

# Синтез преобразователей кодов, предназначенных для сокращения длины двоичных кодируемых слов

П.Н. Бибилло

Объединенный институт проблем информатики НАН Беларуси, bibilo@newman.bas-net.by

**Аннотация**— Рассматривается задача синтеза комбинационных схем кодовых преобразователей, предназначенных для сокращения длины слов из заданного набора кодируемых двоичных слов. Кодирование предполагает, что различные двоичные слова (наборы бит) будут закодированы различными двоичными кодами меньшей длины (разрядности). Предлагаемые способы решения данной задачи основаны на составлении и логической минимизации таких форм систем не полностью определенных булевых функций как дизъюнктивные нормальные формы (ДНФ) и бинарные диаграммы решений, называемые BDD-представлениями (BDD - Binary Decision Diagram). Минимизация составляемых функциональных описаний ориентирована на уменьшение аппаратной сложности комбинационных схем в базе библиотечных элементов либо программируемых элементов FPGA, реализующих преобразователи кода рассматриваемого класса.

**Ключевые слова**— преобразователь кода, система булевых функций, дизъюнктивная нормальная форма (ДНФ), разложение Шеннона, Binary Decision Diagram (BDD), синтез логической схемы, VHDL, СБИС.

## I. ВВЕДЕНИЕ

В цифровых вычислительных и управляющих системах широкое распространение получили преобразователи кодов - устройства, предназначенные для преобразования двоичного кода из одной формы в другую. Для представления информации используют разнообразные двоичные и двоично-десятичные коды чисел, прямой, обратный, дополнительный и их модификации [1], унитарные коды, двоичные коды чисел по модулю [2] и т.д. Проектирование таких преобразователей кодов опирается на известные формы задания входной и выходной информации, т. е. кодируемых двоичных слов и кодов (кодирующих слов). Однако при проектировании цифровых систем могут возникать проблемы передачи по шинам данных длинных двоичных слов, т. е. таких, разрядность которых превышает разрядность шины данных. Например, по 16-разрядной шине данных требуется передавать 18-разрядные или 17-разрядные слова. Конечно, каждое такое слово можно передать за два такта функционирования системы, однако такой подход снижает общее быстродействие всей системы. Одним из подходов для решения таких проблем является разработка комбинационных схем, которые

осуществляют преобразование длинных двоичных кодируемых слов в более короткие. При получении информации, переданной по шине данных, может потребоваться обратный преобразователь, осуществляющий преобразование переданного слова в исходную форму.

В данной работе рассматривается задача логического проектирования кодовых преобразователей, предназначенных для сокращения длины каждого слова из заданного набора кодируемых двоичных слов. Кодирование предполагает, что различные двоичные слова (наборы бит) одинаковой длины будут закодированы различными двоичными кодами меньшей длины (разрядности). Естественно, коды также должны иметь одинаковую длину. Предлагаемые способы решения задачи основаны на составлении и логической минимизации различных форм функционального описания проектируемых преобразователей кода, в качестве таких форм выступают дизъюнктивные нормальные формы (ДНФ) [3] и бинарные диаграммы решений, называемые BDD-представлениями [4]. Минимизация функциональных описаний ориентирована на уменьшение аппаратной сложности преобразователей кода и существенным образом использует такие модели функционирования проектируемых цифровых устройств, как не полностью определенные двоичные (булевы) функции и многозначные функции, зависящие от булевых переменных. Уменьшение сложности одноконтурных комбинационных схем, реализующих функции преобразователей кодов, и является основной целью логического проектирования преобразователей кодов рассматриваемого класса.

## II. ОПРЕДЕЛЕНИЯ И ПОСТАНОВКИ ЗАДАЧ

Булевыми называются двоичные (0, 1) функции  $f(x) = f(x_1, x_2, \dots, x_n)$  двоичных (булевых) переменных  $x_1, x_2, \dots, x_n$ . Пусть  $V^x$  – булево пространство, построенное над переменными булева вектора  $x = (x_1, x_2, \dots, x_n)$ . Элементами этого пространства являются  $n$ -компонентные наборы (векторы) нулей и единиц. Булева функция, значения 0, 1 которой определены на всех элементах  $V^x$ , называется *полностью определенной*. Если же на некоторых элементах булева пространства  $V^x$  значения функции

не определены, то такая функция называется не полностью определенной, или *частичной*. Частичная булева функция принимает единичное значение на элементах  $x^*$  подмножества  $M_f^1$  булева пространства  $V^x$  и нулевое значение на элементах подмножества  $M_f^0$ . На всех остальных элементах пространства  $V^x$ , образующих подмножество  $M_f^-$  пространства  $V^x$ , значения частичной функции не определены, подмножество  $M_f^-$  будем называть также областью неопределенных значений частичной функции. Неопределённое значение функции обозначается символом « $\leftarrow$ ». Частичная функция  $f_i$  реализуется частичной (либо полностью определенной) функцией  $f_j$ , если и только если  $M_{f_i}^1 \subseteq M_{f_j}^1$ ,  $M_{f_i}^0 \subseteq M_{f_j}^0$ . Функцию  $f_j$  называют *доопределением* функции  $f_i$ . Для пары полностью определенных булевых функций  $f_i, f_j$  отношение реализации является отношением равенства. Под *векторной* булевой функцией  $f(x)$  будем понимать упорядоченную систему частичных булевых функций  $f(x) = (f_1(x), \dots, f_m(x))$ , значениями векторных функций на элементах  $x^*$  булева пространства являются  $m$ -компонентные троичные векторы  $f(x^*)$ .

**Задача 1.** Пусть задана булева (двоичная) матрица  $B$ , имеющая  $k$  различных строк  $b^i$  и  $n$  столбцов, при этом  $k \leq 2^{n-1}$ . Требуется закодировать строки  $b^i$  матрицы  $B$  различными минимальными по длине  $m$  булевыми векторами (строками)  $c^i$ , чтобы комбинационная логическая схема, осуществляющая преобразование строк  $b^i$  в строки  $c^i$ , имела возможно меньшую сложность.

Пусть кодирующие комбинации  $c^i$ , соответствующие строкам матрицы  $B$ , образуют булеву матрицу  $C$ . Другими словами, для булевой матрицы  $B$  требуется получить булеву матрицу  $C$ , содержащую  $k$  различных строк, и  $m$  столбцов, где  $m = \lceil \log_2 k \rceil$ , через  $\lceil a \rceil$  обозначается ближайшее целое число, большее либо равное  $a$ . Назовем задачу 1 задачей синтеза схемы кодового преобразователя (либо преобразователя кодов).

Обозначим столбцы булевой матрицы  $B$  через  $x_1, x_2, \dots, x_n$ , столбцы матрицы  $C$  - через  $c_1, c_2, \dots, c_m$ . Пара матриц  $(B, C)$  задает *частичную* векторную булеву функцию  $c(x) = (c_1(x), \dots, c_m(x))$ ,  $x = (x_1, x_2, \dots, x_n)$ , значениями данной векторной функции на двоичных наборах из матрицы  $B$  являются  $m$ -компонентные *двоичные наборы* (строки) из матрицы  $C$ . На наборах, принадлежащих множеству  $V^x \setminus B$ , значения векторной функции  $c(x)$  не определены - соответствующие *троичные векторы*

значений компонентных функций состоят только из неопределенных значений « $\leftarrow$ ».

Перейдем к обсуждению постановки задачи 1.

1. Для решения задачи синтеза любой логической схемы (не только схемы кодового преобразователя) требуется указать базис синтеза (набор логических элементов), часто называемый также технологической библиотекой синтеза [5].

2. Логические схемы могут быть одноктактными (срабатывать в течение одного такта функционирования дискретной системы) либо иметь конвейерную организацию и осуществлять требуемое преобразование набора значений входных сигналов в значения выходных сигналов за несколько тактов смены значений синхросигнала.

3. Синтез схем осуществляется традиционно в два этапа: на первом этапе выполняется технологически независимая оптимизация, на втором этапе - технологическое отображение (technology mapping), когда оптимизированные логические описания булевых функций покрываются функциональными описаниями библиотечных элементов [4, 5].

4. Будем считать, что на вход синтезированной логической схемы при ее функционировании могут поступать только входные наборы из матрицы  $B$  (т. е. наборы из множества  $V^x \setminus B$  никогда не должны поступать на вход схемы), что позволит доопределять исходную частичную векторную функцию до полностью определенной произвольным образом.

Примем следующие *соглашения* (ограничения). Пусть библиотекой логических элементов является библиотека КМОП элементов, описанная в [6], системой синтеза логических схем - синтезатор LeonardoSpectrum [5], а программами технологически независимой оптимизации - программы системы FLC2 [7] логической оптимизации функциональных описаний систем булевых функций. Синтез будет выполняться с целью получения одноктактных (не конвейерных) реализаций схем. Синтезатор LeonardoSpectrum после синтеза схемы подсчитывает сложность (площадь) схем из библиотечных элементов как сумму площадей всех логических элементов схемы и выдает значение данного параметра под названием *Area* (площадь). Заметим, что задержка синтезированной схемы вычисляется в виде значения параметра *Delay*.

Решение задачи 1 сводится к последовательному решению взаимосвязанных задач 2 - 4. Критерием оптимизации при решении задач 2 и 3 является сложность функционального описания системы ДНФ либо сложность BDD-представления, задающих функции кодового преобразователя. Критерием оптимизации при решении задачи 4 является сложность комбинационной схемы кодового преобразователя. Для промышленных синтезаторов логических схем уменьшение сложности (либо

задержки) результирующей логической схемы может достигаться установкой соответствующих управляющих опций синтезатора, в нашем случае LeonardoSpectrum.

**Задача 2.** Найти кодирующую матрицу  $C$  и, возможно, доопределение частичной векторной функции  $c(x) = (c_1(x), \dots, c_m(x))$  на наборах из множества  $V^x \setminus B$  и получить неоптимизированное представление компонентных функций  $c_1(x), \dots, c_m(x)$  кодового преобразователя в виде ДНФ либо СДНФ (совершенной ДНФ).

Доопределение предполагает замену троичных векторов – значений функции  $c(x)$ , состоящих только из неопределенных значений «-» полностью определенными (0, 1) двоичными векторами. В этом случае векторная функция  $c(x) = (c_1(x), \dots, c_m(x))$  является полностью определенной.

**Задача 3.** Выполнить логическую минимизацию частичной либо полностью определенной векторной функции  $c(x) = (c_1(x), \dots, c_m(x))$  и получить ее оптимизированное (минимизированное) алгебраическое представление.

Булевы векторы  $c(b^i)$  значений функции  $c(x) = (c_1(x), \dots, c_m(x))$  на наборах  $b^i$  являются кодами строк  $b^i$  матрицы  $B$ . Для обратного преобразования в роли исходной матрицы  $B$  выступает матрица  $C$ . Пара матриц  $(C, B)$  задает систему булевых функций, которую надо реализовать комбинационной схемой обратного преобразования булевых строк матрицы  $C$  в булевы строки матрицы  $B$ . Частичные векторные булевы функции  $c(x) = (c_1(x), \dots, c_m(x))$  являются промежуточными математическими моделями при нахождении кодов строк матрицы  $B$ .

Решение задачи 3 будем выполнять двумя программами системы FLC-2 [7]. Первой программой является программа Minim раздельной минимизации системы частичных либо полностью определенных функций в классе ДНФ, т. е. каждая из компонентных функций  $c_1(x), \dots, c_m(x)$  будет минимизироваться независимо от других. Результат работы программы Minim – минимизированная система ДНФ, задающая полностью определенные функции кодового преобразователя. В качестве второй программы выступает программа BDD\_Builder совместной минимизации многоуровневых BDDI-представлений систем полностью определенных булевых функций [8]. Результат работы программы BDD\_Builder – взаимосвязанные логические уравнения (формулы разложений Шеннона), задающие полностью определенные функции кодового преобразователя.

Результатом решения задачи 3 является минимизированное представление полностью

определенной векторной функцией  $c(x)$ , являющееся исходными данными для решения задачи 4.

**Задача 4.** Синтезировать логическую схему в заданном базисе логических элементов по минимизированному представлению векторной функции  $c(x)$ .

### III. СПОСОБЫ РЕШЕНИЯ ЗАДАЧ 2, 3

**Способ 1 (тривиальный).** Закодируем строки матрицы  $B$  двоичными наборами, задающие их номера (нумерация начинается с нуля), выполним тривиальное (без комбинаторного поиска) доопределение системы частичных функций – заменим неопределенные значения системы функций  $c(x)$  на наборах из множества  $V^x \setminus B$ , нулевыми строками. Заметим, что номера строк (коды) можно задавать не по порядку, а **случайным** образом размещая их в кодирующей матрице  $C$ , однако это не меняет суть способа 1.

**Способ 2. Замена двоичных строк матрицы  $B$  троичными строками, кодирование строк номерами.** В способе 2 каждая двоичная строка матрицы  $B$  (полная элементарная конъюнкция) заменяется троичной строкой – элементарной конъюнкцией. Получение такой строки сводится к расширению единственного набора из области единичных значений функции  $k_i$  до интервала за счет наборов из области неопределенных значений этой функции.

**Пример 1.** Пусть булева матрица  $B$  имеет 16 строк и пять столбцов (табл. 1).

Таблица 1

*Исходная булева матрица  $B$  и функциональное описание кодового преобразователя, полученное способом 3*

Матрица $B$						Кодирование (эвристика 1)								
						$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$c_1$	$c_2$	$c_3$	$c_4$
0	0	0	0	0	0	0	0	0	0	0				
0	0	1	0	1		-	0	1	-	1	0	0	1	1
0	1	0	1	0		-	1	-	1	-	1	1	1	1
0	1	1	0	0		-	1	1	-	0	0	1	1	0
1	0	0	0	1		1	0	-	-	1	1	1	1	0
1	0	1	1	0		1	-	1	1	-	1	1	0	1
1	1	0	0	0		1	1	-	-	0	1	1	0	0
0	0	0	0	1		0	-	0	-	1	1	0	1	1
0	0	1	1	0		0	-	1	1	-	0	1	1	1
0	1	0	0	0		0	1	0	0	-	1	0	0	0
0	1	1	0	1		0	1	-	-	1	1	0	0	1
1	0	0	1	0		1	-	0	1	-	0	1	0	1
1	0	1	0	0		1	-	1	0	-	0	0	0	1
1	1	0	0	1		1	1	-	-	1	1	0	1	0
0	0	0	1	0		0	0	0	1	-	0	1	0	0
0	0	1	0	0		0	0	1	0	0	0	0	1	0

Троичные векторы, полученные из строк матрицы  $B$  показаны в средней части табл. 1. Например, троичный вектор  $k_2 = (-01-1)$ , соответствующий элементарной конъюнкции  $k_2 = \bar{x}_2 x_3 x_5$ , получается при расширении булева набора  $(00101)$  до интервала, используя наборы  $(00111)$ ,  $(10101)$ ,  $(10111)$  из области  $V^x \setminus B$ .

**Способ 3. Замена двоичных строк матрицы  $B$  троичными строками, кодирование строк матрицы  $B$  согласно эвристики 1.**

В способе 3 каждая двоичная строка матрицы  $B$  (полная элементарная конъюнкция) заменяется троичной строкой – элементарной конъюнкцией так же, как и в способе 2. Однако кодирование полученных троичных векторов выполняется с помощью *эвристики 1: троичные векторы, содержащие много определенных (0, 1) элементов, кодируются булевыми векторами, содержащими возможно меньшее число единичных элементов.*

Данная эвристика ориентирована на сокращение общего числа литералов в системе ДНФ, задающей векторную полностью определенную функцию. Составим множество строк (конъюнкций), содержащих пять определенных элементов (литералов):  $M^5 = \{(00000), (00100)\}$ . Аналогично:  $M^4 = \{(0100-), (0001-)\}$ ,  $M^3 = \{(-01-1), (-1-1-), (-11-0), (10--1), (1-11-), (11--0), (0-0-1), (0-11-), (01--1), (1-01-), (1-10-), (11--1)\}$ ,  $M^2 = \{(-1-1-)\}$ . В примере булева строка  $(00000)$  кодируется комбинацией  $(0000)$  (табл. 1), поэтому полная элементарная конъюнкция  $\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5$  будет отсутствовать в ДНФ каждой из четырех компонентных функций  $c_1, c_2, c_3, c_4$ . Строка  $(00100)$  кодируется вектором  $(0010)$  с одной единичной компонентой, поэтому полная элементарная конъюнкция, содержащая пять литералов, войдет в ДНФ только одной функции  $c_3$ . Строки из множества  $M^4$  кодируются векторами  $(1000), (0100)$ , соответственно. Затем кодируются строки из множества  $M^3$ , используя оставшийся вектор  $(0001)$  с одной единичной компонентой и векторы с двумя и тремя единичными компонентами. Для единственного троичного вектора  $(-1-1-)$  из множества  $M^2$  присваивается кодовая комбинация  $(1111)$  с четырьмя единичными компонентами. Формульное задание системы ДНФ, полученной в способе 3, содержит 96 литералов.

После того как коды присвоены, можно выполнить минимизацию в классе ДНФ полученной системы полностью определенных булевых функций – решать задачу 3. В результате такой минимизации число литералов в системе ДНФ сокращается с 96 до 47. При синтезе логических схем из библиотечных элементов

уменьшение числа литералов в представлениях систем полностью определенных булевых функций приводит к менее сложным схемам [9]. Поэтому число литералов – это основной критерий минимизации алгебраических представлений функций на этапе технологически независимой оптимизации – первом этапе синтеза логических схем.

**Способ 4. Минимизация в классе BDD частичной многозначной функции, зависящей от булевых переменных. Доопределение BDD «вертикальное», кодирование строк матрицы  $B$  согласно эвристики 2.**

Данный способ состоит из пяти этапов.

**Этап 1.** Составление по матрице  $B$  функционального описания не полностью определенной многозначной ( $k$ -значной) функции: каждой строке  $b^i$  матрицы  $B$  ставится в соответствие значение  $p^i$  многозначной переменной  $p$ , являющейся функцией  $p(x_1, \dots, x_n)$ , зависящей от булевых переменных  $x_1, \dots, x_n$ .

**Этап 2.** Построение такого графа BDD многозначной функции  $p(x_1, \dots, x_n)$ , который имеет возможно меньшее число функциональных вершин. Листовые вершины минимизированного графа BDD соответствуют значением  $k$ -значной функции, так как в результате минимизации числа вершин BDD каждое неопределенное значение «-» доопределяется некоторым значением  $p^i$  функции  $p(x_1, \dots, x_n)$ . Заметим, что BDD, реализующие  $k$ -значные функции булевых аргументов, являются частным случаем  $k$ -значных функций, зависящих от многозначных аргументов  $x_i$  и представляющих эти функции диаграмм решений, рассмотренных в [10].

**Этап 3.** Кодирование значений  $k$ -значной функции булевыми кодами по *эвристике 2: соседние листовые вершины графа BDD кодируются соседними (различающимися значениями одной компоненты)  $t$ -компонентными булевыми векторами.* Для оставшихся листовых вершин коды присваиваются произвольным образом.

**Этап 4.** Получение системы ДНФ полностью определенных булевых функций по минимизированному графу BDD и кодированию листовых вершин.

**Этап 5.** Для полученной системы ДНФ выполняются программы логической минимизации.

Рассмотрим применение способа 4 для примера той матрицы  $B$ . В табл. 2 дана 16-значная функция переменная  $p$  (функция  $p(x_1, \dots, x_5)$ ) – это результат этапа 1. На этапе 2 рассмотрим только одну перестановку  $\langle x_1, x_2, x_3, x_4, x_5 \rangle$  переменных, по которой построим BDD. Заметим, что всех перестановок  $n!$  (факториал числа  $n$ ) и задача поиска

перестановки, обеспечивающей минимальное число функциональных вершин BDD, является сложной комбинаторной задачей, которой посвящено большое число работ для случая 2-значных (булевых) функций. Краткий обзор таких работ представлен в [4, с. 39].

Таблица 2

Многозначная функция и функциональное описание кодового преобразователя, полученное способом 4

Многозначная функция		Кодирование (эвристика 2)						
$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$c_1$	$c_2$	$c_3$	$c_4$
0	0	0	0	0	1	1	0	1
0	0	1	0	1	1	0	0	1
0	1	0	1	0	0	1	1	0
0	1	1	0	0	0	1	0	0
1	0	0	0	1	0	0	1	0
1	0	1	1	0	1	0	1	0
1	1	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0
0	0	1	1	0	1	1	1	0
0	1	0	0	0	0	1	1	1
0	1	1	0	1	0	1	0	1
1	0	0	1	0	1	0	0	1
1	0	1	0	0	1	1	1	1
1	1	0	0	1	1	1	1	1
0	0	0	1	0	1	0	1	1
0	0	1	0	0	1	0	0	0

Построим граф BDD функции  $p(x_1, x_2, x_3, x_4, x_5)$  из табл. 2 для последовательности (перестановки) переменных  $\langle x_1, x_2, x_3, x_4, x_5 \rangle$ . Листовые вершины в графе BDD будут задавать определенные либо неопределенные «-» значения функции. Граф BDD, в котором проведено сокращение функциональных вершин путем доопределения (замены) неопределенных значений определенными, показан на рис. 1. Алгоритм «вертикального» доопределения BDD состоит в замене неопределенных значений листовых вершин определенными значениями соседних определенных листовых вершин. Соседними являются вершины пары кофакторов (подфункций)  $g_0, g_1$  получающихся в результате разложения Шеннона  $g = \bar{x}_i g_0 \vee x_i g_1$  какого-либо кофактора  $g$ . На этапе 4 осуществляется замена каждого многозначной

подфункции своей подсистемой булевых функций, зависящих от соответствующих множеств булевых переменных. Уравнения разложения Шеннона многозначной функции и ее многозначных подфункций заменяются уравнениями с булевыми переменными. Элиминация (устранение) промежуточных переменных позволяет получить систему ДНФ полностью определенных функций, заданную в правой части табл. 2.

**Способ 5. Минимизация в классе BDD частичной многозначной функции, зависящей от булевых переменных, доопределение BDD «горизонтальное», кодирование строк матрицы B согласно эвристики 2.**

В способе 5 выполняются этапы 1 – 3 из способа 4. «Горизонтальное» доопределение BDD описано в [4, с. 109]. Затем по полученной системе ДНФ определяется код каждой строки  $b^i$  матрицы B. Для этого находится элементарная конъюнкция, которую имплицитно полная элементарная конъюнкция, соответствующая строке  $b^i$ .

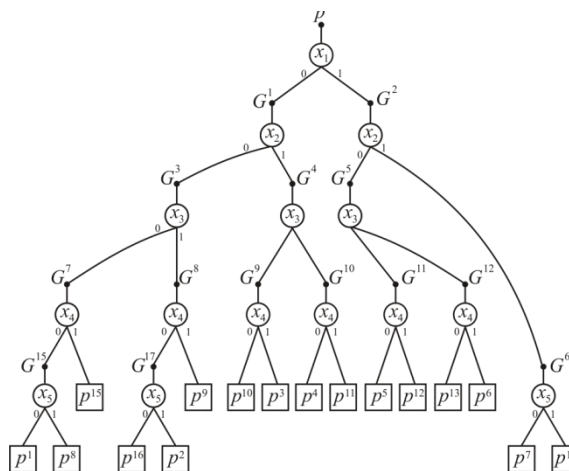


Рис. 1. Граф BDD для кодирования соседних вершин

#### IV. ПРАКТИЧЕСКИЙ ПРИМЕР

Рассмотрим пример, когда матрица B имеет параметры  $n=17, k=2^{16}=65536$ . Тогда  $m = \lceil \log_2 k \rceil = 16$  и требуется построить кодовый преобразователь 17-разрядных двоичных слов в 16-разрядные. Оптимизация по способу 5 (однако со случайным присвоением 16-разрядных кодов) позволяет получить функциональное BDD-описание, содержащее 8 553 двухходовых конъюнкций, 4 119 двухходовых дизъюнкций и синтезировать схему, содержащую 43 818 транзисторов. Синтез был проведен в библиотеке проектирования системы CMOSLD [6]. Функциональное описание схемы для обратного преобразования (реализация системы СДНФ булевых функций, заданной парой булевых матриц (C, B)), содержало 5 473 двухходовых конъюнкций, 2 920 двухходовых дизъюнкций, после BDD-оптимизации и синтеза была получена логическая схема, содержащая 22 232 транзисторов. Схемная реализация

того же примера кодового преобразователя для FPGA xc7k70tfbv676-1 семейства Kintex-7 [11] осуществлялась в системе автоматизированного проектирования Vivado [12], опции синтеза – Vivado Synthesis Default. Сложность схем оценивалась в числе программируемых элементов LUT-6, имеющих 6 входных переменных (LUT – Look-Up Table (таблица, реализующая логическую функцию)). Сложность схемы кодового преобразователя составила 2 014 LUT-6, сложность схемы обратного преобразования строк матрицы  $C$  в строки матрицы  $B$  составила 940 LUT-6. Таким образом, построенный кодовый преобразователь позволил сократить вдвое число тактов для передачи информации по 16-разрядной шине, однако для этого потребовались соответствующие аппаратные затраты. Результаты вычислительных экспериментов показывают, что при одинаковом кодировании строк матрицы  $B$ , но при разных доопределениях частичной векторной функции  $c(x)$  сложности схем и их задержки могут иметь значительные различия. Был проведен эксперимент, когда по матрицам  $B$ ,  $C$  составлялись VHDL-описания систем частичных функций [4, с. 183], которые отправлялись на синтез в LeonardoSpectrum. В этом случае технологически независимую оптимизацию выполнял синтезатор. Оказалось, что во всех случаях технологически независимая оптимизация системы FLC-2 приводила к лучшим результатам синтеза.

#### V. МЕТОДИКА ЛОГИЧЕСКОГО ПРОЕКТИРОВАНИЯ КОДОВЫХ ПРЕОБРАЗОВАТЕЛЕЙ

Предложенная методика синтеза кодовых преобразователей включает: 1) выбор вида описания, предназначенного для описания функций проектируемого преобразователя кодов. Были рассмотрены два вида – ДНФ и BDD; 2) выбор способа доопределения исходной системы частичных функций на множестве наборов  $V^x \setminus B$ ; 3) способы (эвристики) кодирования наборов матрицы  $B$ ; 4) методы (программы) оптимизации полученной системы функций - раздельная минимизация в классе ДНФ, совместная минимизация в классе BDDI, совместная минимизация в классе BDD частичных функций. Для различных исходных булевых матриц  $B$  и различных библиотек проектирования могут быть эффективными различные способы решения задач 2 - 4, однако практика проектирования показывает, что во многих случаях минимизация BDD и BDDI-представлений и новых разложений [13] для логической оптимизации приводит к более простым комбинационным схемам, синтезируемым из библиотечных элементов [4, 8]. Для «обратных» кодовых преобразователей кодирование строк матрицы  $C$  уже имеется (это строки матрицы  $B$ ), поэтому логическая оптимизация сводится к выбору такого доопределения частичной векторной функции, заданной парой матриц  $(C, B)$ , для которого сложность функционального описания и, соответственно, логической схемы обратного преобразователя была бы по возможности меньшей.

#### VI. ЗАКЛЮЧЕНИЕ

В докладе предложена методика проектирования кодовых преобразователей, использующая различные способы составления функциональных описаний кодовых преобразователей в виде систем частичных булевых функций и различные методы логической минимизации таких систем функций. Предложенная методика проектирования кодовых преобразователей рассмотренного в статье класса может быть использована и при проектировании кодовых преобразователей других классов, если описания их функций сводятся к системам частичных булевых функций.

#### ЛИТЕРАТУРА

- [1] Применение интегральных микросхем в электронной вычислительной технике: Справочник / Под ред. Б.Н. Файзулаева, Б.В. Тарабрина. – М. Радио и связь. 1987. 384 с.
- [2] Стемповский А.Л., Корнилов А.И., Семенов М.Ю. Особенности реализации устройств с цифровой обработкой сигналов в интегральном исполнении с применением модулярной арифметики // Информационные технологии. 2004. № 2. С. 2–9.
- [3] Brayton K. R., Hachtel G. D., McMullen C., Sangiovanni-Vincentelli A. L. Logic Minimization Algorithm for VLSI Synthesis. – Boston, Kluwer Academic Publishers, 1984, 193 p.
- [4] Бибило П. Н. Применение диаграмм двоичного выбора при синтезе логических схем. Минск : Беларус. навука, 2014. 231 с.
- [5] Бибило П.Н. Системы проектирования интегральных схем на основе языка VHDL. StateCAD, ModelSim, LeonardoSpectrum. – М.: СОЛОН-Пресс, 2005. 384 с.
- [6] Система логического проектирования функциональных блоков заказных КМОП СБИС с пониженным энергопотреблением / П.Н. Бибило [и др.] // Микроэлектроника. 2017. Т. 46, № 1. С.72–88.
- [7] Бибило П.Н., Романов В.И. Система логической оптимизации функционально-структурных описаний цифровых устройств на основе продукционно-фреймовой модели представления знаний // Проблемы разработки перспективных микро- и нанoeлектронных систем (МЭС). 2020. Выпуск 4. С. 9-16. doi:10.31114/2078-7707-2020-4-9-16
- [8] Бибило П.Н., Ланкевич Ю.Ю. Экспериментальное исследование эффективности программ минимизации BDD-представлений систем булевых функций при синтезе комбинационных схем из библиотечных КМОП // Проблемы разработки перспективных микро- и нанoeлектронных систем (МЭС). 2020. Выпуск 1. С. 33-39. doi:10.31114/2078-7707-2020-1-33-39
- [9] Брейтон Р. К., Хэчтел Г. Д., Санджованни-Винчензелли А. Л. Синтез многоуровневых комбинационных логических схем // ТИИЭР. 1990. Т. 78, № 2. С. 38–83.
- [10] E Kam T., Villa T., Brayton R. K., Sangiovanni-Vincentelli A. L. Multi-Valued Decision Diagrams for Logic Synthesis and Verification // Memorandum No. UCB/ERL M96/75, 1996. 39 p.
- [11] Соловьев, В. В. Архитектуры ПЛИС фирмы Xilinx: FPGA и CPLD 7-й серии. – М.: Горячая линия–Телеком, 2016. 392 с.
- [12] Тарасов И. Е. ПЛИС Xilinx. Языки описания аппаратуры VHDL и Verilog, САПР, приемы

# Hardware Implementation of Code Converters Designed to Reduce the Length of Binary Encoded Words

P.N. Bibilo

United Institute of Informatics Problems of NAS of Belarus, bibilo@newman.bas-net.by

**Abstract** — The problem of synthesis of combinational circuits of code converters designed to reduce the length of words from a given set of encoded binary words is considered. The encoding assumes that different binary words will be encoded by different binary codes of shorter length. Code converters of this type are designed to reduce the length of binary words transmitted in digital systems over data buses when the bit depth of the transmitted words exceeds the bit depth of the data bus. For example, 18-bit or 17-bit words need to be transmitted over a 16-bit data bus. Each such word can be transmitted in two cycles of operation of a digital system, however, this approach reduces the overall performance of the system. One of the approaches to solve such problems is the development of combinational circuits that convert long binary encoded words into shorter ones. The proposed methods for solving the problem of synthesizing circuits of code converters are based on the compilation and logical minimization of such forms of systems of incompletely specified Boolean functions as disjunctive normal forms (DNF) and binary decision diagrams called BDD representations (BDD Binary Decision Diagram). Using BDD to minimize representations of  $k$ -valued functions that depend on Boolean variables is also proposed. Technologically independent logical minimization of functional descriptions of the designed code converters is proposed to be performed by programs for minimizing systems of Boolean functions in the DNF class and programs for joint minimization of BDD representations of systems of completely specified Boolean functions. Minimization of functional descriptions is aimed at reducing the hardware complexity of combinational circuits in the basis of library elements or programmable FPGA elements implementing code converters of the class in question.

**Keywords** — code converter, system of Boolean functions, Disjunctive Normal Form, Binary Decision Diagram, Shannon expansion, digital logic synthesis, VHDL, VLSI.

## REFERENCES

- [1] Primenenie integral'nyh mikroskhem v elektronnoj vychislitel'noj tekhnike: Spravochnik (Application of integrated circuits in electronic computing: Reference) / Pod red. B.N. Fajzulaeva, B.V. Tarabrina. – M. Radio i svyaz'. 1987. 384 s.
- [2] Stempkovskij A.L., Kornilov A.I., Semenov M. YU. Osobennosti realizacii ustrojstv s cifrovoj obrabotkoj signalov v integral'nom ispolnenii s primeneniem modulyarnoj arifmetiki (Features of the implementation of devices with digital signal processing in integrated design using modular arithmetic) // Informacionnye tekhnologii. 2004. № 2. S. 2–9.
- [3] Brayton K. R., Hachtel G. D., McMullen C., Sangiovanni-Vincentelli A. L. Logic Minimization Algorithm for VLSI Synthesis. Boston, Kluwer Academic Publishers, 1984. 193 p.
- [4] Bibilo P.N. Primenenie diagramm dvoichnogo vybora pri sinteze logicheskikh shem (Application of Binary Decision Diagrams at synthesis of logical circuits) Minsk: Belarus. navuka, 2014. 231 s.
- [5] Bibilo P.N. Cistemy proektirovaniya integral'nyh skhem na osnove yazyka VHDL. StateCAD, ModelSim, LeonardoSpectrum (Integrated Circuit Design Systems Based on the VHDL Language. StateCAD, ModelSim, LeonardoSpectrum). – M.: SOLON-Press, 2005. 384 s.
- [6] Sistema logicheskogo proektirovaniya funkcional'nyh blokov zakaznyh KMOP SBIS s ponizhennym energopotrebleniem (System of Logical Design of Functional Blocks of Custom CMOS VLSI with Reduced Power Consumption) / P.N. Bibilo [i dr.] // Mikroelektronika. 2017. T. 46, № 1. S.72–88.
- [7] Bibilo P.N., Romanov V.I. The system of logical optimization of functional structural descriptions of digital circuits based on production-frame knowledge representation model // Problems of Perspective Micro- and Nanoelectronic Systems Development - 2020. Issue 4. P. 9-16. doi:10.31114/2078-7707-2020-4-9-16
- [8] Bibilo P.N., Lankevich Y.Y. Experimental Research of Effectiveness of Programs for Minimizing BDD Representations of Boolean Function Systems in Synthesis of Combinatorial CMOS Circuits // Problems of Perspective Micro- and Nanoelectronic Systems Development - 2020. Issue 1. P. 33-39. doi:10.31114/2078-7707-2020-1-33-39
- [9] Brayton R. K., Hachtel G. D., Sangiovanni-Vincentelli A. L. Synthesis of multi-level combinational logic circuits, Trudy Institute inzhenerov po jelektronike i radiotekhnike, 1990, vol. 78, no. 2, pp. 38–83.
- [10] Kam T., Villa T., Brayton R.K., Sangiovanni-Vincentelli A. L. Multi-Valued Decision Diagrams for Logic Synthesis and Verification // Memorandum No. UCB/ERL M96/75, 1996. - 39 p.
- [11] Solov'ev V. V. Arhitektury PLIS firmy Xilinx: FPGA i CPLD 7-j serii. (XILINX FPGA Architectures: FPGA and CPLD 7-Series). – M.: Goryachaya liniya–Telekom, 2016. 392 s.
- [12] Tarasov I. E. PLIS Xilinx. YAzyki opisaniya apparatury VHDL i Verilog, SAPR, priemy proektirovaniya (XILINX FPGA. Hardware Description Languages VHDL and Verilog, CAD, Design Techniques). – M.: Goryachaya liniya–Telekom, 2020. 538 s.
- [13] Amaru L. G. New Data Structures and Algorithms for Logic Synthesis and Verification. – Springer, 2017. – 156 p.