Экспериментальное исследование эффективности программ минимизации BDD-представлений систем булевых функций при синтезе комбинационных схем из библиотечных КМОП элементов

Бибило П.Н., Ланкевич Ю.Ю.

Объединённый институт проблем информатики НАН Беларуси,

bibilo@newman.bas-net.by, yurafreedom18@gmail.com

Цели и задачи

Цель: исследовать эффективность отечественных САПР для предварительной оптимизации систем булевых функций при синтезе комбинационных схем из библиотечных КМОП элементов.

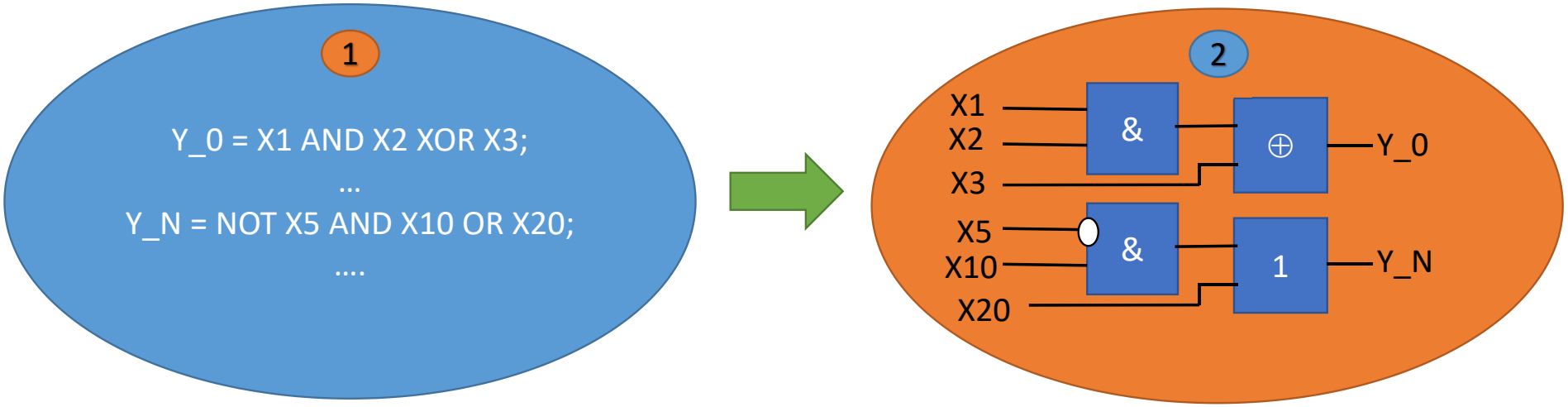
Задачи:

- оптимизировать набор исходных систем булевых функций с помощью программ минимизации BDD-представлений;
- используя программу минимизации булевых сетей, оптимизировать набор систем булевых функций, полученных с помощью программ минимизации BDD-представлений;
- используя *LeonardoSpectrum,* произвести синтез исходных систем булевых функций и систем, полученных при оптимизации с помощью программ минимизации BDD-представлений и программы минимизации булевых сетей;
- провести анализ полученных результаты синтеза.

Синтез схем комбинационной логики

Выделяют два больших этапа:

1. технологически независимую оптимизацию реализуемых систем булевых функций;
2. технологическое отображение.



Технологически независимая оптимизация систем булевых функций

Основными методами технологически независимой оптимизации являются методы, базирующиеся на разложении Шеннона.

В работе приводятся результаты экспериментального исследования эффективности применения двух оптимизационных процедур перед синтезом логических схем в *LeonardoSpectrum*:

- BDD-оптимизации функционального описания систем булевых функций с минимизацией числа коэффициентов (подфункций) разложений Шеннона.
- оптимизация булевых сетей, базирующаяся на разложении Шеннона и минимизации числа узлов булевой сети с поиском инверсных подфункций. Узлами сети являются простейшие булевы функции И, ИЛИ.

Разложение Шеннона

Разложением Шеннона полностью определенной булевой функции $f = f(\boldsymbol{x})$ по переменной x_i называется представление

$$f = f(\boldsymbol{x}) = \bar{x}_i f(x_1,\ldots,x_{i-1},0,x_{i+1},\ldots,x_n) \vee x_i f(x_1,\ldots,x_{i-1},1,x_{i+1},\ldots,x_n) \quad (1)$$

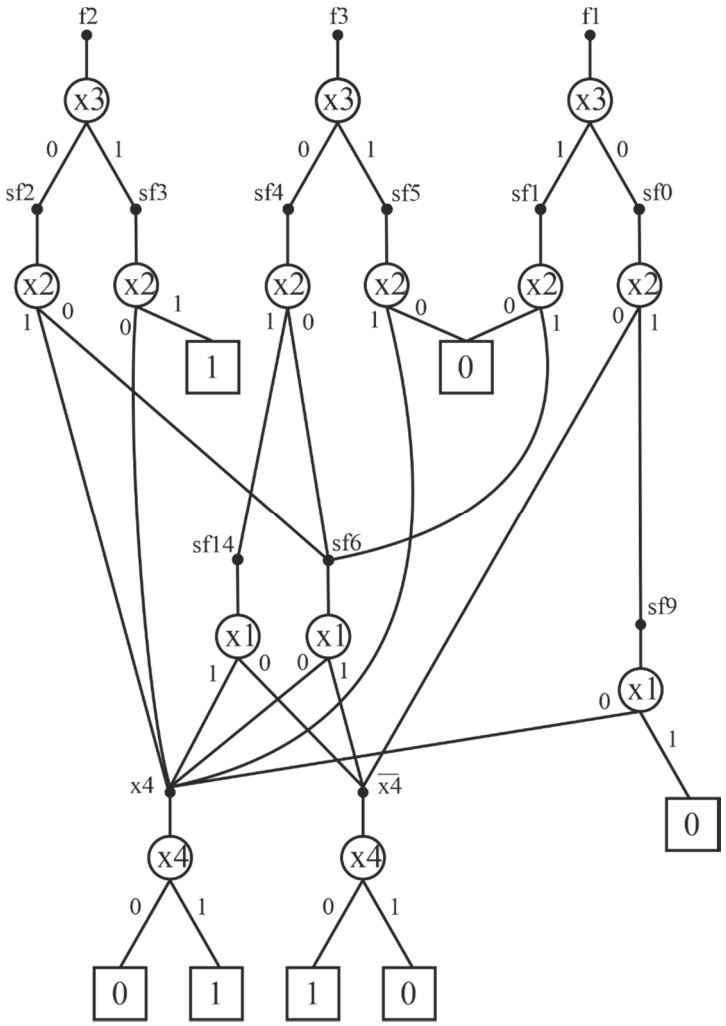
Функции $f_0 = f(x_1,\ldots,x_{i-1}, 0, x_{i+1},\ldots, x_n)$, $f_1 = f(x_1,\ldots,x_{i-1}, 1, x_{i+1},\ldots, x_n)$, в правой части (1) называются **коэффициентами** разложения по переменной x_i , либо просто **подфункциями**.

Binary Decision Diagram

Диаграмма двоичного выбора (Binary Decision Diagram – BDD) – графическое представление систем булевых функций на базе разложения Шеннона.

BDD используются для:

- решения SAT проблем;
- формальной верификации алгоритмических описаний цифровых систем;
- технологически независимой оптимизации.



BDD минимизация

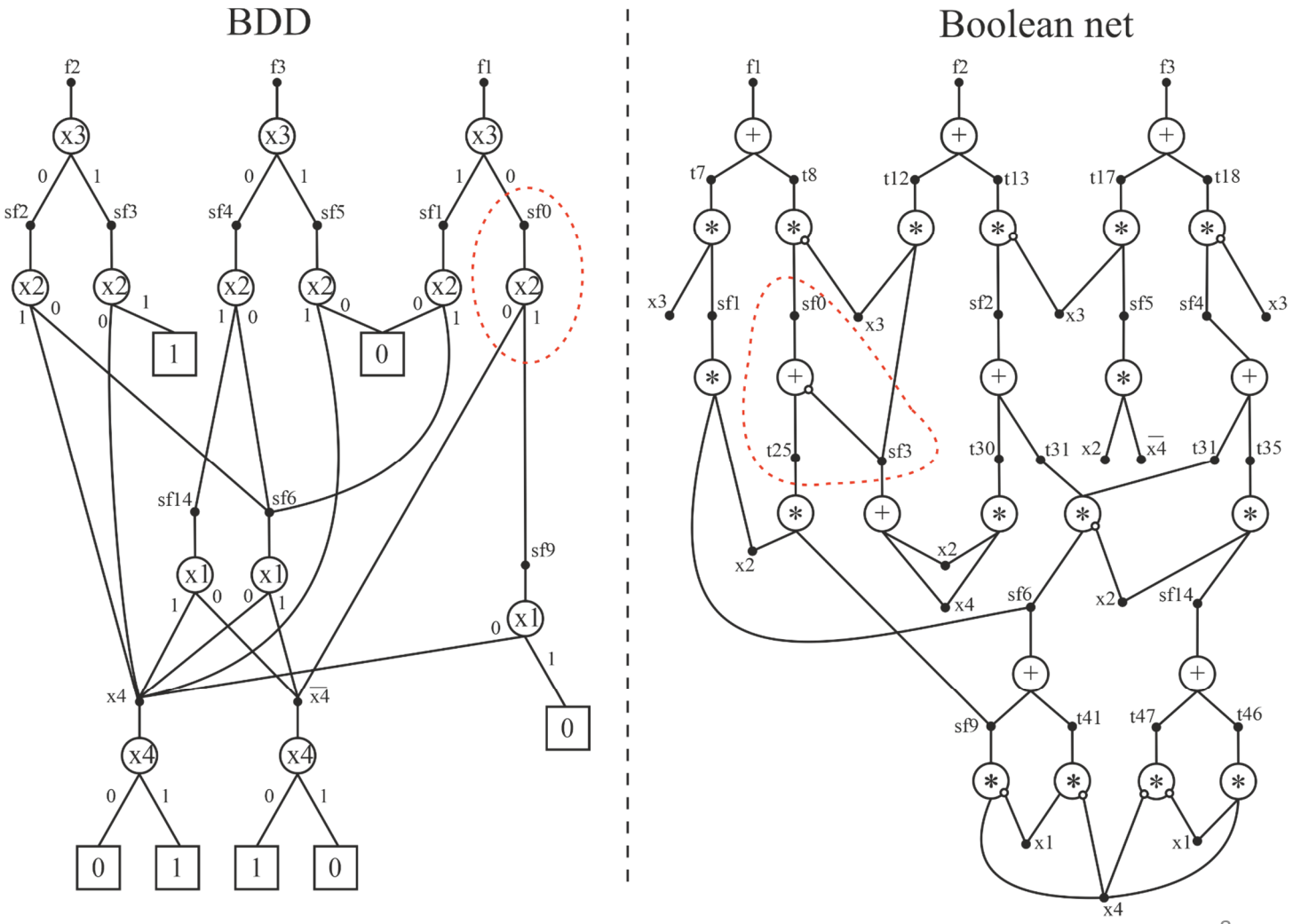
Каждая из подфункций разложения Шеннона

$$f_0 = f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) \text{ и } f_1 = f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)$$

может быть разложена по одной из переменных из множества $\{ x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n \}$. Процесс разложения подфункций заканчивается, когда все n переменных будут использованы для разложения, либо когда все подфункции выродятся до констант 0, 1. На каждом шаге разложения выполняется сравнение на равенство полученных подфункций (для BDDI учитывается равенство с инверсией, $f_i = f_j = \overline{f_k}$) и оставляется одна из нескольких попарно равных подфункций.

Соответствие булевой сети и BDD

Любой граф BDD можно представить в виде булевой сети, при этом булева сеть будет иметь большее число узлов, чем соответствующая ей BDD.



Логическая оптимизация булевых сетей, базирующаяся на разложении Шеннона

Каждая из подфункций

$$f_0 = f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) \text{ и } f_1 = f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)$$

может быть разложена по одной из переменных из множества $\{ x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n \}$.

На каждом шаге разложения выполняется сравнение на равенство, либо на равенство инверсий всех полученных **булевых функций** (**узлов**) в булевой сети. Выбор порядка следования переменных зависит от количества узлов, полученных при разложении по каждой из переменных (из всех возможных на данном шаге).

Процесс разложения подфункций заканчивается, когда все n переменных будут использованы для разложения, либо когда все подфункции выродятся до констант 0, 1.

Пример оптимизации булевой сети двухразрядного сумматора

На рисунке представлен пример оптимизации булевой сети двухразрядного сумматора:

s1 = **b1*****b̄2** + **b1*****b̄2**;

s2 = c̄1***ā1*****a2** + c̄1***a1*****ā2** +
+ c1***a1*****a2**;

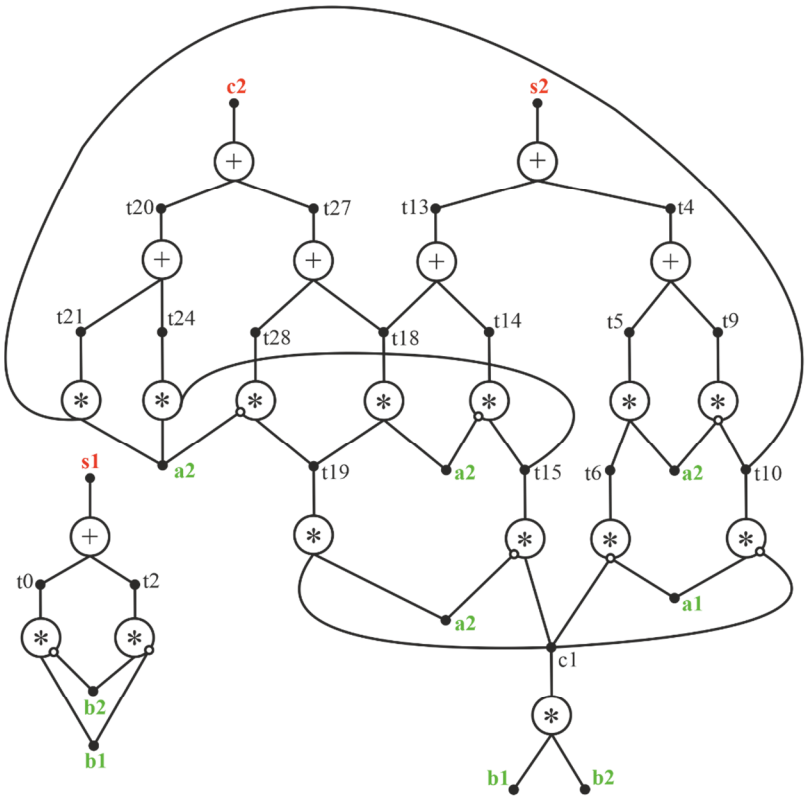
c2 = c̄1***a1*****a2** + c1***ā1*****a2** +
+ c1***a1*****ā2** + c1***a1*****a2**;

c1 = **b1*****b2**;

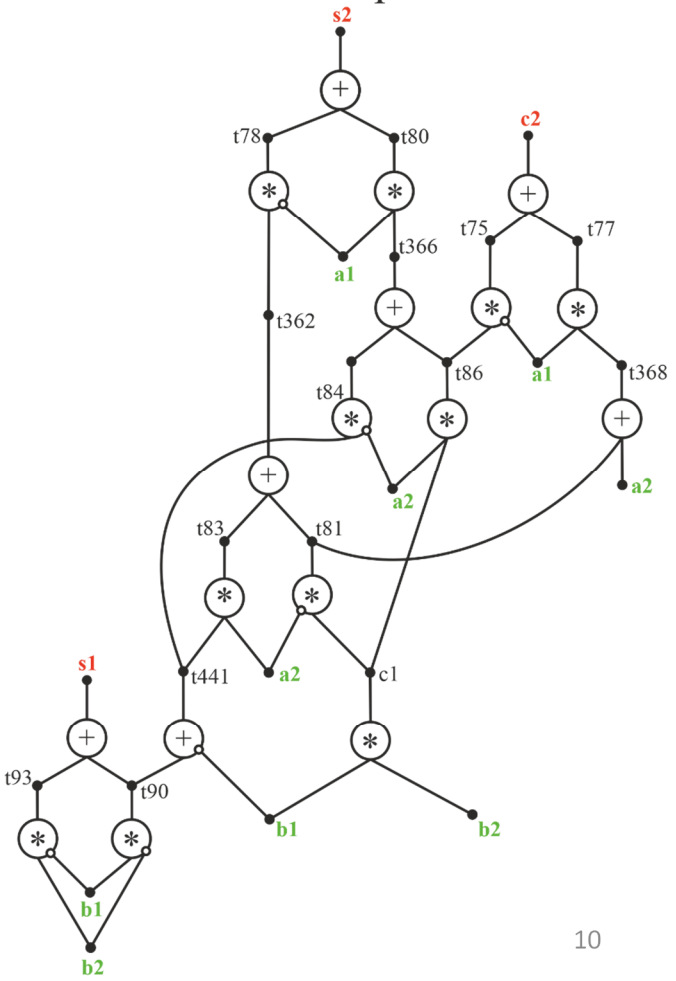
Исходная сеть содержит
7 дизъюнкций и
14 конъюнкций.

Оптимизированная сеть
содержит 7 дизъюнкций и
11 конъюнкций, что на
3 конъюнкции меньше.

Исходная сеть



Оптимизированная сеть

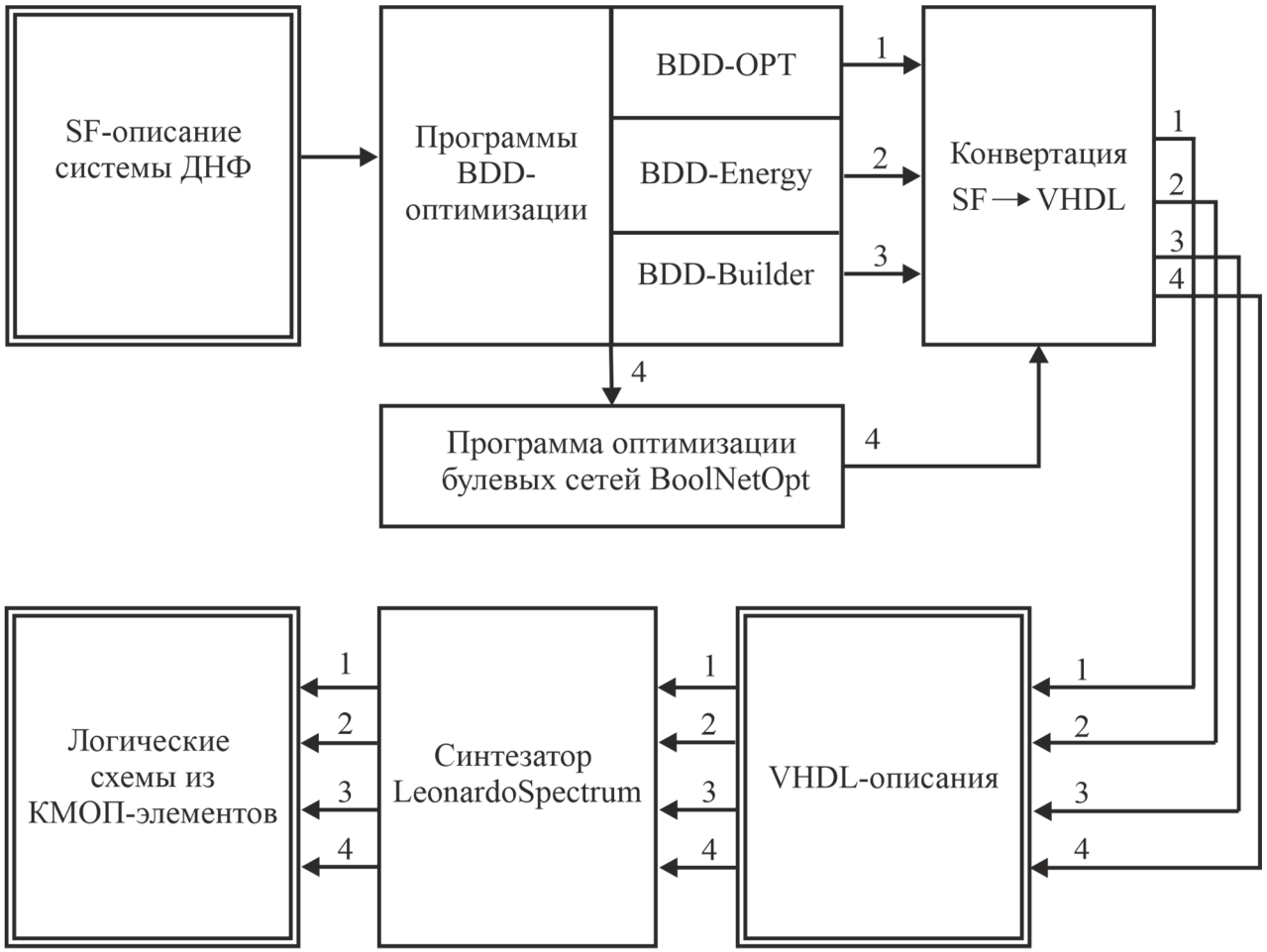


Программы минимизации систем булевых функций

Список используемых отечественных программ минимизации:

- BDD_Builder – минимизация BDD/BDDI
- BDD_OPT – минимизации BDD
- BDD_Energy – минимизация BDD
- BoolNetOpt – минимизация на основе булевых сетей

Маршрут экспериментов



Результаты экспериментов

S_{ASIC} - суммарная площадь элементов схемы в условных единицах;
τ - задержка схемы (нс).

После проведения синтеза выбирались те BDD описания, которые приводили к схемам меньшей площади, и для этих BDD-описаний выполнялась программа BoolNet_Opt оптимизации булевых сетей, после чего осуществлялся синтез схем. Результаты этого эксперимента приведены в таблице, где символом «*» помечается решение, полученное программой BDD_OPT.

Пример	Синтез по исходным описаниям		Лучшее схемное решение, полученное программами BDD-минимизации BDD_Builder, Tie_Energy, BDD_OPT		Улучшение решений BDD-минимизации программой BoolNet_Opt	
	S_{ASIC}	τ	Наименьшая площадь S_{ASIC},	Наименьшая задержка τ,	S_{ASIC}	τ
Rd73	21 684	**3.67**	**15 925**	3.94	16 428	3.89
Dc2	26 745	4.17	**21 505**	**3.33**	22 136	3.49
Dist	83 136	7.03	68 601	5.09*	**64 549**	**5.06**
M2	62 161	7.31	47 536	**4.25**	**45 583**	4.76
M3	82 227	7.09	58 813	**5.18**	**52 982**	5.73
Radd	12 092	**2.89**	8 465	3.34	**7 399**	3.99
Root	55 750	6.47	26 717	4.53*	**26 538**	**4.37**
Z9sym	59 896	9.90	15 909	**4.77**	**15 367**	4.95
Tial	303 100	8.01	**260 636***	**7.71**	294 858	9.80
Intb	382 844	9.26	**229 433**	8.27	297 888	**3.99**
B2	**159 834**	11.36	172 043	**7.84**	162 350	8.52
Sin_16	-	-	9 068 945*	**16.28***	**9 049 560**	16.30
Syst_4	185 206	7.64	**115 791**	**6.74***	125 879	7.17
Syst_8	-	-	7 961 142	18.49	**7 934 408**	**17.26**
Vtx1	**18 386**	**4.08**	25 015	4.37*	33 407	6.26
X9dn	**19 195**	5.24	24 312*	**5.00**	20 473	5.75
Too_large_matr	429 459	12.98	511 759	15.88	**416 575**	**12.55**
X1_matr	**67 312**	**3.99**	79 565	5.00	98 917	5.70
Dalu_matr	**49 673**	**4.22**	81 116	7.17	51 715	4.44
Soar	150 995	5.94	**135 298**	**5.67**	152 992	6.39
X3_matr	206 957	**6.69**	242 406	7.91	**205 428**	6.97
Frg2_matr	**406 062**	**10.83**	507 841	10.94	507 841	10.94

Анализ результатов

Из трех программ оптимизации BDD представлений 13 лучших результатов было получено с помощью BDD_Builder, 9 – с помощью BDD_Energy, 3 – с помощью BDD_OPT.

В 14 случаях из 20, удалось улучшить результаты синтеза с помощью отечественных программ оптимизации.

В 10 случаях из 22, удалось улучшить результаты синтеза с помощью дополнительной оптимизации используя программу на базе булевых сетей.

После оптимизации на базе BDD, площадь в среднем уменьшилась на 18,66%, а после дополнительной оптимизации на базе булевых сетей – на 20,1%.

Заключение

Исследованные программы технологически независимой оптимизации являются эффективными, прошли экспериментальную проверку на примерах схем практической размерности и включены в отечественные системы автоматизированного проектирования. Их использование позволяет во многих случаях улучшать результаты синтеза логических КМОП схем в промышленном синтезаторе *LeonardoSpectrum*.

Спасибо за внимание!