

УМЕНЬШЕНИЕ АППАРАТУРНЫХ ЗАТРАТ В УСТРОЙСТВЕ УПРАВЛЕНИЯ С РАЗДЕЛЕНИЕМ КОДОВ

Ключевые слова: *граф-схема алгоритма, управляющее устройство, композиционное микропрограммное устройство управления, разделение кодов, программируемые логические интегральные схемы типа CPLD, адресация микрокоманд.*

ВВЕДЕНИЕ

Практически любая цифровая система включает устройство управления (УУ), координирующее взаимодействие системных блоков [1]. Функционирование УУ часто может определяться моделью цифрового автомата [2]. Как правило, при реализации схемы УУ возникает задача уменьшения аппаратных затрат [3]. Методы решения этой задачи зависят от особенностей алгоритма управления и элементного базиса. Так, при линейном характере алгоритма управления целесообразно использовать модель композиционного микропрограммного устройства управления (КМУУ) [4, 5]. В настоящей работе рассмотрена задача реализации схемы КМУУ в базисе программируемых логических интегральных схем (ПЛИС) типа CPLD (Complex Programmable Logic Devices) [6, 7]. Подобные ПЛИС состоят из макроячеек типа ПМЛ (программируемая матричная логика) и матрицы межсоединений. Особенностью ПМЛ является достаточно большое количество входов (несколько десятков) и ограниченное количество термов [8, 9]. Для уменьшения количества макроячеек в схеме УУ необходимо уменьшить количество термов в реализуемых функциях. Для этого можно использовать несколько источников кодов состояний [10].

Для КМУУ в качестве состояний использованы операторные линейные цепи (ОЛЦ). В настоящей статье предложены методы, основанные на применении нескольких источников кодов ОЛЦ. При этом алгоритм управления цифровой системы представлен в виде линейной граф-схемы алгоритма (ГСА) [1]. Существует много различных моделей КМУУ [4], их выбор определяется особенностями ГСА, по которой реализуется схема УУ. Далее рассмотрена модель КМУУ с разделением кодов [4]. Отметим, что предлагаемые методы можно использовать для любой модели КМУУ.

БАЗОВАЯ МОДЕЛЬ КМУУ С РАЗДЕЛЕНИЕМ КОДОВ

Пусть ГСА Γ представлена множествами вершин B и дуг E , соединяющих эти вершины. При этом $B = \{b_0, b_E\} \cup B_1 \cup B_2$, где b_0, b_E — начальная и конечная вершины ГСА соответственно; B_1 — множество операторных вершин, $|B_1| = M$; B_2 — множество условных вершин. В вершинах $b_q \in B_1$ записаны наборы микроопераций $Y(b_q) \subseteq Y$, где $Y = \{y_1, \dots, y_N\}$ — множество микроопераций. В вершинах $b_q \in B_2$ записаны элементы множества логических условий $X = \{x_1, \dots, x_L\}$. Введем определения [4, 5], необходимые для дальнейшего изложения.

Определение 1. Операторной линейной цепью ГСА Γ называется конечная последовательность операторных вершин $\alpha_g = \langle b_{g1}, \dots, b_{gF_g} \rangle$ такая, что для любой

пары соседних компонентов кортежа α_g существует дуга $\langle b_{gi}, b_{gi+1} \rangle \in E$, где i — номер компонента кортежа α_g ($i=1, \dots, F_g - 1$), G — число ОЛЦ в ГСА Γ .

Определение 2. Операторная вершина $b_q \in D^g$, где $D^g \subseteq B_1$ — множество операторных 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 $b_t \notin D^g$.

Определение 4. Операторные линейные цепи α_i, α_j называются псевдоэквивалентными ОЛЦ (ПОЛЦ), если существуют дуги $\langle b_i, b_t \rangle, \langle b_j, b_t \rangle \in E$, где b_i, b_j — соответственно выходы ОЛЦ α_i, α_j .

Каждая ОЛЦ α_g имеет произвольное число входов, образующих множество $I(\alpha_g) = \{I_g^1, I_g^2, \dots\}$, и только один выход O_g . Пусть для ГСА Γ найдено разбиение $C = \{\alpha_1, \dots, \alpha_G\}$ множества B_1 на ОЛЦ и множество C включает минимально возможное число ОЛЦ, для чего используется методика [4].

Определение 5. Граф-схема алгоритма Γ называется линейной, если выполняется условие

$$\frac{M}{G} \geq 2. \quad (1)$$

Таким образом, ГСА является линейной, если количество ее операторных вершин хотя бы в два раза превышает минимальное число ее ОЛЦ. При выполнении условия (1) для реализации УУ целесообразно использовать модель КМУУ [4].

Отметим, что каждая вершина $b_q \in B_1$ соответствует микрокоманде MI_q , хранимой в управляющей памяти (УП) по адресу $A(b_q)$. Для адресации микрокоманд достаточно $R = \lceil \log_2 M \rceil$ бит. Пусть каждая ОЛЦ $\alpha_g \in C$ включает F_g компонентов и $Q = \max(F_1, \dots, F_G)$. Закодируем каждую ОЛЦ $\alpha_g \in C$ двоичным кодом $K(\alpha_g)$ разрядности $R_1 = \lceil \log_2 G \rceil$, а каждый компонент $b_q \in B_1$ — двоичным кодом $K(b_q)$ разрядности $R_2 = \lceil \log_2 Q \rceil$.

Используем для кодирования ОЛЦ переменные $\tau_r \in \tau$, а для кодирования компонентов — переменные $T_r \in T$, где $|\tau| = R_1, |T| = R_2$. Кодирование компонентов выполняется в естественном порядке, т.е.

$$K(b_{gi+1}) = K(b_{gi}) + 1, \quad (2)$$

где $g=1, \dots, G, i=1, \dots, F_g$. Пусть для ГСА Γ выполняется условие

$$R_1 + R_2 = R. \quad (3)$$

В этом случае для ее интерпретации можно использовать модель КМУУ с разделением кодов (рис. 1), обозначаемую в дальнейшем U_1 .

В КМУУ U_1 схема адресации микрокоманд (САМ) реализует систему функций возбуждения триггеров счетчика СТ и регистра Рг

$$\begin{aligned} \Phi &= \Phi(\tau, X), \\ \Psi &= \Psi(\tau, X). \end{aligned}$$

При этом адрес микрокоманды MI_q представляется в виде $A(b_q) = K(\alpha_g) * K(b_q)$, где вершина b_q входит в состав ОЛЦ $\alpha_g \in C$, * — знак операции конкатенации.

По сигналу Start в Рг и СТ заносится начальный адрес микропрограммы, а триггер выборки (ТВ) устанавливается в состояние логической единицы. При этом Fetch = 1, что разрешает выборку команд из УП. Если считанная микрокоманда

манда не соответствует выходу ОЛЦ, то одновременно с микрооперациями $Y(b_q)$ формируется переменная y_0 . Если $y_0 = 1$, то к содержимому СТ прибавляется единица и адресуется следующий компонент текущей ОЛЦ. Если выход ОЛЦ достигнут, то $y_0 = 0$. При этом САМ формирует адрес входа следующей ОЛЦ. По достижению окончания микропрограммы формируется сигнал y_E , триггер ТВ обнуляется, и выборка микрокоманд прекращается.

Выходные переменные y_0 , y_E и $y_n \in Y$ зависят только от содержимого регистра и счетчика. Таким образом, КМУУ U_1 является автоматом Мура [1], при этом ОЛЦ $\alpha_g \in C$ — аналог состояния автомата. Следовательно, для оптимизации схемы КМУУ можно использовать все известные методы оптимизации автомата Мура [7, 10, 11]. Естественно, они должны быть адаптированы к особенностям КМУУ.

Особенность КМУУ U_1 и заключается в отсутствии зависимости между кодами цепей и их компонентами. Кроме того, ГСА Г может включать нетривиальные классы ПОЛЦ. При синтезе схемы КМУУ U_1 блоки САМ, СТ, Рг и ТВ реализуются на макроячейках ПМЛ. Для формирования схемы УП необходимы внешние блоки памяти (ВБП). Особенности макроячеек ПМЛ уже упоминались ранее (большое количество входов и ограниченное количество термов). Особенности ВБП является дискретность числа их выходов. Как правило, выходы принадлежат множеству $Q(t) = \{1, 2, 4, 8, 16, 32\}$. С учетом описанных особенностей можно оптимизировать число макроячеек в схеме адресации.

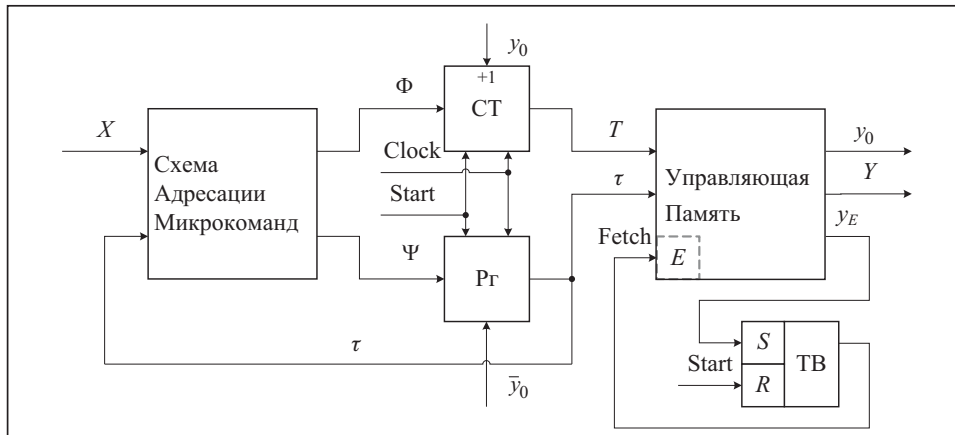


Рис. 1. Структурная схема КМУУ U_1

ОСНОВНАЯ ИДЕЯ ПРЕДЛАГАЕМОГО МЕТОДА

Пусть ОЛЦ $\alpha_g \in C_1$, если выход O_g не связан с конечной вершиной ГСА Г. Найдем разбиение P_C множества C_1 на классы ПОЛЦ. Выполним кодирование ОЛЦ $\alpha_g \in C$ так, чтобы максимально возможное число классов $B_i \in P_C$, где $|P_C| = I$, представлялось одним обобщенным интервалом R_1 -мерного булева пространства. Пусть n_i — число обобщенных интервалов, представляющих класс $B_i \in P_C$. Представим множество P_C в виде $P_C = P_A \cup P_B$. При этом множества P_A и P_B строятся следующим образом:

$$(n_i = 1) \rightarrow B_i \in P_A;$$

$$(n_i > 1) \rightarrow B_i \in P_B.$$

Источником кодов классов $B_i \in P_A$ является регистр Рг. При этом код класса $B_i \in P_A$ определяется соответствующим интервалом R_1 -мерного булева пространства.

Закодируем классы $B_i \in \Pi_B$ двоичными кодами $C(B_i)$ разрядности

$$R_3 = \lceil \log_2(|\Pi_B| + 1) \rceil, \quad (4)$$

используя переменные из множества $Z = \{z_1, \dots, z_{R_3}\}$. Для формирования кодов $C(B_i)$ необходим блок преобразователя кодов (БПК), который реализует систему функций

$$Z = Z(\tau). \quad (5)$$

Пусть для реализации памяти используются блоки ППЗУ, имеющие S входов и t выходов. Примем для упрощения, что $R = S$. Тогда для реализации памяти необходимо

$$m_1 = \left\lceil \frac{N+2}{t} \right\rceil \quad (6)$$

микросхем ППЗУ. Числитель формулы (6) определяется числом микроопераций $y_n \in Y$ и двумя дополнительными переменными: y_0 и y_E . Очевидно, что часть выходов блока УП могут быть избыточными и определяются формулой

$$R_4 = m_1 t - (N + 2). \quad (7)$$

Свободные выходы можно использовать для реализации части кода $C(B_i)$. При этом БПК должен генерировать только

$$R_5 = R_3 - R_4 \quad (8)$$

переменных. Отметим, что в выражении (5) единица добавляется, если $B_i \notin \Pi_B$.

Итак, в общем случае для представления кодов классов ПОЛЦ можно использовать три источника: регистр, преобразователь кодов и управляющую память. Этот подход порождает КМУУ U_2 , приведенное на рис. 2, где множество $Z^1 \subseteq Z$ представляет разряды кода $C(B_i)$, генерируемые преобразователем кодов. Аналогично множество $Z^2 \subseteq Z$ представляет разряды кода $C(B_i)$, генерируемые памятью. Очевидно, что выполняются условия $Z^1 \cap Z^2 = \emptyset$ и $Z^1 \cup Z^2 = Z$.

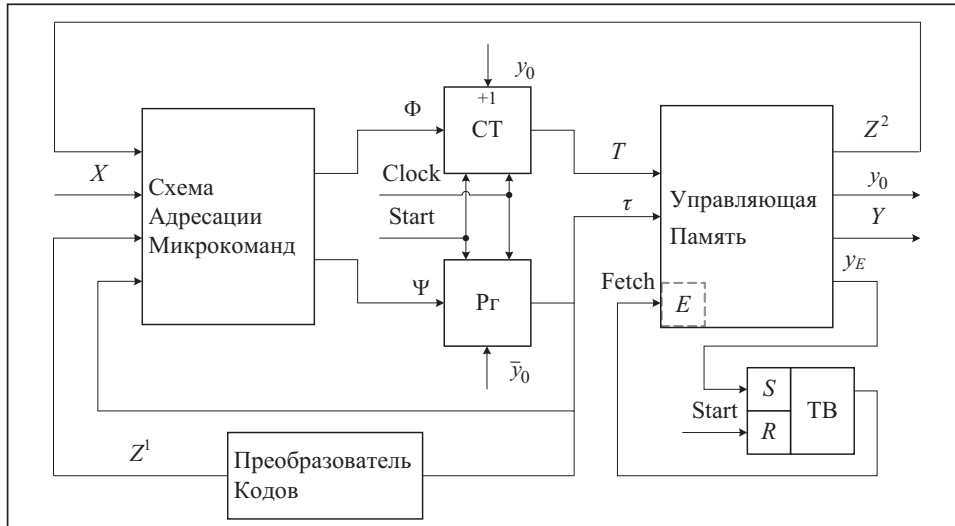


Рис. 2. Структурная схема КМУУ U_2

Принципы функционирования КМУУ U_1 и U_2 совпадают. Отметим, что в зависимости от различных условий модель U_2 можно преобразовать в одну из шести моделей, приведенных в табл. 1. Здесь в столбцах Рг, БПК и УП записаны источники кодов классов псевдоэквивалентных цепей. Отметим, что модели U_1, U_3 и U_4 известны [4], а модели U_2, U_5-U_7 оригинальны. В настоящей работе предложен метод синтеза КМУУ U_2 . Методы синтеза для моделей U_5-U_7 порождаются из предлагаемого метода удалением некоторых этапов.

Таблица 1. Модели КМУУ, порождаемые КМУУ U_2

Модель	Источники кодов классов псевдоэквивалентных цепей			Условия применения модели
	Рг	БПК	УП	
U_1	1	0	0	$\Pi_A = \Pi_C; \Pi_B = \emptyset.$
U_2	1	1	1	$\Pi_A \neq \emptyset; \Pi_B \neq \emptyset; 0 < R_4 < R_3.$
U_3	0	1	0	$\Pi_A = \emptyset; \Pi_B = \Pi_C; R_4 = 0.$
U_4	0	0	1	$\Pi_A = \emptyset; \Pi_B = \Pi_C; R_4 \geq R_3.$
U_5	0	1	1	$\Pi_A = \emptyset; \Pi_B = \Pi_C; 0 < R_4 < R_3.$
U_6	1	0	1	$\Pi_A \neq \emptyset; \Pi_B \neq \emptyset; R_4 \geq R_3.$
U_7	1	1	0	$\Pi_A \neq \emptyset; \Pi_B \neq \emptyset; R_4 = 0.$

Предлагаемый метод включает следующие этапы:

- 1) формирование для ГСА Γ множеств C, C_1 и Π_C ;
- 2) оптимальное кодирование ОЛЦ $\alpha_g \in C_1$ и их компонентов;
- 3) формирование множеств Π_A и Π_B ;
- 4) кодирование классов $B_i \in \Pi_B$ кодами $C(B_i)$;
- 5) определение параметров R_4, R_5 и их соотношений;
- 6) формирование содержимого УП;
- 7) формирование таблицы переходов КМУУ;
- 8) формирование таблицы истинности БПК;
- 9) синтез схемы КМУУ в заданном базисе.

Напомним, что применение модели U_1 имеет смысл только при выполнении условий (1) и (3). Если условие (1) не выполняется, то ГСА Γ не является линейной и необходимо использовать модель автомата Мура [11]. Если для линейной ГСА условие (3) не выполняется, то это приводит к резкому росту числа ВВП в блоке УП по сравнению с известными моделями КМУУ [4]. Рассмотрим некоторые этапы предлагаемого метода.

Этап 1 выполняется по известной методике [4], при этом число G цепей $\alpha_g \in C$ является минимально возможным. Разбиение Π_C формируется на основе определения 4 тривиальным образом.

Оптимальное кодирование ОЛЦ $\alpha_g \in C_1$ выполняется по методике, аналогичной оптимальному кодированию состояний автомата Мура [11]. Для решения этой задачи можно использовать известные методы символьной оптимизации [3]. Компоненты всех цепей $\alpha_g \in C$ кодируются тривиальным образом. Десятичный эквивалент кода первого компонента равен нулю, второго — единице, третьего — двум и т.д.. Такое кодирование удовлетворяет условию (2).

Множества Π_A и Π_B формируются тривиальным образом на основе результата оптимального кодирования ОЛЦ $\alpha_g \in C_1$. Классы $B_i \in \Pi_B$ целесообразно кодировать так, чтобы уменьшить количество термов в функциях $z_r \in Z$. Для этого можно использовать простое правило: чем больше обобщенных интервалов необходимо для представления класса $B_i \in \Pi_B$, тем меньше единиц должен содержать код $C(B_i)$. Это правило является адаптацией известного метода кодирования состояний, ориентированного на D -триггеры [1].

Этап 5 сводится к выполнению простейших арифметических операций. Содержимое УП представляется таблицей со столбцами $A(b_q)$, $Y(b_q)$, y_0 , y_E , q . Перед построением таблицы происходит преобразование исходной ГСА [4]. Если вершина $b_q \in D^g$ и $b_q \neq O_g$, то в нее вводится переменная y_0 . При этом в столбце y_0 строки q , $q = \overline{1, G}$, таблицы записывается единица. Если $\langle b_q, b_E \rangle \in E$, то в вершину $b_q \in B_1$ вводится переменная y_E . При этом в столбце y_E строки q , $q = \overline{1, G}$, таблицы также записывается единица.

Таблица переходов КМУУ строится на основе системы обобщенных формул перехода [4], каждый элемент которой имеет вид

$$B_i \rightarrow \bigvee_{h=1}^{H_i} X_h b_h \quad (i = \overline{1, I}), \quad (9)$$

где X_h — конъюнкция логических условий, определяющая переход из выхода ОЛЦ $\alpha_g \in B_i$ в вершину b_h , H_i — число переходов из выхода O_g . Системе (9) соответствует таблица переходов КМУУ, имеющая столбцы: B_i , $K(B_i)$, $C(B_i)$, b_h , $A(b_h)$, X_h , Φ_h , Ψ_h , h . В этой таблице столбец Φ_h (Ψ_h) содержит функции возбуждения триггеров счетчика (регистра), соответствующие единицам в коде компонента (коде ОЛЦ) b_h ($\alpha_g \in B_i$, где $b_h \in D^g$). Число строк H в таблице определяется суммой:

$$H = \sum_{i=1}^I H_i. \quad (10)$$

Таблица истинности БПК имеет столбцы: $V(B_i^j)$, B_i , $C(B_i)$, Z_k^1 , k . Здесь $V(B_i^j)$ соответствует j -му обобщенному интервалу, в который входит класс $B_i \in \Pi_B$, $j \leq |B_i|$. Множество Z_k^1 содержит переменные $z_r \in Z$, равные единице в коде $C(B_i)$ для k -й строки таблицы ($k = \overline{1, K}$). При разбиении множества Z целесообразно проранжировать функции (5) по возрастанию числа термов. Затем первые R_5 функций определяют переменные $z_r \in Z^1$. Оставшиеся переменные $z_r \in (Z \setminus Z^1)$ должны формироваться блоком памяти. Для этого в таблицу УП вводится столбец $Z(b_q)$. Пусть $b_q = O_g$, где $\alpha_g \in B_i$ и $B_i \in \Pi_B$. Тогда в ячейку памяти с адресом $A(b_q)$ записываются значения переменных $z_r \in Z^2$, которые определяются на основе анализа кода $C(B_i)$.

Для реализации схемы КМУУ необходимо из таблицы его переходов получить систему функций

$$\Psi = \Psi(\tau, Z, X), \quad (11)$$

$$\Phi = \Phi(\tau, Z, X),$$

а из таблицы истинности БПК получить функции

$$Z^1 = Z^1(\tau). \quad (12)$$

Дальнейшая реализация сводится к использованию стандартных пакетов [8, 9] и выходит за рамки настоящей статьи.

ПРИМЕР ПРИМЕНЕНИЯ ПРЕДЛОЖЕННОГО МЕТОДА

Пусть для некоторой ГСА Γ_1 получено множество ОЛЦ $C = \{\alpha_1, \dots, \alpha_{16}\}$, $\alpha_{16} \notin C_1$, и данная граф-схема включает $M = 48$ операторных вершин, при

этом $L=5, N=13$. Пусть ОЛЦ $\alpha_g \in C_1$ образуют разбиение $\Pi_C = \{B_1, \dots, B_6\}$, где $B_1 = \{\alpha_1\}$, $B_2 = \{\alpha_2, \alpha_3, \alpha_4\}$, $B_3 = \{\alpha_5, \alpha_6, \alpha_7\}$, $B_4 = \{\alpha_8, \dots, \alpha_{11}\}$, $B_5 = \{\alpha_{12}, \alpha_{13}, \alpha_{14}\}$, $B_6 = \{\alpha_{15}\}$, и максимальное число компонентов $Q=4$. Тогда $R_2=2, T = \{T_1, T_2\}$. Итак, $M=48, G=16$, следовательно $R=6, R_1=4$, условие (3) выполняется и использование модели U_1 имеет смысл.

Закодируем ОЛЦ $\alpha_g \in C$ как показано на рис. 3. В этой карте Карно ОЛЦ $\alpha_{16} \notin C_1$ рассмотрена как несущественная, что позволяет использовать набор 0110 для минимизации числа обобщенных интервалов. Из рис. 3 имеем: $V(B_1) = 0000$, $V(B_2) = 01**$, $V(B_3^1) = 00*1$, $V(B_3^2) = 001*$, $V(B_4) = 10**$, $V(B_5^1) = 110*$, $V(B_5^2) = 11*1$, $V(B_6) = *110$. Таким образом, $\Pi_A = \{B_1, B_2, B_4, B_6\}$ и $\Pi_B = \{B_3, B_5\}$. Поскольку $\Pi_A \neq \emptyset$ и $\Pi_B \neq \emptyset$, из табл. 1 следует, что можно использовать модели U_2, U_6 и U_7 .

Поскольку каждый класс $B_i \in \Pi_B$ представляется двумя обобщенными интервалами, эти классы кодируются тривиальным образом. Из формулы (4) следует, что $R_3=2$, т.е. $Z = \{z_1, z_2\}$. Пусть $C(B_3) = 01$, $C(B_5) = 10$, а код 00 соответствует условию $B_i \in \Pi_A$.

Пусть для реализации УП используются микросхемы ППЗУ, у которых $S=6, t=8$. Поскольку $N=13$, из формулы (7) имеем $R_4=1$. Таким образом, для реализации КМУУ необходимо выбрать модель U_2 . Из (8) имеем $R_5=1$. Пусть $Z^1 = \{z_1\}$ и $Z^2 = \{z_2\}$.

Предположим, что ОЛЦ α_5 имеет четыре компонента: $\alpha_5 = \langle b_{15}, b_{16}, b_{17}, b_{18} \rangle$. Очевидно, $K(b_{15}) = 00, \dots, K(b_{18}) = 11$. Из карты Карно (см. рис. 3) можно получить следующие адреса: $A(b_{15}) = 000100, \dots, A(b_{18}) = 000111$. Так как $\alpha_5 \in B_3$ и $C(B_3) = 01$, то $z_2 = 1$ для ячейки с адресом 000111. Пусть $Y(b_{15}) = \{y_3, y_4\}$, $Y(b_{16}) = \{y_2, y_6\}$, $Y(b_{17}) = \{y_7, y_9\}$ и $Y(b_{18}) = \{y_1, y_8\}$. Тогда содержимое УП для ОЛЦ $\alpha_5 \in C$ будет иметь вид, приведенный в табл. 2.

Аналогично можно получить содержимое УП для остальных ОЛЦ $\alpha_g \in C$. Отметим, что в столбце $Y(b_q)$ записаны микрооперации $y_n \in Y$, которые при реализации схемы УП заменяются своими кодами [4].

Пусть система обобщенных формул перехода для ГСА Γ_1 имеет следующий вид:

$$\begin{aligned}
 B_1 &\rightarrow x_1 b_4 \vee \overline{x_1 x_2} b_8 \vee \overline{x_1 x_2 x_3} b_{10} \vee \overline{x_1 x_2 x_3} b_{12}; \\
 B_2 &\rightarrow x_4 b_{15} \vee \overline{x_4 x_5} b_{19} \vee \overline{x_4 x_5} b_{22}; \\
 B_3 &\rightarrow x_2 x_3 b_{24} \vee \overline{x_2 x_3} b_{28} \vee \overline{x_2 x_5} b_{32} \vee \overline{x_2 x_5} b_{34}; \\
 B_4 &\rightarrow x_1 x_2 b_{17} \vee \overline{x_1 x_2} b_{37} \vee \overline{x_1 x_4} b_{39} \vee \overline{x_1 x_4} b_{41}; \\
 B_5 &\rightarrow b_{44}; \\
 B_6 &\rightarrow x_4 b_{38} \vee \overline{x_4} b_{46}.
 \end{aligned} \tag{13}$$

		$\tau_3\tau_4$			
		00	01	11	10
$\tau_1\tau_2$	00	α_1	α_5	α_6	α_7
	01	α_2	α_3	α_4	α_{16}
	11	α_{12}	α_{13}	α_{14}	α_{15}
	10	α_8	α_9	α_{10}	α_{11}

Рис. 3. Оптимальные коды ОЛЦ для ГСА Γ_1

Таблица 2. Фрагмент содержимого УП для ОЛЦ α_5

$A(b_q)$	$Y(b_q)$	y_0	y_E	Z^2	q
000100	y_3, y_4	1	0	0	15
000101	y_2, y_6	1	0	0	16
000110	y_7, y_9	1	0	0	17
000111	y_1, y_8	0	0	1	18

Таблица 3. Фрагмент таблицы переходов КМУУ U_2

B_i	$K(B_i)$	$C(B_i)$	b_h	$A(b_h)$	X_h	Φ_h	Ψ_h	h
B_3	****	01	b_{24}	100000	x_2x_3	—	D_1	8
			b_{28}	100100	x_2x_3	—	D_1D_4	9
			b_{32}	101100	x_2x_5	—	$D_1D_3D_4$	10
			b_{34}	101000	x_2x_5	—	D_1D_3	11
B_4	10**	00	b_{17}	000110	x_1x_2	D_5	D_4	12
			b_{37}	110000	x_1x_2	—	D_1D_2	13
			b_{39}	110100	x_1x_4	—	$D_1D_2D_4$	14
			b_{41}	111100	x_1x_4	—	$D_1D_2D_3D_4$	15

Рассмотрим фрагмент таблицы переходов КМУУ U_2 (табл. 3) для классов $B_3 \in \Pi_B$ и $B_4 \in \Pi_B$. Пусть $b_{24} = I_8^1$, $b_{28} = I_9^1$, $b_{32} = I_{10}^1$, $b_{34} = I_{11}^1$, $b_{17} = I_5^2$ и $K(b_{17}) = 10$, $b_{37} = I_{12}^1$, $b_{39} = I_{13}^1$ и $b_{41} = I_{14}^1$. Очевидно, для всех вершин, кроме b_{17} , выполняется условие $K(b_q) = 00$ (для рассматриваемых классов $B_3, B_4 \in \Pi_C$). Фрагмент таблицы переходов КМУУ U_2 имеет восемь строк. Так как формулы для B_1 и B_2 из (13) содержат семь термов, то первый переход для класса B_3 имеет восьмой номер.

Очевидна связь табл. 3 с системой (13) и рис. 3, из которого берутся коды ОЛЦ $\alpha_g \in C$. Таблица переходов является основой для формирования систем (10), (11), каждая функция которых представляется в виде

$$D_r = \bigvee_{h=1}^H C_{rh} \left(\bigwedge_{r=1}^R \tau_r^{l_{rh}} \right) \wedge \left(\bigwedge_{r=1}^R z_r^{m_{rh}} \right) \wedge X_h, \quad (14)$$

где булева переменная $C_{rh} = 1$, если и только если в h -й строке таблицы записано D_r ($r = \overline{1, R}$). Второй член формулы (14) соответствует коду $K(B_i)$, где $l_{rh} \in \{0, 1, *\}$ — значение r -го разряда кода ($r = \overline{1, R_1}$). При этом $\tau_r^0 = \overline{\tau_r}$, $\tau_r^1 = \tau_r$, $\tau_r^* = 1$. Третий член формулы (14) соответствует коду $C(B_i)$, где $m_{rh} \in \{0, 1\}$ — значение r -го разряда кода ($r = \overline{1, R_3}$). Количество строк в таблице определяется формулой (10). В рассмотренном примере $H = 18$.

Например, с учетом закона склеивания [3], можно получить следующую формулу:

$$D_1 = \overline{z_1 z_2} \vee \overline{\tau_1 \tau_2} \overline{z_1 z_2 x_1 x_2} \vee \overline{\tau_1 \tau_2} \overline{z_1 z_2 x_1}, \quad (15)$$

где первый член соответствует h -м строкам 8–11 табл. 3, второй — h -й строке 13, третий — h -м строкам 14, 15. Напомним, что для ПЛИС типа CPLD необходимо выполнить раздельную минимизацию реализуемых систем булевых функций [7, 10].

Рассмотрим таблицу истинности БПК (табл. 4), имеющую две строки. Из этой таблицы формируется система (12), каждая функция которой имеет вид

$$z_r = \bigvee_{k=1}^K C_{rk} \left(\bigwedge_{r=1}^R \tau_r^{l_{rk}} \right), \quad (16)$$

где булева переменная $C_{rk} = 1$, если и только если в k -й строке таблицы записана переменная $z_r = Z^1$. Вторые члены формул (14) и (16) аналогичны. На основании табл. 4 можно получить следующую формулу: $z_1 = \tau_1 \tau_2 \overline{\tau_3} \vee \tau_1 \tau_2 \tau_4$.

Как уже отмечалось, этап реализации схемы КМУУ в данной статье не рассмотрен. Однако авторами реализована САПР, позволяющая совместно с пакетом WebPack фирмы Xilinx синтезировать схемы КМУУ U_1-U_7 .

Таблица 4. Таблица истинности БПК

$V(B_i^j)$	B_i	$C(B_i)$	Z_k^1	l_C
110*	B5	10	z_1	1
11*1	B5	10	z_1	2

РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЙ

Для проведения исследований с целью определения области применения той или иной структуры был разработан программный комплекс (рис. 4), состоящий из следующих элементов:

- fsmEditor — программа для создания и редактирования ГСА с помощью графических примитивов (встроенный модуль fsmGen позволяет пакетно генерировать ГСА с заданными параметрами);
- VHDL Generator — программа, позволяющая на основе XML-описания алгоритма и необходимых параметров создать VHDL-модель управляющего автомата, синтезируемого по ГСА;
- tclGen — утилита, необходимая для создания скрипта управления работой системы XST, входящей в Xilinx WebPack, на основе данных, полученных от VHDL Generator (возможна пакетная генерация на основе набора созданных ранее файлов *.vhd);
- Xilinx WebPack — программный пакет, предназначенный для проектирования устройств на микросхемах ПЛИС фирмы Xilinx (входящий в пакет модуль XST на основе входных файлов *.vhd и *.mif согласно скрипту управления, переданному от tclGen, генерирует файл прошивки микросхемы, а также множество файлов отчета, среди которых находятся данные об аппаратурных затратах, требуемых для реализации автомата на данной микросхеме);
- rptAnlZr — утилита, пакетно обрабатывающая наборы файлов *.rpt, полученных от Xilinx WebPack, и формирующая на их основе таблицы данных в формате *.mat;
- MathWorks MatLab — пакет прикладных программ для решения задач технических вычислений, являющийся мощным средством построения графиков на основе данных, полученных от rptAnlZr.

Для эксперимента выбраны ГСА со следующими параметрами:

- количество вершин от 10 до 400 с шагом 10;
- доля операторных вершин от 50 до 90 % с шагом 10 %;
- количество микроопераций $N = 15$;
- количество логических условий $L = 5$.

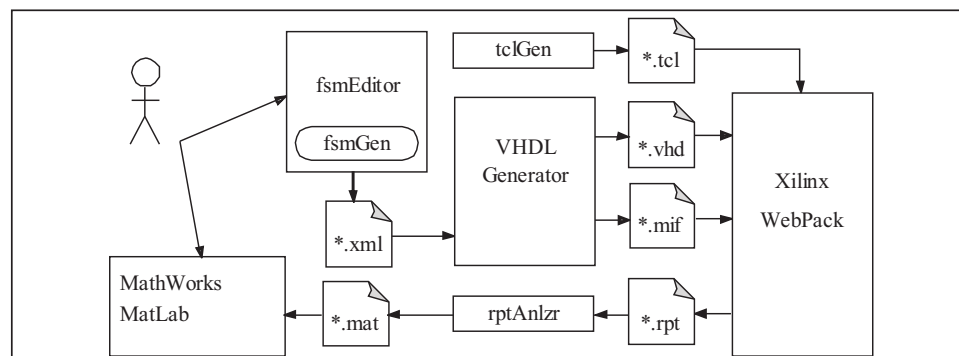


Рис. 4. Программный комплекс для проведения исследований

Для каждой ГСА синтезирована структура КМУУ с разделением кодов U_1 (смси_cs), а также исследуемая структура U_2 (смси_fcs). Для сравнения с классом автоматов с жесткой логикой также синтезированы автоматы Мура (Moore). Каждое измерение, представленное на графиках, является средним значением результатов синтеза для пяти различных ГСА с одинаковыми параметрами.

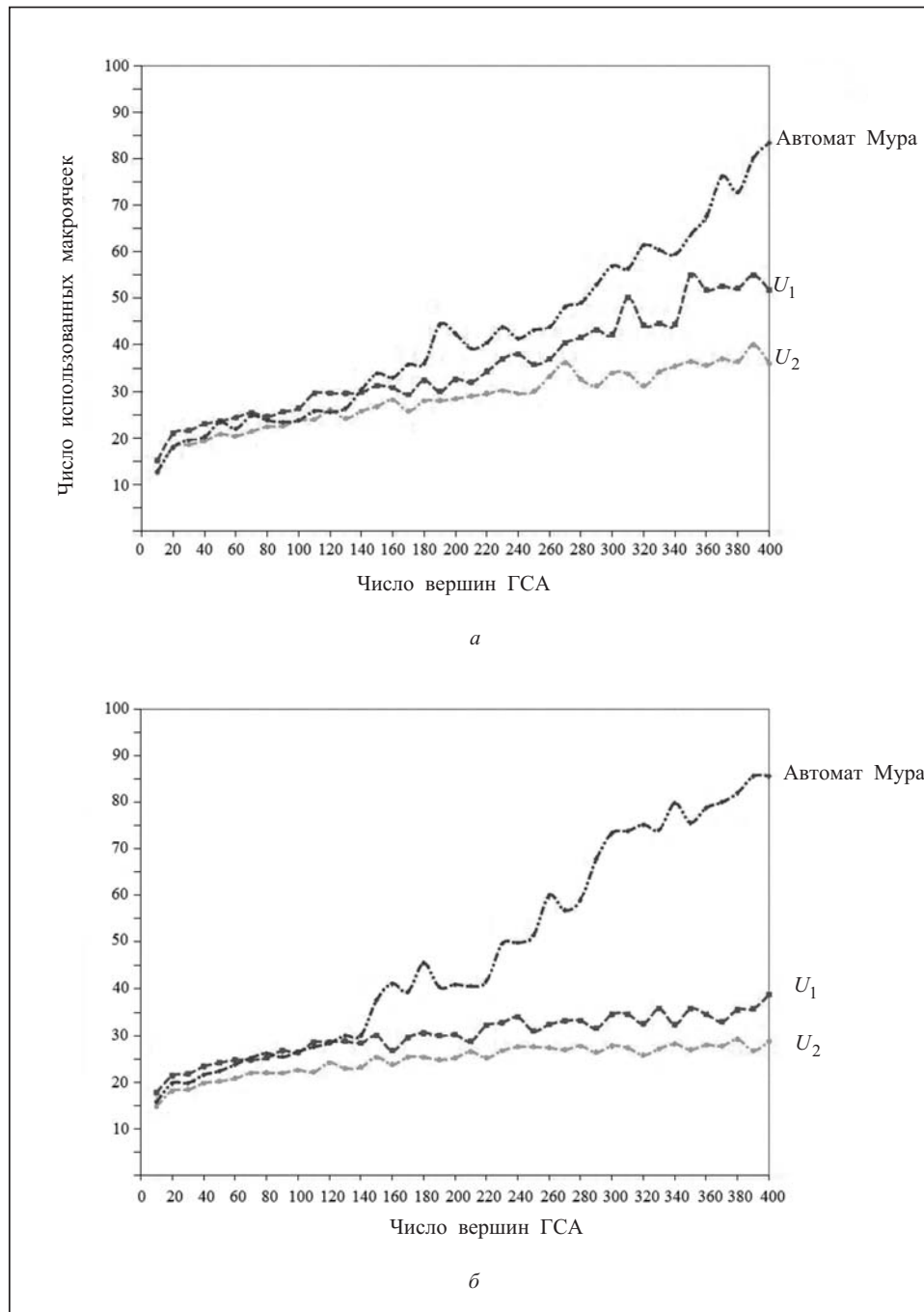


Рис. 5. Аппаратурные затраты при реализации структур КМУУ U_1 , U_2 , и Мура: ГСА содержит 70 % операторных вершин (*a*) и 90 % операторных вершин (*б*)

Исследование аппаратурных затрат показало эффективность применения предлагаемых методик для всех ГСА (рис. 5). Так, модель КМУУ U_2 требует меньше аппаратурных затрат в сравнении с базовым КМУУ U_1 и автоматом Мура. Отметим, что с увеличением доли операторных вершин в ГСА аппаратурные затраты для реализации автомата Мура увеличиваются, а для реализации устройства класса КМУУ — уменьшаются. Максимальный выигрыш при использовании структуры U_2 по сравнению с применением структуры автомата Мура достигал приблизительно 40 макроячеек для ГСА с 70 % операторных вершин и 55 макроячеек для ГСА с 90 % операторных вершин.

ЗАКЛЮЧЕНИЕ

Проведенные исследования позволили выявить уменьшение аппаратурных затрат при реализации КМУУ с несколькими источниками кодов в базе CPLD-микросхем. Это возможно благодаря таким главным факторам, как большой коэффициент объединения по входу современных макроячеек программируемой матричной логики, а также естественная избыточность микросхем ППЗУ, обусловленная тем, что количество их выходов ограничено определенным множеством чисел.

Отметим, что метод применим только для интерпретации линейных ГСА.

Дальнейшие исследования связаны с анализом процесса синтеза других моделей КМУУ, представленных в табл. 1, с сравнением параметров синтезированных схем, составлением аналитических зависимостей между параметрами исходных ГСА и характеристиками получаемых схем, которые позволят определять устройство, наилучшим образом удовлетворяющее выдвигаемым требованиям.

СПИСОК ЛИТЕРАТУРЫ

1. Baranov S. Logic and system design of digital systems. — Tallinn: TUT press, 2008. — 266 p.
2. Глушков В.М. Синтез цифровых автоматов. — М.: Физматгиз, 1962. — 476 с.
3. De Micheli G. Synthesis and optimization of digital circuits. — New York: McGraw-Hill, 1994. — 541 p.
4. Barkalov A., Titarenko L. Logic synthesis for compositional microprogram control units. — Berlin: Springer, 2008. — 272 p.
5. Баркалов А.А., Титаренко Л.А., Ефименко К.Н. Оптимизация схем композиционных микропрограммных устройств управления, реализуемых на ПЛИС // Кибернетика и системный анализ. — 2011. — № 1. — С. 179–188.
6. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах программируемой логики. — СПб: БХВ-Петербург, 2002. — 636 с.
7. Соловьёв В.В., Климович А.С. Логическое проектирование цифровых систем на основе программируемых логических интегральных схем. — М.: Горячая линия — Телеком, 2008. — 376 с.
8. Altera devices overview. — http://www.altera.com/products/devices/common/dev-family_overview.html.
9. Xilinx CPLDs. — http://www.xilinx.com/products/silicon_solutions/cplds/index.htm.
10. Баркалов А.А., Титаренко Л.А., Цололо С.А. Оптимизация схемы автомата Мура, реализуемой в базе ПЛИС // Кибернетика и системный анализ. — 2009. — № 5. — С. 180–186.
11. Barkalov A., Titarenko L. Logic synthesis for FSM-based control units. — Berlin: Springer, 2009. — 233 p.

Поступила 13.10.2011

После доработки 21.02.2012