

**СПОСОБ И СИСТЕМА ИНЪЕКЦИИ ОШИБОК ДЛЯ ТЕСТИРОВАНИЯ
СБОЕУСТОЙЧИВЫХ ПРОЦЕССОРОВ БОРТОВЫХ СИСТЕМ КОСМИЧЕСКИХ АППАРАТОВ**

С. А. Чекмарёв

Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева
Российская Федерация, 660014, г. Красноярск, просп. им. газ. «Красноярский рабочий», 31
E-mail: zaq259@yandex.ru

Предложен способ инъекции ошибок для тестирования сбоеустойчивости бортовых процессоров на основе модификации метода внутрикристалльной отладки с использованием тестового порта процессора. В отличие от базового метода инъекция ошибок производится не с помощью внешнего управляющего компьютера, а с помощью встроенного в тестируемую систему аппаратного инжектора ошибок. Для представляемого способа разработана архитектура системы инъекции ошибок. Модуль инъекции, используя порт тестирования процессора, может прочитать данные по требуемому адресу внутренней памяти, внести ошибку, инвертировав бит в соответствии с требуемым номером, данные с ошибкой обратно записать по данному адресу. Представлен пример реализации предлагаемой системы. Описана процедура внесения и исправления одиночного сбоя во внутрикристалльной памяти процессора. Показано, что предлагаемый подход не привносит большой избыточности в основную архитектуру ядра процессора. В результате экспериментов примерно 90 % вносимых ошибок были обнаружены, из них не менее 92 % ошибок исправлены. Применение представленного аппаратного инжектора обеспечивает большую автономность и реалистичность проведения испытаний процессора на сбоеустойчивость.

Ключевые слова: внутрикристалльная отладка, одиночный сбой, инъекция ошибок, LEON3, система на кристалле, ПЛИС.

Vestnik SibGAU
2014, No. 4(56), P. 132–138**METHOD AND SYSTEM OF ERROR INJECTION FOR TESTING THE FAULT TOLERANCE
PROCESSOR ON-BOARD CONTROL SYSTEM OF SPACECRAFT**

S. A. Chekmaryov

Siberian State Aerospace University named after academician M. F. Reshetnev
31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660014, Russian Federation
E-mail: zaq259@yandex.ru

Error injection method for testing onboard processors' fault-tolerance is proposed. It is based on the modification of the on-chip debugging method which uses the test port of the processor. Unlike the basic method, error injection is made by a built-in hardware error injector rather than by an external computer. An error injection system architecture is developed for the proposed method. Using the processor test port injection module can read the data with the required address in the internal memory; inject errors by inverting the required bit; write back data with injected errors by the initial address. An implementation example of the proposed system is demonstrated. The procedure of injecting and correcting a single failure in the processor on-chip memory is described. It is shown that the proposed method does not introduce much excessiveness in the core architecture of the processor. As a result of experiments about 90 % of injected errors were detected and at least 92 % of them were corrected. Implementation of the hardware injector provides greater autonomy and realistic testing of the processor's fault-tolerance.

Keywords: On-chip debugger, single event upset, fault injection, System on Chip, FPGA.

Введение. В настоящее время, в связи с успехами микроэлектронных технологий в области конструктивно-топологических мер защиты от ионизирующего излучения и достигнутым уровнем схемотехнических способов обеспечения отказоустойчивости на уровне логических ячеек интегральной схемы [1], усилия

в области проектирования сбоеустойчивой архитектуры микропроцессоров в основном концентрируются на защите от одиночных ошибок (Single-event Upset – SEU) внутренней памяти процессора: кэш-памяти и файла регистров [2].

Важным элементом создания сбоеустойчивого процессора является отладка, тестирование и испытания тех или иных вариантов построения его сбоеустойчивой архитектуры. Тестирование и испытания должны удостоверить эффективность используемых методов и средств защиты внутренней памяти процессора от одиночных сбоев SEU [3]. Сбоеустойчивость процессоров в процессе тестирования подтверждается с помощью тех или иных методов технологии внедрения (инъекции) одиночных ошибок в его внутреннюю память [4].

Имеется несколько подходов к инъекции аппаратных ошибок во внутреннюю память процессора. В данной работе применительно к программируемым логическим интегральным схемам (ПЛИС) типа FPGA (Field Programmable Gate Array) рассмотрен метод, который основывается на возможностях OCD-методов, использующих для внесения ошибок блоки отладки ядра процессора (порты тестирования, test ports), которыми оснащены все современные микропроцессоры [5].

OCD-методы, используя порт тестирования процессора, для внесения ошибки выполняют следующие действия: остановка процессора, чтение и внесение изменений в регистры процессора и кэш-память, продолжение работы процессора. Доступ к устройствам отладки осуществляется через внешний стандартный интерфейс, через который отлаживаемая система подсоединяется к управляющему компьютеру. Управляющий компьютер обеспечивает все основные функции системы инъекции ошибок (fault injection environment) в устройство, находящееся под тестированием. Подобным образом в [6] OCD-метод использован для проверки архитектурных решений по сбоеустойчивости к SEU процессорного ядра, встроенного в antifuse-FPGA.

В данной работе предлагается изменить OCD-метод применительно к перепрограммируемым FPGA, в первую очередь flash-FPGA, не имеющих возможности частичной реконфигурации, характерной для SRAM-FPGA, и тем самым соответствующего метода инъекции ошибок [7]. Все основные функции по внесению ошибок, такие как генерация списка вносимых ошибок, непосредственное внесение ошибок во внутреннюю память процессора, первичное сохранение и классификация последствий от внесенных ошибок, реализованы в самой FPGA. Управляющему компьютеру отводится второстепенная роль: он осуществляет общее управление экспериментами по инъекции ошибок, обеспечивает накопление уже частично обработанных данных, полученных в результате проведенных экспериментов, для их окончательного анализа. Подход обеспечивает слабую связь с внешним управляющим компьютером и, таким образом, высокую скорость, большую реалистичность и автономность проведения экспериментов по инъекции ошибок.

Архитектура системы инъекции ошибок. Архитектура системы для реализации предлагаемого подхода инъекции ошибок представлена на рис. 1. Системой под тестированием (System Under Tests – SUT) является полнофункциональный прототип (инженерный образец) нерезервированного бортового компьютера, основанного на софт-процессоре, размещенном в FPGA [8]. От летного образца бортового компьютера его отличает применение электронной компонентной базы некосмического исполнения. Подобные компьютеры используются при создании отдельных видов бортовой аппаратуры, в том числе и для малых космических аппаратов [9]. Таким образом, возможна отработка защиты софт-процессора от SEU в уже законченной системе, предназначенной для практического использования. Данная возможность является дополнительным достоинством предлагаемого метода.

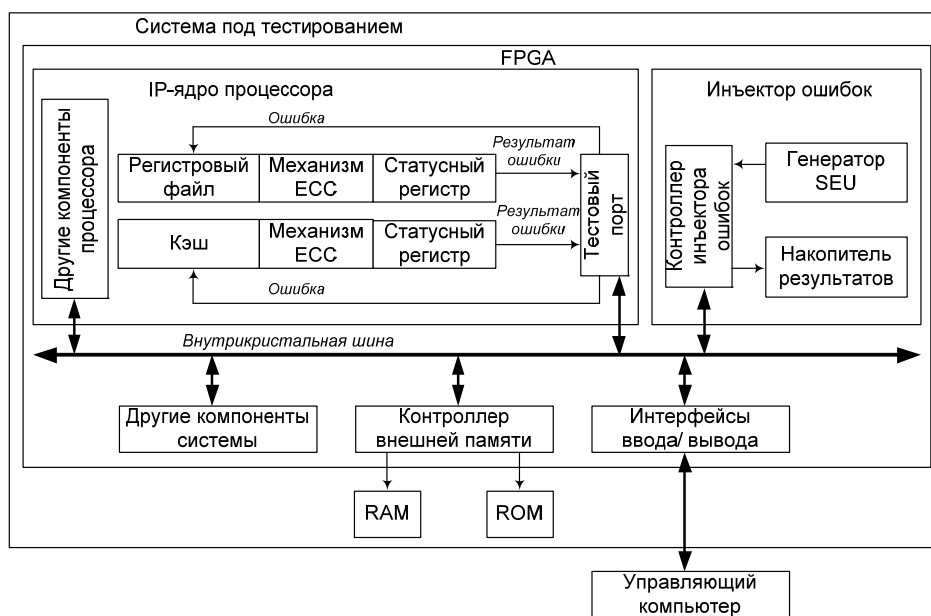


Рис. 1. Архитектура системы инъекции ошибок

Архитектура представляет собой систему на кристалле (СнК). Она состоит из процессорного ядра и модуля инъекции ошибок, размещенных в FPGA, и внешнего управляющего компьютера. Основные компоненты архитектуры системы инъекции ошибок располагаются в FPGA. Процессор и модуль инъекции полностью независимы и связаны внутрикристалльной шиной. Управляющий компьютер подключается к модулю интерфейсов ввода/вывода, также связанного с внутренней шиной СнК. Он осуществляет общее управление и накопление результатов экспериментов.

Ядро процессора оснащено тестовым портом, являющимся стандартным компонентом для большинства современных микропроцессоров. Тестовый порт может иметь прямой внешний интерфейс для взаимодействия с управляющим компьютером, а может, как в нашем случае, быть соединен с внутрикристалльной шиной. Такая архитектура характерна, например, для семейства процессоров Leon [10], для процессоров ARM [11]. Эта особенность является ключевой для предлагаемой архитектуры. Она позволяет все компоненты для инъекции ошибок, как блоки на внутрикристалльной шине, разместить в FPGA, а внешнему компьютеру отвести вспомогательную роль.

Модуль инъекции, используя порт тестирования, может прочитать данные по требуемому адресу внутренней памяти; внести ошибку, инвертировав бит в соответствии с требуемым номером; данные с ошибкой обратно записать по данному адресу. С другой стороны, с помощью порта тестирования модуль инъекции может прочитать информацию о факте выявления ошибки во внутренней памяти, обнаруженной с помощью механизма ECC. Эта информация находится в статусном регистре ECC, хранящем адрес внутренней памяти, по которому обнаружена ошибка, и признак – ошибка при обнаружении исправлена или нет.

Модуль инъекции ошибок состоит из трех блоков: контроллера модуля инъекции ошибок (Fault Injection – FI), генератора SEU, накопителя результатов. Контроллер FI:

- в целом координирует работу компонент модуля инъекции;
- используя информацию, полученную от генератора SEU, заносит ошибки в требуемые биты в требуемые адреса внутренней памяти;
- читает в регистре статуса ECC информацию о факте обнаружения механизмом ECC ошибки, проверяет признак «ошибка исправлена / не исправлена» и сохраняет информацию о результатах внесения ошибки в накопитель.

Генератор SEU в режиме реального времени генерирует новую ошибку для занесения во внутреннюю память. Он определяет три параметра: вид ошибки – однократная (ошибка в одном бите) или двукратная (ошибка в двух соседних битах); адрес ошибки во внутренней памяти; момент времени для инъекции ошибки. Закон генерации ошибок по данным параметрам может быть случайным равномерным, а может быть более сложным, учитывающим, например, процессы деградации и накопления радиации электронными компонентами с течением времени эксплуатации. Последнее может потребоваться для про-

ведения ускоренных испытаний, для подтверждения сбоеустойчивости процессора на конец срока эксплуатации, например, бортового компьютера в космическом аппарате. В простейшем случае генератор SEU может быть просто статической таблицей событий, параметры которых жестко заданы. Данный вариант может быть использован для создания системы контролепригодности ядра процессора в части сбоеустойчивости к SEU.

Накопитель результатов внесения ошибок сохраняет результаты экспериментов. Он располагается во внутрикристалльной памяти FPGA и может быть достаточно большим для FPGA большого размера и зависеть от величины процессорного ядра. В простейшем случае накопитель может содержать только количество обнаруженных ошибок и количество исправленных (или неисправленных) ошибок для каждого вида внутренней памяти процессора. Для проведения более глубокого анализа результатов для каждого случая обнаружения ошибки в накопителе может сохраняться ее адрес во внутренней памяти процессора. В этом случае накопитель представляет таблицу (несколько таблиц), содержимое которой по окончании эксперимента передается в управляющий компьютер. Если время эксперимента велико, то для предотвращения переполнения накопителя предусмотрена возможность его периодического считывания. Возможен также вариант использования под накопитель результатов экспериментов внешней памяти SUT. В этом случае контроллер FI должен быть «мастером» на внутрикристалльной шине.

Пример реализации архитектуры системы инъекции ошибок. Представленная архитектура инъекции ошибок реализована в ПЛИС-устройстве на базе flash-FPGA ProASIC3E A3PE3000, являющемся инженерным образцом бортового компьютера. Оно было разработано в СибГАУ для малого космического аппарата TabletSat [12]. Видоизмененная под данный пример реализации архитектура системы инъекции ошибок представлена на рис. 2. В качестве IP-процессора использовано открытое на уровне VHDL-кода модифицированное процессорное ядро LEON3. Размер кэш-данных 8 KB, размер кэш-команд равен 8 KB, размер регистрового файла – 128 32-разрядных слов. Связь между процессорным ядром и модулем инъекции осуществляется по внутрикристалльной шине AMBA [13]. AMBA имеет две шины: высокоскоростную AHB и низкоскоростную APB. Тестовый порт DSU (Debug Set Unit) расположен на высокоскоростной шине AHB. Управляющий компьютер подключается к низкоскоростной шине APB через интерфейс UART.

Отработка представленной на рис. 2 архитектуры проводилась в процессе экспериментов по внесению ошибок в регистровый файл и во внешнее ОЗУ процессора. Целью проводимых работ являлась реализация данной архитектуры и проверка ее работоспособности с оценкой вносимой избыточности в FPGA.

Механизм ECC [14] был реализован на базе помехоустойчивого модифицированного кода Хэя [15]. После каждого случая обнаружения ошибки статусный регистр ECC сохранял 2 признака: «ошибка

обнаружена» и «ошибка исправлена / не исправлена». Генератор SEU был представлен таблицей однократных ошибок, случайно распределенных по адресам внутрикристалльной (регистровый файл АЛУ) и внешней (ОЗУ) памяти. Данная таблица также содержит в себе значения интенсивности инъекции сбоев, которая определяется временными задержками между событиями «инъекция завершена». Таблица однократных сбоев инициализируется при включении. Имеется возможность получить к ней доступ и инициализировать вручную перед проведением эксперимента. Кроме того, контроллер инжектора ошибок может самостоятельно инициализировать данную таблицу. Для этого в его состав были включены генераторы псевдослучайных последовательностей с настраиваемым диапазоном выдаваемых значений. Возможность заполнения листа ошибок случайными значениями позволяет использовать малые размеры таблицы ошибок при внесении большого объема инъекций. Накопитель результатов инъекции размещался во внутрикристалльной памяти FPGA, его размер был равен количеству ошибок в листе ошибок SEU.

Обращение к данным в ОЗУ с целью запуска механизма ECC производилось с помощью тестовой программы, целью которой являлась организация чтения максимально возможного пространства ОЗУ. В программе выделялся и инициализировался массив данных, равный размеру ОЗУ. Далее в цикле происходил расчет контрольной суммы данных, расположенных в массиве. Если механизм ECC отрабатывал вносимые ошибки, то контрольная сумма после каждой итерации должна оставаться одной и той же. Обращение к регистровому файлу производилось в моменты работы процессора.

Поскольку в процессе работы процессора даже над выполнением предложенной тестовой программы невозможно гарантировать, что после внесения ошибки при чтении каждой ячейки памяти ошибка будет обнаружена ранее, чем эта ячейка будет переписана другими данными, не следует ожидать, что каждая инъецированная ошибка будет обнаружена, а затем исправлена. Учитывая это обстоятельство, был реализован асинхронный режим инъекции ошибок и обращения к памяти, когда модуль инъекции ошибок и тестовая программа работают независимо друг от друга. При этом критерием работоспособности предложенного подхода являлось не то, что каждая внесенная ошибка обнаружена и исправлена при сохранении контрольной суммы после каждой итерации, а то, что если ошибка обнаружена, то она исправлена, а контрольная сумма осталась неизменной после каждой итерации тестовой программы.

Для оценки избыточности, вносимой в FPGA, рассмотренная архитектура была реализована для разного количества ошибок, определенных в листе событий SEU. Результаты синтеза для нескольких ПЛИС компаний ACTEL и ALTERA представлены в табл. 1. Из табл. 1 следует, что предложенный метод не является ресурсозатратным. С ростом размера листа ошибок количество логических блоков, необходимых для реализации метода, остается практически неизменным. Потребность во внутрикристалльной памяти для

многократно программируемых и однократно программируемых flash-ПЛИС компании Microsemi в процентном соотношении увеличивается быстрее. В первую очередь это связано с особенностями данных ПЛИС. Однако применение в предложенном методе функции автоматического заполнения позволяет использовать лист ошибок меньшего объема. Он может быть автоматически проинициализирован для внесения новой порции инъекций. Тем самым уменьшится потребность в используемых блоках памяти.

В табл. 2 представлены результаты экспериментов по внесению ошибок в память. Как и ожидалось, не все вносимые ошибки были обнаружены. Часть инъекций в ОЗУ пришлось на неиспользуемые области памяти, часть – на перезаписываемые участки памяти ПО. Результаты экспериментов показывают, что примерно 90 % вносимых ошибок были обнаружены.

В случаях обнаружения ошибок не менее 92 % ошибок были исправлены с помощью кода Хсяо. Данные результаты связаны с интенсивностью внесения ошибок и объемом памяти. В первом эксперименте при инъекции 100 ошибок в ОЗУ объемом 16 МБ сбой распределился равномерно по всему объему памяти. Все они были обнаружены и исправлены ввиду отсутствия двойных ошибок в памяти. Данные в регистровом файле обновляются очень часто ввиду постоянного обращения процессора к регистровой памяти. Поэтому многие инъекции оказались неэффективными. Для того чтобы повысить эффективность внесения ошибок, пришлось значительно увеличить интенсивность инъекций. В результате, в последующих экспериментах при инъекции 500, 1000 и 2000 ошибок удалось смоделировать ситуации для проверки функционала обнаружения двойных ошибок в ОЗУ и регистровом файле, а также отработать реакцию тестового ПО на данные события. Код Хсяо не исправляет двойные ошибки, поэтому не все ошибки, приведенные в результатах тестирования, были исправлены. Таким образом, меняя интенсивность сбоев, можно промоделировать поведение системы, отработать возможные внештатные ситуации. В целом же результаты экспериментов соответствуют ожидаемым, а предложенная архитектура системы инъекции ошибок соответствует задаче тестирования сбоеустойчивой архитектуры процессорных IP-блоков.

Заключение. Рассмотренный способ и архитектура системы внесения ошибки являются развитием OCD-методов инъекции для процессорных ядер, встроенных в FPGA-устройства. Их основное отличие состоит в том, что вся инфраструктура по внесению ошибок для классификации и первичного накопления последствий ошибок содержится в FPGA. Управляющий компьютер в процессе эксперимента по внесению ошибок выполняет вспомогательную роль пост-обработчика его результатов. Это увеличивает реалистичность проведения эксперимента, несколько увеличивает его скорость, а также позволяет проводить эксперимент в уже законченных образцах FPGA-устройств. Последнее обстоятельство открывает возможность использования данного метода для контролепригодности сбоеустойчивых процессоров.

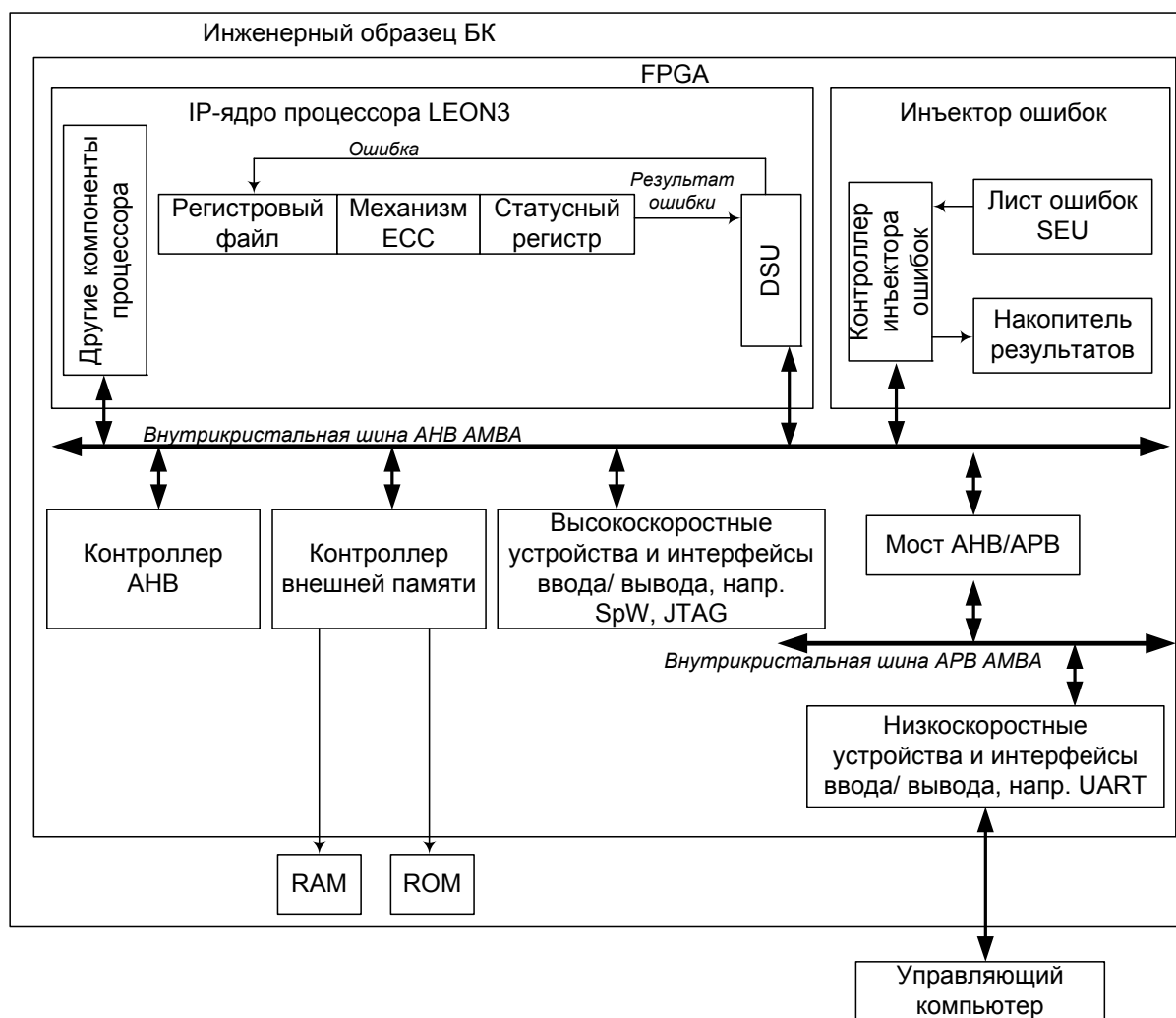


Рис. 2. Видоизмененная под пример реализации архитектура системы инъекции ошибок

Таблица 1

Результаты синтеза инжектора ошибок для разных ПЛИС

Размер листа ошибок SEU	ACTEL A3PE3000L		ACTEL RTAX 1000S		ALTERA Cyclone IV	
	Размер		Размер		Размер	
	Логических блоков (% от всех логических блоков)	Блоков памяти (% от всех блоков памяти)	Логических блоков (% от всех логических блоков)	Блоков памяти (% от всех блоков памяти)	Логических блоков (% от всех логических блоков)	Блоков памяти (% от всех блоков памяти)
100	918 (1,2 %)	2 (1,7 %)	288 (1,5 %)	1 (2 %)	370 (< 1 %)	2,176 (< 1 %)
500	1001 (1,3 %)	3 (2,6 %)	386 (2,1 %)	3 (8 %)	369 (< 1 %)	8 (< 1 %)
2000	1096 (1,45 %)	9 (8 %)	448 (2,4 %)	9 (25 %)	379 (< 1 %)	34 (< 1 %)

Таблица 2

Результаты инъекции ошибок в память процессора LEON3

Количество внесенных ошибок SEU	Регистровый файл (внутрикристальная память)		ОЗУ (внешняя SRAM память)	
	Обнаружено / %	Исправлено / %	Обнаружено / %	Исправлено / %
100	87/87	87/100	100/100	100/100
500	463/92,6	463/100	473/94	457/96
1000	958/95,8	949/99	917/91,7	826/91
2000	1920/96	1845/96	1791/89	1649/92

Предлагаемый подход является малоинтрузивным, он не привносит большой избыточности в основную архитектуру ядра процессора. Модуль внесения ошибок не встраивается в ядро процессора. Он связан с процессором внутрикристалльной шиной. Частичные изменения ядра процессора относятся только к механизму ECC.

Данная архитектура в наибольшей степени удовлетворяет целям разработки и отладки нового эффективного решения для обеспечения сбоеустойчивости процессора. Метод может применяться как для отработки HDL-описаний сбоеустойчивых архитектур для микропроцессоров, изготавливаемых в виде корпусированных интегральных схем, так и для софтверных процессоров (процессорных ядер, процессорных сложных функциональных блоков), для которых результат отладки и испытаний HDL-описания является уже конечной стадией их создания. Отметим, что сбоеустойчивые процессорные ядра занимают свою вполне определенную «нишу» в космическом электронном приборостроении, в частности, они широко используются при создании малых космических аппаратов.

Благодарности. Исследование выполнено при поддержке Министерства образования и науки Российской Федерации, соглашение № 14.574.21.0041 от 19.06.2014 г.

Acknowledgments. The study was supported by the Ministry of Education and Science, the agreement of 19.06.2014 № 14.574.21.0041.

Библиографические ссылки

1. Юдищев В. Радиационно стойкие интегральные схемы. Надежность в космосе и на Земле // *Электроника: наука, технология, бизнес*. 2007. № 5. С. 72–77.
2. Осипенко П. Одиночные сбои – вызов для современных микропроцессоров // *Электронные компоненты*. 2009. № 7. С. 12–15.
3. Таперо К. И., Улимов В. Н., Членов А. М. Радиационные эффекты в кремниевых интегральных схемах космического применения. М.: БИНОМ. Лаборатория знаний, 2012. С. 234.
4. Hsueh M.-C., Tsai T. K., Iyer R. K. Fault Injection Techniques and Tools // *Computer*. 1997. P. 75–82.
5. Fidalgo A. V., Alves G. R., Ferreira J. M. Real Time Fault Injection Using Enhanced OCD – A Performance Analysis // 21st IEEE Intern. Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'06). 2006. P. 254–264.
6. Gaisler J. LEON3-FT-RTAX SEU test results // GAISLER RESEARCH. 2005. P. 8.
7. Qiang Zhang, Jun Zhou, Xiaozhou Yu. A Kind of Low-cost Non-intrusive Autonomous Fault Emulation System // *Computer and Information Science*. 2011. Vol. 4, No. 1. P. 90–99.
8. Концепция создания бортового комплекса управления для малых космических аппаратов / В. Х. Ханов [и др.] // *Вестник СибГАУ*. 2012. № 5. С. 144–149.
9. Гобчанский О. Проблемы создания бортовых вычислительных комплексов малых космических аппаратов // *Современные технологии автоматизации*. 2001. № 4. С. 28–34.
10. SPARC V8 32-bit Processor LEON3 // LEON3-FT Companion Core Data Sheet Template Design. 2010. P. 41.
11. Козлов-Кононов Д. Процессорные ядра семейства Cortex: сочетание высокой производительности и низкого энергопотребления // *ЭЛЕКТРОНИКА: наука, технология, бизнес*. 2010. № 8. С. 16–24.
12. Спутникс [Электронный ресурс]. URL: <http://www.sputnix.ru/ru/technologies> (дата обращения: 25.08.2013).
13. AMBA Specification rev. 2.0, May 99, IHI 0011A, ARM Ltd. [Электронный ресурс]. URL: <http://www.arm.com> (дата обращения: 25.08.2013).
14. GRLIB IP Library User's Manual // Aeroflex Gaisler. 2010. P. 78.
15. Hsiao M. Y. A Class of Optimal Minimum Odd-Weight-Column SEC-DED Codes // *IBM J. Res. Develop.* 1970. № 14. P. 395–401.

References

1. Yudin V. [Radiation hardened integrated circuit reliability in space and on earth]. *Elektronika, Nauka, Tekhnologiya, Biznes*. 2007, no. 5, p. 72–77 (In Russ.).
2. Osipenko P. [Single event upset – a challenge for modern microprocessors]. *Elektronnye komponenty*. 2009, no. 7, p. 12–15 (In Russ.).
3. Ulimov V. N., Tapero K. I. *Radiatsionnye efekty v kremnievykh integral'nykh skhemakh kosmicheskogo primeneniya* [Radiation effects in silicon integrated circuits for space application]. Moscow, Binom. Laboratoriya znaniy Publ., 2012, 234 p.
4. M.-C. Hsueh, T. K. Tsai, R. K. Iyer. Fault Injection Techniques and Tools. *Computer*, 1997, p. 75–82.
5. Andre V. Fidalgo, Gustavo R. Alves, Jose M. Ferreira, Real Time Fault Injection Using Enhanced OCD – A Performance Analysis, dft, 21st IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'06), 2006, p. 254–264.
6. Jiri Gaisler, LEON3-FT-RTAX SEU test results GAISLER RESEARCH, 2005, p. 8.
7. Qiang Zhang, Jun Zhou, Xiaozhou Yu A Kind of Low-cost Non-intrusive Autonomous Fault Emulation System. *Computer and Information Science*. 2011, vol. 4, no. 1, p. 90–99.
8. Hanov V. Kh., Vergazov M. Ju., Chekmarev S. A., Shahmatov V. A., Lukin F. A. [The concept of creating an onboard control system for satellites]. *Vestnik SibGAU*. 2012, no. 5 (45), p. 144–149 (In Russ.).
9. Gobchanskij O. [Problems of creation of onboard computing systems on satellites] *Sovremennye tehnologii avtomatizacii*. 2001, no. 4, p. 28–34 (In Russ.).
10. SPARC V8 32-bit Processor LEON3 LEON3-FT Companion Core Data Sheet Template Design Gaisler Research, 2010. p. 41.
11. Kozlov-Kononov D. [Family of processor cores Cortex: a combination of high performance and low

power consumption]. *ELEKTRONIKA: Nauka, Tekhnologiya, Biznes*. 2010, no. 8, p. 16–24 (In Russ.).

12. Sputniks. Available at: <http://www.sputnix.ru/ru/technologies> (accessed: 25.08.2013).

13. AMBA Specification rev. 2.0, May 99, IHI 0011A, ARM Ltd. Available at <http://www.arm.com> (accessed: 25.08.2013).

14. GRLIB IP Library User's Manual. Aeroflex Gaisler. 2010. p. 78.

15. M. Y. Hsiao. A Class of Optimal Minimum Odd-Weight-Column SEC-DED Codes. *IBM J. Res. Develop.* 1970. no 14, p. 395–401.

© Чекмарёв С. А., 2014