

Министерство образования и науки Украины
Севастопольский национальный технический
университет



РАЗРАБОТКА МИКРОПРОЦЕССОРНОЙ СИСТЕМЫ ПОЛУЧЕНИЯ, ОБРАБОТКИ И ОТОБРАЖЕНИЯ ИНФОРМАЦИИ НА ОСНОВЕ 8-РАЗРЯДНОГО МИКРОПРОЦЕССОРА

Методические указания
к курсовой работе по дисциплине
“Микропроцессоры и ЭВМ”
для студентов по направлению “Приборы”
дневной и заочной форм обучения

Севастополь
2011



Разработка микропроцессорной системы получения, обработки и отображения информации на основе 8-разрядного микропроцессора. Методические указания к курсовой работе по дисциплине “Микропроцессоры и ЭВМ” / Сост.: Л.Е. Карташов – Севастополь: издательство СевНТУ, 2011. – 24 с.

Руководство предназначается для оказания помощи студентам при выполнении проектирования цифровых микропроцессорных систем в рамках курсовой работы по дисциплине “Микропроцессоры и ЭВМ”. В нем определены цели и задачи курсовой работы, задание, требования к объёму и содержанию отчётной документации, критерии оценки и порядок защиты результатов, даны методические советы по выполнению задания.

Объектом разработки является прототип микропроцессорной системы на основе 8-разрядного микропроцессора, содержащий необходимый набор устройств для построения микропроцессорной системы измерительного прибора, предназначенной для получения, обработки и отображения информации.

Методические указания предназначены для студентов дневной и заочной форм обучения по направлению 6.051003 “Приборы”.

Методические указания рассмотрены и утверждены на заседании кафедры АПС, протокол № 7 от 07.04.2011 г.

Методические указания рассмотрены и утверждены на научно-методическом совете СевНТУ, протокол №__ от 20__ г.

Допущено учебно-методическим центром СевНТУ в качестве методических указаний.

Рецензенты: Волков С.П., к.т.н., доцент кафедры АТПП

СОДЕРЖАНИЕ

Введение	4
1. Цели курсовой работы	4
2. Типовое задание. Наименование темы	5
3. Цели разработки	5
4. Исходные данные	5
4.1. Общие исходные данные	5
4.2. Индивидуальные исходные данные	6
5. Задачи и результат проектирования	7
6. Требования к отчётной документации	8
7. Методические рекомендации по выполнению типового задания	8
7.1. Общий подход к проектированию. Этапы проектирования	8
7.2. Этап структурного проектирования	10
7.2.1. Разработка структурной схемы модуля центрального процессора	10
7.2.2. Сигналы системной магистрали	13
7.2.3. Замечания к разработке структурных схем памяти и блока синхронизации....	13
7.2.4. Разработка структурной схемы подсистемы ввода-вывода	14
7.3. Этап функционально-логического проектирования	15
7.3.1. Разработка принципиальной схемы микропроцессорной системы	15
7.3.2. Замечания к разработке принципиальной схем памяти и подсистемы ввода–вывода	15
7.4. Разработка алгоритмов управляющей программы	18
8. Порядок проверки и защиты курсовых работ. Критерии оценки	21
Библиографический список	23

ВВЕДЕНИЕ

Данное руководство предназначается для оказания помощи студентам в приобретении первого опыта проектирования цифровых микропроцессорных систем. В нем определены цели и задачи курсовой работы (КР), многовариантное типовое задание, требования к объёму и содержанию отчётной документации, критерии оценки и порядок защиты результатов, даны методические советы по выполнению задания. Ориентация на начинающих разработчиков повлияла на выбор тем КР и уровня сложности заданий и послужила причиной изложения некоторых положений и рекомендаций, очевидных для искушённых инженеров.

Тематика КР и решаемые в них вопросы согласованы с квалификационной характеристикой к минимуму содержания и уровню подготовки бакалавра по направлению 6.051003.

Объектом разработки в КР является прототип микропроцессорной системы на основе 8-разрядного микропроцессора, содержащий необходимый набор устройств для построения микропроцессорной системы измерительного прибора, предназначенной для получения, обработки и отображения информации: микропроцессор, блок синхронизации, основную память и периферийные устройства. В отличие от полной версии в прототипе игнорируются функции, реализация которых относится к проблемам ещё неизученных студентами дисциплин.

В КР совмещается функциональное и алгоритмическое проектирование. Функциональное (схемное) проектирование осуществляется на двух иерархических уровнях: структурном и функционально-логическом. Алгоритмическое проектирование связано с формированием структуры алгоритмов и составлением программ, позволяющих реализовать возложенные на разрабатываемую микропроцессорную систему задачи.

Варианты типового задания имеют разную сложность. Она определяется преподавателем при выдаче задания подбором индивидуальных исходных данных на разработку с учётом ожидаемой трудоёмкости, нормы времени на самостоятельную работу, уровня подготовки и пожеланий студента.

1. ЦЕЛИ КУРСОВОЙ РАБОТЫ

Курсовая работа направлена на достижение студентами двух целей.

Первая цель predetermined квалификационными требованиями по направлению 6.051003 и соответствующими им целью и задачами изучения дисциплины «Микропроцессоры и ЭВМ», установленными образовательно-профессиональной программой СевНТУ по направлению 6.051003, и состоит:

- в освоении элементов методологии проектирования аппаратных средств вычислительной техники;
- в приобретении практического опыта разработки функциональных устройств и узлов приборов, выполнения схмотехнических расчетов и оформления схемной документации;
- в закреплении и углублении знаний о современной элементной базе.

Вторая цель выходит за рамки дисциплины и основывается на общих требованиях квалификационной характеристики к образованности бакалавра и на тенден-

циях образования. Она может быть обозначена как развитие познавательной активности и интеллектуальных умений и навыков, особенно значимых в условиях непродолжительной жизни прикладных знаний.

2. ТИПОВОЕ ЗАДАНИЕ. НАИМЕНОВАНИЕ ТЕМЫ

В типовом задании принято общее наименование темы, адекватное сущности КР: “Разработка микропроцессорной системы получения, обработки и отображения информации на основе 8–разрядного микропроцессора”.

Для краткости в названии не отражены специфические особенности конкретной КР и опущено подразумеваемое упоминание о разработке программного обеспечения для микропроцессорной системы.

3. ЦЕЛИ РАЗРАБОТКИ

Целью разработчикам ставятся наиболее существенные желаемые характеристики прототипа микропроцессорной системы получения, обработки и отображения информации:

- высокое быстродействие (производительность);
- простота и надежность схемных решений.

Под надежностью понимаются не количественные показатели, а отсутствие в схемах критических временных соотношений, рисков сбоя сигналов, использование унифицированных способов обмена между устройствами и т.п.

Дополнительные цели устанавливаются по усмотрению разработчика.

4. ИСХОДНЫЕ ДАННЫЕ

Исходные данные представляют собой первичное описание прототипа микропроцессорной системы, являющейся объектом разработки.

Задание устанавливает требования к количеству архитектурных характеристик и параметров прототипа микропроцессорной системы. Некоторые ограничения введены с целью уменьшения трудозатрат и упрощения схемных решений.

Выдаваемые студентам задания на КР включают в себе общие и индивидуальные исходные данные.

4.1. Общие исходные данные

В состав разрабатываемого объекта входят: программируемый микропроцессор, постоянные запоминающие устройства (ПЗУ) и оперативные запоминающие устройства (ОЗУ), реализующие функции основной памяти, средства интерфейса, обеспечивающие информационную и электрическую совместимость процессора и ЗУ, блок синхронизации и системный контроллер, параллельный интерфейс, предназначенный для управления и получения информации с восьмиразрядного АЦП. Другие периферийные устройства ввода-вывода указаны в индивидуальных исходных данных. Интерфейс реализуется на основе магистрали и функциональных узлов, являющихся составной частью микропроцессорной системы.

Прототип микропроцессорной системы должен соответствовать следующим требованиям к ее характеристикам и параметрам:

- разрядность шины адреса – 16;
- разрядность шины данных – 8;

- структура магистрали – трёхшинная;
- обмен данными процессор – память (рекомендуемый) – синхронный без квитирования (без подтверждения готовности памяти и циклов ожидания процессора);
- микропроцессорная система должна быть оснащена схемными средствами поддержки начальной установки при включении, и элементами управления операций получения и отображения информации.

4.2. Индивидуальные исходные данные

Индивидуальные исходные данные определяют требования к составу и элементной базе микропроцессорной системы, а также к функциональному составу устройств ввода-вывода и ёмкости памяти. Алгоритмы и код управляющих программ, необходимых для реализации функций прибора, устанавливает преподаватель при выдаче задания.

Микросхемы, обязательные для применения при формировании адресной шины микропроцессорной системы, их наличие в ее структуре и некоторые другие требования приведены в таблице 1. Таблица 2 определяет ёмкость оперативного запоминающего устройства и постоянного запоминающего устройства микропроцессорной системы. В таблице 3 указаны устройства ввода-вывода, тип и разрядность устройств вывода информации.

Не указанная в таблицах 1–3 элементная база должна обоснованно выбираться разработчиком. Вариант индивидуальных исходных данных задается набором трёх строк в таблице 1–3.

Табличные исходные данные могут модифицироваться для уравнивания или, наоборот дифференциации сложности и трудоёмкости, или с целью разнообразия заданий.

Таблица 1 – Состав и элементная база формирователей шины адреса

	Процессорный модуль, регистровый файл,	Разрядность
	Двунаправленный формирователь K589АП16	4
	Двунаправленный формирователь КР580ВА86	8
	Регистр с выходными формирователями КР580ИР82	8
	Многорежимный буферный регистр К589ИР12	8

Таблица 2 – Требования к ОЗУ и ПЗУ

№	Объем памяти		№	Объем памяти	
	ПЗУ, КВ	ОЗУ, КВ		ПЗУ, КВ	ОЗУ, КВ
1	0,5	2	6	1	2
2	2	1	7	3	1
3	1	4	8	2	2
4	3	1	9	1	0,25
5	1	0,25	10	0,5	4

Таблица 3 – Функциональный состав устройств ввода-вывода и индикаторов

№	Устройств ввода-вывода			Индикаторы ¹⁾	
				Способ индикации	Количество
1	KP580BB55	KP580BH59	KP580BI53	Статический	4
2	KP580BB51	KP580BH59	KP580BI53	Динамический	3
3	KP580BB55	KP580BH59	KP580BI53	Динамический	3
4	KP580BB51	KP580BH59	KP580BI53	Статический	4
5	KP580BB55	KP580BH59	KP580BI53	Динамический	3
6	KP580BB51	KP580BH59	KP580BI53	Статический	2
7	KP580BB55	KP580BH59	KP580BI53	Динамический	4
8	KP580BB51	KP580BH59	KP580BI53	Статический	4
9	KP580BB55	KP580BH59	KP580BI53	Динамический	3
10	KP580BB51	KP580BH59	KP580BI53	Статический	2
1) В качестве индикаторов использовать светодиодные семисегментные.					

Общие и индивидуальные исходные данные определяют конечное множество базовых вариантов типового задания. Они служат основой для генерации частично пересекающихся с базовыми, но своеобразных заданий различной сложности путём внесения от одного до трёх дополнений или/и изменений в общие и индивидуальные исходные данные.

Так, например, для хорошо подготовленных студентов, тяготеющих к проектной работе, могут быть сформированы задания повышенной сложности за счёт включения в них некоторых из перечисленных ниже требований:

- Обеспечить обмен данными процессор-память с использованием динамической памяти.
- Реализовать вывод информации с помощью контроллера клавиатуры и дисплея – KP580BB79.
- Реализовать синхронный (асинхронный) протокол обмена с квитированием по последовательному каналу с удаленным контроллером.

Увеличение трудоёмкости, вызываемое усложнением задания, может компенсироваться понижением степени детализации части микропроцессорной системы до уровня функциональных схем.

Задание на КР формулируется руководителем на официальном бланке задания с указанием темы, индивидуальных исходных данных, задач проектирования или перечня разделов пояснительной записки, графического материала и срока представления КР к защите.

5. ЗАДАЧИ И РЕЗУЛЬТАТ ПРОЕКТИРОВАНИЯ

Процесс разработки ещё не существующего прототипа базе микропроцессорной системы заключается в составлении на основе исходных данных описания ее структуры на различных уровнях детализации, в описании основных параметров, характеристик, взаимодействий между различными частями и алгоритмов функционирования при автоматическом выполнении процедуры получения, обработки и отображения информации.

В процессе разработки решаются следующие задачи:

- анализ и регламентация архитектурных характеристик и ров, необходимых для построения структуры микропроцессорной системы;
- разработка структурной и принципиальной электрических схем с обоснованием выбора элементной базы, не указанной в задании;
- разработка алгоритмов и кода программы, реализующей требуемый процесс измерения, обработки и вывода информации;
- анализ принятых решений, включая схемотехнические расчёты, необходимые для проверки работоспособности и оценки быстродействия с учётом ограничений элементной базы.

Результатом разработки являются промежуточные и конечные проектные решения, представляемые в виде расчётно-пояснительной записки, структурной и принципиальной электрических схем. Допускается разработка не полной принципиальной схемы, а её отдельных фрагментов, раскрывающих наиболее сложные и/или оригинальные узлы.

6. ТРЕБОВАНИЯ К ОТЧЁТНОЙ ДОКУМЕНТАЦИИ

Объём расчётно-пояснительной записки 20–30 страниц рукописного или около 20 страниц отпечатанного на принтере текста, включая иллюстрации.

Титульный лист записки оформляется в соответствии с принятыми в СевНТУ правилами.

Пояснительная записка должна содержать задание на КР, перечень разделов, введение, решение задачи, заключение, и список использованной литературы. Структура и стиль пояснительной записки должны соотноситься с проектным характером работы и блочно-иерархическим принципом нисходящего проектирования.

Содержание пояснительной записки должно соответствовать заданию и охватывать решение всех предусмотренных им задач.

Схемы оформляются согласно требованиям ЕСКД. Фрагменты структурной схемы помещаются в пояснительную записку. Рисунки, таблицы нумеруются и снабжаются наименованиями. Принципиальная электрическая схема выполняется на отдельном листе (листах) ватмана стандартного формата, все обозначения в ней должны быть ясно различимы. Допускается в принципиальной схеме не указывать цоколёвку микросхем, с тем, чтобы опустить стереотипную работу.

7. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ ТИПОВОГО ЗАДАНИЯ

7.1. Общий подход к проектированию. Этапы проектирования

Аппаратные средства вычислительной техники, в том числе сложные по выполняемым функциям и составу цифровые устройства и их композиции разрабатывают на основе давно сложившегося блочно-иерархического принципа. Сущность его заключается в следующем.

Проектирование аппаратуры, создаваемой на базе микросхем разной степени интеграции, проводят в нисходящей последовательности, с возрастающей степенью подробности описания на двух иерархических уровнях.

На высшем (структурном) уровне разрабатывают структурную схему аппаратуры, основываясь на декомпозиции системных функций с учётом возможностей элементной базы. Структурная схема содержит в качестве элементов условные графические изображения (обозначения) функциональных комплексов (композиций устройств) и/или устройств, узлов и даёт общее представление об объекте, определяет его составные части, их назначение и взаимосвязи.

На соседнем функционально-логическом уровне разрабатывают функциональные и принципиальные электрические схемы.

Функциональная схема формирует представление о работе аппаратуры с учётом наиболее существенных факторов и функциональных частей.

Принципиальная схема определяет полный набор элементов, их взаимосвязей и детально описывает принцип работы аппаратуры.

Каждая из названных схем описывает внутреннее устройство, или структуру аппаратных средств как совокупность элементов и их взаимосвязей, но с разной степенью подробности. (Слово элемент означает не материальную часть сложного целого, а понятие, смысл которого изменяется при переходе с одного уровня проектирования на другой).

На верхнем уровне описания проектируемого объекта определяется функциональное назначение каждого структурного элемента и требования к нему, которые служат исходными данными (заданием) для разработки этого элемента на соседнем нижнем уровне.

Переходя на этот нижний уровень, элементы верхнего уровня описывают не в целом, а по отдельности, что уменьшает размерность задачи и позволяет выполнять проектные работы параллельно. В качестве составных частей элемента верхнего уровня используют условные графические обозначения элементов более низкого ранга, присущие нижнему уровню.

Подобно описанному процессу блочно-иерархический принцип применим и к алгоритмическому проектированию, которое в КР выполняется для реализации функций получения, обработки и отображения информации.

Задача разработки схемы, реализующей заданный набор функций, в общем случае имеет множество решений и обычно сводится к поиску и систематизации типовых (существующих) и построению новых вариантов, формулированию критерия предпочтения вариантов и к выбору из конечного множества решений наиболее удачного на основе этого критерия. В создании новых схемных решений большую роль играют неформальные интуитивно-эвристические приёмы и элементы изобретательства. Процесс проектирования имеет итерационный характер и в ряде случаев может повторяться неоднократно.

Разработку прототипа микропроцессорной системы, представляющего собой совокупность достаточно сложных устройств, целесообразно выполнять на основе блочно-иерархического принципа, так как применение его значительно упростит решение задачи и сократит затраты времени.

Проектирование микропроцессорной системы включает два аспекта: схемный и алгоритмический.

Схемное проектирование в соответствии с постановкой задачи и блочно-иерархическим принципом разделяется на два этапа: структурного и функционально-логического проектирования.

Исходные данные содержат большую часть характеристик процессорной системы, предопределяющих ее структуру. Однако в задании преднамеренно не регламентированы некоторые архитектурные характеристики и параметры, необходимые для построения схем. Чтобы получить полное корректное первичное описание на архитектурном уровне, достаточное для схемного воплощения, надо конкретизировать требования к системному интерфейсу; распределить адресное пространство памяти и пространство ввода-вывода; подключить элементы управления. Решение этих вопросов зависит от особенностей элементной базы и влияет на схемные решения и потому отнесено к этапу структурного проектирования.

На втором этапе осуществляется во взаимосвязи функционально-логическое проектирование микропроцессорной системы и алгоритмическое проектирование управляющих программ.

7.2. Этап структурного проектирования

На данном этапе разрабатывается структурная схема микропроцессорной системы, затем структурные схемы ее элементов. На основе полученных результатов формируются требования к структурным элементам более низкого ранга, и осуществляется переход на функционально-логический уровень проектирования, т.е. ко второму этапу.

Некорректность первичного описания микропроцессорной системы устраняется установлением требований к характеристикам и значениям параметров по мере возникновения надобности в них.

7.2.1. Разработка структурной схемы модуля центрального процессора

Проектирование структуры заключается в определении состава микропроцессорной системы и способов взаимодействия содержащихся в нем устройств.

Структурная схема komponуется на основе ее архитектурных характеристик. По типовому заданию в состав модуля центрального процессора микропроцессорной системы входят микропроцессор, формирователи шины адреса, системный контроллер, память ROM(ПЗУ) и RAM(ОЗУ), блок синхронизации на основе генератора тактовых импульсов и интерфейс: шины адреса ABUS, данных DBUS и управления CBUS, магистрали и элементы интерфейса памяти с микропроцессором.

Обобщенная структура приведена на рисунке 1. В индивидуальных исходных данных указаны (и должны обозначаться на схеме) разрядность шин ABUS и DBUS, емкость ROM и RAM. Состав сигналов шины управления/синхронизации CBUS определяется разработчиком.

Рассмотрение способов взаимодействия элементов и формирование расширенных требований к ним включает следующие вопросы: организация памяти и выбор способа декодирования адресов, подходы к реализации синхронного обмена данными по магистрали и формированию сигналов управления, электрический аспект интерфейса с памятью.

Заданием предусмотрена линейная структура памяти. Использование микросхем памяти одинакового объема при удачном распределении адресного пространства разработчиком может дать в ряде случаев однотипные схемные решения.

Средства декодирования адресов обозначены на рисунке 1 как «ресный селектор» в составе ROM и RAM.

Незанятые памятью области адресного пространства следует считать резервом для функционального расширения микропроцессорной системы.

Рекомендуемый типовым заданием синхронный способ обмена требует формирования сигналов $R/\overline{W}$ управления чтением/записью в синхронные моменты времени.

При выполнении заданий, требующих применения динамической памяти в составе микропроцессорной системы, организации взаимодействия процессор – DRAM необходимо уделить особое внимание.

Как известно, для сохранения данных в динамической памяти обычно применяют алгоритм циклической регенерации, реализуемый контроллером DRAM (на рисунке 1 он не показан).

В циклах регенерации память недоступна для чтения–записи, что интерпретируют как захват магистрали памятью. Необходимость выполнения очередного цикла регенерации определяет таймер контроллера.

Если процессор не управляет регенерацией, то он взаимодействует с памятью наряду с контроллером регенерации, оба они претендуют на использование магистрали. Уровни приоритета их запросов доступа к памяти могут быть постоянными или динамически изменяются. Обменом и регенерацией управляет арбитр в соответствии с заданными алгоритмами обработки запросов и приоритетами, а процессор вынужден проверять, разрешает ли ему арбитр доступ к памяти, и работает с циклами ожидания. Этот универсальный способ осуществления обмена процессор – DRAM широко применяется и не противоречит требованиям задания. Однако его не следует использовать одновременно с синхронным (без квитирования) принципом обмена процессора со статической памятью (ROM или RAM), т.к. такое решение противоположно понятию и концепции интерфейса. В разрабатываемой микропроцессорной системе должен быть реализован единый принцип обмена информацией, унифицированный набор магистральных управляющих сигналов и единый способ синхронизации.

Для осуществления синхронного без квитирования принципа обмена процессор – DRAM есть необходимые предпосылки. При выполнении каждой машинной команды существует не менее одного цикла, в течение которого процессор или декодирует команду, или выполняет операцию над уже имеющимися данными и не обращается к памяти. О каждом цикле доступности магистрали процессор должен сообщить контроллеру DRAM (и другим устройствам гипотетической микропроцессорной системы) магистральным сигналом, разрешая бесконфликтную, или прозрачную регенерацию. Контроллер DRAM, получив данный сигнал, использует свободный цикл для регенерации, если это необходимо.

Анализ взаимодействий и взаимосвязей нужно продолжить, выявить некоторые детали, для того чтобы определить требования к интерфейсной части каждого структурного элемента модуля центрального процессора и детально описать сигналы магистрали. Учитывая важность этого вопроса, в пояснительной записке его следует выделить в самостоятельный подраздел.

7.2.2. Сигналы системной магистрали

Требования к типу выходного электрического интерфейса источников магистральных сигналов процессора рационально определять, полагая, что в составе гипотетической микропроцессорной системы имеется хоть одно устройство, выполняющее функцию управления обменом. Не затрагивая иные аспекты интерфейса, кроме электрического, отметим, что процессор, отдавая магистраль в распоряжение другого активного устройства, должен перевести магистральные буферы всех совместно используемых линий в высокоимпедансное состояние.

В активном состоянии магистральные формирователи должны обладать достаточно высокой нагрузочной способностью, а любой подключаемый к магистрали модуль должен создавать лимитированную разработчиком нагрузку на линии шин.

Сигналы магистрали целесообразно описать в табличной форме по аналогии с характеристикой сигналов стандартных интерфейсов: обозначение, название сигнала, тип выхода, активный уровень, функциональное назначение. Не исключено, что на последующих этапах разработки в исходный перечень магистральных сигналов понадобится внести поправки. В общем случае процессор имеет 10 линий шины управления. Каждая линия управления имеет свое функциональное назначение:

- RESET – начальная установка;
- SYNC – сигнал синхронизации по машинным циклам;
- DBIN – строб ввода данных;
- WR – строб вывода данных;
- READY – сигнал готовности периферийных модулей к обмену;
- WAIT – сигнал ожидания готовности;
- INTE – разрешение прерывания;
- INT – запрос векторного прерывания;
- HOLD – запрос доступа к магистрали;
- HLDA – подтверждение доступа к магистрали.

Для нормального функционирования микропроцессорной системы данных управляющих сигналов недостаточно. Так, в данном наборе сигналов нельзя отличить циклы обращения к памяти от циклов ввода–вывода. Расширение числа управляющих сигналов выполнено с помощью специального 8–разрядного слова состояния SW (Status Word). SW фиксируется в системном контроллере, и определяет тип текущего машинного цикла, в зависимости от которого формируется один из пяти управляющих стробов системной магистрали: MEMR, MEMW, I/OR, I/OW, INTA.

Последующие действия заключаются в систематизации и анализе требований к каждому структурному элементу модуля центрального процессора, на основе которых строятся их структурные схемы.

7.2.3. Замечания к разработке структурных схем памяти и блока синхронизации

Начальное представление о составе памяти, сформированное в ходе анализа способов обмена данными процессор–память, необходимо представить в виде структурной схемы с учетом требований задания. Этот материал будет использован после выбора элементной базы памяти.

Блок синхронизации, как относительно простое устройство, может быть описан, начиная с уровня принципиальных схем.

7.2.4. Разработка структурной схемы подсистемы ввода-вывода

Основные задачи этой части первого этапа – разработка схемно-ориентированной структуры подсистемы ввода-вывода микропроцессорной системы для заданного набора БИС, формирование требований к структурным элементам ввода-вывода микропроцессорной системы, не обеспеченным заданной элементной базой, и определение назначения внешних устройств.

Структурная схема подсистемы ввода-вывода, как и структура модуля центрального процессора, определяется на основе ее архитектурных характеристик с учетом структурно-алгоритмических особенностей заданных компонентов.

Проектирование структуры подсистемы ввода-вывода микропроцессорной системы осуществляется в такой последовательности:

- изучение литературы, анализ архитектурных характеристик заданных БИС;
- анализ исходных данных, устранение некорректности первичного описания подсистемы ввода-вывода микропроцессорной системы;
- определение состава подсистемы ввода-вывода микропроцессорной системы, предварительный выбор микросхем дешифрации внешних устройств, обслуживания и др., необходимых для компоновки структуры подсистемы ввода-вывода;
- компоновка структурной схемы подсистемы ввода-вывода микропроцессорной системы.

Решению данной проектной задачи предшествует основательная работа с литературой.

В общем случае в подсистеме ввода-вывода микропроцессорной системы можно выделить следующие устройства:

- К589ИР12 – многорежимный буферный регистр;
- КР580ВВ51/ВВ51А – программируемый связной адаптер;
- КР580ВИ53, 8254 – программируемый интервальный таймер;
- КР580ВВ55/ВВ55А – программируемый периферийный адаптер;
- КР580ВТ57 – контроллер прямого доступа к памяти;
- КР580ВН59 – программируемый контроллер прерываний;
- КР580ВВ79 – контроллер клавиатуры и дисплея;
- КР580ВГ75 – контроллер ЭЛТ;
- КР580ВК91А, КР580ВА93 – приемопередатчики шины IEEE-488.

Техника формирования структуры подсистемы ввода-вывода аналогична приемам, использованным для построения модуля центрального процессора микропроцессорной системы.

Взаимосвязи между выделенными структурными элементами подсистемы ввода-вывода определяются на основе анализа их функционального назначения и четкого представления о движении информационных и управляющих потоков.

Определяя конфигурацию подсистемы ввода-вывода, следует классифицировать ее внутренние шины и линии по функциональному назначению и рассматривать их организацию по функциональным группам: система шин адреса/данных; адреса портов ввода-вывода; шины, передающие статусные сигналы; линии синхронизации.

Систему шин адреса/данных подсистемы ввода–вывода предопределяет количество портов ввода/вывода и количество источников и приемников адреса/данных, существующее для периферийных устройств, включая элементы интерфейса.

Результаты данного раздела работ:

- структурная схема микропроцессора;
- задания на разработку принципиальных схем структурных элементов МП, не обеспеченных специализированными микропроцессорными компонентами, для следующего этапа проектирования.

7.3. Этап функционально-логического проектирования

7.3.1. Разработка принципиальной схемы микропроцессорной системы

Проектирование схемы микропроцессорной системы получения, обработки и отображения информации на данном этапе выполняется в такой последовательности:

- разработка фрагментов принципиальной схемы микропроцессорной системы по исходным данным (заданиям), полученным на предыдущем этапе (например, схем блока подсистемы ввода–вывода и т.п.);
- выбор элементной базы и разработка принципиальной схемы модуля центрального процессора микропроцессорной системы;
- разработка фрагментов принципиальной схемы подсистемы ввода–вывода по результатам предыдущего этапа;
- разработка фрагментов принципиальной схемы средств поддержки начальной установки, и элементов управления операциями получения и отображения информации.
- компоновка общей принципиальной схемы микропроцессорной системы получения, обработки и отображения информации;
- расчет характеристик электрических свойств системных магистралей.

Форма описания функций структурного элемента микропроцессорной системы, способа построения его функциональной схемы с применением неформальных инженерных приемов и/или методом формального синтеза выбирается разработчиком. Элементная база и варианты схемных решений отбираются, исходя из целей проектирования. При выборе элементной базы существенным моментом является согласование электрических параметров шинных формирователей и микросхем памяти.

Результатом этой части этапа являются:

- принципиальная электрическая схема микропроцессорной системы получения, обработки и отображения информации;
- результаты расчета электрических параметров микропроцессорной системы.

Все принципиальные электрические схемы оформляются по ГОСТ 2.702–75.

7.3.2. Замечания к разработке принципиальной схем памяти и подсистемы ввода–вывода

Подсистему статической оперативной (RAM) и постоянной (ROM, EPROM)

или EEPROM) памяти можно разрабатывать в такой последовательности:

- выбор микросхем памяти по заданной информационной емкости (организации). Допускается применять модули EEPROM и RAM зарубежных изготовителей;
- компоновка накопителя; выбор элементной базы и разработка принципиальной схем интерфейса памяти с микропроцессором, проверка выполнения условий работоспособности;
- распределение адресного пространства микропроцессорной системы совместно с построением схемы декодирования адресов и формирования сигналов выбора кристалла (страниц);
- Разработка DRAM отличается большей сложностью и может выполняться в такой очередности:
- выбор БИС/СБИС или модулей памяти зарубежных фирм-изготовителей в соответствии с требуемой емкостью и быстродействием;
- распределение адресного пространства, если оно не выполнено ранее, выбор схемы декодирования адресов;
- построение накопителя, если не применяется готовый модуль; выбор режима регенерации, построение номинальных временных диаграмм для режимов регенерации и обращения (с использованием справочных значений параметров);
- определение состава, выбор элементной базы, разработка функциональной и принципиальной схем контроллера DRAM, обеспечивающего номинальные временные диаграммы на шине накопителя во всех режимах;
- проверка выполнения условий работоспособности.

Завершающий шаг аналогичен последнему действию в процессе разработки статической памяти.

Применяя приборы DRAM высокого быстродействия, необходимо строить схемы формирования временной диаграммы сигналов для накопителя по синхронному принципу. Проектирование DRAM поддерживается справочной литературой и консультациями руководителя.

Разработка блока синхронизации оставляется разработчикам без комментария для самостоятельных изысканий.

При проектировании принципиальной схемы подсистемы ввода–вывода следует учесть, что периферийные БИС серии KP580 удовлетворяют электрическим и логическим спецификациям на микропроцессорную шину Microbus фирмы National Semiconductor – унифицированную 8-разрядную шину, объединяющую в функционально законченный модуль отдельные компоненты, размещенные в непосредственной близости друг от друга. В 1978 г. она была признана стандартным интерфейсом периферийных БИС (стандарт P696–3 IEEE), обеспечивающим совместимость уже существующих микропроцессорных кристаллов с новыми.

Основная задача проектирования подсистемы ввода–вывода на базе микропроцессора и периферийных БИС состоит в выборе необходимых сигналов магистралей и сопряжении с ней приборов. Функциональные требования, накладываемые на магистрали, сформулированы исходя из возможностей физических интерфейсов.

микропроцессора и периферийных БИС. Другие требования предъявляются к интерфейсам периферийных устройств, призванным расширить функциональные возможности микропроцессорных систем на аппаратном уровне. Попытка унифицировать эти требования привела к разработке ряда стандартных внутренних интерфейсов периферийных БИС, к числу которых относится и шина Microbus. В случае простейших микропроцессорных систем, состоящих из нескольких кристаллов, она может интерпретироваться как системная.

Двухнаправленная трехстабильная шина данных D7–D0 (Data Bus) используется для ввода–вывода данных, информации о состоянии и управляющих слов, организованных в 8-разрядные наборы. В простейших случаях эти линии могут быть одноподключенными.

Наличие напряжения низкого уровня на линии выбора кристалла CS (Chip Select) показывает, что производится обращение к данному прибору. Сигнал CS должен действовать в сочетании с другими сигналами, такими как строб чтения RD или записи WR. Так, при напряжении низкого уровня на входной линии RD (Read) разрешает прибору выдачу информации на шину D7–D0 при условии, что он выбран ($CS = 0$). Необходимо, чтобы на срезе стоба RD на линиях шины D7–D0 находились действительные данные. В случае невыполнения условия $CS=0$ передатчики шины данных прибора должны находиться в высокоимпедансном состоянии. Аналогично низкий уровень напряжения на входном контакте WR (Write) в совокупности с низким уровнем на входе CS дает возможность БИС принять данные из шины D7–D0. Для приборов первого поколения информация на шине данных D7–D0 должна быть истинной во время действия сигнала записи WR. БИС второго поколения требуют ее истинности только на срезе сигнала записи.

Во многих МС предполагается использование отдельных стробов чтения–записи при обращении к памяти – MEMR, MEMW и к портам ввода–вывода – I/OR, I/OW (изолированный ввод–вывод). В тех случаях, когда БИС содержит и память, и порты ввода–вывода, должны быть предусмотрены обе пары стробов чтения–записи.

Сигналы адресной шины A15–A0 применяются для адресации внутренних регистров БИС сопряжения. Сигналы адреса, поступающие на интерфейс, выбирают один из его внутренних регистров или ячеек памяти и в сочетании с входными сигналами CS, WR или RD определяют тип выполняемого обмена: ввода–вывод данных, ввод управляющей информации или вывод состояния БИС.

Выходной сигнал готовности к обмену READY предназначен для реализации асинхронного обмена по шине Microbus. Стробы RD и WR будут сняты только после появления активного уровня на линии READY, что гарантирует успешный обмен данными с медленно действующими интерфейсными БИС. В приборах с быстрым доступом ответный сигнал READY может отсутствовать.

Сигнал сброса RESET (допускается вариант прямого сигнала RESET) выдается микропроцессором при включении источника питания или нажатии кнопки сброса. При получении данного сигнала интерфейсная БИС устанавливается в исходное состояние, при этом шина данных БИС должна быть отключена от системной шины Microbus.

Перекрестные устройства часто выдают в систему запросы на прерывание IRQ0, IRQ1, ..., которые служат признаками программно не предсказуемых событий. Сигнал прерывания IRQ, после того как он выработан асинхронно по отноше-

нию к другим сигналам шины, должен оставаться в активном состоянии до тех пор, пока не будет обработан. Он снимается только по срезу соответствующего stroba RD или WR.

Микросистемы, построенные с расчетом на универсальную 8-разрядную шину Microbus, обеспечиваются средствами прямого сопряжения с большинством выпускаемых сегодня периферийных БИС, ориентированных на стандарт шины Microbus. Благодаря этому возможна дальнейшая модификация микропроцессорных систем, что очень важно для сложных устройств.

Сигналы магистрали CB модуля центрального процессора микропроцессорной системы не обеспечивают систему полным набором сигналов шины Microbus. Отсутствуют такие важные для системы линии, как линии выбора кристалла, линии запросов на прерывание, а также набор сигналов прямого доступа к памяти.

Сигналы CS обычно формируются из комбинации старших разрядов адресной шины A15–A0 логикой выбора кристалла, которая представляет собой комбинационный дешифратор и легко реализуется с помощью ПЗУ, ПЛМ или логических схем малой степени интеграции. Для получения оставшихся сигналов шины Microbus в микропроцессорную систему необходимо ввести контроллер обработки прерываний и контроллер ПДП, реализуемые с помощью БИС KP580BH59 и KP580BT57 соответственно.

При подключении к периферийным БИС датчиков, индикаторов, управляющих элементов и т.п., в случае недостатка линий ввода–вывода следует воспользоваться логическими схемами малой степени интеграции – дешифраторами и мультиплексорами. Для подключения исполнительных устройств большой мощности (реле, ЭПК и т.п.) необходимо использовать устройства гальванической развязки.

7.4. Разработка алгоритмов управляющей программы

Если целевая функция микропроцессорной системы сформулирована, т.е. задача на разработку поставлена, то для получения текста исходной программы необходимо выполнить ряд последовательных действий:

- 1) подробное описание задачи;
- 2) анализ задачи;
- 3) инженерную интерпретацию задачи, желательно с привлечением того или иного аппарата формализации (граф автомата, сети Петри, матрицы состояний и связности и т.п.);
- 4) разработку общей блок-схемы алгоритма работы микропроцессорной системы;
- 5) разработку детализированных блок-схем отдельных процедур, выделенных на основе модульного принципа составления программ;
- 6) детальную проработку интерфейса и внесение исправлений в общую и детализированные блок-схемы;
- 7) распределение рабочих регистров и памяти микропроцессора;
- 8) формирование текста исходной программы.

В результате работы по трем первым пунктам данного перечня получают так называемую функциональную спецификацию прикладной программы, в которой основное внимание уделяется детализации способов формирования входной и выходной информации.

На языке схем алгоритмов разработчик описывает метод, выбранный для решения поставленной задачи. Часто бывает, что одна и та же задача может быть решена различными методами. Способ решения задачи, выбранный на этапе ее инженерной интерпретации, на основе которого формируется блок-схема алгоритма, определяет не только качество разрабатываемой прикладной программы, но и качественные показатели конечного продукта.

В основу разработки блок-схема алгоритма положена процедура модульного проектирования.

Так как алгоритм есть точно определенная процедура, предписывающая микропроцессорной системе однозначно определенные действия по преобразованию исходных данных в обработанные выходные данные, то разработка блок-схемы алгоритма требует предельной точности и однозначности используемой атрибутики: символических имен переменных, констант, подпрограмм, символических адресов таблиц, портов ввода-вывода и т.п. Основное внимание при разработке блок-схемы следует уделить тому разделу функциональной спецификации прикладной программы, в котором приводится описание аппаратуры сопряжения микропроцессорной системы с объектом получения информации. (Это описание должно быть детализировано вплоть до электрических и временных характеристик каждого входного и выходного сигнала или устройства.)

Разработка прикладной программы должна осуществляться с использованием метода декомпозиции, при котором вся задача последовательно разделяется на меньшие функциональные модули, каждый из которых можно анализировать, разрабатывать и отлаживать отдельно от других. При выполнении прикладной программы в микропроцессорной системе управление передается от одного функционального модуля к другому. Схема связности этих функциональных модулей, каждый из которых реализует некоторую процедуру, образует общую (или системную) блок-схема алгоритма прикладной программы. Это разделение задачи на модули выполняется последовательно до такого уровня, когда разработка блок-схемы модуля становится простым и понятным. Метод последовательной декомпозиции обладает достаточной гибкостью, что позволяет привести степень детализации блок-схема алгоритма в соответствие со сложностью процедуры. Язык графических образов блок-схемы можно использовать на любом уровне детализации описания модулей вплоть до того, что каждому оператору блок-схемы будет соответствовать единственная команда микропроцессора.

Структурное программирование есть процесс построения прикладной программы из строгого набора программных модулей, каждый из которых реализует определенную процедуру обработки данных. Программные модули должны иметь только одну точку входа и одну точку выхода.

Разработка блок-схемы функционального модуля программы имеет ярко выраженный итеративный характер, т.е. требует многократных проб, прежде чем возникает уверенность, что алгоритм реализации процедуры правильный и заверченный. Вне зависимости от функционального назначения процедуры при разработке ее блок-схемы необходимо придерживаться следующей очередности работы:

1. Определить, что должен делать модуль (это уже было сделано при разработке системной блок-схемы алгоритма, но теперь разработчик имеет дело с фрагментом прикладной программы, а не с целой программой, и, следовательно, может

потребоваться доопределение и уточнение целевого назначения процедуры).

2. Определить способы получения модулем исходных данных (от датчиков через порты ввода, или из таблиц в памяти, или через рабочие регистры). Для реализации ввода исходных данных в модуль в его блок-схемы надо включить соответствующие операторы.

3. Определить необходимость какой-либо предварительной обработки введенных исходных данных (маскирование, сдвиг, масштабирование, перекодировка). Если до использования данных требуется их предобработка, то в состав блок-схемы включаются соответствующие операторы.

4. Определить способ преобразования входных данных в требуемые выходные. Используя операторы процедур и условные операторы, отобразить на языке блок-схем выбранный метод содержательной обработки исходных данных.

5. Определить способы выдачи из модуля обработанных данных (передать в память, или в вызывавшую программу, или в порты вывода информации). Необходимые действия отобразить в блок-схеме.

6. Определить необходимость какой-либо постобработки выводимых данных (изменение формата, перекодирование, масштабирование, маскирование). Ввести в блок-схему операторы подготовки данных для вывода из модуля.

7. Вернуться к пункту 1 перечня и проанализировать полученный результат. Выполнить итеративную корректировку блок-схемы с целью сделать ее простой, логичной, стройной и обладающей четким графическим образом.

8. Проверить работоспособность алгоритма на бумаге путем подстановки в него действительных данных.

9. Рассмотреть предельные случаи и попытаться определить границы значения информационных объектов, с которыми оперирует алгоритм и за пределами которых он теряет свойства конечности, сходимости и результативности.

10. Провести мысленный эксперимент по определению работоспособности алгоритма в реальном масштабе времени, когда стохастические события, происходящие в объекте управления, могут оказать влияние на работу алгоритма. При этом самому тщательному анализу следует подвергнуть реакцию алгоритма на возможные прерывания с целью определения критических операторов, которые необходимо защитить от прерываний. Кроме того, в ходе этого мысленного эксперимента следует проанализировать логику алгоритма с целью определения таких последовательностей операторов, при выполнении которых микропроцессор может пропустить кратковременные события в объекте получения информации. При обнаружении таких ситуаций в логику блок-схемы следует внести коррективы.

Практика разработки программного обеспечения для микропроцессорных систем показала, что последовательное использование описанной поэтапной процедуры проектирования алгоритмов, составляющей основу метода структурного программирования, позволяет получать работоспособные прикладные программы.

Преобразование разработанной блок-схемы в исходный текст программы является тривиальным. Но прежде чем приступить к написанию программы, необходимо специфицировать память.

Спецификация памяти (и рабочих регистров) заключается в определении адреса первой команды прикладной программы, действительных начальных адресов стека, таблиц данных, буферных зон передачи параметров между процедурами, подпрограмм обслуживания прерываний и т.д. При этом следует помнить, что в микропроцессоре память программ и память данных могут быть физически и логически разделены.

Большинство прикладных задач получения информации и управления объектами вследствие того, что они должны решаться в реальном времени, предъявляет столь высокие требования по быстродействию, что для их решения основным языковым средством написания прикладных программ будет язык ассемблера. Это положение о преимущественном использовании языка ассемблера подкрепляется и тем обстоятельством, что микропроцессоры имеют ограниченный объем памяти и, следовательно, критичны к длине прикладных программ.

В данной курсовой работе, исходя из разработанной принципиальной схемы микропроцессорной системы и функций, которые на нее возлагаются, необходимо разработать укрупненную блок-схему (граф) управляющей программы. В блок-схеме должны быть предусмотрены процедуры начальной инициализации системы, взаимодействие с устройствами управления ходом получения и вывода информации, процедура обработки информации и реакции на внештатные ситуации.

Для одной из процедур, заданной руководителем, необходимо разработать временную диаграмму (циклограмму) работы системы и программу на ассемблере.

Результаты данного раздела работы:

- укрупненная блок-схема (граф) управляющей программы;
- временная диаграмма (циклограмма) заданной процедуры;
- подробную блок-схему процедуры;
- текст программы на ассемблере для указанной процедуры.

8. ПОРЯДОК ПРОВЕРКИ И ЗАЩИТЫ КУРСОВЫХ РАБОТ. КРИТЕРИИ ОЦЕНКИ

Курсовая работа сдается на проверку в установленный графиком самостоятельной работы срок за два-три дня до ее защиты комиссии из двух преподавателей.

С письменными замечаниями руководителя студент может ознакомиться накануне или за два-три часа до защиты.

Проекты, представляющие собой одновариантные решения, описанные как данность, и работы, содержащие принципиальные ошибки, возвращаются на доработку. Доработке подлежат также КР с большим количеством непринципиальных разнотипных ошибок в схемах, в расчетах и в иллюстрациях.

Результаты проектирования и их защиты оцениваются по четырехбалльной системе.

Итоговая оценка определяется в соответствии с требованиями Государственного образовательного стандарта к знаниям, умениям и навыкам, с целями и задачами курсовой работы, с требованиями к пояснительной записке, схемам и является мерой степени достижения студентами двух целей КР.

Оценка “отлично” ставится за доброкачественную работу, выполненную в полном объеме в срок, и грамотные ответы на возникающие в ходе защиты вопросы. Допускаются две-три неточности в пояснительной записке и в схемах.

Критерии снижения оценки относительно высшего балла:

- непринципиальные ошибки в схемах, в расчетах, нерациональный выбор пути или варианта решения – до 1.25 балла в зависимости от характера и количества ошибок;
- неумение письменно логично и ясно изложить результаты проектирования с использованием профессиональной лексики и лаконичных форм описания (таблиц, временных диаграмм и т.п.), плохое структурирование материала, отсутствие в работе списка литературных источников и другие подобные недочеты – до 0.5 балла (суммарно);
- нарушение правил, установленных ЕСКД – до 1 балла.

Поощряется не являющееся обязательным модельное исследование оригинального функционального узла микропроцессорной системы повышением оценки на 0.25–1.0 балл в зависимости от сложности, объема и качества этой части работы.

При определении итоговой оценки учитываются не только особенности и количество допущенных в работе ошибок, но и уровень знаний и эрудиция студента, показанные им на защите, аргументированность ответов на вопросы и замечания и умение найти правильное решение взамен ошибочного или неудачного.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Васильев В.И. Электронные промышленные устройства / В.И. Васильев, Ю.М. Гусев, В.Н. Миронов – М.: Высш. шк., 1988. – 287 с.
2. Янсен И. Курс цифровой электроники. Микрокомпьютеры / И. Янсен – М.: Мир, 1987 – 406 с.
3. Самофалов К.Г. Микропроцессоры / К.Г. Самофалов – К.: Техника, 1986. – 278 с.
4. Каффрон Дж. Расширение микропроцессорных систем / Дж. Каффрон, В. Лонг – М.: Машиностроение, 1987. – 320 с.
5. Хвощ С.Т. Микропроцессоры и микро-ЭВМ в системах автоматического управления / С.Т. Хвощ, Н.Н. Варлинский, Е.А. Попов; под общ. ред. С.Т.Хвоща. – Л.: Машиностроение, Ленинградское отделение, 1987. – 640 с.
6. Щелкунов Н.Н. Микропроцессорные средства и системы / Н.Н. Щелкунов, А.П. Дианов – М.: Радио и связь, 1989. – 288 с.
7. Безуглов Д. А. Цифровые устройства и микропроцессоры / Д. А. Безуглов, И. В. Калиенко – М.: Феникс, 2008. – 480 с.
8. А. К. Нарышкин Цифровые устройства и микропроцессоры / М.: Академия, 2008. – 320 с.
9. Басманов А.С. Микропроцессоры и однокристальные микро-ЭВМ. Номенклатура и функциональные возможности / Басманов А.С., Широков Ю.Ф.; под ред. В.Г. Дормачева. – М.: Энергоатомиздат, 1988. – 128 с.
10. Федорков Б.Г. Микросхемы ЦАП и АЦП: функционирование, параметры, применение / Б.Г. Федорков, В.А. Телец – М.: Энергоатомиздат, 1990 – 320 с.

Заказ № _____ от « ____ » _____ 2011. Тираж _____ экз.

Изд-во СевНТУ