ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

КОНТРОЛЬ И ДИАГНОСТИКА СЛОЖНЫХ СИСТЕМ

Методические рекомендации к лабораторным работам для студентов направления подготовки 09.03.01 «Информатика и вычислительная техника» дневной формы обучения

Могилев 2018

УДК 004.2
ББК 32.973.202
К 65

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Автоматизированные системы управления» «4» сентября 2018 г., протокол № 2

Составитель канд. техн. наук, доц. А. В. Кушнер

Рецензент канд. техн. наук, доц. М. Н. Миронова

В методических рекомендациях кратко изложены теоретические сведения, необходимые для лабораторных работ. Рекомендации составлены в соответствии с рабочей программой по дисциплине «Контроль и диагностика сложных систем» для направления подготовки 09.03.01 «Информатика и вычислительная техника».

Учебно-методическое издание

КОНТРОЛЬ И ДИАГНОСТИКА СЛОЖНЫХ СИСТЕМ

Ответственный за выпуск	С. С. Сергеев
Технический редактор	С. Н. Красовская
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 16 экз. Заказ №

Издатель и полиграфическое исполнение:
Государственное учреждение высшего профессионального образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 24.01.2014.
Пр. Мира, 43, 212000, Могилев.

Содержание

Инструкция по охране труда при проведении лабораторных работ

1 Для работы на ПЭВМ в компьютерном классе допускаются студенты, прошедшие обучение по мерам безопасности, проверку знаний с оформлением протокола.

2 Студенты допускаются в класс только в отведенное для них время, в соответствии с графиком работы класса.

3 Студенты должны соблюдать правила внутреннего распорядка. Не допускается находиться в классах в верхней одежде, в состоянии алкогольного, токсического и наркотического опьянения и курения.

4 Не допускается находиться в классе посторонним лицам.

5 При эксплуатации ПЭВМ необходимо помнить, что питающее напряжение сети 220 В является опасным для жизни человека. В связи с этим необходимо соблюдать правила техники безопасности при работе с высоким напряжением.

6 Не допускается проведение ремонта ПЭВМ непосредственно в компьютерном классе.

7 В компьютерном классе должен быть уголок охраны труда, укомплектованный огнетушителем и аптечкой.

8 Студенты при работе на ПЭВМ должны соблюдать правила пожарной безопасности, уметь пользоваться углекислотным огнетушителем.

9 О любых фактах травмирования необходимо незамедлительно сообщить преподавателю, проводящему занятия в классе, лаборатории, аудитории.

10 Каждый студент, должен уметь оказывать первую (доврачебную) медицинскую помощь потерпевшим при несчастном случае.

11 О неисправностях оборудования студент должен сообщить преподавателю или обслуживающему персоналу.

12 Студенты на ПЭВМ должны соблюдать требования личной гигиены.

13 Перед началом работы студенту необходимо:

– убедиться в том, что закрыты все крышки и кожухи устройств, входящих в ПЭВМ;

– убедиться в том, что изоляция электрических проводов не имеет видимых повреждений.

14 При включении ПЭВМ студент обязан соблюдать следующую последовательность включения оборудования:

– включается блок питания (если он имеется);

– включаются периферийные устройства: принтер, монитор, сканер и др.;

– включается системный блок (процессор);

– при необходимости включается местное освещение.

15 Запрещается приступать к работе при:

– обнаружении неисправности оборудования;

– поврежденных кабелях электропитания;

– отсутствии заземляющего устройства ПЭВМ;

– отсутствии защитного фильтра;

– разобранном системном блоке (процессоре).

16 Студент в компьютерном классе обязан:

– производить работу на ПЭВМ только в присутствии инженера или преподавателя, проводящего занятия в классе;

– выполнять только ту работу, которая ему поручена и по которой он проинструктирован;

– содержать рабочее место в порядке и чистоте в течение всего занятия;

– соблюдать правила эксплуатации оборудования;

– соблюдать санитарные правила и нормы;

– при работе соблюдать рекомендуемое расстояние от экрана монитора до глаз (60…70 см).

17 Длительность работы с компьютером не должна превышать у студентов вузов первых трех курсов – не более трех часов в день, у старшекурсников – не более четырех часов.

18 Для предупреждения развития переутомления при работе с ВДТ и ПЭВМ необходимо осуществлять комплекс профилактических мероприятий:

– устраивать перерывы после каждого академического часа занятий, независимо от учебного процесса, длительностью не менее 10 мин;

– проводить во время перерывов сквозное проветривание компьютерного класса с обязательным выходом из него;

– проводить упражнения для глаз через каждые 20…25 мин работы на ВДТ и ПЭВМ. При появлении зрительного дискомфорта, выражающегося в быстром развитии усталости глаз, рези, мелькании точек перед глазами и т. п., упражнения для глаз проводятся индивидуально, самостоятельно и раньше указанного времени.

19 Студенту при работе на ПЭВМ запрещается:

– прикасаться к задней стенке системного блока (процессора) при включенном питании;

– производить какие-либо переключения любого периферийного оборудования;

– загромождать верхние панели устройств ненужными бумагами и посторонними предметами;

– допускать попадание влаги на поверхность системного блока, монитора, рабочую поверхность клавиатуры, дисковода, принтера и других устройств;

– проводить самостоятельно вскрытие и ремонт оборудования.

20 При обнаружении обрыва проводов питания, заземления и других повреждений электрооборудования, появления запаха гари, возникновении необычного шума немедленно отключить питание и сообщить о случившемся непосредственному руководителю или лицу, осуществляющему техническое обслуживание оборудования.

21 При возгорании оборудования отключить питание и сообщить о случившемся своему преподавателю, с последующим вызовом пожарной службы

по телефонц 101 и принятием мер к тушению очага возгорания с помощью углекислотного огнетушителя.

22 В случае поражения электрическим током необходимо:

– немедленно освободить пострадавшего от воздействия тока;

– уложить его ровно и удобно на твердую поверхность;

– расстегнуть и расслабить одежду;

– создать приток свежего воздуха;

– проверить наличие дыхания, пульс, состояние зрачка (широкий зрачок указывает на резкое ухудшение кровоснабжения мозга);

– вызвать врача по телефону 103;

– если пострадавший находится в сознании – обеспечить ему до прибытия врача полный покой, наблюдая за пульсом и дыханием;

– если пострадавший без сознания – давать ему нюхать нашатырный спирт и при необходимости проводить искусственное дыхание и закрытый массаж сердца.

23 По окончаний работы пользователь обязан:

– произвести закрытие всех активных задач;

– извлечь сменные носители (гибкий или компакт диск, USB накопитель);

– отключить питание системного блока (процессора);

– отключить питание всех периферийных устройств;

– отключить блок питания (если он имеется);

– осмотреть и привести в порядок рабочее место.

1 Подготовка исходных данных и формирование структурных моделей для интегральных схем комбинационного и последовательностного типов

Цель работы: изучить методы подготовки данных и формирование исходных данных для построения моделей интегральных схем. Научиться постановке задачи тестирования схем. Ознакомиться с приемами документирования.

Приборы и оборудование

Персональный компьютер, среда программирования C++, C#, Java.

Общая часть

Для автоматизации тестирования схем вычислительных и цифровых устройств необходимо иметь возможность смоделировать данные устройства и проводить исследования на модели, применяя мощный аппарат вычислительных средств и алгоритмов.

Все цифровые и вычислительные устройства состоят из простых элементов ограниченного ассортимента. Имея модели элементов, мы можем, группируя

элементарные модели, сгенерировать модель устройства. Создавая модели, мы должны упростить реальные элементы, абстрагируясь от некоторых свойств и возможностей элементов, которые не важны или не окажут влияния на поведение модели на требуемом уровне абстракции модели. Кроме построения модели, часто бывает необходимо построить графическую модель элементов, позволяющую документировать процесс тестирования, а иногда и визуально отображать и управлять процессом тестирования.

Порядок выполнения работы

1 Проанализировать выданную преподавателем схему. Выявить необходимый набор элементов.

2 Разделить набор элементов на последовательностные и комбинационные типы элементов.

3 Определить возможные типы отказов для всех элементов.

4 Поставить задачу для диагностики всех видов отказов.

5 Поставить задачу для моделирования всех элементов с возможностью эмуляции всех возможных ошибок.

6 Создать графические модели элементов для документирования.

Контрольные вопросы

1 В чем отличие схем комбинационного и последовательностного типов?

2 В чем заключается процесс моделирования?

3 Каков состав и назначение «постановки задачи»?

2 Подготовка исходных данных для построения функциональных моделей интегральных схем комбинационного и последовательностного типов. Разработка алгоритма и составление программы

Цель работы: освоить приемы построения функциональных моделей интегральных схем комбинационного и последовательностного типов.

Приборы и оборудование

Персональный компьютер, среда программирования C++, C#, Java.

Общая часть

Для автоматизации тестирования схем вычислительных и цифровых устройств необходимо иметь возможность смоделировать данные устройства и проводить исследования на модели, применяя мощный аппарат вычислительных средств и алгоритмов. Все цифровые и вычислительные устройства состоят

из простых элементов ограниченного ассортимента. Имея модели элементов, мы можем, группируя элементарные модели, сгенерировать модель устройства. Создавая модели, мы должны упростить реальные элементы, абстрагируясь от некоторых свойств и возможностей элементов, которые не важны или не окажут влияния на поведение модели на требуемом уровне абстракции модели.

При построении функциональной модели элементы обычно идеализируют во временной и амплитудной областях. Логические элементы могут выдавать только идеальные нулевые и единичные сигналы, причем, если это не оговорено особо, все сигналы вырабатываются одновременно, т. е. элементы работают в едином кванте времени. Для получения возможности функционального моделирования каждый элемент должен иметь только одно выходное значение, т. е. реальный элемент может быть смоделирован группой функциональных моделей.

Порядок выполнения работы

1 На основе построенных ранее моделей разработать алгоритм программы, моделирующей работу элементов схемы.

2 Построить программу, моделирующую работу элементов схемы с возможностью эмуляции возможных отказов.

3 Проверить правильность работы программ на тестовом примере, составленном самостоятельно.

Контрольные вопросы

1 Как определить возможные отказы устройства?

2 В чем особенности модели комбинационных и последовательностных устройств?

3 Почему так велико различие в программах, моделирующих комбинационные и последовательностные устройства?

4 По какому принципу Вы составили тестовый пример?

3 Подготовка тестов и моделирование моделей для интегральных схем комбинационного и последовательностного типов

Цель работы: освоить приемы построения моделей интегральных схем комбинационного и последовательностного типов и составление программ моделирования процесса тестирования и эмуляции неисправностей.

Приборы и оборудование

Персональный компьютер, среда программирования C++, C#, Java.

Общая часть

Для автоматизации тестирования схем вычислительных и цифровых устройств необходимо иметь возможность смоделировать данные устройства и проводить исследования на модели, применяя мощный аппарат вычислительных средств и алгоритмов. Все цифровые и вычислительные устройства состоят из простых элементов ограниченного ассортимента. Имея модели элементов, мы можем, группируя элементарные модели, сгенерировать модель устройства. Создавая модели, мы должны упростить реальные элементы, абстрагируясь от некоторых свойств и возможностей элементов, которые не важны или не окажут влияния на поведение модели на требуемом уровне абстракции модели.

При построении функциональной модели, элементы обычно идеализируют во временной и амплитудной областях. Логические элементы могут выдавать только идеальные нулевые и единичные сигналы, причем, если это не оговорено особо, все сигналы вырабатываются одновременно, т. е. элементы работает в едином кванте времени. Для получения возможности функционального моделирования каждый элемент должен иметь только одно выходное значение, т .е. реальный элемент может быть смоделирован группой функциональных моделей.

Порядок выполнения работы

1 Разработать программу, моделирующую работу схемы с возможностью эмуляции возможных ошибок.

2 Разработать алгоритм ручного построения дерева тестов для автоматического тестирования.

3 Создать программу формирования данных для ручного построения дерева тестов для автоматического тестирования.

4 Проверить работу программы на тестовых примерах с неисправностями, созданными самостоятельно.

Контрольные вопросы

1 Каков порядок выполнения компонентов, моделирующих работу элементов схемы?

2 По какому принципу Вы составили тестовый пример?

3 Является ли полученное Вами дерево оптимальным? Почему?

4 Проверка правильности функционирования программ, описывающих работу интегральных схем комбинационного и последовательностного типов

Цель работы: освоить приемы построения моделей схем и составление программ моделирования процесса тестирования и эмуляции неисправностей. Освоить методы автоматического построения алгоритмов автоматического тестирования.

Приборы и оборудование

Персональный компьютер, среда программирования C++, C#, Java.

Общая часть

Для автоматизации тестирования схем вычислительных и цифровых устройств необходимо иметь возможность смоделировать данные устройства и проводить исследования на модели, применяя мощный аппарат вычислительных средств и алгоритмов. Все цифровые и вычислительные устройства состоят из простых элементов ограниченного ассортимента. Имея модели элементов, мы можем, группируя элементарные модели, сгенерировать модель устройства. Создавая модели, мы должны упростить реальные элементы, абстрагируясь от некоторых свойств и возможностей элементов, которые не важны или не окажут влияния на поведение модели на требуемом уровне абстракции модели.

При построении функциональной модели, элементы обычно идеализируют во временной и амплитудной областях. Логические элементы могут выдавать только идеальные нулевые и единичные сигналы, причем, если это не оговорено особо, все сигналы вырабатываются одновременно, т. е. элементы работает в едином кванте времени. Для получения возможности функционального моделирования каждый элемент должен иметь только одно выходное значение, т. е. реальный элемент может быть смоделирован группой функциональных моделей.

Порядок выполнения работы

1 Разработать алгоритм автоматического построения дерева тестов для автоматического тестирования.

2 Разработать алгоритм формирования данных для автоматического построения дерева тестов для автоматического тестирования.

3 Создать программу автоматического построения дерева тестов для автоматического тестирования.

4 Протестировать работу полученных программ.

Контрольные вопросы

1 Для любой ли схемы можно составить программу автоматического построения дерева тестов для автоматического тестирования?
2 Можно ли применить полученное дерево для ручного тестирования?
3 Является ли полученное Вами дерево оптимальным? Почему?
4 По какому принципу Вы составили тестовый пример?

5 Моделирование логического анализатора

Цель работы: изучить работу логического анализатора. Освоить принципы записи данных и уменьшения их объема при неизменности записываемых сигналов. Освоить принципы запуска и останова логического анализатора.

Приборы и оборудование

Персональный компьютер, среда программирования C++, C#, Java.

Общая часть

Логический анализатор – это устройство, записывающее значения входных сигналов в течение заданного периода времени с целью их дальнейшего анализа, который может производиться либо оператором, либо самим устройством, сравнивая сигналы с эталонными.

Логический анализатор содержит анализатор входных сигналов, производящий преобразование сигналов для их последующей записи, устройства памяти, сохраняющей выборки сигналов, выходного интерфейса, преобразующего выборки к виду, который может быть воспринят внешними устройствами или оператором и устройства синхронизации, управляющего запуском и остановом логического анализатора.

Основным лимитирующим фактором является объем памяти, который возрастает при увеличении времени регистрации и разрешающей способности. Решением проблемы может являться применение адаптивных логических анализаторов, которые записывают выборки не в равномерно распределенные периоды времени, а только в моменты времени, в которых входные сигналы изменяют свои значения. При этом требуется, естественно, записывать и значение времени.

Порядок выполнения работы

1 Разработать алгоритм работы логического анализатора.
2 Разработать алгоритм работы адаптивного логического анализатора.

3 Разработать программу, моделирующую логический анализатор.

4 Разработать программу запуска анализатора по сигналу, изменению сигнала, кодовому слову, последовательности кодовых слов.

Контрольные вопросы

1 В чем заключается сущность адаптивного логического анализатора?

2 В каких случаях оправдано применение адаптивного логического анализатора?

3 Приведите пример применения останова логического анализатора по кодовому слову (последовательности кодовых слов).

6 Построение схемы сигнатурного анализатора. Выбор функции получения сигнатуры

Цель работы: изучить принцип работы сигнатурного анализатора. Научиться выбору и построению функций перемешивания, оценке их сложности и возможности применения при сигнатурном анализе. Ознакомиться с методами упрощения функций перемешивания при сохранении допустимого качества.

Приборы и оборудование

Персональный компьютер, среда программирования C++, C#, Java.

Общая часть

Сигнатурный анализ основан на получении значения функции входных переменных и внутреннего состояния анализатора в предыдущий момент времени.

Сигнатурный анализатор является последовательным устройством, т. е. процесс анализа занимает несколько циклов, в каждом из которых используется свой набор входных переменных, а состоянием устройства является состояние анализатора в предыдущий момент.

К функции предъявляется требование: функция должна иметь различные значения для различных состояний тестируемого устройства в равных условиях. Так как функция должна быть реализована аппаратно, она должна быть максимально проста. Противоречие разрешается упрощением функции с одновременным увеличением числа итераций.

Порядок выполнения работы

1 Оценка тестируемого устройства, определение требуемой сложности функции перемешивания.

2 Выбор функции перемешивания, оценка возможности ее аппаратной реа-

лизации и достаточности для тестирования.

3 Построение функциональной схемы сигнатурного анализатора.

Контрольные вопросы

1 На чем основано применение сигнатурного анализа?

2 Какие требования предъявляются к функции перемешивания?

3 Чем достигается компромисс между требуемым качеством перемешивания и возможностью аппаратной реализации?

4 Возможно ли построение универсального сигнатурного анализатора?

7 Моделирование сигнатурного анализатора

Цель работы: на основе моделей элементов схем построить модель сигнатурного анализатора. Научиться построению таблиц сигнатур и приемам работы с сигнатурным анализатором при диагностике устройств.

Приборы и оборудование

Персональный компьютер, среда программирования C++, C#, Java.

Общая часть

Сигнатурный анализ основан на получении значения функции входных переменных и внутреннего состояния анализатора в предыдущий момент времени.

Сигнатурный анализатор является последовательным устройством, т. е. процесс анализа занимает несколько циклов, в каждом из которых используется свой набор входных переменных, а состоянием устройства является состояние анализатора в предыдущий момент.

К функции предъявляется требование: функция должна иметь различные значения для различных состояний тестируемого устройства в равных условиях. Так как функция должна быть реализована аппаратно, она должна быть максимально проста. Противоречие разрешается упрощением функции с одновременным увеличением числа итераций.

Порядок выполнения работы

1 Разработать схему сигнатурного анализатора.

2 Построить таблицу сигнатур для всех возможных состояний модели диагностируемого устройства.

3 Проверить работу анализатора, эмулируя отказы в диагностируемом устройстве.

Контрольные вопросы

1 Сравните схемы логического и сигнатурного анализаторов. В чем их сходство и различие?

2 Чем обусловлен малый, по сравнению с логическим анализатором, требуемый объем памяти сигнатурного анализатора?

3 Каков, по Вашему мнению, минимально необходимый объем памяти?

4 Почему, несмотря на удобство и малые аппаратные затраты, сигнатурный анализ редко употребляется в реальных условиях?

8 Проектирование тестопригодных схем

Цель работы: освоить понятия тестопригодности и глубины тестирования. Научиться приемам построения тестопригодных схем заданной глубины тестирования и расстановки контрольных точек.

Приборы и оборудование

Персональный компьютер, среда программирования C++, C#, Java.

Общая часть

Тестопригодность – приспособленность схемы к обнаружению неисправностей, их локализации и реализации тестового диагностирования в пределах допустимых финансовых затрат. Нужно сокращать затраты в любом из перечисленных разделов: на ЭВМ, программы, время тестирования, обучения, обслуживания и т. д.

Различают три подхода к проектированию тестопригодных схем:

1) численная оценка управляемости и наблюдаемости, количественная мера тестопригодности;

2) структурное проектирование (иногда самотестируемых) схем с использованием свойств сканируемого пути с простотой доступа к внутренним точкам устройства. Эффективные методы генерации тестов, реализующих методы сканирования;

3) разработка практических указаний для сокращения затрат на процедуры генерации тестов и реализацию тестирования.

Порядок выполнения работы

1 Проанализировать задание, выданное преподавателем.

2 Разработать структурную схему устройства.

3 Оценить тестопригодность разработанного устройства.

4 Оценить глубину тестирования, возможную для данной схемы.

5 Расставить контрольные точки и разработать алгоритм диагностирования устройства.

6 Дать рекомендации по изменению схемы для изменения глубины тестирования.

Контрольные вопросы

1 Каковы требования к тестопригодности схем?
2 Что такое глубина тестирования?
3 Зависит ли тестопригодность от глубины тестирования?
4 Возможны ли абсолютно нетестопригодные схемы?

9 Контроль и тестирование вычислительных систем

Цель работы: научиться определять круг возможных неисправностей по функциональной схеме вычислительного устройства. Ознакомиться с методами ускоренного тестирования.

Приборы и оборудование

Персональный компьютер, среда программирования C++, C#, Java.

Общая часть

Оценочные и отладочные комплексы.

Предназначены для отладки микропроцессорных систем на программном уровне.

Оценочные комплексы – микро-ЭВМ в минимальном режиме, на базе которого создается микропроцессорная система с подключаемой клавиатурой и дисплеем, а также с возможностью подключения аппаратуры пользователя. Оценочные комплексы являются хорошим средством обучения и оценки возможностей микропроцессора, средством для макетирования, дают возможность выполнения программ в РВ и на реальном МП, но не дают возможности генерировать ПО, занимают ресурсы систем, не позволяют собирать информацию о поведении и управлении поведением систем в реальном времени.

Отладочные комплексы отличаются от оценочных развитым ПО, увеличенной емкостью памяти и усложненным интерфейсом. Изготавливаются в виде конструктивно-законченного блока, объединяющего несколько плат и имеющего пульт.

Их использование дает возможность программирования на языке Ассемблера и языках высокого уровня, высокий набор внешних устройств, в том числе и дискретного накопителя, наличие развитой ОС. Так же, как в оценочных

комплексах, основой системы является микро-ЭВМ на элементарной базе, используемой в проектируемой или тестируемой системе.

Отладочные комплексы часто имеют интерфейсы для подключения внешних устройств, однако сами устройства не включаются в состав комплекса.

Комплексы развития.

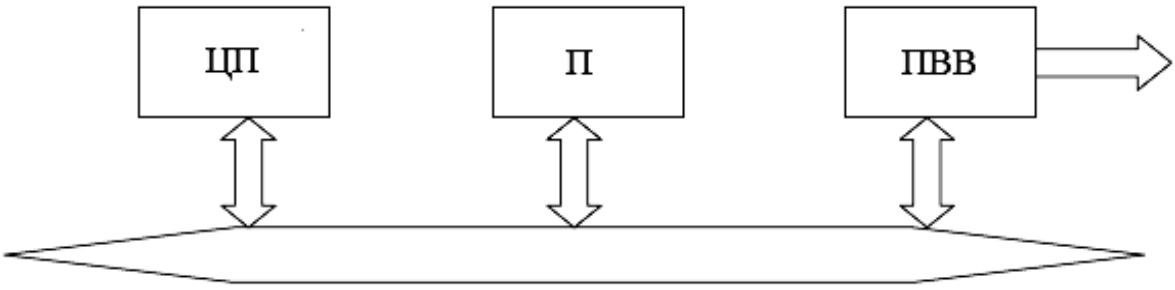
Предназначены для отладки микропроцессорных систем и позволяют управлять поведением систем, собирать информацию о поведении системы, моделировать (эмулировать) недостающие устройства в режиме РВ или близкого к нему. Комплексы развития характеризуются типом и числом эмулируемых МП и числом одновременно работающих пользователей.

Особенностью комплексов развития является наличие внутреннего эмулятора, который моделирует поведение и электро-физические характеристики процессора или устройств, собирает информацию о поведении системы.

Внутренний эмулятор может прервать функционирование диагностируемой системы и появление заданного события, пускать выполнение с заданной команды, выполнять программы в пошаговом или автономном режимах, изменять состояние памяти и регистров МП, порядков ввода-вывода.

Внутренний эмулятор подключается к диагностируемой системе в том месте, где должен находиться МП, эмуляция устройств позволяет использовать ресурсы комплекса развития для поэтапной отладки. В частности сбора информации, внутрисхемный эмулятор обладает возможностями ЛА с синхронной записью данных. Позволяет собирать статические данные о времени выполнения тех или иных участков программы.

Комплексы развития делятся на однопроцессорные (рисунок 9.1) и многопроцессорные; многопроцессорные одномагистральные и многопроцессорные многомагистральные.



ЦП – центральный процессор; ПВВ – порты ввода-вывода; П – память

Рисунок 9.1 – Комплекс развития с одномагистральной однопроцессорной структурой

Многопроцессорные одномагистральные (рисунок 9.2) обычно используют один процессор для управления комплексом и один или несколько в составе внутренних эмуляторов. Все процессоры работают общей шиной и используют разделяемую память и устройства. Недостатком является то, что с шиной одновременно может работать только один процессор.

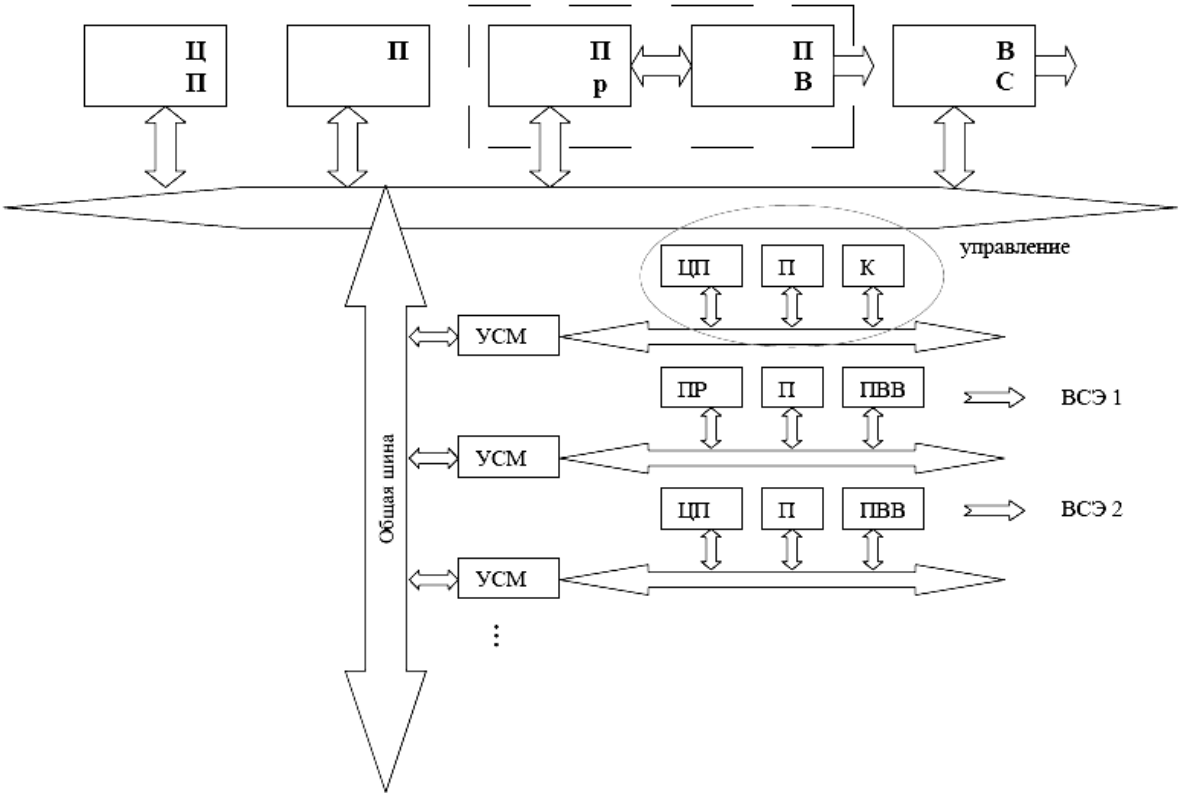


Рисунок 9.2 – Комплекс развития с одномагистральной многопроцессорной структурой

Этого недостатка лишены многомагистральные структуры, каждый внутрисхемный эмулятор имеет собственный процессор, собственную память и магистраль, связанную с общей магистралью, что позволяет вести независимую эмуляцию по всем канала.

Программное обеспечение комплекса развития обычно состоит из:

– операционной системы;

– редактора текста;

– кроссассемблера;

– кросскомпилятора.

Кроссассемблер и кросскомпилятор обеспечивают разработку программ для конкретного МП и драйверов внутрисхемных эмуляторов.

Применение логических моделей для построения алгоритмов диагностики.

Логические модели применяются на более высоком уровне представления системы, чем функциональные. Обычно логические модели строят из функциональных.

Особенности логических моделей следующие:

1) элементы системы теряют не только свое физическое представление, но и функцию, которую они выполняют;

2) любой элемент может находиться в одном из двух состояний: исправном и неисправном;

3) все сигналы могут быть одного из двух типов: допустимые и недопустимые;

4) исправный элемент при поступлении допустимого сигнала на вход вырабатывает допустимый выходной сигнал. Любой элемент, получающий недопустимый сигнал вырабатывает недопустимый выходной сигнал. Неисправный элемент вырабатывает недопустимый выходной сигнал независимо от входного;

5) все элементы работают в едином временном кванте.

Данные правила позволяют представить логическую модель в виде однонаправленного графа и применять для исследования моделей методы исследования графов (рисунок 9.3).

Обычно при исследовании логических моделей применяется таблица состояний.

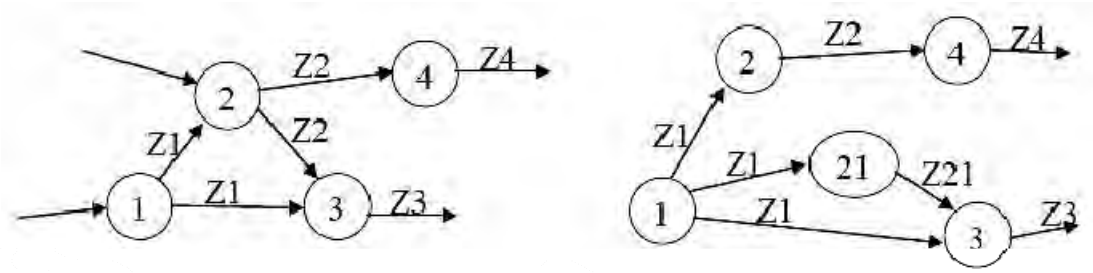


Рисунок 9.3 – Схемы состояний

Так как мы исследуем изолированную систему, мы предполагаем, что все входные сигналы, поступающие в систему, допустимы, поэтому мы можем этими сигналами пренебречь.

Для удобства допустимый сигнал и исправное состояние элемента обозначают 1, а недопустимый сигнал и неисправное состояние элемента обозначают 0.

Несмотря на то, что мы отказались от функциональности элементов, мы понимаем, что любой элемент все равно исполняет какую-либо функцию, поэтому принимается еще одно условие – каждый элемент вырабатывает только один выходной сигнал. Если от элемента требуется выдавать несколько сигналов, этот элемент заменяется несколькими элементами.

В таблицу заносятся состояния сигналов при различных состояниях системы (таблица 9.1).

Таблица 9.1 – Состояния сигналов при различных состояниях системы

Состояние системы	Состояние сигнала					
	*S*0	*S*1	*S*2	*S*21	*S*3	*S*3
*Z*1	1	0	1	1	1	1
*Z*2	1	0	0	1	1	1
*Z*21	1	0	1	0	1	1
*Z*3	1	0	1	0	0	1
*Z*4	1	0	0	1	1	0

Состояние *S*0 – все элементы исправны, *S*1 – неисправный первый элемент и т. д.

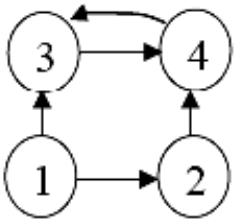
Таблица не может содержать столбца, состоящего из одних единиц, кроме первого.

Таблица не может содержать двух одинаковых строк.

Таблица не может содержать двух одинаковых столбцов.

Если эти правила не соблюдаются, то система становится не тестопригодной и должна быть преобразована в тестопригодную.

Преобразование заключается в разрыве циклических связей, либо в объединении нескольких элементов в один. В приведенной выше системе нужно объединить третий и четвертый элементы (рисунок 9.4).



	S0	S1	S2	S3	S4
Z1	1	0	1	1	1
Z2	1	0	0	1	1
Z3	1	0	1	0	0
Z4	1	0	1	0	0

Рисунок 9.4 – Циклические связи

Для исследования таблицы состояний применяются такие же методы, как и для исследования функциональных моделей.

Порядок выполнения работы

1 Построить функциональную схему устройства.

2 Определить возможные отказы функциональных блоков устройства.

3 Разработать алгоритм выявления отказавшего элемента.

4 Разработать упрощенный тест устройства, позволяющий оценить вероятность исправности устройства на основе построенной таблицы диагнозов определить неисправный элемент.

Контрольные вопросы

1 Какого рода неисправности возможны в логическом и вычислительном устройствах?

2 Различны ли виды неисправностей комбинационных и последовательностных элементов?

3 Почему обычно нельзя провести полный перебор возможных состояний устройства?

4 Если полный перебор возможных состояний устройства возможен, будет ли его достаточно для определения исправности устройства?

10 Контроль и тестирование оперативных запоминающих устройств

Цель работы: ознакомиться с принципами и приемами тестирования оперативных устройств. Освоить алгоритмы ускоренного тестирования.

Приборы и оборудование

Персональный компьютер, среда программирования C++, C#, Java.

Общая часть

Основным требованием, предъявляемым к неразрушающим тестам, является необходимость восстановления исходного состояния объекта тестирования после выполнения процедуры тестирования. Поэтому все тестовые воздействия, направленные на активизацию неисправностей и проявление их в виде ошибок на выходах схемы, должны носить обратимый характер. Простейшим способом реализации этого требования для случая неразрушающего тестирования памяти является сохранение всего ее содержимого в резервном буфере с последующим выполнением одного из классических разрушающих алгоритмов.

Существует несколько методов неразрушающего тестирования ОЗУ.

1 Тестирование по методу Николаидиса.

2 Симметричное тестирование памяти.

3 Локально-симметричные алгоритмы тестирования памяти.

Порядок выполнения работы

1 Проанализировать полученный от преподавателя набор методов тестирования.

2 Разработать алгоритмы реализации заданных методов тестирования.

3 Разработать программы тестирования оперативных запоминающих устройств.

4 Проверить работу программ.

5 Оценить полноту реализации методов и полноту тестирования заданными методами.

Контрольные вопросы

1 Почему при тестировании оперативных запоминающих устройств не применяют полный перебор возможных состояний устройства?

2 Какого рода отказы возможны в оперативных запоминающих устройствах?

3 Имеется ли какой-либо метод, проверяющий все возможные виды отказов?

4 Обеспечивает ли данный Вам набор методов полное тестирование всех видов отказов?

Список литературы

1 **Бочкарев, С. В.** Диагностика и надежность автоматизированных технологических систем : учебное пособие / С. В. Бочкарев, А. И. Цаплин, А. Г. Схиртладзе. – Старый Оскол : ТНТ, 2013. – 616 с.

2 **Бороденко, В. А.** Сборник задач по теории автоматического управления : учебно-методическое пособие / В. А. Бороденко. – Павлодар : Кереку, 2009. – 112 с.

3 **Ким, Д. П.** Сборник задач по теории автоматического управления. Линейные системы : учебное пособие / Д. П. Ким, Н. Д. Дмитриева. – Москва : Физматлит, 2008. – 328 с.

4 **Клавдиев, А. А.** Теория автоматического управления в примерах и задачах : учебное пособие в 2 ч. / А. А. Клавдиев. – Санкт-Петербург : СЗТУ, 2005. – Ч. 1. – 74 с.