ИНФОРМАТИКА

2012 октябрь-декабрь № 4

УДК 519.7

Ю.В. Поттосин

ИТЕРАТИВНЫЙ МЕТОД ЭНЕРГОСБЕРЕГАЮЩЕГО КОДИРОВАНИЯ СОСТОЯНИЙ ДИСКРЕТНОГО АВТОМАТА

Рассматривается задача кодирования состояний дискретного автомата с целью уменьшения интенсивности переключений элементов памяти в реализующей схеме. Определение значений внутренних переменных сводится к задаче нахождения максимального разреза во взвешенном графе.

Введение

В последнее время при проектировании дискретных устройств управления на основе сверхбольших интегральных схем большое внимание уделяется проблеме снижения энергопотребления проектируемой схемы. Это обусловлено, с одной стороны, стремлением увеличить время действия источника энергии в портативных приборах и, с другой стороны, стремлением снизить остроту проблемы отвода тепла при проектировании сверхбольших интегральных схем. Поэтому одним из основных критериев оптимизации при проектировании дискретных устройств является величина потребляемой схемой энергии.

Как отмечено в работах [1, 2], потребляемая мощность схемы, построенной на основе КМОП-технологии, пропорциональна интенсивности переключений логических элементов и элементов памяти. Это дает возможность частично решать данную проблему на уровне логического проектирования. В частности, снижения энергопотребления можно добиваться при кодировании состояний автомата [3–5]. Кодировать состояния при этом надо таким образом, чтобы при переходе автомата из одного состояния в другое меняли свое состояние как можно меньше элементов памяти.

Первоначально критерием оптимизации при кодировании состояний была простота описания булева автомата, представляющего собой систему выходных булевых функций и функций возбуждения элементов памяти автомата [6]. Существует два подхода к решению этой задачи. Один из них предложен в статье [7] и направлен на упрощение двухуровневого представления системы булевых функций, т. е. получение как можно меньшего общего числа различных элементарных конъюнкций в системе дизъюнктивных нормальных форм. Другой подход [8] использует структуру разбиений на множестве состояний автомата, и целью кодирования при этом подходе является ослабление зависимости функций от их аргументов, т. е. считается, что функция тем проще, чем от меньшего числа аргументов она зависит.

Первый подход [7] положен в основу методов, предлагаемых в работах [5, 9], где кодирование состояний автомата сводится к оптимальной укладке графа поведения автомата в булевом гиперкубе. Этот подход рассчитан на общую модель автомата. В настоящей работе для решения задачи кодирования состояний используется подход, предложенный в работе [10] и описанный в работе [11]. Он близок к подходу [8], используемому также в работах [3, 4]. Предлагаемый далее метод ориентирован на модель автомата с абстрактным состоянием [6]. Метод назван итеративным, поскольку его выполнение состоит в последовательности итераций, на каждой из которых вводится внутренняя переменная, являющаяся компонентой кода состояния автомата.

1. Описание метода

Данный метод выгодно применять, когда для задания автомата используется модель с абстрактным состоянием и число его состояний невелико по сравнению с числом выходных булевых переменных.

Примером задания автомата с абстрактным состоянием являются следующие две матрицы:

$$\boldsymbol{U}=\begin{array}{cc} \begin{array}{cccc} x_1 & x_2 & x_3 & q \end{array} & \\ \begin{bmatrix} - & - & - & 1 \\ 0 & - & - & 2 \\ 1 & 0 & - & 2 \\ 1 & 1 & - & 2 \\ - & - & 0 & 3 \\ - & 0 & 1 & 3 \\ - & 1 & 1 & 3 \\ - & - & - & 4 \\ - & 0 & - & 5 \\ - & 1 & 0 & 5 \\ - & 1 & 1 & 5 \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \end{array}, \quad \boldsymbol{V}=\begin{array}{c} \begin{array}{ccccc} q^{+} & y_1 & y_2 & y_3 & y_4 \end{array} \\ \begin{bmatrix} 2 & 0 & - & 1 & 0 \\ 3 & 1 & 0 & - & 1 \\ 3 & 0 & - & 1 & 0 \\ 5 & - & 1 & 1 & - \\ - & 0 & 1 & - & 0 \\ 5 & 0 & 0 & - & 0 \\ 2 & 0 & - & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ - & 0 & 1 & 1 & - \\ - & 0 & 1 & 0 & - \\ 4 & - & 0 & 0 & 0 \end{bmatrix} \end{array},$$

где, например, вторая сверху пара строк представляет сопровождаемый выходными сигналами $y_1 = 1$, $y_2 = 0$, $y_4 = 1$ при неопределенном значении y_3 переход автомата из состояния 2 в состояние 3 при $x_1 = 0$ и любых значениях x_2 и x_3.

Рассмотрим троичные векторы, являющиеся частями строк матриц $\boldsymbol{U}$ и $\boldsymbol{V}$. *Противоречием* в задании функции y_i в матрицах $\boldsymbol{U}$ и $\boldsymbol{V}$ назовем пару неортогональных троичных векторов $\boldsymbol{x}_s$ и $\boldsymbol{x}_t$, которым соответствуют противоположные значения y_i (0 и 1). Например, функция y_4 имеет противоречия в виде пар строк (1, 2), (2, 5), (2, 6), (2, 7), (2, 8) и (2, 11).

Процесс кодирования состояний заключается в устранении подобных противоречий путем введения внутренних булевых переменных $z_1, z_2, \dots, z_k$ и соответственно функций $z_1^{+}, z_2^{+}, \dots, z_k^{+}$. При введении новых функций вводятся, возможно, и новые противоречия, которые устраняются тем же путем. Значения переменных $z_1, z_2, \dots, z_k$ и функций $z_1^{+}, z_2^{+}, \dots, z_k^{+}$ должны быть согласованы со значениями соответствующих многозначных переменных q и q^{+}, т. е. одним и тем же значениям переменной q или q^{+} должны соответствовать одни и те же значения булевых переменных.

Результат указанного процесса в виде системы булевых функций в значительной степени зависит от порядка выбора функций для устранения противоречий. С целью упрощения получаемых функций рекомендуется для такого выбора использовать следующие соображения.

Текущая ситуация в данном процессе характеризуется функциями $y_1, y_2, \dots, y_m$, $z_1^{+}, z_2^{+}, \dots, z_l^{+}$ с противоречиями относительно аргументов $x_1, x_2, \dots, x_n$, $z_1, z_2, \dots, z_l$ (в начальной ситуации имеются только $y_1, y_2, \dots, y_m$ и $x_1, x_2, \dots, x_n$). Каждый шаг процесса заключается во введении внутренней переменной и придании ей значений, устраняющих противоречия в выбранной функции. При этом в матрицы $\boldsymbol{U}$ и $\boldsymbol{V}$ добавляется новый столбец. Вектор, представляющий часть строки матрицы $\boldsymbol{U}$ без компоненты q, обозначим $\boldsymbol{x}'$. Компонентами этого вектора являются переменные $x_1, x_2, \dots, x_n$ и $z_1, z_2, \dots, z_l$.

Для всех функций y_i и z_j^{+} построим *взвешенные графы противоречий*. Вершинам этих графов соответствуют состояния автомата, и две вершины q_g и q_h связаны ребром, если и только если q_g и q_h находятся в строках матрицы $\boldsymbol{U}$, троичные векторы которых $\boldsymbol{x}'_s$ и $\boldsymbol{x}'_t$ образуют противоречие для y_i. Вес w_{gh} ребра $q_g q_h$ определяется как $w_{gh} = m_{gh}/p_{gh}$, где m_{gh} – число пар строк матрицы $\boldsymbol{U}$, соответствующих противоречиям, в каждой из которых присутствуют q_g и q_h, а p_{gh} – увеличенное на единицу (чтобы не было деления на 0) число переходов между состояниями q_g и q_h заданного автомата не важно в какую сторону.

В работе [5] этот фактор выражается через вероятности переходов между состояниями, вычисляемые по методу Чэпмена – Колмогорова [12]. Однако этот метод пригоден только для полностью определенного автомата, который из любого состояния может перейти через какую-то последовательность переходов в любое его состояние. Здесь же рассматривается более общий случай не полностью определенного автомата.

Для устранения противоречий рассматриваем матрицы смежности $\boldsymbol{D}(y_i)$ взвешенных графов противоречий и в первую очередь выбираем ту функцию y_i, матрица $\boldsymbol{D}(y_i)$ которой обладает максимальной суммой значений элементов.

Рассмотрим, как вводится новая переменная z_{l+1} со значениями на некоторых состояниях автомата. Здесь уместно обратиться к задаче нахождения максимального разреза взвешенного графа, т. е. такого разбиения множества вершин V на два подмножества A и B, чтобы сумма весов ребер, соединяющих вершины из A с вершинами из B, была максимальной. На состояниях, соответствующих вершинам из множества A, переменной z_{l+1} приписывается значение 0 (или 1), а на состояниях, соответствующих вершинам из B, – значение 1 (или 0). Для разбиения множества V на подмножества A и B используем «жадный» алгоритм из работы [13], который представляет собой последовательность итераций, на каждой из которых в подмножестве B выбирается вершина v и переносится в подмножество A. Начальными значениями являются $A = \varnothing$ и $B = V$, а вершина v выбирается следующим образом.

Пусть d – сумма весов ребер, инцидентных вершине v, и d_A – сумма весов ребер, соединяющих вершину v с вершинами из A. Перенос вершины v из B в A сопровождается изменением суммы весов ребер, соединяющих вершины из A с вершинами из B, на величину $d - 2d_A$. На первом шаге эта величина равна степени переносимой вершины, а на последующих шагах она может быть отрицательной. Каждый раз выбирается та вершина, для которой величина $d - 2d_A$ максимальна, и процесс заканчивается, когда для всех вершин из подмножества B эта величина перестает быть положительной.

2. Пример

Для рассматриваемого примера величины p_{gh} представим в виде следующей матрицы (в силу симметричности подобных матриц здесь и в дальнейшем представляем только их верхнюю половину):

$$\begin{array}{cc} \begin{array}{cccc} 2 & 3 & 4 & 5 \end{array} & \\ \begin{bmatrix} 9 & 1 & 9 & 1 \\ & 9 & 1 & 3 \\ & & 1 & 3 \\ & & & 3 \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array}.$$

Получим матрицы смежности взвешенных графов противоречий, элементами которых являются веса соответствующих ребер:

$$\boldsymbol{D}(y_1) = \begin{array}{cc} \begin{array}{cccc} 2 & 3 & 4 & 5 \end{array} & \\ \begin{bmatrix} 1/9 & 0 & 1/9 & 0 \\ & 3/9 & 1 & 2/3 \\ & & 3 & 0 \\ & & & 2/3 \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array}, \quad \boldsymbol{D}(y_2) = \begin{array}{cc} \begin{array}{cccc} 2 & 3 & 4 & 5 \end{array} & \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ & 1/9 & 1 & 1 \\ & & 1 & 1/3 \\ & & & 2/3 \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array},$$

$$\boldsymbol{D}(y_3) = \begin{array}{cc} \begin{array}{cccc} 2 & 3 & 4 & 5 \end{array} & \\ \begin{bmatrix} 0 & 1 & 1/9 & 2 \\ & 1/9 & 2 & 2/3 \\ & & 1 & 1/3 \\ & & & 1/3 \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array}, \quad \boldsymbol{D}(y_4) = \begin{array}{cc} \begin{array}{cccc} 2 & 3 & 4 & 5 \end{array} & \\ \begin{bmatrix} 1/9 & 0 & 0 & 0 \\ & 3/9 & 1 & 1/3 \\ & & 0 & 0 \\ & & & 0 \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array}.$$

Для функций y_1, y_2, y_3 и y_4 суммы значений равны соответственно $5\frac{8}{9}$, $4\frac{1}{9}$, $7\frac{5}{9}$ и $1\frac{7}{9}$. Делаем попытку устранить противоречия функции y_3, имеющей наибольший для этого показатель. Некоторые противоречия удается устранить введением внутренней переменной z_1.

Последовательность итераций при поиске максимального разреза графа, соответствующего функции y_3, отражена в табл. 1, где пустая клетка в столбце B говорит о том, что соответствующая вершина перешла в множество A. Наибольшие степень вершины и величины $d-2d_A$ выделены. В результате получены $A = \{1, 4\}$ и $B = \{ 2, 3, 5\}$.

Таблица 1

Начальная ситуация		Шаг 1		Шаг 2	
B	d	B	$d-2d_A$	B	$d-2d_A$
1	$3\frac{1}{9}$	1	$\mathbf{2\frac{7}{9}}$		
2	$2\frac{7}{9}$	2	$-2\frac{7}{9}$	2	$-2\frac{7}{9}$
3	$2\frac{1}{9}$	3	$\frac{1}{9}$	3	$-1\frac{8}{9}$
4	$\mathbf{3\frac{4}{9}}$				
5	3	5	$2\frac{1}{3}$	5	$-1\frac{2}{3}$

Состояниям 1, 4 припишем $z_1 = 0$, а состояниям 2, 3, 5 – $z_1 = 1$. Появилась новая функция z_1^+, которая также имеет противоречия в задании. Ребра, соответствующие устраненным противоречиям, удаляются из всех графов. Получим следующие матрицы:

$$
\boldsymbol{U} = \begin{array}{c} x_1 \quad x_2 \quad x_3 \quad z_1 \quad q \\ \begin{bmatrix} - & - & - & 0 & 1 \\ 0 & - & - & 1 & 2 \\ 1 & 0 & - & 1 & 2 \\ 1 & 1 & - & 1 & 2 \\ - & - & 0 & 1 & 3 \\ - & 0 & 1 & 1 & 3 \\ - & 1 & 1 & 1 & 3 \\ - & - & - & 0 & 4 \\ - & 0 & - & 1 & 5 \\ - & 1 & 0 & 1 & 5 \\ - & 1 & 1 & 1 & 5 \end{bmatrix} \end{array}, \quad
\boldsymbol{V} = \begin{array}{c} z_1^+ \quad q^+ \quad y_1 \quad y_2 \quad y_3 \quad y_4 \\ \begin{bmatrix} 1 & 2 & 0 & - & 1 & 0 \\ 1 & 3 & 1 & 0 & - & 1 \\ 1 & 3 & 0 & - & 1 & 0 \\ 1 & 5 & - & 1 & 1 & - \\ - & - & 0 & 1 & - & 0 \\ 1 & 5 & 0 & 0 & - & 0 \\ 1 & 2 & 0 & - & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ - & - & 0 & 1 & 1 & - \\ - & - & 0 & 1 & 0 & - \\ 0 & 4 & - & 0 & 0 & 0 \end{bmatrix} \end{array},
$$

$$
\boldsymbol{D}(y_1) = \begin{array}{cc} \begin{array}{cccc} 2 & 3 & 4 & 5 \end{array} & \\ \begin{bmatrix} 0 & 0 & 1/9 & 0 \\ & 3/9 & 0 & 2/3 \\ & & 0 & 0 \\ & & & 0 \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array}, \quad
\boldsymbol{D}(y_2) = \begin{array}{cc} \begin{array}{cccc} 2 & 3 & 4 & 5 \end{array} & \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ & 1/9 & 0 & 2/3 \\ & & 0 & 1/3 \\ & & & 0 \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array}, \quad
\boldsymbol{D}(y_3) = \begin{array}{cc} \begin{array}{cccc} 2 & 3 & 4 & 5 \end{array} & \\ \begin{bmatrix} 0 & 0 & 1/9 & 0 \\ & 1/9 & 0 & 2/3 \\ & & 0 & 1/3 \\ & & & 0 \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array},
$$

$$
\boldsymbol{D}(y_4) = \begin{array}{cc} \begin{array}{cccc} 2 & 3 & 4 & 5 \end{array} & \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ & 3/9 & 0 & 1/3 \\ & & 0 & 0 \\ & & & 0 \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array}, \quad
\boldsymbol{D}(z_1^+) = \begin{array}{cc} \begin{array}{cccc} 2 & 3 & 4 & 5 \end{array} & \\ \begin{bmatrix} 0 & 0 & 1/9 & 0 \\ & 0 & 0 & 2/3 \\ & & 0 & 1/3 \\ & & & \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array}.
$$

Суммы элементов матриц $\boldsymbol{D}(y_1)$, $\boldsymbol{D}(y_2)$, $\boldsymbol{D}(y_3)$, $\boldsymbol{D}(y_4)$ и $\boldsymbol{D}(z_1^+)$ равны соответственно $1\frac{1}{9}$, $1\frac{1}{9}$, $1\frac{2}{9}$ 1 и $1\frac{1}{9}$. Видно, что наибольшим числом противоречий опять обладает функция y_3. Последовательность итераций при поиске максимального разреза графа, соответствующего функции y_3, отражена в табл. 2.

Таблица 2

Начальная ситуация		Шаг 1		Шаг 2	
B	d	B	$d-2d_A$	B	$d-2d_A$
1	$\frac{1}{9}$	1	$\frac{1}{9}$		
2	$\frac{7}{9}$	2	$-\frac{5}{9}$	2	$-\frac{5}{9}$
3	$\frac{4}{9}$	3	$-\frac{2}{9}$	3	$-\frac{2}{9}$
4	$\frac{1}{9}$	4	$\frac{1}{9}$	4	$-\frac{1}{9}$
5	**1**				

Введем переменную z_2, придав ей значение 0 для состояний 1, 5 и значение 1 для состояний 2, 3, 4. Получим матрицы

$$\boldsymbol{U}=\begin{array}{c}\begin{matrix} x_1 & x_2 & x_3 & z_1 & z_2 & q \end{matrix}\\ \begin{bmatrix} - & - & - & 0 & 0 & 1\\ 0 & - & - & 1 & 1 & 2\\ 1 & 0 & - & 1 & 1 & 2\\ 1 & 1 & - & 1 & 1 & 2\\ - & - & 0 & 1 & 1 & 3\\ - & 0 & 1 & 1 & 1 & 3\\ - & 1 & 1 & 1 & 1 & 3\\ - & - & - & 0 & 1 & 4\\ - & 0 & - & 1 & 0 & 5\\ - & 1 & 0 & 1 & 0 & 5\\ - & 1 & 1 & 1 & 0 & 5 \end{bmatrix}\end{array},\quad \boldsymbol{V}=\begin{array}{c}\begin{matrix} z_1^{+} & z_2^{+} & q^{+} & y_1 & y_2 & y_3 & y_4 \end{matrix}\\ \begin{bmatrix} 1 & 1 & 2 & 0 & - & 1 & 0\\ 1 & 1 & 3 & 1 & 0 & - & 1\\ 1 & 1 & 3 & 0 & - & 1 & 0\\ 1 & 0 & 5 & - & 1 & 1 & -\\ - & - & - & 0 & 1 & - & 0\\ 1 & 0 & 5 & 0 & 0 & - & 0\\ 1 & 1 & 2 & 0 & - & 0 & 0\\ 0 & 0 & 1 & 1 & 0 & 0 & 0\\ - & - & - & 0 & 1 & 1 & -\\ - & - & - & 0 & 1 & 0 & -\\ 0 & 1 & 4 & - & 0 & 0 & 0 \end{bmatrix}\end{array},$$

$$\boldsymbol{D}(y_1)=\begin{array}{cc}\begin{matrix}2 & 3 & 4 & 5\end{matrix} & \\ \begin{bmatrix} 0 & 0 & 0 & 0\\ & 3/9 & 0 & 0\\ & & 0 & 0\\ & & & 0 \end{bmatrix} & \begin{matrix}1\\2,\\3\\4\end{matrix}\end{array}\quad \boldsymbol{D}(y_2)=\begin{array}{cc}\begin{matrix}2 & 3 & 4 & 5\end{matrix} & \\ \begin{bmatrix} 0 & 0 & 0 & 0\\ & 1/9 & 0 & 0\\ & & 0 & 0\\ & & & 0 \end{bmatrix} & \begin{matrix}1\\2,\\3\\4\end{matrix}\end{array}\quad \boldsymbol{D}(y_3)=\begin{array}{cc}\begin{matrix}2 & 3 & 4 & 5\end{matrix} & \\ \begin{bmatrix} 0 & 0 & 0 & 0\\ & 1/9 & 0 & 0\\ & & 0 & 0\\ & & & 0 \end{bmatrix} & \begin{matrix}1\\2,\\3\\4\end{matrix}\end{array}$$

$$\boldsymbol{D}(y_4)=\begin{array}{cc}\begin{matrix}2 & 3 & 4 & 5\end{matrix} & \\ \begin{bmatrix} 0 & 0 & 0 & 0\\ & 1/3 & 0 & 0\\ & & 0 & 0\\ & & & 0 \end{bmatrix} & \begin{matrix}1\\2,\\3\\4\end{matrix}\end{array}\quad \boldsymbol{D}(z_1^{+})=\begin{array}{cc}\begin{matrix}2 & 3 & 4 & 5\end{matrix} & \\ \begin{bmatrix} 0 & 0 & 0 & 0\\ & 0 & 0 & 0\\ & & 0 & 0\\ & & & 0 \end{bmatrix} & \begin{matrix}1\\2,\\3\\4\end{matrix}\end{array}\quad \boldsymbol{D}(z_2^{+})=\begin{array}{cc}\begin{matrix}2 & 3 & 4 & 5\end{matrix} & \\ \begin{bmatrix} 0 & 0 & 0 & 0\\ & 1/3 & 0 & 0\\ & & 0 & 0\\ & & & 0 \end{bmatrix} & \begin{matrix}1\\2.\\3\\4\end{matrix}\end{array}$$

Для устранения оставшихся противоречий достаточно ввести переменную z_3 со значением 0 для состояния 2 и со значением 1 для состояния 3. Окончательно получим следующие матрицы:

$$
\boldsymbol{U}=\begin{array}{c}
\begin{array}{cccccc} x_1 & x_2 & x_3 & z_1 & z_2 & z_3 \end{array}\\
\begin{bmatrix}
- & - & - & 0 & 0 & -\\
0 & - & - & 1 & 1 & 0\\
1 & 0 & - & 1 & 1 & 0\\
1 & 1 & - & 1 & 1 & 0\\
- & - & 0 & 1 & 1 & 1\\
- & 0 & 1 & 1 & 1 & 1\\
- & 1 & 1 & 1 & 1 & 1\\
- & - & - & 0 & 1 & -\\
- & 0 & - & 1 & 0 & -\\
- & 1 & 0 & 1 & 0 & -\\
- & 1 & 1 & 1 & 0 & -
\end{bmatrix}
\end{array},\quad
\boldsymbol{V}=\begin{array}{c}
\begin{array}{ccccccc} z_1^+ & z_2^+ & z_3^+ & y_1 & y_2 & y_3 & y_4 \end{array}\\
\begin{bmatrix}
1 & 1 & 0 & 0 & - & 1 & 0\\
1 & 1 & 1 & 1 & 0 & - & 1\\
1 & 1 & 1 & 0 & - & 1 & 0\\
1 & 0 & - & - & 1 & 1 & -\\
- & - & - & 0 & 1 & - & 0\\
1 & 0 & - & 0 & 0 & - & 0\\
1 & 1 & 0 & 0 & - & 0 & 0\\
0 & 0 & - & 1 & 0 & 0 & 0\\
- & - & - & 0 & 1 & 1 & -\\
- & - & - & 0 & 1 & 0 & -\\
0 & 1 & - & - & 0 & 0 & 0
\end{bmatrix}
\end{array}.
$$

Для оценки качества полученного результата введем величину $S=\sum_{i,j} P_{ij}s_{ij}$, где P_{ij} – число переходов между состояниями q_i и q_j в любую сторону, s_{ij} – число переключаемых элементов при переходе между состояниями q_i и q_j. Суммирование ведется по всем переходам между состояниями в заданном автомате. Будем считать, что чем меньше эта величина, тем лучше результат кодирования. В рассмотренном примере $S = 40$. Если кодировать состояния произвольно, например использовать двоичное представление последовательности натуральных чисел с нулем, то получим результат в виде матриц

$$
\boldsymbol{U}=\begin{array}{c}
\begin{array}{cccccc} x_1 & x_2 & x_3 & z_1 & z_2 & z_3 \end{array}\\
\begin{bmatrix}
- & - & - & 0 & 0 & 0\\
0 & - & - & 0 & 0 & 1\\
1 & 0 & - & 0 & 0 & 1\\
1 & 1 & - & 0 & 0 & 1\\
- & - & 0 & 0 & 1 & 0\\
- & 0 & 1 & 0 & 1 & 0\\
- & 1 & 1 & 0 & 1 & 0\\
- & - & - & 0 & 1 & 1\\
- & 0 & - & 1 & - & -\\
- & 1 & 0 & 1 & - & -\\
- & 1 & 1 & 1 & - & -
\end{bmatrix}
\end{array},\quad
\boldsymbol{V}=\begin{array}{c}
\begin{array}{ccccccc} z_1^+ & z_2^+ & z_3^+ & y_1 & y_2 & y_3 & y_4 \end{array}\\
\begin{bmatrix}
0 & 0 & 1 & 0 & - & 1 & 0\\
0 & 1 & 0 & 1 & 0 & - & 1\\
0 & 1 & 0 & 0 & - & 1 & 0\\
1 & - & - & - & 1 & 1 & -\\
- & - & - & 0 & 1 & - & 0\\
1 & - & - & 0 & 0 & - & 0\\
0 & 0 & 1 & 0 & - & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 0 & 0\\
- & - & - & 0 & 1 & 1 & -\\
- & - & - & 0 & 1 & 0 & -\\
0 & 1 & 1 & - & 0 & 0 & 0
\end{bmatrix}
\end{array}.
$$

Для этого варианта кодирования $S = 44$.

Заключение

Предлагаемый метод кодирования состояний автомата рассчитан на использование его в автоматизированной системе логического проектирования. Сравнение результатов кодирования состояний изложенным методом и методом произвольного кодирования состояний показывает, что применение первого метода позволяет снизить интенсивность переключений элементов памяти при переходах между состояниями в автомате. Минимизация интенсивности переключений не противоречит минимизации количества элементов в схеме, что также ведет к снижению энергопотребления. Предлагаемый метод допускает частичное совместное решение этих двух задач на этапе кодирования состояний. При этом, как уже было сказано, существует два критерия минимизации: число различных элементарных конъюнкций в дизъюнктивных нормальных формах функций и количество аргументов, от которых зависит отдельная функция. Выбор их определяется используемой элементной базой. Этим двум критериям соответствуют два подхода к решению рассматриваемой задачи, один из которых, соответствующий второму упомянутому критерию, использован в данной работе.

Список литературы

1. Мурога, С. Системное проектирование сверхбольших интегральных схем. В 2-х кн. / С. Мурога. – М. : Мир, 1985. – Кн. 1. – 288 с.

2. Pedram, M. Power minimization in IC design: Principles and applications / M. Pedram // ACM Trans. Design Automat. Electron. Syst. – 1996. – Vol. 1. – P. 3–56.

3. Kashirova, L. State assignment of finite state machine for decrease of power dissipation / L. Kashirova, A. Keevallik, M. Meshkov // Second International Conference Computer-Aided Design of Discrete Devices, CAD DD'97, Minsk, Republic of Belarus, November 12–14, 1997. – Minsk : Institute of Engineering Cybernetics NASB, 1997. – Vol. 1. – P. 60–67.

4. Sudnitson, A. Partition search for FSM low power synthesis / A. Sudnitson // Fourth International Conference Computer-Aided Design of Discrete Devices, CAD DD'2001, Minsk, November 14–16, 2001. – Minsk : Institute of Engineering Cybernetics NASB, 2001. – Vol. 1. – P. 44–49.

5. Закревский, А.Д. Алгоритмы энергосберегающего кодирования состояний автомата / А.Д. Закревский // Информатика. – 2011. – № 1(29). – С. 68–78.

6. Закревский, А.Д. Логические основы проектирования дискретных устройств / А.Д. Закревский, Ю.В. Поттосин, Л.Д. Черемисинова. – М. : Физматлит, 2007. – 592 с.

7. Armstrong, D.B. A programmed algorithm for assigning internal codes for sequential machines / D.B. Armstrong // IRE Trans., EC-11. – 1962. – № 4. – P. 466–472.

8. Hartmanis, J. Algebraic Structure Theory of Sequential Machines / J. Hartmanis, R.E. Stearns. – N.Y. : Prentis-Hall Inc., 1966. – 208 p.

9. Поттосин, Ю.В. Кодирование состояний дискретного автомата, ориентированное на уменьшение энергопотребления реализующей схемы / Ю.В. Поттосин // Прикладная дискретная математика. – 2011. – № 4(14). – С. 62–71.

10. Поттосин, Ю.В. Итеративный способ кодирования состояний дискретного автомата / Ю.В. Поттосин // Автоматизация логического проектирования дискретных устройств : сб. науч.тр. – Минск : Ин-т техн. кибернетики АН БССР, 1980. – Вып. 2. – С. 16–26.

11. Поттосин, Ю.В. Основы теории проектирования цифровых устройств / Ю.В. Поттосин. – Saarbrücken : LAP LAMBERT Academic Publishing, 2011. – 336 с.

12. Macii, E. High-level power modeling, estimation and optimization / E. Macii, M. Pedram, F. Somenzi // IEEE Trans. on Comp.-Aided Design of IC and Systems. – 1998. – Vol. 17, № 11. – P. 1061–1079.

13. Закревский, А.Д. Раскраска графов при декомпозиции булевых функций / А.Д. Закревский // Логическое проектирование : сб. науч. тр. – Минск : Ин-т техн. кибернетики НАН Беларуси, 2000. – Вып. 5. – С. 83–97.

Поступила 20.06.2012

Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: pott@newman.bas-net.by

Yu.V. Pottosin

ITERATIVE LOW POWER STATE ASSIGNMENT OF A DISCRETE AUTOMATON

The problem of state assignment of a discrete automaton with the purpose of decreasing the switching activity of memory elements in the implementing circuit is considered. The determination of values of internal variables is reduced to finding a maximal cut in a weighted graph.