Оригинальная статья
DOI: 10.21683/1729-2646-2018-18-2-3-10-17

Имитационная модель надежности резервированной вычислительной системы с периодическим восстановлением информации

Игорь В. Егоров, *Санкт-Петербургский государственный политехнический университет, Санкт-Петербург, Россия*

Игорь В. Егоров

***Резюме**. Для цифровых информационно-управляющих систем, выполненных по современной нанотехнологии, характерна повышенная чувствительность к попаданию частиц высокой энергии при работе в условиях радиации. Наиболее часто она проявляется в периодическом возникновении мягких отказов, то есть искажений информационных битов в элементах памяти системы при сохранении аппаратурой работоспособности. Причиной этому являются ложные импульсы тока на выходах логических элементов, возникающие вследствие ионизации подзатворной области полупроводника транзистора после попадания в него частицы высокой энергии. Для борьбы с мягкими отказами в систему внедряют механизмы самовосстановления, обеспечивающие периодическую перезапись искаженных данных корректными. При таком подходе к проектированию значительно возрастает значимость анализа надежности разрабатываемой системы. Так как регулярное возникновение мягких отказов является по сути штатным режимом работы системы в условиях повышенной радиации, анализ надежности необходимо многократно проводить еще на этапе ее проектирования, поскольку только таким образом можно своевременно оценить качество выбранных архитектурных решений. Однако специфика разрабатываемых отказоустойчивых программно-аппаратных систем, связанная с наличием в них детерминированных процессов восстановления, ограничивает применимость известных методов анализа надежности. Поведение этих систем затруднительно формализовать в виде модели надежности в рамках классической теории надежности, ориентированной на оценку аппаратных структур. Выявлено, что применение традиционных методов анализа надежности (таких, как использование марковской модели или логико-вероятностных методов) требует принятия ряда допущений, которые приводят к появлению недопустимых погрешностей в результатах оценки или к невозможности ее проведения. **Цель**. Разработка модели и методов анализа надежности, позволяющих оценивать надежность программно-аппаратных систем с периодическим восстановлением. **Результаты**. Разработана имитационная модель, предназначенная для оценки надежности сложных восстанавливаемых информационно-управляющих систем. Модель представляет собой сеть ориентированных графов состояний, которая позволяет описать поведение восстанавливаемой системы с учетом наличия в ней вычислительных процессов и процессов восстановления, работающих по детерминированным алгоритмам. На основании имитационной модели разработано программное средство анализа надежности, позволяющее получить вероятностную оценку характеристик надежности отдельных узлов системы и всей ее структуры в целом путем компьютерного моделирования внутренних процессов возникновения отказов и восстановлений. Данное средство может быть использовано при комплексной оценке надежности программно-аппаратных систем, которая предполагает проведение анализа восстанавливаемых узлов со сложным поведением с помощью разработанной имитационной модели, а их работу в совокупности с простыми аппаратными элементами, такими как источники питания и предохранители, – с помощью традиционных аналитических методов анализа надежности. Такой подход к оценке надежности реализован в программном средстве анализа надежности «Digitek Reliability Analyzer». **Практическая значимость**. Применение разработанных имитационной модели и средства анализа надежности на этапе проектирования позволяет своевременно оценить качество синтезируемой отказоустойчивой восстанавливаемой системы с точки зрения надежности и выбрать наилучшее архитектурное решение, что имеет большую практическую значимость.*

***Ключевые слова:** модель надежности, имитационная модель, мягкий отказ, теория надежности, восстанавливаемая система.*

***Формат цитирования:** Егоров И.В. Имитационная модель надежности резервированной вычислительной системы с периодическим восстановлением информации // Надежность. 2018. Т. 18, № 3. С. 10-17. DOI: 10.21683/1729-2646-2018-18-3-10-17*

Введение

Из работ [1–3], посвященных анализу эффектов в полупроводниковых структурах при воздействии радиации, известно, что частица высокой энергии, попавшая в МОП транзистор, может вызывать ионизацию подзатворной области полупроводника. По этой причине на выходе вентиля, в схему которого входит транзистор, возможно появление кратковременного ложного сигнала (импульса напряжения), длительность которого обычно находится в диапазоне до 1–2 нс. При переходе к современной нанотехнологии производства интегральных схем такие наведенные ложные импульсы представляют опасность, так как по своим характеристикам они сравнимы с полезными сигналами и могут привести к искажению информации в вычислительной системе.

Если ложный импульс изменяет состояние триггера или другого запоминающего элемента, то происходит событие, которое принято называть «мягким отказом» (soft error). Его суть заключается в том, что с точки зрения надежности оно может привести к отказу, так как изменение внутреннего состояния памяти системы влияет на ее функционирование. В то же время, аппаратура при этом остается работоспособной, а это значит, что состояние системы может быть восстановлено путем перезаписи искаженных данных корректными.

Исследования в области повышения радиационной стойкости информационно-управляющих систем [4–6], в частности, проведенные группой сотрудников Санкт-Петербургского политехнического университета [7–11], показывают, что именно внедрение средств периодического восстановления позволяет качественно повысить устойчивость системы к мягким отказам.

Функционирование таких самовосстанавливаемых систем имеет ряд специфических особенностей, которые сказываются на методике оценки их надежности и делают малоприменимыми традиционные методы анализа надежности [12–16]. При этом анализ надежности для данных систем играет первостепенную роль в процессе проектирования, поскольку периодическое возникновение мягких отказов, по сути, является для них штатным режимом работы. Следовательно, синтез такого рода систем невозможен без детальной оценки надежности, которая позволяет оценить качество проектируемой структуры. По этой причине разработка новых моделей и методов анализа надежности, покрывающих особенности самовосстанавливаемых систем, устойчивых к мягких отказам, является на данный момент актуальной задачей.

Традиционные аналитические модели надежности вычислительных систем и их ограничения

В течение многолетнего развития теории надежности было разработано множество моделей и методов анализа надежности. Большинство из них ориентировано на решение следующей практической задачи: при условии, что некоторая аппаратная система работает в стационарном режиме, когда с течением времени отдельные ее элементы случайным образом отказывают, оценить время наработки системы на отказ и выявить наиболее структурно важные с точки зрения надежности элементы.

Однако при анализе надежности систем с периодическим восстановлением, работающих в условиях регулярного возникновения мягких отказов, приходится учитывать следующие их особенности:

• механизм самовосстановления состояния памяти предполагает, что искаженные информационные биты будут периодически перезаписываться корректными. Очевидно, что восстановление происходит не в случайные моменты времени, а в детерминированные;

• анализируемые системы представляют собой программно-аппаратные комплексы, где зачастую именно программная составляющая (вычислительный процесс) определяет их функционирование. Аппаратная часть, в свою очередь, может рассматриваться как некоторый ресурс, который должен находиться в исправном состоянии на момент обращения вычислительного процесса к нему. Помимо основного вычислительного процесса существует и процесс восстановления, который также в определенные моменты времени требует доступа к ресурсу (памяти).

Обе вышеперечисленные особенности усложняют требования к проектированию высоконадежных систем. С одной стороны, проектировщик системы должен обеспечить как можно меньший период восстановления в каждом ее узле. С другой – процесс восстановления не должен блокировать ресурсы, необходимые основному вычислительному процессу. Эти противоречивые требования необходимо учитывать как непосредственно при проектировании (синтезе) системы, так и при анализе ее надежности: имея информацию об алгоритме работы основного вычислительного процесса, можно определить предельно допустимый период восстановления в узлах и оценить надежность проектирований системы с учетом этих условий. Если требования к надежности не соблюдены, необходимо еще на этапе проектирования модифицировать архитектуру системы в пользу применения дополнительных средств повышения надежности [7]. При таком подходе анализ надежности является инструментом синтеза отказоустойчивый системы.

На практике наиболее часто применяют следующие традиционные аналитические подходы к оценке надежности:

1) комбинаторная оценка.

Для восстанавливаемого аппаратного узла с фиксированным периодом восстановления анализируются возможные комбинации событий (отказов элементов) и влияния этих событий на другие элементы, соединенные с отказавшим. В результате строится вероятностная функция, связывающая интенсивности возникновения

отказов в элементах узла с интенсивностью отказа самого узла. Для типовых структур формула уже известна заранее, и достаточно просто подставить в нее параметры возникновения отказов и моментов восстановлений [13]. При такой оценке приходится ограничивать количество рассматриваемых событий, возникающих за период восстановления, чтобы значительно сократить количество анализируемых комбинаций и упростить тем самым итоговое выражение, что приводит к росту погрешности результатов.

2) марковская модель.

Если комбинации произошедших в системе отказов элементов выделить в состояния системы, а всем событиям отказов и восстановлений поставить в соответствие переходы между этими состояниями, то систему можно представить в виде марковской модели. В этом случае результат получается путем решения системы алгебраических уравнений Чепмена̃ Колмогорова [16]. При этом моменты всех переходов должны подчиняться экспоненциальному закону распределения случайной величины, что приводит к погрешности, которая может быть весьма значительной [12]. Также при увеличении количества элементов системы значительно возрастает число состояний в модели.

3) логико-вероятностные методы.

Известными способами строится логико-вероятностная функция работоспособности системы [17,18]. При ее построении сохраняются ограничения на закон распределения моментов отказов и восстановлений, характерные для применения марковской модели.

Поскольку все вышеописанные методы имеют ограничения применительно к анализируемым системам, приводят к погрешностям в результатах оценки, а также достаточно трудоемки при использовании на практике, целесообразна разработка программных средств, позволяющих автоматизированно оценивать надежность систем с периодическим восстановлением после мягких отказов.

Имитационная модель надежности восстанавливаемой вычислительной системы

Так как анализируемые системы обладают достаточно сложным поведением, более подходящим для оценки их надежности представляется применение имитационных моделей, нежели методов аналитической оценки. В то же время использование имитационных моделей общего назначения (таких как сети Петри) не представляется возможным на практике, поскольку требует от пользователя (проектировщика системы) значительных трудозатрат на построение адекватной модели. Более подходящей для практического применения представляется специализированная имитационная модель, которая оперировала бы такими терминами теории надежности как «отказ», «восстановление», но позволяла бы при этом моделировать поведение широкого ряда структур. Программное средство, оперирующее этой моделью, должно автоматически рассчитывать искомые характеристики надежности, такие как функция работоспособного состояния и среднее время наработки до отказа. Этот подход позволит быстро вносить изменения в имитационную модель и производить перерасчет характеристик надежности с учетом этих изменений, что особенно важно на этапе проектирования.

С этой целью автором была разработана имитационная модель надежности, основанная на представлении системы в виде ориентированного графа состояний, содержащего следующие основные элементы:

• состояния – определяются набором отказавших элементов. Каждое состояние классифицируется как работоспособное либо неработоспособное (при этом следует разделять состояния «мягкого», то есть, восстанавливаемого отказа и невосстанавливаемого отказа);

• переходы между состояниями – обычно происходят при возникновении мягких отказов, либо невосстанавливаемых отказов, либо восстановлений после мягкого отказа.

Переходы между состояниями могут происходить как в случайный момент времени, так в детерминированный. Поэтому для каждого перехода задается закон распределения случайной величины момента его появления (нормальное распределение, экспоненциальное распределение или детерминированный момент возникновения) и характеристики распределения (интенсивность возникновения события). При моделировании необходимо отличать события, привязанные к состоянию (которые возникают через определенный момент после перехода в это состояние), и не связанные с конкретным состоянием. К примеру, момент восстановления не связан с текущим состоянием, поскольку он возникает с фиксированной периодичностью независимо от момента отказа какого-либо элемента, который привел переход системы в текущее состояние. Для моделирования таких событий на уровне модели введена особая сущность, называемая генератором глобальных событий (Global Events Generator). Она содержит описание правил возникновения всех событий, не зависящих от текущего состояния системы.

Анализируемая система может содержать значительное число элементов, а каждое состояние имитационной модели в общем случае основано на совокупности состояний всех ее элементов. Это приводит к значительному росту числа состояний и усложнению модели. Для устранения этой проблемы имеется возможность описания модели не в виде единственного графа состояний, а в форме сети, состоящей из множества графов. Переходы в каждом графе этой сети могут осуществляться:

• по причине события, связанного с текущим состоянием в графе;

• при получении сигнала от глобального генератора событий;

• при возникновении события в каком-либо другом графе сети (такие события именуются внешними).

Для определения условия, при котором система считается вышедшей из строя, в модели задается параметр – функция работоспособности (Health Function). Функция работоспособности в контексте имитационной модели представлена в виде перечисления графов, которые должны находиться в работоспособном состоянии, чтобы система считалась работоспособной.

Разработанное автором программное средство, оперирующее данной имитационной моделью, функционирует следующем образом. Описание имитационной модели (сети графов) загружается из файлов в формате xml, после чего производится заданное в параметрах модели число экспериментов. В ходе каждого эксперимента моделируется возникновение случайных и детерминированных событий, описанных в модели, и измеряется время до отказа системы (момент, когда хотя бы один из графов, перечисленных в функции работоспособности, находится в неработоспособном состоянии). Поскольку в эксперименте моделируются случайные события, результат оценки также является случайной величиной. Для оценки погрешности результатов производится моделирование заданного числа экспериментов (параметр модели), на основании которых вычисляется стандартное отклонение случайной величины результата. При достаточном объеме статистики, собранной из результатов экспериментов, может быть оценена вероятностная функция работоспособности системы от времени.

Пример использования имитационной модели для оценки надежности восстанавливаемого узла вычислительной системы

В качестве примера произведем оценку времени наработки до отказа узла цифровой информационно-управляющей системы, работающего с некоторым тактовым периодом T и имеющего следующую структуру (рисунок 1):

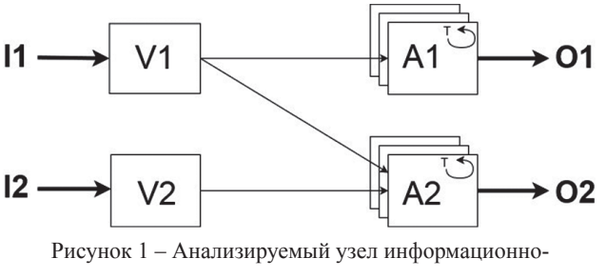

Рисунок 1 – Анализируемый узел информационно-управляющей системы

A1, A2 – функциональные модули, каждый из которых троирован и мажорирован с целью повышения надежности, а также имеет встроенные механизмы, обеспечивающие восстановление состояния модуля каждый такт (с частотой 1/T). Модули обеспечивают выставление выходных сигналов O1 и O2 соответственно.

V1, V2 – мажоритарные элементы, обеспечивающие выставление корректных входных данных I1 и I2 для элементов A1 и A2 по правилу голосования 2 из 3.

O1, O2 – выходные сигналы узла, корректность которых определяет работоспособность всего узла. Так как A1 и A2 троированы и мажорированы, данные соответствующих выходов O1 и O2 становятся некорректными, если 2 и более экземпляра модулей A1 и A2 находятся в неработоспособном состоянии. До тех пор, пока мягкому отказу подвержен только один экземпляр, модуль находится в деградировавшем состоянии, но на работоспособность системы это не влияет.

Узел подвержен потоку мягких отказов, в результате чего троированные экземпляры модулей A1 и A2 в среднем каждый сотый такт (то есть с известной частотой 1/100T) случайным образом переходят в неработоспособное состояние. В момент восстановления все деградировавшие экземпляры A1 и A2 переходят в начальное работоспособное состояние. В мажоритарных элементах мягких отказов не происходит, так как в них отсутствуют запоминающие элементы, состояние которых может быть искажено. Однако они могут быть источником кратковременных ложных импульсов (с известной частотой 1/1000T), которые, в свою очередь могут вызвать мягкий отказ в A1, A2.

Имитационная модель для данного примера имеет следующее визуальное представление (рисунок 2):

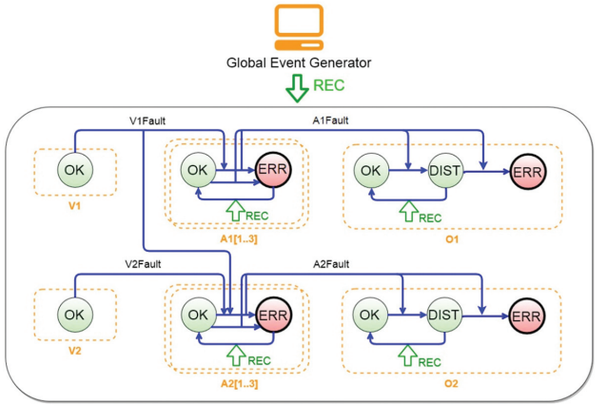

Рисунок 2 – Визуальное представление имитационной модели

На рисунке 2 пунктиром обведены отдельные графы, входящие в состав имитационной модели. Однотипные графы с идентичной структурой (троированные модули A1, A2) сгруппированы с помощью двойной пунктирной линии. В каждом графе круги, обведенные тонким контуром, обозначают работоспособные состояния, жирным – неработоспособные. Переходы между состояниями обозначены стрелками, соединяющими состояния. Стрелка, входящая в переход, обозначает условие возникновения этого перехода. В качестве нее может

выступать переход, произошедший в текущем графе, либо переход в другом графе либо событие генератора глобальных событий (широкая стрелка). Если в переходе отсутствуют входящие стрелки, это означает, что он зависит только от текущего состояния в графе и не управляется никакими внешними событиями.

Рассмотрим принцип функционирования модели на примере блока A1. Примитивный граф V1, содержащий единственное состояние OK, моделирует работу мажоритара V1, который является источником отказа V1Fault, влияющего как на модуль A1, так и на A2. Этот переход в свою очередь порождает событие A1Fault. При этом в графе O1, моделирующем состояние выходной линии O1, происходит переход в состояние DIST, свидетельствующее о том, что один из троированных экземпляров A1 неработоспособен. Отметим, что после этого один экземпляр графа A1, в котором произошел отказ, продолжает хранить неработоспособное состояние ERR и перестает быть источником отказов для O1. Возникновение события восстановления REC переводит в состояние OK как неработоспособный экземпляр A1, так и O1. Отказ O1 произойдет только в том случае, если событие A1Fault возникнет дважды за период восстановления, другими словами, если два различных экземпляра модуля A1 перейдут в неработоспособное состояние в течение периода восстановления. Модель событий для модуля A2 строится аналогичным образом.

Для получения численных оценок в модели необходимо задать параметры переходов и глобальных событий. В текстовом описании модели (в формате xml) это делается следующим образом:

```
   <topLevelDescription>
      <globalEvents>
          <event eventName="REC"
distributionType=
             "CONSTANT" intensity="1.0"/>
      </globalEvents>
      <healthFunction parameters="O1,O2"/>
      <graphs>
          <graph filePath="voter1.gr"
           graphName="V1"/>
          <graph filePath="voter2.gr"
          graphName="V2"/>
          <graph filePath="A1.gr"
          graphName="A1[1..3]"/>
          <graph filePath="A2.gr"
          graphName="A2[1..3]"/>
          <graph filePath="O1.gr"
          graphName="O1"/>
          <graph filePath="O2.gr"
          graphName="O2"/>
      </graphs>
  </topLevelDescription>
```

Секция globalEvents описывает события, порождаемые глобальным генератором событий. Каждое событие (это справедливо не только для глобальных событий, но и для переходов в графе) определяется записью event, которая содержит следующие параметры:

• eventName – имя данного события в модели;

• distributionType – закон распределения случайной величины момента его появления (CONSTANT – детерминированный с заданной периодичностью, EXPONENTIAL – экспоненциальное с заданной интенсивностью, GAUSSIAN – нормальное с заданной интенсивностью);

• intensity – задает интенсивность возникновения событий, распределенных по экспоненциальному и нормальному закону распределения, для детерминированных определяет фиксированный период между возникновением событий.

Запись healthFunction определяет функцию работоспособности. В ее единственном параметре (parameters) через запятую перечисляются названия графов, которые должны находиться в работоспособном состоянии для того, чтобы система считалась работоспособной.

Секция graphs задает перечень графов, включенных в имитационную модель. Каждому графу соответствует запись типа graph с параметрами filePath (путь к xml-файлу, содержащему описание графа) и graphName (имя графа). В случае, когда модель содержит несколько идентичных графов (в рассматриваемом примере это – троированные модули A1 и A2), имеется возможность указать в качестве имени графа массив вида A1[1..3], в результате чего в модели фактически будут использованы 3 графа с именами A1[1], A[2], A[3].

С учетом параметров отказов, использованных в данном примере (интенсивность мягких отказов модулей A1 и A2 равна 1/100T), и принимая за единицу измерения один такт работы модулей, описание графа A1 выглядит следующим образом:

```
<description>
    <states>
        <state name="OK" isfail="false"
         initialProbability="1.0"/>
        <state name="ERR" isfail="true"
         initialProbability="0.0"/>
    </states>
    <links>
        <link firstNode="ERR"
         lastNode="OK" eventName="REC"/>
        <link firstNode="OK" lastNode=
         "ERR" eventName="V11Fault"
         generateBefore="A1Fault"/>
        <link firstNode="OK" lastNode=
         "ERR" intensity="0.01"
         distributionType="EXPONENTIAL"
         generateBefore="A1Fault"/>
    </links>
</description>
```

Описание состоит из перечня состояний графа (states) и переходов (links). Каждое состояние имеет имя (name), признак того, является ли оно работоспособным (isFail) и вероятность нахождения графа в этом состоянии на

момент начала моделирования (сумма этих вероятностей для всех состояний графа должна равняться 1). Каждый переход (link) имеет те же параметры, что и событие (event) глобального генератора событий, а дополнительно к этому в нем определены исходящее (firstNode) и входящее (lastNode) состояния и название внешнего события, которое дополнительно генерируется при этом переходе (в параметре generateBefore, если внешнее событие необходимо сгенерировать до осуществления перехода в текущем графе, либо в параметре generateAfter, если внешнее событие должно возникнуть после перехода). Названия состояний и событий, использованных в описании модели, соответствуют использованным на рисунке 2.

Запустив процедуру расчета для 1000 экспериментов, получаем следующий результат (в тактах работы устройства):

Среднее время наработки до отказа = 1202,5 (такта);

Погрешность результата: ± 37,3 (такта).

Таким образом, была произведена оценка среднего времени наработки до отказа узла вычислительной системы, состоящего из восстанавливаемых структурных блоков. Сравнение качества результатов анализа надежности восстанавливаемых узлов, полученных с помощью имитационной модели и традиционных методов анализа надежности, рассмотрено в работе [12].

Рассмотренная имитационная модель применима для оценки восстанавливаемых узлов со сложным поведением и восстановлением и внедрена в программное средство анализа надежности «Digitek Reliability Analyzer» [19]. В то же время, программно-аппаратная система помимо таких узлов содержит базовые блоки, такие как элементы питания, тактовые генераторы, предохранители и т.д. Оценку влияния этих элементов на надежность удобнее производить традиционными методами. Поэтому при использовании средства «Digitek Reliability Analyzer» предлагается производить анализ сложных восстанавливаемых узлов с помощью имитационной модели, а затем использовать модель надежности верхнего уровня, содержащую как базовые аппаратные элементы, так и сложные восстанавливаемые узлы, представленные в виде «черного ящика». В качестве входных параметров для них задаются характеристики надежности, предварительно рассчитанные с помощью имитационной модели. Расчет надежности с помощью модели верхнего уровня в «Digitek Reliability Analyzer» производится с помощью логико-вероятных методов и позволяет добиться точных аналитических оценок.

В качестве подобного примера рассмотрим систему, содержащую дублированный восстанавливаемый узел (рисунок 1), надежность которого была оценена выше с помощью имитационной модели, и связанные с ним аппаратные узлы. Структурная схема надежности устройства, разработанная в программном средстве «Digitek Reliability Analyzer», выглядит следующим образом (рисунок 3).

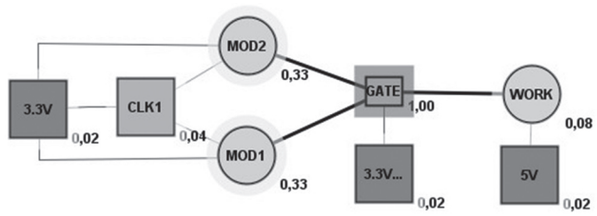

Рисунок 3 – Структурная схема надежности информационной системы

Структурная схема (рисунок 3) содержит два экземпляра проанализированного выше восстанавливаемого узла (MOD1, MOD2) с подключенными к ним источником питания (3,3V) и тактовым генератором (CLK1). Выходы MOD1 и MOD2 подключены к коммутатору GATE, который обеспечивает выставление корректных данных конечной рабочей станции (WORK), пока хотя бы один из модулей MOD1, MOD2 функционирует. Элементу GATE для работы требуется источник питания (3,3V...). Система считается работоспособной, пока функционирует рабочая станция WORK, для работы которой требуется наличие неискаженных данных на информационной линии, идущей от коммутатора (GATE), исправность элемента питания 5V и отсутствие собственных внутренних отказов.

Для каждого элемента структурной схемы задаются параметры его собственных внутренних отказов. Для элементов MOD1 и MOD2 используются значения, рассчитанные с помощью имитационной модели (среднее время наработки до отказа MOD1 и MOD2 равно 1202,5 такта). Надежность тактового генератора и источников питания известна из спецификации на соответствующее оборудование. Далее с помощью программного средства «Digitek Reliability Analyzer» автоматически рассчитывается вероятностная функция работоспособности системы P(t), график которой изображен на рисунке 4 (время t выражено в количестве тактов работы модулей MOD1, MOD2).

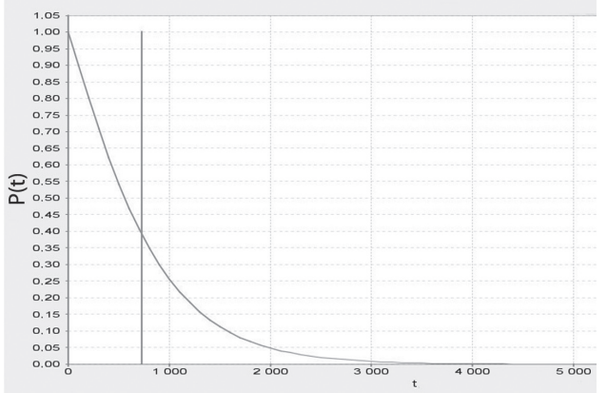

Рисунок 4 – Рассчитанная функция работоспособности информационной системы

Вертикальной линией на рисунке 4 отмечено среднее время работы системы до отказа (приблизительно 725

тактов). Для оценки вклада отдельных элементов в это значение, программное средство измеряет структурную значимость каждого из них. Она отображается рядом с правым нижним углом элемента (рисунок 3), лежит в интервале от 0 (самый малозначимый с точки зрения надежности элемент) до 1 (наиболее значимый с точки зрения надежности элемент) и зависит от текущего времени и введенных характеристик надежности элементов. Чем больше значение структурной значимости, тем больший «прирост» надежности системы дает увеличение надежности этого элемента. Для рассматриваемого примера (рисунок 3) наибольшей структурной значимостью, равной 1, обладает коммутатор (GATE). Восстанавливаемые модули MOD1, MOD2 имеют в три раза меньшую структурную значимость (0,33). Это означает, что для большего повышения надежности системы, скорее всего, следует в первую очередь повысить отказоустойчивость коммутатора. После этого необходимо произвести пересчет характеристик надежности и оценить полученный результат: среднее время наработки до отказа должно возрасти, а структурная значимость элементов – перераспределиться (коммутатор перестанет быть наиболее важным элементом). Эти сведения дают проектировщику возможность оценить качество текущего решения и выбрать дальнейшее направление по улучшению надежности путем модификации архитектуры системы.

Заключение

Разработанная автором имитационная модель надежности структурно-сложной системы позволяет автоматизированно оценивать надежность восстанавливаемых программно-аппаратных систем, обладающих сложным алгоритмом функционирования. Особенно актуально ее использование при проектировании информационно-управляющих систем, работающих в условиях регулярного возникновения мягких отказов (к примеру, вследствие неблагоприятной радиационной обстановки).

Разработанная имитационная модель позволяет описывать реакции системы на возникновение случайных событий – отказов (невосстанавливаемых и восстанавливаемых) в ее элементах, а также неслучайных событий, возникающих в соответствии с алгоритмом вычислительного процесса или в результате работы встроенных механизмов самовосстановления. Имитационная модель обладает достаточным уровнем абстракции для описания широкого круга систем. В то же время формат ее хранения позволяет разрабатывать пользовательские представления модели, более удобные для восприятия специалистом – проектировщиком цифровых систем.

Использование имитационной модели для оценки надежности наиболее сложных узлов совместно с применением известных аналитических методов для анализа надежности общей структуры системы позволяет облегчить процесс проектирования высоконадежных радиационно-стойких систем, шаг за шагом предоставляя разработчику системы информацию, необходимую для выбора лучшей архитектуры, удовлетворяющей заданным требованиям к надежности.

Библиографический список

1. Edmonds, D.L. An introduction to space radiation effects on microelectronics [Текст] / D.L. Edmonds, C.E. Barnes, L.Z. Scheick // Pasadena, USA: NASA, Jet propulsion laboratory, California institute of technology, 2000.

2. Amusan, O.A. Single event upsets in deep-submicrometer technologies due to charge sharing [Текст] / O.A. Amusan et al. // IEEE Transactions on Device and Materials Reliability. – 2008. – Vol. 8. – No. 3. – P. 582–589.

3. Жаднов, В.В. Прогнозирование показателей надежности бортовой аппаратуры космических аппаратов при воздействии ионизирующих излучений низкой интенсивности [Текст] / В.В. Жаднов, М.А. Артюхова // Надежность. – 2015. – № 1 (52). – С. 13–18.

4. Бочков, К.А. Механизмы и вероятности функциональных сбоев микроэлектронной элементной базы под действием импульсных электромагнитных помех [Текст] / К.А. Бочков, Д.В. Комнатный // Надежность. – 2015. – № 3 (54). – С. 65–72.

5. Rollins, N. Evaluating TMR Techniques in the Presence of Single Event Upsets [Текст] / N. Rollins et al. // Washington DC (USA). – 2003. – P. 1–5.

6. Егоров, И.В. Анализ проблемы повышения радиационной стойкости информационно-управляющих систем на этапе функционально-логического проектирования [Текст] / И.В. Егоров, В.Ф. Мелехин // Информационно-управляющие системы. – 2016. – № 1(80). – С. 26–31.

7. Егоров, И.В. Анализ процессов в конечном автомате при воздействии радиации. Оценка вероятности искажения информации [Текст] / И.В. Егоров, В.Ф. Мелехин // Информационно-управляющие системы. – 2016. – № 3 (82). – С. 24–33.

8. Егоров, И.В. Анализ показателей надежности и сложности реализации различных вариантов структур автомата с памятью при потоке мягких отказов [Текст] / И.В. Егоров, В.Ф. Мелехин // Информационно-управляющие системы. – 2017. – № 3(88). – С. 34–46.

9. Максименко, С.Л. Методология проектирования восстанавливаемых встраиваемых вычислительных систем [Текст] / С.Л. Максименко и др. // Университетский научный журнал. – 2014. – № 8. – С. 144–153.

10. Максименко, С.Л. Анализ проблемы построения радиационно-стойких информационно-управляющих систем [Текст] / С.Л. Максименко, В.Ф. Мелехин, А.С. Филиппов // Информационно-управляющие системы. – 2012. – № 2 (57). – С. 18–25.

11. Егоров, И.В. Методы и средства анализа надежности структурных блоков с резервированием и периодическим восстановлением информации на различных этапах

проектирования вычислительных систем [Текст] / И.В. Егоров, В.Ф. Мелехин // Информационно-Управляющие Системы. – 2016. – № 2 (81). – С. 26–34.

12. Максименко, С.Л. Анализ надежности функциональных узлов цифровых СБИС со структурным резервированием и периодическим восстановлением работоспособного состояния [Текст] / С.Л. Максименко, В.Ф. Мелехин // Информационно-Управляющие Системы. – 2013. – № 2 (63). – С. 18–23.

13. Глухих, М.И. Расчет показателей надежности по модели структуры вычислительной системы [Текст] / М.И. Глухих // Вычислительные, измерительные и управляющие системы: сборник научных трудов / под ред. Ю.Б. Сениченкова. – СПбГПУ, СПб. – С. 57–64.

14. Черкесов, Г.Н. Надёжность аппаратно-программных комплексов [Текст]: Учебное пособие / Г.Н. Черкесов. – СПб: Питер, 2005. – 479 с.

15. Перегуда, А.И. Математическая модель надежности комплекса “объект защиты – система безопасности” при нечеткой исходной информации [Текст] / А.И. Перегуда // Надежность. – 2014. – № 1 (48). – С. 99–113.

16. Черкесов, Г.Н. Логико-вероятностные методы расчета надежности структурно-сложных систем [Текст] / Г.Н. Черкесов, А.С. Можаев // Качество и надежность изделий. – 1991. – № 3 (15). – С. 3–65.

17. Хахулин, Г.Ф. Имитационная модель надежностной структуры летательных аппаратов военного назначения и ее использование в задачах исследования процессов их послепродажного обслуживания [Текст] / Г.Ф. Хахулин, Ю.П. Титов // Надежность. – 2014. – № 3 (50). – С. 3–15.

18. Digitek Labs. Reliability Analysis [Electronic resource]. URL: http://www.digiteklabs.ru/en/research/relanalyzer (accessed: 17.03.2018).

Сведения об авторе

Игорь В. Егоров – аспирант кафедры компьютерных систем и программных технологий института компьютерных наук и технологий Санкт-Петербургского государственного политехнического университета, Санкт-Петербург, Россия, e-mail: ig-ego@mail.ru

Поступила: 03.04.2018