

АНАЛИЗ МЕТОДОВ МОДЕЛИРОВАНИЯ НЕИСПРАВНОСТЕЙ ЦИФРОВЫХ УСТРОЙСТВ, ОПИСАННЫХ НА ЯЗЫКЕ VHDL

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Пурхади Амин Джамшид

Золоторевич Л.А. – к. т. н., доцент

Рассматриваются подходы к моделированию неисправностей СБИС и устройств цифровой электроники, представленных на языке VHDL. Дается анализ эффективности практических решений.

Моделирование неисправностей цифровых устройств и систем является базовой задачей при анализе контролирующей способности тестов, при построении тестов контроля на разных этапах проектирования, анализе контролепригодности объектов проектирования. Традиционно данная задача решалась в классе константных неисправностей применительно к структурным представлениям цифровых устройств на основе интерпретационных и компилятивных моделей. При этом известны практические реализации на основе эффективных методов, основанных на параллельном и сквозном моделировании неисправностей [1]. Применение языков высокого уровня для описания и моделирования изделий электронной техники поставило задачу моделирования неисправностей объектов, представленных на языке VHDL или Verilog в структурном виде или на уровне межрегистровых передач (RTL). Эффективное решение задачи в такой постановке требует разработки компиляторов с языка описания, так как внутреннее представление проекта при использовании фирменных компиляторов не доступно для внешнего использования, что существенно усложняет задачу. В последнее время актуальность проблемы разработки тестов и анализа их полноты продолжает повышаться, а поиск эффективных решений задачи анализа функционирования устройств при наличии неисправностей при нисходящем проектировании на основе VHDL идет по двум различным направлениям. Одно из развиваемых направлений исследований основано на применении аппаратных прототипов проектируемых устройств [2,3], другое – на моделировании неисправностей [4-6]. Для сокращения времени моделирования и связанных с ним сроков проектирования в некоторых случаях предпочтение отдается созданию аппаратного прототипа. В работе [6] рассматривается создание прототипа на основе применения FPGA. При этом задача решается не на основе выполнения повторного синтеза каждого неисправного прототипа, а путем частичного изменения ресурсов устройства на основе их реконфигурации в процессе функционирования, что сокращает время проведения эксперимента.

Проектирование «толерантных к неисправностям» электронных систем требует разработки новых методов и средств построения тестов, моделирования неисправностей, а также методы оценки уровня надежности, которые также предполагают моделирование неисправностей. При этом существует необходимость решения задачи моделирования неисправностей во время всего процесса проектирования на разных уровнях представления объекта, в том числе на уровне поведенческого описания объекта, когда структура устройства не известна. Такой подход позволяет определять и решать ряд потенциальных проблем на раннем этапе проектирования, что, в свою очередь, упрощает процесс проектирования и сокращает его длительность.

В работе приводится анализ методов моделирования неисправностей цифровых объектов, описанных на языке VHDL, которые основаны на применении фирменных компиляторов, расширении возможностей языка описания, применении встроенных команд системы моделирования.

Известны два подхода к моделированию неисправностей на структурном уровне описания объектов на языке VHDL [5, 7, 8]. Первый основан на введении *диверсантов* в описание компоненты и замене исходных постоянных компонент так называемыми мутантами. Предполагается изменение (мутация) описания архитектурного тела, которое будет происходить автоматически или вручную с целью отображения функционирования устройства при появлении определенной неисправности. При этом полученные «мутанты» ведут себя идентично оригинальным компонентам, за исключением интервала времени при введении неисправности. Данный подход применяется в работе [8] и позволяет описывать неисправности широкого класса. Второй подход к введению неисправностей состоит в манипулировании переменными и сигналами модели во время процесса моделирования с использованием встроенных команд системы моделирования. В работе [5,6] описан инструментальный комплекс MESHISTO, который базируется на обоих указанных выше подходах.

В работе [7] предлагается программная система VERIFY, которая позволяет вводить неисправности в цифровую систему на различных уровнях абстракции. Она основывается на динамических мутантах, где описание неисправности является составной частью поведенческого описания компонент. Для этих целей был расширен язык VHDL с тем, чтобы можно было описать тип, частоту появления и среднее время существования неисправности. Для того, чтобы можно было вводить неисправности в соответствии с этим описанием, разработан компилятор и система моделирования. Естественный путь обмена информацией с компонентой в VHDL – это применение сигналов. Поэтому была использована концепция сигнала для описания неисправности, связанной с компонентой, и в то же время – с имитатором, чтобы активизировать неисправность в заданное время. Каждая из возможных неисправностей соответствующей компоненты может быть описана отдельным сигналом. Имеется две возможности внесения неисправностей. Первая из них – сделать сигналы введения неисправностей FIS

(fault injection signals) и их параметры видимыми для системы моделирования, включив сигналы в описание интерфейса (entity), вторая – сохранить FIS прозрачными для других компонент. В первом случае FIS должны быть описаны как порты VHDL в интерфейсе компоненты. Это требовало бы сделать FIS всех поведенческих компонент цифровой схемы видимыми на верхнем уровне иерархии системы. Для каждого из FIS должен быть задан «путь» через все уровни иерархии. В работе обеспечена полная прозрачность описания неисправностей каждой компоненты. С этой целью введен новый для синтаксиса VHDL сигнал. В данном случае FIS описываются как внутренние сигналы, имеющие расширение на два дополнительных параметра: среднее время встречаемости неисправности и ее средняя длительность, к примеру:

```
SIGNAL i_stuck_at_0: BOOLEAN INTERVAL 10000 h DURATION 5 ns.
```

В работах [9,11] приведен подход, основанный на изменении VHDL – кода описания объекта внедрением в него так называемых диверсантов. В данном случае для каждого проекта необходимо разрабатывать модель на языке VHDL, которая способна моделировать исправное устройство и устройство с периодически вносимыми неисправностями. Такой подход способен обеспечить внесение и моделирование неисправностей константного типа и обрыва в линиях связи между компонентами внедрением, к примеру, в исходный код процесса, приведенного на рис. 1. Метод не требует расширения языка описания и разработки компилятора.

```
entity error is
  port(ip,c:in BIT;
        op:inout BIT);
end error;
architecture error_arch of error is
begin
  T: process (ip,c)
  begin
    if c='0' then op<=ip;
      else op<='0','1' after 100 ns;
    end if;
  end process T;
end error_arch;
```

Рис. 1

В фирменных программах моделирования на языке VHDL имеются встроенные команды языка написания сценариев TCL, которые можно использовать для интерактивного внесения неисправностей в объект без применения корректировки исходного кода.

Задача моделирования неисправностей переносится в последнее время на уровень RTL и рассматривается в более широком классе неисправностей [12,13].

Список использованных источников:

1. Zolotorevitch L.A., Baturitsky M.A. Deduktive switch-level CMOS-VLSI fault simulation //The International conference computer-aided design of discrete devices (CAD DD'95). – V. 2. – Minsk-Szczecin, 1995. – P. 157-164.
2. J. Arlat, M. Aguera, L. Amat, Y. Crouzet, J.C. Fabre, J.-C. Laprie, E. Martins, D. Powell. Fault Injection for Dependability Validation: A Methodology and some Applications // IEEE Transactions on Software Engineering.- Vol. 16.- No. 2.- 1990.
3. J. Karlsson, P. Liden, P. Dahlgren, R. Johansson, U. Gunneflo. Using Heavy-Ion Radiation to Validate Fault- Handling Mechanisms// IEEE Micro.- Vol. 14.- No. 1. 1994.- P. 8-32.
4. T. A. Delong, B. W. Johnson, and J. A. Profeta. A fault injection technique for VHDL behavioral-level models// IEEE Design Test Comput.- Vol. 13.-1996.- P. 24–33.
5. E. Jenn, J. Arlat, M. Rimen, J. Ohlsson, and J. Karlsson. Fault injection into VHDL models: The MEFISTO tool // In 24th Int. Symp. Fault-Tolerant Comput. - June 1994. - P. 66–75.
6. J. Boué, P. Pétilton, and Y. Crouzet. MEFISTO-L: A VHDL-based fault injection tool for the experimental assessment of fault tolerance // In 28th FTCS.- June 1998.- P. 168–173.
7. Sieh V., Tschache O., Balbach F.. VERIFY: evaluation of reliability using VHDL-models with embedded fault descriptions // Proc. 27th Int. symp. on fault-tolerant comp., (FTCS-27). – Chicago, June 1997. – P. 32-36.
8. Goswami K. K., Iyer, R. K. A simulation-based study of a triple modular redundant system using DEPEND // Proc. of the 5th Int. conference on fault- tolerant computing systems. – Paris, 1991. – P. 300-311.
9. Золоторевич, Л.А. Моделирование неисправностей в структурах СБИС на VHDL/Л.А. Золоторевич // Информатика. – 2005. - №1. – С. 89 – 94.
10. Золоторевич, Л.А. Моделирование неисправностей СБИС на поведенческом уровне на языке VHDL /Л.А. Золоторевич // Информатика. - 2005. - №3. - С. 135-144.
11. Иванюк, А.А. Моделирование функциональных неисправностей цифровых устройств средствами языка VHDL/ А.А. Иванюк // Информатика. - 2007. - №1. - С. 31-39.
12. Zolotorevitch, L.A. Development of tests for VLSI circuit testability at the upper design levels / L.A. Zolotorevitch, A. V. Il'inkova // Automation and Remote Control. – USA, NY, Plenum Press. – Vol. 71 Issue 9. – September 2010. – P. 1888-1898.