

А.А. Баркалов, Я.Е. Визор, А.Н. Мирошкин

## Модификация алгоритма кодирования полей совместимых микроопераций

Предложена модификация алгоритма кодирования полей совместимых микроопераций, основанная на использовании копий некоторых сигналов, что дает определенную гибкость в формировании классов совместимых микроопераций. Описаны условия применения предложенной модификации алгоритма, показана ее эффективность в сравнении с классическим алгоритмом.

The modification of the algorithm of encoding of the sets of compatible microinstructions is suggested. The modification is based on using the copies of some signals. This leads to the more flexible forming of the sets of compatible microinstructions. The conditions of using the suggested modification of the algorithm are described, its efficiency as compared to the classic algorithm is shown.

Запропоновано модифікацію алгоритму кодування полів сумісних мікрооперацій, засновану на використанні копій деяких сигналів, що дає певну гнучкість у формуванні класів сумісних мікрооперацій. Описано умови запропонованої модифікації алгоритму, показано її ефективність в порівнянні з класичним алгоритмом.

**Введение.** При проектировании цифровых управляющих устройств (УУ) одной из актуальных задач является минимизация аппаратных затрат, необходимых для его реализации [1]. В настоящее время базис программируемых логических интегральных схем (ПЛИС) [2, 3] широко используется для реализации схем устройств управления [4, 5]. Если доля операторных вершин в интерпретируемой ГСА превышает 75%, речь идет о линейной ГСА. Для ее интерпретации целесообразно использовать композиционное микропрограммное устройство управления (КМУУ) [6]. Современные семейства микросхем *FPGA* содержат настраиваемые блоки встроенной памяти (БВП), что дает возможность реализовать КМУУ на одной микросхеме. Поскольку объем встроенной памяти невелик, актуальной становится проблема сокращения разрядности слова микропрограммы. В настоящей статье предлагается модификация метода кодирования полей совместимых микроопераций, что при выполнении определенных условий приводит к уменьшению разрядности управляющего слова.

*Цель исследования* – уменьшение аппаратных затрат в схеме КМУУ путем модификации алгоритма кодирования полей совместимых микроопераций и кодирования классов

псевдоэквивалентных операторных линейных цепей (ОЛЦ).

*Задача исследования* – разработка метода синтеза, позволяющего уменьшить аппаратные затраты, необходимые для реализации устройства управления.

### Основные положения

С момента предложения М. Уилксом в 1951 г. принципа микропрограммного управления [7] возникла проблема оптимального кодирования операционной части микрокоманд. Широкое применение получили три основных стратегии кодирования: максимальное, унитарное и кодирование полей совместимых микроопераций.

При максимальном кодировании операционная часть представляется в виде кода разрядности

$$m = \lceil \log_2 T \rceil, \quad (1)$$

где  $T$  – число наборов микроопераций (МО). При помощи дешифратора такой код преобразуется в унитарный код  $Y_1, \dots, Y_T$ , из которого по логике «ИЛИ» формируются сигналы микроопераций.

Недостаток данной стратегии – значительные временные задержки, вносимые дешифратором и схемой «ИЛИ» при большом количестве МО.

При унитарном кодировании каждой МО соответствует один разряд в операционной части микрокоманды (МК). Преимущество схемы – максимальная гибкость микропрограммирования и максимальное быстродействие. Однако разрядность МК линейно растет при увеличении количества МО.

Компромиссным решением является кодирование полей совместимых микроопераций, при котором множество МО делится на подмножества совместимых МО. Совместимыми называются такие МО, которые никогда не встречаются вместе в микрокомандах микропрограммы. В каждом подмножестве выполняется максимальное кодирование микроопераций. Полученный код размещается в соответствующем поле операционной части. Для получения сигналов МО каждому полю ставится в соответствие дешифратор.

Классический метод кодирования полей совместимых микроопераций предполагает разбиение множества микроопераций  $Y = \{y_1, \dots, y_N\}$  на подмножества (классы)  $G_1, \dots, G_I$ . Каждой группе  $G_i$  ставится в соответствие поле в МК разрядности

$$B_i = \lceil \log_2(|G_i| + C_i) \rceil, \quad (2)$$

где  $|G_i|$  – количество МО в классе  $G_i$ ;  $C_i$  – переменная, принимающая единичное значение в том случае, если существует такой набор  $Y_k$ , для которого  $Y_k \cap G_i = \emptyset$ .

Разрядность операционной части МК определяется как

$$B = \sum_{i=1}^I B_i. \quad (3)$$

Предложенный С. Шварцем алгоритм разбиения микроопераций на подмножества и дальнейшие модификации этого алгоритма базируются на следующих ограничениях:

- каждая МО  $y_n \in Y$  содержится только в одном классе  $G_i$  ( $i = 1, \dots, I$ );
- микрооперации  $y_n, y_m$ , принадлежащие одному набору  $Y_i$  ( $T = 1, \dots, T$ ), должны быть включены в разные классы  $G_i$ .

На данных ограничениях основывается и известный метод ветвей и границ [8], который дает решение задачи разбиения при меньших затратах времени. Алгоритм использует общую идею поиска с возвратом и состоит из четырех этапов:

1. Определение корня дерева поиска.
2. Выбор очередной размещаемой микрооперации.
3. Размещение очередной микрооперации.
4. Вычисление нижней границы решения.

На первом этапе алгоритма выбирается МК  $Y_i$ , содержащая максимальное количество микроопераций. Далее формируются корень дерева поиска, включающий в себя  $m$  одноэлементных множеств, где  $m = |Y_i|$ . После выбора корня остается распределить  $N - m$  микроопераций. Для упорядочения процесса выбора очередной микрооперации задается очередь МО, основанная на показателе совместимости  $E_n$ :

$$E_n = |Q_n|, \quad (4)$$

где  $Q_n$  – множество МО, совместимых с  $y_n$ . Первой в очередь размещается МО, имеющая минимальное значение показателя  $E_n$ . Классы  $G_i$  в свою очередь также упорядочиваются в соответствии с показателем  $Z_i$ , выражающим степень близости микроопераций из класса  $G_i$  с оставшимися нераспределенными МО.

$$Z_i = \left| \bigcap_{y_n \in G_i} Q_n \right|, \quad (5)$$

т.е. первым в очереди стоит класс, позволяющий разместить максимальное количество нераспределенных МО. Если при выборе очередной МК  $Y_i$  принадлежащая ей МО  $y_n$  не может быть включена ни в одну из  $m$  групп из-за нарушения ограничения 2, то число групп разбиения увеличивается на единицу и производится попытка разбиения множества МО  $Y$  на  $m + 1$  группу. Процедура продолжается до нахождения разбиения множества МО на  $m + q$  групп, удовлетворяющего условиям 1, 2 при минимально возможном значении  $q$ .

Рассмотрим применение данного метода на примере. Пусть некоторая ГСА  $\Gamma_1$  состоит из микрокоманд, приведенных в табл. 1.

Таблица 1. Список микрокоманд ГСА  $\Gamma_1$

МК	МО	МК	МО	МК	МО
$Y_1$	$y_1, y_2, y_3$	$Y_6$	$y_2, y_5, y_8$	$Y_{11}$	$y_1, y_2, y_8, y_9$
$Y_2$	$y_1, y_4$	$Y_7$	$y_1, y_4, y_6, y_8$	$Y_{12}$	$y_2, y_7, y_9$
$Y_3$	$y_1, y_4, y_5$	$Y_8$	$y_1, y_2, y_3, y_7$	$Y_{13}$	$y_3, y_4, y_8$
$Y_4$	$y_1, y_2, y_6$	$Y_9$	$y_1, y_4, y_9, y_{10}$	$Y_{14}$	$y_3, y_7, y_{10}$
$Y_5$	$y_4, y_6, y_7$	$Y_{10}$	$y_2, y_5, y_{10}$	$Y_{15}$	$y_1, y_6, y_7, y_{10}$

На первом шаге алгоритма из набора  $Y_7$  с максимальным количеством МО создаются четыре класса:  $G_1 = \{y_1\}$ ,  $G_2 = \{y_4\}$ ,  $G_3 = \{y_6\}$ ,  $G_4 = \{y_8\}$ . Для оставшихся МО определены множества совместимости  $Q_i$ :

$$\begin{aligned} Q_2 &= \{y_4\}; & Q_7 &= \{y_5, y_8\}; \\ Q_3 &= \{y_5, y_6, y_9\}; & Q_9 &= \{y_3, y_5, y_6\}; \\ Q_5 &= \{y_3, y_6, y_7, y_9\}; & Q_{10} &= \{y_8\}. \end{aligned} \quad (6)$$

Показатели совместимости для указанных множеств  $E_2 = E_{10} = 1$ ,  $E_7 = 2$ ,  $E_3 = E_9 = 3$ ,  $E_5 = 4$ , следовательно, очередь МО на распределение выглядит так:

$$y = (y_2, y_{10}, y_7, y_3, y_9, y_5). \quad (7)$$

Для классов  $G_i$  определим показатели  $Z_i$ , означающие количество нераспределенных МО, попарно совместимых со всеми МО из класса  $G_i$ :

$$Z_1 = 0; Z_2 = 1; Z_3 = 3; Z_4 = 2. \quad (8)$$

Из (8) следует, что для размещения очередной МО на втором шаге алгоритма классы следует расположить в следующем порядке:  $G_3$ ,  $G_4$ ,  $G_2$ ,  $G_1$ . Поскольку в классе  $G_3$  и  $G_4$  наблюдаются несовместимые с  $y_2$  микрооперации, распределение выполним в класс  $G_2$ . Продолжим распределение оставшихся микроопераций в классы, на каждом шаге алгоритма выполняя их упорядочение в порядке убывания параметра  $Z_i$ . Если при распределении МО  $y_n$

нарушается ограничение 2 для всех имеющихся классов, то образуется новое одноэлементное множество  $S_q = \{y_n\}$ , после чего необходимо продолжить поиск решения с учетом нового множества. В табл. 2 приведено решение для разбиения множества микроопераций ГСА  $\Gamma_1$  по описанному алгоритму.

Таблица 2. Анализ разбиения на множества совместимых микроопераций ГСА  $\Gamma_1$

Множество, $G_i$	Элементы, $y_i$	Количество элементов, $ G_i $	Разрядность кода, $\text{int}(\log_2  G_i )$
$G_1$	$y_1$	1	1
$G_2$	$\emptyset, y_2, y_4$	3	2
$G_3$	$\emptyset, y_3, y_5, y_6, y_9$	5	3
$G_4$	$\emptyset, y_8, y_{10}$	3	2
$G_5$	$y_7$	1	1

Общая разрядность операционной части составляет  $B = 1 + 2 + 3 + 2 + 1 = 9$  бит.

Современные микросхемы *FPGA* [5, 6] включают в себя блоки *EMB*, конфигурация которых (число слов  $N_W$  и число выходов  $N_O$ ) может меняться при сохранении постоянной емкости

$$V_{EMB} = N_W \cdot N_O. \quad (9)$$

У типичных представителей семейства *FPGA*  $V_{EMB} = 4K$  бит при конфигурациях  $4K*1$ ,  $2K*2$ ,  $1K*4$ ,  $512*8$ , т.е.  $N_O = \{1, 2, 4, 8\}$ . Для реализации блока БМО необходимы *EMB* с количеством слов  $N_W = 2^{R_A}$ , где  $R_A = \lceil \log_2 M \rceil$ , при этом число выходов  $t_F$  определяется как

$$t_F = \frac{V_{EMB}}{2^{R_A}}. \quad (10)$$

Таким образом, для реализации схем блока БМО необходимо

$$n_1 = \left\lceil \frac{B}{t_F} \right\rceil \quad (11)$$

блоков *EMB*, где  $B$  – общая разрядность операционной части.

Пусть ГСА  $\Gamma_1$  содержит  $M = 300$  вершин, тогда разрядность адреса  $R_A = \lceil \log_2 300 \rceil = 9$ .

Число выходов блока  $EMB$   $t_F = 4K / 2^9 = 8$ . Для реализации схемы блока БМО необходимо  $n_1 = \lceil 9/8 \rceil = 2$  блока  $EMB$ . Для определения эффективности использования блоков встроеной памяти введем показатель

$$\omega = \frac{V_{MP}}{V_{RAM}}, \quad (12)$$

где  $V_{MP}$  – объем микропрограммы,  $V_{RAM}$  – объем используемой памяти.

$$V_{MP} = M \cdot B, \quad (13)$$

$$V_{RAM} = n_1 \cdot V_{EMB}. \quad (14)$$

Для рассмотренного примера  $V_{MP} = 300 \cdot 9 = 2700$  бит,  $V_{RAM} = 2 \cdot 4K = 8K$ ,  $\omega = 2700/8192 = 0,33$ . Следовательно, используется лишь 33% от выделенного объема памяти.

### Основная идея предлагаемого метода

Модификация данного метода разбиения множества микроопераций на подмножества совместимых микроопераций основано на том, что ограничения 1 и 2 снимаются, что дает определенную свободу в формировании множеств. Для дальнейшего описания методики необходимо ввести несколько определений.

**Определение 1.** *Независимой* называется микрооперация, входящая только в один из классов  $G_i$ . Соответственно, микрооперация, которая входит больше, чем в один класс, называется *зависимой*. Множество независимых МО обозначим как  $Y_I$ , а зависимых – как  $Y_D$ .

**Определение 2.** Микрооперации  $y_i$  и  $y_j$  называются *условно-совместимыми*, если принадлежат к различным множествам ( $y_i \in Y_I$ ,  $y_j \in Y_D$ ,  $i \neq j$ ), относятся к одному классу ( $y_i, y_j \in G_k$ ), встречаются вместе в микрокоманде  $Y_n$ , и существует такой класс  $G_m$ , что  $Y_n \cap G_m = y_j$ .

**Определение 3.** Микрооперации  $y_i$  и  $y_j$  называются *эквивалентными*, если во всех вершинах ГСА они встречаются только вместе.

Модифицированный алгоритм разбиения множества МО на классы состоит из следующих шагов:

1. Выделить группы эквивалентных микроопераций. В каждой группе оставить по одному элементу для уменьшения количества микроопераций и, как следствие, – сложности алгоритма разбиения.

2. Найти микрокоманду  $Y_i$  такую, что  $|Y_i| = \max(|Y_1|, \dots, |Y_N|) = m$ . Сформировать  $m$  классов, в которые включить микрооперации  $y_i \in Y_i$ , ( $i = 1, \dots, m$ ).

3. Для оставшихся  $N - m$  микроопераций определить множества попарно совместимых микроопераций. Расположить нераспределенные микрооперации в порядке увеличения мощности соответствующих множеств. Зависимые МО расположить в конце списка.

4. Выполнить распределение согласно классическому алгоритму деления на классы совместимых микроопераций, анализируя наборы микроопераций, содержащие зависимые МО, в последнюю очередь.

5. При распределении зависимой МО распределить ее копии в минимально необходимое количество классов.

Применение алгоритма продемонстрируем на примере ГСА  $\Gamma_1$ .

Использование предлагаемого метода возможно как для нахождения решения, начиная с первого шага, так и для модификации готового решения, найденного одним из известных методов. Применим его к решению, приведенному в табл. 2.

Эффективность предлагаемого алгоритма зависит от правильности выбора зависимых МО. В качестве предполагаемых зависимых микроопераций авторы статьи предлагают выбирать такие, которые образуют одноэлементные множества. В приведенном примере такими являются МО  $y_1$  и  $y_7$ . Показатели совместимости для этих микроопераций  $E_1 = 0$  и  $E_7 = 2$ , следовательно, выберем  $y_7$  в качестве зависимой МО.

Для возможности применения метода необходимо, чтобы предполагаемая микрооперация была попарно или условно совместимой со всеми МО из множества  $Y$ . Микрооперация  $y_7$

встречается в микрокомандах  $Y_5, Y_8, Y_{12}, Y_{14}, Y_{15}$ . Для каждой микрооперации найдем класс  $G_i$  такой, чтобы  $Y_i \cap G_i = \emptyset$ :

$$Y_5 : \{G_1, G_4\}, Y_8 : \{G_4\}, Y_{12} : \{G_1, G_4\}, \\ Y_{14} : \{G_1, G_2\}, Y_5 : \{G_2\}.$$

Из рассмотрения необходимо исключить те множества, включение дополнительного элемента в которые увеличит разрядность, необходимую для кодирования. Поскольку класс  $G_1$  – одноэлементное множество, его необходимо исключить:  $Y_5 : \{G_4\}, Y_8 : \{G_4\}, Y_{12} : \{G_4\}, Y_{14} : \{G_2\}, Y_5 : \{G_2\}$ .

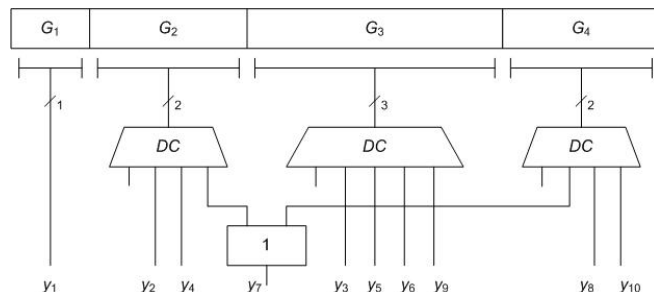
В результате выполнения алгоритма не получено пустых множеств, следовательно, МО  $y_7$  условно совместима с остальными элементами множества  $Y$ . После включения ее в классы  $G_2$  и  $G_4$  решение изменится так, как показано в табл. 3.

**Таблица 3.** Модификация разбиения на множества совместимых микроопераций ГСА  $G_1$

Множество, $G_i$	Элементы, $y_i$	Количество элементов, $ G_i $	Разрядность кода, $\text{int}(\log_2  G_i )$
$G_1$	$y_1$	1	1
$G_2$	$\emptyset, y_2, y_4, y_7$	3	2
$G_3$	$\emptyset, y_3, y_5, y_6, y_9$	5	3
$G_4$	$\emptyset, y_8, y_{10}, y_7$	3	2

Общая разрядность операционной части составляет  $B = 1 + 2 + 3 + 1 = 8$  бит. Для ГСА  $G_1$ , которая содержит  $M = 300$  вершин, разрядность адреса  $R_A = \lceil \log_2 300 \rceil = 9$ . Число выходов блока ЕМВ  $t_F = 4K / 2^9 = 8$ . Для реализации схемы блока БМО необходим  $n_1 = \lceil 8 / 8 \rceil = 1$  блок ЕМВ. Размер микропрограммы составляет  $V_{MP} = 300 \cdot 8 = 2400$  бит,  $V_{RAM} = 1 \cdot 4K = 4K$ , показатель эффективности использования ресурсов памяти  $\omega = 2400 / 4096 = 0,586$ . Следовательно, в сравнении с применением классического метода снизились затраты ресурсов памяти (ис-

пользуется на один блок ЕМВ меньше). Одновременно с этим повысилась эффективность использования занимаемых ресурсов (с 33 до 58,6 %).



**Рис. 1.** Структурная схема формирования МО ГСА  $G_1$

**Заключение.** Предлагаемый метод основан на включении копий определенных микроопераций в различные классы совместимых микроопераций. При выполнении необходимых условий такой метод позволяет снизить разрядность операционной части микрокоманды в сравнении с классической стратегией кодирования полей совместимых микроопераций, что позволяет сократить аппаратные затраты на реализацию таких автоматов на заказных микросхемах и в некоторых случаях позволяет уменьшить количество стандартных микросхем.

Отметим, что наблюдается некоторое ухудшение временных показателей, поскольку в схему вводится подсхема формирования зависимых микроопераций. Иногда такое ухудшение допустимо, если применение предложенного метода даст существенную экономию аппаратных затрат при реализации схемы устройства.

Практическая значимость предложенного метода состоит в сокращении аппаратных затрат в сравнении с известными методами реализации КМУУ с общей памятью.

Дальнейшие исследования связаны с формализацией алгоритма выбора зависимых МО, что должно привести к увеличению эффективности предложенного метода.

1. De Micheli G. Synthesis and Optimization of Digital Circuits. – NY: McGraw-Hill, 1994. – 636 p.
2. Грушвицкий Р.И., Мурсаев А.Х., Узрюмов Е.П. Проектирование систем с использованием микросхем программируемой логики. – СПб: БХВ–Петербург, 2002. – 608 с.

3. Соловьев В.В. Проектирование цифровых схем на основе программируемых логических интегральных схем. – М.: Горячая линия-ТЕЛЕКОМ, 2001. – 636 с.
4. Baranov S. Logic Synthesis for Control Automata – Boston: Kluwer Acad. Publi., 1994 – 312 p.
5. Баркалов А.А., Бабаков Р.М., Ахмад Бадер. Исследование аппаратурных характеристик автомата Мили с кодированием фрагмента микроопераций по VHDL-моделям // Искусственный интеллект. – 2007. – № 1. – С. 117–122.
6. Barkalov A., Titarenko L. Logic Synthesis for Compositional Microprogram Control Units. – Berlin: Springer, 2008. – 272 p.
7. Баркалов О.О. Синтез операційних пристроїв. – Донецьк: РВА ДонНТУ, 2003. – 305 с.
8. Bayer J., Koyama B. On the minimization of the width of the control memory of microprogramed processors // IEEE Trans. On Computers. – 1979. – № 4. – P. 310–316.
9. Embedded memory in Altera FPGA's. – <http://www.altera.com/technology/memory/embedded/mem-embedded.html>
10. Embedded memory in Altera FPGA's. – [http://www.xilinx.com/products/design\\_resources/mem\\_corner/resource/xawembedded.htm](http://www.xilinx.com/products/design_resources/mem_corner/resource/xawembedded.htm)

Поступила 01.10.2009

Тел. для справок: + 375 (17) 284-2084 (Донецк, Киев)

E-mail: A.Barkalov@iie.uz.zgora.pl, yaviz@ukr.net

© А.А. Баркалов, Я.Е. Визор, А.Н. Мирошкин, 2009

### Правила подготовки материалов

К рассмотрению принимаются не опубликованные ранее работы по тематике, приведенной на второй странице обложки журнала. Все статьи рецензируются. Решение редколлегии по содержанию каждого номера журнала утверждается ученым советом МНУЦИТиС. Одобренные к печати материалы редактируются. В случае отклонения рукописи один экземпляр и рецензия возвращаются автору. В одном номере журнала публикуется только одна статья автора, в том числе и в соавторстве.

В редакцию необходимо представить:

1. Рукопись (2 экз.), напечатанную через два интервала, объемом не более 16 страниц, на одной стороне листа формата А4 (кегель 12). Один экземпляр должен быть подписан автором(ами).

Страницы оригинала должны быть пронумерованы и иметь поля: левое – 25 мм, правое – 10 мм, верхнее – 20 мм, нижнее – 25 мм.

2. Аннотацию (2 экз.), напечатанную на отдельной странице (до 5 строк) с указанием фамилии автора(ов) и названия статьи на русском, украинском и английском языках; через два интервала.

3. Сопроводительное письмо организации за подписью руководителя.

4. Дискету 3,5" с текстом статьи, аннотацией и иллюстрациями.

5. Сведения об авторе(ах) – фамилия, имя, отчество, ученая степень, место работы, должность, адрес, телефон, факс, e-mail.

6. Копию квитанции о подписке на журнал УСИМ (не менее чем на полгода).

В начале статьи необходимо указать индекс УДК. Используемая литература приводится общим списком в конце статьи в порядке упоминания. Графики, рисунки и таблицы с подписями должны быть распечатаны на отдельных страницах либо выполнены тушью для сканирования.

Для подготовки текста на дискете необходимо использовать редактор *Microsoft Word* любой версии (шрифт *Times New Roman*; кегль 12, интервал двойной; отступ 1 см.), для набора формул – редактор *Microsoft Equation Editor v. 2.0/3.0* из состава *Microsoft Office*. Иллюстрации могут быть выполнены в любом графическом редакторе.

Материалы можно высылать электронной почтой (по адресу [gor@usm.freenet.kiev.ua](mailto:gor@usm.freenet.kiev.ua)) с обязательным дублированием на бумаге в двух экземплярах или почтой (простое письмо).

В соответствии с постановлением президиума ВАК Украины от 15.01.2003 г. № 7-05/1 «Про підвищення вимог до фахових видань, внесених до переліків ВАК України» статьи, принимаемые к опубликованию, должны состоять из следующих элементов:

- постановка проблемы и ее связь с научными или практическими заданиями;
- анализ последних исследований и публикаций (где начато разрешение данной проблемы), на которые опирается автор;
- выделение неразрешенной части общей проблемы, чему посвящена предлагаемая статья;
- формулировка цели статьи (постановка задачи);
- изложение основного материала исследований с полным обоснованием полученных научных результатов;
- выводы из данного исследования и перспективы дальнейших разработок в данном направлении.

Редакция обращается с просьбой к авторам, желающим опубликовать статью в нашем журнале на украинском или английском языке, прилагать к направляемым материалам русский аналогичный вариант текста.

*Редколлегия*