

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Северо-Кавказский филиал ордена Трудового Красного Знамени федерального
государственного бюджетного образовательного учреждения высшего образования
«Московский технический университет связи и информатики»



Методические указания
по выполнению курсового проекта

по дисциплине

МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ

направление подготовки 09.03.01 Информатика и вычислительная техника
(профиль "Вычислительные машины, комплексы, системы и сети"
"Программное обеспечение и интеллектуальные системы")

Ростов-на-Дону
2019

УДК 681.3.06 (076)
ББК 32.07

Чикалов А.Н. Микропроцессорные системы. Методические указания по выполнению курсового проекта. Ростов-на-Дону: Северо-Кавказский филиал МТУСИ, 2019.- 44с.

В пособии изложены цели и задачи дисциплины, структура дисциплины и краткое содержание каждого раздела. Кроме того, приведены индивидуальные задания на курсовой проект, правила оформления и примеры выполнения этапов курсового проекта.

Методические указания содержат основную информацию, необходимую для выполнения курсового проекта по дисциплине "Микропроцессорные системы". Курсовой проект предусматривает разработку аппаратных средств до уровня принципиальной электрической схемы и части управляющей программы простейшей микропроцессорной системы (МПС) на базе процессоров семейства Intel x86. Такие МПС широко используются как встраиваемые системы управлений в различную аппаратуру управления и связи.

Идея работы заимствована в материалах: Курасов П.В. Методические указания и задания на курсовой проект по дисциплине Микропроцессорные системы. М.: МТУСИ, 2006.

Методические указания предназначены для студентов, обучающихся по направлению 09.03.01 "Информатика и вычислительная техника" профиль "Вычислительные машины, комплексы, системы и сети", "Программное обеспечение и интеллектуальные системы" очной и заочной форм обучения. Пособие может быть использовано также в процессе самостоятельной работы.

Рассмотрено и одобрено
на заседании кафедры ИВТ
Протокол от 26 августа 2019 г. № 1

Рецензент Зав.кафедрой ИВТ д.т.н. профессор Соколов С.В.

Содержание

1.	Цели дисциплины	4
	Задачи дисциплины	4
	Трудоемкость дисциплины	6
2.	Содержание дисциплины	6
3.	Лабораторные работы и практические занятия	12
4.	Список литературы	14
5.	Задание на курсовой проект	14
6.	Требования к оформлению курсового проекта	16
7.	Пример оформления курсового проекта	18
8.	Методические указания для выполнения задания	24
	8.1. Блок центрального процессора	24
	8. 2. Блок дешифрации адресов	29
	8.3. Блок памяти	37
	8.4. Блок внешнего интерфейса	40
	8.5. Основная программа МПС	41

1. ЦЕЛИ ДИСЦИПЛИНЫ

Целями изучения дисциплины "Микропроцессорные системы" является приобретение студентами базовых знаний в области интегральных микропроцессорных устройств и микроконтроллеров, необходимых для реализации информационных технологий и построения телекоммуникационных и управляющих устройств с требуемыми функциональными возможностями.

Задачи дисциплины

Бакалавр по направлению подготовки 09.03.01 должен решать задачи в рамках *проектной* профессиональной деятельности.

Процесс изучения дисциплины направлен на формирование следующих компетенций:

ПК-1: способен производить разработку и отладку программного кода, интегрировать программные модули и компоненты, проектировать программное обеспечение.

В результате изучения дисциплины для достижения перечисленных задач обучаемый должен:

знать:

- архитектуру, возможности, устройство и функционирование микропроцессорных систем, коммуникационное оборудование;
- методы и приемы формализации, алгоритмизации, программирования и оформления программного кода;
- компоненты программно-технических архитектур, существующие приложения и интерфейсы взаимодействия с ними;

уметь:

- разрабатывать программное обеспечение с использованием языков и сред программирования, выполнять определение и манипулирование данными;
- осуществлять тестирование, отладку и оптимизацию программного обеспечения;
- использовать выбранную среду программирования для разработки процедур интеграции программных модулей;

владеть:

- приемами анализа возможностей и разработки требований к программному обеспечению;
- методами проектирования программного обеспечения и баз данных;

- методами и средствами интеграции модулей и компонент программного обеспечения, приемами развертывания и обновления программного обеспечения

Трудоёмкость дисциплины

Общая трудоёмкость дисциплины 180 часов, в том числе аудиторных занятий:

- заочная форма обучения – 20 часов;
- очная форма обучения - 80 часов.

2. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

1. Схемы запоминающих устройств
1.1. Постоянные ЗУ. Назначение и классификация схем памяти. Обобщенная структурная схема полупроводникового ЗУ. Временные диаграммы работы ЗУ. Основные параметры и характеристики полупроводниковых ЗУ. Классификация ПЗУ. Особенности структурной схемы ПЗУ. Особенности ИМС ПЗУ
1.2. Оперативные ЗУ. Оперативные ЗУ. Статические и динамические ОЗУ. Схемотехника ОЗУ. Построение модулей ОП. Схемы регенерации. Нарастивание емкости модулей. Семейство ОЗУ в сериях ИМС. Параметры и характеристики ИМС ОЗУ
1.3. Организация модулей ЗУ. Нарастивание емкости и разрядности памяти. Принципы управления модулями памяти. Построение дешифраторов блоков памяти
1.4. Принципы управления памятью в МПС. Типы адресов при работе с памятью. Распределение адресного пространства. Страничная и сегментная организация памяти
1.5. Программирование схем ПЗУ. Технология программирования схем ПЗУ. Применение схем с программируемой логикой: ПЛИС, ПЛМ, FPGA
Вопросы и упражнения для самоконтроля: <ol style="list-style-type: none">1. Сформулируйте назначение входов и выходов микросхемы памяти 155PE3.2. Перечислите способы записи и стирания информации в ПЗУ.3. Разработать индикатор телевизионных каналов на базе ПЗУ.4. Разработать дешифратор на ПЗУ 155PE3, тип выходов которого (прямые или инверсные) выбирается управляющим сигналом.5. В чем состоят особенности микросхем динамической памяти?6. Какие сигналы присутствуют на выходах микросхем 573РФ2 и 155PE3 при пассивных сигналах на входах выбора микросхемы?7. Как обозначаются на корпусах различные типы микросхем?8. Как, помимо программирования и управления старшим адресным разрядом можно изменить направление движения бегущей строки?

9. Сформулируйте на уровне структурной схемы предложения по созданию информационного табло по принципу бегущей строки, позволяющего отображать произвольную текстовую и числовую информацию.
10. Нарисуйте структурную схему ПЗУ. Каково назначение основных структурных элементов? Каково назначение входов и выходов ПЗУ?
11. Назовите основные параметры ПЗУ, охарактеризуйте их физический смысл.
12. Какие типы выходов применяются в микросхемах ПЗУ? Для чего их используют?
13. Какое состояние имеют элементы памяти ПЗУ до программирования? Как это определить?
14. Каково назначение входов и выходов ИМС 155PE3? Каково ее условное графическое обозначение на схемах?
15. Каков порядок считывания данных из ИМС ПЗУ?
16. Для каких целей используют ПЗУ?
17. В чем состоят особенности микросхем динамической памяти?
18. Каково назначение основных входов и выходов в микросхемах памяти?
19. Каким образом увеличить разрядность шины данных в микросхемах памяти? Нарисуйте схему такого наращивания.
20. Каким образом увеличить число ячеек блока памяти? Нарисуйте схему такого наращивания.
21. Сформулируйте этапы синтеза дешифратора блока памяти.
22. Синтезируйте дешифратор с заданной областью адресов блока памяти.
23. Каким образом можно увеличить число выходов дешифратора адресов памяти?
24. Каким образом дешифратор подключается к сигнальным выводам микропроцессора?
25. Каков коэффициент расширения выводов шины адреса микропроцессора?
26. Каким образом увеличивают коэффициент расширения выводов адреса микропроцессора?
27. Как реализуются совмещенные и отдельные адресные пространства МП? Какие достоинства

2. Организация микропроцессорных систем

2.1. Основные принципы работы МП. Базовые понятия. Архитектуры Фон Неймана и Гарвардская. Модель Глушкова. Алгоритм управления ЦП

2.2. Классификация микропроцессорных устройств. Классификация МП-устройств. Состав МП-комплекта. Современные МП высокой производительности. Процессоры Alpha, PA, IA-64, Rxxxx, UltraSPARC. Способы повышения производительности

2.3. Архитектура микропроцессорных устройств. Структурная схема МП и ее работа. Программная модель МП. Регистр флагов. Слово состояния процессора. Шины семейств PC-bus. Шина PCI. Шина Q-bus

2.4. Управляющий цикл процессора. Организация внешних связей МП.

<p>Машинный цикл процессора. Типы управляющих конструкций</p>
<p>2.5. Система команд микропроцессора. Способы адресации. Команды пересылки, арифметические, логические, передачи управления, управления процессом Регистр флагов. Слово состояния процессора. Организация внешних связей МП. Машинный цикл процессора. Функционирования МП при выполнении команд работы со стеком и указателем стека. Выполнение команд условного и безусловного перехода, вызова и возвращения из подпрограмм, разработка подпрограммы, разработка обработчиков</p>
<p>2.6. Программное обеспечение МП. Структура ПО МПС. Технология создания исполняемых модулей. Операторы Ассемблера. Совмещенное и раздельное адресное пространство. Подключение памяти и внешних устройств к шинам. Способы организации модулей</p>
<p>2.7. Организация параллельного интерфейса МП. Назначение, программная модель параллельного интерфейса. Протоколы Centronics, IEEE 1284. Режимы работы. Программирование работы</p>
<p>2.8. Организация последовательного интерфейса МП. Назначение, программная модель последовательного интерфейса. Протоколы RS-232, USB. Режимы работы. Программирование работы</p>
<p>2.9. Организация работы счетчиков-таймеров МП. Назначение, программная модель таймеров. Режимы работы. Программирование работы</p>
<p>2.10. Организация прерываний и ПДП МП. Назначение, программная модель контроллеров. Режимы работы. Таблица векторов. Приоритезация и маскирование. Механизм обслуживания запросов. Программирование работы контроллеров. Программирование контроллера ПДП. Схема взаимодействия с процессором. Структурная схема контроллера. Программирование контроллера прерываний</p>
<p>2.11. Взаимодействие МП с устройствами ввода-вывода. Принципы работы устройств. Программные модели устройств. Принципы программирования контроллеров</p>
<p>2.12. Проектирование МПС. Применение МПС. Этапы проектирования. Основные этапы разработки аппаратного и программного обеспечения. Кросс-ассемблеры. Моделирующие программы. CASE-технологии. Системы САПР. Отладка аппаратного и программного обеспечения. Логические и сигнатурные анализаторы. Разработка и отладка программы</p>
<p>Вопросы и упражнения для самоконтроля:</p> <ol style="list-style-type: none"> 1. Почему пересылки между А, В, С, D, H, L занимают всего 1 машинный цикл, а с М значительно длиннее? 2. Какие команды занимают 1 байт, какие 2 байта и 3 байта? Почему? 3. Что происходит при выполнении MOV A,A? 4. Каково состояние регистров-источников при окончании пересылки? 5. Сравните время выполнения команд NOP и MOV C,C. 6. Занимается ли ячейка памяти при выполнении PUSH, POP, если при инициализации SPHL HL=0FFFFh? Проверить на эмуляторе. 7. Сколько внешних устройств можно адресовать?

8. Сформируйте точку красного цвета в середине экрана монитора эмулятора.
9. Где располагаются второй и третий байт команды?
10. Какие способы адресации используются при выполнении команд пересылки?
11. Чем отличается текстовый режим работы монитора от графического режима?
12. В каких кодах выполняются арифметические команды? Как формируются эти коды?
13. Как машина различает: числа со знаком или нет? положительное число или отрицательное?
14. Как выполняются арифметические и логические операции с двумя байтами (двухбайтными числами)?
15. Какие виды циклических операций существуют? Чем они различаются?
16. Что такое признаки (флаги) ЦП?
17. Зачем в системе команд операция сдвига через признак С?
18. Разработайте алгоритм и программу сложения двух символов. На монитор вывести: Символ + Символ = Результат.
19. Каков формат выдачи данных на монитор? Разработайте подпрограмму вывода данных на монитор.
20. Как в машине выполняется управляющая конструкция типа "развилка"? Какие данные и как обрабатывает машина при ее реализации?
21. Как выполняется команда RET? Состояние каких регистров при этом изменяется?
22. Зачем организуются подпрограммы?
23. Какова технология разработки программ?
24. Состояние каких регистров изменяется при выполнении команд CPI и CMP?
25. Почему вывод сообщения оформляется в формате подпрограммы?
26. Как выполняется команда CALL? Состояние каких регистров изменяется?
27. Как конструируются циклы?
28. В каком формате вводятся данные с клавиатуры?
29. Какие коды имеют числовые символы?
30. Как из кода символа получить число, ему соответствующее?
31. Как из одноразрядного числа получить код, ему соответствующий?
32. Сформулируйте алгоритм преобразования кода символов многоразрядного числа в двоичное число?
33. Каковы задачи компоновщика?
34. Какова структура файла типа .COM?
35. Какие данные хранятся в файле типа .PRN?
36. Что находится в файле типа .SYM?
37. Каковы задачи компилятора?

38. Что такое "символическое имя"? Как оно записывается в исходном тексте и чему соответствует в исполняемом файле?
39. Каковы требования к текстовому редактору системы программирования?
40. Какие задачи выполняет символьный отладчик?
41. Что такое "точка останова"?
42. Какова методика отладки программы?
43. Можно ли отладить программу без символьного отладчика?
44. В файле с каким расширением предполагается настройка адресов? В чем она заключается?
45. Чем отличается строковый и числовой формат данных?
46. Как из кода символа получить число, ему соответствующее?
47. Сформулируйте алгоритм преобразования кода символов многозначного числа в двоичное число?
48. Что такое виртуальные символические, виртуальные и физические адреса?
49. Какие директивы символьного отладчика используются при отладке программы?
50. Как проверяется правильность программ?
51. Как отладить программу без символьного отладчика?
52. В чем заключается работа загрузчика?
53. Разработайте алгоритм вывода числа в шестнадцатеричном формате.
54. Разработайте алгоритм вывода числа в двоичном формате.
55. Сформулируйте назначение параллельного интерфейса.
56. Какова программная модель параллельного интерфейса?
57. Каково назначение типовых регистров параллельного интерфейса?
58. Нарисуйте схему подключения микросхемы параллельного интерфейса к шинам МП.
59. Какие существуют режимы работы параллельного интерфейса?
60. Нарисуйте схему подключения внешних устройств к параллельному интерфейсу.
61. Разработайте программу инициализации параллельного интерфейса для основного режима.
62. Разработайте программу инициализации параллельного интерфейса для стробируемого однонаправленного режима.
63. Разработайте программу инициализации параллельного интерфейса для стробируемого двунаправленного режима.
64. Сформулируйте назначение последовательного интерфейса.
65. Какие коды называются последовательными и параллельными?
66. Какова программная модель последовательного интерфейса?
67. Каково назначение типовых регистров последовательного интерфейса?
68. Нарисуйте схему подключения микросхемы последовательного интерфейса к шинам МП.

69. Какие существуют режимы работы последовательного интерфейса?
70. Нарисуйте схему подключения внешних устройств к последовательному интерфейсу.
71. Разработайте программу инициализации последовательного интерфейса для работы в асинхронном режиме.
72. Разработайте программу инициализации последовательного интерфейса для работы в режиме внутренней синхронизации.
73. Разработайте программу инициализации последовательного интерфейса для работы в режиме внешней синхронизации.

3. Специализированные контроллеры

3.1. Классификация контроллеров. Характеристика семейств микроконтроллеров. Архитектура и схемотехника контроллеров. Особенности системы команд. Структурная схема МК и ее работа. Программная модель МК. Организация периферии МК. Применение МК

3.2. Этапы проектирования ЦУ. Сущность процесса проектирования. Классификация подходов к проектированию. Этапы проектирования цифровых устройств. Этапы проектирования, поддающиеся автоматизации. Средства автоматизированного проектирования. Модели и языки описания устройств Средства автоматизированного проектирования. Модели и языки описания устройств

Вопросы и упражнения для самоконтроля:

1. Чем отличается "структура" от "архитектуры"?
2. Какие внешние устройства могут быть в составе микроконтроллера?
3. Какие типы памяти есть в микроконтроллере и каково их назначение?
4. Каково назначение интерфейса SPI?
5. Для чего создается модуль загрузки ISP?
6. Как используются регистры общего назначения?
7. Каково назначение аналогового компаратора?
8. Каково назначение последовательного приемопередатчика?
9. Каково назначение последовательного интерфейса TWI?
10. Что входит в состав программной модели микроконтроллера?
11. Какие системы счисления используются при вводе программ в AVR? Как перевести данные из одной системы счисления в другую?
12. Какова программная модель микроконтроллера ATtiny2313?
13. Какова организация блоков памяти микроконтроллера?
14. Каково назначение стека в микроконтроллере? Где стек организуется?
15. Как адресуются регистры общего назначения микроконтроллера? Сколько их?
16. Как адресуются внешние устройства микроконтроллера?
17. Какие внешние устройства имеются у микроконтроллера ATtiny2313?
18. Каким образом компилятору задают область размещения кодов: в ОЗУ, постоянной памяти данных, памяти программ?
19. Какова разрядность счетчика команд? Почему такая?
20. Какие коды используются при выполнении арифметических команд?

21. Каковы операции получения дополнительного кода?
22. Приведите примеры команд арифметических, поясните смысл выполняемых преобразований.
23. Приведите примеры команд логических, поясните смысл выполняемых преобразований.
24. Приведите примеры команд сдвига, поясните смысл выполняемых преобразований.
25. Приведите примеры команд арифметического и логического сдвига, поясните различие в выполняемых преобразованиях.
26. Для какой цели используется команда сдвига через признак переноса.
27. Для какой цели используется команда сложения с признаком переноса?
28. Приведите примеры команд операций с битами, поясните смысл выполняемых преобразований.
29. Какие операторы применяются в ассемблере AVR?
30. Какие команды позволяют выполнить обращение к подпрограмме?
31. В чем отличие команд вызова подпрограмм?
32. Как инициализируется стек при запуске МК?
33. Какие группы команд условного перехода имеются в МК?
34. Как выполняется команда CALL?
35. Как выполняется команда RET?
36. Зачем нужна команда сброса сторожевого таймера WDR?
37. Как выполняются команды сравнения?
38. Каким образом передаются параметры подпрограмм?
39. Что такое расширенный переход?

4. Курсовое проектирование. Разработка микропроцессорной системы

3. ЛАБОРАТОРНЫЕ РАБОТЫ И ПРАКТИЧЕСКИЕ ЗАНЯТИЯ

Лабораторная работа №1. Исследование функционирования МП при выполнении команд пересылки. Исследование работы МП и состояния логической модели МП при выполнении команд однобайтных и двухбайтных пересылок, обмена байтами, ввода и вывода

Лабораторная работа №2. Исследование функционирования МП при выполнении арифметических и логических команд. Исследование работы МП и состояния логической модели МП при выполнении команд арифметических, логических, сдвига, преобразователя кода в число и вывода на монитор

Лабораторная работа №3. Исследование функционирования МП при выполнении команд передачи управления. Исследование работы МП и состояния логической модели МП при выполнении команд условного и безусловного перехода, вызова и возвращения из подпрограмм, разработка подпрограммы

Лабораторная работа №4. Анализ функционирования МП при выполнении программ. Исследование работы МП и состояния логической модели МП при загрузке и выполнении программы
Лабораторная работа №5. Анализ режимов программирования параллельного интерфейса. Анализ режимов работы, программной модели. Разработка программы управления
Лабораторная работа №6. Анализ режимов программирования последовательного интерфейса. Анализ режимов работы, программной модели. Разработка программы управления
Лабораторная работа №7. Исследование МК при выполнении команд пересылки, арифметических и логических команд. Среда AVR Studio. Изучение работы МК и состояния логической модели МК при выполнении команд пересылок, обмена байтами, ввода и вывода, арифметико-логических команд
Лабораторная работа №8. Исследование МК при выполнении команд передачи управление и выполнении программ. Среда AVR Studio. Изучение работы МК и состояния логической модели МК при выполнении команд условного и безусловного перехода, вызова и возвращения из подпрограмм, разработка подпрограммы
Практическое занятие №1. Программирование ПЗУ. Программирование ПЗУ для реализации настраиваемых дешифраторов, сумматоров, логических преобразователей
Практическое занятие №2. Использование ПЗУ в динамических устройствах. Использование ПЗУ в генераторах сигналов, устройствах отображения информации
Практическое занятие №3. Разработка модуля памяти. Определение рабочих адресов, требований к параметрам, построение модуля памяти, разработка адресного дешифратора
Практическое занятие №4. Изучение логической структуры микроконтроллера. Изучение структурной схемы, программной модели, способов включения микроконтроллера в состав устройства

4. СПИСОК ЛИТЕРАТУРЫ

1. Угрюмов Е.П. Цифровая схемотехника. Спб: БХВ-Петербург, 2012, 800с.
2. Гуров В.В. Микропроцессорные системы: Учебник. - М.: НИЦ ИНФРА-М, 2016. -336с.
3. Юров В.И. Ассемблер. Учебник для вузов. СПб.: Питер. 2003. -624с.
4. Микропроцессорный комплект К1810: Структура, программирование, применение: Справочная книга / Ю.М. Казаринов, В.Н. Номоконов, ГС. Подклетнов, Ф.В. Филиппов; Под ред. Ю.М. Казаринова. - М.; Высшая школа, 1990. - 269 с.

5. Дао Л. Программирование микропроцессора 8088. - М.: Мир, 1988. 357с.
6. ГОСТ 2.102-95 Основные требования к текстовым документам.
7. Положение о контрольных и курсовых работах (проектах) СКМ-П-01/2-51-17. СКФ МТУСИ, 2017.
http://www.skf-mtusi.ru/files/info/docs/kursovykh_kontrolnykh_rabotakh_01.09.17.pdf
8. Жуковский А.Г., Манин А.А. Руководство по подготовке курсовых работ (проектов) и выпускных квалификационных работ. Ростов-га-Дону: СКФ МТУСИ, 2019. - 61с.
9. Микропроцессоры и микропроцессорные комплекты интегральных микросхем: Справочник: В 2 т. / Н.Н. Аверьянов и др.; Под ред. В.А. Шахнова. - М.: Радио и связь, 1988.
10. Лю Ю-Чжен, Гибсон Г. Микропроцессоры семейства 8086/8088. Архитектура, программирование и проектирование микропроцессорных систем. - М.: Радио и связь, 1987. - 512с.
11. Лебедев О.Н. Микросхемы памяти и их применение. - М.: Радио и связь, 1990.-160 с.
12. Чикалов А.Н. Микропроцессоры. Периферийные устройства. Методические указания к лабораторным и практическим занятиям. Ростов-на-Дону: Северо-Кавказский филиал МТУСИ, 2019.- 37 с.
13. Интегральные микросхемы запоминающих устройств
<https://www.integral.by/ru/products/integrated-circuits/storage-devices>
14. Режимы адресации <http://e-al.narod.ru/cpress05/cpu2.htm>
15. Литература для студентов <https://studfiles.net>
16. Справочник по ИМС
<https://www.qrz.ru/reference/kozak/mem/mem00.shtml>

5. ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ

Разработать простейшую микропроцессорную систему (МПС) на базе процессоров семейства Intel x86, обеспечивающую управление заданными периферийными программируемыми микросхемами. Составить программу инициализации периферийных программируемых микросхем на языке Ассемблера. Описания БИС даны в литературе [3, 4].

Вариант индивидуального задания выбирается по последним двум цифрам зачетной книжки (табл.1). Для определения варианта в диапазоне значений 31-60 предварительно необходимо вычесть 30, в диапазоне 61-90 предварительно вычесть 60, в диапазоне 91-99 - вычесть 90.

Две последние цифры зачетной книжки определяют номер варианта для выбора ИМС ОЗУ.

Две последние цифры зачетной книжки, прочитанные справа налево, определяют номер варианта для выбора ИМС ПЗУ и типа и адреса внешнего устройства (устройства ввода-вывода, блока внешнего устройства).

Например, две последние цифры - 24. Следовательно, ИМС ОЗУ выбираются по строке №24, а тип ИМС ПЗУ и тип внешнего устройства выбирается по номеру $42 - 30 = 12$.

Курсовой проект оформляется в соответствии с правилами, действующими в СКФ МТУСИ.

Материалы проекта должны содержать:

1. Данные индивидуального задания;
2. Схему электрическую структурную микропроцессорной системы в соответствии с заданием, назначением элементов схемы;
3. Блок центрального процессора с изложением принципов его работы, назначением элементов схемы, с указанием разрядности шины данных и адреса, назначением сигналов шины управления. Краткое описание работы ЦП при обмене данными;
4. Программную модель ЦП и ее описание;
5. Условное графическое обозначение используемых ИМС памяти в соответствии с заданием с указанием назначения выводов и структуры накопителя;
6. Аналитические расчеты блоков дешифрации адресов схем памяти и двух внешних устройств, схемы электрические функциональные этих блоков;
7. Обоснования и схемы электрические функциональные блоков памяти;
8. Условное графическое обозначение, назначение выводов схемы внешнего интерфейса в соответствии с вариантом. Структуру управляющего слова, описание режима работы программируемой микросхемы внешнего интерфейса, коды задания режимов. Один из возможных режимов работы программируемой микросхемы студент выбирает самостоятельно, но вариант должен отличаться от приведенного в методическом пособии в качестве примера;
9. Исходный текст программы инициализации микропроцессорной системы с комментариями. Указать объекты, подлежащие инициализации, адреса размещения кода инициализации. Для устройства ввода-вывода в соответствии с вариантом исходный текст должен соответствовать описанному режиму;
10. Список использованной литературы, на которую в тексте пояснительной записки сделаны ссылки.

Варианты индивидуальных заданий

№ вар.	Тип ЦП	Нач. адр. ОЗУ	Объем ОЗУ, Кб	БИС ОЗУ	Объем ПЗУ, Кб	БИС ПЗУ	Нач. адр. ввода-вывода	БВИ
1	8086	0	16	537PY8	256	573PФ7	20h	1
2	8086	2000h	8	537PY8	128	573PФ7	40h	2
3	8086	4000h	16	537PY8	256	573PФ7	60h	3
4	8086	6000h	8	537PY8	128	573PФ7	80h	1
5	8086	8000h	16	537PY8	256	573PФ7	0A0h	2
S	8086	10000h	64	537PY18	64	573PФ6	0C0h	3
7	8086	18000h	32	537PY18	32	573PФ6	0E0h	1
8	8086	20000h	64	537PY18	64	573PФ6	100h	2
9	8086	28000h	32	537PY18	32	573PФ6	120h	3
10	8086	30000h	64	537PY18	64	573PФ6	140h	1
11	8086	40000h	128	537PY9	16	573PФ5	160h	2
12	8086	50000h	64	537PY9	8	573PФ5	180h	3
13	8086	80000h	128	537PY9	16	573PФ5	1A0h	1
14	8086	90000h	64	537PY9	8	573PФ5	1C0h	2
15	8086	0A0000h	128	537PY9	16	573PФ5	1E0h	3
16	8086	40000h	256	537PY13	16	573PФ2	200h	1
17	8086	60000h	128	537PY13	8	573PФ2	220h	2
18	8086	80000h	256	537PY13	16	573PФ2	240h	3
19	8086	0A0000h	128	537PY13	8	573PФ2	260h	1
20	8086	40000h	256	537PY13	16	573PФ2	280h	2
21	8086	0	512	537PY7	64	573PФ4	2A0h	3
22	8086	80000h	256	537PY7	32	573PФ4	2C0h	1
23	8086	0	512	537PY7	64	573PФ4	2E0h	2
24	8086	40000h	256	537PY7	32	573PФ4	300h	3
25	8086	0	512	537PY7	64	573PФ4	320h	1
26	8086	0C000h	16	537PY10	256	573PФ7	340h	2
27	8086	0E000h	8	537PY10	128	573PФ7	360h	3
28	8086	10000h	16	537PY10	256	573PФ7	380h	1
29	8086	12000h	8	537PY10	128	573PФ7	3A0h	2
30	8086	14000h	16	537PY10	256	573PФ7	3C0h	3

Код типа периферийной БИС в блоке внешнего интерфейса (БВИ):

- 1 - i8251,
- 2 - i8253,
- 3 - i8255.

6. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ КУРСОВОГО ПРОЕКТА

Работа должна быть оформлена в соответствии с требованиями Положения о контрольных и курсовых работах (проектах) СКФ МГУСИ и положений ГОСТ 2.102-95 Основные требования к текстовым документам.

В соответствии с Положением (Раздел 2) пояснительная записка должна содержать 30-40 стр. машинописного текста размером 14 с полуторным интервалом с выравниванием по ширине.

Каждый лист пояснительной записки должен иметь рамку и чертежный штамп.

На защиту студент представляет пояснительную записку на бумажном и электронном носителе, файл разработанной базы данных. Пояснительная записка должна быть размещена в портфолио студента.

В процессе защиты автору предстоит обосновать принятые решения и продемонстрировать работающую базу данных. Автор для доклада может воспользоваться подготовленной презентацией.

Пояснительная записка должна содержать следующие элементы:

1. Титульный лист (приложение А Положения);
2. Задание (приложение Б Положения);
3. СОДЕРЖАНИЕ;
4. ВВЕДЕНИЕ (содержит описание состояния проблемы, актуальность цели и задачи проекта);
5. Разделы и подразделы;
6. ЗАКЛЮЧЕНИЕ (включает выводы и рекомендации);
7. СПИСОК ЛИТЕРАТУРЫ;
8. ПЕРЕЧЕНЬ СОКРАЩЕНИЙ (при необходимости);
9. Приложения (при необходимости).

Записка должна быть подписана автором с указанием даты. Листы записки должны быть скреплены в виде брошюры.

7. ПРИМЕР ОФОРМЛЕНИЯ КУРСОВОГО ПРОЕКТА

Федеральное агентство связи
Северо-Кавказский филиал ордена Трудового Красного Знамени
федерального государственного бюджетного образовательного учреждения
высшего образования
«Московский технический университет связи и информатики»

Кафедра Информатики и вычислительной техники

Допустить к защите
«__» _____ 20__ г.

(подпись, Фамилия, И.О.)

Курсовой проект

по дисциплине: Микропроцессорные системы

по теме: Микропроцессорная система

Студент Капустин А.М.

Направление Информатика и ВТ

Группа ВМ-41 Курс 4

Шифр ст. билета 13227

Вариант 27

Дата «__» _____ 20__ г.

Подпись _____

Ростов-на-Дону

2019

СОДЕРЖАНИЕ

Введение	21
1. Схема электрическая структурная МПС	22
2. Блок центрального процессора	
3. Программная модель ЦП и ее описание	
4. Условное графическое обозначение ИМС памяти и структура накопителей	
5. Аналитические расчеты блока дешифрации адресов	
6. Разработка схемы блоков дешифрации адресов	
7. Разработка блока памяти МПС	
8. Анализ работы УВВ и проектирование программы инициализации	
Заключение	
Список литературы	

					<i>СКФ МТУСИ 09.03.01 13227 ПЗ</i>			
Изм	Лист	№ документа	Подп.	Дата				
Разработал		Капустин А.М.			Микропроцессорная система	Лит.	Лист	Листов
Проверил		Капустин А.М.					4	21
Рецензент		Молчанов А.Н.				<i>каф.ИВТ гр. ВМ-41</i>		
Н. контр								
Утвердил		Орлов С.В.						

ВВЕДЕНИЕ

Успешное решение задач управления технологическими процессами и средствами связи предполагает применение и совершенствование средств вычислительной техники в контуре управления. Основой таких средств являются промышленно выпускаемые микропроцессоры и контроллеры. Микропроцессор с 16-разрядной шиной данных является одним из типовых средств для подобного применения. На базе процессора 8086 выпускаются одноплатные управляющие микроЭВМ и микроконтроллеры, универсальные микроЭВМ, высокопроизводительные микропроцессорные системы.

Архитектура процессора 8086 легла в основу всех дальнейших процессоров Intel, осознавшей удобство и универсальность его концепции. Именно благодаря акценту на совместимость с предшествующими платформами с точки зрения программного обеспечения архитектура 8086 стала ключевой и послужила основой для большинства последующих процессоров. Каждый следующий процессор строился на фундаменте предыдущего, обрастая новыми технологиями, инструкциями и блоками, но по своей сути немногим отличался от 8086.

В настоящей работе выполнена разработка микропроцессорной системы на базе микропроцессора Intel 8086, ее блока памяти и средств управления типовыми периферийными схемами, предложена программа инициализации устройств ввода-вывода.

Для проекта по варианту №27 заданы следующие исходные данные:

№ варианта	Тип ЦП	Нач. адрес ОЗУ	Объем ОЗУ, Кб	БИС ОЗУ	Объем ПЗУ, Кб	БИС ПЗУ	Нач. адр. ввода-вывода	УВВ
27	8086	8000h	16	537РУ8	256	573РФ7	0A0h	1

					СКФ МТУСИ 09.03.01 13227 ПЗ			Лист
								22
Изм	Лист	№докум.	Подпись	Дата				

1 Схема электрическая структурная МПС

Структурная схема МПС предполагает шинную организацию. Это позволяет достаточно просто подключить внешние устройства, управлять ими и обеспечить в дальнейшем наращивание состава периферийных устройств.

Схему можно разделить на блок центрального процессора, включающую тактовый генератор, центральный процессор и системный интерфейс; блок дешифрации адресов; блока памяти и блока внешних интерфейсных схем. Структурная схема МПС представлена на рисунке 1.1.

Тип центрального процессора (ЦП) определяет разрядность шины адреса (20 линий) и шины данных (16 разрядов), системной шины, объем адресного пространства для адресации памяти и портов ввода-вывода, организацию (разрядность) памяти и портов ввода-вывода, принципы построения системного интерфейса и набор управляющих сигналов системной шины. Характеристики ЦП 8086 в сравнении с аналогичными процессорами представлены в таблице 1.1.

Системный интерфейс образуется системным контроллером и буферными схемами. Системный контроллер формирует управляющие сигналы МПС. Он вырабатывает сигналы чтения/записи памяти и УВВ, дешифрируя информацию, выдаваемую ему процессором по шине состояния о типе шинного цикла, выполняемого в данный момент процессором.

Буферные схемы системного интерфейса обеспечивают формирование шин адреса и данных, временное хранение этих сигналов в целях требуемой синхронизации и увеличение электрической мощности. Последняя функция обеспечивает подключение нескольких входов подключаемых микросхем к выводам процессора.

					<i>СКФ МТУСИ 09.03.01 13227 ПЗ</i>	<i>Лист</i>
						23
<i>Изм</i>	<i>Лист</i>	<i>№докум.</i>	<i>Подпись</i>	<i>Дата</i>		

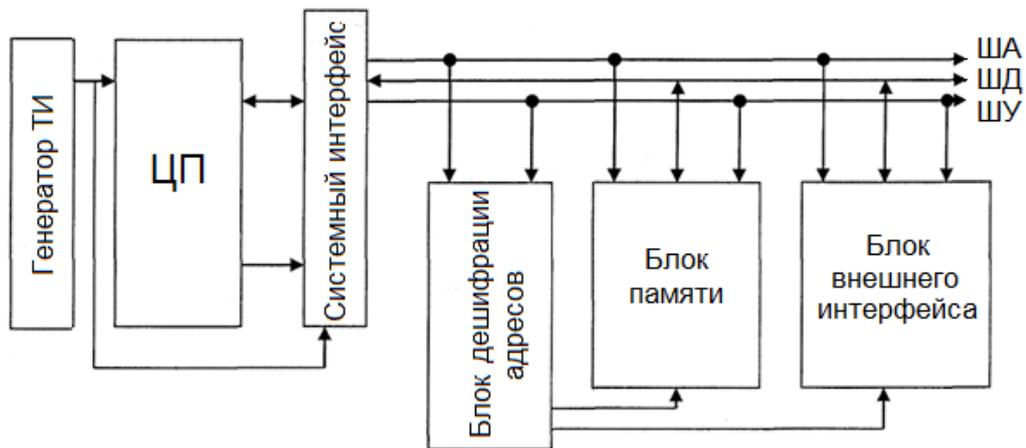


Рисунок 1.1 - Схема электрическая структурная МПС

Таблица 1.1 - Основные характеристики процессоров

Тип	Число сигналов адреса	Число сигналов данных	Макс, объем памяти	Макс, число портов	БИС синхронизатора	БИС системного контроллера
8086	20	16	1Mb	64Kb	8284	8288
8088	20	8	1Mb	64Kb	8284	8288
80C186	20	16	1Mb	64Kb	Встроен	8288
80C188	20	8	1Mb	64Kb	Встроен	8288
80286	24	16	16Mb	64Kb	82284	82288
1810BM86	20	16	1Mb	64Kb	1810ГФ84	1810ВГ88

.....
и т.д.

.....

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ВЫПОЛНЕНИЯ ЗАДАНИЯ

8.1. БЛОК ЦЕНТРАЛЬНОГО ПРОЦЕССОРА

Блок центрального процессора проектируемой МПС включает в себя: синхрогенератор, БИС центрального процессора и микросхемы системного интерфейса, который, в свою очередь, состоит из БИС системного контроллера, формирующего сигналы системной шины управления (ШУ), микросхем буферных регистров-защелок адреса, формирующих сигналы системной шины адреса (ША), и микросхем повторителей с тремя состояниями выходов, формирующих сигналы системной шины данных (ШД). Схема блока ЦП представлена на рис.8.1.

В процессоре используется 16-разрядная мультиплексированная двунаправленная единая шина адреса/данных. Старшие адресные линии A16 - A19 формируются самостоятельно. Для использования линий адреса и данных их обязательно необходимо разделить с помощью внешних схем. Адресные линии (16 младших A0-A15 и A16-A19) выдаются в первом такте цикла и должны быть сохранены в течение всего цикла на регистрах-защелках по заднему фронту синхросигнала ALE. В качестве регистров используются 8-разрядные микросхемы с трехстабильными выходными буферами и высокой нагрузочной способностью K1810ИР82.

Вход \overline{OE} буферных регистров разрешает выдачу данных на выходы DO0...DO7. В однопроцессорных системах без прямого доступа к памяти этот контакт заземляется.

Сигнал ВНЕ=0 формируется одновременно с адресными сигналами и его также необходимо зафиксировать в защелке. Сигнал используется как еще один сигнал системной шины управления, сопровождающий передачу 8-битовых данных по старшей половине ШД D8-D15 (его иногда называют сигналом "выборки старшего байта", т.е. байта по нечетному адресу). Он своим нулевым значением сопровождает выдачу либо слова (2 байта), либо старшего байта данных. Этот сигнал используется в блоке памяти как дополнительный адресный вход, определяющий доступ к старшему банку памяти либо к внешнему устройству с байтной организацией, подключенному к старшей половине мультиплексированной шины. Адресные комбинации для передачи различных типов данных представлены в табл.8.1.

Таблица 8.1

Адресные комбинации для передачи байтов и слов
по 16-разрядной шине данных МПС

ВНЕ	A0	Разрядность передаваемых данных
0	0	Все слово, оба байта по линиям D ₀ ...D ₁₅
0	1	Старший байт по линиям D8...D15 (нечетный адрес)
1	0	Младший байт по линиям D0... D7 (четный адрес)
1	1	Нет передачи

К внешним устройствам ЦП обращается по 16-разрядной шине адреса A0 - A15. Суммарная емкость портов 64К.

Во второй половине цикла по линиям AD0 - AD15 передаются сигналы данных D0 - D15, которые сопровождаются стробом DEN. Задачу буферизации данных выполняют двунаправленные 8-битовые шинные формирователи K1810BA86, которые усиливают сигналы системной шины.

Для вариантов МПС с микросхемой синхрогенератора БИС K1810ГФ84 она подключается к ЦП по стандартной схеме, вариант которой показан на рис.8.1.

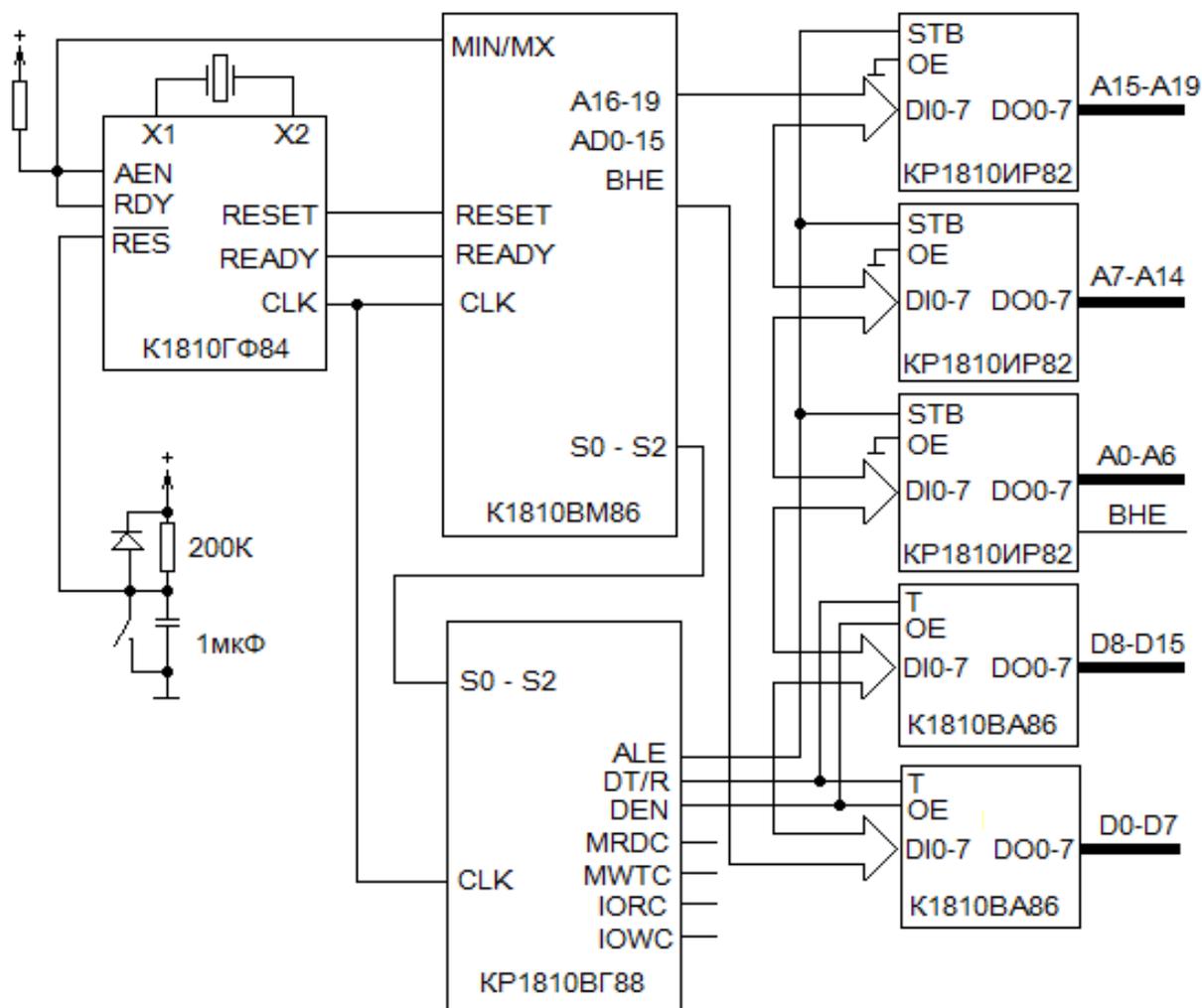


Рис.8.1. Блок центрального процессора

В таблице 8.2 поясняется назначение выводов БИС синхрогенератора (нумерация выводов для аналогичных сигналов 8284 и 82284 не совпадает, обозначение ПУ - периферийные устройства).

Назначение и подключение выводов БИС синхрогенератора К1810ГФ84

Сигнал	Направление	Назначение	Куда подключается в учебной МПС или что подается
X1	Вход	Для подключения кварцевого резонатора	К выводу кварцевого резонатора
X2	Выход	Тоже	Тоже
TANK	Вход	Для подключения LC-фильтра	0
F/C	Вход	Выбор режима с внешним генератором	0
EFI	Вход	Сигнал от внешнего генератора	0
CSYN	Вход	Синхронизация нескольких БИС синхрогенераторов	0
RES	Вход	Сигнал сброса при включении питания или от кнопки "Reset"	К RC-цепочке и кнопке "Reset"
RDY1	Вход	Сигнал готовности ПУ к обмену данными (1)	1
RDY2	Вход	Сигнал готовности ПУ к обмену данными (2)	1
AEN1	Вход	Сигнал разрешения формирования RDY1	1
AEN2	Вход	Сигнал разрешения формирования RDY2	1
CLK	Выход	Основной системный синхросигнал для ЦП	К входу CLK ЦП и входу CLK БИС системного контроллера
RESET	Выход	Сигнал системного сброса	К входу RESET ЦП и входам RESET периферийных БИС (если необходимо)
READY	Выход	Сигнал готовности ПУ к обмену данными	К входу READY ЦП и входу READY системного контроллера
PCLK	Выход	Синхросигнал для периферийных БИС $F_{PCLK}=F_{CLK}/2$	К входам CLK периферийных БИС (если необходимо)
OSC	Выход	Синхросигнал с частотой кварцевого резонатора $F_{OSC}=3 F_{CLK}$	Не используется

В таблице:

0 - постоянно подан уровень логического нуля, т.е. вход заземлен;

1 - постоянно подан уровень логической единицы, т.е. вход подключен к проводнику питания +5В через токоограничивающий резистор сопротивлением примерно 1кОм.

БИС системного контроллера K1810ВГ88 также подключается к ЦП и синхрогенератору по стандартной схеме (см.рис.8.1). В таблице 8.3 поясняется назначение выводов БИС системного контроллера.

Системный контроллер управляет обменом данными, выдавая сигналы на шинные формирователи (DT/R, DEN) регистры, фиксаторы адреса (ALE), устройства ввода-вывода (IOWC, IORC) и память (MRDC, MWTC). Выработка сигналов происходит по синхросигналам генератора CLK на основании сигналов состояния процессора S0, S1, S2.

Таблица 8.3

Назначение и подключение выводов БИС K1810ВГ88

Сигнал	Направление	Назначение	Куда подключается в учебной МПС
SO	Вход	Сигнал состояния ЦП. Определяет тип выполняемой передачи	К выходу SOЦП
SI	Вход	Тоже	К выходу S1 ЦП
S2 (M/IO)	Вход	Тоже	К выходу S2 ЦП (к выходу M/IOЦП)
CLK	Вход	Системный синхросигнал	К выходу CLK синхрогенератора
AEN	Вход	Управление выдачей сигналов на ШУ	0
CEN	Вход	То же, используется при каскадировании контроллеров (вместе с сигналами DEN и PD)	1
IOB	Вход	Выбор режима работы	0
MRDC	Выход	Сигнал чтения из памяти	Используется в блоке дешифрации адресов
MWTC	Выход	Сигнал записи в память	Ко входам W/R БИС ОЗУ в блоке памяти. Используется также в блоке дешифрации адресов
AMWC	Выход	Опережающий сигнал записи в память	Не используется
IORC	Выход	Сигнал чтения в процессор из порта ввода-вывода по ШД: порт выставляет данные на	Ко входам RD периферийных БИС. Используется также в блоке дешифрации

		ШД	адресов
IOWC	Выход	Сигнал записи из процессора в порт ввода-вывода	Ко входам WR периферийных БИС. Используется также в блоке дешифрации адресов
AIOWC	Выход	Опережающий сигнал записи в порт ввода-вывода	Не используется
DT/R	Выход	Направление передачи данных: 0 - прием, входными линиями являются DO0...DO7; 1 - передача, входными линиями являются DI0...DI7	Ко входам T микросхем повторителей на шине данных
DEN	Выход	Разрешение передачи данных (0 - разрешает)	Ко входам OE микросхем повторителей на шине данных (через инвертор)
MC/PD	Выход	Считывание номера ведомого контроллера прерываний	Не используется
ALE	Выход	Строб регистров-защелок адреса для фиксации битов	Ко входам строга записи микросхем регистров-защелок адреса (C или STB)
READY (только для 82286)	Вход	Сигнал готовности ПУ к обмену данными	К выходу READY БИС синхрогенератора 82284

К входу RES МП подключается RC-цепочка, которая формирует сигнал сброса автоматически при включении питания либо с помощью кнопки (см. рис.8.1).

Работа процессора осуществляется совместно с системным контроллером. Это так называемый максимальный режим работы шины адреса/данным МП. Диаграммы работы МП в максимальном режиме представлены на рис.8.2 [2].

Во время тактов T1 и половины T2 выставляется адрес, сигнал ВНЕ и они фиксируются стробом ALE на регистрах-защелках. На регистрах адрес сохраняется до окончания цикла.

Во время T2 и T3 выставляется сигнал направления передачи данных DT/R, формируются сигналы чтения из памяти и УВВ (MRDC, IORC) или записи в память и УВВ (MWTC, IOWC).

Во время T4 снимаются активные сигналы и переводится в третье состояние выходы шины адреса/данных. На этом цикл завершается.

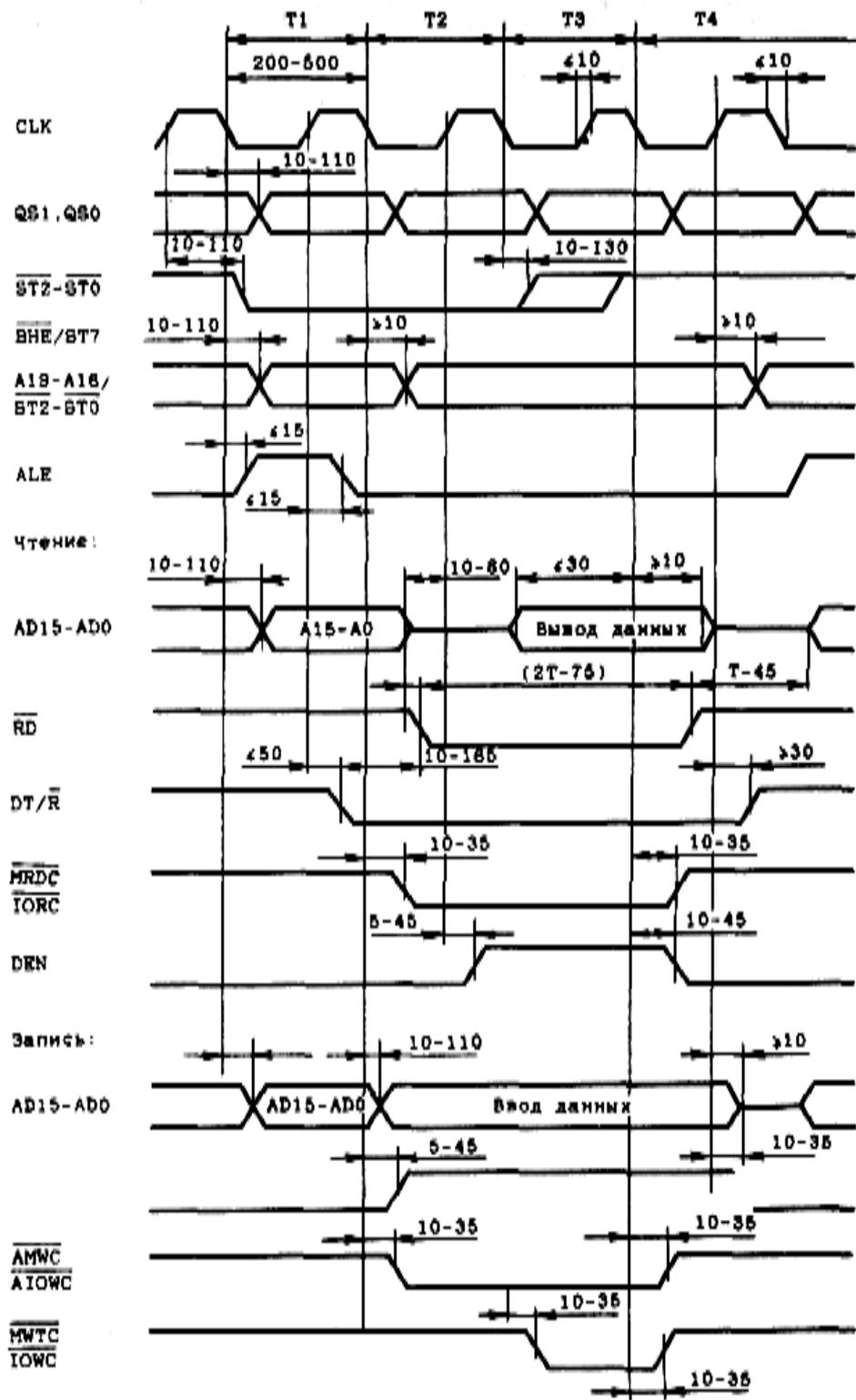


Рис.8.2. Временные диаграммы работы МП 8086 в максимальном режиме

8.2. БЛОК ДЕШИФРАЦИИ АДРЕСОВ

Для разработки блока дешифрации адресов следует учесть структуру накопителей. В МПС необходимо обеспечить обмен как двухбайтными словами, так и отдельными байтами. Поэтому блок памяти следует организовать

в виде двух 8-разрядных банков. Младший банк, хранящий данные D0-D7, должен иметь четные адреса обращения. Старший банк, хранящий данные D8-D15, должен иметь нечетные адреса обращения. Формировать тип передаваемых данных необходимо комбинациями сигналов адресного A0 и разрешения передачи старшего байта ВНЕ так, как указано в таблице 8.1. В соответствии с этой таблицей, 2-байтное слово должно располагаться начиная с четного адреса (A0=0). В памяти в младшем адресе сохраняется младший байд 2-х байтного числа, а в старшем адресе - старший байт 2-х байтного числа. Адресом всего числа является адрес его самого младшего байта.

Выравнивания многобайтных чисел по четным адресам не требуется. Если число начинается с четного адреса, то обмен осуществляется за одну операцию передачи по шине сразу 16 разрядов данных. Для многоразрядных чисел, начинающихся с нечетного адреса, обмен будет выполнен за два обращения к памяти, что снижает производительность процессора.

Сигналы A0 и ВНЕ должны подаваться на входы выбора соответственно младшего и старшего банка данных. Если потребуется страничная организация банков памяти, то удобно выбрать микросхемы памяти с двумя входами выбора для дополнительной селекции выбираемого банка данных. Такой вариант получится самым оптимальным. Структурная схема организации страницы памяти представлена на рис.8.3. Очевидно, что одна страница должна образовываться минимум двумя ИМС с байтной организацией.

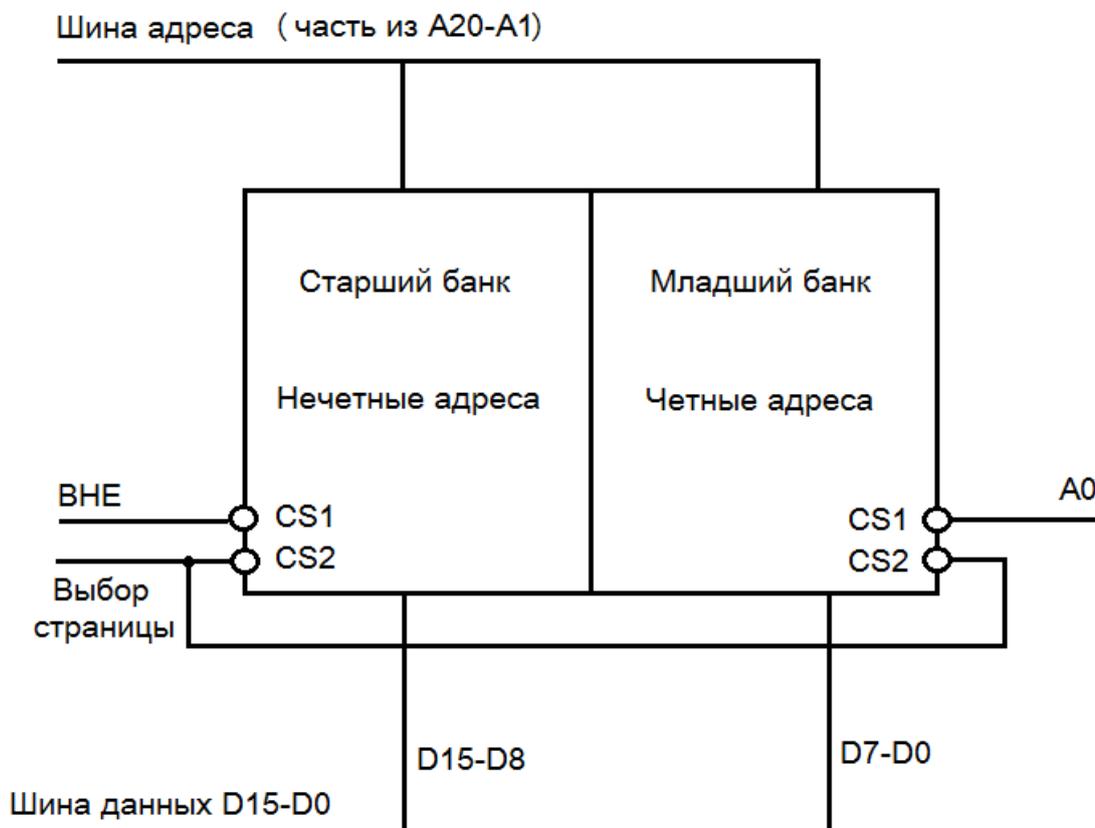


Рис.8.3. Структура страницы блока памяти

Блок дешифрации адресов проектируемой МПС состоит из трех частей: схемы формирования сигналов выбор кристалла (CS) для страниц (микросхем) ОЗУ блока памяти, схемы формирования сигналов CS для страниц (микросхем) ПЗУ блока памяти и схемы формирования сигналов CS для программируемых периферийных БИС блока внешнего интерфейса.

Исходными данными для разработки блока дешифрации адресов являются указанные в индивидуальном задании: начальный адрес ОЗУ, объем ОЗУ, тип микросхем ОЗУ, объем ПЗУ, тип микросхем ПЗУ, начальный адрес области портов ввода-вывода, а также разрядность системной шины данных, которая, в свою очередь, определяется заданным типом центрального процессора МПС. На начальном этапе разработки необходимо:

- 1) найти число БИС ОЗУ в блоке памяти;
- 2) найти число БИС ПЗУ в блоке памяти;
- 3) определить начальный адрес ПЗУ;
- 4) определить конечный адрес ОЗУ;
- 5) определить номера разрядов шины адреса для формирования общего сигнала выборки страниц ОЗУ ($SELECT_{RAM}$) и общего сигнала выборки страниц ПЗУ ($SELECT_{ROM}$);
- 6) вычислить начальный адрес ввода-вывода для второй периферийной БИС (для первой он задан в задании).

Процесс разработки блока дешифрации адресов проиллюстрирован на следующем примере с использованием гипотетических типов ИМС.

Дано:

- 1) начальный адрес ОЗУ-38000H,
- 2) объем ОЗУ-32Kb,
- 3) БИС ОЗУ статического типа K537PУХХ с организацией 8Kx1 (объемом 8Kбит),
- 4) объем ПЗУ-8Kb,
- 5) БИС ПЗУ типа K573PФХ с организацией 2Kx8 (объемом 2Kb),
- 6) начальный адрес ввода-вывода для первой периферийной БИС таймера - 270H,
- 7) ЦП типа 8086 с 16-разрядной системной шиной данных.

Решение:

Условные графические обозначения ИМС, указанных в задании представлены на рис.8.4.

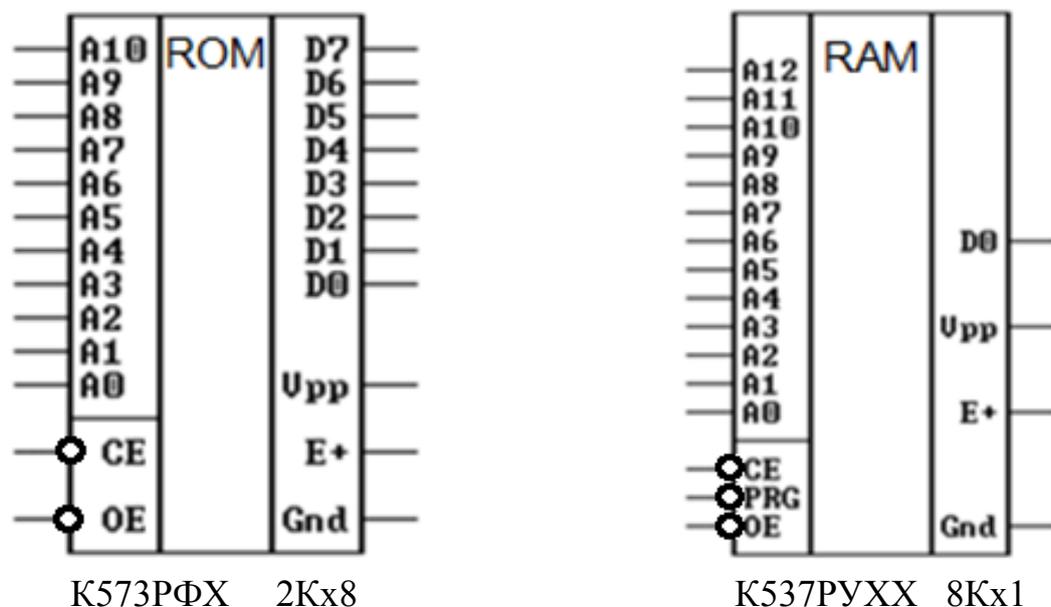


Рис.8.4. УГО ИМС памяти

У обоих ИМС имеется 2 входа разрешения с инверсным активным сигналом. Вход PRG у ИМС ПЗУ используется для программирования и в штатной работе процессора не применяется.

1). На основе исходных данных число банков в блоке памяти $32\text{Kb}/8\text{Kb}=4$, т.е. общий объем ОЗУ делится на объем одного банка. Объем банка определяется количеством адресуемых ячеек ИМС памяти. Если организация накопителя ОЗУ 8-разрядная, то число банков равно числу ИМС ОЗУ: $N_{\text{BANK}}=N_{\text{RAM}}$.

Однако если ИМС ОЗУ имеет другую организацию, то необходимо на основе имеющейся ИМС ОЗУ организовать банк с байтной структурой. Для рассматриваемого случая каждый банк необходимо создавать из восьми ИМС, соединив параллельно все ее входы. Такая схема представлена на рис.8.5. В этом случае выходы данных образуют единый байт с исходным количеством ячеек памяти. Поэтому в этом случае $N_{\text{RAM}}=32$ при $N_{\text{BANK}}=4$.

С учетом принципов построения 16-разрядного блока памяти (см. рис.8.3) количество страниц равно двум.

2). Аналогично число БИС ПЗУ в блоке памяти $N_{\text{ROM}} = 8\text{Kb}/2\text{Kb}=4$. Для ПЗУ организация накопителя байтная, поэтому число блоков и число ИМС совпадают. Количество страниц для схем ПЗУ равно двум.

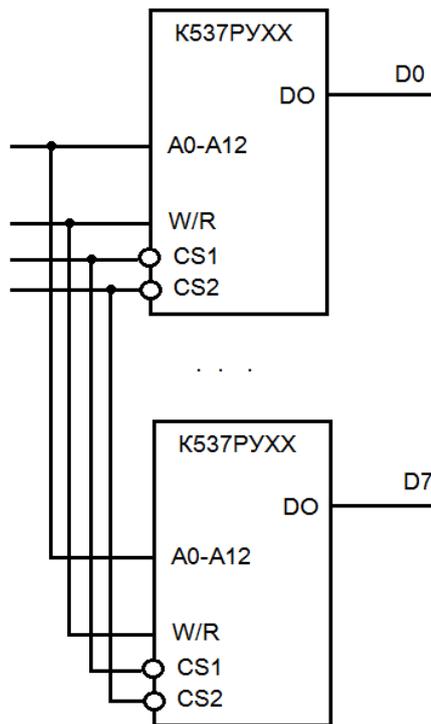


Рис.8.5. Организация байтной структуры банка памяти

3). Для определения начального адреса ПЗУ необходимо из всего объема адресного пространства процессора в 1Mb (100000H) вычесть заданный объем ПЗУ (число ячеек), т.е. 8Kb ($8 * 2^{10} = 2000H$). Получается:

$$100000H - 2000H = FE000H.$$

Для вычисления шестнадцатеричного кода числа 8K достаточно понимать, что 1K образуется из 10 двоичных разрядов: 100 0000 0000. Чтобы убедиться в этом, достаточно проследить веса разрядов по мере смещения в сторону старших разрядов: последний ноль имеет вес 512. Следовательно, следующий разряд с единицей имеет вес 1024. Такое число и обозначается как 1K. Это число в шестнадцатеричной системе записывается как 400H. Однако дальнейшие преобразования выполнять в шестнадцатеричной системе довольно трудно, поэтому лучше продолжать работать в двоичной системе.

Для получения 8K необходимо 1K умножить на 8, что соответствует трем сдвигам влево. Поэтому 8K эквивалентно числу 10 0000 0000 0000. Переведя это число в шестнадцатеричную систему (каждую тетраду представив одной шестнадцатеричной цифрой), получим как раз 2000H.

Следует заметить, что умножение в шестнадцатеричной системе даст такой же результат. Если 400H умножить на 8, то должны получить впереди десятичное число 32. Но в шестнадцатеричной системе это число 20H. Поэтому итоговый результат получается такой же: 2000H.

ПЗУ всегда находится в самом конце первого мегабайта адресного пространства, т.к. после включения питания ЦП выбирает для исполнения первую команду по адресу FFFF0H. Это справедливо и для процессора 80286.

4). Для определения конечного адреса ОЗУ необходимо к заданному начальному адресу (38000H) прибавить число ячеек ОЗУ (объем ОЗУ для рассматриваемого примера - 32Kb или $32 * 2^{10}=8000H$) и вычесть единицу. Получаем:

$$38000H + 8000H - 1 = 3FFFFH.$$

5). Определить номера разрядов шины адреса для формирования общего сигнала выборки ОЗУ ($SELECT_{RAM}$). Это те из 20-ти разрядов адреса, двоичные значения которых остаются постоянными для всех адресов из диапазона ОЗУ: 38000H ... 3FFFFH. Нумеруются разряды шины адреса соответственно A19 ... A0 слева направо.

$$38000H = \underline{0011} \underline{1000} \underline{0000} \underline{0000} \underline{0000}_2$$

$$3FFFFH = \underline{0011} \underline{1111} \underline{1111} \underline{1111} \underline{1111}_2.$$

Разряды, сохраняющие постоянное значение, подчеркнуты. Они соответствуют сигналам системной шины адреса A₁₉ ... A₁₅.

Таким образом, сигналы A₁₉..A₁₅ подаются на блок дешифрации адресов и формируют общий сигнал выборки ОЗУ:

$$SELECT_{RAM} = \bar{A}_{19} \& \bar{A}_{18} \& A_{17} \& A_{16} \& A_{15} \& (\overline{MRDC} \vee \overline{MWTC}).$$

В формировании сигнала $SELECT_{RAM}$ участвуют также сигналы MRDC (сигнал чтения из памяти), MWTC (сигнал записи в память) системной шины управления, т.к. ОЗУ в блоке памяти должно быть выбрано (активно) только во время циклов чтения или записи в память. Во время циклов ввода-вывода ОЗУ не выбрано (пассивно; хранит ранее записанную информацию).

С помощью сигнала $SELECT_{RAM}$ и нескольких старших разрядов из числа оставшихся - A₁₄..A₀ необходимо сформировать сигналы CS_x для выборки отдельных страниц ОЗУ. Для заданного числа страниц (N_{STR}) число дополнительных разрядов ША для дешифрации $n = \log_2 N_{STR} = \log_2 2 = 1$. Таким образом, для формирования отдельных сигналов CS₁, CS₂ должен использоваться разряд A₁₄ системной шины адреса. Остальные 14 младших разрядов системной шины адреса (A₁₃..A₀) на блок дешифрации адресов не подаются, а подключаются непосредственно к адресным входам и входам CS микросхем ОЗУ в блоке памяти. Число сигналов CS в этом примере равно числу страниц ОЗУ, т.е. двум.

Перебирая все возможные комбинации для сигналов A₁₄, запишем логические выражения для отдельных сигналов CS выборки страниц ОЗУ в рассматриваемом примере:

$$\overline{CS}_1 = SELECT_{RAM} \& \bar{A}_{14}$$

$$\overline{CS}_2 = SELECT_{RAM} \& A_{14}.$$

Блок дешифрации адресов может строиться на базе отдельных логических элементов, микросхем дешифраторов, БИС ПЗУ или ПЛМ. Однако в блоке дешифрации адресов проектируемой МПС наиболее рационально формировать сигнал $SELECT_{RAM}$ с помощью логических элементов и подавать его на вход выборки дешифратора, сигнал A14 - на адресные входы дешифратора, а сигналы CS1,..CS2 — снимать с выхода микросхемы дешифратора 2x4 с несколькими входами разрешения. Такая схема представлена на рис.8.6.

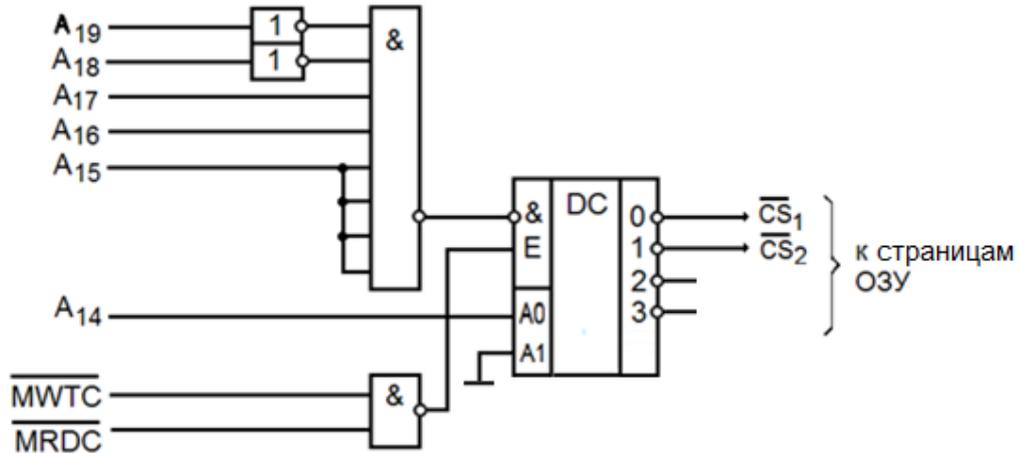


Рис.8.6. Схема дешифрации адреса ОЗУ

Аналогичные рассуждения позволяют получить логическое выражение для общего сигнала выборки ПЗУ:

$$FE000H = 1111\ 1110\ 0000\ 0000\ 0000$$

$$FFFFFH = \underline{1111\ 1111\ 1111\ 1111\ 1111}$$

$$SELECT_{ROM} = A_{19} \& A_{18} \& A_{17} \& A_{16} \& A_{15} \& A_{14} \& A_{13} \& \overline{MRDC}$$

Здесь используется только управляющий сигнал чтения из памяти MRDC, т.к. в процессе функционирования МПС из ПЗУ только считываются коды машинных команд управляющей программы. При этом для формирования отдельных сигналов выборки страниц ПЗУ используется разряд A12 системной шины адреса. Требуется только 1 разряд, т.к. количество страниц ПЗУ как и ОЗУ равно двум. Остальные 12 младших разрядов системной шины адреса (A11...A0) на блок дешифрации адресов не подаются, а подключаются непосредственно к адресным входам микросхем ПЗУ в блоке памяти. Замечание относительно разрядности системной шины данных и организации памяти в равной мере справедливо и для ПЗУ.

На рис.8.7 для рассматриваемого примера показан вариант схемы формирования сигналов CS выборки страниц ПЗУ блока памяти.

б). Порты ввода-вывода адресуются 16-ю младшими разрядами системной шины адреса. В проектируемой МПС достаточно использовать лишь

10 младших разрядов (A9...A0), т.к. адреса схем начинаются только с адреса 270H. При этом два младших разряда подключается непосредственно к адресным входам периферийных БИС в блоке внешнего интерфейса. Таким образом, в блоке дешифрации адресов будут использоваться только 8 разрядов ША (A9.. A2).

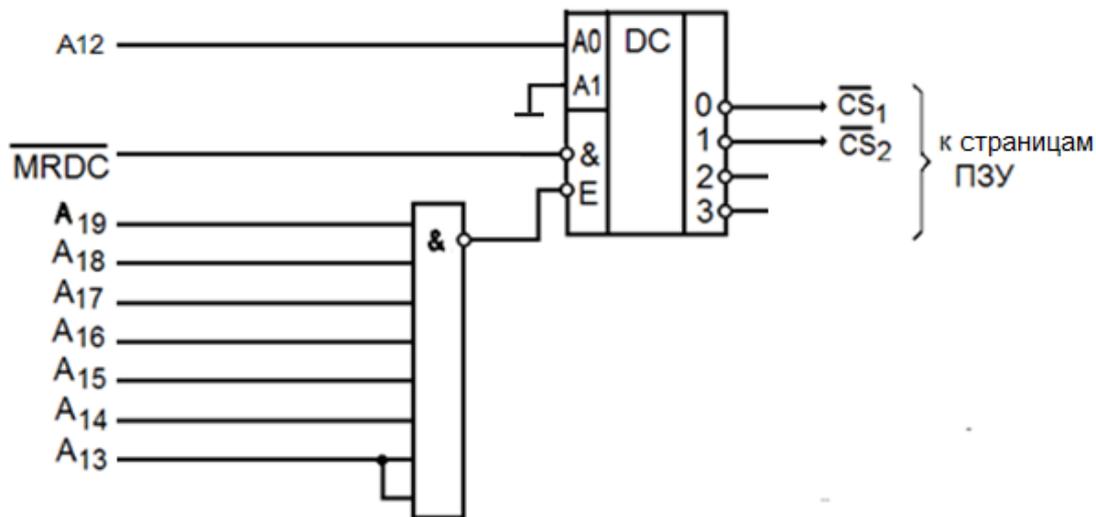


Рис.8.7. Схема дешифрации страниц блока ПЗУ

Для вариантов с 16-разрядной шиной данных обращение к внутренним регистрам периферийной БИС в управляющей программе выполняется только командами передачи байта по четному адресу из области ввода-вывода.

В этом случае сигнал A₀ системной шины адреса для выбора портов не используется (и всегда равен нулю в четном значении адреса), а адресные входы периферийной БИС подключаются к системной шине адреса со сдвигом на один разряд, т.е. вход A₀ к линии A₁ шины, вход A₁ к линии A₂ шины и т.д.

Отсюда следует, что начальный адрес ввода-вывода для последующей периферийной БИС (если такой требуется) на 8 больше заданного начального адреса ввода-вывода для предыдущей БИС. То есть для рассматриваемого примера

$$\text{Addr.2} = \text{Addr.1} + 8\text{H} = 270\text{H} + 8\text{H} = 278\text{H}.$$

Представив начальные адреса в виде двоичных комбинаций адресных сигналов, получаем логические выражения для сигналов выборки двух периферийных БИС:

$$\begin{aligned} \overline{CS_1} &= A_9 \& \overline{A_8} \& \overline{A_7} \& A_6 \& A_5 \& A_4 \& \overline{A_3} \& (\overline{IORC} \vee \overline{IOWC}) \\ \overline{CS_2} &= A_9 \& \overline{A_8} \& \overline{A_7} \& A_6 \& A_5 \& A_4 \& A_3 \& (\overline{IORC} \vee \overline{IOWC}) \end{aligned}$$

Адресная линия A₃ позволяет различить две ИМС внешнего устройства.

Следует учесть, что адресные линии A15-A10 не учтены в выражениях для выбора внешних устройств (эти разряды в выражениях несущественны). Поэтому случайно использованные адреса в этом диапазоне разрядов будут активизировать внешние схемы. Обращаться по ним недопустимо.

В формировании сигналов выборки участвуют также сигналы IORC, IOWC системной шины управления, т.к. каждая периферийная БИС в блоке внешнего интерфейса должна быть выбрана (для обмена данными между ЦП и ее внутренними регистрами) только во время циклов ввода-вывода. Во время циклов обращения к памяти программируемые периферийные БИС отключены от системной шины данных, т.е. соответствующие их выходы находятся в третьем состоянии (Z-состоянии).

На рис.8.8 для рассматриваемого примера показана схема формирования сигналов CS выборки двух периферийных БИС блока внешнего интерфейса. Схемы рис.8.6, 8.7 и 8.8. совместно образуют функциональную схему разработанного блока дешифрации адресов для приведенного примера. После выбора конкретных типов микросхем логики и дешифраторов на базе функциональной схемы строится принципиальная электрическая схема блока.

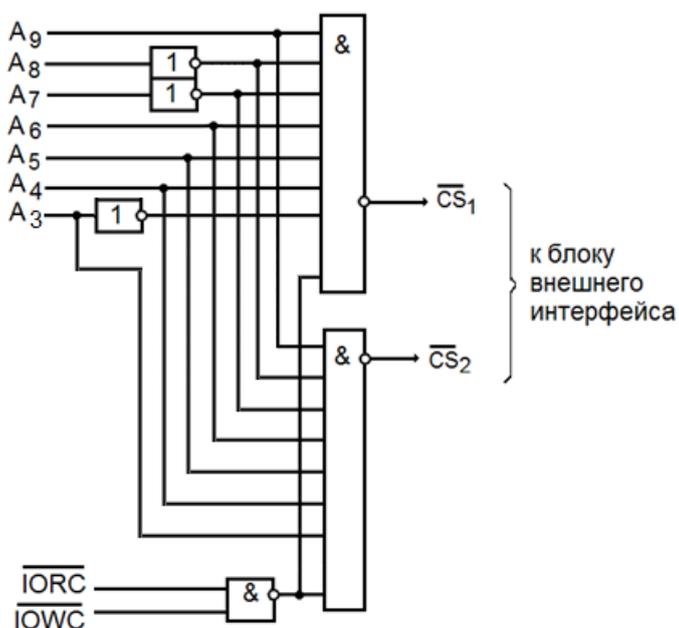


Рис.8.8.Схема дешифрации адреса периферийных БИС

8.3. БЛОК ПАМЯТИ

Наиболее просто блок памяти реализуется для вариантов с 8-разрядной системной шиной данных, в которых заданы БИС ОЗУ статического типа. В этом случае все одноименные входы-выходы D0.-D7 всех БИС блока соединяются параллельно и подключаются к соответствующим линиям данных системной шины. Аналогично все одноименные входы адреса всех БИС блока соединяются параллельно и подключаются к соответствующим линиям адреса системной шины, не использованным в блоке дешифрации адресов.

Отдельные сигналы CS, сформированные в блоке дешифрации адресов, подаются на входы CS соответствующих БИС памяти. Если микросхема памяти имеет несколько таких входов, все они соединяются параллельно, либо на дополнительные входы CS постоянно подается активный логический уровень. Все входы W/R БИС ОЗУ соединяются параллельно и подключаются к выходу MWTC системного контроллера, т.е. к линии сигнала "запись в память" системной шины управления.

Для вариантов с 16-разрядной системной шиной данных, в которых заданы БИС ОЗУ статического типа, блок памяти немного сложнее. Все БИС ОЗУ, как и все БИС ПЗУ, необходимо разбить попарно. Каждая пара, образуемая младшим и старшим банком данных, представляет собой страницу памяти. По одному из входов CS микросхемы в паре объединяются и на них подается соответствующий сигнал CS_x выбора страницы из блока дешифрации адресов. Микросхемы в паре активизируются одновременно, только если в цикле чтения или записи, начиная с четного адреса, происходит передача слова, т.е. если в цикле используются все 16 линий данных системной шины. В других случаях выбирается только одна микросхема из пары. Логика выбора для различных циклов памяти при наличии 16-разрядной системной шины данных была отражена в табл.8.1.

Одна БИС из пары хранит байты по четным адресам и ее входы-выходы данных должны быть подключены к одноименным линиям D0 ... D7 шины данных. Другая БИС из пары хранит байты по нечетным адресам и ее входы-выходы данных должны быть подключены к линиям D8...D15 шины данных.

Если БИС ПЗУ и БИС ОЗУ статического типа имеют минимум два входа выборки CS, то вторые входы выборки кристалла необходимо использовать для индивидуальной активизации каждой БИС ОЗУ и ПЗУ. На второй инверсный вход CS 1-й микросхемы из пары (D0-D7, четные адреса) подается младший адресный сигнал системной шины - A0. А на второй инверсный вход CS 2-й микросхемы из пары (D8-D15, нечетные адреса) подается управляющий сигнал ВНЕ ("выборка старшего байта") от центрального процессора МПС. Все одноименные входы адреса всех БИС блока соединяются параллельно и подключаются к линиям адреса системной шины, не использованным в блоке дешифрации адресов, начиная с адресной линии системной шины A1. Другими словами, адресный вход A0 каждой микросхемы памяти подключается к линии A1 системной шины адреса, адресный вход A1 микросхемы памяти подключается к линии A2 системной шины и т.д.

Если ИМС памяти имеет один вход CS, то на нем по схеме логического умножения активных сигналов должны быть объединены сигнал CS блока дешифрации и сигнал A0 (для младшего банка четные адреса) и ВНЕ (для старшего банка нечетные адреса) соответственно.

Описанная схема включения (рис.8.9) обеспечивает необходимый порядок выборки микросхем ОЗУ и ПЗУ в блоке памяти со словной организа-

цией.

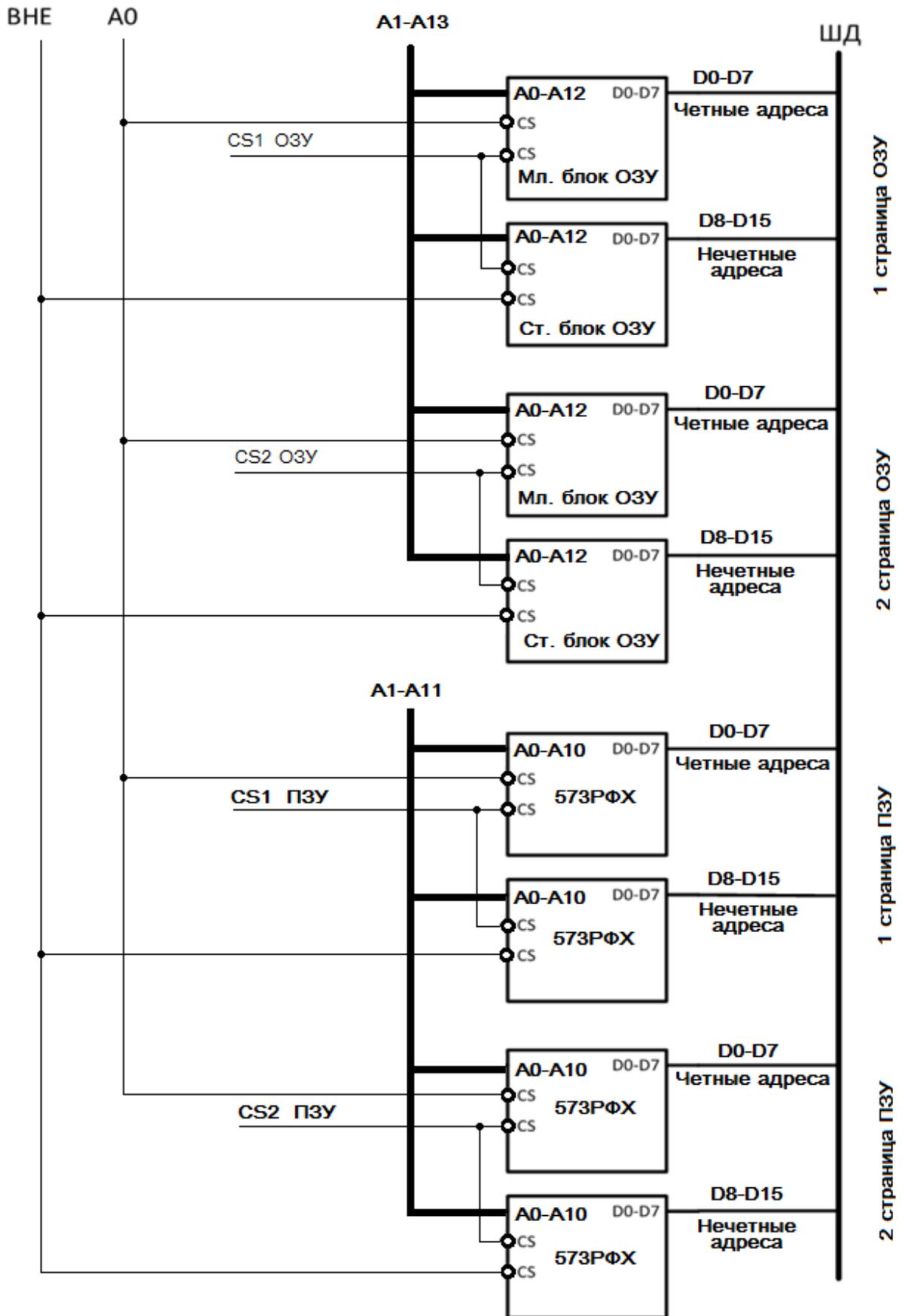


Рис.8.9. Вариант построения блока памяти ОЗУ и ПЗУ

Для вариантов с БИС ОЗУ динамического типа (заданный объем оперативной памяти - 64Kbit и более) при расчете числа микросхем необходимо пользоваться формулой:

$$N_{\text{RAM}} = (M_{\text{ОБЩ}}/M_1)*8,$$

где $M_{\text{ОБЩ}}$ - заданный объем ОЗУ в килобайтах,

M_1 -число ячеек в одной БИС заданного типа в килобитах. БИС ОЗУ динамического типа имеют битовую организацию [5].

Для управления мультиплексированием адресных сигналов, формирования сигналов RAS (выбор строки) и CAS (выбор столбца), а также для организации режима регенерации микросхем ОЗУ динамического типа в блоке памяти целесообразно использовать специализированную БИС - контроллер динамического ОЗУ, например, отечественную микросхему К1810ВТ03 [2, раздел 5.5]. Типовые структурные схемы блока динамического ОЗУ с использованием БИС К1810ВТ03 можно найти в литературе [2, рис. 5,45 на с.190, рис. 5.47 на с.191].

8.4. БЛОК ВНЕШНЕГО ИНТЕРФЕЙСА

Несмотря на то, что в индивидуальных вариантах задания для использования в учебной МПС указаны различные программируемые периферийные БИС, все они имеют ряд одноименных выводов, аналогичных по своему функциональному назначению:

1) входы-выходы данных (D0.-D7), через которые центральный процессор передает байты команд во внутренние регистры периферийной БИС во время выполнения программы ее начальной инициализации (настройки режима работы), а затем обменивается информацией с внешним устройством, которым управляет периферийная БИС;

2) адресные входы (обычно от одного до четырех) для выбора одного из внутренних регистров периферийной БИС перед обменом данными;

3) вход выборки кристалла CS, на который необходимо в цикле ввода-вывода подавать активный уровень сигнала (лог. 0), иначе обмен данными с ЦП будет невозможен;

4)управляющие входы чтения (RD) и записи (WR), на которые необходимо подать сигналы IORD и IOWR системной шины управления соответственно;

5) вход для сигнала системного сброса RST (может обозначаться и как RESET), переводящего все внутренние регистры периферийной БИС и ее управляющий микропрограммный автомат в исходное состояние;

6) вход тактового синхросигнала CLK (имеется не у всех программируемых БИС) для тактирования и синхронизации всехвнутренних последовательностных устройств БИС. Периферийные БИС некоторых типов имеют

собственный интегрированный на кристалле синхрогенератор и выводы для подключения внешнего частото задающего элемента -кварцевого резонатора.

На рис.8.10 показана схема подключения выводов программируемого таймера учебной МПС к системной шине, блоку дешифрации адресов и системному синхрогенератору. Выводы и сигналы, специфичные для каждого конкретного типа БИС можно уточнить в [11].

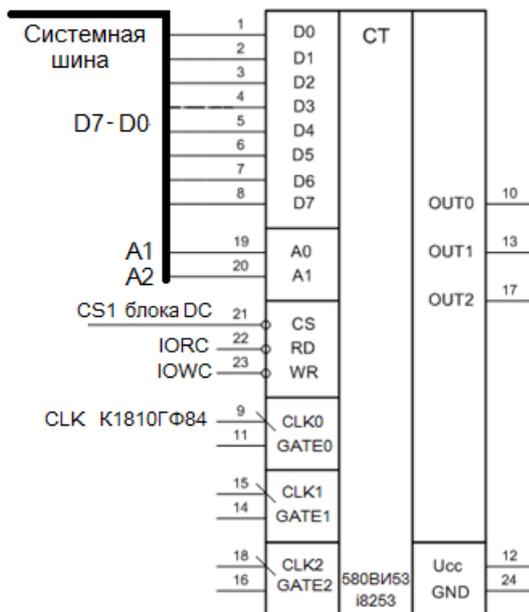


Рис.8.10. Типовая схема включения таймера в состав МПС

8.5. ОСНОВНАЯ ПРОГРАММА МПС

Поскольку проектируемая МПС относится к классу встраиваемых систем, ее программное обеспечение включает в себя единственную управляющую программу, хранящуюся в ПЗУ. Благодаря этому, управляющая программа начинает исполняться сразу же после включения питания системы. На рис.8.11 показана укрупненная схема алгоритма управляющей программы. Схема состоит из процедуры инициализации всех программируемых периферийных БИС системы, процедуры инициализации внешних по отношению к МПС устройств и главного рабочего цикла. Кроме того, в состав управляющей программы могут входить процедуры обработки прерываний (аппаратных, по запросу периферийных БИС или внешних устройств и программных).

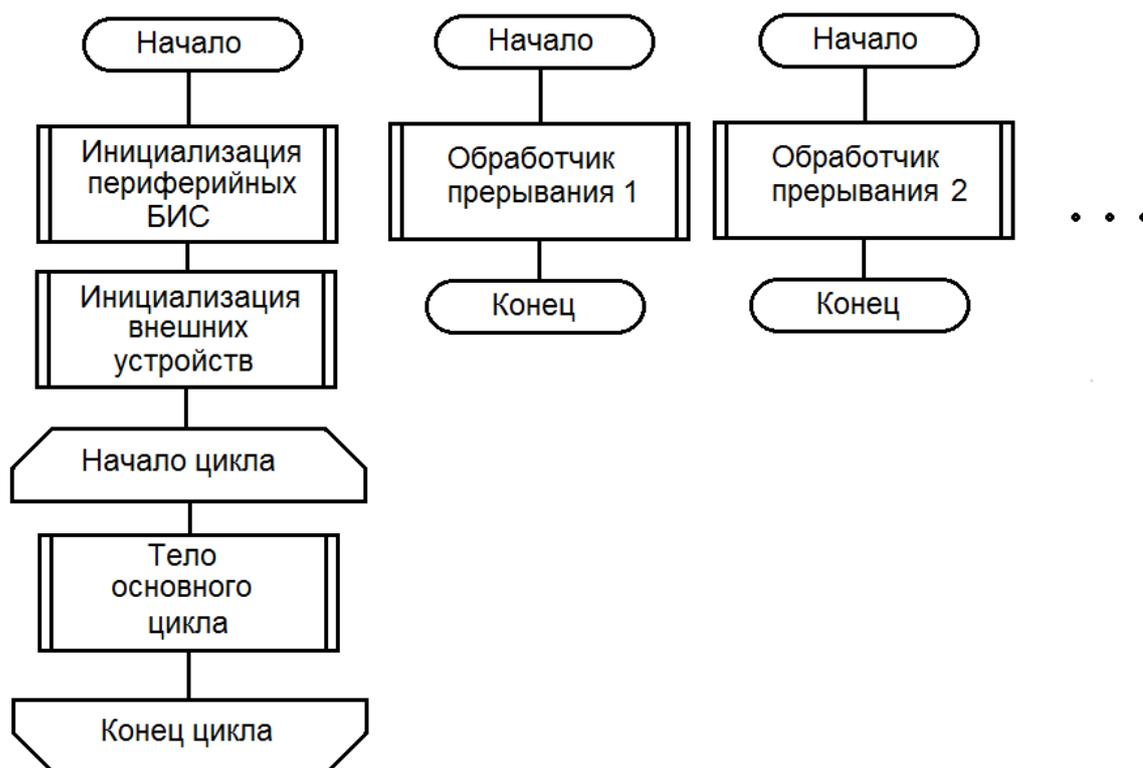


Рис. 8.11. Укрупненная схема алгоритма управляющей программы МПС

В рамках курсового проектирования необходимо реализовать процедуру инициализации программируемых периферийных БИС системы на языке Ассемблер (для процессоров Intel x86) [2]. Описания БИС даны в литературе [11]. По заданию преподавателя также может быть реализована процедура инициализации внешних устройств или главный рабочий цикл управляющей программы. В главном рабочем цикле может быть использован механизм "активного ожидания" (в противоположность механизму аппаратных прерываний) для контроля состояния внешнего устройства, которым будет управлять проектируемая МПС.

После включения питания первая команда управляющей программы должна быть выбрана центральным процессором по абсолютному адресу FFFF0H из ПЗУ. Поэтому, приступая к кодированию программы, необходимо определить номер ячейки ПЗУ для первого байта первой команды. Этой командой должна быть команда безусловного межсегментного ("длинного") перехода по абсолютному адресу, т.е. команда JMP сегмент:смещение. В команде JMP указывается вычисленный ранее абсолютный адрес начала области ПЗУ в адресном пространстве процессора, который соответствует первой ячейке ПЗУ в блоке памяти по выполняемому варианту. Адрес FFFFFH в адресном пространстве процессора соответствует номеру последней ячейки ПЗУ.

В рассмотренном выше примере используется ПЗУ объемом 8Кб, что соответствует ячейкам с номерами от 0 до 1FFFH. Тогда, с учетом вышеиз-

ложенного, абсолютному адресу FFFF0H в адресном пространстве процессора соответствует ячейка ПЗУ с номером 1FF0H. ПЗУ должно быть заранее запрограммировано, и делается это автономно от МПС системы, с последующим подключением в процессорный блок. Поэтому программа должна предполагать размещение данных в объеме ПЗУ 8кв. После первого обращения процессора по адресу ПЗУ FFFF0H должен быть выполнен межсегментный (длинный) переход на начало ИМС ПЗУ с адресом 0000H. Начальный адрес области ПЗУ процессора (сегментный адрес) должен быть записан при этом в сегментный регистр команд CS. Формат команда межсегментного перехода это обеспечивает и формирует 5-байтную команду. Все последующие команды размещаются в ПЗУ с начальным смещением 0000H.

Инициализация МПС предполагает инициализацию сегмента стека (например, для рассматриваемого примера объемом 256b в конце адресов ОЗУ), сегмента данных, начиная с первого адреса ОЗУ и т.д.

Адреса ПЗУ в этом примере начинаются с FF00H, поэтому начальный сегмент адресов ПЗУ равен FE00H.

Адреса ОЗУ начинаются с адреса 3800H, поэтому сегмент данных должен начинаться с 3800H. Конечный адрес ОЗУ равен 3FFFFH, поэтому сегмент стека (размером 256b = 100H байт) должен начинаться с номера 3FFFFH – 100H = 3EEFH.

Поэтому для рассмотренного примера начало исходного текста программы инициализации для ИМС ПЗУ примет следующий вид:

```

ORG 1FF0H      ;ячейка ПЗУ, выбираемая по адресу МПС 0FFFF0h
JMP 0FE00h:0   ;межсегментный переход на начальный адрес ПЗУ

ORG 0          ;первая ячейка ПЗУ (адрес МПС0FE000h)

START:         ;метка начала процедуры инициализации
mov ax,3EEFH   ;установка регистра сегмента стека
mov ss,ax      ;размером 100H

mov ax,3800h   ;установка регистра сегмента данных на
mov ds,ax      ;начальный адрес области ОЗУ
...

```

Директива ORG Ассемблера задает компилятору смещение для следующей команды, тем самым обеспечивая верный формат объектного двоичного файла, который затем будет использован программатором ПЗУ.

Каждый канал таймера программируется индивидуально в соответствие с управляющим словом. Структура управляющего слова таймера представлена на рис.3.3.

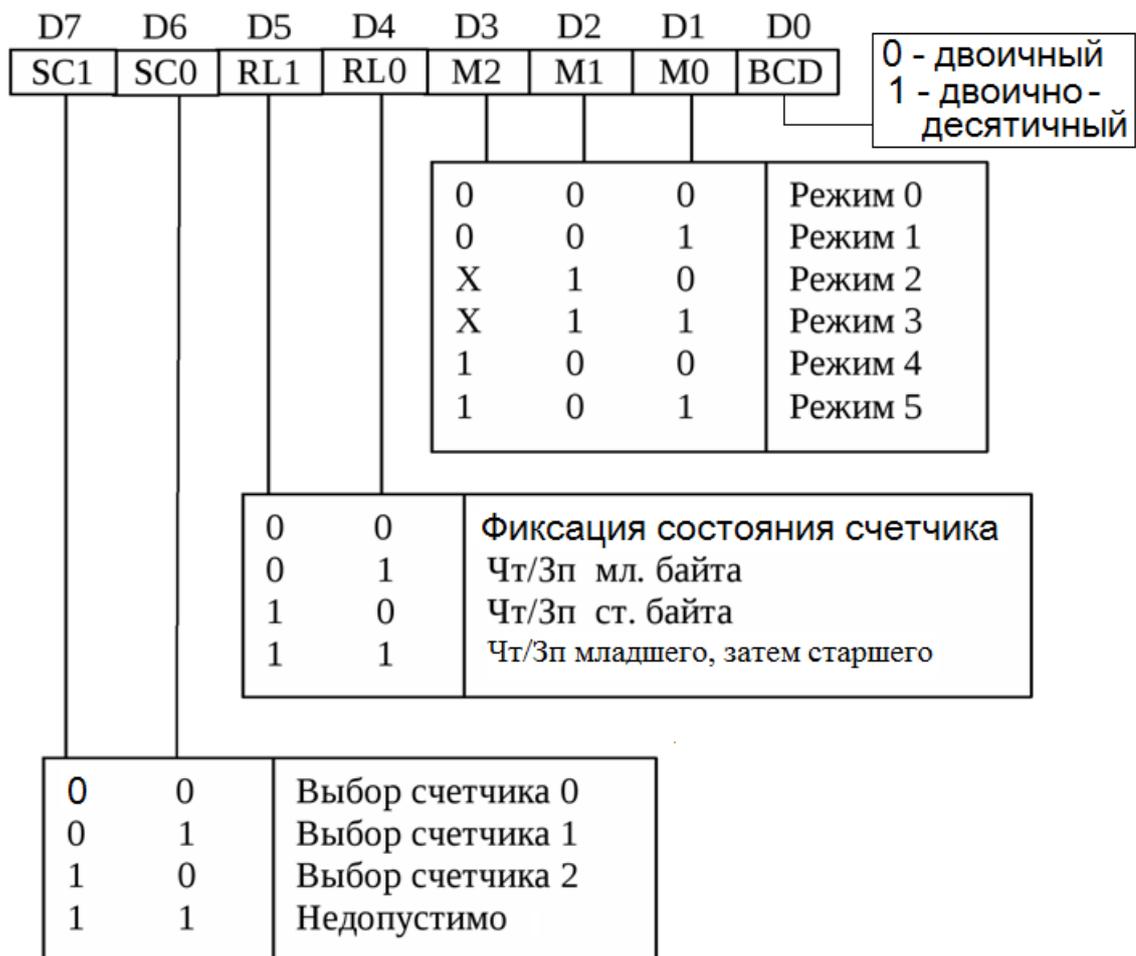


Рис.3.3. Структура управляющего слова таймера

Поле BCD (Binary-Coded Decimal) определяет способ кодирования константы предустановки счетчика.

Двоично-десятичный формат предполагает запись десятичного разряда числа в виде его четырехбитного двоичного кода. Например, десятичное число 311 будет записано в двоичном коде как 100110111, а в двоично-десятичном коде как 001100010001. Это упрощает ввод чисел, однако уменьшает диапазон вводимых чисел при той же разрядности. При BCD кодировке диапазон допустимых чисел для 16 разрядов счетчика 0 - 9999, а в двоичном варианте 0 - 65535.

Программная модель таймера имеет 4 буфера, адресуемых по следующим адресам:

Адрес на входах A1 A0	Буфер
00	Канал 0
01	Канал 1
10	Канал 2
11	Управляющее слово

Пример программирования таймера.

Запрограммировать счетчикСТ0в режим генератора импульсов для получения частоты $f_{\text{вых}}= 1\text{кГц}$. Адрес обращения к таймеру 270H. Частота входного сигнала CLK= 5МГц.

Для режима 2 (делителя) значения коэффициента деления:

$$N = f_{\text{CLK}} / f_{\text{вых}} = 5000000/1000 = 5000.$$

В двоичном представлении десятичное $5000 = 0001\ 0011\ 1000\ 1000 = 1388\text{H}$.

Коэффициент деления можно представить и в кодеBCD: $5000 = 0101\ 0000\ 0000\ 0000 = 5000\text{H}$.

Управляющее слово в соответствии с рис.3.3 для программирования счетчикаСТ0в режиме2, с двоичным способом кодирования:

D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	1	0	1	1	0

Управляющее слово - 36H.

Тогда программа для таймера К580ВИ53 в кодах процессора i8086 будет иметь вид:

```
ADR53 EQU 270H ; директива присваивания имени ADR53
; значения адреса таймера
MOV AL, 36H ; Задание управляющего слова
OUT ADR+6, AL ; используются только четные адреса
MOV AL, 88 ; Запись младшего байта
OUT ADR53, AL ; кода предустановки
MOV AL, 13H ; Запись старшего байта
OUT ADR3, AL ; кода предустановки
```

После выполнения программы на выводеOUT0импульсы с частотой 1 кГц будут до тех пор, пока не будет перепрограммирован таймер, или выключен источник тактовых сигналов 0-канала таймера, или снят сигнал разрешения GATE.