



# Содержание

Введение от авторов русского перевода .....	11
---	----

## ЧАСТЬ I. ВВЕДЕНИЕ

<b>Глава 1. Воздействие радиации и отказоустойчивые методы для ПЛИС и графических процессоров</b> .....	<b>13</b>
1.1. Введение .....	13
1.2. Радиационные эффекты .....	14
1.3. Устранимые ошибки в ПЛИС .....	16
1.3.1. Сбои, вызванные одиночным воздействием на ПЛИС со статическим ОЗУ .....	17
1.3.2. Сбои, вызванные одиночным воздействием на ПЛИС на основе флэш-памяти .....	18
1.3.3. Сбои, вызванные одиночным воздействием на ПЛИС с токопроводящими перемычками .....	20
1.4. Устранимые ошибки в графических процессорах .....	21
1.5. Отказоустойчивые методы .....	21
1.5.1. Отказоустойчивые методы для ПЛИС .....	22
1.5.2. Отказоустойчивые методы для графических процессоров .....	23
1.6. Определение радиационной чувствительности ПЛИС и графических процессоров .....	24
1.6.1. Внесение ошибок .....	24
1.6.2. Методы радиационных испытаний по прогнозу и измерениям одиночных сбоев в ПЛИС и графических процессорах .....	25
Литература .....	27

## ЧАСТЬ II. ОБЛАСТИ ПРИМЕНЕНИЯ

<b>Глава 2. Бразильский наноспутник с приемником глобальной системы спутниковой навигации и функцией сопровождения на реконфигурируемой системе на кристалле</b> .....	<b>29</b>
2.1. Введение .....	30
2.2. CONASAT .....	30
2.2.1. Проект CONASAT .....	30
2.2.2. Архитектура спутников созвездия CONASAT .....	31
2.3. Архитектура программного обеспечения приемников глобальной системы спутниковой навигации (GPS) .....	32
2.4. Проектирование аппаратных средств .....	34
2.4.1. Входной каскад .....	34
2.4.2. Модуль обработки сигналов основной полосы частот .....	34
2.4.3. Модуль обработки приложений .....	36
2.4.4. Устранение SEU в имеющихся в продаже ПЛИС и системах на кристалле .....	36
2.4.5. Предлагаемая архитектура .....	37
2.4.6. Сокращение времени холодного старта .....	37
2.5. Рыночные решения .....	38
2.6. Выводы .....	39
Литература .....	39
<b>Глава 3. Обзор и исследование методов обнаружения и устранения одиночных сбоев для гетерогенных систем на основе ПЛИС</b> .....	<b>41</b>
3.1. Введение .....	42
3.2. Смягчение последствий устранимых ошибок в ASIP-процессорах .....	43

3.2.1. Методы аппаратного устранения сбоев.....	44
3.2.2. Методы устранения сбоев, основанные на программном обеспечении.....	46
3.2.3. Обсуждение.....	47
3.3. Быстрое восстановление после сбоев в конфигурационной памяти ПЛИС.....	48
3.4. Комплекс аппаратуры RUSH проекта QB50 и эксперимент.....	51
3.5. Выводы.....	53
Литература.....	54

### ЧАСТЬ III. ПЛИС НА ОСНОВЕ СТАТИЧЕСКОГО ОЗУ

<b>Глава 4. Метод внесения ошибок, ориентированный на ПЛИС со статическим ОЗУ.....</b>	<b>56</b>
4.1. Введение.....	57
4.2. Внесение сбоев в ПЛИС на основе статического ОЗУ.....	58
4.2.1. Внесение ошибок, ориентированных на регистры пользователя.....	58
4.2.2. Внесение ошибок в конфигурационные регистры.....	58
4.2.3. Сравнение статического анализа и анализа в нулевой временной координате.....	59
4.3. FT-UNSHADES2 для использования с ПЛИС.....	60
4.4. Частный случай.....	62
4.5. Выводы.....	64
Литература.....	65

<b>Глава 5. Система внесения сбоев для оценки чувствительности soft-процессора в базисе ПЛИС Virtex-5.....</b>	<b>67</b>
5.1. Введение.....	67
5.2. Сопутствующие работы.....	70
5.3. Платформа XRTC Virtex-5 (XRTC-V5FI).....	70
5.3.1. Архитектура.....	71
5.3.2. Атрибуты.....	72
5.3.3. Методология.....	73
5.4. Внесение ошибок в soft-процессор.....	74
5.4.1. Soft-процессор, использованный в экспериментах.....	75
5.4.2. Тестовые проекты для soft-процессоров.....	75
5.5. Результаты и их анализ.....	76
5.5.1. Исходная и нормированная чувствительность.....	77
5.5.2. Эксперимент «сброс – восстановление».....	78
5.6. Выводы.....	78
Литература.....	79

<b>Глава 6. Энергосберегающая адаптивная платформа FDIR, применяющая модули гетерогенных систем на кристалле.....</b>	<b>81</b>
6.1. Сопутствующие работы.....	84
6.2. Платформа FDIR, адаптируемая к вычислительной нагрузке.....	85
6.2.1. Система управления восстановлением после сбоя.....	85
6.2.2. Разнородные сигналы «Подтверждение соединения».....	86
6.2.3. Адаптивная система управления.....	87
6.3. Эталонное тестирование приложений.....	89
6.3.1. Кластеризация методом $k$ -средних.....	89
6.4. Эксперименты.....	90
6.4.1. Установка испытаний прототипа.....	91
6.4.2. Настройка эксперимента.....	91
6.5. Результаты.....	92
6.6. Выводы и перспективы.....	94
Литература.....	95

<b>Глава 7. Гибридная очистка конфигурации для ПЛИС 7-й серии компании Xilinx</b>	<b>97</b>
7.1. Введение	97
7.2. Очистка конфигурации	99
7.3. Конфигурация ПЛИС 7-й серии Xilinx	100
7.4. Архитектура гибридной очистки	102
7.5. Радиационные испытания	104
7.6. Выводы и будущее направление работы	106
Литература	107
<b>Глава 8. Анализ потребляемой мощности систем с n-модульным резервированием на основе ПЛИС со статическим ОЗУ</b>	<b>108</b>
8.1. Введение	108
8.2. Моделирование потребляемой мощности ПЛИС со статическим ОЗУ	110
8.2.1. Учет потребляемой мощности для реализации метода nMR на ПЛИС	112
8.3. Расчет потребляемой мощности на учебных примерах схем, реализованных в базе ПЛИС со статическим ОЗУ	115
8.3.1. Учебный пример № 1: MiniMIPS	115
8.3.2. Учебный пример № 2: цепочка сумматоров	119
8.4. Выводы	122
Литература	122
<b>Глава 9. Ядро управления отказоустойчивой системой для динамической частичной реконфигурации ПЛИС</b>	<b>124</b>
9.1. Введение	125
9.2. Классический метод динамической частичной конфигурации (DPR)	126
9.3. Предлагаемое ядро управления DPR	129
9.3.1. Архитектура DPRM	129
9.3.2. Отказоустойчивость DPRM	132
9.4. Установка для испытаний и результаты внесения ошибок	133
9.5. Выводы и направление будущей работы	135
Литература	135
<b>Глава 10. Наземная платформа внесения множественных сбоев для ПЛИС со статическим ОЗУ на основе наземных радиационных экспериментов</b>	<b>137</b>
10.1. Введение	137
10.2. Сопутствующая работа	138
10.3. Аппаратная реализация платформы внесения множественных сбоев	139
10.3.1. Организация конфигурационной памяти в ПЛИС Virtex-5	140
10.3.2. Метод проведения мероприятий по внесению сбоев	142
10.4. Метод для захвата и моделирования однобитовых сбоев	145
10.4.1. Моделирование с использованием базы данных предыдущих наземных радиационных экспериментов	145
10.4.2. Моделирование SEU с помощью созданных компьютером данных	147
10.5. Результаты мероприятий по внесению сбоев и их сравнение	148
10.6. Выводы	151
Литература	151
<b>ЧАСТЬ IV. ПЛИС С ФЛЭШ-ПАМЯТЬЮ</b>	
<b>Глава 11. Радиационные эффекты в 65-нм ПЛИС</b>	<b>153</b>
11.1. Введение	153
11.2. Конфигурационные ячейки флэш-памяти	154
11.3. Радиационные испытания	157

11.3.1. Радиационные испытания на воздействия суммарной дозы ионизирующего излучения	158
11.3.2. Радиационные испытания на одиночные события, вызванные ионизирующим излучением	159
11.4. Результаты радиационных испытаний на воздействия суммарной дозы ионизирующего излучения	160
11.4.1. Воздействие суммарной дозы ионизирующего излучения на ячейки флэш-памяти	160
11.4.2. Воздействие суммарной дозы ионизирующего излучения на КМОП-транзисторы	162
11.4.3. Влияние суммарной дозы ионизирующего излучения на задержку распространения	163
11.4.4. Влияние суммарной дозы ионизирующего излучения на токи источников питания в режиме ожидания	165
11.5. Результаты радиационных испытаний на одиночные события, вызванные ионизирующим излучением	166
11.5.1. Тиристорный эффект в ПЛИС	166
11.5.2. SEU для ячеек флэш-памяти	166
11.5.3. SEU для встроенного статического ОЗУ	167
11.5.4. SEU триггеров логических ресурсов ПЛИС	169
11.6. Будущая работа	171
Литература	171
<b>Глава 12. Применение метода С-кратного замедления с корректировкой временных параметров в маломощных и критических для обеспечения безопасности применениях</b>	<b>173</b>
12.1. Введение	173
12.2. Предпосылки	174
12.3. Организация главы и полученные результаты	174
12.4. Метод CSR	175
12.4.1. Теория метода CSR	175
12.4.2. CSR на RTL	175
12.4.3. Проверка изменений при проектировании CSR	176
12.5. Потребляемая мощность проектов, использующих метод CSR	176
12.5.1. Обзор	176
12.5.2. Применение в CSR обоих фронтов тактовых сигналов	177
12.5.3. R при выполнении идентичных потоков	177
12.6. Обнаружения с помощью CSR одиночных сбоев (SEU)	178
12.6.1. Обнаружение SEU стандартным CSR	178
12.6.2. Восстановление	179
12.6.3. Уменьшение сдвиговых регистров	179
12.7. Результаты	180
12.8. Краткое изложение результатов	184
Литература	184
<b>Глава 13. Усовершенствование применения функции обнаружения и исправления ошибок в радиационно-стойких ПЛИС</b>	<b>186</b>
13.1. Введение	186
13.2. Коды SEC-DED для обнаружения и устранения ошибок	188
13.3. Описание алгоритма	189
13.3.1. Цели оптимизации	189
13.3.2. Пошаговая процедура	190
13.4. Результаты	191
13.5. Выводы	195
Литература	196

<b>Глава 14. Вызванные воздействием нейтронов одиночные сбои в комбинированных аналого-цифровых ПЛИС с флэш-памятью</b>	<b>198</b>
14.1. Введение	198
14.2. Платформа аналого-цифровой СнК SmartFusion	200
14.3. Предложенный метод для исследуемого случая, использующий резервирование	201
14.3.1. Аналого-цифровая испытываемая схема с разнообразием резервирования проектов (DDR — Design Diversity Redundancy)	201
14.3.2. Комплементарные цифровые проекты	203
14.4. Установка для испытаний на воздействие нейтронов	203
14.5. Результаты испытаний	205
14.5.1. Аналого-цифровая схема с разнообразием резервирования (DDR-DMR)	205
14.5.2. Моделирование перераспределения заряда в SAR АЦП	207
14.5.3. Комплементарные цифровые проекты	210
14.6. Выводы	212
Литература	212

## ЧАСТЬ V. ВСТРОЕННЫЕ ПРОЦЕССОРЫ В СнК

<b>Глава 15. Устранение случайных ошибок во встраиваемых микропроцессорных ядрах СнК на основе ПЛИС</b>	<b>214</b>
15.1. Введение	215
15.2. Допущения	216
15.3. Модель поведенческих ошибок микропроцессорного ядра при SEE	216
15.4. Методы обнаружения ошибок	218
15.4.1. Методы повышения отказоустойчивости данных	218
15.4.2. Методы контроля потока управления	219
15.4.3. Помехоустойчивость	222
15.4.4. Гибридный метод	222
15.5. Ошибки SEE в процессорных ядрах СнК	223
15.5.1. Конструкция сторожевого устройства для СнК	225
15.5.2. Дублирование программного уровня для СнК	227
15.6. Вариант использования	228
15.7. Выводы	230
Литература	230
<b>Глава 16. Устранение случайных ошибок в soft-процессорах ПЛИС</b>	<b>233</b>
16.1. Введение	233
16.1.1. Необходимость устранения сбоев	234
16.1.2. Возможные варианты решения	234
16.2. ПЛИС в качестве технологической платформы для soft-ядер	234
16.2.1. Альтернативные варианты	235
16.3. Аппаратные методы	236
16.3.1. Защита памяти на основе резервирования информации	236
16.3.2. Защита памяти с помощью схемной логики	236
16.4. Программно-реализуемые методы	237
16.4.1. Методы защиты потока управления программы	237
16.4.2. Методы защиты данных	239
16.5. Гибридные методы	242
16.6. Выводы	247
Литература	247

<b>Глава 17. Сокращение скрытого увеличения задействованных вычислительных ресурсов программой устранения случайных ошибок с помощью селективного повышения отказоустойчивости</b>	<b>252</b>
17.1. Введение	252
17.2. Селективное программно-реализуемое повышение отказоустойчивости	254
17.2.1. Селективный метод SWIFT-R	255
17.3. Селективное аппаратно-реализуемое повышение отказоустойчивости	260
17.3.1. Селективное TMR	262
17.4. Совместное повышение отказоустойчивости: разработка метода селективного повышения отказоустойчивости с аппаратной и программной реализацией	264
17.4.1. Совместное повышение отказоустойчивости на конкретном примере: S-SWIFT-R + S-TMR	266
17.5. Выводы	269
Литература	269
<b>Глава 18. Сокращение дополнительных вычислительных ресурсов в программно-реализуемых методах обеспечения отказоустойчивости потока данных</b>	<b>272</b>
18.1. Введение	273
18.2. Программное обеспечение отказоустойчивости	274
18.2.1. Методы для потока управления	274
18.2.2. Методы для потока данных	274
18.3. Метод и его реализация	276
18.4. Результаты	280
18.5. Выводы	282
Литература	283
<b>Глава 19. Отказоустойчивые методы для процессоров с soft-ядром, использующие интерфейс трассировки</b>	<b>285</b>
19.1. Введение	285
19.2. Связанные с рассматриваемой проблемой работы	287
19.3. Интерфейс трассировки	289
19.4. Проверка выполнения	289
19.4.1. Результаты экспериментов	291
19.5. Проверка потока управления	292
19.5.1. Прогноз для счетчика команд	292
19.5.2. Проверка на основе сигнатуры	293
19.5.3. Двойной контроль потока управления	294
19.5.4. Результаты экспериментов	295
19.6. Выводы	296
Литература	297
<b>ЧАСТЬ VI. ПАРАЛЛЕЛЬНЫЕ АРХИТЕКТУРЫ И ГРАФИЧЕСКИЕ ПРОЦЕССОРЫ</b>	<b>299</b>
<b>Глава 20. Влияние случайных ошибок на графические процессоры</b>	<b>299</b>
20.1. Введение	300
20.2. Архитектура GPU и уязвимость к воздействиям радиации	301
20.3. Экспериментальная установка	304
20.4. Результаты экспериментов	306
20.4.1. Структуры основных испытаний	306
20.4.2. Динамические испытания	309
20.5. Выводы	313
Литература	314
<b>Список сокращений и аббревиатур</b>	<b>316</b>

## Введение от авторов русского перевода

Даже работающие на уровне земли заказные специализируемые большие интегральные схемы (БИС) могут подвергаться воздействиям нейтронов, которые приводят к одиночным событиям или сбоям (SEE), вызывающим отказы радиоэлектронной аппаратуры (РЭА), не говоря об условиях эксплуатации в космическом пространстве. Высокоэнергетические нейтроны появляются в результате столкновений космических галактических лучей с атомами в атмосфере и могут приводить к изменению состояния конфигурационных ячеек памяти и последовательной логики, вызывая появление случайных одиночных сбоев (SEU) или изменяя функцию схемы в комбинаторной логике, а также приводя к случайному одиночному сбою переключения (SET).

Одиночные сбои снижают надежность систем, если не принимать никаких мер для их предотвращения. В космосе, за пределами защиты магнитосферы Земли, SEU становятся крайне острой проблемой из-за высоких уровней радиации. SEU могут приводить к серьезным последствиям для космического аппарата, таким как потеря информации, сбой в функционировании или утрата контроля, например ошибочное выполнение прикладной программы.

При непрерывном снижении технологических проектных норм изготовления вызванные воздействием радиации ошибки стали главной проблемой. С уменьшением геометрических размеров транзистора появляется вероятность того, что одна падающая частица будет воздействовать не на один, а на несколько транзисторов, создавая множественные сбои ячеек в массивах памяти (MCU).

В настоящее время как зарубежные компании, так и отечественные производители выпускают радиационно-стойкие ПЛИС. Например, ПЛИС компании Xilinx серии Virtex-5QV с конфигурационными ячейками памяти на статическом ОЗУ или структурированные ПЛИС компании MicroSemi серии RT ProASIC3 с ячейками на основе флэш-памяти и ПЛИС с токопроводящими перемычками серии RTAX («антифьюз»). Радиационно-стойкие ПЛИС серии Virtex-5QV обеспечивают высокую защиту от SEU, SET, MBU, защелкиваний (SEL), выдерживают высокие дозы ионизирующего излучения (TID).

Стойкость к одиночным сбоям в радиационно-стойких ПЛИС обеспечивается специальными схемотехническими методами проектирования (RHBD) и особыми технологическими приемами (RHBP). RHBD предполагает использование тройного модульного резервирования (TMR) конфигурации и управляющей логики JTAG-интерфейса, двойных триггеров в конфигурационной памяти, двойных триггеров-защелок и SET-фильтров на всех входах триггеров конфигурируемых логических блоков, встроенных схем обнаружения и исправления ошибок в блочной памяти (ECC, SECDED). Большинство ПЛИС компании Xilinx поддерживают частичную динамическую реконфигурацию (DPR), которая выполняется во время работы системы через внутренний порт доступа к конфигурационным битам (ICAP). Динамическая реконфигурация часто используется в различных областях, таких как мультимедийные средства информации, авионика, аэрокосмическая промышленность, а также в других интеллектуальных системах для изменения их функциональности и локализации сбоев. Процесс реконфигурации – один из наиболее важных этапов исключения скопления сбоев в ячейках статического ОЗУ, которые должны быть устранены в течение определенного времени, требуемого для восстановления. Для поддержки радиационно-стойких ПЛИС компанией Xilinx разработана программно-аппаратная платформа для оценки устойчивости конфигурационной памяти ПЛИС с soft-процессорами к SEU и SET-методом внесения ошибок (Fault Injector).

Имеющиеся в продаже зарубежные COTS БИС, в т. ч. и ПЛИС, в последние годы широко используются в РЭА с длительным сроком активного существования. Использование COTS БИС позволяет создавать современное высокотехнологическое РЭА, т. к. они содержат самые инновационные решения при низкой стоимости и наличии в свободной



продаже. COTS ПЛИС предоставляют возможность разработки специализированных микропроцессорных систем в составе РЭА.

Однако COTS ПЛИС и графические процессоры специально не разработаны для обеспечения стойкости к воздействию радиации. Для проектов, реализуемых в базе COTS ПЛИС, необходимо использовать отказоустойчивые методы разработки: программно-реализуемые методы обеспечения отказоустойчивости (SWIFT), в частности селективное программно-реализуемое отказоустойчивое восстановление (S-SWIFT-R), гибридные методы с программным резервированием с той или иной аппаратной поддержкой, разнообразие резервирования проектов (DDR), коды обнаружения и исправления ошибок (ECC) и др.

Авторы перевода надеются, что сборник научных трудов под редакцией Ф. Кастеншмидт и П. Реха «ПЛИС и параллельные архитектуры для применения в аэрокосмической области. Программные ошибки и отказоустойчивое проектирование» будет полезен не только инженерно-техническим работникам, занимающимся применением серийно выпускаемых ПЛИС в авиации, космонавтике, в приборостроении для транспорта и других критических важных областях народного хозяйства, но и магистрантам, обучающимся по направлению подготовки 11.04.04 «Электроника и микроэлектроника», а также аспирантам, проходящим обучение по направлению подготовки 11.06.01 «Электроника, радиотехника и системы связи».

# Часть I. Введение

## ГЛАВА I

### ВОЗДЕЙСТВИЕ РАДИАЦИИ И ОТКАЗОУСТОЙЧИВЫЕ МЕТОДЫ ДЛЯ ПЛИС И ГРАФИЧЕСКИХ ПРОЦЕССОРОВ

*Фернанда Кастенсмиidt и Паоло Реh*

**Краткое содержание.** В этой книге приводится понятие устранимых ошибок, возникающих в ПЛИС типа ППВМ (FPGA — Field Programmable Gate Array) и графических процессорах. В главах рассматриваются радиационные эффекты в ПЛИС, отказоустойчивые методы для ПЛИС, применение серийно выпускаемых ПЛИС в авиации и космонавтике, экспериментальные данные о воздействии радиации на ПЛИС, встроенные в ПЛИС процессоры под воздействием радиации и внесение ошибок в ПЛИС. Поскольку специализированная архитектура параллельной обработки, как в случае графического процессора, стала более востребованной в авиации и космонавтике благодаря высоким вычислительным возможностям, также приводятся результаты анализа поведения графического процессора под воздействием радиации.

#### I.1. Введение

Программируемые пользователем вентиляемые матрицы (ППВМ) весьма привлекательны для авиации и космонавтики, а также для многих наземных областей применения, в которых необходим высокий уровень надежности, таких как автомобили, банковские серверы, перерабатывающие предприятия и т.п. Большой объем ресурсов, доступный в программируемых логических устройствах, может, например, увеличить возможности бортовых компьютеров на спутниках и в автомобильной промышленности. Поскольку ПЛИС может конфигурироваться/программироваться на месте эксплуатации, исправление проектных

---

F. Kastensmidt (✉) • P. Rech  
Federal University of Rio Grande do Sul, Porto Alegre, Brazil  
e-mail: fglima@inf.ufrgs.br; prech@inf.ufrgs.br

© Springer International Publishing Switzerland 2016  
F. Kastensmidt, P. Rech (eds.), *FPGAs and Parallel Architectures  
for Aerospace Applications*, DOI 10.1007/978-3-319-14352-1\_1

решений может выполняться вплоть до самого конца процесса разработки. Кроме того, новые области применения и новые функции могут быть внесены уже после запуска спутника или корректироваться при наличии электромагнитных помех. Современные ПЛИС как системы на кристалле (СнК, SoC — System-on-Chip) содержат различные программные и аппаратные процессорные ядра, встроенные цифровые сигнальные процессоры, блоки памяти и большое число конфигурируемых логических блоков, которые можно настраивать индивидуально для реализации проектных решений пользователя.

Графические процессоры (GPU — Graphics Processing Units) традиционно используются для ускорения получения графических изображений в персональных компьютерах или портативных устройствах. В мультимедийных приложениях надежность не имеет большого значения, т. к. вероятность сбоя слишком низкая, а имеющиеся ошибки вполне «терпимы», поскольку глаз человека не может их различить. Тем не менее последние графические процессоры стали применяться в тех областях, в которых надежность имеет достаточно большое значение. Благодаря их эффективности, вычислительным возможностям и низкому потреблению электроэнергии по сравнению с традиционными вычислительными процессорами, эти последние разработки графических процессоров фактически являются частью проектов в авиации и космонавтике. Возможности параллельных вычислений графических процессоров могут использоваться для сжатия изображений на спутниках, чтобы ограничить полосу пропускания, необходимую для их отправки на Землю. Кроме того, графические процессоры применяются в современных системах содействия управлению транспортным средством (ADAS — Advanced Driver Assistance Systems) и помогают водителям избегать аварийных ситуаций. И наконец, графические процессоры широко используются в качестве ускорителей в высокопроизводительных вычислительных центрах (НРС — High Performance Computing centers). Крупные НРС-центры имеют тысячи графических процессоров, которые работают параллельно, значительно повышая вероятность появления искажений по крайней мере в одном графическом процессоре под действием радиации.

К сожалению, как ПЛИС, так и графические процессоры оказались очень чувствительными к радиации, главным образом потому, что они изготавливаются по нанометровым технологическим процессам. Важно измерить экспериментально частоту появления устранимых ошибок доступных ресурсов, а также частоту появления ошибок на выходе определенных приложений, чтобы оценить, удовлетворяют ли они требованиям надежности проекта. Экспериментальное снятие характеристик используемых программируемых компонент и графического процессора является обязательным условием для подтверждения возможности их применения при кратковременных отказах. Методика испытания и снятия характеристик ПЛИС и графических процессоров при воздействии радиации необходима для надлежащего выбора и оценки отказоустойчивых методов, чтобы сделать эти компоненты более стойкими к воздействию радиации. Радиационные эксперименты, несмотря на их сложность и стоимость, являются единственным известным и сертифицированным способом для точного измерения вероятности отказа в современных интегральных схемах.

## 1.2. Радиационные эффекты

Интегральные схемы, работающие в среде, находящейся под воздействием радиации, чувствительны к кратковременным сбоям, вызванным взаимодействием ионизирующих частиц с кремнием. Частица считается ионизирующей, если она способна разделить атом на ионы. Ионизирующее излучение вызывает отказы в электронных устройствах, поскольку наведенный заряд может возмущать состояние транзистора. Этот заряд может быть наведен непосредственно (если ионизирующая частица заряжена) или опосредованно. Удары нейтронов, например, создают вторичные частицы (альфа-частицы, ионы, протоны),

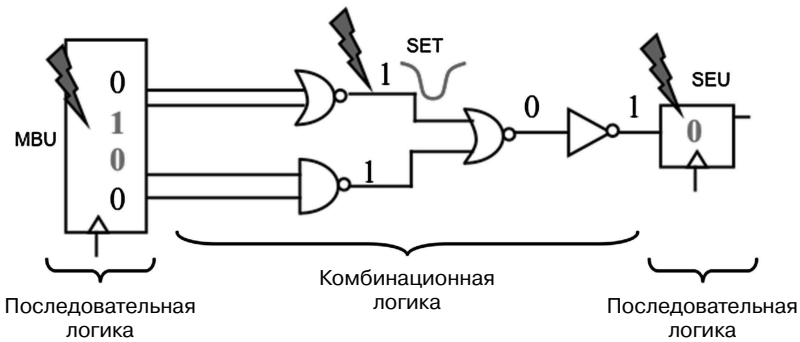


Рис. 1.1. SEU и MBU в последовательной логической схеме и SET в комбинационной логической схеме

которые имеют заряд и могут возмущать транзистор. Взаимодействие ионизирующих частиц с транзисторами может вызвать кратковременные и постоянные сбои, в зависимости от положения и количества переданного заряда (напрямую или опосредованно) в материал в результате столкновений частиц с кремнием.

Эффекты, вызванные воздействием ионизирующего излучения, носящие название одиночных событий (SEE — Single Event Effects), могут приводить к восстанавливаемым (кратковременным) или постоянным (невосстанавливаемым) отказам [1]. Когда SEE проявляется кратковременно, его принято называть устранимой ошибкой (или однократным восстанавливаемым отказом), поскольку устройство не получило необратимого повреждения. Примером устранимой ошибки является одиночный сбой ячейки памяти (SEU — Single Event Upset) и случайный одиночный сбой переключения выхода логического элемента (SET — Single Event Transient). Примером SEU является инвертирование разрядов, которые возникают, когда при ударе ионизирующих частиц по транзистору ячейки памяти наводится заряд, достаточный для изменения состояния на противоположное. Эта ячейка памяти по-прежнему работает надлежащим образом, выполняя операции чтения и записи, но хранившаяся в ней информация повреждена. При ударе ионизирующей частицы по логическому элементу создается пилообразный скачок напряжения, который при защелкивании приводит к SET. И опять же логический элемент не поврежден, поскольку новая операция в конечном счете будет выполнена правильно. Следует отметить, что понятие устранимой ошибки не уменьшает серьезности вызванных радиацией ошибок.

В противоположность этому кратковременный и стохастический характер устранимых ошибок приводит к чрезвычайной сложности их идентификации и исправления. Постоянная неисправность в ячейке памяти означает, что эту ячейку можно отметить как неиспользуемую, тогда как вероятность появления SEU делает всю матрицу ячеек памяти потенциально неисправной. Следует отметить, что с уменьшением размеров транзистора появляется вероятность того, что одна падающая частица будет воздействовать не на один, а на несколько транзисторов, создавая множественные сбои ячеек (MCU — Multiple Cell Upset) в массивах памяти. Если поврежденные биты принадлежат одному слову памяти, MCU называют многобитовыми сбоями (MBU — Multiple Bit Upset). MBU особенно критичны, поскольку они разрушают эффективность кодирования с исправлением ошибок (коды коррекции ошибок) (ECC — Error Correcting Codes). На рис. 1.1 приведены примеры единичных, многократных и случайных сбоев переключения в интегральных схемах.

Ионизирующее излучение может также вызывать постоянные отказы, такие как одиночное событие радиационного защелкивания (SEL — Single Event Latchup), одиночный пробой подзатворного диэлектрика мощного МОП-транзистора (SEGR — Single Event Gate Rupture) или единичный пробой сток/истоковых областей МОП-транзистора (SEB — Single Event Burnout). И наконец, накопление воздействий частиц приводит к суммарной

дозе ионизирующего излучения (TID — Total Ionizing Dose), которая проявляется в ухудшении рабочих характеристик транзисторов, поскольку изменяет пороговое напряжение и ток утечки.

Радиационное окружение состоит из различных частиц, генерируемых солнечной и звездной активностью [2]. Космическое пространство наполнено галактическими космическими лучами, которые являются потоком тяжелых ионов, создаваемых взрывами сверхновых звезд или столкновениями небесных тел. Высвобожденные в результате атомы, блуждающие во Вселенной, теряют протоны или электроны, тем самым приобретая заряд. В результате взаимодействия с магнитными полями планет и звезд эти ионы приобретают ускорение, достигая энергий порядка гигаэлектрон-вольт (ГэВ). Солнце создает потоки протонов и электронов, которые достигают Земли с низкими энергиями, поскольку у них не было достаточного времени для ускорения.

Эти частицы можно разделить на два основных типа: 1) частицы с высокой энергией, такие как нейтроны, электроны, протоны и тяжелые ионы, и 2) электромагнитное излучение (фотоны), к которому относятся рентгеновское излучение, гамма-лучи и ультрафиолетовое (УФ) излучение. Основными источниками частиц с высокой энергией, вносящими свой вклад в радиационные эффекты, являются протоны и электроны, захваченные поясами Ван Аллена, тяжелые ионы, захваченные в ионосфере, галактические космические лучи и солнечные вспышки. Заряженные частицы взаимодействуют с атомами кремния, вызывая возбуждение электронов и ионизацию атомов.

На уровне Земли наиболее часто вызывают сбои нейтроны. Нейтроны появляются в результате взаимодействий в верхних слоях атмосферы космических ионов с молекулами кислорода и азота. Следует отметить, что если солнечный ветер оказывается захваченным в поясах Ван Аллена из-за низкой энергии образующих его частиц, то галактические космические лучи обладают достаточно высокой энергией, чтобы пройти через эти пояса и проникнуть в верхние слои земной атмосферы. Поток нейтронов сильно зависит от таких ключевых параметров, как высота, широта и долгота. В этом потоке есть нейтроны с высокой энергией, которые взаимодействуют с материалом, создавая свободные электронно-дырочные пары, и нейтроны с низкой энергией. Эти нейтроны взаимодействуют с содержащимся в полупроводниковом материале бором, создавая другие частицы. Альфа-частицы являются вторичными частицами, которые возникают в результате взаимодействия с радиоактивными примесями, содержащимися в самом устройстве или в материале корпуса, и создают наибольшие проблемы. Выбор используемых материалов состоит в сведении к минимуму эмиссии альфа-частиц. Однако это не может устранить проблему полностью.

Поскольку высокоэнергетическая частица, проходя сквозь представляющий интерес материал, например через отрицательно смещенный *p-n*-переход, вдоль всего пути следования, теряет при этом свою энергию, что подробно описывается в [3]. Эта энергия рассчитывается как линейная передача энергии (LET — Linear Energy Transfer) и определяется как потеря энергии частицы вдоль пройденного пути, нормированная на плотность материала. Она обычно выражается в МэВ·см<sup>2</sup>/мг. Общее число зарядов пропорционально LET летающих частиц. В зависимости от способа изготовления и электрических характеристик каждого чувствительного элемента, таких как сопротивление и емкость, создаются кратковременные скачки напряжения различной амплитуды и длительности.

### 1.3. Устранимые ошибки в ПЛИС

Программируемые пользователем вентиляльные матрицы (ППВМ) являются реконфигурируемыми интегральными схемами на основе регулярных структур с высокой плотностью логических элементов, которые настраиваются конечным пользователем под различные проектные решения. Архитектура ПЛИС основана на матрице логических блоков

и межсоединениях, настраиваемых программируемыми коммутаторами. Для работы коммутаторов используются несколько различных технологий программирования. В настоящее время используются технологии программирования коммутаторов трех типов: статическое ОЗУ (оперативное запоминающее устройство), в котором роль программируемого коммутатора выполняет проходной транзистор или мультиплексор, управляемый состоянием бита статического ОЗУ (ПЛИС со статическим ОЗУ (SRAM — Static Random Access Memory)); проводящий мостик (antifuse) — электрически программируемый коммутатор образует низкоомный канал между двумя металлическими слоями (ПЛИС на основе проводящей перемычки); стираемое программируемое постоянное запоминающее устройство (СППЗУ, EPROM — Erasable Programmable Read-Only Memory) — электрически стираемое перепрограммируемое ПЗУ (ЭСППЗУ, EEPROM — Electrically Erasable Programmable Read-Only Memory) или флэш-ячейка (FLASH), когда коммутатор представляет собой транзистор с плавающим затвором, который можно отключить, подав заряд на плавающий затвор.

Настройки на основе статического ОЗУ являются энергозависимыми. Это означает, что ПЛИС на основе статического ОЗУ можно перепрограммировать на месте их работы столько раз, сколько это необходимо, и что они теряют содержащуюся в них информацию при отключении запоминающего устройства от электропитания. Решения с проводящими перемычками являются энергонезависимыми, поэтому сохраняют конфигурацию даже при отключенном электропитании и могут программироваться только один раз. Каждая ПЛИС имеет определенную архитектуру. Компании, разрабатывающие программируемую логику, такие как Xilinx, MicroSemi, Aegoflex (имеющая лицензию на ПЛИС компании Quicklogic), Atmel и Honeywell (имеющая лицензию на ПЛИС компании Atmel), предлагают стойкие к воздействию радиации семейства ПЛИС. Каждая компания применяет разные методы подавления сбоев с учетом характеристик архитектуры.

### 1.3.1. Сбои, вызванные одиночным воздействием на ПЛИС со статическим ОЗУ

ПЛИС со статическим ОЗУ состоят из матрицы конфигурируемых логических блоков (КЛБ, CLB — Configurable Logic Blocks), сложной структуры трассировочных ресурсов, матрицы встроенных элементов памяти (блоков ОЗУ, RAM — Random Access Memory), матрицы блоков цифровой обработки сигналов (ЦОС-блоки, DSP — Digital Signal Processing) и набора управляющей и контролирующей логики. Блоки КЛБ состоят из таблиц преобразований (LUT — Look-up Table), которые обеспечивают выполнение комбинационной логики, и D-триггеров, тактируемых фронтом синхросигнала (DFF — Flip-Flops), отвечающих за последовательностную логику. Структура трассировочных ресурсов может быть очень сложной и состоять из миллионов предустановленных соединений, которые могут настраиваться с помощью мультиплексоров и коммутаторов для получения требуемой маршрутизации.

Конфигурация всех КЛБ, маршрутизации, блоков ОЗУ, ЦОС-блоков и модулей ввода-вывода задается с помощью установки битов конфигурационной памяти, данный набор битов носит название битового потока или прошивки. В зависимости от размера устройства ПЛИС битовый поток может содержать миллионы бит. Ячейки памяти, которые хранят битовый поток внутри ПЛИС, образуют ячейки памяти статического ОЗУ, поэтому их можно перепрограммировать и они энергозависимы. Возникновение SEE в конфигурационной памяти ПЛИС на основе статического ОЗУ может спровоцировать инвертирование битов. Такое инвертирование битов может изменить маршрутизацию, конфигурацию LUT или триггера в КЛБ и привести к катастрофическим последствиям в проекте, поскольку SEE может изменить его функциональность.

SEE в битах конфигурационной памяти ПЛИС на основе статического ОЗУ приводит к постоянному эффекту, и его можно устранить только перепрограммированием

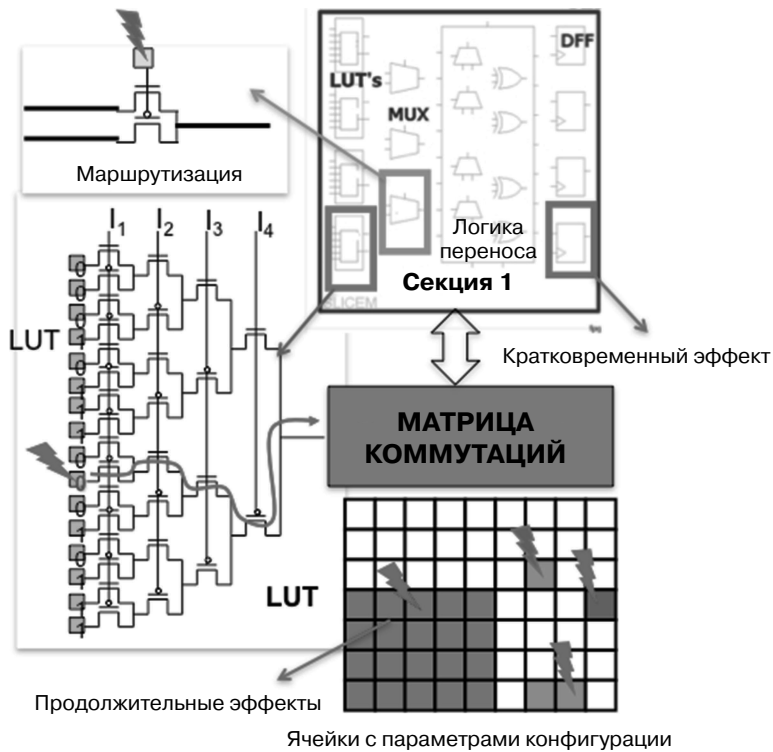


Рис. 1.2. Пример возникновения SEU в LUT и в маршрутизации ПЛИС на основе статического ОЗУ

ПЛИС [4]. В комбинационной логике последствия SEE связаны с постоянным сбоем (ноль или единица) в одной или нескольких ячейках конфигурации LUT. На рис. 1.2 можно видеть примеры возникновения SEU в конфигурационных ячейках LUT и в ячейке, контролирующей соединения маршрутизации. SEE в архитектуре маршрутизации может подключать или отключать межсоединение в матрице. Это также относится к постоянному эффекту. Результатом такого воздействия может быть изменение электрической схемы, что приводит к изменению логики работы схемы, или появление короткого замыкания в комбинационной логике, интерпретируемой ПЛИС. Может пройти большое количество тактов синхросигнала, прежде чем постоянная ошибка такого рода будет выявлена и запущатся необходимые восстановительные процедуры по загрузке неповрежденного битового потока. За это время ошибка может распространяться в оставшуюся часть проекта.

Инвертирование битов также может происходить в триггерах КЛБ, используемых для реализации последовательностной логики пользователя. В этом случае инвертирование бита имеет кратковременный эффект и последующая загрузка триггера устранил ее.

### 1.3.2. Сбои, вызванные одиночным воздействием на ПЛИС на основе флэш-памяти

ПЛИС на основе флэш-памяти имеют перестраиваемую конфигурацию матрицы, состоящую из логических ячеек ПЛИС (VersaTiles) и трассировочных ресурсов, которые программируются включением или отключением коммутаторов (конфигурационных

ключей), построенных на базе транзисторов с плавающим затвором (FG — Floating Gate) ( $n$ -канальный МОП-транзистор с многоуровневым затвором) [5]. Цепь такого коммутатора состоит из двух  $n$ -канальных МОП-транзисторов: 1) считывающего транзистора для программирования плавающего затвора и считывания тока во время измерения порогового напряжения и 2) переключающего транзистора для включения и отключения линии передачи данных в ПЛИС (рис. 1.3). Эти два транзистора совместно используют одни и те же управляющий и плавающий затворы. Пороговое напряжение определяется накопленным зарядом в плавающем затворе. Рис. 1.3 иллюстрирует логические ячейки ПЛИС (VersaTiles), используемые для построения типовых логических элементов. VersaTiles подключаются по четырехуровневой иерархии трассировочных ресурсов: сверхбыстрые локальные ресурсы, эффективные ресурсы маршрутизации для длинных линий, высокоскоростные ресурсы маршрутизации для очень длинных линий, высокопроизводительные сети VersaNet.

Каждая ячейка VersaTile может реализовать любую трехходовую логическую функцию, что эквивалентно трехходовой таблице преобразования (3-LUT). Но важно заметить, что электрическая реализация VersaTile полностью отличается от электрической реализации таблицы преобразования (LUT — Lookup Table). Поэтому VersaTiles при

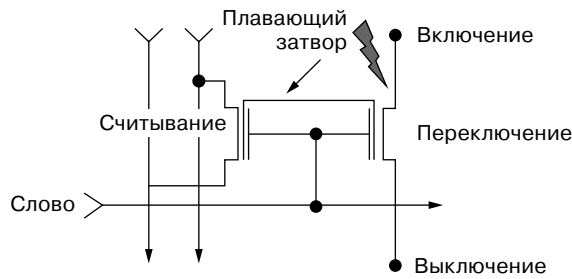


Рис. 1.3. SET в программируемом коммутаторе ПЛИС на основе флэш-памяти

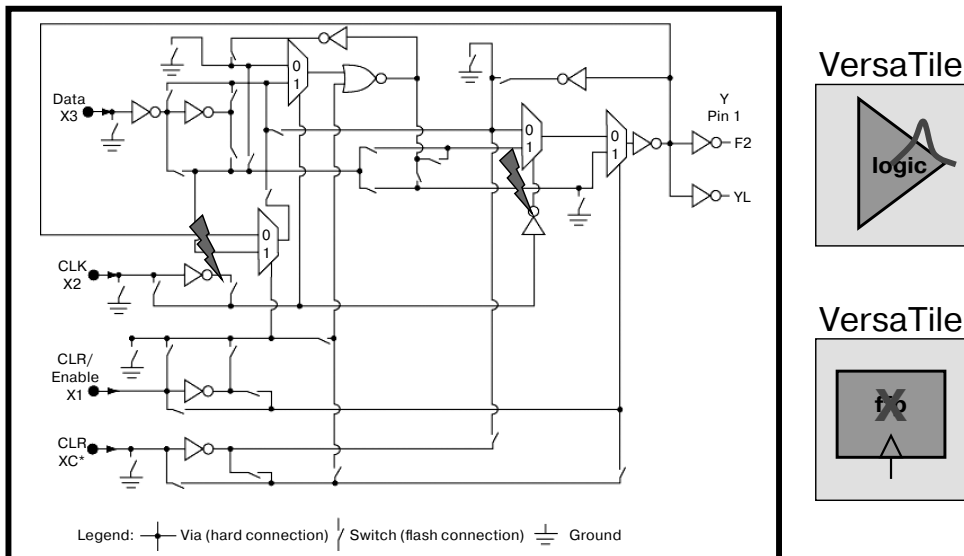


Рис. 1.4. SET и SEU в ячейках VersaTile ПЛИС на основе флэш-памяти



различных воздействия могут иметь отличное электрическое поведение по сравнению с трехходовыми LUT. VersaTiles также могут реализовать триггер-защелку с очисткой и сбросом или D-триггер с очисткой или сбросом или включить D-триггер с очисткой и сбросом посредством транзисторной логики и линий обратной связи внутри логической ячейки VersaTile. Для каждой конфигурации ячейки VersaTile число переключателей плавающего затвора и транзисторов в критическом пути меняется. Случайные сбои переключения (SET — Single Event Transient) могут воздействовать на область стока транзистора в отключенном состоянии, как это представлено на рис. 1.3, вызывая кратковременный импульс в конфигурационных коммутаторах, или воздействовать на чувствительные области транзисторов в ячейках VersaTile, создавая импульсные помехи или инвертирование битов в зависимости от конфигурации данной ячейки (рис. 1.4). В главе 11 подробно рассматривается оценка ошибок, вызванных радиацией, в ПЛИС, созданных по технологии 65 нм на основе флэш-памяти. В главе 14 приводится обзор влияния воздействия нейтронов на ПЛИС по флэш-технологии, предназначенной для аналого-цифровой обработки сигналов.

### 1.3.3. Сбои, вызванные одиночным воздействием на ПЛИС с токопроводящими перемычками

ПЛИС с токопроводящими перемычками основаны на постоянной матрице, состоящей из комбинационных С-ячеек и последовательностных R-ячеек, окруженных регулярными трассировочными каналами. Все варианты трассировки и конфигурации С- и R-ячеек осуществляются расплавлением необходимых токопроводящих перемычек (программируемых переключателей). Результаты наземных радиационных испытаний показали, что программируемые переключатели на основе технологий ONO (Oxide — Nitride — Oxide, оксид — нитрид — оксид) и МІМ (Metal — Insulator — Metal, металл — изолятор — металл) являются стойкими к воздействию ионизации и накопленной дозы облучения [6]. Поэтому настраиваемая пользователем трассировка не чувствительна к SEU, только комбинационная логика и триггеры, используемые для описания пользовательской последовательностной логики, оказываются чувствительными к SEE.

Другим хорошо известным вариантом ПЛИС на основе расплавляемых токопроводящих перемычек являются ППВМ компаний Aeroflex и QuickLogic. Их архитектура составлена из матрицы конфигурируемых логических ячеек, используемых для реализации комбинационной логики, и триггеров, которые окружены трассировочной структурой. Программируемые переключатели, получившие название ViaLink, применяются для настройки всех вариантов пользовательских конфигураций.

Чтобы подытожить воздействия SEU и SET на ПЛИС, в табл. 1.1 показаны восприимчивые части архитектуры и разделение сбоев на кратковременные и продолжительные, когда для устранения сбоя требуется реконфигурация.

Таблица 1.1. Последствия воздействий SEU и SET в ПЛИС

ПЛИС	SEU/SET в логической схеме основного блока конфигурации	Трассировочные соединения	Конфигурируемые коммутаторы
На основе статического ОЗУ	Продолжительный	Продолжительный	Продолжительный
На основе флэш-памяти	Кратковременный	Нет	Нет
На основе проводящей перемычки	Кратковременный	Нет	Нет

## 1.4. Устранимые ошибки в графических процессорах

Модули графических процессоров представляют собой сложные системы параллельных вычислений, которые используют большие структуры памяти, такие как кэш второго и первого уровней (L2 и L1) или регистровые файлы, высокоэффективные арифметическо-логические устройства (АЛУ, ALU — Arithmetic Logic Units), а также планировщики и диспетчеры задач.

Радиация может вызывать в структурах памяти графического процессора как одиночные обратимые сбои, так и множественные сбои. Если радиация повреждает регистр, то процесс, использующий этот регистр для вычислений, вероятнее всего, выдаст неверный результат. Специфика параллельных вычислений приводит к тому, что ошибки в кэш-памяти графического процессора более существенны, чем в традиционных ЦПУ. Фактически кэш-память первого уровня (L1) используется совместно всеми параллельными процессами вычислений в потоковом мультипроцессоре (SM — Streaming Multiprocessor), тогда как кэш L2 распределяется между всеми SM. Поэтому ошибка в кэше L1 может, в наиболее неблагоприятном случае, распространиться на все выполняемые текущим SM параллельные процессы. Аналогичным образом ошибка в кэше L2 может повлиять на абсолютно все процессы, выполняемые графическим процессором [7].

Соударение падающей частицы с вентилем может привести к случайному сбою переключения. Что касается SEU, то его серьезность и общий суммарный эффект SET зависят от области соударения. Если SET воздействует на вентиль внутри отдельного ядра, то поток операций, выполняемых этим ядром, вероятно, даст на выходе единственный сбой. Однако если SET приводит к сбою планировщика или диспетчера параллельных вычислений, это может повлиять на вычисления нескольких процессов, а также привести к аварийному завершению приложения или зависанию системы [8].

Чтобы иметь всестороннюю оценку чувствительности графического процессора, недостаточно измерить чувствительность к воздействию радиации отдельных ресурсов, таких как модули памяти или логические вентили. Необходимо также проанализировать, как эти ресурсы используются в вычислениях. Для этого можно провести радиационные эксперименты на репрезентативном наборе приложений, чтобы иметь достаточно данных для обобщения на остальные алгоритмы. Альтернативно этому можно вычислить коэффициент уязвимости архитектуры (AVF — Architectural Vulnerability Factor), т.е. вероятность выдачи поврежденным ресурсом ошибки в выходных данных, как в [9]. В главе 20 подробно рассматриваются возможные радиационные воздействия на графические процессоры и показывается возможный путь оценки поведения графического процессора под воздействием радиации.

## 1.5. Отказоустойчивые методы

Отказоустойчивость определяется как набор методов по обеспечению рабочего режима, при котором выполняется системная функция, несмотря на наличие (ограниченного набора) сбоев. Проблема отказоустойчивости полупроводниковых устройств обрела важность после того, как несколько лет назад начали возникать сбои космической аппаратуры. С тех пор интерес к изучению отказоустойчивых методов для поддержания работоспособности интегральных схем (ИС, IC — Integrated Circuits) в таких неблагоприятных условиях окружающей среды стал расти, ведомый всеми возможными областями применения стойких к воздействию радиации ИС, как, например, полеты в космосе, спутниковые системы, высокоэнергетические эксперименты в физике и т.п. Системы космических кораблей

включают множество разнообразных аналоговых и цифровых компонентов, которые потенциально чувствительны к воздействию радиации, и поэтому для обеспечения их надежности необходимо использовать отказоустойчивые методы.

### 1.5.1. Отказоустойчивые методы для ПЛИС

Различные отказоустойчивые методы применимы для ПЛИС в соответствии с типом технологии конфигурирования, архитектурой и заданной средой эксплуатации. Эти методы могут быть реализованы пользователем с применением высокоуровневого языка описания аппаратных средств (HDL — Hardware Description Language), прежде чем проект будет синтезирован в ПЛИС. В этой книге авторы преимущественно рассматривают методы, которые могут применяться самим пользователем при проектировании с помощью языка HDL.

Главные методы основаны на одном из двух принципов: пространственной или временной избыточности (резервировании) [10]. Пространственная избыточность сводится к  $n$ -кратному дублированию конкретного модуля  $n$  идентичными модулями, все выходные сигналы которых направляются в мажоритарную схему голосования. Обычно  $n$  является нечетным числом. Схема голосования определяет правильный выходной сигнал по большинству равных выходных сигналов. Наиболее распространенным является вариант  $n$ -модульного резервирования (nMR — n-modular redundancy) с  $n$ , равным 3, получивший название тройного модульного резервирования (TMR — Triple Modular Redundancy). В этом случае схема голосования позволяет выбрать два одинаковых значения из трех, которые не содержат ошибок. TMR может использоваться различным образом: с крупноблочной реализацией или разбиением на небольшие блоки, с добавлением дополнительных схем мажоритарного голосования. Существует локальный вариант TMR, когда три раза дублируются лишь триггеры, и глобальный TMR, также известный как XTMR, когда три раза дублируется вся комбинационная и последовательностная логика. Возможно и использование разнообразного тройного модульного резервирования Diverse TMR, в этом случае каждый дублирующий модуль имеет архитектуру, отличную от остальных.

При разработке маршрутизации можно выбирать различные методы для увеличения или уменьшения коэффициента разветвления, времени задержки и числа соединений, которые могут иметь различное влияние на чувствительность к SEE. В дополнение для ПЛИС со статическим ОЗУ реконфигурация является необходимым средством устранения сбоев в прошивке. Полное непрерывное восстановление конфигурации вслепую носит название очистки (scrubbing), которая отвечает за полное восстановление конфигурации ПЛИС, используя эталонный битовый поток (golden bitstream). Также можно использовать частичное реконфигурирование.

Для встроенных процессоров применяются разные методы смягчения чувствительности к сбоям, основанные на программной или процессорной избыточности, как, например, жесткая конфигурация (lock-step) и повторные вычисления. Программные методы обеспечения отказоустойчивости используют информационную избыточность, анализ потока команд управления и сравнения для обнаружения ошибок во время выполнения программы. Для этой цели в программных методах применяют дополнительные инструкции в коде программы либо для повторного выполнения команд, либо для сохранения и сверки подходящей информации в элементах памяти. Недавно было реализовано программное средство, которое позволяет автоматически вставлять такие инструкции в код C или в ассемблер, значительно сокращая затраты на повышение отказоустойчивости.

При временной избыточности в течение определенного времени два или три раза захватывается одна и та же величина, чтобы путем голосования избежать кратковременного сбоя. Полученные значения сдвигаются на время задержки [11]. Смысл этого подхода состоит в том, чтобы определить два не содержащих сбоя значения из трех для игнорирования ошибки.

**Таблица 1.2.** Перечень методов устранения, которые могут применяться пользователем в проектных решениях с ПЛИС

Метод устранения	Уровень абстрагирования	SET	SEU
Локальное TMR	HDL		X
Глобальное TMR или XTMR	HDL	X	X
Крупноблочное TMR	HDL	X	X
Разнообразное TMR (DTMR)	HDL	X	X
Включение схемы голосования	HDL	X	X
Размещение и трассировка с учетом надежности	Маршрут проектирования ПЛИС	X	X
Временная избыточность	HDL	X	
Избыточность встроенного процессора	HDL/на базе ПО	X	X
Очистка/частичная реконфигурация	Система		X

Каждый из этих методов позволяет защитить от SEU или SET либо от того и другого, как это показано в табл. 1.2, и они будут рассмотрены в отдельных главах этой книги.

Очень часто системы на кристалле (СнК), реализованные в ПЛИС, используют набор уже упоминавшихся методов устранения сбоев. В главах 2 и 3 представлены проекты на основе ПЛИС как СнК со статическим ОЗУ со встроенным процессорным ядром для спутниковых приложений, в которых используются некоторые отказоустойчивые методы.

В главе 6 подробно описывается обнаружение отказов, их изоляция и восстановление структуры с использованием преимуществ, предоставляемых ресурсами, доступными в гетерогенных системах. В главе 7 предлагается новый метод очистки для конфигурационных ячеек памяти ПЛИС. В главе 8 оцениваются требования к электропитанию для п-модульной избыточности, а в главе 9 представлено отказоустойчивое ядро управления для динамической частичной реконфигурации ПЛИС. В главе 12 предлагается использовать метод C-Slow с восстановлением синхронизации для обеспечения безопасности критически важных приложений. В главе 13 предлагается более эффективное исполнение функции обнаружения и исправления ошибок в стойких к воздействию радиации ПЛИС. В главе 15 приводятся методы повышения отказоустойчивости для встроенных процессоров, а в главах 16 и 19 — метод повышения отказоустойчивости для soft-процессоров. В главах 17 и 18 изучается, как уменьшить дополнительные затраты ресурсов для распространенных методов повышения отказоустойчивости электрических схем и процессоров.

### 1.5.2. Отказоустойчивые методы для графических процессоров

Поскольку графические процессоры изначально предназначались для ускорения графической визуализации, исследования их надежности находятся в зачаточном состоянии. Большинство из доступных графических процессоров не предполагают каких-либо вариантов обеспечения надежности, отдавая предпочтение производительности, а не отказоустойчивости. И только недавно в графические процессоры, производимые для рынка высокопроизводительных средств вычисления, стали включать коды исправления ошибок в основных структурах памяти (в L1 и L2 кэш-памяти и во внутренних регистрах памяти). Имеющаяся в наличии система ECC исправляет одиночные и обнаруживает двойные ошибки (SEDED — Single Error Correction Double Error Detection). С ее помощью можно исправлять SEU и обнаруживать MBU. Экспериментально было установлено, что около 30 % вызванных воздействием радиации сбоев в структурах памяти современных графических процессоров относятся к категории множественных сбоев. Благодаря использованию

чередования памяти (т. е. логические биты, принадлежащие одному и тому же слову, физически разделены) только 5% ошибок являются множественными, воздействующими на биты одного и того же слова. Кроме того, MBU с более чем двумя искаженными битами экспериментально никогда не наблюдались. Таким образом, SECDED ECC представляется достаточной для того, чтобы гарантировать высокую надежность. Несмотря на это, логические ресурсы вычислительных структур и планировщиков заданий остаются без защиты, таким образом, внутренние триггеры и очереди не охвачены системой ECC. Поэтому ECC может не гарантировать высокий уровень надежности [12].

Недавно были предложены некоторые программные методы повышения отказоустойчивости для параллельных кодов. Основная идея заключается в дублировании параллельных задач для идентификации сбоев или в добавлении процедур кодирования и декодирования для обнаружения и в конечном счете устранения сбоев. Дублирование со сравнением (DWC — Duplication With Comparison) очень легко описывается в графическом процессоре, поскольку основной принцип программирования этого устройства — это параллелизм [12]. Если даже DWC и представляется обещающим и эффективным для обнаружения ошибок методом, он вносит отнюдь не пренебрежимо малые вычислительные издержки. Это приводит к тому, что избыточность может оказаться неприемлемой для высокопроизводительных вычислительных систем или встроенных систем со строгими ограничениями энергопотребления. Важно также вдумчиво дублировать параллельные процессы, избегая выполнения потоков, принадлежащих одному и тому же домену, на одном потоковом мультипроцессоре, т. к. эти потоки будут совместно использовать одну и ту же кэш-память. Ошибка в совместно используемой области затем распространится в обе копии, оставаясь при этом необнаруженной. Другим принципом обеспечения отказоустойчивости, применяемым в параллельных кодах, является алгоритмическая отказоустойчивость (ABFT — Algorithm Based Fault Tolerance). ABFT основана на кодировании входных данных, модификации алгоритма для работы с закодированными данными и, наконец, декодировании выходных данных с обнаружением и исправлением ошибок. ABFT алгоритмозависима и требует подробного анализа алгоритма и серьезного усложнения исходного кода для реализации. На данный момент единственными алгоритмами, для которых возможно применение принципа ABFT, являются алгоритмы перемножения матриц и быстрое преобразование Фурье [7, 13]. В главе 20 приводится обзор принципов обеспечения отказоустойчивости, применяемых для работы современных параллельных процессоров.

## 1.6. Определение радиационной чувствительности ПЛИС и графических процессоров

### 1.6.1. Внесение ошибок

При работе с ПЛИС важным этапом разработки является проверка работы методов отказоустойчивости, которая обычно осуществляется внесением ошибок. Первоначальный битовый поток, используемый для конфигурации ПЛИС, может быть изменен с помощью электрической схемы или инструментов в ПК путем инвертирования разрядов битового потока по одному за раз. Это инвертирование имитирует SEU в ячейках конфигурационной памяти. Результаты испытания схемы можно непрерывно контролировать (DUT — Design Under Test) для анализа влияния внесенной неисправности на ее работу. Если была обнаружена ошибка, это означает, что используемый метод обеспечения отказоустойчивости не надежен для устранения этого конкретного сбоя (SEU) в заданном конфигурационном бите памяти.

Можно вносить неисправности во все конфигурационные биты для анализа наиболее критичных блоков проекта [14]. Это может помочь конструкторам на ранних этапах разработки выбрать наиболее подходящий принцип отказоустойчивости еще до проведения наземных радиационных испытаний. Весь процесс внесения ошибок может занять от нескольких часов до нескольких дней, в зависимости от количества инвертируемых битов и подключения к схеме управления внесением неисправностей. Когда вся система (управление внесением неисправностей + испытуемая система + эталонная прошивка) реализована на аппаратном уровне (на плате), без связи с ПК, этот процесс ускоряется на порядки.

В главах 4 и 5 представлены некоторые методы для внесения ошибок в ПЛИС со статическим ОЗУ. В главе 10 представлена платформа внесения ошибок, которая воспроизводит множественные сбои и накопление сбоев на основе реальных радиационных экспериментов.

Однако процесс внесения ошибок в графических процессорах имеет некоторые ограничения. Только малая часть ресурсов графического процессора доступна пользователю, и для того, чтобы получить к ним доступ в целях внесения неисправностей, необходимо изменить ход выполнения алгоритма, что влечет за собой элементы неестественного поведения. Для графического процессора существует одна утилита внесения неисправностей — GPU-Qin [15], которая позволяет пользователю вносить неисправности только в конкретные значения.

## 1.6.2. Методы радиационных испытаний по прогнозу и измерениям одиночных сбоев в ПЛИС и графических процессорах

Тестирование ПЛИС под воздействием радиации зависит от протокола испытаний, разработанного для каждого типа ПЛИС и архитектуры проекта. В этом разделе подробно рассмотрим радиационные испытания ПЛИС со статическим ОЗУ. Существует два типа испытаний: статическое и динамическое. Статическое испытание можно проводить, например, на ПЛИС со статическим ОЗУ, тогда испытание будет состоять из следующих шагов: конфигурирование ПЛИС *эталонным* битовым потоком (прошивкой), содержащим тестируемый проект, а затем непрерывное считывание данных из конфигурационной памяти ПЛИС с помощью инструмента iMPACT компании Xilinx через интерфейс JTAG. В этом эксперименте управляющий ПК сравнивает *эталонный* битовый поток со *считанным* битовым потоком. При обнаружении разницы конфигурация ПЛИС восстанавливается с помощью *эталонного* битового потока, а различия сохраняются в компьютере. Сбои определяются как любое инвертирование бита в конфигурационной памяти, обнаруженное процедурой *считывания*. В этом случае можно рассчитать частоту сбоев в битах конфигурационной памяти для заданного потока.

Сечение взаимодействия в расчете на бит показывает чувствительные участки устройства и используется для сравнения радиационной чувствительности между различными устройствами. Рассчитывается согласно формуле (1.1)

$$\sigma_{\text{SEU-бит}} = \frac{N_{\text{SEU}}}{\Phi_{\text{нейтрон}} \cdot N_{\text{бит}}}, \quad (1.1)$$

где  $N_{\text{SEU}}$  — число SEU в битах конфигурационной памяти,  $\Phi_{\text{нейтрон}}$  — интегральный поток нейтронов, а  $N_{\text{бит}}$  — общее число битов в устройстве. Интегральный поток измеряется в нейтронах на  $\text{см}^2$  и получается умножением потока нейтронов на время нахождения устройства в этом потоке.

Динамическое испытание направлено на анализ выводов реализованного проекта в ПЛИС. В этом случае ожидаемая частота ошибок намного ниже, чем при статических испытаниях. Для ПЛИС со статическим ОЗУ, по данным доклада о надежности компании

Xilinx [16], чтобы вызвать одну ошибку на выходе испытуемого проекта, в среднем необходимо спровоцировать 20 сбоев в битах конфигурационной памяти. Это соотношение, конечно же, может варьироваться в зависимости от плотности логических элементов, размещения, трассировки и выбранной архитектуры проекта. При использовании избыточности, такой как TMR или n-MR, число скопившихся сбоев в битовом потоке, не приводящих к функциональной ошибке, может значительно возрасти. Для вариантов ПЛИС с флэш-памятью и с расплавляемыми токопроводящими перемычками частота устранимых ошибок связана только с восприимчивостью конфигурируемой логики к SET и SEU (инвертированию разрядов), поскольку программируемые ячейки памяти (ПЛИС с флэш-памятью и токопроводящими перемычками) не восприимчивы к кратковременным сбоям.

Статические испытания графических процессоров выполняются по тому же принципу, что и ПЛИС. По сути, известная модель загружается в структуры основной памяти устройства и затем считывается. Для доступа к структурам памяти нет специального порта, поэтому испытания должны быть спроектированы так, чтобы использовать все преимущества обычных ресурсов графического процессора для записи, а затем считывания шаблона конфигурации. Динамические испытания графического процессора требуют выбора подходящих эталонных тестов для запуска на устройстве. Следует заметить, что выбранные эталонные тесты должны соответствовать рабочей нагрузке реального применения. Иначе результаты испытаний будут действительны только для тестируемой конфигурации. Обычно эталонные тесты выполняются с предварительно выбранными входными данными, а результаты сверяются с предварительно вычисленной копией. При обнаружении расхождений их следует считать ошибкой. Для оценки сечения взаимодействия необходимо оценить поток частиц, падающий на устройство, только в процессе выполнения программы, исключая время проверки результатов. Альтернативно этому можно рассчитать сечение взаимодействия, разделив наблюдаемую частоту появления ошибок (ошибки/с) на средний поток, используемый установкой при испытаниях (частицы/(см<sup>2</sup> с)).

Во всем мире есть только несколько установок, которые обеспечивают подходящие потоки и спектр энергий, чтобы согласовать шкалу экспериментальных результатов с ожидаемой частотой ошибок в реальных условиях. Примерами таких установок являются LANSCE в Лос-Аламосе, NM в США, TSL в Уппсале (Швеция), TRIUMF в Ванкувере (Канада) и ISIS в Дидкоте (Великобритания).

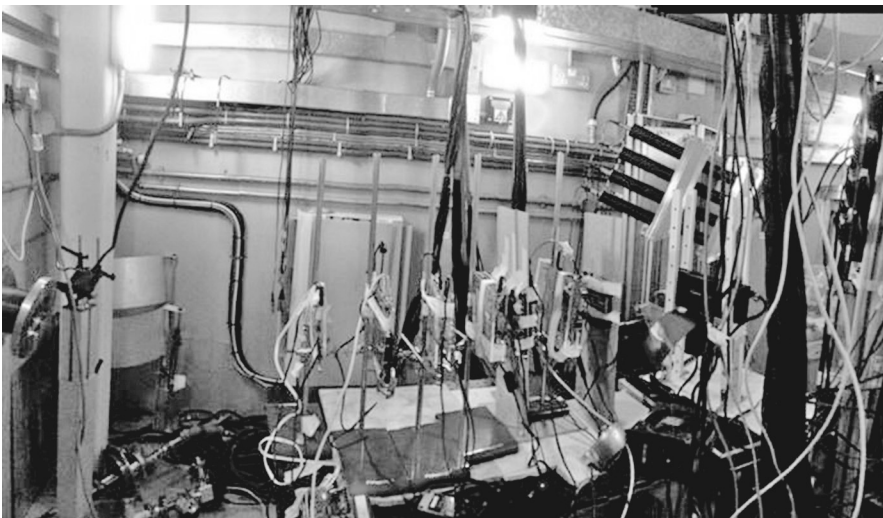
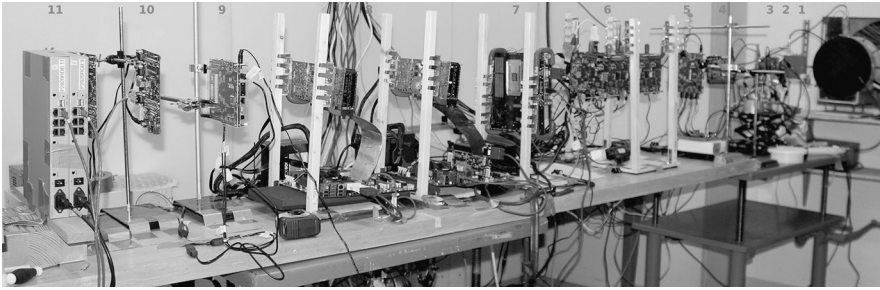


Рис. 1.5. Нейтронная экспериментальная установка в ISIS для ПЛИС и графических процессоров



**Рис. 1.6.** Нейтронная экспериментальная установка в Лос-Аламосе для ПЛИС и графических процессоров

В этой книге используются результаты экспериментов, проведенных в Национальной лаборатории (LANL — Los Alamos National Laboratory's) «Лос-Аламосский центр по изучению физики нейтронов» (LANSCE — Los Alamos Neutron Science Center) по облучению интегральных схем и корпусов электроники, а также в Резерфордской лаборатории ISIS в Дидкоте (Великобритания), использующих плоский пучок нейтронов. Как показано в [17], обе эти установки обеспечивают источник нейтронов с белым спектром, который имитирует спектр энергии потока нейтронов в атмосфере. Спектр ISIS имеет более низкий уровень высокоэнергетических нейтронов по сравнению с LANSCE и спектром в атмосфере Земли. Связь между энергией нейтронов и сечением взаимодействия современных устройств все еще остается нерешенной проблемой. Тем не менее пучок нейтронов ISIS, как удалось установить экспериментально, оказался подходящим для имитации пучка LANSCE и радиационной среды земной атмосферы [17].

На рис. 1.5 и 1.6 показаны схемы экспериментов по изучению воздействий нейтронов в научном центре ISIS в Великобритании и в Лос-Аламосе соответственно, состоящие из многих типов ПЛИС и графических процессоров, работающих параллельно.

## Литература

1. Nicolaidis M. (2011). *Soft errors in modern electronic systems*. Springer, New York, p. 318.
2. Stassinopolous E. G., Raymond J. P. (1988). The space radiation environment for electronics. *Proc IEEE* 76:1423–1442.
3. Dodd P. E., Massengill L. W. (2003). Basic mechanisms and modeling of single-event upset in digital microelectronics. *IEEE Trans. Nucl. Sci.* 50(3):583–602.
4. Kastensmidt F. L., Reis R., Carro L. (2006). *Fault-tolerance techniques for SRAM-based FPGAs (frontiers in electronic testing)*. Springer, New York.
5. Microsemi. ProASIC3, IGLOO and SmartFusion flash family FPGAs datasheet. [www.microsemi.com](http://www.microsemi.com).
6. Rezgui S., Louris P., Sharmin R. (2010). SEE characterization of the new RTAX-DSP (RTAX-D) antifuse-based FPGA. *IEEE Trans. Nucl. Sci.* 57(6):3537–3546.
7. Rech P., Aguiar C., Frost C., Carro L. (2013). An efficient and experimentally tuned software-based hardening strategy for matrix multiplication on GPUs. *IEEE Trans. Nucl. Sci.* 60(4):2797–2804.
8. Rech P., Pilla L., Navaux P. O. A., Carro L. (2014). Impact of GPU's parallelism management on safety-critical and HPC applications reliability. В: *Proceeding IEEE international conference on dependable systems and networks (DSN)*, June 2014, pp. 455–466.



9. Mukherjee S. S., Emer J., Reinhardt S. K. (2005). The soft error problem: an architectural perspective. В: High-performance computer architecture, 2005. HPCA-11. 11th international symposium on, 12–16 Feb. 2005, pp. 243–247.
10. Schrimpf R. D., Fleetwood D. M. (2004). Radiation effects and soft errors in integrated circuits and electronic devices. Word Scientific, Singapore.
11. Anghel L., Alexandrescu D., Nicolaidis M. (2000). Evaluation of a soft error tolerance technique based on time and/or space redundancy. In: The Proceedings of symposium on integrated circuits and systems design, SBCCI, 13, pp. 237–242.
12. Oliveira D. A. G., Rech P., Pilla L. L., Navaux P. O. A., Carro L. (2014). GPGPU ECC efficiency and efficacy. In: International symposium on defect and fault tolerance in VLSI and nanotechnology systems.
13. Pilla L. L., Rech P., Silvestri F., Frost C., Navaux P. O. A., Sonza Reorda M., Carro L. (2014). Software-based hardening strategies for neutron sensitive FFT algorithms on GPUs. IEEE Trans. Nucl. Sci. 61(4):1874–1880.
14. Sterpone L., Violante M. (2007). A new partial reconfiguration-based fault-injection system to evaluate SEU effects in SRAM-based FPGAs. IEEE Trans. Nucl. Sci. 54(4):965–970.
15. Fang B., Pattabiraman K., Ripeanu M., Gurumurthi S. (2014). GPU-Qin: A methodology for evaluating the error resilience of GPGPU applications. In: Proceedings of the IEEE international symposium on performance analysis of systems and software (ISPASS).
16. Xilinx, Inc. (2013). Device reliability report third quarter 2013. [http://www.xilinx.com/support/documentation/user\\_guides/ug116.pdf](http://www.xilinx.com/support/documentation/user_guides/ug116.pdf).
17. Violante M., Sterpone L., Manuzzato A., Gerardin S., Rech P., Bagatin M., Paccagnella A., Andreani C., Gorini G., Pietropaolo A., Cargarilli G., Pontarelli S., Frost C. (2007). A new hardware/software platform and a new 1/E neutron source for soft error studies: testing FPGAs at the ISIS facility. IEEE Trans. Nucl. Sci. 54(4):1184–1189.

## Часть II. Области применения

### ГЛАВА 2

# БРАЗИЛЬСКИЙ НАНОСПУТНИК С ПРИЕМНИКОМ ГЛОБАЛЬНОЙ СИСТЕМЫ СПУТНИКОВОЙ НАВИГАЦИИ И ФУНКЦИЕЙ СОПРОВОЖДЕНИЯ НА РЕКОНФИГУРИРУЕМОЙ СИСТЕМЕ НА КРИСТАЛЛЕ

*Глауберто Л. А. Альбукерке, Манозль Дж. М. Карвалью и Карлос Вальдerraма*

**Краткое содержание.** В этой главе представлена гибкая архитектура GPS-приемника с функцией частичного реконfigurирования (PR — Partial Reconfiguration) для реализации в базе реконfigurируемой системы на кристалле, состоящей из ПЛИС и двухъядерного процессора архитектуры ARM (СНК ПЛИС). За счет встроенной системы коррекции ошибок данное устройство соответствует требованиям, предъявляемым к наноспутникам в созвездии CONASAT. Приемник имеет функцию частичной реконfigurации, что улучшает его рабочие характеристики, позволяет совместно использовать аппаратные средства и оптимизировать потребляемую мощность. Кроме того, обеспечиваются все преимущества восстановления конфигурации на орбите. Предложенная архитектура в соответствии с требованиями CONASAT использует компоненты, имеющиеся в свободной продаже.

---

G. L. A. Albuquerque (✉)  
Barreira do Inferno Launch Center—CLBI, Parnamirim, Brazil  
e-mail: glauberto@engineer.com

M. J. M. Carvalho  
CRN—Centro Regional do Nordeste, Instituto Nacional de Pesquisas Espaciais—INPE, Natal, Brazil  
e-mail: manael@crn.inpe.br

C. Valderrama  
SEMI—Electronics and Microelectronics Department, University of Mons, Mons, Belgium  
e-mail: carlos.valderrama@umons.ac

## 2.1. Введение

Стандарт CubeSats стал доступной альтернативой для космических программ развивающихся стран [1], а в некоторых случаях привлекательным и для развитых стран. Технические условия CubeSat позволяют уменьшить затраты на запуск и время на разработку небольших спутниковых систем. Технические условия, разрабатываемые с 1999 г. Калифорнийским политехническим государственным университетом в сотрудничестве со Стендфорским университетом, помогли ученым во всем мире в развитии науки и исследовании космоса. Хотя CubeSats первоначально предназначался для образовательных целей, в наши дни имеются коммерческие, военные и межпланетные программы, которые используют этот стандарт как ценную альтернативу для многих типов космических программ [2–4].

Достижения в области электроники и микроэлектромеханических систем (МЭМС) в сочетании с такими технологиями, как программно-определяемая радиосистема (корректировка САВ) (SDR — Software Defined Radio) и цифровая обработка сигналов (DSP), способствовали снижению затрат и обеспечили их развитие. В частности, программируемые пользователем вентиляльные матрицы (ПЛИС) доказали свою рентабельность в различных областях, выходящих за пределы SDR. В дополнение к гибкости, обеспечиваемой способностью к реконфигурации, ПЛИС обладает преимуществом низкого электропотребления [5], что само по себе является очень важным параметром для применений в космосе. Спутник достаточно легко может получить электроэнергию, преобразуя солнечную энергию, однако это сопровождается значительным увеличением массы. Таким образом, для снижения суммарного объема потребляемой мощности спутники должны проектироваться на основе малогабаритных устройств с низким электропотреблением.

Помимо специфики каждой космической программы, комплекс космической аппаратуры всех спутников содержит некоторые комплексы связи и навигационное управление, для которых предусмотрены приемники глобальной навигационной спутниковой системы (GNSS — Global Navigation Satellite System). Кроме того, эти подсистемы должны быть устойчивыми к внешним воздействиям и надежными настолько, чтобы без сбоев работать в неблагоприятной окружающей среде. В соответствии с этим возможность частичной динамической реконфигурации (PDR — Partial Dynamic Reconfiguration) ПЛИС может быть дополнительным преимуществом при работе в космосе [6, 7]. Эта процедура не только позволяет адаптировать комплекс космической аппаратуры на орбите, но также обеспечивает определенную степень стойкости к воздействию радиации (например повторную инициализацию давшей сбой системы, ее замену и обновление).

В этой главе предлагается архитектура недорогого GPS-приемника, использующего имеющиеся в продаже СнК с ПЛИС, что соответствует требованиям, предъявляемым к спутникам созвездия CONASAT. Этот приемник использует преимущества современной СнК с ПЛИС и метода частичной реконфигурации для работы в космосе.

## 2.2. CONASAT

### 2.2.1. Проект CONASAT

CONASAT является проектом, основанным на применении созвездия наноспутников, который финансируется Бразильским национальным институтом исследований космоса (INPE — National Institute for Space Research). Его главной задачей является сбор данных об окружающей среде с тысяч платформ (DCPs — Data Collection Platforms), распределенных по всей территории Бразилии и на ее побережье. Это созвездие заменит использо-

вавшиеся ранее спутники SCD1 и SCD2, которые все еще активны, хотя у них уже истек проектный срок службы.

Основные задачи и принципы при разработке проекта в рамках CONASAT [8]:

- разработать экспертизу в области космических программ, особенно для наноспутников;
- обеспечить минимально возможный уровень затрат при достаточном уровне надежности и среднем сроке службы 5 лет;
- использовать в максимально возможной степени компоненты, имеющиеся в свободной продаже, а также серийно выпускаемые подсистемы;
- иметь гибкую платформу и модульную аппаратную платформу, которую могли бы использовать последующие поколения спутников этого созвездия;
- спутники в рамках проекта CONASAT должны быть совместимы с наноспутниками CubeSat;
- этот проект должен создавать новые возможности для развития бразильской высокотехнологичной промышленности.

Проект CONASAT займет в бразильской системе сбора экологических данных (SBCDA — System for Environmental Data Collection) сегмент, требующий высокого пространственного разрешения. Бразилия уже производит собственные платформы сбора данных (DCP) и некоторые части спутников системы CONASAT. Насколько это возможно, остальные части спутника будут производиться с участием бразильских экспертов. Например, используемый в настоящее время протокол связи между DCP и спутниками будет изменен, чтобы можно было обеспечивать двусторонний обмен данными.

Спутники CONASAT будут использоваться на низких околоземных орбитах (LEO — Low Earth Orbits, высоты от 500 до 800 км). Таким образом, они не будут все время находиться над территорией Бразилии. Все время, когда спутники будут находиться за пределами территории Бразилии, они будут обеспечивать работу других служб. Например, планируется включить в систему SBCDA мониторинг рыболовных судов. Для этого необходимо, чтобы части CONASAT, установленные на аппаратных средствах с изменяемой конфигурацией, поддерживали выполнение заданий по требованию. Что касается радиационной стойкости, то орбиты спутников на высоте около 600 км находятся в области низкой плотности ионов.

### 2.2.2. Архитектура спутников созвездия CONASAT

Типовая архитектура спутника, показанная на рис. 2.1, ничем не отличается от остальных. Она включает полное резервирование основных подсистем, включая систему *управления электропитанием*. Поэтому можно считать, что имеются две спутниковые системы в одной механической инфраструктуре. Этот выбор обусловлен стремлением к увеличению надежности всей системы благодаря возможности использования покупных серийных изделий. Другой причиной такого решения является значение среднего времени между отказами (MTBF — Mean Time Between Failures) у имеющихся в продаже модулей CubeSat, которые не рассчитаны на срок службы средней продолжительности.

Подсистема *управления резервированием* всякий раз принимает решение, какую из подсистем следует активировать. Подсистема *управления положением корабля* содержит магнитный датчик крутящих моментов (iMTQ — includes a Magnetorquer), звездный гироскоп, трехкоординатный гироскоп, астронавигатор и маховики. Этот спутник также использует приемник глобальной системы спутниковой навигации (GPS-приемник) для упрощения прогнозирования орбиты. Применение многочисленных датчиков подчиняется принципу обеспечения максимальной надежности. Однако, тогда как совместное использование датчиков повышает эффективность, это одновременно приводит к увеличению веса спутника и потребляемой им мощности. Более того, возможность обрабатывать сигналы GPS-приемника должна быть адаптирована к орбитальной скорости.

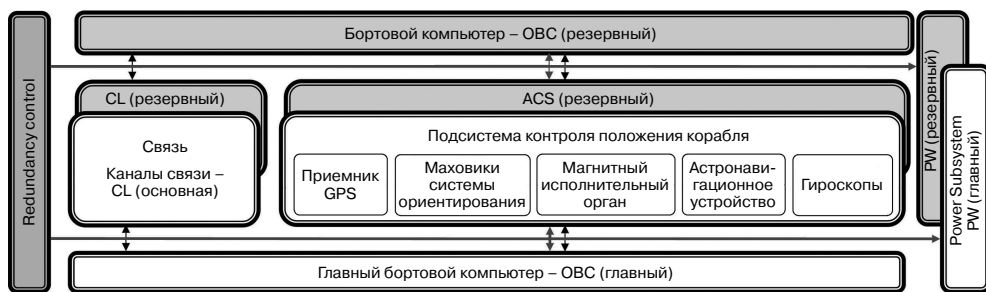


Рис. 2.1. Функциональная архитектура CONASAT (взято из [8])

Поэтому способность космического GPS-приемника обрабатывать данные необходимо тщательно адаптировать. *Подсистема связи* состоит из СВЧ-линии связи передачи данных на спутник и линии передачи данных со спутника дециметрового диапазона (S-полосы). Она отвечает за ретрансляцию на наземные станции данных, полученных с платформ DCP. Подсистемы *питания* и *управления положением корабля* выполняют на орбите настолько узкоспециализированные задачи, что не могут выполнять дополнительные функции.

Таким образом, GPS-приемник является единственной подсистемой, которая нуждается в модификации прямо на орбите для успешного выполнения текущего задания по требованию. По этой причине приемник должен быть создан на программной платформе. Более того, из-за требований к рабочим характеристикам и мощности эта универсальность должна поддерживаться аппаратными средствами с возможностью реконфигурирования. Однако на рынке нет такого GPS-приемника, который мог бы на орбите перестраивать свою конфигурацию и соответствовать требованиям стандарта CubeSat. При оптимальном выборе ПЛИС неиспользуемые элементы логики могут обеспечивать дополнительную функциональность или даже, когда приемник находится в режиме ожидания, давать возможность совместно использовать всю платформу. Это позволит уменьшить размер и число электронных устройств с благоприятным влиянием на потребление энергии и общий вес спутника.

### 2.3. Архитектура программного обеспечения приемников глобальной системы спутниковой навигации (GPS)

Как можно было видеть ранее, основанный на программном обеспечении метод для приемников глобальной спутниковой системы навигации был естественным выбором для рассматриваемой схемы, особенно в тех случаях, когда сигналы GPS из созвездия спутников используют цифровую модуляцию (BPSK — Binary Phase-Shift Keying, двоичная фазовая манипуляция). Принимая это во внимание, сборка GPS-приемника (или другой системы спутниковой навигации), несмотря на некоторые трудности, не является невыполнимой задачей [9]. Благодаря большей распространенности GPS в данной области в оставшейся части этой главы будет рассматриваться GPS в качестве основы для объяснения предложенной архитектуры.

В зависимости от используемого набора ИС (чипсета) можно выделить два варианта исполнения (подхода к проектированию) приемников: программное и аппаратное. Приемники аппаратного исполнения используют заказные специализированные большие интегральные схемы для выполнения всех заданий по сопровождению и навигации. Серийно

выпускаемые приемники имеют ограниченное применение или не применяются совсем в авиации или аэрокосмической/космической отрасли. В приемниках программного исполнения процесс обработки сигналов программируемый, для чего применяется процессор общего назначения (GPP — General Purpose Processor), процессор обработки цифровых сигналов, графический процессор или даже аппаратные средства с реконфигурируемой ПЛИС. Иногда разработчики используют комбинацию этих устройств [10–12].

На рис. 2.2 можно видеть базовую архитектуру GPS-приемника. Хотя имеющиеся различного типа приемники спутниковой навигации спроектированы для различных областей применения, все их базовые архитектуры содержат одни и те же функциональные блоки.

Принимаемый антенной радиосигнал необходимо предварительно усилить и отфильтровать, на входном каскаде лежит задача аналого-цифрового преобразования и сжатия сигнала за счет понижения частоты перед оцифровкой. Блок обработки сигналов основной полосы частот обнаруживает и контролирует каждый поступающий сигнал для расчета собственного положения и скорости. На каждый отслеживаемый спутник необходимо иметь один такой блок, занимающийся извлечением наблюдательной и навигационной информации из канала. Теоретически приемник может отслеживать до 12 спутников системы GPS, однако для расчета собственного местоположения приемнику достаточно использовать только четыре. После правильного отслеживания сигналов полученные данные измерений направляются в блок обработки приложений.

Блок приложений использует информацию из следящих систем для разных целей. Типичным применением этой информации является контроль параметров ионосферы,

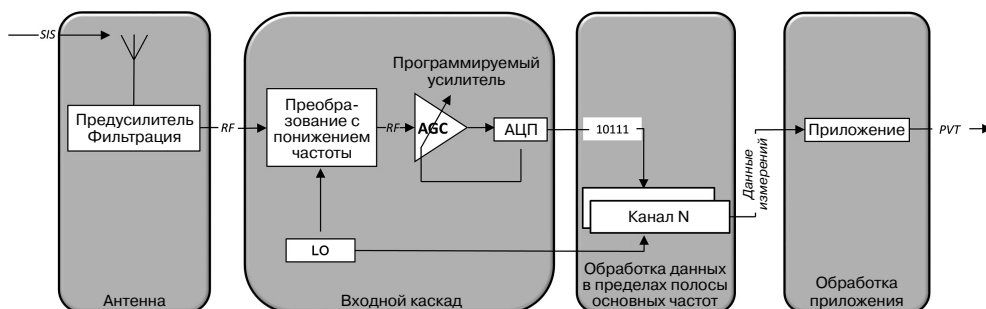


Рис. 2.2. Архитектура типowego GPS-приемника [13]

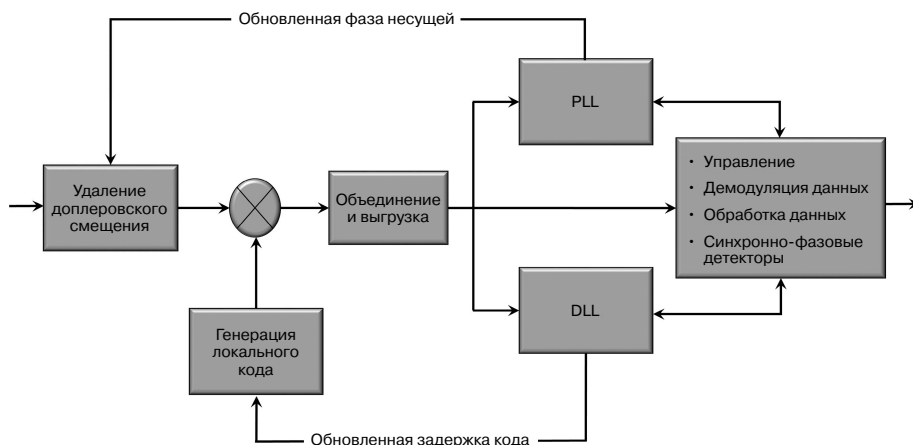


Рис. 2.3. Обработка сигнала полосы основных частот [13]

расчеты дифференциальной глобальной системы определения местоположения (DGPS — Differential Global Positioning System), статическая и кинематическая съемка.

Время обработки *сигналов основной полосы частот* определяет две категории приемников: реального времени и постобработки (постпроцессинга). В случае постобработки (постпроцессинга) информация по полосе основных частот используется для получения корреляций между принимаемыми сигналами и опорным сигналом. В результате образуются промежуточные данные, сохраняемые для последующей обработки в пакетном режиме с помощью дополнительных алгоритмов. При этом приемник не способен определять положение в реальном времени. Эта задержка критична при навигации на орбитальных скоростях, а также влечет за собой использование дополнительной вычислительной мощности и контролирующих алгоритмов.

*Обработка сигналов основной полосы частот* включает все алгоритмы обнаружения и сопровождения «видимого» сигнала GPS с помощью синхронизации на основе известного кода псевдослучайной последовательности (PRN — Pseudorandom Noise code) и удаления максимального количества ошибок. Этот процесс основан на принципе корреляции сигналов: принимаемый сигнал многократно коррелируется с опорным сигналом ожидаемого PRN-кода, который известен априори. Функциональная конструкция приведена на рис. 2.3. Для получения из корреляции достоверного уровня значимости в приемнике генерируется локальный опорный сигнал, учитывая фазу несущей частоты сигнала, задержку кода, доплеровскую частоту и PRN-код [12]. Для получения максимальной корреляции блоки DLL (Delay Locked Loop, цифровая автоподстройка по задержке) и PLL (Phase Locked Loop, фазовая автоподстройка частоты) отвечают за отслеживание задержки кода и несущей соответственно.

## 2.4. Проектирование аппаратных средств

### 2.4.1. Входной каскад

Даже для программного исполнения приемников системы спутникового позиционирования входным каскадом чаще всего являются устройства в базисе СБИС. На рынке представлены десятки вариантов, в том числе и с перенастраиваемой конфигурацией, которые появились совсем недавно [14]. Бразильские ученые использовали ИС GP2000 в GPS-приемнике для геофизических ракет [15]. Более того, как было показано в [16], ИС GP2000 в достаточной степени защищена от воздействия радиации, что позволяет использовать ее на низких околоземных орбитах без существенных изменений. Однако большинство остальных GPS-приемников для космического применения используют в качестве входного каскада ИС GP2015, изготавливаемые компаниями DLR и Surrey Technologies [17, 18]. Поэтому семейство ИС GP2015 можно рассматривать как сертифицированный вариант.

### 2.4.2. Модуль обработки сигналов основной полосы частот

Хотя модуль GP2015 входного каскада является хорошим выбором для этого приемника, применение других ИС этого семейства приведет нас к необходимости использования аппаратной реализации приемника, преимущества программной реализации в отношении универсальности алгоритма и связанной с этим эффективности обработки данных будут потеряны.

Входной каскад на основе ИС GP2015 на частоте выборки 5,71 МГц предоставляет 2-битные выборки. Ширина полосы частот, требуемая данной частотой:

$$f_s = 5,71 \text{ Мвыб/с}, \quad (2.1)$$