

DOI <https://doi.org/10.15407/usim.2018.02.042>

УДК: 004.2

Р.М. БАБАКОВ, канд. техн. наук, доцент, Донецкий нац. ун-т имени Василя Стуса (Украина), ул. 600-летия, 21, Винница, 21021, Украина, r.babakov@donnu.edu.ua

А.А. БАРКАЛОВ, д-р. техн. наук, профессор, Ун-т Зеленогурский (Польша), ул. Подгорная, 50, 65-246, Зеленая Гура (Польша), a.barkalov@imei.uz.zgora.pl

УМЕНЬШЕНИЕ МАКСИМАЛЬНОГО КОЛИЧЕСТВА СУЩЕСТВЕННЫХ ВХОДНЫХ ПЕРЕМЕННЫХ В МИКРОПРОГРАММНОМ АВТОМАТЕ С ОПЕРАЦИОННЫМ АВТОМАТОМ ПЕРЕХОДОВ

Предложено использование известного метода уменьшения максимльного количества существенных входных переменных для снижения аппаратурных затрат в логической схеме микропрограммного автомата с операционным автоматом переходов и заменой входных переменных.

Ключевые слова: микропрограммный автомат, операционный автомат переходов, замена входных переменных, аппаратурные затраты.

Введение

Устройство управления — один из основных блоков современных вычислительных систем [1]. Реализация устройства управления в виде микропрограммного автомата (МПА) способствует увеличению быстродействия схемы устройства при относительно высоких аппаратурных затратах [2]. В связи с наблюдаемым сегодня увеличением сложности имплементируемых МПА алгоритмов управления актуальна задача минимизации аппаратурных затрат в логической схеме МПА. Одно из решения данной задачи — разработка новых структур МПА с оптимизированными затратами аппаратуры [3, 4]. К таким структурам относятся, в частности, микропрограммный автомат с операционным автоматом переходов (МПА с ОАП), в котором схема формирова-

ния переходов реализуется в виде операционного автомата [5].

Помимо структурных решений, существует ряд подходов к уменьшению аппаратурных затрат, не приводящих к модификации структуры автомата [2, 3]. Возможность и целесообразность их использования в структуре МПА с ОАП к настоящему моменту остается неисследованной.

Анализ последних исследований и публикаций

В работе [5] предложена структура МПА с ОАП, в которой функция переходов реализуется совместной работой операционного автомата переходов и Z -подсхемы. ОАП представляет собой операционный автомат, выполняющий над кодом текущего состояния и

входными сигналами одну из операций переходов (ОП), определяемую кодом Z :

$$T = T(X, T, Z). \quad (1)$$

Каждая ОП реализует некоторое подмножество переходов автомата в соответствии с разбиением функции переходов на множество частичных функций [6]. Код текущего состояния автомата хранится в регистре памяти (РП), который в ОАП есть одновременно регистром исходных данных и регистром результата.

Код Z формируется отдельным блоком, называемым Z -подсхемой. В зависимости от входных сигналов отдельная ОП может быть сопоставлена либо каждому состоянию, либо каждому переходу автомата. В первом случае на вход Z -подсхемы подается только код текущего состояния, во втором — код текущего состояния и множество входных сигналов [7].

Также в структуре МПА с ОАП присутствует схема формирования микроопераций (СФМО), которая формирует множество выходных сигналов автомата и синтезируется по системе булевых уравнений каноническим способом [2, 5, 6].

В работе [8] рассмотрены модификации структур МПА с ОАП, в которых использован метод замены входных переменных. Данный метод заключается в том, что множество входных переменных X с помощью специальной M -подсхемы преобразуется во множество промежуточных переменных P таким образом, что $|X| \ll |P|$. Это позволяет уменьшить количество сигналов на входах Z -подсхемы, ОАП и СФМО, снизив таким образом затраты аппаратуры на реализацию данных блоков. Метод замены входных переменных эффективен в том случае, если уменьшение аппаратурных затрат в Z -подсхеме, ОАП и СФМО превышает затраты на реализацию M -подсхемы.

Уменьшение максимального количества существенных входных переменных

Одним из способов задания МПА служит граф-схема алгоритма (ГСА) [2]. Сегодня из-

вестен ряд методов оптимизации логической схемы МПА, в основе которых лежат допустимые преобразования исходной ГСА. Пример таких преобразований — введение в исходную ГСА дополнительных состояний [9, 10]. Данное действие сопровождается увеличением времени выполнения алгоритма, а также возможным ростом аппаратурных затрат, вызванным увеличением разрядности кода состояния. Если последствия введения дополнительных состояний не противоречат критериям проектирования, данный подход может быть использован для снижения затрат аппаратуры в схеме микропрограммного автомата с операционным автоматом переходов.

Пусть G — максимальное количество входных переменных автомата, существенно влияющих на автоматные переходы. Это значит, что ни при одном автоматном переходе не анализируется более G логических условий. Дальнейшее снижение аппаратурных затрат в рассматриваемых структурах МПА с ОАП, возможно путем уменьшения величины G . Для этого может быть использован известный подход, заключающийся в следующем [10]:

- Пустая операторная вершина может быть добавлена в любую ветвь, соединяющую выход одной условной вершины со входом другой условной вершины.
- Количество добавляемых вершин должно быть минимально достаточным для снижения величины G до требуемого значения.

В предельном случае G может быть равно единице, т.е. любой автоматный переход зависит не более чем от одного логического условия.

Рассмотрим применение данного подхода к МПА с ОАП с заменой входных переменных [8]. Пусть МПА задан ГСА G , отмеченной состояниями автомата Мили, фрагмент которой приведен на рис. 1 и определяет переходы из состояний $a_1 - a_4$. Выполним замену входных переменных по методике, изложенной в [10], в соответствии с табл. 1. После замены логических условий фрагмент ГСА G примет вид рис. 2.

Использование структуры МПА, рассмотренной в работе [8], позволяет сопоставлять

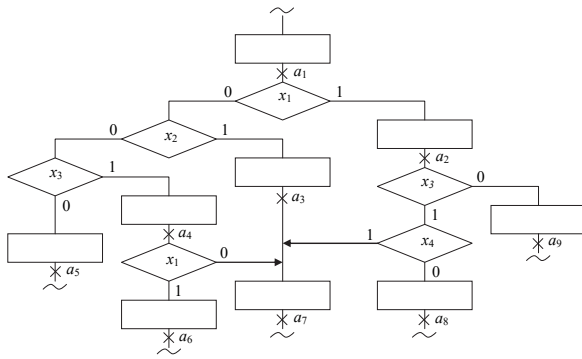


Рис. 1. Фрагмент ГСА Γ

Таблица 1. Замена входных переменных (фрагмент ГСА Γ)

a_i	p_1	p_2	p_3
a_1	X_1	x_2	X_3
a_2	X_4		X_3
a_3	—	—	—
a_4	x_1		

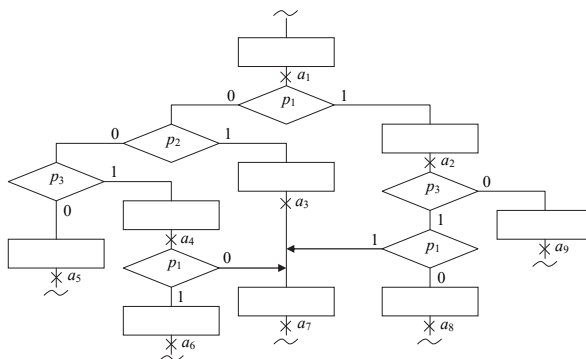


Рис. 2. Фрагмент ГСА Γ после замены входных переменных

операции переходов отдельным состояниям автомата. На рис. 2 переходы из состояний $a_1 - a_4$ зависят от разного количества входных переменных. Это не позволяет в рамках данного фрагмента сопоставить одну и ту же ОП более чем одному состоянию: состоянию a_1 должна быть сопоставлена ОП, использующая в качестве аргументов три переменных $p_1 - p_3$, состоянию a_2 — ОП, использующая две переменных p_1 и p_3 , состоянию a_3 — ОП, не использующая входные переменные в качестве аргументов, состоянию a_4 — ОП, использую-

щая одну переменную p_1 . Таким образом, все четыре ОП будут различными независимо от того, каким образом будет происходить замена переменных X переменными P .

Преобразуем фрагмент ГСА, изображенный на рис. 1, таким образом, чтобы любой переход в рамках фрагмента зависел не более чем от одного логического условия. Для этого добавим пустые операторные вершины в каждую ветвь, соединяющую условные вершины, что приведет к появлению трех дополнительных состояний $a_{10} - a_{12}$. Замена входных переменных в полученном фрагменте приводит к тому, что в каждой условной вершине будет записана переменная p_1 (рис. 3).

Проведенные преобразования дают формальную возможность сопоставить состояниям, выходы которых соединены с входом условной вершины (состояниям $a_1, a_2, a_4, a_{10}, a_{11}, a_{12}$) одну и ту же операцию переходов, использующую в качестве аргументов код текущего состояния и значение переменной p_1 . При этом состоянию a_3 по-прежнему должна быть сопоставлена отдельная ОП, реализующая безусловный переход, не зависящий от p_1 . Таким образом, минимально допустимое количество ОП для данного фрагмента равно двум.

Рассмотрим пример, демонстрирующий возможность использования двух ОП для реализации всех переходов в рамках фрагмента, изображенного на рис. 3, содержащего три типа переходов: безусловный переход, переход по условию $p_1 = 0$ и переход по условию $p_1 = 1$. В абстрактном автомате эти переходы выполняются под воздействием абстрактных входных сигналов $z_0 - z_2$, структурные коды которых определяются значениями переменной p_1 . Пусть $K_S(z_0) = \langle - \rangle$, $K_S(z_1) = \langle 0 \rangle$, $K_S(z_2) = \langle 1 \rangle$.

Проведем синтез для фрагмента ГСА на рис. 3. Зададим две абстрактных подалгебры переходов, определяемых выражениями (2) и (3), первая из которых реализует во фрагменте все условные переходы, вторая — единственный безусловный переход из состояния a_3 в состояние a_7 .

$$\left\{ \begin{aligned} G_{\delta_1} &= \langle A_{\delta_1}, F_{\delta_1} \rangle = \langle \{A_{\delta_1}, Z_{\delta_1}\}, \{\delta_1\} \rangle; \\ A_{\delta_1} &= \{a_1, \dots, a_{12}\}; \\ Z_{\delta_1} &= \{z_1, z_2\}; \\ \delta_1 &= \{ \langle a_1, z_1, a_{10} \rangle, \langle a_1, z_2, a_2 \rangle, \\ &\quad \langle a_2, z_1, a_9 \rangle, \langle a_2, z_2, a_{12} \rangle, \\ &\quad \langle a_{10}, z_1, a_{11} \rangle, \langle a_{10}, z_2, a_3 \rangle, \\ &\quad \langle a_{11}, z_1, a_5 \rangle, \langle a_{11}, z_2, a_4 \rangle, \\ &\quad \langle a_{12}, z_1, a_8 \rangle, \langle a_{12}, z_2, a_7 \rangle, \\ &\quad \langle a_4, z_1, a_7 \rangle, \langle a_4, z_2, a_6 \rangle \}. \end{aligned} \right. \quad (2)$$

$$\left\{ \begin{aligned} G_{\delta_2} &= \langle A_{\delta_2}, F_{\delta_2} \rangle = \langle \{A_{\delta_2}, Z_{\delta_2}\}, \{\delta_2\} \rangle; \\ A_{\delta_2} &= \{a_3, a_7\}; \\ Z_{\delta_2} &= \{z_0\}; \\ \delta_2 &= \{ \langle a_3, z_0, a_7 \rangle \}. \end{aligned} \right. \quad (3)$$

В системе (2) G_{δ_1} — абстрактная подалгебра переходов, носитель A_{δ_1} которой образован множеством состояний A_{δ_1} и множеством абстрактных входных сигналов Z_{δ_1} , а сигнатура F_{δ_1} образована единственной частичной функцией переходов δ_1 [7]. Функция δ_1 представлена множеством векторов вида $\langle a_i, z_k, a_j \rangle$, каждый из которых соответствует переходу из состояния a_i в состояние a_j под воздействием абстрактного входного сигнала z_k . Аналогичным образом организована абстрактная подалгебра G_{δ_2} , задаваемая выражением (3).

Сформируем промежуточную алгебру переходов G_{I_1} , изоморфную подалгебре G_{δ_1} . Промежуточные коды состояний выберем из множества натуральных чисел, промежуточные коды входных сигналов будем интерпретировать как логические величины, задаваемые на множестве {ИСТИНА, ЛОЖЬ}. Пусть $K_{I_1}(z_1) = \text{ЛОЖЬ}$, $K_{I_1}(z_2) = \text{ИСТИНА}$.

Предположим, что общее количество состояний в ГСА Γ таково, что минимально достаточная разрядность структурного кода состояния $R = 5$. Зададим промежуточные коды состояний в диапазоне $[0; 31]$ так, как показано в табл. 2.

Сопоставим состояниям $a_1, a_2, a_4, a_{10}, a_{11}, a_{12}$ следующую ОП:

$$O_i: K_{I_1}(a^{t+1}) = \begin{cases} (K_{I_1}(a^t) \oplus 10100_2 + 2_{10}) \bmod 32, & \text{если } p_1 = 0; \\ (K_{I_1}(a^t) \oplus 10100_2 + 5_{10}) \bmod 32, & \text{если } p_1 = 1. \end{cases} \quad (4)$$

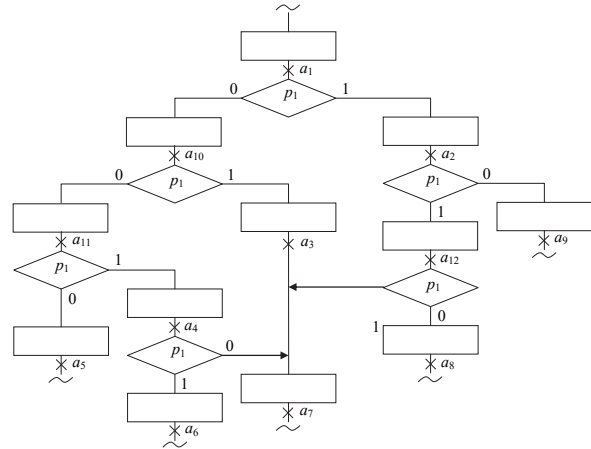


Рис. 3. Преобразованный фрагмент ГСА Γ

Таблица 2. Кодирование состояний (фрагмент ГСА Γ)

a_i	$K_I(a_i)$	a_i	$K_I(a_i)$
a_1	21	a_7	8
a_2	6	a_8	5
a_3	28	a_9	20
a_4	18	a_{10}	3
a_5	15	a_{11}	25
a_6	11	a_{12}	23

В данной интервальной функции промежуточный код текущего состояния $K_{I_1}(a^t)$ сначала приводится к форме пятиразрядного двоичного вектора, над которым выполняется поразрядная операция сложения по модулю 2 с вектором—константой 10100₂. Полученный результат интерпретируется как двоичное целое без знака и суммируется с целочисленной константой, значение которой зависит от значения переменной p_1 . Над результатом сложения выполняется операция «mod 32», приводящая результат ОП к диапазону $[0; 31]$.

Операция O_1 позволяет реализовать все переходы из состояний $a_1, a_2, a_4, a_{10}, a_{11}, a_{12}$ при

условии использования промежуточных кодов состояний из табл. 2. Например, переход из состояния a_{10} по условию $p_1 = 0$ выполняется следующим образом:

- $K_{I_1}(a_{10}) = 3_{10} = 00011_2$;
- $00011_1 \oplus 10100_2 = 10111_2$;
- $10111_2 = 23_{10}$;
- $23 + 2 = 25$;
- $25 \bmod 32 = 25 = K_{I_1}(a_{11})$.

С учетом табл. 2 и выражения (4), промежуточная алгебра G_{I_1} есть следующая система:

$$\left\{ \begin{array}{l} G_{I_1} = \langle A_{I_1}, F_{I_1} \rangle = \langle \{K_{I_1}(A_{\delta_1}), K_{I_1}(Z_{\delta_1})\}, \{O_1\} \rangle; \\ A_{I_1} = \{3, 5, 6, 8, 11, 15, 18, 20, 21, 23, 25, 28\}; \\ Z_{I_1} = \{\text{ЛОЖЬ, ИСТИНА}\} \\ O_1 = \{ \langle 21, \text{ЛОЖЬ}, 3 \rangle, \langle 21, \text{ИСТИНА}, 6 \rangle, \\ \langle 6, \text{ЛОЖЬ}, 20 \rangle, \langle 6, \text{ИСТИНА}, 23 \rangle, \\ \langle 3, \text{ЛОЖЬ}, 25 \rangle, \langle 3, \text{ИСТИНА}, 28 \rangle, \\ \langle 25, \text{ЛОЖЬ}, 15 \rangle, \langle 25, \text{ИСТИНА}, 18 \rangle, \\ \langle 23, \text{ЛОЖЬ}, 5 \rangle, \langle 23, \text{ИСТИНА}, 8 \rangle, \\ \langle 18, \text{ЛОЖЬ}, 8 \rangle, \langle 18, \text{ИСТИНА}, 11 \rangle \}. \end{array} \right. \quad (5)$$

Такая сигнатура алгебры G_{I_1} представлена единственной операцией переходов O_1 , образованной векторами вида $\langle K_{I_1}(a_i), K_{I_1}(z_k), K_{I_1}(a_j) \rangle$, каждый из которых может быть поставлен в соответствие одному из векторов частичной абстрактной функции переходов δ_1 .

Сформируем структурную подалгебру переходов G_{d_1} , изоморфную алгебрам (2) и (5) [11, 12]. Для это зададим состояниям из множества A_{δ_1} структурные коды, эквивалентные пятиразрядному двоичному представлению соответствующих промежуточных кодов из множества A_{I_1} . Получаемая в результате структурная подалгебра имеет вид (6).

В данной системе d_1 – частичная структурная функция переходов, представляемая множеством векторов вида $\langle K_S(a_i), K_S(z_k), K_S(a_j) \rangle$, каждый из которых может быть поставлен в соответствие одному из векторов частичной абстрактной функции переходов δ_1 . Таким образом, можно говорить о существова-

нии взаимного изоморфизма $G_{\delta_1} \leftrightarrow G_{I_1} \leftrightarrow G_{d_1}$.

Его суть состоит в том, что все автоматные переходы, определяемые абстрактной функцией переходов δ_1 , рассматриваются как преобразования скалярных величин (целых чисел) с помощью ОП O_1 и в структурном (двоичном) виде реализуются по закону, определяемому структурной функцией переходов d_1 .

$$\left\{ \begin{array}{l} G_{d_1} = \langle A_{d_1}, F_{d_1} \rangle = \langle \{K_S(A_{d_1}), K_S(Z_{d_1})\}, \{d_1\} \rangle; \\ K_S(A_{d_1}) = \{K_S(a_1), \dots, K_S(a_{12})\} = \\ = \{00011, 00101, 00110, 01000, 01011, 01111, \\ 10010, 10100, 10101, 10111, 11001, 11100\}; \\ K_S(Z_{d_1}) = \{K_S(z_1), K_S(z_2)\} = \{0, 1\}; \\ d_1 = \{ \langle 10101, 0, 00011 \rangle, \langle 10101, 1, 00110 \rangle, \\ \langle 00110, 0, 10100 \rangle, \langle 00110, 1, 10111 \rangle, \\ \langle 00011, 0, 11001 \rangle, \langle 00011, 1, 11100 \rangle, \\ \langle 11001, 0, 01111 \rangle, \langle 11001, 1, 10010 \rangle, \\ \langle 10111, 0, 00101 \rangle, \langle 10111, 1, 01000 \rangle, \\ \langle 10010, 0, 01000 \rangle, \langle 10010, 1, 01011 \rangle \}. \end{array} \right. \quad (6)$$

Сформируем промежуточную алгебру переходов G_{I_2} , изоморфную подалгебре G_{δ_2} . Поскольку $A_{\delta_2} = \{a_3, a_7\} \subseteq A_{\delta_1}$, структурные коды $K_S(a_3)$ и $K_S(a_7)$ в подалгебре G_{d_2} должны совпадать с соответствующими структурными кодами в подалгебре G_{d_1} . Зная структурные коды состояний и входных сигналов, можно немедленно сформировать структурную подалгебру G_{d_2} , изоморфную подалгебре (3):

$$\left\{ \begin{array}{l} G_{d_2} = \langle A_{d_2}, F_{d_2} \rangle = \langle \{K_S(A_{d_2}), K_S(Z_{d_2})\}, \{d_2\} \rangle; \\ K_S(A_{d_2}) = \{K_S(a_3), K_S(a_7)\} = \{11100, 01000\}; \\ K_S(Z_{d_2}) = \{K_S(z_0)\} = \{-\}; \\ d_2 = \{ \langle 11100, -, 01000 \rangle \}. \end{array} \right. \quad (7)$$

Отметим, что в рассматриваемом примере преобразование $K_S(a_3) = 11100$ в $K_S(a_7) = 01000$, определяемое частичной структурной функцией переходов функцией d_2 , может быть реализовано путем выполнения поразрядной операции сложения по модулю 2 кода $K_S(a_3)$ с двоичным вектором 10100. Это позволяет задать операцию O_2 промежуточной алгебры G_{I_2} в следующем виде:

$$O_2: K_{I_2}(a^{t+1}) = K_{I_2}(a^t) \oplus 10100_2. \quad (8)$$

Исходя из (8), промежуточные коды состояний, образующие носитель $K_{I_2}(A_{\delta_2})$ алгебры G_{I_2} , есть двоичные векторы, совпадающие со структурными кодами соответствующих состояний подалгебры (7): $K_{I_2}(a_3) = 11100$, $K_{I_2}(a_7) = 01000$. Поскольку сигнал z_0 не участвует каким-либо образом в операции O_2 , для промежуточной алгебры G_{I_2} он формальный и $K_{I_2}(Z_{\delta_2}) = Z_{\delta_2} = \{z_0\}$. Теперь промежуточная алгебра переходов G_{I_2} может быть задана следующей системой:

$$\begin{cases} G_{I_2} = \langle A_{I_2}, F_{I_2} \rangle = \\ \quad = \langle \{K_{I_2}(A_{\delta_2}), K_{I_2}(Z_{\delta_2})\}, \{O_2\} \rangle; \\ K_{I_2}(A_{\delta_2}) = \{11100, 01000\}; \\ K_{I_2}(Z_{\delta_2}) = Z_{\delta_2} = \{z_0\}; \\ O_2 = \{ \langle 11100, z_0, 01000 \rangle \}. \end{cases} \quad (9)$$

Отметим, что промежуточная алгебра (9) изоморфна подалгебрам (3) и (9). Существование изоморфизмов $G_{\delta_1} \leftrightarrow G_{I_1} \leftrightarrow G_{d_1}$ и $G_{\delta_2} \leftrightarrow G_{I_2} \leftrightarrow G_{d_2}$ подтверждает возможность реализации всех переходов во фрагменте ГСА на рис. 3 с учетом выбранных операций переходов [11, 12]. Графически это показано на рис. 4, где в каждой операторной вершине указаны промежуточные и структурные (двоичные) коды состояния, следующего за данной вершиной, а каждая дуга графа отмечена операцией, реализующей данный переход.

Таким образом, уменьшение максимального количества существенных входных переменных при использовании структуры МПА с ОАП и заменой входных переменных может в некоторых случаях приводить к уменьшению количества используемых ОП. Следствием этого есть уменьшение количества комбинационных схем в операционной части ОАП, приводящее к снижению аппаратных затрат в схеме автомата [13]. В тех случаях, когда уменьшение количества ОП сопровождается уменьшением разрядности кода ОП, имеет место дополнительное снижение аппаратур-

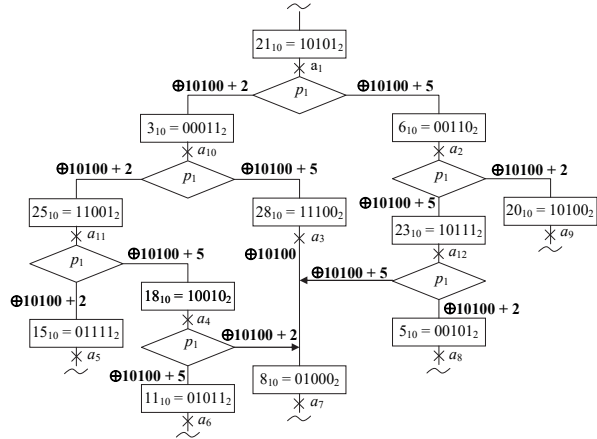


Рис. 4. Графическое представление результатов синтеза (фрагмент ГСА Γ)

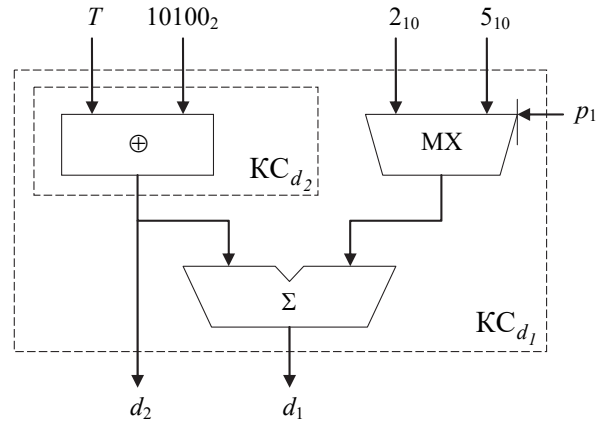


Рис. 5. Фрагмент функциональной схемы операционной части ОАП, соответствующий операциям переходов O_1 и O_2

ных затрат в схеме ОАП путем упрощения мультиплексора результата, а также в Z -подсхеме за счет уменьшения числа выходов.

Итак, отметим, что операция поразрядного сложения по модулю 2 с константой 10100_2 , являющаяся в выражении (4) внутренней функцией по отношению к ОП O_1 , в точности совпадает с ОП O_2 , определяемой выражением (8). Это позволяет синтезировать комбинационные схемы данных ОП так, как показано на рис. 5. В данной схеме все линии связи, за исключением одноразрядной линии, по которой подается сигнал p_1 , пятиразрядные, а в качестве констант 2_{10} и 5_{10} на входы мультиплексора МХ подаются их двоичные целочис-

ленные формы 00010_2 и 00101_2 соответственно. В результате комбинационная схема $КС_{d_2}$, реализующая ОП O_2 , — часть комбинационной схемы $КС_{d_1}$, реализующей операцию O_1 . Безусловно, данную особенность следует рассматривать как сугубо частный случай, однако подобные ситуации необходимо всегда иметь в виду при оптимизации логической схемы операционного автомата переходов.

Помимо снижения аппаратных затрат в схеме ОАП, уменьшение количества входных переменных, существенно влияющих на автоматные переходы, в случае автомата Мили приводит к дополнительному уменьшению числа входов блока СФМО. Отметим также, что при использовании в МПА с ОАП метода уменьшения количества существенных входных переменных следует помнить, что уменьшение величины G неизбежно сопровождается увеличением числа состояний автомата и в некоторых случаях может приводить к увеличению разрядности R кода состояния. Увеличение R приводит к росту аппаратных затрат во всех блоках логической схемы МПА независимо от его структурной организации.

Заключение

В статье предлагается способ оптимизации аппаратных затрат в структуре МПА с ОАП и заменой входных переменных. Способ заключается в применении к МПА с ОАП известного метода уменьшения количества существенных входных переменных. Данный подход не приводит к изменениям в структуре автомата, однако позволяет в некоторых случаях добиться уменьшения количества операций, используемых в ОАП, а также снизить аппаратные затраты в блоке СФМО автомата Мили.

Вместе с тем, численные показатели эффективности использования данного подхода не являются очевидными из структуры МПА и имплементируемого им алгоритма управления, и требуют проведения дополнительных исследований. Практическое применение рассмотренного подхода требует разработки соответствующих формализованных методов синтеза МПА с ОАП, ориентированных на использование современного элементного базиса программируемых интегральных схем [14].

СПИСОК ЛИТЕРАТУРЫ

1. Глушков В.М. Синтез цифровых автоматов, М.: Физматгиз, 1962, 476 с.
2. Баранов С.И. Синтез микропрограммных автоматов, Л.: Энергия, 1979, 232 с.
3. Logic Synthesis for FPGA-Based Finite State Machines / A. Barkalov, L. Titarenko, M. Kolopienczyk et al, Springer, 2016, 280 p.
4. DeMicheli G. Synthesis and Optimization of Digital Circuits, NY: McGraw-Hill, 1994, 576 p.
5. Баркалов А.А., Бабаков Р.М. Реализация функции переходов микропрограммного автомата на базе операционного автомата // УСиМ, 2015, № 5, С. 22–29.
6. Barkalov A.A., Babakov R.M. Algebraic Interpretation of a Microprogram Finite-State Machine with Datapath of Transitions, Cybernetics and Systems Analysis, 52, Issue 2, 2016, P. 191–198.
7. Бабаков Р.М., Ярош И.В. Формирование кодов операций переходов в микропрограммном автомате с операционным автоматом переходов // Сб. науч. трудов ДонНТУ. Серия «Информатика, кибернетика и вычислительная техника», Красноармейск, ДонНТУ, 2015, 1 (20), С. 11–16.
8. Бабаков Р.М., Баркалов А.А. Модификация микропрограммного автомата с операционным автоматом переходов и заменой входных переменных // УСиМ, 2017, № 6, С. 35–40.
9. Скларов В.А. Синтез автоматов на матричных БИС, Минск: Наука и техника, 1984, 256 с.
10. Баркалов А.А. Синтез устройств управления на программируемых логических устройствах, Донецк, ДонНТУ, 2002, 262 с.
11. Бабаков Р.М. Промежуточная алгебра переходов в микропрограммном автомате // Радиотехника, информатика, управление, 2016, № 1, С. 64–73.
12. Бабаков Р.М. Математическая модель микропрограммного автомата с операционным автоматом переходов // Сб. науч. трудов ДонНТУ. Серия «Информатика, кибернетика и вычислительная техника», Красноармейск, ДонНТУ, 2016, 1 (22), С. 54–57.
13. Бабаков Р.М., Ярош И.В. Операционный автомат переходов // Сб. науч. трудов ДонНТУ. Серия: «Вычислительная техника и автоматизация», Там же, 2015, 1 (28), С. 33–40.

14. Грушевицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах программируемой логики, БХВ-Петербург, 2002, 608 с.

Поступила 03.04.2018

REFERENCES

1. *Glushkov V.M.* Sintez tsifrovih avtomatov, Moscow, 1962, 476 p.
2. *Baranov S.I.* Sintez mikroprogrammnykh avtomatov, Leningrad, 1979, 232 p.
3. *Logic Synthesis for FPGA-Based Finite State Machines / A. Barkalov, L. Titarenko, M. Kolopienchyk et al.*, Springer, 2016, 280 p.
4. *DeMicheli G.* Synthesis and Optimization of Digital Circuits, New York, 1994, 576 p.
5. *Barkalov A.A., Babakov R.M.* Upravljajushie sistemi i mashini, N 5, 2015, P. 22–29.
6. *Barkalov A.A., Babakov R.M.* Algebraic Interpretation of a Microprogram Finite-State Machine with Datapath of Transitions, Cybernetics and Systems Analysis, Volume 52, Issue 2, 2016, P. 191–198.
7. *Babakov R.M., Yarosh I.V.* Sbornik nauchnykh trudov DonNTU. Seriya: «Informatika, kibernetika i vychislitel'naya tehnika», N 1, 2015, P. 11–16.
8. *Barkalov A.A., Babakov R.M.* Upravljajushie sistemi i mashini, N 6, 2016, P. 35–40.
9. *Sklyarov V.A.* Sintez avtomatov na matrichnykh BIS, Minsk, 1984, 256 p.
10. *Barkalov A.A.* Sintez ustroystv upravleniya na programmiruemykh logicheskikh ustroystvakh, Donetsk, 2002, 262 p.
11. *Babakov R.M.* Radiotekhnika, informatika, upravlenie. N 1, 2016, P. 64–73.
12. *Babakov R.M.* Sbornik nauchnykh trudov DonNTU. Seriya: «Informatika, kibernetika i vychislitel'naya tehnika», N 1, 2016, P. 54–57.
13. *Babakov R.M., Yarosh I.V.* Sbornik nauchnykh trudov DonNTU. Seriya: «Vychislitel'naya tehnika i avtomatizatsiya», N 1, 2015, P. 33–40.
14. *Grushvitskiy R.I., Mursaev A.X., Ugrumov E.P.* Proektirovanie sistem na mikroshemah programmiruemoj logiki, St. Peterburg, 2002, 608 p.

Received 03.04.2018

R.M. Babakov, PhD in Techn. Sciences, Associate Professor, Vasyl' Stus Donetsk National University, 600-richa str., 21, Vinnitsya, 21021, Ukraine, (+380 50) 295-0650, r.babakov@donnu.edu.ua

A.A. Barkalov, Doctor in Techn. Sciences, Professor, University of Zielona Gora, Podgorna str., 50, Zielona Gora, 65246, Poland, (+48 68) 326-2693, a.barkalov@imei.uz.zgora.pl

REDUCTION OF THE MAXIMUM NUMBER OF SIGNIFICANT INPUT VARIABLES IN THE MICROPROGRAM FINITE STATE MACHINE WITH DATAPATH OF TRANSITIONS

Introduction. The object of research is the microprogram finite state machine with datapath of transitions with the input variables replacement. In digital devices, microprogram finite state machine performs the functions of control unit and coordinates the functionality of other system blocks. One of the current scientific and practical problems is the reduction of hardware expenses in the logic circuit of the microprogram finite state machine. One way to solve this problem is the development of new microprogram structures and methods for their synthesis. One of such structures is a microprogram finite state machine with datapath of transitions. In this structure, the transition formation circuit is implemented in the form of datapath consisting of a separate operational blocks. Each operational block implements the individual law of converting state codes and input signals, realizing a disjoint subset of microprogram transitions. Herewith, the hardware expenses in the operational block do not depend or depend insignificantly on the number of microprogram transitions it implements. This allows to achieve the reducing of hardware expenses in comparison with the implementation of transition function of the finite state machine by the canonical method using the system of Boolean equations. The approach, consisting of the circuit representation forming the transitions of the microprogram finite state machine in the form of datapath, is called the operational realization of the transition function of the finite state machine. The use of the input variables replacement method makes it possible to further reduce the hardware expenses in some blocks of the finite state machine.

Purpose. The purpose of this article is to research a new way for optimization of the hardware expenses in logical circuit of finite state machine with datapath of the input variables transitions and replacement.

Methods. It is proposed to use the known method of decreasing the maximum number of significant input variables. The method consists of adding new states, which leads to a decrease in the number of input signals analyzed in one state machine transition.

Results. The proposed approach does not lead to a change in the structure of the microprogram finite state machine with datapath of the input variables transitions and replacement. It can be achieved by reducing the number of transition operations used. It is also possible to cut hardware expenses simultaneously in other structure blocks by reducing the number of intermediate signals replacing the finite state machine input signals.

Conclusion. The reduction of the maximum number of significant input variables allows, under certain conditions, to cut the amount of hardware expenses in the logic circuit of the microprogram finite state machine with datapath of transitions and the input variables replacement. The disadvantage of this approach is the increase in the execution time of the algorithm, interpreted by the finite state machine.

Keywords: microprogram finite state machine, datapath of transitions, replacement of input variables, hardware expenses.

Р.М. Бабаков, канд. техн. наук, доцент, Донецький нац. ун-т ім. Василя Стуса (Україна), вул. 600-річчя, 21, Вінниця, 21021, Україна, (+380 50) 295-0650, r.babakov@donnu.edu.ua

О.О. Баркалов, д-р. техн. наук, професор, У-т Зеленогурський (Польща), вул. Підгірна, 50, Зелена Гора, 65246, Польща, (+48 68) 326-2693, a.barkalov@imei.uz.zgora.pl

ЗМЕНШЕННЯ МАКСИМАЛЬНОЇ КІЛЬКОСТІ СУТТЄВИХ ВХІДНИХ В МІКРОПРОГРАМНОМУ АВТОМАТІ З ОПЕРАЦІЙНИМ АВТОМАТОМ ПЕРЕХОДІВ

Вступ. In this article, the object of research is the microprogram finite state machine with datapath of transitions with replacement of input variables. In digital devices, microprogram finite state machine performs the functions of control unit and coordinates the functionality of other blocks of the system. One of the topical scientific and practical problems is the reduction of hardware expenses in the logic circuit of the microprogram finite state machine. One way to solve this problem is the development of new microprogram finite state machine structures and methods for their synthesis. One of such structures is a microprogram finite state machine with datapath of transitions. In this structure, the transition formation circuit is implemented in the form of an datapath consisting of a number of separate operational blocks. Each operational block implements individual law of converting state codes and input signals, realizing a disjoint subset of microprogram transitions. Herewith, the hardware expenses in the operational block do not depend or depend insignificantly on the number of microprogram transitions it implements. This allows under certain conditions to achieve reducing of hardware expenses in comparison with the implementation of transition function of the finite state machine by the canonical method using the system of Boolean equations. The approach, consisting in the representation of the circuit for forming the transitions of the microprogram final-state machine in the form of an datapath, was called the operational realization of the transition function of the finite state machine. The use of the method of replacement of input variables in this structure makes it possible to further reduce the hardware expenses in some blocks of the finite state machine.

Мета. The purpose of this article is to research a new way for optimization of hardware expenses in logical circuit of finite state machine with datapath of transitions and replacement of input variables.

Методи. It is proposed to use the known method of decreasing the maximum number of significant input variables. The method consists in adding new states, which leads to a decrease in the number of input signals analyzed in one state machine transition.

Результати. The proposed approach does not lead to a change in the structure of the microprogram finite state machine with datapath of transitions and the replacement of input variables. Reducing the hardware expenses in the datapath of transitions can be achieved by reducing the number of transition operations used. It is also possible to simultaneously reduce hardware expnses in other blocks of the structure by reducing the number of intermediate signals replacing the input signals of the finite state machine.

Висновки. Reduction of the maximum number of significant input variables allows under certain conditions to reduce the amount of hardware expenses in the logic circuit of the microprogram finite state machine with datapath of transitions and the replacement of input variables. The disadvantage of this approach is the increase in the execution time of the algorithm, interpreted by the finite state machine.

Ключові слова: мікропрограмний автомат, операційний автомат переходів, заміна вхідних змінних, апаратурні витрати.