

УДК 519.714.5

РЕАЛИЗАЦИЯ В FPGA РАЗРЕЖЕННЫХ МАТРИЧНЫХ ФОРМ СИСТЕМ ДНФ БУЛЕВЫХ ФУНКЦИЙ



П.Н. Бибило

Заведующий лабораторией логического проектирования ОИПИ НАН Беларуси, доктор технических наук, профессор bibilo@newman.bas-net.by



С.Н. Кардаш

Старший научный сотрудник лаборатории логического проектирования ОИПИ НАН Беларуси, кандидат технических наук kardas77@gmail.com

П.Н. Бибило

С 1994 г. является заведующим лабораторией логического проектирования Объединенного института проблем информатики НАН Беларуси, с которым неразрывно связана вся его трудовая деятельность. Основные научные результаты Бибило П.Н. относятся к теории логического проектирования цифровых устройств, основные направления – разработка методов синтеза цифровых устройств с использованием современной элементной базы сверхбольших интегральных схем (СБИС), автоматизация процессов проектирования (синтеза, моделирования и функциональной верификации) заказных цифровых СБИС, разработка систем автоматизированного проектирования (САПР).

С.Н. Кардаш

Окончил БГУ им. Ленина. Область научных интересов – синтез матричных структур заказных цифровых СБИС, разработка программ логической оптимизации систем булевых функций.

Аннотация. Рассматривается проблема выбора лучших методов и программ для схемной реализации в *FPGA* разреженных систем ДНФ (дизъюнктивных нормальных форм) полностью определенных булевых функций. Для матричных форм разреженных систем ДНФ троичная матрица, задающая элементарные конъюнкции, содержит большую долю неопределенных значений, соответствующих в алгебраической записи отсутствующим литералам булевых входных переменных, а булева матрица, задающая вхождения конъюнкций в ДНФ функций, содержит большую долю нулевых значений. Для проектирования схем *FPGA* показана эффективность комбинированного подхода, использующего сначала программы блочного покрытия системы ДНФ с последующим применением программ минимизации многоуровневых представлений блоков в виде булевых сетей, минимизированных на основе разложений Шеннона.

Ключевые слова: система булевых функций, дизъюнктивная нормальная форма (ДНФ), минимизация ДНФ, *Binary Decision Diagram (BDD)*, булева сеть, разложение Шеннона, блочное покрытие системы ДНФ, синтез логической схемы, *FPGA*, *VHDL*.

Введение. Проблема эффективной схемной реализации цифровых комбинационных блоков в *FPGA* является актуальной при создании средств автоматизированного проектирования цифровых систем. Важным аспектом этой проблемы является то, что современные синтезаторы логических схем чувствительны к форме задания проектной информации, в качестве которой выступают *VHDL* либо *Verilog* описания [1] моделей функционирования комбинационных схем – те или иные формы задания систем полностью определенных булевых функций. Синтез логических схем выполняется в два этапа – технологически независимая оптимизация представлений систем булевых функций (этап 1) и технологическое отображение (этап 2) в заданный базис (библиотеку) программируемых элементов *FPGA*. Важнейшим является первый этап, на котором выбирается форма представления системы булевых функций и осуществляется

минимизация этой формы. Результат выполнения первого этапа определяет важнейшие параметры синтезированной на втором этапе логической схемы – площадь, временную задержку и энергопотребление. Синтезаторы логических схем, входящие в САПР *FPGA* [2], имеют в своем составе программы логической оптимизации, что позволяет провести синтез исходных не минимизированных описаний систем булевых функций, в том числе и заданных в виде систем ДНФ. Однако, как показано в [3], предварительная технологически независимая оптимизация многоуровневых представлений систем функций на основе разложения Шеннона методами минимизации *BDD* (*BDD* – *Binary Decision Diagram*, бинарная диаграмма решений) позволяет во многих случаях улучшать результаты схемной реализации блоков комбинационной логики в *FPGA*.

В данной работе рассматриваются разреженные системы ДНФ полностью определенных булевых функций. Для матричных форм таких систем ДНФ троичная матрица, задающая элементарные конъюнкции, содержит большую долю неопределенных значений, а булева матрица, задающая вхождения конъюнкций в ДНФ функций, содержит большую долю нулевых значений и, следовательно, небольшую долю единичных значений. Для разреженных систем ДНФ булевых функций предлагаются алгоритмы их блочного покрытия. Проводится экспериментальное исследование эффективности применения блочных многоуровневых представлений при синтезе *FPGA*. Синтезированные схемы сравниваются по площади и временной задержке. Многоблочные многоуровневые представления сравниваются по результатам синтеза с отдельными и совместными многоуровневыми представлениями исходной системы ДНФ и минимизированными двухуровневыми представлениями, под которыми понимаются совместно минимизированные ДНФ. В качестве базовых многоуровневых представлений использованы бинарные диаграммы решений с инверсными кофакторами (*BDDI*-представления) и булевы сети (*Bool*-представления). Минимизация *BDDI*-представлений выполняется по матричным заданиям систем ДНФ булевых функций, *Bool*-представлений – по логическим уравнениям, задающим те же системы ДНФ. В результате проведенных экспериментов установлено: для разреженных систем ДНФ наряду с методами минимизации систем функций в классе ДНФ эффективным методом технологически независимой оптимизации является комбинированный метод, включающий блочное покрытие системы ДНФ и последующую минимизацию многоуровневых представлений блоков, при этом функции блоков минимизируются в классе булевых сетей с использованием разложений Шеннона.

Многоуровневые *BDDI*-представления систем булевых функций. Под *BDDI*-представлением (*BDDI* – *Binary Decision Diagram with Inverse cofactors*) понимается ориентированный бесконтурный граф, задающий последовательные разложения Шеннона булевой функции $f(\mathbf{x})=f(x_1, \dots, x_n)$, $\mathbf{x}=(x_1, \dots, x_n)$, либо системы $\mathbf{f}(\mathbf{x})=(f^1(\mathbf{x}), \dots, f^m(\mathbf{x}))$ булевых функций по всем переменным x_1, x_2, \dots, x_n при заданном порядке (перестановке) переменных, по которым проводятся разложения, при условии нахождения пар взаимно инверсных кофакторов.

Разложением Шеннона булевой функции $f(\mathbf{x})$ по переменной x_i называется представление

$$f(\mathbf{x}) = \bar{x}_i f_0 + x_i f_1. \quad (1)$$

Функции $f_0=f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$, $f_1=f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$ в правой части (1) называются кофакторами (cofactors, англ.) разложения по переменной x_i . Каждый из кофакторов $f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$, $f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$ может быть разложен по одной из переменных из множества $\{x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n\}$. Процесс разложения кофакторов

заканчивается, когда все n переменных будут использованы для разложения. $BDDI$ -представлению соответствует совокупность взаимосвязанных формул разложения Шеннона. Сравнение кофакторов на равенство и нахождение взаимно инверсных кофакторов осуществляется с использованием полиномов Жегалкина – канонических представлений булевых функций [4], либо сравнением ДНФ, задающих кофакторы.

Минимизация сложности $BDDI$ заключается в нахождении последовательности (*перестановки*) переменных разложения Шеннона, при которой число кофакторов является наименьшим [4]. Если для каждой функции системы соответствующая ей $BDDI$ строится независимо, то такая минимизация называется *раздельной*. При построении раздельных $BDDI$ могут появляться одинаковые кофакторы в $BDDI$ различных функций системы, однако данный факт при построении $BDDI$ для каждой отдельной функции во внимание не принимается.

Многоуровневые Bool-представления систем булевых функций.

Bool-представление системы булевых функций соответствует булевой сети (ориентированному бесконтурному графу), функциями вершин которой могут быть логические операции «конъюнкции» либо «дизъюнкции» (возможно с инверсией) над литералами булевых переменных. *Литерал* – это булева переменная либо ее инверсия. Логическая минимизация булевых сетей на основе разложения Шеннона заключается в поиске такой перестановки переменных разложения, при которой число литералов в булевой сети является наименьшим. В булевой сети разложение Шеннона записывается в виде трех формул

$$f(x) = w_0 \vee w_1; \quad w_0 = \bar{x}_i f_0; \quad w_1 = x_i f_1. \quad (2)$$

После разложения Шеннона по очередной переменной минимизация булевой сети сводится к следующему: ищутся вершины булевой сети, опирающиеся на одинаковые подсети, после чего проводится сокращение сети и находятся уравнения, соответствующие редуцированной сети [5].

Раздельные BDDI- и Bool-минимизации для выделенных подсистем системы булевых функций (либо для отдельных функций системы) заключаются в нахождении своей перестановки $\langle x_1, x_2, \dots, x_n \rangle$ переменных разложения для каждой из подсистем функций, в то время как при *совместной BDDI- и Bool-минимизации* используется одна и та же перестановка переменных разложения для всех функций $f^1(x), \dots, f^m(x)$ системы $f(x)$. Для некоторых систем функций преимущество при синтезе имеет совместная минимизация, для других – раздельная $BDDI$ - либо $Bool$ -минимизация. Обычно раздельная минимизация позволяет получать схемы, характеризуемые большим быстродействием.

Минимизация в классе ДНФ. Кратчайшей системой ДНФ D_f для системы булевых функций $f(x) = (f^1(x), \dots, f^m(x))$ называется такая система ДНФ, которая содержит минимальное число общих элементарных конъюнкций, на которых заданы ДНФ D_{f_i} , $i=1, \dots, m$, всех функций $f^i(x)$ системы $f(x) = (f^1(x), \dots, f^m(x))$. Задача совместной минимизации системы булевых функций: для заданной системы $f(x) = (f^1(x), \dots, f^m(x))$ булевых функций найти кратчайшую систему ДНФ D_f . Совместная минимизация систем булевых функций в экспериментах выполнялась программой *Espresso ПС* [6].

Блочное покрытие системы ДНФ булевых функций. Пусть (T^x, B^f) – пара матриц, задающая матричную форму системы ДНФ булевых функций $f(x) = (f^1(x), \dots, f^m(x))$, $x = (x_1, \dots, x_n)$, где T^x – троичная матрица, задающая общие элементарные конъюнкции, B^f –

булева матрица, единичные элементы которой отмечают вхождения элементарных конъюнкций в ДНФ функций [7]. Система ДНФ

$$\begin{aligned}
 f^1 &= x_1 \bar{x}_2 \bar{x}_3 x_4 \quad x_2 x_3 \bar{x}_5 \quad \bar{x}_1 x_4 x_5; \\
 f^2 &= x_5 x_6 \bar{x}_7 \bar{x}_8 \bar{x}_9 \quad \bar{x}_4 x_5 \bar{x}_6 x_8 x_9 x_{10} \quad x_4 x_5 x_8; \\
 f^3 &= x_5 x_6 \bar{x}_7 \bar{x}_8 \bar{x}_9 \quad \bar{x}_4 x_5 \bar{x}_6 x_8 x_9 x_{10} \quad x_1 \bar{x}_2 \bar{x}_4 x_5 \quad x_2 x_3 \bar{x}_5 \quad \bar{x}_1 x_4 x_5; \\
 f^4 &= x_5 x_6 \bar{x}_7 \bar{x}_8 \bar{x}_9 \quad \bar{x}_4 x_5 \bar{x}_6 x_8 x_9 x_{10} \quad \bar{x}_1 \bar{x}_2 x_8 x_{10} \quad x_1 \bar{x}_8 x_{10} \quad x_4 x_5 x_8; \\
 f^5 &= x_2 x_8 \bar{x}_9 \quad x_1 \bar{x}_8 x_{10}; \\
 f^6 &= x_1 x_2 \bar{x}_8 x_9 x_{10} \quad \bar{x}_1 \bar{x}_2 x_8 x_{10} \quad x_2 x_8 \bar{x}_9
 \end{aligned} \tag{3}$$

задается парой матриц (T^x, B^f) в табл. 1.

Таблица 1. Матричная форма системы ДНФ булевых функций

Номер строки	Троичная матрица T^x										Булева матрица B^f					
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	f^1	f^2	f^3	f^4	f^5	f^6
1		1	1	-	-	-	-	0	1	1	0	0	0	0	0	1
2		-	-	-	-	1	1	0	0	0	0	1	1	1	0	0
3		-	-	-	0	1	0	-	1	1	0	1	1	1	0	0
4		0	0	-	-	-	-	-	1	-	0	0	0	1	0	1
5		1	0	-	0	1	-	-	-	-	1	0	1	0	0	0
6		-	1	1	-	0	-	-	-	-	1	0	1	0	0	0
7		-	1	-	-	-	-	-	1	0	0	0	0	0	1	1
8		1	-	-	-	-	-	-	0	-	0	0	0	1	1	0
9		0	-	-	1	1	-	-	-	-	1	0	1	0	0	0
10		-	-	-	1	1	-	-	1	-	0	1	0	1	0	0

Число n переменных равно 10 ($n=10$), число функций $m=6$, ДНФ заданы на $k=10$ общих элементарных конъюнкциях, число d операторов дизъюнкции в (3) равно 15 ($d=15$). Площадь $Q(T^x, B^f)$ матриц будем вычислять по формуле (4) и выразать в числе бит:

$$Q(T^x, B^f) = (n+m)k \text{ (бит)}. \tag{4}$$

Рассмотрим систему ДНФ, каждая элементарная конъюнкция которой включает не более t литералов. Это значит, что в каждой строке троичной матрицы T^x находится не более t определенных 0, 1 элементов (остальные элементы равны «-»). Пусть $p \leq t$. Рассмотрим пару (T_{H_i}, B_{H_i}) подматриц, где T_{H_i} – строчная (образованная некоторыми строками матрицы T^x) подматрица матрицы T^x , B_{H_i} – подматрица матрицы B^f , заданная на том же подмножестве строк, что и T_{H_i} . Назовем пару (T_{H_i}, B_{H_i}) блоком H_i .

Блок H_i назовем (p,s,q) -ограниченным, если

- число столбцов T_{H_i} , содержащих определенные элементы 0, 1, не превышает p ;
- число ненулевых столбцов матрицы B_{H_i} не превышает q ;
- число строк подматриц T_{H_i}, B_{H_i} не превышает s .

Блок (T_{H_i}, B_{H_i}) назовем (p,q) -ограниченным, если значение параметра s не ограничивается, т. е. всегда предполагается $s=k$. Блок назовем p -ограниченным, если значения параметров s и q не ограничиваются, т. е. предполагается всегда $s=k$ и $q=m$. Легко видеть, что пара (T_{H_i}, B_{H_i}) задает соответствующую систему ДНФ булевых функций. Рассмотрим три пары подматриц, заданных в табл. 2 – 4.

Таблица 2. Система ДНФ булевых функций блока H_1

Номер строки	Троичная матрица T_{H_1}	Булева матрица B_{H_1}
	$x_1 \ x_2 \ x_8 \ x_9 \ x_{10}$	$f_1^4 \ f_5^5 \ f_6^6$
1	1 1 0 1 1	0 0 1
4	0 0 1 - 1	1 0 1
7	- 1 1 0 -	0 1 1
8	1 - 0 - 1	1 1 0

Таблица 3. Система ДНФ булевых функций блока H_2

Номер строки	Троичная матрица T_{H_2}	Булева матрица B_{H_2}
	$x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9$	$f_2^2 \ f_1^3 \ f_2^4$
2	- 1 1 0 0 0	1 1 1
3	0 1 0 - 1 1	1 1 1
10	1 1 - - 1 -	1 0 1

Таблица 4. Система ДНФ булевых функций блока H_3

Номер строки	Троичная матрица T_{H_3}	Булева матрица B_{H_3}
	$x_1 \ x_2 \ x_3 \ x_4 \ x_5$	$f_1^4 \ f_2^3$
5	1 0 - 0 1	1 1
6	- 1 1 - 0	1 1
9	0 - - 1 1	1 1

Пара (T_{H_1}, B_{H_1}) является $(5,4,3)$ -ограниченной и имеет площадь 32 бит; пара (T_{H_2}, B_{H_2}) является $(6,3,3)$ -ограниченной и имеет площадь 27 бит, пара (T_{H_3}, B_{H_3}) является $(5,3,2)$ -ограниченной и имеет площадь 21 бит.

Множество $\{H_1, \dots, H_v\} = \{(T_{H_1}, B_{H_1}), \dots, (T_{H_v}, B_{H_v})\}$ блоков назовем блочным дизъюнктивным покрытием (далее *блочным покрытием*) пары матриц (T^x, B^f) , если каждый единичный элемент матрицы B^f входит только в одну из подматриц B_{H_i} , а каждая строка матрицы T^x входит хотя бы в одну из подматриц $T_{H_i}, i=1, \dots, v$.

Блочное покрытие (p,q) -ограниченными блоками будет являться разбиением исходной системы ДНФ на *непересекающиеся* подсистемы функций, если все функции каждого блока H_i будут зависеть от одного и того же подмножества переменных, мощность которого не более p .

Если же для подсистемы, состоящей из q функций, число ее аргументов превышает число p , то для реализации блочного разложения требуются операции дизъюнкции для некоторых (либо всех) функций нескольких (возможно всех) блоков.

Задача 1 блочного покрытия системы ДНФ по критерию минимальности числа блоков: найти покрытие пары (T^x, B^f) возможно меньшим числом (p,q) -ограниченных блоков $(T_{H_i}, B_{H_i}), i=1, \dots, v$.

Задача 2 блочного покрытия системы ДНФ по критерию минимальности площади блоков: найти покрытие пары (T^x, B^f) возможно меньшим числом p -ограниченных блоков $(T_{H_i}, B_{H_i}), i=1, \dots, w$, имеющих возможно меньшую суммарную площадь с учетом площадей логических элементов, реализующих дизъюнкции функций.

Алгоритмы и программы решения задач блочного покрытия системы ДНФ описаны в [8]. Они являются эвристическими и итерационными – на каждой итерации формируется один блок на основе эвристик выбора столбцов и строк соответствующих подматриц очередного блока. После этого обнуляются те единичные элементы булевой матрицы B^f , которые принадлежат сформированному блоку. Процесс формирования блоков заканчивается, когда все элементы матрицы B^f станут нулевыми.

Применение программы решения задачи 2 (для $p=6$) блочного покрытия с минимизацией площади блоков для системы ДНФ из табл. 1 позволяет получить три блока H_1, H_2, H_3 , заданных в табл. 2–4, соответственно. В табл. 3 ДНФ функций f^2, f_2^4 одинаковы, т. е. $f^2 = f_2^4$, в табл. 4 также задаются одинаковые ДНФ $f_1^4 = f_2^3$, поэтому можно сказать, что блочное покрытие является приемом логической оптимизации и позволяет выделять *общие подфункции* в дизъюнктивных разложениях системы ДНФ. Изменяя параметры p, q , можно получать различные блочные покрытия системы ДНФ, характеризуемые различной суммарной площадью, различным числом блоков и различным числом дизъюнкций, требуемых для формирования выходных функций.

Разреженные системы ДНФ. Под *разреженностью* α *троичной матрицы* T^x будем понимать отношение числа неопределенных элементов «–» к числу всех элементов этой матрицы и выражать это отношение в процентах. Например, троичная матрица T^x (табл. 1) содержит 62 неопределенных значения «–», общее число элементов матрицы T^x равно 100 (матрица состоит из 10 столбцов и 10 строк), следовательно, $\alpha = 62\%$. Под *разреженностью* β *булевой матрицы* B^f будем понимать отношение числа единичных элементов к числу всех ее элементов и выражать это отношение в процентах. В булевой матрице B^f (табл. 1) число единичных элементов равно 21, следовательно, $\beta = 21/60 = 0,35$, что составляет 35%. Чем большее значение имеет параметр α и чем меньшее значение имеет параметр β , тем более разреженной является матричная форма системы ДНФ.

Исходные данные для экспериментов. Системы булевых функций для экспериментов (табл. 5) были заданы в двух формах – матричной форме и форме логических уравнений.

Таблица 5. Исходные данные – разреженные системы ДНФ

Схема	n	m	k	d	α	β
C8	28	18	70	103	89,5	22,0
DALU	75	16	194	1145	94,0	41,3
LAL	26	19	117	67	83,8	28,2
PM1	16	13	42	27	83,6	30,0
SCT	19	15	64	76	98,9	36,7
TTT2	24	21	222	203	80,7	28,1
Alu4	14	8	1 028	1 020	45,2	59,7
Apex5	117	88	1 227	1 142	95,0	5,7
I2c	147	142	1 357	1 251	98,6	2,2
X1	51	35	324	289	87,0	22,0

Окончание таблицы 5

Схема	n	m	k	d	α	β
$X3$	135	99	915	523	92,4	6,1
$X4$	94	71	371	277	94,5	9,2
$Bloki1$	15	16	355	506	58,1	28,5
$Bloki2$	15	16	90	101	64,2	24,6
$Pozd\ 1$	30	30	754	1 516	79,2	17,5
$Pozd\ 2$	30	30	605	1 805	75,2	22,2

Примеры $Bloki1$, $Bloki2$, $Pozd\ 1$, $Pozd\ 2$ – это специально сгенерированные «блочные» системы ДНФ. Остальные 12 примеров (табл. 5) – это разреженные системы ДНФ (*Combinational Multi-Level Examples*), взятые из библиотеки примеров *LGSynth91*. Исходные функциональные описания примеров $C8$, $DALU$, LAL , $PM1$, SCT , $TTT2$, $X1$, $X3$, $X4$ не содержат инверсий литералов входных переменных, т. е. троичные матрицы T^x для этих примеров содержат только 1 и «-». В табл. 5 используются следующие обозначения: n – число аргументов системы ДНФ булевых функций; m – число функций; k – число общих элементарных конъюнкций; d – число дизъюнкций в системе ДНФ булевых функций; α – разреженность (в процентах) троичной матрицы T^x ; β – разреженность (в процентах) булевой матрицы B^f .

Эксперименты. Эксперименты были организованы следующим образом. Сначала (кроме эксперимента 1) выполнялась *раздельная, совместная либо блочная* (комбинированная) логическая минимизация. *BDDI*-минимизация выполнялась для матричных форм, *Bool*-минимизация – для логических уравнений, задающих те же системы функций. Программы блочного покрытия выполнялись только для матричных представлений, обработка многоблочных логических сетей осуществлялась с помощью стратегий системы логической оптимизации *FLC2* [9].

Перечислим программы системы *FLC2*, участвующие в экспериментах. *BDDI*-минимизация отдельных ДНФ и совместная *BDDI*-минимизация систем ДНФ выполнялась с помощью программы *BDD_Builder* [5]; раздельная и совместная *Bool*-минимизация – с помощью модификации программы *BoolNet_Opt* [5]. Для решения задачи 1 блочного покрытия систем ДНФ использовалась программа *RAZ* [8], для решения задачи 2 – программа *RAZ_Area* [8]. Как уже говорилось, совместная минимизация систем булевых функций в классе ДНФ выполнялась программой *Espresso IIC* [6].

После логической минимизации минимизированные описания представлений систем функций конвертировались в *VHDL* описания в систему *Vivado*. Синтезатор системы *Vivado* имеет свои средства логической минимизации, этот синтезатор перерабатывает входное описание, получает свое (внутреннее) описание, по которому и синтезируется схема. В качестве *FPGA* использовалась микросхема *xc7k70tfbg676-1* семейства *Kintex-7* [10], эксперименты выполнялись в системе *Vivado* [1] (версия 2019.1), стратегия синтеза «*Vivado Synthesis Defaults*», стратегия имплементации – «*Vivado Implementation Defaults*».

Эксперимент 1. Для систем функций, заданных логическими уравнениями, логическая оптимизация не выполнялась, исходные описания сразу конвертировались в *VHDL* описания. Полученные схемные реализации названы базовыми. На рис. 1 показана логическая схема, полученная в эксперименте 1 по не минимизированной системе ДНФ (табл. 1) на этапе логического синтеза, после выполнения имплементации схема реализуется структурой из семи программируемых элементов *LUT(5)*, имеющих пять входов.

В экспериментах 2 – 10 исходной информацией были матричные формы систем ДНФ булевых функций.

Эксперимент 2. Совместная минимизация систем функций в классе ДНФ с помощью программы *Espresso ПС* [6].

Эксперимент 3. Разбиение системы на подсистемы, состоящие из отдельных функций, затем для каждой из функций исходной системы, заданной в матричной форме, выполнялась раздельная *BDDI*-минимизация.

Эксперимент 4. Разбиение системы на подсистемы, состоящие из отдельных функций, затем матричное описание каждой из функций переводилось в логические уравнения, после этого для каждой из функций выполнялась раздельная *Bool*-минимизация.

Эксперимент 5. Совместная *BDDI*-минимизация исходной системы ДНФ булевых функций.

Эксперимент 6. Матричное описание системы ДНФ булевых функций переводилось в логические уравнения, затем для системы функций, заданных логическими уравнениями, выполнялась совместная *Bool*-минимизация.

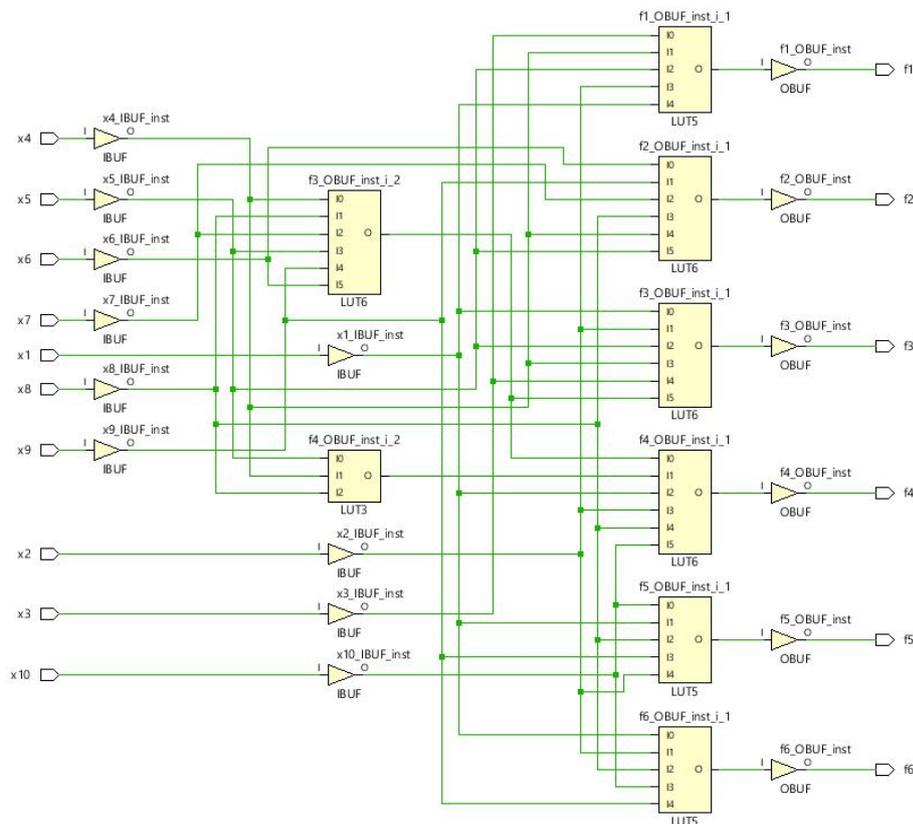


Рисунок 1. Реализация в *FPGA* после этапа логического синтеза системы ДНФ (табл.1) в эксперименте 1

Эксперимент 7. Блочное покрытие матричной формы системы ДНФ по критерию минимальности числа (p, q) -ограниченных блоков (p – число входов, q – число выходов блока), затем для функций каждого блока выполнялась совместная *BDDI*-минимизация.

Эксперимент 8. Блочное покрытие матричной формы системы ДНФ по критерию минимальности числа (p, q) -ограниченных блоков (p – число входов, q – число выходов блока), затем для каждого блока выполнялась совместная *Bool*-минимизация.

Эксперимент 9. Блочное покрытие матричной формы системы ДНФ по критерию минимальности суммарной площади блоков, затем для функций каждого блока выполнялась совместная *BDDI*-минимизация.

Эксперимент 10. Блочное покрытие матричной формы системы ДНФ по критерию минимальности суммарной площади блоков, затем для функций каждого блока выполнялась совместная *Bool*-минимизация. На рис. 2 показана логическая схема, полученная в эксперименте 10 по минимизированной (блочное покрытие, затем *Bool*-минимизация) системе ДНФ (табл. 1) на этапе логического синтеза, после выполнения имплементации схема реализуется структурой из шести программируемых элементов *LUT*(5).

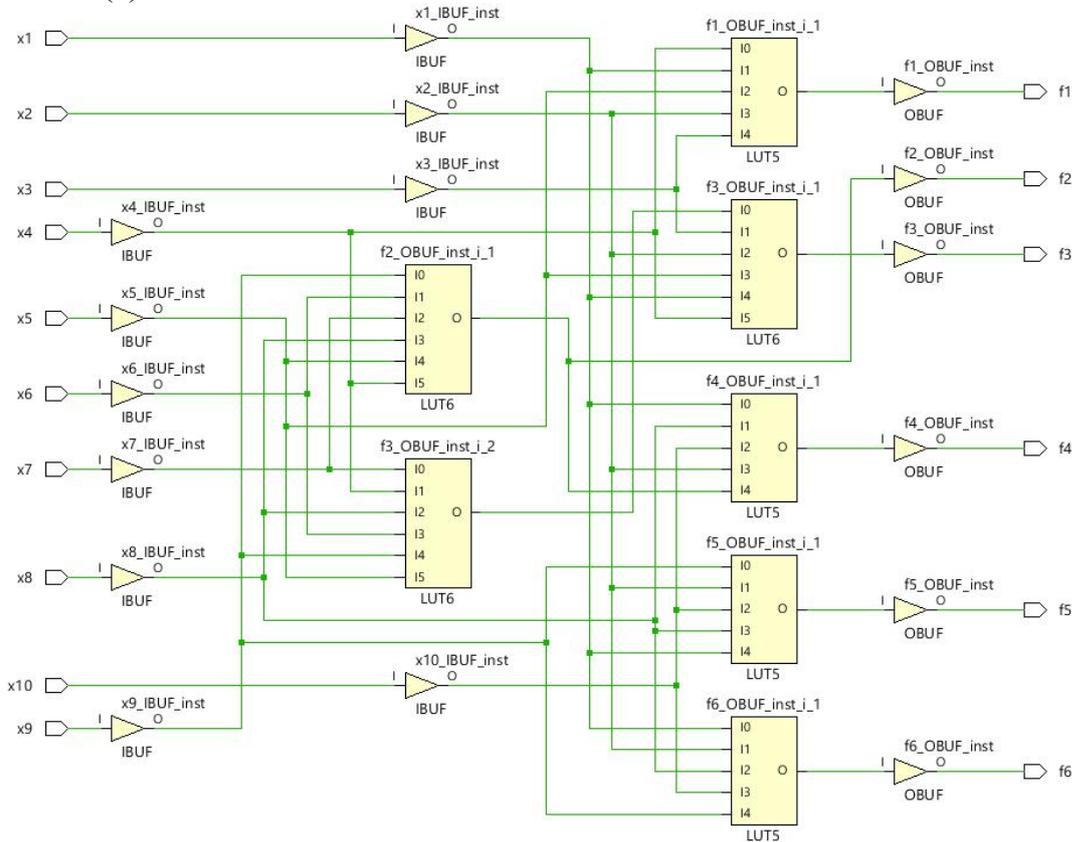


Рисунок 2. Реализация в *FPGA* логической сети (табл. 2- 4) в эксперименте 10

Результаты экспериментов и их обсуждение. Результаты экспериментов для примеров из табл. 5 представлены в табл. 6 – 9, где символом * отмечены лучшие решения для испытываемого примера – меньшие значения параметров площади и временной задержки, при этом сравнение проводилось по всем десяти экспериментам.

Символом # помечены решения, улучшающие базовые решения, т. е. отмечены меньшие значения параметра площади либо задержки, чем соответствующие значения параметров из эксперимента 1.

В табл. 6 – 9 используются следующие обозначения:

Z - число литералов в задании системы булевых функций;

k_{Espr} – число элементарных конъюнкций в совместно минимизированной системе ДНФ с помощью программы *Espresso*;

LUT – число программируемых элементов *LUT* (*Look-Up Table* (таблица, реализующая логическую функцию) – настраиваемый элемент, входящий в состав *FPGA*;

Delay – временная задержка схемы (нс);

p – число входов блока;

q – число выходов блока;

r – число блоков после разбиения системы функций на подсистемы (случай $r=1$ соответствует совместной реализации системы).

Таблица 6. Результаты экспериментов 1, 2

Имя примера	Эксперимент 1 Базовые решения		Эксперимент 2			
	<i>LUT</i>	<i>Delay</i>	<i>k</i>	<i>k_{Espr}</i>	<i>LUT</i>	<i>Delay</i>
<i>C8</i>	16	7.425	70	70	16	#7.217
<i>DALU</i>	*51	9.239	194	194	*51	*#8.680
<i>LAL</i>	28	7.753	117	117	28	8.012
<i>PM1</i>	12	6.909	42	42	12	6.988
<i>SCT</i>	21	7.330	64	64	#20	*#7.057
<i>TTT2</i>	37	7.909	222	222	37	#7.869
<i>Alu4</i>	266	10.786	1 028	575	273	11.638
<i>Apex5</i>	173	13.639	1 227	1 088	#169	14.158
<i>I2c</i>	255	14.559	1 357	805	#253	15.391
<i>X1</i>	73	*8.705	324	274	74	9.238
<i>X3</i>	*152	13.326	915	915	174	13.620
<i>X4</i>	*84	12.996	371	371	101	#12.201
<i>Bloki1</i>	279	10.307	355	240	*#266	#9.721
<i>Bloki2</i>	87	8.198	90	80	#82	8.510
<i>Pozd_1</i>	995	13.396	754	555	*#865	13.519
<i>Pozd_2</i>	977	*13.435	605	442	*#902	14.501
Улучшено базовых решений			-	-	8	6

Таблица 7. Результаты экспериментов 3, 4

Имя примера	Эксперимент 3		Эксперимент 4		
	<i>LUT</i>	<i>Delay</i>	<i>LUT</i>	<i>Delay</i>	
<i>C8</i>	*#15	#7.317	*#15	#7.157	
<i>DALU</i>	*51	#9.202	*51	#8.748	
<i>LAL</i>	30	8.056	28	7.837	
<i>PM1</i>	13	#6.837	12	7.003	
<i>SCT</i>	#18	7.810	#18	7.713	
<i>TTT2</i>	#36	8.279	#36	8.415	
<i>Alu4</i>	226	10.992	#200	*#9.533	
<i>Apex5</i>	#172	14.015	174	14.405	
<i>I2c</i>	255	*#13.766	#251	#14.321	
<i>X1</i>	73	8.860	#72	9.046	
<i>X3</i>	176	#13.279	176	14.116	
<i>X4</i>	102	#11.957	100	13.086	
<i>Bloki1</i>	#267	#9.914	#270	*#9.521	
<i>Bloki2</i>	#79	#7.803	#82	8.314	
<i>Pozd_1</i>	1 295	15.293	1 273	13.624	
<i>Pozd_2</i>	1 730	13.913	1 751	14.554	
Улучшено базовых решений		7	8	9	5

Таблица 8. Результаты экспериментов 5, 6, 7

Имя примера	Эксперимент 5		Эксперимент 6		Эксперимент 7	
	<i>LUT</i>	<i>Delay</i>	<i>LUT</i>	<i>Delay</i>	<i>LUT</i>	<i>Delay</i>
<i>C8</i>	16	7.503	16	7.549	*#15	#7.325
Имя примера	Эксперимент 5		Эксперимент 6		Эксперимент 7	
	<i>LUT</i>	<i>Delay</i>	<i>LUT</i>	<i>Delay</i>	<i>LUT</i>	<i>Delay</i>
<i>DALU</i>	*51	#9.075	52	9.305	*51	#9.235
<i>LAL</i>	30	*#7.547	*#26	7.874	#27	#7.593
<i>PM1</i>	13	#6.805	13	6.909	*#11	*#6.692
<i>SCT</i>	*#17	7.543	#20	#7.276	#18	7.366
<i>TTT2</i>	37	#7.833	37	8.616	*#35	8.323
<i>Alu4</i>	194	#9.917	#182	#9.816	262	11.224
<i>Apex5</i>	*#162	13.932	183	13.767	224	#13.324
<i>I2c</i>	255	14.321	#245	14.560	258	14.930
<i>X1</i>	#71	9.269	73	9.231	#70	9.198
<i>X3</i>	175	13.647	175	*#13.177	175	14.438
<i>X4</i>	100	#12.127	100	#11.941	101	#12.115
<i>Bloki1</i>	290	#9.965	285	#9.829	279	#9.905
<i>Bloki2</i>	#82	#7.844	#83	#8.019	*#73	8.225
<i>Pozd_1</i>	1 057	14.072	1 030	13.646	1 059	14.079
<i>Pozd_2</i>	1 176	14.435	1 036	14.233	1 172	14.280
Улучшено базовых решений	4	8	5	6	8	7

Таблица 9. Результаты экспериментов 8, 9, 10

Имя примера	Эксперимент 8		Эксперимент 9		Эксперимент 10	
	<i>LUT</i>	<i>Delay</i>	<i>LUT</i>	<i>Delay</i>	<i>LUT</i>	<i>Delay</i>
<i>C8</i>	*#15	#7.325	16	*#7.123	*#15	7.447
<i>DALU</i>	*51	#9.124	*51	9.422	*51	#8.908
<i>LAL</i>	28	#7.596	29	7.974	28	#7.628
<i>PM1</i>	*#11	#6.740	*#11	7.151	13	7.033
<i>SCT</i>	*#17	7.810	22	7.444	#18	7.704
<i>TTT2</i>	37	*#7.746	#36	#7.836	37	#7.815
<i>Alu4</i>	204	#9.593	226	10.992	*#181	#9.699
<i>Apex5</i>	180	15.126	#171	14.314	#172	*#13.201
<i>I2c</i>	261	#13.841	#254	14.598	*#242	#14.472
<i>X1</i>	*#69	8.969	#70	9.198	#72	9.153
<i>X3</i>	176	16.358	174	16.097	174	14.766
<i>X4</i>	101	#12.689	101	#12.337	101	*#11.715
<i>Bloki1</i>	282	#10.203	#269	#9.915	#270	10.339
<i>Bloki2</i>	#84	8.398	#80	*#7.791	#80	8.215
<i>Pozd_1</i>	1 011	*#13.022	1 088	14.597	1 036	14.528
<i>Pozd_2</i>	1 032	14.478	1 172	14.280	1 045	14.106
Улучшено базовых решений	6	10	8	5	9	7

Алгоритмы и программы [8] блочного покрытия для матричных представлений позволяют выделять блоки в блочных системах ДНФ функций и в разреженных системах ДНФ. Это подтверждается схемной реализацией специально сгенерированных блочных систем ДНФ. Анализ экспериментов, проведенных в системе в *Vivado*, приводит к следующим выводам.

Практически все программы предварительной технологически независимой оптимизации позволяют часто улучшать результаты схемной реализации в *FPGA* по сравнению с непосредственным (без оптимизации) базовым синтезом в *Vivado*, однако ни одна из программ предварительной логической минимизации не имеет явного преимущества по числу лучших результатов, помеченных символом «*». Наряду с совместной минимизацией в классе ДНФ, эффективной для специально сгенерированных блочных систем ДНФ и выполняемой программой *Espresso*, отдельная *BDDI*- и *Bool*-минимизация оказываются достаточно эффективными при синтезе схем *FPGA*. Однако выполнение блочных покрытий матричных представлений систем ДНФ функций и *Bool*-минимизация полученных блоков являются более эффективными, так как в большем числе случаев позволяют уменьшить площади схем либо их временные задержки. Результаты эксперимента 10 (табл. 9) показывают, что, как и в случае заказных СБИС, блочное покрытие с *Bool*-минимизацией блоков является наиболее эффективным при реализации разреженных матричных форм систем ДНФ полностью определенных булевых функций. Целесообразно в дальнейших исследованиях организовывать перебор блочных покрытий, оценивания их не только по площади, но и по суммарному числу литералов в блоках покрытия. Данный подход, использующий перебор блочных покрытий и их оценку по числу литералов, оказался эффективным при синтезе схем заказных СБИС, когда целевая библиотека синтеза состояла из КМОП элементов.

Заключение. Система *FLC-2* [9] включает разнообразные программы логической минимизации различных форм представлений систем булевых функций. В системе *FLC-2* можно провести эффективную технологически независимую оптимизацию разреженных систем ДНФ, используя программы блочного покрытия с последующей минимизацией *Bool*-представлений либо *BDDI*-представлений полученных блоков в зависимости от целевой библиотеки синтеза – библиотечные элементы либо программируемые элементы *FPGA*. Это не исключает применения и других программ логической минимизации, особенно тогда, когда системы ДНФ не являются разреженными. В этих случаях могут быть эффективными комбинированные методы многоэтапной совместной минимизации многоуровневых представлений, либо подходы, основанные на выделении из системы функций таких подсистем, для которых преимущество при синтезе имеет совместная минимизация.

Список литературы

- [1] Тарасов, И. Е. ПЛИС Xilinx. Языки описания аппаратуры VHDL и Verilog, САПР, приемы проектирования / И. Е. Тарасов. – М.: Горячая линия – Телеком, 2020. – 538 с.
- [2] Chen, D. *FPGA design automation: a survey* / D. Chen, J. Cong, P. Pan // *Foundations and Trends in Electronic Design Automation*. – 2006. – Vol. 1. – № 3. – P. 195–330.
- [3] Бибило П.Н., Ланкевич Ю.Ю., Романов В.И. Логическая минимизация при синтезе комбинационных структур в *FPGA* // *Информатика*. – 2021. – Т. 18. – № 1. – С. 7–24.
- [4] Бибило, П. Н. Использование полиномов Жегалкина при минимизации многоуровневых представлений систем булевых функций на основе разложения Шеннона / П. Н. Бибило, Ю. Ю. Ланкевич // *Программная инженерия*. – 2017. – № 8. – С. 369–384.
- [5] Бибило, П.Н. Экспериментальное сравнение эффективности алгоритмов оптимизации *BDD*-представлений систем булевых функций / П.Н. Бибило, Ю.Ю. Ланкевич // *Программные продукты и системы*. – 2020. – Т. 33. – № 3. – С. 449–463.
- [6] *Logic Minimization Algorithm for VLSI Synthesis* / K. R. Brayton [et al.]. – Boston: Kluwer Academic Publishers, 1984. – 193 p.
- [7] Закревский, А. Д. Логические основы проектирования дискретных устройств / А. Д. Закревский, Ю. В. Поттосин, Л. Д. Черемисинова. – М.: Физматлит, 2007. – 592 с.

[8] Кардаш, С. Н. Построение блочных разбиений систем булевых функций на основе задачи покрытия булевых матриц / С. Н. Кардаш // IX Международная научно-практическая конференция «BIG DATA and Advanced Analytics» (BIG DATA 2023): Материалы междунар. науч. конф., (Республика Беларусь, Минск, 17–18 мая 2023 г.). – Минск: БГУИР, 2023. – Часть 2. – С. 326 – 330.

[9] Бибило, П.Н. Система логической оптимизации функционально-структурных описаний цифровых устройств на основе продукционно-фреймовой модели представления знаний / П.Н. Бибило, В.И. Романов // Проблемы разработки перспективных микро- и нанoeлектронных систем. 2020. – Выпуск 4. – С. 9–16.

[10] Соловьев, В. В. Архитектуры ПЛИС фирмы Xilinx: FPGA и CPLD 7-й серии / В. В. Соловьев. – М.: Горячая линия – Телеком, 2016. – 392 с.

Авторский вклад

Бибило Петр Николаевич – постановка задачи исследования, проведение экспериментов в системах Vivado и FLC-2, подготовка текста доклада.

Кардаш Сергей Николаевич – разработка программ блочного покрытия и разбиения систем ДНФ булевых функций, исследование эффективности этих программ.

IMPLEMENTATION IN FPGA OF SPARSE SYSTEMS OF DISJUNCTIVE NORMAL FORMS OF BOOLEAN FUNCTIONS

P. N. Bibilo

*Head of Laboratory of UIIP of NAS of Belarus,
Dr. of Technical Sciences, Professor*

S. N. Kardash

*Senior Researcher of UIIP of NAS of Belarus,
PhD of Technical Sciences*

Abstract. The problem of choosing the best methods and programs for circuit implementation as part of FPGA sparse DNF systems of fully defined Boolean functions is considered. For matrix forms of sparse DNF systems, the ternary matrix specifying elementary conjunctions contains a large proportion of undefined values corresponding to missing literals of Boolean input variables, and the Boolean matrix specifying the occurrences of conjunctions in DNF functions contains a large proportion of zero values. For the FPGA circuits, the effectiveness of a combined approach is shown, which first uses block coverage programs of the DNF system, followed by the use of programs to minimize multilevel block representations in the form of Boolean networks minimized based on Shannon expansion.

Keywords: Boolean function system, Disjunctive Normal Form (DNF), DNF minimization, Binary Decision Diagram (BDD), Boolean network, Shannon expansion, block cover of the DNF system, logic synthesis, FPGA, VHDL.