A. Kh. Mursaev
Saint Petersburg Electrotechnical University «LETI»

IMPLEMENTATION OF FUNCTIONAL TRANSFORMATION
IN THE PROGRAMMABLE LOGIC CHIPS

Functional transformation, that is, the formation of the code, which numerical equivalent is a defined function of the input code numerical equivalent, is a widespread operation in the measurement, control and similar systems. The implementation of such transformations in universal calculators (e. g. microprocessors) often does not meet the requirements of consumers in terms of performance or energy effectiveness. A review of the functional transformation methods is presented: tabular approximation, polynomial approximation, calculations according Volder method. The implementations into program logic microchips using different variants of spatial and temporal distribution of calculations are considered. It is shown that performance in the range up to 107 transformations per second with moderate costs is achievable.

Functional transformation, hardware implementation, programmable logic

УДК 519.7

С. И. Тодиков
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Минимизация мультиопераций в классе ключевых стандартных форм

Рассматривается проблема минимизации мультиопераций в классе ключевых стандартных форм. В основе работы лежит разработанный алгоритм минимизации мультиопераций для n = 2, 3 в классе ключевых стандартных форм. Данный алгоритм основан на анализе мультиопераций и последующим наилучшим заменам всех нулевых элементов для получения минимального представления мультиоперации в классе ключевых стандартных форм. С помощью разработанного алгоритма получены все минимальные представления мультиопераций для n = 2, 3 в классе ключевых стандартных форм, средняя сложность минимального представления мультиопераций в классе ключевых стандартных форм и количественное распределение мультиопераций по сложностям полученных минимальных представлений. Произведено сравнение полученных результатов минимизации мультиопераций в классе ключевых стандартных форм с минимизацией мультиопераций в классе стандартных форм. При сравнении результатов делается вывод, что минимальное представление мультиопераций в классе стандартных форм лучше, чем в классе ключевых стандартных форм.

Мультиоперация, суперклоны, ключевая стандартная форма, алгоритм, минимизация

В теории дискретных функций, помимо всюду определенных функций k-значной логики, изучаются и функции, определенные не на всех наборах. В этом случае неопределенность понимается по-разному, в зависимости от рассматриваемого класса задач. Не всюду определенными функциями на конечном множестве являются и мультиоперации. У данного вида функций неопределенность понимается как неодноэлементное (в том числе и пустое) подмножество конечного множества, на котором эти функции заданы. В настоящее время изучаются разные вопросы теории мультиопераций, в частности вопросы полноты [1], связь клонов функций и клонов мультиопераций [2], рассматриваются вопросы функциональной разделимости [3].

В данной работе затрагивается вопрос минимизации мультиоперации в классе ключевых стандартных форм, приводится алгоритм минимизации и результаты минимизации мультиопераций ранга 2.

Основные понятия. Пусть 2^A – множество всех подмножеств A. Отображение из A^n в 2^A называется n-местной мультиоперацией на A.

Множество всех n-местных мультиопераций на A будем обозначать через H_A^n. При $|A| = k$ будем использовать обозначение H_k^n. Мультиоперации заданные на k-элементном множестве A будем называть мультиоперациями ранга k.

Используя главные операции суперклона [4] можно определить:

1) операцию ε_i^n, выделяющую мультиоперацию-проекцию по i-му аргументу

$$\varepsilon_i^n = e_i^n,\ e_i^n(a_1, \ldots, a_n) = \{a_i\};$$

2) операцию транспозиции i-го с j-м аргументов

$$(\tau_{i,j} f)(a_1, ..., a_i, ..., a_j, ..., a_n) = \\ = f(a_1, ..., a_j, ..., a_i, ..., a_n);$$

3) операцию отождествления j-го аргумента с i-м аргументом

$$(\Delta_{i,j} f) = (a_1, \ldots, a_{j-1}, a_{j+1}, \ldots, a_n) = \\ = f(a_1, \ldots, a_{j-1}, a_i, a_{j+1}, \ldots, a_n);$$

4) операцию подстановки на место i-го аргумента

$$(f *_i g)(a_1, \ldots, a_{n+m-1}) = \\ = \bigcup_{b \in g(a_i, \ldots, a_{i+m-1})} f(a_1, \ldots, a_{i-1}, b, a_{i+m}, \ldots, a_{n+m-1});$$

5) частичную операцию суперпозиции

$$(f * (f_1, ..., f_n))(a_1, ..., a_m) = \\ = \bigcup_{b_i \in f_i(a_1, ..., a_m)} f(b_1, ..., b_n);$$

6) операцию разрешимости по i-му аргументу

$$(\mu_i f)(a_1, ..., a_n) = \{a \mid a_i \in \\ \in f(a_1, ..., a_{i-1}, a, a_{i+1}, ..., a_n)\}.$$

Следуя [5], n-местную мультиоперацию f на множестве $A = \{1, 2\}$ будем представлять как отображение:

$$f\{1,2\}^n \to \{0, 1, 2, 3\},$$

используя следующую кодировку:

$$\{\varnothing\} \to 0;\ \{1\} \to 1;\ \{2\} \to 2;\ \{1, 2\} \to 3.$$

Также n-местную мультиоперацию f можно представить в виде вектора всех ее значений $(\alpha_1, ..., \alpha_{2^n})$, где $f(a_1, \ldots, a_n) = \alpha_i$, если $(a_1 - 1, \ldots, a_n - 1)$ есть представление числа $i - 1$ в двоичной системе счисления.

Определим бинарную мультиоперацию $\cap \in H_k^2$ как $\cap(a, b) = \{a\} \cap \{b\}$. В дальнейшем будем использовать суффиксную форму записи $\cap(a, b) = a \cap b$. Эта мультиоперация коммутативная и ассоциативная. Также она принадлежит любому суперклону [4], что делает естественным ее использование для построений формульных представлений мультиопераций.

Через $d_{i,\alpha}^i \in H_k^n$ обозначим следующие мультиоперации:

$$d_{i,\alpha}^n = (2^k - 1, \ldots, 2^k - 1, \alpha^i, 2^k - 1, \ldots, 2^k - 1), \\ (1 \le i \le k^n),$$

где $\alpha \in \{0, \ldots, 2^k - 1\}$, в частности $d_{1,\alpha}^0 = (\alpha)$. Если $\alpha = 2^k - 1$, то используем обозначение d^n, т. е. $d^n = (2^k - 1, \ldots, 2^k - 1)$.

В [2] была введена ключевая стандартная форма мультиопераций

$$f(x_1, \ldots, x_n) = \bigcap_j d_j(x_{j_1}, \ldots, x_{j_m}),$$

когда выполняется $d_j \in \left(f,\ d_{1,1}^1,\ \ldots,\ d_{s,2^{s-1}}^1,\ \ldots, d_{k,2^{k-1}}^1\right)$, $s \in \{1, \ldots, k\}$.

Также в [2] была введена лемма, говорящая о том, что если $f \in H_k^n$ и $f \ne d^n$, то существует разложение по аргументу x_i

$$f(x_0, \ldots, x_n) = f_0 \cap f_1 \cap f_2 \cap \ldots \\ \cap f_{2^{s-1}} \cap \ldots \cap f_{2^{k-1}},$$

где в f_0 аргумент x_i – фиктивный, а f_2^{k-1} такие, что 2^{r-1} – остаточные по аргументу x_i при $r \ne s$, равном d^{n-1}, при этом выполняется $f_0,\ f_{2^{s-1}} \in \\ \in \left(f,\ d_{1,1}^1,\ \ldots,\ d_{s,2^{s-1}}^1,\ \ldots,\ d_{k,2^{k-1}}^1\right)$, $s \in \{1,\ \ldots,\ k\}$.

Пример 1. Представим мультиоперацию $f \in H_2^n$ заданной векторно $f(x_1,\ x_2,\ x_3) = (20312001)$ в ключевой стандартной форме:

$$f(x_1, x_2, x_3) = d_{2,1}^1(x_3) \cap d_{1,2}^1(x_2) \cap d_{7,0}^3(x_1, x_2, x_3).$$

Также данную мультиоперацию можно представить в следующем виде:

$$f(x_1, x_2, x_3) = d^1_{2,1}(x_3) \cap d^1_{1,2}(x_2) \cap$$
$$\cap d^2_{2,0}(x_2, x_3) \cap d^2_{3,2}(x_1, x_3) \cap$$
$$\cap d^2_{4,1}(x_1, x_3) \cap d^2_{3,2}(x_1, x_2) \cap d^2_{4,1}(x_1, x_2) \cap$$
$$\cap d^3_{6,0}(x_1, x_2, x_3) \cap d^3_{7,0}(x_1, x_2, x_3).$$

Видно, что представление мультиопераций в ключевой стандартной форме не единственное.

Алгоритм минимизации мультиопераций в классе ключевых стандартных форм. Под минимальностью будем понимать представление мультиоперации в виде ключевой стандартной формы с наименьшим количеством компонент пересечения. Количество компонент пересечения в представлении назовем его сложностью.

Работа алгоритма заключается в наилучших заменах нулевых элементов мультиоперации с помощью леммы, так как именно на этих шагах ключевая стандартная форма дает разные сложности.

На вход алгоритма подается вектор, представляющий мультиоперацию.

На выходе алгоритма получим минимальную ключевую стандартную форму, представляющую мультиоперацию и сложность ее формы.

Работу алгоритма можно представить следующими шагами:

Шаг 1. Выполнить основные шаги алгоритма представления мультиоперации в ключевой стандартной форме до момента замены нулевых элементов мультиоперации, где начинает работать лемма.

Шаг 2. Если в мультиоперации нет ни одного нулевого элемента, то перейти к шагу 1, иначе – к шагу 3

Шаг 3. Если в мультиоперации нулевой элемент можно заменить на элемент 2^{k-1}, то производим замену. Если на данном шаге все нулевые элементы заменяются на элементы 2^{k-1}, то возвращаемся на шаг 1.

Шаг 4. Если с помощью замены нулевых элементов можно построить одинаковую последовательность элементов в мультиоперации, то производим замену, создаем последовательность и переходим к шагу 1, иначе – к шагу 5.

Шаг 5. Производим замену нулевых элементов с помощью элементов, которые получаются по остаточной мультиоперации от последней переменной в мультиоперации и возвращаемся к шагу 1.

Продемонстрируем работу алгоритма. Процесс нахождения ключевой стандартной формы представим в виде трех таблиц. В таблицах приведены мультиоперации, получаемые после применения леммы по каждой переменной. Для большего понимания алгоритма все шаги по переменной x_1 рассмотрим более подробно.

Пример 2. На вход алгоритма подается мультиоперация $f(x_1, x_2, x_3)$ = (12022110). Выполняем основные шаги алгоритма представления мультиоперации в ключевой стандартной форме [4] по переменной x_1 до срабатывания леммы:

1) $g_0 = \left(d^1 * (\mu_x f)\right) = (33303303)$;

2) $f_0 = (\mu_x g_0) = (33123312)$;

3) $h_1 = \left(\left(\mu d^1_{1,1}\right) * (\mu_x f)\right) = (32202303)$.

В мультиоперации h_1 есть нулевые элементы, которые можно заменить на элемент 2^{k-1} (для ранга 2 элемент 2^{k-1} равен 3). После произведенных замен получаем мультиоперацию g_1:

4) $g_1 = (32232333)$;

5) $f_1 = (\mu_x g_1) = (12233333)$;

6) $h_2 = \left(\left(\mu d^1_{2,2}\right) * (\mu_x f)\right) = (13303101)$.

В мультиоперации h_2 есть нулевые элементы. Замена аналогична первому случаю, т. е. их можно заменить на элемент 2^{k-1}. После произведенных замен получаем мультиоперацию g_2:

7) $g_2 = (13333131)$;

8) $f_2 = (\mu_x g_1) = (33332131)$.

Для большей наглядности представим полученный результат алгоритма по переменной x_1 в виде табл. 1.

На данном этапе работа с переменной x_1 закончена. Дальше алгоритм производит такие же шаги с переменными x_2 и x_3. Далее в табл. 2 представлен результат работы алгоритма с переменной x_2, а в табл. 3 представлен результат работы алгоритма с переменной и x_3.

Далее из табл. 3 убираем мультиоперации вида d^2, т. е. f = (33333333), и повторяющиеся мультиоперации. В итоге мы получаем минимальное представление мультиоперации в ключевой стандартной форме. Сложность нашей мультиоперации f равна 7 и ее представление имеет следующий вид:

$$f(x_1x_2x_3) = d^2_{3,1}(x_1x_2) \cap d^2_{4,2}(x_1x_2) \cap$$
$$\cap d^3_{1,1}(x_1x_2x_3) \cap d^3_{2,2}(x_1x_2x_3) \cap d^3_{3,2}(x_1x_2x_3) \cap$$
$$\cap d^2_{3,1}(x_1x_3) \cap d^3_{4,2}(x_1x_2x_3).$$

Таблица 1

x_1	x_2	x_3	f	f_0	f_1	f_2
1	1	1	1	3	1	3
1	1	2	2	3	2	3
1	2	1	0	1	2	3
1	2	2	2	2	3	3
2	1	1	2	3	3	2
2	1	2	1	3	3	1
2	2	1	1	1	3	3
2	2	2	0	2	3	1

Таблица 2

Мультиоперация	Значение мультиоперации							
f_{00}	3	3	3	3	3	3	3	3
f_{01}	3	3	3	3	3	3	3	3
f_{02}	3	3	1	2	3	3	1	2
f_{10}	3	3	3	3	3	3	3	3
f_{11}	1	2	3	3	3	3	3	3
f_{12}	3	3	2	3	3	3	3	3
f_{20}	3	3	3	3	3	1	3	1
f_{21}	3	3	3	3	2	3	3	3
f_{22}	3	3	3	3	3	3	3	3

Таблица 3

Мультиоперация	Значение мультиоперации							
f_{000}	3	3	3	3	3	3	3	3
f_{001}	3	3	3	3	3	3	3	3
f_{002}	3	3	3	3	3	3	3	3
f_{010}	3	3	3	3	3	3	3	3
f_{011}	3	3	3	3	3	3	3	3
f_{012}	3	3	3	3	3	3	3	3
f_{020}	3	3	1	3	3	3	1	3
f_{021}	3	3	3	2	3	3	3	2
f_{022}	3	3	3	3	3	3	3	3
f_{100}	3	3	3	3	3	3	3	3
f_{101}	3	3	3	3	3	3	3	3
f_{102}	3	3	3	3	3	3	3	3
f_{110}	1	3	3	3	3	3	3	3
f_{111}	3	2	3	3	3	3	3	3
f_{112}	3	3	3	3	3	3	3	3
f_{120}	3	3	2	3	3	3	3	3
f_{121}	3	3	3	3	3	3	3	3
f_{122}	3	3	3	3	3	3	3	3
f_{200}	3	3	3	3	3	3	3	3
f_{201}	3	3	3	3	1	3	1	3
f_{202}	3	3	3	3	3	3	3	3
f_{210}	3	3	3	3	2	3	3	3
f_{211}	3	3	3	3	3	3	3	3
f_{212}	3	3	3	3	3	3	3	3
f_{220}	3	3	3	3	3	3	3	3
f_{221}	3	3	3	3	3	3	3	3
f_{222}	3	3	3	3	3	3	3	3

Результаты минимизации мультиопераций. Для тестирования алгоритма была реализована компьютерная программа и проведены следующие эксперименты.

Проведена минимизация всех мультиопераций в классе ключевых стандартных форм для $n = 2$ и $n = 3$.

Таблица 4

L	K	P, %
1	28	10.51
2	122	47.76
3	92	36.22
4	14	5.51

Таблица 5

L	K	P, %
1	82	0.12
2	1548	2.36
3	9560	14.60
4	22 580	34.45
5	21 560	32.89
6	8648	13.20
7	1472	2.25
8	86	0.13

Таблица 6

L	K	P, %
1	24	9.45
2	124	48.82
3	92	36.22
4	14	5.51

Таблица 7

L	K	P, %
1	78	0.12
2	1765	2.69
3	13 319	20.32
4	28 966	44.2
5	17 144	26.16
6	3724	5.69
7	512	0.78
8	26	0.04

Мультиоперации вида $d^0_{1,0}$ и $d^0_{1,3}$ – специфические для данного алгоритма, и найти их сложность по алгоритму невозможно, но они имеют сложность 1, так как их представление в ключевой стандартной форме очевидно и сложность каждого представления равна 1. Для остальных мультиопераций с помощью алгоритма были найдены все минимальные представления и их сложности.

Полученные данные позволили определить среднюю сложность минимального представления мультиопераций в классе ключевых стандартных форм для $n = 2, 3$ и количественное распределение мультиопераций по сложностям полученных минимальных представлений.

$L_{\text{aver}}(2) = 2.36$, $L_{\text{aver}}(3) = 4.46$, где L_{aver} – средняя сложность минимального представления мультиопераций в классе ключевых стандартных форм.

Распределение полученных минимальных представлений при $n = 2$ отображено в табл. 4, а при $n = 3$ – в табл. 5. Используем следующие обозначения: L – сложность мультиопераций; K – количество мультиопераций, имеющих соответствующую сложность; P – доля от числа мультиопераций, %.

Данные результаты позволяют сравнить минимизацию мультиопераций в классе ключевых стандартных форм с минимизацией в классе стандартных форм [6]. Средние сложности минимального представления мультиопераций в классе стандартных форм для $n = 2, 3$ имеют следующие значения: $L_{\text{aver}}(2) = 2.38$, $L_{\text{aver}}(3) = 4.14$. В табл. 6 и 7 показаны распределения минимальных представлений при $n = 2$ и при $n = 3$ для класса стандартных форм.

При сравнении видно, что для мультиопераций от $n = 2$ минимизация почти совпала, а средние сложности минимального представления почти равны. Для мультиопераций от $n = 3$ минимизация по сложностям совпала, но значения P различаются, причем в классе стандартных форм они лучше. Также видно, что средняя сложность минимального представления в классе стандартных форм при $n = 3$ немного лучше, чем в классе ключевых стандартных форм. Это говорит о том, что минимальное представление мультиопераций в классе стандартных форм лучше, чем в классе ключевых стандартных форм, но так как класс ключевых стандартных форм есть часть класса стандартных форм, то получение таких результатов вполне естественно. Полученные результаты являются оригинальным вкладом в изучение теории мультиопераций.

СПИСОК ЛИТЕРАТУРЫ

1. Пантелеев В. И. Критерий полноты для недоопределенных частичных булевых функций // Вести НГУ. Сер. Математика, механика, информатика. 2009. Т. 9, вып. 3. С. 95–114.

2. Peryazev N. A., Sharankhaev I. K. Galois theory for clones and superclones // Discrete Mathematics and Applications. 2016. Vol. 26, № 4. P. 227–238.

3. Шаранхаев И. К. О методе декомпозиции мультифункций // Тр. IX Междунар. конф. «Дискретные модели в теории управляющих систем». М.: МАКС Пресс, 2015. С. 266–267.

4. Перязев Н. А. Стандартные формы мультиопераций в суперклонах // Изв. Иркут. гос. ун-та. Сер. Математика. 2010. Т. 3, № 4. С. 88–95.

5. Перязев Н. А. Клоны, ко-клоны, гиперклоны и суперклоны // Учен. зап. Казан. гос. ун-та. Сер. Физ.-мат. науки. 2009. Т. 151, кн. 2. С. 120–125.

6. Перязев Н. А., Яковчук И. А. Минимизация мультиопераций в классе стандартных форм // Изв. Иркут. гос. ун-та. Сер. Математика. 2009. Т. 2, № 2. С. 117–125.

S. I. Todikov
Saint Petersburg Electrotechnical University «LETI»

MINIMISATION OF MULTIOPERATIONS IN A CLASS KEY STANDERD FORMS

Addresses the problem of minimizing multi-operations in a class of key standard forms. The work is based on the developed algorithm for minimizing multi-operations for n = 2 and n = 3 in the class of key standard forms. This algorithm is based on the analysis of multi-operations and the subsequent best replacements of all zero elements in order to obtain a minimal representation of multi-operations in the class of key standard forms. Using the developed algorithm, all minimal representations of multioperations for n = 2 and n = 3 in the class of key standard forms, the average complexity of the minimal representation of multioperations in the class of key standard forms and the quantitative distribution of multi-operations by the complexity of the obtained minimal representations are obtained. The obtained results of minimizing multi-operations in the class of key standard forms are compared with the minimization of multi-operations in the class of standard forms. When comparing the results, it is concluded that the minimal representation of multi-operations in the class of standard forms is better than in the class of key standard forms.

Multiprocess, superclones, key standard form, algorithm, minimization

УДК 004.056.53

Е. В. Шкляр, Е. Г. Воробьев, М. Ф. Савельев
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Распознавание клавиатурного почерка в браузере

Рассматривается тема распознавания клавиатурного почерка на современных мобильных и стационарных устройствах с установленными веб-браузерами. Выбор темы обусловлен массовым распространением Интернета в России и Мире – согласно исследованиям, к пользователям Интернета так или иначе относится половина населения планеты. Тем не менее, такой тип аутентификации практически не описан в существующих государственных стандартах РФ и мало исследован. В работе рассмотрены возможные методы решения задачи, возможные проблемы и ошибки, а также описан подход к разработке прототипа такой системы. Также приведен процесс разработки и схема работы прототипа системы распознавания клавиатурного почерка, работающая на устройствах любого типа. В работе предлагается инновационный метод аутентификации, не требующий от пользователя для входа в систему ничего, кроме ввода случайного текста. При успешной реализации этого метода станет возможной аутентификация на сайтах без логина и пароля.

Клавиатурный почерк, браузер, биометрия, аутентификация пользователя, мобильные устройства

Клавиатурный почерк – подвид поведенческой подгруппы аутентификации по неотчуждаемым признакам. Это довольно небольшая область, по всем аспектам и ответвлениям которой написано всего несколько десятков статей. В англоязычных источниках клавиатурный почерк называется «keystroke dynamics» [1]. Если переводить дословно, то это «динамика нажатия клавиш на клавиатуре».

В настоящей статье использована классификация методов аутентификации, предложенная исследователями С. Лью и М. А. Сильверман [2]. Они выделяют три типа аутентификации – по знанию, по владению и по неотчуждаемым признакам. К первому типу относятся пароли и PIN-коды, ко второму – смарт-карты, ключи и USB-токены, а третий признак охватывает широкую и популярную сейчас область биометрической аутентификации.

В наше время массово распространены и доступны мобильные устройства с дактилоскопическими датчиками или датчиками сетчатки глаза. Клавиатурный почерк представляет собой очень