

ՀԱՅԱՍՏԱՆԻ ՀԱՆՐԱՊԵՏՈՒԹՅԱՆ ԿՐԹՈՒԹՅԱՆ ԵՎ ԳԻՏՈՒԹՅԱՆ
ՆԱԽԱՐԱՐՈՒԹՅՈՒՆ

ՀԱՅԱՍՏԱՆԻ ԱԶԳԱՅԻՆ ՊՈԼԻՏԵԽՆԻԿԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ

ԱՄԻՐՔԱՆՅԱՆ ԿԱՐԵՆ ՍՊԻՐԻԴՈՎԻՇ

ԹԵՍԱՎՈՐՄԱՆ ԱՎՏՈՄԱՏԱՑՎԱԾ ՀԱՄԱԿԱՐԳԻ ՄՇԱԿՈՒՄԸ
ՍՏԱՏԻԿ ՀԻՇՈՂՈՒԹՅԱՆ ՍԱՐՔԵՐԻ ՀԱՄԱՐ

Ե.13.02 «Ավտոմատացման համակարգեր» մասնագիտությամբ
տեխնիկական գիտությունների թեկնածուի գիտական աստիճանի
հայցման ատենախոսության

ՄԵՂՍԱԳԻՐ

Երևան 2015

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ АРМЕНИЯ

НАЦИОНАЛЬНЫЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ АРМЕНИИ

АМИРХАНИЯН КАРЕН СПИРИДОНОВИЧ

**РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ТЕСТИРОВАНИЯ
СТАТИЧЕСКИХ УСТРОЙСТВ ПАМЯТИ**

АВТОРЕФЕРАТ

диссертации на соискание ученой степени кандидата
технических наук по специальности 05.13.02-
“Системы автоматизации”

Ереван 2015

Ատենախոսության թեման հաստատվել է Հայաստանի ազգային պոլիտեխնիկական համալսարանում (ՀԱՊՀ):

Գիտական ղեկավար՝ ֆիզ.-մաթ. գ.թ. Վ. Ա. Վարդանյան

Պաշտոնական ընդդիմախոսներ՝ տ.գ.դ., պրոֆ. Ռ.Ռ. Վարդանյան
տ.գ.թ., Խ.Գ. Շառոյան

Առաջատար կազմակերպություն՝ Ռուս-հայկական (սլավոնական) համալսարան

Ատենախոսության պաշտպանությունը տեղի կունենա 2015թ. դեկտեմբերի 24-ին, ժամը 14:00-ին, ՀԱՊՀ-ում գործող «Կառավարման, ավտոմատացման և էլեկտրոնիկայի» 032 Մասնագիտական խորհրդի նիստում (հասցեն՝ 0009, Երևան, Տերյան փ., 105, 17 մասնաշենք):

Ատենախոսությանը կարելի է ծանոթանալ ՀԱՊՀ-ի գրադարանում:
Սեղմագիրն առաքված 2015թ. նոեմբերի 24-ին:

032 Մասնագիտական խորհրդի
գիտական քարտուղար, տ.գ.դ.



Ա. Գ. Ավետիսյան

Тема диссертации утверждена в Государственном национальном политехническом университете Армении

Научный руководитель: к. ф-м.н. В.А. Варданян

Официальные оппоненты: д.т.н. проф. Р.Р. Варданян

к.т.н. Х.Г. Шароян

Ведущая организация: Российско-армянский (славянский) университет

Защита диссертации состоится 24-го декабря 2015г. в 14:00 ч. на заседании Специализированного совета 032 - “Управления, автоматизации и электроники”, действующего при Национальном политехническом университете Армении (НПУА), по адресу: 0009, г. Ереван, ул. Теряна, 105, корпус 17.

С диссертацией можно ознакомиться в библиотеке НПУА.

Автореферат разослан 24-го ноября 2015г.

Ученый секретарь
Специализированного совета 032
д.т.н.



А.Г. Аветисян

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Одной из важных особенностей современных интегральных полупроводниковых приборов (ИПП) является содержание в этих устройствах большого количества встроенных устройств памяти (ВУП), различных по количественному составу, функциональному назначению и структурному исполнению. В данной диссертационной работе рассмотрены вопросы, касающиеся исключительно статических устройств памяти (СУП). В дальнейшем, говоря о СУП или о ВУП, будем подразумевать исключительно статические устройства памяти (Static Random Access Memory – SRAM). Наряду с развитием технологии изготовления полупроводниковых приборов количество ячеек ВУП неуклонно растет и в зависимости от типа устройства может достигать в современных ИПП нескольких сот миллионов. По имеющимся прогнозам, проведенным на базе статистических исследований, к 2017 году в больших ИПП поверхность, занимаемая ВУП, будет достигать порядка 70% всей поверхности кристалла микросхем. В то же время известно, что ячейка памяти проектируется и изготавливается по специальной технологии, обеспечивающей максимальную компактность и плотность топологии ячеек памяти. Поэтому становится очевидным, что область, занимаемая ячейками памяти в кристалле ИПП, наиболее уязвима для проявления физических и технологических дефектов. С уменьшением размеров транзисторов и одновременным многократным увеличением их количества в микросхемах ИПП, а также с усложнением технологии изготовления ИПП, выход годных микросхем (ВГМ) имеет тенденцию к понижению. ВГМ определяется как соотношение общего числа произведенных микросхем к числу годных. Очевидно, что на повышение ВГМ решающее влияние оказывает надежность технологии изготовления устройств памяти.

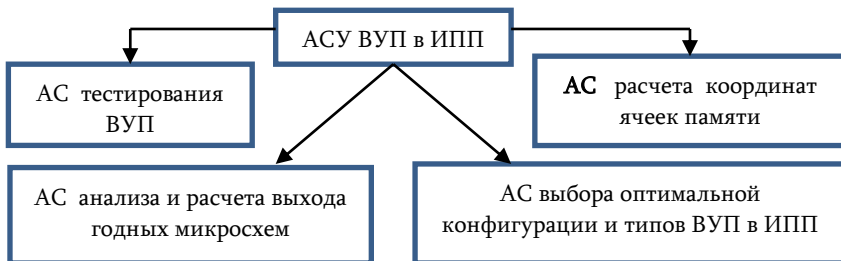


Рис. 1. Элементы АСУ ВУП ИПП

В процессе работы с устройствами памяти возникает ряд задач, требующих неперенного применения автоматизированных систем управления (АСУ). На рис. 1 представлена блок-схема АСУ ВУП ИПП. Элементами АСУ ВУП ИПП являются:

- автоматизированная система (АС) тестирования ВУП;
- АС расчета координат ячеек памяти;
- АС выбора оптимальной конфигурации и типов ВУП в ИПП;
- АС анализа и расчета выхода годных микросхем.

Исследования показали, что решение перечисленных задач невозможно без достоверных данных о строении ВУП. Эти задачи требуют от разработчиков микросхем и ВУП, в частности, знания и применения в программных средствах достоверной информации о структуре используемых СУП (рис. 2). В современных компиляторах, генерирующих ВУП, достоверная информация о структуре памяти содержится в производственном файле компилятора, представляемого в графическом формате – GDSII (Graphic Database System II). Из-за большого объема файла графического формата GDSII его обработка требует много времени и ресурсов, и поэтому GDSII файл крайне неудобен для применения в качестве входной информации для прикладных программ.

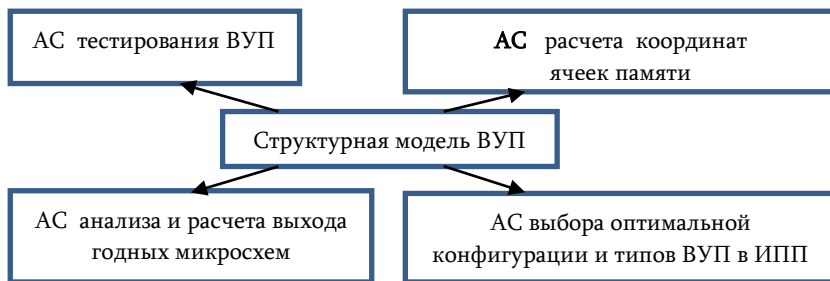


Рис. 2. Применение структурной модели в АСУ ВУП ИПП

Данная диссертационная работа посвящена решению задачи получения, обработки, проверки и применения информации о структуре СУП. В качестве альтернативы GDSII файлу предлагается использовать структурную модель памяти (СМП), написанную на программном языке описания скриптов - Tool Command Language (TCL), и содержащую всю необходимую информацию о строении СУП.

С целью повышения ВГМ в ВУП применяются специальные схематические и технологические решения, в том числе: а) еще на стадии проектирования в матрицу ВУП включаются запасные строки и колонки ячеек памяти с целью их дальнейшего использования для восстановления неисправных элементов (колонок, строк или ячеек) памяти. Причем процесс восстановления может осуществляться как непосредственно после производственного тестирования (тем самым увеличивая процент ВГМ), так и в момент включения питания микросхемы в процессе эксплуатации микросхемы (тем самым увеличивая надежность и срок работы устройства); б) включение в состав ВУП специального тестирующего процессора, обеспечивающего самотестирование ВУП (Built-in

Self Test – BIST) и самовосстановление работы ВУП (Built-in Redundancy Allocation - BIRA).

По сравнению с внешними системами, встроенные системы тестирования обладают важными свойствами, определяющими их превосходство и широкое распространение:

1. Встроенные системы более “мощные” и они позволяют осуществлять проверку ИПП в широком диапазоне изменяемых внешних параметров процесс/напряжение/температура (Process/Voltage/Temperature PVT), производя исследование и контроль в широком диапазоне частот и напряжений практически для неограниченного количества сигналов ИПП.

2. Внедрение и использование встроенных систем осуществляется быстрее внешних устройств, что и требует обеспечения надежного, физического подключения. Кроме того, встроенные системы более гибкие и позволяют использовать многочисленные готовые, многократно проверенные программные и схемотехнические решения, например, подключения P&P (Plug and Play), JTAG (Joint Test Action Group) и других устройств.

3. Использование встроенных систем дешевле по сравнению с внешними системами. Для сравнения отметим, что стоимость только одного современного осциллографа составляет несколько сот тысяч долларов США (не считая коммуникаций для подключения тестируемых ИПП и программного обеспечения), а стоимость встроенных систем исчисляется десятками тысяч долларов США.

Очевидно, что запасные элементы и дополнительный тестовый процессор увеличивают суммарную занимаемую площадь ВУП на кристалле микросхемы, что повышает стоимость производства микросхемы, но получаемый конечный суммарный выход годных микросхем и повышение надежности работы ИПП полностью оправдывают дополнительные произведенные затраты.

Одним из важных элементов описанной системы является процесс “BIST” - самотестирования ВУП, который подразумевает использование высокоэффективных и вместе с тем максимально компактных алгоритмов тестирования устройства памяти. Такими алгоритмами являются линейно-сложные, так называемые “Марч” (от английского “March”) алгоритмы тестирования, которые, несмотря на простоту исполнения, позволяют обнаруживать практически все известные типы неисправностей в различных узлах ВУП.

Объект исследования. В работе объектами исследования являются:

- образцы встроенных СУП, которые были сгенерированы с помощью компиляторов памяти различных технологических процессов (начиная с 250 нм до современных 14 нм) изготовления ИПП;
- строение образцов ВУП, физическое строение ячеек памяти;
- самотестируемые и восстанавливаемые системы памяти;

- получение и проверка структурной модели СУП;
- физические дефекты, возникающие в ячейках памяти, матрице памяти и в других функциональных узлах образцов СУП;
- применение структурной модели СУП в задачах тестирования, обнаружения дефектов, диагностики и восстановления работоспособности ВУП.

Целью работы является разработка автоматической системы тестирования СУП. Исходя из цели исследования, в работе были поставлены и решены следующие задачи:

- исследование и классификация элементов структурной модели СУП;
- создание программных средств АС для генерации и проверки СМП;
- создание программной среды для автоматического получения СМП из графического GDSII файла образца памяти;
- создание программных средств, позволяющих: а) внедрять различные физические и/или поведенческие дефекты в узлы СУП; б) создавать, исследовать и отлаживать тестовые алгоритмы, обнаруживающие эти встроенные дефекты.

Диссертация представляет собой научно обоснованное теоретическое исследование, обеспечивающее практическое решение перечисленных задач.

Методы исследования. В процессе исследования были использованы основные положения теории проектирования цифровых, аналоговых и смешанных ИС; основные положения технологии производства полупроводниковых приборов; методы математического моделирования электронных схем; методы ориентированного программного обеспечения.

Научная новизна работы.

1. Создана классификация примитивов (элементов) структуры статических устройств памяти, разработан язык описания структурной модели статической памяти.
2. Разработана и внедрена программная среда автоматической проверки СМП (Structural Information Verification flow – SIV).
3. Разработана и внедрена программная среда, обеспечивающая автоматическое извлечение СМП из графического GDSII файла образца памяти (Structural Information Extraction flow – SIE).
4. Разработана и внедрена программная среда (Defect Injection flow - DI), обеспечивающая в автоматическом режиме:
 - внедрение различных физических и поведенческих дефектов в узлы СУП;
 - описание, исследование и отлаживание тестовых алгоритмов, обнаруживающих встроенные дефекты в различных узлах СУП.

Достоверность научных положений подтверждается результатами различных исследований и анализом результатов схемотехнического (SPICE) моделирования, произведенными для более чем 800 различных типов (большой плотности, быстродействующих, с низкой потребляемой мощностью и т.д.) и

различных технологических процессов изготовления (от 250 нм до 14 нм) компиляторов ВУП, с использованием технологий производства различных производителей, таких как TSMC, UMC, IBM, Intel, Samsung и т.д.

Практическая ценность работы. В настоящее время разработанная модель СМП является обязательной частью всех компиляторов памяти, проектируемых в компании Synopsys, и широко используется в АСУ ВУП, таких как: АС тестирования, АС восстановления работоспособности СУП, АС планирования конфигураций и группирования ВУП в ИПП. Разработанная АС SIV обеспечивает в автоматическом режиме проверку файла СМП компиляторов памяти, а АС SIE - автоматическое получение структурной модели образцов памяти из графического файла GDSII образца памяти. Программная среда DI широко используется как инструмент внедрения и исследования дефектов в MOSFET и FinFET СУП, генерации “Марч” тестовых алгоритмов для исследуемых дефектов.

На защиту выносятся следующие научные положения:

1. Структурная модель СУП, язык описания структурной модели.
2. Алгоритм проверки структурной модели СУП и созданная на его базе АС проверки модели СУП.
3. Алгоритм автоматического извлечения СМП из графического файла GDSII образца СУП и созданная на его базе АС извлечения СМП образца.
4. Методика внедрения различных физических и поведенческих дефектов в узлы СУП и создания тестовых “Марч” алгоритмов, обнаруживающих встроенные дефекты в различных узлах СУП.

Внедрение. Разработанная модель описания структуры СУП памяти для компиляторов ВУП, программные средства SIV, SIE, DI внедрены в АС-му тестирования компании Synopsys, а также используются в АС-ах заказчиков компании Synopsys, таких как APPLE, Samsung, AMD, TSMC и др. В перечисленных компаниях проведены работы по обучению персонала.

Апробация работы. Основные научные и практические результаты диссертации докладывались на:

- 11-ом Международном научно-техническом симпозиуме “International on-line Testing Symposium”, IOLTS’11 (Афины, Греция, 2005);
- 8-й Международной научно-технической конференции “International Conference on Computer Science and Information Technologies”, CSIT 2011 (Ереван, Армения, 2011);
- 10-ом Международном симпозиуме “EAST-WEST DESIGN & TEST SYMPOSIUM”, EWDTs 2012 (Харьков, Украина, 2012);
- семинаре Государственного инженерного университета Армении (2014).

Публикации. Основные положения диссертации опубликованы в четырёх патентах и четырёх научных работах, список которых приводится в конце автореферата.

Структура и объем работы. Диссертационная работа состоит из введения, трех глав, основных выводов, списка литературы из 103 наименований и 6 приложений. Основной текст работы составляет 126 страниц, включая 68 рисунков и 17 таблиц. Общий объем работы составляет 148 страниц. Диссертационная работа написана на армянском языке.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность темы диссертации, сформулированы цель и задачи исследования, представлены научная новизна, практическая значимость и основные научные положения, выносимые на защиту.

В первой главе рассмотрены общие принципы построения, а также назначение отдельных узлов статических устройств памяти. Детально изложены принципы построения декодеров строк и колонок СУП. На рис. 3 представлена блок-схема СУП. Как видно из схемы, СУП состоит из 8 основных блоков.

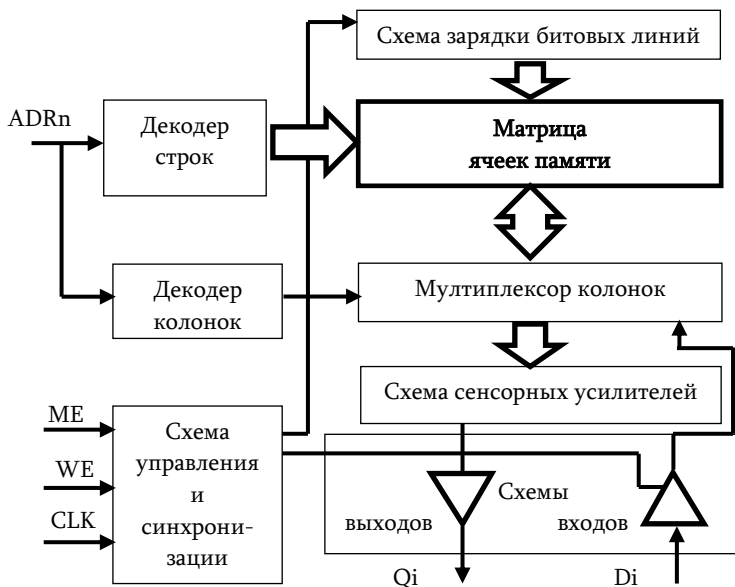


Рис. 3. Блок-схема статического устройства памяти:
 ADR_n – адресная шина, ME – сигнал выборки памяти, WE – сигнал разрешения записи, CLK – синхронизирующие импульсы,
 Q_i / D_i – выходные /входные сигналы данных

Показано, что физическое строение этих блоков СУП имеет важное значение в решении ряда задач АСУ, таких как: а) расчет координат ячеек памяти; б)

тестирование СУП; в) решение задачи выбора оптимальной конфигураций для используемых ВУП и г) расчет выхода годных микросхем.

В процессе исследований было установлено, что топология структуры СУП состоит из двух частей: матрицы ячеек памяти (МЯП) и зоны топологии остальных узлов СУП. Матрица памяти содержит элементы СУП, построенные исключительно из ячеек памяти. Это сами ячейки памяти (memory bit-cells area), запасные (redundancy), а также дополнительные, балансирующие (dummy) строки и колонки.

Определение 1. Топология элементов, расположенных вне МЯП, называется “страпом” (от англ. Strap). Установлено, что для прикладных программ, используемых в СУП, имеют значение не функциональность “страпа”, а его размер (высота и ширина) и физическая позиция в топологии СУП.

Исследования показали, что в процессе работы над топологией СУП, разработчик оптимизирует как топологию библиотечных элементов СУП, так и их расположение на подложке микросхемы. В процессе этой оптимизации топологии возникает несоответствие логического распределения библиотечных элементов (ячеек, узлов и сигналов СУП) с физическим расположением этих блоков и сигналов на кристалле микросхемы СУП.

Определение 2. Несоответствие строения микросхемы к логическому описанию СУП называется скремблированием (от англ. scrambling). Например, возникновение скремблирования в СУП приводит к несоответствию логической нумерации строк (колонок) в матрице СУП относительно физического распределения строк (колонок) в этой матрице. Логической нулевой строке R0 на физическом уровне в результате скремблирования может соответствовать третья строка – PR2 (если считать с нуля) на матрице ячеек памяти. Нулевому логическому столбцу C0 может соответствовать четвертая физическая строка PC5 матрицы ячеек. В результате при обращении к СУП по логическому нулевому адресу {R0, C0} в нашем примере физический адрес на матрице ячеек будет {PR2, PC5}. Очевидно, что скремблирование может оказать существенное влияние на результаты работы тестового алгоритма и привести к невозможности обнаружения дефекта в матрице ячеек.

В процессе исследования физического строения образцов микросхем СУП в рамках диссертационной работы была разработана классификация примитивов скремблирования, описывающая физическое строение образцов (структура образцов), а впоследствии - и компиляторов СУП. Было установлено, что структура памяти описывается 19-ю элементами структурного скремблирования, дано описание каждого из 19 элементов скремблирования СПУ.

Процесс разработки компиляторов СУП включает создание документа описания скремблинг-элементов устройства (СЭ СУП). В компиляторах СЭ СУП описывались в виде: набора таблиц, уравнений или текста, что сильно затрудняло автоматизацию использования СЭ СУП в прикладных программах.

В процессе исследований была определена структурная модель образца СУП как набор функций описаний всех СЭ СУП и straps топологии образца СУП.

$$M_{\text{comp}} = \{ \text{Scr1}, \text{Scr2}, \dots, \text{Scr } n, \text{St1}, \text{St2}, \dots, \text{Stm} \},$$

где M_{comp} - структурная модель компилятора/образца СУП; Scr n – описание скремблинг-элементов СУП; Stm – описание “стрэпов” в топологии компилятора/образца.

В качестве языка описания Структурной модели СУП было предложено использовать язык описания скриптов - TCL, а все элементы СЭСУП представлять в виде TCL процедур. В таблице представлены некоторые скремблинг-элементы структуры СУП (СЭ СУП).

Таблица

Элементы СЭ СУП

Название структурного элемента	Название процедуры	Входные параметры	Выходные параметры
Адрес скремблирование	address_scramble	-	Описание элементов адреса
Функция декодера строк	row_scramble	Логический номер строки	Физический адрес строки
Функция декодера колонок	column_scramble	Логический номер колонки	Физический адрес колонки
VX/ВЫХ скремблирование	io_cell_scramble	Логический номер VX/ВЫХ	Физический адрес VX/ВЫХ

Кроме того, было изучено влияние СЭ СУП на обнаружение неисправностей тестовыми алгоритмами (ТА) BIST. В процессе работы было выявлено, что использование структурной модели в ТА на 30% увеличивает эффективность в обнаружении неисправностей.

Применение предложенной структурной модели дало возможность автоматизировать процесс использования скремблинг-элементов или группы элементов СЭ СУП в прикладных АС тестирования и восстановления работоспособности СУП, расчета координат ячеек памяти в файлах GDSII, выбора оптимальной конфигурации и типов ВУП в ИПП, расчета ВГМ и т.д. В процессе работ генерации структурной модели компиляторов СУП и применения СМП в прикладных программах было выявлено, что основным недостатком процесса получения СМП является отсутствие возможности полного автоматизированного получения СМП и, как следствие, наличие ошибок в СМП из-за влияния человеческого фактора. Для исключения ошибок в СМП компиляторов были разработаны и внедрены: 1) АС проверки файла описания структурной модели компилятора СУП (SIV flow); 2) программная

среда автоматизированного получения структурной модели образца СУП (SIE flow).

Детально рассмотрена схематика стандартной, 6-ти транзисторной ячейки СУП; описаны входные параметры СУП и даны основные формулы расчета физических и геометрических параметров, характеризующих компиляторы СУП.

Во второй главе представлены: 1) методика программирования GDSII файла образца с использованием ячеек программирования «0» и «1»; 2) блок-схемы и подробное описание работы узлов автоматизированных систем SIV и SIE; 3) некоторые результаты экспериментов.

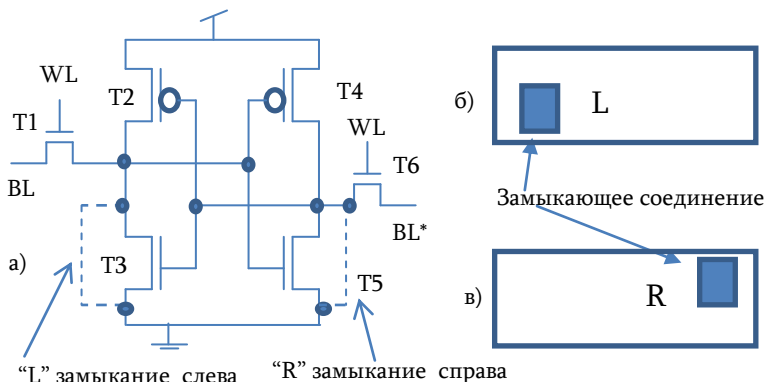


Рис. 4. Программирующие ячейки

Строение ячейки статической памяти (как схематика, так и топология) полностью симметрично (рис. 4 а). Симметричность ячейки памяти делает строение топологии матрицы ячеек СУП регулярным, что является серьезным препятствием для процесса определения скремблинг-примитивов СУП. С целью преодоления данной проблемы, была разработана и применена методика программирования ячейки в матрице СУП. Основная идея данной методики – нарушить симметричность ячейки памяти, тем самым изменив неопределенное состояние ячейки на predetermined состояние «0» или «1». Процесс программирования осуществляется путем замыкания на «общий» провод истока транзистора Т3 для случая «L» или истока транзистора Т5 для случая «R». В этом случае ячейка памяти переходит в состояние с predetermined значением «0» или «1» (см. рис. 4 а).

Для осуществления процесса программирования в топологии создаются программирующие ячейки. Примеры топологий программирующих ячеек «L» и «R» приведены на рис. 4 б и в. Сам процесс программирования топологии матрицы ячеек памяти осуществляется с помощью программы MPT (Memory

Programming Tool), которая производит программирование GDSII файла по образцу шаблона программирования (ШП).

<p>а)</p> <pre> 00000000000000000000 00001000010000100001 00001100011000110001 00001110011100111001 </pre>	<p>б)</p> <pre> 00000000000000000000 10000100000000100001 11000110000000110001 11100111000000111001 </pre>
--	--

Рис. 5. Примеры файла шаблона программирования

На рис. 5 приведены примеры файла шаблона программирования для матрицы ячейки памяти, содержащей: 4 строки, 16 колонок и 4 запасные колонки, расположенные с левой стороны (рис. 5 а) и в центре (рис. 5 б) матрицы. Программа МРТ устанавливает программирующие ячейки “L” и “R” поверх ячеек памяти, тем самым меняя результирующую топологию матрицы.

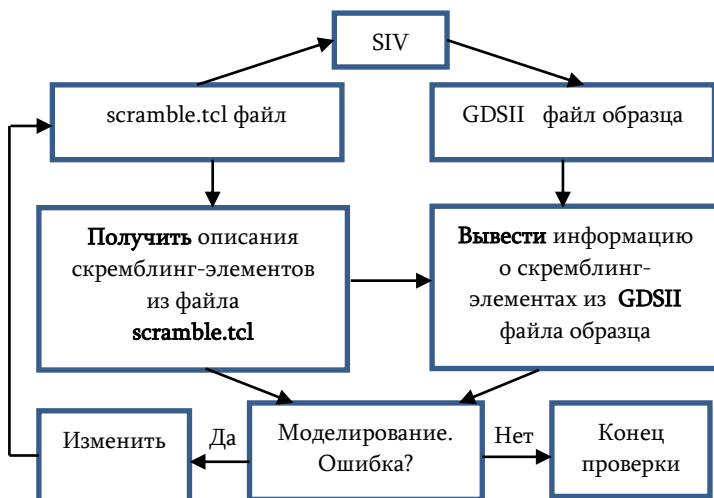


Рис. 6. Принцип работы АС проверки модели СУП

АС SIV - это программная среда, обеспечивающая автоматическую проверку правильности описания СЭСУП в **scramble.tcl** файле компилятора памяти. Проверка производится путем сравнения скремблинг-примитивов, описанных в файле **scramble.tcl** структурной модели, с эталонными. В данном случае эталонными являются скремблинг-примитивы, полученные из графического GDSII файла(ов) СУП. На рис. 6 представлен принцип работы АС SIV. Отметим, что для получения информации о скремблинг-элементах из GDSII файлов образцов SIV осуществляет следующие этапы обработки: 1) на основе информации из файла **scramble.tcl** определяются необходимые для полной проверки конфигурации GDSII файлов и создаются индивидуальные ШП

GDSII файлов; 2) в соответствии с п. 1 создаются **GDSII** файлы СУП, которые проходят специальную обработку по индивидуальным ШП с помощью программы MPT; 3) из обработанных **GDSII** файлов выводятся SPICE файлы и производится моделирование (симуляция) полученных SPICE файлов; 4) если моделирование не выявило ошибок, то описание скремблинг-примитивов в файле scramble.tcl компилятора СУП также не содержит ошибок.

AC SIE - программная среда, которая позволяет в автоматическом режиме получить структурную модель (описание СЭ СУП и “страпов”) образца СУП из графического файла **GDSII** образца СУП. Принцип работы SIE среды аналогичен принципу работы SIV, с той лишь разницей, что в этом случае решается обратная задача, т.е. для данной конфигурации образца СУП генерируется набор ШП, который позволяет однозначно определять типы скремблинг-примитивов в данном образце СУП. На рис. 7 показана функциональная схема работы SIE среды.

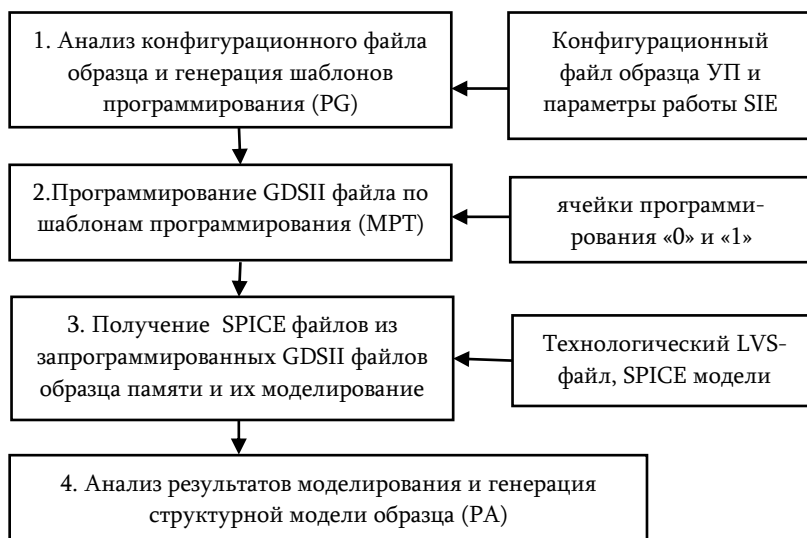


Рис. 7. Функциональная схема работы AC SIE

Pattern generator (PG) – процедура, генерирующая ШП;

Pattern analyzer (PA) – процедура, анализирующая результаты моделирования;

Memory Programming Tool (MPT) – программа, программирующая GDSII файл;

Layout vs. SPICE (LVS) – правила извлечения SPICE файла из GDSII файла

Количество ШП зависит от геометрических параметров (количества строк и колонок) строения топологии образца. Наименьшее количество ШП требуется для случая, когда в образце одинаковое число строк и колонок, т.е. матрица ячеек памяти имеет квадратную форму.

Для каждого шаблона программирования производится следующая последовательность операций: 1) программирование GDSII файла с использованием файла шаблона; 2) извлечение SPICE файла из запрограммированного GDSII файла образца; 3) моделирование SPICE файла; 4) анализ результата моделирования – получение информации о скремблинг-примитивах образца. Пункты 1 - 4 повторяются для каждого ШП. Полученный в результате работ SIE файл структурной модели образца не требует дополнительной проверки и готов к применению в прикладных программах, в том числе и в тестовых алгоритмах проверки ВУП.

Как уже отмечалось, применение небольших, но эффективных тестовых алгоритмов в ВУП способствует значительному повышению надежности ИПП, а также увеличению выхода годных микросхем.

В третьей главе описана автоматизированная программная среда *DI*, обеспечивающая: 1) внедрение различных физических и поведенческих дефектов в узлы СУП; 2) описание, исследование и отлаживание “Марч” тестовых алгоритмов, обнаруживающих встроенные дефекты в различных узлах СУП. На рис. 8 дана функциональная блок-схема программной среды *DI*.

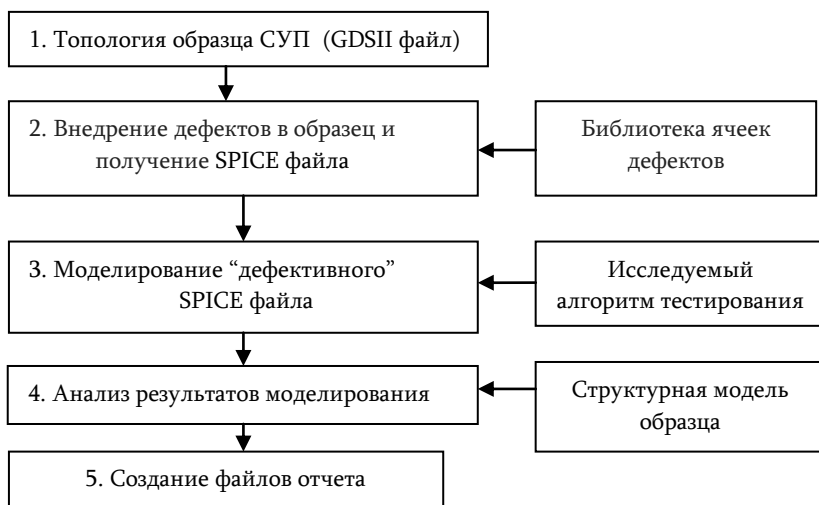


Рис. 8. Функциональная блок-схема AC DI

В программной среде *DI*, внедрение дефектов (рис. 8. (2)) может быть осуществлено тремя основными способами:

1. Ручной способ внедрения физических дефектов в графический GDSII файл образца. Данный способ может применяться для внедрения дефектов как в МЯП, так и во все остальные функциональные узлы (адрес-декодер, схемы ВХ/ВЫХ, узлы управления и синхронизации и т.д.) СУП. Ручной метод используется,

например, для внедрения различных дефектов, связанных с нарушением технологии изготовления (Process Variation defects (PV)) в СУП.

2. Автоматический способ внедрения физических дефектов в GDSII файл. Данный способ внедрения осуществляется с помощью MPT инструмента и может быть применен только для внедрения физического дефекта (или множества дефектов) в МЯП. Он позволяет произвести внедрение одновременно не более двух типов («0» и «1») физических дефектов и требует предварительного создания ячеек программирования дефектов, а также файла ШП, описывающего расположение дефектов в теле МЯП.

	11111111		00000000		xxxxx1x
	11111111		xxxxxxx		xxx1xxxx
A.	11111111	B.	00000000	B.	xxxxx1xx
	11111111		xxxxxxx		xxxx101x
	11111111		00000000		xx0xxxxx

Рис. 9. Примеры файлов ШП дефектов

На рис. 9 представлены примеры файлов ШП с дефектами для МЯП, состоящей из (физических) 5 строк и 8 колонок. В общем случае ШП позволяет описать произвольное расположение дефектов. На Рис. 9 А представлен пример файла ШП, в котором все ячейки матрицы содержат дефекты только типа «1»; на рис. 9 Б представлен пример, когда дефекты типа «0» распределены в строках 0, 2, 4 матрицы; на Рис. 9 В – пример смешанно-хаотичного расположения дефектов типов «0» и «1» в МЯП.

3. Третий способ создания дефектов позволяет внедрять и исследовать дефекты, имеющие поведенческий характер. В данном случае дефект внедряется непосредственно в тело SPICE файла или в файлы используемых SPICE моделей.

Исследуемый тестовый “Марч” алгоритм в виде векторного SPICE файла применяется в процессе моделирования. На рис. 10 показан пример файла описания тестового алгоритма.

```

Q w a n b r a          a = 0101 b = !a
P w b r b n a          stop adress 0x9
U w b d9 b r b         inst_max_adr 0xf
D r b d16 a r b w a    start_adr 0x2
D r a

```

Рис. 10. Файл описания тестируемого алгоритма в AC DI

AC DI поддерживает следующие элементы “Марч” алгоритма:

1. Типы адресаций: **U** (upward addressing) - снизу вверх; **D** (Downward addressing - reverse to U) - сверху вниз; **P** (Ping-pong addressing) -

перепрыгивающая адресация, **Q** (Reverse Ping-pong addressing) – обратная к **P** адресация.

2. Типы операций: **w** (write) – писать; **r** (read) – читать; **n** (NO) – нет операции (сигнал CLK не останавливается); **d** (delay) – задержка, содержащая N количество NO операций $d=n*N$, где N – целое число.

3. Типы данных: «a» и «b», где $b = !a$, а «a» может принимать значения {(00),(11),(01),(10)} длиной в слово СУП. Кроме описания “Марч” алгоритма, файл алгоритма содержит значения некоторых дополнительных параметров: “*start_adr*” – определяет начальный адрес алгоритма, “*stop_adress*” – конечный адрес алгоритма и “*inst_max_adr*” – максимальный допустимый адрес образца.

Полученный после обработки SPICE файл (см. Рис. 8 (2)) может содержать в себе резистивные дефекты: резистивное замыкание или резистивно-открытый дефект. Наличие этих дефектов приводит к неисправностям, обнаружение которых и является целью разрабатываемого тестового алгоритма. В процессе SPICE моделирования (рис. 8 (3)) определяется, будет ли обнаружена неисправность исследуемым тестовым алгоритмом. АС позволяет автоматически анализировать результаты моделирования (рис. 8 (4)) в заданном диапазоне сопротивления полученной резистивной неисправности и, тем самым, определять эффективность тестового алгоритма. Очевидно, что в процессе моделирования можно применять различные значения внешних параметров PVT, тем самым создавая необходимые дополнительные стрессовые режимы для исследуемого алгоритма. Результаты экспериментов, проведенных для различных типов исследуемых дефектов с использованием различных технологий изготовления СУП, показали высокую эффективность DI инструмента в деле создания ТА.

ОСНОВНЫЕ ВЫВОДЫ ПО ДИССЕРТАЦИОННОЙ РАБОТЕ

1. Исследована и представлена классификация структурных примитивов статических устройств памяти. На базе этой классификации предложен язык описания структурной модели для образцов и компиляторов СУП. Сгенерированы файлы структурной модели для более чем 800 компиляторов СУП. Разработана и внедрена методика использования структурной модели в различных программах АС СУП [5].

2. Разработана методика автоматической проверки файла структурной модели. С целью исследования строения МЯП разработана методика программирования ячеек матрицы памяти. Предложен алгоритм создания ШП, обеспечивающий достоверную проверку структурных примитивов. Создана программная среда (АС SIV) автоматизированной проверки файла структурной модели [6].

3. Разработана методика автоматического извлечения структурной модели образца из графического GDSII файла изготовления образца СУП. Предложен

алгоритм создания ШП, обеспечивающий достоверную идентификацию структурных примитивов в образце СУП. Создана программная среда (AC SIE) автоматизированного извлечения структурной модели образца из графического файла образца памяти [4,8].

4. Разработана методика внедрения физических и поведенческих дефектов в различные функциональные узлы образца СУП. Предложен алгоритм создания и проверки “Марч” тестовых алгоритмов СУП с использованием резистивных и поведенческих неисправностей. Создана программная среда (AC DI) автоматизированной проверки “Марч” тестовых алгоритмов. Исследованы Process Variation, Random Telegraph Noise, Address Decoders delay и другие типы дефектов в MOSFET и FinFET образцах СУП [1,2,3,7].

Основные результаты диссертации опубликованы в следующих работах:

1. **Aleksanyan K., Amirkhanyan K., Zorian Y., Vardanian V.**, Impact of Soft Error Challenge on SoC Design // 11th IEEE Int: On-Line Testing Symposium, Athena, Greece, July 2005. - P. 63-68.
2. **Amirkhanyan. K.** Defect injection and a memory test algorithms verification flow // Computer Science and Information Technologies, CSIT. – 2011. -P. 283-286.
3. **Amirkhanyan K., Davtyan A., Harutyunyan G., et al.** Application of defect injection flow for fault validation in memories // 10 th IEEE East-West Design & Test Symposium, EWDTS. - 2012. – P. 1-4.
4. **Ամիրխանյան Կ.** Հիշողության նմուշների կառուցվածքային մոդելի ավտոմատ դուրսբերումը գրաֆիկական GDSII ձևաչափի ֆայլից // ՀԳԱԱ և ՀՊՀՀ «ՏԵՂԵԿԱԳԻՐ» ՏԳ սերիա. – 2014. հատոր 67, N4. - էջ.- 458-466.
5. US patent 7786840. Memory Modeling Using an Intermediate Level Structural Description / **K. Aleksanyan, K. Amirkhanyan, V. Vardanian, Y. Zorian** - August 3, 2010. – 17p.
6. US patent 8112730. Various methods and apparatuses for memory modeling using a structural primitive verification for memory compilers / **K. Amirkhanyan, K. Aleksanyan, S. Karapetyan, A. Shubat, S. Shoukourian, V. Vardanian, Y. Zorian** - Feb.7, 2012. – 22 p.
7. US patent 8850277 B2. Detecting Random Telegraph noise induced failures in an electronic memory/**K. Amirkhanyan, A. Davtyan, G. Harutyunyan, T. Melkumyan, S. Shoukourian, V. Vardanian, A. Shubat, Y. Zorian** - Sep.30, 2014. – 14p.
8. US Patent application number 20130346056. Generation of Memory Structural Model Based on Memory Layout / **K. Amirkhanyan, K. Darbinyan, A. Davtyan, G.**

ԱՄՓՈՓԱԳԻՐ

Ժամանակակից կիսահաղորդչային սարքերի ընդգծված հատկություններից մեկն այն է, որ այդ սարքերը պարունակում են մեծ ծավալի ներդրված մեկ կամ ավելի, բազմակի ֆունկցիոնալ դեր կատարող հիշողության սարքեր (ՀՄ): Ներդրված հիշողության բջիջների (ՀԲ) գումարային քանակն այդ սարքերում շատ դեպքերում հասնում է հարյուրավոր միլիոնների: Ընդ որում, հիշողության բջիջների քանակը սարքերում տարեցտարի աճում է և, ըստ վիճակագրական տվյալների, 2017 թվականին ՀԲ-երը կգրադեցնեն նախագծվող սարքերի ամբողջ ֆիզիկական մակերևույթի մոտ 70 տոկոսը: Այդ պատճառով՝

ա) այդ բջիջների կառուցվածքը (layout) նախագծվում է առավելագույնս խիտ, բ) հիշողության սարքերի (ՀՄ) արտադրության օգտակար ելքի ավելացման նպատակով հիշողության սարքերում տեղադրվում են լրացուցիչ, պահեստային տողեր և սյուներ: Հետագայում այդ միավորներն օգտագործվում են արտադրության կամ էլ շահագործման ժամանակ առաջացած անսարքությունների պատճառով խափանված ՀՄ-ի աշխատանքը վերականգնելու նպատակով,

գ) հիշողության սարքերում տեղադրվում են ներկառուցված, հիշողության հետ միաժամանակ և նույն տեխնոլոգիայով պատրաստվող լրացուցիչ էլեկտրոնային սխեմաներ, որոնց նպատակն է հիշողության հանգույցների ինքնաթեստավորումը, անսարքությունների հայտնաբերումը և, հնարավորության դեպքում, հիշողության սարքի աշխատունակության վերականգնումը:

Պետք է նշել, որ ժամանակակից միկրոսխեմաներում, որոնք հանդիսանում են բարդ համակարգեր (SoC - System on Chip), լայնորեն օգտագործվում են ներդրված գործիքների հիման վրա կառուցված ներդրված թեստավորման համակարգեր (ՆԹՀ): ՆԹՀ համատարած օգտագործումը պայմանավորված է հետևյալ երեք հանգամանքներով.

1. արտաքին թեստավորման գործիքների հետ համեմատած՝ ներդրված թեստավորման գործիքներն ավելի «հզոր» են: Ներդրված թեստավորման գործիքները տարբեր պայմանների համար կարող են դիտարկել ցանկացած տեսակի և քանակի ազդանշաններ,
2. ներդրված թեստավորման գործիքների իրականացումը կատարվում է ավելի արագ: Ներդրված գործիքների դեպքում կիրառվում է «միացրու և օգտագործիր» (plug and play) մոտեցումը՝ օգտագործելով ստանդարտ JTAG (Joint Test Action Group) թեստային միացումը և արդեն մշակված և բազմակի անգամ ստուգված ծրագրային միջոցները,
3. ներդրված գործիքներն անհամեմատ էժան են:

ՀՄ-երի հետ աշխատելիս ի հայտ են գալիս ավտոմատացման մի շարք խնդիրներ, ինչպիսիք են՝ ա) հիշողության սարքի թեստավորումը, բ) հիշողության բջիջների կոորդինատների հաշվարկը, գ) նախագծի մեջ օգտագործվող հիշողության սարքերի օպտիմալ տեսակների և կոնֆիգուրացիաների որոշումը, դ) նախագծի և, մասնավորապես, հիշողության սարքի արտադրության օգտակար ելքի կանխատեսման հաշվարկը և վերլուծումը: Նշված խնդիրները ճարտարագետից պահանջում են հիշողության կառուցվածքի՝ ՀՄ-երի գոտիների բաշխման և կառուցվածքային խճողումների (scrambling), մասին տեղեկությունների տիրապետում և օգտագործում:

Ատենախոսության ընթացքում հետազոտվել են տարբեր տեխնոլոգիաների և տեսակների ստատիկ պատահական դիմումով հիշողության սարքերը (ՄՊԴՀ)՝ ՀՄ-երի կառուցվածքները, հիշողության սարքերի խճողումների առաջացման պատճառները և դրանց ազդեցությունը հիշողության սարքերի թեստավորման վրա: Դասակարգվել են կառուցվածքային խճողումների տեսակները և մշակվել են կառուցվածքային մոդելի (ԿՄ) ներկայացման ձևերը [5], որոնց հիման վրա գեներացվել են հիշողության կոմպիլյատորների կառուցվածքային մոդելները: Նկարագրվել են ՀՄ-ի արտադրության օգտակար ելքի բարձրացման ձևերը, մասնավորապես, հիշողության թեստավորման արդյունավետության ազդեցությունը ՀՄ-ի օգտակար ելքի վրա:

Հետազոտությունների ընթացքում ակնհայտ դարձավ, որ հավաստի ԿՄ ստանալու գործընթացում մարդկային գործոնը բացասական ազդեցություն ունի: Այդ պատճառով ատենախոսության աշխատանքի սահմաններում մշակվել և իրականացվել են հետևյալ ավտոմատացման համակարգեր (ԱՀ)՝

1. Structural Information Verification (SIV) [6] ԱՀ-ի նպատակն է կատարել հիշողության կոմպիլյատորի կառուցվածքային գոտիների և կառուցվածքային խճողումների նկարագրող ֆայլի ճշտության ստուգումը: Ատենախոսությունում ներկայացված են SIV ԱՀ-ի ֆունկցիոնալ բլոկ-սխեման, և հանգույցների աշխատանքի մանրամասն նկարագրությունը:

2. Structural Information Extraction (SIE) [4,8] ԱՀ-ի, որը հնարավորություն է տալիս դուրս բերել ՄՊԴՀ նմուշի կառուցվածքային մոդելը նմուշի գրաֆիկական **GDSII** ձևաչափով ներկայացված ֆայլից: Ատենախոսությունում ներկայացված են SIE ԱՀ-ի ֆունկցիոնալ բլոկ-սխեման, և հոսքի հանգույցների աշխատանքի մանրամասն նկարագրությունը, մոդելի դուրս բերման հետ կապված հիմնախնդիրները, բարդությունները և սահմանափակումները:

3. Թերությունների ներդրման և հիշողության թեստավորման ալգորիթմների ստուգման (ԹՆևՀԹԱՍ) ԱՀ-ի [2,3], որի մեջ իրականաց-

ված է ներդրված թեստային թերությունների վերլուծության վրա հիմնված մեթոդը: ԹՆևՀԹԱՍ հնարավորություն է տալիս՝

- ՀՄ-ի նմուշի բոլոր ֆունկցիոնալ հանգույցների բջիջների տոպոլոգիայում ներարկել տարբեր տեսակի թերություններ:

- Վերլուծել ներարկված իրատեսական թերությունների ազդեցությունը հիշողության ֆունկցիոնալության վրա:

- Անսարքության դիմադրության լայն տիրույթում ընկած արժեքների համար ավտոմատացված ձևով մոդելավորել հիշողության նմուշի GDSII ֆայլից ստացված SPICE ձևաչափով էլեկտրական սխեման:

- Ստուգել և ապահովել հիշողության ինքնաթեստավորման ալգորիթմի թերությունների հայտնաբերման ունակությունը:

Ատենախոսության հիմնական արդյունքներն են՝

1. Մշակվել է ՄՊԴՀ-երի կառուցվածքային մոդելը: Կատարվել է հիշողության կառուցվածքային մոդելի տարրերի դասակարգումը, մշակվել են տարրերի ներկայացման ձևերը, տարրերի ներկայացման լեզուն [5]:
2. Մշակվել և իրականացվել է ԱՀ, որը թույլ է տալիս ստուգել հիշողության կոմպիլյատորի կառուցվածքային մոդելը՝ համեմատելով այն հիշողության գրաֆիկական ներկայացման ֆայլերի՝ GDSII, կառուցվածքների հետ [6]:
3. Մշակվել և իրականացվել է ԱՀ, որը թույլ է տալիս նմուշի գրաֆիկական ներկայացման GDSII ֆայլից ստանալ հիշողության նմուշի կառուցվածքային մոդելը [4,8]:
4. Մշակվել և իրականացվել է ԱՀ, որը թույլ է տալիս հիշողության սարքերում ներդնել տարբեր տեսակի թերություններ և մշակել այդ թերությունները հայտնաբերող Մարշ տեսակի արդյունավետ թեստային ալգորիթմներ [2,3]:

Ատենախոսությունում ներկայացվել են կատարված փորձերի արդյունքներից մի քանիսը [1,2,3,7]:

KAREN SPIRIDON AMIRKHANYAN

MEMORY STRUCTURAL MODEL, SOFTWARE DEVELOPMENT AND APPLICATIONS

SUMMARY

Today an SRAM memory is an inseparable part of all types of electronic chips, and it exhibits a tendency to increase. As the statistics show, the area of the embedded SRAM in 2015 will reach over 70% of a System on Chip (SoC). The common number of bit-cells in an embedded memory in SoC is increasing and nowadays it reached hundreds millions per SoC chip. Meantime, the layout of the SRAM is becoming denser in comparison with the layout of other parts of SoC and consequently is more prone to faults. The built-in self-test (BIST) is the basic method of testing and monitoring of the embedded SRAM. Two basic parameters that characterize BIST are: 1. Layout area overhead, and 2. Coverage of the used Test Algorithm. These parameters of BIST are interrelated because a more complicated Test Algorithm requires more hardware, thus more layout area for implementation. To increase the yield of SRAM memories in the chip, some redundant rows and columns are included in the SRAM memory area. Those redundant elements are used for repairing the faulty cells of the memory area.

There are automation tasks related with the embedded SRAMS. These tasks are development of an efficient embedded test algorithm, calculation of the coordinates of an SRAM memory bit-cell, optimization of SoC SRAM types, configurations and planning, as well as yield calculation of SRAM chips. All those tasks require the correct and complete information on the structure of SRAM memories. In the scope of the dissertation the structure of more than 800 different types of compilers were investigated. In the chapter one of dissertation, the reasons of the scrambling in SRAM and its impact on SRAM applications were investigated. As a result of investigation, the classification of the memory instance structural primitives was proposed [5]. The main 16 types of the memory scrambling information were investigated and described in [5]. In the memory compilers, the memory instance graphical GDSII file contains the memory structural information, but processing of the GDSII file requires powerful computer resources and, at the same time, it is very time-consuming. In the old memory compilers (such as 130 nm, 180 nm and older), information on the memory compiler structure was presented by means of different types of tables and text descriptions which were uncomfortable for automation. To fix this issue, the *memory structural (MS) model* of SRAM memory instances and the memory compilers were developed and implemented for Synopsys memory compilers. The high level scripting language - TCL (Tool Command Language) was chosen for MS model description.

The process of MS models development and application showed that verification and validation of the structural model were very important problems. To resolve the structural model file verification problem, the Structural Information Verification (SIV) flow was developed and implemented [6]. The SIV flow allows automatically verify the memory compiler structural model file (scramble.tcl) by comparing the scramble information from the file vs. the structural information from the memory instances GDSII files. In the chapter two of the dissertation, the flowchart and the detailed description of SIV flow are presented. After the memory structural primitives are described (MS model in the memory compiler level) in a formal way the model is verified by layout (GDSII). The comparison flow assumes generation of: a set of the memory configuration, memory layout generation, pattern programming, SPICE net-list extraction, net-list simulation, vector output prediction and result comparison.

In the chapter two of the dissertation, the flowchart and the detailed description of Structural Information Extraction (SIE) flow are presented also. SIE flow allows automated extraction of the SRAM *instance* structural model from the graphical GDSII file of the instance. SIE flow assumes generation of: a set of the programming patterns, the instance GDSII file programming, SPICE netlist extraction, netlist simulation, simulation result analysis and the instance structural model generation. The problems and limitations and some practical results of the SIE flow are presented in the chapter.

In the chapter three the Defects Injection (DI) flow is presented [2,3]. The described DI flow gives us a possibility to inject different types of defects (resistive opens, resistive shorts, process variation defects and so on) in different blocks of the layout of the memory instance, then to verify if the used in the memory BIST algorithm detects a fault caused by an injected realistic defect. The flow gives us an opportunity of modeling other defects also by means of the used SPICE simulator tools. We can test the memory Test Algorithm completely or step by step for each March element separately. On the other hand, during the defect analysis we can determine: if the injected defect is realistic or not for this particular implementation of the layout library cell (memory cell, address decoder cell, sense-amplifier cell etc.), if the used Test Algorithm detects the studied (injected) defect. SRAM Address Decoder defects, Random Telegraph Noise (RTN) fault [7] and the Process Variation defects of a particular technology on the memory layout were investigated by the DI flow. A March test algorithm for detection of the mentioned defects was generated and verified. The dissertation is presented in Armenian.

The main statements of the dissertation:

- The Structural model of SRAM memories was developed and implemented.
- SIV - Structural Information Verification flow was developed and implemented.
- SIE - Structural Information Extraction flow was developed and implemented.
- DI - Defects Injection flow was developed and implemented.