

анализатором, порождаемым полиномом $M_1M_3M_5$ - образующим примитивного БЧХ-кода, исправляющего три ошибки, но имеют одинаковые границы достоверности. Действительно, если в качестве примитивного полинома взять полином $M_1=x^7+x^3+1$, то минимальные многочлены M_3, M_5, M_9 соответствующего элемента поля $GF(2^m)$ имеют вид: $M_3=x^7+x^3+x^2+x+1$, $M_5=x^7+x^4+x^3+x^2+1$, $M_9=x^7+x^5+x^4+x^3+x^2+x+1$. Заметим, что обширные таблицы примитивных полиномов, необходимые для построения таких сигнатурных анализаторов, приведены, например в [2].

Литература

1. Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А. Теория кодов, исправляющих ошибки: Пер. с англ. - М.: Связь, 1979. - 744 с.
2. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки: Пер. с англ. - М.: Мир, 1976. - 594 с.
3. Ярмолик В. Н. Контроль и диагностика цифровых узлов ЭВМ. - Мн.: Наука и техника, 1988. - 240 с.

ОБЗОР НОВЫХ СПОСОБОВ ТЕСТИРОВАНИЯ FPGA

Данилов М.А.

Брестский политехнический институт.

Введение

В последние годы наблюдается стремительный прорыв в области микроэлектронных устройств на основе регулярных структур, таких как Field Programmable Gate Arrays (FPGAs). В связи с этим остро стоит вопрос тестирования и диагностирования таких изделий. В данной статье содержится

обзор последних достижений в области тестирования и диагностирования FPGAs.

1. Постановка задачи.

FPGAs представляют собой сложные изделия, состоящие из большого числа однородных ячеек, называемых логическими блоками. Поэтому тестирование возможно в трех случаях:

- a) тестирование FPGAs на заводе изготовителе;
- b) тестирование неконфигурированных FPGAs перед программированием пользователем;
- c) тестирование во время работы.

2. Обзор известных способов тестирования.

Первые два случая а) и б) охватывает метод граничного сканирования (Boundary Scan) [1], который является стандартом IEEE 1149.1-1990. Он заключается в изготовлении на кристалле дополнительных узлов, производящих под управлением специализированной микросхемы тестирование FPGA. Данный стандарт описывает интерфейс между тестируемой FPGA и специализированной микросхемой BOUNDARY SCAN MASTER фирмы Lucent Technologies.

Задача тестирования неконфигурированных FPGAs перед программированием их пользователем сводится к конфигурированию тестируемой микросхемы таким образом, чтобы минимальным количеством тестовых последовательностей и в минимальные сроки произвести исчерпывающее тестирование изделия.

Данный метод рассматривается в [2], где авторы, беря за основу FPGA XILINX 4000, предлагают разделять логические блоки данных FPGA на комбинационную и последовательностную схемы с целью отдельного их

тестирования. Кроме того, предлагаются минимальные тестовые наборы для тестирования всех составных частей логических блоков, а также 8 способов конфигурирования логических блоков (4 для комбинационной части и 4 для последовательностной), позволяющие производить исчерпывающее тестирование отдельного логического блока. Авторы предлагают также способ организации массива логических блоков, для наиболее оптимального приложения тестовых воздействий и транспортировки результатов на внешние выводы.

Другим направлением в тестировании неконфигурированных FPGA является применение самотестирования [3]. В этой области выделяются два возможных варианта: самотестирование при помощи дополнительных схем на кристалле и самотестирование, при котором задействуются только стандартные логические блоки FPGA. Второй вариант представляется наиболее перспективным, поскольку позволяет экономить около 33% места на кристалле. Идея метода заключается в такой организации логических блоков, при которой они делятся на три вида: генератор тестовых воздействий, анализатор ответов и тестируемые логические блоки. Для исчерпывающего тестирования после одного прохода тестирования логические блоки меняются функциями: генератор тестовых воздействий становится тестируемым блоком, анализатор ответов становится генератором тестовых воздействий, а один из тестируемых ранее логических блоков становится анализатором ответов. Для диагностирования двухмерной матрицы логических блоков выполняется сначала тестирование по строкам, затем по столбцам.

В случае с) тестирование во время работы достигается путем специального проектирования логических ячеек [4]. Каждая логическая ячейка кроме основного сигнала выхода (результат функции) генерирует сигнал, комплиментарный основному. Затем специально спроектированная ячейка проверяет комплиментарность этих сигналов. При каскадировании в случае сложных вычислений на вход последующей ячейки поступают и прямой и комплиментарный выходные сигналы предыдущей, транспортируя тем самым сигнал ошибки на ячейку проверки. Авторы приводят пример проек-

тирования схемы путем декомпозиции исходной функции. Очевидно, что данный метод требует дополнительных аппаратных ресурсов.

Заключение

Среди перспективных методов тестирования и диагностирования следует выделить непрерывное тестирование в процессе работы микросхемы, а также метод самотестирования и диагностирования на основе стандартных логических блоков. Оптимальным видится метод совмещения непрерывного тестирования с диагностированием, в случае сбоев, по методу [3].

Литература

1. Lucent Technologies "Field Programmable Gate Arrays", Data Book, October 1996.
2. M. Renovell, J.M. Portal, J. Figueras, Y. Zorian "Testing Unconfigured FPGA Logic Modules" 15th IEEE VLSI Test Symposium, CA, USA, May 1997.
3. C. Stroud, E. Lee, M. Abramovici "BIST-Based Diagnostics of FPGA Logic Blocks", Proc. IEEE International Test Conference, pp. 539-547.
4. A.L. Burrell, P.K. Lala "On-Line Testable Logic Design For FPGA Implementation", Proc. IEEE International Test Conference, pp. 471-478.