УДК 681.325.53

А.Г. ЛУКАШЕНКО, Р.Е. ЮПИН, Д.А. ЛУКАШЕНКО
Черкасский государственный технологический университет

# АЛГОРИТМЫ РАСЧЕТА КОРРЕКТИРУЮЩИХ КОНСТАНТ ПРИ ТАБЛИЧНО-АЛГОРИТМИЧЕСКИХ МЕТОДАХ АППАРАТУРНОЙ РЕАЛИЗАЦИИ ФУНКЦИОНАЛЬНЫХ ЗАВИСИМОСТЕЙ

*Предложены алгоритмы расчета корректирующих констант для таблично-аддитивного и таблично-логического методов аппаратурной реализации функциональных зависимостей, позволяющих на этапе проектирования сократить время подготовки базы данных при заполнении числового блока памяти. Преимуществом алгоритмов является отсутствие необходимости разработки специальной программы. Это уменьшает стоимость микропроцессорных систем и расширяет рынок сбыта.*

*The algorithms calculate the correction constants for table–additive and table–logical methods of hardware implementation of functional dependencies, allowing, at the design stage to reduce the time of preparation of the database at filling of a numerical memory unit. The advantage of the algorithms is the lack of need for special software. This reduces the cost of microprocessor systems and expanding market.*

Ключевые слова: алгоритмы, база данных, числовой блок памяти, корректирующая константа.

Постоянно растущая сложность задач, решаемых вычислительными устройствами в лазерных технологических комплексах (ЛТК) для управления траекторией лазерного луча, повышает требования к:

- точности;
- малому времени задержки сигнала при обработке информации;
- достоверности результатов решения;
- простоте программирования;
- удобству обслуживания;
- ограничению на габаритные размеры, вес;
- низкой стоимость устройств и к другим параметрам, тесным образом связанных с проблемой для повышения эффективности работы современных микропроцессорных структур ЛТК.

Этим требованиям отвечает аппаратурная реализация заданной функциональной зависимости, построенная классическим табличным методом [1–4, 7].

Однако объем таблиц растет с увеличением точности вычисления, например, при погрешности равной или меньшей чем $\pm\delta = 0{,}5 \cdot 2^{-32}$ составит порядка $32 \cdot 2^{32}$. Это приводит к низкому проценту выхода годных кристаллов с пластины и, как следствие, к высокой стоимости СБИС.

Изыскание методов аппаратурной реализации функциональных зависимостей привели к таблично-алгоритмическим методам (ТАМ), которые включают таблично-аддитивный и таблично-логический методы.

Суть ТАМ заключается в представлении *n*-разрядного аргумента в виде $m$ кортежей $a_i$, по $r$ разрядам в каждом. Воспроизведение *n*-разрядного кода значений функции $f(x)$ выполняется корректировкой кортежа $a_i$, в соответствующий кортеж кода функции $b_i$ с помощью таблицы соответствующих корректирующих констант $\Delta_I$. Причем для таблично-аддитивного метода корректирующие константы имеют соответствующий знак ($\pm\Delta$) [1].

Таблично-логический метод отличается от предыдущего тем, что корректирующая константа не имеет отрицательного знака [3, 6–7].

Пусть $n$-значимое число воспроизводимой функции $f_i(x)$ изображается, как кортеж переменных

$$b_{m-1}, \ldots, b_i, \ldots, b_2, b_1, b_0, \tag{1}$$

а значения входной независимой переменной $(x)$ как кортеж переменных

$$a_{m-1}, \ldots, a_i, \ldots, a_2, a_1, a_0, \tag{2}$$

причем каждая переменная $a_i$, $b_i$, в свою очередь, изображается $r$-значимым числом, тогда вычисление значения соответствующей корректирующей константы $\Delta_i$ определяется – как сумма по $\mathrm{mod}\,2$ соответствующих кортежей зависимой переменной и независимой переменной:

$$\begin{aligned}
&(b_0 \oplus a_0) = \Delta_0;\\
&(b_1 \oplus a_1) = \Delta_1;\\
&(b_2 \oplus a_2) = \Delta_2;\\
&\vdots\\
&(b_i \oplus a_i) = \Delta_i;\\
&\vdots\\
&(b_m \oplus a_m) = \Delta_m,
\end{aligned} \tag{3}$$

где зависимая переменная имеет вид:

$$b_i = f_i S_i ... h_i = f_i g^{(n-1)-ri} + S_i g^{(n-2)-ri} + ... + h_i g^{(n-r)-ri}, \quad (4)$$

а независимая переменная –

$$a_i = c_i d_i ... l_i = c_i g^{(n-1)-ri} + d_i g^{(n-2)-ri} + ... + l_i g^{(n-r)-ri}. \quad (5)$$

Тогда корректирующая константа записывается следующим образом:

$$\Delta_i = (f_i \oplus c_i) g^{(n-1)-ri} + (S_i \oplus d_i) g^{(n-2)-ri} + ... + (h_i \oplus l_i) g^{(n-r)-ri} = F_i N_i Z_i. \quad (6)$$

При этом соответствующее значение функции примет вид:

$$f_i(x) = f_i g^{(n-1)-ri} + S_i g^{(n-2)-ri} + ... + h_i g^{(n-r)-ri} =$$
$$= (F_i \oplus c_i) g^{(n-1)-ri} + (N_i \oplus d_i) g^{(n-2)-ri} + ... + (Z_i \oplus l_i) g^{(n-r)-ri}, \quad (7)$$

где $F_i = f_i \oplus c_i$;

$N_i = S_i \oplus d_i$;

$Z_i = h_i \oplus l_i$.

Таким образом, формирование кода значения функции можно записать следующим образом:

$$f(x) = f_i S_i ... h_i = \{c_i d_i ... l_i\} \oplus \{F_i N_i ... Z_i\}. \quad (8)$$

Основными общими действиями методики воспроизведения значения функции в этих методах на этапе проектирования является:

- предварительное представление в виде кортежей кодовых последовательностей аргумента и соответствующих функциональной зависимости цифровых кодов заданной функции;
- определение значений корректирующих констант для каждой пары кортежей соответствующих массивам заданных последовательностей аргумента и функции.

Воспроизведение функции осуществляется с помощью соответствующей корректирующей константы. Поэтому разработка алгоритмов для расчета значений корректирующих констант на этапе проектирования функционального цифрового преобразователя является задачей первоочередной и особенно актуальной для специализированных сложных функций.

Пример [5] обобщенной цифровой модели аппаратурной реализации заданной функциональной зависимости $f(x)$ и построенной на базе таблично-логического метода аппаратурной реализации представлен на рис. 1.

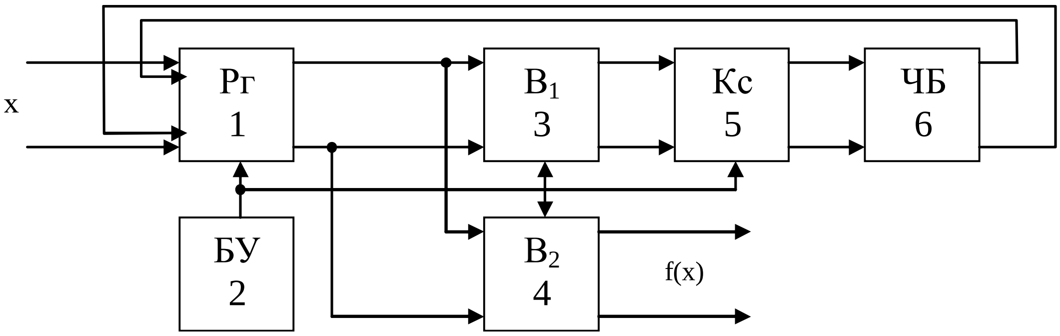


**Рис. 1. Образно-знаковая модель преобразователя реализованного на базе таблично-логического метода для *m*=1: 1 – входной регистр; 2 – блок управления; 3, 4 – блоки вентилей; 5 – комбинационная схема адреса; 6 – числовой блок памяти**

Особенностью модели является наличие обратной связи, по которой единицы с выхода числового блока памяти преобразуют кодовую последовательность аргумента в соответствующую кодовую последовательность значения *f(x)*, без длительных операций и выходного регистра [7].

Алгоритм расчета корректирующих констант строится на соотношениях между известными значениями входных кодов аргумента и соответствующими значениями выходных кодов специализированной функции. С целью сокращения объема памяти создается база данных соответствующих значений корректирующих констант как разность между ними.

Для таблично-логического метода аппаратурной реализации эта разница определяется как сумма по $\mod 2$. Алгоритм расчета корректирующих констант для этого метода имеет следующий вид (рис. 2) и содержит следующие операции:

1. Начало.
2. Ввод данных

$$\left(X_{lz}, Y_{lz}, \textit{при } l = \overline{1, 2^r}, \ z = \overline{1, n}\right).$$

3. Вычисление количества кортежей

$$(m := n / r).$$

4. Начало цикла по строкам

$$\left(j := 1, \ldots, 2^r\right);$$

5. Начало цикла по кортежам текущей строки

$$\left(i := 0, \overline{m-1}\right).$$

6. Вычисление значения корректирующей константы для данного кортежа

$$\left(\Delta_{i\,j} := X_{i\,j} \oplus Y_{i\,j}\right).$$

7. Запись значения корректирующей константы в память

$$\left(\Delta_{i\,j}\right).$$

8. Если есть в строке еще кортежи перейти к следующему кортежу и выполнить шаг 6, иначе перейти к шагу 9.

9. Если есть еще строки перейти к следующей строке и выполнить шаг 5, иначе перейти к шагу 10.

10. Конец.

Из алгоритма (рис. 2) и перечня операций для таблично-логического метода аппаратурной реализации видно, что использование операции суммирования по $\mathrm{mod}\,2$ для соответствующих кортежей $a_i$, $b_i$ при получении значений корректирующих констант $\Delta_l$, не требует разработки специальной программы и этим значительно сокращается время на этапе проектирования.

Для таблично-аддитивного метода аппаратурной реализации разница между $a_i$, $b_i$ определяется как арифметическая сумма. Этот метод целесообразен при необходимости иметь сумматор в вычислительном устройстве.

Алгоритм расчета корректирующих констант для таблично-аддитивного метода аппаратурной реализации представлен на рис. 3 и содержит такие основные операции:

1. Начало.
2. Ввод данных

$$\left(X_{l\,z},\, Y_{l\,z},\, \textit{при}\ \ l = \overline{1,\, 2^r},\, z = \overline{1,\, n}\right).$$

3. Вычисление количества кортежей

$$\left(m := n / r\right).$$

4. Начало цикла по строкам

$$\left(j := \overline{1,\, 2^r}\right).$$

5. Начало цикла по кортежам текущей строки $\left(i := 0, \overline{m-1}\right)$.

6. Обнуление значения переноса $\left(c := 0\right)$.

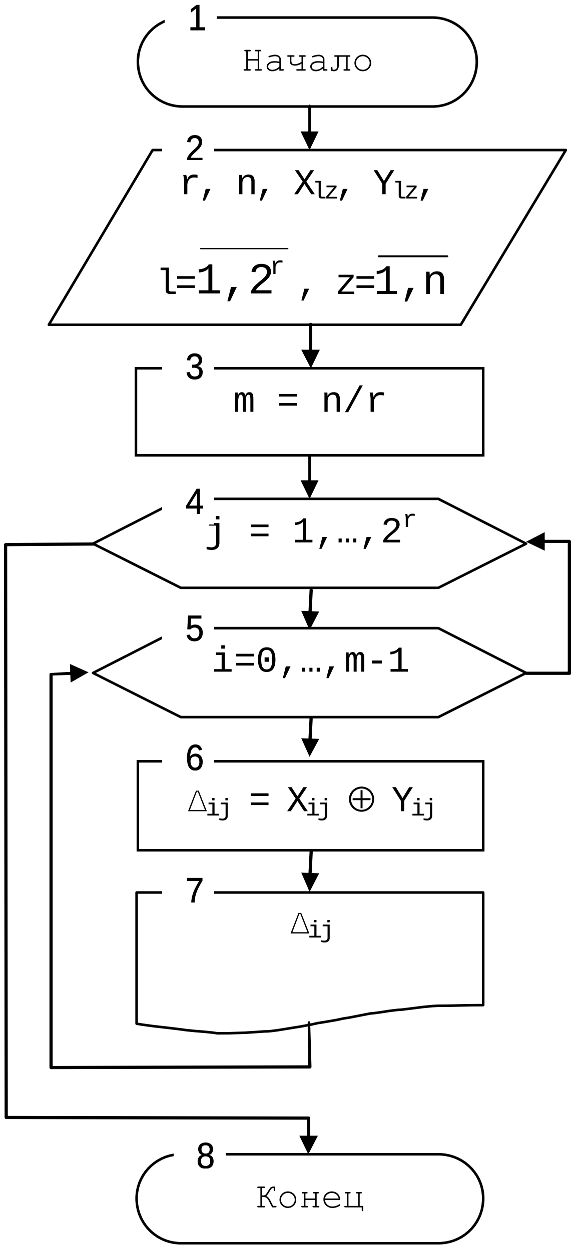


**Рис. 2. Алгоритм расчета кортежей корректирующей константы для таблично-логического метода аппаратурной реализации**

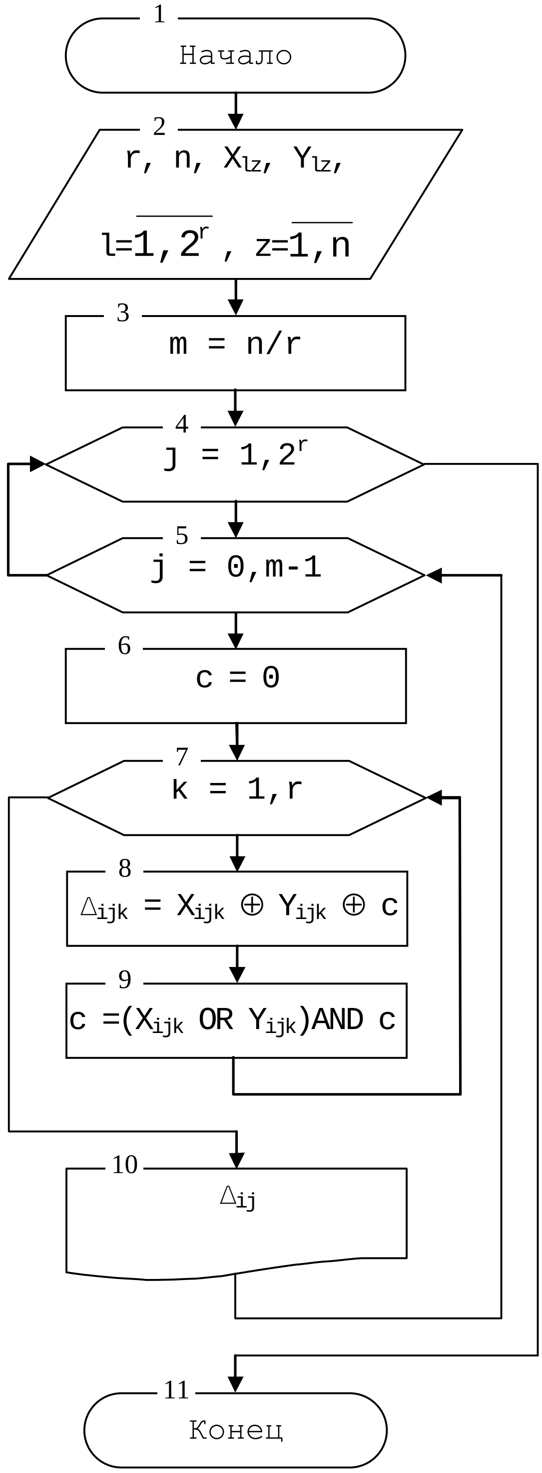


**Рис. 3. Алгоритм расчета кортежей корректирующей константы для таблично-аддитивного метода аппаратурной реализации**

1. Начало цикла по разрядам текущего кортежа $(k := 1, r)$.

2. Вычисление значения корректирующей константы для данного кортежа

$$\left(\Delta_{i\ j\ k} := X_{i\ j\ k} \oplus Y_{i\ j\ k} \oplus c\right).$$

3. Вычисление переноса в старший разряд

$$\left(c := (X_{i\ j\ k}\ OR\ Y_{i\ j\ k})\ AND\ c\right).$$

4. Если есть в кортеже еще разряды, то перейти к следующему разряду и выполнить шаг 8, иначе перейти к шагу 11.

5. Запись значения корректирующей константы в память $\left(\Delta_{i\ j}\right)$.

6. Если есть в строке еще кортежи перейти к следующему кортежу и выполнить шаг 6, иначе перейти к шагу 13.

7. Если есть еще строки перейти к следующей строке и выполнить шаг 5, иначе перейти к шагу 14.