

DOI: 10.15587/2312-8372.2017.108845

ПРИМЕНЕНИЕ МЕТОДА ЗАМЕНЫ ВХОДНЫХ ПЕРЕМЕННЫХ В МИКРОПРОГРАММНОМ АВТОМАТЕ С ОПЕРАЦИОННЫМ АВТОМАТОМ ПЕРЕХОДОВ

Бабаков Р. М.

1. Введение

Одним из центральных узлов современных вычислительных систем является устройство управления, характеристики которого во многом определяют характеристики системы в целом [1, 2]. Наблюдаемый сегодня рост сложности вычислительных задач увеличивает требования к быстродействию вычислительных систем. Одним из способов повышения быстродействия устройств управления является использование высокоэффективных структурных решений, одним из которых является микропрограммный автомат (МПА) [2]. МПА с канонической структурой наряду с высоким быстродействием характеризуется высокими аппаратными затратами на реализацию логической схемы устройства [2, 3]. Это делает актуальной задачу разработки новых структур и методов синтеза МПА, направленных на оптимизацию аппаратных затрат при сохранении приемлемого быстродействия.

В настоящее время известно множество методов оптимизации МПА, приводящих к различным структурным реализациям [3–6]. Одним из таких методов является метод замены входных переменных, позволяющий уменьшить количество различных булевых термов в системе уравнений функции переходов автомата [5, 6]. Другим структурным решением является микропрограммный автомат с операционным автоматом переходов (МПА с ОАП), в котором функция переходов представляется в виде множества частичных функций и имеет схемную интерпретацию в виде операционного автомата [7, 8].

В работах [2, 5, 6] показана возможность совместного использования различных методов оптимизации, позволяющая при определенных условиях повысить эффективность результирующих структур МПА с точки зрения аппаратных затрат. В отношении микропрограммного автомата с операционным автоматом переходов вопросы применения известных методов оптимизации аппаратных затрат к настоящему моменту остаются неисследованными.

2. Объект исследования и его технологический аудит

Объектом данного исследования является микропрограммный автомат с операционным автоматом переходов [7, 8], структурная схема которого представлена на рис. 1 и в рамках данной работы обозначена символом U_1 .

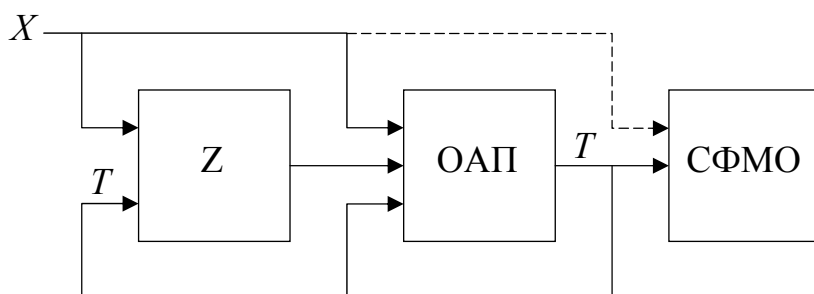


Рис. 1. Структурная схема микропрограммного автомата с операционным автоматом переходов (структура U_1)

Структура включает следующие блоки (рис. 1):

- ОАП: операционный автомат переходов, выполняющий преобразования кода состояния автомата с помощью множества операций переходов (ОП) под управлением кода ОП, представленного множеством сигналов Z ;
- Z -подсхема: формирует код операции переходов Z на основании кода текущего состояния T и сигналов логических условий (ЛУ) X ;
- СФМО: схема формирования микроопераций, формирующая множество микроопераций Y .

Наличие связи, показанной пунктиром на рис. 1, позволяет рассматривать данную структуру как автомат Мили, отсутствие связи – как автомат Мура.

В МПА с ОАП абстрактная функция переходов разбивается на множество частичных функций, каждая из которых реализует собственное подмножество автоматных переходов [9]. Разбиение выполняется таким образом, чтобы для каждой частичной функции мог быть задан единый закон арифметико-логического преобразования кодов состояний (операцию переходов, ОП), позволяющий реализовать все переходы из соответствующего подмножества. Данный закон реализуется в виде отдельной комбинационной схемы, затраты аппаратуры в которой не зависят от количества реализуемых ею автоматных переходов. Множество комбинационных схем, соответствующее множеству частичных функций переходов, образуют операционную часть (ОЧ), в которой выходы комбинационных схем мультиплексируются под управлением сигналов Z .

ОЧ совместно с регистром памяти (РП) образуют операционный автомат переходов, в котором РП является единственной регистровой схемой и в каждой операции выступает одновременно в роли регистра исходных данных и регистра результата (рис. 2).

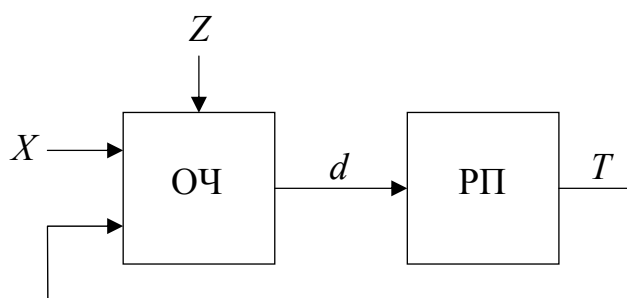


Рис. 2. Структурная схема операционного автомата переходов

Пусть МПА задан граф-схемой алгоритма (ГСА), содержащей множество состояний $A = \{a_1, \dots, a_M\}$ и множество логических условий $X = \{x_1, \dots, x_L\}$ [2]. С увеличением сложности алгоритмов управления, имплементируемых схемой МПА с ОАП, зачастую связано увеличение числа L логических условий, выступающих в качестве входных сигналов микропрограммного автомата. В структуре U_1 (рис. 1) увеличение L приводит к увеличению входных сигналов в Z -подсхеме по линии X . Это во многих случаях делает невозможным синтез данного блока в базисе запоминающих устройств (в частности, в базисе блочной памяти ПЛИС типа FPGA, [10]), поскольку требует использования запоминающих устройств с большим числом адресных входов [2, 5].

Единственным приемлемым базисом для синтеза Z -подсхемы остается базис LUT-элементов, являющийся основным ресурсом кристалла ПЛИС [10]. При этом увеличение аппаратных затрат в логической схеме МПА с ОАП, связанное с реализацией Z -подсхемы в базисе LUT-элементов, ухудшает такие характеристики схемы, как стоимость, габариты, энергопотребление, надежность и др.

Таким образом, одним из способов уменьшения аппаратных затрат в логической схеме микропрограммного автомата с операционным автоматом переходов является реализация Z -подсхемы в базисе блочной памяти ПЛИС FPGA. Это возможно при условии снижения количества входных сигналов Z -подсхемы. Последнее может быть достигнуто путем увеличения числа уровней преобразования логических сигналов в схеме автомата за счет применения известного метода замены входных переменных [5, 6].

3. Цель и задачи исследований

Цель исследования – применение метода замены входных переменных к микропрограммному автомату с операционным автоматом переходов. Это позволило бы при определенных условиях снизить аппаратные затраты в логической схеме МПА с ОАП. Полученная в результате структура может оказаться более предпочтительной по сравнению с другими известными структурами микропрограммных автоматов.

Для достижения поставленной цели необходимо:

1. Разработать структурную схему МПА с ОАП и заменой входных переменных.
2. Определить условия эффективности данной структуры по сравнению со структурой U_1 .
3. Продемонстрировать возможность применения данного подхода.

4. Исследование существующих решений проблемы

Разработка методов оптимизации аппаратных затрат в логических схемах устройств управления является классической задачей теории цифровых автоматов. По данному направлению в литературе следует, в первую очередь, выделить работы [2–6], содержащие ряд оригинальных методик и структурных решений, ориентированных на использование гетерогенного элементного базиса.

Также в рамках данной проблемы могут быть обозначены следующие направления:

- структурная декомпозиция логической схемы автомата [11–13];
- специальное кодирование состояний [14–16];
- синтез с учетом особенностей элементного базиса [17–19].

Так, в работе [11] предложен метод преобразования кодов объектов, в соответствии с которым в структуре МПА увеличивается количество уровней преобразования логических сигналов. При этом за счет упрощения каждого из уровней, а также за счет обеспечения возможности реализации схемы в гетерогенном элементном базисе, суммарные затраты аппаратуры в схеме автомата снижаются. Подобные методы, ориентированные на конкретный элементный базис (PAL, FPGA), изложены в работах [12, 13].

Другой подход к снижению аппаратурных затрат заключается в использовании специальных способов кодирования состояний [14–16]. При этом, помимо снижения стоимости схемы, в ряде случаев достигается также увеличение быстродействия, снижение энергопотребления, увеличение помехозащищенности и другие положительные эффекты.

Также альтернативным вариантом достижения экономии аппаратурных затрат является использование особенностей элементного базиса. В современных ПЛИС типа FPGA встречаются такие типовые функциональные узлы, как мультиплексоры, регистровые схемы, блочная и распределенная память и другие. По мнению авторов работ [17–19], использование данных узлов в схеме МПА позволяет уменьшить число используемых LUT-элементов, задействовав их для реализации других блоков вычислительной системы (например, операционного устройства).

Таким образом, результаты анализа позволяют сделать вывод о существовании множества различных путей оптимизации аппаратурных затрат в схемах цифровых устройств управления. Тем не менее, возможность и целесообразность использования данных подходов в микропрограммном автомате с операционным автоматом переходов сегодня остается неисследованной.

5. Методы исследования

Характерной особенностью МПА является то, что почти всегда для любого состояния $a_i \in A$ выполняется условие:

$$|X(a_i)| \ll L, \quad (1)$$

где $X(a_i)$ – подмножество X , элементы которого существенно влияют на переходы из состояния a_i [6]. Выполнение для заданной ГСА неравенства (1) позволяет применить к МПА с ОАП известный метод замены входных переменных, соответствующих сигналам логических условий [5, 6].

Пусть МПА с ОАП задан ГСА, в которой любой переход зависит не более чем от G логических условий. В этом случае на вход Z -подсхемы и ОАП будет подаваться множество сигналов $P = \{p_1, \dots, p_G\}$, а в структуру автомата добавляется специальная M -подсхема, реализующая функцию:

$$P = P(X, T). \quad (2)$$

Добавление М-подсхемы в структуру U_1 приводит к структуре МПА с ОАП и заменой входных переменных (рис. 3). Обозначим данную структуру символом U_2 .

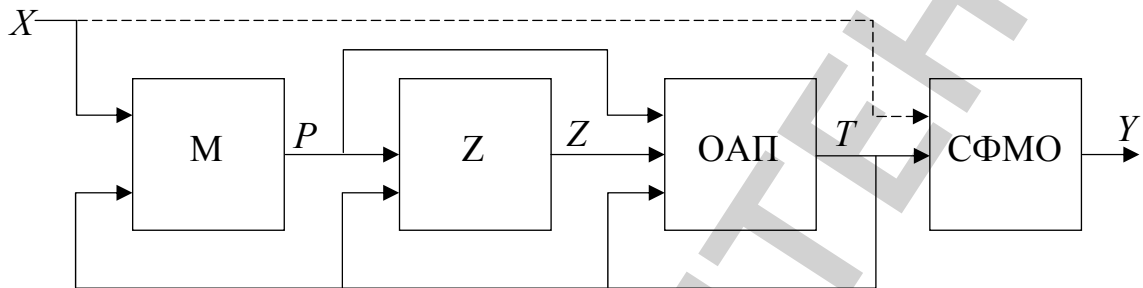


Рис. 3. Структурная схема микропрограммного автомата с операционным автоматом переходов и заменой входных переменных (структура U_2)

Здесь Z-подсхема реализует функцию:

$$Z = Z(P, T), \quad (3)$$

и строится по системе уравнений:

$$\begin{cases} z_1 = z_1(T_1, \dots, T_R, p_1, \dots, p_G); \\ z_2 = z_2(T_1, \dots, T_R, p_1, \dots, p_G); \\ \dots \\ z_{R_z} = z_{R_z}(T_1, \dots, T_R, p_1, \dots, p_G), \end{cases} \quad (4)$$

где R – разрядность кода состояния автомата. М-подсхема синтезируется в соответствии со следующей системой уравнений:

$$\begin{cases} p_1 = p_1(T_1, \dots, T_R, x_1, \dots, x_L); \\ p_2 = p_2(T_1, \dots, T_R, x_1, \dots, x_L); \\ \dots \\ p_G = p_G(T_1, \dots, T_R, x_1, \dots, x_L). \end{cases} \quad (5)$$

Составим неравенство, при выполнении которого аппаратные затраты в логической схеме МПА со структурой U_2 будут меньше, чем в схеме со структурой U_1 :

$$H_M^{U_2} + H_Z^{U_2} + H_{Oч}^{U_2} + H_{рп}^{U_2} + H_{сфмо}^{U_2} < H_Z^{U_1} + H_{Oч}^{U_1} + H_{рп}^{U_1} + H_{сфмо}^{U_1}. \quad (6)$$

Здесь каждое слагаемое есть численно выраженные затраты аппаратуры в блоке, определяемом нижним индексом, входящим в структуру МПА, определяемую верхним индексом. При использовании базиса ПЛИС FPGA в качестве единицы измерения аппаратурных затрат могут быть использованы таблицы поиска (LUT, Look-Up Tables), являющиеся регулярными функциональными узлами FPGA [10].

Полагая, что $H_{рп}^{U_1} = H_{рп}^{U_2}$ и $H_{сфмо}^{U_1} = H_{сфмо}^{U_2}$, сведем неравенство (6) к следующему:

$$H_M^{U_2} + H_Z^{U_2} + H_{Oч}^{U_2} < H_Z^{U_1} + H_{Oч}^{U_1}. \quad (7)$$

В левой части неравенства дополнительным слагаемым является величина $H_M^{U_2}$, соответствующая аппаратурным затратам в М-подсхеме. При построении схемы МПА в гетерогенном элементном базисе М-подсхема может быть реализована на стандартных мультиплексорах или ПЛМ с большим числом входов и малым числом выходов [5, 6].

Метод кодирования входных переменных предполагает, что рост аппаратурных затрат за счет добавления М-подсхемы сопровождается одновременным их снижением в Z-подсхеме. Так, кодирование входных переменных позволяет снизить число входов Z-подсхемы до величины $(G + R)$, где R – разрядность структурного (двоичного) кода состояния автомата. Это не только уменьшает затраты в Z-подсхеме, но и упрощает ее реализацию в базисе запоминающих устройств. Таким образом, при $G \ll L$ в общем случае можно ожидать выполнения неравенства $H_Z^{U_2} < H_Z^{U_1}$.

Что касается ОАП, то кодирование входных переменных не оказывает принципиального влияния на его внутреннюю структуру. В структуре U_2 на вход ОАП будет поступать G структурных (двоичных) сигналов p_i вместо L сигналов логических условий в случае структуры U_1 . Это может приводить к появлению общих схемных фрагментов в различных комбинационных схемах ОЧ, как следствие, уменьшению аппаратурных затрат в схеме ОАП структуры U_1 по сравнению со структурой U_2 .

6. Результаты исследований

В качестве результатов проведенных исследований проиллюстрируем предложенный подход на примере синтеза МПА со структурой U_2 . Пусть МПА задан ГСА Γ , отмеченной состояниями автомата Мура (рис. 4). ГСА содержит $M=10$ состояний $a_0 - a_9$, для кодирования которых требуется $R=4$ двоичных разряда. Применим к ГСА Γ метод замены входных переменных. Согласно рис. 4, значение $G=2$, то есть $P = \{p_1, p_2\}$.

Сформируем таблицу замены входных переменных $x_1 - x_5$ переменными p_1, p_2 , для чего применим методику, изложенную в [5]. Таблица кодирования логических условий представлена табл. 1.

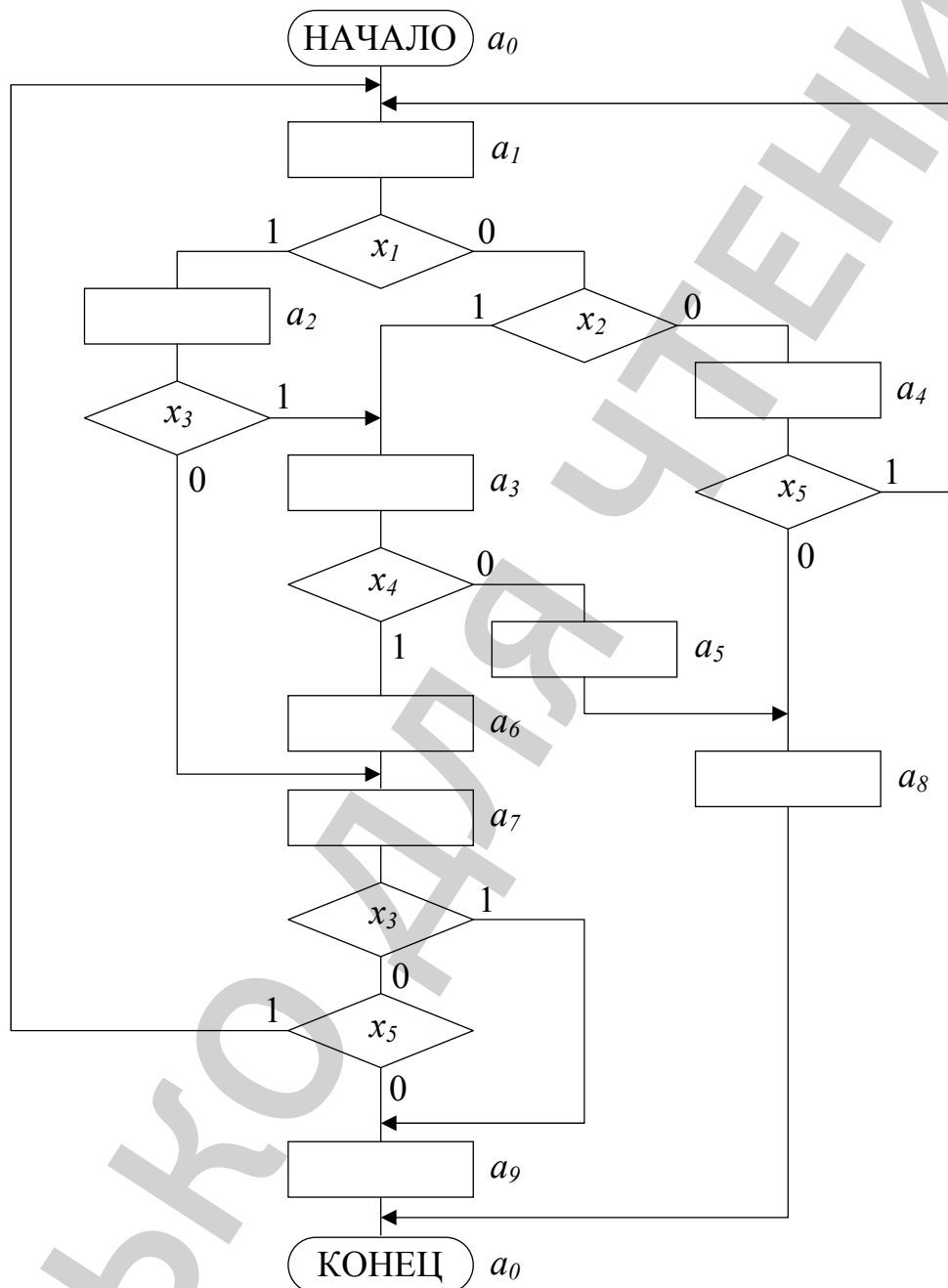


Рис. 4. Граф-схема алгоритма Γ

Таблица 1

Таблица замены входных переменных (ГСА Γ)

a_i	p_1	p_2	a_i	p_1	p_2
a_0	—	—	a_5	—	—
a_1	x_1	x_2	a_6	—	—
a_2	x_3	—	a_7	x_3	x_5
a_3	x_4	—	a_8	—	—

a_4	–	x_5	a_9	–	–
-------	---	-------	-------	---	---

Выполним синтез автомата по ГСА, изображенной на рис. 4, в которой вместо переменных x_1, x_3, x_4 используется переменная p_1 , вместо переменных x_2, x_5 – переменная p_2 .

По аналогии с [20], закодируем состояния $a_0 - a_9$ промежуточными кодами K_I из множества целых чисел в диапазоне $[0; 15]$ следующим образом:

$$K_I(a_0) = 3, K_I(a_1) = 8, K_I(a_2) = 4, K_I(a_3) = 0, K_I(a_4) = 13, \\ K_I(a_5) = 11, K_I(a_6) = 5, K_I(a_7) = 2, K_I(a_8) = 6, K_I(a_9) = 7.$$

Структурные (двоичные) коды состояний, представляемые двоичными векторами $\langle T_1, T_2, T_3, T_4 \rangle$, выберем равными соответствующим промежуточным кодам, представленным в формате четырехразрядных двоичных беззнаковых чисел:

$$K_S(a_0) = 0011, K_S(a_1) = 1000, K_S(a_2) = 0100, K_S(a_3) = 0000, \\ K_S(a_4) = 1101, K_S(a_5) = 1011, K_S(a_6) = 0101, K_S(a_7) = 0010, \\ K_S(a_8) = 0110, K_S(a_9) = 0111.$$

Зададим следующие операции переходов, определенные на множестве промежуточных кодов состояний:

$$O_1: K_I(a^{t+1}) = K_I(a^t) + 5, \quad (8)$$

$$O_2: K_I(a^{t+1}) = K_I(a^t) + 11, \quad (9)$$

$$O_3: K_I(a^{t+1}) = K_I(a^t) \times 4, \quad (10)$$

$$O_4: K_I(a^{t+1}) = K_I(a^t) / 2. \quad (11)$$

При использовании данных ОП будем полагать, что результат каждой ОП всегда приводится к диапазону $[0; 15]$, что выражается математической операцией «mod 16» (взятие остатка от целочисленного деления на 16).

Закодируем операции $O_1 - O_4$ уникальными двухразрядными кодами $\langle z_1, z_2 \rangle$:

$$K_Z(O_1) = 00, K_Z(O_2) = 01, K_Z(O_3) = 10, K_Z(O_4) = 11.$$

Сопоставим каждому переходу автомата одну из ОП $O_1 - O_4$ так, как показано в таблице переходов (табл. 2) [2, 5].

В табл. 2 столбец O_i содержит операцию переходов, выполняемую при переходе из состояния a_m по условию P_b .

В графической форме результат синтеза показан на рис. 5.

Таблица 2

Таблица переходов микропрограммного автомата со структурой U_2

a_m	$K_S(a_m)$	a_s	$K_S(a_s)$	P_h	O_h	Z_h	h
a_0	0011	a_1	1000	1	O_1	–	1
a_1	1000	a_2	0100	P_1	O_4	$z_1 z_2$	2
		a_3	0000	$\bar{P}_1 P_2$	O_3	z_1	3
		a_4	1101	$\bar{P}_1 \bar{P}_2$	O_1	–	4
a_2	0100	a_3	0000	P_1	O_3	z_1	5
		a_7	0010	\bar{P}_1	O_4	$z_1 z_2$	6
a_3	0000	a_5	1011	P_1	O_2	z_2	7
		a_6	0101	\bar{P}_1	O_1	–	8
a_4	1101	a_1	1000	P_2	O_2	z_2	9
		a_8	0110	\bar{P}_2	O_4	$z_1 z_2$	10
a_5	1011	a_8	0110	1	O_2	z_2	11
a_6	0101	a_7	0010	1	O_4	$z_1 z_2$	12
a_7	0010	a_9	0111	P_1	O_1	–	13
		a_1	1000	$\bar{P}_1 P_2$	O_3	z_1	14
		a_9	0111	$\bar{P}_1 \bar{P}_2$	O_1	–	15
a_8	0110	a_0	0011	1	O_4	$z_1 z_2$	16
a_9	0111	a_0	0011	1	O_4	$z_1 z_2$	17

По табл. 2 строится система уравнений (4), каждое уравнение которой определяется следующим выражением:

$$z_i = \bigvee_{h=1}^H C_{ih} A_m^h P_h. \quad (12)$$

В выражении (12) использованы следующие обозначения:

- C_{ih} : булева переменная, принимающая единичное значение, если и только если в строке h операционной таблицы переходов в столбце Z_h записана переменная z_i ;
- A_m^h : значение столбца $K_S(a_m)$ в строке h ;
- P_h : конъюнктивный терм, образованный переменными из множества P , указанными в столбце P_h строки h таблицы переходов.

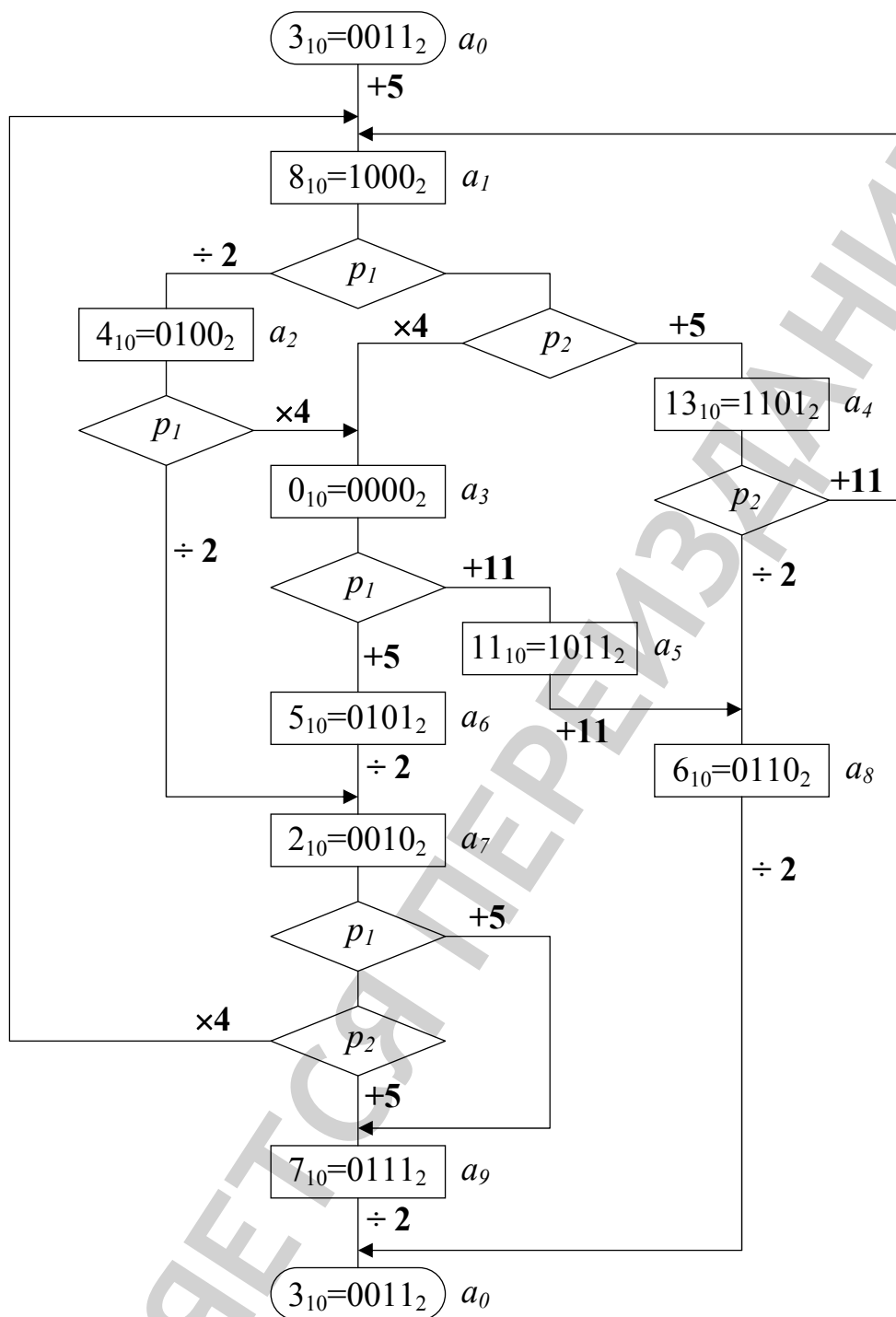


Рис. 5. Результат синтеза микропрограммного автомата со структурой U_2

Для синтеза М-подсхемы необходимо построить систему уравнений (5) по табл. 1. Методика синтеза М-подсхемы, учитывающая особенности используемого элементного базиса, подробно изложена в [2].

Рассмотренный пример синтеза МПА с ОАП со структурой U_2 позволяет сделать вывод о возможности использования метода замены входных переменных в микропрограммном автомате с операционным автоматом переходов. При этом условия целесообразности применения данного подхода определяются выражением (7).

7. SWOT-анализ результатов исследований

Strength. Преимуществом предлагаемой в работе структуры микропрограммного автомата с операционным автоматом переходов и заменой входных переменных является меньшее число входных сигналов Z-подсхемы по сравнению со структурой-прототипом. Это дает возможность использовать для синтеза Z-подсхемы базис запоминающих устройств, представителем которого могут выступать модули блочной памяти ПЛИС типа FPGA.

В то же время добавляемая в структуру M-подсхема может быть реализована в базисе мультиплексоров, также являющихся стандартными функциональными узлами современных FPGA. Достигаемая за счет этого экономия LUT-элементов позволяет использовать их для синтеза других узлов МПА или вычислительной системы. В конечном итоге данный подход позволяет синтезировать на базе той же серии ПЛИС автоматы более высокой сложности, либо использовать для заданного автомата микросхемы меньшей сложности.

Weakness. Уменьшение количества входов Z-подсхемы в предлагаемой структуре МПА с ОАП возможно за счет введения дополнительного блока – M-подсхемы. Если выбранная ПЛИС не содержит в своем составе достаточное количество мультиплексоров, для реализации M-подсхемы может быть использован базис LUT-элементов, что может нивелировать аппаратный выигрыш в Z-подсхеме. В любом случае эффективность предлагаемой структуры по критерию аппаратных затрат в сравнении со структурой-прототипом не является абсолютной и является различной в каждом конкретном случае. Если для заданного автомата предлагаемая в данной работе структура обладает большими затратами аппаратуры, чем структура-прототип, для проектирования МПА следует использовать структуру-прототип (или другую, более оптимальную структуру МПА, отвечающую заданным критериям проектирования).

Opportunities. Практическое использование устройств управления на базе микропрограммного автомата с операционным автоматом переходов и заменой входных переменных требует разработки формализованных методов синтеза данной структуры. Также должна быть определена область ее эффективно применения, выражаемая совокупностью значений или диапазонов значений параметров автомата. Современный процесс проектирования цифровых устройств невозможен без использования специализированных САПР, что приводит к задаче разработки синтезируемой VHDL-модели, ориентированной на определенный элементный базис (например, базис FPGA) [9]. Также с помощью подобной модели для заданного автомата могут быть экспериментально определены затраты аппаратуры, выражаемые, например, в количестве используемых LUT-элементов и блоков внутренней памяти ПЛИС.

Threats. В настоящее время известно множество структур МПА, ориентированных на оптимизацию различных параметров схемы автомата. При этом до сих пор не существует однозначных способов выбора той или иной структуры автомата в каждом конкретном случае. Попытки разработать специализированные САПР, позволяющие в автоматическом или автоматизированном режиме выбрать оптимальную структуру автомата из множества известных структур в соответствии с заданными критериями проектирования, остаются нереализо-

ванными. Выбор оптимальной структуры «вручную» требует от проектировщика знания большого числа различных структур и методов их синтеза. При этом всегда существует риск, что выбранная проектировщиком структура не является оптимальной в конкретном случае. Это негативно отражается на стоимости конечных изделий на базе МПА с выбранной структурой.

Таким образом, SWOT-анализ результатов исследований позволяет обозначить основные направления для дальнейших исследований. Среди них: разработка формального метода синтеза микропрограммного автомата с операционным автоматом переходов и заменой входных переменных, построение синтезируемой VHDL-модели данной структуры и определение области эффективного использования последней.

8. Выводы

1. В данной работе предложена новая структура микропрограммного автомата с операционным автоматом переходов, использующая принцип замены входных переменных. В данной структуре уменьшается число входных переменных Z -подсхемы, что способствует уменьшению числа компонент булевых термов в системе уравнений, реализуемой Z -подсхемой. В результате имеет место снижение аппаратных затрат в Z -подсхеме при одновременном увеличении затрат за счет введения в структуру МПА с ОАП дополнительной M -подсхемы, реализующей замену входных переменных.

2. Сформулировано условие эффективности предложенной структуры по критерию аппаратных затрат в сравнении с известной структурой МПА с ОАП без замены входных переменных. Условие определяется неравенством (7), каждая часть которого есть сумма аппаратных затрат в отдельных блоках соответствующей структуры.

3. Рассмотрен пример синтеза МПА с ОАП и заменой входных переменных по граф-схеме алгоритма. Данный пример отражает основные этапы синтеза автомата и в дальнейшем может быть использован при разработке формализованного метода синтеза данной структуры.

Литература

1. Glushkov, V. M. Sintez tsifrovyyh avtomatov [Text] / V. M. Glushkov. – Moscow: Fizmatgiz, 1962. – 476 p.
2. Baranov, S. I. Sintez mikroprogrammnyh avtomatov [Text] / S. I. Baranov. – Leningrad: Energiia, 1979. – 232 p.
3. Drozdov, E. A. Optimizatsiia struktur tsifrovyyh avtomatov [Text] / E. A. Drozdov. – Moscow: Soetskoe radio, 1975. – 352 p.
4. Barkalov, A. A. Sintez mikroprogrammnyh ustroystv upravleniia [Text] / A. A. Barkalov, A. V. Palagin. – Kyiv: Institute of Cybernetics of NAS of Ukraine, 1997. – 135 p.
5. Barkalov, A. A. Sintez ustroystv upravleniia na programmirovemykh logicheskikh ustroystvakh [Text] / A. A. Barkalov. – Donetsk: Donetsk National Technical University, 2002. – 262 p.

6. Baranov, S. I. Tsifrovye ustroistva na programmirovemyh BIS s matrichnoi strukturoi [Text] / S. I. Baranov, V. A. Skliarov. – Moscow: Radio i sviaz', 1986. – 272 p.
7. Barkalov, A. A. Operational formation of state codes in microprogram automata [Text] / A. A. Barkalov, R. M. Babakov // Cybernetics and Systems Analysis. – 2011. – Vol. 47, No. 2. – P. 193–197. doi:[10.1007/s10559-011-9301-y](https://doi.org/10.1007/s10559-011-9301-y)
8. Babakov, R. Research of efficiency of microprogram final-state machine with datapath of transitions [Text] / R. Babakov, A. Barkalov, L. Titarenko // 2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM). – IEEE, 2017. – P. 203–206. doi:[10.1109/cadsm.2017.7916115](https://doi.org/10.1109/cadsm.2017.7916115)
9. Babakov, R. M. Algebraic Interpretation of a Microprogram Finite-State Machine with Datapath of Transitions [Text] / R. M. Babakov, A. A. Barkalov. – Cybernetics and Systems Analysis. – 2016. – Vol. 52, No. 2. – P. 191–198. doi:[10.1007/s10559-016-9814-5](https://doi.org/10.1007/s10559-016-9814-5)
10. Grushvitskii, R. I. Proektirovanie sistem na mikroshemah programmirovemoi logiki [Text] / R. I. Grushvitskii, A. H. Mursaev, E. P. Ugriumov. – St. Pererburg: BHV-Peterburg, 2002. – 608 p.
11. Barkalov, A. A. Structural decomposition as a tool for the optimization of an FPGA-based implementation of a mealy FSM [Text] / A. A. Barkalov, L. A. Titarenko, A. A. Barkalov // Cybernetics and Systems Analysis. – 2012. – Vol. 48, No. 2. – P. 313–322. doi:[10.1007/s10559-012-9410-2](https://doi.org/10.1007/s10559-012-9410-2)
12. Kania, D. Decomposition-based synthesis and its application in PAL-oriented technology mapping [Text] / D. Kania // Proceedings of the 26th Euromicro Conference. EUROMICRO 2000. Informatics: Inventing the Future. – Maastricht: IEEE Computetr Society Press, 2000. – P. 138–145. doi:[10.1109/eurmic.2000.874626](https://doi.org/10.1109/eurmic.2000.874626)
13. Rawski, M. The Influence of Functional Decomposition on Modern Digital Design Process [Text] / M. Rawski, T. Luba, Z. Jachna, P. Tomaszewicz // Design of Embedded Control Systems. – Boston: Springer US, 2005. – P. 193–204. doi:[10.1007/0-387-28327-7_17](https://doi.org/10.1007/0-387-28327-7_17)
14. Amann, R. Optimal state chains and state codes in finite state machines [Text] / R. Amann, U. G. Baitinger // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 1989. – Vol. 8, No. 2. – P. 153–170. doi:[10.1109/43.21834](https://doi.org/10.1109/43.21834)
15. Bacchetta, P. Low-power state assignment techniques for finite state machines [Text] / P. Bacchetta, L. Daldos, D. Sciuto, C. Silvano // 2000 IEEE International Symposium on Circuits and Systems. Emerging Technologies for the 21st Century. Proceedings (IEEE Cat No.00CH36353). – 2000. – Vol. 2. – P. 641–644. doi:[10.1109/iscas.2000.856410](https://doi.org/10.1109/iscas.2000.856410)
16. Czerwinski, R. B09: State assignment method for high speed FSMS [Text] / R. Czerwinski, D. Kania // IFAC Proceedings Volumes. – 2004. – Vol. 37, No. 20. – P. 216–221. doi:[10.1016/s1474-6670\(17\)30599-2](https://doi.org/10.1016/s1474-6670(17)30599-2)
17. Barkalov, A. A. Partial reconfiguration of compositional microprogram control units implemented on FPGAs [Text] / A. A. Barkalov, M. Wegrzyn,

R. Wiszniewski // IFAC Proceedings Volumes. – Vol. 39, No. 21. – P. 116–119. doi:[10.1016/s1474-6670\(17\)30169-6](https://doi.org/10.1016/s1474-6670(17)30169-6)

18. Bomar, B. W. Implementation of microprogrammed control in FPGAs [Text] / B. W. Bomar // IEEE Transactions on Industrial Electronics. – 2002. – Vol. 49, No. 2. – P. 415–422. doi:[10.1109/41.993275](https://doi.org/10.1109/41.993275)

19. Solovjev, V. Synthesis of sequential circuits on programmable logic devices based on new models of finite state machines [Text] / V. Solovjev // Proceedings Euromicro Symposium on Digital Systems Design. – IEEE Computer Society Press, 2001. – P. 170–173. doi:[10.1109/dsd.2001.952274](https://doi.org/10.1109/dsd.2001.952274)

20. Babakov, R. M. Intermediate algebra of transitions in microprogram final-state machine [Text] / R. M. Babakov // Radio Electronics, Computer Science, Control. – 2016. – No. 1. – P. 64–73. doi:[10.15588/1607-3274-2016-1-8](https://doi.org/10.15588/1607-3274-2016-1-8)

НЕ ЯВЛЯЕТСЯ ПЕРВИЗУМЕМ