[ВВЕДЕНИЕ 4](#_Toc357802398)

[1 АНАЛИЗ БОРТОВОГО ИНТЕГРИРОВАННОГО ВЫЧИСЛИТЕЛЬНОГО КОМПЛЕКСА(БИВК) 5](#_Toc357802399)

[1.1 СОСТАВ БИВК 6](#_Toc357802400)

[1.2 ОПИСАНИЕ И СИСТЕМА РЕКОНФИГУРАЦИИ БИВК 8](#_Toc357802401)

[1.3 СТРУКТУРА РЕЗЕРВИРОВАНИЯ БИВК 15](#_Toc357802402)

[1.3.1 ПЕРВЫЙ КОНТУР РЕЗЕРВИРОВАНИЯ 16](#_Toc357802403)

[1.3.2 ВТОРОЙ КОНТУР РЕЗЕРВИРОВАНИЯ 18](#_Toc357802404)

[1.3.3 ЛОКАЛЬНЫЙ КОНТУР РЕЗЕРВИРОВАНИЯ АСВТМИ 19](#_Toc357802405)

[1.3.4 ОБЩИЕ ПОЛОЖЕНИЯ РЕКОНФИГУРАЦИИ БИВК 20](#_Toc357802406)

[1.3.5 АВТОМАТИЧЕСКАЯ РЕКОНФИГУРАЦИЯ 21](#_Toc357802407)

[1.4 МЕТОДЫ ОЦЕНКИ ПОКАЗАТЕЛЕЙ БЕЗОТКАЗНОСТИ БИВК 24](#_Toc357802408)

[1.4.1 МАРКОВСКИЕ ПРОЦЕССЫ 26](#_Toc357802409)

[1.4.2 МЕТОД МИНИМАЛЬНЫХ ПУТЕЙ И СЕЧЕНИЙ 28](#_Toc357802410)

[1.4.3 ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ 30](#_Toc357802411)

[1.5 Обзор программного обеспечения имитационного моделирования 32](#_Toc357802412)

[1.5.1 ANYLOGIC 32](#_Toc357802413)

[1.5.2 GPSS 32](#_Toc357802414)

[1.6 ОПИСАНИЕ ПОДСИСТЕМЫ АСОНИКА-К-РЭС 33](#_Toc357802415)

[1.6.1 Выводы 35](#_Toc357802416)

[1.7 ОПИСАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ АСОНИКА-К-РЭС 35](#_Toc357802417)

[1.8 Создание модели БИВК при помощи АСОНИКА-К-РЭС 38](#_Toc357802418)

[1.9 ВЕРИФИКАЦИЯ МОДЕЛИ БИВК 42](#_Toc357802420)

[1.10 МОДЕЛИРОВАНИЕ МОДЕЛИ БИВК 44](#_Toc357802421)

[1.11 АНАЛИЗ РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ БИВК 46](#_Toc357802422)

[1.12 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПРИ РАБОТЕ С АСОНИКА-К-РЭС 46](#_Toc357802423)

[1.13 ПРЕДЛОЖЕНИЯ, РЕКОМЕНДАЦИИ ПО РАЗВИТИЮ ПОДСИСТЕМЫ АСОНИКА-К-РЭС 48](#_Toc357802424)

[1.14 РЕЗУЛЬТАТЫ РАСЧЕТА 49](#_Toc357802425)

[2 КОНСТРУКТИВНО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 51](#_Toc357802426)

[2.1 ТЕХНОЛОГИЯ СОЗДАНИЯ БЛОК-СХЕМЫ В СООТВЕТСТВИИ С ЕСПД 51](#_Toc357802427)

[2.2 СОЗДАНИЕ БЛОК-СХЕМ МЕТОДИК В СООТВЕТСТВИИ С ЕСПД 52](#_Toc357802428)

[3 ЗАКЛЮЧЕНИЕ 54](#_Toc357802430)

[4 ОХРАНА ТРУДА 55](#_Toc357802431)

[4.1 ЭЛЕКТРОБЕЗОПАСНОСТЬ 55](#_Toc357802432)

[4.2 ОБЕСПЕЧЕНИЕ ЭЛЕКТРОБЕЗОПАСНОСТИ ТЕХНИЧЕСКИМИСПОСОБАМИ И СРЕДСТВАМИ 57](#_Toc357802433)

[4.3 РАСЧЕТ ЗАЩИТНОГО ЗАНУЛЕНИЯ НА РАБОЧЕМ МЕСТЕ 60](#_Toc357802434)

[5 ЭКОЛОГИЧЕСКАЯ ЧАСТЬ ПРОЕКТА 62](#_Toc357802435)

[5.1 МИКРОКЛИМАТ В РАБОЧЕЙ ЗОНЕ 62](#_Toc357802436)

[5.2 ЗАЩИТА ОТ ШУМА 64](#_Toc357802437)

[6 ПРИЛОЖЕНИЕ 68](#_Toc357802439)

# ВВЕДЕНИЕ

Одним из приоритетных направлений развития отечественной промышленности является космическая отрасль. Для нее создается крайне разнообразное оборудование и системы, включающие в себя и электронные средства (ЭС) различной сложности.

ЭС являются основной функциональной составляющей космических аппаратов, с их помощью производится управление двигателями, контроль орбиты, связь с землей и центром управления. ЭС производят сбор, обработку и передачу телеметрии, просто выполняют функцию космического аппарата. Так же работа ЭС в условиях нахождения в космическом пространстве подвержена ряду факторов негативно сказывающихся на работе ЭС, такие как вакуум, жесткое солнечное излучение, большие перепады температур, высокая вероятность электростатических разрядов по площади поверхности космических аппаратов. Вследствие этого, при проектировании бортовых электронных средств для космических аппаратов в техническом задании всегда указываются требования по надежности, и, при проектировании отдельное внимание уделяется обеспечению требований надежности.

Задачей работы является применение метода исследований надежности реконфигурируемых электронных средств на основе имитационного моделирования.

# АНАЛИЗ БОРТОВОГО ИНТЕГРИРОВАННОГО ВЫЧИСЛИТЕЛЬНОГО КОМПЛЕКСА(БИВК)

Бортовой интегрированный вычислительный комплекс (БИВК), это современное высокотехнологичное ЭС, разработанное ведущими предприятиями НИИ «Аргон» и НТЦ «Модуль». БИВК (рис.2.1) предназначен для использования в околоземном космическом пространстве в составе бортовых вычислительных сетей КА.



Рис. 2.1 – БИВК

Вследствие этого в ТЗ на разработку жестко устанавливаются необходимые показатели надежности. Для обеспечения заданного уровня надежности и в силу структурных особенностей БИВК разработчиками была применена система резервирования и реконфигурации. Общая структура БИВК представлена на рис. 1.1.

БИВК состоит из 2-х комплектов модулей. Физически однотипные модули расположены на одной ячейке и запитываются от различных источников питания.

## СОСТАВ БИВК

Группы модулей объединяются в секции в соответствии с реализуемыми ими функциями и конструктивным исполнением.

Секция 1 - секция отключаемых источников питания ВИП1.

Секция 2 – секция вычислителя в составе следующих модулей: ВМ, ИМ, МПР.

Секция 3 – секция выдачи управляющих воздействий в составе следующих модулей: МНК, МЦАП.

Секция 4 – секция приема/выдачи команд управления и реконфигурации в составе следующих модулей: МРК, МАКБ МШВ.

Секция 5 – секция не отключаемых источников питания ВИП2.

Секция 6 - секция мультиплексора аппаратуры сбора телеметрической информации

Секция 7 – измерительная секция 0,

Секция 8 – измерительная секция 1,

Секция 9 - секция вторичных источников питания.

Назначение модулей секции отключаемых источников питания

В составе секции имеется только один модуль вторичных источников питания номиналами +3.3, 5.0В, +12.0В, -12.0В предназначенный для подачи питания на секцию вычислителя. В состав блока входят поляризованные реле (дистанционные переключатели) служащие для отключения/включения ВИП по внешним командам.

Назначение модулей секции вычислителя (СВ).

Вычислительный модуль предназначен для программной реализации алгоритмов управления и контроля бортовых систем КА и состоит из следующих составных частей:

* микропроцессор,
* оперативное запоминающее устройство,
* электрически перепрограммируемое запоминающее устройство,
* контроллер внутренней шины БИВК,
* контроллер прерываний,
* сторожевой и универсальные таймера.

Интерфейсный модуль предназначен для реализации требований по мультиплексному каналу обмена в соответствии со стандартом MIL-STD-1553В. В состав модуля входят два независимых канала МКО, причем каждый из каналов может использоваться в режиме контроллера или в режиме оконечного устройства, задаваемым бортовым программным обеспечением.

Модуль приема внешних прерываний осуществляет прием сигналов на 16 входов, в том числе, 10 сигналов прерывания типа СК (не запитанный контактный датчик типа контакт реле) или НК (бесконтактный датчик – не запитанный электронный ключ) и 6 импульсных сигналов прерывания БИВК.

## ОПИСАНИЕ И СИСТЕМА РЕКОНФИГУРАЦИИ БИВК

Описание функционирования БИВК, применимое к расчету показателей надежности.

Модули телеметрии полукомплектов А и Б санкционируют следующим образом.

Модуль МПС(А) обеспечивает питание следующих групп модулей:

* модуль ММ(А) + модуль ММХ8(А);
* 2 группы модулей МКПС(А);
* 11 групп модулей МУП2(А). причем каждый модуль МУП2(А) связан по цепям питания с одним из измерительных модулей МПТ32(А), МПА64(А), МПЦ80(А), МПН32(А).

Каждый из двух модулей МКПС(А) связан функционально с группой измерительных модулей:

* модуль МКПС1(А) связан с тремя модулями МПТ32(А) (группа 1 измерительных модулей), двумя модулями МПЦ80(А) (группа 3 измерительных модулей) и одним модулем МПН32(А) (группа 4 измерительных модулей);
* модуль МКПС2(А) связан с двумя модулями МПТ32(А) (группа 1 измерительных модулей), одним модулем МПА64(А) (группа 2 измерительных модулей) и двумя модулями МПЦ80(А) (группа 3 измерительных модулей).

Аналогично модуль МПС(Б) обеспечивает питанием такие же модули полукомплекта Б.

При отказе модуля МПС(А) отключаются все модули полукомплекта А, перечисленные выше, и в работу включается модуль МПС(Б) и все модули полукомплекта Б.

При отказе модулей ММ(А) или ММХ8(А) в работу включаются модуль МПС(Б) и все модули полукомплекта Б, модули ММ(Б) и ММХ8(Б) начинают функционировать вместо отключенных модулей ММ(А) и ММХ8(А).

При отказе одного из модулей МКПС(А) в работу включаются модуль МКПС(Б) и все модули полукомплекта Б кроме измерительных модулей и модулей МУП2, связанных с модулем МКПС(Б), соответствующем исправному модулю МКПС(А). Вместо отказавшего модуля МКПС(А) и связанной с ним цепочки измерительных модулей с соответствующими им модулями МУП2(А) начинает функционировать модуль МКПС(Б) и связанная с ним цепочка измерительных модулей и модулей МУП2(Б).

При отказе одного из модулей МУП2(А) или одного из измерительных модулей полукомплекта А в работу включаются следующие модули:

* МПС(Б);
* ММ(Б);
* ММХ8(Б);
* МКПС1(Б);
* МКПС2(Б);
* измерительный модуль полукомплекта Б вместе со своим модулем МУП2, соответствующие отказавшему модулю полукомплекта А.

Все остальные измерительные модули и соответствующие им модули МУП2 полукомплекта Б остаются в выключенном состоянии.

Вместо отказавшего модуля МУП2(А) или измерительного модуля полукомплекта А начинают функционировать соответствующий измерительный модуль полу комплекта Б и связанный с ним по питанию модуль МУП2(Б).

Комплект А

Команды на вкл/откл ВИП

Команды на вкл/откл ВИП

Комплект Б

ВИП1

ВИП1

27ВА

27ВБ

ИТХБ

ИТХА

ККСБ

ККСА

ВМ

КПУА

ВМ

КПУБ

ТКБ

ТКА2

ИМКО

МКО1

МКО1

ИМ

ИМ

МКО2

ИМКО

МКО2

ИППР

**…**

ПРЕР1

ИППР

МПР

МПР

ПРЕР1

**…**

ПРЕР16

ПРЕР16

ИМСБ

ИМСА

МНК

**…**

НК1А

МНК

НК1Б

**…**

НК32А

НК32Б

МЦАП

МЦАП

ЦАП1А

**…**

**…**

ЦАП1Б

ИИКУ

ИИКУ

ЦАП4А

ЦАП4Б

**…**

КХ1А

**…**

КХ1Б

МРК

КХ16А

КХ16Б

МРК

**…**

КY1А

КY1Б

**…**

КY16А

КY16Б

От/в

БА КИС

От/в

БА КИС

МАК

МАК

КИ\_КИС

КИ\_КИС

МШВ

1 кГц

МШВ

1 кГц

ВИП2

ВИП2

27ВС

27ВД

Комплекты А, Б модулей сбора, обработки и выдачи телеметрической информации

Рис. 2.2 Структурная схема БИВК( часть 1)

**На рис 2.2( часть 1) приняты следующие сокращения названия модулей:**

ВМ – вычислительный модуль,

ИМ – интерфейсный модуль,

МНК – модуль выдачи независимых команд,

МЦАП – модуль выдачи аналоговых команд управления,

МПР – модуль приема прерываний,

ВИП1 – отключаемый вторичный источник питания

МАК – модуль автоконфигурации и команд,

МШВ - модуль шкалы времени,

МРК – модуль разовых команд,

ВИП2 – не отключаемый вторичный источник питания

На структурной схеме( часть 1) приняты следующие сокращения внешних и внутренних связей БИВК:

ИМСА(Б) – шины интерфейса межмодульных связей

МКО1,2 – мультиплексные каналы обмена

ИМКО – интерфейс мультиплексных каналов обмена

ККСА(Б) – каналы контроля состояния

КПУА(Б)- каналы прямого управления

ТКА(Б) – технологические каналы RS-232

ИТХА(Б) – технологические интерфейсы

ПРЕР1-16 –сигналы внешних прерываний

ИППР – интерфейс приема прерываний

ЦАП1-4 – каналы выдачи аналоговых сигналов вольтового диапазона

НК1-32 – каналы выдачи независимых команд

КХ1-КХ16, КУ1-КУ16 – сигналы выдачи разовых команд через матрицу

ИИКУ – исполнительный интерфейс команд управления

27В – первичное питание БИВК

1кГц – внешняя частота для шкалы времени

КИ\_КИС – командный интерфейс с бортовой аппаратурой командно-измерительной системы (БА\_КИС).Рис. 2.3 Структурная схема БИВК (часть 2)



**На рис 2.3( часть 2) приняты следующие сокращения названия модулей:**

ММХ8 – модуль мультиплексора,

МПАД32 - модуль измерения 32 сигналов АД6,

МПТ44 - модуль измерения 44 сигналов ТД

МПА44 - модуль измерения 44 сигналов АД

МПР80 - модуль измерения 80 сигналов ЦД

МКС - модуль контроллера секции

ВИП3 - вторичный отключаемый источник питания

Где АД6 – сигналы аналоговых датчиков с диапазоном изменения от 0 В до 6 В,

ТД - сигналы от температурных и генераторных датчиков,

АД - сигналы от аналоговых датчиков с диапазоном от минус 6 В до + 6 В

ЦД - сигналы от цифровых датчиков типа «запитанный электронный ключ», датчиков типа «незапитанный электронный ключ», цифровых датчиков типа «сухой контакт»

КТМ КИС - канал передачи телеметрической информации в КИС,

ТКТМ – технологический канал телеметрии,

ОЗУТМА - оперативное запоминающее устройство для хранения и формирования телеметрических сообщений

## СТРУКТУРА РЕЗЕРВИРОВАНИЯ БИВК

Структура резервирования БИВК с аппаратно-программной точки зрения имеет 2 контура, построенных по иерархическому принципу, т.е. когда подчиненный контур обеспечивает дополнительные возможности резервирования для вышестоящего при условии, что базовые элементы в каждой ступени резервирования вышестоящего контура исправны.

Оба этих контура являются глобальными, т.е. обеспечивающими полноценное функционирование БИВК по целевому назначению.

Так же в БИВК имеется локальный контур резервирования, реализующий требования ТЗ об автономном функционировании аппаратуры сбора, обработки и выдачи телеметрической информации (АСВТМИ) и принципа – «телеметрия должна умирать последней».

### ПЕРВЫЙ КОНТУР РЕЗЕРВИРОВАНИЯ

Первый контур обеспечивает резервирование БИВК на уровне функциональных секций аппаратуры, питаемых от одного и того же источника питания, т.е. базовым элементом этого контура по отношению к контуру 2 являются вторичные источники питания.

Этот контур имеет 3 ступени резервирования и представлен на рис.2,4.



Рис.2.4 Первый контур системы резервирования.

Блок-схема управления электропитанием БИВК представлена на рис.2.5. Источником команд управления питанием могут являться Земля, вычислительный модуль и модуль реконфигурации.



Рис.2.5 Блок-схема управления электропитанием БИВК

### ВТОРОЙ КОНТУР РЕЗЕРВИРОВАНИЯ

Второй контур обеспечивает резервирование БИВК на уровне доступности аппаратных ресурсов БИВК от секции вычислителя полукомплекта А или Б (в составе модулей ВМ, ИМ, МПР) посредством резервированного интерфейса межмодульных связей.

Этот контур имеет 5 ступеней резервирования и представлен на рис.1.6.



Рис.2.6 Второй контур системы резервирования.

Следует отметить, что для цепочек модулей внутри прямоугольников существует такая группа отказов в каждом из них, при возникновении которых другие модули остаются работоспособными и, таким образом, полноценно функционирующая секция может быть построена из «кусочков» разных полукомплектов, поскольку полукомплекты модулей каждой секции имеют разные адреса в адресном пространстве ИМС.

### ЛОКАЛЬНЫЙ КОНТУР РЕЗЕРВИРОВАНИЯ АСВТМИ

Локальный контур резервирования предназначен для поддержания автономного функционирования АСВТМИ независимо от работоспособности остальной части БИВК.

Этот контур имеет 4 ступени резервирования и представлен на рис.2.7.



Рис.2.7 Локальный контур резервирования АСВТМИ.

Следует отметить, что для цепочек модулей внутри прямоугольников отказ любого модуля в каждом из них (кроме модуля МКС), или даже нескольких модулей, не приводит к полному отказу всей цепочки, поскольку, если такие же модули остаются работоспособными в аналогичной цепочке, то эти отказы могут парироваться аппаратно-программными средствами самой АСВТМИ - без участия МАК, МРК, головного ВМ (при условии, что запитаны обе цепочки) и, таким образом, АСВТМИ остается полностью функциональной.

### ОБЩИЕ ПОЛОЖЕНИЯ РЕКОНФИГУРАЦИИ БИВК

Процедура реконфигурации БИВК может инициироваться тремя путями: внешняя (принудительная), по командам с Земли, автоматическая, инициируемая исключительно аппаратными средствами БИВК и реконфигурация инициируемая и проводимая под управлением БПО.

Аппаратура, непосредственно используемая при проведении реконфигурации – модули МАК и МРК. Общий порядок прохождения реконфигурации следующий:

* возникновение запроса на исполнение разовой (одиночной) команды конфигурации или последовательности команд;
* арбитраж запроса, декодирование команды или последовательности команд (МАК);
* исполнение команды или последовательности команд (МРК).

Запросы могут возникать от 3-х источников: от БА КИС (Земля, только разовые команды), от ВМ (БПО, разовые команды и последовательности команд), от логики аппаратной конфигурации/реконфигурации и восстановления (узел МАК, разовые команды и последовательности команд)

Ниже речь пойдет только об автоматической реконфигурации и реконфигурации под управлением БПО.

### АВТОМАТИЧЕСКАЯ РЕКОНФИГУРАЦИЯ

Логика аппаратной конфигурации/реконфигурации и восстановления обеспечивает непрерывный мониторинг ограниченного количества сигналов от критически важных устройств БИВК.

К таким устройствам относятся модули ВМ и модули ММХ полукомплектов А и Б соответственно, т. е. модули в которых расположен «интеллект», обеспечивающий функционирование БИВК по целевому назначению.

Эти модули поставляют в оба МАК апериодические сигналы «ВМА\_ЖИВ», «ВМБ\_ЖИВ», «ММХА\_ЖИВ», «ММХБ\_ЖИВ» генерируемые прикладным программным обеспечением.

Данные сигналы сбрасывают в МАК индивидуальные сторожевые таймеры, которые, в случае превышения некоторого временного интервала, будут генерировать запрос на реконфигурацию, которая может иметь от одной до 4-х стадий, в упрощенном виде представленных для ВМА(Б) на рис.2.8.



Рис.2.8 Процедура автоматической реконфигурации.

Следует отметить, что аналогичная процедура для модулей ММХ дополняется на каждой стадии прерыванием на активный, в текущий момент, модуль ВМ.

Таким образом, автоматическая реконфигурация обеспечивает поддержание работоспособности БИВК на уровне первого контура резервирования.

## МЕТОДЫ ОЦЕНКИ ПОКАЗАТЕЛЕЙ БЕЗОТКАЗНОСТИ БИВК

Согласно ГОСТ 51901.5-2005выбор метода расчета надежности зависит от следующих критериев:

* Сложность системы. Сложные системы, включающие резервирование.
* Новизна системы. Новая системы требует более глубокого анализа, чем разработанная ранее.
* Качественный или количественный анализ. Действительно ли количественный анализ необходим?
* Единичные или многократные неисправности. Существенно ли влияние комбинации неисправностей или можно ими пренебречь?
* Поведение системы зависит от времени или последовательности событий. Имеет ли значение для анализа последовательность событий (например, система отказывает только в случае, если событию А предшествует событие В, но не наоборот) или поведение системы зависит от времени (например, ухудшение режимов работы после отказа или выполнения функции)?
* Возможность использования метода для зависимых событий. Зависят ли характеристики отказа или восстановления отдельного элемента от состояния системы в целом?
* восходящий или нисходящий анализ. Обычно применение восходящих методов является более простым. Применение нисходящих методов требует осмысления и творческого подхода и имеет больше возможностей для ошибок;
* распределение требований надежности. Может ли метод быть приспособлен к количественному распределению требований надежности.
* квалификация исполнителя. Какой требуется уровень образования или опыта для правильного применения метода.
* применимость. Например, регулирующая сторона или заказчик обычно применяет метод
* необходимость инструментальной поддержки. Нуждается ли метод в компьютерной поддержке или он может быть выполнен вручную?
* проверки правдоподобия. Можно ли проверить правдоподобие результатов вручную? Если нет, являются ли инструментальные средства доступными?
* работоспособность инструментальных средств. Действительно ли инструментальные средства доступны. Имеют ли эти инструментальные средства общий интерфейс с другими инструментальными средствами анализа, чтобы результаты могли многократно использоваться или передаваться.
* стандартизация. Существует ли стандарт, устанавливающий требования к представлению его результатов

### МАРКОВСКИЕ ПРОЦЕССЫ

Расчет надежности восстанавливаемых систем проводят при помощи метода расчета с использованием аппарата теории Марковских процессов.  
 Марковский процесс является марковской цепью с различными состояниями и может быть представлен матрицей значений переходных вероятностей.



Элемент матрицы переходов рij соответствует вероятности того, что марковская цепь, находящаяся в состоянии Si, на очередном шаге перейдет из состояния Si в состояние Sj  
Из самого опреде­ления марковских цепей выходит, что переходные вероятности не зависят от номера шага, т. е. процесс переходов одинаковый во времени. Исходя из этого, Марковская цепь — дискретный случайный процесс с дискретным временем, переходы которого осу­ществляются через некоторый интервал времени Δ*t* из од­ного состояния в другое в счетном множестве состояний.  
 В марковском процессе длительность пребывания в каж­дом из состояний Si,- величина случайная, имеющая свое рас­пределение *Fk (t)* для каждого *k-гo* состояния. Причем все распределения *Fk (t)* подчиняются экспоненциальному закону.  
 Марковский процесс обладает характерными свойствами, определяемыми в первую очередь, экспоненциальностью рас­пределения времени пребывания процесса в каждом из сос­тояний: стационарностью, когда переходные вероятности и длитель­ности пребывания в том или ином состоянии не зависят от то­го, в какой момент времени рассматривается этот процесс; ординарностью, когда за бесконечно малый интервал време­ни не может произойти более одного перехода из одного со­стояния в другое состояние; отсутствием последействия, когда вся последующая траек­тория перехода процесса из состояния в состояние зависит толь­ко от текущего состояния и не зависит от всей прошлой истории развитии этого процесса или от любой его части.[1]

Использование Марковских процессов не подходят для расчета данной реконфигурируемой ЭС, потому что в данной системе большое количество состояний.

Для примера рассчитаем количество состояний при двух отказавших блоках, по формуле -.

.

Получаем 435 сочетаний при двух отказавших блоках. С количеством отказавших блоков будет расти количество сочетаний, что делает применение метода неоправданным.

### МЕТОД МИНИМАЛЬНЫХ ПУТЕЙ И СЕЧЕНИЙ

Минимальным путем называется такой j-й минимальный путь, который состоит из минимальной совокупности Мi под­систем, необходимой для обеспечения безотказной работы системы независимо от состояния остальных подсистем. В структуре системы имеется, как правило, несколько мини­мальных путей. Характерным признаком минимального пути считается то, что отказ хотя бы одной подсистемы пути (если работоспособны только подсистемы пути) влечет за собой отказ системы.  
  
 Минимальное сечение — это такое *k-e* минимальное сече­ние, состоящее из минимальной совокупности подсистем Nk, чей одновременный отказ влечет за собой отказ системы не­зависимо от состояния остальных подсистем. Характерная особенность минимального сечения — восстановление хотя бы одной подсистемы в минимальном сечении (если остальные подсистемы работоспособны) влечет за собой восстановление системы.

По методу минимальных путей и сечений возможно полу­чить только оценки PН(t) и РВ(t)- вероятности безотказной работы системы снизу и сверху соответственно. Вероятность безотказной работы системы оценивается в пределах

PН(t)≤РC(t)≤РВ(t).

Вероятность PН(t) выражается как вероятность безотказ­ной работы системы вспомогательной, составленной из групп после­довательно включенных подсистем, соответствующих всем минимальным сечениям всей системы. Каждая группа сос­тоит из параллельно включенных подсистем соответствующе­го минимального сечения.  
Вероятность РВ(t) выражается как вероятность безотказ­ной работы вспомогательной системы, составленной из па­раллельно включенных групп подсистем, соответствующих всем минимальным путям системы.  
Каждая группа состоит из последовательно включенных подсистем соответствующего минимального пути.[2]

Метод минимальных путей и сечений не дает точных оценок, дает только интервал. Так же метод не учитывает изменение (невозможно учесть изменение) интенсивности отказов в процессе работы, что дает вероятность дополнительной ошибки, сложно учитывать реконфигурируемые системы, так как схема расчета надежности изменяется всякий раз с отключением элементов.

### ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ

Имитационное моделирование — это метод исследования, при котором система заменяется моделью, с достаточной точностью описывающей реальную систему и ее развитие во времени. Цель имитационного моделирования состоит в воспроизведении поведения исследуемой системы на основе результатов анализа наиболее существенных взаимосвязей между ее элементами, или, другими словами, — разработке симулятора исследуемого объекта для проведения различных экспериментов.

Над моделью проводятся эксперименты, в ходе которых определяется состояние моделируемой системы в каждый момент времени. По их итогам можно получить данные о исследуемом процессе, аналогичные тем, которые могли бы быть получены в ходе натурного эксперимента.

При построении модели любого процесса или объекта методом имитационного моделирования можно выделить следующие общие шаги:

* Формулировка задачи;
* Сбор данных и создание концептуальной модели;
* Определение валидности концептуальной модели;
* Программирование модели;
* Определение валидности запрограммированной модели;
* Проектирование, управление и анализ экспериментов;
* Документирование и представление результатов моделирования.

Этот алгоритм, несмотря на крайне общую формулировку, позволяет построить модель практически любого объекта, в том числе и структурно сложной радиоэлектронной аппаратуры, и провести исследование процессов, протекающие в нем. При задаче оценки надежности РЭА необходимо построить модель, отражающую поток отказов компонентов, и на основе критериев отказов определяет состояние РЭА. Такая модель многократно прогоняется от начального состояния до состояния отказа с фиксацией прошедшего модельного времени.

Результатом является множество реализаций наработки на отказ модели, на основе которого можно сделать оценку средней наработки на отказ и/или вероятности безотказной работы. Также можно оценить и другие показатели надежности, фиксируя параметры работы модели.

Поток отказов моделируется при помощи источника псевдослучайных чисел путем розыгрыша времени до отказа компонента с помощью следующей формулы:

 (4.1)

– функция, обратная закону распределения вероятности безотказной работы,

P – разыгранное случайное число в диапазоне [0,1).

Для случая экспоненциального распределения отказов это выражение принимает вид:

 (4.2)

Способы задания алгоритмов реконфигурации, эффекта их действия и критериев отказов не могут быть описаны в общем виде, так как сильно зависят от программных средств, которые используются для создания модели. Это могут быть как классические языки программирования, так и специализированные языки или средства имитационного моделирования. При этом они не содержат специализированных моделей для описания реконфигурируемой РЭА, поэтому для проведения расчета необходимо эти модели реализовать и верифицировать, что само по себе является сложной задачей. Значительно упростить задачу можно, используя программное средство АСОНИКА-К-РЭС, которое содержит типовые заготовки для моделей реконфигурируемых РЭА.

## Обзор программного обеспечения имитационного моделирования

### ANYLOGIC

AnyLogic - программное обеспечение для имитационного моделирования, разработанное российской компанией «Экс Джей Текнолоджис». Продукт получил название AnyLogic, потому что он поддерживает все три известных метода моделирования:

* системная динамика;
* дискретно-событийное (процессное) моделирование;
* агентное моделирование.

Минусом AnyLogic при попытке расчета надежности РЭА является сложность изучения, необходимость специализированной подготовки, отсутствие готовых моделей компонентов для расчета надежности РЭА. Однако их реализация и набор готовых шаблонов есть в системе АСОНИКА-К-РЭС

### GPSS

Gpss – разрабатывался как язык для моделирования систем массового облуживания, исходя из этого стандартные объекты моделирования имеют узкую направленность, чем затрудняют моделирование электронных средств. GPSS имеет синтаксис, заметно отличающийся от классических языков программирования, и слабые алгоритмические возможности.

GPSS является способом алгоритмизации дискретных динамических систем. Ориентирована на описание параллельных процессов в динамической системе. Примеры моделируемых объектов: транспортные объекты, склады, производственные системы, магазины, торговые объекты, сети ЭВМ, системы передачи сообщений. Алгоритмическая схема может быть использована для оформления сложных формальных схем. Формальные модели таких объектов: СМО и стохастические сети, автоматы, сети Петри, агрегаты и др.

На основе опыта моделирования, можно сделать вывод о сложности применения языка GPSS для моделирования электронных средств с реконфигурируемой структурой. Моделирование электронных средств, состоящих более чем 4-5 компонентов и использующих схему резервирования сложнее однократного нагруженного резерва, приводит к сильному усложнению программы и практически неизбежно сложно устранимым ошибкам.

## ОПИСАНИЕ ПОДСИСТЕМЫ АСОНИКА-К-РЭС

Целью работы является получение общих представлений о порядке выполнения расчетов надежности в «Системе расчета надежности реконфигурируемых электронных средств» АСОНИКА-К-РЭС, а также приобретение навыков самостоятельного расчета надежности реконфигурируемых РЭА.

Кроме изменения элементной базы и внесения резервирования для повышения надежности в современной радиоэлектронной аппаратуре ответственного применения зачастую используется принцип реконфигурации составных частей. Основная идея этого метода состоит в перераспределении функций отказавших компонентов между исправными элементами РЭА. Под реконфигурируемыми системами понимаются такие системы, в процессе функционирования которых происходят перераспределения нагрузок, подключение и отключение дополнительных модулей в зависимости от решаемых задач или в случае отказов. Использование таких систем технически выгодно, так как позволяет эффективно расходовать энергию и ресурсы компонентов. Возможность реконфигурации структуры закладывается на этапе проектирования РЭА и тесно переплетена с особенностями исполняемых задач, общей структурой РЭА и многими другими факторами.

В силу этого алгоритмы реконфигурации строго индивидуальны для различных РЭА, и среди них невозможно выделить типовые группы. Для реконфигурируемых РЭА характерны следующие особенности при расчете показателей надежности:

− Изменение интенсивности отказов одних элементов при отказе других

(взаимосвязи между отказами компонентов);

− Уникальные алгоритмы реконфигураций;

− Сложные критерии отказов.

Эти особенности делают неприменимыми аналитические методы исследования надежности. Для таких расчетов целесообразно использовать имитационное моделирование, как более гибкий метод.

Подсистема АСОНИКА-К-РЭС предназначена для проведения расчетов показателей надежности радиоэлектронной аппаратуры.

АСОНИКА-К-РЭС представляет собой компилятор для построения программной модели РЭА на основе формальной модели в формате специализированного языка. Далее над построенной моделью, возможно провести имитационные эксперименты и определить показатели надежности.

АСОНИКА-К-РЭС позволяет создавать формальные модели радиоэлектронной аппаратуры со сложной структурой, верифицировать модель, проводить имитационное моделирование отказов, собирать и анализировать статистику в процессе моделирования и на ее основе рассчитывать показатели надежности.

Программа может применяться для расчета радиоэлектронной аппаратуры, функционирование которой возможно описать в рамках синтаксиса языка ИМОСЭТ.

Исходными данными для проведения исследований надежности с применением программы АСОНИКА-К-РЭС являются:

1. Интенсивности отказов компонентов во всех режимах работы

2. Описание структуры резервирования и реконфигурации

3. Критерии отказа

4. Описание функционирования

По этим данным необходимо составить текстовую модель ЭС, ее верифицировать и провести серию экспериментов.

В общем виде входные данные представляют собой описание модели в текстовом виде – формальная модель.

Выходными данными являются файлы с графиками, результатами обработки модели сообщения системы. Для главного компонента модели: график интенсивности отказов, средняя наработка на отказ, вероятность безотказной работы, коэффициент готовности; для остальных компонентов: статистика состояния на конец эксперимента, среднее время нахождения в каждом состоянии.

### Выводы

Использование других систем имитационного моделирования для расчета надежности электронных средств обладает те ми же недостатками, то есть отсутствие специализированных моделей для компонентов электронных средств, сложность задания алгоритмов резервирования, необходимость специальной подготовки для использования этих программ.

## ОПИСАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ АСОНИКА-К-РЭС

Для описания компонентов моделируемой РЭА используется встроенный язык программирования, при помощи, которого становится возможным описать компоненты по необходимым параметрам, для расчета надежности. Для построения формальной модели используется специализированный язык описания отказов РЭА. В язык встроены модели законов распределения, модели компонентов РЭА, структурные элементы, моделирующие события и специализированные алгоритмы исполнения структурных элементов языка. При помощи объектов языка моделируются элементы РЭА, как нижнего уровня, так и составных, моделируются действия алгоритмов реконфигурации.

Основным элементом формальной модели является компонент РЭА который задается при помощи оператора knot. Компоненты делятся на два типа: элементарный и составной. Элементарный компонент -это модель нижнего уровня разукрупнения структуры РЭА, в нем нет составных компонентов, и исходными данными для него являются законы распределения (в частном случае - закон распределения отказов). Составной компонент с точки зрения структуры РЭА - это компонент более высокого уровня, у него есть дочерние элементы, и законы распределения в явном виде для него неизвестны. Внешне для модели любой компонент описывается состоянием и режимом. Состояние определяется внутренними процессами, протекающими в компоненте (в конечном счете, они моделируются законами распределения). Режимы предназначены для моделирования влияния других компонентов и изменяются вследствие действия алгоритмов резервирования или реконфигурации.

Перед описанием компонента необходимо объявить закон распределения отказов. В общем виде описание одного компонента будет выглядеть следующим образом.

distribution Dis\_K\_example (1e-5);

Этим оператором объявляется экспоненциальный закон распределения с параметром λ (интенсивность отказов) = 1\*10-5 и именем Dis\_K\_example. Далее это имя можно использовать при описании компонента:

Knot komponent

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

| Normal |

Work | Dis\_K\_example;

tableStateChange:

|Normal |

Work |Fail;

};

knot– ключевое слово, означающее объявления компонента,

component – имя компонента

state: Fail, Work; – перечисление состояний компонента, состояние

указанное первым считается необратимым отказом, если оно присвоено

компоненту, то изменить его уже нельзя.

mode: Normal; – перечисление режимов работы компонента

startState: Work; – состояние при начале эксперимента

startMode: Normal; – режим при начале эксперимента

cntrlMode: unDistribution; – тип контроля, значение unDistribution

означает, что данный компонент изменит состояние по прошествии времени, разыгранного генератором случайных чисел.

tableDistribution:

| Normal|

Work| Dis\_K\_example; – таблица, определяющая по какому

распределению разыгрывается время, которое компонент будет находится в данном состоянии.

tableStateChange:

|Normal |

Work |Fail;– таблица, указывающая, в какое состояние перейдет

данный компонент по истечении разыгранного времени. РЭА выбранный в качестве примера отказывает, следовательно, указано состояние Fail.

Для построения модели РЭА необходимо описать каждый компонент отдельно. Текст модели БИВК представлен в приложении. Данный пример является ознакомительным, для дачи представления о моделировании в системе АСОНИКА-К-РЭС.

## Создание модели БИВК при помощи АСОНИКА-К-РЭС

## Описание: БИВК

Рис 2.9. Схема модуля телеметрии

Описание функционирования БИВК, применимое к расчету показателей надежности.

Модули телеметрии полукомплектов А и Б санкционируют следующим образом ( рис 2.9).

Модуль МПС(А) обеспечивает питание следующих групп модулей:

* модуль ММ(А) + модуль ММХ8(А);
* 2 группы модулей МКПС(А);
* 11 групп модулей МУП2(А). причем каждый модуль МУП2(А) связан по цепям питания с одним из измерительных модулей МПТ32(А), МПА64(А), МПЦ80(А), МПН32(А).

Каждый из двух модулей МКПС(А) связан функционально с группой измерительных модулей:

* модуль МКПС1(А) связан с тремя модулями МПТ32(А) (группа 1 измерительных модулей), двумя модулями МПЦ80(А) (группа 3 измерительных модулей) и одним модулем МПН32(А) (группа 4 измерительных модулей);
* модуль МКПС2(А) связан с двумя модулями МПТ32(А) (группа 1 измерительных модулей), одним модулем МПА64(А) (группа 2 измерительных модулей) и двумя модулями МПЦ80(А) (группа 3 измерительных модулей).

Аналогично модуль МПС(Б) обеспечивает питанием такие же модули полукомплекта Б.

При отказе модуля МПС(А) отключаются все модули полукомплекта А, перечисленные выше, и в работу включается модуль МПС(Б) и все модули полукомплекта Б.

При отказе модулей ММ(А) или ММХ8(А) в работу включаются модуль МПС(Б) и все модули полукомплекта Б, модули ММ(Б) и ММХ8(Б) начинают функционировать вместо отключенных модулей ММ(А) и ММХ8(А).

При отказе одного из модулей МКПС(А) в работу включаются модуль МКПС(Б) и все модули полукомплекта Б кроме измерительных модулей и модулей МУП2, связанных с модулем МКПС(Б), соответствующем исправному модулю МКПС(А). Вместо отказавшего модуля МКПС(А) и связанной с ним цепочки измерительных модулей с соответствующими им модулями МУП2(А) начинает функционировать модуль МКПС(Б) и связанная с ним цепочка измерительных модулей и модулей МУП2(Б).

При отказе одного из модулей МУП2(А) или одного из измерительных модулей полукомплекта А в работу включаются следующие модули:

* МПС(Б);
* ММ(Б);
* ММХ8(Б);
* МКПС1(Б);
* МКПС2(Б);
* измерительный модуль полукомплекта Б вместе со своим модулем МУП2, соответствующие отказавшему модулю полукомплекта А.

Все остальные измерительные модули и соответствующие им модули МУП2 полукомплекта Б остаются в выключенном состоянии.

Вместо отказавшего модуля МУП2(А) или измерительного модуля полукомплекта А начинают функционировать соответствующий измерительный модуль полу комплекта Б и связанный с ним по питанию модуль МУП2(Б).

## ВЕРИФИКАЦИЯ МОДЕЛИ БИВК

После создания формальной модели БИВК необходимо эту модель верифицировать. Необходимо проверить параметры реконфигурации. Для этого в режиме отладки модели проверяем параметры реконфигурации и функционирования, по данному описанию функционирования.

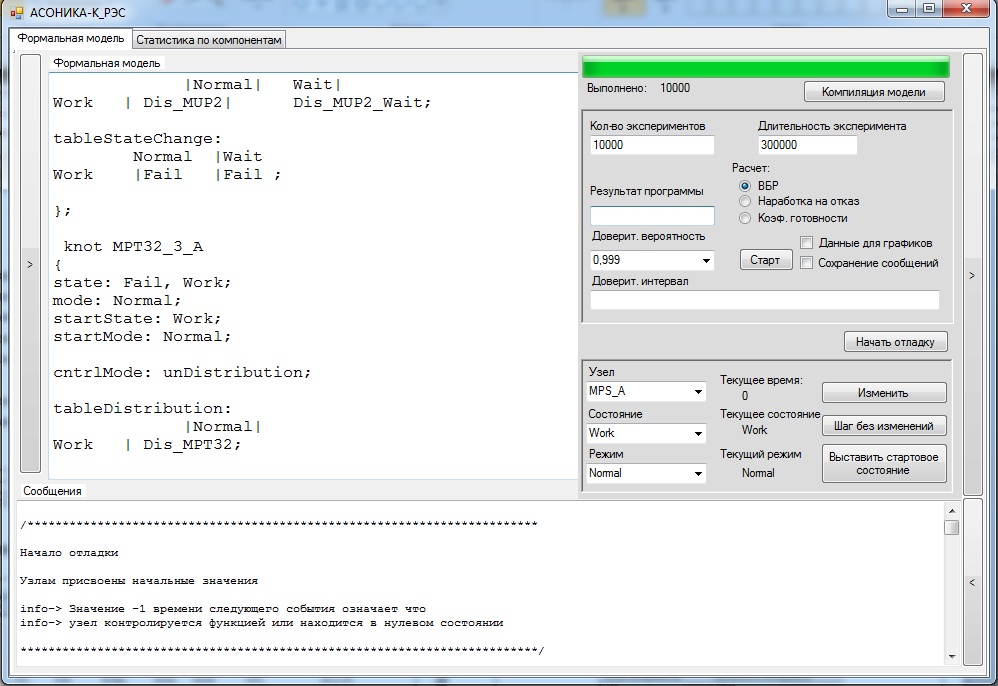


Рис. 2.10. Главное окно АСОНИКА-К-РЭС

Для проверки параметров реконфигурации, в панели управления отладкой формальной модели, для нужного узла выбираем режим и состояние, нажимаем «Изменить» (рис. 2.10).

Все изменения наблюдаем в панели отображения элементов формальной модели (рис.2.11).

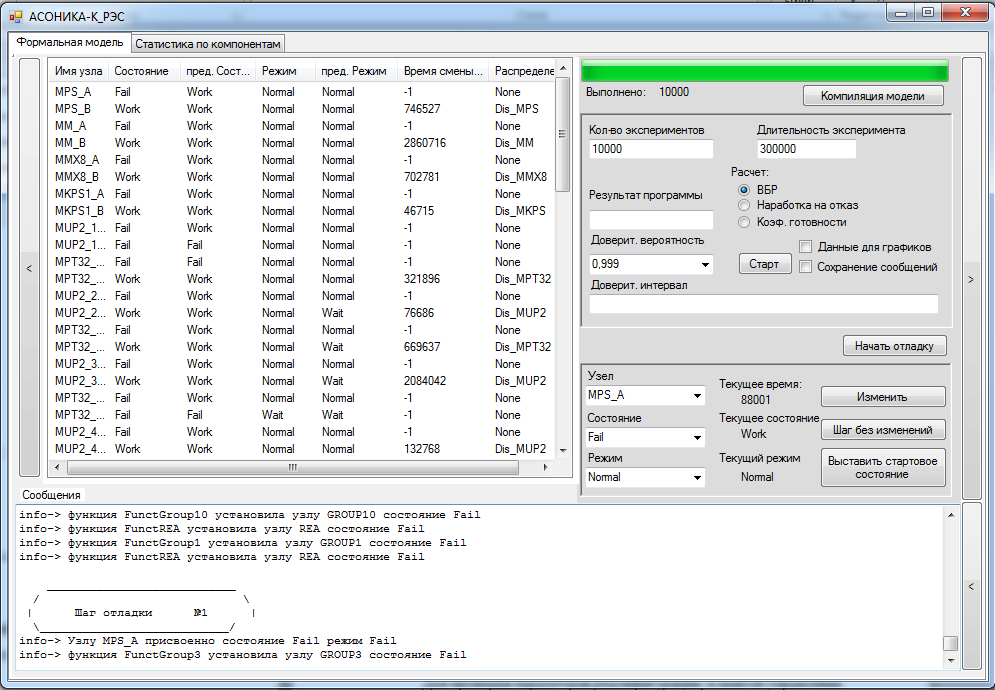


Рис. 2.11. Панель отображения элементов формальной модели

В этой же панели показана информация о работе компонентов. В таблице панели отображения элементов формальной модели видно следующее (рис. 2.11.) - имя узла, состояние, предыдущее состояние, состояние до изменения, режим, предыдущий режим, время смены состояния, распределения.

## МОДЕЛИРОВАНИЕ МОДЕЛИ БИВК

После заключения о правильности формальной модели можно перейти к расчету, для этого в блоке управления параметрами моделирования. Для всех расчетов необходимо указать кол-во экспериментов, а в случае расчета ВБР время (длительность эксперимента).

На рис. 2.10 справа изображен блок управления параметрами моделирования.

От количества проведенных экспериментов зависит точность расчетов, ее оценка в виде ширины доверительного интервала для различных доверительных вероятностей выводится в соответствующее поле. При проведении расчета интенсивности отказов возможно сохранить в файл «graf.dat» численные ряды полученных значений вероятности безотказной работы и интенсивности отказов. Для этого перед началом моделирования необходимо установить флаг «данные для графиков». На основе данного файла возможно построить графики ВБР и интенсивности отказов исследуемой РЭА в программе Microsoft Office Excel.

Информация оператору о ходе эксперимента указывается в графе Сообщения, с помощью неё можно определить текущее системное время и подробную информацию о выполнении функций и событий.

После верификации модели, можно приступить к расчету надежности. В панели управления экспериментом выбираем необходимый расчет, в данном случае вероятность безотказной работы, количество экспериментов, доверительную вероятность и длительность (указывается в минутах), нажимаем кнопку Старт (рис. 2.10).

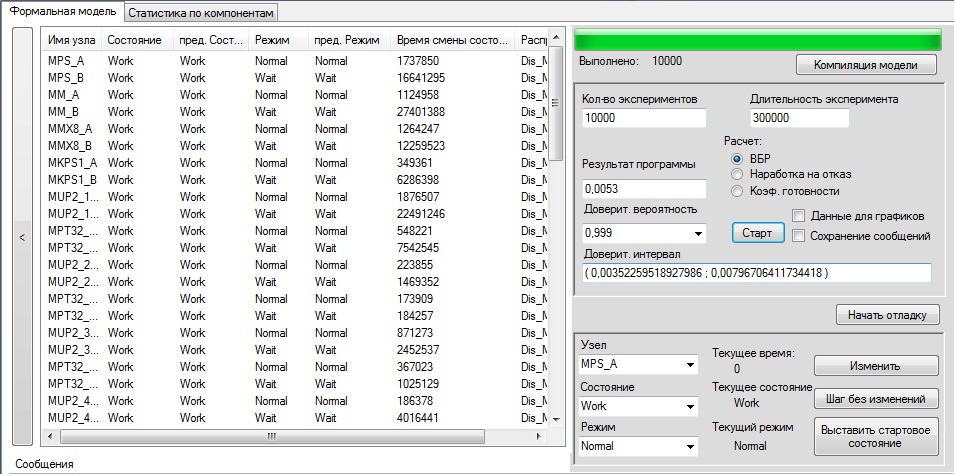


Рис. 10.1 Расчет вероятности безотказной работы в АСОНИКА-К-РЭС

При расчете ВБР в поле “Доверительный интервал” и “Результат” появится результат расчета (рис. 2.10).

В панели контроля элементов формальной модели для остальных компонентов: статистика состояния на конец эксперимента, среднее время нахождения в каждом состоянии

## АНАЛИЗ РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ БИВК

После верификации над моделью были проведены имитационные эксперименты, в количестве 100 000. Это дает значение вероятности безотказной работы БИВК за 10 лет (87600 часов) до 4 знака. На основе полученных данных, были построены график вероятности безотказной работы (рис. 14.1) и график интенсивности отказов (рис 14.2). Расчет проводился методом перебора состояний, но из-за сложности после первого шага реконфигураций отказ любого компонента считался отказом всего БИВК, что верно лишь для случая отказавшего модуля МКПС, в то время как отказ большинства компонентов не приводит к отказу БИВК. За счет этого в результате расчетов была получена нижняя оценка вероятности безотказной работы за интервал времени 10 лет.

## МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПРИ РАБОТЕ С АСОНИКА-К-РЭС

Исходными данными для проведения исследований надежности с применением программы АСОНИКА-К-РЭС являются:

1. Интенсивности отказов компонентов во всех режимах работы

2. Описание структуры резервирования и реконфигурации

3. Критерии отказа

4. Описание функционирования

По этим данным необходимо составить текстовую модель ЭС, ее верифицировать и провести серию экспериментов. Результатами моделирования является средняя наработка на отказ и график ВБР. При построении формальной модели рекомендуется придерживаться приведенной ниже методики.

При построении формальной модели рекомендуется придерживаться приведенной ниже методики:

1. Анализ исходных данных.

1.1. Составление списка компонентов (элементарных и составных).

1.2. Составление списка состояний и режимов для каждого элементарного компонента.

Состояния и режимы должны отличаться распределениями или функционально.

Под функциональным различием понимается влияние состояния одного компонента на другие.

1.3. Составление полного списка возможных распределений (интенсивностей отказов).

1.4. Установление соответствия компонент – состояние – режим – распределение (интенсивность отказов).

1.5. Составление списка стартовых состояний компонентов .

1.6. Анализ резервированных групп.

1.6.1. Составление списка резервированных групп.

1.6.2. Составление списка компонентов резервированной группы (для каждой группы).

1.6.3. Составление списка критериев отказов (для каждой резервированной группы).

1.7. Составление списка реконфигураций (причина - следствие).

2. Программирование формальной модели.

2.1. Ввод списка законов распределения в рамках синтаксиса языка (по спискам пункта) в формате объявления distribution.

2.2. Ввод моделей элементарных компонентов в формате объявления knot (по спискам пунктов ,,,).

2.3. Ввод дополнительных моделей составных компонентов для каждой резервной группы (по спискам пункта ).

2.4. Ввод критериев отказа для резервированных групп в виде вычислительных процедур (function).

2.5. Ввод дополнительных составных компонентов (при необходимости).

2.6. Ввод действий реконфигурации (switch\_event).

2.7. Ввод главного компонента – программной модели ЭС (general knot).

2.8. Ввод условий отказа ЭС в виде вычислительной процедуры(function).

2.9. Поиск и исправление синтаксических ошибок.

При построении формальной модели рекомендуется компоновать блоки в следующей последовательности:

1. Объявления законов распределения (distribution).

2. Объявления элементарных компонентов.

3. Объявления составных компонентов.

4. Функции вычисляющие состояние составных компонентов.

5. События (switch\_event).

Такой порядок расположения блоков модели обеспечивает

своевременное объявление всех объектов и удобство отладки и редактирования формальной модели.

Перед началом построения формальной модели исследуемой РЭА рекомендуется внимательно ознакомиться с ним.

Для написания текста формальной модели может использоваться как окно программы «АСОНИКА-К-РЭС», так и любой текстовый редактор.

## ПРЕДЛОЖЕНИЯ, РЕКОМЕНДАЦИИ ПО РАЗВИТИЮ ПОДСИСТЕМЫ АСОНИКА-К-РЭС

Для удобства работы в системе АСОНИКА-К-РЭС мною были предложены следующие рекомендации:

* Для снижения количества времени и облегчения работы при расчетах надежности, добавить функцию создания графиков вероятности безотказной работы и интенсивности отказов на основе полученных данных внутри системы.
* Добавить возможность использовать систему в полноэкранном режиме.
* Добавить возможность открывать и сохранять модели внутри системы.
* Переход от построения формальной модели в текстовом виде, к построению при помощи графических моделей компонентов и переходом системы на графическую среду моделирования.
* Возможность смены состояний и режимов в панели отображения элементов

## РЕЗУЛЬТАТЫ РАСЧЕТА

П

После верификации над моделью были проведены имитационные эксперименты, в количестве 100 000. Это дает значение вероятности безотказной работы БИВК за 10 лет (87600 часов) до 4 знака. На основе полученных данных, были построены график вероятности безотказной работы (рис. 2.12) и график интенсивности отказов (рис 2.13).

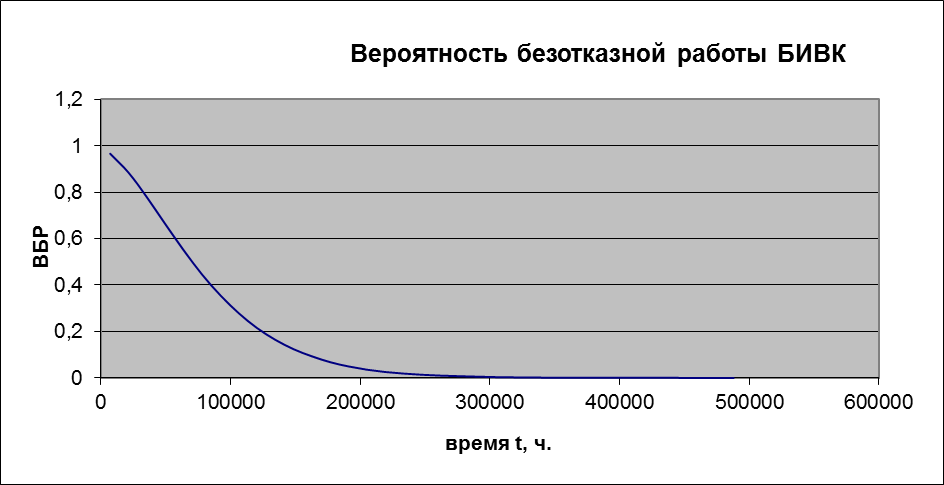


Рис. 2.12 График вероятности безотказной работы БИВК

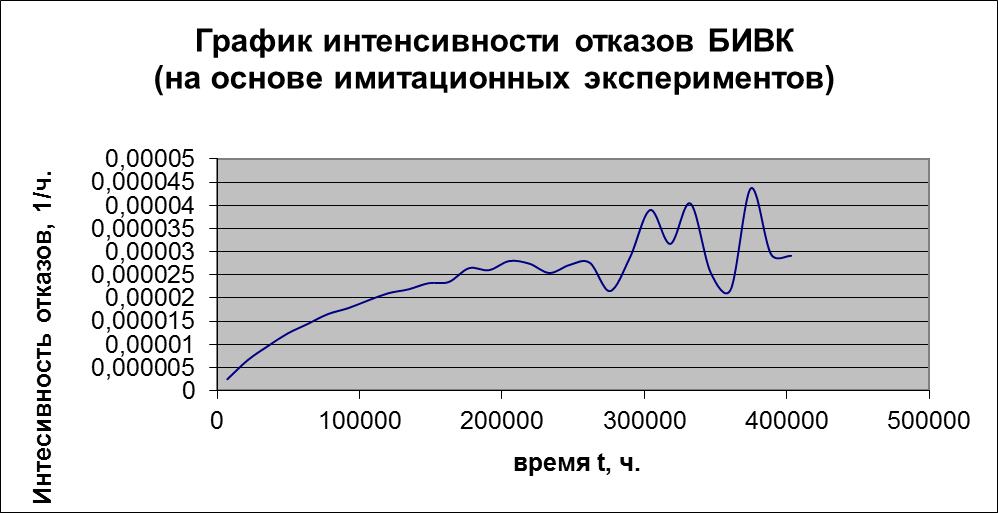


Рис. 2.13 График интенсивности отказов БИВК

Значение ВБР от 10 лет по рузельтатам имитационного моделирования – 0,42476, что ожидаемо выше полученных данных произведенных аналитическим расчетом – 0,3807.

# КОНСТРУКТИВНО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

## ТЕХНОЛОГИЯ СОЗДАНИЯ БЛОК-СХЕМЫ В СООТВЕТСТВИИ С ЕСПД

Единая система программной документации (ЕСПД) – комплекс государственных стандартов, устанавливающих взаимоувязанные правила разработки, оформления и обращения программ и программной документации.  
  
 В стандартах ЕСПД установлены требования, регламентирующие разработку, сопровождение, изготовление и эксплуатацию программ, что обеспечивает возможность:

* унификации программных изделий для взаимного обмена программами, применения ранее разработанных программ в новых разработках;
* снижения трудоемкости и повышения эффективности разработки, сопровождения, изготовления и эксплуатации программных изделий;
* автоматизации изготовления и хранения программной документации.

Сопровождение программы включает анализ функционирования, развитие и совершенствование программы, а также внесение изменений в нее с целью устранения ошибок.[1][4]

## СОЗДАНИЕ БЛОК-СХЕМ МЕТОДИК В СООТВЕТСТВИИ С ЕСПД

## При помощи блок схем были оформлены методические указания при создании формальной модели в системе АСОНИКА-К-РЭС. Подробный алгоритм описан в пункте 2.13 “Методические указания при работе с асоника-к-рэс”.

На рис 3.2. показана общая методика в виде блок-схемы в соответствии с ЕСПД.

Первый этап “Сбор исходных данных”, подразумевает сбор необходимых входных данных для работы в системе АСОНИКА-К-РЭС:

1. Интенсивности отказов компонентов во всех режимах работы

2. Описание структуры резервирования и реконфигурации

3. Критерии отказа

4. Описание функционирования

Второй этап «Создание модели БИВК», на этом этапе оператор создает модель ЭС, стараясь придерживаться следующей методики:

1. Объявления законов распределения (distribution).

2. Объявления элементарных компонентов.

3. Объявления составных компонентов.

4. Функции вычисляющие состояние составных компонентов.

5. События (switch\_event).

Третий этап «Верификация модели», проверка подлинности и соответствия формальной модели описанию функционирования модели.

Четвертый этап «Расчет надежности», установка необходимых параметров и непосредственно эксперименты по расчету надежности.

Пятый этап «Анализ результатов моделирования», на основе полученных данных, построение графиков вероятности и безотказной работы.



Рис. 3.2. Методика создания формальной модели при помощи АСОНИКА-К-РЭС

# ЗАКЛЮЧЕНИЕ

В результате проведенного дипломного проектирования были изучены и описаны системы реконфигурации БИВК. Были рассмотрены различные методы оценки показателей безотказности, проведен обзор программного обеспечения имитационного моделирования. Изучено программное средство имитационного моделирования АСОНИКА-К-РЭС. При помощи которого в результате дипломного проектирования стала модель ЭС с реконфигурируемой структурой. А так же расчет надежности по созданной модели.

В ходе работы был применен новый метод расчета показателей надежности реконфигурируемой электронной аппаратуры. Метод основан на специализированном языке описания отказов используемый в программном средстве АСОНИКА-К-РЭС, который позволяет создать модель ЭС и провести эксперименты по расчету показателей надежности.

В данном дипломном проектировании было продемонстрировано создание модели расчета надежности бортового интегрированного вычислительного комплекса. Тем самым можно заключить о достижении поставленной в проектировании цели.

# ОХРАНА ТРУДА

## ЭЛЕКТРОБЕЗОПАСНОСТЬ

Электробезопасность - система организационных и технических мероприятий и средств, обеспечивающих защиту людей от вредного и опасного воздействия электрического тока, электрической дуги, электромагнитного поля и статического электричества.

Электробезопасность должна обеспечиваться:

* конструкцией электроустановок;
* техническими способами и средствами защиты;
* организационными и техническими мероприятиями.
* Электроустановки и их части должны быть выполнены таким образом, чтобы работающие не подвергались опасным и вредным воздействиям электрического тока и электромагнитных полей, и соответствовать требованиям электробезопасности.ГОСТ 12.1.019-96 (2001)

Проходя через тело человека, электрический ток оказывает следующие воздействия:

* Термическое — нагрев тканей и биологической среды;
* Электролитическое — разложение крови и плазмы;
* Биологическое — способность тока возбуждать и раздражать живые ткани организма;
* Механическое — возникает опасность механического травмирования в результате судорожного сокращения мышц.

Электрический ток, воздействуя на человека, приводит к травмам:

Общие травмы:

* 1 степень - судорожное сокращение мышц, без потери сознания;
* 2 степень - судорожное сокращение мышц, с потерей сознания;
* 3 степень - потеря сознания с нарушением работы органов дыхания и кровообращения;
* 4 степень – остановка сердца и состояние клинической смерти.

Местные травмы:

* Электрические ожоги;
* Электрический шок;
* Электроавтольмия.

Тяжесть поражения электрическим током зависит от:

* Величины тока.
* Времени протекания.
* Пути протекания.
* Рода и частоты тока.
* Сопротивления человека.
* Окружающей среды.
* Состояния человека.
* Пола и возраста человека.

Наиболее опасным является переменный ток 20 - 100Гц. Так как компьютер питается от сети переменного тока частотой 50Гц, то этот ток является опасным для человека.

## ОБЕСПЕЧЕНИЕ ЭЛЕКТРОБЕЗОПАСНОСТИ ТЕХНИЧЕСКИМИСПОСОБАМИ И СРЕДСТВАМИ

Для обеспечения защиты от случайного прикосновения к токоведущим частям необходимо применять следующие способы и средства:

* защитные оболочки;
* защитные ограждения (временные или стационарные);
* безопасное расположение токоведущих частей;
* изоляцию токоведущих частей (рабочую, дополнительную, усиленную, двойную);
* изоляцию рабочего места;
* малое напряжение;
* защитное отключение;
* предупредительная сигнализация, блокировка, знаки безопасности.

Для обеспечения защиты от поражения электрическим током при прикосновении к металлическим нетоковедущим частям, которые могут оказаться под напряжением в результате повреждения изоляции, применяют следующие способы:

* защитное заземление;
* зануление;
* выравнивание потенциала;
* систему защитных проводов;
* защитное отключение;
* изоляцию нетоковедущих частей;
* электрическое разделение сети;
* малое напряжение;
* контроль изоляции;
* компенсацию токов замыкания на землю;
* средства индивидуальной защиты.

Технические способы и средства применяют раздельно или в сочетании друг с другом так, чтобы обеспечивалась оптимальная защита. Требования к техническим способам и средствам защиты должны быть установлены в стандартах и технических условиях.

Для защиты от поражения электрическим током используется зануление - преднамеренное электрическое соединение с нулевым защитным проводником металлических токоведущих проводников, которые могут оказаться под напряжением (Рис.1).

Зануление применяется в трехфазных сетях с глухо заземленной нейтралью, в установках до 1000В и является основным средством обеспечения электробезопасности.

Принцип защиты пользователей при занулении заключается в отключении сети за счет тока короткого замыкания, который вызывает отключение ПЭВМ от сети.

Типовая конфигурация компьютеризированного рабочего места:

ПК на основе процессора Intel Core 2 Duo c необходимым набором устройств ввода-вывода и хранения информации (ZIP-drive, CD/DVD-RW, Floppy “3.5”);

Лазерный принтер Samsungscx-3200;

цветной монитор BenQGL2250 “21” (TCO''03):

* разрешение по горизонтали (max) - 1920 пикселей;
* разрешение по вертикали (max) - 1080 пикселей;
* частота кадровой развертки при максимальном разрешении - 60 Гц;

|  |  |
| --- | --- |
|  |  |

Из рис. 1 видно, что для схемы зануления необходимо наличие глухого заземления нейтрали обмоток источника тока и повторного заземления нулевого защитного проводника. Заземление нейтрали обмоток источника тока, питающего сеть, предназначено для снижения напряжения занулённых корпусов (а следовательно, нулевого защитного проводника) относительно земли до безопасного значения при замыкании фазы на землю. Повторное заземление нулевого защитного проводника практически не влияет на отключающую способность схемы зануления и в этом смысле без него можно обойтись. Однако при отсутствии повторного заземления нулевого защитного проводника возникает опасность для людей, прикасающихся к занулённому оборудованию в период, пока существует замыкание фазы на корпус. Это связано с тем, что напряжение относительно земли участка нулевого защитного проводника за местом обрыва и всех присоединённых к нему корпусов исправного электрооборудования окажется близким по значению фазному напряжению сети. Это напряжение будет существовать длительно, поскольку повреждённая установка автоматически не отключится, и её трудно обнаружить среди исправных установок, чтобы отключить вручную.[6]

## РАСЧЕТ ЗАЩИТНОГО ЗАНУЛЕНИЯ НА РАБОЧЕМ МЕСТЕ

|  |
| --- |
|  |
| Рис 2.: Схема защитного зануления |

 (1), где:



 - ток короткого замыкания [А];

Uф - фазовое напряжение [B];

rm - сопротивление катушек трансформатора [Ом];

rнзп - сопротивление нулевого защитного проводника [Ом].

По заданным параметрам определим возможный Jк.з.

Uф = 220 В

rm =0,350 Ом (по паспорту )

(2), где:



- удельное сопротивление материала проводника[(Ом\*мм)÷м];



l - длина проводника [м];

s –поперечное сечение проводника [мм2].

Р медь= 0,0175 Ом\*м

=425 м ; =155 м ; =45 м







; 7,218 Ом







По величине определим с каким необходимо включить в цепь питания ПЭВМ автомат.





, где K – качество автомата.

К=3.



Следовательно, для отключения ПЭВМ от сети в случае короткого замыкания или других неисправностей в цепь питания ПЭВМ необходимо ставить автомат с

 = 10 А.

# ЭКОЛОГИЧЕСКАЯ ЧАСТЬ ПРОЕКТА

## МИКРОКЛИМАТ В РАБОЧЕЙ ЗОНЕ

Оптимальная температура для легких работ в теплый период года равна 22...25°С, а допустимая (при значительном избытке явной теплоты) на 5 °С выше средней температуры воздуха наружи в 13 ч самого жаркого месяца, но не выше 28 °С. Относительная влажность при этом должна быть менее 55 %. Скорость движения воздуха в помещении 0,2...0,5 м/с (оптимальная скорость 0,2 м/с). В холодный период года на тех же работах оптимальная температура составляет 20... 23°С, допустимая — 19... 25°С. В ряде случаев допускают определенные отклонения от норм. Регистрирующим прибором-термографом типа М-1В измеряют температуру воздуха, а психрометром — влажность. Рабочим местом считается такое, где работающий находится более половины своего рабочего времени или более 2 ч непрерывно. Если же люд работают в различных местах рабочей зоны, то она вся считаете постоянным рабочим местом.  
        Микроклимат в рабочей зоне определяется действующими на организм человека сочетаниями температуры, влажности и скорости движения воздуха, а также температурой окружающих поверхностей. Повышенная влажность затрудняет теплоотдачу организма путем испарений при высокой температуре воздуха и способствует перегреву, а при низкой температуре, наоборот, усиливает теплоотдачу, способствуя переохлаждению. Оптимальны такие параметры микроклимата, которые при длительном и систематическом воздействии на человека обеспечивают сохранение нормального функционального и теплового состояния организма без напряжения реакций терморегуляции, что создает ощущение теплового комфорта и служит предпосылкой для высокой работоспособности. Оптимальный микроклимат можно поддерживать только в том случае, если предприятие оснащено кондиционерами и установками кондиционирования микроклимата. В других возможных случаях надлежит обеспечивать допустимые микроклиматические условия, то есть такие, при которых и возможны возникновения напряжения при терморегуляции организма, но не выходящие за пределы физиологических приспособительных возможностей. При этом не возникают нарушения состояния здоровья, но может наблюдаться ухудшение самочувствия и понижение работоспособности.  
       Нормы гигиенические зависят от категорий работ по степени физической нагрузки, а также от тёплого или холодного периода года и от избытка явной теплоты, поступающей в помещение оборудования, нагретых материалов, отопительных приборов, людей и солнечного света. То есть зависят от разности между явной теплотой и теплопотерями при расчете параметров наружного воздуха учете всех мероприятий по уменьшению теплопотерь. Избытки теплоты считают незначительными, если они составляют не более 23 Дж/(м3 х с), если же избытки теплоты больше 23 Дж/(м3 х с), то помещение относится к горячему цеху.

Гигиенические нормы на параметры микроклимата в рабочей зоне даны в ГОСТ 12.1.005-88 (2001). Рабочей зоной, считается пространство высотой до 2 метров над уровнем пола или площадки, на которой расположены рабочие места.

Рабочим местом считается то место, где работающий человек находится более половины своего рабочего времени или более двух часов непрерывно. Если же человек работает в различных местах рабочей зоны, то она вся считается постоянным рабочим местом.

## ЗАЩИТА ОТ ШУМА

По характеру спектра шум следует подразделять на:

* широкополосный с непрерывным спектром шириной более одной октавы;
* тональный, в спектре которого имеются выраженные дискретные тона. Тональный характер шума для практических целей (при контроле его параметров на рабочих местах) устанавливают измерением в треть октавных полосах частот по превышению уровня звукового давления в одной полосе над соседними не менее чем на 10 дБ.

По временным характеристикам шум следует подразделять на:

* постоянный, уровень звука которого за 8-часовой рабочий день (рабочую смену) изменяется во времени не более чем на 5 дБ *А* при измерениях на временной характеристике “медленно” шумомера по ГОСТ 17187;
* непостоянный, уровень звука которого за 8-часовой рабочий день (рабочую смену) изменяется во времени более чем на 5 дБ *А* при измерениях на временной характеристике “медленно” шумомера по ГОСТ 17187.

Непостоянный шум следует подразделять на:

* колеблющийся во времени, уровень звука которого непрерывно изменяется во времени;
* прерывистый, уровень звука которого ступенчато изменяется (на 5 дБ *А* и более), причем длительность интервалов, в течение которых уровень остается постоянным, составляет 1 с и более;
* импульсный, состоящий из одного или нескольких звуковых сигналов, каждый длительностью менее 1 с, при этом уровни звука, измеренные в дБ *AI* и дБ *А* соответственно на временных характеристиках “импульс” и “медленно” шумомера по ГОСТ 17187, отличаются не менее чем на 7 дБ.

Согласно ГОСТ 12.1.003-83 (1999) при разработке технологических процессов, проектировании, изготовлении и эксплуатации машин, производственных зданий и сооружений, а также при организации рабочих мест следует принимать все необходимые меры по снижению шума, воздействующего на человека, до значений, не превышающих допустимые.

В помещениях, где размещены различные шумные агрегаты, вычислительные машины (принтеры и тому подобные), уровень шума не должен превышать 75дБА, в тех помещениях, где стоят персональные машины, уровень шума допускается максимум 65 дБА.

Категорически запрещается даже кратковременное пребывание в зонах с октавными уровнями звукового давления свыше 135 дБ в любой октавной полосе.

Защита от шума должна обеспечиваться разработкой шумобезопасной техники, применением средств и методов коллективной защиты, в том числе строительно-акустических, применением средств индивидуальной защиты.

В первую очередь следует использовать средства коллективной защиты. По отношению к источнику возбуждения шума коллективные средства защиты подразделяются на средства, снижающие шум в источнике его возникновения, и средства, снижающие шум на пути его распространения от источника до защищаемого объекта.

Снижение шума в источнике осуществляется за счет улучшения конструкции машины или изменения технологического процесса. Средства, снижающие шум в источнике его возникновения в зависимости от характера шумообразования подразделяются на средства, снижающие шум механического происхождения, аэродинамического и гидродинамического происхождения, электромагнитного происхождения.

Методы и средства коллективной защиты в зависимости от способа реализации подразделяются на строительно-акустические, архитектурно-планировочные и организационно - технические и включают в себя:

* изменение направленности излучения шума;
* рациональную планировку предприятий и производственных помещений;
* акустическую обработку помещений;
* применение звукоизоляции. [4]

Средства индивидуальной защиты включают в себя беруши.

Список литературы

1. Единая система программной документации. Общие положения

ГОСТ 19.001-77

2. Строгалев, В.П. имитационное моделирование. / В.П. Строгалев, И.О. Толкачева. – М.:Издательство МГТУ им. Н.Э. Баумана, 2008. – 280 с.

3. *Карпов Ю. Г.* Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. — СПб: БХВ-Петербург, 2006. — 400 с. — ISBN 5-94157-148-8

# 4. "Cистема стандартов безопасности труда. шум. общие требования безопасности. гост 12.1.003-83 (ст сэв 1930-79)" (утв. постановлением госстандарта ссср от 06.06.83 n 2473) (ред. от 19.12.88)

5. Менеджмент риска. Руководство по применению методов анализа надежности

ГОСТ Р 51901.5-2005

6. Лабораторный практикум по дисциплине "Безопасность жизнедеятельности" для студентов всех специальностей: учебное пособие / Ю.А. Амелькович, Ю.В. Анищенко, А.Н. Вторушина, М.В. Гуляев, М.Э. Гусельников, А.Г. Дашковский, Т.А. Задорожная, В.Н. Извеков, А.Г. Кагиров, К.М. Костырев, В.Ф. Панин, А.М. Плахов, С.В. Романенко. - Томск: Издательство Томского политехнического университета, 2010. - 236 с.

# ПРИЛОЖЕНИЕ

**МОДЕЛЬ БИВК ИСПОЛЬЗУЮЩАЯСЯ ПРИ РАСЧЕТЕ НАДЕЖНОСТИ С СИСТЕМЕ АСОНИКА-К-РЭС**

distribution Dis\_MPS (1,27e-6);

distribution Dis\_MPS\_Wait (1,2e-7);

distribution Dis\_MM (0,695e-6);

distribution Dis\_MM\_Wait (6,9e-8);

distribution Dis\_MMX8 (1,852e-6);

distribution Dis\_MMX8\_Wait (1,8e-7);

distribution Dis\_MKPS (1,587e-6);

distribution Dis\_MKPS\_Wait (1,58e-7);

distribution Dis\_MUP2 (1,228e-6);

distribution Dis\_MUP2\_Wait (1,22e-7);

distribution Dis\_MPT32 (3,105e-6);

distribution Dis\_MPT32\_Wait (3,05e-7);

distribution Dis\_MPA64 (2,079e-6);

distribution Dis\_MPA64\_Wait (2,02e-7);

distribution Dis\_MPC80 (2,237e-6);

distribution Dis\_MPC80\_Wait (2,17e-7);

distribution Dis\_MPN32 (2,937e-6);

distribution Dis\_MPN32\_Wait (2,88e-7);

knot MPS\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MPS;

tableStateChange:

Normal|

Work |Fail ;

};

knot MPS\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MPS| Dis\_MPS\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MM\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MM;

tableStateChange:

Normal|

Work |Fail ;

};

knot MM\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MM| Dis\_MM\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MMX8\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MMX8;

tableStateChange:

Normal|

Work |Fail ;

};

knot MMX8\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MMX8| Dis\_MMX8\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MKPS1\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MKPS;

tableStateChange:

Normal|

Work |Fail ;

};

knot MKPS1\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MKPS| Dis\_MKPS\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MUP2\_1\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MUP2;

tableStateChange:

Normal|

Work |Fail ;

};

knot MUP2\_1\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MUP2| Dis\_MUP2\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MPT32\_1\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MPT32;

tableStateChange:

Normal|

Work |Fail ;

};

knot MPT32\_1\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MPT32| Dis\_MPT32\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MUP2\_2\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MUP2;

tableStateChange:

Normal|

Work |Fail ;

};

knot MUP2\_2\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MUP2| Dis\_MUP2\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MPT32\_2\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MPT32;

tableStateChange:

Normal|

Work |Fail ;

};

knot MPT32\_2\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MPT32| Dis\_MPT32\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MUP2\_3\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MUP2;

tableStateChange:

Normal|

Work |Fail ;

};

knot MUP2\_3\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MUP2| Dis\_MUP2\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MPT32\_3\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MPT32;

tableStateChange:

Normal|

Work |Fail ;

};

knot MPT32\_3\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MPT32| Dis\_MPT32\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MUP2\_4\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MUP2;

tableStateChange:

Normal|

Work |Fail ;

};

knot MUP2\_4\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MUP2| Dis\_MUP2\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MPC80\_4\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MPC80 ;

tableStateChange:

Normal|

Work |Fail ;

};

knot MPC80\_4\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MPC80 | Dis\_MPC80\_Wait ;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MUP2\_5\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MUP2;

tableStateChange:

Normal|

Work |Fail ;

};

knot MUP2\_5\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MUP2| Dis\_MUP2\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MPC80\_5\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MPC80;

tableStateChange:

Normal|

Work |Fail ;

};

knot MPC80\_5\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MPC80| Dis\_MPC80\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MUP2\_6\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MUP2;

tableStateChange:

Normal|

Work |Fail ;

};

knot MUP2\_6\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MUP2| Dis\_MUP2\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MPN32\_6\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MPN32;

tableStateChange:

Normal|

Work |Fail ;

};

knot MPN32\_6\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MPN32| Dis\_MPN32\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MKPS2\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MKPS;

tableStateChange:

Normal|

Work |Fail ;

};

knot MKPS2\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MKPS| Dis\_MKPS\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MUP2\_7\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MUP2;

tableStateChange:

Normal|

Work |Fail ;

};

knot MUP2\_7\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MUP2| Dis\_MUP2\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MPT32\_7\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MPT32;

tableStateChange:

Normal|

Work |Fail ;

};

knot MPT32\_7\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MPT32| Dis\_MPT32\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MUP2\_8\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MUP2;

tableStateChange:

Normal|

Work |Fail ;

};

knot MUP2\_8\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MUP2| Dis\_MUP2\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MPT32\_8\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MPT32;

tableStateChange:

Normal|

Work |Fail ;

};

knot MPT32\_8\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MPT32| Dis\_MPT32\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MUP2\_9\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MUP2;

tableStateChange:

Normal|

Work |Fail ;

};

knot MUP2\_9\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MUP2| Dis\_MUP2\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MPA64\_9\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MPA64;

tableStateChange:

Normal|

Work |Fail ;

};

knot MPA64\_9\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MPA64| Dis\_MPA64\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MUP2\_10\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MUP2;

tableStateChange:

Normal|

Work |Fail ;

};

knot MUP2\_10\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MUP2| Dis\_MUP2\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MPC80\_10\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MPC80;

tableStateChange:

Normal|

Work |Fail ;

};

knot MPC80\_10\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MPC80| Dis\_MPC80\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MUP2\_11\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MUP2;

tableStateChange:

Normal|

Work |Fail ;

};

knot MUP2\_11\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MUP2| Dis\_MUP2\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot MPC80\_11\_A

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unDistribution;

tableDistribution:

|Normal|

Work | Dis\_MPC80;

tableStateChange:

Normal|

Work |Fail ;

};

knot MPC80\_11\_B

{

state: Fail, Work;

mode: Normal,Wait;

startState: Work;

startMode: Wait;

cntrlMode: unDistribution;

tableDistribution:

|Normal| Wait|

Work | Dis\_MPC80| Dis\_MPC80\_Wait;

tableStateChange:

Normal |Wait

Work |Fail |Fail ;

};

knot GROUP1

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unFunction;

tableStateChange:

Normal

Work |FunctGroup1 ;

};

knot GROUPMKPS1

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unFunction;

tableStateChange:

Normal

Work |FunctGroupMKPS1 ;

};

knot GROUP2

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unFunction;

tableStateChange:

Normal

Work |FunctGroup2 ;

};

knot GROUP3

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unFunction;

tableStateChange:

Normal

Work |FunctGroup3 ;

};

knot GROUP4

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unFunction;

tableStateChange:

Normal

Work |FunctGroup4 ;

};

knot GROUP5

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unFunction;

tableStateChange:

Normal

Work |FunctGroup5 ;

};

knot GROUP6

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unFunction;

tableStateChange:

Normal

Work |FunctGroup6 ;

};

knot GROUPMKPS2

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unFunction;

tableStateChange:

Normal

Work |FunctGroupMKPS2 ;

};

knot GROUP7

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unFunction;

tableStateChange:

Normal

Work |FunctGroup7 ;

};

knot GROUP8

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unFunction;

tableStateChange:

Normal

Work |FunctGroup8 ;

};

knot GROUP9

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unFunction;

tableStateChange:

Normal

Work |FunctGroup9 ;

};

knot GROUP10

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unFunction;

tableStateChange:

Normal

Work |FunctGroup10 ;

};

knot GROUP11

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unFunction;

tableStateChange:

Normal

Work |FunctGroup11 ;

};

knot GROUP0

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unFunction;

tableStateChange:

Normal

Work |FunctGroup0 ;

};

general knot REA

{

state: Fail, Work;

mode: Normal;

startState: Work;

startMode: Normal;

cntrlMode: unFunction;

tableStateChange:

Normal

Work |FunctREA ;

};

switch\_Event MPS\_A\_FAIL (->MPS\_A:Fail)

{

set\_state(MM\_A:Fail);

set\_state(MMX8\_A:Fail);

set\_state(MKPS1\_A:Fail);

set\_state(MUP2\_1\_A:Fail);

set\_state(MPT32\_1\_A:Fail);

set\_state(MUP2\_2\_A:Fail);

set\_state(MPT32\_2\_A:Fail);

set\_state(MUP2\_3\_A:Fail);

set\_state(MPT32\_3\_A:Fail);

set\_state(MUP2\_4\_A:Fail);

set\_state(MPC80\_4\_A:Fail);

set\_state(MUP2\_5\_A:Fail);

set\_state(MPC80\_5\_A:Fail);

set\_state(MUP2\_6\_A:Fail);

set\_state(MPN32\_6\_A:Fail);

set\_state(MKPS2\_A:Fail);

set\_state(MUP2\_7\_A:Fail);

set\_state(MPT32\_7\_A:Fail);

set\_state(MUP2\_8\_A:Fail);

set\_state(MPT32\_8\_A:Fail);

set\_state(MUP2\_9\_A:Fail);

set\_state(MPA64\_9\_A:Fail);

set\_state(MUP2\_10\_A:Fail);

set\_state(MPC80\_10\_A:Fail);

set\_state(MUP2\_11\_A:Fail);

set\_state(MPC80\_11\_A:Fail);

set\_mode (MPS\_B:Normal);

set\_mode(MM\_B:Normal);

set\_mode(MMX8\_B:Normal);

set\_mode(MKPS1\_B:Normal);

set\_mode(MUP2\_1\_B:Normal);

set\_mode(MPT32\_1\_B:Normal);

set\_mode(MUP2\_2\_B:Normal);

set\_mode(MPT32\_2\_B:Normal);

set\_mode(MUP2\_3\_B:Normal);

set\_mode(MPT32\_3\_B:Normal);

set\_mode(MUP2\_4\_B:Normal);

set\_mode(MPC80\_4\_B:Normal);

set\_mode(MUP2\_5\_B:Normal);

set\_mode(MPC80\_5\_B:Normal);

set\_mode(MUP2\_6\_B:Normal);

set\_mode(MPN32\_6\_B:Normal);

set\_mode(MKPS2\_B:Normal);

set\_mode(MUP2\_7\_B:Normal);

set\_mode(MPT32\_7\_B:Normal);

set\_mode(MUP2\_8\_B:Normal);

set\_mode(MPT32\_8\_B:Normal);

set\_mode(MUP2\_9\_B:Normal);

set\_mode(MPA64\_9\_B:Normal);

set\_mode(MUP2\_10\_B:Normal);

set\_mode(MPC80\_10\_B:Normal);

set\_mode(MUP2\_11\_B:Normal);

set\_mode(MPC80\_11\_B:Normal);

} ;

switch\_Event MM\_A\_FAIL (->MM\_A:Fail|->MMX8\_A:Fail)

{

set\_mode (MPS\_B:Normal);

set\_mode(MM\_B:Normal);

set\_mode(MMX8\_B:Normal);

set\_mode(MKPS1\_B:Normal);

set\_mode(MUP2\_1\_B:Normal);

set\_mode(MPT32\_1\_B:Normal);

set\_mode(MUP2\_2\_B:Normal);

set\_mode(MPT32\_2\_B:Normal);

set\_mode(MUP2\_3\_B:Normal);

set\_mode(MPT32\_3\_B:Normal);

set\_mode(MUP2\_4\_B:Normal);

set\_mode(MPC80\_4\_B:Normal);

set\_mode(MUP2\_5\_B:Normal);

set\_mode(MPC80\_5\_B:Normal);

set\_mode(MUP2\_6\_B:Normal);

set\_mode(MPN32\_6\_B:Normal);

set\_mode(MKPS2\_B:Normal);

set\_mode(MUP2\_7\_B:Normal);

set\_mode(MPT32\_7\_B:Normal);

set\_mode(MUP2\_8\_B:Normal);

set\_mode(MPT32\_8\_B:Normal);

set\_mode(MUP2\_9\_B:Normal);

set\_mode(MPA64\_9\_B:Normal);

set\_mode(MUP2\_10\_B:Normal);

set\_mode(MPC80\_10\_B:Normal);

set\_mode(MUP2\_11\_B:Normal);

set\_mode(MPC80\_11\_B:Normal);

} ;

switch\_Event MKPS1\_A\_FAIL (->MKPS1\_A:Fail)

{

set\_state(MUP2\_1\_A:Fail);

set\_state(MPT32\_1\_A:Fail);

set\_state(MUP2\_2\_A:Fail);

set\_state(MPT32\_2\_A:Fail);

set\_state(MUP2\_3\_A:Fail);

set\_state(MPT32\_3\_A:Fail);

set\_state(MUP2\_4\_A:Fail);

set\_state(MPC80\_4\_A:Fail);

set\_state(MUP2\_5\_A:Fail);

set\_state(MPC80\_5\_A:Fail);

set\_state(MUP2\_6\_A:Fail);

set\_state(MPN32\_6\_A:Fail);

set\_mode (MPS\_B:Normal);

set\_mode(MM\_B:Normal);

set\_mode(MMX8\_B:Normal);

set\_mode(MKPS1\_B:Normal);

set\_mode(MUP2\_1\_B:Normal);

set\_mode(MPT32\_1\_B:Normal);

set\_mode(MUP2\_2\_B:Normal);

set\_mode(MPT32\_2\_B:Normal);

set\_mode(MUP2\_3\_B:Normal);

set\_mode(MPT32\_3\_B:Normal);

set\_mode(MUP2\_4\_B:Normal);

set\_mode(MPC80\_4\_B:Normal);

set\_mode(MUP2\_5\_B:Normal);

set\_mode(MPC80\_5\_B:Normal);

set\_mode(MUP2\_6\_B:Normal);

set\_mode(MPN32\_6\_B:Normal);

set\_mode(MKPS2\_B:Normal);

} ;

switch\_Event MKPS2\_A\_FAIL (->MKPS2\_A:Fail)

{

set\_state(MUP2\_7\_A:Fail);

set\_state(MPT32\_7\_A:Fail);

set\_state(MUP2\_8\_A:Fail);

set\_state(MPT32\_8\_A:Fail);

set\_state(MUP2\_9\_A:Fail);

set\_state(MPA64\_9\_A:Fail);

set\_state(MUP2\_10\_A:Fail);

set\_state(MPC80\_10\_A:Fail);

set\_state(MUP2\_11\_A:Fail);

set\_state(MPC80\_11\_A:Fail);

set\_mode (MPS\_B:Normal);

set\_mode(MM\_B:Normal);

set\_mode(MMX8\_B:Normal);

set\_mode(MKPS1\_B:Normal);

set\_mode(MKPS2\_B:Normal);

set\_mode(MUP2\_7\_B:Normal);

set\_mode(MPT32\_7\_B:Normal);

set\_mode(MUP2\_8\_B:Normal);

set\_mode(MPT32\_8\_B:Normal);

set\_mode(MUP2\_9\_B:Normal);

set\_mode(MPA64\_9\_B:Normal);

set\_mode(MUP2\_10\_B:Normal);

set\_mode(MPC80\_10\_B:Normal);

set\_mode(MUP2\_11\_B:Normal);

set\_mode(MPC80\_11\_B:Normal);

} ;

switch\_Event MUP\_6\_A\_FAIL (->MUP2\_6\_A:Fail | ->MPN32\_6\_A:Fail )

{

set\_mode (MPS\_B:Normal);

set\_mode(MM\_B:Normal);

set\_mode(MMX8\_B:Normal);

set\_mode(MKPS1\_B:Normal);

set\_mode(MKPS2\_B:Normal);

set\_mode(MUP2\_6\_B:Normal);

set\_mode(MPN32\_6\_B:Normal);

} ;

switch\_Event MUP\_5\_A\_FAIL (->MUP2\_5\_A:Fail | ->MPC80\_5\_A:Fail )

{

set\_mode (MPS\_B:Normal);

set\_mode(MM\_B:Normal);

set\_mode(MMX8\_B:Normal);

set\_mode(MKPS1\_B:Normal);

set\_mode(MKPS2\_B:Normal);

set\_mode(MUP2\_5\_B:Normal);

set\_mode(MPC80\_5\_B:Normal);

} ;

switch\_Event MUP\_4\_A\_FAIL (->MUP2\_4\_A:Fail | ->MPC80\_4\_A:Fail )

{

set\_mode (MPS\_B:Normal);

set\_mode(MM\_B:Normal);

set\_mode(MMX8\_B:Normal);

set\_mode(MKPS1\_B:Normal);

set\_mode(MKPS2\_B:Normal);

set\_mode(MUP2\_4\_B:Normal);

set\_mode(MPC80\_4\_B:Normal);

} ;

switch\_Event MUP\_3\_A\_FAIL (->MUP2\_3\_A:Fail | ->MPT32\_3\_A:Fail )

{

set\_mode (MPS\_B:Normal);

set\_mode(MM\_B:Normal);

set\_mode(MMX8\_B:Normal);

set\_mode(MKPS1\_B:Normal);

set\_mode(MKPS2\_B:Normal);

set\_mode(MUP2\_3\_B:Normal);

set\_mode(MPT32\_3\_B:Normal);

} ;

switch\_Event MUP\_2\_A\_FAIL (->MUP2\_2\_A:Fail | ->MPT32\_2\_A:Fail )

{

set\_mode (MPS\_B:Normal);

set\_mode(MM\_B:Normal);

set\_mode(MMX8\_B:Normal);

set\_mode(MKPS1\_B:Normal);

set\_mode(MKPS2\_B:Normal);

set\_mode(MUP2\_2\_B:Normal);

set\_mode(MPT32\_2\_B:Normal);

} ;

switch\_Event MUP\_1\_A\_FAIL (->MUP2\_1\_A:Fail|->MPT32\_1\_A:Fail)

{

set\_mode (MPS\_B:Normal);

set\_mode(MM\_B:Normal);

set\_mode(MMX8\_B:Normal);

set\_mode(MKPS1\_B:Normal);

set\_mode(MKPS2\_B:Normal);

set\_mode(MUP2\_1\_B:Normal);

set\_mode(MPT32\_1\_B:Normal);

} ;

function FunctREA

{

return GROUP0 & GROUPMKPS1 & GROUP1 & GROUP2 & GROUP3 & GROUP4 & GROUP5 & GROUP6& GROUPMKPS2& GROUP7& GROUP8& GROUP9& GROUP10& GROUP11;

};

function FunctGroup0

{

return (MPS\_A&MM\_A&MMX8\_A)|( MPS\_B&MM\_B&MMX8\_B);

};

function FunctGroupMKPS1

{

return (MPS\_A&MKPS1\_A)|( MPS\_B&MKPS1\_B);

};

function FunctGroup1

{

return (MPS\_A&MUP2\_1\_A &MPT32\_1\_A)|( MPS\_B&MUP2\_1\_B&MPT32\_1\_B);

};

function FunctGroup2

{

return (MPS\_A&MUP2\_2\_A &MPT32\_2\_A)|( MPS\_B&MUP2\_2\_B&MPT32\_2\_B);

};

function FunctGroup3

{

return (MPS\_A&MUP2\_3\_A &MPT32\_3\_A)|( MPS\_B&MUP2\_3\_B&MPT32\_3\_B);

};

function FunctGroup4

{

return (MPS\_A&MUP2\_4\_A &MPC80\_4\_A)|( MPS\_B&MUP2\_4\_B&MPC80\_4\_B);

};

function FunctGroup5

{

return (MPS\_A&MUP2\_5\_A&MPC80\_5\_A)|( MPS\_B&MUP2\_5\_B&MPC80\_5\_B);

};

function FunctGroup6

{

return (MPS\_A&MUP2\_6\_A&MPN32\_6\_A)|( MPS\_B&MUP2\_6\_B&MPN32\_6\_B);

};

function FunctGroupMKPS2

{

return (MPS\_A&MKPS2\_A)|( MPS\_B&MKPS2\_B);

};

function FunctGroup7

{

return (MPS\_A&MUP2\_7\_A &MPT32\_7\_A)|( MPS\_B&MUP2\_7\_B&MPT32\_7\_B);

};

function FunctGroup8

{

return (MPS\_A&MUP2\_8\_A &MPT32\_8\_A)|( MPS\_B&MUP2\_8\_B&MPT32\_8\_B);

};

function FunctGroup9

{

return (MPS\_A&MUP2\_9\_A &MPA64\_9\_A)|( MPS\_B&MUP2\_9\_B&MPA64\_9\_B);

};

function FunctGroup10

{

return (MPS\_A&MUP2\_10\_A &MPC80\_10\_A)|( MPS\_B&MUP2\_10\_B&MPC80\_10\_B);

};

function FunctGroup11

{

return (MPS\_A&MUP2\_11\_A&MPC80\_11\_A)|( MPS\_B&MUP2\_11\_B&MPC80\_11\_B);

};