

УДК 519.6

П.Н. Бибило<sup>1</sup>, А.М. Седун<sup>2</sup>**О РЕАЛИЗАЦИИ УСТРОЙСТВ ПРОВЕРКИ СИММЕТРИЧНОСТИ  
БУЛЕВЫХ ФУНКЦИЙ**

*Приводятся результаты экспериментов схемной реализации на FPGA (Field-Programmable Gate Arrays – программируемые пользователем вентильные матрицы) устройств проверки симметричности булевых функций, задаваемых векторами их значений согласно таблицам истинности.*

Возможности микроэлектронной технологии позволяют все чаще переходить от программной к аппаратной реализации алгоритмов. Как правило, аппаратная реализация оказывается более быстродействующей, что облегчает решение задач большой размерности. Примерами аппаратно реализованных комбинаторных алгоритмов являются криптоалгоритмы [1] и алгоритмы решения задачи «выполнимость» [2], аппаратно реализуются также трудоемкие задачи моделирования и синтеза топологии интегральных схем. Задача проверки симметричности булевой функции часто возникает при проектировании логических схем [3; 4, с. 48]. Полностью определенная булева функция  $f(x_1, x_2, \dots, x_n)$  называется симметричной, если она не изменяется при любой перестановке переменных; в противном случае – несимметричной.

Рассмотрим задачу синтеза устройства  $S$ , позволяющего определить, является ли симметричной заданная полностью определенная булева функция  $f(x_1, x_2, \dots, x_n)$ , значения которой подаются на вход этого устройства.

Для сокращения числа входных полюсов устройства  $S$  будем задавать булеву функцию двоичным вектором (массивом) – столбцом ее значений согласно таблице истинности. Для сокращения аппаратной сложности логической схемы устройство  $S$  должно проверять свойство симметричности функции за  $n!$  тактов изменения сигнала синхронизации  $Clk$ , т. е. в каждом такте функционирования устройство должно проверять, не изменилась ли функция после одной перестановки переменных.

Алгоритм проверки симметричности булевой функции состоит в последовательном переборе всех  $n!$  перестановок множества  $X = \{x_1, x_2, \dots, x_n\}$  аргументов и проверке на равенство двух функций: исходной и полученной в результате каждой конкретной перестановки аргументов. Если окажется, что для всех  $n!$  перестановок функция не изменяется, то на единственном выходном полюсе устройства  $S$  формируется значение 1.

Проектирование устройства проверки симметричности булевой функции было проведено с помощью современных средств. Так, поведение устройства было описано на языке VHDL [4], моделирование проведено в системе ModelSim, синтез устройства – в синтезаторе LeonardoSpectrum [4]. Разработанный VHDL-алгоритм проверки функции на симметричность является параметризованным – настраиваемым на заданное число  $n$  аргументов функции – и использует представленный в [5, с. 185] алгоритм порождения перестановок номеров аргументов функции в лексикографическом порядке. Пусть  $\Pi = (\pi_1, \pi_2, \dots, \pi_n)$  – начальная перестановка множества  $X = \{x_1, x_2, \dots, x_n\}$  аргументов булевой функции  $f(x_1, x_2, \dots, x_n)$ . Для получения последующей перестановки из перестановки  $\Pi$  справа налево просматривается перестановка  $\Pi$  и находится самая правая позиция, в которой  $\pi_i < \pi_{i+1}$ . Если такая позиция  $i$  найдена, то ищется  $\pi_j$  – такой наименьший элемент, который расположен справа от  $\pi_i$  и больший его. Затем осуществляется транспозиция элементов  $\pi_i$  и  $\pi_j$  и отрезок  $\pi_{i+1}, \dots, \pi_n$ , элементы которого расположены в порядке убывания, переворачивается. Результатом последнего действия является следующая перестановка.

Например, пусть имеется перестановка (5, 4, 7, 6, 3, 2, 1) для  $n = 7$ . Если просматривать ее справа налево, то  $\pi_i = 4$ ,  $\pi_{i+1} = 7$ . Затем ищется справа от  $\pi_i = 4$  наименьший элемент, который больше  $\pi_j$ ; этим элементом является  $\pi_j = 6$ . Осуществляется транспозиция элементов  $\pi_i = 4$ ,  $\pi_j = 6$ ; получается перестановка (5, 6, 7, 4, 3, 2, 1). После этого отрезок  $\pi_{i+1}, \dots, \pi_n$ , равный (7, 4, 3, 2, 1), переворачивается, в результате получается следующая перестановка (5, 6, 7, 4, 3, 2, 1).

Особенностью алгоритма является то, что в процессе его работы каждый раз по переднему фронту сигнала *Clk* генерируется следующая перестановка. Поэтому, если тактовых импульсов будет больше  $n!$ , алгоритм не останавливается, а генерирует очередную перестановку и дальше проверяет функцию на симметричность, т. е. алгоритм проверки симметричности является циклическим с длиной цикла  $n!$ . Чтобы проверить функцию на симметричность, требуется  $n!$  раз нулевое значение сигнала *Clk* изменить на единичное. Сброс устройства (переход на начало цикла генерации перестановок) осуществляется по единичному значению сигнала *Rst*. В этот момент на вход устройства может быть подана другая функция, для которой требуется проверка на симметричность. Длина такта сигнала *Clk* определяется после синтеза схемы и должна быть больше ее задержки. Входными портами устройства являются: *Clk* – синхросигнал; *Rst* – сигнал сброса, по единичному значению которого устройство начинает перебор перестановок; *F* – массив (столбец) значений функции. Таким образом, устройство *S* имеет  $2^n + 2$  входных порта и один выходной порт *Rez*. Если *Rez* = 0, то функция несимметричная, если *Rez* = 1, то функция симметричная.

В качестве элементной базы схемной реализации устройства *S* были выбраны программируемые логические интегральные схемы типа FPGA семейства VIRTEX II PRO фирмы Xilinx [6]. Синтез схемы проводился на персональной ЭВМ с микропроцессором Pentium IV тактовой частоты 2,4 ГГц (таблица).

Результаты синтеза устройства S

$n$	Число входов схемы ( $2^n + 2$ )	Сложность схемы (LUT)	Задержка схемы (нс)	Время синтеза схемы	Число тактов моделирования ( $n!$ )
4	18	240	26	3 с	24
5	34	848	45	19 с	120
6	66	3067	60	12 мин 10 с	720
7	130	11110	65	2ч 25 мин 04 с	5040

Сложность схемы подсчитывалась в числе программируемых элементов FPGA – четырех-входовых функциональных генераторов LUT (Look Up Table) [6]. Приняв длину такта синхросигнала 70 нс, за время  $5040 \times 70$  нс = 0,0004 с на FPGA с помощью реализованного алгоритма можно определить, является ли произвольная булева функция от семи переменных симметричной либо несимметричной.

Таким образом, современная микроэлектронная база FPGA позволяет успешно реализовать аппаратным способом VHDL-алгоритм определения симметричности полностью определенных булевых функций, зависящих не более чем от семи аргументов.

### Список литературы

1. Панасенко, С.П. Алгоритм шифрования ICE / С.П. Панасенко // ИНФОРМОСТ. Радиоэлектроника и телекоммуникации. – 2006. – № 3. – С. 34–36.
2. Skliarova, I. Reconfigurable Hardware SAT Solvers: A Survey of Systems / I. Skliarova, A.V. Ferrari // IEEE Trans. on Computers. – 2004. – Vol. C-53. – № 11. – P. 1449–1461.
3. Поваров, Г.Н. О групповой инвариантности булевых функций / Г.Н. Поваров // Применение логики в науке и технике. – М. : Изд-во АН СССР, 1960. – С. 263–340.
4. Бибило, П.Н. Синтез логических схем с использованием языка VHDL / П.Н. Бибило. – М. : Солон-Р, 2002. – 384 с.
5. Рейнгольд, Э. Комбинаторные алгоритмы. Теория и практика / Э. Рейнгольд, Ю. Нивергельт, Н. Део. – М. : Мир, 1980. – 478 с.

---

6. Кузелин, О.М. Современные семейства ПЛИС фирмы Xilinx : справ. пособие / О.М. Кузелин, Д.А. Кнышев, Ю.В. Зотов. – М. : Горячая линия – Телеком, 2004. – 440 с.

Поступила 10.08.08

<sup>1</sup>Объединенный институт проблем  
информатики НАН Беларуси,  
Минск, Сурганова, 6  
e-mail: bibilo@newman.bas-net.by

<sup>2</sup>Белорусский государственный  
экономический университет,  
Минск, пр. Партизанский, 26  
e-mail: sedun@bseu.by

**P.N. Bibilo, A.M. Sedun**

**IMPLEMENTATION OF CIRCUIT DEVICES  
FOR BOOLEAN FUNCTIONS SYMMETRY TEST**

The circuit devices for verifying the symmetry of Boolean functions determined in terms of truth tables were tested. The experiments were performed on FPGA (the Field-Programmable Gate Arrays). The resultant experimental data are reported and discussed.