, к которому подключен выход INT ведомой.

Для МПр серии K580 для ведущего ПКП вывод $\sim EN/\sim SP$ является входом и на него подается уровень логической единицы (напряжение питания), для ведомых БИС - уровень логического нуля, а для МПр серии K1810 вывод $\sim EN/\sim SP$ ведущего ПКП является выходом и сигнал $\sim EN$ поступает на вход шинного формирователя BA86 контроллера шины, а тип БИС ПКП (ведомая или ведущая) указывается в команде инициализации ICW4. Выводы $\sim EN/\sim SP$ ведомых БИС являются входами, на которые подается уровень логического нуля аппаратно, а при выдаче вектора на шину данных из ведомой БИС по второму $\sim INTA$ сигнал для шинного формирователя $\sim EN$ формируется ведущим ПКП, так как выводы $\sim EN/\sim SP$ нельзя объединять по монтажному "ИЛИ" (они не имеют третьего состояния).

При каскадном включении, если сигнал INT, поступивший в МПр, выработан сигналом IRQ от ведущего ПКП, то формирование адреса ППОП или адреса вектора таблицы IDT осуществляется этой же БИС аналогично тому, как это происходит при работе с одной БИС. Если же выработка сигнала INT вызвана по сигналу IRQ, поступившему на вход ведомой микросхемы, то формирование адреса ППОП осуществляется следующим образом. Для МПр серии K580 при поступлении первого сигнала $\sim INTA1$ ведущая БИС выдает на шину данных код команды CALL, а на шину CAS2-CAS0 - код номера ведомой БИС, вызвавшей прерывание. Ведомые БИС по этой же шине принимают этот код и сравнивают со своим номером, загруженным в ICW3 при начальной инициализации. По второму и третьему сигналам $\sim INTA$ код адреса ППОП выдается на шину данных той ведомой БИС, номер которой совпал с кодом на шине CAS2-CAS0.

Для МПр серии K1810 по сигналу $\sim INTA1$ выполняется только выдача номера запроса IRQ, вызвавшего выработку сигнала INT на шину CAS2-CAS0 из ведущего ПКП, а по $\sim INTA2$ по совпавшему номеру ведомая БИС ПКП выдает на шину данных адрес вектора прерывания таблицы IDT, а ведущая БИС сопровождает его выдачей сигнала $\sim EN$, если в команде ICW4 ведущей был запрограммирован бит разрешения данных $\sim EN$.

Заметим, что ведущая и ведомые БИС должны иметь различные адреса портов ввода-вывода, могут программироваться независимо друг от друга и на различные режимы обслуживания запросов.

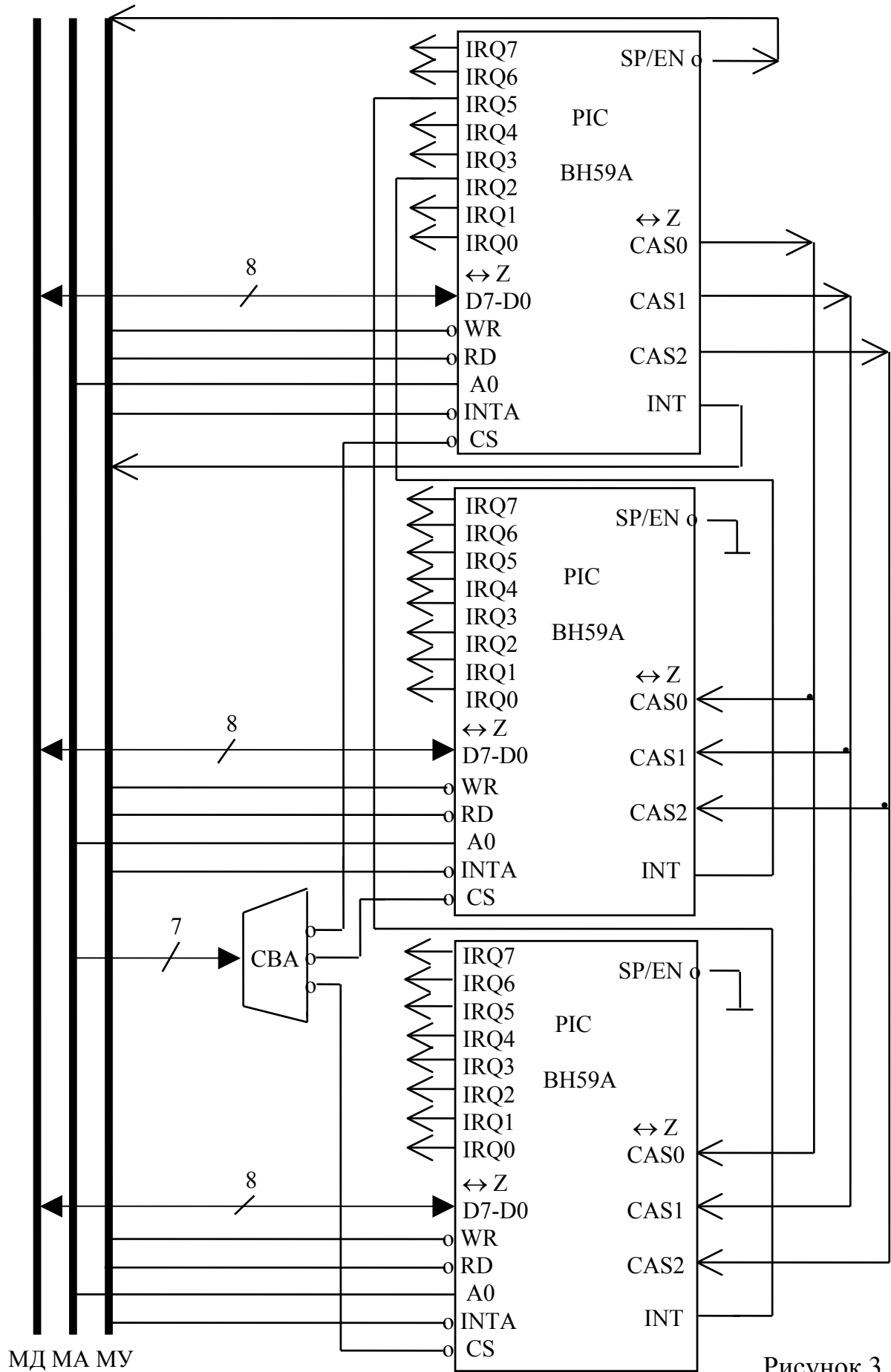


Рисунок 3.14

В ПКП ВН59А устранен недостаток нарушения режима вложенности обработки запросов при каскадном включении от ведомых ПКП путем введения дополнительного режима строгого упорядочения приоритетов, который задается командой ICW4 (бит PSV D4=1) только в ведущем ПКП. При этом в ведущем ПКП снимается требование строгого упорядочения приоритетов и разрешается прием новых запросов со входов IRQ, только что принятых к обслуживанию (или еще называют разрешение приоритета ведомого), если бит AEOI=0. С точки зрения технической реализации это означает, что схема сравнения COM разрешает выделение приоритетного запроса, если его номер меньше или равен номеру ISR-бита ведущего, установленного в "1" (если PSV=0, то сравнение производится только на меньше (рисунок 3.26)).

В этом случае ведущий ПКП разрешает незамаскированным IRQ с более высоким приоритетом в ведомой БИС проходить на выход INT ведущего, даже если соответствующий бит ISR регистра в ведущем установлен в "1". То есть, если в ведомом ПКП во время обработки запросов, приходит запрос с более высоким приоритетом, то новый IRQ вновь сформирует сигнал INT в ведущем ПКП.

При использовании режима строгого упорядочения приоритетов несколько изменяется и организация возврата из ППОП. Для входов IRQ ведущего ПКП в ППОП необходима загрузка одного OCW2 только в ведущий ПКП, а для IRQ ведомых ПКП требуется загрузка одного или двух OCW2 по следующему алгоритму:

1. Сначала OCW2 неконкретного EOI загружается в ведомый ПКП, который вызвал прерывание.

2. Затем считывается и проверяется значение ISR регистра этого ведомого (предварительно в OCW3 должен быть установлен режим чтения ISR регистра, если он не был установлен при инициализации или при установке режима специального маскирования или программного опроса).

3. Если регистр ISR ведомого ПКП равен нулю, то OCW2 загружается в ведущий ПКП, иначе OCW2 в ведущий не загружается.

Например, два ведомых ПКП подключаются ко входам IRQ1 и IRQ2 ведущего. Тогда приоритеты входов IRQ распределяются следующим образом:

Ведущий: IRQ0 - высший приоритет

Ведомый 1: IRQ0, IRQ1, IRQ2, IRQ3, IRQ4, IRQ5, IRQ6, IRQ7

Ведомый 2: IRQ0, IRQ1, IRQ2, IRQ3, IRQ4, IRQ5, IRQ6, IRQ7

Ведущий: IRQ3, IRQ4, IRQ5, IRQ6, IRQ7 - низший приоритет.

Пусть запросы не маскируются и поступает запрос на вход IRQ4 первого ведомого:

1 ситуация: AEOI=0 и PSV=0 для ведущего ПКП.

После первого \sim INTA ISR регистр ведущего равен: 00000010

ISR регистр 1-го ведомого равен: 00010000

Тогда ни один из вновь поступивших запросов с более высоким приоритетом IRQ0-IRQ3 первого ведомого не может прервать выполнение текущей ППОП до сброса ISR-бита в ведущем ПКП. Только один запрос IRQ0 ведущего может прервать текущую ППОП.

2 ситуация: AEOI ведущего = 1, PSV=0.

По второму сигналу \sim INTA в ведущем ПКП будет автоматически сброшен ISR-бит с наивысшим приоритетом, если не установлен режим автоматического сдвига приоритетов (дополнительная команда OCW2) и теперь запросы IRQ0-IRQ3 первого ведомого и IRQ0 ведущего с более высоким приоритетом могут прервать выполнение текущей ППОП, но и запросы по входам IRQ2-IRQ7 ведущего с более низким приоритетом могут прервать выполнение текущей ППОП, включая IRQ0-IRQ7 второй ведомой через вход IRQ2 ведущей БИС ПКП.

3 ситуация: AEOI=0, PSV=1 для ведущей.

В данной ситуации запросы только IRQ0-IRQ3 первой ведомой и IRQ0 ведущей с более высоким приоритетом могут прервать выполнение текущей ППОП. При этом необходимо учитывать, что при обработке прерывания от ведущего ПКП (запросы IRQ0, IRQ3-IRQ7 ведущего) могут прерывать свои собственные ППОП, для чего при входе в ППОП программист должен предусмотреть их маскирование по этим входам до окончания выполнения ППОП.

В данном примере также необходимо учитывать в ППОП от ведомых БИС последовательность сброса ISR-бита в ведомом и ведущем ПКП по ранее рассмотренному алгоритму. Например, при обработке прерывания IRQ4 от первого ведомого поступает запрос IRQ2, а затем IRQ0 от первого ведомого. Тогда в ISR-регистре первого ведомого будет сформирован код 00010101 и если при возврате

из прерывания в ППОП ведомого ПКП сразу сбросить ISR-бит в ведомом и ведущем, то сразу будут разрешены прерывания от источников с более низким приоритетом от ведущего и второго ведомого ПКП до завершения обработки ППОП с более высоким приоритетом (рисунок 3.16).

3.4 Влияние помех на входах IRQ в режиме UP=1 (IRQ по уровню)

На входах IRQ при бите UP=1 в ICW1 сигнал запроса должен сохранять высокий уровень до окончания первого сигнала $\sim\text{INTA}$. Если это условие не выполняется (IRQ снимается раньше до поступления $\sim\text{INTA}$), то ПКП "моделирует" "1" на входе IRQ7 (запрос с низшим приоритетом), так как сигнал IRQ вызовет выработку сигнала INT, процессор в ответ выставит 2 или 3 ответных сигнала подтверждения прерывания, а в регистре ISR бит не будет установлен (рисунок 3.15).

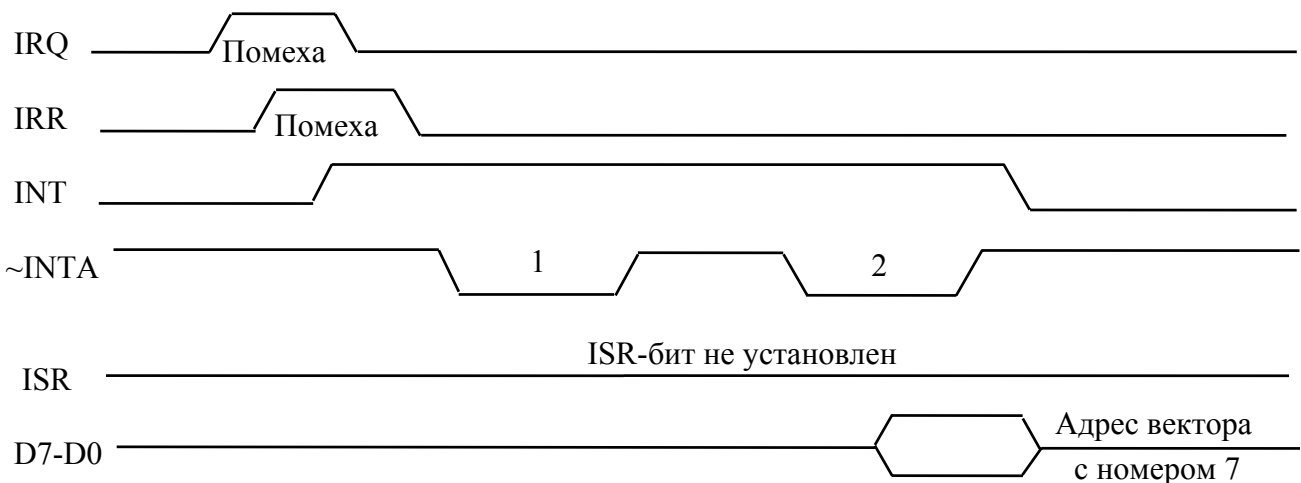


Рисунок 3.15 - Временные диаграммы работы ПКП в режиме обслуживания для МПр серии K1810 при UP=1 с выработкой сигнала помехи

Таким образом, если источник прерывания ко входу IRQ7 не подключен (вход заземлен), то по второму сигналу $\sim\text{INTA}$ на МД выдается адрес вектора с номером 7 и в этой ППОП выполняется "очистка" помех, например, сброс ISR-бита с номером 7 (OCW2 формата SEOI) или просто возврат из прерывания.

Если ко входу IRQ7 подключен источник прерывания, то помеха также может быть обнаружена, так как запрос от устройства установит ISR-бит с номером 7, а "запрос" от помехи на входе IRQ7 не воздействует на установку ISR-бита ($ISR[7]=0$), т.е. в ППОП по вектору 7, считывая ISR регистр и проверяя бит 7, можно отличить эти ситуации (рисунок 3.17).

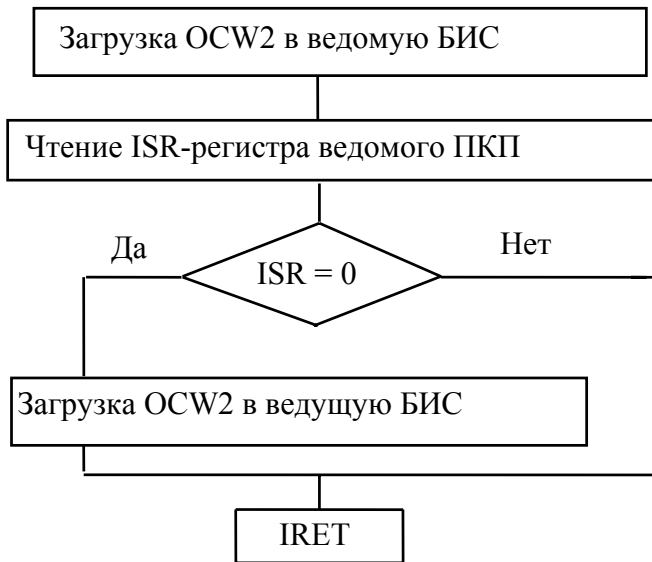


Рисунок 3.16 - Схема алгоритма конца прерывания для ведомой БИС при каскадном включении

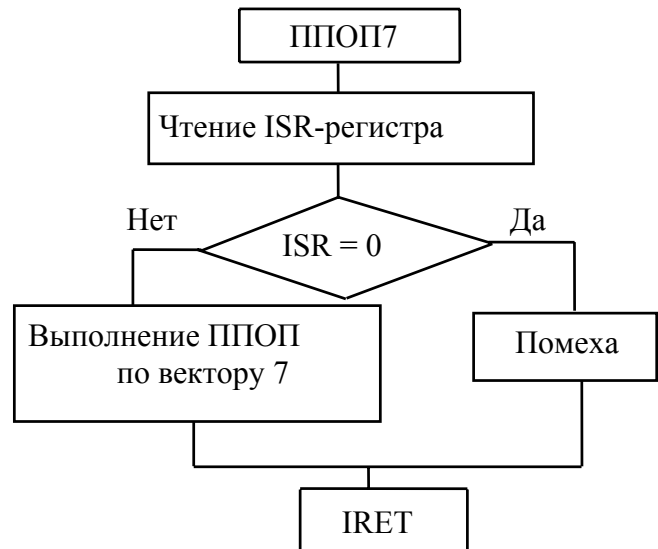


Рисунок 3.17 - Схема алгоритма конца прерывания при UP=1 для определения влияния "помехи"

Контрольные вопросы для самопроверки

1. Каковы особенности работы БИС ПКП ВН59А в режиме специального маскирования и области применения?
2. Сравните по архитектуре и функциональным возможностям блоки приоритетных прерываний К1810ВН59А и К1804ВН1.
3. Перечислите последовательность программирования ПКП ВН59А при каскадном и автономном включении.
4. В каких случаях применяется режим циклического сдвига приоритетов?
5. Для каких целей в ПКП используется команда OCW2? В каких режимах работы ПКП используется каждый формат OCW2?
6. Для каких целей используется режим с установкой бита PSV в ICW4 и как этот режим реализуется?
7. Для чего необходима выработка сигнала $\sim EN$ при работе с микропроцессором К1810ВМ86?

4 Организация временных интервалов в микроЭВМ

4.1 Программируемый таймер К580ВИ54

Программируемый таймер К580ВИ54 предназначен для задания временных интервалов в МПС и может использоваться в следующих режимах работы:

- ◆ прерывания по окончании счета;
- ◆ программируемого ждущего мультивибратора (одновибратора) ;
- ◆ программируемого делителя частоты (генератора тактовых импульсов с заданной частотой) ;
- ◆ программируемого генератора меандра (генератора прямоугольных импульсов (делителя частоты на 2)) ;
- ◆ одиночного программно-управляемого строба (счетчика событий) ;
- ◆ одиночного аппаратно-формируемого строба (счетчика событий с автозагрузкой).

На рисунке 4.1 приведено условное графическое обозначение таймера ВИ54, а в таблице 4.1 - назначение выводов БИС. Микросхема выполнена по n-МОП технологии в 24-выводном корпусе типа 2120.24.

Таблица 4.1 - Назначение выводов БИС ВИ54

09 —	CLK0	СТ ВИ54	↔ Z	
11 —	CE0		D0	— 08
			D1	— 07
15 —	CLK1		D2	— 06
14 —	CE1		D3	— 05
			D4	— 04
18 —	CLK2		D5	— 03
16 —	CE2		D6	— 02
			D7	— 01
23 —	WR		OUT0	— 10
22 —	RD		OUT1	— 13
21 —	CS		OUT2	— 17
19 —	A0			
20 —	A1			

Обозначение	Назначение
~CS	Вход выбора микросхемы
~WR	Сигнал записи с шины данных
~RD	Сигнал чтения на шину данных
D7-D0	Двунаправленная шина данных
CLK2-CLK0	Входы тактовых импульсов каналов 2, 1, 0
CE2-CE0	Входы разрешения работы каналов 2, 1, 0
OUT2-OUT0	Выходы каналов 2, 1, 0
A1-A0	Адресные входы выбора канала таймера и управляющего слова

Рисунок 4.1 Условное графическое обозначение БИС ПТ К580ВИ54

Структурная схема таймера представлена на рисунке 4.2. БИС содержит три независимых идентичных канала, каждый из которых включает 16-разрядный вычитающий счетчик СТ с частотой счета по входу CLK до 2 МГц с возможностью установки произвольного начального кода из буферного регистра RgB. Счет может выполняться в двоичном или двоично-десятичном коде 2421 над одно- или двухбайтными числами.

Буферный регистр RgB предназначен для загрузки и хранения начального значения счета, загружаемого из МПР с шины данных. Управление режимами осуществляется с помощью управляющего слова CW (Command Word), загружаемого также из МПР по шине данных в регистр управляющего слова RgCW.

Связь ПТ с МПР осуществляется через двунаправленную 8-разрядную шину данных под управлением пяти сигналов: A1, A0, \sim CS, \sim WR, \sim RD схемой управления чтением/записью в соответствии с таблицей 4.2. Буфер данных предназначен для сопряжения ПТ с магистралью данных микропроцессора и используется для программирования режимов работы таймера, загрузки буферного регистра RgB и считывания текущего значения счета и слова состояния каналов таймера.

Таблица 4.2 - Коды выборки каналов таймера ВИ54 и CW

A1 A0	\sim RD	\sim WR	\sim CS S	Выполняемая операция
0 0 0 1 1 0	1 1 1	0 0 0	0 0 0	Загрузка в счетчики СТ0, СТ1, СТ2 начального значения с шины данных: CTN:= D7-D0
0 0 0 1 1 0	0 0 0	1 1 1	0 0 0	Считывание содержимого счетчиков или регистра приказов обратного считывания каналов 0, 1, 2 на шину данных: D7-D0 = CTN V RgPR
1 1	1	0	0	Запись управляющего слова для установки режима работы канала или приказов "чтения на лету" или обратного считывания
1 1 X X	0 X	1 X	0 1	Не воздействует: шина данных в Z-состоянии

Схема управления СС (Channel Control) синхронизирует работу счетчика в соответствии с запрограммированным режимом и работу канала с микропроцессором.

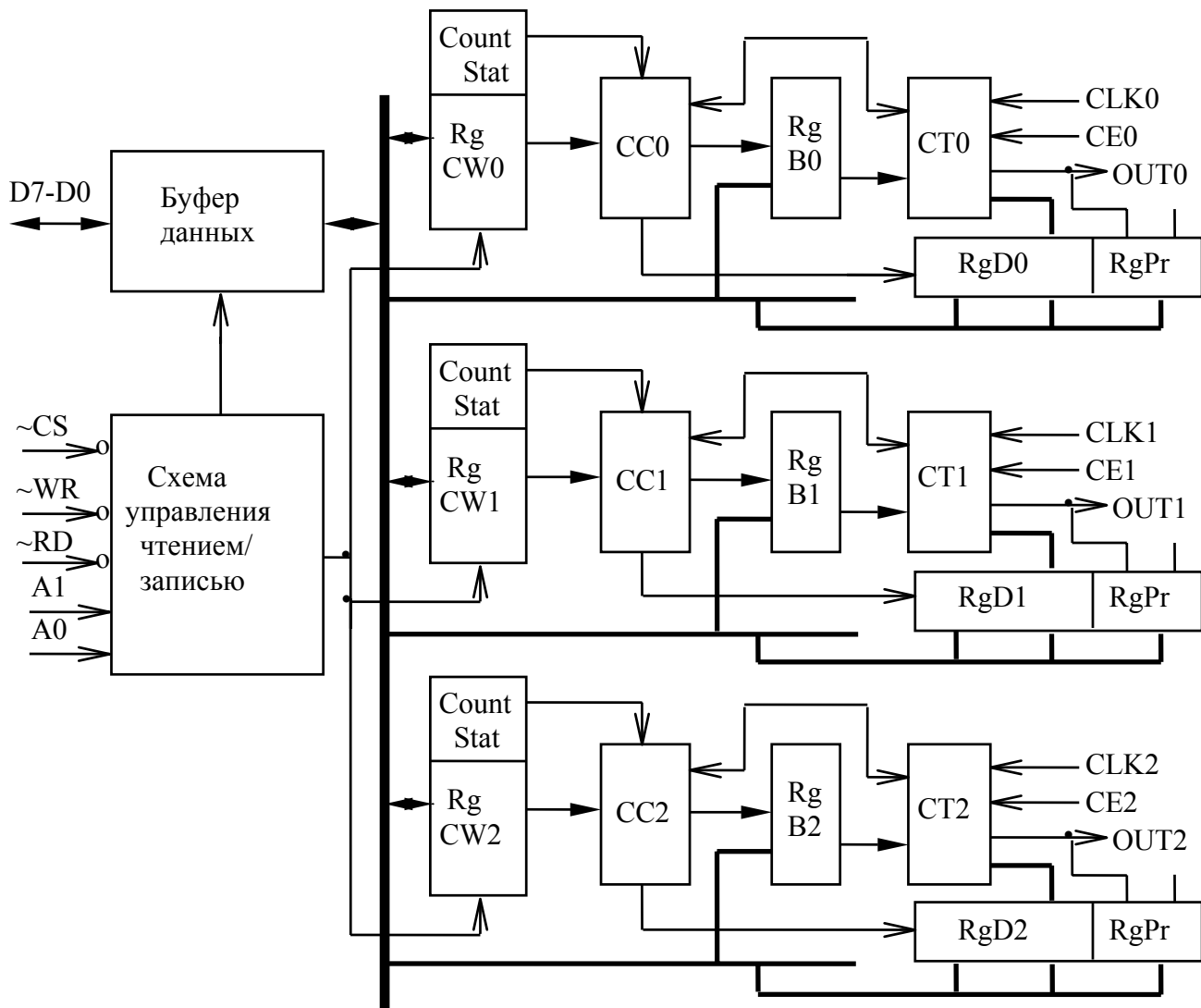


Рисунок 4.2 - Структурная схема программируемого таймера К1810ВИ54

Функционирование каналов осуществляется параллельно и независимо друг от друга. При включении питания состояния счетчиков, регистров и режимы работы оказываются неопределенными. Поэтому перед началом работы каждый канал должен быть инициализирован индивидуально загрузкой соответствующего управляющего слова CW и начальных данных счета CT .

4.2 Программирование таймера и режимы работы

Каждый из счетчиков в зависимости от настройки (инициализации) может быть либо двоичным, либо двоично-десятичным, а также работать в одном из шести режимов - 0, 1, 2, 3, 4, 5. Для программирования режима работы каждого из счетчиков в 6-разрядный регистр управляющего слова RgCW необходимо загрузить из МПР соответствующий код CW. При этом на адресных входах A1, A0 порта таймера должны быть поданы сигналы высокого уровня, а также сформирован сигнал выбора микросхемы $\sim CS=0$ из старших разрядов адреса портов таймера A7-A2 и сигнал $\sim WR=0$. Запись управляющих слов для различных каналов можно производить в любой последовательности. Из таблицы 4.2 видно, что в зависимости от комбинаций управляющих сигналов на входах $\sim CS$, $\sim RD$, $\sim WR$, A1, A0 к шине данных D7-D0 подключаются различные каналы таймера и выполняется запись либо управляющего слова CW, либо начальное значение счета в RgB, а считывать либо текущее значение счета из счетчиков, либо слово состояния каждого канала таймера. Формат управляющего слова CW для задания режимов работы таймера приведен на рисунке 4.3, которое кроме режима работы (поле M2-M0) определяет код счета (двоичный или 2-10, поле BCD) и формат обмена данными с микропроцессором при загрузке начальных данных счета: только старшего байта, только младшего байта или всего 16-разрядного слова (поле RL1-RL0). Кроме этого в этом же поле можно задать приказ фиксации текущего состояния счетчика, который будет рассмотрен ниже. Поле SC1-SC0 используется для указания номера канала, к которому относится данное управляющее слово CW, а также для задания приказа обратного считывания.

После записи управляющего слова CW в счетчик загружается начальное значение, два или один байт, при двухбайтных данных сначала записывается младший байт, затем - старший. Работа канала таймера разрешается только после загрузки последнего байта данных. Отметим, что последовательность инициализации и переинициализации каналов может быть произвольной, однако, для каждого канала сначала загружается CW, а затем начальные данные.

Во время функционирования таймера на вход CLK каждого из счетчиков могут поступать тактовые импульсы, а на вход CE - управляющие сигналы.

При рассмотрении режимов работы таймера необходимо обратить внимание на следующие различия и особенности:

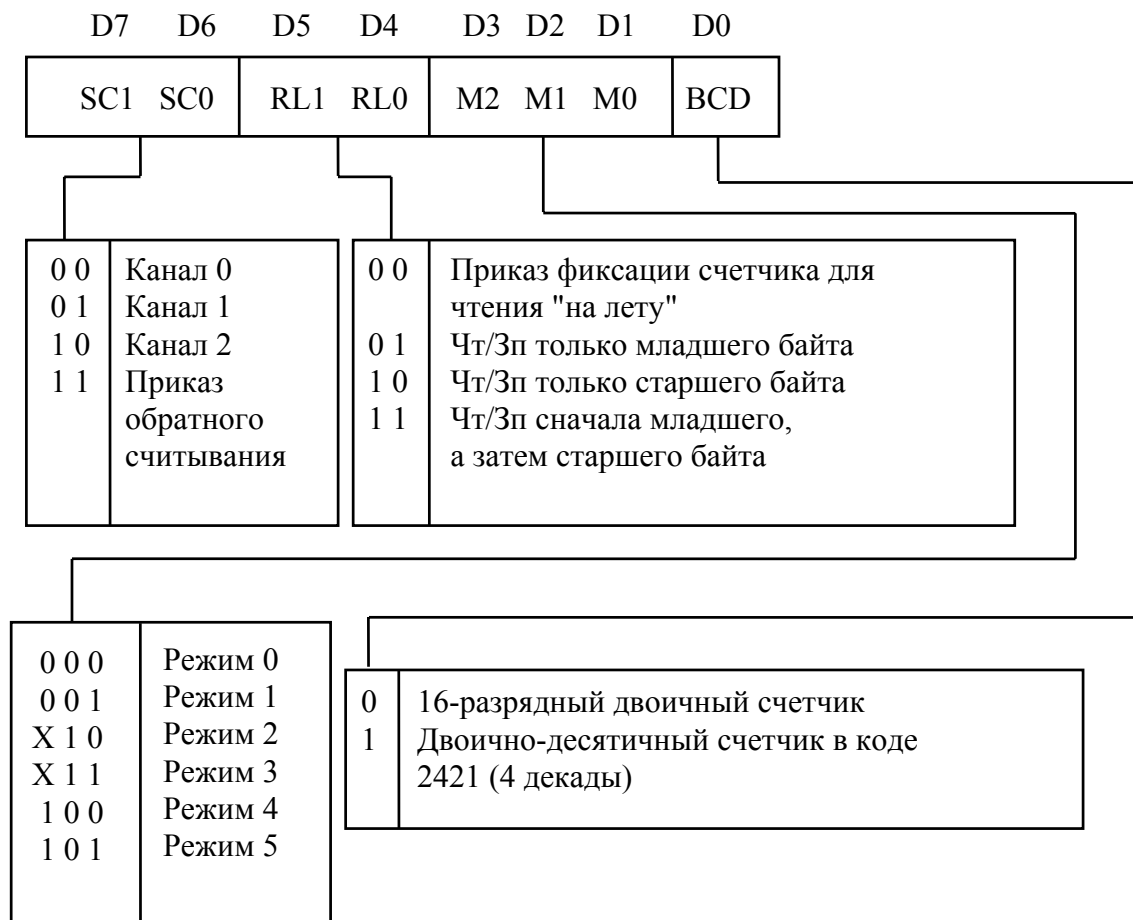


Рисунок 4.3 - Формат управляющего слова для задания режима работы

- ◆ уровень сигнала на выходе OUT таймера изменяется в зависимости от режима работы канала, текущего значения счета и изменения сигнала на входе СЕ;
- ◆ уровень сигнала на входе СЕ запрещает или разрешает счет, а фронт этого сигнала при перепаде из нуля в единицу разрешает, запускает или перезапускает счет;
- ◆ перезагрузка таймера новыми начальными данными во время счета либо запускает счет сразу с новыми начальными данными, либо начинает счет при следующем перезапуске счета;

- ◆ некоторые режимы являются автоперезапускаемыми по окончании счета, другие требуют загрузки новых начальных данных, а третьи требуют внешнего запуска по фронту сигнала СЕ.

4.3. Режимы работы таймера

Режимы работы таймера можно условно разбить на три группы: программируемый одновибратор (режимы 0 и 1), делитель частоты (режимы 2 и 3) и счетчик событий (режимы 4 и 5).

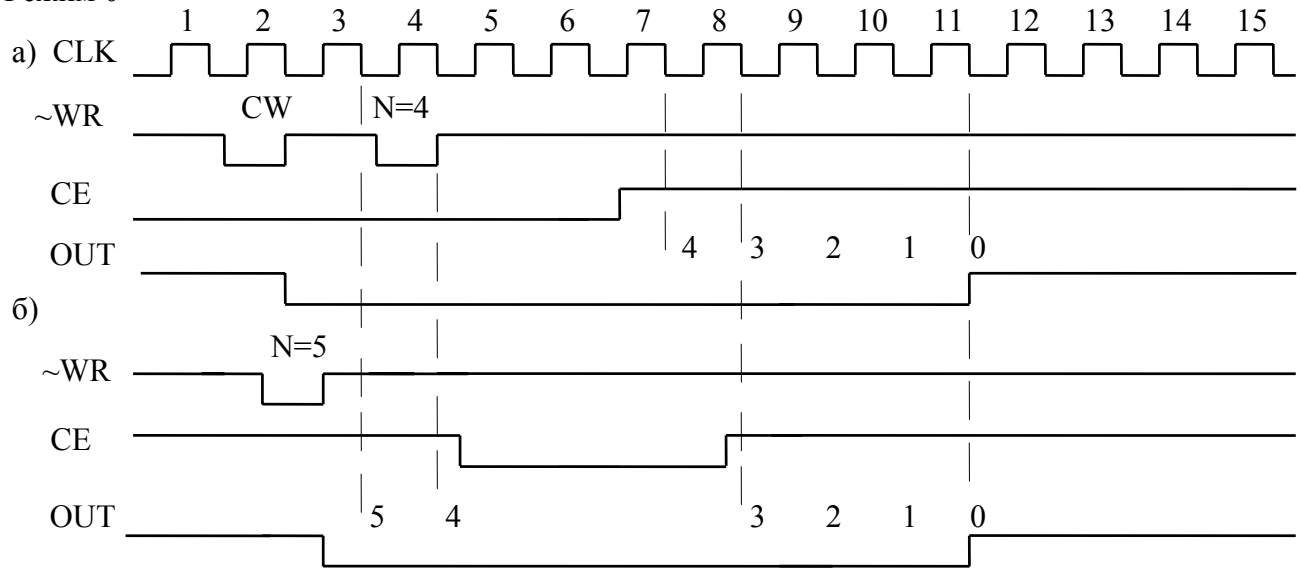
Режим 0 (прерывание по окончании счета). По окончании записи СW или загрузки начальных данных до окончания счета (СТ#0) на выходе OUT устанавливается сигнал низкого уровня (рисунок 4.4, а). После окончания счета (СТ=0) уровень сигнала на выходе становится высоким и сохраняется до следующей загрузки данных в счетчик. Перезапуск канала производится только при загрузке новых начальных данных. При этом перезагрузка СТ во время счета младшим байтом останавливает текущий счет, а загрузка старшего байта числа запускает счетчик сначала (рисунок 4.4, б). Счет возможен только при сигнале СЕ=1. Низкий уровень этого сигнала или ниспадающий фронт запрещает счет (рисунок 4.4, б).

Режим 1 (программируемый одновибратор (мультивибратор) рисунок 4.4 в,г). Выход OUT генерирует “0” по заднему фронту импульса синхронизации CLK только при внешнем запуске канала по фронту сигнала СЕ, и счетчик начинает считать. При переходе СТ в “0” на выходе OUT устанавливается уровень “1”. В отличие от режима 0 новое число, загруженное во время счета, не влияет на состояние выхода счетчика, а учитывается на следующем запуске по фронту сигнала СЕ. Кроме того, в данном режиме программируется не момент перехода к высокому уровню сигнала на выходе OUT, а длительность отрицательного импульса.

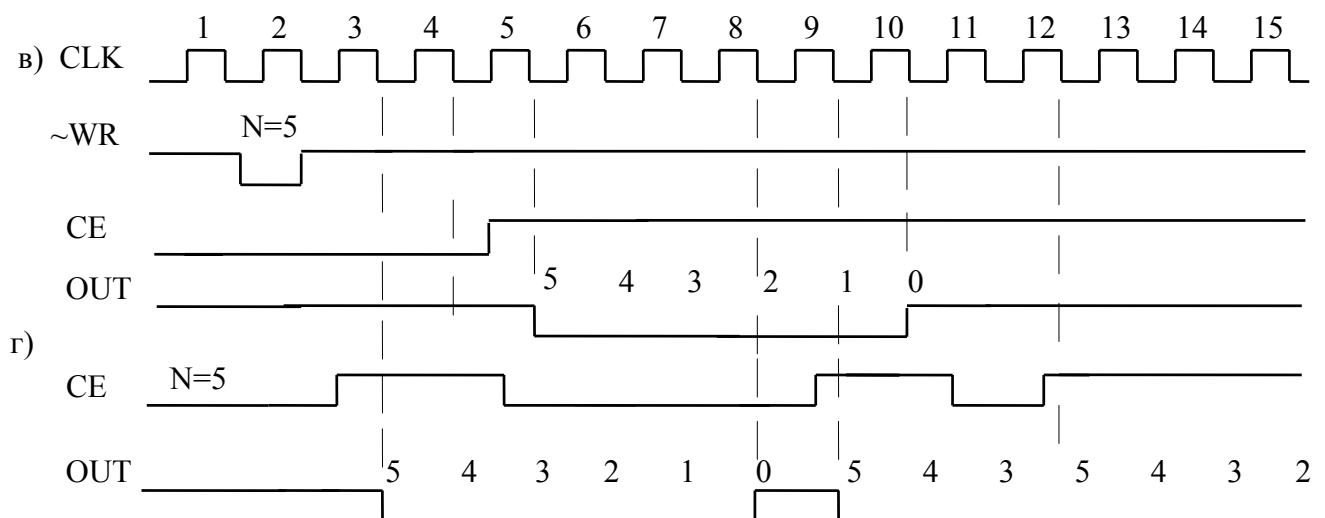
Особенностью данного режима является также то, что перезапуск счетчика можно осуществить без предварительной перезагрузки данных, которая необходима в режиме 0. Поэтому при подаче сигнала с выхода OUT канала на вход СЕ обеспечивается автоматический перезапуск счетчика по фронту сигнала с выхода

OUT и канал работает в режиме мультивибратора. Чтение текущего состояния счетчика возможно в любое время.

Режим 0



Режим 1



Режим 2

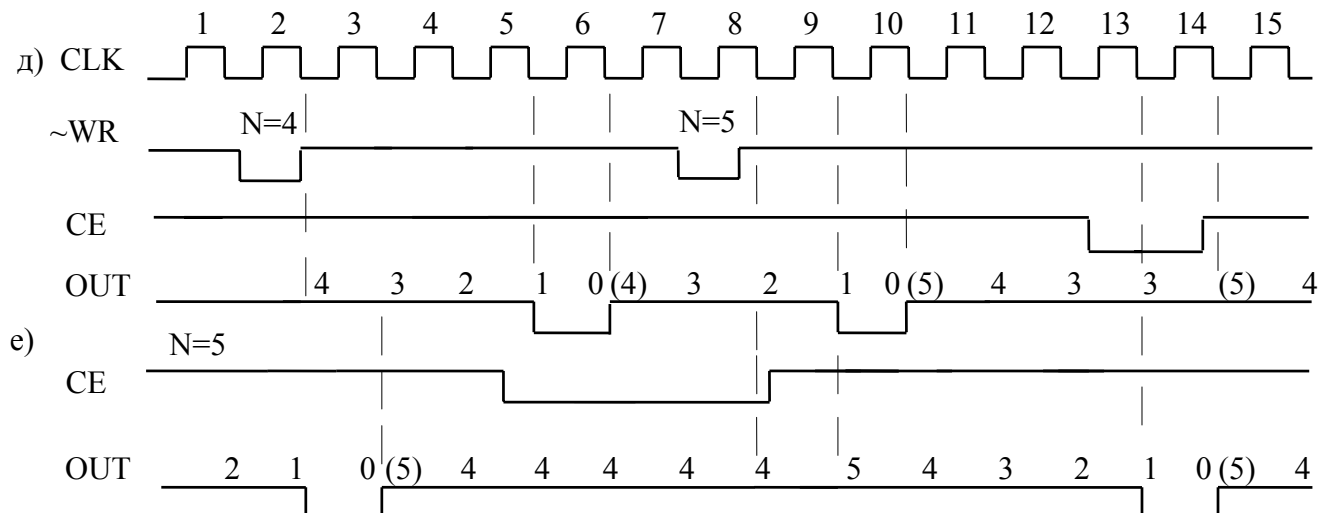


Рисунок 4.4 - Временные диаграммы работы таймера в режимах 0, 1, 2

Режим 2 (программируемый генератор (делитель) частоты) (рисунок 4.4, д, е). В этом режиме после просчета числа, записанного в СТ (RgB), на выходе OUT в К-м такте (СТ=1) появляется отрицательный импульс длительностью в один период тактовой частоты, т.е. длительность положительной части периода равна $(K-1) * \tau_{CLK}$, а отрицательной τ_{CLK} и осуществляется автоматический перезапуск канала СТ с загруженной константы. Длительность периода генератора определяется числом К, загруженным в RgB канала. Перезагрузка счетчика новым значением во время счета не приводит к изменению длительности текущего периода, а счетчик перестраивается на новый коэффициент пересчета на следующем запуске. При подаче на управляющий вход СЕ нуля счет приостанавливается, и на выходе OUT устанавливается “1”. Перезапуск счетчика осуществляется по фронту сигнала СЕ из “0” в “1”. Вход СЕ может служить для аппаратной синхронизации счетчика.

Режим 3 (программируемый генератор меандра или делитель частоты на 2)(рисунок 4.5, ж, з). Этот режим аналогичен режиму 2, с той лишь разницей, что если в счетчик, настроенный на данный режим, загружено четное число, то длительности положительных и отрицательных полупериодов на выходе OUT равны между собой и определяются по формуле $(K * \tau_{CLK}) / 2$. В случае нечетного К отрицательный полупериод выходного сигнала OUT меньше положительного на τ_{CLK} , т.е. $T_o = ((K-1) / 2) * \tau_{CLK}$, а $T_p = ((K+1) / 2) * \tau_{CLK}$.

Загрузка счетчика новым числом во время счета не влияет на текущий счет, а последующий просчет или перезапуск по фронту сигнала СЕ будет осуществляться с новым коэффициентом пересчета. Низкий уровень сигнала на входе СЕ (или спад сигнала) запрещает счет, и на выходе счетчика устанавливается сигнал высокого уровня. Сигнал СЕ=1 разрешает счет, а положительный перепад (из “0” в “1”) запускает счетчик с начального состояния. Следует отметить, что числом 3 счетчики в этом режиме загружать нельзя.

Режим 4 (счетчик событий или одиночного программно-управляемого строба) (рисунок 4.5, и, к). После записи CW на выходе OUT устанавливается уровень “1”. Запуск счета осуществляется сразу после загрузки начальных данных в RgB при СЕ=1. После окончания счета (СТ=0) на выходе OUT формируется отрица-

тельный импульс длительностью τ_{clk} . Запись данных в RgB во время счета младшего байта не влияет на текущий счет, а запись старшего байта перезапускает счетчик с нового значения. Уровень сигнала $CE=0$ или отрицательный фронт на входе CE запрещает счет, а сигнал высокого уровня - разрешает, т.е. счет продолжается с прерванного значения счета, а не сначала. Перезапуск счетчика осуществляется загрузкой начальных данных в RgB.

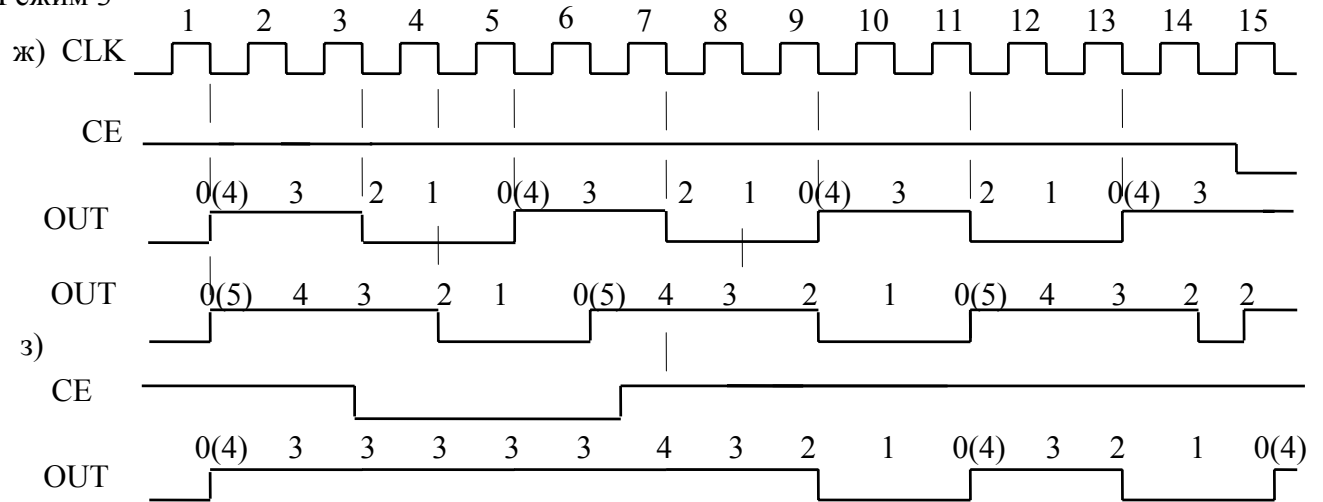
Режим 5 (счетчик событий с автозагрузкой или одиночно-управляемого строба) (рисунок 4.5, л, м). Отличие от режима 4 состоит в том, что счетчик является перезапускаемым, т.е. по фронту сигнала CE (из “0” в “1”) выполняется перезапуск счета сначала. Уровень сигнала $CE=0$ не приостанавливает счет, поэтому чтение текущего значения счетчика в МПР возможно в любое время. Перезагрузка новых начальных данных в RgB во время счета не воздействует на текущий счет, а новый счет начинается после перезапуска по фронту сигнала CE .

Во всех режимах сигнал CE является управляющим: запрещает счет низким уровнем и (или) перезапускает фронтом. Функции входа CE приведены в таблице 4.4.

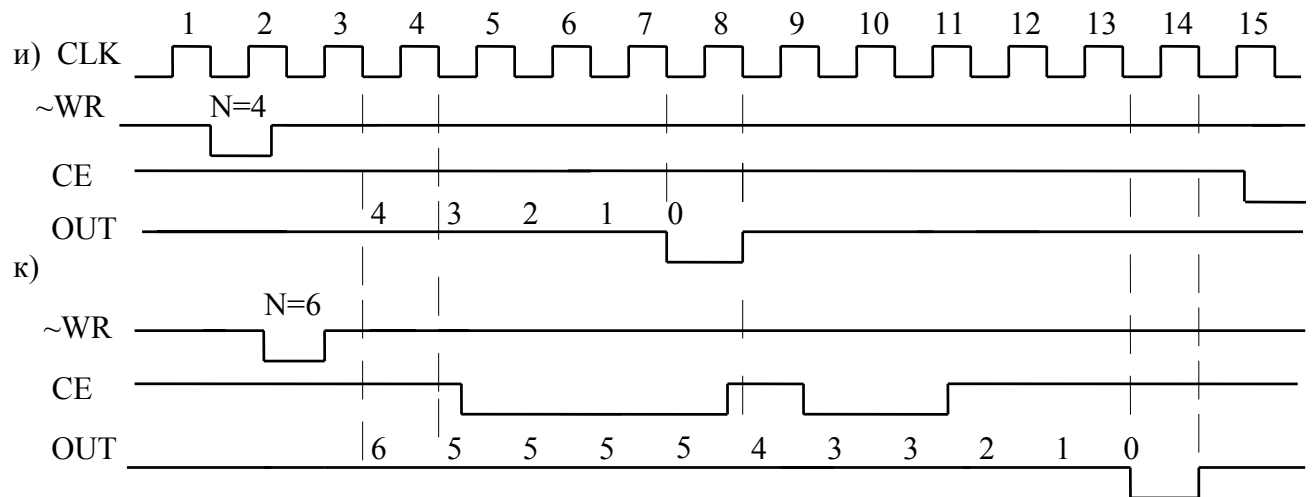
В ряде случаев необходимо контролировать текущее состояние СТ, например, когда СТ используется в качестве счетчика событий или реальных часов. Существует два способа чтения содержимого СТ: с остановом счета и без останова (чтение “на лету”).

В первом случае работа счетчика приостанавливается либо снятием сигнала на входе CE ($CE=0$), либо блокированием поступления тактовых импульсов CLK. Чтение содержимого счетчика осуществляется либо с помощью двух команд IN (сначала читается младший байт, затем старший, если в управляющем слове $D5=1$, $D4=1$), либо по одной команде IN (считывается только младший байт, если в CW биты $D5D4=01$, или старший байт, если $D5D4=10$ (рисунок 4.3)).

Режим 3



Режим 4



Режим 5

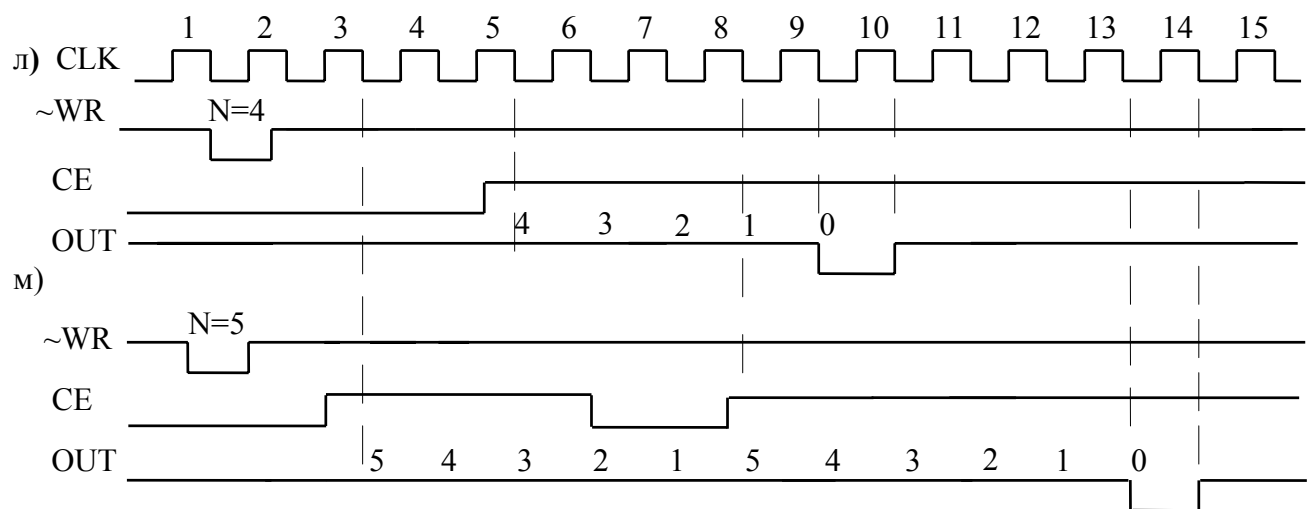


Рисунок 4.5 - Временные диаграммы работы таймера в режимах 3, 4, 5

Таблица 4.4 - Назначение сигнала на входе СЕ

Режим	Состояние сигнала на входе СЕ		
	Низкий уровень "0" или спад сигнала	Нарастание сигнала перепад из "0" в "1"	Высокий уровень (лог. "1")
0 и 4	Запрещает счет	* * * * *	Разрешает счет
1	* * * * *	1. Запускает счетчик сначала; 2. Сбрасывает в "0" выход канала OUT после прихода следующего CLK	* * * * *
2 и 3	1. Запрещает счет; 2. Сразу устанавливает на выходе OUT канала высокий уровень	Запускает счетчик на выполнение счета сначала	Разрешает счет
5	* * * * *	Запускает счетчик на выполнение счета сначала	* * * * *

Чтение без останова счета возможно лишь в случае предварительной загрузки управляющих слов для фиксации текущего состояния счетчика в специальном буферном регистре данных канала RgD. В таймере ВИ54 имеется два вида приказов: приказ фиксации счетчика (или чтения “на лету”) и приказ обратного считывания.

Приказ фиксации счетчика для чтения "на лету". При загрузке данного управляющего слова в дополнительном регистре RgD фиксируется текущее состояние счетчика канала, которое в дальнейшем может быть считано в микропроцессор для анализа, при этом считываются в МкПр только те байты, на загрузку которых был запрограммирован канал таймера. Если канал был запрограммирован на загрузку и младшего, и старшего байтов, то они оба обязательно должны быть считаны, иначе нарушается последовательность считывания байт при последующих попытках чтения. В приказе фиксации счетчика биты D5, D4 равны нулю, а биты D7, D6 указывают на номер канала, в котором выполняется фиксация текущего состояния счетчика (рисунок 4.3).

Приказ обратного считывания. Загрузка приказа обратного считывания выполняется, если биты D7, D6 управляющего слова равны 11. Приказ обратного считывания выполняет две функции:

- 1) подготавливает состояние счетчика (режим счета или счет закончен) и его выхода OUT (бит D4 \sim Stat=0) (рисунок 4.6);

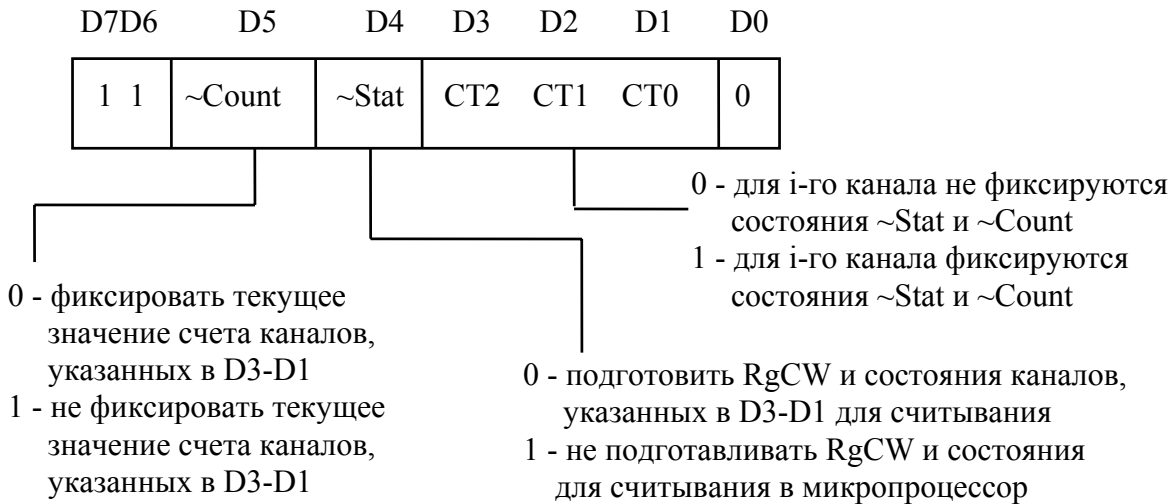


Рисунок 4.6 - Формат приказа обратного считывания

- 2) фиксирует текущее значение счета канала в регистре RgD (как и при чтении на "лету") (бит D5 \sim Count=0). Как и для приказа фиксации счетчика при загрузке приказа обратного считывания осуществляется защелка текущего значения счета в RgDi, если бит \sim Count=0 для тех каналов, которые отмечены "1" в разрядах D3-D1 приказа, а на последующих операциях считывания выполняется выдача на шину данных D7-D0 и ввод в МкПр регистра состояния (признаков) и управляющего слова соответствующего канала, если бит \sim Stat канала равен 0, иначе, если бит \sim Stat канала равен 1, по команде чтения IN в микропроцессор считывается текущее значение счета из RgDi, зафиксированное в момент загрузки приказа обратного считывания. Таким образом, в приказе обратного считывания бит D5 \sim Count=0 указывает на необходимость фиксации состояния текущего состояния счетчика в RgDi, а бит D4 \sim Stat=0 указывает на необходимость подготовки регистров состояния каналов и управляющих слов для считывания, причем по первой команде чтения IN в МкПр будет считано значение двухразрядного регистра состояния канала

RgPr на момент выполнения команды IN (рисунок 4.2) (выхода OUT канала и признака состояния счетчика канала Null Count) для тех счетчиков, биты CT_i которых равны 1, например, CT₂=1, CT₁=1, CT₀=0, т.е. из счетчиков 2 и 1. Допускается признак фиксации состояния $\sim\text{Count}=0$ и признак подготовки регистра управления и состояния (признаков) указывать в одной команде. Если бит $\sim\text{Stat}$ был установлен в 0, то содержимое всех регистров управления CW и состояния, для которых биты D3-D1 установлены в “1”, должны быть считаны в МкПр.

Формат регистра состояния, считываемого после загрузки приказа обратного считывания, показан на рисунке 4.7. Если бит $\sim\text{Count}$ был равен 1 для соответствующего канала, то на шину D7-D0 этого канала таймера выводится код текущего значения счета непосредственно из счетчика (чтение с остановом), даже если бит канала $\sim\text{Stat}_i=0$.

Алгоритм действия приказа обратного считывания можно описать с помощью таблицы 4.5.

Управляющие слова приказов фиксации счетчиков и обратного считывания загружаются, если биты адреса A1A0 порта ввода-вывода таймера равны 11. Считывание текущего значения счетчика RgD и регистра состояния для приказа обратного считывания выполняется по командам IN с указанием номера порта, из канала которого выполняется считывание. Поэтому при программировании каналов таймера необходимо учитывать, что перемежать загрузку управляющих слов фиксации счетчиков и приказов обратного считывания без команд чтения каналов недопустимо, так как команды чтения имеют один и тот же формат, а источник чтения определяется последней загруженной командой приказа.

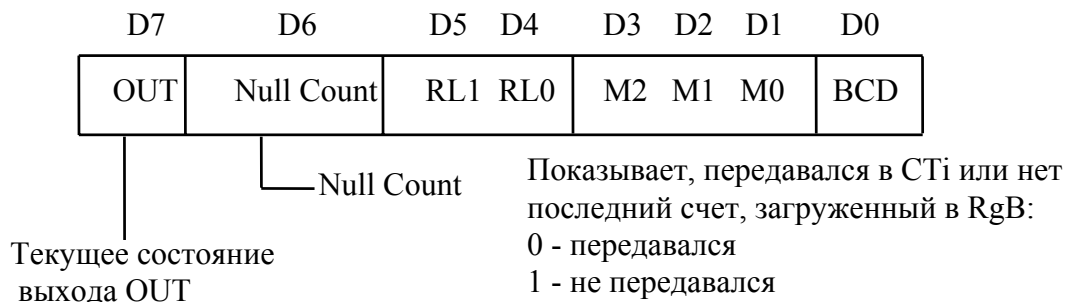


Рисунок 4.7 - Формат регистра состояния таймера при чтении по приказу обратного считывания

Таблица 4.5

~Count	~Stat	CT2 CT1 CT0	Комментарии
0	1	1 0 1	Защелкнуть в RgD каналов 2 и 0 текущее значение счета и подготовить для считывания в МПр содержимое RgD2 и RgD0
1	0	1 0 0	Подготовить для считывания в МПр текущего состояния канала 2 (OUT2.Pr2.CW2[D5-0])
1	0	0 0 1	Подготовить для считывания в МПр регистр состояния канала 0 (OUT0.Pr0.CW0)
0	0	1 1 0	Защелкнуть в RgD каналов 2 и 1 текущее значение счета и подготовить для считывания в МПр сначала содержимое регистров состояния каналов 2 и 1 OUT2.Pr2.CW2 и OUT1.Pr1.CW1, а затем значения RgD2 и RgD1

Примечание: при каждом считывании в МПр значений регистра состояния признак ~Stat=0 канала сбрасывается, указывая на источник чтения RgD канала (если бит канала ~Count=0), а при считывании текущего значения счета канала из RgD источником чтения является непосредственно счетчик канала с остановом счета по входам CE или CLK (кроме режимов 1 и 5).

Также следует заметить, что управляющие слова приказов фиксации счетчиков при загрузке в каналы таймера нигде не фиксируются, а декодируются и используются в качестве управляющего сигнала для фиксации текущего значения счета в RgDi. При загрузке приказа обратного считывания значения бит ~Stat каналов запоминаются в отдельном разряде управляющего слова CW канала, а при считывании этот бит сбрасывается. Значение бита ~Count=0 при загрузке приказа обратного считывания используется для защелки данных в RgD тех каналов, для которых биты D3-D1 установлены в 1, и определения номеров каналов, в которых была защелкнута текущая информация. Пример использования команд приказа:

```
MOV AL, 01110100b ; установка режима 2 канала 1 с загрузкой
OUT 43h, AL        ; младшего и старшего байтов
MOV AL, 01000000b ; загрузка приказа фиксации счетчика
OUT 43h, AL        ; канала 1
IN AL, 41h         ; чтение младшего байта канала 1
MOV BL, AL         ;
IN AL, 41h         ; чтение старшего байта канала 1
MOV BH, AL         ; в BX текущее значение счета канала 1
MOV AL, 11001010b ; загрузка приказа обратного
```

OUT 43h, AL	; считывания для каналов 2 и 0
IN AL, 40h	; чтение состояния канала 0 и сброс бита
MOV BL, AL	; ~Stat
IN AL, 42h	; чтение состояния канала 2 и сброс бита ~Stat
MOV CL, AL	; в BL и CL состояния каналов 0 и 2
IN AL, 42h	; чтение младшего, затем старшего байтов
MOV BL, AL	; канала 2 и сброс бита ~Count
IN AL, 42h	;
MOV BH, AL	; в BX текущее значение счета канала 2
IN AL, 40h	; чтение младшего, затем старшего байтов
MOV BL, AL	; канала 0 и сброс бита ~Count
IN AL, 40h	;
MOV BH, AL	; в BX текущее значение счета канала 0

4.4 Применение ПТ ВИ54 для организации общесистемных средств счета времени

Программируемый таймер ВИ54 является прибором широкого назначения и может иметь неограниченный спектр применений в зависимости от задач пользователя: прерывания в операционных системах с разделением времени через равномерные интервалы для переключения программ; вывод точных временных сигналов с программируемыми периодами (генераторы) в устройствах ввода-вывода (например, в АЦП); программируемая генерация скорости передачи данных в бодах; измерение временной задержки между внешними событиями; подсчет числа событий во внешнем эксперименте и ввод показателя в ЭВМ; прерывание процессора после появления запрограммированного числа внешних событий; часы суточного времени и другие.

На рисунке 4.8 приведена схема подключения таймера ВИ54 к магистрали МПр. В адресном пространстве ввода-вывода ПТ представлен четырьмя адресами: три первых определяют номер канала таймера, а четвертый - регистры управляющих слов каналов RgCW и приказов.

В зависимости от области применения можно получить различные варианты схем таймеров, расширяя разрядность СТ кратно 16, обеспечить реализацию различных режимов в каждом канале, управлять счетом с помощью управляющих сигналов на входе СЕ.


```

ST1: Mov AL, 00111001b ; Загрузка CW в канал 0 для 4 режима
      Out 43h, AL       ; Счет в двоично-десятичном коде
      Mov AL, 0FFh      ; Загрузка начальных данных
      Out 40h, AL       ; Младший байт
      Out 40h, AL       ; Старший байт

```

Текст основной программы

```

Mov AL, 00000000b ; Загрузка приказа фиксации счета в RgD0
Out 43h, AL       ; чтения "на лету"
In AL, 40h        ; Чтение младшего байта СТ0
Mov BL, AL        ; Сохранение байта в регистре ВХ
In AL, 40h        ; Чтение старшего байта СТ0
Mov BH, AL       ; В регистре ВХ обратный код числа команд
Neg BX          ; В регистре ВХ количество команд в программе

```

При достижении СТ0=0 будет выработан сигнал запроса на прерывание, сигнализирующий о том, что количество команд в программе превысило 9999.

Во втором задании на вход CLK1 подаются синхроимпульсы с частотой 2 МГц, т.е. длительность периода составляет 500 нс. Отсюда, для выработки сигнала прерывания через каждые 3,635 мс необходимо загрузить код 7270 или 1C66h и запрограммировать канал 1 на режим 2, при котором СТ1 автоматически перезапускается на новый счет в режиме генератора:

```

ST2: Mov AL, 01110100b ; Загрузка CW в канал 1 для 2 режима
      Out 43h, AL       ;
      Mov AL, 66h       ; Загрузка начальных данных
      Out 41h, AL       ; Младший байт
      Mov AL, 1Ch       ;
      Out 41h, AL       ; Старший байт

```

В задании 3 для выработки сигнала IRQ2 на вход CLK2 необходимо подавать сигналы обмена от ПУ, а выход OUT2 через схему "ИЛИ" соединить со входом CE2, который в режиме 5 автоматически перезапускает СТ2 на новый счет. Для запуска счета после инициализации канала необходимо предусмотреть схему внешнего запуска аппаратными или программными средствами через отдельный порт ввода-вывода. Предварительно в СТ2 загружается код 38=26h:

```

ST3: Mov AL, 10011011b ; Загрузка CW в канал 2 для 5 режима
      Out 43h, AL       ;

```

Mov AL, 26h ; Загрузка начальных данных
 Out 42h, AL ; Только младший байт

Для чтения содержимого СТ2 без нарушения счета достаточно выполнить команду IN, так как режим 5 допускает чтение без останова:

In AL, 42h ; Чтение младшего байта в аккумулятор.

Контрольные вопросы для самопроверки

1. Назовите режимы работы таймера K580ВИ54.
2. Какое назначение регистра RgB в структуре таймера?
3. В каких режимах возможно считывание содержимого счетчика таймера без останова и без загрузки приказа фиксации счета?
4. Чем различаются режимы одиночного программно-управляемого и аппаратно-управляемого строба?
5. Какая максимальная частота подается на вход CLK таймера?
6. В какой последовательности выполняется инициализация каналов таймера?
7. Какие функции выполняет сигнал на входе CE при работе таймера в различных режимах работы?
8. Для каких целей выполняется загрузка приказа обратного считывания и какой формат имеет считанное слово состояния таймера?
9. Разработайте схему включения таймера, проведите необходимые расчеты и составьте программу инициализации каналов таймера, которые индицируют время в часах, минутах и секундах.
10. Какой формат имеют приказы фиксации счета и обратного считывания и по какому номеру порта ввода-вывода они загружаются?
11. К каким действиям приводит перезагрузка начальных данных в RgB канала при различных режимах работы таймера?
12. В какое состояние устанавливается выход OUT канала после загрузки управляющего слова установки режима работы канала таймера для режимов с 0 по 5?

5 Организация ввода и отображения данных в МПС

Пульты управления (или консоли) небольших систем часто реализуются как устройства ввода и вывода простых клавиатуры и индикатора без применения устройства отображения на ЭЛТ. С помощью клавиатуры в МПС данные, адреса памяти и машинные коды вводятся в 16-ричной системе счисления. Кроме цифровых клавиш клавиатура имеет функциональные клавиши для ввода признаков управления. При выводе адреса памяти и данные отображаются с помощью светодиодных индикаторов. Примером такой системы ввода и вывода информации может служить учебный стенд “Электроника-580”.

Для клавиатуры применяются клавишные переключатели, представленные матрицей и не имеющие никаких электронных схем. На рисунке 5.1 показано подключение 64-клавишной клавиатуры к магистрали микроЭВМ через два порта (канала) ввода-вывода программируемого параллельного адаптера (интерфейса) (ППА) K580BB55. При нажатии клавиши соответствующие строка и столбец замыкаются. Код строки и столбца нажатой клавиши образуют кодовое слово этой клавиши. Таким образом, для ввода символа в микроЭВМ решается задача:

1. Определение факта нажатия клавиши на клавиатуре.
2. Нахождение номера (кодového слова) нажатой клавиши.
3. Осуществление передачи управления на соответствующую подпрограмму преобразования кода клавиши в код символа (кодогенератор) (решается программными или аппаратными средствами).

Процесс определения кодového слова (номера клавиши) называется сканированием клавиатуры и реализуется следующим образом. Из микроЭВМ через выходной порт (канал А ППА BB55) посылается код “0” в строку 0 и сигнал “1” во все остальные строки. Затем считываются и проверяются линии столбцов через порт В ППА, и если в ней нет сигнала “0” (нет нажатой клавиши в строке 0), то процесс сканирования повторяется для строки 1, затем 2 и т.д. Когда хотя бы в одном разряде выходного слова канала В фиксируется “0”, то обнаруживается нажатая клавиша, позиция строки которой известна по выводимой комбинации через канал В (в унитарном коде), а позиция столбца - по результату ввода (коду сканирования клавиатуры через канал А). Например, если была нажата клавиша

во второй строке третьего столбца, то через канал В будет принят код 11110111, а через канал А при этом был выдан код 11111011. Объединяя позиции строки и столбца обнаруженного “0” путем перевода унитарных кодов в восьмиричные, получим местоположение нажатой клавиши 32 (в 8-ричной системе счисления) (третий столбец, вторая строка).

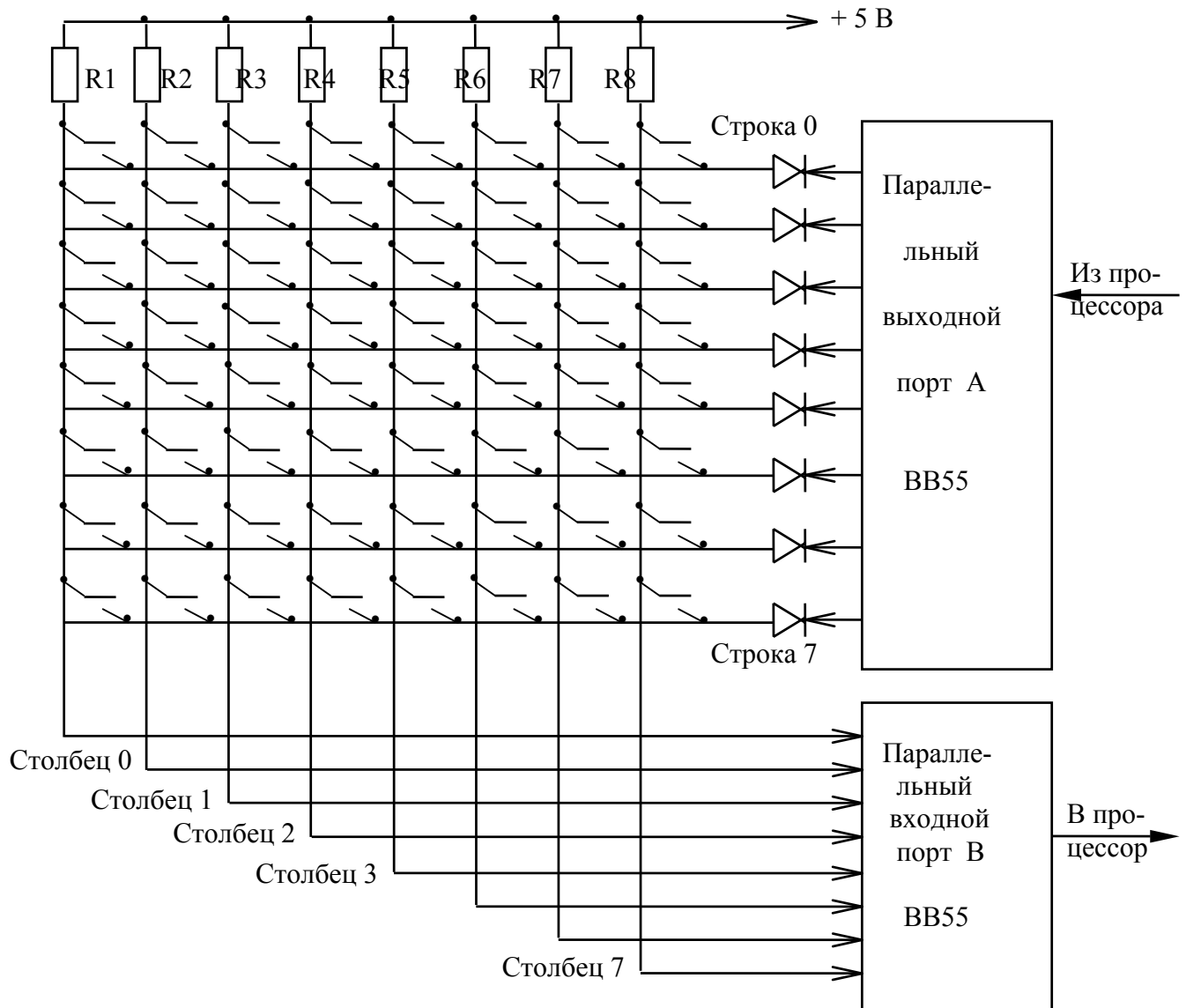


Рисунок 5.1 - Организация механической клавиатуры

С клавиатурой связаны две сложности:дребезг контактов и одновременное нажатие нескольких клавиш. Дребезг контактов возникает при переходе из одного устойчивого состояния в другое (замкнуто или разомкнуто). Продолжительность дребезга не постоянна, но обычно меньше 10 мс. Подавление дребезга осуществляется либо аппаратно, либо программно, последнее требует больших затрат процессорного времени. Для исключения ситуации одновременного срабатывания

нескольких клавиш ввод кодов замкнутых клавиш производят в порядке их обнаружения.

В настоящее время для индикации информации существует большое разнообразие цифровых и алфавитно-цифровых индикаторов. Наиболее распространенными для отображения 16-ричных цифр являются семисегментные светодиодные индикаторы (рисунок 5.2), где А - G и DP (десятичная точка) - входы индикатора, на которые подается код символа.

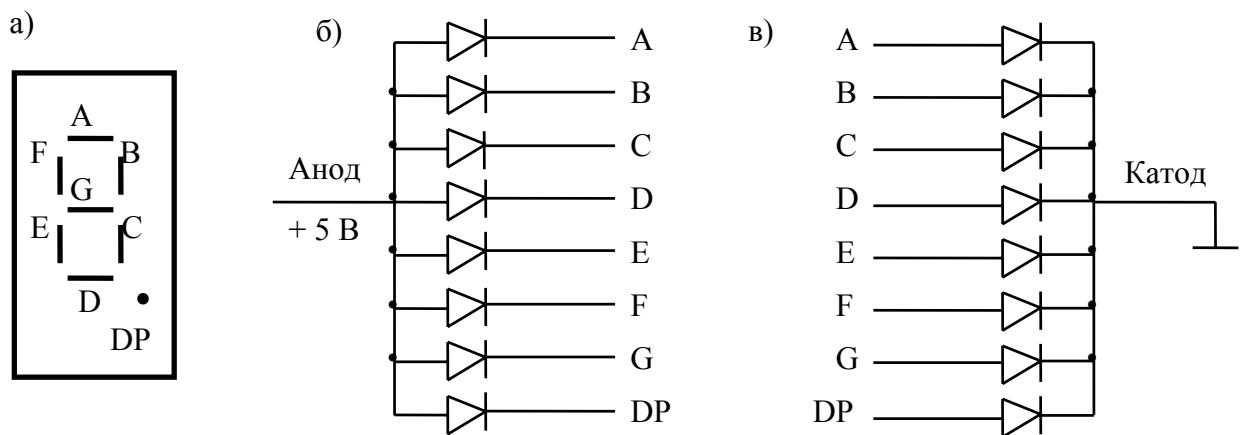


Рисунок 5.2 - Семисегментный светодиодный индикатор а), индикатор с общим анодом б) и с общим катодом в)

На рисунке 5.3 показана упрощенная структура подключения восьми индикаторов без использования буферных регистров для кодов символов. Дешифратор в любой момент времени выбирает только один из восьми индикаторов на 1 мс для подсветки выводимого на линии А-Г, DP кода символа, а после прохождения всех разрядов последовательность операций повторяется с нулевого разряда, т.е. выполняется регенерация изображения на индикаторах (подсветка), что создает иллюзию непрерывно работающего дисплея. Потребляемый ток включенного сегмента индикатора превышает нагрузочную способность ТТЛ-схем, поэтому в цепях разрядов и сегментов обычно используют транзисторные ключи-усилители.

Другим примером индикатора 16-ричных цифр может служить точечная матрица из 20 светодиодов. Индикатор имеет встроенный регистр-защелку, дешифратор и усилители тока. На вход индикатора подается 4-разрядный двоичный код цифры, который запоминается в регистре-защелке и преобразуется в видимое изображение эквивалентной 16-ричной цифры, при этом регенерация изображе-

ния не требуется, т.к. данные постоянно хранятся в регистре-защелке (рисунок 5.4).

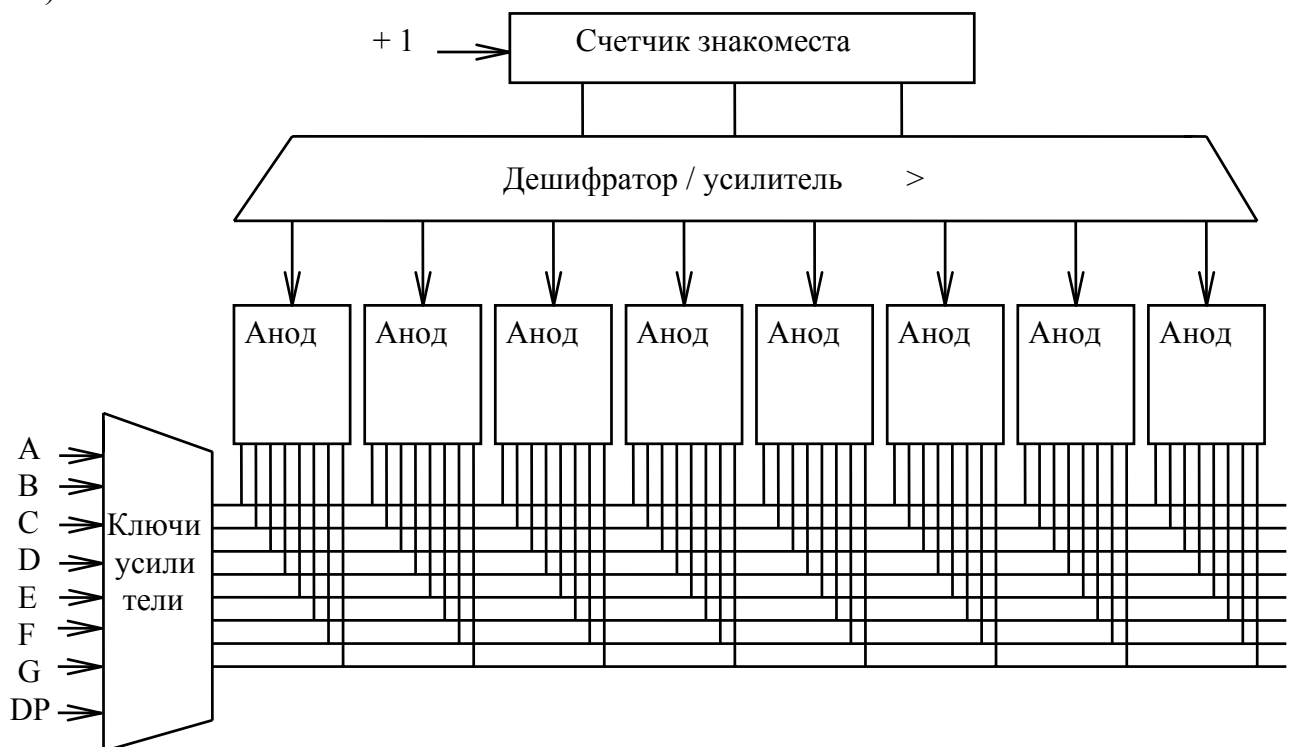


Рисунок 5.3 - Структурная схема подключения индикаторов

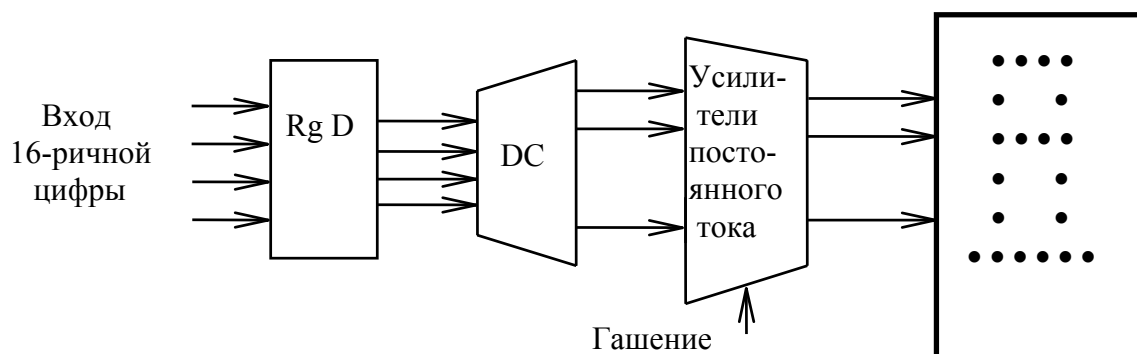


Рисунок 5.4 - Точечно-матричный 16-ричный индикатор

В качестве устройств ввода могут использоваться также датчики, представляющие собой переключательные элементы, управляемые от каких-либо объектов, например, датчики пожарной сигнализации, предельных режимов работы аппаратуры и т.д., как правило, отличающиеся по принципу действия от клавиатуры только отсутствием многократного срабатывания (дребезг контактов) при включении/выключении и длительностью замыкания контактов.

Клавиатура и индикаторы обычно подключаются к магистрали микроЭВМ через программируемые параллельные адаптеры ввода-вывода ВВ55 и при вводе и выводе по прерываниям процессор должен выполнять процедуры сканирования клавиатуры (периодического ввода кодов клавиш и программной обработки введенной информации) и регенерации индикатора. Для освобождения процессора от этих процедур в состав МПК К580 входит БИС программируемого контроллера клавиатуры и индикации (ПККИ) К580 ВВ79.

5.1. Программируемый контроллер клавиатуры и индикации К580ВВ79

Программируемый контроллер клавиатуры и индикации (ПККИ) предназначен для реализации обмена информацией между МПр и матрицей клавиш (датчиков) и блоком индикации и может выполнять следующие функции:

- ◆ ввод информации (по прерыванию) в МПС с помощью матрицы клавиш или датчиков;
- ◆ вывод информации на 8- или 16-разрядный алфавитно-цифровой (цифровой) индикатор

и позволяет полностью освободить МПр от операций сканирования клавиатуры, устранениядребезга контактов при замыкании клавиш и регенерации изображения на индикаторах.

На рисунке 5.5 представлено условное графическое обозначение БИС ПККИ ВВ79, а в таблице 5.1 - назначение выводов.

Структурная схема ПККИ приведена на рисунке 5.6, в которой можно выделить три основных блока: управления, интерфейсов индикации и клавиатуры, при этом блоки интерфейсов индикации и клавиатуры работают независимо друг от друга.

Блок управления. В состав блока входят: буфер шины данных, схема управления вводом-выводом, схема управления и синхронизации, счетчик сканирования и регистры управляющих слов.

Буфер шины данных служит для организации обмена с магистралью данных МПр при записи управляющих слов (CW) и данных в программируемый интер-

фейс контроллера, чтения слова состояния ПККИ и содержимого ОЗУ клавиатуры (М-ОЗУ) и ОЗУ индикации (отображения) и имеет двунаправленную 8-разрядную шину D7-D0 на три состояния.

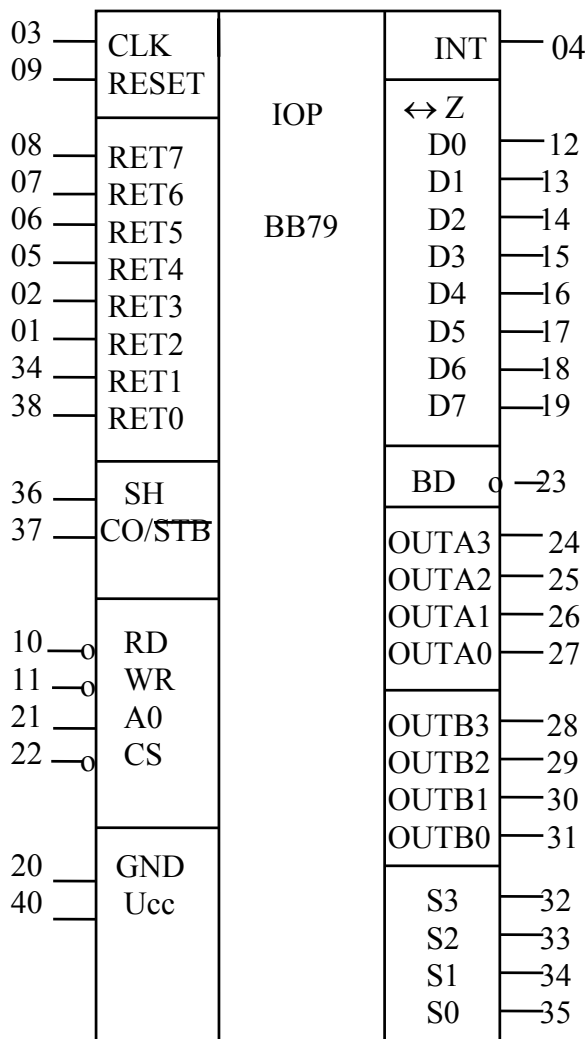


Рисунок 5.5 - Условное графическое обозначение БИС K580BB79

Таблица 5.1 - Назначение выводов БИС ПККИ K580BB79

Обозначение	Назначение выводов
RET7-RET0	Входы линии возврата
CLK	Сигнал синхронизации
INT	Выход запроса на прерывание
RESET	Начальная установка
~RD	Сигнал чтения на МД
~WR	Сигнал записи с МД
D7-D0	Двунаправленная шина данных
A0	Сигнал команда/данные
~CS	Сигнал выбора БИС
~BD	Сигнал гашения отображения (индикации)
OUTA3-0	Выходы кодов символов дисплея А
OUTB3-0	Выходы кодов символов дисплея В
SH	Сигнал сдвига (верхний/нижний регистр)
C0/~STB	Сигнал управления/строб
GND	Общий
Ucc	Напряжение питания +5 В

Схема управления вводом-выводом управляет обменом между МПр и ПККИ с помощью управляющих сигналов ~CS, ~RD, ~WR, A0 в режимах записи и чтения.

Тип информации определяется значением разряда адреса порта ПККИ A0. Если A0=1, то записываемая в ПККИ информация является кодом управляющего слова CW, а считанная - кодом слова состояния ПККИ (PSW). Если A0=0, то записываемая и считываемая информация является словом данных ПККИ.

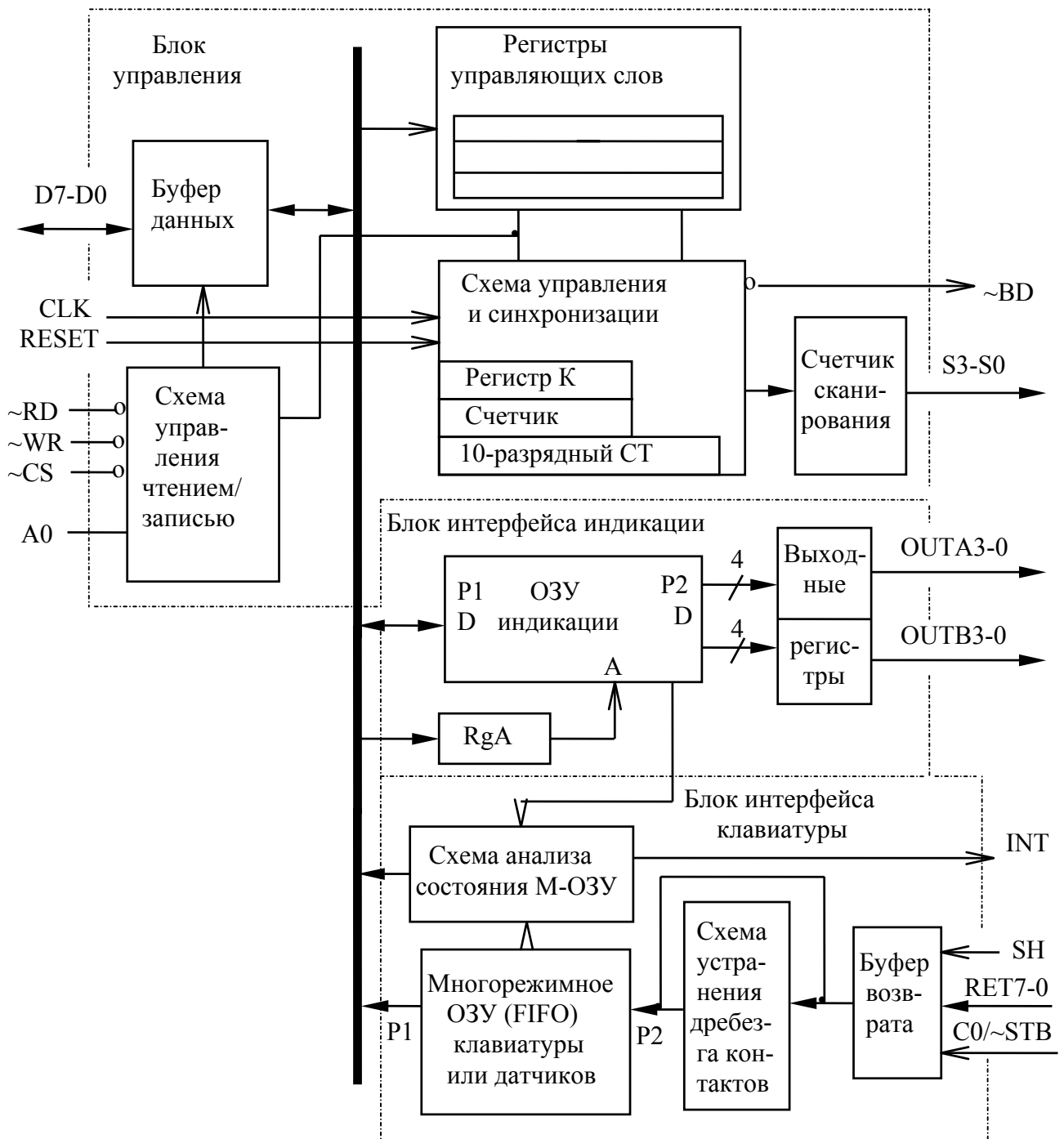


Рисунок 5.6 - Структурная схема ПККИ K580BB79

Регистры управляющих слов служат для записи и хранения управляющих слов CW , определяющих режимы работы ПККИ, а с помощью схем управления и синхронизации производится их дешифрация и выработка сигналов, которые управляют функционированием всех узлов БИС. Кроме того, с помощью 5-разрядного регистра коэффициента пересчета, вычитающего счетчика с предустановкой и 10-разрядного счетчика (четыре старших разряда которого выполняют

функции счетчика сканирования) схема управления и синхронизации делит внешнюю частоту синхросерии CLK, поступающей на вход ПККИ от ГТИ микропроцессора на K (K - число, записанное в регистр коэффициента пересчета), так как внутренняя частота синхронизации должна быть не более 100 кГц, что обеспечивает время просмотра (сканирования) клавиатуры за 5,12 мс и время ожидания успокоения дребезга контактов - 10,24 мс (два цикла сканирования). При этом тактовая частота внешнего генератора импульсов $T_{CLK} \leq 2$ МГц, а длительность такта внутренней синхронизации составляет 10 мкс.

Схема управления и синхронизации также вырабатывает синхросигнал на выход $\sim BD$ для гашения индикации во время смены символов на ней длительностью 160 мкс при загрузке управляющего слова CW5 - запрет выдачи информации по каналам A и B.

Счетчик сканирования (CTS) служит для формирования кодов, используемых для опроса матрицы клавиатуры (датчиков), и индикации, например, при загрузке управляющего слова CW0 “Установка режимов работы интерфейсов клавиатуры и индикации” производится настройка схемы для выдачи состояния счетчика сканирования, который может работать в двух режимах:

- ◆ кодированном (обычного двоичного счета);
- ◆ декодированном (режим сдвигового регистра с бегущим нулем).

В кодированном режиме для получения восьми сигналов опроса (сканирования) столбцов матрицы клавиатуры размерностью 8x8 необходимо использовать внешний дешифратор на 3 входа (8 выходов). В декодированном режиме на выходах счетчика сканирования S3-S0 генерируются 4 сигнала опроса, что исключает использование дополнительного дешифратора, и размерность матрицы клавиатуры не должна превышать 4x8 (32 клавиши).

Так как один и тот же счетчик вырабатывает сигналы опроса клавиатуры и регенерации изображения на индикаторе, то при работе счетчика в декодированном режиме будут работать только 4 разряда индикатора.

Блок интерфейса индикации включает ОЗУ индикации (отображения), регистр адреса ОЗУ индикации и выходные регистры блока индикации.

ОЗУ индикации (ОЗУИ) представляет собой двухпортовое запоминающее устройство емкостью 16 8-разрядных слов и предназначено для хранения информации, которая может отображаться на 8- или 16-разрядный индикатор. ОЗУ индикации состоит из двух независимых частей по 16 4-разрядных слов каждая, причем информация из каждой части может выдаваться на индикацию независимо на выходы OUTA3-OUTA0 и OUTB3-OUTB0 и запись в ОЗУИ также может осуществляться раздельно в каждую из частей или параллельно в обе части. После загрузки в БИС управляющих слов чтения или записи ОЗУИ его содержимое может быть прочитано или изменено микропроцессором по командам IN и OUT через системную шину данных.

Четырехразрядный регистр адреса предназначен для адресации ячеек ОЗУИ, который снабжен схемами автоинкрементирования и сброса в “0”. Также имеется возможность занесения начального адреса ОЗУИ с помощью управляющих слов, загружаемых в ПККИ для обеспечения вывода символа на индикацию с произвольной позиции или записи в ОЗУИ новой информации (обновления содержимого части ОЗУИ или чтения содержимого ОЗУИ в микропроцессор).

Блок интерфейса клавиатуры включает буфер возврата (регистр ввода), схему анализа и устранениядребезга контактов клавиатуры, многорежимное МОЗУ клавиатуры (датчиков), схему анализа состояния клавиатуры.

Буфер возврата (регистр ввода) служит для приема кода состояния счетчика сканирования, снимаемого с выходных шин (столбцов) матрицы клавиш или датчиков в каждом цикле сканирования, кроме режима “Ввода по стробу”. Каждый вход RET7-RET0 буфера имеет высокое внутреннее сопротивление, обеспечивающее непосредственное подключение к нему выходной шины матрицы клавиатуры. В режиме стробируемого ввода буфер возврата является регистром, снабженным схемами стробирования по положительному фронту сигнала на входе C0/~STB, а по линиям RET7-RET0 с клавиатуры поступает непосредственно код клавиши, т.е. формирование кода клавиши осуществляется в блоке клавиатуры и его выдача сопровождается формированием сигнала C0/~STB.

Схема анализа и устранениядребезга контактов клавиатуры подключается к выходам буфера возврата, служит для выявления запрещенных ситуаций при нажатии клавиш и не допускает повторного ввода кода клавиши, который может

произойти за счет дребезга контактов в режимах стандартной и “игровой” клавиатуры. Эта схема запоминает номер сработавшего датчика на первом цикле сканирования (факт нажатия клавиши), ждет в течение двух циклов сканирования клавиатуры (время устранения дребезга контактов) и проверяет, осталась ли клавиша в сработавшем состоянии. Если да, то срабатывание истинное и сформированный код клавиши вводится в М-ОЗУ, если нет - то оно ложное и игнорируется.

Например, при внутренней частоте контроллера 100 кГц время задержки схемы (устранения дребезга) равно 10,24 мс. Таким образом, всякое срабатывание клавиши на время, меньшее времени выдержки (2 цикла сканирования), будет игнорироваться. Таким образом, программируя внутреннюю частоту контроллера, можно в широких пределах изменять время просмотра матрицы клавиш и время выдержки схемы анализа и устранения дребезга контактов применительно к рабочим характеристикам клавиатуры, исключив ложные срабатывания.

Многорежимное М-ОЗУ клавиатуры работает в режиме сканирования клавиатуры и организовано по принципу ОЗУ магазинного типа “первый зашел - первый вышел” (FIFO) емкостью восемь 8-разрядных слов для клавиатурных режимов работы (кроме режима анализа матрицы датчиков). То есть каждый вновь вводимый в ОЗУ код клавиши записывается в автоматически адресуемую ячейку (через указатель стека записи), а затем считывается в том порядке, в котором вводится (через указатель стека чтения). Таким образом, М-ОЗУ клавиатуры предназначено для хранения кодов клавиш, поступающих через буфер возврата и схему анализа и устранения дребезга контактов клавиш или в обход ее для режима ввода по стробу с выходных шин матрицы клавиш.

В режиме анализа матрицы датчиков (опроса датчиков) М-ОЗУ работает как обычное адресуемое ОЗУ с произвольным доступом формата 8 слов x 8 бит при $S=0$ в CW2 или в режиме с автоинкрементированием адреса при $S=1$. Каждое слово имеет свой адрес. При этом в данном режиме в М-ОЗУ хранятся состояния всех строк датчиков (в остальных режимах (таблица 5.4) - записываемые коды клавиш), **а адрес записи равен содержимому счетчика сканирования в текущий момент времени.** Если схема анализа состояния М-ОЗУ обнаружит изменение состояния хотя бы одного датчика по сравнению с предыдущим циклом сканирования (опроса), то в конце цикла сканирования генерируется сигнал запроса на

прерывание INT. Информация из М-ОЗУ в режиме анализа матрицы датчиков считывается по адресу, установленному в RgA ОЗУ индикации с помощью CW2 “Чтение ОЗУ клавиатуры (датчиков)”. В режимах ввода с клавиатуры (стандартной и “игровой”) и ввода по стробу (таблица 5.4) М-ОЗУ клавиатуры работает как обычный 8-уровневый стек типа FIFO (ввод через порт P2), т.е. коды клавиш автоматически записываются в последовательные ячейки М-ОЗУ указателя стека записи, а считывание данных происходит через порт P1 в порядке записи данных через указатель стека чтения М-ОЗУ. После загрузки в ПККИ управляющего слова CW2 информация, хранящаяся в М-ОЗУ, может быть считана микропроцессором по команде IN.

Схема анализа состояния М-ОЗУ клавиатуры включает регистр состояния, а также схему формирования запроса на прерывание INT. Разряды регистра состояния ПККИ содержат сведения о наличии введенных кодов клавиш и отображают следующие характеристики М-ОЗУ и состояние БИС (рисунок 5.7):

- ◇ количество символов в очереди FIFO, кроме режима анализа матрицы датчиков (разряды D3-D0), причем значение разряда D3=1 используется как признак (флажок) того, что М-ОЗУ заполнено;

- ◇ признаки ошибок обращения при записи в заполненное М-ОЗУ (D5) или попытки чтения микропроцессором из М-ОЗУ, не содержащего информации (D4) (FIFO пуст);

- ◇ признак “Неготовность дисплея” (разряд D7) устанавливается и сигнализирует об ошибке при вводе данных в ОЗУИ, указывая на запрет доступа (записи) к ОЗУ индикации, когда выполняется команда CW6 “Программный сброс” формата записи константы в ОЗУИ или CW5 “Запрет записи в ОЗУ индикации - гашение” по обоим каналам (А и В) ОЗУИ;

- ◇ признак S/E (разряд D6) устанавливается в зависимости от режима работы контроллера:

♦ в режиме “Анализа матрицы датчиков”, если обнаружено изменение состояния хотя бы одного датчика по сравнению с предыдущим циклом сканирования с блокированием дальнейшей записи в М-ОЗУ;

♦ в режиме игровой клавиатуры с запрещением одновременного нажатия двух и более клавиш (или специального анализа ошибки), если обнаружено одновременное срабатывание двух и более клавиш с блокированием дальнейшей записи в М-ОЗУ.

Слово состояния ПККИ в основном используется для программного опроса контроллера с целью анализа его состояния в конкретных ситуациях.

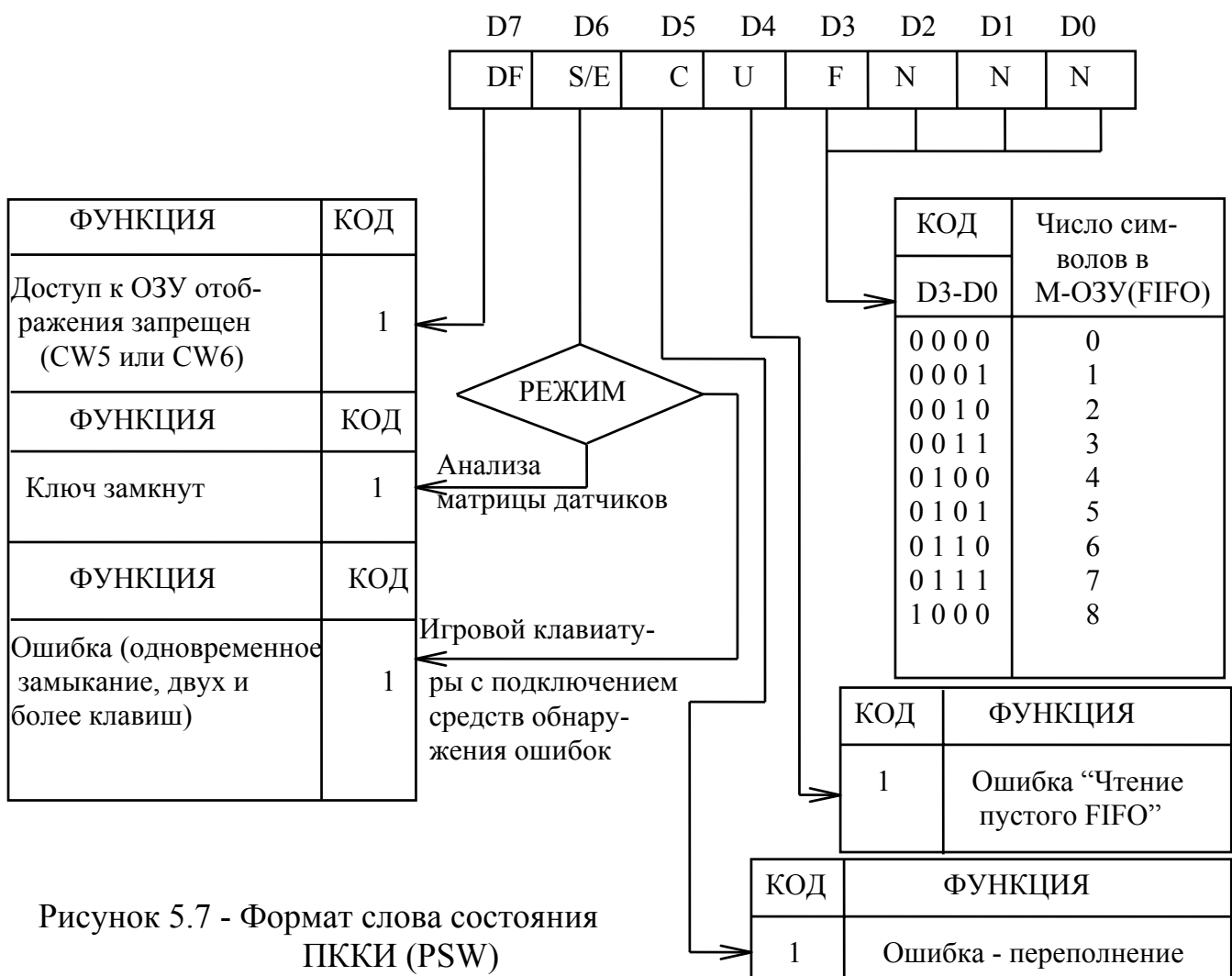


Рисунок 5.7 - Формат слова состояния ПККИ (PSW)

5.2 Программирование и режимы работы БИС ПККИ K580BB79

После включения питания ПККИ должен быть установлен в исходное состояние путем подачи на вход RESET сигнала высокого уровня длительностью не менее $6 \times T_{CLK}$. При этом обнуляются счетчик сканирования, RgA ОЗУ индикации, указатели стека М-ОЗУ, регистр слова состояния, двоичный 10-разрядный счетчик, в регистр коэффициента пересчета записывается число 31, на выходах INT, \sim BD, OUTA и OUTB устанавливается сигнал низкого уровня. Если аппаратный сброс (подача сигнала RESET) выполняется после записи управляющего слова CW0 “Установка режима работы интерфейса клавиатуры и индикации”, то в общем случае необходимо заново программировать режимы работы BB79, так как по умолчанию программируемый интерфейс настраивается на режим кодированного сканирования клавиатуры с запрещением ввода двух и более одновременно нажатых клавиш (режим стандартной клавиатуры), а также на вывод информации на 16-разрядную индикацию с разрешением ввода символов слева направо (начиная со старших разрядов). Также источником чтения назначается М-ОЗУ, что соответствует загрузке управляющего слова CW2. Отметим, что при аппаратном сбросе содержимое ячеек ОЗУИ и М-ОЗУ клавиатуры не обнуляется.

Начальная установка БИС BB79 (настройка) осуществляется загрузкой управляющего слова CW0 “Установка режима работы интерфейса клавиатуры и индикации”, причем значение на входе A0=1. Блоки интерфейса клавиатуры и индикации хотя и используют для сброса общий счетчик сканирования, функционируют независимо и устанавливаются после загрузки управляющего слова.

Программирование (настройка) ПККИ выполняется по командам OUT путем записи управляющих слов CW в регистры блока управления, после чего МПр может записывать командой OUT, читать командой IN информацию из ОЗУИ, М-ОЗУ клавиатуры и его слово состояния. Операции чтения/записи выполняются в зависимости от состояния управляющих сигналов \sim CS, \sim RD, \sim WR, A0 (таблица 5.2).

Таблица 5.2 - Операции чтения/записи

Входные сигналы A0 ~RD ~WR ~CS	В и д о п е р а ц и и
1 1 0 0	Запись команды RgCW := D7-D0
0 1 0 0	Запись данных ОЗУИ := D7-D0
1 0 1 0	Чтение слова состояния D7-D0 = Rg PSW
0 0 1 0	Чтение данных D7-D0 = M-ОЗУ v ОЗУИ
X X X 1	Микросхема не выбрана. Шина данных находится в третьем состоянии.

В ПККИ возможна запись восьми управляющих слов (таблица 5.3): три команды являются общими и используются во всех режимах; три CW управляют работой дисплея (индикацией) и два - режимами ввода, которые передаются по магистрали данных от МПр при ~CS=0 и A0=1 и загружаются по нарастающему фронту сигнала ~WR.

Таблица 5.3 - Команды процессора для установки режимов работы

Т и п команды	К о м а н д а (управляющее слово CW)	Код CW D7 D6 D5	Код операции D4 D3 D2 D1 D0
Общие команды	CW0: Установка режима работы интерфейса клавиатуры и индикации	0 0 0	T2 T1 K2 K1 KD
	CW1: Программирование скорости ввода-вывода (синхронизации)	0 0 1	P4 P3 P2 P1 P0
	CW6: Программный сброс	1 1 0	C4 C3 C2 C1 C0
Управле- ние работой дисплея	CW3: Чтение ОЗУ индикации	0 1 1	S A3 A2 A1 A0
	CW4: Запись в ОЗУ индикации	1 0 0	S A3 A2 A1 A0
	CW5: Запрет записи в ОЗУ индикации - гашение	1 0 1	X A B A B
Управле- ние вводом инфор- мации	CW2: Чтение М-ОЗУ клавиатуры (датчиков)	0 1 0	S X A2 A1 A0
	CW7: Снятие запроса на прерывание - подключение средств обнаружения ошибок	1 1 1	E X X X X

Рассмотрим форматы управляющих слов и преобразования, выполняемые в ПККИ ВВ79 при их загрузке.

При загрузке **CW0 “Установка режима работы интерфейса клавиатуры и индикации”** производится настройка клавиатуры на один из восьми режимов работы (таблица 5.4) и дисплея на один из четырех режимов (таблица 5.5). Формат CW0 приведен на рисунке 5.8.

Возможно задание одного из трех базовых режимов ввода информации:

1. Ввод данных с клавиатуры (последовательный опрос матрицы клавиш, включающий два режима);

- ◆ режим стандартной клавиатуры;
- ◆ режим “игровой” клавиатуры;

2. Последовательный опрос матрицы датчиков (или режим анализа матрицы датчиков);

3. Строблируемый ввод (или ввод по стробу - сигналу синхронизации выдачи данных с клавиатуры ~STB).

В каждом из режимов ввода возможны следующие два варианта управления работой сканирующих линий:

- ◆ режим кодирования (с дешифрированием состояния счетчика сканирования) - информация на линиях S3-S0 изменяется в двоичном коде;
- ◆ режим декодирования - информация на линиях S3-S0 изменяется в унитарном коде с бегущим нулем, и при этом возможна работа без внешнего дешифратора.

Таблица 5.4 - Кодирование режимов работы интерфейса клавиатуры

K2 K1 K0	Режим	Модификации режимов интерфейса клавиатуры	
0 0 0	Режим ввода с клавиатуры	Режим стандартной клавиатуры с блокировкой ввода двух и более одновременно нажатых клавиш	Кодирование линий S3-S0
0 0 1			Декодирование линий S3-S0
0 1 0		Режим “игровой” клавиатуры с разрешением ввода нескольких клавиш	Кодирование линий S3-S0
0 1 1			Декодирование линий S3-S0
1 0 0	Режим анализа матрицы датчиков	Кодирование линий S3-S0	
1 0 1		Декодирование линий S3-S0	
1 1 0	Режим стробированного ввода	Кодирование линий S3-S0	
1 1 1		Декодирование линий S3-S0	

Функционирование ПККИ в различных режимах будет рассмотрено ниже.

Таблица 5.5 - Режимы работы дисплея

T2 T1	Режимы работы интерфейса клавиатуры
0 0	Вывод 8-разрядных символов на индикацию с размещением их слева направо (начиная со старшего разряда индикатора)
0 1	Вывод 16-разрядных символов на индикацию с размещением их слева направо (начиная со старшего разряда индикатора) (устанавливается после аппаратного сброса по сигналу RESET)
1 0	Вывод 8-разрядных символов в один и тот же разряд со сдвигом ранее отображаемой информации влево (начиная с младшего разряда индикатора)
1 1	Вывод 16-разрядных символов в один и тот же разряд со сдвигом ранее отображаемой информации влево (начиная с младшего разряда индикатора)

D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	T2	T1	K2	K1	KD	CW0

KD = 0 - кодированный выход счетчика сканирования;

KD = 1 - декодированный выход счетчика сканирования.

K2 K1

0 0 - режим стандартной клавиатуры

0 1 - режим “игровой” клавиатуры

1 0 - режим анализа матрицы датчиков

1 1 - режим ввода по стробу

T1=0 - вывод на 8-разрядный индикатор

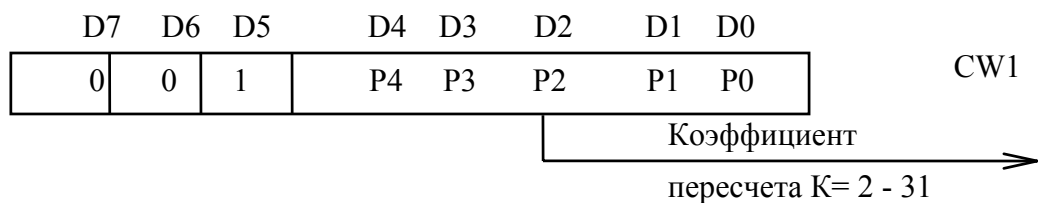
T1=1 - вывод на 16-разрядный индикатор

T2=0 - вывод цифр слева направо (начиная со старшего разряда дисплея)

T2=1 - вывод цифр справа налево (начиная с младшего разряда дисплея)

Рисунок 5.8 - Формат управляющего слова CW0 “Установка режима работы интерфейса клавиатуры и индикации

“Программирования скорости ввода-вывода (синхронизации)” CW1.



Загрузка данного управляющего слова выполняется при настройке ПККИ для согласования внешней синхросерии CLK микропроцессора с внутренней частотой БИС ВВ79, которая должна быть не более 100 кГц. Коэффициент деления частоты, размещаемый в разрядах D4-D0 CW1, должен обеспечивать внутреннюю синхронизацию ПККИ с частотой $F_{CLK}/K \leq 100$ кГц.

При аппаратном сбросе в регистр коэффициента пересчета заносится число 31, а при программном сбросе не изменяется. После загрузки CW1 происходит перенос коэффициента пересчета $K = 2 - 31$ в вычитающий счетчик с предустановкой в течение от 0 до $64 T_{CLK}$.

“Сброс” (программная очистка) CW6. Формат управляющего слова CW6 предусматривает несколько возможных вариантов сброса интерфейсов клавиатуры и индикации (рисунок 5.9):

- ◆ сброс интерфейса индикации:
 - а) запись в ОЗУИ константы гашения (пробела): $C4=1, C1-C0=0$;
 - б) загрузка константы гашения в буфер: $C4=0, C1-C0=0$;
- ◆ сброс интерфейса клавиатуры: $C4=0, C1=1, C0=0$;
- ◆ два варианта совместного сброса интерфейсов клавиатуры и индикации:
 - а) $C4=1, C1=1, C0=0$;
 - б) $C4=0, C1=0, C0=1$.

Сброс интерфейса индикации выполняется после загрузки CW6 формата 110.1УУ00, где УУ - код константы очистки, записываемый в ОЗУИ в течение цикла сканирования: все нули, все единицы или код 20h (цикл сканирования определяется форматом загруженного управляющего слова CW0: для декодированного выхода CTS цикл сканирования по умолчанию равен 4, а для кодированного выхода CTS - определяется разрядностью индикатора 8 или 16). Код константы, записываемый в ОЗУИ, определяется типом семисегментного индикатора (рису-

нок 5.2 - все нули или все единицы) или кодом символа (20h код пробела для КОИ8).

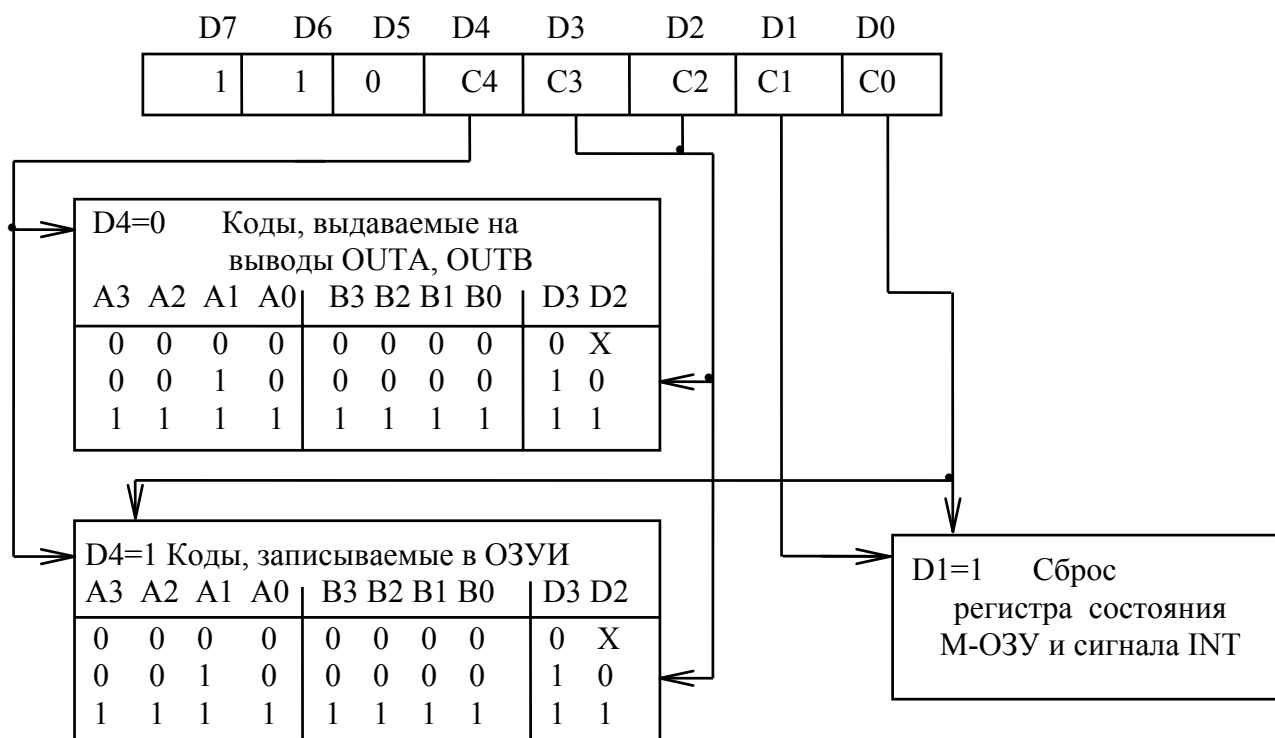


Рисунок 5.9 - Формат управляющего слова CW6 "Сброс"

При загрузке данного формата управляющего слова в слове состояния ПККИ бит 7 устанавливается в "1" и запрещается запись в ОЗУИ со стороны микропроцессора. В течение цикла сканирования в ОЗУИ записывается константа, определяемая значением бит D3, D2 CW6, а по окончании записи бит D7 PSW автоматически сбрасывается. Операция очистки занимает около 200 мкс, и процессор должен анализировать состояние разряда D7 слова состояния после загрузки команды CW6 "Сброс ОЗУИ", прежде чем будет выполнено следующее обращение к контроллеру для записи новой информации в ОЗУИ.

Если в разряде D4 CW6 будет записан нуль, то после загрузки управляющего слова в ПККИ на выходах OUTA и OUTB в зависимости от содержимого разрядов D3 и D2 появятся нули, единицы или код 20h, т.е. на время загрузки CW6 выполняется гашение индикатора и константа гашения записывается в специальный буферный регистр, которая в дальнейшем используется при гашении индикатора после загрузки CW5.

Сброс интерфейса клавиатуры выполняется при D1=1, D4=0, D0=0 в формате CW6. При этом в ПККИ сбрасывается регистр слова состояния М-ОЗУ, указатели стека FIFO и на выходе INT устанавливается сигнал низкого уровня, что соответствует ситуации: М-ОЗУ пусто и разрешена запись новых кодов клавиш. Единица в разряде D0 CW6 оказывает такое же действие, как единицы в разрядах D4 и D1.

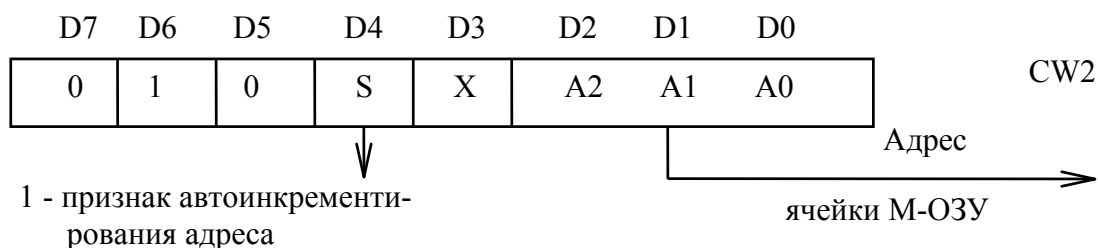
Преобразования, выполняемые при загрузке CW6 различных форматов, приведены в таблице 5.6.

Таблица 5.6 - Установка ПККИ при аппаратном и программном сбросе

Источник	RESET	Программный сброс CW6				
		CW0 1100NN10 1100NN00 1101NN00 110XNN01				
Режим работы	CW0 01000	Определ. командой	W	W	W	W
Коэффициент деления К	31	W	W	W	W	W
Счетчик сканирования	Сброс	W	W	W	W	Сброс
Источник чтения	М-ОЗУ	W	W	W	W	W
Регистр адреса ОЗУИ	0	0	W	W	0	0
Указатели стека М-ОЗУ	0	0	0	W	W	0
Слово состояния	0	0	0	W	W	0
Запрос INT	0	0	0	W	W	0
ОЗУ индикации	W	W	W	W	NN	NN
Код очистки на выходах OUTA, OUTB	0	W	W	NN	NN	NN

W - остается без изменения.

“Чтение М-ОЗУ (FIFO) клавиатуры (датчиков)” CW2.



В зависимости от режима работы, на который настроена БИС ПККИ, чтение содержимого М-ОЗУ клавиатуры производится различным образом. В клавиатурных режимах (последовательного опроса клавиатуры) и ввода по стробу данные из М-ОЗУ (FIFO) клавиатуры читаются после загрузки управляющего слова CW2, в разрядах D4-D0 которого может быть записана произвольная информация, так как в качестве адреса чтения используется указатель стека чтения FIFO.

В режиме анализа матрицы датчиков М-ОЗУ работает в режиме ЗУ с произвольным доступом, а в качестве регистра адреса используется RgA ОЗУИ и после загрузки CW2 содержимое М-ОЗУ читается микропроцессором по командам IN по адресу, указанному в разрядах D2-D0 CW2. Если разряд D4=1 в CW2, то после каждого чтения адрес увеличивается на “1”, т.е. выполняется автоинкрементирование и информация будет считываться из последовательных ячеек М-ОЗУ, иначе (S=0) чтение будет происходить из одной и той же ячейки до подачи новой команды CW2 с другим кодом адреса AAA.

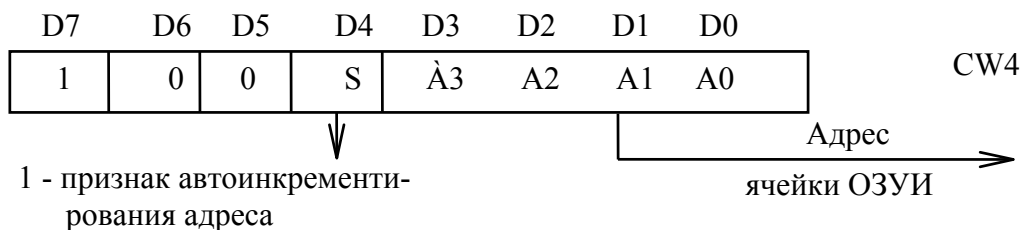
“Снятие запроса на прерывание - подключение средств обнаружения ошибки” CW7.

D7	D6	D5	D4	D3	D2	D1	D0	
1	1	1	E	X	X	X	X	CW7

В зависимости от режима, на который настроена БИС ПККИ, загрузка CW7 приводит к разным последствиям. В режиме анализа матрицы датчиков (E=X) после загрузки CW7 на выходе INT устанавливается низкий уровень сигнала и разрешается запись в М-ОЗУ клавиатуры (датчиков) (или просто снятие запроса на прерывание).

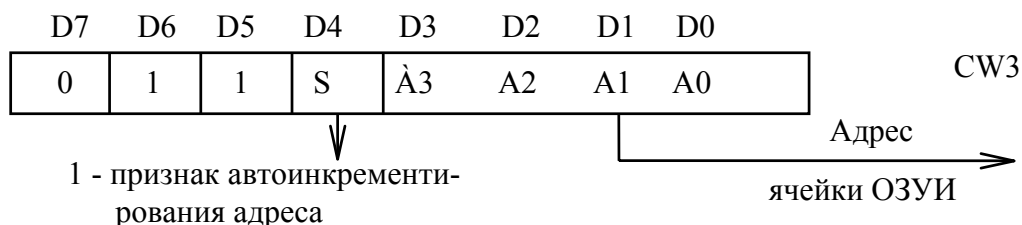
В режиме “игровой” клавиатуры с разрешением замыкания двух и более одновременно нажатых клавиш единица, записанная в разряд D4 (E) CW7, обеспечивает подключение специальных средств обнаружения ошибок, которые переводят ПККИ в режим, при котором фиксируется факт одновременного нажатия нескольких клавиш в одном цикле проверки надребезг контактов и устанавливают в “1” флаг ошибки (разряд D6 слова состояния), запрещающий запись кодов клавиш в М-ОЗУ клавиатуры и устанавливающий на выходе INT сигнал высокого уровня.

“Запись в ОЗУ индикации” CW4.



После загрузки CW4 микропроцессор по командам OUT может записывать информацию в ОЗУ индикации по адресу, указанному в разрядах D3-D0. Если разряд D4=1, то адрес ячейки после каждой операции записи увеличивается на единицу для адресации в следующую ячейку по принципу указателя стека.

“Чтение ОЗУ индикации” CW3.



Значения разрядов те же, что и в предыдущей команде. После загрузки CW3 микропроцессор по командам IN может читать информацию из ОЗУ индикации. CW3 устанавливает, помимо адреса ячейки, еще и внутренний переключатель, в результате чего источником считываемой информации становится ОЗУ индикации, а не М-ОЗУ клавиатуры. Загрузка CW2 сбрасывает этот переключатель, и источником чтения становится М-ОЗУ клавиатуры, т.к. чтение М-ОЗУ и ОЗУИ осуществляется микропроцессором по одной и той же команде IN с четным портом ввода-вывода (A0=0). Микропроцессор всегда будет читать информацию из источника, определяемого последней по времени командой CW3 “Чтение ОЗУ индикации” или CW2 “Чтение М-ОЗУ клавиатуры”. Также следует учитывать следующие особенности использования RgA ОЗУИ:

- ♦ адрес в RgA ОЗУИ может устанавливаться по трем командам: CW3, CW4 и CW2 (только в режиме анализа матрицы датчиков (АМД));

- ◆ при загрузке адреса можно использовать два режима чтения: с автоинкрементированием по принципу указателя стека и без изменения адреса чтения/записи (в режиме произвольного доступа).

Поэтому при совместном использовании этих команд необходимо учитывать следующее:

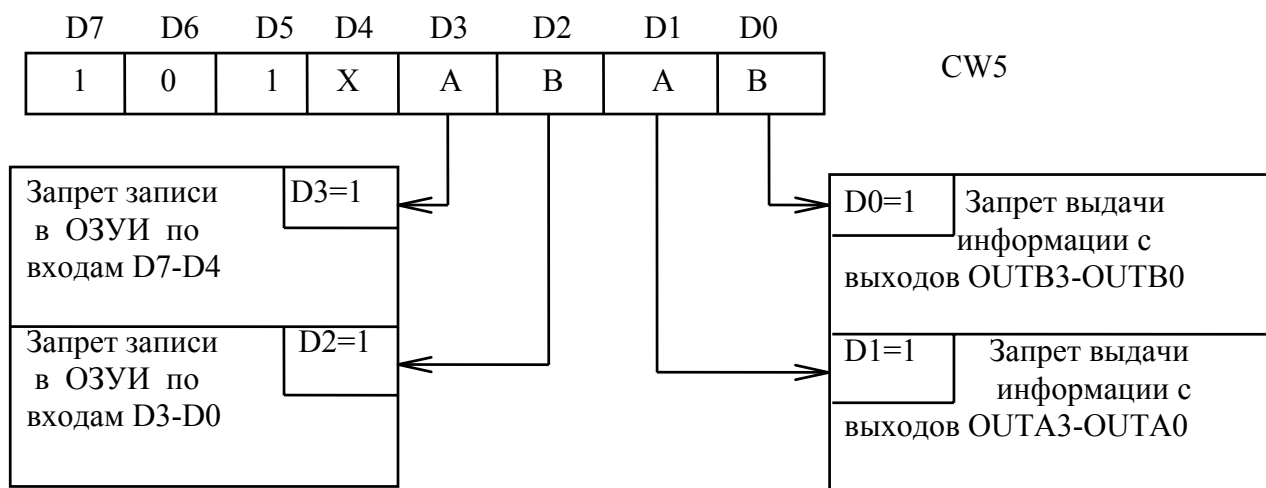
- ◆ для команд CW3, CW4, CW2 (в режиме АД) без автоинкрементирования необходимо перед командой IN задавать адрес чтения ОЗУИ или М-ОЗУ, а перед командой OUT - адрес записи ОЗУИ;
- ◆ команды CW3, CW4, CW2 (в режиме АД) с автоинкрементированием нельзя использовать в одном цикле (ППОП), т.к. при каждом чтении или записи адрес будет автоматически изменяться на “1”. Например, в режиме АД, если загружены CW2 и CW4 с автоинкрементированием, то адрес чтения М-ОЗУ будет определяться последней загруженной командой CW, а адрес записи в ОЗУИ будет на “1” больше адреса чтения.

Поэтому рекомендуется при использовании этих команд с автоинкрементированием сначала выполнить операции чтения М-ОЗУ всех переменных в цикле, а затем - запись массива данных в ОЗУИ. Во всяком случае недопустимо перемежать различные виды обмена (запись и считывание, обращение по чтению к ОЗУИ и М-ОЗУ клавиатуры) без предварительной отправки управляющего слова, устанавливающего код регистра адреса и внутренний переключатель чтения ОЗУИ и М-ОЗУ клавиатуры.

Также отметим, что в режимах вывода символов на 4- и 8-разрядный дисплей в ОЗУИ ячейки с адресами 4-15 и 8-15 соответственно не используются для отображения и при построении микроконтроллерных систем их может использовать микропроцессор для расширения емкости ОЗУ данных. Доступ к этим ячейкам возможен только по командам IN и OUT.

“Запрет записи в ОЗУ индикации - гашение” CW5.

При загрузке CW5 в ПККИ можно запретить запись информации в ОЗУ индикации по входам D7-D0 при наличии единиц в разрядах слова D3, D2 или запретить отдельно запись информации в каждую половину ОЗУ по входам D7-D4 магистрали данных (D3=1) или по входам D3-D0 (D2=1).



Кроме того, единица в разряде D1 CW5 запрещает выдачу информации (гашение индикации) из выходного регистра канала А (OUTA3-OUTA0), а единица в разряде D0 - выдачу информации из выходного регистра канала В (OUTB3-OUTB0). Команда применяется, например, при отображении данных на дисплее из 32 знакомест (две линейки индикаторов по 16 отображаемых символов) для вывода данных на одну из линеек индикатора или запрещения ввода соответственно младшей или старшей половины байта. При запрещении выдачи данных выходы OUTA и OUTB устанавливаются в 0000, 1111 или 00100000 (20h) в зависимости от того, какой код константы гашения был предварительно загружен командой CW6. Заметим, что если выдача информации запрещается по каналам А и В одновременно, то на выходе \sim BD появляется сигнал низкого уровня.

Отмена режима отображения данных, установленного командой CW5, производится при выполнении этой же команды с инверсными значениями требуемых битов.

5.3 Работа ПККИ K580BV79

Работу ПККИ рассмотрим отдельно для блоков интерфейса клавиатуры (датчиков) и интерфейса индикации.

Контроллер можно настроить на один из трех базовых режимов работы клавиатуры (таблица 5.4, рисунок 5.8) и один из двух базовых режимов работы индикации (таблица 5.5).

1. В режимах стандартной и “игровой” клавиатуры осуществляется ввод кода сработавшей клавиши в матрице, причем коды выдаются в той последовательности, в которой схема анализа обнаруживает сработавшие клавиши.

В режиме стандартной клавиатуры осуществляется ввод только одной единственной клавиши из всего набора (этот режим также часто называют с запрещением ввода кодов двух и более одновременно нажатых клавиш). Если после анализа схемой устранениядребезга контактов (СУДК) на протяжении двух циклов сканирования матрицы (10,24 мс) других сработавших клавиш не было обнаружено, то в ПККИ формируется код клавиши, в разряд D7 которого может помещаться единица - признак кода КОИ-8, а единица в разряде D6 может выступать в роли признака верхнего/нижнего регистра клавиатуры. Разряды D7 и D6 (C0/~STB - управление и SH - верхний/нижний регистр) отражают состояние двух дополнительных клавиш, подключенных непосредственно к входам C0/~STB (D7) и SH (D6).

D7	D6	D5	D4	D3	D2	D1	D0
C0/~STB	SH	Номер строки		Номер столбца			

СУДК выполняет преобразование унитарного кода строки и столбца в двоичный и в разряды D5-D3 помещается номер строки матрицы замкнутой клавиши - значение S2-S0 счетчика сканирования, а в разряды D2-D0 - номер столбца матрицы замкнутой клавиши. Код данного слова заносится в FIFO (М-ОЗУ клавиатуры). При этом вырабатывается сигнал запроса на прерывание INT, информирующий о наличии информации в FIFO, если М-ОЗУ до этого было пусто. При каждом считывании из FIFO на выходе INT появляется сигнал низкого уровня, а затем, если очередь еще не пуста, - вновь запрос на прерывание INT.

Если М-ОЗУ (FIFO) полностью загружено, то запись нового слова не производится, а устанавливается признак переполнения (разряд D4) в слове состояния контроллера (рисунок 5.7).

Следует отметить, что в общем случае разряды D7 и D6 вводимого кода могут использоваться при вводе с расширенной клавиатуры, число клавиш которой может быть увеличено до 256.

В режиме стандартной клавиатуры схема анализа и устранениядребезга контактов выполняет следующие функции:

1. Определяет факт нажатия клавиши и формирует код этой клавиши, не записывая его в М-ОЗУ клавиатуры, а фиксируя в отдельном буферном регистре.

2. Затем в течение двух циклов сканирования (10,24 мс) проверяет, замкнута ли данная клавиша. Если замкнута, то ее код вводится в М-ОЗУ клавиатуры по адресу указателя стека записи.

3. В случае, если обнаружены другие нажатия клавиш, проверяет, отжаты ли они раньше, чем данная. При соблюдении этого условия нажатие считается одиночным и код этой клавиши вводится в М-ОЗУ. В противном случае, когда данная клавиша отжата раньше других, ее код игнорируется.

4. Если одновременно нажаты две или более клавиш, матрица сканируется до тех пор, пока не обнаружится ситуация, когда одна из клавиш остается прижатой, а остальные отжаты (пункты 1-3).

В режиме “игровой” клавиатуры (с разрешением ввода кодов N одновременно нажатых клавиш) коды клавиш вводятся в FIFO (М-ОЗУ) в том порядке, в котором они обнаружены при сканировании клавиатуры после двух циклов сканирования устранениядребезга контактов.

Существует разновидность режима “игровой” клавиатуры, который устанавливается при записи в контроллер CW7 “Снятие запроса на прерывание - подключение средств обнаружения ошибок”. В этом режиме контроллер дополнительно формирует сигнал ошибки, записываемый в бит D6 слова состояния ПККИ, в том случае, если во время одного из циклов проверки надребезг контактов будет обнаружено две и более одновременно нажатых клавиш. Установка флага ошибки S/E в слове состояния запрещает запись информации в М-ОЗУ (FIFO) и устанавливает сигнал прерывания INT=1. При обнаружении одиночного

срабатывания код клавиши вводится в М-ОЗУ и также формируется сигнал запроса на прерывание INT. Как видно данный режим является модификацией режима стандартной клавиатуры, но с запретом записи в М-ОЗУ и выработкой сигнала INT в двух случаях: при вводе кода клавиши в М-ОЗУ при одиночном нажатии и при обнаружении одновременного срабатывания нескольких клавиш. Данный режим удобно использовать для формирования звукового сигнала при обнаружении факта игнорирования ввода нажатых клавиш в М-ОЗУ. В ППОП предварительно необходимо прочитать в МПр слово состояния ПККИ и, если его бит D6=0 выполнить процедуру чтения кода клавиши в микропроцессор, иначе (D6=1) выдать на динамик звуковой сигнал и сбросить флаг ошибки в слове состояния ПККИ путем загрузки управляющего слова CW6 с битом D1=1 (сброс интерфейса клавиатуры), после чего снова разрешается запись в М-ОЗУ.

2. В режиме анализа матрицы датчиков сигналы, снимаемые с матрицы датчиков (состояния датчиков) через входы RET7-RET0, минуя СУДК, поступают непосредственно в М-ОЗУ в унитарном коде, доступ к которому разрешен в каждом цикле сканирования матрицы датчиков. Ввод состояний датчиков в М-ОЗУ осуществляется в обратном коде (с инверсией). Таким образом, в этом режиме схема анализа и устранениядребезга контактов отключена, а М-ОЗУ работает при чтении в режиме произвольного доступа или автоинкрементирования, при этом в качестве источника адреса записи в М-ОЗУ выступает счетчик сканирования (последовательный доступ), а чтения - регистр адреса ОЗУ индикации.

Состояния датчиков, подключенных ко входам C0/~STB и SH ПККИ, в этом режиме игнорируются. Так как состоянию каждого датчика соответствует один разряд, то в М-ОЗУ в каждом цикле опроса хранится отображение всей матрицы датчиков в виде бит-карты (в унитарном коде).

Сигнал прерывания INT=1 вырабатывается, если в конце цикла сканирования матрицы датчиков (CTS=7 или 15 при кодированном выходе или CTS=0111 - при декодированном) обнаружится, что хотя бы один датчик изменил свое состояние на противоположное по сравнению с предыдущим циклом сканирования. При этом дальнейшая запись в М-ОЗУ запрещается и в слове состояния PSW ПККИ устанавливается бит D6=1, а при чтении PSW в разрядах D3-D0 отображается номер строки матрицы в момент чтения (содержимое RgA ОЗУИ).

Работа ПККИ в этом режиме имеет ряд особенностей:

- ◆ при срабатывании двух и более датчиков в одной строке матрицы вырабатывается один сигнал запроса на прерывание INT, а если произошло несколько срабатываний в разных строках матрицы - вырабатывается K сигналов INT, где K - число строк, кроме нулевой, в которых хотя бы один датчик изменил свое состояние на противоположное;

- ◆ исключение составляет нулевая строка, в которой множественные изменения датчиков вызывают такое же число сигналов прерывания L и запретов записи в М-ОЗУ.

Таким образом, число сигналов INT составляет K срабатываний в строках с первой по седьмую плюс число сработавших датчиков в нулевой строке и минимум может составлять от одного до максимум $K=7$, $L=8$, т.е. $N=K+L=15$.

Сигнал INT может быть снят и запись в М-ОЗУ разрешена в следующих ситуациях:

- ◆ при загрузке управляющего слова CW7 независимо от значения признака автоинкрементирования установленного в CW2;

- ◆ после выполнения первой команды чтения IN в микропроцессор, если в RgA при загрузке CW2 режим автоинкрементирования не был установлен;

- ◆ после выполнения N команд чтения IN в микропроцессор, если в RgA при загрузке CW2 режим автоинкрементирования был установлен, т.е. число команд чтения должно равняться числу сигналов прерывания, выработанных при вводе нового состояния матрицы датчиков.

При снятии сигнала запроса на прерывание бит D6 в PSW автоматически сбрасывается. Эти свойства необходимо учитывать при составлении ППОП в зависимости от области применения и количества датчиков в матрице. Кроме матрицы датчиков, ко входам порта ввода RET7-RET0 можно подключать любые логические схемы, работающие под управлением сигналов счетчика сканирования (например, до 8 мультиплексированных портов ввода сигналов с объекта управления, состояние которых будет периодически анализироваться контроллером (рисунок 5.10).

3. Режим ввода по стробу предназначен для работы с матрицами клавиатуры, использующих эффект Холла или явления ферромагнетизма, для снятия информации с которых требуются стробирующие импульсы.

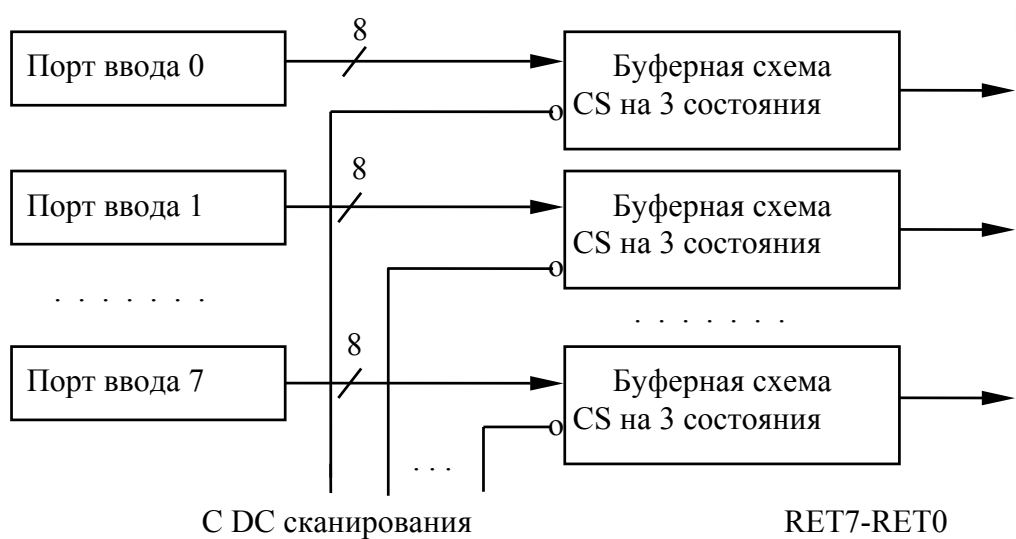


Рисунок 5.10 - Схема подключения портов ввода датчиков

В этом режиме с выходов матрицы клавиатуры снимается код клавиши и через линии RET7-RET0 записывается в буфер возврата по срезу сигнала на входе C0/~STB, длительность которого должна быть не меньше периода сигнала синхронизации CLK, а по фронту импульса строба вводимый код переписывается в М-ОЗУ, минуя схему анализа и устранениядребезга контактов в инверсном коде. В этом режиме обращение к М-ОЗУ происходит также по принципу очереди FIFO на 8 слов и при записи кода клавиши в М-ОЗУ формируется сигнал запроса на прерывание INT. При этом вход C0/~STB используется для подачи стробимпульса, а вход SH не используется.

Таким образом, выделим особенности формирования сигнала запроса на прерывание INT в различных режимах:

♦ в режимах стандартной, “игровой” клавиатуры и ввода по стробу высокий уровень сигнала INT является признаком того, что в М-ОЗУ, работающем по принципу FIFO, есть информация, которую надо прочесть микропроцессором. В режиме программного опроса (прерывания в МПр запрещены) наличие информации в М-ОЗУ можно проанализировать путем чтения слова состояния ПККИ (би-

ты D3-D0), а при аппаратной реакции на прерывание, например, с использованием БИС ПКП K1810BH59A;

- ♦ при считывании кода клавиши из FIFO в этих режимах сигнал INT сбрасывается и снова становится активным, если в FIFO имеется еще непрочитанная информация;

- ♦ в режиме “Анализа матрицы датчиков” появление сигнала INT указывает на изменение состояния датчиков, а в режиме “Подключение средств обнаружения ошибок” на ввод кода клавиши в М-ОЗУ или одновременное срабатывание двух и более клавиш. При этом в обоих режимах запись в М-ОЗУ блокируется (в последнем случае только при обнаружении двух и более одновременно нажатых клавиш).

В этих режимах целесообразно использовать сигнал INT в качестве сигнала прерывания для запуска подпрограммы анализа возникшей ситуации.

Также отметим, что при чтении слова состояния ПККИ в МПр разряды D4 и D5 автоматически сбрасываются.

При работе блока интерфейса клавиатуры микропроцессор параллельно может выполнять загрузку управляющих слов CW2, CW3, CW4 (таблица 5.3) и по командам OUT и IN выполнять операции обмена информацией с клавиатурой и дисплеем.

Рассмотрим функционирование **блока интерфейса индикации**.

Дисплейная часть (индикаторы) рассчитана на обслуживание “мультиплексированных секций без встроенной памяти”, например, цифровой индикатор калькулятора. Каждая дисплейная секция имеет информационные входы и разрешающий вход (рисунок 5.2). Информация на индикаторе отображается в виде знака только при подаче разрешающего сигнала (анод).

Необходимый для отображения знака код на информационных входах секции определяется типом дисплейной секции (рисунки 5.2 и 5.4).

Мультиплексированные дисплейные секции позволяют использовать один информационный канал для поочередного обслуживания всех дисплейных секций (рисунок 5.3) путем подачи кода символа параллельно на входы всех индикаторов, а выбор отображаемого символа осуществляется сигналом выбора индикатора с декодированных выходов счетчика сканирования (регенерация изображения),

что за счет инерционности зрения создает эффект непрерывной работы всех индикаторов.

Отображаемые коды хранятся в ОЗУ индикации (регенерации), из которого циклически считываются и передаются на входы индикаторов коды символов под управлением счетчика сканирования (источником адреса чтения ОЗУИ и регенерации изображения на индикаторе является счетчик сканирования).

В структуре контроллера BB79 можно выделить ОЗУ индикации емкостью 16 слов x 8 бит (рисунок 5.6), регистр адреса RgA и выходные регистры OUTA и OUTB. Выходные сигналы счетчика сканирования управляют выборкой информации из ОЗУИ и после каждого обращения инкрементируют код адреса CTS, что обеспечивает процесс регенерации изображения на индикаторах. В моменты изменения состояния CTS вырабатывается сигнал гашения ($\sim BD$) длительностью 160 мкс, используемый для гашения индикатора при смене символов.

Заметим, что ОЗУИ не связано аппаратно по данным с клавиатурной частью контроллера. Обмен данными с ОЗУ индикации возможен только через МПр при выполнении команд обмена данными ($A0=0$) с контроллером. Из М-ОЗУ данные считываются в микропроцессор в виде кода клавиши, а в зависимости от решаемой задачи код клавиши необходимо преобразовать программными средствами в требуемый код символа для отображения на индикаторе. Код отображаемого символа может быть представлен в одном из следующих вариантов:

- ◆ в семисегментном коде символа, не требующего дальнейшего преобразования для вывода на индикатор;
- ◆ в виде кода клавиши (код клавиши и символа совпадают);
- ◆ в стандартном коде представления символов в микроЭВМ - код КОИ-8;
- ◆ в виде любого двоично-десятичного или шестнадцатеричного кода символа (цифровой код).

Преобразования кода клавиши или выводимой из МПр выходной информации в код символа выполняются в микропроцессоре программно с помощью специальной программы кодогенератора или аппаратно при записи данных из МПр в ОЗУИ на основе ПЗУ кодогенератора или при чтении кода клавиши из М-ОЗУ в МПр. В зависимости от способа представления кода символа в ОЗУИ использу-

ются и различные способы подключения линеек индикаторов к выходным линиям OUTA, OUTB контроллера (рисунки 5.11-5.13).

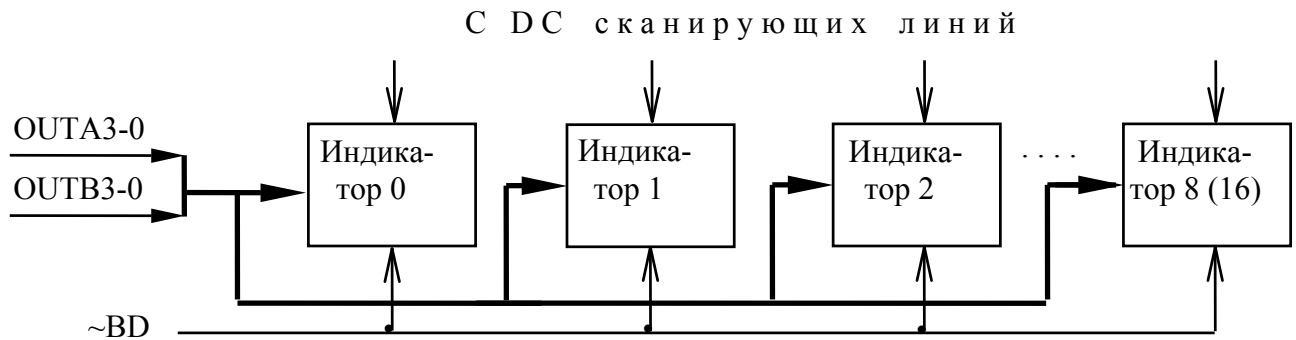


Рисунок 5.11 - Схема подключения дисплея без знакогенератора

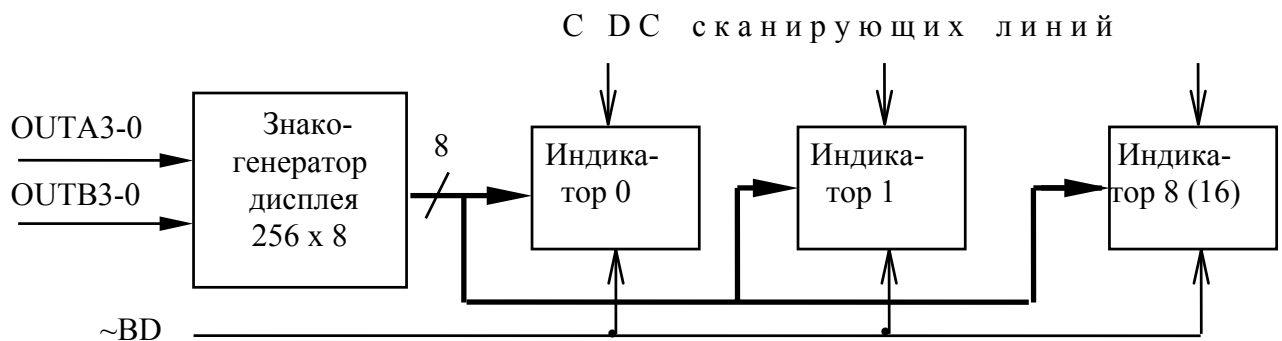


Рисунок 5.12 - Схема подключения дисплея со знакогенератором

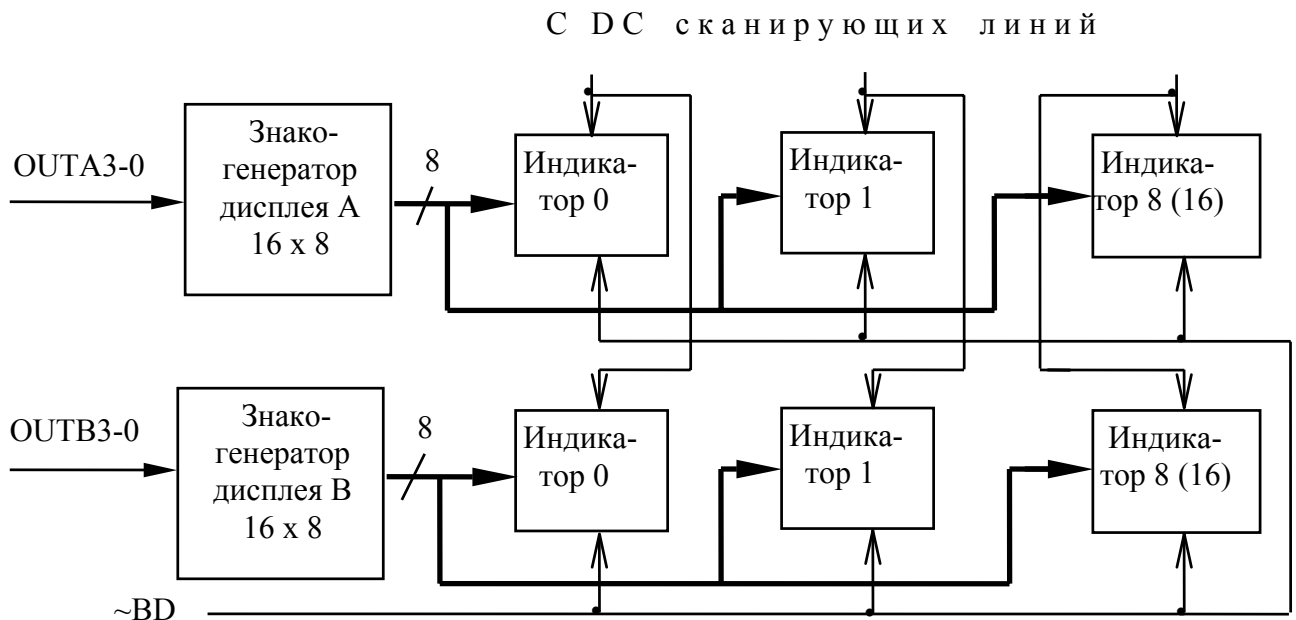


Рисунок 5.13 - Схема подключения двух линеек индикаторов

В первом случае (рисунок 5.11) код символа записывается из МПр в ОЗУИ в семисегментном коде и схема не требует наличия на выходах OUTA, OUTB знакогенератора. Во втором случае (рисунок 5.12) код символа представлен любым 8-разрядным кодом (КОИ-8, кодом клавиши и т.д.), что требует на выходах OUTA, OUTB знакогенератора емкостью 256 байт для преобразования кода символа в семисегментный код. И в третьем случае используются две линейки индикаторов и коды символов должны быть представлены любым кодом от одного до четырех бит (в двоичной, восьмиричной, шестнадцатеричной системах счисления или в любом двоично-десятичном коде). При этом запись кодов символов в ОЗУИ из МПр осуществляется отдельно по старшей или младшей тетраде с запретом записи в одну из половинок ОЗУИ (с загрузкой CW5), а на каждом из выходов OUTA, OUTB необходимо иметь по знакогенератору емкостью по 16 байт для преобразования кода символа в семисегментный код.

Коды символов из ОЗУИ выдаются синхронно с работой счетчика сканирования, и существует жесткое соответствие кода счетчика сканирования и адреса ячейки ОЗУИ, из которой выдается информация. Так как счетчик сканирования дешифрируется для выработки сигналов включения индикаторов дисплейных секций, то это обеспечивает отображение в каждой из секций содержимого одной и той же ячейки ОЗУИ (для вывода информации на индикатор со старших разрядов).

Настройка блока интерфейса индикации осуществляется при загрузке управляющего слова CW0 в зависимости от содержимого разрядов D4, D3 на режим 8-разрядной индикации, сканируемой за 5,12 мс, или 16-разрядную, сканируемую за 10,24 мс при внутренней частоте генератора 100 кГц. Также при настройке задается способ выдачи кодов со счетчика сканирования (выходы S3-S0) (таблица 5.4). При декодированном режиме линий сканирования S3-S0 информация появляется только в первых четырех разрядах дисплея, отменяя режим вывода на 8- и 16-разрядную индикацию. Также задается один из двух режимов вывода символов: слева направо (начиная со старшего разряда индикатора) или справа налево (начиная с младшего разряда индикатора).

Программным путем (загрузкой CW5 “Запрет записи в ОЗУ индикации - гашение”) можно запретить выдачу любой одной или обеих тетрад содержимого выходных регистров индикации OUTA, OUTB. При этом в случае запрета выдачи обеих тетрад сигнал низкого уровня на выходе \sim BD будет оставаться на время действия управляющего слова (режим гашения индикатора), а запрет записи в старшую или младшую тетрады ОЗУИ используется для организации вывода данных на две линейки индикаторов.

В режиме заполнения дисплея со стороны старших разрядов (вывод слева направо) каждому адресу ОЗУ индикации имеется прямое соответствие номера индикатора в секции.

Нулевой адрес ОЗУИ соответствует старшему разряду дисплея, а адрес 15 (или 7 - при отображении 8 символов) - младшему разряду дисплея (рисунок 5.14а). Последовательная запись информации в ячейки ОЗУИ с адреса 0 вызывает последовательное заполнение дисплейных секций со стороны старших разрядов. После заполнения всех 16 (8) ячеек ОЗУИ новая информация записывается опять в ячейку с адресом 0 и эта информация вновь отображается в старшем разряде дисплея и т.д.

Режим заполнения со стороны младших разрядов дисплея (справа налево) применяется в большинстве калькуляторов. Первая вводимая цифра помещается в младшем разряде дисплея, следующая - туда же, а ранее введенная цифра сдвигается в сторону старших разрядов (влево) и т.д. (рисунок 5.14б). Цифры, сдвигаемые из старшего разряда дисплея, теряются. При установке режима заполнения дисплея справа теряется прямое соответствие адресов ячеек ОЗУ индикации и номеров (позиции) индикаторов дисплея, в которых отображается информация, так как адрес чтения из ОЗУИ определяется как $CTS + N$, где N - число введенных символов в ОЗУ индикации.

При каждом вводе новой цифры в ОЗУИ в режиме автоинкрементирования адреса код RgA ОЗУИ изменяется на единицу, поэтому дисплей работает правильно только после предварительной установки адреса на нулевую ячейку ОЗУИ. Запись кода цифры в произвольную ячейку ОЗУИ в режиме вывода с младших разрядов индикатора с автоинкрементированием не рекомендуется, так как может вызвать непредсказуемый результат.

а) 16-разрядный дисплей, вывод слева направо

Адрес ОЗУ индикации							
0	1	2	3		13	14	15
1							
1	2						
1	2	3					

Выдача 1-го числа
Выдача 2-го числа
Выдача 3-го числа

1	2	3	4		14	15	16
17	2	3	4		14	15	16
17	18	3	4		14	15	16

Выдача 16-го числа
Выдача 17-го числа
Выдача 18-го числа

б) 16-разрядный дисплей, вывод справа налево

Адрес ОЗУ индикации							
1	2	3	4		15	0	
1)						1	

Адрес ОЗУ индикации							
2	3	4	5		0	1	
2)					1	2	

Адрес ОЗУ индикации							
3	4	5	6		1	2	
3)					2	3	

Адрес ОЗУ индикации							
0	1	2	3		14	15	
16)	1	2	3	4		15	16

Адрес ОЗУ индикации							
1	2	3	4		15	0	
17)	2	3	4	5		16	17

Адрес ОЗУ индикации							
2	3	4	5		0	1	
18)	3	4	5	6		17	18

в) 16-разрядный дисплей, вывод справа налево с использованием CW4

“Запись в ОЗУИ” с позиции номер 6

CW4 = 100 1 0110

Адрес ОЗУ индикации															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0
1)					1										

Адрес ОЗУ индикации															
2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	1
2)				1	2										

Адрес ОЗУ индикации															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16)	11	12	13	14	15	16	1	2	3	4	5	6	7	8	9

Адрес ОЗУ индикации															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0
17)	12	13	14	15	16	17	2	3	4	5	6	7	8	9	10

Рисунок 5.14 - Режим заполнения дисплея слева направо (а), справа налево (б), и справа налево с позиции 6

Запись данных в ОЗУИ без автоинкрементирования ведется в одну и ту же ячейку и отображается в одном и том же разряде дисплея.

Если запись в ОЗУИ начать не с адреса 0, то заполнение дисплея начинается с разряда, соответствующего адресу ячейки ОЗУИ, установленному SW4 “Запись в ОЗУИ”. Новые разряды будут появляться именно в этой позиции дисплея (для вывода справа налево), а ранее введенные будут сдвигаться по кольцу через старший разряд в нулевой разряд дисплея и пропадать при переходе к позиции, в которой появляются новые разряды (рисунок 5.14в), а при выводе слева направо новые разряды будут появляться в следующей позиции дисплея, а ранее введенные будут затираться новой информацией.

5.4 Применение ПККИ K580BB79

Программируемый контроллер клавиатуры и индикации широко применяется для организации ввода информации с клавиатуры или матрицы датчиков, а также для отображения результатов ввода и вычислений на дисплее, построенном на линейках индикаторов до 32 разрядов, при проектировании микроконтроллерных систем управления объектами.

В зависимости от запрограммированного режима работы клавиатуры и индикации можно получить большое число структур и типов матрицы клавиатуры или датчиков с различными характеристиками времени сканирования и индикации. На рисунке 5.15 приведен пример устройства ввода с матрицы клавиатуры 8 х 8 и клавиш “управления” C0/~STB и верхний/нижний регистр SH с 16-разрядной линейкой индикаторов, с записью кодов символов в ОЗУИ в семисегментном коде. Выполним настройку ПККИ на режим “игровой” клавиатуры с подключением средств обнаружения ошибок с кодированным выходом счетчика сканирования, а блок интерфейса индикации - на режим вывода символов на 16-разрядный дисплей с размещением цифр слева направо.

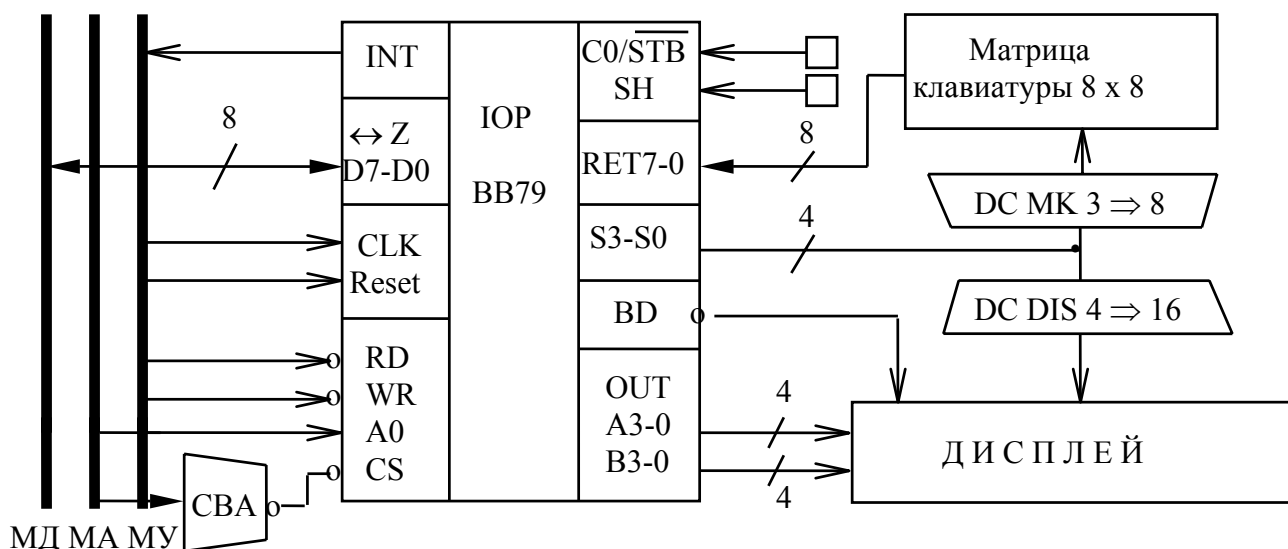


Рисунок 5.15 - Структурная схема подключения ПККИ к микропроцессору

Частота синхронизации в МПС $F_{CLK} = 2$ МГц. Адрес порта ПККИ равен 40h. Составить программу настройки ПККИ на заданный режим и текст ППОП для ввода кодов клавиш с выработкой звукового сигнала при нажатии двух и более одновременно нажатых клавиш и отображения введенных символов на индикатор. Адрес порта программируемого контроллера прерываний ВН59А равен 30h, таймера - 20h.

Настройка ПККИ:

```

MOV AL, 00001010b ; CW0
OUT 41h, AL ; загрузка CW0 - игровая клавиатура
MOV AL, 00110100b ; CW1
OUT 41h, AL ; загрузка CW1: K = 20 (2000/K <= 100 кГц)
MOV AL, 11010000b ; CW6
OUT 41h, AL ; загрузка CW6: сброс ОЗУИ константа = 00h
M1: IN AL, 41h ; чтение слова состояния ПККИ
AND AL, 80h ; формирование флага в PSW микропроцессора
JNZ M1 ; ожидание записи в ОЗУИ константы гашения
MOV AL, 10010000b ; CW4
OUT 41h, AL ; загрузка CW4 - запись ОЗУИ с автоинкрементом
MOV AL, 11110000b ; AL := CW7
OUT 41h, AL ; загрузка CW7 - подключение средств обна
; ружения ошибок
RET ; возврат из подпрограммы настройки ПККИ

```

Текст ППОП:

```

PROG: PUSH AX ; сохранение в стеке AX и BX
      PUSH BX
      IN AL, 41h ; чтение в AL слова состояния ПККИ

```

```

AND AL, 40h      ; выделение бита D6 в PSW
JZ  M2           ; если D6=0 - переход на ввод кода клавиши в МПр
MOV  AL, 00010000b ; загрузка CW в 0-ой канал таймера
OUT  23h, AL     ;
MOV  AL, 10h     ; загрузка начальных данных в канал 0 для вы-
OUT  20h, AL     ; работки звукового сигнала и запуск счета
MOV  AL, 11000010b ; загрузка CW6 для сброса интерфейса
OUT  41h, AL     ; клавиатуры: разрешения записи в М-ОЗУ
JMP  M3          ; на возврат из ППОП
M2:  MOV  AL, 01000000b ; загрузка CW2 "Чтение М-ОЗУ" в ПККИ
      OUT  41h, AL     ;
      IN  AL, 40h; чтение кода клавиши из М-ОЗУ типа FIFO
      LEA  BX, TABLE ; загрузка в BX начального адреса таблицы
      XLAT                                ; программного знакогенератора и преобразо-
                                ; вание кода клавиши в 7-сегментный код
      OUT  40h, AL     ; запись кода символа в ОЗУИ
M3:  MOV  AL, 00100000b ; загрузка OCW2 в ПКП ВН59А
      OUT  30h, AL     ;
      POP  BX          ; восстановление из стека BX и AX
      POP  AX          ;
      IRET             ; возврат из ППОП

```

Заметим, что команду XLAT можно заменить двумя эквивалентными командами для обращения к таблице программного знакогенератора:

```

ADD BX, AX      ; нахождение адреса 7-сегментного кода в TABLE
MOV  AX, [BX]   ; загрузка кода символа в AX

```

В программе необходимо зарезервировать массив из 256 байт с начальным адресом TABLE и выполнить прошивку программного знакогенератора для преобразования кода клавиши в семисегментный код символа. Если в ОЗУИ код символа записывается в коде КОИ-8, то в массиве TABLE необходимо заменить 7-сегментные коды символов на коды КОИ-8, а на выходах ОУТА, ОУТВ ПККИ поставить ПЗУ знакогенератора емкостью 256 байт с прошивкой преобразования кода КОИ-8 в 7-сегментный код:

Адрес ячейки ПЗУ 3Г в 8сс (код КОИ-8)	Код прошивки	Символ на дисплее
60	00111111	0
61	00000110	1
62	01011011	2

63	01001111	3
64	01100110	4

и т.д.

При записи в ОЗУИ считанного кода клавиши из М-ОЗУ в ППОП не требуется выполнять никаких преобразований, а только изменить прошивку в знакогенераторе на выходах ОУТА, ОУТВ ПККИ.

На рисунках 5.11-5.13 приведены примеры подключения дисплея к контроллеру. Из рассмотренного примера видно, что функции кодогенератора и знакогенератора может выполнять либо МПр путем программного преобразования кодов клавиш в 7-сегментный код или код представления символа в памяти микропроцессора (КОИ-8), либо на основе ПЗУ или ПЛМ. На рисунке 5.10 представлен пример подключения восьми портов ввода для режима анализа матрицы датчиков, а на рисунке 5.16 пример работы дисплея для вывода на 8-разрядный индикатор со старших и младших разрядов индикатора двух трехразрядных чисел с позиций 0 и 5.

Как видно из рисунка, попытка вывода второго числа в режиме вывода начиная с младших разрядов индикатора приводит к непредсказуемым последствиям, так как нет соответствия между адресом записи в ОЗУИ и номером знакоместа на линейке индикаторов.

Рассмотрим пример программного опроса ПККИ, когда прерывания в МПр запрещены для ввода кодов клавиш (режим работы - из предыдущего примера, но без подключения средств обнаружения ошибок) (программу настройки необходимо дополнить командой запрета прерываний в МПр CLI и убрать загрузку CW7):

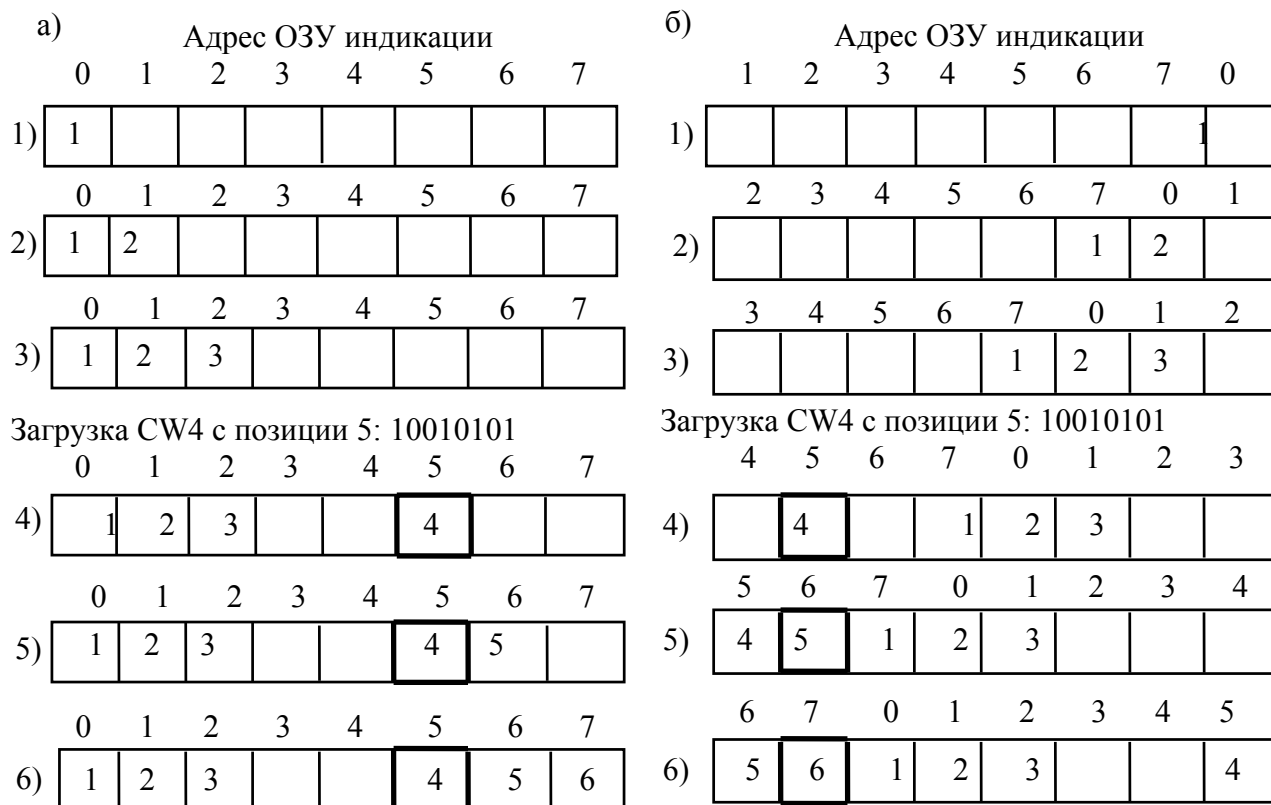


Рисунок 5.16 - Применение 8-разрядного дисплея для вывода со старших а) и младших б) разрядов индикатора двух 3-разрядных чисел

PO: PUSH AX ; сохранение в стеке AX, BX, CX
 PUSH BX
 PUSH CX
 IN AL, 41h; чтение слова состояния ПККИ
 AND AL, 0Fh ; выделение четырех младших разрядов PSW
 MOV CL, AL ; организация счетчика циклов кодов клавиш М-ОЗУ
 JZ M4 ; нет введенных в М-ОЗУ кодов клавиш
 MOV AL, 01000000b ; загрузка CW2 “Чтение М-ОЗУ” в ПККИ
 OUT 41h, AL ;

M5: IN AL, 40h; чтение кода клавиши из М-ОЗУ типа FIFO
 LEA BX, TABLE ; загрузка в BX начального адреса таблицы прог-
 XLAT ; рамного знакогенератора и преобразование
 ; кода клавиши в 7-сегментный код
 OUT 40h, AL ; запись кода символа в ОЗУИ
 DEC CL ; изменение счетчика циклов
 JNZ M5 ; возврат на ввод кодов клавиш в МПр пока CL ≠ 0

M4: POP CX ; восстановление из стека CX, BX, AX
 POP BX ;
 POP AX ;
 RET ; возврат из подпрограммы

Если код символа из МПр поступает в ОЗУИ в виде 4-разрядного двоично-десятичного или 16-ричного кода, то при выводе информации на нижнюю линейку индикаторов необходимо предварительно запретить запись в ОЗУИ по старшим разрядам байта путем загрузки CW5:

```
MOV AL, 10101000b ; загрузка CW5 с запретом записи по каналу A
OUT 41h, AL ;
```

A для вывода на верхний дисплей установить в CW5 запрет записи по каналу B, при этом код символа в байте должен находиться в старших разрядах байта.

При использовании режима “Анализ матрицы датчиков” необходимо учитывать, что в М-ОЗУ хранится бит-карта состояний матрицы датчиков, а выработка сигнала запроса на прерывание INT при обнаружении изменения состояния хотя бы одного из датчиков сопровождается запретом записи в М-ОЗУ. Отсюда на ППОП возлагаются следующие функции:

- ♦ в памяти микропроцессора необходимо хранить состояние матрицы датчиков из 8 байт до выработки сигнала запроса на прерывание INT (при настройке ПККИ на режим анализа матрицы датчиков этот массив должен быть обнулен);

- ♦ в ППОП необходимо считать в МПр новый массив состояния матрицы датчиков и выполнить сравнение одноименных строк матрицы с целью определения номера строки, в которой датчик изменил свое состояние на противоположное, и номеров сработавших датчиков в строке (сравнение можно выполнять последовательно по мере считывания строк состояния датчиков), а также в каком состоянии находится датчик (включено или выключено);

- ♦ при необходимости можно подсчитать число сработавших датчиков и вывести на индикатор или вывести номера сработавших датчиков, если их одновременное срабатывание не превышает 16 (по два индикатора на номер датчика при 32-разрядном дисплее);

- ♦ при считывании состояния матрицы датчиков из М-ОЗУ в режиме автоинкрементирования необходимо учитывать, что множественное срабатывание датчиков в нулевой строке плюс количество строк с 1-ой по 7-ую, в которых сработало произвольное число датчиков, вызывает выработку такого же количества сигналов INT и запретов записи в М-ОЗУ и при считывании восьми строк из М-ОЗУ сигнал INT может остаться активным, а запись новых состояний матрицы

датчиков на последующих циклах сканирования будет блокирована. Поэтому возможны следующие варианты для снятия сигнала запроса на прерывание INT и разрешения записи в М-ОЗУ:

- а) в ППОП выполнить загрузку CW7, которое сразу снимает сигнал INT и разрешает запись в М-ОЗУ;
- б) выполнить не 8, а 15 чтений из М-ОЗУ, после чего даже в самом худшем случае, когда одновременно сработают все датчики, сигнал INT будет снят. То есть для снятия сигнала запроса на прерывание INT в режиме автоинкрементирования необходимо выполнить столько чтений М-ОЗУ, сколько сигналов INT было выработано при сканировании и вводе состояний матрицы датчиков в М-ОЗУ (минимум одно чтение, максимум - 15 чтений М-ОЗУ);
- с) сначала выполнить чтение состояний всей матрицы датчиков в режиме автоинкрементирования, а затем загрузить CW2 без автоинкрементирования, и по первой же команде чтения из М-ОЗУ сигнал запроса на прерывание будет снят (или наоборот, сначала загрузить CW2 без автоинкрементирования и выполнить чтение М-ОЗУ, а затем CW2 с автоинкрементированием и считать оставшуюся часть строк М-ОЗУ).

Разрешение записи в М-ОЗУ по описанной методике можно выполнять с первых команд ППОП или перед командой IRET в зависимости от решаемой задачи, так как для быстродействующих систем реального времени могут быть потеряны состояния сработавших датчиков (за время выполнения ППОП датчик может переключиться дважды: включиться и выключиться).

На рисунке 5.17 приведен алгоритм работы клавиатурного интерфейса в различных режимах работы.

Многообразие режимов работы программируемого контроллера ВВ79 открывает широкие возможности его применения в зависимости от задач пользователя.

Контрольные вопросы для самопроверки

1. Перечислите режимы работы ПККИ ВВ79.
2. Какие типы индикаторов могут использоваться для вывода данных при применении БИС К580ВВ79 ?
3. Какую организацию имеет ОЗУ индикации и какие данные по форме представления в него могут записываться ?
4. Назначение слова состояния ПККИ и области его использования при организации процедур ввода с клавиатуры (датчиков) и вывода на дисплей ?
5. Перечислите управляющие слова, загружаемые в ПККИ, области и особенности их использования.
6. Объясните назначение управляющего слова CW7 “Снятие запроса на прерывание - подключение средств обнаружения ошибки”.
7. Как осуществляется ввод кода клавиши в режиме стандартной, “игровой” клавиатуры и ввода по стробу ?
8. Какие действия выполняются в интерфейсе контроллера индикации при загрузке управляющего слова CW6 формата с битом D4=1 или D0=1?
9. Чем отличается режим стандартной клавиатуры от режима "игровой" клавиатуры с подключением средств обнаружения ошибок?

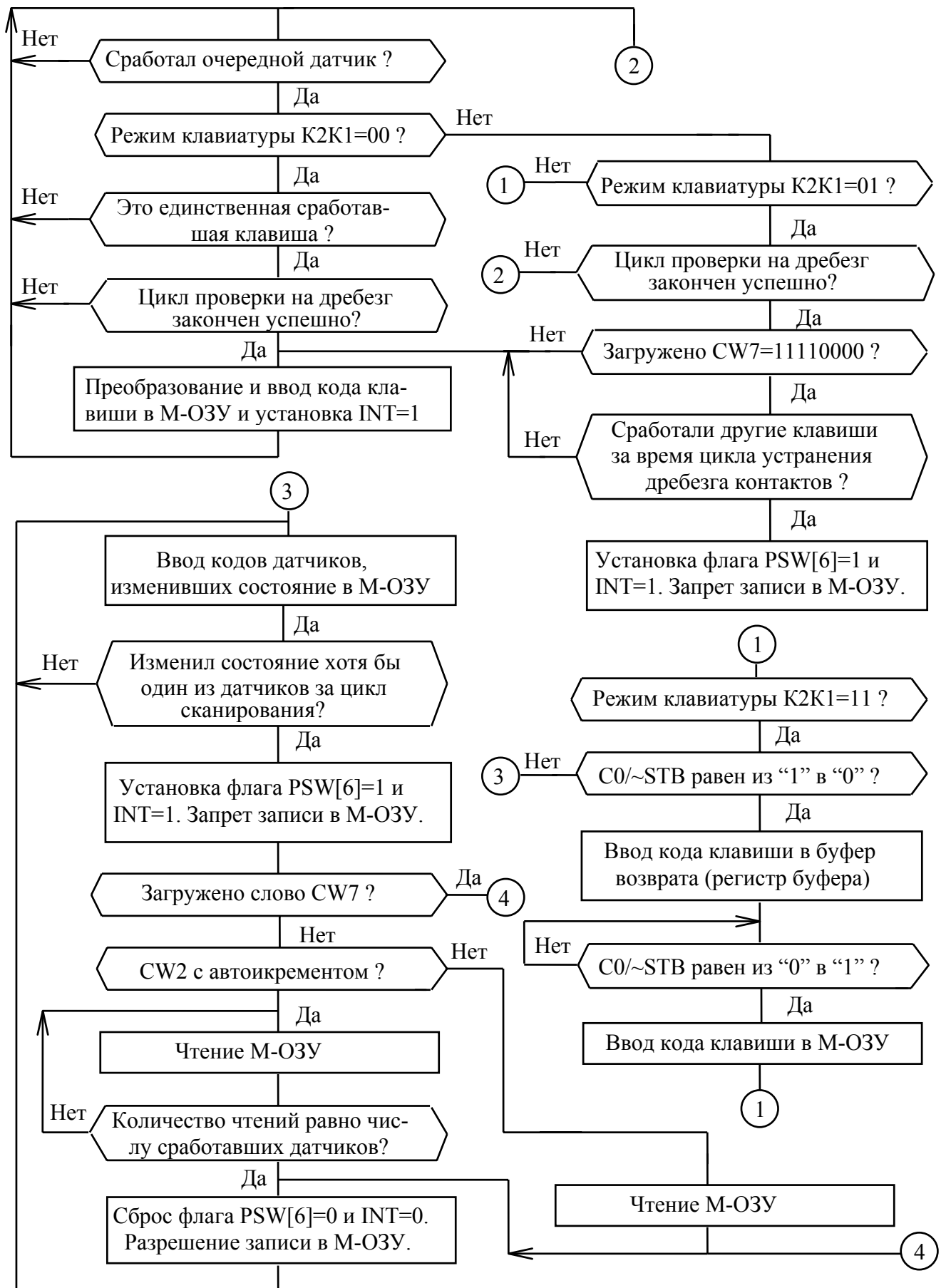


Рисунок 5.17 - Алгоритм работы интерфейса клавиатуры ПККИ

6 Проектирование микропроцессорных систем на основе однокристальных микропроцессоров

6.1 Базовые конфигурации микропроцессора ВМ86

Основная проблема, которую необходимо решить при проектировании МПС на основе однокристальных МкПр - это организация внешней системной шины (интерфейса МПС). Выходные схемы центрального процессора (ЦП) имеют ограниченную нагрузочную способность, кроме того, шина адреса и данных ЦП является совмещенной из-за ограничений по количеству выводов микросхемы. Поэтому для организации магистралей адреса, данных и управления могут потребоваться приемники и передатчики, регистры-защелки и схемы управления ими.

ЦП ВМ86 имеет 20 линий адреса и 16 линий данных, совмещенных с линиями адреса, вход синхронизации от ГИ и 16 линий управления, предназначенных для сигналов квитирования во время передачи данных и внешнего управления ЦП. Некоторые линии управления также являются совмещенными и используются для различных целей в зависимости от режима работы ЦП.

Главной структурной особенностью МПС является магистрально-модульный принцип их построения, при котором МПС разбивается на ряд функционально-законченных устройств - модулей. Связь между модулями осуществляется через единую внутрисистемную магистраль, что подразумевает общий для всех модулей состав шин (интерфейс), единый способ представления информации на них и общие правила организации процедур передачи информации по шине.

Микропроцессор К1810ВМ86 может работать в двух режимах, который задается уровнем сигнала на входе MN/~MX.

Минимальный режим (MN/~MX=1) предназначен для построения простейших однопроцессорных МПС, при котором сам МПр практически обеспечивает управление системной магистралью (не требуется дополнительных БИС шинного формирователя ВА86 и системного контроллера шины ВГ88).

Максимальный режим (MN/~MX=0) предназначен для построения мульти-процессорных систем различной конфигурации. Для организации системной шины на основе локальной шины микропроцессора требуется дополнительная логика на основе системного контроллера шины ВГ88, который выполняет функции управления системной шиной. Контроллер шины ВГ88 преобразует биты состояния ЦП ВМ86 ~S2~S0 в расширенный набор сигналов управления, т.е. формирует магистраль управления (МУ) шины для организации обмена между ОП, ПЗУ, ПУ, другими процессорами с ЦП (рисунок 6.1).

Регистр-защелка на основе БИС К580ИР82 служит для организации магистрали адреса (МА), так как локальная шина адреса/данных ЦП ВМ86 является совмещенной, шинный формирователь (ШФ) К580ВА86 служит для организации магистрали данных (МД) и перевода ее в третье состояние и работает под управлением контроллера шины ВГ88. На рисунке 6.1 показана структурная схема МПС в режиме максимального включения ЦП с сильно связанной конфигурацией с подключением математического сопроцессора ВМ87.

Биты состояния ~S2~S0 ЦП определяют тип выполняемой передачи:

000 - подтверждение прерывания: сигнал ~INTA;

001 - чтение из порта ввода/вывода: сигнал ~RDIO;

010 - запись в порт ввода/вывода: сигнал ~WRIO;

011 - останов: HALT;

100 - выборка команды: сигнал ~RDM и внутренний сигнал ЦП M1, который определяет цикл шины чтения памяти для выборки команды;

101 - чтение из памяти: сигнал ~RDM;

110 - запись в память: сигнал ~WRM;

111 - пассивный, определяет такт как холостой:

МД находится в z-состоянии, а преобразования выполняются в ЦП.

По сигналам состояния контроллер шины ВГ88 формирует набор интерфейсных сигналов для МУ и сигналов для управления другими БИС контроллера шины (ВА86 и ИР82):

ALE - показывает на наличие адреса на линиях AD из ЦП;

DEN - сигнал, информирующий, что ЦП готов выдавать или принимать данные через ШФ;

DT/~R - показывает направление передачи данных через ШФ: 1 - выдавать из ЦП, 0 - принимать данные в ЦП.

МПр ВМ86 может быть синхронизирован по отношению к сопроцессорам с помощью команды WAIT и входного сигнала ~TEST.

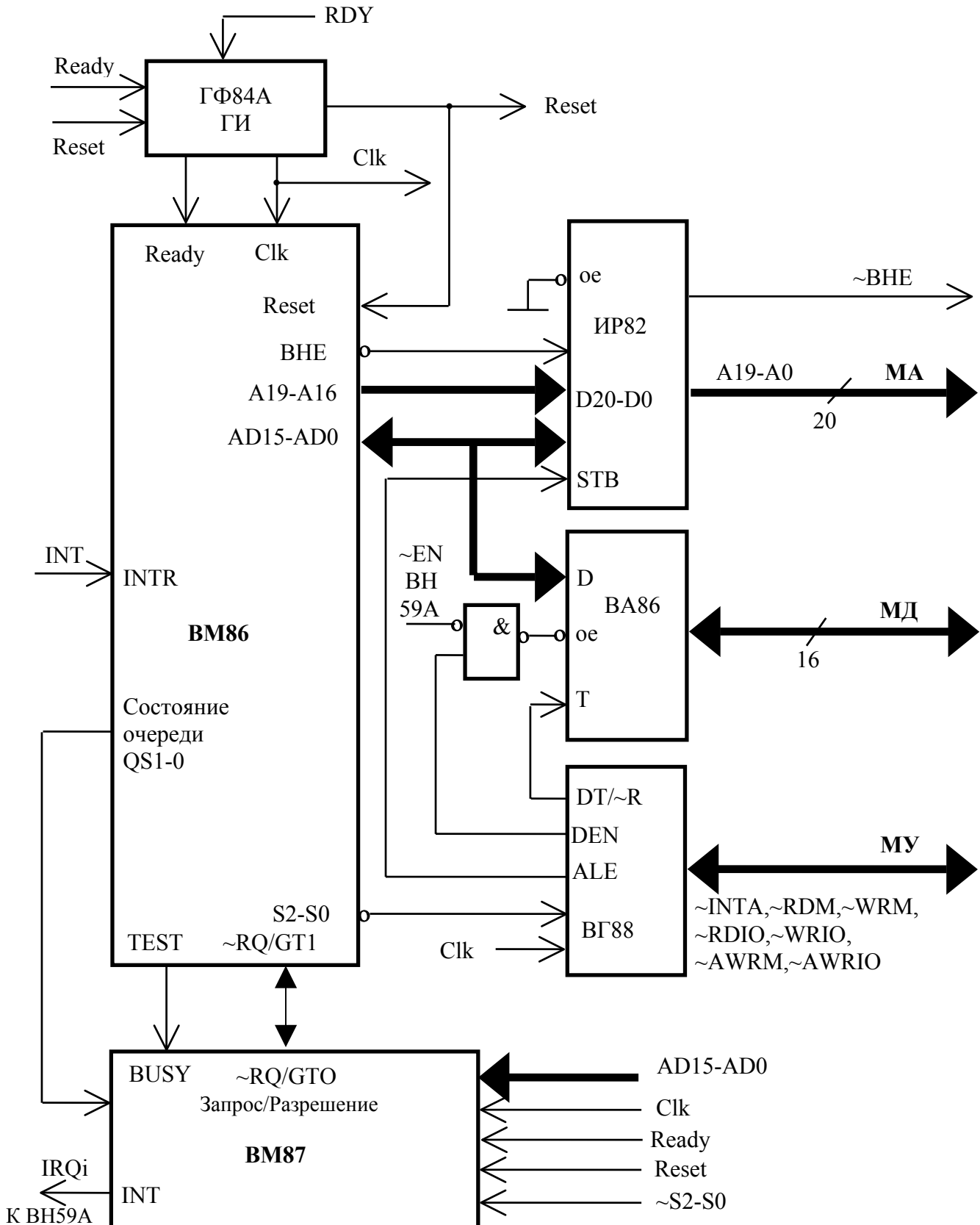


Рисунок 6.1 - Организация интерфейса микропроцессора ВМ86

На схеме рисунка 6.1 показано включение математического сопроцессора VM87 с сильно связанной конфигурацией, что позволяет значительно расширить возможности ЦП при обработке данных с ПЗ, двоично-десятичных чисел и целых чисел с повышенной разрядностью. Команды, передаваемые по шине данных, поступают на оба процессора одновременно, и формируется очередь команд. Пока ЦП выполняет свои команды, сопроцессор просто "прослеживает" их прохождение, не выполняя никаких действий до тех пор, пока не встречается команда для сопроцессора, каждая из которых в первом байте содержит КОП ESC=11011, а следующие три бита (X) в общем случае указывают на номер используемого сопроцессора при наличии нескольких сопроцессоров или на тип используемых данных (два бита поле MF и один бит X для КОП). Второй байт команды сопроцессора содержит поля:

mod Y r/m определяющие вид адресации и КОП совместно с

$\begin{array}{c} 0\ 0 \\ \hline \end{array}$ $\begin{array}{c} 0\ 1\ 0 \\ \hline \end{array}$ $\begin{array}{c} 1\ 0\ 1 \\ \hline \end{array}$ тремя или одним битами X байта ESC.

EA КОП EA или КОП при r/m=11 (EA указывает на способ формирования исполнительного адреса).

Первой командой очереди команд для сопроцессора является команда WAIT, следующей ESC с кодом операции в полях X и Y одной из команд сопроцессора LOAD, MUL, STO и т.д. Команда ESC обычно предваряется командой WAIT, которая переводит ЦП в состояние ожидания. Сопроцессор, приступая к выполнению своей команды, выставляет сигнал \sim BUSY занятости, который и вводит ЦП в состояние ожидания на время выполнения своей команды (вход \sim TEST ЦП).

При необходимости пересылки данных из сопроцессора в память сопроцессор посылает запрос на предоставление шины по линии \sim RQ/ \sim GT, а ЦП по этой же линии выдает сигнал разрешения на использования шины, т.к. ЦП имеет более высокий приоритет на использование шины. Закончив обмен, сопроцессор вновь посылает в ЦП сигнал по линии \sim RQ/ \sim GT, информируя об освобождении шины, снимает сигнал \sim BUSY и выводит ЦП из состояния ожидания.

В ряде случаев при выполнении команд в сопроцессоре могут возникать исключительные ситуации: переполнение, потеря значимости и т.п. Сопроцессор в таких ситуациях формирует сигнал запроса INT, который поступает на один из

входов контроллера прерываний ВН59А, а далее прерывание обрабатывается обычным способом с помощью подпрограмм обработки прерываний исключительных ситуаций.

В данной схеме к ЦП, кроме математического сопроцессора, можно подключить и другие разнотипные сопроцессоры, например, ввода-вывода, приоритет которого будет определяться местоположением сопроцессора по отношению к ЦП.

Физически оперативная память (ОП) для МПр ВМ86 организуется как два банка памяти по 512 К байт: один банк представляет старший байт слова (D15-D8), другой - младший (D7-D0). Для адресации ячеек памяти в каждом банке используются разряды A19-A1 магистрали адреса. Микропроцессор вырабатывает сигналы \sim ВНЕ и A0, используемые для выбора соответствующего банка памяти.

Младший байт адресуется при четном значении адреса A0=0, а старший, содержащий нечетно адресуемые байты, выбирается при \sim ВНЕ=0. Совместное использование сигналов \sim ВНЕ и A0 показывает, каким образом данные передаются по магистрали данных ЦП, для чего в структуру интерфейса МД входит схема перестановщика байтов:

	\sim ВНЕ A0		
Зп/Чт слова по четному адресу	0	0	AD15-AD0
(данные выравнены в ОП)			
Зп/Чт байта по четному адресу	1	0	AD7-AD0
Зп/Чт байта по нечетному адресу	0	1	AD15-AD8
Зп/Чт слова по нечетному адресу	1	1	AD7-AD0, AD15-AD8
(данные не выравнены по границе слова в ОП)			

Последняя комбинация используется для словных команд при невыравненных данных по границе слова и требует двух обращений к ОП. В первом цикле шины младший байт передается по линиям AD15-AD8, а во втором - старший байт по линиям AD7-AD0.

На рисунке 6.2 показана схема подключения к интерфейсу МПр ПЗУ и ОЗУ статического типа. Подключение периферийных БИС рассмотрено в разделах 3-5. Схема включает два селектора выбора адреса, которые формируют сигнал выбора

типа памяти $\sim CS$ (ПЗУ или ОЗУ в адресном пространстве МПр). ОП содержит два банка памяти, выбор которых осуществляется сигналами $\sim VNE$ и $A0$.

В состав интерфейса шины управления входит также логика подтверждения обмена, формирующая сигнал $\sim XACK$ - готовности данных на МД схемой задержки для медленно действующих устройств, если цикла шины недостаточно для обеспечения выборки или записи данных в память, и логика запрета, основанная на сигналах $\sim INH1$ и $\sim INH2$, первый из которых обеспечивает выбор требуемого для обмена устройства (ПЗУ, ОЗУ или ПУ), так как обращение к памяти и ПУ осуществляется по одним и тем же адресам, но для обращения к памяти используются интерфейсные сигналы $\sim RDM$, $\sim WRM$, а к ПУ - $\sim RDIO$, $\sim WRIO$. Сигнал $\sim INH2$ сигнализирует о занятости шины. Схемы одновибраторов ОБ служат для формирования сигнала готовности данных RDY для обеспечения требуемой длительности сигналов $\sim RDM$ и $\sim WRM$ при обращении к ПЗУ и ОЗУ.

6.2 Цикл шины центрального процессора ВМ86

Командный цикл процессора по отношению к системной магистрали можно представить состоящим из циклов шины для обмена данными между ЦП и другими устройствами, подключаемыми к общей шине (ОШ), и тактов, выполняемых в ЦП по преобразованию адресов и данных внутри ЦП, а ОШ во время их выполнения находится в пассивном (отключенном) состоянии.

Продолжительность цикла шины для ВМ86 составляет 4 такта $T1$ - $T4$ плюс неопределенное число тактов ожидания T_w , если время доступа к устройству для операций чтения и записи не обеспечивается длительностью цикла шины, а для старших моделей - два такта.

Таким образом, командный цикл процессора по отношению к интерфейсу ввода-вывода можно представить в виде: первый цикл шины - чтение команды, холостой такт для вычисления в ЦП исполнительного адреса, второй цикл шины - чтение операнда, два холостых такта на выполнение операции в ЦП и цикл шины записи результата.

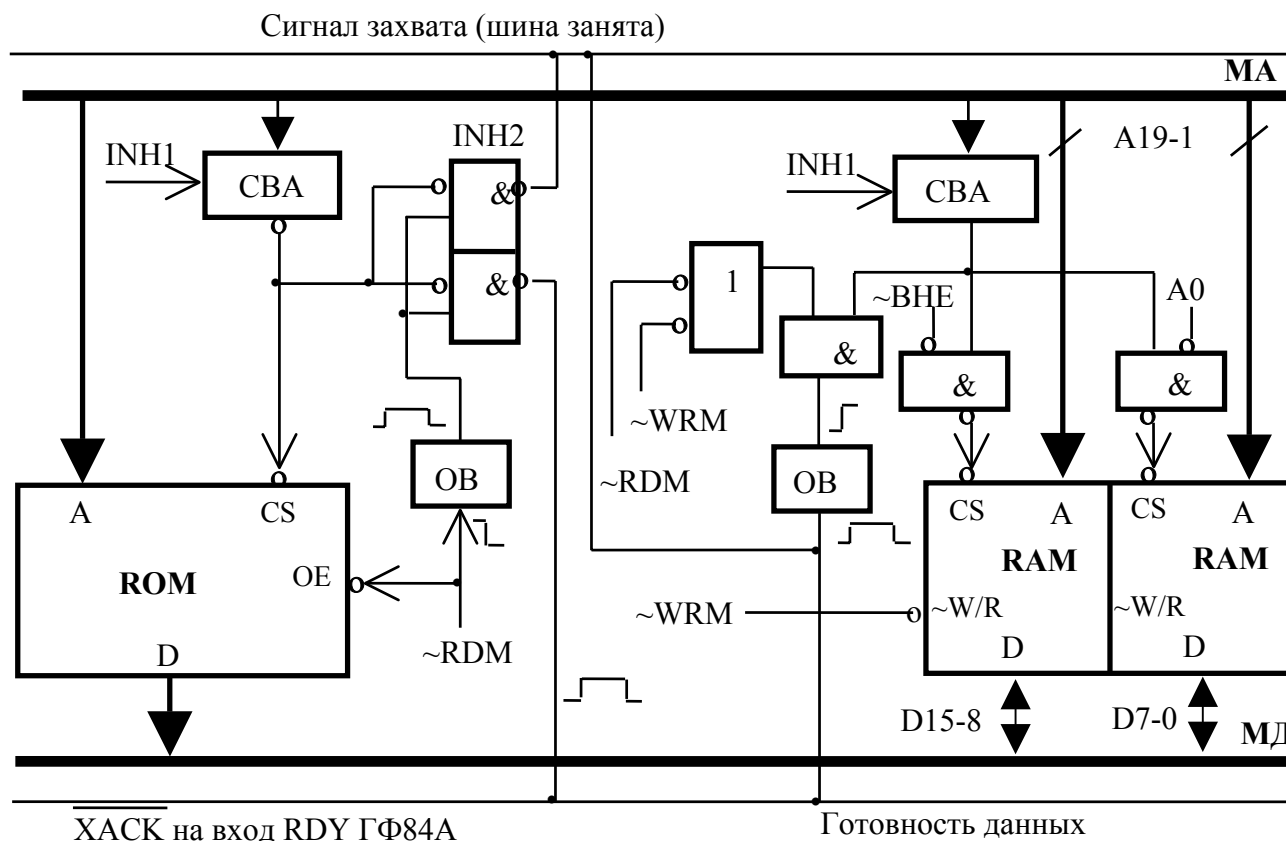


Рисунок 6.2 - Схема подключения ПЗУ и ОЗУ к интерфейсу МП86

На рисунке 6.3 приведены временные диаграммы циклов чтения и записи. В такте T1 на локальную шину адреса/данных МП86 всегда выдается адресная информация, а контроллер шины ВГ88 (КШ) выдает сигнал ALE, по срезу которого адрес фиксируется в регистре-защелке RgA ИР82. В такте T2 происходит переключение направления работы канала адреса/данных, т.е. адрес снимается и из ЦП выдается номер регистра кодового сегмента по линиям адреса A17-16 и формируется сигнал DEN для разрешения выходов из Z-состояния ШФ для выдачи или приема данных. При чтении во втором такте КШ формирует сигнал $\sim RD$. Передача данных происходит в тактах T3 и T4. В такте T3, если времени цикла шины достаточно для доступа к памяти или ПУ, данные помещаются на МД, и в такте T4 ЦП принимает данные в приемник, а КШ снимает сигнал $\sim RD$ и DEN. Приемник данных в ЦП определяется кодом команды и микропрограммой командного цикла процессора.

При записи в такте T1 выполняются те же действия, что и при чтении. В такте T2 ЦП выставляет данные для записи и удерживает их до такта T4. В такте T3 КШ формирует сигнал $\sim WR$, а в T4 снимает его и сигнал DEN для запрещения

ШФ. Сигнал DT/ \sim R при чтении имеет низкий уровень, а при записи высокий и определяет направление передачи данных через ШФ.

МУ ЦП позволяет управлять длительностью цикла шины при выдаче и приеме данных по МД. Для этого в ЦП служит вход READY (готовности ЦП или готовности данных). Если к началу такта T4 сигнал READY не поступил в ЦП, то он между тактами T3 и T4 вводит такты ожидания Tw, а в такте Tw выполняются те же действия, что и в такте T3. Таким образом увеличивается время доступа к памяти или ПУ на время, кратное одному такту. Для реализации данной процедуры устройство, к которому осуществляется обращение, должно формировать сигнал готовности данных, который подается на вход RDY генератора импульсов (ГИ) ГФ84А, а в ответ на него ГИ формирует сигнал READY для ЦП.

На временной диаграмме также показаны сигналы опережающей записи в память и ПУ. Они выполняют те же функции, что и сигналы \sim WRM и \sim WRIO, но выдаются на один такт синхронизации раньше.

В зависимости от типа используемого интерфейса и модели ЦП (структуры ЦП) выделяют и различное количество типов цикла шины. Например, для шины ISA существует 4 индивидуальных типа циклов:

- ◆ доступ к ресурсу при обмене данными между ЦП и другими устройствами;
- ◆ прямого доступа к памяти (ПДП), если контроллер ПДП является задатчиком для выполнения обмена между памятью и ПУ;
- ◆ регенерации для регенерации динамической памяти;
- ◆ захвата шины от внешнего модуля, чтобы стать задатчиком (ведущим) на шине.

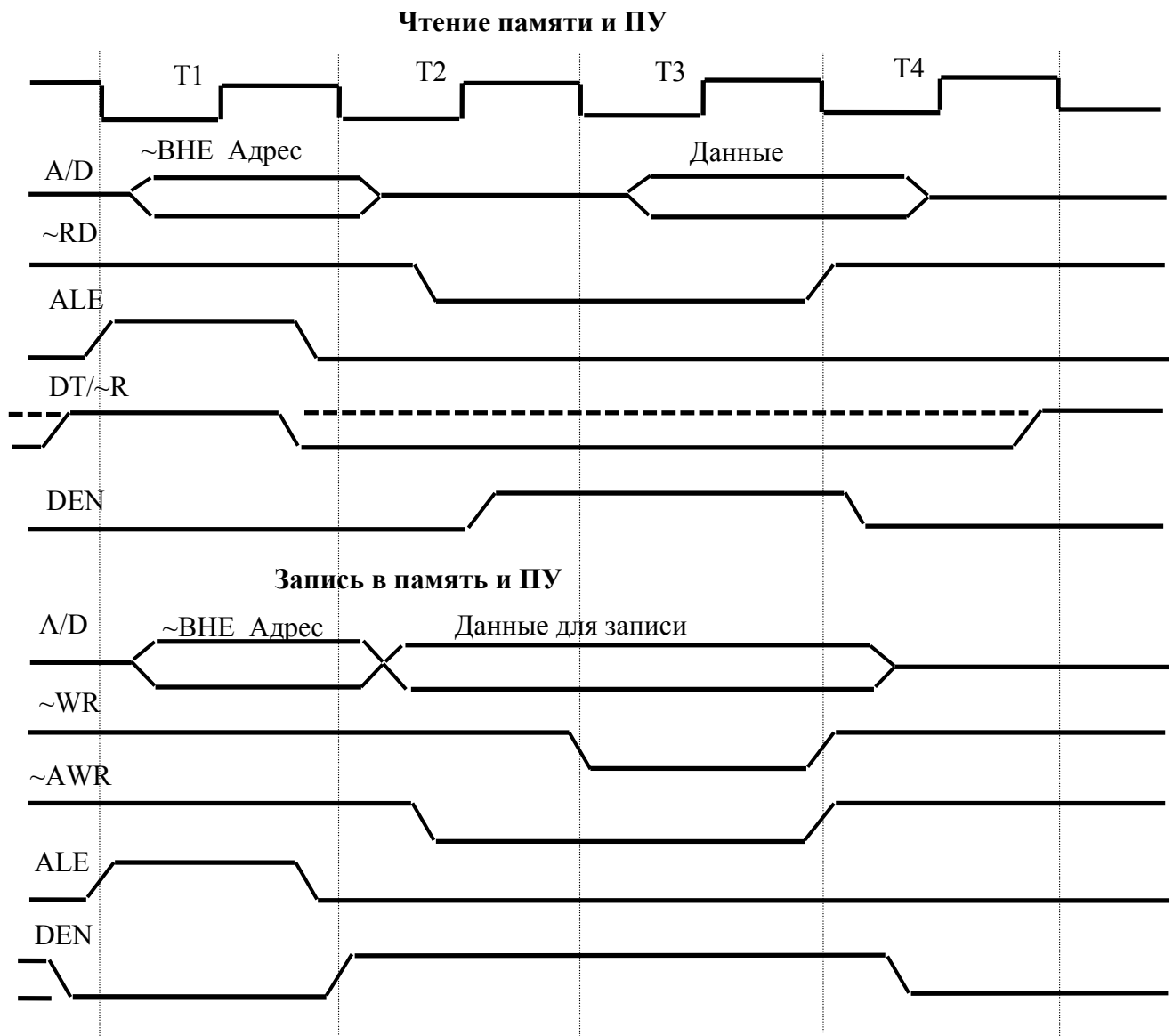


Рисунок 6.3 - Временные диаграммы циклов чтения и записи МПр

Структурно циклы отличаются по типу задатчиков на шине и видами ресурсов доступа к ней, а внутри - продолжительностью цикла.

Для цикла доступа к ресурсу существует 3 вида циклов:

- ◆ цикл с 0 тактов ожидания: самый короткий цикл для организации пакетного доступа к памяти при расслоении обращений ОП;
- ◆ нормальный цикл, не требующий формирования сигнала готовности данных;
- ◆ удлиненный цикл с формированием сигнала готовности данных.

В циклах ПДП и регенерации существует два вида циклов: нормальный и удлиненный.

В старших моделях центральных процессоров нормальный цикл шины включает два такта, а удлиненный цикл организуется аналогично ЦП ВМ86 путем многократного повторения такта T2 до формирования сигнала готовности данных RDY. При этом количество различных типов циклов шины резко возрастает, что связано с использованием оперативной памяти с расслоением обращений с конвейеризацией доступов, наличием внутренней и внешней КЭШ памяти и другими особенностями структуры процессоров.

Контрольные вопросы для самопроверки

1. Как организуется связь БИС ЦП ВМ86 с внешними устройствами?
2. Какие функции выполняют БИС ИР82, ВА86 и ВГ88 в составе контроллера шины?
3. Поясните работу ЦП при выполнении циклов чтения и записи шины при подключении ПУ и памяти.
4. Перечислите известные вам типы циклов шины и их назначение.

Литература

1. Бакшаев А.М. Практикум по микропроцессорным периферийным БИС с использованием диалоговых систем управления.- Киров, КирПИ, 1992.- 118с.
2. Страбыкин Д.А., Бакшаев А.М. Практикум по микропроцессорным устройствам с использованием диалоговых систем управления.- Горький, изд. ГГУ, 1989.- 105 с.
3. Казаринов Ю.М., Номоконов В.Н., Филлипов Ф.В. Применение микропроцессоров и микроЭВМ в радиотехнических системах. Учеб. пособие для радиотехнич. спец. вузов.-М.:Высш. шк., 1988.-207 с.: ил.
4. Мик Дж., Брик Дж, Проектирование микропроцессорных устройств с разрядно - модульной организацией: В 2-х кн. Пер с англ.-М.: Мир,1982. - 480 с.

5. Микропроцессоры и микропроцессорные комплекты интегральных микросхем: Справочник: В 2 т./В.-Б.Б.Абрайтис, Н.Н.Аверьянов, А.И.Белоус и др.; Под ред. В.А.Шахнова.- М.: Радио и связь, 1988.-Т.1.- 386с.
6. Микропроцессоры и микропроцессорные комплекты интегральных микросхем: Справочник: В 2 т./ Н.Н.Аверьянов, А.И.Березенко и др.; Под ред. В.А.Шахнова.- М.: Радио и связь, 1988.-Т.2.- 386с.
7. Микропроцессоры: В 3 т. Архитектура и проектирование микроЭВМ. Организация вычислительных процессов: Учебник для втузов / П.В.Нестеров, В.Ф. Шаньгин, В.Л.Горбунов и др.; Под ред. Л.Н.Преснухина, М.: Высшая школа. 1986.- Т.1.- 495с.
8. Микропроцессоры: системы программирования и отладки./ В.А.Мясников, М.Б.Мясникова, М.Б.Игнатьева. - М.: Энергоатомиздат, 1985.- 272с.
9. Микропроцессоры и микроЭВМ в системах автоматического управления: Справочник./ С.Т.Хвощ, Н.Н.Варлинский, Е.А.Попов; Под общ.ред. С.Т.Хвоща.- Л.: Машиностроение. Ленинград. отделение, 1987.- 640 с.
10. Самофалов К.Г.,Викторов О.В. Микропроцессоры.- Б-ка инженера.- К.: Техника, 1989.- 312 с.
11. Щелкунов Н.Н., Дианов А.Н. Микропроцессорные средства и системы.- М.: Радио и связь, 1989.- 288 с.
12. Майоров В.Г., Гаврилов А.И. Практический курс программирования микропроцессорных систем.- М.: Машиностроение, 1989.- 272 с.
13. Лю Ю-Чжен, Гибсон Г. Микропроцессоры семейства 8086/8088. Архитектура, программирование и проектирование микрокомпьютерных систем: Пер. с англ.- М.: Радио и связь, 1987.- 512 с.