

СИНТЕЗ КОМПОЗИЦИОННОГО МИКРОПРОГРАММНОГО УСТРОЙСТВА УПРАВЛЕНИЯ НА FPGA

Баркалов А.А., Ковалев С.А., Ефименко К.Н.

Донецкий национальный технический университет, г. Донецк

кафедра электронных вычислительных машин

E-mail: barkalov@cs.dgtu.donetsk.ua

Abstract

Barkalov A.A., Kovalev S.A., Efimenko K.N. Synthesis of compositional microprogram control unit of FPGA. The method of compositional microprogram control unit (CMCU) design permitting to decrease the amount of functions of addressing FSM depending on logic conditions and internal variables is proposed. Method is oriented on the application in the basis of FPGA. The method is based on the usage of the senior bits of the operational linear chains output address to initialize the transitions between different operational linear chains. The application of proposed method permits to decrease the amount of LUT-elements of FPGA, needed for implementation of CMCU.

1. Введение

Устройство управления (УУ) является одной из основных частей любой цифровой системы [1]. Успехи микроэлектроники привели к появлению «систем–на–кристалле», произвольная логика в которых, как правило, реализуется на FPGA (field-programmable gate array) [2,3]. В этой связи возникает необходимость разработки новых и совершенствования известных методов синтеза УУ, ориентированных на этот базис.

Характерной особенностью FPGA является наличие логических элементов табличного типа, так называемых LUT–элементов (look–up table). Число входов LUT–элементов ограничено и не превышает 6 [2,3], что вызывает необходимость функциональной декомпозиции систем булевых функций, задающих закон функционирования УУ [4]. Один из путей уменьшения числа LUT–элементов в схеме УУ — уменьшение числа функций с большим числом аргументов. В литературе решению этой задачи уделялось достаточно большое внимание [5,6]. Однако и в настоящее время она остается актуальной. Наиболее эффективным путем решения рассматриваемой задачи может быть переход к двухуровневой структуре УУ, в которой система выходных функций (микроопераций) определена более, чем на 50% возможных термов. В этом случае для реализации системы микроопераций целесообразно использовать блоки ЕАВ (embedded array block), входящие в состав современных систем на кристалле [2,3]. При таком подходе управляющая память, реализованная на основе ЕАВ, хранит микрокоманды, а произвольная логика их адресации реализуется на FPGA. Ограниченность ресурсов ЕАВ вызывает необходимость уменьшения разрядности микрокоманд. Всем этим целям идеально отвечают композиционные микропрограммные устройства управления (КМУУ), управляющая память которых не содержит адресов переходов [7]. В настоящей работе предлагается метод синтеза КМУУ с минимально возможным числом выходов схемы адресации микрокоманд.

2. Основные определения

Пусть алгоритм управления цифровой системы задан в виде граф–схемы алгоритма (ГСА) Γ [8], операторные вершины которой образуют множество $B = \{b_1, \dots, b_K\}$. В вершине $b_k \in B$ записан набор одновременно выполняемых микроопераций (микрокоманда) $Y(b_k) \subseteq Y$, где $Y = \{y_1, \dots, y_N\}$. В условных вершинах ГСА Γ записываются элементы множества логических условий $X = \{x_1, \dots, x_L\}$. Кроме операторных и условных вершин ГСА Γ содержит начальную b_0 и конечную вершины b_E . Пусть E — множество дуг ГСА Γ .

Введём ряд определений [7], необходимых для дальнейшего изложения материала.

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

Определение 2. Входом ОЛЦ α_g называется вершина $b_q \in V$, такая что существует дуга $\langle b_t, b_q \rangle \in E$, где b_t — условная или начальная вершина ГСА Γ , или операторная вершина, не входящая в ОЛЦ α_g .

Определение 3. Выходом ОЛЦ α_g называется вершина $b_q \in V$, такая что существует дуга $\langle b_q, b_t \rangle \in E$, где b_t — условная или конечная вершина ГСА Γ , или операторная вершина, не входящая в ОЛЦ α_g .

Обозначим через $D^g \subseteq V$ множество операторных вершин, входящих в ОЛЦ $\alpha_g \in C$, где $C = \{\alpha_1, \dots, \alpha_G\}$ — множество ОЛЦ ГСА Γ , удовлетворяющее условию

$$\begin{aligned} D^g \cap D^q &= \emptyset \quad (g \neq q, g, q \in \{1, \dots, G\}); \\ V &= D^1 \cup D^2 \cup \dots \cup D^G; \\ D^g &\neq \emptyset \quad (g = 1, \dots, G). \end{aligned} \tag{1}$$

Пусть для каждой ОЛЦ $\alpha_g \in C$ выполнена естественная адресация микрокоманд

$$A(b_{gi+1}) = A(b_{gi}) + 1 \quad (i = \overline{1, F_g - 1}), \tag{2}$$

где $A(b_q)$ — адрес микрокоманды, соответствующей вершине $b_q \in V$.

В этом случае ГСА Γ может быть интерпретирована КМУУ с базовой структурой (рис. 1), называемым в дальнейшем КМУУ U_1 .

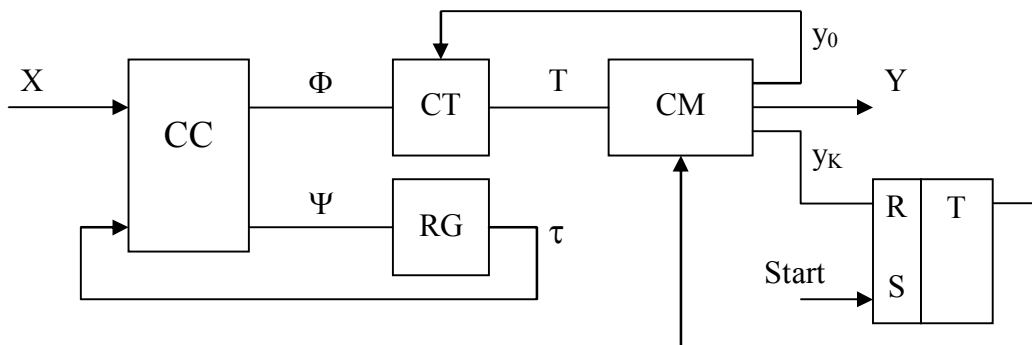


Рисунок 1 — Структурная схема КМУУ U_1

Здесь комбинационная схема CC и регистр RG образуют автомат адресации микрокоманд S_1 , а счетчик CT , управляющая память CM и триггер T образуют микропрограммное устройство управления S_2 с естественной адресацией микрокоманд [9], что соответствует условию (2). Регистр RG хранит код $K(a_m)$ текущего состояния $a_m \in A$, где $A = \{a_1, \dots, a_M\}$ — множество внутренних состояний автомата адресации, и имеет R разрядов, где $R = \lceil \log_2 M \rceil$. Для кодирования состояний $a_m \in A$ используются переменные $\tau_r \in \tau = \{\tau_1, \dots, \tau_R\}$, регистр RG имеет входы типа D . Счетчик CT хранит адреса $A(b_k)$ микрокоманд, представляемых переменными $T_r \in T = \{T_1, \dots, T_{R_1}\}$, где $R_1 = \lceil \log_2 K \rceil$. Микрокоманды $Y(b_k)$ хранятся в управляющей памяти, имеющей в случае унитарного кодирования микроопераций [9] $2^{R_1} \times (N + 2)$ бит. Один из дополнительных разрядов используется для хранения сигнала y_0 , по которому осуществляется режим адресации (2), второй — для хранения сигнала y_k , по которому происходит завершение функционирования устройства.

По сигналу Start содержимое RG и СТ обнуляются, что соответствует исходному состоянию КМУУ, триггер Т устанавливается в единицу, что позволяет считывать микрокоманды из СМ. При переходах внутри ОЛЦ $\alpha_g \in C$ сигнал $y_0 = 1$, при этом состояние автомата S_1 не меняется. Если выход текущей ОЛЦ достигнут, то $y_0 = 0$ и автомат S_1 формирует функции возбуждения регистра $\Psi = \{\Psi_1, \dots, \Psi_R\}$ и счетчика $\Phi = \{\Phi_1, \dots, \Phi_{R1}\}$

$$\begin{aligned} \Phi &= \Phi(\tau, X), \\ \Psi &= \Psi(\tau, X). \end{aligned} \tag{3}$$

Функции Φ и Ψ формируют в регистре код $K(a_S)$ состояния перехода автомата S_1 и адрес входа очередной ОЛЦ $\alpha_g \in C$ в счетчике соответственно. При достижении микрокоманды $Y(b_k) \subseteq Y$ такой, что $\langle b_k, b_E \rangle \in E$, формируется сигнал $y_k = 1$, триггер Т обнуляется и функционирование КМУУ U_1 прекращается.

Такие УУ имеют ряд положительных качеств [7], делающих целесообразным их применение в системах на кристалле:

1. Число функций с числом аргументов до $L + R$ ограничено параметром $t_1 = R + R_1$, что позволяет уменьшить число LUT-элементов в схеме УУ по сравнению, например, с реализацией УУ в виде микропрограммного автомата [8].

2. Управляющая память содержит только микрооперации $y_n \in Y$ и не содержит адреса переходов, что при принятой стратегии кодирования микроопераций [9] минимизирует требуемое число выходов блоков ЕАВ.

3. Согласно условию (1), число микрокоманд в СМ равняется K , что является минимально возможным параметром для микропрограммных устройств управления [9]. Это позволяет минимизировать ёмкость используемых блоков ЕАВ.

Итак, использование модели КМУУ для интерпретации ГСА позволяет уменьшить число LUT-элементов в схеме УУ и использовать минимально возможные ресурсы блоков ЕАВ для реализации системы микроопераций. В настоящей работе предлагается метод уменьшения числа выходов схемы СС, что приводит к дальнейшему уменьшению числа LUT-элементов в схеме УУ.

3. Основная идея метода

Пусть алгоритм управления цифровой системы задан ГСА Γ_1 (рис. 2).

Используем результаты работы [7] и построим для ГСА Γ_1 множество $C = \{\alpha_1, \alpha_2, \dots, \alpha_5\}$, где $\alpha_1 = \langle b_1, b_2, b_3 \rangle$, $\alpha_2 = \langle b_4, b_8, b_{12} \rangle$, $\alpha_3 = \langle b_5, b_9 \rangle$, $\alpha_4 = \langle b_6, b_{10} \rangle$, $\alpha_5 = \langle b_7, b_{11} \rangle$. Очевидно, что полученное множество C удовлетворяет условию (1). Выполним адресацию микрокоманд (2) и построим таблицу содержимого управляющей памяти (табл. 1).

Таблица 1 — Содержимое управляющей памяти КМУУ U_1 (Γ_1)

A(b _k)	Y(b _k)	Комментарии	A(b _k)	Y(b _k)	Комментарии
0000	y ₀ y ₁ y ₂	b ₁ I ₁ ¹	0110	y ₀ y ₂	b ₅ I ₃ ¹
0001	y ₀ y ₃ y ₄	b ₂ I ₁ ²	0111	y ₃ y ₄ y _k	b ₉ O ₃
0010	y ₂	b ₃ O ₁	1000	y ₀ y ₅	b ₆ I ₄ ¹
0011	y ₀ y ₃ y ₄	b ₄ I ₂ ¹	1001	y ₁ y ₂ y _k	b ₁₀ O ₄
0100	y ₀ y ₁ y ₂	b ₈	1010	y ₀ y ₃ y ₄	b ₇ I ₅ ¹
0101	y ₅ y _k	b ₁₂ O ₂	1011	y ₅ y _k	b ₁₁ O ₅

Здесь U_1 (Γ_1) обозначает, что КМУУ U_1 интерпретирует ГСА Γ_1 , I_g^j — j-й вход g-й ОЛЦ ($j \leq F_g$), O_g — выход g-й ОЛЦ ($g = \overline{1, G}$).

Как видно из табл. 1, все вершины, кроме выходов ОЛЦ, содержат сигнал y_0 для организации режима адресации (2), вершины, связанные с конечной вершиной b_E , содержат сигнал y_k для организации режима останова.

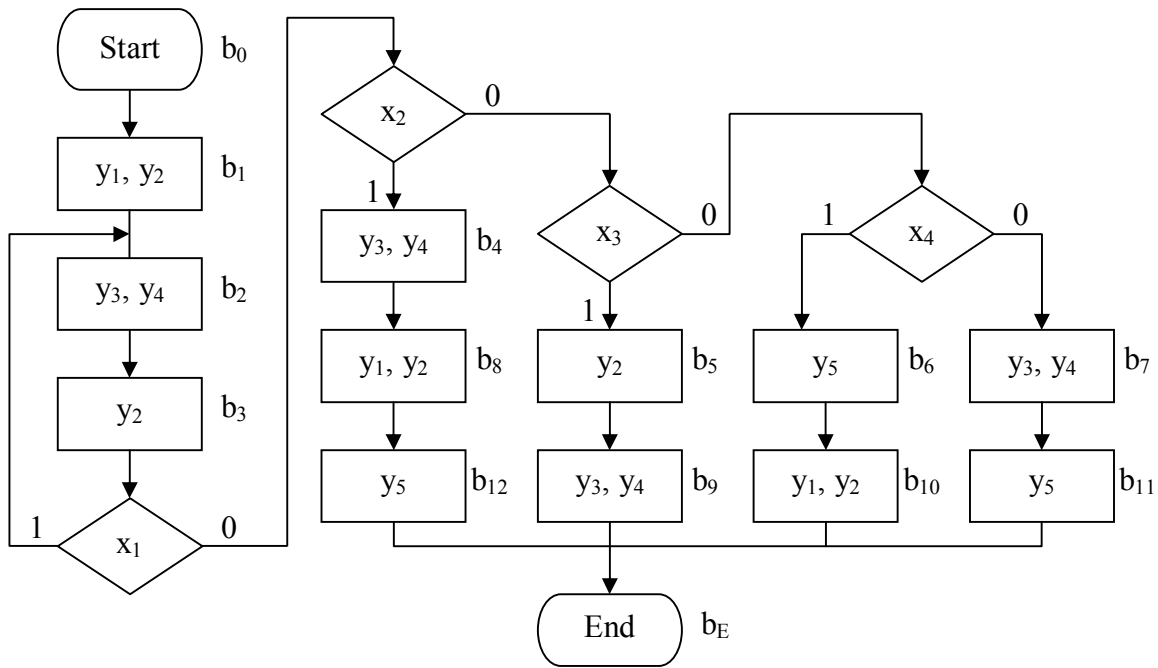


Рисунок 2 — Исходная граф-схема алгоритма Γ_1

Как видно из табл. 1, старшие $R_2 = 3$ разряда адресов выходов ОЛЦ $\alpha_g \in C$ для всех ОЛЦ имеют разные значения. В общем случае

$$R_2 = \lceil \log_2 G \rceil. \tag{4}$$

Таким образом, старшие R_2 разряда адреса выхода ОЛЦ могут использоваться для её однозначной идентификации.

В настоящей работе предлагается использовать это свойство для минимизации числа выходов схемы СС, чему соответствует композиционное микропрограммное устройство управления U_2 (рис. 3). Организация режима останова в КМУУ U_1 и U_2 идентична, поэтому триггер Т на рис. 3 не показан.

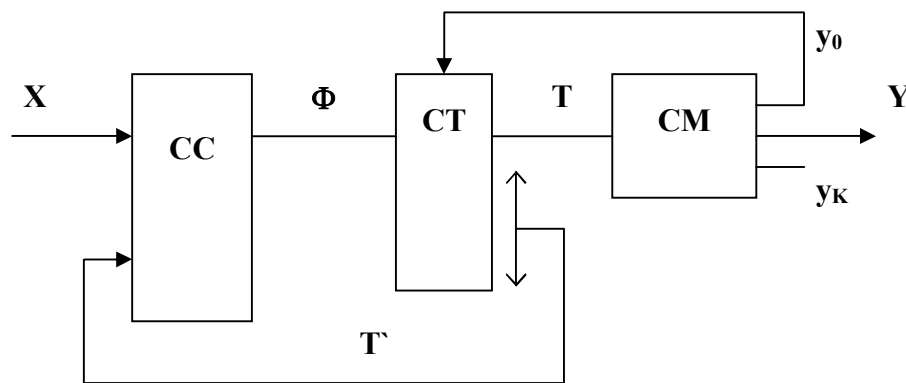


Рисунок 3 — Структурная схема КМУУ U_2

Принцип функционирования КМУУ U_1 и U_2 отличается только при организации перехода между различными ОЛЦ $\alpha_g \in C$. В этом случае сигнал y_0 не формируется и адрес входа очередной ОЛЦ заносится в СТ функциями

$$\Phi = \Phi(T', X), \tag{5}$$

где $\Gamma \subseteq T$, $|\Gamma| = R_2$, $\Gamma = \{T_1, \dots, T_{R_2}\}$. В настоящей работе предлагается метод синтеза КМУУ U_2 .

4. Метод синтеза композиционного микропрограммного устройства управления U_2

Предлагаемый в работе метод синтеза КМУУ U_2 включает следующие этапы:

1. Формирование множества ОЛЦ, адресация микрокоманд и формирование содержимого управляющей памяти. Этот этап выполняется по известной методике [7]. Для КМУУ U_2 (Γ_1) $R_2 = 3$, $\Gamma = \{T_1, T_2, T_3\}$.

2. Формирование таблицы переходов КМУУ. Эта таблица является основой для формирования системы функций (5) и последующего синтеза схемы СС.

Таблица переходов КМУУ U_2 содержит столбцы: O_g , $SA(O_g)$, I_q^j , $A(I_q^j)$, X_h , Φ_h , h , где O_g — выход ОЛЦ $\alpha_g \in C$; $SA(O_g)$ — старшие R_2 разрядов адреса микрокоманды, соответствующей выходу ОЛЦ $\alpha_g \in C$; I_q^j — j -й вход ОЛЦ $\alpha_g \in C$ ($j = \overline{1, F_g - 1}$); $A(I_q^j)$ — адрес входа I_q^j ; X_h — входной сигнал, определяющий переключение счетчика СТ из адреса выхода ОЛЦ α_g в адрес входа $A(I_q^j)$; $\Phi_h \subseteq \Phi$ — набор функций возбуждения счетчика СТ, принимающих единичное значение для переключения СТ из $A(O_g)$ в $A(I_q^j)$; $h = \overline{1, N}$ — номер перехода.

Для КМУУ U_2 (Γ_1) таблица переходов (табл. 2) содержит $N = 5$ строк.

Таблица 2 — Таблица переходов КМУУ U_2 (Γ_1)

O_g	$SA(O_g)$	I_q^j	$A(I_q^j)$	X_h	Φ_h	h
O_1	001	I_1^2	0001	x_1	D_4	1
		I_2^1	0011	$\overline{x_1 x_2}$	$D_3 D_4$	2
		I_3^1	0110	$\overline{x_1 x_2 x_3}$	$D_2 D_3$	3
		I_2^1	1000	$\overline{x_1 x_2 x_3 x_4}$	D_1	4
		I_3^1	1011	$\overline{x_1 x_2 x_3 x_4}$	$D_1 D_3 D_4$	5

Отметим, что переходы из выходов ОЛЦ $\alpha_g \in C$, связанных с конечной вершиной, не рассматриваются. Это связано с тем, что при достижении таких выходов функционирование КМУУ U_2 прекращается.

3. Формирование системы функций возбуждения счетчика. Система (5) формируется по таблице переходов КМУУ U_2 в виде

$$\varphi_r = \bigvee_{h=1}^N C_{rh} E_g^h X_h \quad (r = \overline{1, R_1}) \tag{6}$$

Здесь C_{rh} — булева переменная, равная единице, если и только если в h -й строке таблицы переходов записана функция $\varphi_r = 1$ ($h = \overline{1, N}$); E_g^h — конъюнкция переменных $T_r \in \Gamma$, соответствующая адресу $SA(O_g)$ выхода O_g из h -й строки таблицы:

$$E_g^h = \bigwedge_{r=1}^{R_2} T_r^{lgr}, \tag{7}$$

где $lgr \in \{0, 1\}$ — значение r -го разряда адреса $SA(O_g)$,

$$T_r^0 = \overline{T_r}, \quad T_r^1 = T_r \quad (r = \overline{1, R_2}).$$

Использование выражений (6) и (7) для нашего примера приводит к системе (5) в виде:

$$\begin{aligned}
 D_1 &= \overline{T_1 T_2 T_3 x_1 x_2 x_3 x_4} \vee \overline{T_1 T_2 T_3 x_1 x_2 x_3 x_4}, \\
 D_2 &= \overline{T_1 T_2 T_3 x_1 x_2 x_3}, \\
 D_3 &= \overline{T_1 T_2 T_3 x_1 x_2} \vee \overline{T_1 T_2 T_3 x_1 x_2 x_3} \vee \overline{T_1 T_2 T_3 x_1 x_2 x_3 x_4}, \\
 D_4 &= \overline{T_1 T_2 T_3 x_1} \vee \overline{T_1 T_2 T_3 x_1 x_2} \vee \overline{T_1 T_2 T_3 x_1 x_2 x_3 x_4}
 \end{aligned}$$

4. Синтез логической схемы КМУУ. Синтез сводится к реализации системы (5) на FPGA и реализации управляющей памяти на EAB. Вторая из этих задач является тривиальной, а первая достаточно полно рассмотрена в литературе [4]. Вопросы синтеза схемы КМУУ U_2 выходят за рамки нашей статьи.

Моделирование предлагаемого метода показало, что уменьшение числа LUT-элементов по сравнению с базовой структурой определяется соотношением между величинами R и R_1 , и может быть выражено коэффициентом

$$\eta = \frac{R + R_1}{R_1} = 1 + \frac{R}{R_1} \quad (8)$$

5. Заключение

Предлагаемый в работе метод синтеза КМУУ позволяет сократить число функций, реализуемых на LUT-элементах, от $t_1 = R + R_1$ (КМУУ с базовой структурой) до $t_2 = R_1$. Как показали исследования авторов, оптимизация числа LUT-элементов в схеме КМУУ U_2 по сравнению с КМУУ U_1 пропорциональна коэффициенту η (8).

При этом время такта КМУУ U_1 и U_2 совпадает, то есть оптимизация по аппаратным затратам не приводит к потере производительности. Недостаток метода заключается в увеличении числа сигналов обратной связи в КМУУ U_2 при выполнении условия

$$R_2 > R. \quad (9)$$

Как показали исследования, предлагаемый метод целесообразно применять, если условие (9) нарушается, то есть при $R_2 = R$. В этом случае число LUT-элементов в схеме СС КМУУ U_2 может уменьшаться на 35–42% по сравнению с аналогичным параметром КМУУ U_1 .

Литература

1. Баркалов А. А. Синтез операционных устройств. — Донецк: ДонНТУ, 2003. — 306 с.
2. Salcic Z. VHDL and FPLDs in digital systems design, prototyping and customization. — Kluwer Academic Publishers, 1998. — 549 p.
3. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах программируемой логики. — СПб: БХВ-Петербург, 2002. — 608 с.
4. Синтез цифровых устройств / Под редакцией Т. Лубы. — Варшава: ВКЛ, 2003. — 206 с.
5. Miller D.M., Maslov D., Dueck G.W. A transformation based algorithm for reversible logic synthesis // Proceedings – Design Automation Conference, 2003, pp. 318–323.
6. Mo F., Brayton R.K. Whirlpool PLAs: A regular logic structure and their synthesis // IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, 2002, pp. 543–550.
7. Баркалов А. А. Синтез устройств управления на программируемых логических устройствах — Донецк: ДонНТУ, 2002. — 262 с.
8. Баранов С. И. Синтез микропрограммных автоматов. — Л.: Машиностроение, 1978. — 234 с.
9. Баркалов А.А., Палагин А.В. Синтез микропрограммных устройств управления. — Киев: ИК НАН Украины, 1997. — 136 с.