УДК 681.325

В.П. Супрун, Д.А. Городецкий

МЕТОД БЛОЧНО-СТРУКТУРНОГО СИНТЕЗА ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ МОДУЛЯРНОЙ АРИФМЕТИКИ

Рассматривается задача логического проектирования вычислительных устройств модулярной арифметики. Предлагается метод блочно-структурного синтеза, применение которого позволяет получать логические схемы устройств, превосходящие существующие аналоги по сложности (числу входов логических элементов), по быстродействию и (или) числу внешних выводов. Использование метода приводится для случая представления входных и выходных данных в унитарных кодах.

Введение

Известно, что использование остаточных классов [1–3] и модулярных систем счисления [4] позволяет повысить производительность вычислительных структур за счет организации параллельной и независимой обработки малоразрядных остатков от деления чисел на выбранные натуральные модули – основания модулярных систем счисления. Кроме того, модулярные коды позволяют обнаруживать и исправлять ошибки как при хранении и передаче числовой информации, так и при выполнении (вычислении) арифметических операций [1, 2].

Очевидно, что повысить производительность вычислительных систем можно путем создания относительно простых вычислительных устройств $\Re$, предназначенных для выполнения (вычисления) арифметических операций по модулю P, – устройств модулярной арифметики.

Задача синтеза логических схем $S(\Re)$ вычислительных устройств $\Re$ является весьма сложной, а эффективность методов ее решения определяется, в частности, такими параметрами, как конструктивная сложность (число элементов или число входов логических элементов) $l(\Re)$, глубина (максимальное число последовательно соединенных логических элементов от входа к выходу) $g(\Re)$ и число внешних выводов (суммарное число входов и выходов) $m(\Re)$.

Одним из подходов к эффективному решению этой задачи является разработка вычислительных устройств $\Re_1, \Re_2, \ldots, \Re_m$, реализующих элементарные арифметические операции, в качестве строительных блоков для проектирования вычислительных устройств $\Re$, предназначенных для реализации более сложных арифметических операций.

В настоящей статье предлагается метод блочно-структурного синтеза логических схем вычислительных устройств $\Re$, в основе которого лежит использование «блоков» $\Re_1, \Re_2, \ldots, \Re_m$. Применение метода демонстрируется на примере проектирования устройств модулярной арифметики при условии представления входных и выходных операндов в унитарных кодах по модулю три.

1. Вычислительные устройства модулярной арифметики

Устройства модулярной арифметики относятся к области вычислительной техники и микроэлектроники и могут использоваться для построения систем передачи и обработки дискретной информации, средств аппаратурного контроля и цифровых устройств, работающих в системе остаточных классов.

Вычислительные устройства $\Re_1, \Re_2, \ldots, \Re_m$ модулярной арифметики по модулю P ориентированы прежде всего на вычисление элементарных арифметических операций $A + B = S \pmod P$ и $A * B = R \pmod P$. Такие устройства могут использоваться в качестве са-

мостоятельных функциональных узлов, а также служить основой для построения других (более сложных) вычислительных устройств модулярной арифметики.

Как известно [3], при проектировании модулярных параллельных вычислительных структур применяется способ кодирования информации посредством двоичных унитарных кодов. Если для некоторого операнда A имеет место $A = i \pmod P$, то $i \in \{0, 1, \ldots, P-1\}$. При кодировании A в унитарных кодах по модулю P используется P разрядов $(a_0, a_1, \ldots, a_{P-1})$. Здесь $a_i = 1$ тогда и только тогда, когда $A = i \pmod P$, где $i = 0, 1, \ldots, P-1$.

Унитарные коды применяются, например, в цифровых устройствах, работающих в системе остаточных классов, для построения систем аппаратного контроля, а также в вычислительных устройствах, реализующих алгоритмы модулярной арифметики [3].

Отметим, что преобразование P-разрядного унитарного кода в $]\log_2 P[$-разрядный позиционный код и обратно осуществляется с помощью шифратора и дешифратора соответственно.

2. Определение формулы *W* (*A*, *B*, *C*, …)

В настоящей статье рассматриваются только две элементарные арифметические операции: сложение $A + B = S \pmod P$ и умножение $A * B = R \pmod P$ в унитарных кодах. Первую операцию реализует устройство $\Re_1$ (модулярный сумматор), а вторую – устройство $\Re_2$ (модулярный умножитель). Сложная арифметическая операция F задается формулой $W(A, B, C, \ldots)$, для которой дается следующее рекурсивное определение:

1. Выражения $A + B = S \pmod P$ и $A * B = R \pmod P$ являются (простейшими) формулами.

2. Если W_1 и W_2 есть формулы, то новые образования $(W_1 + W_2)$ и $(W_1 * W_2)$ также являются формулами.

3. Других формул нет.

С учетом того что операция умножения «сильнее» операции сложения, в записи формул некоторые пары скобок опускаются. Кроме того, опускаются внешние скобки.

Например, выражения $A * B + C = F \pmod P$, $A * (B + C) = F \pmod P$, $A * B + C * D = F \pmod P$ и $(A + B) * (C + D) = F \pmod P$ являются формулами.

3. Метод блочно-структурного синтеза

Общая идея метода блочно-структурного синтеза логических схем $S(\Re)$ вычислительных устройств модулярной арифметики $\Re$ по модулю P состоит в следующем.

Пусть требуется синтезировать логическую схему $S(\Re)$ вычислительного устройства $\Re$ по модулю P, которое предназначено для реализации арифметической операции, заданной посредством формулы $W(A, B, C, \ldots)$.

В соответствии с формулой $W(A, B, C, \ldots)$ первоначально необходимо построить структуру устройства $\Re$, состоящую из «блоков» двух типов – сумматора $\Re_1$ и умножителя $\Re_2$. Затем необходимо подобрать логические схемы «блоков» $\Re_1$ и $\Re_2$ такие, что после их подстановки в структуру устройства $\Re$ можно было бы «на стыках» объединить одноименные логические элементы И, ИЛИ и СЛОЖЕНИЕ ПО МОДУЛЮ ДВА, уменьшая тем самым сложность и (или) глубину логической схемы $S(\Re)$ вычислительного устройства $\Re$. Кроме того, если на этом этапе использовать логические схемы «блоков» $\Re_1$ и $\Re_2$ с меньшим чис-

лом внешних выводов, то появляется возможность удалить из логической схемы $S(\Re)$ часть логических элементов.

Естественно, что для повышения эффективности применения метода блочно-структурного синтеза требуется иметь несколько логических схем «блоков» $\Re_1$ и $\Re_2$, отличающихся друг от друга типом использованных логических элементов, сложностью $l(\Re_j)$, глубиной $g(\Re_j)$ и (или) числом внешних выводов $m(\Re_j)$, где $j = 1, 2$.

Применение метода блочно-структурного синтеза проиллюстрируем на примере синтеза нескольких логических схем вычислительного устройства $\Re$, ориентированного на вычисление операции $A * B + C * D = F$ в унитарных кодах по модулю три.

4. Логические схемы модулярного сумматора $\Re_1$ и модулярного умножителя $\Re_2$

Модулярный сумматор $\Re_1$ и модулярный умножитель $\Re_2$ выполняют операции $A + B = S$ и $A * B = R$ по модулю три при условии, что операнды A и B, а также результаты вычислений S и R представлены в унитарных кодах, т. е. $A = (a_0, a_1, a_2)$, $B = (b_0, b_1, b_2)$, $S = (s_0, s_1, s_2)$ и $R = (r_0, r_1, r_2)$, где $a_k = 1$, $b_k = 1$, $s_k = 1$ и $r_k = 1$ тогда и только тогда, когда $A = k \pmod 3$, $B = k \pmod 3$, $S = k \pmod 3$ и $R = k \pmod 3$, где $k = 0, 1, 2$.

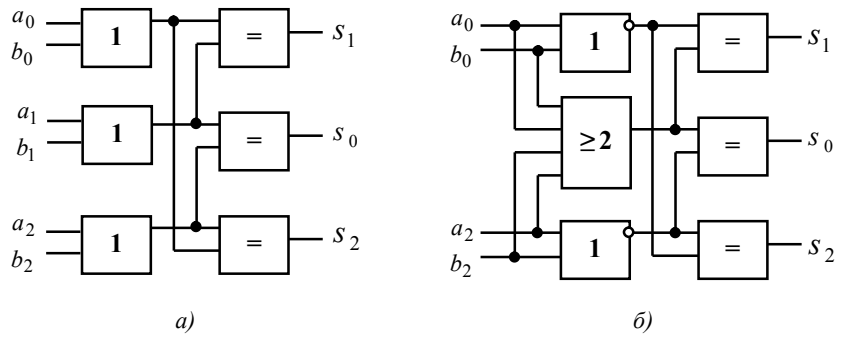
Ниже представлена таблица истинности логических функций $s_0, s_1, s_2, r_0, r_1, r_2$, которые реализуются на выходах устройств $\Re_1$, $\Re_2$ и зависят от переменных $a_0, a_1, a_2, b_0, b_1, b_2$.

Таблица истинности логических функций, реализуемых на выходах устройств $\Re_1$ и $\Re_2$

Входы						Выходы					
Унитарный код первого операнда $A = (a_0, a_1, a_2)$			Унитарный код второго операнда $B = (b_0, b_1, b_2)$			Унитарный код результата сложения $S = (s_0, s_1, s_2)$			Унитарный код результата умножения $R = (r_0, r_1, r_2)$		
a_0	a_1	a_2	b_0	b_1	b_2	s_0	s_1	s_2	r_0	r_1	r_2
1	0	0	1	0	0	1	0	0	1	0	0
1	0	0	0	1	0	0	1	0	1	0	0
1	0	0	0	0	1	0	0	1	1	0	0
0	1	0	1	0	0	0	1	0	1	0	0
0	1	0	0	1	0	0	0	1	0	1	0
0	1	0	0	0	1	1	0	0	0	0	1
0	0	1	1	0	0	0	0	1	1	0	0
0	0	1	0	1	0	1	0	0	0	0	1
0	0	1	0	0	1	0	1	0	0	1	0

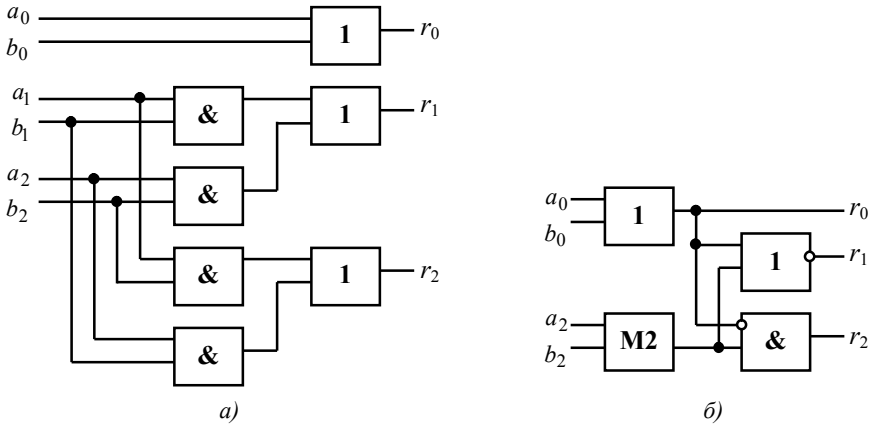
Рассмотрим две логические схемы $S_1(\Re_1)$, $S_2(\Re_1)$ модулярного сумматора $\Re_1$ (рис. 1) и две логические схемы $S_1(\Re_2)$, $S_2(\Re_2)$ модулярного умножителя $\Re_2$ (рис. 2). Логическая схема $S_1(\Re_1)$ [5] имеет наименьшую конструктивную сложность среди известных аналогов, а логическая схема $S_2(\Re_1)$ – минимально возможное число внешних выводов. Логическая схема $S_1(\Re_2)$ синтезирована на основе использования ДНФ логических функций r_0, r_1, r_2, а логическая схема $S_2(\Re_2)$ [6] имеет минимально возможное число внешних выводов и является наиболее простой (по числу входов логических элементов) среди известных аналогов.

Следует отметить, что существует довольно-таки много логических схем модулярного сумматора $\Re_1$ и модулярного умножителя $\Re_2$, отличающихся друг от друга типом используемых элементов, числом внешних выводов, конструктивной сложностью и/или числом уровней.



а) *б)*

Рис. 1. Логические схемы модулярного сумматора $\mathfrak{R}_1$: *а)* схема $S_1(\mathfrak{R}_1)$; *б)* схема $S_2(\mathfrak{R}_1)$



а) *б)*

Рис. 2. Логические схемы модулярного умножителя $\mathfrak{R}_2$: *а)* схема $S_1(\mathfrak{R}_2)$; *б)* схема $S_2(\mathfrak{R}_2)$

5. Примеры применения метода блочно-структурного синтеза

В качестве примера применения метода блочно-структурного синтеза рассмотрим задачу построения логических схем вычислительного устройства $\mathfrak{R}$, предназначенного для вычисления (реализации) арифметической операции $A*B+C*D=F$ в унитарных кодах по модулю три.

С помощью данного метода удалось синтезировать три логические схемы вычислительного устройства $\mathfrak{R}$: $S_1(\mathfrak{R})$, $S_2(\mathfrak{R})$ и $S_3(\mathfrak{R})$. Рассмотрим подробнее применение метода.

В соответствии с методом блочно-структурного синтеза первоначально определим структуру проектируемого устройства $\mathfrak{R}$. Очевидно, что структура $\mathfrak{R}$ содержит один «блок» модулярного сумматора $\mathfrak{R}_1$ и два «блока» модулярного умножителя $\mathfrak{R}_2$, выходы которого соединены с входами первого «блока» $\mathfrak{R}_1$.

Для построения логической схемы $S_1(\mathfrak{R})$ заменим «блоки» $\mathfrak{R}_1$ и $\mathfrak{R}_2$ на логические схемы $S_1(\mathfrak{R}_1)$ и $S_1(\mathfrak{R}_2)$ (см. рис. 1, *а* и 2, *а*). В результате такой замены получим некоторую логическую схему устройства $\mathfrak{R}$, которую обозначим через $S_1^*(\mathfrak{R})$. Так как второй и третий уровни логической схемы $S_1^*(\mathfrak{R})$ составляют логические элементы ИЛИ, то после их объединения (за счет увеличения числа входов) получаем логическую схему $S_1(\mathfrak{R})$, приведенную на рис. 3, *а* [7].

Для построения второй логической схемы $S_2(\mathfrak{R})$ заменим «блоки» $\mathfrak{R}_1$ и $\mathfrak{R}_2$ на логические схемы $S_2(\mathfrak{R}_1)$ и $S_2(\mathfrak{R}_2)$ (см. рис. 1, *б* и 2, *б*). В результате такой замены получим некоторую промежуточную логическую схему устройства $\mathfrak{R}$, которую обозначим $S_2^*(\mathfrak{R})$.

Так как в логической схеме $S_2(\mathfrak{R}_1)$ отсутствуют входы, на которые подаются значения a_1 и b_1, то из логической схемы $S_2^*(\mathfrak{R})$ можно удалить два элемента ИЛИ-НЕ, на выходах которых в логической схеме $S_2(\mathfrak{R}_2)$ реализуется логическая функция r_1. Полученная в результате таких преобразований логическая схема $S_2(\mathfrak{R})$ показана на рис. 3, *б*.

Для построения логической схемы $S_3(\mathfrak{R})$ (рис. 3, *в*) необходимо воспользоваться логическими схемами $S_2(\mathfrak{R}_1)$ и $S_1(\mathfrak{R}_2)$ (см. рис. 1, *б* и 2, *а*).

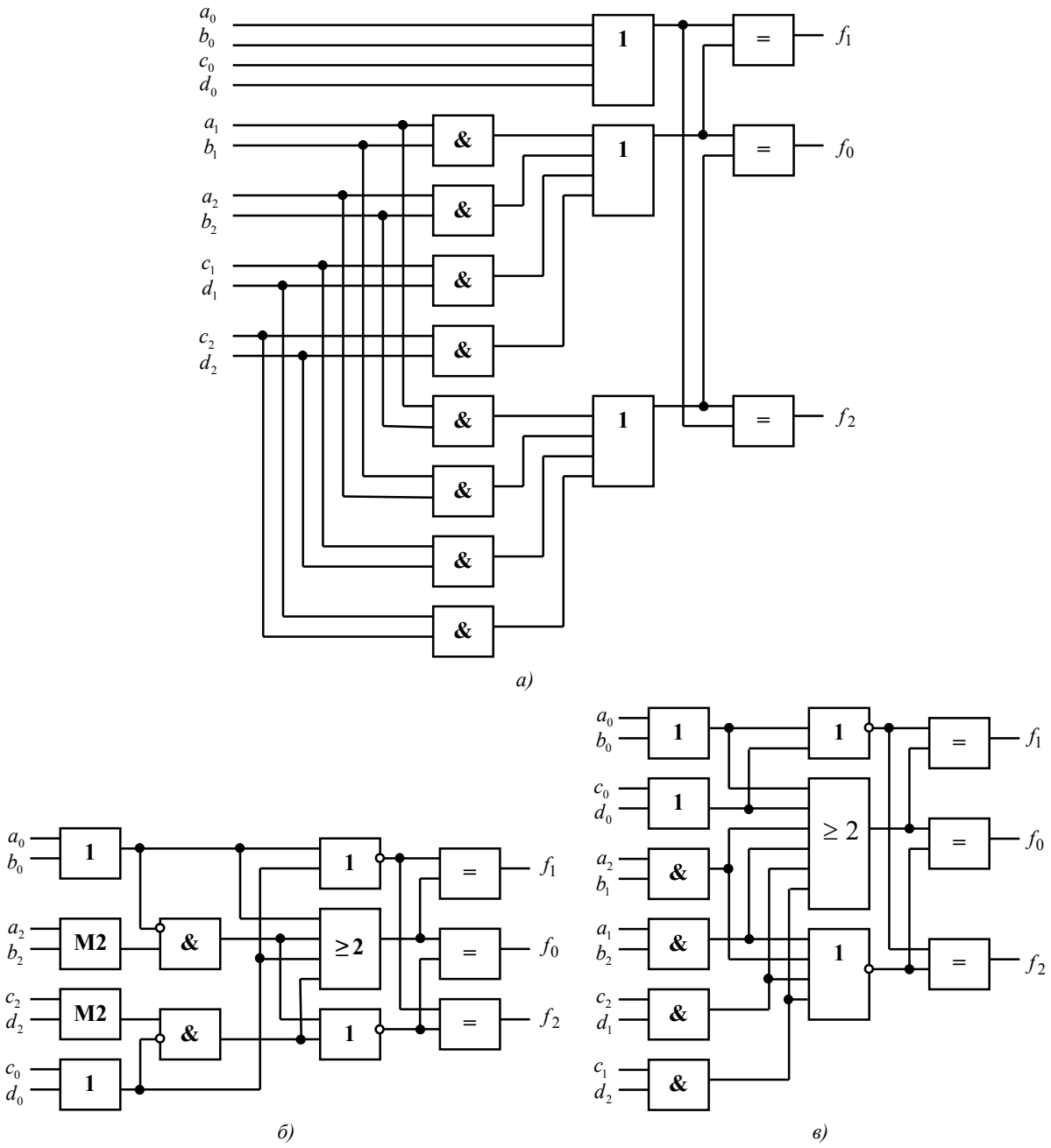


Рис. 3. Логические схемы: *а)* $S_1(\mathfrak{R})$; *б)* $S_2(\mathfrak{R})$; *в)* $S_3(\mathfrak{R})$

Логическая схема $S_1(\mathfrak{R})$ имеет конструктивную сложность (по числу входов логических элементов) $l(S_1)=34$, глубину $g(S_1)=3$ и число внешних выводов $m(S_1)=15$. Логические схе-

мы $S_2(\Re)$ и $S_3(\Re)$ имеют следующие характеристики: $l(S_2)=26$, $g(S_2)=4$, $m(S_2)=11$ и $l(S_3)=30$, $g(S_3)=3$, $m(S_3)=15$. Очевидно, что синтезированные логические схемы $S_1(\Re)$, $S_2(\Re)$ и $S_3(\Re)$ являются конкурентоспособными и каждая из них может быть использована при проектировании других (более сложных) вычислительных устройств модулярной арифметики.

Заключение

Для повышения эффективности применения метода блочно-структурного синтеза необходимо иметь несколько логических схем, которые реализуют элементарные арифметические операции сложения и умножения и отличаются друг от друга типом использованных при их синтезе логических элементов, числом уровней и (или) числом внешних выводов.

С помощью данного метода удалось синтезировать эффективные логические схемы вычислительных устройств модулярной арифметики в унитарных (и позиционных) кодах по модулю три и по модулю пять [8]. В частности, синтезированы и запатентованы логические схемы вычислительных устройств модулярной арифметики, реализующих операции $A*B+C=F$ (патент РБ № 9189), $A*B+C*D=F$ (патенты РБ № 9341, 10535), $(A+B)*(C+D)=F$ (патенты РБ № 9477, 10350), $A+B+C+D=F$ (патенты РБ № 9600, 10201). В перечисленных выше патентах РБ на изобретение удалось оптимизировать конструктивную сложность, глубину или число внешних выводов, а иногда – две (или даже три) характеристики логических схем $S(\Re)$ одновременно.

Список литературы

1. Торгашев, В.А. Система остаточных классов и надежность ЦВМ / В.А. Торгашев. – М. : Сов. радио, 1973. – 120 с.

2. Долгов, А.И. Диагностика устройств, функционирующих в системе остаточных классов / А.И. Долгов. – М. : Радио и связь, 1982. – 64 с.

3. Модулярные параллельные вычислительные структуры нейропроцессорных систем / Н.И. Червяков [и др.]. – М. : Физматлит, 2003. – 288 с.

4. Коляда, А.А. Модулярные структуры конвейерной обработки информации / А.А. Коляда, И.Т. Пак. – Минск : Университетское, 1992. – 256 с.

5. Сумматор унитарных кодов по модулю три : пат. 3270 Респ. Беларусь, МПК G 06 F 7/49 / В.П. Супрун ; опубл. 30.03.00 // Афіцыйны бюл. / Нац. цэнтр інтэллектуал. уласнасці. – 2000. – № 1. – С. 187.

6. Устройство для умножения N чисел в унитарных кодах по модулю три : пат. 6586 Респ. Беларусь, МПК G 06 F 7/49 / В.П. Супрун, А.М. Седун ; опубл. 30.12.04 // Афіцыйны бюл. / Нац. цэнтр інтэллектуал. уласнасці. – 2004. – № 4. – С. 194.

7. Вычислительное устройство унитарных кодов по модулю три : пат. 9341 Респ. Беларусь, МПК G 06 F 7/38, 7/48 / В.П. Супрун, Д.А. Городецкий ; опубл. 30.06.07 // Афіцыйны бюл. / Нац. цэнтр інтэллектуал. уласнасці. – 2007. – № 3. – С. 141.

8. Городецкий, Д.А. Вычислительные устройства унитарных кодов по модулю три с минимальным числом внешних выводов / Д.А. Городецкий, А.М. Седун, В.П. Супрун // Материалы 4-й Междунар. науч.-техн. конф. «Проблемы проектирования и производства радиоэлектронных средств». – Новополоцк, 2006. – Т. 2. – С. 34–38.

Поступила 12.11.08

Белорусский государственный университет,
Минск, пр. Независимости, 4
e-mail: suprun@bsu.by

V.P. Suprun, D.A. Gorodetsky

A METHOD OF BLOCK-STRUCTURED SYNTHESIS OF COMPUTING DEVICES IN MODULAR ARITHMETIC

A problem of the logic design of computing devices in modular arithmetic is considered. A method of Block-structured synthesis that outperforms the existing approaches in terms of the complexity (the number of logic elements inputs), performance and (or) external outputs, is proposed. The method is used for presentation of input and output data in unitary codes.