

ДИАГНОСТИРОВАНИЕ ПЛИС НА ОСНОВЕ МОДЕЛЕЙ КЛЕТОЧНЫХ АВТОМАТОВ

БЕРЕЖНАЯ М.А., ДЕРБУНОВИЧ Л.В., ХАХАНОВ В.И.

Предлагается метод тестирования программируемых логических устройств на стадии входного контроля для автоматизированного проектирования цифровых систем, представленных в виде конечных автоматов.

Введение

Специализированные цифровые системы на основе программируемых логических интегральных схем (ПЛИС) являются новым перспективным средством для решения задач технологического управления. Реализация таких систем предполагает использование как универсального процессора в целях управления вычислениями, так и программируемой логики, ориентированной на параллельную обработку структур данных большого объема. Наряду с низкой стоимостью кристаллов программируемых логических интегральных схем и незначительным временем синтеза структурно- и функционально-сложных вычислительных устройств на их основе существует проблема тестирования последних. Она определяется структурной и функциональной сложностью цифровой системы, имплементированной в ПЛИС, насчитывающей несколько миллионов эквивалентных вентилей на одном кристалле, и отсутствием эффективных методов их анализа в целях верификации и тестирования цифровых систем. В практике проектирования дискретных устройств широко применяются два основных типа программируемых логических интегральных схем: FPGA (Field Programmable Gate Arrays), CPLD (Complex Programmable Logic Device). Существует много архитектурных разновидностей ПЛИС, которые отличаются технологиями программирования.

На рис. 1 представлены типовые структуры FPGA (верхняя часть), CPLD (средняя) и структура типовой ячейки (нижняя часть). Каждая из первых двух есть гибкое и многофункциональное программируемое логическое устройство, позволяющее реализовывать большие цифровые системы на одной микросхеме. Типичная PGA — это микросхема, состоящая из массива идентичных логических ячеек с программируемыми соединениями. Пользователь может запрограммировать функции, реализуемые каждой логической ячейкой, и соединения между ними. Поэтому они являются программируемыми в условиях эксплуатации (field-programmable). Структура ПЛИС типа FPGA представляет собой двухмерную сеть, состоящую из (N×N) конфигурируемых логических блоков (КЛБ), которые имеют от 3 до 5 адресных входов и D-триггер, блоки ввода-вывода данных для соедине-

ния с внешними контактами и программируемую коммутационную матрицу [1].

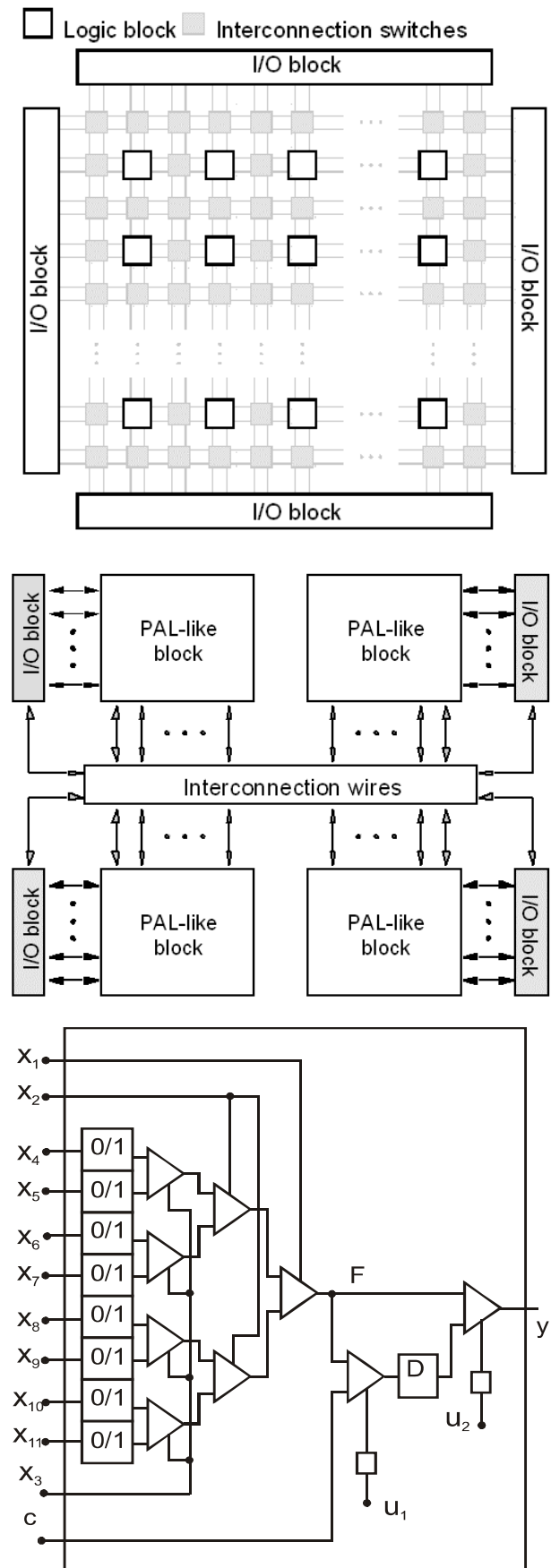


Рис. 1. Структура программируемых схем

Тестовое диагностирование FPGA проводится на этапе входного контроля перед загрузкой «теневого» программы требуемой конфигурации в целях выявления неисправных элементов ПЛИС. Процедуры тестового диагностирования FPGA проанализированы в работах [1-3], в которых решались задачи нахождения минимального числа реконфигураций ПЛИС и соответствующих тестовых последовательностей, позволяющих обнаружить константные неисправности КЛБ.

В [3] рассмотрены методы проверки коммутационных матриц FPGA, позволяющие обнаружить неисправности типа «появление» и «исчезновение» соединений при минимальном числе возможных конфигураций.

В настоящей работе предлагается конфигурировать матрицу КЛБ в виде С-тестируемой однородной структуры клеточных автоматов (СКА) с наблюдаемыми выходами, что обеспечивает исчерпывающее тестирование кратных неисправностей ячеек сети [4]. Разрабатывается методика синтеза проверяющих тестов СКА, у которых модели клеточных автоматов являются сильно связанными и имеют однородные отличительные последовательности для каждого входного символа.

Не снижая общности предложенного подхода, рассмотрим структуру FPGA, в которой каждый КЛБ имеет три адресных входа (x_1, x_2, x_3). Тогда N КЛБ можно сконфигурировать в СКА, как показано на рис. 2.

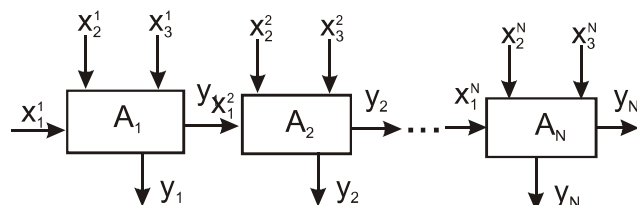


Рис. 2. Однородная структура с наблюдаемыми выходами

Каждая КЛБ A_i представляется моделью автомата Мура с двумя состояниями. Правый выход каждой ячейки, а также выходы y_i коммутируются на наблюдаемые выходы через программируемые блоки входа FPGA. Входы x_i^j ($i = 1, 2, 3; j = \overline{1, N}$) соединяются с программируемыми блоками ввода.

На функциональном уровне описание ячейки клеточного автомата будем рассматривать как таблицу переходов-выходов автомата Мура, задаваемого тройкой (x, s, δ) , у которого функции переходов и выходов совпадают:

$$\delta(s_i, x_a) = \lambda(s_i, x_a) = s_j, \forall s_i, s_j \in S, \forall x_a \in X. \quad (1)$$

Введем следующие обозначения и определения. Пусть $\alpha_1 \approx x(1)$ обозначает приложение логической переменной, набора логических переменных при наличии нескольких входных полюсов x в ячейке клеточного автомата, к входному полюсу (полюсам) $x(1)$.

Вектор $V_j = \{s_1, \alpha_1, \alpha_2, \dots, \alpha_p\}$ представляет двоичный входной набор, в котором $s_1 \approx s(1)$, $\alpha_1 \approx x(1)$, $\alpha_2 \approx x(2)$, $\alpha_p \approx x(p)$.

Вектор $V_i = \{s_1, (\alpha_1, \alpha_2, \dots, \alpha_k)^*\}$ представляет также двоичный входной набор, но в котором $s_1 \approx s(1)$, $\alpha_1 \approx x(1)$, $\alpha_2 \approx x(2)$, $\alpha_k, \alpha_1 \approx x(k+1)$. Таким образом, входные полюсы ОС соединяются с источником питания циклическими повторяющимися логическими переменными, или наборами переменных $\alpha_1, \alpha_2, \dots, \alpha_k$.

Пусть необходимо проверить правильность перехода $\delta_{ij}(s_i, x_j) = s_a$ в ячейке $C(1)$ сети, которая имеет отличительную последовательность $x(0)$. Проверяющий входной набор построим в следующем виде:

$$V(\delta_{ij}) = \{s_i, (x_j X_0 T(s_i))^*\}, \quad (2)$$

где $(R)^*$ – последовательность RRR; $T(s_i)$ – переводящая последовательность, которая переводит автомат из состояния $\delta(s_i, x_j X_0) = q_0$ в состояние s_i ; X_0 – отличительная последовательность.

Действительно, состояние s_a на выходе ячейки $C(1)$ отличается от множества других состояний S/s_a по реакции последующих ячеек, к которым приложена отличительная последовательность X_0 . Это обеспечивается наличием в сети наблюдаемых выходов y_i и свойствами отличительных последовательностей.

Пусть входной набор $x_j X_0 T(s_i)$, прикладываемый к входам X сети, состоит из k входных символов. Так как этот набор циклически повторяется и состояние k -й ячейки $s_k = \delta(x_j X_0 T(s_i)) = s_i$, то приложение $V(\delta_{ij})$ ко входам сети позволяет проверить правильность переходов δ_{ij} в ячейках $C(1), C(k+1), C(2k+1), C(3k+1), \dots$

Определение. Входной набор $V(\delta_{ij})$, определяемый выражением (2), будем называть циклической отличительной последовательностью (ЦОП) одномерной сети клеточных автоматов с наблюдаемыми выходами y_i .

Известно, что в любом сильно связном автомате с n состояниями существует множество переводящих последовательностей $T(s_i), i = \overline{1, n}$, длина которых не превышает $(n-1)$.

Из (2) верхняя граница длины ЦОП $|V(\delta_{ij})|$ определяется неравенством:

$$\left| V(\delta_{ij}) \right| \leq 1 + \frac{n(n-1)}{2} + (n-1) \leq \frac{n}{2}(n+1). \quad (3)$$

Минимальная нижняя граница длины ЦОП $\left| V(\delta_{ij}) \right|$ имеет вид

$$\left| V(\delta_{ij}) \right| \geq \left\lceil \frac{\log_2 n}{\log_2 r} \right\rceil + n, \quad (4)$$

где r – число выходных символов ячейки сети.

Тест $V(\delta_{ij})$ проверяет правильность перехода δ_{ij} в ячейках сети $A(1), A(k+1), A(2k+1), \dots$. Свойство сильносвязности автоматной модели ячейки сети упрощает процедуру нахождения множества тестов, проверяющих этот переход во всех ячейках сети. Эта процедура сводится к циклическому сдвигу входного слова $x_j X_0 T(s_j)$. При этом выполнение

$\left(\left| V(\delta_{ij}) \right| - 1 \right)$ циклов сдвига теста $V(\delta_{ij})$ позволяет получить $\left| V(\delta_{ij}) \right|$ входных ЦОП, проверяющих правильность перехода во всех ячейках сети. Тогда верхняя и нижняя границы длины полного теста, проверяющего правильность всех $(m \times n)$ переходов ячейки клеточного автомата с учетом (3) и (4), равны соответственно:

$$l(T) \geq \frac{1}{2} \cdot m \cdot n \cdot (n+1), \quad (5)$$

$$l(T) \leq m \cdot n^2 \cdot \left(\left\lceil \frac{\log_2 n}{\log_2 r} \right\rceil + n \right). \quad (6)$$

Из (5) и (6) следует, что длина проверяющего теста не зависит от размерности сети или числа N ячеек сети. Таким образом, сети рассматриваемого класса являются S -тестируемыми.

Процесс генерации тестовых последовательностей для СКА с наблюдаемыми выходами Y , каждая ячейка которой описывается ТПВ сильносвязного автомата, имеющего отличительную последовательность, можно представить следующим алгоритмом:

1. Построить таблицу переходов-выходов автомата Мура или Мили по заданной логической схеме ячейки сети. Пусть ТПВ имеет n строк и m столбцов.
2. Построить отличительное дерево преемников автомата и найти отличительную последовательность X_0 .
3. Для каждого перехода ТПВ

$$\delta_{ij}(s_i, x_j) = s_a, i = \overline{1, n}, j = \overline{1, m},$$

найти множество конечных состояний

$$\delta(s_a X_0) = s_k.$$

4. Найти множество переводящих последовательностей $T(s_k, s_i)$ по ТПВ для каждого перехода, где множество состояний $\{s_k\}$ определено на шаге 3.

5. Построить множество ЦОП $V(\delta_{ij})$ для каждого перехода автоматной диаграммы, используя X_0 и $T(s_k, s_i)$. Пусть длина ЦОП $\left| V(\delta_{ij}) \right| = t$.

6. Для каждого проверяющего теста $V(\delta_{ij})$ выполнить $(t-1)$ циклических сдвигов, которые определяют ЦОП, проверяющих правильность всех $(m \times n)$ переходов ТПВ в каждой ячейке сети.

7. Конец алгоритма.

Применение ЦОП, полученных в соответствии с предложенным алгоритмом, к сетям клеточных автоматов позволяет проверить соответствие таблицы истинности проверяемой ячейки таблице истинности исправной ячейки.

Для проверки такой сети предлагается использовать две автоматные модели Мура КЛБ. Два состояния автомата (s_0 и s_1) соответствуют состояниям Д-триггера КЛБ.

Если число адресных входов КЛБ $k = 3$, то входы (x_2^i, x_3^i) формируют 4 входных символа

$$(x_0 = 00, x_1 = 01, x_2 = 10, x_3 = 11),$$

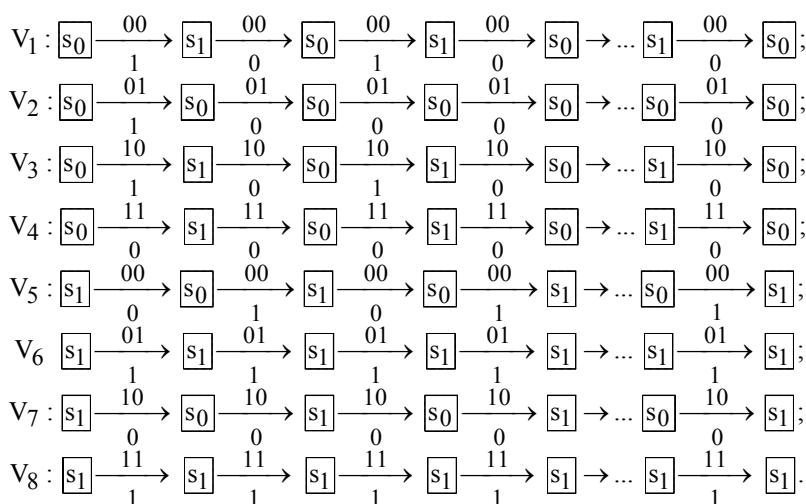
которые подаются на вертикальные входы x_i сети (см. рис. 2). Каждый входной символ является отличительным для двух конечно-автоматных моделей $A1$ и $A2$:

$$A1 = \begin{array}{c|cccc|c} \begin{array}{c} s(t+1) \rightarrow \\ s(t) \downarrow \end{array} & x_0 & x_1 & x_2 & x_3 & y(t) \\ \hline s_0 & s_0 & s_1 & s_0 & s_1 & 0 \\ \hline s_1 & s_1 & s_0 & s_1 & s_0 & 1 \end{array};$$

$$A2 = \begin{array}{c|cccc|c} \begin{array}{c} s(t+1) \rightarrow \\ s(t) \downarrow \end{array} & x_0 & x_1 & x_2 & x_3 & y(t) \\ \hline s_0 & s_1 & s_0 & s_1 & s_0 & 0 \\ \hline s_1 & s_0 & s_1 & s_0 & s_1 & 1 \end{array}.$$

Он покрывает множество функциональных неисправностей каждого КЛБ двумя конфигурациями, которые определяются этими таблицами переходов состояний.

Универсальное множество тестов, проверяющее таблицу переходов $A1$ в N клеточных автоматах, которое генерируется в соответствии с ЦОП для случая четного числа ячеек N , представлено в следующем виде:



Аналогично генерируются тестовые последовательности, проверяющие ТПВ клеточных автоматов для модели А2.

Для тестирования управляемости D-триггера по входу С необходимо использовать конфигурацию, в которой $u_1=u_2=1$ и D-триггеры образуют N-разрядный сдвиговый регистр, который проверяющийся двумя циклическими тестами (010101) и (101010).

Таким образом, достаточно использовать $n_c = 5$ конфигураций КЛБ в соответствии с автоматными моделями А1, А2 и универсальное множество тестов, чтобы обнаружить неисправные КЛБ в сети из N идентичных клеточных автоматов.

Вывод

Моделирование полноты сгенерированных тестов для класса одиночных константных неисправностей КЛБ показывает, что тесты, полученные для класса С-тестируемых регулярных структур клеточных автоматов, позволяющие обнаружить кратные неисправности сети функционального уровня, покрывают 100% одиночных константных неисправностей КЛБ.

Литература: 1. *Huang W.K., Lombardi F.* An Approach for testing Programmable/Configurable Field Programmable Gate Arrays // Proc.14th IEEE VLHI Test Symp. IEEE Computer Society Press, Los Alamitos, Calif. 1996. P. 450-455. 2. *Inoue T. et al.* Universal Test Complexity of Field – Programmable Gate Arrays // Proc. Fourth IEEE Asian Test Symp. IEEE CS Press. 1995. P. 259-265. 3. *Renovell M. et al.* Test of RAM – Based FPGAs: Methodology and Application to the Interconnect. // Proc.15th IEEE VLHI Test Symp. IEEE CS Press. 1997. P. 230-237. 4. *Дербунович Л.В.* Синтез легко тестируемых двумерных итеративных логических сетей // В кн.: Тезисы докладов. 6 Всесоюзное совещание по технической диагностике. Ростов-на-Дону. 1987. С.51. 5. *Michinisti H. et al.* A Test Methodology for Configurable Logic Blocks of Lookup Table Based FPGAs // IEICE

Trans. D-1. – December, 1996. Vol. J79 – D1, №12. P. 1141-1150. 6. *Хаханов В.И., Бережная М.А.* Двухтактное кубическое исчисление // АСУ и приборы автоматики. 1997. Вып.106. С. 103-116. 7. *Хаханов В.И., Бережная М.А.* Диагностирование одиночных и кратных неисправностей // АСУ и приборы автоматики. 1997. Вып. 104. С. 17-28.

Поступила в редколлегию 16.05.2002

Рецензент: д-р техн. наук, проф. Кривуля Г.Ф.

Бережная Марина Анатольевна, аспирант кафедры АПВТ ХНУРЭ. Научные интересы: тестопригодное проектирование цифровых устройств. Увлечения: книги, музыка. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 40-93-26. E-mail: hahanov@kture.kharkov.ua

Дербунович Леонид Викторович, д-р техн. наук, профессор кафедры автоматики и управления в технических системах НТУ “ХПИ”. Научные интересы: техническая диагностика и проектирование вычислительных устройств. Увлечения: единоборства, книги, иностранные языки. Адрес: Украина, 61166, Харьков, ул. Фрунзе, 23.

Хаханов Владимир Иванович, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика вычислительных устройств, систем, сетей и программных продуктов. Увлечения: баскетбол, футбол, горные лыжи. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 40-93-26. E-mail: hahanov@kture.kharkov.ua