

А.А. Баркалов, Л.А. Титаренко, К.Н. Ефименко

Оптимизация схем композиционных микропрограммных устройств управления, реализуемых на ПЛИС

Ключевые слова: композиционное микропрограммное устройство управления, ПЛИС, оптимизация, адресация микрокоманд

Key-words: compositional microprogram control unit, FPGA, optimization, microinstruction addressing

1. Введение

Устройство управления (УУ) является важным блоком любой цифровой системы [1]. Для задания закона функционирования УУ часто используется модель цифрового автомата [2]. Одной из актуальных задач, возникающих при синтезе логических схем УУ, является задача уменьшения аппаратных затрат при сохранении заданного быстродействия [3]. Методы решения этой задачи как от используемой модели УУ, так и от характеристик алгоритма управления и элементов, из которых реализуется схема. В настоящее время для реализации схем УУ широко используются программируемые логические схемы (ПЛИС) типа FPGA (field-programmable gate array) [4,5]. Подобные ПЛИС включают логические элементы табличного типа (LUT, look-up tables) и встроенные блоки памяти ЕМВ (embedded memory block). Далее условимся использовать обозначение ПЛИС для FPGA, ЛЭ (логический элемент) для LUT, и ВБП (встроенный блок памяти) для ЕМВ. Особенностью ЛЭ является ограниченное число входов (4-6), что приводит к необходимости функциональной декомпозиции [6] реализуемых систем булевых функций. Это приводит к увеличению числа уровней в схеме УУ и уменьшению быстродействия цифровой системы в целом. В настоящей работе предлагаются методы уменьшения аппаратных затрат при реализации на ПЛИС схем, интерпретирующих линейные граф-схемы алгоритма (ГСА) [7]. Для линейных ГСА характерно наличие цепочек безусловно выполняемых микрокоманд, что позволяет использовать для их интерпретации автомат Мура, названный в [8] композиционным микропрограммным устройством управления (КМУУ).

2. Основы композиционных микропрограммных устройств управления

Пусть алгоритм управления представлен ГСА Γ , характеризуемой множествами вершин $V = E_1 \cup E_2 \cup \{b_0, b_E\}$ и дуг E , соединяющих эти вершины. Здесь -- E_1 множество операторных вершин, содержащих наборы микроопераций из множества $Y = \{y_1, \dots, y_N\}$; E_2 -- множество условных вершин, содержащих элементы множества логических условий $X = \{x_1, \dots, x_L\}$; b_0, b_E -- соответственно начальная и конечная вершины ГСА. Введем ряд определений необходимых для дальнейшего изложения и взятых из [8].

Определение 1. Операторной линейной цепью (ОЛЦ) ГСА Γ называется конечная последовательность операторных вершин $\alpha_g = \langle b_{g1}, \dots, b_{gF_g} \rangle$, такая, что для любой пары её соседних компонент которой существует дуга $\langle b_{gi}, b_{gi+1} \rangle \in E$, где $i=1, \dots, F_g-1$.

Определение 2. Операторная вершина $b_q \in D^g$, где D^g -- множество вершин, входящих в ОЛЦ α_g , называется входом ОЛЦ α_g , если существует дуга $\langle b_t, b_q \rangle \in E$, где $b_t \notin D^g$.

Определение 3. Операторная вершина $b_q \in D^g$, называется выходом ОЛЦ α_g , если существует дуга $\langle b_q, b_t \rangle \in E$, где $b_t \notin D^g$.

Пусть для ГСА Γ получены множество ОЛЦ $C = \{\alpha_1, \dots, \alpha_G\}$, соответствующее разбиению минимальной мощности множества E_1 на G классов, каждый из которых отвечает определению 1. Пусть $I(\Gamma)$, $O(\Gamma)$ -- соответственно множество входов и выходов ОЛЦ $\alpha_g \in C$. Каждая вершина $b_q \in E_1$ соответствует микрокоманде, имеющей адрес $A(b_q)$. Выполним адресацию микрокоманд так, чтобы выполнялось условие

$$A(b_{gi+1}) = A(b_{gi}) + 1, \quad (1)$$

где $g \in \{1, \dots, G\}$, $i \in \{1, \dots, F_g-1\}$. В этом случае ГСА Γ может быть интерпретирована КМУУ U_1 [7], структурная схема которого приведена на рис.1. Это устройство включает блок адресации микрокоманд (БАМ), счетчик (СТ), блок микроопераций (БМО) и триггер выборки (ТВ). КМУУ U_1 функционирует следующим образом.

По сигналу Start в СТ записывается нулевой адрес, соответствующий началу микропрограммы, интерпретирующей ГСА Γ , Одновременно триггер ТВ устанавливается в единичное состояние (Fetch=1) и микрокоманды могут выбираться из блока памяти БМО. Если СТ содержит адрес $A(b_q)$ и $b_q \notin O(\Gamma)$, то одновременно с набором микроопераций $Y(b_q)$, записанных в вершине $b_q \in E_1$, БМО формирует сигнал y_0 . Если $y_0=1$, то содержимое СТ увеличивается на единицу по сигналу Clock. При этом происходит безусловный переход, соответствующий (1). Если $b_q \in O(\Gamma)$, то сигнал y_0 не формируется, а блок БАМ вырабатывает функции возбуждения СТ

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

В этом случае по сигналу Clock в СТ формируется адрес перехода из выхода некоторой ОЛЦ $\alpha_g \in C$. Если $\langle b_q, b_E \rangle \in E$, то блок БМО формирует сигнал y_E , вызывающий установку триггера ТВ в нулевое состояние. При этом Fetch=0, выборка микрокоманд прекращается и КМУУ U_1 прекращает функционирование.

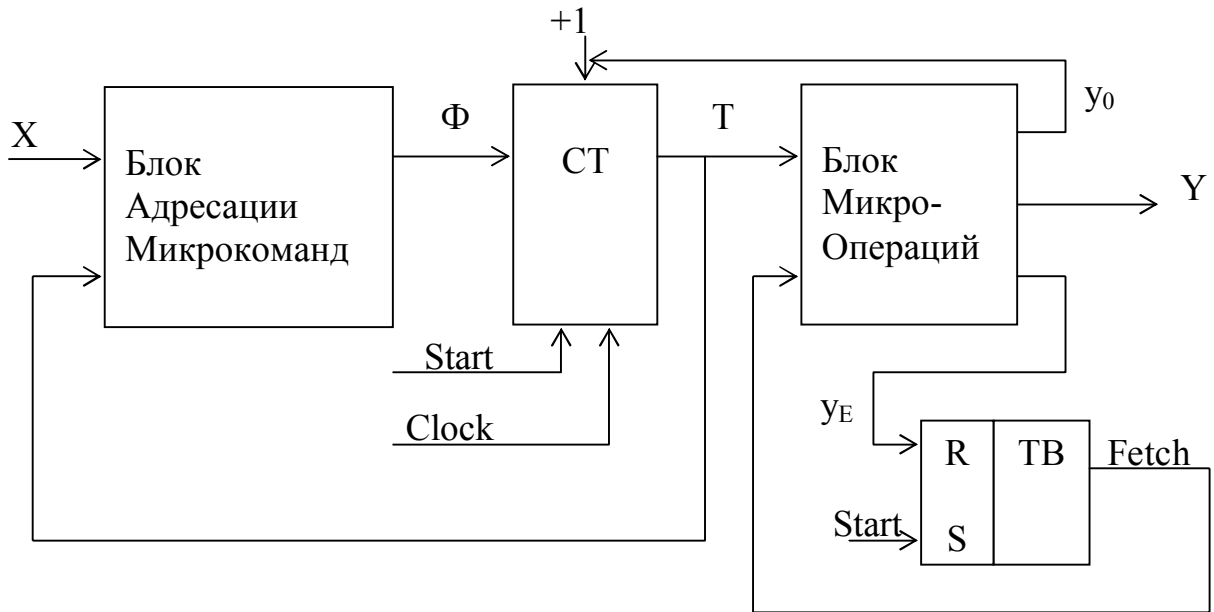


Рис. 1. Структурная схема КМУУ U_1

При реализации схемы КМУУ U_1 на ПЛИС, схема БАМ, СТ и ТВ строится на ЛЭ, а схема БМО реализуется на ВБП. Таким образом, модель позволяет сбалансировано использовать возможности современных ПЛИС. Недостатком модели является значительное число функций обратной связи T , совпадающее с разрядностью адреса микрокоманды

$$R_A = \lceil \log_2 M \rceil, \quad (2)$$

где $M = |E_1|$. В настоящей работе предлагаются два метода адресации микрокоманд, позволяющих уменьшить число сигналов обратной связи в блоке БАМ.

3. Основная идея предлагаемых методов

Выходные сигналы Y КМУУ зависят от содержимого СТ, то есть

$$Y = Y(T);$$

$$y_0 = y_0(T); \quad (4)$$

$$y_E = y_E(T).$$

Таким образом, КМУУ является автоматом Мура, а адреса микрокоманд соответствуют кодам его состояний. Характерной чертой КМУУ является то,

что в качестве кодов, участвующих в формировании функций (2), используются только адреса выходов ОЛЦ. Очевидно, что для однозначной идентификации G ОЛЦ достаточно

$$R_G = \lceil \log_2 G \rceil \quad (5)$$

переменных.

Если начальные адреса микрокоманд изменить так, чтобы для идентификации выходов использовать $R_0 < R_A$ переменных, то сложность функций (2) уменьшится. Очевидно, что в пределе $R_0 = R_G$. Итак, первая возможность оптимизации схемы БАМ заключается в однозначной идентификации выходов ОЛЦ $\alpha_g \in C$ с использованием $R_0 < R_A$ переменных.

Характерной особенностью автомата Мура является наличие классов псевдоэквивалентных состояний [8], соответствующих состояниям эквивалентного автомата Мили. Для оптимизации схемы автомата Мура состояния должны быть закодированы так, чтобы большинство классов псевдоэквивалентных состояний (в пределе -- все классы) представлялись обобщенными интервалами кодирующего пространства. Эта же идея может быть использована для оптимизации схемы КМУУ U_1 .

Назовем ОЛЦ $\alpha_g, \alpha_i \in C$ псевдоэквивалентными ОЛЦ, если их выходы соединены с входом одной и той же вершины ГСА Γ . Найдем разбиение $\Pi_C = \{V_1, \dots, V_l\}$ множества ОЛЦ C на классы псевдоэквивалентных ОЛЦ. Выполним адресацию микрокоманд так, чтобы каждый класс $V_i \in \Pi_C$ представлялся одним обобщенным интервалом R_A -- размерного булева пространства. Такой интервал используется в качестве кода $K(V_i)$ класса $V_i \in \Pi_C$ и участвует в формировании функций (2).

Назовем первый подход уточненной адресацией микрокоманд, а второй -- блочной адресацией микрокоманд. Рассмотрим методы синтеза КМУУ U_2 и U_3 , основанных на этих подходах. При этом структурные схемы КМУУ U_2 -- U_3 совпадают, а разница в аппаратных затратах связана только с разными адресами выходов ОЛЦ $\alpha_g \in C$. Рассмотрим оба предлагаемых метода оптимизации на примере ГСА Γ_1 (рис. 2).

Первый этап синтеза для обоих КМУУ U_2 и U_3 связан с формированием множества ОЛЦ C , удовлетворяющего следующим условиям [8]:

$$\begin{aligned} D^i \cap D^j &= \emptyset \quad (i \neq j, i, j \in \{1, \dots, G\}); \\ D^1 \cup D^2 \cup \dots \cup D^G &= E_1 \\ G &\rightarrow \min. \end{aligned} \quad (6)$$

Методы решения этой задачи достаточно известны [7] и в случае КМУУ, реализуемого по ГСА Γ_1 , дают следующее решение: $C = \{\alpha_1, \dots, \alpha_4\}$, где $\alpha_1 = \langle b_1, b_2, b_3 \rangle$, $\alpha_2 = \langle b_4, b_5, b_6 \rangle$, $\alpha_3 = \langle b_7, b_8 \rangle$, $\alpha_4 = \langle b_9, b_{10}, b_{11} \rangle$. Итак, для ГСА Γ_1

$G=4$, $M=11$, $R_A=4$, $R_G=2$. Для адресации микрокоманд используются элементы множества $T=\{T_1, \dots, T_4\}$. Как правило, счетчик СТ имеет входы типа D [6], поэтому множество $\Phi=\{D_1, \dots, D_4\}$. Условимся в дальнейшем обозначать как $U_i(\Gamma_j)$ тот факт, что ГСА Γ_j интерпретируется КМУУ U_i ($i=1,2,3$).

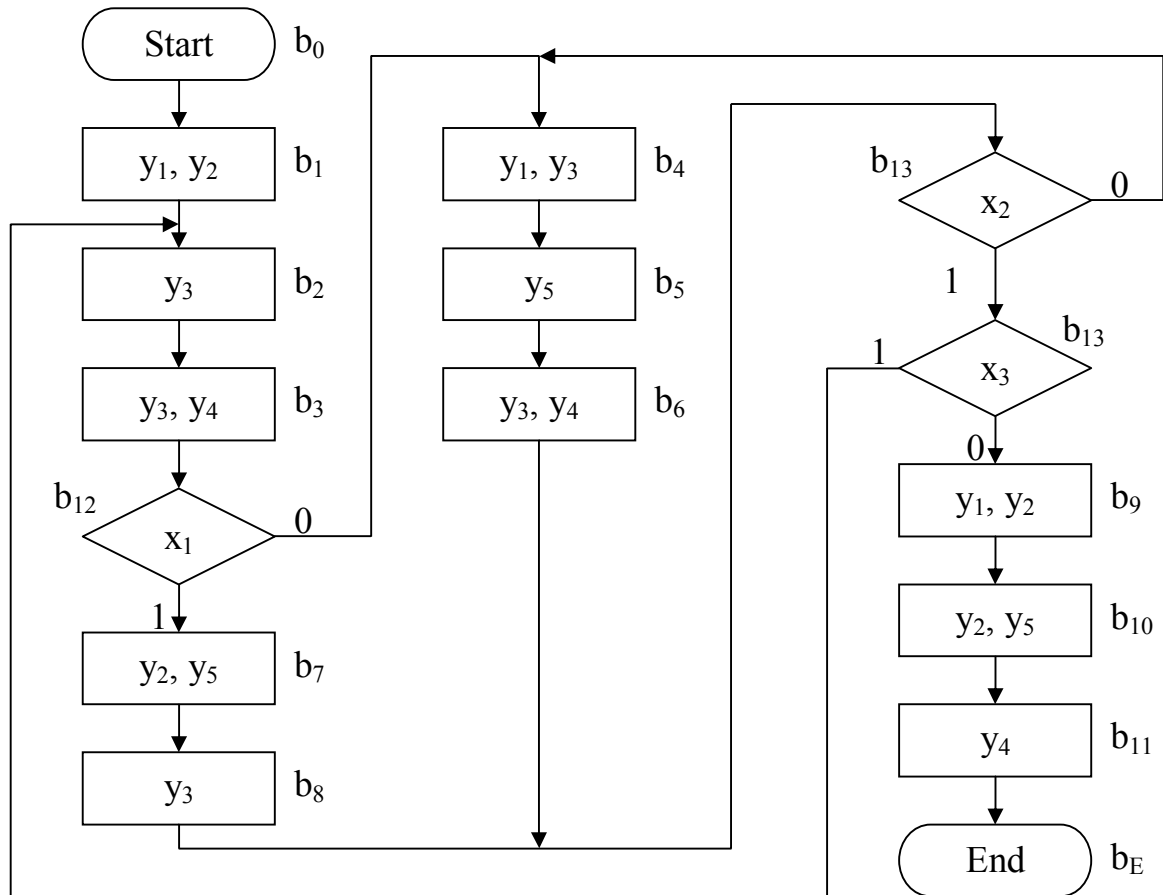


Рис. 2. Исходная граф-схема алгоритма Γ_1

4. Синтез КМУУ с уточненной адресацией микрокоманд

Для адресации микрокоманд КМУУ U_1 используется методика из [8]. При этом алгоритм состоит из G шагов. На первом шаге адресуются микрокоманды, соответствующие вершинам ОЛЦ, вход которой связан с выходом начальной вершины b_0 . В случае КМУУ $U_1(\Gamma_1)$ адресация начинается с ОЛЦ α_1 . На g -м шаге выполняется адресация микрокоманд, соответствующих вершинам ОЛЦ α_g ($g=1, \dots, G$). При этом в качестве первого адреса для ОЛЦ α_g берется увеличенный на единицу адрес выхода ОЛЦ α_{g-1} . В случае КМУУ $U_1(\Gamma_1)$ адреса микрокоманд показаны в табл.1.

Таблица 1

Адресация микрокоманд КМУУ $U_1(\Gamma_1)$

$T_1 T_2 T_3 T_4$	b_q	$T_1 T_2 T_3 T_4$	b_q	$T_1 T_2 T_3 T_4$	b_q	$T_1 T_2 T_3 T_4$	b_q
0000	b_1	0011	b_4	0110	b_7	1001	b_{10}
0001	b_2	0100	b_5	0111	b_8	1010	b_{11}
0010	b_3	0101	b_6	1000	b_9	1011	*

Назовем рассмотренный ранее процесс стандартной адресацией микрокоманд, а полученные адреса – стандартными адресами микрокоманд. Пусть $O_g, A(O_g)$ означают соответственно выход ОЛЦ $\alpha_g \in C$ и адрес этого выхода.

Из табл.1 следует, что $A(O_1)=0010, A(O_2)=0101, A(O_3)=0111, A(O_4)=1010$. Очевидно, адрес $A(O_4)$ не имеет значения, так как этот адрес не участвует в формировании функции (2). Это связано с тем, что при достижении выхода $O_4=b_{11}$ КМУУ формирует сигнал y_E и функционирование прекращается. Анализ адресов $A(O_1) -- A(O_3)$ показывает, что для однозначной идентификации выходов $O_1 -- O_3$ достаточно три переменные: T_1, T_2, T_3 . При этом O_1 характеризуется кодом $K(O_1)=001, O_2 -- K(O_2)=010$ и $O_3 -- K(O_3)=011$. Так как $R_G=2$, то необходимо сделать попытку такой адресации микрокоманд, чтобы выходы ОЛЦ $\alpha_1 -- \alpha_3$ характеризовались двумя переменными $T_r \in T$. Пусть $C_1 \subseteq C$ – множество ОЛЦ, выходы которых не связаны с входом вершины b_E . Как было показано ранее, только выходы ОЛЦ $\alpha_g \in C_1$ должны идентифицироваться при помощи R_G переменных.

В настоящей работе предлагается метод уточненной адресации, минимизирующий величину R_0 :

1. Положить $R_0=R_G$ и выполнить стандартную адресацию микрокоманд.
2. Построить таблицу адресации, имеющую, 2^{R_0} столбцов, отмеченных переменными T_1, \dots, T_{R_0} , и $2^{R_A-R_0}$ строк, отмеченных переменными $T_{R_0+1}, \dots, T_{R_A}$.
3. Если выходы ОЛЦ $\alpha_i, \alpha_j \in C_1$, где $j>i$, находятся в одном столбце таблицы адресации, то осуществим сдвиг информации, начиная с первой компоненты ОЛЦ $\alpha_j \in C_1$. Освобождающиеся клетки таблицы адресации заполним знаком *. Сдвиг продолжаем до тех пор, пока выходы O_i и O_j не окажутся в разных столбцах таблицы адресации.
4. Если в процессе сдвига происходит выход за пределы адресного пространства размерности R_A , то $R_0:=R_0+1$.

5. Если $R_0 < R_A$, то перейти к п. 2.

6. Конец.

Рассмотрим применение этой процедуры к КМУУ $U_1(\Gamma_1)$. Положим $R_0=2$ и построим начальную таблицу адресации, используя стандартные адреса из табл.1 (рис.3).

		T_1T_2			
		00	01	10	11
T_3T_4	00	b_1	b_5	b_9	*
	01	b_2	b_6	b_{10}	*
	10	b_3	b_7	b_{11}	*
	11	b_4	b_8	*	*

Рис. 3. Стандартные адреса микрокоманд КМУУ $U_2(\Gamma_1)$

Выходы ОЛЦ α_1 и α_2 находятся в разных столбцах таблицы адресации, а выходы ОЛЦ α_2 и α_3 -- в одном. Сдвинем информацию на один разряд вниз, начиная с вершины b_7 (рис.4).

		T_1T_2			
		00	01	10	11
T_3T_4	00	b_1	b_5	b_8	*
	01	b_2	b_6	b_9	*
	10	b_3	*	b_{10}	*
	11	b_4	b_7	b_{11}	*

Рис. 4. Уточненные адреса микрокоманд КМУУ $U_2(\Gamma_1)$

Как видно из рис. 4, выходы ОЛЦ $\alpha_g \in C_1 = \{\alpha_1, \alpha_2, \alpha_3\}$ находятся в разных столбцах таблицы кодирования. Следовательно, на рис.4 приведены уточненные адреса микрокоманд. Очевидно, что код выхода O_g совпадает с информацией, идентифицирующий соответствующий столбец. Итак, для КМУУ $U_2(\Gamma_1)$ имеем: $K(O_1)=00$, $K(O_2)=01$ и $K(O_3)=10$. Следовательно, в рассматриваемом примере выходы ОЛЦ $\alpha_g \in C_1$ однозначно идентифицируются при помощи $R_G=2$ разрядов.

Для синтеза схемы КМУУ $U_2(\Gamma_1)$ необходимо выполнить следующие этапы [7]:

1. Формирование системы формул перехода КМУУ.
2. Формирование таблицы переходов КМУУ.
3. Формирование функций возбуждения счетчика СТ.
4. Формирование содержимого блока БМО.
5. Реализация схемы КМУУ в заданном элементном базисе.

Система формул перехода [1] формируется только для выходов ОЛЦ $\alpha_g \in C_1$. В нашем примере эта система имеет следующий вид:

$$\begin{aligned}
 O_1 &\rightarrow x_1 b_7 \vee \overline{x_1} b_4; \\
 O_2 &\rightarrow x_2 x_3 b_2 \vee \overline{x_2} \overline{x_3} b_9 \vee \overline{x_2} b_4; \\
 O_3 &\rightarrow x_2 x_3 b_2 \vee \overline{x_2} \overline{x_3} b_9 \vee \overline{x_2} b_4.
 \end{aligned}
 \tag{7}$$

Таблица переходов КМУУ строится по системе формул перехода и включает столбцы: O_g , $K(O_g)$, b_q , $A(b_q)$, X_h , Φ_h , h . Здесь X_h – входной сигнал, определяющий переход из O_g в b_q и равный конъюнкции некоторых элементов множества X (или их отрицаний); $\Phi_h \subseteq \Phi$ – набор функций возбуждения памяти, принимающих единичное значение для формирования в СТ адреса $A(b_q)$. Для нашего примера таблица переходов имеет $N_2(\Gamma_1)=8$ строк (табл.2), где $N_i(\Gamma_j)$ означает число строк таблицы переходов КМУУ $N_i(\Gamma_j)$.

Таблица 2

Таблица переходов КМУУ $U_2(\Gamma_1)$

O_g	$K(O_g)$	b_q	$A(b_q)$	X_h	Φ_h	h
O_1	00	b_7	0111	$\overline{x_1}$	$D_2 D_3 D_4$	1
		b_4	0011	x_1	$D_3 D_4$	2
O_2	01	b_2	0001	$x_2 \overline{x_3}$	D_4	3
		b_9	1001	$x_2 x_3$	$D_1 D_4$	4
		b_4	0011	$\overline{x_2}$	$D_3 D_4$	5
O_3	10	b_2	0001	$x_2 \overline{x_3}$	D_4	6
		b_9	1001	$x_2 x_3$	$D_1 D_4$	7
		b_4	0011	$\overline{x_2}$	$D_3 D_4$	8

Адреса микрокоманд $A(b_q)$ берутся из рис.4. Связь табл.2 с системой (7) очевидна.

Таблица переходов КМУУ является основой для формирования системы (2), зависящей от термов F_h , где

$$F_h = \left(\bigwedge_{r=1}^{R_0} T_r^{l_r h} \right) \cdot X_h. \quad (8)$$

В формуле (8) первая конъюнкция соответствует коду $K(O_g)$ из h -й строки таблицы ($h=1, \dots, H_2(\Gamma_j)$); $l_r h \in \{0,1\}$ -- значение r -го разряда кода $K(O_g)$ из h -й строки таблицы, $T_r^0 = \overline{T_r}$, $T_r^1 = T_r$ ($r=1, \dots, R_0$). Например, из табл.2 имеем $D_1 = F_4 \vee F_7 = \overline{T_1} T_2 x_2 \overline{x_3} \vee T_1 \overline{T_2} x_2 \overline{x_3}$.

Отметим, что в формуле $D_r \in \Phi$ для КМУУ $U_1(\Gamma_1)$ входят все переменные $T_r \in T$, что повышает вероятность необходимости функциональной декомпозиции для системы (2).

Блок БМО реализуется на ВБП и задается таблицей истинности с входами $T_r \in T$ и выходами $Y \cup \{y_0, y_E\}$. Содержимое этой таблицы строится тривиальным образом:

1. Набор микроопераций $Y(b_q)$ записывается в строку таблицы, соответствующую адресу $A(b_q)$.
2. Если $b_q \neq O_g$, то по адресу $A(b_q)$ записывается переменная y_0 .
3. Если $\langle b_t, b_E \rangle \in E$, то по адресу $A(b_q)$ записывается переменная y_E .

Применение этой процедуры к ГСА Γ_1 приводит к содержимому блока БМО, символическое представление которого показано в табл.3.

Таблица 3

Содержимое блока БМО КМУУ $U_2(\Gamma_1)$

Адрес	Слово	Адрес	Слово	Адрес	Слово	Адрес	Слово
0000	$y_0 y_1 y_2$	0100	$y_0 y_5$	1000	y_3	1100	-
0001	$y_0 y_3$	0101	$y_3 y_4$	1001	$y_0 y_1 y_2$	1101	-
0010	$y_3 y_4$	0110	-	1010	$y_0 y_2 y_5$	1110	-
0011	$y_0 y_1 y_2$	0111	$y_0 y_2 y_5$	1011	$y_4 y_E$	1111	-

Адреса микрокоманд для табл.3 взяты из рис.4, а содержимое слов памяти -- из вершин ГСА Γ_1 . В табл.3 знак “-“ соответствует неопределенному содержанию слова, когда адрес не используется для микрокоманд КМУУ $U_2(\Gamma_1)$.

Реализация схемы КМУУ $U_2(\Gamma_j)$ сводится к реализации системы (2) на ЛЭ табличного типа, а системы (4) -- на встроенных блоках памяти. Для этой цели используются стандартные промышленные пакеты [9,10]. В настоящей статье этот этап не рассматривается.

5. Синтез КМУУ с оптимальной адресацией микрокоманд

Анализ табл.2 показывает, что переходы из выходов O_2 и O_3 идентичны для одинаковых входных наборов X_n . Это связано с тем, что вершины $b_6=O_2$ и $b_8=O_3$ связаны с входом условной вершины b_{13} ГСА Γ_1 (рис.2). Таким образом, ОЛЦ $\alpha_2, \alpha_3 \in C_1$ являются псевдоэквивалентными. Если выходы O_2 и O_3 входят в один обобщенный интервал четырехмерного булева пространства, то число строк таблицы переходов уменьшается по сравнению с $H_2(\Gamma_1)$. В общем случае параметр $H_2(\Gamma)$ определяется следующим образом:

$$H_2(\Gamma) = \sum_{g=1}^G N_g C_g, \quad (9)$$

где C_g -- булева переменная, равная единице, если $\alpha_g \in C_1$, N_g -- число переходов из выхода ОЛЦ $\alpha_g \in C_1$.

Если каждый класс $V_i \in P_C$ представляется одним обобщенным интервалом R_A -размерного булева пространства, то таблица переходов КМУУ включает $H_3(\Gamma)$ строк, где

$$H_3(\Gamma) = \sum_{i=1}^I N_{gi}. \quad (10)$$

В выражении (10) параметр N_{gi} равен числу переходов N_g для любой из ОЛЦ $\alpha_g \in V_i$. При этом разбиение P_C находится для множества C_1 . Как уже отмечалось ранее, оптимальная адресация микрокоманд порождает КМУУ U_3 . Структуры КМУУ U_1 и U_3 совпадают. Метод синтеза КМУУ U_3 включает следующие этапы:

1. Формирование множеств C и C_1 для ГСА Γ .
2. Формирование разбиения P_C множества C_1 на классы псевдоэквивалентных ОЛЦ.
3. Оптимальная адресация микрокоманд.

Далее следует 5 этапов, рассмотренных для КМУУ U_2 . Наиболее трудным из этих этапов является адресация микрокоманд. Для ее выполнения можно использовать модифицированные алгоритмы из работ [11,12], либо алгоритм ESPRESSO [6]. Рассмотрим условия, при которых оптимальная адресация микрокоманд с $|V_i|=1$ для $i = \overline{1, I}$ возможна.

Пусть $R_i = \lceil \log_2 |V_i| \rceil$, тогда для размещения выходов ОЛЦ $\alpha_g \in V_i$ требуется куб булева пространства, размерность которого определяется как

$$V_i = 2^{R_i} \quad (i = \overline{1, I}). \quad (11)$$

Пусть d_i -- максимальное число компонент в ОЛЦ $\alpha_g \in V_i$, $Q_i = \lceil \log_2 d_i \rceil$. В этом случае для размещения компонент ОЛЦ $\alpha_g \in C_1$ требуется куб размерности

$$W_i = 2^{Q_i} \quad (i = \overline{1, I}). \quad (12)$$

Таким образом, для размещения всех компонент ОЛЦ $\alpha_g \in V_i$ требуется куб размером

$$\Delta_i = V_i \times W_i \quad (i = \overline{1, I}). \quad (13)$$

Для адресации микрокоманд используется карта Карно, в которой для адресов микрокоманд, соответствующих компонентам ОЛЦ $\alpha_g \in C_1$, отводится Δ клеток, где

$$\Delta = 2^{R_A} - M_1. \quad (14)$$

Параметр M_1 определяет число компонент для ОЛЦ $\alpha_g \notin C_1$. Если выполняется условие

$$\Delta \geq \sum_{i=1}^I \Delta_i, \quad (15)$$

то адреса компонент любой ОЛЦ $\alpha_g \in C_1$ могут быть расположены в соседних клетках карты Карно, причем компоненты всех ОЛЦ $\alpha_g \in V_i$ расположены в одном кубе размерности Δ_i .

Для рассматриваемого примера $C_1 = \{\alpha_1, \alpha_2, \alpha_3\}$, $P_C = \{b_1, b_2\}$, $V_1 = \{\alpha_1\}$, $V_2 = \{\alpha_2, \alpha_3\}$ и один из вариантов оптимальной адресации микрокоманд представлен на рис.5.

Модификация карты заключается в том, что по вертикали записываются двоичные наборы, следующие в естественном порядке. Кубы, соответствующие классам $V_1, V_2 \in P_C$ и ОЛЦ α_3 , дают следующие коды, однозначно идентифицирующие классы $V_i \in P_C$: $K(V_1) = 00^{**}$, $K(V_2) = *1^{**}$. Как видно из этих кодов классов используются $R_G = 2$ переменных, как и в случае КМУУ $U_2(\Gamma_1)$.

		$T_1 T_2$			
		00	01	11	10
$T_3 T_4$	00	b_1	*	*	*
	01	b_2	*	b_4	b_9
	10	b_3	b_7	b_5	b_{10}
	11	*	b_8	b_6	b_{11}

Рис. 5. Модифицированная карта Карно для оптимальной адресации микрокоманд КМУУ $U_3(\Gamma_1)$

Система формул перехода для КМУУ $U_3(\Gamma_1)$ представлена системой (7). С учетом оптимальной адресации микрокоманд, выход O_1 представляется классом B_1 , а выходы O_2 и O_3 -- классом B_2 . Построим модифицированную систему формул перехода, заменив выходы ОЛЦ $\alpha_g \in B_i$ классами $V_i \in \text{Пс}$:

$$\begin{aligned} B_1 &\rightarrow x_1 b_7 \vee \overline{x_1} b_4; \\ B_2 &\rightarrow x_2 x_3 b_2 \vee x_2 \overline{x_3} b_9 \vee \overline{x_2} b_4. \end{aligned} \quad (16)$$

Таблица переходов КМУУ U_3 строится по модифицированной системе формул перехода и включает столбцы: $B_i, K(B_i), b_q, A(b_q), X_h, \Phi_h, h$. В случае КМУУ $U_3(\Gamma_1)$ адреса микрокоманд берутся из рис.5, а таблица переходов имеет $N_3(\Gamma_1)=5$ строк (табл.3).

Таблица 3

Таблица переходов КМУУ $U_3(\Gamma_1)$

B_i	$K(B_i)$	b_q	$A(b_q)$	X_h	Φ_h	h
B_1	00**	b_7	0110	x_1	$D_2 D_3$	1
		b_4	1101	$\overline{x_1}$	$D_1 D_2 D_4$	2
B_2	*1**	b_2	0001	$x_2 \overline{x_3}$	D_4	3
		b_9	1001	$x_2 x_3$	$D_1 D_4$	4
		b_4	1101	$\overline{x_2}$	$D_1 D_2 D_4$	5

Эта таблица является основой для формирования системы (2), термы которой также определяются выражением (8). Однако, в данном случае первая конъюнкция соответствует коду $K(B_i)$ из h -й строки таблицы, а $l h \in \{0, 1, *\}$ ($h=1, \dots, N_3(\Gamma)$). Литералы этой конъюнкции определяются следующим образом: строки таблицы, $T_r^0 = \overline{T_r}$, $T_r^1 = T_r$, $T_r^* = 1$ ($r=1, \dots, R_A$). Например, из табл.3 имеем:

$$D_1 = F_2 \vee F_4 \vee F_5 = \overline{T_1} \overline{T_2} \overline{x_1} \vee T_2 x_2 \overline{x_3} \vee T_2 \overline{x_2} = \overline{T_1} \overline{T_2} \overline{x_1} \vee T_2 x_2 \vee T_2 \overline{x_3}.$$

Содержимое блока БМО для КМУУ U_3 производится аналогично выполнению этого этапа для КМУУ U_2 . Реализация схемы КМУУ U_3 сводится к реализации схемы блока БАМ на ЛЭ табличного типа, а схемы БМО -- на ВБП. Оба эти этапа в нашей статье не рассматриваются.

6. Заключение

Предложенные методы адресации микрокоманд позволяют уменьшить число аргументов в функциях возбуждения счетчика в пределе до R_G . При этом упрощаются уравнения для выходных функций блока адресации микрокоманд, что приводит к уменьшению аппаратных затрат по сравнению с исходной моделью КМУУ U_1 .

Теоретически все модели U_1 -- U_3 имеют одинаковую длительность такта, но на практике упрощение реализуемых уравнений часто приводит к уменьшению числа уровней в комбинационной схеме. Таким образом, применение предложенных методов адресации может привести к уменьшению числа ЛЭ табличного типа, так и времени такта устройства управления. При этом оптимальная адресация микрокоманд более эффективна, чем уточненная, если выполняется условие (15).

Для исследования эффективности разработанных методов были построены VHDL-модели КМУУ U_1 -- U_3 . Полученные системы уравнений вводились в пакет WebPack фирмы Xilinx [10] для реализации схемы КМУУ. Стандартные тестовые примеры для конечных автоматов [13] не содержат линейных ГСА. Поэтому для их генерации была разработана специальная программа. В результате исследований оказалось, что схемы КМУУ U_2 -- U_3 всегда имели меньше аппаратных затрат (в среднем на 22%), чем схемы эквивалентных КМУУ U_1 . Если условие (15) выполняется, то КМУУ U_3 обладает наилучшими характеристиками по быстродействию и аппаратным затратам по сравнению с КМУУ U_1 и U_2 . Отметим также, что метод оптимальной адресации может быть использован и при реализации схем КМУУ на ПЛИС, основанных на макроячейках программируемой матричной логики (ПМА) [14]. Это связано с тем, что при оптимальной адресации минимизируется число термов в функциях системы (2), а макроячейки ПМЛ имеют ограниченное число термов. Также отметим, что оба метода адресации не требуют увеличения емкости памяти блока БМО по сравнению с базовой схемой КМУУ U_1 .

В печать

На перевод на английский язык согласен

Список литературы

1. Baranov S. Logic and System Design of Digital Systems. -- Tallinn: TUT Press, 2008. -- 266 pp.
2. Глушков В.М. Синтез цифровых автоматов. -- М.: Физматгиз, 1962. --476 с.
3. Minus P., Eliot I. FSM – based Digital Design using Verilog. -- John Wiley & Sons, 2008. -- 351 pp.
4. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах программируемой логики. -- Петербург: БХВ – Петербург, 2002. – 636 с.
5. Maxfield S. The Design Warrior’s Guide to FPGAs. -- Elsevier: Amsterdam, 2004. – 541 pp.
6. De Micheli G. Synthesis and Optimization of Digital Circuits. -- NY: McGraw Hill, 1994. -- 626 pp.
7. Баркалов А.А., Титаренко Л.А. Синтез композиционных микропрограммных устройств управления. --Харьков:Коллегиум,2007.--302с.
8. Баркалов А.А. Микропрограммное устройство управления как композиция автоматов с программируемой и жесткой логикой // Автоматика и вычислительная техника. – 1983. -- №4. -- С. 36-41.
9. www.xilinx.com.
10. www.altera.com.
11. Ачасова С.Н. Алгоритмы синтеза автоматов на программируемых матрицах. -- М.: Радио и связь, 1987. -- 136 с.
12. Закревский А.Д., Поттосин Ю.В., Черемисинова Л.Д. Оптимизация в булевом пространстве. -- Минск: Объед. ин-т проблем информатики, 2004. -- 240 с.
13. Yang S. Logic Synthesis and Optimization Benchmarks User Guide. Techical Report №1991 -- IWLS -- UG -- Sqeyang. -- Microelectronics Center of North Carifornia -- 1991. -- 43 pp.
14. Соловьев В.В., Климович А.С. Логическое проектирование цифровых систем на основе программируемых логических интегральных схем. -- М.: Горячая линия -- Телеком, 2008. – 376 с.