

Министерство образования Республики Беларусь

Учреждение образования
«Гомельский государственный технический
университет имени П.О. Сухого»

Кафедра «Промышленная электроника»

ПРОЕКТИРОВАНИЕ МИКРОПРОЦЕССОРНОЙ СИСТЕМЫ КОНТРОЛЯ И УПРАВЛЕНИЯ ОБЪЕКТОМ

**ПРАКТИЧЕСКОЕ РУКОВОДСТВО
к курсовому проектированию
по дисциплине «Микропроцессорная техника»
для студентов специальности 1-36 04 02
«Промышленная электроника»
дневной и заочной форм обучения**

Гомель 2006

УДК 621.58(075.8)
ББК 32.973.26-04я73
П79

*Рекомендовано научно-методическим советом факультета
автоматизированных и информационных систем ГГТУ им. П. О. Сухого
(протокол № 1 от 14.09.2004 г.)*

Автор-составитель: Э. М. Виноградов

Рецензент: канд. техн. наук, доц. каф. «Автоматизированный электропривод»
ГГТУ им. П. О. Сухого Д. А. Хабибуллин

Проектирование микропроцессорной системы контроля и управления объектом :
П79 практ. рук. к курсовому проектированию по дисциплине «Микропроцессорная техника»
для студентов специальности 1-36 04 02 «Промышленная электроника» днев. и заоч.
форм обучения / авт.-сост. Э. М. Виноградов. – Гомель : ГГТУ им. П. О. Сухого, 2006. –
46 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное
место на HDD 16 Mb ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа:
<http://gstu.local/lib>. – Загл. с титул. экрана.

Практическое руководство содержит методические указания по выполнению курсового
проекта, примеры разработки схем и программ, требования к оформлению проекта.

Для студентов специальности 1-36 04 02 «Промышленная электроника» дневной и заочной
форм обучения.

УДК 621.58(075.8)
ББК 32.973.26-04я73

© Учреждение образования «Гомельский
государственный технический университет
имени П. О. Сухого», 2006

Введение

Развитие и распространение микропроцессоров, а также микроконтроллеров и микроЭВМ на их основе требует квалифицированных специалистов. Они должны, с одной стороны, хорошо знать элементную базу цифровых устройств (микропроцессоров, микросхем различной степени интеграции), с другой стороны, владеть методами алгоритмизации и программирования различных технических задач.

Для закрепления и углубления знаний, приобретенных студентами в курсе лекций «Микропроцессорная техника» программой специальности «Промышленная электроника» предусмотрен курсовой проект.

Целью курсового проекта является приобретение студентами навыков по проектированию и расчету микропроцессорных систем, осуществляющих сбор данных, их обработку, отображение и преобразование в унифицированные информационные сигналы.

Задача курсового проекта – разработка микропроцессорной системы управления технологическим объектом, выбор и расчет элементов системы, разработка ее программного обеспечения, составление структурных, функциональных и принципиальных схем.

1. Общие принципы проектирования микропроцессорных систем

Проектирование МПС состоит обычно из трех этапов [1]:

- 1) выбор элементов и составление структурной, функциональной и принципиальной схем;
- 2) распределение памяти и присвоение элементам, к которым в процессе выполнения программы необходимо обращаться, соответствующих кодов и адресов;
- 3) составление и отладка программы.

Основными исходными данными для проектирования являются: описание или схема алгоритма решаемой задачи, характеристики входных или выходных сигналов, требования к точности, быстродействию, потребляемой мощности, надежности. Часто в задании на проектирование содержатся также рекомендации по применению определенной элементной базы. Во многих случаях указанные данные закладываются в схему устройства вначале лишь в общем виде (посредством выбора подходящих элементов), а затем в процессе проектирования проверяется их выполнение и вносятся коррективы.

На первом этапе проектирования выбирается элементная база и, в первую очередь, микропроцессорный комплект (МПК). Выбор определяет, прежде всего, специфика реализуемых в МПС алгоритмов.

Если МПС предназначена для реализации алгоритмов управления и контроля, важнейшим фактором при выборе МПК является чаще всего простота (схемная и программная) сопряжения МПС с объектом управления. Поскольку в системах автоматического управления таковыми обычно

являются электромеханические и тепловые устройства, обладающие большой инерционностью, к быстродействию МП предъявляются сравнительно невысокие требования. И в таких системах во многих случаях можно использовать 8-разрядные микропроцессоры комплектов КР580 и КР1821 или однокристальные микроЭВМ серий К1816 и КР1830, которые позволяют строить достаточно простые и экономичные схемы.

В некоторых случаях решающим при выборе МПК является возможность использования имеющегося программного обеспечения, поскольку трудоемкость разработки новых программ сравнима и даже выше трудоемкости разработки аппаратурной части. С этой точки зрения достаточно эффективным является МПК КР580, для которого имеется обширная литература по программному обеспечению [2, 3].

На первом этапе проектирования выбираются элементы памяти ПЗУ и ОЗУ, имеющие требуемую емкость и быстродействие, а также интерфейсные модули, обеспечивающие связь МП с внешними устройствами (ВУ). Если система “компактна”, связь обеспечивается, как правило, с помощью параллельных адаптеров или простых регистров; если же ВУ разнесены на большие расстояния, связь организуется с помощью последовательных адаптеров. При этом организация линий связи упрощается, но производительность системы падает. Для частичной разгрузки МП в систему часто включают таймеры, которые выполняют функции времязадающих устройств.

Далее производится проверка электрического сопряжения элементов схемы, и при необходимости вводятся усилители, формирователи, преобразователи уровней и т.д. Затем составляются исходные варианты структурных, функциональных и принципиальных схем.

Выполнение второго этапа начинается с распределения адресного поля между ПЗУ, ОЗУ и портами ввода-вывода. Для обращения к памяти используют, как правило, метод страничной адресации, при котором младшие разряды шины адреса декодируются внутренними дешифраторами микросхем памяти, а старшие разряды используются для выбора страниц. Так, если микросхема ПЗУ или ОЗУ имеет информационную емкость $N = 2^k$ бит, то для адресации ее ячеек используются k линий ША – A_0, A_1, \dots, A_{k-1} . Для выбора страниц остается 16-к линий ША (в МПС на базе КР580). В простых МПС (типа микроконтроллеров), в которых емкость ПЗУ и ОЗУ составляет обычно несколько Кбайт, часто применяют двухстраничную адресацию. При этом выбор (селекция) ПЗУ и ОЗУ осуществляется с помощью одной из неиспользованных линий ША (например, при $A_{15} = 0$ выбирается ПЗУ, а при $A_{15} = 1$ выбирается ОЗУ).

Для адресации портов ввода-вывода используются разряды A_0, A_1, \dots, A_7 ША. Использование одних и тех же адресов и линий ША для памяти и портов возможно, поскольку обращение к этим устройствам осуществляется с помощью разных команд МП, при этом генерируются разные управляющие сигналы, выполняющие селекцию устройств.

Если количество портов ввода или вывода не превышает восьми, то каждому порту ставится в соответствие одна из линий ША, по которой передается сигнал, выбирающий этот порт. Иными словами, каждый порт кодируется унитарным кодом. Такой метод селекции портов часто называют линейным выбором. Декодирование унитарного кода выполняется инверторами. Если портов в МПС более восьми, то они кодируются двоичным позиционным кодом, и для их выбора используются дешифраторы.

На втором этапе выделяются области ПЗУ для хранения основной программы, подпрограмм, констант и т.п. В ОЗУ выделяются области для запоминания поступающих из ВУ данных и промежуточных результатов, а также организации стека. Начало (вершина) стека располагается в последней ячейке ОЗУ.

На третьем этапе составляется программа работы устройства. При этом если решаемая задача простая (программа содержит до ста команд), используется язык Ассемблера выбранного МП, который позволяет непосредственно получить двоичные (“объектные”) коды команд, записываемые далее в ПЗУ. Если же задача достаточно сложная, то используется тот или иной язык высокого уровня (Бейсик, Паскаль, Си). Выбор языка определяется тем, какие имеются у разработчика средства отладки программ и их трансляции в машинные коды выбранного МП. Система команд МП позволяет, как правило, выполнить ту или иную операцию алгоритма множеством способов. Поэтому составленную программу, даже если она правильно решает поставленную задачу, необходимо тщательно проанализировать и по возможности оптимизировать с целью уменьшения емкости используемой памяти и времени выполнения.

При программировании простых МПС (типа микроконтроллеров) обычно используют язык Ассемблера. При отладке программ на ЭВМ, имеющих трансляторы с Ассемблера, программы необходимо дополнить соответствующими псевдокомандами [3].

Широкие возможности использования типовых решений на первом и втором этапах проектирования и ограниченные возможности на третьем этапе приводит к тому, что 70..90 % трудоемкости проектирования специализированной МПС приходится на третий этап.

Эффективным средством разработки программ для специализированных МПС (типа микроконтроллеров) является метод декомпозиции (или “нисходящего” проектирования), при котором вся задача последовательно разделяется на меньшие функциональные модули, каждый из которых можно анализировать, разрабатывать и отлаживать отдельно от других [1]. При выполнении программ в МПС управление передается от одного функционального модуля к другому. Схема связи этих функциональных модулей, каждый из которых реализует некоторую завершающую процедуру, образует общую структурную схему (блок-схему) алгоритма (БСА) программы. Это разделение задачи на модули и submodule выполняется последовательно до такого уровня, когда разработка БСА модуля становится простым (даже

тривиальным) делом. Язык графических образов БСА можно использовать на любом уровне детализации описания модулей вплоть до того, что каждому блоку БСА будет соответствовать единственная команда МП.

2. Задание на проектирование

В курсовом проекте необходимо разработать микропроцессорную систему (МПС) контроля и управления некоторым технологическим объектом. Структура МПС приведена на рис. 2.1. МПС состоит из объекта контроля и управления, микроконтроллера (управляющей микроЭВМ), пульта управления и аппаратуры их взаимной связи. Микроконтроллер (МК) путем периодического опроса принимает информацию об объекте от аналоговых и цифровых датчиков. Выходные сигналы датчиков вследствие их различной физической природы могут потребовать промежуточного преобразования на аналого-цифровых преобразователях (АЦП) или на схемах формирователей сигналов (ФС), которые чаще всего выполняют функции гальванической развязки и формирования уровней двоичных сигналов стандарта ТТЛ.

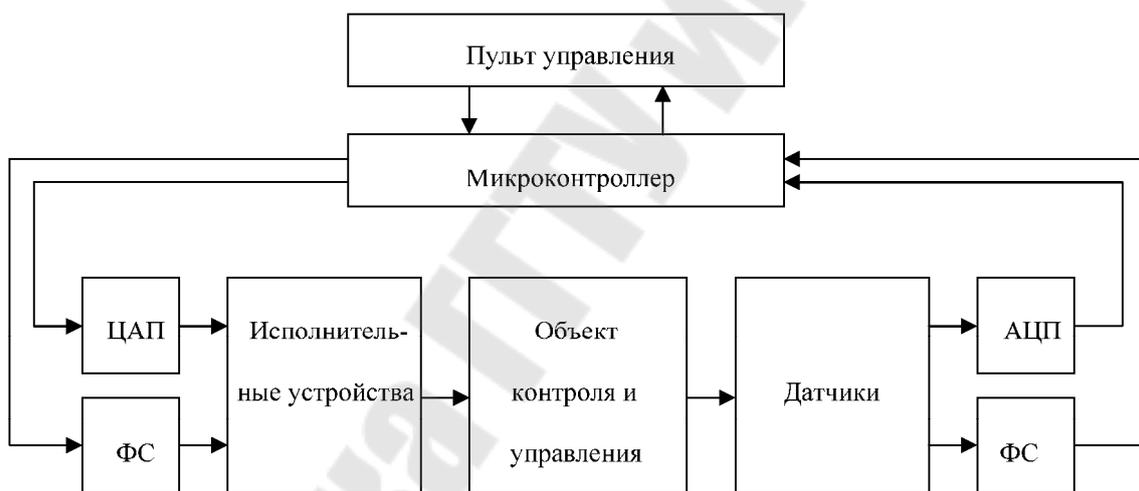


Рис. 2.1. Структура МПС

МК с требуемой периодичностью вырабатывает управляющие сигналы на своих выходных портах в соответствии с законом управления. Некоторая часть управляющих сигналов интерпретируется как совокупность простых двоичных сигналов управления, которые через схемы формирователей сигналов (усилители мощности, реле, оптроны и т.п.) поступают на исполнительные устройства. Другая часть управляющих сигналов представляет собой много-разрядные двоичные коды, которые через цифро-аналоговые преобразователи (ЦАП) воздействуют на исполнительные устройства аналогового типа. С помощью пульта управления (ПУ) оператор получает возможность управлять работой МПС: запускать ее и останавливать, выдавать значение некоторых сигналов, снимать с индикаторов информацию о состоянии объекта и т.п.

В курсовом проекте необходимо разработать структурную схему МПС, функциональные и принципиальные схемы отдельных модулей, разработать программы для МК, обеспечивающие выполнение алгоритмов контроля и управления в соответствии с заданием.

2.1. Алгоритм работы МПС

Алгоритм работы МПС представлен на рис. 2.2. После включения электропитания или после нажатия кнопки “Сброс” на ПУ выполняется начальная установка (инициализация) системы: настройка программируемых БИС (параллельных адаптеров, контроллера прерывания, таймера и т.п.), в выходные каналы засылаются начальные значения управляющих воздействий. Затем разрешаются прерывания, и МП переводится в режим останова.

По сигналу от таймера, работающего с частотой опроса f_{opr} , происходит выход из режима останова и переход к подпрограмме обработки прерывания ISR, блок-схема алгоритма которой приведена на рис. 2.3.



Рис. 2.2 Блок-схема алгоритма работы МПС

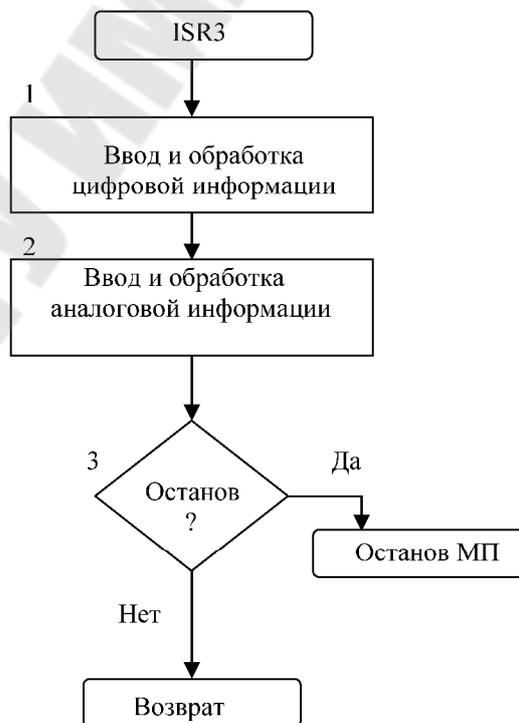


Рис. 2.3. БСА подпрограммы обработки прерывания

Блок 1 выполняет задачу приема и обработки двоичной информации: принимает сигналы от двоичных датчиков X_1, X_2, X_3 и X_4 , вычисляет значения логической (булевой) функции $f(X_1, X_2, X_3, X_4)$ в соответствии с выражением, определенном в задании на курсовой проект, и выдает это значение в качестве управляющего сигнала Y_1 по соответствующему выходному каналу на исполнительное устройство. При единичном значении логической функции $f(X_1, X_2, X_3, X_4)$ МК вырабатывает выходной сигнал ТТЛ - уровня $Y_1=1$ длительностью t_1 (рис. 2.4). БСА обработки цифровых сигналов приведена на рис. 2.5.

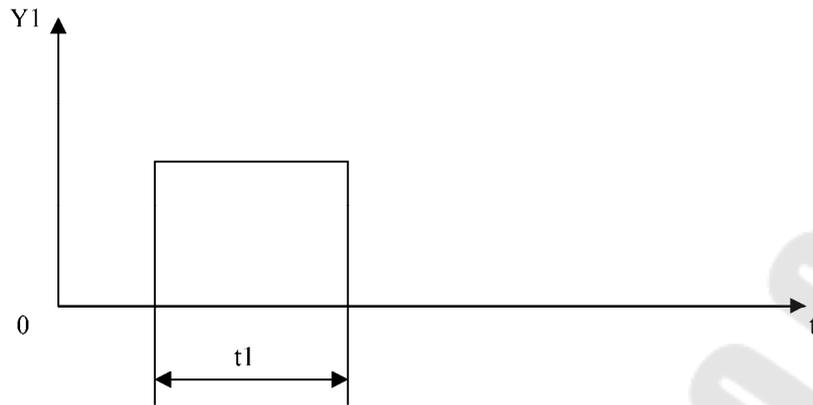


Рис. 2.4. Форма сигнала $Y1$

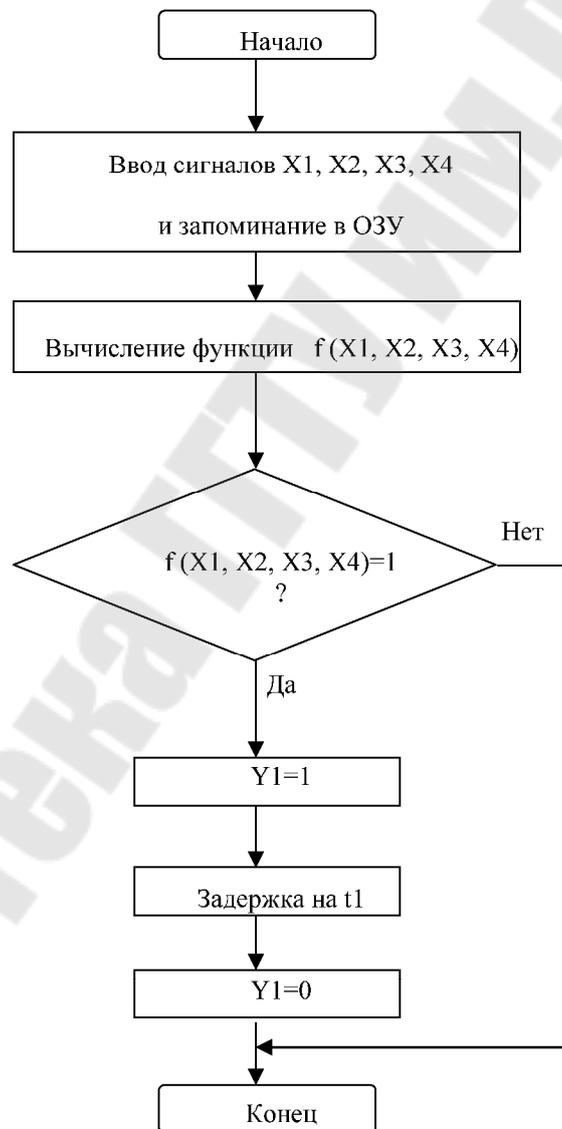


Рис. 2.5. БСА обработки цифровых сигналов

Блок 2 на рис. 2.3 обеспечивает прием сигналов V_1 , V_2 , V_3 и V_4 от аналоговых датчиков (однополярное напряжение от 0 до +10 В), которые преобразуются в цифровую форму с помощью АЦП. В МПС имеется только один АЦП, и сигналы с датчиков V_1 , V_2 , V_3 , V_4 подключаются к нему последовательно во времени с помощью аналогового коммутатора (мультиплексора). С выхода АЦП n -разрядные коды W_1 , W_2 , W_3 и W_4 , представляющие собой целые беззнаковые двоичные числа, поступают на обработку.

Обработка кодов W_1 и W_2 начинается с вычисления функции $g(W_1, W_2, K_1, K_2)$, где коэффициенты K_1 и K_2 – 16-разрядные целые беззнаковые двоичные числа, хранящиеся в ПЗУ. Вид функции $g()$ определяется из задания на курсовой проект. Полученное значение функции $g()$ сравнивается с константой Q , хранящейся в ПЗУ. В зависимости от результата сравнения МК вырабатывает двоичные управляющие сигналы TTL-уровня Y_2 или Y_3 длительностью t_2 или t_3 . БСА обработки сигналов W_1 и W_2 приведена на рис. 2.6.

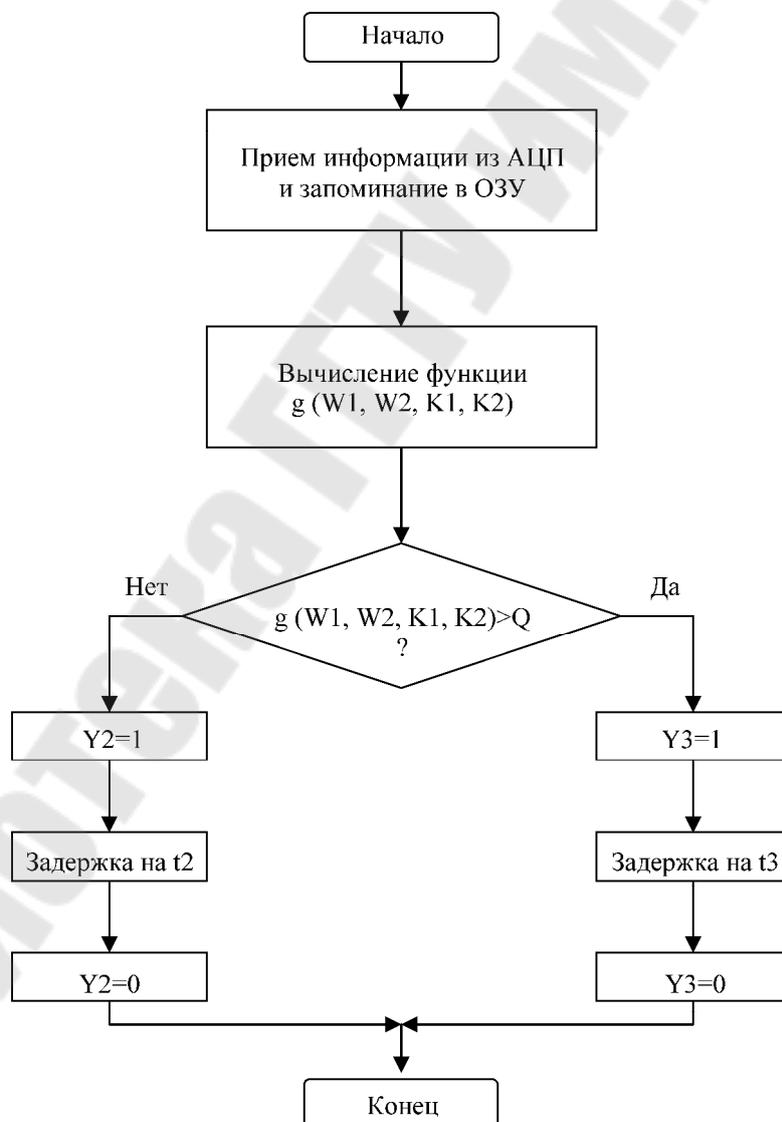


Рис. 2.6 БСА обработки аналоговых сигналов V_1 и V_2

Цифровой код $W3$, образованный аналоговым сигналом $V3$, используется для вычисления функции $e(W4, K3, K4, K5, K6)$, где $K3 \dots K6$ – коэффициенты, хранящиеся в ПЗУ. Конкретный вид функции $e()$ определяется заданием на курсовой проект. При вычислении функции предполагается, что все входящие значения представляют собой дробные 16-разрядные беззнаковые двоичные числа с фиксированной запятой. Полученное значение функции $e()$ подается на ЦАП, на выходе которого формируется аналоговый сигнал $Y4$, поступающий на исполнительные устройства.

Цифровой код $W4$, образованный аналоговым сигналом $V4$, используется для формирования управляющего сигнала $Y5$, представляющего собой последовательность импульсов с частотной или широтной модуляцией (вариант определяется заданием на курсовой проект).

При частотно-импульсной модуляции (ЧИМ) длительность импульса t_i постоянна, а меняется период следования импульсов T , т.е. их частота f в пределах от f_{\min} до f_{\max} прямо пропорционально величине цифрового кода $W4$, поступающего с АЦП (рис. 2.7). Значения t_i , f_{\min} и f_{\max} определяются в задании на курсовой проект.

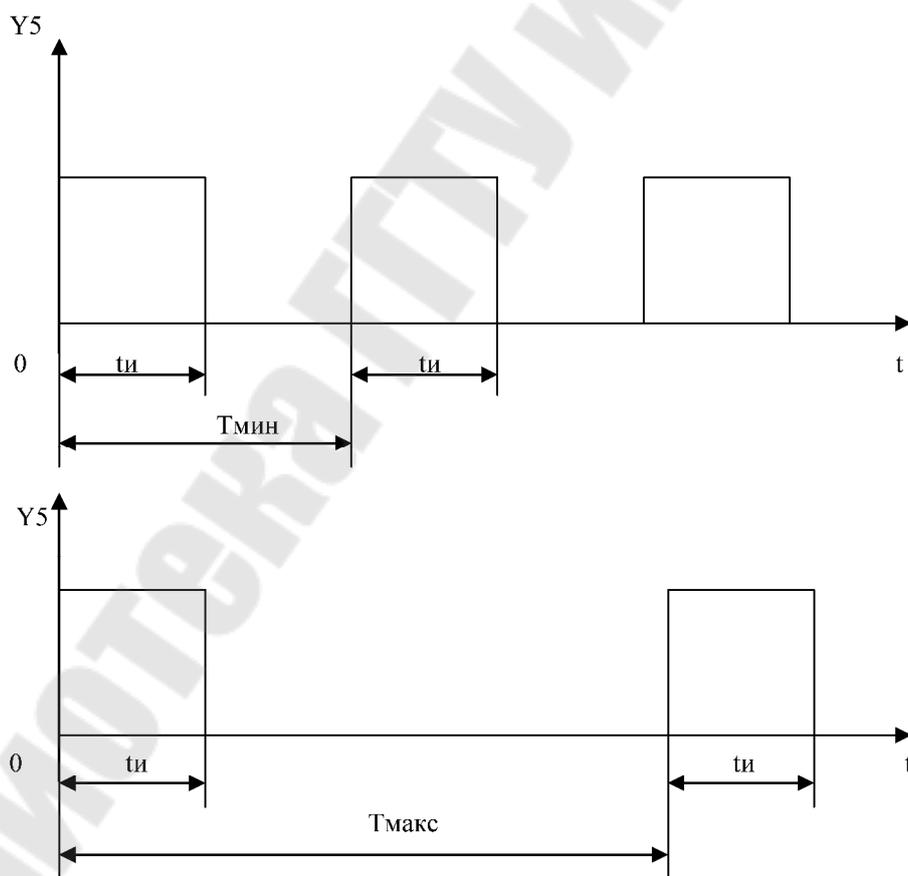


Рис. 2.7. Временные диаграммы сигналов при частотно-импульсной модуляции

При широтно-импульсной модуляции (ШИМ) постоянным является период T следования импульсов, т.е. частота f , а изменяется длительность

t_i в диапазоне от $t_{i.мин}$ до $t_{i.макс}$ прямо пропорционально величине цифрового кода W_4 , поступающего с АЦП (рис. 2.8). Значения T (или частоты f), $t_{i.мин}$, $t_{i.макс}$ определяются в задании на курсовой проект.

Блок 3 на рис. 2.3 обеспечивает циклический режим работы или останов МПС в соответствии с командой, поступающей от оператора с пульта управления.

В системе имеется также двоичный датчик аварийной ситуации, единственный сигнал с которого X_0 должен вызвать аварийный останов системы в любой момент выполнения рабочего цикла программы.

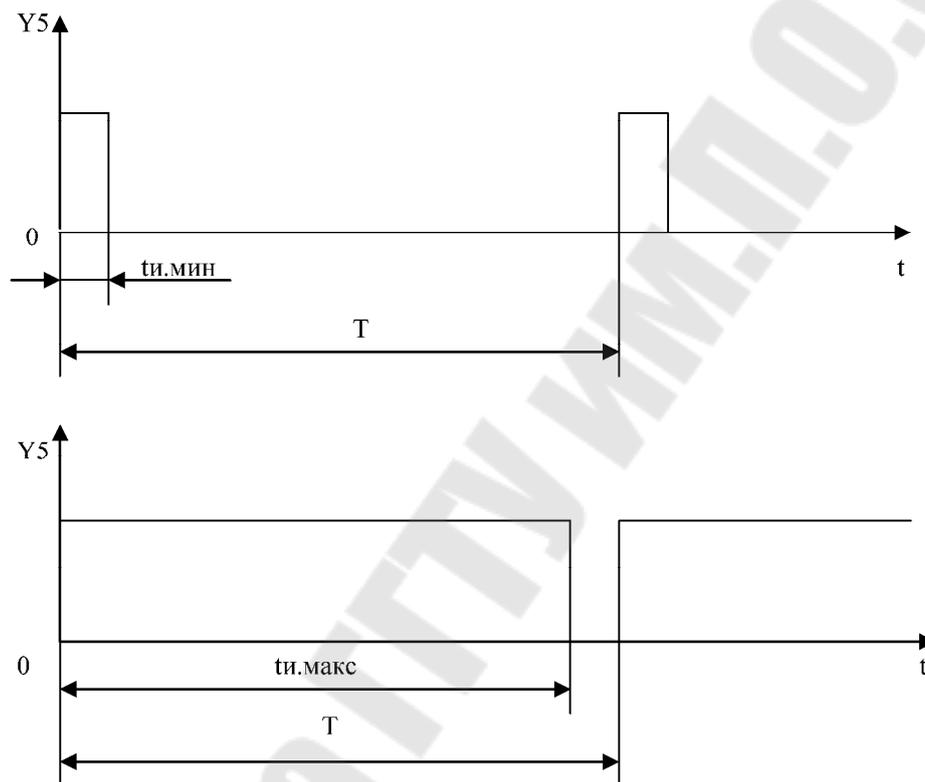


Рис. 2.8. Временные диаграммы сигналов при широтно-импульсной модуляции

2.2. Обработка запросов прерывания

В проектируемой МПС предусмотрены четыре запроса прерывания:

- запрос прерывания по сигналу от аварийного датчика INT0;
- запрос прерывания от пульта управления INT1;
- запрос прерывания от генератора ЧИМ (ШИМ) INT2;
- запрос прерывания от таймера опроса INT3.

Прерывание работы МПС от сигнала аварийного датчика имеет высший приоритет. При этом включается аварийная сигнализация на пульте управления (световая с частотой 1 Гц и звуковая с частотой 500 Гц), а также выводятся на индикацию сигналы датчиков X_1, \dots, X_4 и значения Y_1, Y_2, Y_3 . После этого МП необходимо остановить.

Прерывание от пульта управления (прерывание оператора) обеспечивает выполнение следующих действий:

1. Выдать на индикацию последние значения сигналов $X_1, \dots, X_4, Y_1, Y_2, Y_3$.

2. Выдать на индикацию значения цифровых кодов W_1, W_2, W_3, W_4 (по выбору оператора).

3. Обеспечить выход из прерывания по сигналу оператора (при нажатии кнопки “Пуск”).

2.3. Пульт управления

Пульт управления должен содержать следующие элементы:

1. Светодиоды для индикации значений $X_1, X_2, X_3, X_4, Y_1, Y_2, Y_3$.

2. Четырехпозиционный линейный дисплей на семисегментных светодиодах для отображения кодов W_1, W_2, W_3, W_4 .

3. Клавишный переключатель для выбора W_1, \dots, W_4 .

4. Светодиод индикации и динамик, на которые с таймера МПС подается меандр частотой 1 Гц и 500 Гц соответственно.

5. Кнопку “Сброс”, при нажатии на которую производится начальная установка устройств МПС.

6. Тумблер “Останов”, опрашиваемый в конце каждого цикла выполнения подпрограммы ввода и обработки сигналов.

7. Кнопку “Прерывание”, при нажатии на которую вырабатывается сигнал INT1 запроса прерывания от пульта управления.

8. Кнопку “Пуск”, при нажатии на которую происходит выход из подпрограммы прерывания по запросу оператора.

9. Светодиод “Останов”, индицирующий режим останова МП.

2.4. Структурная схема МПС

Структурная схема разрабатываемой МПС приведена на рис. 2.9. Модуль процессора формирует системную шину, состоящую из 16-разрядной шины адреса ША, 8-разрядной шины данных ШД и шины управления ШУ. К системной шине подключаются все устройства МПС. Модуль памяти включает ПЗУ для хранения команд программы и констант и ОЗУ для записи входных, промежуточных и выходных данных.

Цифровые (двоичные) сигналы X_1, X_2, X_3, X_4 поступают в систему через порт ввода 1. Аналоговые сигналы V_1, V_2, V_3, V_4 через коммутатор (аналоговый мультиплексор) поступают на АЦП. Цифровой код с выхода АЦП поступает на шину данных через порты ввода 2 и 3. Если АЦП выдает 8-разрядный код, то достаточно одного порта. Порт вывода 1 выдает адрес для выбора канала коммутатора, а также сигнал для запуска АЦП. На вход модуля прерывания подаются запросы прерывания: INT0 – от аварийного датчика X0, INT1 – от пульта управления, INT2 – от генератора импульсов с ШИМ (ЧИМ), INT3 – от таймера опроса. Таймер опроса вырабатывает импульсы с частотой $f_{опр}$, которые поступают в модуль прерывания.

Управляющие сигналы Y1, Y2, Y3 вырабатываются формирователями одиночных импульсов определенной длительности. Аналоговый сигнал Y4 вырабатывается с помощью ЦАП, на вход которого подается цифровой код через порты вывода 2 и 3. Периодический сигнал Y5 вырабатывается генератором импульсов с широтной (ШИМ) или частотной (ЧИМ) модуляцией.

Пульт оператора включает в себя: светодиоды индикации, которые управляются портом вывода 4; четырехпозиционный линейный дисплей на семисегментных светодиодных индикаторах, который управляется портами вывода 5, ..., 8; переключатели и кнопки, сигналы от которых поступают через порт ввода 4; устройство аварийной сигнализации, выдающее звуковые и световые сигналы определенной частоты.

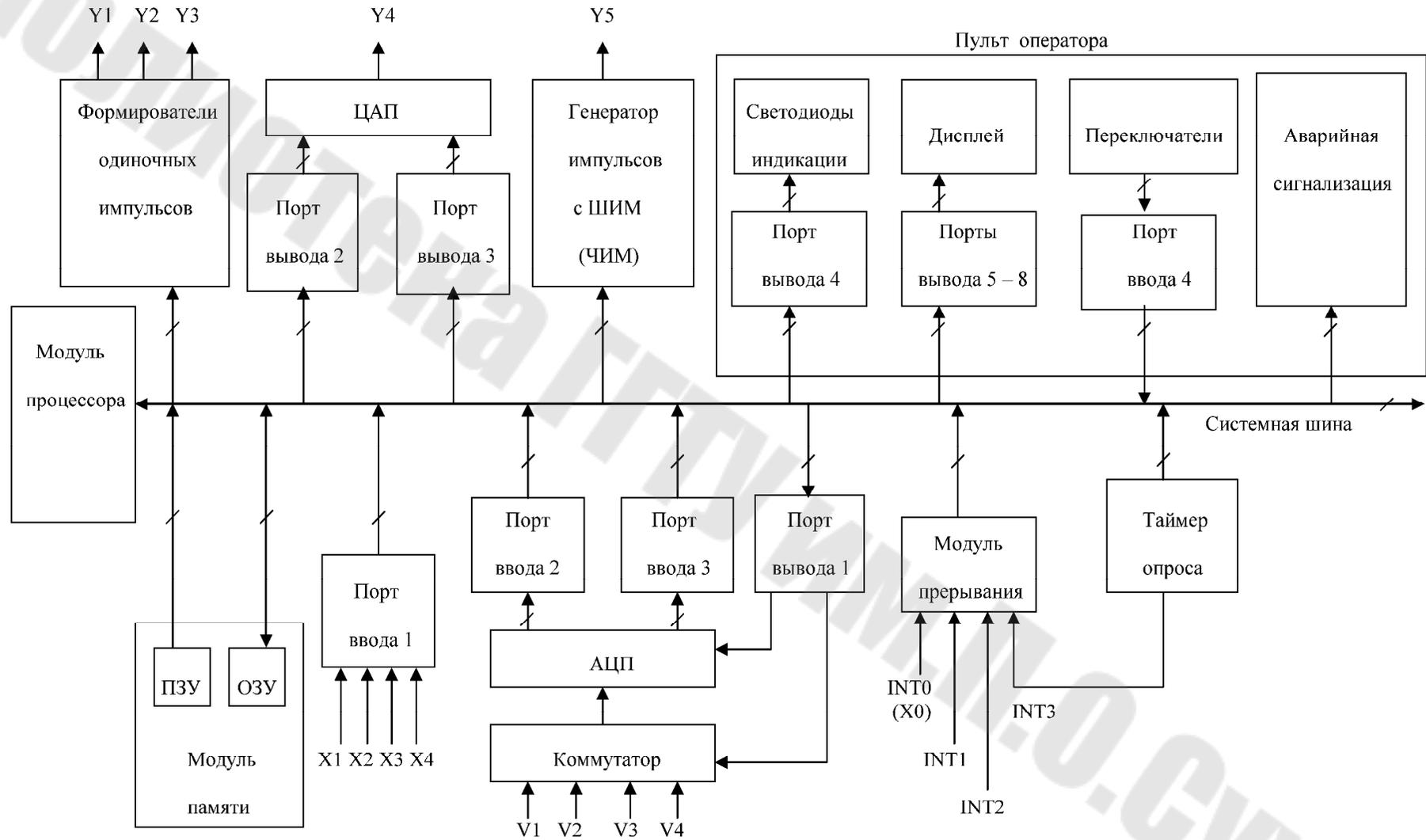


Рис. 2.9. Структурная схема МПС

3. Методические указания по выполнению курсового проекта

Проектирование микропроцессорной системы рекомендуется начинать с разработки аппаратурной части: модуля процессора, модуля памяти, модуля ввода-вывода и т.д. Затем следует разработать блок-схемы алгоритмов программ и написать их на языке Ассемблера.

3.1. Модуль процессора

Функциональная схема модуля процессора с использованием микропроцессорного комплекта КР580 приведена на рис. 3.1. Модуль состоит из микропроцессора КР580ВМ80А (микросхема DD1), генератора тактовых сигналов КР580ГФ24 (DD2), системного контроллера КР580ВК38 (DD3) и двух шинных формирователей КР580ВА86 или КР580ИР82 (DD4, DD5).

Кварцевый резонатор ZQ1 служит для задания тактовой частоты f_T работы МП, причем его частота должна быть в 9 раз больше тактовой, т.е.

$$f_{ZQ} = 9f_T.$$

Начальная установка МП при включении питания выполняется с помощью цепочки R1, C1, присоединенной к входу \overline{RESIN} микросхемы DD2. При R1=10 кОм и C1=1 мкФ обеспечивается надежная начальная установка при всех допустимых значениях f_T . С помощью кнопки SB1 начальная установка (сброс) может производиться в произвольный момент времени. На вход RDYIN подан сигнал лог. 1, который может быть получен от резистора сопротивлением порядка 1 кОм, присоединенным к источнику +5 В. В проектируемой МПС режим прямого доступа к памяти не предусмотрен, поэтому на входы HOLD микросхемы DD1 и \overline{BUSEN} микросхемы DD3 подан сигнал лог.0.

На рис. 3.2 приведена функциональная схема модуля процессора на базе МП КР1821ВМ85А. Вспомогательные микросхемы используются серии КР1533. Микросхемы DD2, DD3 выполняют роль формирователей сигналов шины адреса A0–A15, а DD4 служит для формирования двунаправленной шины данных D0–D7. Микросхемы DD5, DD6 преобразуют сигналы управления МП \overline{RD} , \overline{WR} , IO/ \overline{M} в системные сигналы управления \overline{MEMR} , \overline{MEMW} , \overline{IOR} , \overline{IOW} . Кварцевый резонатор ZQ1 задает частоту внутреннего тактового генератора. Частота кварца должна быть в 2 раза больше требуемой тактовой частоты f_{CLK} работы МП. Цепь R1, C1 выполняет начальную установку МП при включении питания. Обычно R1=10 кОм, C1=1 мкФ. С вывода CLK снимается сигнал тактовой частоты f_{CLK} работы МП, с вывода RESOUT снимается сигнал для начальной установки (сброса) различных устройств МПС. Выводы TRAP, RST7.5, RST6.5, RST5.5 и INTR – это входы запросов прерывания. Если какой-либо из этих входов не используется в системе, то на него необходимо подать низкий уровень (лог. 0).

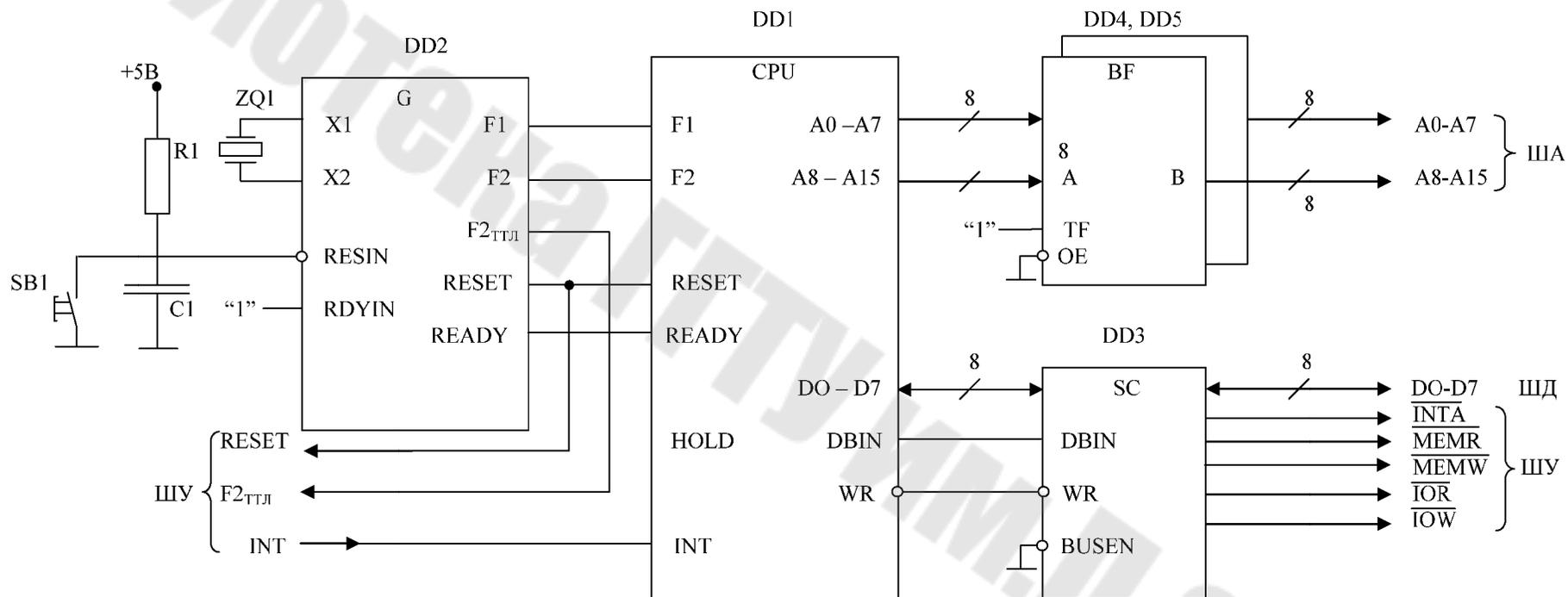


Рис 3.1. Функциональная схема модуля процессора с МП КР580ВМ80А

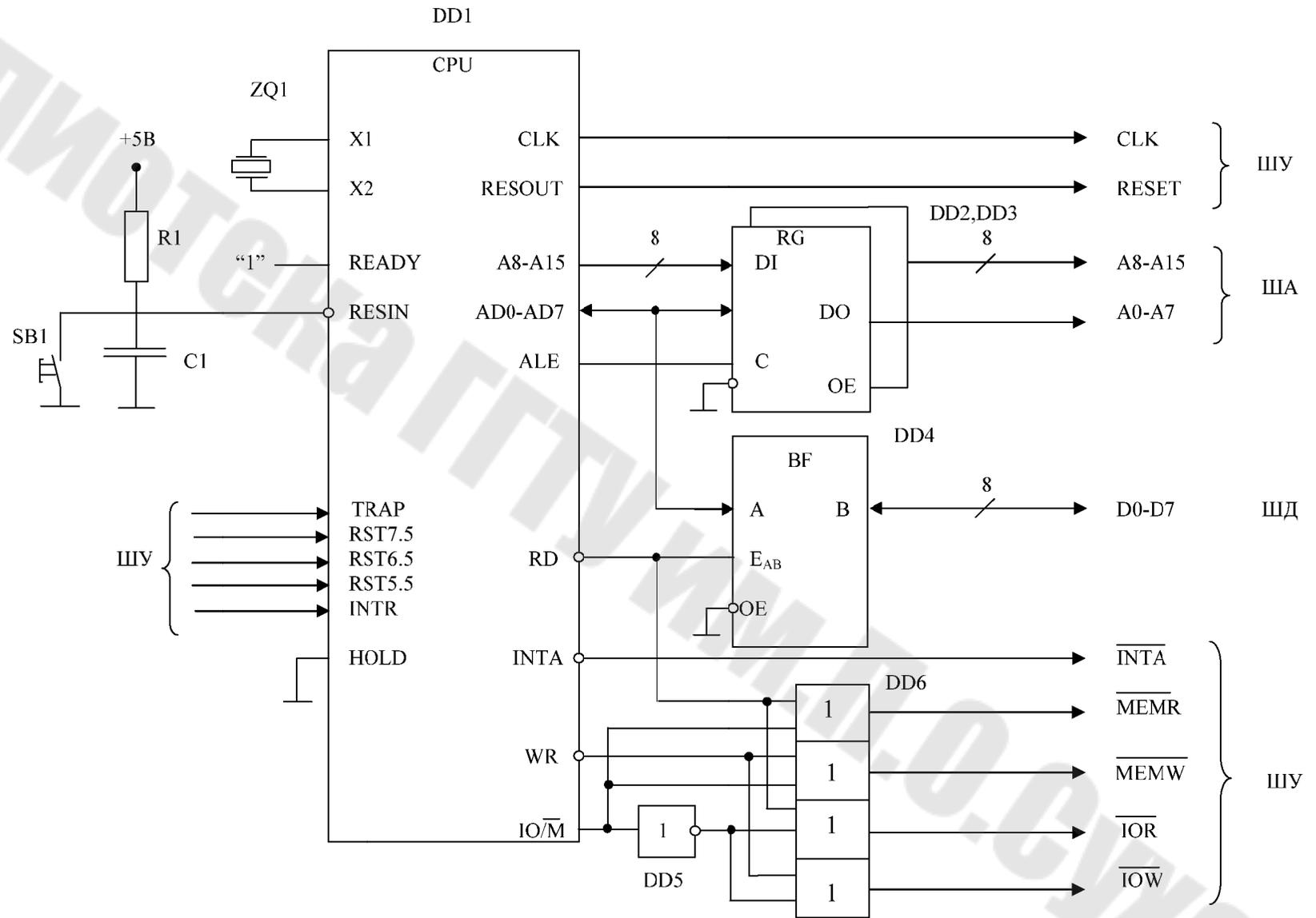


Рис. 3.2. функциональная схема модуля процессора с МП КР1821ВМ85А

3.2. Модуль памяти

Модуль выполняется на микросхемах постоянной (ПЗУ) и оперативной (ОЗУ) памяти. Используется принцип страничной организации памяти. При этом ячейки внутри страницы кодируются двоичным кодом и дешифруются внутренними цепями БИС ЗУ. Страницы выбираются адресным селектором, который часто реализуется на основе дешифраторов.

Сопряжение БИС ЗУ с шиной адреса МПС осуществляется обычно непосредственным подключением адресных входов микросхемы. Сопряжение с шиной данных МПС зависит от типа выходов данных микросхемы. Если выходы имеют 3 состояния, то их можно непосредственно подключать к ШД системы. В противном случае используются шинные формирователи, которые имеют выходные буферы с тремя состояниями. Формирователи также уменьшают емкостную нагрузку на БИС ЗУ и на ШД МПС.

На рис. 3.3 приведена функциональная схема модуля памяти, имеющего две страницы: одну ПЗУ (микросхема DD2) и одну ОЗУ (микросхема DD3) объемом по 2К 8-разрядных ячеек. Выбор ячейки внутри страницы производится с помощью адресных сигналов A0–A10. Выбор страниц выполняется дешифратором DD1, на входы которого подаются старшие разряды адреса A11–A15. Сигналы выбора страниц $\overline{SEL0}$ и $\overline{SEL1}$ с выхода дешифратора подаются на входы \overline{CS} микросхем памяти. Нулевая страница соответствует ПЗУ, а первая – ОЗУ. Они выбираются, когда в разрядах A15–A11 будут коды соответственно 00000 и 00001. Таким образом, ячейки ПЗУ занимают в адресном пространстве область 0000H–07FFH, а ячейки ОЗУ – область 0800H–0FFFH. При подаче на входы \overline{OE} активного низкого уровня сигнала управления \overline{MEMR} открываются выходные буферы выбранной микросхемы (ПЗУ или ОЗУ), и на ШД МПС появляются данные из ячейки памяти. При подаче на вход разрешения записи \overline{WE} микросхемы ОЗУ активного низкого уровня сигнала управления \overline{MEMW} происходит запись данных с ШД МПС в выбранную ячейку.

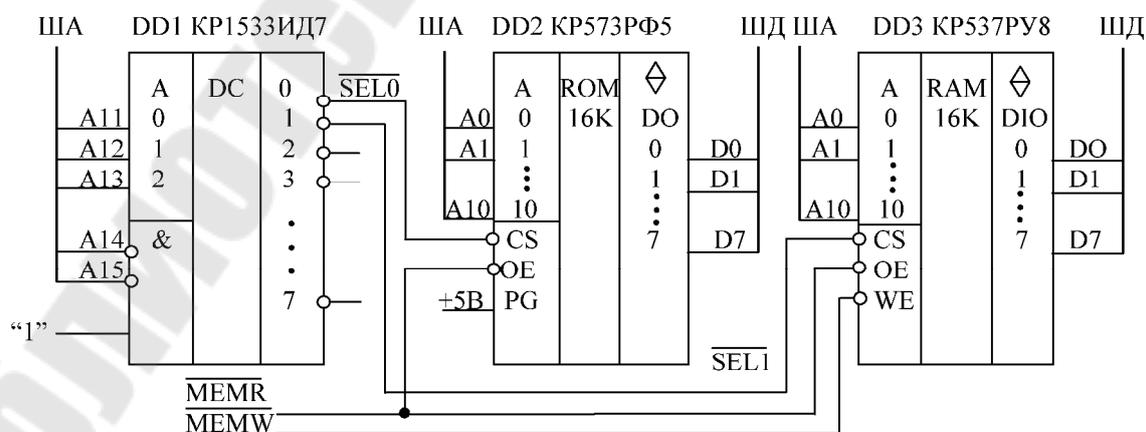


Рис. 3.3. Функциональная схема модуля памяти

При двухстраничной организации модуля памяти можно в качестве селектора страниц использовать один из старших разрядов адреса. Это позволяет обойтись без дешифратора. На рис. 3.4 приведена схема такого двухстраничного модуля памяти. Селектором страниц является разряд A11 шины адреса. При A11=0 выбрана микросхема DD1 (ПЗУ). На входе \overline{CS} микросхемы DD2 (ОЗУ) будет сигнал лог. 1 с выхода инвертора DD3, поэтому DD2 не выбрана. При значении A11=1 наоборот, выбрано ОЗУ, а ПЗУ – нет. В данном модуле ПЗУ имеет адреса 0000H–07FFH, а ОЗУ – адреса 0800H–0FFFH.

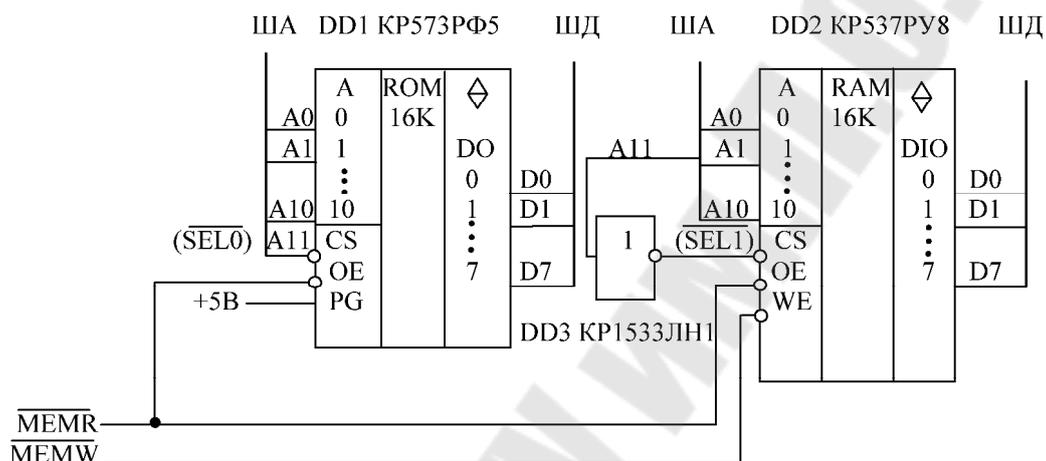


Рис. 3.4. Двухстраничный модуль памяти

На рис. 3.5 приведена схема ПЗУ на микросхеме KP556PT5, которая имеет выходы данных с открытым коллектором (ОК). Для подключения БИС ЗУ к ШД МПС используется шинный формирователь KP580BA86, имеющий выходы с тремя состояниями (ЗС). При чтении из ячеек ПЗУ модуль процессора вырабатывает сигнал \overline{MEMR} , который открывает выходные буферы шинного формирователя, и данные из ячейки памяти поступают на ШД системы.

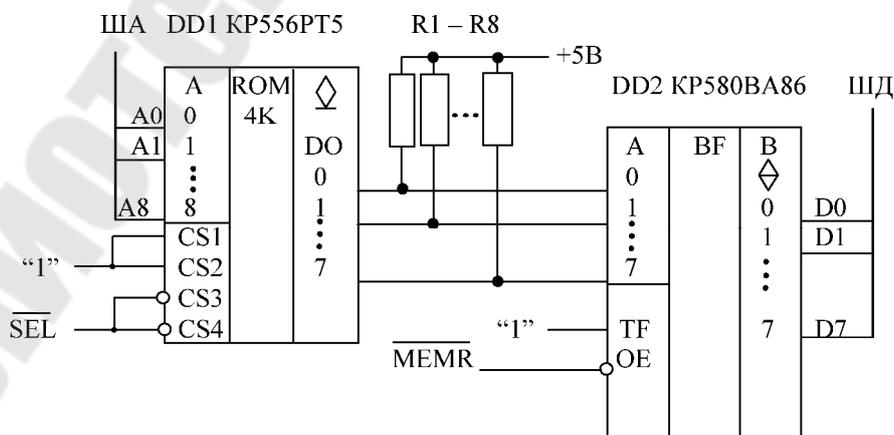


Рис. 3.5. Подключение БИС ПЗУ с буферированием данных

Если микросхема ПЗУ имеет выходы данных с тремя состояниями, но не имеет вывода для управления ими (отсутствует вывод \overline{OE}), то подключить такую БИС к ШД МПС можно двумя способами:

1) с использованием выходного буфера данных (аналогично схеме на рис. 3.5);

2) стробированием сигнала выбора микросхемы \overline{CS} .

На рис. 3.6 приведен пример подключения БИС КР556РТ6 с применением стробирования сигнала \overline{CS} .

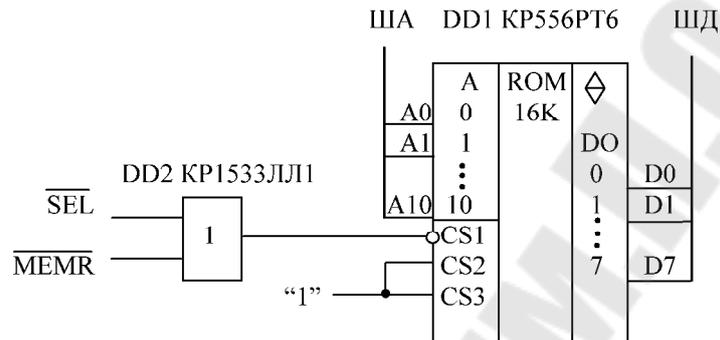


Рис. 3.6. Подключение БИС ПЗУ со стробированием сигнала \overline{CS}

Микросхемы ОЗУ выпускаются двух видов: с отдельными и с общими линиями ввода-вывода данных. Все виды микросхем имеют на выходах данных буферы с тремя состояниями. БИС ОЗУ с отдельными линиями ввода и вывода имеют одноразрядные ячейки памяти, т.е. организацию $1K \times 1$, $2K \times 1$, $4K \times 1$ и т.п. Для построения модуля ОЗУ требуется как минимум 8 микросхем, причем каждая из них будет хранить строго определенный разряд байта данных, т.е. D0, D1, .. , D7. Особенностью этих БИС является то, что они не имеют вывода управления выходными буферами (отсутствует вывод \overline{OE}). Сопряжение таких БИС с ШД МПС возможно двумя способами:

1) с буферированием данных;

2) стробированием сигнала выбора \overline{CS} .

На рис. 3.7 приведена схема модуля ОЗУ емкостью $2K \times 8$ бит с буферированием данных. В качестве буфера используется шинный формирователь КР580ВА86, выходы которого управляются сигналом \overline{MEMR} . На рис. 3.8 приведена схема ОЗУ с использованием стробирования сигнала выбора микросхем \overline{CS} . Схема стробирования выполнена на логических элементах DD9–DD11.

3.3. Модуль прерывания

В микропроцессорном комплекте КР580 имеется микросхема КР580ВН59, которая называется программируемым контроллером прерывания (ПКП). Она позволяет осуществлять прерывания основной программы микропроцессора по одному из восьми запросов прерывания [4]. На рис. 3.9 приведена функциональная схема модуля прерывания с ПКП, реализующего поставленные в задании на проектирование МПС условия: 4 запроса прерывания INT0 – INT3, за-

прос INT0 имеет высший приоритет. Запрос INT0 поступает от аварийного датчика, INT1 – от пульта управления, INT2 – от генератора импульсов ЧИМ (или ШИМ), запрос INT3 – от таймера опроса.

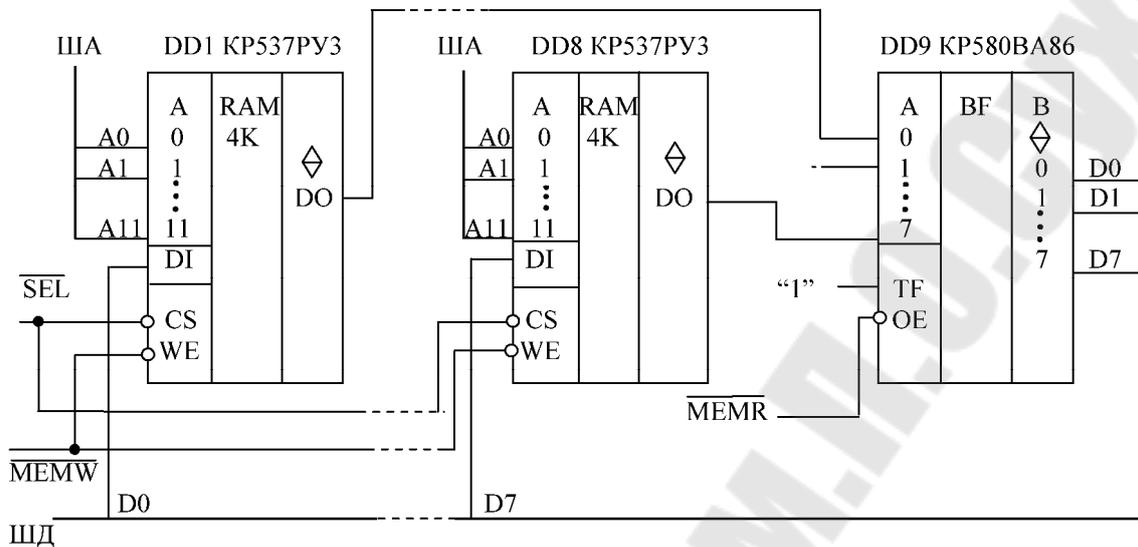


Рис. 3.7. ОЗУ с буферированием данных

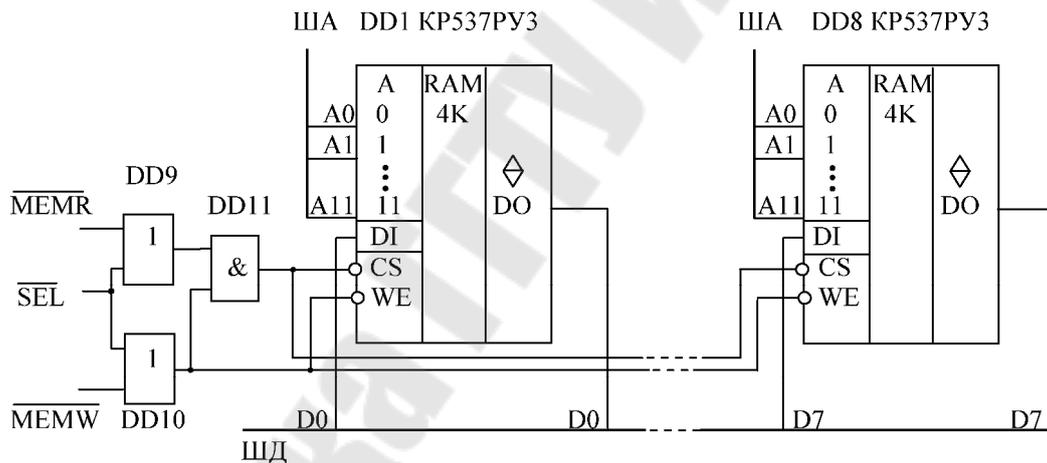


Рис. 3.8. ОЗУ со стробированием сигнала \overline{CS}

Все запросы представляют собой положительные перепады напряжения уровней ТТЛ, т.е. переходы от низкого к высокому уровню. Запросы прерывания INT_i подаются на тактовые входы D-триггеров, которые находятся в исходном нулевом состоянии. При поступлении запроса соответствующий триггер переходит в единичное состояние, с его прямого выхода сигнала высокого уровня поступает на вход IRQ_i ПКП. Когда МП переходит к циклу обслуживания прерывания, по сигналу \overline{INTA} от модуля процессора, поступающего на асинхронный вход \overline{R} , триггер сбрасывается в нулевое состояние, и запрос со входа IRQ_i снимается. Этим предотвращается закливание программы обработки прерывания. Цепочка R1, C1 служит для сброса триг-

геров в нулевое состояние при включении питания. Сигнал $\overline{\text{SELPIC}}$ поступает с адресного селектора и служит для выбора ПКП.

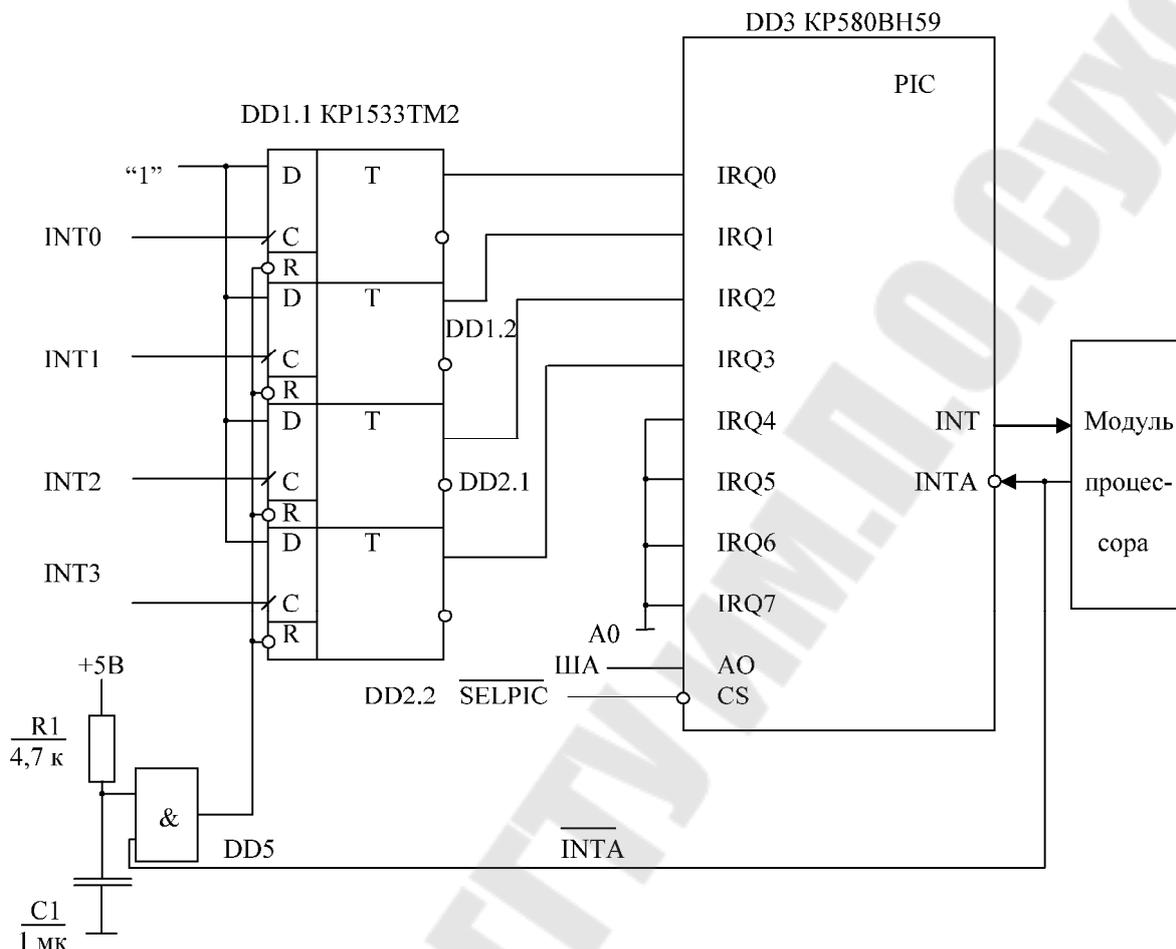


Рис. 3.9. Функциональная схема модуля прерывания с ПКП KP580BH59

Если в модуле процессора используется МП KP1821BM85A, то необходимость в ПКП отпадает, так как сам МП имеет 5 входов запросов прерывания. В этом случае сигнал с выхода триггера DD1.1 следует подать на вход TRAP, с DD1.2 – на RST7.5, с DD2.1 – на RST6.5, и с выхода DD2.2 – на вход RST5.5. На вывод INTR необходимо подать низкий уровень.

3.4. Таймер опроса

Согласно заданию на курсовой проект микроконтроллер должен периодически с частотой опроса $f_{\text{опр}}$ (или периодом $T_{\text{опр}}$) производить ввод сигналов от датчиков, обрабатывать их и выводить на исполнительные устройства. Для этой цели МК должен иметь времязадающее устройство, которое выдает импульсы с частотой $f_{\text{опр}}$. Назовем это устройство таймером опроса. Реализовать его можно с помощью БИС программируемого таймера (ПТ) KP580BI53, который имеет 3 независимых 16-разрядных счетчика, способных работать в шести режимах [4]. Для рассматриваемого случая подходит режим 2 – программируемый делитель частоты. Коэффициент де-

ления определяется числом N , загруженным в счетчик. В качестве входных удобно использовать импульсы тактовой частоты МП, которые выдаются на выводы $F2_{\text{ТТЛ}}$ или CLK шины управления в модуле процессора. Определим необходимое число N при заданных $F2_{\text{ТТЛ}} = 2 \text{ МГц}$ и $f_{\text{опр}} = 2 \text{ Гц}$:

$$N = \frac{F2_{\text{ТТЛ}}}{f_{\text{опр}}} = \frac{2 \cdot 10^6}{2} = 10^6.$$

Максимальное число, которое можно загрузить в 16-разрядный счетчик, равно 65535, что значительно меньше необходимого. Поэтому будем использовать два счетчика, включенных последовательно, каждый из которых будет делить частоту в 1000 раз. Схема включения счетчиков в таймере опроса приведена на рис. 3.10. Используются счетчики CT0 и CT1 программируемого таймера. Сигнал $\overline{\text{SELPT}}_1$ поступает с адресного селектора и служит для выбора микросхемы.

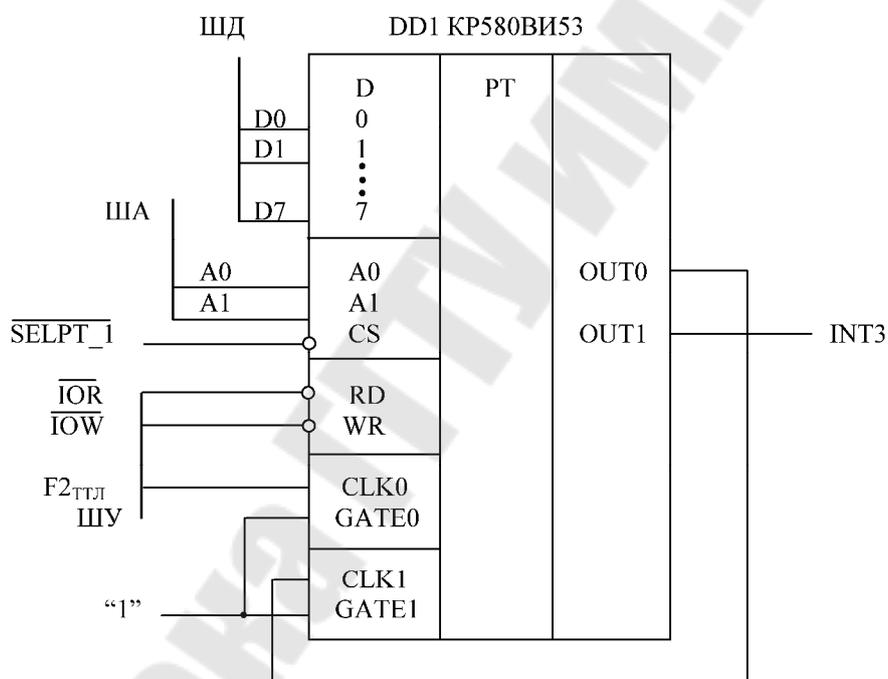


Рис. 3.10. Схема таймера опроса

3.5. Схемы ввода цифровых и аналоговых сигналов

На рис. 3.11 приведена функциональная схема ввода сигналов от датчиков МПС. Порты реализованы на БИС KP580BB55A – программируемом параллельном адаптере (ППА). Порты PA и PB работают на ввод, порт PC – на вывод. Микросхема DA1 – 10-разрядный АЦП K1113PB1 . Младшие разряды выходного кода АЦП поданы на вход порта PB , а два старших разряда D8 , D9 поданы на входы PA6 , PA7 порта PA . Сигнал готовности данных $\overline{\text{DR}}$ АЦП поступает на вход PA5 порта PA . Цифровые входные сигналы X1-X4 подаются на входы PA0-PA3 порта PA . Порт PC адаптера

Таблица 1

Адреса портов (регистров БИС)

Двоичный код								HEX-код	Символическое имя	Примечание
A7	A6	A5	A4	A3	A2	A1	A0			
1	0	0	0	0	0	0	0	80	СТ0_1	1-й программируемый таймер
1	0	0	0	0	0	0	1	81	СТ1_1	
1	0	0	0	0	0	1	0	82	СТ2_1	
1	0	0	0	0	0	1	1	83	РУС53_1	
1	0	0	0	0	1	0	0	84	СТ0_2	2-й программируемый таймер
1	0	0	0	0	1	0	1	85	СТ1_2	
1	0	0	0	0	1	1	0	86	СТ2_2	
1	0	0	0	0	1	1	1	87	РУС53_2	
1	0	0	0	1	0	0	0	88	РУС59	ПКП
1	0	0	0	1	0	0	1	89	РУС59А	
1	0	0	0	1	1	0	0	8С	РА	ППА
1	0	0	0	1	1	0	1	8D	РВ	
1	0	0	0	1	1	1	0	8E	РС	
1	0	0	0	1	1	1	1	8F	РУС55	
1	0	0	1	0	0	0	0	90	IPORT	Порт ввода

3.6. Пульт управления

На рис. 3.14 приведена схема подключения светодиодов для индикации значений сигналов X1, ... , X4, Y1, Y2, Y3 на пульте управления. Микросхемы DD2 – это мощные инверторы с открытым коллектором, например, К155ЛН3. В качестве порта вывода можно использовать регистры КР580ИР82 или серии КР1533, например, КР1533ИР33.

На рис. 3.15 приведена схема включения семисегментных светодиодных индикаторов НГ1-НГ4, входящих в четырехпозиционный линейный дисплей на пульте управления. Дисплей предназначен для отображения 16-разрядных кодов W1-W4, полученных с выхода АЦП, и соответствующих входным аналоговым сигналам V1-V4.

Для управления портами индикации необходим селектор адреса (дешифратор) портов, его схема подключения приведена на рис. 3.16.

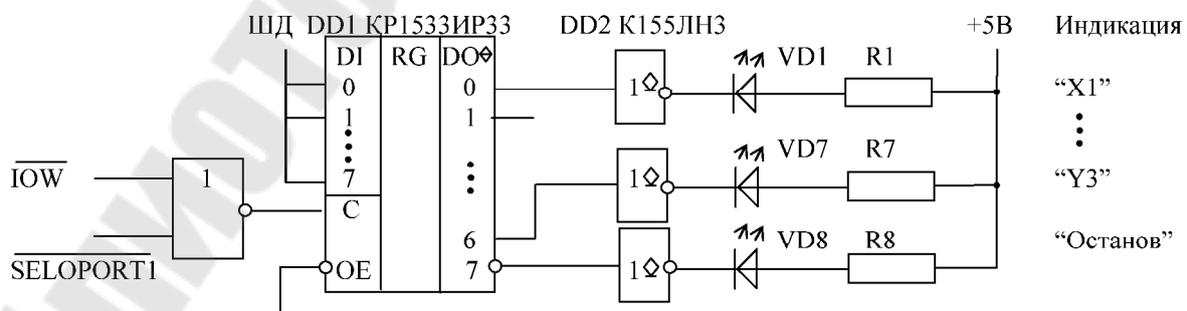


Рис. 3.14. Схема подключения светодиодов индикации

В табл. 2 приведены коды адресов портов и их символические имена.

Таблица 2

Адреса портов

Двоичный код								HEX-код	Символическое имя
A7	A6	A5	A4	A3	A2	A1	A0		
1	0	1	0	0	0	0	0	A0	OPORT 1
1	0	1	0	0	1	0	0	A4	OPORT 2
1	0	1	0	1	0	0	0	A8	OPORT 3
1	0	1	0	1	1	0	0	AC	OPORT 4
1	0	1	1	0	0	0	0	B0	OPORT 5
1	0	1	1	0	1	0	0	B4	OPORT 6
1	0	1	1	1	0	0	0	B8	OPORT 7
1	0	1	1	1	1	0	0	BC	OPORT 8

Примечание. Хотя разряды A1 и A0 могут иметь произвольные значения (они не декодируются), по умолчанию их значения равны нулю. Управление работой МПС с пульта управления выполняется с помощью кнопочных переключателей и кнопок. Их подключение к порту ввода приведено на рис. 3.17. Кнопочные переключатели SA1-SA4 предназначены для выбора кода W1-W4 для вывода на дисплей. Тумблер SA5 служит для останова работы МП. С помощью кнопки SB2 “Пуск” выполняется возврат к основной программе работы МПС. С помощью кнопки SB3 “Прерывание” выдается сигнал запроса прерывания INT1 (прерывание оператора). Триггер служит для устранения влияния дребезга контактов кнопки.

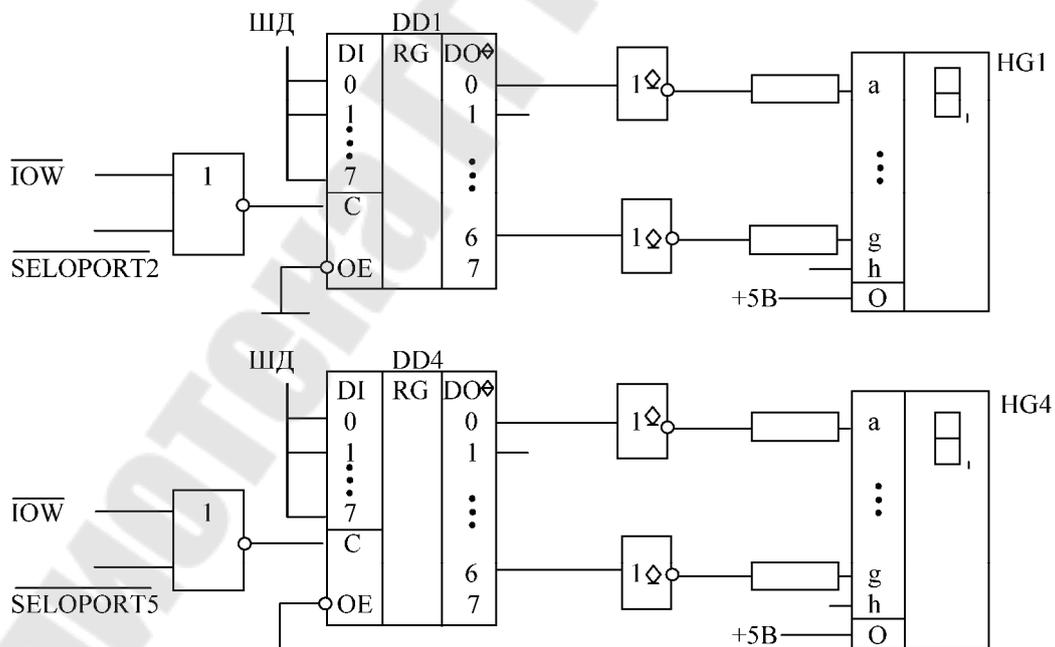


Рис 3.15. Схема подключения семисегментных индикаторов

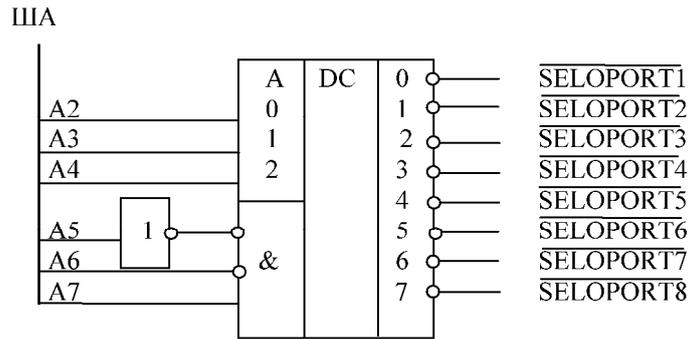


Рис. 3.16. Дешифратор адресов портов для ПУ

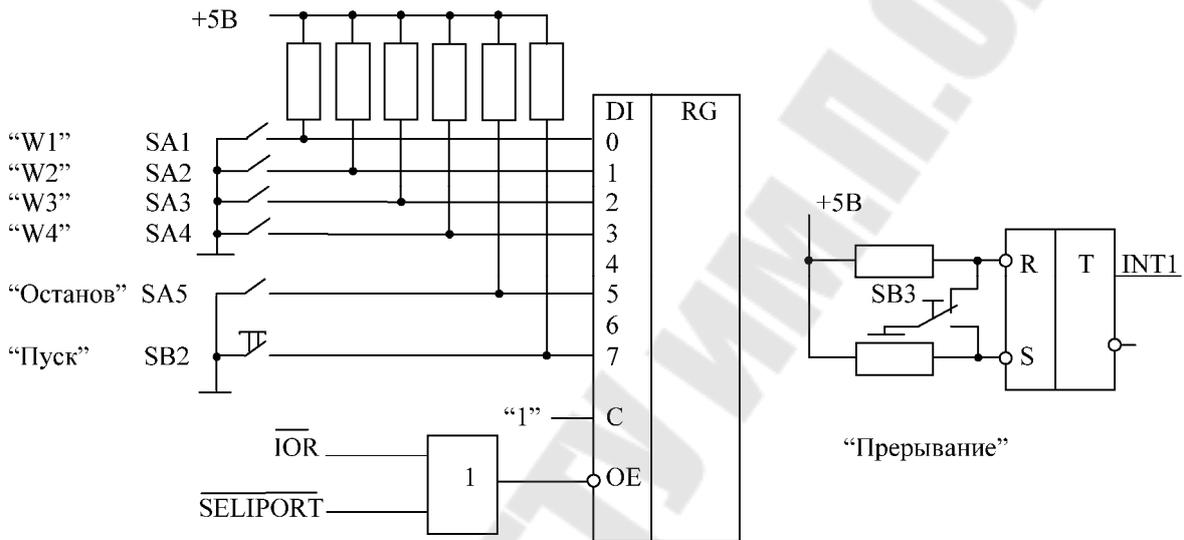


Рис. 3.17. Схемы подключения переключателей и кнопок на ПУ

3.7. Примеры разработки программ работы МПС

Алгоритм работы МПС был приведен на рис. 2.2. Он заключается в инициализации (начальной установке) всех устройств системы, разрешения прерываний и затем останова МП. Дальнейшая работа МПС выполняется как реакция на запросы прерывания от различных устройств (таймера опроса, пульта управления, аварийного датчика, генератора импульсов с ЧИМ или ШИМ), т.е. с помощью подпрограмм обработки запросов прерываний.

При составлении программы на языке Ассемблера необходимо знать адреса, занимаемые ПЗУ и ОЗУ, и адреса портов. Желательно использовать символические адреса констант, портов и ячеек памяти, с которыми оперирует программа.

Будем считать, что модуль памяти проектируемой МПС выполнен по схеме рис. 3.3. ПЗУ имеет адреса 0000H-07FFH, а ОЗУ – адреса 0800H-0FFFH. Адреса и символические имена портов и регистров БИС даны в таблицах 1 и 2.

Для инициализации программируемого таймера, параллельного адаптера, контроллера прерываний необходимо определить управляющие слова для заданных режимов работы БИС [2,4].

Для счетчиков СТ0, СТ1 1-го таймера ПТ_1: режим 2 (делитель частоты), счет двоичный, загрузить 2 байта. Числа N0 и N1 равны 1000=03E8H. Управляющее слово для СТ0: 00110100B=34H, для СТ1: 01110100B=74H.

Для ПКП: адреса подпрограмм обслуживания прерываний равны 0020H, 0024H, 0028H, 002CH. Управляющее слово 1: 00110110B=36H, управляющее слово 2: 00000000B=00H.

Для ППА: режим 0, порты РА и РВ работают на ввод, порт РС – на вывод. Управляющее слово: 10010010B=92H.

Основную программу назовем CONTROL. Текст программы имеет следующий вид.

```

; CONTROL – основная программа работы МПС
СТ0_1      EQU  80H
.....
IPORT      EQU  90H
OPORT1     EQU  A0H
  ⋮
STACK      EQU  0FFFH      ; Адрес вершины стека
           ORG  0000H
CONTROL:   LXI  SP, STACK  ; Определить стек
           JMP  MAIN
           ORG  0020H      ; Вектор прерывания по INT0
           JMP  ISR0

           ORG  0024H      ; Вектор прерывания по INT1
           JMP  ISR1
           ORG  0028H      ; Вектор прерывания по INT2
           JMP  ISR2
           ORG  002CH      ; Вектор прерывания по INT3
           JMP  ISR3
           ORG  0050H
MAIN:      MVI  A, 92H
           OUT  PUC55      ; Инициализация ППА
           XRA  A
           OUT  PC        ; Вывод нулей в порт РС
           OUT  OPORT1    ; Гашение светодиодов
           OUT  OPORT2    ; Гашение дисплея
           OUT  OPORT3
           OUT  OPORT4
           OUT  OPORT5
           MVI  A, 34H
           OUT  PUC53_1    ; Режим СТ0_1
           MVI  A, 74H
           OUT  PUC53_1    ; Режим СТ1_1

```

```

MVI A, 0E8H
OUT CT0_1
MVI A, 03H
OUT CT0_1      ; Число N0 в CT0_1
MVI A, 0E8H
OUT CT1_1
MVI A, 03H
OUT CT1_1      ; Число N1 в CT1_1
MVI A, 36H
OUT PUC59
MVI A, 00H
OUT PUC59A     ; Инициализация ПКП
EI              ; Разрешить прерывания
HLT            ; Остановить МП

```

Если в МПС используется МП КР1821ВМ85А и отсутствует ПКП, то в программу CONTROL нужно внести изменения:

1. Необходимо изменить векторы прерываний (начальные адреса подпрограмм обработки прерываний).

2. Необходимо кроме общего разрешения прерываний разрешить прерывания по входам RST7.5, RST6.5, RST5.5 индивидуально с помощью команды SIM.

Команда SIM загружает в регистр прерываний МП содержимое аккумулятора, которое должно иметь вид:

7	6	5	4	3	2	1	0
0	0	0	R7.5	MSE	M7.5	M6.5	M5.5

Здесь M7.5, M6.5, M5.5 – маски прерываний (1 – маска установлена); MSE – разрешение установки маски (1 – разрешено); R7.5 – разрешение сброса триггера на входе RST7.5 (1 – сброс).

Например, необходимо разрешить прерывания RST7.5 и RST6.5, но запретить RST5.5, выполнить сброс триггера RST7.5. Содержимое аккумулятора должно быть: 00011001B=19H. Программа для МП:

```

MVI A, 19H
SIM

```

Следует отметить, что сброс триггера на входе RST7.5 необходимо выполнять каждый раз перед выходом из подпрограммы обслуживания прерывания по этому входу, иначе МП не будет реагировать на запросы по RST7.5.

Таким образом, для МП КР1821ВМ85А программа CONTROL будет иметь вид:

```

ORG 0000H
CONTROL: LXI SP, STACK
JMP MAIN

```

```

    ORG 0024H      ; Вектор прерывания по входу TRAP
                  ; (запрос INT0)
    JMP ISR0
    ORG 003CH      ; Вектор прерывания по входу RST7.5
                  ; (запрос INT1)
    JMP ISR1
    ORG 0034H      ; Вектор прерывания по входу RST6.5
                  ; (запрос INT2)
    JMP ISR2
    ORG 002CH      ; Вектор прерывания по входу RST5.5
                  ; (запрос INT3)
    JMP ISR3
    ORG 0050H
MAIN:  MVI A, 92H
        ⋮
        MVI A, 18H

    SIM          ; Разрешить прерывания RST7.5,
                  ; RST6.5, RST5.5
    EI           ; Общее разрешение прерываний
    HLT

```

Блок-схема алгоритма (БСА) подпрограммы обработки прерывания ISR3 по запросу INT3 приведена на рис. 2.3. Вначале производится ввод и обработка цифровой информации (входные сигналы X1-X4), затем ввод и обработка аналоговой информации (входные сигналы V1-V4), в конце проверяется состояние тумблера “Останов” на пульте управления.

БСА обработки цифровой информации приведена на рис. 2.5. Рассмотрим программную его реализацию на примере вычисления логической функции двух переменных

$$f(X1, X2) = \overline{X1} \wedge (\overline{X1} \vee X2),$$

где \vee – знак операции логического сложения (функция ИЛИ);
 \wedge – знак операции логического умножения (функция И);
 $\overline{\quad}$ – знак операции логического отрицания (функция НЕ).

Входные сигналы X1, X2 поступают на вход порта PA ППА (см. рис. 3.11). Для временного хранения переменных X_i в программе выделим в ОЗУ массив ячеек с начальным символическим адресом DATA_X. Для хранения значений Y_i будем использовать массив с начальным адресом DATA_Y.

```

; ISR3 – подпрограмма обработки запроса прерывания INT3
DATA_X EQU 0800H
DATA_Y EQU 0810H
ISR3:  XRA A

```

```

OUT PC
OUT OPORT1
OUT OPORT2
OUT OPORT3
OUT OPORT4
OUT OPORT5
; Ввод и обработка сигналов X1, X2
LXI H, DATA_X
IN PA
MOV C, A
MOV M, A ; Запомнить X1 и X2
ANI 0000 0001B ; Выделить разряд D0
INX H
MOV M, A ; Запомнить X1
MOV A, C
RRC
ANI 0000 0001B
INX H
MOV M, A ; Запомнить X2
DCX H
ORA M ; (A)=X1 ∨ X2
CMA ; (A)=  $\overline{X1 \vee X2}$ 
MOV B, A
MOV A, M ; (A)=X1
CMA ; (A)=  $\overline{X1}$ 
ANA B ; (A)=f(X1, X2)=Y1
LXI H, DATA_Y
MOV M, A ; Запомнить Y1
JZ EXIT ; Идти на метку EXIT, если Y1=0
; Формирование импульса длительностью t1

```

FORM_Y1:

Схема формирования импульса Y1 приведена на рис. 3.13. В ней использован счетчик СТ2 1-го таймера ПТ_1, работающий в режиме 0 (программируемая задержка). Для формирования задержки, например, 30 мкс при $F_{2\text{TTL}}=2$ МГц (период $T=0,5$ мкс) в СТ2 нужно загрузить число

$$N2 = \frac{t1}{T} = \frac{30}{0,5} = 60 = 3\text{CH}.$$

Управляющее слово для СТ2: счет двоичный, режим 0, загрузить только младший байт – 10010000B=90H.

```

FORM_Y1: MVI A, 90H
          OUT PUC53_1 ; Загрузить управляющее слово

```

```

MVI A, 3CH
OUT CT2_1      ; Загрузить N2
MVI A, 01H
OUT PUC55      ; Выдать U0=1

```

Примечания:

1. В программе для управления выходами порта РС ППА используется управляющее слово установки сброса битов, с помощью которого можно выводить 1 или 0 в любой разряд РС независимо от других разрядов. Этой возможностью обладает только порт РС.

2. Имеется другая возможность независимого управления разрядами любого порта ППА – с помощью чтения состояния выходов порта и использования логических команд. Например, надо установить 1 на выходе РС0 порта РС:

```

IN PC          ; Чтение состояния выходов РС
ORI 0000 0001B
OUT PC         ; Установить РС0 ← 1

```

Сбросить в 0 разряд РС3 порта РС:

```

IN PC
ANI 1111 0111B
OUT PC         ; Сбросить РС3 ← 0

```

3. Если в качестве порта вывода используется обычный регистр, то определить программно состояния его выходов невозможно. Поэтому для реализации индивидуального управления разрядами порта необходимо после каждого вывода в порт запоминать содержимое аккумулятора в специальной ячейке ОЗУ. Это содержимое будет определять состояние выходов порта. Например:

```

STATUS EQU 0900H      ; Адрес ячейки ОЗУ для хранения со-
                      ; стояния выходов порта вывода
OPORT EQU XXH         ; Адрес порта
                      ; Первый вывод в порт
XRA A
OUT OPORT
STA STATUS            ; Вывод 1 в 0-й разряд порта
LDA STATUS
ORI 0000 0001B
OUT OPORT
STA STATUS           ; Запомнить состояние выходов порта
.....
                      ; Вывод 0 в 3-й разряд порта

```

```
LDA STATUS
ANI 1111 0111B
OUT OPORT
STA STATUS
```

БСА обработки аналоговых сигналов V1 и V2 приведена на рис. 2.6. С помощью АЦП сигналы V1 и V2 преобразуются в двухбайтные целые двоичные числа W1 и W2. Для их сохранения необходимо выделить в ОЗУ массив, начальный адрес которого DATA_W. Предположим, что требуется вычислить функцию

$$g(V1, V2, K1, K2) = W1 + W2 - K1,$$

где K1 и K2 – двухбайтные коэффициенты, хранящиеся в ПЗУ.

Полученное значение функции g() сравнивается с двухбайтной константой Q, хранящейся в ПЗУ. Сравнение можно выполнить с помощью программы вычитания двухбайтных целых чисел g() – Q.

Результат сравнения определяется флагом CY микропроцессора после вычитания старших байтов. Если CY=0, то g() > Q, если CY=1, то g() < Q.

Для размещения двухбайтных констант в ПЗУ можно использовать псевдокоманду DW, которую надо записать в неисполняемую микропроцессором часть программы, например, после команд JMP, RET, HLT:

```
JMP MET1 ; Команда программы
ADR_K1: DW XXXXH ; Символический адрес константы K1
ADR_K2: DW XXXXH
ADR_Q: DW XXXXH
MET1: ..... ; Очередная команда программы
```

Если необходимо поместить константы в заранее заданные ячейки ПЗУ, то начальный адрес задается псевдокомандой ORG:

```
ORG 0700H
ADR_K1: DW XXXXH
```

Программу аналого-цифрового преобразования удобно оформить в виде подпрограммы с именем ADCONV, которая вызывается по мере необходимости. Перед ее вызовом в пару HL необходимо записать адрес ячеек для хранения полученного кода АЦП.

```
LXI H, DATA_W
CALL ADCONV
.....
.....
ADCONV: MVI A, 07H
OUT PUC55 ; Вывод PC3 ← 1
MVI A, 06H
OUT PUC55 ; Вывод PC3 ← 0, запуск АЦП
WAIT: IN PA
```

```

ANI 0010 0000B ; Выделить разряд D5
JNZ WAIT      ; Ожидание готовности данных АЦП
IN PB
MOV M, A      ; Запомнить младший байт кода АЦП
INX H
IN PA
RLC
RLC
ANI 0000 0011B ; Выделить D1, D0
MOV M, A      ; Запомнить старший байт кода АЦП
RET

```

Фрагмент программы ввода и обработки сигналов V1 и V2 имеет вид:

```

; На входах коммутатора A1=A0=0, при этом Y=X1
; Ввод сигнала V1
    LXI H, DATA_W
    CALL ADCONV
; Установить на коммутаторе A0=1, A1=0, при этом Y=X2
    MVI A, 05H
    OUT PUC55      ; Вывод PC2 ← 1
; Ввод сигнала V2
    INX H
    CALL ADCONV
; Вычисление суммы W1+W2
; Пересылка W2 в регистры D, E
    MOV D, M
    MOV E, M
    DCX H
    DCX H
    MOV A, M
    ADD E
    MOV E, A      ; Младший байт суммы
    INX H
    MOV A, M
    ADD D
    MOV D, A      ; Старший байт суммы
; Вычисление g( )=W1+W2-K1
    LXI H, ADR_K1
    MOV A, E
    SUB M
    MOV E, A      ; Младший байт разности
    INX H
    MOV A, D
    SBB M

```

```

MOV D, A ; Старший байт разности
; Вычисление g() – Q
LXI H, ADR_Q
MOV A, E
SUB M
INX H
MOV A, D
SBB M
JC FORM_Y2 ; Если g()–Q < 0
FORM_Y3: ..... ; Если g()–Q > 0
FORM_Y2: .....

```

Программы FORM_Y2 и FORM_Y3 формируют одиночные импульсы на выходах Y2 и Y3 длительностью t2 и t3. Они подобны программе FORM_Y1. Для формирования задержек в них используются счетчики CT0 и CT1 2-го таймера ПТ_2 (см. рис. 3.13).

После окончания обработки аналоговых сигналов необходимо проверить положение тумблера “Останов” на пульте управления (ПУ). Схема включения тумблера приведена на рис. 3.17, где он обозначен как SA5. Если SA5 разомкнут, то происходит возврат в основную программу. Если же SA5 замкнут, то производится включение светодиода с надписью “Останов” на ПУ и останов МП. Программа проверки имеет вид:

```

CHECK: IN IPORT
ANI 0010 0000B ; Выделить разряд D5
JZ STOP ; Если SA5 замкнут
EI
RET
STOP: MVI A, 80H
OUT OPORT1 ; Включить VD8 “Останов”
HLT

```

Подпрограмма ISR1 является обработчиком прерывания по запросу INT1, вырабатываемого при нажатии на кнопку “Прерывание” на ПУ (прерывание оператора). БСА подпрограммы ISR1 приведена на рис. 3.19. Рассмотрим программную реализацию некоторых блоков алгоритма.

При входе в подпрограмму ISR1 нужно сохранить в стеке содержимое всех регистров МП, что может быть необходимо для корректной работы подпрограмм более низкого приоритета – ISR2 и ISR3.

```

ISR1: PUSH PSW ; Сохранить в стеке содержимое
PUSH B ; регистров A, F, B, C, D, E, H, L
PUSH D
PUSH H
.....

```

Проверить, замкнут ли один из переключателей SA1-SA4, можно программой:

```

IN IPORT
ANI 0FH           ; Выделить разряды D3-D0
CPI 0FH
JZ CHECK         ; Переход, если SA1-SA4 разомкнуты
CALL IDENTIF     ; Вызов подпрограммы идентификации
    
```

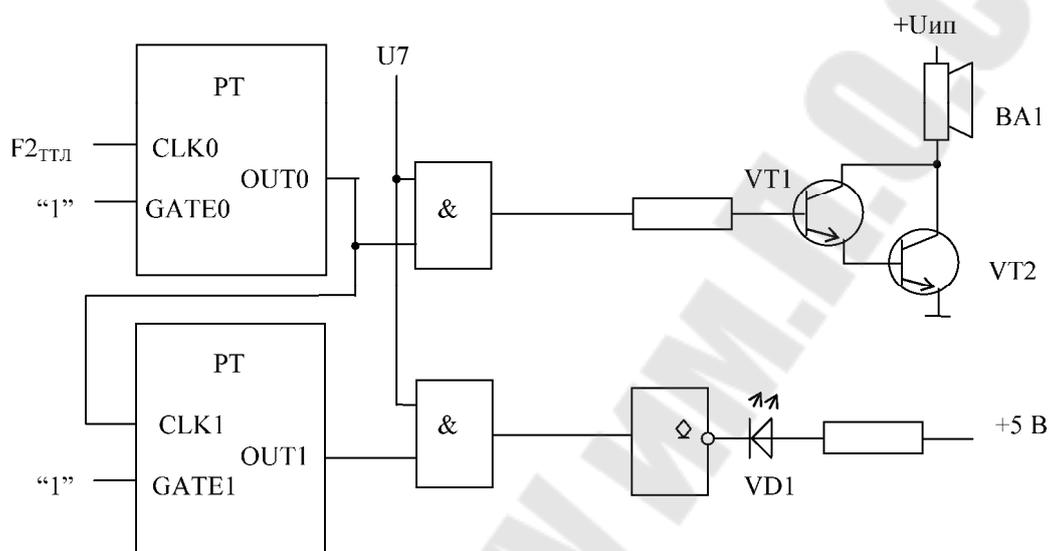


Рис. 3.18. Схема аварийной сигнализации

Определение номера замкнутого переключателя SA1-SA4 выполняет подпрограмма IDENTIF, которая присваивает ему двоичный позиционный код в соответствии с табл. 3.

Таблица 3

Коды для переключателей SA1-SA4

Замкнут SA	Унитарный код	Позиционный код, HEX
SA1	XXXX 1110	00
SA2	XXXX 1101	01
SA3	XXXX 1011	02
SA4	XXXX 0111	03

```

; IDENTIF – подпрограмма идентификации замкнутого SA1-SA4
; Выходной параметр: регистр С – позиционный код переключателя
MASK1 EQU 0000 1110B
MASK2 EQU 0000 1101B
MASK3 EQU 0000 1011B
MASK4 EQU 0000 0111B
IDENTIF: IN IPORT
ANI 0FH
MOV B, A
    
```

```

CPI MASK1
JZ EXIT1           ; Если D0=0
MOV A, B
CPI MASK2
JZ EXIT2           ; Если D1=0
MOV A, B
CPI MASK3
JZ EXIT3           ; Если D2=0
MVI C, 03H        ; Код переключателя SA4
RET
EXIT1: MVI C, 00H  ; Код переключателя SA1
RET
EXIT2: MVI C, 01H  ; Код переключателя SA2
RET
EXIT3: MVI C, 02H  ; Код переключателя SA3
RET

```

Если замкнут один из переключателей SA1-SA4, то на линейный дисплей выводится соответствующий код W1-W4 из массива DATA_W. Схема подключения индикаторов дисплея приведена на рис. 3.15. При программировании вывода на дисплей следует учесть, что на индикаторы необходимо выводить семисегментный двоичный код, а МП оперирует с данными в позиционном двоичном коде. Поэтому необходима перекодировка перед выводом информации на дисплей.

```

; CROSSCODE – подпрограмма перекодировки
; Входной параметр: регистр C – позиционный код символа
; Выходной параметр: регистр A – семисегментный код символа
CROSSCODE: MVI B, 0
            LXI H, TABLE           ; В паре HL – адрес TABLE
            DAD B                   ; (HL)←(HL)+(BC)
            MOV A, M
            RET
TABLE:     DB 3FH                   ; Код символа “0”
            DB 06H                   ; Код символа “1”
            .....
            DB 71H                   ; Код символа “F”

```

Полную таблицу семисегментных кодов можно найти в [10].

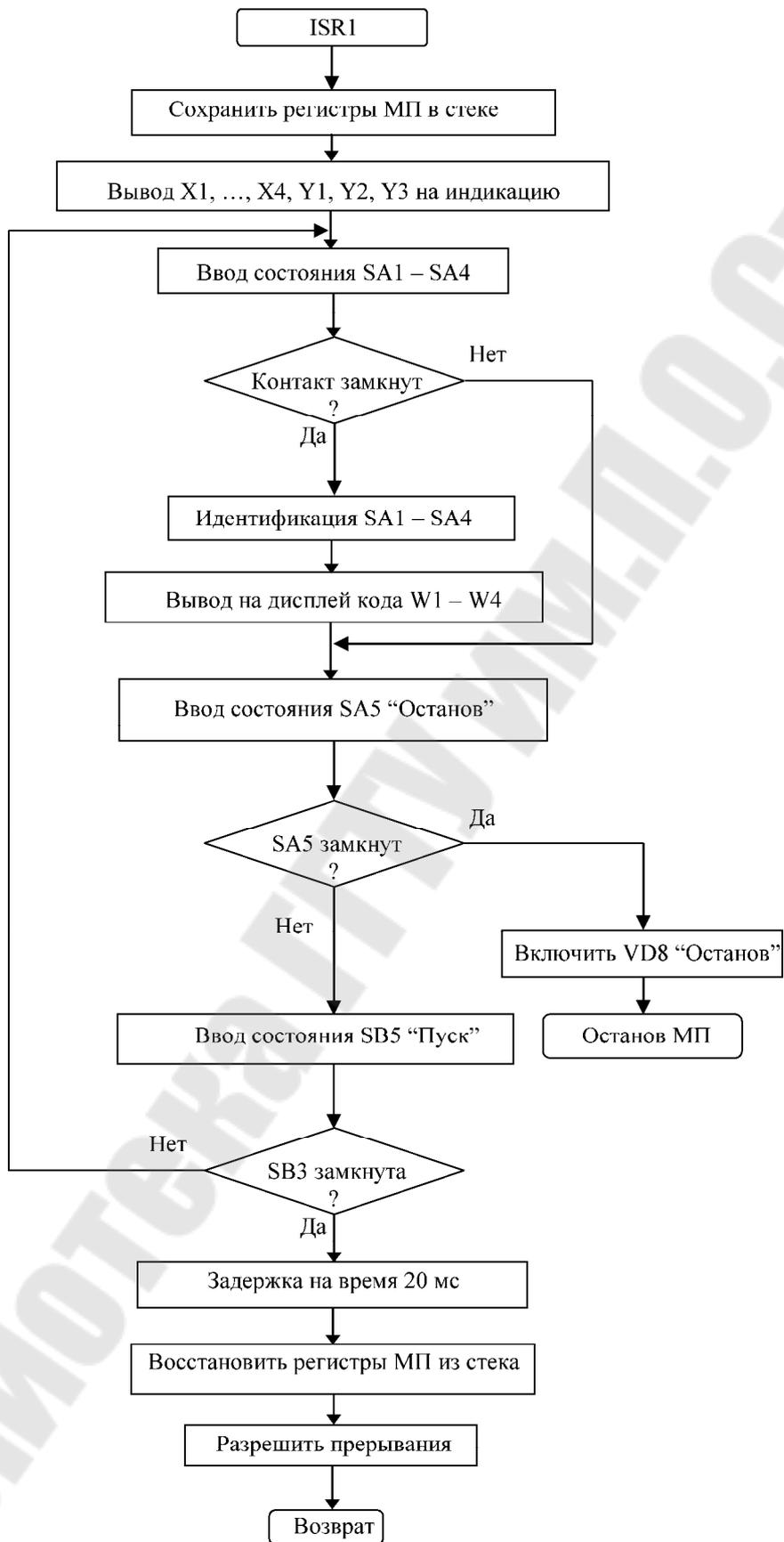


Рис. 3.19. БСА программы работы ПУ

Программа вывода на дисплей имеет вид:

```
DISPLAY:   LXI D, DATA_W
           LDAX D
           ANI 0FH
           MOV C, A
           CALL CROSSCODE ; Перекодировка
           OUT OPORT2    ; Вывод на HG1
           LDAX D
           RRC
           RRC
           RRC
           RRC
           ANI 0FH
           MOV C, A
           CALL CROSSCODE ; Перекодировка
           OUT OPORT3    ; Вывод на HG2
           INX D
           .....
```

Проверка состояния кнопки SB3 “Пуск” может быть выполнена следующим образом:

```
IN IPORT
ANI 80H ; Выделить разряд D7
JNZ BEGIN ; Идти на начало, если не нажата
CALL DELAY ; Подпрограмма задержки
; Подготовка к возврату из прерывания
MVI A, 10H
SIM ; Сброс триггера на входе RST7.5
POP H ; Восстановление регистров
POP D
POP B
POP PSW
EI ; Разрешить прерывания
RET
```

Подпрограмма DELAY обеспечивает задержку около 20 мс, что обеспечивает защиту от дребезга контактов кнопки. Ее можно реализовать программным способом по методике, изложенной в [9].



Рис. 3.20. БСА подпрограммы аварийной сигнализации

По сигналу от аварийного датчика в МПС поступает запрос прерывания INT0 и происходит переход к подпрограмме ISR0. Функциональная схема реализации аварийной сигнализации приведена на рис. 3.18, а БСА программы – на рис. 3.20. В схеме используются счетчики СТ0 и СТ1 1-го таймера ПТ_1, так как при аварии частоту опроса $f_{опр}$ генерировать не нужно. Счетчики работают в режиме 3 – генератора прямоугольных импульсов (меандра). После инициализации счетчиков подается лог. 1 с выхода 7-го разряда порта РС ППА (сигнал U7), который разрешает выдачу импульсов на светодиод VD1 и динамик ВА1. Затем необходимо включить светодиод с надписью “Останов” (VD8 на ПУ) и остановить МП.

При вычислении величины Z , являющейся функцией кода $W3$, $Z=e(W3, K3, K4, K5, K6)$, можно воспользоваться программами арифметических операций, приведенными в [3]. При этом предполагается, что коэффициенты $K3-K6$ подобраны таким образом, что все исходные и промежуточные данные, а также результат – двухбайтные беззнаковые дробные числа с фиксированной запятой, т.е. всегда меньше 1. Полученное значение Z следует сохранить в массиве с именем DATA_Z (сначала младший, потом старший байт) для последующего вывода на ЦАП.

При разработке схемы сопряжения ЦАП с МПС следует иметь в виду, что все разряды кода должны поступать на вход ЦАП одновременно, иначе возникают большие выбросы на его выходе [5]. На рис. 3.21 приведена функциональная схема подключения 10-разрядного ЦАП К572ПА1 к системной шине МПС. Регистры DD2, DD3 служат для вывода на ЦАП полного 10-разрядного кода. Они открываются одновременно с помощью логического элемента DD4.2 при выводе в порт с символическим именем OPORT7. Регистр DD1 принимает из МПС младший байт кода. Программа вывода на ЦАП имеет вид:

```

LXI  H, DATA_Z
MOV  A, M
OUT  OPORT6      ; Вывод мл.байта кода в регистр
  
```

```

; DD1
INX H
MOV A, M
OUT OPORT7 ; Вывод кода на ЦАП

```

На рис. 3.22 приведена функциональная схема генератора сигнала Y5, представляющего собой прямоугольные импульсы с частотной (ЧИМ) или широтной (ШИМ) модуляцией. В схеме используются счетчики СТ1 и СТ2 2-го таймера ПТ_2. Для обоих видов модуляции счетчик СТ1 работает в режиме 2 (делитель частоты), а счетчик СТ2 – в режиме 0 (программируемая задержка). Сигнал U6, снимаемый с выхода РС6 порта ППА разрешает выдачу Y5 и запроса прерывания INT2.

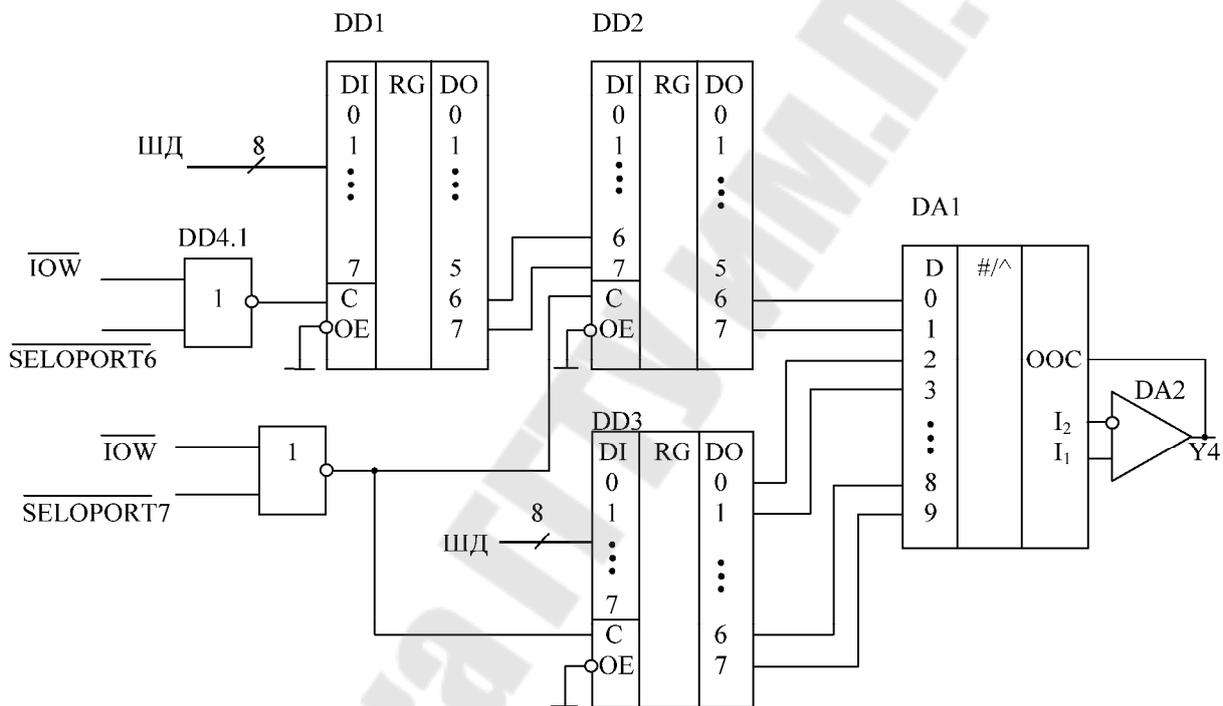


Рис. 3.21. Схема подключения ЦАП к МПС

Рассмотрим программную реализацию генератора на примере формирования импульсов с ШИМ. Исходными данными являются: частота следования $f=1000$ Гц (период $T=1$ мс), минимальная длительность импульса $t_{и.мин}=50$ мкс, максимальная длительность $t_{и.макс}=800$ мкс, тактовая частота $F_{2ТТЛ}=2$ МГц.

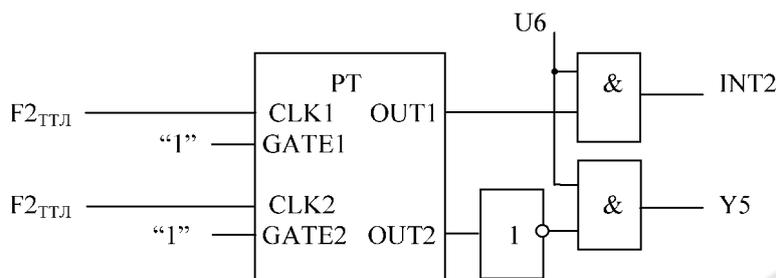


Рис. 3.22. Схема формирования сигнала Y5

Определим управляющие слова для счетчиков. Счетчик СТ1: счет двоичный, режим 2, загружать 2 байта, управляющее слово – 01110100В=74Н. В счетчик надо загрузить число

$$N1 = \frac{2 \cdot 10^6}{1 \cdot 10^3} = 2 \cdot 10^3 = 07D0H.$$

Для СТ2: счет двоичный, режим 0, загружать 2 байта, управляющее слово – 10110000В=В0Н. Число N2, которое надо загрузить в счетчик, вычисляем по формуле

$$N2 = N2_{\text{мин}} + \frac{N2_{\text{макс}} - N2_{\text{мин}}}{3FF} \cdot W4,$$

где $N2_{\text{мин}}$ – минимальное значение, соответствующее $t_{i.\text{мин}}$ и коду АЦП $W4=0000H$;

$N2_{\text{макс}}$ – максимальное значение, соответствующее $t_{i.\text{макс}}$ и коду АЦП $W4=03FFH$ (для 10-разрядного АЦП).

Вид формулы поясняет рис. 3.23.

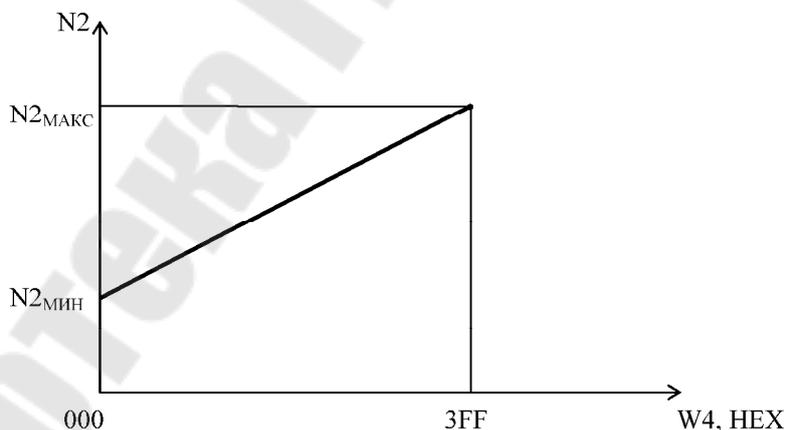


Рис. 3.23. График функции $N2 = F(W4)$

Для вычисления значений N2 можно воспользоваться программами из [3]. Будем предполагать, что двухбайтное число N2 размещается в ОЗУ по адресу DATA_N2 (сначала младший, потом старший байт).

В подпрограмму ISR3 необходимо внести команды, обеспечивающие инициализацию счетчиков и выдачу сигналов Y5, INT2:

```

DI ; Запретить прерывания
MVI A, 74H
OUT PUC53_2 ; Режим CT1_2
MVI A, 0B0H
OUT PUC53_2 ; Режим CT2_2
MVI A, 0D0H
OUT CT1_2
MVI A, 07H
OUT CT1_2 ; Запуск CT1_2
LXI H, DATA_N2
MOV A, M
OUT CT2_2
INX H
MOV A, M
OUT CT2_2 ; Запуск CT2_2
MVI A, 0DH
OUT PUC55 ; Установить  $U_6 \leftarrow 1$ 
EI ; Разрешить прерывания

```

Подпрограмма ISR2, которая вызывается по запросу прерывания INT2 с заданной частотой следования импульсов f , служит для очередного запуска счетчика CT2_2, формирующего однократную задержку длительности t_i . Подпрограмма имеет вид:

```

ISR2:    PUSH PSW
         PUSH H
         LXI H, DATA_N2
         MOV A, M
         OUT CT2_2
         INX H
         MOV A, M
         OUT CT2_2 ; Запуск CT2_2
         POP H
         POP PSW
         EI
         RET

```

Если необходимо получить импульсы Y_5 с частотной модуляцией (ЧИМ), то в счетчик CT2 надо загрузить постоянное число, пропорциональное заданной длительности импульса t_i . В счетчик CT1 надо загружать переменное число N_1 из массива DATA_N1, которое будет пропорционально коду W_4 и находиться в диапазоне от $N_{1\text{мин}}$ до $N_{1\text{макс}}$.

4. Оформление курсового проекта

Курсовой проект состоит из пояснительной записки и графической части.

4.1. Пояснительная записка

Пояснительная записка должна содержать:

- Титульный лист
- Задание на курсовой проект
- Оглавление с указанием страниц
- Введение
- Разработка структурной схемы МПС
- Разработка схем отдельных модулей МПС
- Разработка программного обеспечения работы МПС
- Заключение
- Список литературы
- Приложение

Во введении должны быть сформулированы цели и задачи курсового проекта, кратко охарактеризовано содержание работы.

Структурная схема разрабатывается в соответствии с индивидуальным заданием. Необходимо дать краткое описание состава и назначения основных элементов системы.

В разделе “Разработка схем отдельных модулей МПС” необходимо разработать функциональные схемы модулей процессора, памяти, прерывания, ввода-вывода цифровой и аналоговой информации, пульта управления. Нужно дать краткое описание микросхем, входящих в эти модули (заданных или выбранных самостоятельно). Привести их условное изображение, назначение выводов, таблицы функционирования и т.п.

В разделе “Разработка программного обеспечения работы МПС” необходимо привести карту распределения адресного пространства памяти: подпрограммы, данные, стек, константы и т.п. Также нужно определить адреса всех портов. Необходимо привести блок-схемы алгоритмов работы МПС и их программную реализацию на языке Ассемблера. При составлении программ надо использовать символические адреса ячеек памяти, портов и имена констант. Листинги программ на Ассемблере должны включать псевдокоманды, метки, мнемонику команд и обязательно комментарии. Перевода в машинные коды команд делать не надо.

В заключении необходимо привести основные результаты выполнения курсового проекта, параметры разработанной МПС (например, количество микросхем, объем занимаемой постоянной и оперативной памяти, напряжения и токи, требуемые от источников питания и т.п.)

В приложении помещается перечень элементов принципиальной схемы разработанной МПС.

Пояснительная записка оформляется компьютерными средствами на листах формата А4 (210×297 мм) в соответствии с требованиями ГОСТ 2.105 - 95, должна быть переплетена и подписана. Примерный объем пояснительной записки - 25-30 страниц.

Графическая часть

Графическая часть курсового проекта состоит из двух чертежей формата А1:

1. Принципиальная электрическая схема МПС.
2. Блок-схемы алгоритмов программ работы МПС.

Все элементы принципиальной схемы должны иметь нумерацию, а микросхемы - нумерацию выводов.

Примечание. Нумерация микросхем и других элементов на общей принципиальной схеме, как правило, отличается от нумераций на схемах отдельных модулей.

Графическая часть должна быть выполнена компьютерными средствами или вручную с применением чертежных инструментов и оформлена в соответствии с требованиями ЕСКД.

Список рекомендуемой литературы

1. Каган Б.М., Сташин В.В. Основы проектирования микропроцессорных устройств автоматики. - М.: Энергоатомиздат, 1987.
2. Майоров В.Г., Гаврилов А.И. Практический курс программирования микропроцессорных систем. - М.: Машиностроение, 1989.
3. Гуртовцев А.Л., Гудыменко С.В. Программы для микропроцессоров. - Мн.: Высш. шк., 1989.
4. Микропроцессоры и микропроцессорные комплекты интегральных микросхем: Справочник. В 2 т. / Под ред. В.А. Шахнова. - М.: Радио и связь, 1988.
5. Федорков Б.Г., Телец В.А., Дегтяренко В.П. Микроэлектронные цифроаналоговые и аналогоцифровые преобразователи. - М.: Радио и связь, 1985.
6. Цифровые и аналоговые интегральные микросхемы: Справочник / Под ред. С.В. Якубовского. - М.: Радио и связь, 1990.
7. Полупроводниковые БИС запоминающих устройств: Справочник / Под ред. А.И. Гордонова и М.Н. Дьякова. - М.: Радио и связь, 1987.
8. Лебедев О.Н. Микросхемы памяти и их применение. - М.: Радио и связь, 1990.
9. Методические указания к лабораторным занятиям курса "Микропроцессорная техника", ч. 4 - Гомель: ГПИ, 1995.
10. Методические указания к лабораторным занятиям курса "Микропроцессорная техника", ч. 5 - Гомель: ГПИ, 1997.
11. Методические указания к лабораторным занятиям курса "Микропроцессорная техника", ч. 7 - Гомель: ГГТУ им. П.О. Сухого, 2000.
12. Смоленчук В.С., Рязанцева Н.В. Технические средства микропроцессорных систем. Методические указания по курсовому проектированию. - Гомель: БЕЛИИЖТ, 1993.

Оглавление

Введение.....	3
1. Общие принципы проектирования микропроцессорных систем.....	3
2. Задание на проектирование.....	6
2.1. Алгоритм работы МПС.....	8
2.2. Обработка запросов прерывания.....	11
2.3. Пульт управления.....	11
2.4. Структурная схема МПС.....	13
3. Методические указания по выполнению курсового проекта.....	15
3.1. Модуль процессора.....	15
3.2. Модуль памяти.....	18
3.3. Модуль прерывания.....	20
3.4. Таймер опроса.....	22
3.5. Схемы ввода цифровых и аналоговых сигналов.....	23
3.6. Пульт управления.....	25
3.7. Примеры разработки программ работы МПС.....	27
4. Оформление курсового проекта.....	44
4.1. Пояснительная записка.....	44
4.2. Графическая часть.....	45
Список рекомендуемой литературы.....	46

ПРОЕКТИРОВАНИЕ МИКРОПРОЦЕССОРНОЙ СИСТЕМЫ КОНТРОЛЯ И УПРАВЛЕНИЯ ОБЪЕКТОМ

**Практическое руководство
к курсовому проектированию
по дисциплине «Микропроцессорная техника»
для студентов специальности 1-36 04 02
«Промышленная электроника»
дневной и заочной форм обучения**

Автор-составитель: **Виноградов Эдуард Михайлович**

Подписано в печать 14.04.06.

Формат 60x84/16. Бумага офсетная. Гарнитура Таймс.

Ризография. Усл. печ. л. 2,68. Уч.-изд. л. 3,26.

Изд. № 156.

E-mail: ic@gstu.gomel.by

<http://www.gstu.gomel.by>

Отпечатано на цифровом дуплекаторе
с макета оригинала авторского для внутреннего использования.

Учреждение образования «Гомельский государственный
технический университет имени П.О. Сухого».

246746, г. Гомель, пр. Октября, 48.