

на правах рукописи

Выборнов Пётр Владимирович

**ДИАГНОСТИКА ПАРАЛЛЕЛЬНЫХ МАГИСТРАЛЕЙ И ПАМЯТИ  
БОРТОВЫХ ЦВМ**

Специальность 05.13.15 –  
«Вычислительные машины, комплексы и компьютерные сети»

**АВТОРЕФЕРАТ**

диссертации на соискание ученой степени  
кандидата технических наук

Москва – 2012

Работа выполнена на кафедре «Вычислительные машины, системы и сети» Московского авиационного института (национального исследовательского университета).

Научный руководитель: д.т.н., проф. Гаврилин Б.Н.

Официальные оппоненты: Аксёнова Г.П., к.т.н., с.н.с. лаборатории технической диагностики ИПУ им. В.А. Трапезникова РАН

Александров А.Е., д.т.н., проф. кафедры «Персональные компьютеры и сети» МГУПИ

Ведущая организация: ФГУП «Научно-исследовательский институт авиационных систем» (НИИАС)

Защита состоится «\_\_» \_\_\_\_\_ 2012 г. в \_\_\_\_ часов \_\_\_\_ мин. на заседании диссертационного совета Д212.125.01 Московского авиационного института (национального исследовательского университета) по адресу: 125993, г.Москва, А-80, ГСП-3, Волоколамское ш., д.4, зал заседаний Ученого совета.

С диссертацией можно ознакомиться в библиотеке МАИ.

Автореферат разослан «\_\_» \_\_\_\_\_ 2012 г.

Ученый секретарь  
диссертационного совета Д212.125.01

к.т.н.

Корнеевкова А.В.

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

**Актуальность работы.** Начиная с середины 70-ых годов важной тенденцией в создании навигационно-пилотажных комплексов летательных аппаратов (НПК) было постоянно расширяющееся использование средств микропроцессорной техники. Ныне подавляющая часть функций преобразования информации в НПК, связанных с задачами навигации, процессами управления, стабилизации и наведения, реализуется с помощью бортовых цифровых вычислительных машин (БЦВМ). Информационные связи между удалёнными подсистемами комплекса и структурными блоками бортовых вычислительных систем строятся на основе цифровых последовательных каналов передачи данных.

Эффективная диагностика технического состояния вычислительных средств стала одной из важнейших составляющих обеспечения безотказности систем НПК и надёжности летательных аппаратов в целом. Несмотря на существенный прогресс, достигнутый в диагностике бортового оборудования, в том числе и его вычислительных устройств, в отношении узловой части самих цифровых модулей – вычислительного ядра – уровень автоматизации поиска неисправностей остаётся невысоким. Такое положение в данной области применения вычислительной техники объясняется, с одной стороны, исторически складывавшимися приоритетами исследований в цифровой диагностике, а с другой – особенностями применяемых электронных компонентов.

Проверки, которые производятся после изготовления вычислительных модулей БЦВМ и в ходе их испытаний, показали, что от 70 до 90% (в зависимости от типа модуля) всех неисправностей относятся к вычислительному ядру (ВЯ). Подавляющее большинство дефектов ВЯ, порядка 85-90%, приходится на связи микропроцессора (МП) с микросхемами внешней памяти – структуры демультимплексированных параллельных магистралей (ДПМ). Второе по численности место занимают дефекты кристаллов оперативной памяти, составляя в среднем чуть более 10%; небольшой остаток в 1-2% приходится на постоянную память.

Одной из основных проблем тестирования этих структурных компонентов ВЯ является отсутствие методов программной диагностики связей МП с памятью, обеспечивающих достаточную глубину разделения неисправных состояний и пригодных для поиска наиболее часто встречающихся неисправностей – дефектов монтажа и печати. Эта проблема является специфической для вычислительных устройств, созданных на основе не имеющих граничного сканирования микросхем (микропроцессоров, асинхронной памяти, магистральной логики). К подобным устройствам относятся, в частности, модули БЦВМ инерциальных систем управления (ИСУ) тактических ракет. Магистрали в таких устройствах с точки зрения процесса диагностирования составляют одну целостную подсистему с памятью и не могут рассматриваться как отдельный объект.

Традиционно используемые тесты, обнаруживая факт неработоспособности подсистемы, являются малоинформативными с точки зрения отделения и определения магистральных дефектов и требуют постановки дополнительных, как правило, трудоёмких программных и электроизмерительных экспериментов. В ряде случаев постановка таких экспериментов в условиях проявления дефекта оказывается невозможной.

Вопросы тестирования кристаллов памяти в целом разработаны достаточно глубоко и всесторонне, обеспечивая эффективное обнаружение т.н. классических дефектов. Однако существующие методы обнаружения кодочувствительных дефектов (PSF – pattern sensitive fault) не обеспечивают приемлемой для этапов наладки и испытаний полноты покрытия. Данный факт обусловлен критериями построения этих тестов, которые ориентированы на применение в аппаратуре самотестирования и должны удовлетворять довольно жёстким временным ограничениям, а также объективной сложностью обнаружения этих дефектов. Очевидно, что недостаточная степень выявления кодочувствительных дефектов на этапе испытаний существенно снижает вероятность безотказной работы БЦВМ в условиях эксплуатации.

Таким образом, существует необходимость создания специализированных магистральных тестов и усовершенствования методов проверки памяти в целях повышения эффективности диагностики: увеличения глубины разделения неисправных состояний, охвата максимального количества реально существующих типов дефектов, повышения достоверности результатов и снижения трудоёмкости процесса. Эта задача является актуальной как для этапа отработки опытных образцов БЦВМ, так и при их серийном производстве.

**Целями** диссертационной работы являются: разработка подходов, методов и алгоритмов программной диагностики магистральных структур и памяти вычислительных модулей, создание на их основе тестов для обнаружения и локализации дефектов, а также реализация программно-аппаратного комплекса диагностики вычислительных модулей БЦВМ.

**Основные результаты работы**, выносимые на защиту:

1) Созданы модели неисправностей ДПМ и предложен комплекс подходов и методов построения тестовых последовательностей, позволяющих реализовать максимальную для программного способа диагностики глубину разбиений неисправных состояний ДПМ. Адекватность моделей была проверена на реальных устройствах.

2) Разработаны алгоритмы детектирующей тестовой последовательности и локализующих проверок для ДПМ в системах, содержащих статическую оперативную (SRAM) и постоянную (ROM) память. Обнаруживающая и различающая способности алгоритмов были подтверждены экспериментальным путём. На их основе составлены тесты для обнаружения и поиска магистральных дефектов.

3) Разработан метод генерации шаблонов активации для нередуцированных моделей активных и пассивных кодочувствительных неисправностей PSF3 и PSF4, обеспечивающий их 100%-ное покрытие. Дополнение классических маршевых тестов памяти тестами, составленными по этому методу, позволило существенно увеличить полноту проверки кристаллов SRAM.

4) Спроектирована и изготовлена специализированная ЦВМ, являющаяся ядром комплекса диагностики. ЦВМ осуществляет тестирование удалённого вычислительного модуля, обеспечивая предписываемые алгоритмами форму и временные параметры тестовых воздействий.

5) Подготовлено программное обеспечение комплекса диагностики, которое включает операционную систему ЦВМ (ОС) и программу монитора. Разработан интерпретируемый язык CDS, позволивший выполнять программы диагностических тестов в режиме отладки, что обеспечивает наблюдаемость, управляемость и простоту организации процесса диагностирования, т.е. его эффективность.

**Методы исследования.** Для решения поставленных задач в ходе диссертационного исследования были использованы методы цифровой схемотехники и двоичной алгебры, технической диагностики, тестирования оперативных запоминающих устройств, технологии низкоуровневого и процедурно-ориентированного программирования.

**Научная новизна работы.** Новыми результатами работы являются алгоритм диагностики ДПМ с максимальной глубиной разбиения неисправных состояний и метод обнаружения кодочувствительных дефектов, обладающий 100%-ным покрытием для дефектов третьего и четвёртого порядков (PSF3 и PSF4). Более детально они представляются следующим образом:

1) при построении алгоритма диагностики ДПМ применён сигнально-ориентированный подход, подразумевающий нацеленность элементарных проверок на определение состояния магистрального сигнала или группы сигналов;

2) разработана функциональная классификация магистральных сигналов и связанных с ними неисправностей, и созданы явные модели поведения неисправной системы для всех видов магистральных дефектов;

3) введены логически неустойчивые типы поведения в модели неисправной системы для повышения достоверности диагноза;

4) предложен принцип разбиения множества одиночных неисправностей на ранжированные подмножества и маскирования дефектов подчинённых рангов, который определяет последовательность обнаружения групп дефектов при наличии их произвольных сочетаний;

5) разработаны методы комбинирования неисправных разрядов, на основе которых осуществляется локализация обрывов и замыканий на шинах адреса и данных;

6) предложены шаблоны тестового заполнения для ROM, как в варианте полного объёма, так и в свободных областях, позволяющие выполнять исчерпывающую диагностику связей с МП без необходимости перезаписи;

7) предложен принцип упорядочивания множества ячеек массива SRAM и их блоков в зависимости от чётности позиции в адресном пространстве, введены операции регулярного преобразования шаблонов и сформулирована и доказана теорема покрытия парных комбинаций;

8) на основе принципа чётности позиции, использования регулярных преобразований и теоремы разработан и обоснован метод прямой генерации активирующих шаблонов для нередацированных моделей PSF3 и PSF4.

**Практическая значимость** состоит в создании эффективных программных и программно-аппаратных средств тестовой диагностики параллельных магистралей и памяти, которые позволили автоматизировать поиск неисправностей в подсистеме магистраль–память, радикально снизив трудоёмкость наладки вычислительных модулей. На их основе созданы автоматизированные рабочие места для контроля и наладки опытных образцов БЦВМ в ГосНИИП и серийных образцов на заводе-изготовителе БЦВМ.

Алгоритмическая основа магистральных тестов допускает их применение для любой элементной базы, гарантирует нечувствительность к разбросу характеристик драйверов и приёмников сигналов и обеспечивает получение достоверных результатов при произвольных сочетаниях дефектов и наличии логической неустойчивости. Применение бит-ориентированных маршевых тестов, разработанных в работе тестов для PSF3/PSF4, а также тестов переключения адреса и насыщения позволило существенно увеличить полноту проверки кристаллов SRAM. Разработанный комплекс диагностики, наряду со стандартной наладкой, позволяет проводить тестирование памяти при испытаниях модулей в специальных условиях, во время которых проявляется значительная часть дефектов. Комплекс обеспечивает простоту организации, управляемость и наблюдаемость процесса диагностирования. Схемотехнические решения ЦВМ комплекса и её программное обеспечение являются перспективными для использования в задачах диагностики интерфейсных узлов БЦВМ, полунатурного моделирования и отладки рабочих программ, а также в качестве монитора мультиплексированной магистрали.

**Апробация работы.** Результаты работы представлялись на конференциях молодых учёных и специалистов ГосНИИП (2006 г., 2008 г.), конференциях Московского отделения МОО «Академия навигации и управления движением» (г. Москва, ЦНИИАГ, 2009 г., 2011 г.), конференции «Львовские чтения» (г. Свердловск, ОКБ «Новатор», 2010 г.).

**Публикации.** По теме диссертации опубликовано 2 научные работы в журналах, рекомендованных ВАК РФ; результаты работы также нашли отражение в пояснительных записках к двум техническим проектам (ГосНИИП).

**Структура и объём работы.** Диссертация состоит из введения, трёх глав, заключения, списка литературы и трёх приложений, оформленных в виде отдельной книги. Основной текст изложен на 185-и страницах и содержит 33 рисунка и 26 таблиц. Список источников включает 63 наименования.

## СОДЕРЖАНИЕ РАБОТЫ

**Введение** носит постановочный характер, формулируются цель и задачи работы, и обосновывается её актуальность, в том числе на статистических данных выявляемых дефектов. Составляется схема исследования и даётся краткая характеристика основных разделов диссертации.

**Первая глава** посвящена разработке методов и алгоритмов диагностики магистральных структур вычислительных модулей.

Описывается типовая схема вычислительного ядра модуля (рис.1).

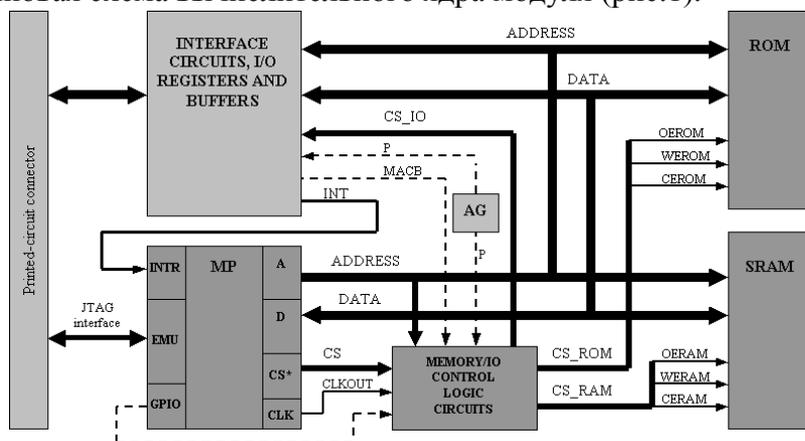


Рис.1. Обобщённая функциональная схема вычислительного модуля БЦВМ

Детально рассматриваются магистральные циклы обмена процессора с памятью, приводятся временные диаграммы сигналов ДПМ (рис.2).

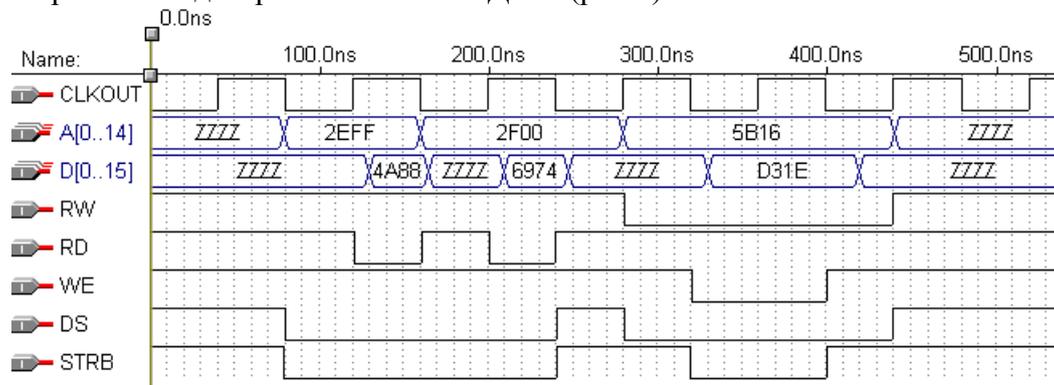


Рис.2. Временные диаграммы магистральных сигналов (read-read-write)

**Функциональная классификация сигналов.** Все сигналы диагностируемого модуля разбиваются на пять функциональных групп: 1) **CS** – линии питания микросхем подсистемы и образующие по форме временной диаграммы шесть подгрупп сигналы управления памятью, к которым относятся процессорные стробы, тактовая частота магистрали, управляющие адресные разряды, биты регистров I/O доступа к памяти MACB и производные перечисленных сигналов; 2) **A** – адресные разряды магистральной шины, участвующие в физической адресации ячеек памяти; 3) **D** – линии данных; 4) **F** – внешние сигналы постоянного уровня; 5) **P** – периодические не синхронизированные с CLKOUT внешние сигналы, производные непроцессорных генераторов, если они есть в устройстве (AG на рис.1.). Группы 1, 2 и 3 образуют магистральные сигналы (I), группы 4 и 5 - внешние (E).

Физическими дефектами исследуемого объекта являются: замыкания магистральных сигналов между собой (SH-I1-I2-...) или со сторонними сигналами (SH-I-E), обрывы сигнальных линий и отсутствие питания микросхем памяти и управляющей логики (BR-I), общие для подсистемы дефекты логических элементов MCLC и сигнальных трактов (SPF-CS), дефекты переключения сигналов (SWF-I), вызванные высоким сопротивлением

сигнальных цепей, и дефекты кристаллов памяти (MCD). Все возможные одиночные неисправности подсистемы подразделены на 20 классов, выделяемых соответственно функции участвующих в ней сигналов.

Класс образуется одним или несколькими видами неисправностей. Вид есть типовое искажение формы и уровней определённых сигналов, в ряде случаев сопровождающееся и нарушением работы электронных схем, для которых эти сигналы являются входными. Например, виды SH-WE-A[x], SH-RD-A[x], SH-CE-A[x], SH-DS-A[x] и SH-CLKOUT-A[x] составляют класс SH-CS-A. Искажение уровней вызывается либо дефектами сигнальных линий, либо дефектами элементов магистральной логики. Созданная функциональная классификация в основном совпадает с разбиением видов дефектов на ранжированные группы и характеризует влияние дефектов на обмены с памятью с точки зрения процессора и, следовательно, способ их обнаружения.

**Логически неустойчивое поведение.** В работах по диагностике межкомпонентных связей в качестве описания одиночных дефектов используются детерминированные логические модели замыканий для двух сигналов. Известно несколько таких моделей, а в качестве рабочих, как правило, применяются монтажное И (WAND) или монтажное ИЛИ (WOR); обрывы считаются константой лог.1 либо лог.0. Не ставя под сомнение следующего из этих моделей общего принципа обнаружения замыканий посредством подачи парафазных наборов, в диссертационной работе было показано, что совокупность этих моделей не является достаточной для отражения поведения замкнутых сигналов.

Во-первых, на практике не являются исключением кратные замыкания, вовлекающие более чем две сигнальные линии, логический тип которых, как правило, не совпадает с типом простого. Во-вторых, логический тип простых и кратных замыканий во многих случаях не может быть описан детерминированной функцией (рис.3). Применение вышеописанных моделей к потерявшей детерминизм неисправной системе может привести к недостоверному диагнозу – дефект может быть пропущен или определён неправильно. Поэтому необходимы такие модели, которые позволили бы описать и кратные замыкания, и возникающую логическую неопределённость. Для этого замыкания сигнальных линий были представлены в электротехнических понятиях.

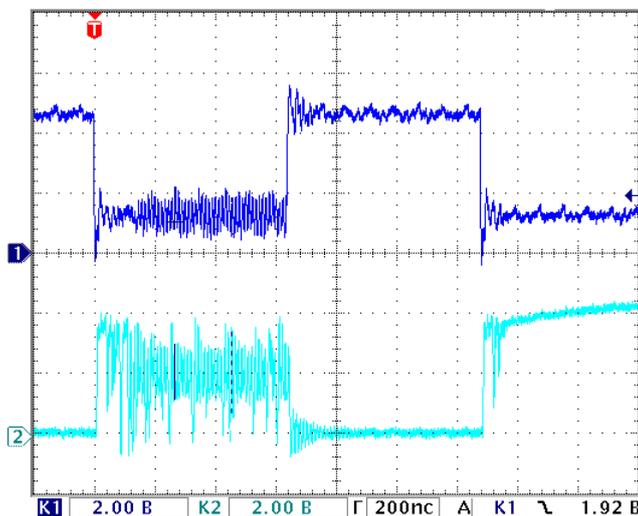


Рис.3. Замыкание адресных линий с неопределённым уровнем (канал K1 – тёмный), приводящее к попеременной коммутации разных ячеек SRAM на разряд D[n] шины данных (канал K2 – светлый)

Техника проверок объектов с детерминированным поведением заключается в установлении правильности выполнения единичной операции. На основе рассмотрения и анализа реальных замыканий был определён принцип построения проверок для объектов с логической неустойчивостью: критерием исправности сигнала должно быть отсутствие неправильных результатов при многократном повторении операции с инверсными входными

наборами. Элементарные проверки для таких объектов должны строиться исходя из предположения, что неисправный объект может вести себя и в соответствии с любой детерминированной моделью, и переходить к логически неустойчивому функционированию в произвольный момент времени. Для реализации данного подхода в работе были предложены и использованы зависимые от логического контекста модели замыканий:

**S0** (suppressing 0) – на сигнальных линиях устанавливается потенциал, всегда воспринимающийся приёмником при данной логической конфигурации сигналов как устойчивый лог.0 (может быть и выше стандартного порогового уровня  $U_{вх0max}$ ).

**S1** (suppressing 1) – устанавливается потенциал  $U_{зам}$ , всегда воспринимаемый приёмником как устойчивая лог.1 (может быть и ниже стандартного  $U_{вх1min}$ ).

**INDTL** (indetermined level) – результирующий потенциал приёмником воспринимается неустойчиво, то как лог.1, то как лог.0 в течение одного цикла или в разных циклах.

Аналогичные логические типы были введены и для описания обрывов.

**Модели неисправностей.** Пусть  $E$  – множество технических состояний подсистемы «магистраль-память», которое включает исправное состояние  $E_0$  и подмножество из всех остальных, неисправных состояний  $E_i$  ( $i=1,2,\dots,|C|$ ), где  $C$  – множество всех одиночных неисправностей (простых дефектов)  $S$  и их возможных сочетаний. Тогда  $Z=\Psi(X,Y_{n-1},n)$  – некоторая форма представления передаточной функции исправного дискретного последовательностного объекта диагностики ( $E_0$ ), где  $X$  – набор входных переменных в цикле  $n$ ,  $Y_{n-1}$  – значение внутренних переменных после записи в элементы памяти в цикле  $n-1$ ,  $Z$  – выходные функции в цикле  $n$  после определённого времени установки выходов. Объект, находящийся в  $i$ -неисправном состоянии ( $E_i$ ), реализует передаточную функцию  $Z_i=\Psi_i(X,Y_{n-1},n)$ . Функция  $Z_i$  с входным набором  $X_t$  и значением внутренних переменных  $Y_{n-1}$  описывает элементарную проверку  $\pi_i$ , а используемые выходные переменные, или результат проверки, принято обозначать  $R_{i\bar{t}}$ .

При проверке магистрали подсистема в большинстве случаев может быть представлена как квазикомбинационное устройство. Для этого передаточная функция (модель объекта) должна представлять не единичное обращение, а пару совмещённых операций записи и чтения. Наличие инициализирующих записей выражает  $Y_{n-1}$  (содержимое ячеек SRAM) через входные переменные  $X=(A,Dp)$ , и каждая элементарная проверка описывается функцией без участия членов  $Y_{n-1}$  и  $n$ .

Очевидно, что передаточной функцией исправной подсистемы будут тождества:

$Dr[A](n)=Dp[A](i=\max(0, n-1))$  при обращении к SRAM,

$Dr[A](n)=Dm[A]$  при обращении к ROM,

где  $Dr[A](n)$  – данные, читаемые процессором в  $n$ -цикле из адреса  $A$ ;  $Dp[A](i=\max(0, n-1))$  – данные, выставляемые процессором на магистраль в  $i$ -цикле записи в память по адресу  $A$ ,  $i$ -цикл является последней записью по адресу  $A$ ;  $Dm[A]$  – данные в ячейке ROM с адресом  $A$ .

При наличии неисправности связей  $S_i$  подсистема функционирует в соответствии с одной из передаточных функций  $\Psi_i$ . Соответствующие ей функции элементарных проверок  $Dr=\Phi_i(A,Dp)$  с необходимыми значениями  $A$  и  $Dp$  (об их выборе см. далее) составляются в виде синтаксических диаграмм записи-чтения. Для каждого вида магистральных дефектов составляется несколько таких диаграмм, для каждого логического типа замыкания и набора  $(A,Dp)$ . Синтаксические диаграммы являются текстовым представлением поведения объекта, которое получается, если вместо временных диаграмм исправных сигналов (рис. 2) подставить диаграмму при их замыкании, и проследить работу памяти в этих условиях; постановка физических экспериментов позволила проверить, уточнить и в ряде случаев дополнить модели дефектов. Выглядит этот способ представления следующим образом (для примера взят вид замыкания SH-WE-A[x], логического типа S1, из класса SH-A-CS):

**A(0000)Dp[0000] WR + Aw[x]=0 mem(0000)←[0000] RD + Ar[x]=1 mem(0100)→ [PPPP] RES=CNE**, где

**A(0000)Dp[0000]** – входные наборы адреса A(в круглых скобках) и данных Dp[в квадратных скобках], условные позиции неисправных разрядов A и/или D выделены подчёркиванием

**WR(RD)** – маркеры начала описания записи(чтения), атрибуты { + | – | **wda** | **wdd** | **osc** | **asw** | **atd** } – обращение происходит нормально | стробирование подавлено | запись по фронту установки данных | запись по фронту снятия данных | отключающееся чтение | самопереключение адреса | изменение логического состояния разряда адреса при выставлении данных памятью

**Aw[x]=0** – фактическое значение неисправных разрядов в цикле записи, символ “/”, когда присутствует, разделяет альтернативные варианты

**mem(0000)←[0000]** – изменение содержимого ячеек памяти в цикле записи

**Ar[x]=1** – фактическое значение неисправных разрядов в цикле чтения в момент фиксации процессором состояния магистральной шины данных

**mem(0100)→[PPPP]** – ячейки, из которых фактически производится чтение, и данные получаемые процессором D<sub>r</sub> [в квадратных скобках], являющиеся результатом R<sub>ci</sub> элементарной проверки. Значением данных, кроме определённого числа, может быть: **XXXX** – неопределённый результат чтения; **PPPP** (previous value) – предыдущее значение в ячейке, не изменяемое данной элементарной проверкой

**RES=CNE** – атрибуты выходных данных, могут быть следующими: {EQ|NE|CNE|LP} – D<sub>r</sub>=D<sub>p</sub> | D<sub>r</sub>≠D<sub>p</sub> безусловно | D<sub>r</sub>≠D<sub>p</sub> условно | значение низкой вероятности.

Поведенческие модели неисправной подсистемы определяют элементарные проверки, по которым дефекты могут быть обнаружены и отделены друг от друга.

**Методика разбиений.** Если множество допустимых одиночных неисправностей объекта S, а множество их сочетаний – C (S⊆C), то фактическая j-неисправность подсистемы в общем случае представляется C<sub>j</sub>=S<sub>k</sub>+S<sub>l</sub>+...+S<sub>p</sub>, причём на дефекты S<sub>k</sub>...S<sub>p</sub> не налагается ограничений по числу и видовой принадлежности. Особенностью систем с неоднородными сигналами является то, что детектирующие проверки для некоторого подмножества неисправностей могут быть осуществлены только при отсутствии некоторых других их подмножеств. Для выбора последовательности обнаружения неисправностей при наличии произвольных их сочетаний в работе предложен принцип разбиения на ранжированные подмножества и маскирования дефектов подчинённых рангов.

Принцип состоит в следующем: всё множество S магистральных дефектов группируется в непересекающиеся ранжированные подмножества F<sub>t</sub> (t – ранг подмножества) таким образом, чтобы любой одиночный дефект подмножества текущего ранга можно было обнаружить при отсутствии дефектов старших рангов и независимо от наличия или отсутствия любых дефектов из подмножеств подчинённых рангов.

Самым старшим рангом является первый, для него все остальные подмножества, имеющие ранг t>1, являются подчинёнными. Критерием включения простого дефекта S<sub>i</sub> в подмножество старшего ранга F<sub>t</sub> из исходного F', ранг которого принимается равным t+1, является возможность физически осуществить и однозначно интерпретировать такую элементарную проверку, которая при наличии любых дефектов ранга t+1 даст результат R<sub>ci</sub> (ответ исправного объекта) при отсутствии S<sub>i</sub> и R<sub>ci</sub>≠R<sub>ci</sub> – при его наличии. Сначала берется исходное множество F'=S, t=1 и выделяется F<sub>1</sub>, затем t=2, F'=S-F<sub>1</sub>, и уже из нового F' выделяется группа дефектов F<sub>2</sub> и так далее вплоть до последнего класса или вида дефектов. В результате применения этой процедуры исходное множество было разделено на 13 подмножеств F<sub>t</sub> (t=1,2,...13), так, что S=F<sub>1</sub>∪F<sub>2</sub>∪...∪F<sub>13</sub>. Каждому подмножеству F<sub>t</sub> соответствует свой этап теста, предназначенный для обнаружения и определения дефектов только из этого подмножества, называемых *целевыми дефектами*. Этапы теста выполняются поочередно в ранговом порядке соответствующих подмножеств, начиная с самого старшего и кончая младшим.

**Синтез проверок алгоритма диагноза.** На основе определённой выше последовательности и предложенных моделей и принципов синтезируются проверки

алгоритма магистрального теста для систем, содержащих SRAM и ROM. Тест состоит из основной (детектирующей) последовательности и локализирующих процедур. Основная тестовая последовательность состоит из детектирующих проверок для каждого подмножества  $F_t$ , это путь по графу алгоритма диагноза от состояния «полностью неопределённо» до «исправно». Локализирующие процедуры выполняют окончательное разделение и определение неисправностей внутри целевого подмножества, обнаруженного проверками основной последовательности, и являются составными частями соответствующих этапов теста. Алгоритмы проверок представлены в виде Фортран-подобного псевдокода.

Выбор входных векторов  $A$  и  $D_p$  как для детектирования  $F_t$ , так и для определения всех простых дефектов  $S_i \in F_t$  является основной задачей, решаемой при построении каждого этапа, или фазы теста. Рассмотрение каждого из 13-ти этапов теста начинается с анализа моделей целевых дефектов, на основе которого определяются алгоритм и тестовые вектора детектирующей проверки. В соответствии с принципом ранжированных разбиений выбираются такие вектора, которые при наличии любых неисправностей  $F'$  из возможных на данной фазе теста позволят выделить дефекты, относящиеся к подмножеству  $F_t$ , или сделать заключение об отсутствии таких дефектов.

Детектирующие проверки устанавливают лишь факт неработоспособности тех или иных групп сигналов или разрядов, в практическом же смысле всегда желательно, чтобы тест производил окончательное разбиение неисправных состояний, что свело бы к минимуму дополнительные операции по определению причин неисправностей или совсем бы их исключило. В работе, в частности, был предложен метод для локализации групп замыканий и обрывов на шинах адреса и данных, получивший название *метода комбинирования*.

Поскольку замыкание любого вида проявляется только тогда, когда на участвующие в нём разряды (сигналы) подаются разноимённые логические уровни, то отсутствия его проявления можно добиться, подавая на эти разряды одинаковые логические значения. Следовательно, принадлежность одного или нескольких разрядов к группе замыкания может быть установлена по факту пропадания дефекта при условии, что на все или часть этих разрядов подано одно и то же логическое значение, отличающееся от значения разрядов, не входящих в группу. Для проведения такой проверки должны быть составлены все возможные комбинации разрядов в пределах подозреваемого множества. Отличием локализирующей проверки от детектирующей является то, что тестовые наборы первой содержат не единственную 1(0) на фоне остальных нулевых (единичных) разрядов, а их пары, тройки и т.д. Комбинации должны составляться по нарастанию порядка: сначала проверяются все возможные пары разрядов, затем тройки и т.д. Совокупность комбинаций, при которых проведение проверки не обнаруживает дефекта, и будет содержать искомые разряды.

На основе метода комбинирования было составлено три разных алгоритма локализации для следующих групп дефектов: 1) SH-A-D, 2) SH-A-A, BR-A, SH-A-F, 3) SH-D-D.

Задача этапов проверки связей с постоянной памятью заключается главным образом в установлении целостности сигнальных линий, поскольку большинство замыканий элиминируется предыдущими этапами теста. Была разработана схема области тестового заполнения ROM. Первые две четверти области заполняются инверсными шаблонами и предназначены для проверок сигналов  $D$  и  $CS$ . Верхняя её половина содержит адресный код для обнаружения неисправностей линий  $A$ .

Испытания разработанного алгоритма полностью подтвердили его детектирующую и различающую способность, в том числе при наличии сложных комбинаций дефектов и логической неустойчивости. Неисправности, связанные с замыканиями и обрывами на шинах данных и адреса и составляющие порядка 80-85% встречающихся магистральных дефектов, определяются алгоритмом вплоть до разрядов и групп замыканий, не требуя никаких дополнительных действий для своей локализации. Таким образом, фактически достигнута максимальная для программного способа глубина диагностики магистральных структур.

**Вторая глава** посвящена анализу существующих методов тестирования SRAM и разработке нового метода обнаружения кодочувствительных неисправностей. Тестирование кристаллов памяти является второй частью процесса диагностирования подсистемы «магистраль-память».

Рассматриваются электрическая схема SRAM и модели классических функциональных дефектов, статических – залипания SAF, обрыва линии SOF, перехода TF, взаимовлияния CF, дешифратора адреса ADF и динамических – удержания данных DRF, восстановления усилителей записи/чтения и переключения адреса. Описание сопровождается указанием возможных физических причин дефекта, а также операций, необходимых для его детектирования.

Наиболее эффективным средством тестирования памяти являются маршевые тесты, сочетающие высокую покрывающую способность в отношении дефектов классических типов и низкую сложность порядка  $N$  ( $N$  – количество ячеек массива). Маршевый тест состоит из последовательности маршевых элементов. Маршевый элемент есть некоторая последовательность операций чтения и/или записи, применяемая к каждой ячейке массива перед тем, как будет осуществлён переход к следующей ячейке. Перебор ячеек может производиться в одном из двух порядков следования адресов: прямом или обратном. Прямая последовательность перебора адресов обозначается значком  $\uparrow$ , обратный –  $\downarrow$ . Чаще всего применяются наиболее простые в реализации счётчиковые последовательности, прямая (от 0 до  $N-1$ ) и реверсивная (от  $N-1$  до 0). Когда порядок перебора ячеек значения не имеет, перед маршевым элементом ставится значок  $\downarrow$ . Операции чтения ожидаемого значения  $e$  принято обозначать  $re$ , записи –  $we$ ,  $e \in \{0,1\}$ .

Для обнаружения тех или иных дефектов, в том числе и в случаях, когда проявление одного дефекта может маскироваться наличием другого, маршевый тест должен содержать элементы с определённой последовательностью операций. Эти вопросы изложены по основополагающим в данной теме работам. Так, например, тест обнаруживает все ADF при наличии двух маршевых элементов:  $\uparrow(re, \dots, w\bar{e})$  и  $\downarrow(r\bar{e}, \dots, we)$ . Первый элемент читает значение  $e$  из ячейки 0, затем записывает инверсное значение  $\bar{e}$  в ячейку 0, читает значение  $e$  из ячейки 1, затем записывает  $\bar{e}$  в ячейку 1, ... , читает  $e$  из ячейки  $N-1$ , записывает  $\bar{e}$  в ячейку  $N-1$ . Второй элемент выполняет аналогичные операции с инверсными значениями, но в обратном порядке.

Были проанализированы наиболее известные маршевые тесты и показано, что среди них существуют такие, которые обеспечивают полное покрытие для всех типов статических дефектов, включая и случаи связанных. В качестве примера можно привести тест March B длиной  $17N$ :

**March B:**  $\{\downarrow(w0); \uparrow(r0, w1, r1, w0, r0, w1); \uparrow(r1, w0, w1); \downarrow(r1, w0, w1, w0); \downarrow(r0, w1, w0)\}$ .  
M0 M1 M2 M3 M4

Обнаружение динамических дефектов SRAM и ROM затруднений не представляет и производится либо путём сжатия читаемых данных и последующего сравнения с эталонным значением, либо прямым сравнением получаемого массива. Главным условием проведения подобных проверок, как и в магистральных тестах, является чтение (запись) в прилежащих циклах.

Наиболее труднообнаруживаемым типом дефектов SRAM являются кодочувствительные неисправности (*pattern sensitive faults* – PSF), вовлекающие более двух ячеек массива (т.н. неклассические дефекты). Для подобных дефектов поведение одной ячейки зависит от содержимого или логических переходов соседних ячеек. Выделяют базовую ячейку (*base cell*), которая является аналогом жертвы в классических дефектах взаимовлияния CF, и соседние ячейки (*neighborhood cells*), выступающие в роли агрессоров. Если принимается, что их местоположение может быть произвольными, то говорят о нередуцированной модели PSF. Если допускается, что массив представляет собой одну прямоугольную матрицу, последовательность расположения ячеек в которой соответствует их адресам, и в дефекте могут участвовать только физически смежные ячейки, то говорят о редуцированной модели.

Нередуцированные модели являются универсальными, редуцированные корректно могут описать только ограниченное число случаев с известной топологией кристалла.

В зависимости от эффекта влияния на базовую ячейку большинство исследователей выделяет два класса кодочувствительных дефектов: пассивные (*passive PSF* – PPSF), когда состояние базовой ячейки не может быть изменено для определённого кода в соседних ячейках, и активные (*active PSF* – APSF), когда базовая ячейка изменяет своё состояние при наличии определённого кода в соседних ячейках и изменении состояния на противоположное в одной из них. В качестве объекта исследования обычно рассматриваются дефекты PPSF<sub>k</sub>, где *k* обозначает количество произвольных ячеек массива емкостью *N* бит, участвующих в неисправности, причём одна из них является базовой. Основной проблемой обнаружения кодочувствительных дефектов является генерация наборов активации дефекта для всех возможных конфигураций и значений соседних ячеек, и предельное покрытие любого метода определяется отношением количества наборов активации дефекта, генерируемых тестом, к общему числу возможных наборов. В работе были рассмотрены четыре существующих метода обнаружения этих дефектов, три из них используют нередуцированные модели, а один – редуцированные. Все они используют многократные проходы маршевых тестов.

Первый метод основан на изменении *начального адреса* при каждом проходе теста, сам порядок перебора адресов остаётся неизменным. Покрывающая способность метода зависит от выбора начального адреса повторного применения теста. Адрес должен выбираться с таким расчётом, чтобы для любой базовой ячейки код во всевозможных влияющих ячейках максимально отличался от того кода, который был при предыдущем применении теста. Максимальное отличие будет достигаться в том случае, когда во всевозможных *k*–1 влияющих ячейках (каковыми могут быть любые *k*–1 из *N*–1) будут сформированы двоичные комбинации с максимальным Хэмминговым расстоянием. Это условие выполняется для двухпроходных маршевых тестов, когда разница между начальными адресами первого и второго применения теста равна одной второй от всего размера памяти. В общем случае при многократном использовании маршевых тестов разница между начальными адресами проходов теста должна быть равна 1/*q* от всего объёма памяти, где *q* – количество проходов маршевых тестов.

Однократный проход теста March C– даёт следующие покрытия для PPSF<sub>k</sub>: для *k*=3 – 50,0%, для *k*=4 – 25,0% и для *k*=5 – 12,5%. Для *k*=5 двукратный проход даёт 24,5%, трёхкратный – 32,9%, *N*-кратный – 50,0%. Существует методический предел, достигаемый при *N*-кратном использовании теста: нельзя достигнуть 100%-ого покрытия для кодочувствительных дефектов с *k*>3. Следует отметить, что это предельное покрытие на практике не может быть реализовано, так как длина теста приобретает порядок *N*<sup>2</sup>.

Альтернативным подходом является многократное тестирование с *различными адресными последовательностями*. Под адресной последовательностью понимается упорядоченная последовательность максимальной длины из *m*-разрядных двоичных векторов  $A(j)=b_{m-1}b_{m-2}b_{m-3}\dots b_2b_1b_0$ , где  $b_i \in \{0,1\}$ ,  $i \in \{0,1,2,\dots,m-1\}$ , а  $j \in \{0,1,2,\dots,2^{m-1}\}$ , однократно принимающих всевозможные значения  $\{0,1,2,\dots,2^{m-1}\}$ . В качестве меры отличия последовательностей используется метрика среднего Хэммингова расстояния АНД между соседними элементами. Так, для случая счётчиковой последовательности АНД[*A<sub>C</sub>*]=2, а для последовательности кода Грея АНД[*A<sub>G</sub>*]=1, что свидетельствуют об их незначительном отличии и невысокой полноте покрытия кодочувствительных неисправностей при реализации двукратного тестирования памяти. С целью достижения максимального отличия адресных последовательностей рассматриваются последовательности, для которых данная характеристика близка к величине разрядности *m* адресов памяти – анти-Грея (*A<sub>G\*</sub>*) и МНД (*A<sub>H</sub>*).

По результатам моделирования двукратного применения теста March C– наибольшее покрытие было получено для сочетания *A<sub>C</sub>*–*A<sub>G\*</sub>*: для PPSF<sub>3</sub> – 71,7%, для PPSF<sub>5</sub> – 22,0%. Отметим, что предыдущий метод при двукратном тестировании обеспечивает несколько более высокое покрытие. Насколько можно судить по вышедшим в печать работам,

дальнейшая разработка метода использования различных адресных последовательностей (трёхкратного и т.д.) развития не получила, что, вероятно, связано с трудностью использования метрики АНД для получения большего количества последовательностей.

Метод *начальных заполнений* массива использует повторные проходы неразрушающих версий маршевых тестов для различных начальных значений массива. Потенциальным достоинством неразрушающих тестов является возможность использования любого начального заполнения памяти для генерации наборов активации PSF, в то время как разрушающие версии формируют однотипные заполнения. Эффективность данного подхода зависит от степени изменения содержимого памяти перед повторным применением теста.

В случае  $m$  проходов теста, который генерирует только одно состояние в соседних ячейках, оптимальным набором заполнений  $V_0, V_1, \dots, V_{m-1}$  будет такое, в котором минимальное значение из Хэмминговых расстояний между каждой парой векторов  $HD(V_i, V_j)$ , где  $i, j \in \{0, 1, 2, \dots, m-1\}$ , будет наибольшим. Эффективные алгоритмы существуют для генерирования наборов, состоящих не более чем из четырёх таких шаблонов, чем и ограничивается покрывающая способность данного метода. По результатам моделирования для тестов типа MATS+ покрытие дефектов PPSF3 составило 47,23%, PPSF4 – 24,54%.

В методах для *редуцированных моделей* используется прямая генерация заполнений, поэтому подобные тесты отличаются высоким покрытием при небольшой длине. В работе был рассмотрен маршевый тест MT-R3CF длиной  $30n$  для обнаружения активных APSF3 и пассивных PPSF3 кодочувствительных дефектов с покрытием 100%. Самый главный недостаток подобных методов является обратной стороной их преимущества: в большинстве практических случаев гарантировать достоверность редуцированных моделей, а значит и покрывающую способность таких тестов не представляется возможным.

Таким образом, все методы для нередуцированных моделей используют опосредованные оценки различия состояний массива, которые позволяют максимизировать покрытие при заданном числе проходов теста, но не позволяют достигнуть полного покрытия даже для наиболее распространённых дефектов наименьшего порядка.

В работе был развит новый метод обнаружения PSF3 и PSF4 для нередуцированных моделей – *метод чётности позиции*, – обеспечивающий 100%-ное покрытие для этих неисправностей. Он использует прямую генерацию активирующих наборов посредством начальных заполнений массива и многократные проходы неразрушающего маршевого теста  $\{\uparrow(re, w\bar{e}, r\bar{e}, we, re); \uparrow(re)\}$ . Ниже излагается предложенный принцип чётности позиции, вводится ряд понятий, формулируется теорема и показывается основанный на ней способ генерации заполнений.

Под *чётностью позиции* или *расположения* ячейки одноразрядной памяти понимается чётность адреса этой ячейки. Любой чётный адрес одноразрядной памяти размером  $N$  представляется выражением  $A=2n$ , а нечётный –  $A=2n+1$ , где  $n \in (0, 1, \dots, N/2)$ . Чётность позиции вводится лишь для того, чтобы разделить и упорядочить обращения к ячейкам двух подмножеств, никакого смысла касательно геометрического расположения ячеек на кристалле она не несёт.

Подобно ячейке одноразрядной памяти, чётность расположения слова в многоразрядной памяти будем определять чётностью его адреса, а чётность позиции ячейки внутри слова многоразрядной памяти – чётностью номера бита (биты 0,2,4,6,... являются чётными, а 1,3,5,7,... – нечётными). Характеристика чётности может быть приложена к блокам ячеек или слов любого размера, при этом вместо чётности физических адресов необходимо оперировать чётностью порядковых номеров блоков. Ячейки одноразрядной памяти и слова многоразрядной памяти, адреса которых различаются на 1, будем называться смежными.

Шаблон называется минимальная по размеру последовательность бит, повторение которой даёт требуемое регулярное заполнение ячеек массива. В общем случае шаблон обозначается  $(P, Q, R, \dots)$ , где  $P, Q, R, \dots$  –  $n$ -разрядные числа, размещаемые в памяти непосредственно друг за другом в порядке возрастания адресов. Подразделение последовательности бит шаблона на  $n$ -разрядные числа обусловлено в основном

разрядностью комбинаций, для генерации которых создаётся шаблон, и выбранным размером слова, внутри которого эти комбинации должны быть созданы.

*Определение 1.* Элементарной инверсной парой (1,0):(0,1) называется пара двухбитных шаблонов (1,0) и (0,1), применяемых к двум смежным ячейкам одноразрядной памяти. Напомним, что такие ячейки всегда имеют различную чётность расположения.

*Определение 2.* Инверсной парой масштаба  $x$ , обозначаемой  $M[x](1,0):(0,1)$ , называются два шаблона (P,Q) и (Q,P), состоящие из полностью единичного  $P=11\dots1$  и полностью нулевого  $Q=00\dots0$  чисел, разрядность которых одинакова и равна  $x$ . Инверсная пара масштаба  $x=1$  есть элементарная инверсная пара. Инверсная пара масштаба  $x=2$  есть шаблоны (11,00):(00,11), а парой масштаба  $x=3$  являются значения (111,000):(000,111). Будем заполнение шаблоном (P,Q) называть прямым, а (Q,P) – инверсным.

Условимся считать, что в элементарных инверсных парах младшая ячейка всегда расположена по чётному адресу, а в масштабированных парах младший блок всегда имеет чётный порядковый номер.

*Теорема покрытия парных комбинаций.* Дан массив одноразрядной памяти размером  $N$ , где  $N=2^p$ ,  $p \in (1,2,3,\dots)$ . Адреса ячеек массива пробегают все значения от 0 до  $N-1$ . Заполнения массива инверсными парами  $M[x](0,1):(1,0)$ , где  $x=2^y$ , а  $y$  пробегает все значения от 0 до  $p-1$ , создают в двух любых произвольно взятых ячейках массива  $C_\alpha$  и  $C_\beta$  комбинации (0,1) и (1,0).

Пусть массив одноразрядной памяти содержит  $N=32$  ячейки (рис.4). Тогда  $p=\log_2 32=5$ , и  $y$  принимает значения 0,1,2,3,4, а соответствующие им значения  $x$  – 1,2,4,8,16. Тогда 5 инверсных пар заполнений массива в байтовом представлении будут выглядеть так:

- 1) 55h 55h 55h 55h : AAh AAh AAh AAh ( $y=0, x=1$ )
- 2) 33h 33h 33h 33h : CCh CCh CCh CCh ( $y=1, x=2$ )
- 3) 0Fh 0Fh 0Fh 0Fh : F0h F0h F0h F0h ( $y=2, x=4$ )
- 4) FFh 00h FFh 00h : 00h FFh 00h FFh ( $y=3, x=8$ )
- 5) FFh FFh 00h 00h : 00h 00h FFh FFh ( $y=4, x=16$ )

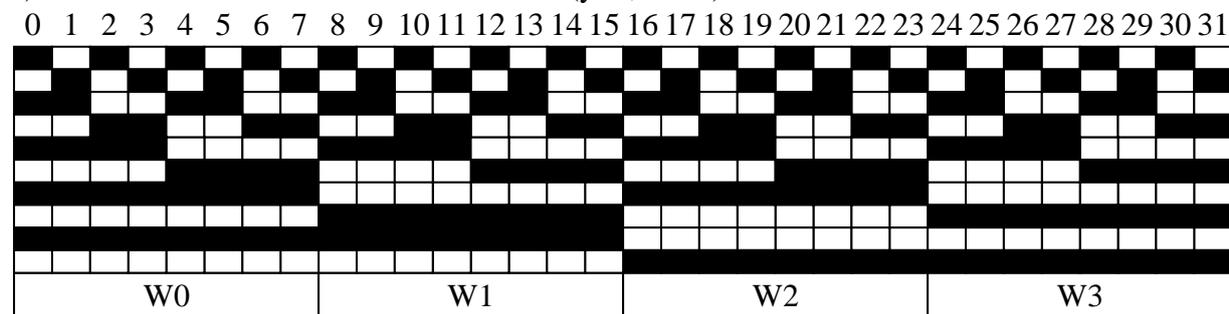


Рис.4. Заполнения массива  $N=32$  инверсными парами масштабов 1, 2, 4, 8, 16

Адреса ячеек массива проставлены в верхней части рисунка. Каждый ряд прямоугольников представляет собой заполнение массива определённым шаблоном; ячейки, принимающие единичное значение, окрашены чёрным. В паре рядов с одинаковым масштабом верхний ряд показывает прямое заполнение, а нижний – инверсное. Масштаб  $x$  увеличивается сверху вниз. В самом низу показано разделение массива на 4 блока (байта)  $W_0$ - $W_3$  по 8 ячеек.

Теорема непосредственно даёт метод генерации наборов активации дефектов PPSF3 и APSF3, вовлекающих три ячейки  $C_\alpha$ ,  $C_\beta$  и  $C_\gamma$  с произвольным расположением в адресном пространстве, из которых любая может быть базовой. Для определённости будем считать базовой, например, ячейку  $C_\gamma$ , а  $C_\beta$  – агрессором (для APSF3). Две соседние ячейки  $C_\alpha$  и  $C_\beta$  в ходе заполнений массива инверсными парами примут комбинации (1,0) и (0,1), а комбинации (0,0) и (1,1) будут получены при сплошных нулевых и единичных заполнениях. Проверка переключения каждой ячейки  $C_\gamma$  после заполнения массива обнаруживает PPSF3. Первое чтение  $C_\gamma$ , которая в паре с  $C_\alpha$  должна содержать требуемый набор активации, после запускающего перехода в  $C_\beta$  обнаруживает APSF3. Для обнаружения таких конфигураций

APSF3, в которых адрес базовой ячейки меньше адреса агрессора, тестом должна быть выполнена проверка сохранности шаблона после прохода элемента  $\uparrow(re, w\bar{e}, r\bar{e}, we, re)$ , выполняющего запускаящие переходы: для этого добавлен элемент  $\uparrow(re)$ . С учётом сплошных количество заполнений составит  $2\log_2 N + 2$  (множитель 2 перед логарифмом отражает то, что заполнение для каждого масштаба  $x$  производится двумя шаблонами –  $(P, Q)$  и  $(Q, P)$ ). Каждое заполнение требует  $N/W$  записей в память, где  $W$  есть разрядность памяти. Так как после заполнения к каждой из  $N$  ячеек маршевыми элементами  $\{\uparrow(re, w\bar{e}, r\bar{e}, we, re); \uparrow(re)\}$  будет произведено 6 обращений, то сложность теста будет вычисляться как  $6N(2\log_2 N + 2) + N/W(2\log_2 N + 2) = 2N(6 + 1/W)(\log_2 N + 1)$ .

*Определение 3.* Перестановочной парой шаблонов называются двухсловные шаблоны  $(P, Q)$  и  $(Q, P)$ , в которых числа  $P$  и  $Q$  меняются местами; оба числа записываются в смежные слова или блоки ячеек (слов). Условимся, что младшее число всегда пишется в блок с чётным порядковым номером, старшее – в блок с нечётным номером. На рис. 5 заполнение ряда 1 (сверху) выполнено шаблоном  $(05h, AAh)$ , а заполнение ряда 4 – шаблоном  $(AAh, 05h)$ ; эти шаблоны образуют перестановочную пару. Отметим, что инверсная пара масштаба  $x$  есть частный случай перестановочной пары из  $x$ -разрядных чисел вида  $P=00\dots 0$  и  $Q=11\dots 1$

*Определение 4.* Масштабированной перестановочной парой шаблонов масштаба  $x$ , обозначаемой  $M[x](P, Q):(P, Q)$ , называются два шаблона  $(xP, xQ)$  и  $(xQ, xP)$ , в которых  $xP$  и  $xQ$  получаются соответственно из чисел  $P$  и  $Q$  расширением каждого их бита начиная с младшего на  $x$  разрядов. На рис. 5 масштабированный шаблон  $M[2](05h, AAh)=(0033h, CCCCCh)$  находится в ряду 3, а  $M[2](AAh, 05h)=(CCCCCh, 0033h)$  – в ряду 6.

*Определение 5.* Повторной (реплицированной) парой перестановки  $R[x](P, Q):(Q, P)$  для исходной пары  $(P, Q):(Q, P)$  является пара шаблонов  $(PP\dots P, QQ\dots Q):(QQ\dots Q, PP\dots P)$ , в которой числа  $P$  и  $Q$  повторяются как целое  $x$  раз. На рис. 5 повторный шаблон  $R[2](05h, AAh)=(0505h, AAAAh)$  находится в ряду 2, а  $R[2](AAh, 05h)=(AAAAh, 0505h)$  – в ряду 5. Заметим, что для инверсных пар процедуры повтора и масштабирования дают один и тот же результат.

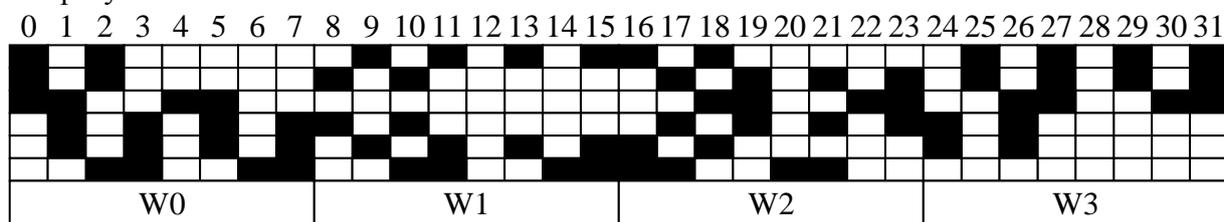


Рис.5. Масштабирование и повторение пары шаблонов  $(05h, AAh):(AAh, 05h)$

Для активации дефектов PSF4 в каждом произвольном триplete ячеек массива  $C_\alpha, C_\beta$  и  $C_\gamma$ , должны быть созданы все восемь возможных комбинаций. Проверка состояний и обоих переключений каждой ячейки массива  $C_e$ , которая, как и в предыдущем случае, производится элементами  $\{\uparrow(re, w\bar{e}, r\bar{e}, we, re); \uparrow(re)\}$  после каждого начального заполнения, обнаруживает дефект.

Когда соседние ячейки, или ячейки-агрессоры имеют одинаковые значения активации  $C_\alpha C_\beta C_\gamma = 000$  или  $C_\alpha C_\beta C_\gamma = 111$ , дефект обнаруживается первыми проходами маршевого теста со сплошными заполнениями  $00h$  и  $FFh$ . Очевидно, что при остальных шести наборах  $\{100, 010, 110, 001, 101, 011\}$  две из трёх соседних ячеек всегда имеют одинаковое значение, которое будем обозначать  $e$ , а одна –  $\bar{e}$ . Далее, значком  $*$  перед значением будем обозначать чётность позиции бита (ячейки), отличную от чётности двух других. Распределение ячеек по словам (байтам) может быть следующим: все три ячейки  $C_\alpha, C_\beta$  и  $C_\gamma$  могут быть в одном слове  $W_\nu$ , две любые могут быть в одном слове  $W_\mu$ , третья – в другом  $W_\nu$ , все ячейки могут быть в разных словах  $W_\lambda, W_\mu$  и  $W_\nu$ .

Для определения заполнений, активирующих PSF4, были рассмотрены все возможные сочетания значений и чётности позиций трёх ячеек-агрессоров (табл. 1). Для случаев, когда

ячейки с одинаковым значением имеют одинаковые чётности позиции в слове или когда такого совпадения нет, но все три ячейки размещаются в одном слове, было составлено 24 однобайтовых шаблона. Для случаев, когда две из трёх ячеек находятся в одном слове, составлено 26 перестановочных пар шаблонов. Из теоремы покрытия парных комбинаций следует, что определённое количество операций репликации этих исходных пар перестановки, зависящее от размера памяти, покрывает все возможные комбинации в таких триплетах ячеек. Наконец, теорема позволяет свести случаи расположения ячеек в трёх разных словах к первым двум случаям применением операций масштабирования ко всем исходным шаблонам.

Таблица 1. Позиции ячеек-агрессоров и шаблоны активации PPSF4 и APSF4 (повторяющиеся шаблоны выделены серым)

Расположение соседних ячеек	Шаблоны активации PSF4
$(e,e,e) \in \forall$ , $(e,e,*e) \in \forall$	00h,FFh
$(e,e,*\bar{e}) \in \forall$	55h,AAh
$(e,*e,\bar{e}) \in W_v$	22h, 88h, 0Ah, A0h, 77h, DDh, 5Fh, F5h, 11h, 44h, 05h, 50h, BBh, EEh, AFh, Fah
$(e,e,\bar{e}) \in W_v$	22h, 88h, 0Ah, A0h, 28h, 82h, 11h, 44h, 05h, 50h, 14h, 41h
$(e,\bar{e}) \in W_\mu$ , $*e \in W_v$	R[x](P,Q):(Q,P) для пар чисел (P,Q): чётная позиция *e – (22h,00h), (22h,55h), (88h,00h), (88h,55h), (0Ah,00h), (0Ah,55h), (A0h,00h), (A0h,55h); нечётная позиция *e – (11h,00h), (11h,AAh), (44h,00h), (44h,AAh), (05h,00h), (05h,AAh), (50h,00h), (50h,AAh)
$(e,\bar{e}) \in W_\mu$ , $e \in W_v$	R[x](P,Q):(Q,P) для пар чисел (P,Q): чётная позиция ячеек – (11h,00h), (11h,55h), (44h,00h), (44h,55h), (05h,00h), (05h,55h), (50h,00h), (50h,55h); нечётная позиция ячеек – (22h,00h), (22h,AAh), (88h,00h), (88h,AAh), (0Ah,00h), (0Ah,AAh), (A0h,00h), (A0h,AAh)
$(e,*e) \in W_\mu$ , $\bar{e} \in W_v$	R[x](00h,FFh):(FFh,00h)
$(e,e) \in W_\mu$ , $\bar{e} \in W_v$	R[x](00h,FFh):(FFh,00h)
$(*e,\bar{e}) \in W_\mu$ , $e \in W_v$	R[x](55h,AAh):(AAh,55h)
$C_\alpha \in W_\lambda$ , $C_\beta \in W_\mu$ , $C_\gamma \in W_v$	M[x](P) и M[x](P,Q):(Q,P) всех предыдущих вариантов, кроме $(e,e,e) \in \forall$ , $(e,e,*e) \in \forall$

Сложность полученного теста  $N(6+1/W)(100\log_2 N-398)$ , формула верна для  $N>128$ . Обратим внимание, что данные заполнения обнаруживают и все PSF3. Если принять разрядность микросхем равной 8, приближённо длина теста составит  $612N(\log_2 N-4)$ .

**Третья глава** посвящена технической реализации программно-аппаратного комплекса диагностирования магистральных структур и памяти вычислительных модулей.

Существует три возможных варианта размещения и запуска программ-тестов, предназначенных для определения неисправностей процессорных (системных) магистралей и памяти вычислительных модулей. Какой из них может быть реализован, зависит от того, имеет ли микропроцессор модуля внутрисхемную эмуляцию, и от типа встроенной памяти. Процессоры без эмуляции допускают только такой вариант, когда тест системной магистрали проводится внешним задатчиком. Именно для модулей данного типа, в основном, и предназначается разработанный комплекс. Для тестирования мультиплексированной периферийной шины была выбрана схема двусторонней проверки, в которой функции генератора входных сигналов и приёмника выходных поочередно

выполняют задатчик магистрали (процессор модуля) и ведомый. Данная схема обеспечивает более детальную диагностику по сравнению со схемой односторонней проверки.

Система диагностирования состоит из тестируемого вычислительного модуля ВМ, диагностической ЦВМ (в дальнейшем ЦВМ), компьютера-монитора и устройства управления и индикации (рис. 6). Основой комплекса является диагностическая ЦВМ, которая выполняет тесты системной (процессорной) магистрали, микросхем памяти и мультиплексированной магистрали ввода-вывода (МПМ) модуля. Компьютер-монитор, соединяемый с ЦВМ по последовательному каналу ПК, служит для управления её работой и отображения поступающей из неё информации. Предусмотрена возможность автономной работы ЦВМ без компьютера, в этом случае некоторые простые функции монитора реализуются с помощью устройства управления и индикации.

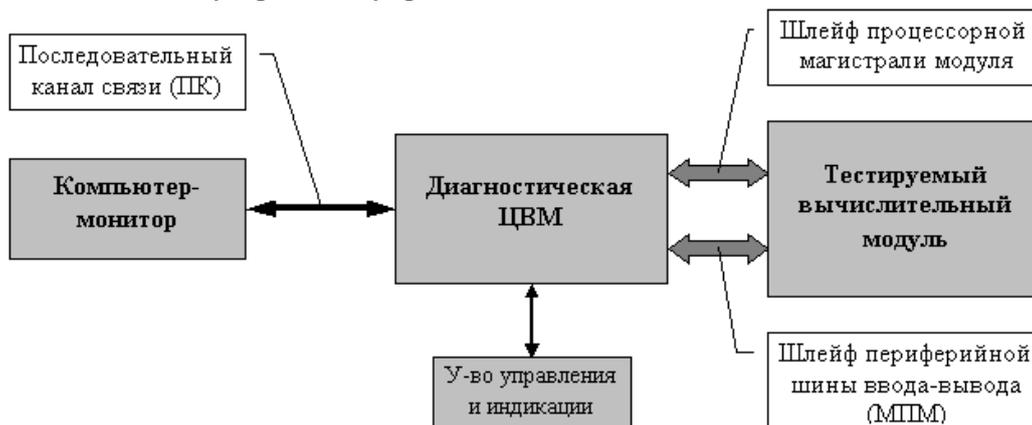


Рис. 6. Система диагностирования

Применение микропроцессора ЦВМ в качестве задатчика системной магистрали модуля вместо адаптеров, устанавливаемых в разъёмы периферийных шин ISA, EISA или PCI компьютеров, даёт два существенных преимущества:

- возможно добиться полной идентичности магистральных сигналов по формам диаграмм и временным параметрам при тестировании и в рабочем режиме, что является критичным для достоверного обнаружения ряда типов дефектов,
- тестируемый модуль вместе с диагностической ЦВМ может быть расположен на удалении от компьютера-монитора, что необходимо при проверках в специальных условиях (например, в температурных камерах).

Вычислительное ядро ЦВМ включает микропроцессор МР со встроенными Flash-модулями программной памяти и дополнительную внешнюю память программ SRAM (рис. 7). Блок соединения магистралей ВСС (Bus Connection Circuitry) в режиме тестирования (TEST=1) предоставляет процессору ЦВМ доступ к пространству данных и кода системной магистрали модуля через адресное окно DS:2000h-3FFFh, а к пространству ввода-вывода – через окно IS:8000h-9FFFh. Узловыми частями блока являются управляемые магистральные передатчики адреса и сигналов управления, двунаправленный приёмопередатчик данных и схема расширения/перестановки адреса.

Блок DPRAM/MPXBI (Dual port RAM/Multiplexed bus interface) служит для связи с мультиплексированной параллельной магистралью модуля. Он состоит из двухпортовой памяти данных DPRAM, регистра и внутренней шины адреса правого порта, схемы управления доступом к памяти со стороны МПМ, схем генерации прерывания и установки разовых команд, а также двунаправленного параллельного порта для связи ЦВМ с устройством индикации. Введение памяти с управляемым доступом позволяет использовать ЦВМ не только в целях двустороннего тестирования МПМ, но и в составе комплексов полунатурного моделирования, для чего предусмотрена также обработка прерывания LINT

от левого порта DPRAM. Память блока целиком или частично также может быть использована в качестве обычной памяти данных.

Два участка ПК связаны через конвертор USB-UART. Он поддерживает полноскоростной режим стандарта USB 2.0 и позволяет задавать скорость передачи по UART от 300 бит/с до 3 Мбит/с, различные форматы слова и режимы контроля чётности. Наличие приёмного и отправочного FIFO-буферов обеспечивает высокую пропускную способность ПК за счёт того, что минимизируется число кадров, требующихся для передачи массивов по USB, и время, которое затрачивается операционной системой ЦВМ на передачу данных.

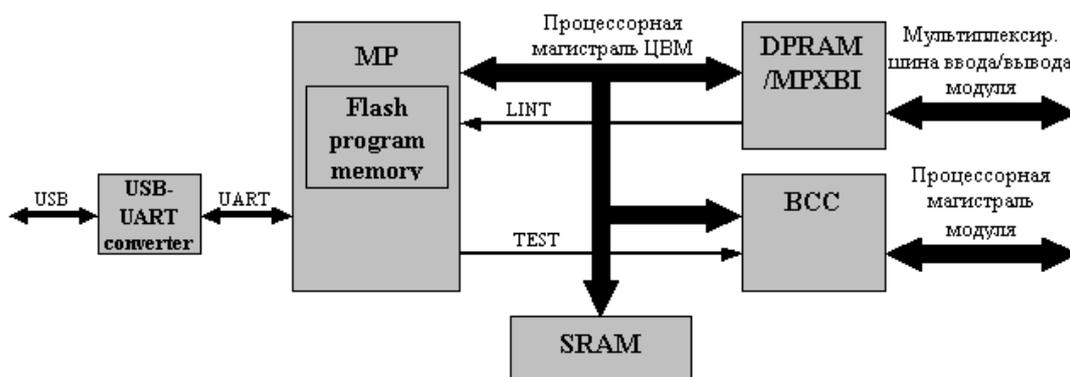


Рис. 7. Структурная схема диагностической ЦВМ.

В разделе описания аппаратной части ЦВМ приводится обоснование применяемых схемотехнических решений, а также детально рассматривается работа спроектированных узлов. Логические схемы блоков ВСС и DPRAM/MPXVI реализованы на ПЛИС EPM7160S. Исходные файлы для компилятора MAX+PLUS II создавались на языке описания цифровых устройств AHDL. Средствами из этого же программного пакета было проведено моделирование работы схем и их элементов, проверка на неперевышение ограничивающих временных параметров, а также осуществлена разводка и программирование ПЛИС.

Так как выполнение целевых программ (тестов) производится удалённой ЦВМ, возникла задача организации её взаимодействия с монитором. С учётом возможности использования машины в различных режимах, а также других перспектив её применения оказалось целесообразным создать общий программный интерфейс обмена сообщениями между ЦВМ и монитором.

Как показывает опыт, наиболее эффективно процесс поиска неисправностей реализуется при выполнении диагностических тестов в режиме отладки. Существенным моментом здесь является не только получение итоговых результатов проверок, но и достаточно полная визуализация контекста и возможность оперативного вмешательства специалиста. Типичными случаями такого вмешательства являются переход на операции заикливания (при необходимости произвести электроизмерительный эксперимент), выбор из нескольких возможных операций, изменение тестовых значений, создание точек останова. Одновременно с этим потребность реализации стандартных действий, в частности, проверок теста, в виде низкоуровневых процедур послужила причиной разработки интерпретируемого языка, получившего рабочее название CDS (Communication Device Sequence).

В работе рассматриваются те компоненты системного программного обеспечения комплекса, которые непосредственно обеспечивают выполнение тестовых программ.

Операционная система (ОС) является базовой частью программного обеспечения ЦВМ и хранится в постоянной памяти. Она обеспечивает загрузку, запуск и выполнение целевых программ, их взаимодействие с аппаратной частью и связь с монитором по

последовательному интерфейсу, а для интерпретируемых CDS-программ – также и режим отладки. ОС включает восемь функциональных единиц:

- блок инициализации
- приёмопередатчик сообщений последовательного канала
- исполнительный блок системных команд
- интерпретатор CDS
- отладчик CDS
- обработчики прерываний таймера и двухпортовой памяти DPRAM
- набор процедур для вызова загружаемым кодом
- блок перепрограммирования Flash-памяти ЦВМ

ОС ЦВМ обеспечивает четыре режима выполнения целевых функций, которые должны выбираться исходя из приоритета требований, которые предъявляет каждая конкретная задача. Для задач диагностики оптимальным является режим выполнения интерпретируемого CDS-кода.

Подробно рассматривается организация обмена ЦВМ с монитором по последовательному каналу, типы и форматы сообщений, работа блоков инициализации и системных команд, приёмопередатчик ПК и режимы его функционирования.

Дается общая характеристика языка. CDS является стековым языком типа ассемблера – большая часть команд использует в качестве исходных операндов и/или операндов назначения ячейки вершины стека данных. Набор команд языка был составлен исходя из анализа разработанных тестов, а также программ-драйверов для устройств связи в комплексах полунатурного моделирования. Поэтому так же, как и для всех типичных операций устройств связи, язык предоставляет для проведения тестирования процессорной магистрали и памяти модуля специальные команды. Язык CDS поддерживает основные положения концепции модульного программирования, обеспечивая функциональную декомпозицию задачи и простоту информационных и управляющих связей между структурными единицами программы.

Описывается структура исполняемого модуля, состоящего из секций предопределённых типов, режимы адресации и форматы команд. Рассматривается набор команд, при этом наибольшее внимание уделяется специфическим высокоуровневым механизмам языка, с помощью которых осуществляется обмен данными через ПК, вызов процедур, обработка прерываний, размещение массивов, тестирование подключаемого модуля, работа с таймером. Среди других особенностей языка следует упомянуть наличие команд быстрого вычисляемого перехода, перемещения и обработки массивов, эффективной организации циклов, доступа к пространству данных и ввода-вывода.

Приводится структура и алгоритм работы интерпретатора CDS-программ. Рассматриваются принципы построения и совместная работа процедур-отладчиков в ЦВМ и мониторе, обеспечивающие стандартный набор отладочных операций и визуализацию контекста CDS-программ. Следует отметить, что механизм отладки является прозрачным и при отсутствии точек останова не препятствует выполнению программ в реальном времени.

**В заключении** сформулированы основные результаты диссертационной работы.

1) Разработана функциональная классификация магистральных сигналов и связанных с ними неисправностей, созданы явные модели поведения неисправной подсистемы для всех видов магистральных дефектов. Сформирован комплекс подходов и методов построения тестовых последовательностей, позволяющих реализовать максимальную для программного способа глубину разбиения неисправных состояний параллельных магистралей вычислительных модулей.

2) На основе полученных моделей, принципов и методов разработаны алгоритмы проверок детектирующей тестовой последовательности и процедур локализации неисправностей ДПИМ в системах, содержащих статическую оперативную (SRAM) и постоянную (ROM) память. Испытания полностью подтвердили детектирующую и

различающую способности алгоритма, в том числе при наличии множественных дефектов и логической неустойчивости.

3) Проведён анализ моделей дефектов и современных способов тестирования SRAM, который показал, что существующие методы обнаружения кодочувствительных дефектов PSF не обеспечивают удовлетворительной для наладочных испытаний полноты покрытия. Сопоставление методов позволило наметить подходы, которые позволяют достигать полного покрытия и для нередуцированных моделей PSF. Кроме этого, было показано, что задача обнаружения дефектов классических типов полностью решается существующими бит-ориентированными маршевыми тестами.

4) Разработан новый метод обнаружения активных и пассивных кодочувствительных неисправностей PSF3 и PSF4, имеющий 100%-ное покрытие. Для этого был предложен принцип чётности позиции ячеек и их блоков в адресном пространстве памяти, введены понятия инверсной пары, шаблона, перестановочной пары шаблонов, операции их регулярного преобразования и сформулирована теорема покрытия парных комбинаций.

5) Спроектирована специализированная ЦВМ для проведения тестов параллельных магистралей и памяти модулей. Она обеспечила идентичность временных параметров сигналов в тестовом и рабочем режиме и возможность проверки удалённого объекта, что требуется при испытаниях в специальных условиях. На её основе создан программно-аппаратный комплекс диагностики.

6) Подготовлено системное программное обеспечение комплекса – операционная система ЦВМ и программа-монитор, разработаны интерпретируемый язык CDS и средства создания и отладки CDS-программ, обеспечивающие эффективную организацию процесса диагностирования.

7) На базе разработанных средств созданы автоматизированные рабочие места для контроля и наладки опытных образцов БЦВМ в ГосНИИП и серийных образцов на заводе-изготовителе БЦВМ. Разработаны диагностические тесты для системных и периферийных магистралей и памяти вычислительных модулей различных типов, построенных на основе микропроцессоров 1867ВЦ6, 1867ВЦ2Т, 1867ВЦ5Т и Л1867ВМ2.

#### **Публикации по теме диссертации:**

1. Выборнов П.В., Гаврилин Б.Н. О подходах к составлению диагностических тестов вычислительного ядра БЦВМ. – Вопросы оборонной техники, серия 9, 2010, вып. 3-4, с. 140-148.

2. Выборнов П.В., Гаврилин Б.Н. Детектирующая тестовая последовательность для магистральных структур в системах с оперативной памятью. – Вестник Южно-Уральского государственного университета, серия «Компьютерные технологии, управление, радиоэлектроника», 2011, вып. 14, №23[240], стр. 53-59.

3. Аносов Б.Н., Выборнов П.В., Гаврилин Б.Н., Ефремов Д.В., Зверобой Б.М., Карашук А.В. и др. Пояснительная записка СЧ технического проекта АБ-151. – ФГУП «ГосНИИП», 2009.

4. Арндарчук В.Б., Выборнов П.В., Данилушкин Ю.В., Карашук А.В. Средства диагностики БЦВУ Заря-74. Пояснительная записка к техническому проекту. – ФГУП «ГосНИИП», 2010.