

Повышение стойкости к воздействию ТЗЧ устройств на базе ПЛИС с использованием инструмента Synplify Premier

С. Белоусов¹

УДК [004.9::658.512.22]::[004.31::629.78] | ВАК 05.13.12

В настоящее время программируемые логические интегральные схемы (ПЛИС) получили широкое распространение в ряде областей, в особенности характеризующихся относительно небольшими объемами партий и высокой специализацией изделий с точки зрения их функциональности. Среди преимуществ ПЛИС для таких применений – возможность разработки специализированных устройств обработки данных и цифрового управления в значительно более короткие сроки и с меньшими затратами, чем в случае создания заказных ИС. Одной из сфер, в которых активно применяются ПЛИС, является космическая аппаратура – область, предъявляющая особые требования к компонентам, в том числе в отношении радиационной стойкости.

ПЛИС традиционно используются для построения критически важных функциональных узлов, таких как средства контроля и управления бортовыми системами сбора и обработки информации. Это означает, что к устройствам на основе ПЛИС предъявляются повышенные требования в отношении устойчивости к отказам, вызываемым внешними воздействиями. Одним из основных таких воздействий в приложении к аппаратуре космического применения является влияние на электронную аппаратуру тяжелых заряженных частиц (ТЗЧ), приводящих к случайному возникновению индуцированного заряда в проводниках микросхемы и, как следствие, к образованию нежелательных коротких паразитных импульсов (glitch), которые могут быть в дальнейшем интерпретированы системой как полезные сигналы, что, в свою очередь, может вызывать сбои и даже выход системы из строя (рис. 1),

Непрерывающееся уменьшение геометрических размеров транзисторов на современных кристаллах, а также снижение напряжения питания ПЛИС приводят к тому, что влияние ТЗЧ на работоспособность бортовых систем неуклонно растет.

Традиционно подобные проблемы решаются инженерами методом резервирования, которое может быть реализовано как в рамках отдельной ПЛИС, так и за счет дублирования

целых функциональных блоков в бортовой электронной системе. В первом случае разработчики модифицируют исходный RTL-код вручную, увеличивая количество блоков с одинаковой функцией и добавляя схемы контроля, позволяющие отследить корректность выполнения ими соответствующих операций. Во втором случае резервирование происходит на уровне аппаратуры, с физическим разделением однотипных блоков. Оба подхода имеют существенные недостатки: дополнительные временные затраты на разработку схемы резервирования и увеличение используемой площади ПЛИС в первом случае и существенное удорожание и увеличение массогабаритных параметров разрабатываемой аппаратуры – во втором. Для улучшения характеристик создаваемых изделий и сокращения времени их разработки целесообразно применение специализированных инструментов автоматизации проектирования, учитывающих соответствующие требования к проектам.

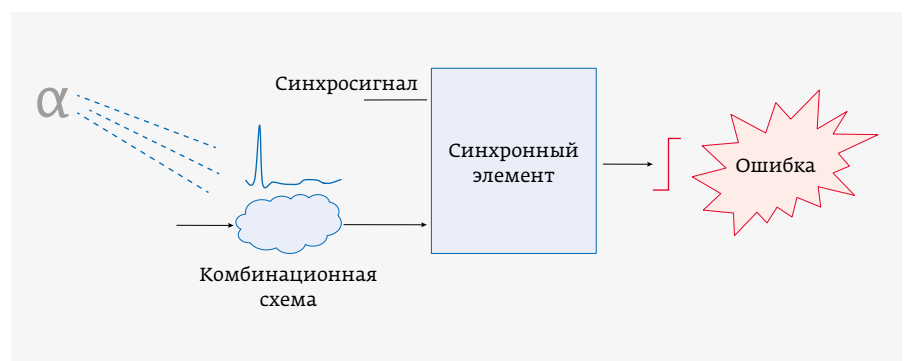


Рис. 1. Влияние ТЗЧ на электронное оборудование

¹ Компания Synopsys, старший инженер по применению.

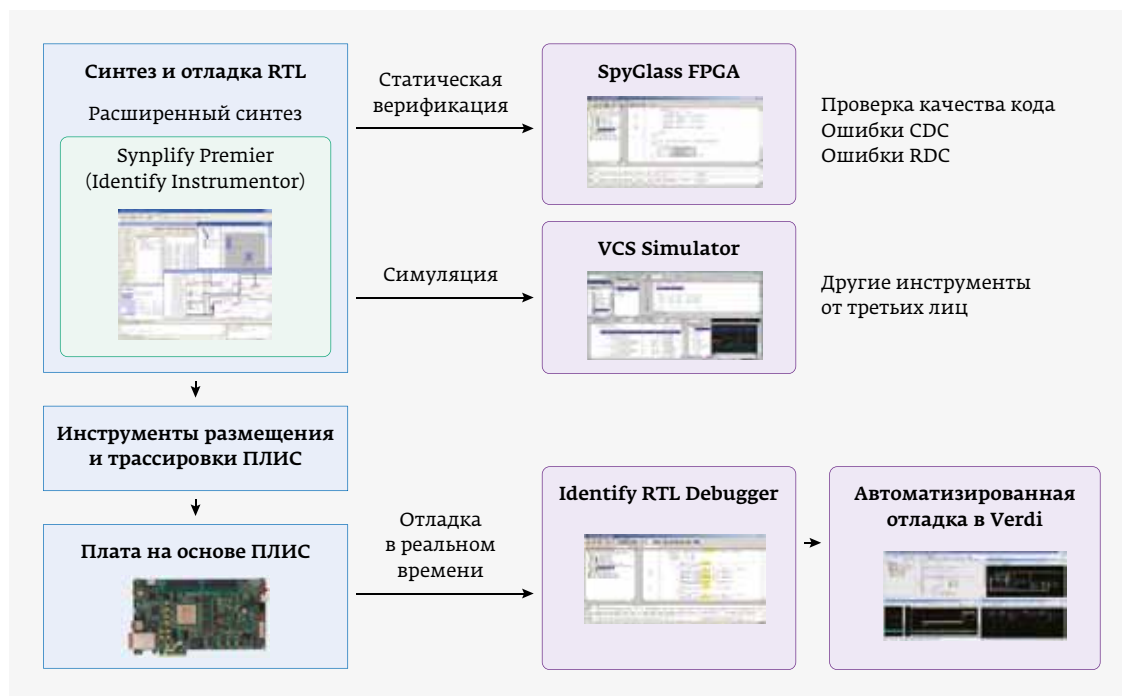


Рис. 2. Платформа Synplify Premier и ее интеграция с другими инструментами для разработки проектов ПЛИС

Инструмент Synplify Premier от компании Synopsys широко известен среди инженеров, разрабатывающих проекты под ПЛИС различной архитектуры. Он обладает рядом преимуществ перед стандартными средствами автоматизированного проектирования, предоставляемыми компаниями – производителями ПЛИС. В частности, за счет использования улучшенного алгоритма синтеза (Placement-Aware Logic Synthesis – логический синтез с учетом размещения), доступного в Synplify Premier, можно добиться лучших показателей по занимаемой площади и быстродействию разрабатываемого устройства. Кроме того, данный инструмент имеет нативную интеграцию с платформой статической верификации SpyGlass, симулятором VCS и графической средой отладки Verdi (рис. 2), что существенно ускоряет процесс разработки, отладки и интеграции устройства, выполненного на базе ПЛИС. В системы, которые предполагается использовать

в условиях воздействия ТЗЧ, инструмент Synplify Premier позволяет автоматически внедрять специальные механизмы, повышающие их устойчивость к данному воздействию. Речь идет об алгоритмах троирования, схемах детектирования и исправления единичных сбоев в работе аппаратуры, а также о механизмах защиты конечных автоматов от случайных сбоев. Далее в статье будут рассмотрены все указанные выше способы защиты, и для ПЛИС каждого типа будет выбран наиболее оптимальный вариант защиты проекта от воздействия ТЗЧ.

На рынке представлено большое количество ПЛИС различных типов: многократно программируемые ПЛИС на основе SRAM, однократно программируемые ПЛИС (Antifuse), ПЛИС на основе флеш-памяти. Сравнение перечисленных типов ПЛИС представлено в табл. 1.

Разработчики при построении аппаратуры выбирают тип ПЛИС на основе таких параметров как емкость,

Таблица 1. Сравнение типов ПЛИС

| | ПЛИС на основе SRAM | ПЛИС на основе флеш-памяти | ПЛИС типа Antifuse |
|--------------------------------|---------------------------------------|--|--|
| Элемент хранения | SRAM-ячейка | Ячейка с плавающим затвором | Металлическое соединение |
| Реконфигурация | Быстрая, неограниченная, динамическая | Медленная, ограниченная, при выключении | Невозможна |
| Плотность | Высокая | Средняя | Средняя |
| Устойчивость к воздействию ТЗЧ | Низкая | Средняя, элемент хранения не подвержен влиянию ТЗЧ | Высокая, элемент хранения не подвержен влиянию ТЗЧ |

скорость, стоимость и встроенные системы защиты от помех. В дальнейшем для каждого типа ПЛИС применяются различные алгоритмы повышения устойчивости к ТЗЧ, связанные с конкретной технологией их реализации. Эти алгоритмы могут оказывать существенное влияние на площадь и временные характеристики устройства. Таким образом, перед разработчиком стоит задача определить, какой именно алгоритм защиты стоит применить к проекту на ПЛИС определенного типа для достижения оптимального быстродействия и устойчивости к помехам.

Существующие технологии повышения устойчивости схем к ТЗЧ можно разделить по видам функциональных блоков, к которым они применимы:

1. комбинационные схемы и секвенциальная логика;
2. конечные автоматы;
3. память RAM.

Ниже рассмотрены основные технологии, доступные в инструменте Synplify Premier для каждого вида функциональных блоков, реализованных в ПЛИС различных типов.

ТЕХНОЛОГИЯ НА ОСНОВЕ РЕЗЕРВИРОВАНИЯ

Как было сказано выше, технология резервирования активно применяется в настоящее время для повышения устойчивости космической аппаратуры к воздействию ТЗЧ. Троирование (Triple Modular Redundancy – TMR) обеспечивает устойчивость системы к внешним

воздействиям за счет утроения количества функциональных блоков и добавления схемы оценки полученных результатов (рис. 3). Подобный подход позволяет детектировать единичные сбои в работе устройства и исправлять их за счет сравнения ошибочных результатов, полученных от одного блока, с корректными результатами от двух других блоков.

ИСПОЛЬЗОВАНИЕ ТРОИРОВАНИЯ В ПЛИС ТИПА SRAM

ПЛИС на основе многократно программируемой памяти SRAM подвержены влиянию ТЗЧ в первую очередь. Самой их чувствительной частью является блок памяти, хранящий файл конфигурации, который служит для формирования матрицы соединений и логических функций устройства. Ошибка в конфигурационной памяти приводит к некорректной коммутации всей электрической схемы в ПЛИС, что фатально сказывается на функционировании электронной аппаратуры. Поэтому применение технологии троирования в проектах, реализованных на ПЛИС типа SRAM, необходимо как на уровне логики, так и на уровне регистров.

Функциональные блоки в составе системы могут иметь в своей структуре обратную связь.

В таких блоках возникновение ошибки, вызванной ТЗЧ, приводит к ее «залипанию» на неопределенное число тактов. Для обеспечения устойчивости системы к сбоям

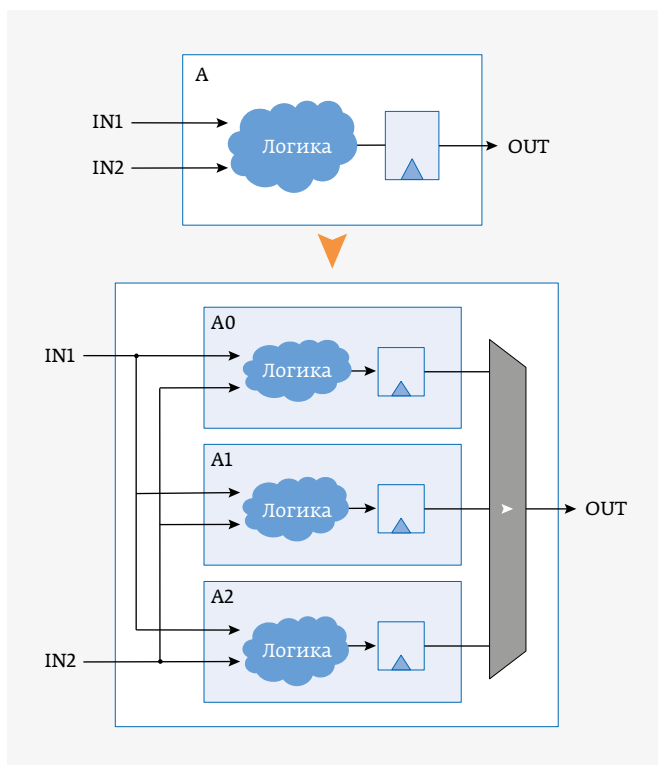


Рис. 3. Троирование

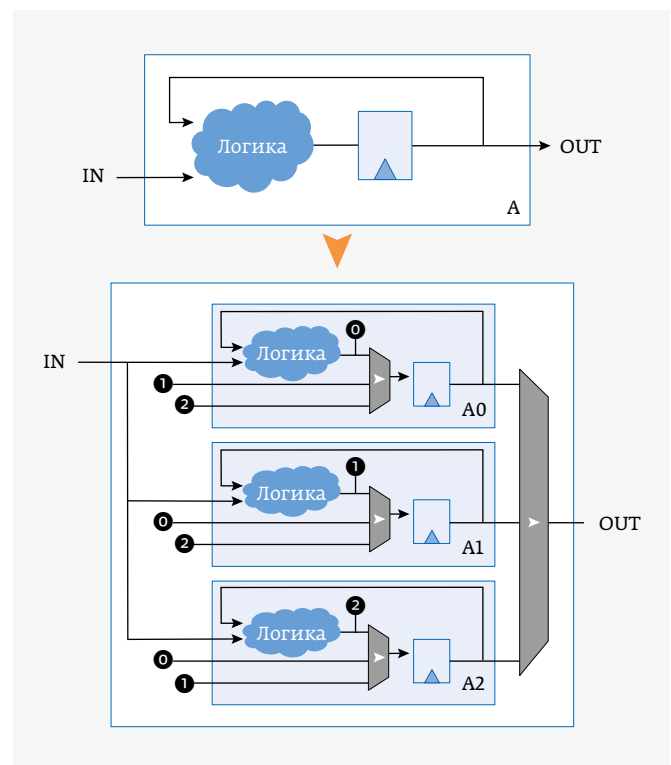


Рис. 4. DTMR в блоках с обратной связью

используются дополнительные блоки мажоритарного принятия решения на выходе триггеров, входящих в цепь обратной связи (рис. 4). Реализация технологии распределенного троирования (Distributed TMR – DTMR) в инструменте Synplify Premier позволяет разработчикам применять ее автоматически. Также инструмент дает возможность при необходимости автоматически применять троирование одновременно для нескольких блоков (рис. 5).

Теоретически разработчик с помощью инструмента Synplify Premier может добавить DTMR к каждой электрической цепи, присутствующей в проекте, однако это приводит к существенному увеличению затрачиваемых ресурсов ПЛИС, особенно если речь идет о блоках с обратной связью. Для случаев, когда размер проекта критичен и в первую очередь необходимо детектирование ошибки, а не ее автоматическое исправление, бывает достаточно блочного TMR. В этом случае выполняется утроение действующей логики и используется единственный блок мажоритарного принятия решения на уровне блока с сигналом «Ошибка» (рис. 6).

При использовании блочного TMR возможно «залипание» ошибки, способное вывести систему из строя со временем. Для предотвращения подобной ситуации должна быть разработана схема обработки ошибок, которая в случае ее возникновения, например, осуществляет

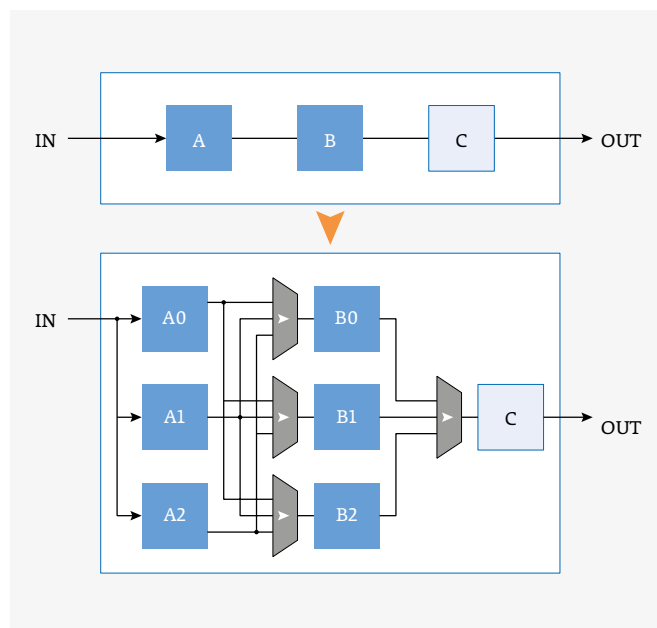


Рис. 5. Применение технологии DTMR к нескольким блокам

сброс состояния блока в исходное, устраняя тем самым данную ошибку. Инструмент Synplify Premier предоставляет возможность разработчикам автоматически

Верификация: Нахождение ошибок в СнК раньше и быстрее, ранний запуск ПО и проверка всей системы



ПЛАТФОРМА ДЛЯ ВЕРИФИКАЦИИ

Отладка, планирование и покрытие

| | | | | |
|------------------------------|--------------------------------------|---------------|----------|------------------|
| Виртуальное прототипирование | Статическая и формальная верификация | Моделирование | Эмуляция | Прототипирование |
|------------------------------|--------------------------------------|---------------|----------|------------------|

IP для верификации, модели и базы данных

- Самые быстрые движки в промышленности
- Унифицированная компиляция с VCS
- Унифицированная отладка с Verdi
- Естественная интеграция

Подробности на www.synopsys.com Москва, Смоленская площадь, дом 3 Тел: +7 (495) 933 1015

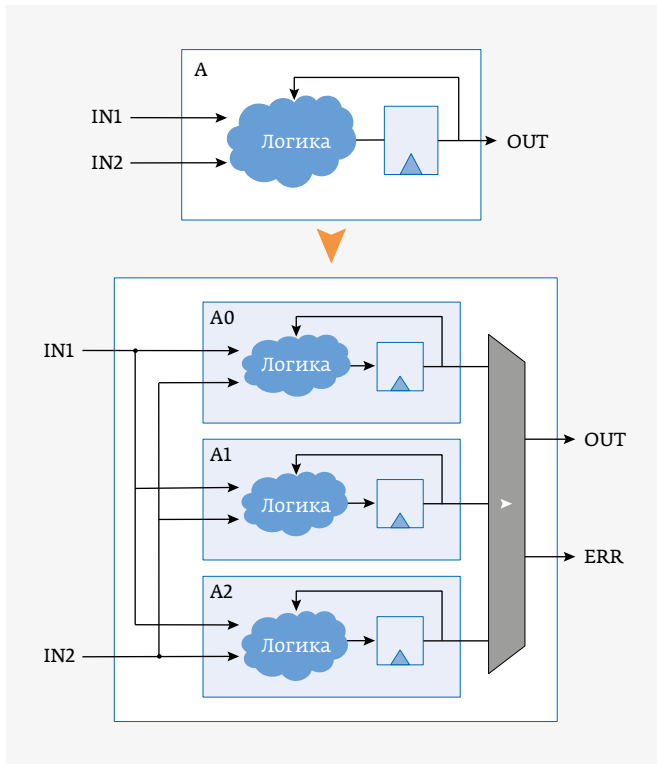


Рис. 6. Троирование на уровне блока

имплементировать схему детектирования ошибок для таких случаев, что будет рассмотрено ниже в соответствующем разделе.

ИСПОЛЬЗОВАНИЕ ТРОИРОВАНИЯ В ПЛИС ТИПА ANTIFUSE И НА ОСНОВЕ ФЛЕШ-ПАМЯТИ

ПЛИС на основе технологии Antifuse или флеш-памяти подвержены влиянию ТЗЧ на блоки конфигурационной памяти в меньшей степени, чем ПЛИС типа SRAM, или не подвержены ему вовсе. Однако проблема, связанная с влиянием индуцированной помехи на тракт тактовой частоты, для них остается актуальной. Поэтому для улучшения радиационной стойкости ПЛИС данных типов применяется локальное троирование синхронных элементов системы (рис. 7).

ОБЕСПЕЧЕНИЕ УСТОЙЧИВОСТИ К ОШИБКАМ В КОНЕЧНЫХ АВТОМАТАХ

Конечные автоматы являются критически важными компонентами систем, реализуемых на ПЛИС. Все состояния и переходы между ними должны определяться однозначно, и ни одно состояние не должно приводить к «зависанию» автомата. Однако при возникновении случайных эффектов, вызванных влиянием ТЗЧ, конечный автомат может перейти в непредсказуемое состояние, выход из которого не предусмотрен. Для того чтобы избежать подобных эффектов, в проектах следует применять

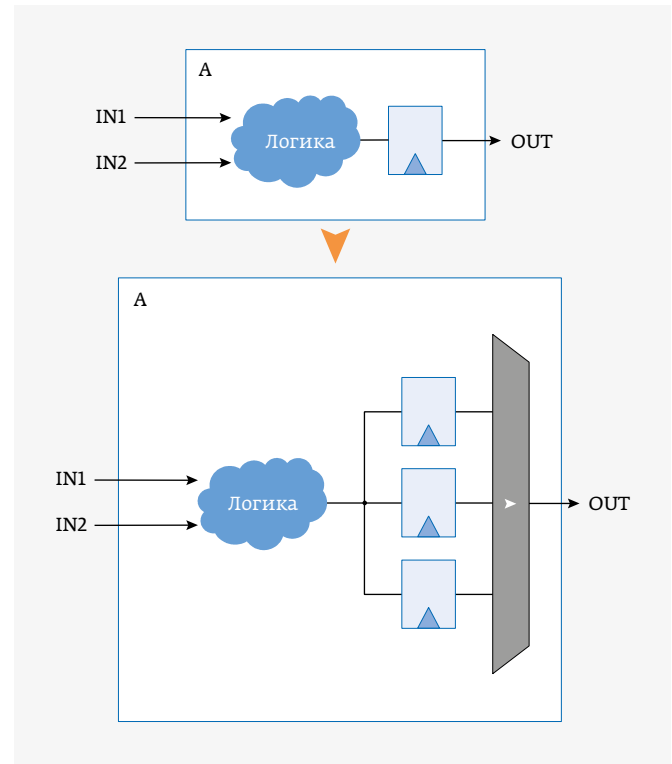


Рис. 7. Локальное троирование

специальные методы защиты конечных автоматов от нежелательных состояний.

Рассмотренная выше технология троирования может обеспечить такую защиту, однако это произойдет ценой трехкратного увеличения используемых ресурсов ПЛИС и ограничения по быстродействию. Именно поэтому в инструменте Synplify Premier реализован специализированный механизм защиты конечных автоматов от возникновения нежелательных состояний, который требует существенно меньших затрат ресурсов и не накладывает ограничений на быстродействие защищенного конечного автомата. Рассмотрим пример, приведенный на рис. 8.

```
always @ (posedge clk)
  if (rst)
    present_state <= s0;
  else
    present_state <= next_state;

always @ (in1, present_state)
  case (present_state)
    s0 : . . . . . ;
    s1 : . . . . . ;
    s2 : . . . . . ;
    s3 : . . . . . ;
  default : next_state <= s3;
  endcase
```

Рис. 8. Пример реализации конечного автомата

Для обеспечения стабильных, независимых от помех переходов в данном конечном автомате с использованием инструмента Synplify Premier можно реализовать следующие способы его защиты без непосредственной модификации исходного RTL-кода:

- механизм сброса системы в рабочее состояние в случае возникновения непредусмотренного состояния. Данный механизм применяется, если в Synplify Premier установлен атрибут *syn_encoding=safe*. В таком случае система, оказавшись в непредусмотренном состоянии из-за влияния ТЗЧ, будет принудительно возвращена в состояние *s0*;
- сброс системы в состояние *default* в случае возникновения непредусмотренного состояния. В данном случае система перейдет в состояние *s3*. При этом разработчик должен сам реализовать механизм обработки случайной ошибки в состоянии *s3*;
- использование в регистре, определяющем состояние конечного автомата, устойчивого к ошибкам кода Хэмминга (7, 4). Данный подход позволяет автоматически детектировать и исправлять единичные сбои в регистре, переводя систему в рабочее состояние.

ОБЕСПЕЧЕНИЕ УСТОЙЧИВОСТИ К ОШИБКАМ В БЛОКАХ ПАМЯТИ RAM

По аналогии с защитой конфигурационной памяти в ПЛИС типа SRAM для защиты блоков памяти, хранящих данные, может применяться технология троирования. Однако в связи с ограниченным количеством доступных блоков RAM в ПЛИС применение данной технологии не всегда возможно или целесообразно.

Во многих ПЛИС изначально реализован механизм детектирования и исправления ошибок в блоках RAM. Инструмент Synplify Premier обладает информацией о доступных технологиях повышения устойчивости к ошибкам в конкретной ПЛИС, что позволяет комбинировать базовые технологии, доступные «из коробки», и расширенные, предоставляемые инструментом Synplify Premier, для достижения лучшей защиты устройства от случайных ошибок.

ДЕТЕКТИРОВАНИЕ ОШИБОК

В ситуациях, когда разработчик ограничен в доступных ресурсах ПЛИС, применение технологии троирования становится нежелательным. Однако даже в этих случаях, если речь идет о сложных функциональных блоках в составе оборудования космического применения, детектировать ошибки необходимо для их дальнейшей обработки аппаратурой. В данном случае достаточно продублировать необходимые блоки и добавить компаратор для детектирования различия в результатах (рис. 9).

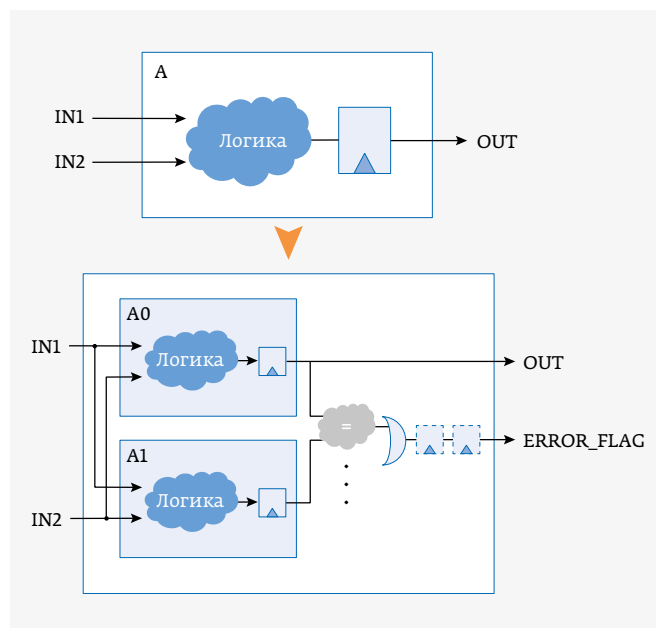


Рис. 9. Дублирование блока и имплементация компаратора

При работе с инструментом Synplify Premier необходимо лишь определить блоки, ошибки в которых требуется детектировать. Инструмент Synplify Premier во время синтеза автоматически продублирует блоки, выбранные разработчиком, и добавит соответствующий компаратор для детектирования ошибок и генерации соответствующего сигнала.

ЗАКЛЮЧЕНИЕ

Synplify Premier от компании Synopsys – не только мощный инструмент синтеза проектов ПЛИС, позволяющий уменьшить занимаемую проектом площадь и увеличить его быстродействие. Инструмент обладает рядом встроенных технологий, таких как распределенное и блочное троирование, устойчивые к ошибкам конечные автоматы, автоматическое внедрение системы детектирования ошибок на основе дублирования функциональных блоков. Эти технологии могут обеспечить повышенную устойчивость разрабатываемой системы к влиянию случайных помех, вызванных воздействием ТЗЧ, что, в свою очередь, крайне важно при создании электронной аппаратуры для космических применений. Данные технологии позволяют разработчику, комбинируя различные механизмы защиты, добиваться оптимального соотношения между занимаемой площадью, быстродействием и уровнем защищенности аппаратуры от ТЗЧ, а также благодаря их автоматическому применению существенно сокращать вероятность появления ошибок в проекте, вызванных человеческим фактором, и ускорять процесс проектирования устройства в целом. ●