

Синтез микропрограммного устройства управления с разделением кодов и модификацией операторных линейных цепей

Баркалов А.А.
University of Zielona Gora, Poland
A.Barkalov@iie.uz.zgora.pl

Зеленева И.Я., Мирошкин А.Н.
Кафедра ЭВМ ДонНТУ
irina@cs.dgtu.donetsk.ua
miroshkin@cs.dgtu.donetsk.ua

Аннотация

Баркалов А.А., Зеленева И.Я., Мирошкин А.Н. Синтез микропрограммного устройства управления с разделением кодов и модификацией операторных линейных цепей. В статье предлагается метод синтеза композиционного микропрограммного устройства управления. Метод ориентирован на уменьшение количества макроячеек PAL в комбинационной части устройства управления. Метод базируется на модификации операторных линейных цепей путем добавления в каждый управляющих микрокоманд, которые содержат коды классов псевдоэквивалентных цепей. К приведенному методу приведен пример.

Анотація

Баркалов О.О., Зеленцова І.Я., Мірошкін О.М. Синтез мікропрограмного пристрою керування з розділенням кодів та модифікацією операторних лінійних ланцюгів. У статті пропонується метод синтезу композиційного мікропрограмного пристрою керування. Метод орієнтований на зменшення кількості макрокомірок PAL у комбінаційній частині пристрою керування. Метод базується на модифікації операторних лінійних ланцюгів шляхом додавання до кожного керуючих мікрокоманд, які містять коди класів псевдоеквівалентних ланцюгів. До запропонованого методу приведений приклад.

Abstract

Barkalov A.A., Zeleneva I.Y., Miroshkin A.N. Syntheses of microprogram control unit with code sharing and modification of operational linear chains. The method of design for compositional microprogram control

units with code sharing is proposed. The method is oriented on reduction in the number of PAL macrocells in the combinational part of control unit. The method is based on modification of each operational linear chains by some additional control microinstructions, which contain codes of the classes of pseudoequivalent chains. Proposed method is illustrated by example.

Введение

Устройство управления является одним из основных блоков любой цифровой системы [1]. Если интерпретируемый алгоритм носит линейный характер, то для его реализации целесообразно использовать композиционное микропрограммное устройство управления (КМУУ) [2]. В настоящее время базис CPLD (complex programmable logic devices) [3,4] широко используется для реализации схем устройств управления [5,6]. При этом актуальной является проблема оптимизации аппаратных затрат в схеме устройства. В случае CPLD эта проблема решается путем преобразования исходных дизъюнктивных нормальных форм (ДНФ) функций автомата с целью минимизации числа термов в них, а, следовательно, и необходимого для их реализации количества макроячеек программируемой логики (ПМЛ). В настоящей работе предлагается один из путей решения этой проблемы: реализация управляющего устройства в виде КМУУ с разделением кодов [2] и модификацией системы микрокоманд.

Целью исследования является оптимизация комбинационной схемы КМУУ за счет введения дополнительных микрокоманд безусловного перехода и использования наличия классов псевдоэквивалентных операторных линейных цепей.

Задачей исследования является разработка метода синтеза КМУУ, позволяющего уменьшить число макроячеек ПМЛ в схеме формирования функций возбуждения памяти. При этом алгоритм управления представляется в виде граф-схемы алгоритма (ГСА) [7].

Особенности КМУУ с разделением кодов

Обозначим в некоторой ГСА Γ множество вершин $B = \{b_0, b_E\} \cup E_1 \cup E_2$ и множество дуг E . В этом множестве b_0 – начальная вершина, b_E – конечная вершина, E_1 – множество операторных вершин, E_2 – множество условных вершин. Операторная вершина $b_q \in E_1$ содержит набор микроопераций $Y(b_q) \subseteq Y$, где $Y = \{y_1, \dots, y_N\}$ – множество микроопераций (выходных сигналов). Условная вершина $b_q \in E_2$ содержит один элемент множества логических условий $X = \{x_1, \dots, x_L\}$ (входных сигналов). Под

линейной понимается ГСА, в которой число операторных вершин превосходит 75% общего числа вершин.

Операторная линейная цепь (ОЛЦ) является последовательностью операторных вершин граф-схемы алгоритма. Сформируем множество ОЛЦ $C = \{\alpha_1, \dots, \alpha_G\}$. При этом каждая пара соседних вершин ОЛЦ $\alpha_g \in C$ соответствует некоторой дуге из множества E [2]. Каждая ОЛЦ $\alpha_g \in C$ имеет произвольное число входов и только один выход O_g . Формальные определения ОЛЦ, их входов и выходов можно найти в [2]. Отметим, что каждая вершина $b_q \in E_1$ соответствует микрокоманде MI_q , хранимой в управляющей памяти (УП) по адресу A_q . Для адресации микрокоманд достаточно

$$R = \lceil \log_2 M \rceil \quad (1)$$

бит, где $M = |E_1|$. Пусть каждая ОЛЦ $\alpha_g \in C$ включает F_g компонент и пусть $Q = \max(F_1, \dots, F_G)$. Закодируем каждую ОЛЦ $\alpha_g \in C$ двоичным кодом $K(\alpha_g)$ разрядности

$$R_1 = \lceil \log_2 G \rceil, \quad (2)$$

а каждую компоненту $b_q \in E_1$ – двоичным кодом $K(b_q)$ разрядности

$$R_2 = \lceil \log_2 Q \rceil. \quad (3)$$

Используем для кодирования ОЛЦ элементы $\tau_r \in \tau$, а для кодирования компонент – элементы $T_r \in T$, где $|\tau| = R_1$, $|T| = R_2$. Кодирование компонент выполняется в естественном порядке, то есть

$$K(b_{gi}) = K(b_{gi-1}) + 1, \quad (4)$$

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

$$R_1 + R_2 = R \quad (5)$$

В этом случае для её интерпретации может быть использована модель КМУУ с разделением кодов, обозначаемая в дальнейшем символом U_1 [2]. Структурная схема КМУУ U_1 показана на рис.1.

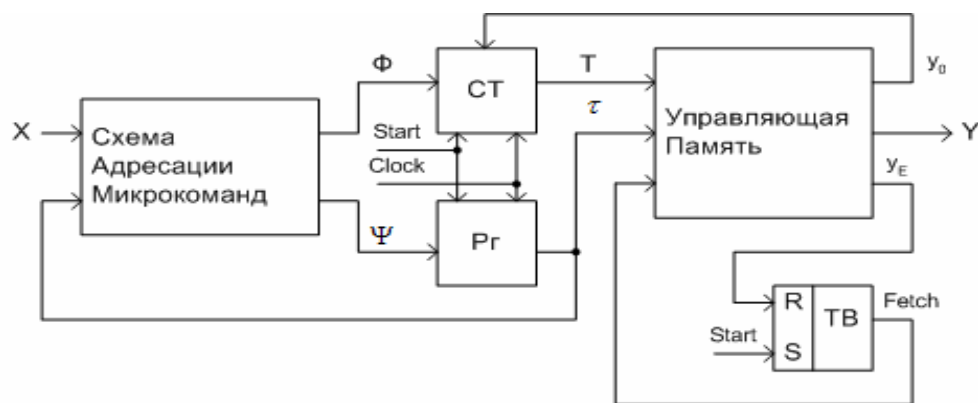


Рисунок 1 – Структурная схема КМУУ U_1

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

$$\begin{aligned}\Phi &= \Phi(\tau, X), \\ \Psi &= \Psi(\tau, X).\end{aligned}\tag{6}$$

При этом адрес микрокоманды MI_q представляется в виде

$$A(b_q) = K(\alpha_g) * K(b_q),\tag{7}$$

где b_q входит в 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 единичное состояние. При этом сигнал Fetch=1, что разрешает выборку микрокоманд из УП. Если считанная микрокоманда не соответствует выходу ОЛЦ, то одновременно с микрооперациями $Y(b_q)$ формируется сигнал y_0 . Если $y_0=1$, то к содержимому СТ прибавляется единица и адресуется следующая компонента текущей ОЛЦ. Если выход ОЛЦ достигнут, то $y_0=0$. При этом адрес входа следующей ОЛЦ формируется схемой САМ. При достижении окончания микропрограммы формируется сигнал y_E , триггер ТВ обнуляется и выборка микрокоманд прекращается.

Число термов в схеме САМ может быть уменьшено путем введения преобразователя кодов ОЛЦ в коды классов псевдоэквивалентных ОЛЦ (ПОЛЦ) [2]. ОЛЦ $\alpha_i, \alpha_j \in C$ называются псевдоэквивалентными, если их выходы связаны со входом одной и той же вершины ГСА Г. Однако реализация такого преобразователя требует дополнительных ячеек CPLD. В настоящей работе предлагается использовать свободные ячейки управляющей памяти с целью снижения аппаратурных затрат в схеме устройства.

Основная идея предлагаемого метода

Пусть $C_1 \subset C$ – множество ОЛЦ, выходы которых не связаны с вершиной $b_E \in b$. Найдем разбиение $\Pi_C = \{B_1, \dots, B_I\}$ множества C_1 на классы ПОЛЦ, где I – количество классов ПОЛЦ. Пусть для каждой ОЛЦ $\alpha_g \in C_1$ выполняется условие

$$2^{R_2} > F_g.\tag{8}$$

Закодируем классы $B_i \in \Pi_C$ двоичными кодами $K(B_i)$ разрядности

$$R_3 = \lceil \log_2 I \rceil\tag{9}$$

и используем для кодирования элементы множества Z , где $|Z| = R_3$. Введем в каждую ОЛЦ $\alpha_g \in C_1$ дополнительную микрокоманду MC_q , содержащую

$y_0 = 0$ и код $K(B_i)$, где $\alpha_g \in B_i$. При этом все микрокоманды MI_q содержат $y_0 = 1$.

В этом случае для интерпретации ГСА Γ предлагается КМУУ U_2 , структура которого приведена на рис.2.

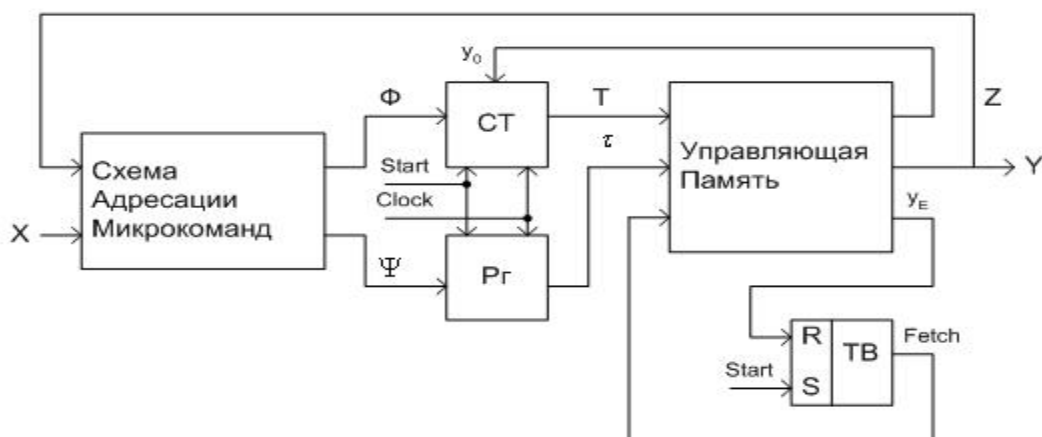


Рисунок 2 – Структурная схема КМУУ U_2

В КМУУ U_2 схема САМ реализует функции

$$\Phi = \Phi(Z, X), \quad (10)$$

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

а остальные элементы имеют то же назначение, что и для КМУУ U_1 .

Функции (10)-(11) генерируются, если из УП выбирается управляющая микрокоманда. В этом случае операционный автомат цифровой системы не получает микрооперации и находится в ждущем состоянии. Этого можно достичь, например, путем введения зависимости синхронизации операционного автомата от сигнала y_0 .

Отметим, что все элементы КМУУ U_2 (кроме УП) реализуются в составе кристалла CPLD. Для реализации УП необходимо использовать внешние ПЗУ.

В настоящей работе предлагается метод синтеза КМУУ U_2 , включающий следующие этапы:

1. Формирование множеств C, C_1, P_C для ГСА Γ .
2. Ввод дополнительных компонент в ОЛЦ $\alpha_g \in C_1$.
3. Кодирование ОЛЦ, их компонент и классов.
4. Формирование содержимого управляющей памяти.
5. Формирование таблицы переходов КМУУ.
6. Синтез логической схемы КМУУ.

Исследование эффективности предложенного метода

Найдём область эффективного применения КМУУ U_2 , используя для этого вероятностный подход, рассмотренный в [2]. Согласно этому подходу каждая ГСА Γ характеризуется долей операторных вершин p_1 . В случае линейных ГСА $p_1 \geq 0,75$. В исследовании используются матричные модели КМУУ [7], а не схемы в определённом базисе. При этом аппаратные затраты характеризуются площадью матриц, занимаемых схемами. Вывод об эффективности предложенного метода делается на основании исследования отношения

$$f = \frac{S(U_2)}{S(U_1)} \quad (12)$$

где $S(U_i)$ - площадь матричной реализации КМУУ U_i ($i=1,2$).
Матричная схема КМУУ U_1 представлена на рис.3.

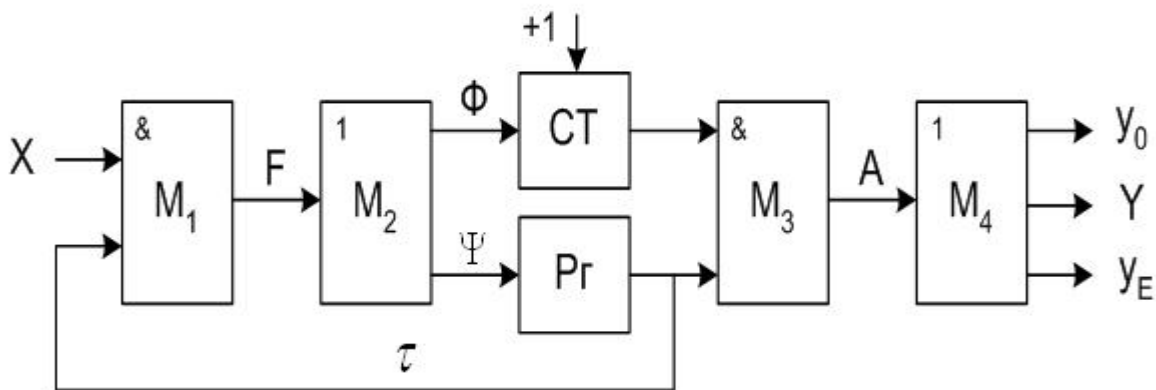


Рисунок 3 – Матричная реализация КМУУ U_1

На рис.3 опущены некоторые детали (Clock, Start, TB, Fetch) и связи, не влияющие на результат исследований. Здесь конъюнктивная матрица M_1 реализует систему термов $F = \{F_1, \dots, F_H\}$, соответствующих строкам таблицы переходов КМУУ. Дизъюнктивная матрица M_2 реализует функции (6), зависящие от термов $F_h \in F$. Конъюнктивная матрица M_3 реализует переменные $A_q \in A$, где A – множество адресов микрокоманд. Дизъюнктивная матрица M_4 реализует микрооперации $y_n \in Y$ и дополнительные переменные Y_0, Y_E . Площади $S(M_j)_1$ матриц M_j КМУУ U_1 ($j=1, \dots, 4$) определяются следующим образом:

$$S(M_1)_1 = 2(L + R_1)H; \quad (13)$$

$$S(M_2)_1 = HR; \quad (14)$$

$$S(M_3)_1 = 2R * 2^R; \quad (15)$$

$$S(M_4)_1 = 2^R(N + 2). \quad (16)$$

Матричная реализация КМУУ U_2 имеет такой же вид, как и в случае U_1 . Однако система обратных связей у обоих КМУУ отличается. При этом площади $S(M_j)_2$ определяются следующим образом:

$$S(M_1)_2 = 2(L + R_3)H_0; \quad (17)$$

$$S(M_2)_2 = H_0R; \quad (18)$$

$$S(M_3)_2 = S(M_3)_1; S(M_4)_2 = S(M_4)_1. \quad (19)$$

В формулах (17) и (18) переменная H_0 равняется числу строк таблицы переходов автомата Мили, эквивалентного определяющему САМ автомату Мура. Для уменьшения числа параметров в уравнениях (13) – (19) используем результаты работы [2]. Пусть K - число вершин ГСА Γ , тогда

$$R = \lceil \log_2(p_1K) \rceil; \quad (20)$$

$$L = (1 - p_1)K / 1,3; \quad (21)$$

$$H = 17,4 + 1,7k_1p_1K; \quad (22)$$

$$R_1 = \lceil \log_2(k_1p_1K) \rceil. \quad (23)$$

Как следует из (23), $G = k_1M$, где $k_1 \leq 1$ – коэффициент, обратно пропорциональный средней длине ОЛЦ (числу ее компонент) в интерпретируемой ГСА Γ . Для определения параметра R_3 необходимо найти число состояний эквивалентного автомата Мили для блока САМ:

$$I = 2,75 + 0,34k_1p_1K \quad (24)$$

Теперь формула (24) может быть использована как аргумент в (9). Параметр H_0 можно найти следующим образом:

$$H_0 = 4,4 + 1,1 * I \quad (25)$$

Для определения области, где КМУУ U_2 эффективнее КМУУ U_1 , необходимо исследовать функцию

$$f = \frac{S(M_1)_2 + \dots + S(M_4)_2}{S(M_1)_1 + \dots + S(M_4)_1} \quad (26)$$

зависящую от переменных K , p_1 , k_1 , N . Некоторые результаты наших исследований показаны на рис.4 и рис.5.

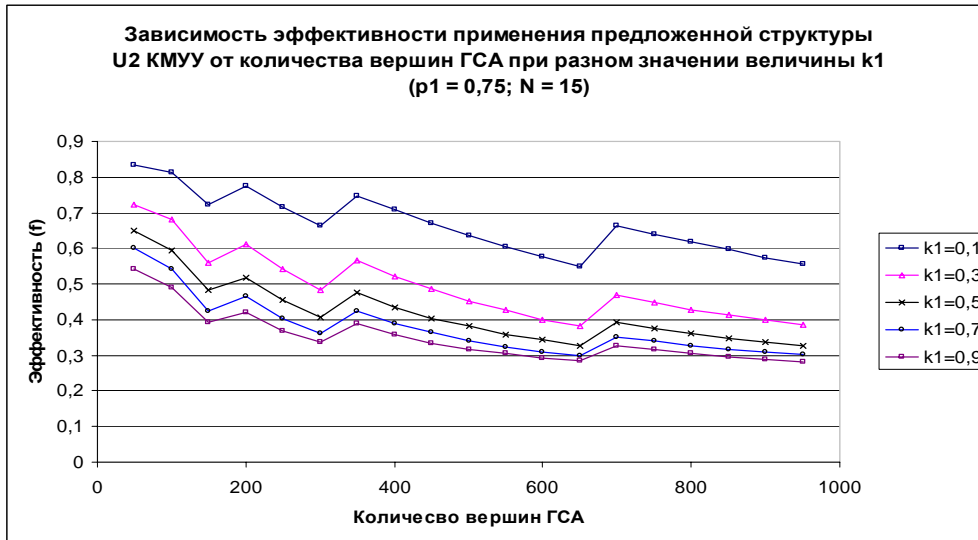


Рисунок 4 – Зависимость эффективности применения предложенной структуры U_2 КМУУ от количества вершин ГСА при разном значении величины k_1 ($p_1 = 0,75$; $N = 15$)

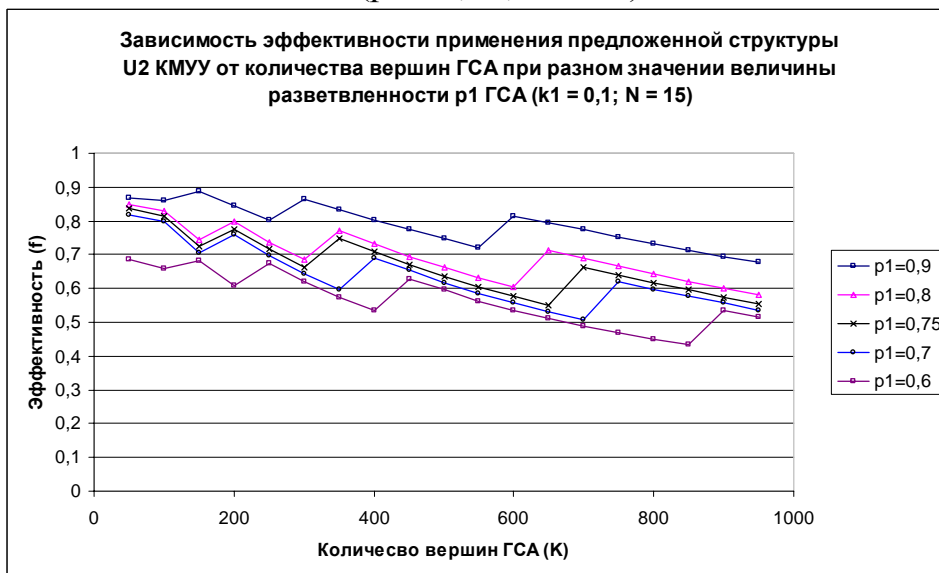


Рисунок 5 – Зависимость эффективности применения предложенной структуры U_2 КМУУ от количества вершин ГСА при разном значении величины разветвленности p_1 ГСА ($k_1 = 0,1$; $N = 15$)

Как следует из рис.4 и рис.5, при выполнении условия (8) КМУУ U_2 всегда требует меньше аппаратных затрат, чем КМУУ U_1 . При этом выигрыш увеличивается по мере увеличения коэффициента k_1 . Средний выигрыш при $p_1=0.75$, $k_1=0.1$ и $N=15$ составляет 47 %.

Как показывают исследования, использование структуры U_2 КМУУ для ГСА с большим коэффициентом разветвления дает меньший эффект. Средний выигрыш при $p_1=0.75$, $k_1=0.1$ и $N=15$ составляет 33 %.

Заключение

Предлагаемый метод модификации операторных линейных цепей КМУУ ориентирован на уменьшение числа макроячеек ПМЛ в схеме формирования адреса микрокоманд. При этом число микросхем ППЗУ, реализующих управляющую память, совпадает с этим числом для базовой структуры U_1 КМУУ с разделением кодов.

Метод основан на кодировании классов псевдоэквивалентных ОЛЦ, что позволяет уменьшить длину таблицы переходов по сравнению с этим параметром эквивалентного КМУУ U_1 .

Недостатком предлагаемого метода является увеличение числа тактов выполнения алгоритма в сравнении с КМУУ U_1 . Однако, побочным эффектом при уменьшении числа макроячеек ПМЛ может являться уменьшение числа уровней комбинационной схемы и, следовательно, уменьшение времени такта. Таким образом, окончательный вывод о среднем времени выполнения алгоритма может быть сделан только после реализации схем КМУУ $U_1(\Gamma_j)$ и $U_2(\Gamma_j)$. Исследования данного метода на произвольном наборе ГСА показали, что число макроячеек уменьшается до 30 %, а число уровней – на 2-3. Напомним, что применение этого метода возможно только для линейных ГСА при условии, что введение дополнительных микрокоманд не увеличивает разрядность адреса в сравнении с КМУУ U_1 .

Научная новизна предложенного метода заключается в использовании особенностей КМУУ (наличие классов псевдоэквивалентных ОЛЦ) и алгоритма управления (возможность введения дополнительных компонент ОЛЦ) для уменьшения числа макроячеек ПМЛ в схеме адресации микрокоманд.

Практическая значимость метода заключается в уменьшении аппаратных затрат при реализации схемы управляющего устройства, а также в улучшении его временных характеристик за счет уменьшения количества уровней схемы.

Дальнейшие направления исследований связаны с возможностью применения предложенного метода для базиса FPGA (Field-programmable gate arrays) [9], а также с разработкой САПР для синтеза схем композиционных микропрограммных устройств управления.

Литература

1. De Micheli G. Synthesis and Optimization of Digital Circuits. – NY: McGraw-Hill, 1994. – 636 pp.

2. Баркалов А.А. Синтез устройств управления на программируемых логических устройствах. Донецк: ДНТУ, 2002. – 262 с.
3. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем с использованием микросхем программируемой логики. – СПб: БХВ. – Петербург, 2002. – 608 с.
4. Соловьев В.В. Проектирование цифровых схем на основе программируемых логических интегральных схем. – М.: Горячая линия-ТЕЛЕКОМ, 2001. – 636 с.
5. Baranov S. Logic Synthesis for Control Automata – Boston: Kluwer Academic Publishers, 1994. – 312 pp.
6. Баркалов А.А., Бабаков Р.М., Ахмад Бадер. Исследование аппаратных характеристик автомата Мили с кодированием фрагмента микроопераций по VHDL- моделям // Искусственный интеллект. – 2007, №1, - С. 117-122.
7. Maxfield C. The Design Warrior's Guide for FPGA – Amsterdam : Elseteir, 2004. – 541 pp.



Баркалов Александр Александрович.

Д.т.н., професор кафедри ЕОМ Донецького національного технічного університету.
Наукові інтереси: моделювання цифрових пристроїв керування.



Зеленьова Ірина Яківна.

К.т.н., доцент кафедри ЕОМ Донецького національного технічного університету.
Наукові інтереси: цифрові пристрої керування, інтелектуальні системи прийняття рішень.



Мірошкін Олександр Миколайович.

У 2007 році отримав диплом магістра зі спеціальності "Системне програмування" у Донецькому національному технічному університеті. З 2008 року працює на кафедрі "Електронні обчислювальні машини".
Наукові інтереси: моделювання комп'ютерних пристроїв та систем.