УДК 004.312.02

Н.Г. КОРОБКОВ[1], Е.Н. КОРОБКОВА[2], В.Г. РУБАНОВ[2]

[1] ***Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина***
[2] ***Белгородский государственный технологический университет им. В.Г. Шухова, Россия***

СИНТЕЗ ЦИКЛИЧЕСКИХ КОНЕЧНЫХ АВТОМАТОВ С ПЕРЕСТРАИВАЕМОЙ ДЛИНОЙ ЦИКЛА, ОСНОВАННЫЙ НА ПРЕДСТАВЛЕНИИ ФУНКЦИЙ В ОБОБЩЁННОЙ ФОРМЕ

Предлагается метод синтеза конечных циклических цифровых автоматов с перестраиваемой длиной цикла, основанный на представлении функций настройки в обобщённой форме в сжатой области определения, отождествляя координаты точек области с состояниями автомата, а значение функции в каждой точке сжатой области представляется произведением литералов настроечных переменных, определяющих настройку на заданную длину цикла. Выполнен синтез заданного автомата, приведена его схема, даны рекомендации по выбору оптимального кодирования режимов настройки.

Ключевые слова: *цифровые автоматы, логические функции, синтез, минимизация, оптимизация.*

Введение

Постановка проблемы. Работа посвящена синтезу циклических конечных автоматов с программируемой длиной цикла, используемых при построении делителей частоты повторения импульсов, различных датчиков и циклических генераторов дискретных интервалов времени, программируемых интервальных таймеров, устройств управления шаговыми двигателями, устройств управления микропрограммными автоматами с перестраиваемой длительностью микрокоманд, генераторов тона в аудио системах, генераторов световых эффектов, генераторов циклограмм и т.д.

Большинство методов синтеза цифровых устройств ориентировано на современную элементную базу программируемых интегральных схем (ПЛИС) двух разновидностей – PLD (Programmable Logic Devices) и FPGA (Field Programmable Gate Array), хотя явно это может и не подчёркиваться [1, 2].

Традиционные методы синтеза принято делить на два больших класса: двухуровневый и многоуровневый синтез. Двухуровневый синтез предполагает прохождение сигнала со входа комбинационной схемы на выход через два уровня логических элементов, обычно это элементы И и ИЛИ, что соответствует структуре PLD.

Многоуровневый синтез сводится к синтезу логической сети, узлами которой являются некоторые логические элементы, что соответствует синтезу комбинационных схем на FPGA.

Одной из основных задач двухуровневого и, как последующего его развития, многоуровневого синтеза схем является минимизация логических функций в классе дизъюнктивных нормальных форм (ДНФ).

В связи с бурным развитием программируемых интегральных схем и внедрением их в практику проектирования цифровых устройств [1] вновь возрос интерес к логическому синтезу, причём, не только на уровне автоматизированного проектирования, но и на более низком - «ручного синтеза». Проблема «ручного синтеза» остаётся актуальной при синтезе цифровых устройств, выполненных на микросхемах малого и среднего уровня интеграции, при разработке библиотек стандартных элементов и узлов с последующим их использованием на более высоком уровне автоматизированного проектирования на основе PLD и FPGA.

В связи с этим исследования, направленные на разработку новых элементов и узлов и совершенствование методов проектирования, обеспечивающих улучшение их основных характеристик, а также снижение времени и стоимости разработки никогда не потеряют своей актуальности.

Анализ исследований и публикации, посвящённых проблеме проектирования блоков и узлов, многие из которых вошли в библиотеку типовых, посвящено большое число работ, простое перечисление которых представляет собой далеко не тривиальную задачу [2]. При этом следует отметить, что, не смотря на достаточную широту известных библиотек типовых узлов, являющихся основными кирпичиками в арсенале разработчиков цифровых устройств, циклические автоматы с перестраиваемой длиной цикла не представлены.

Известные методы проектирования позволяют по заданному алгоритму построить любой конечный

автомат. Построение циклического автомата, как конечного автомата с жёсткой логикой, не вызывает никаких проблем для случая фиксированной длины цикла. При проектировании циклического автомата с перестраиваемой длиной цикла, появляются проблемы, связанные даже не с синтезом схемы автомата как такового, а с нахождением минимального представления функции, определяющей настройку автомата на заданную длину цикла, которая зависит от переменных, определяющих состояние автомата ($Q_{i-1}\cdots Q_0$), и переменных, определяющих настройку на заданную длину цикла ($a_{i-1}\cdots a_0$).

Поскольку функция недоопределена примерно на половине наборов, то возникают проблемы, связанные с нахождением минимальной формы. Существующие методы минимизации таких функций основаны на переборе и анализе всех возможных вариантов доопределения, число которых равно 2^m (m – число недоопределённых наборов). Это число довольно велико даже при относительно небольших значениях разрядности автомата.

В работах [3 – 7] исследован метод синтеза, основанный на представлении функций в обобщённой форме. При этом отмечено, что представление функций в форме ОЛФ в одних случаях выступает просто как способ, обеспечивающий уменьшение числа точек области определения, что само по себе немаловажно, поскольку позволяет уменьшить размерность таблиц функционирования и, как следствие, сократить время проектирования. В других случаях такое представление выступает как значительно большее, чем простое уменьшение числа точек, обеспечивающее возможность анализа функций с последующим упрощением алгоритма преобразования и оптимизации по заданному критерию и контроля достоверности полученных результатов [8, 9].

Цель статьи – продолжить исследования, направленные на приложение аппарата обобщённых логических функций к разработке методов оптимизации проектирования циклических конечных автоматов с перестраиваемой длиной цикла.

Метод решения

Ключевым моментом предлагаемого метода является разбиение множества всех переменных ($Q_{i-1}\cdots Q_0\, a_{i-1}\cdots a_0$), определяющих функцию настройки, на два подмножества: подмножество переменных, определяющих состояние автомата ($Q_{i-1}\cdots Q_0$), и подмножество переменных, определяющих настройку автомата на заданную длину цикла ($a_{i-1}\cdots a_0$), с последующим представлением функции настройки в сжатой области определения, отождествляя координаты точек области с состояниями автомата, а значение функции в каждой точке отмечаем минтермом, образуемым литерами настроечных переменных ($a_{i-1}\cdots a_0$). Такой подход позволяет не только сократить число точек области определения, но и существенно упростить процедуру нахождения оптимального доопределения функции настройки в недоопределённых точках.

Суть любого способа построения цифрового автомата с перестраиваемой длиной цикла состоит в укорочении числа состояний типового циклического устройства (суммирующего или вычитающего счётчика, сдвигающего регистра с обратной связью) до заданного числа, определяемого настроечными переменными.

Предлагаемый метод проектирования рассмотрим на примере использования четырёхразрядного двоичного суммирующего счётчика со входом синхронной установки в нулевое состояние R (Reset).

Известно, что счётчик с синхронным Reset имеет два режима: режим счёта и режим синхронной установки в нулевое состояние. Режим счёта имеет место при $R = 0$. В этом режиме с поступлением тактирующих импульсов на вход C счётчик переходит из любого исходного состояния в следующее. При $R = 1$ с поступлением тактирующего импульса на вход C счётчик переходит из любого исходного состояния в нулевое состояние.

Наличие этих режимов позволяет использовать счётчик для построения циклического автомата с перестраиваемым числом состояний от минимального, равного 1, до максимального, равного 16.

Сигнал перевода счётчика в нулевое состояние можно представить как функцию, определяемую состоянием автомата ($Q_3Q_2Q_1Q_0$), и настроечными переменными ($a_3a_2a_1a_0$):

$$R = F(Q_3Q_2Q_1Q_0a_3a_2a_1a_0).$$

Если счётчик находится в некотором r-м состоянии, то для перевода его в состояние нуля необходимо обеспечить формирование сигнала R=1.

Трактуя минтерм m_r, определяемый значениями переменных $Q_3Q_2Q_1Q_0$, соответствующих этому состоянию, как координату точки сжатой области определения, а минтерм k_s, определяемый настроечными переменными $a_3a_2a_1a_0$, как настройку на переход счётчика из состояния «r» в состояние «0». Единичное значение функции настройки (R), соответствующее этой точке, можно представить в виде произведения этих минтермов: $m_r \cdot k_s$. Значения кодов настройки (кодирование минтермов), обеспечивающих переход из каждого состояния в нулевое, может быть произвольным. Как будет показано ниже, от характера кодирования режимов настройки зависит сложность функции настройки.

Исходя из смыслового содержания алгоритма настройки индексы минтермов настройки удобно

отождествить с длиной цикла, начиная от 1 до 16 k_1 обеспечивает неизменность исходного состояния (вырожденный цикл); k_2 обеспечивает два состояния и так далее, k_0 – 16 состояний. При таком кодировании режимов настройки функцию настройки можно будет представить в виде логической суммы произведений $m_{i-1} \cdot k_i$:

$$R = \bigvee_{i=1}^{15} m_{i-1} \cdot k_i ,$$

что соответствует представлению функции в СДНФ.

Схемная реализация функции, представленной в СДНФ, как правило, сложна, как в классе элементов малого уровня интеграции, также в классе ПЛИС. При оценке сложности функции настройки исходим из необходимого числа логических элементов с учётом числа входов каждого из них, и суммарного числа входов-выходов (по Квайну). В соответствии с этой оценкой для реализации её в классе элементов малого уровня интеграции потребуется 15 восьмивходовых элементов И, один пятьнадцативходовый элемент ИЛИ и 8 инверторов с общим числом входов-выходов, равным 176.

Реализация схемы в классе элементов среднего уровня интеграции с использованием двух четырёхадресных дешифраторов, один из которых формирует минтермы, образуемые переменными $Q_3Q_2Q_1Q_0$, а второй – переменными $a_3a_2a_1a_0$, с последующим перемножением соответствующих минтермов и логическим суммированием, позволяет уменьшить общее число микросхем, но сами микросхемы имеют более сложную структуру.

Представление функции в минимальной форме (в ДНФ или КНФ) позволяет сократить, общее число логических элементов, число входов их, и, как следствие, суммарное число входов-выходов.

Минимизации функции настройки известными методами сопряжена с определёнными трудностями, обусловленными не столько числом переменных, сколько наличием недоопределённых наборов, число которых (будет показано ниже) довольно велико.

Для более глубокого представления и анализа алгоритма функционировании проектируемого автомата, дающего возможность формализовать процедуру выбора вариантов оптимального кодирования вариантов настройки, исходя из критерия минимально возможной сложности функции настройки, представим алгоритм функционирования автомата в виде графа (рис. 1), отождествляя вершины его с состояниями автомата, а дуги отмечаем условиями перехода счётчика из r го состояния ($r = i - 1$) в состояние «0» (минтермами k_i, обуславливающими эти переходы). Из графа видно, что если значение настроечных переменных равно 0001, что соответствует минтерму m_1, с поступлением каждого очередного тактирующего импульса нулевое состояние счётчика буде оставаться неизменным (счётчик из 0 будет переходить в 0). И только если значение кода настройки отлично от 0001, счётчик перейдёт в единичное состояние. Если при этом значение набора настроечных переменных равно 0010, что соответствует минтерму m_2, то с поступлением следующего тактирующего импульса счётчик перейдёт в нулевое состояние, затем снова в единичное и т.д., пока не будет изменён код настройки.

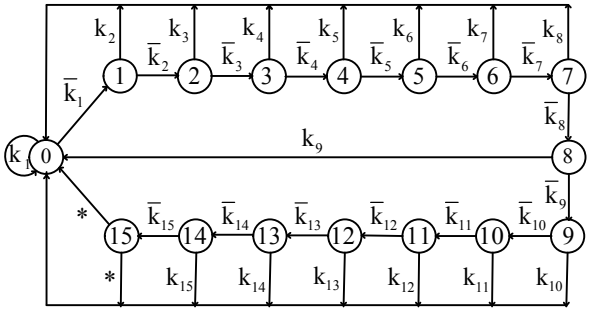

Рис. 1. Граф переходов циклического автомата (первый вариант кодировании режимов настройки)

Аналогичная ситуация будет иметь место при любой другой настройке.

Минтермы m_{i-1}, определяемые переменными $Q_3Q_2Q_1Q_0$, трактуем как координаты точек области определения, а минтермы k_i, определяемые настроечными переменными $a_3a_2a_1a_0$, как значения функции настройки в этих точках. Представляем область определения функции настройки в виде шестнадцатиэлементной карты Карно (рис. 2), окрашенной интервалами значений переменных $Q_3Q_2Q_1Q_0$. При этом особо следует подчеркнуть, что при записи значений функций в точках сжатой области определения в каждую из них (кроме нулевой точки) будет входить не только один полностью определённый минтерм, определяющий единичное значение функции настройки в этой точке, но и некоторые недоопределённые минтермы (отмеченные в клетках карты «звёздочкой»), соответствующие режимам настройки на меньшую длину цикла (фиктивным режимам настройки для рассматриваемой точки). В частности, в нулевой точке области определения будет представлен только один полностью определённый минтерм k_1. Во все следующие точки входит отмеченное значение этого минтерма – k_1^*. В первой точке области определения будет представлен полностью определённый минтерм k_2 и отмеченный минтерм k_1^*. Отмеченное значение минтерма k_2^* будет представлено во всех следующих точках области определения. Во второй точке области определения будет представлен полностью определённый минтерм k_3 и отмеченные минтермы k_1^* и k_2^*. Отмеченное значение минтерма k_3^* будет представлено во всех следующих точках области определения. И так далее вплоть до 15-й точки области определения, где будет представлен полностью определённый минтерм k_0 и все остальные отмеченные

минтермы, начиная от k_1^* до k_{15}^*. Как было показано в [7], суть любого отмеченного минтерма состоит в том, что при нахождении минимальной формы функции настройки он может быть доопределён значением полностью определённого минтерма с тем же индексом или значением 0.

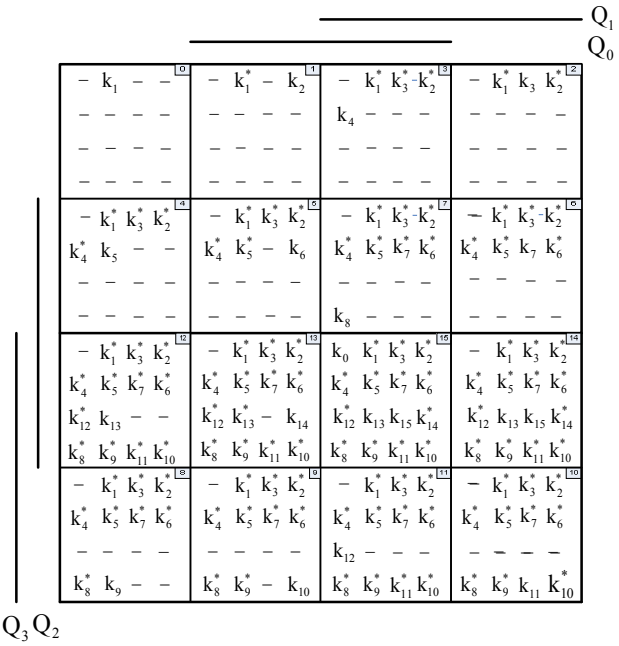

Рис. 2. Область определения функции настройки (первый вариант кодировании режимов настройки)

С целью нахождения оптимальных вариантов доопределения функции отмеченными минтермами при записи их в точках сжатой области определения (клетках карты) располагают в форме шестнадцатиэлементной матрицы с соседним кодированием по настроечным переменным, что позволяет существенно упростить процедуру доопределения.

Комментарий к процедуре выделения правильных конфигураций с доопределением определённых минтермов отмеченными опускаем, записывая только результаты его в виде покрываемых определённых минтермов, входящих в эти конфигурации (k_i), номеров клеток, образующих конфигурации ($\langle \dots \rangle$), и соответствующих им простых импликант (I_i).

$k_1 : I_1 = k_1 = \bar{a}_3\bar{a}_2\bar{a}_1a_0$;

$k_2 : \langle 1,3,5,7,13,15,9,11\rangle - I_2 = k_2Q_0 = \bar{a}_3\bar{a}_2a_1\bar{a}_0Q_0$;

$k_3 : \langle 2,3,6,7,14,15,10,11\rangle - I_3 = (k_3 \vee k_2)Q_1 = \bar{a}_3\bar{a}_2a_1Q_1$;

$k_4 : \langle 3,7,15,11\rangle - I_4 = k_4Q_1Q_0 = \bar{a}_3a_2\bar{a}_1\bar{a}_0Q_1Q_0$;

$k_5 : \langle 4,5,6,7,12,13,14,15\rangle - I_5 = (k_5 \vee k_4)Q_2 = \bar{a}_3a_2\bar{a}_1Q_2$;

$k_6 : \langle 5,7,13,15\rangle - I_6 = (k_6 \vee k_2)Q_2Q_0 = \bar{a}_3a_1\bar{a}_0Q_2Q_0$;

$I_6 = (k_6 \vee k_4)Q_2Q_0 = \bar{a}_3a_2\bar{a}_0Q_2Q_0$;

$k_7 : \langle 6,7,14,15\rangle - I_7 = (k_7 \vee k_6 \vee k_4 \vee k_5)Q_2Q_1 = \bar{a}_3a_2Q_2Q_1$;

$I_7 = (k_7 \vee k_6 \vee k_2 \vee k_3)Q_2Q_1 = \bar{a}_3a_1Q_2Q_1$;

$k_8 : \langle 7,15\rangle - I_8 = k_8Q_2Q_1Q_0 = a_3\bar{a}_2\bar{a}_1\bar{a}_0Q_2Q_1Q_0$;

$k_9 : \langle 8,9,10,11,12,13,14,15\rangle - I_9 = (k_9 \vee k_8)Q_3 = a_3\bar{a}_2\bar{a}_1Q_3$;

$k_{10} : \langle 9,11,13,15\rangle - I_{10} = (k_{10} \vee k_8)Q_3Q_0 = a_3\bar{a}_2\bar{a}_0Q_3Q_0$;

$I_{10} = (k_{10} \vee k_2)Q_3Q_0 = \bar{a}_2a_1\bar{a}_0Q_3Q_0$;

$k_{11} : \langle 10,11,14,15\rangle - I_{11} = (k_{11} \vee k_{10} \vee k_8 \vee k_9)Q_3Q_1 = a_3\bar{a}_2Q_3Q_1$;

$k_{12} : \langle 11,15\rangle - I_{12} = (k_{12} \vee k_8)Q_3Q_1Q_0 = a_3\bar{a}_1\bar{a}_0Q_3Q_1Q_0$;

$I_{12} = (k_{12} \vee k_4)Q_3Q_1Q_0 = a_2\bar{a}_1\bar{a}_0Q_3Q_1Q_0$;

$k_{13} : \langle 12,13,14,15\rangle - I_{13} = (k_{13} \vee k_{12} \vee k_8 \vee k_9)Q_3Q_2 = a_3\bar{a}_1Q_3Q_2$;

$I_{13} = (k_{13} \vee k_{12} \vee k_4 \vee k_5)Q_3Q_2 = a_2\bar{a}_1Q_3Q_2$;

$k_{14} : \langle 13,15\rangle - I_{14} = (k_{14} \vee k_{10} \vee k_{12} \vee k_8)Q_3Q_2Q_0 = a_3\bar{a}_0Q_3Q_2Q_0$;

$I_{14} = (k_{14} \vee k_{10} \vee k_2 \vee k_6)Q_3Q_2Q_0 = a_1\bar{a}_0Q_3Q_2Q_0$;

$k_{15} : \langle 14,15\rangle - I_{15} = (k_{15} \vee k_{14} \vee k_{13} \vee k_{12} \vee k_8 \vee k_9 \vee \vee k_{10} \vee k_{11})Q_3Q_2Q_1 = a_3Q_3Q_2Q_1$;

$I_{15} = (k_{15} \vee k_{14} \vee k_{13} \vee k_{12} \vee k_4 \vee k_5 \vee k_6 \vee \vee k_7)Q_3Q_2Q_1 = a_2Q_3Q_2Q_1$.

Выполняя логическое сложение полученных простых импликант, можно записать минимальную ДНФ функции настройки в виде:

$$R = \bigvee_{i=1}^{15} I_i.$$

При этом следует заметить, что поскольку для выбранного кодирования режимов настройки 7 импликант ($I_6, I_9, I_{10}, I_{12}, I_{13}, I_{14}, I_{15}$) можно реализовать двумя способами, то число минимальных ДНФ довольно велико (равно 2^7).

Оценивая сложность схемной реализации в классе элементов малого уровня интеграции замечаем, что число элементов сократилось только на 4 инвертора, но поскольку число входов каждого из элементов И уменьшено, то всё это позволило сократить общее число входов-выходов до 117. Это конечно немного проще, чем реализация функции, представленной в СДНФ, но выигрыш небольшой.

В силу громоздкости полученных выражений, запись минимальной ДНФ для данного варианта кодирования режимов настройки и схему автомата не приводим.

Анализируя возможные варианты размещения минтермов в точках области определения функции настройки не трудно заметить, что если выбрать такое кодирование режимов настройки, которое обеспечит соседство определённого минтерма максимальным числом (кратным степени двойки) недоопределённых минтермов, то выражения для простых импликант можно упростить.

В качестве одного из таких вариантов можно предложить вариант, когда индекс минтерма, определяющего переход из i-го состояния в нулевое совпадает с номером состояния, т.е. с индексом минтерма, определяющего это состояние.

Граф переходов автомата для предложенного варианта кодирования приведен на рис. 3.

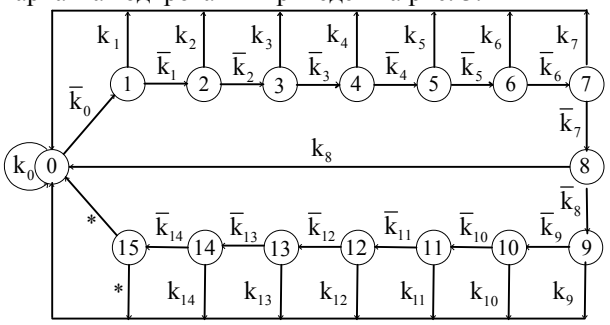

Рис. 3. Граф переходов автомата (второй вариант кодирования режимов настройки)

Значение кода настройки, равное нулю, обеспечивает переход счётчика из нулевого состояния в нулевое, значение кода настройки, равное единице, обеспечивает переход счётчика из единичного состояния в нулевое, значение кода настройки, равное двойке, обеспечивает переход счётчика из состояния двойки в нулевое и так далее, вплоть до минтерма k_{14}. Формирование импликанты, соответствующей 15-му состоянию, не обязательно, поскольку счётчик из этого состояния переходит в нулевое, независимо от значения сигнала на входе R .

Как и в первом варианте, функцию настройки представляем в шестнадцатиэлементной карте Карно, приведенной на рис. 4.

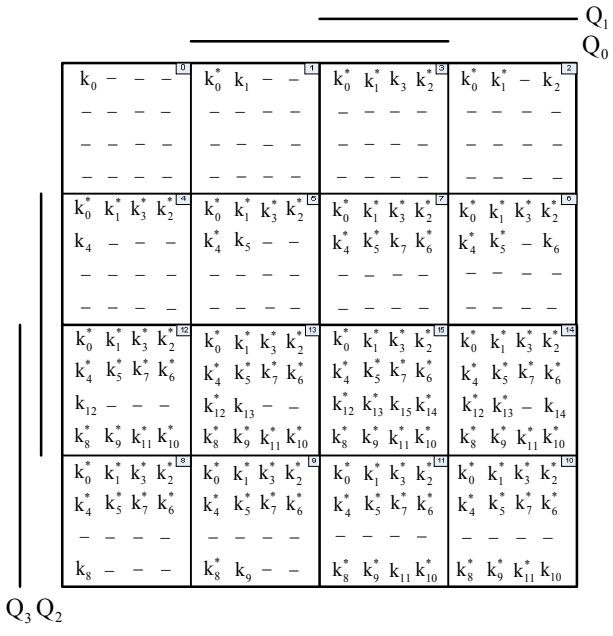

Рис. 4. Область определения функции настройки (второй вариант кодировании режимов)

В каждой клетке карты представлен один определённый минтерм, индекс которого совпадает с состоянием автомата (номером клетки), и некоторые недоопределённые минтермы.

В нулевой точке области определения будет представлен только один полностью определённый минтерм k_0^*. Отсюда следует, что если значения настроечных переменных будут равно нулю, то состояние автомата будет оставаться неизменным, следовательно, во все следующие точки области определений можно записать отмеченное значение этого минтерма – k_0^*. В первой точке будет представлен полностью определённый минтерм k_1 и отмеченный минтерм k_0^*. Во второй точке области определения будет представлен полностью определённый минтерм k_2 и отмеченные минтермы k_0^* и k_1^*. Отмеченное значение минтерма k_2^* будет представлено во всех следующих точках области определения. И так далее, вплоть до 15 точки области определения, где будет представлен полностью определённый минтерм k_{15} и все остальные отмеченные минтермы, начиная от k_0^* до k_{14}^*.

Также, как и выше комментарий к процедуре выделения правильных конфигураций максимально возможной площади с доопределением определённых минтермов отмеченными опускаем, записывая только результаты его в виде покрываемых определённых минтермов, входящих в эти конфигурации, номеров клеток, образующих конфигурации, и соответствующих им простых импликант.

$k_0 : I_0 = k_0 = \bar{a}_3\bar{a}_2\bar{a}_1\bar{a}_0$;

$k_1 : \langle 1,3,5,7,13,15,9,11\rangle - I_1 = (k_1 \vee k_0)Q_0 = \bar{a}_3\bar{a}_2\bar{a}_1Q_0$;

$k_2 : \langle 2,3,6,7,14,15,10,11\rangle - I_3 = (k_2 \vee k_0)Q_1 = \bar{a}_3\bar{a}_2\bar{a}_0Q_1$;

$k_3 : \langle 3,7,15,11\rangle - I_3 = (k_3 \vee k_2 \vee k_1 \vee k_0)Q_1Q_0 = \bar{a}_3\bar{a}_2Q_1Q_0$;

$k_4 : \langle 4,5,6,7,12,13,14,15\rangle - I_4 = (k_4 \vee k_0)Q_2 = \bar{a}_2\bar{a}_1\bar{a}_0Q_2$;

$k_5 : \langle 5,7,13,15\rangle - I_5 = (k_4 \vee k_5 \vee k_0 \vee k_1)Q_2Q_0 = \bar{a}_2\bar{a}_1Q_2Q_0$;

$k_6 : \langle 6,7,14,15\rangle - I_6 = (k_6 \vee k_4 \vee k_2 \vee k_0)Q_2Q_1 = \bar{a}_2\bar{a}_0Q_2Q_1$;

$k_7 : \langle 7,15\rangle - I_7 = (k_7 \vee k_6 \vee k_5 \vee k_4 \vee k_3 \vee k_2 \vee \vee k_1 \vee k_0)Q_2Q_1Q_0 = \bar{a}_3Q_2Q_1Q_0$;

$k_8 : \langle 8,9,10,11,12,13,14,15\rangle - I_8 = (k_8 \vee k_0)Q_3 = \bar{a}_2\bar{a}_1\bar{a}_0Q_3$;

$k_9 : \langle 9,11,13,15\rangle - I_9 = (k_9 \vee k_8 \vee k_0 \vee k_1)Q_3Q_0 = \bar{a}_2\bar{a}_0Q_3Q_0$;

$k_{10} : \langle 10,11,14,15\rangle - I_{11} = (k_{10} \vee k_8 \vee k_2 \vee k_0)Q_3Q_1 = \bar{a}_2\bar{a}_0Q_3Q_1$;

$k_{11} : \langle 11,15\rangle - I_{11} = (k_{11} \vee k_{10} \vee k_9 \vee k_8 \vee k_3 \vee \vee k_2 \vee k_1 \vee k_0)Q_3Q_1Q_0 = \bar{a}_2Q_3Q_1Q_0$;

$k_{12} : \langle 12,13,14,15\rangle - I_{12} = (k_{12} \vee k_8 \vee k_4 \vee k_0)Q_3Q_2 = \bar{a}_1\bar{a}_0Q_3Q_2$;

$k_{13} : \langle 13,15\rangle - I_{13} = (k_{13} \vee k_{12} \vee k_8 \vee k_9 \vee k_4 \vee \vee k_5 \vee k_0 \vee k_1)Q_3Q_2Q_0 = \bar{a}_1Q_3Q_2Q_0$;

$k_{14} : \langle 14,15\rangle - I_{14} = (k_{14} \vee k_{10} \vee k_{12} \vee k_8 \vee k_6 \vee \vee k_4 \vee k_2 \vee k_0)Q_3Q_2Q_1 = \bar{a}_0Q_3Q_2Q_1$.

Оценивая сложность схемной реализации рассмотренного варианта кодирования режимов настройки, следует отметить, что не смотря на то, что число элементов по сравнению с предыдущем вари-

антом не сократилось, схемная реализация рассматриваемого варианта предпочтительнее, поскольку все элементы И однотипны (четырёхвходовые), что позволило не только сократить общее число входов-выходов, но и существенно уменьшить общее число корпусов типовых микросхем по сравнению с предыдущим вариантом и обеспечить регулярность структуры при её реализации на плате.

При реализации схемы в классе ПЛИС можно предложить этот вариант, поскольку в кристалле одинаково доступны как прямые, а также инверсные значения переменных.

Если схема ориентирована на микросхемы малого и среднего уровня интеграции, то наличие инверсных значений настроечных переменных при выбранном варианте настройки вызывает необходимость постановки четырёх инверторов.

Некоторое усложнение схемы не особенно существенно. Основной недостаток такого варианта состоит в увеличении задержки. Её можно исключить, изменяя коды настройки на инверсные значения, т.е. режим перехода автомата из нулевого состояния в нулевое настраивается минтермом k_{15}, из первого в нулевое – минтермом k_{14} и так далее, вплоть до минтерма k_1, обеспечивающего переход из состоянии «14» в состояние «15».

Граф переходов автомата для предложенного варианта кодирования приведен на рис. 5

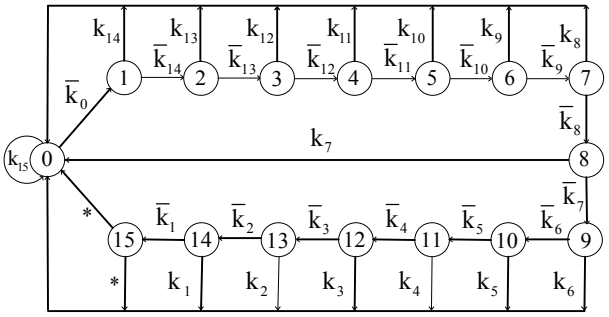

Рис. 5. Граф переходов автомата (третий вариант кодирования режимов настройки)

При нахождении функции настройки для этого варианта кодирования можно было бы воспользоваться рассмотренной выше методикой, представляя функцию настройки в карте. Однако, поскольку коды настройки в каждой из точек области определения функции R представлены противоположными значениями, то достаточно в полученных в предыдущем варианте выражениях инверсные значения настроечных переменных заменить прямыми. В результате такой подстановки получаем простые импликанты, определяющие функцию настройки автомата, приведенную на рис. 6.

$k_0 : I_0 = k_0 = a_3a_2a_1a_0$;

$k_1 : \langle 1,3,5,7,13,15,9,11\rangle - I_1 = (k_1 \vee k_0)Q_0 = a_3a_2a_1Q_0$;

$k_2 : \langle 2,3,6,7,14,15,10,11\rangle - I_3 = (k_2 \vee k_0)Q_1 = a_3a_2a_0Q_1$;

.....

$k_{13} : \langle 13,15\rangle - I_{13} = (k_{13} \vee k_{12} \vee k_8 \vee k_9 \vee k_4 \vee k_5 \vee \vee k_0 \vee k_1)Q_3Q_2Q_0 = a_1Q_3Q_2Q_0$;

$k_{14} : \langle 14,15\rangle - I_{14} = (k_{14} \vee k_{10} \vee k_{12} \vee k_8 \vee k_6 \vee \vee k_4 \vee k_2 \vee k_0)Q_3Q_2Q_1 = a_0Q_3Q_2Q_1$.

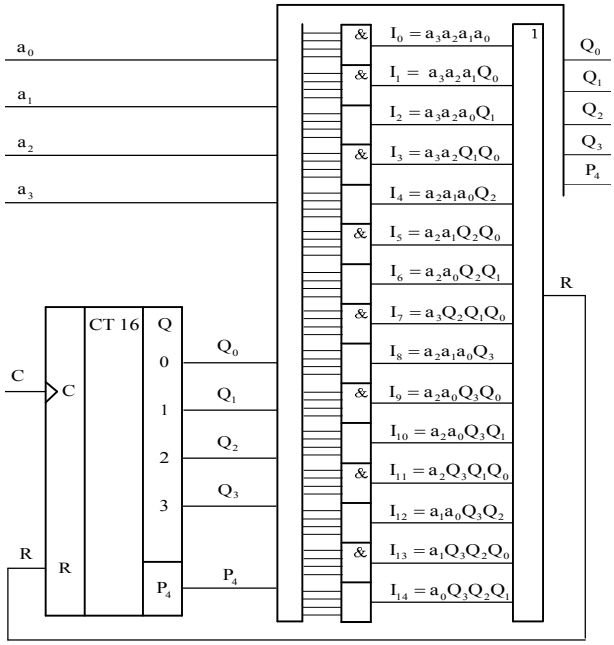

Рис. 6. Схема циклического конечного автомата с перестраиваемой длиной цикла

Заключение

Впервые предложено использование представления логических функций в обобщённой форме при синтезе цифровых автоматов с перестраиваемыми режимами функционирования, что позволяет, в отличие от известных подходов, сократить число анализируемых точек области определения функций и упростить выбор оптимальных вариантов, определив тем самым *научную новизну* и *практическую значимость* работы.

Сравнивая предложенный метод синтеза с известными, заключаем, что он выгодно *отличается* от них, поскольку в какой-то мере совместил в себе простоту визуального анализа, присущего известным табличным методам, с математической строгостью, характерной для аналитических методов.

Одним из направлений *дальнейших исследований* является описание предложенного подхода на языке описания аппаратуры (VHDL), позволяющего создавать сложные и большие проекты, ориентированные на современную элементную базу ПЛИС.

Литература

1. Уэйкерли Дж. Ф. Проектирование цифровых устройств: пер. с англ. / Дж. Уэйкерли. – М.: Постмаркет, 2002. – 544 с.

2. Соловьев В.В. Проектирование цифровых систем на основе программируемых логических интегральных схем / В.В. Соловьев. – М.: Горячая линия-Телеком, 2001. – 636 с.

3. Обобщённые логические функции и системы на программируемой логике: учеб. пособие / Н.Г. Коробков, Е.Н. Коробкова, В.Г. Рубанов, В.С. Харченко; под общ. ред. В.С. Харченко. – Минво образования и науки Украины, Нац. аэрокосмический ун-т. им. Н.Е. Жуковского «ХАИ». – Х.: Изд-во НАКУ «ХАИ», 2008. – 351 с.

4. Рубанов В.Г. Логическое проектирование цифровых устройств, основанное на представлении функций в обобщённой форме: моногр. / В.Г. Рубанов, Е.Н. Коробкова. – Белгород: Издательство БГТУ, 2009. – 335 с.

5. Коробков Н.Г. Разработка алгоритма минимизации обобщённых логических функций с зависимыми параметрами / Н.Г. Коробков, Е.Н. Коробкова // Системи обробки інформації. – Х.: НАНУ, ПАНМ, ХВУ, 2002. – Вип. 2(18). – С. 225-230.

6. Рубанов В.Г. Визуально-алгебраический способ минимизации логических функций / В.Г. Рубанов, Е.Н. Коробкова // Радіоелектроніка та нформатика. – Х.: ХНУРЭ. – 2004. – № 2. – С. 112-115.

7. Коробкова Е.Н. Представление и минимизация недоопределенных логических функций в сжатых картах / Е.Н. Коробкова // Системи обробки інформації. – Х.: ХВУ, 2003. – Вип. 4. – С. 35-50.

8. Рубанов В.Г. Оценка достоверности минимизации логических функций / В.Г. Рубанов, Е.Н. Коробкова // Телекоммуникации. – 2008. – №1. – С. 44-48.

9. Рубанов В.Г. Контроль достоверности минимизации логических функций / В.Г. Рубанов, Е.Н. Коробкова // Радіоелектроніка та інформатика. – Х.: ХНУРЭ. – 2006. – № 2. – С. 62-68.

Поступила в редакцию 7.04.2009

Рецензент: д-р техн. наук, профессор, заведующий кафедрой информационных систем О.Е. Федорович, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.

СИНТЕЗ КІНЦЕВИХ ЦИКЛІЧНИХ ЦИФРОВИХ АВТОМАТІВ З ПЕРЕСТРОЮВАНОЮ ДОВЖИНОЮ ЦИКЛУ, ЩО БАЗУЮТЬСЯ НА ПРЕДСТАВЛЕННІ ФУНКЦІЙ В УЗАГАЛЬНЕНІЙ ФОРМІ

М.Г. Коробков, О.М. Коробкова, В.Г. Рубанов

Запропоновано метод синтезу кінцевих циклічних цифрових автоматів з перестроюваною довжиною циклу, що базується на представленні функцій настроювання в узагальненій формі в стиснутій області визначення, уточнюючи координати точок області зі станами автомата, а значення функції у кожній точці стиснутої області надається добутком літералів настроєних змінних, що визначають настройку на задану довжину циклу. Виконано синтез заданого автомата, наведена його схема, надаються рекомендації з вибору оптимального кодування режимів настройки.

Ключові слова: логічні функції, цифрові автомати, синтез, оптимізація.

SYNTHESIS OF CYCLIC FINITE-STATE AUTOMATONS WITH PROGRAMMABLE CYCLE LENGTH BASED ON REPRESENTING FUNCTIONS IN GENERALIZED FORM

N.G. Korobkov, E.N. Korobkova, W.G. Rubanov

The method of synthesis of cyclic finite-state automatons with programmable cycle length is proposed. It is based on representing functions in generalized form within shortened definition domain by refining domain coordinates in accordance with current automaton state. Function value is represented by conjunction of tuning variables literals setting the necessary cycle length. An example of automaton synthesis is given. The automaton scheme as well as recommendations on optimal coding is presented.

Key words: logical functions, digital automatons, synthesis, optimization.

Коробков Николай Григорьевич – канд. техн. наук, доцент, доцент кафедры компьютерных систем и сетей Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ», Харьков, Украина.

Коробкова Елена Николаевна – канд. техн. наук, старший преподаватель кафедры технической кибернетики Белгородского государственного технологического университета им. В.Г. Шухова, Белгород, Россия.

Рубанов Василий Григорьевич – д-р техн. наук, профессор, заведующий кафедрой технической кибернетики Белгородского государственного технологического университета им. В.Г. Шухова, Белгород, Россия.