Эффективность проектирования заказных схем в синтезаторе LeonardoSpectrum

Николай Авдеев, Пётр Бибило (г. Минск, Беларусь)

В статье сравнивается эффективность результатов проектирования заказных СБИС в различных версиях синтезатора LeonardoSpectrum. Эффективность определяется уменьшением площади и увеличением быстродействия синтезируемых схем. Сравнение версий проведено на примерах VHDL-описаний проектов цифровых схем. Установлено, что первые версии программы были ориентированы на получение схем меньшей площади, более поздние – на уменьшение задержки; в последних версиях достигается компромисс между этими характеристиками СБИС.

Введение

После этапа моделирования и верификации проекта цифровой системы (схемы) наступает решающий этап проектирования – синтез логических схем в выбранной целевой технологической библиотеке. Для реализации схемы на программируемых логических интегральных схемах (ПЛИС) применяются различные пакеты программ: XST в системе проектирования ISE (Xilinx), Sinplify (Synopsys), LeonardoSpectrum [1] (Mentor Graphics).

Скрипт для выполнения синтеза схемы b2 в эксперименте 1

```
clean_all
set PROJECT "b2"
set ENTITY_NAME b2
set VHDL_FILE_NAMES "b2.vhd"
set_working_dir $PROJECT
# чтение проекта
read -design $ENTITY_NAME \
  $VHDL_FILE_NAMES
# чтение библиотеки power
load_library power.syn
# оптимизация и синтез
optimize -target power -macro \
  -area -effort standard \
  -hierarchy flatten
optimize_timing
report_area   -cell_usage
report_delay  -num_paths 1 \
  -critical_paths \
  -clock_frequency
# write netlist
write b2_area_standard_sch.vhd ;
```

Таблица 1. Примерная эквивалентность версий синтезатора LeonardoSpectrum

Версия	Условное название для эквивалентных версий
2003b.35	L2003
2004a.30	L2006
2005a.82	
2006a.59	
2007a.37	L2011
2008a.5	
2009a.6	
2010a.7	
2011a.4	

Синтезатор LeonardoSpectrum используется, в том числе, и для проектирования заказных СБИС. Пользователь может выполнить синтез в собственной технологической библиотеке, подготовив описания элементов целевой библиотеки, либо осуществлять управление процессом синтеза, выбирать критерии и режимы оптимизации схемы и устанавливать технологические ограничения, например требование к задержке. Программа может работать как в ОС Linux, так и в ОС Windows. На настоящий момент сменилось большое число её версий.

Для пользователей LeonardoSpectrum актуальным является вопрос о том, *насколько повышается качество (уменьшается площадь, повышается быстродействие) получаемых логических схем со сменой версий программы*. С этой целью в статье проанализирован опыт работы с различными версиями синтезатора LeonardoSpectrum и приведены результаты экспериментов практических примеров.

Примеры и эксперименты

Все примеры представлены в виде функциональных либо алгоритмических описаний на языке VHDL. Первую группу составляют функциональные описания комбинационной логики, соответствующие представлениям логических функций в виде систем ДНФ (двухуровневые И/ИЛИ представления) и в виде многоуровневых представлений в базисе И, ИЛИ, НЕ. Данные примеры взяты из источника [2].

Другую группу примеров составляют алгоритмические описания проектов цифровых систем, взятые из практики проектирования либо из Интернета [3] (примеры dec_8b10b, enc_8b10b). Во всех экспериментах была использована одна и та же библиотека проектирования отечественных КМОП СБИС [4], содержащая 35 элементов типа 2И-НЕ, 3И-НЕ, 2-ИЛИ-НЕ, 3-ИЛИ-НЕ (и других) и три типа триггеров.

Эксперимент 1 проводился с целью выбора критерия оптимизации (*Area, Delay*) и режима трудоёмкости оптимизации (*Optimize effort*), при которых синтез приводил к получению схем, меньших по площади и задержке. Устанавливаемый в программе параметр режима «Optimize effort» может принимать значения *remap*, *quick*, *standard* [5].

Смысл этих значений следующий:

- *remap* выполняет локальную оптимизацию и технологическое отображение в целевую библиотеку;
- *quick* (режим по умолчанию) выполняет одну итерацию оптимизации на основе быстрых алгоритмов и применяется на начальном этапе проектирования;
- *standard* выполняет несколько итераций оптимизации и технологического отображения на основе различных методов, при этом получаемые результаты, как правило, лучше, чем в режиме *quick*.

Синтезатором можно управлять как с помощью графического интерфейса, так и с помощью TCL-скриптов. В листинге приведён скрипт (сценарий), используемый при синтезе в эксперименте 1, строка комментариев начинается с символа #. При проведении эксперимента 1 в скрипте вместо параметра *area* команды *optimize* использовался параметр *delay*, а вместо значения *standard* параметра *effort* использовались также значения *remap* либо *quick*.

По результатам сравнения синтезированных проектов схем версии программы LeonardoSpectrum можно условно поделить на три группы – L2003, L2006

и L2011, дающие примерно одинаковые результаты синтеза. Основанием для такого деления являются существенные различия в площади и задержке получаемых логических схем (см. табл. 1).

Результаты первого эксперимента по сравнению трёх базовых версий L2003, L2006, L2011 приведены в таблице 2, где знаком «*» отмечены лучшие результаты – параметры площади (в условных единицах) и задержки (в наносекундах) отдельно для каждой из версий. Жирным шрифтом выделены лучшие решения совместно по трём версиям синтезатора. В столбце «Критерий оптимизации» приведён критерий *Area* или *Delay*, в столбце «Трудоёмкость оптимизации» указано значение этого параметра, при котором получена схема меньшей площади либо с меньшей задержкой.

Таблица 2. Результаты эксперимента 1

Вид проекта	Название схемы	Критерий оптимизации	Трудоёмкость оптимизации	L2003		L2006		L2011		Сравнение с L2011			
				S	τ	*S*	τ	*S*	τ	S_{L2003}/S_{L2011}	$\tau_{L2003}/\tau_{L2011}$	S_{L2006}/S_{L2011}	$\tau_{L2006}/\tau_{L2011}$
Комбинационная логика	b2	area	standard	***141581**	13,38	*211588	11,10	*158801	11,36	0,89	1,18	1,33	0,98
		delay	quick	313892	*9,83	471711	***6,53**	313892	*9,83	1,00	1,00	1,57	0,73
	Chkn	area	quick	106667	*7,48	119685	7,28	106667	7,48	1,00	1,00	1,12	0,97
		area	standard	***83934**	11,47	*96556	9,98	*92823	*6,70	0,90	1,71	1,04	1,49
		delay	quick	119747	9,10	132397	***6,53**	119747	9,10	1,00	1,00	1,12	0,76
	dk48	area	quick	30082	7,44	41510	6,06	*30082	7,44	1,00	1,00	1,38	0,81
		area	standard	***30043**	7,54	*41404	6,84	*30082	7,44	1,00	1,01	1,38	0,92
		delay	quick	34144	*6,88	45477	***6,01**	34144	*6,88	1,00	1,00	1,19	0,98
		delay	standard	33240	7,43	42140	6,80	34144	*6,88	1,00	1,00	1,33	0,87
	FRG2	area	standard	***168522**	7,64	*196689	6,73	*250112	*7,56	0,67	1,01	0,79	0,89
		delay	standard	188554	*6,89	255430	***5,02**	270809	8,41	1,00	1,00	1,30	0,66
	l8	area	quick	159292	7,82	264726	5,81	*159292	*7,82	1,00	1,00	1,66	0,74
		area	standard	***118692**	7,32	*160012	5,73	*159292	*7,82	0,75	0,94	1,00	0,73
		delay	standard	126884	*6,72	206187	***4,82**	176992	8,71	1,00	1,00	2,06	0,68
	lbm	area	quick	42654	5,18	*51498	5,10	*42654	5,18	1,00	1,00	1,21	0,98
		area	standard	***41231**	6,61	54126	5,97	*42654	5,18	0,97	1,28	1,27	1,15
		delay	remap	97103	*4,87	122643	***3,88**	97103	*4,87	0,97	1,28	1,27	1,15
	in0	area	remap	219707	*8,09	291081	7,77	219707	*8,09	1,00	1,00	1,32	0,96
		area	standard	***115964**	13,27	*147624	12,91	*131509	8,83	0,88	1,50	1,12	1,46
		delay	remap	260480	8,93	349849	***6,80**	260480	8,93	0,88	1,50	1,12	1,46
	in2	area	remap	213005	*7,60	271221	6,60	213005	7,60	1,00	1,00	1,27	0,87
		area	standard	***102131**	9,36	*113944	9,63	*117448	*6,91	0,87	1,35	0,97	1,39
		delay	quick	172974	8,63	192605	***5,52**	172974	8,63	1,00	1,00	1,22	0,79
	prom2	area	quick	602791	*11,50	*721873	9,70	602791	11,50	1,00	1,00	1,20	0,84
		area	standard	***555656**	13,31	723386	11,52	*556309	*11,41	1,00	1,17	1,30	1,01
		delay	quick	688198	11,54	837469	***8,02**	688198	11,54	1,00	1,00	1,03	0,73
	Tial	area	quick	323690	*8,37	348125	7,75	323690	8,37	1,00	1,00	1,08	0,93
		area	standard	***276456**	12,59	*301186	11,41	*303100	*8,01	0,91	1,57	0,99	1,42
		delay	quick	359185	9,21	372671	***7,00**	359185	9,21	1,00	1,00	1,06	0,74
Не полностью определённые булевы функции	verg_1	area	standard	***370250**	16,83	*520129	13,58	*446880	13,62	0,83	1,24	1,16	1,00
		delay	quick	980389	*13,59	1334998	***10,83**	980389	*13,59	–	–	–	–
	verg_2	area	quick	972505	*15,56	1276230	13,89	972505	15,56	1,00	1,00	1,31	0,89
		area	standard	***540981**	28,09	*762881	24,35	*658713	*15,02	0,82	1,87	1,16	1,62
		delay	quick	1048705	16,79	1443490	***12,95**	1048705	16,79	–	–	–	–
Алгоритмические описания	Belts	area	quick	4934277	*65,26	34493590	70,44	4934277	65,26	1,00	1,00	6,99	1,08
		area	standard	***4577274**	121,35	*12757498	99,71	*4650707	77,45	0,98	1,57	2,74	1,29
		delay	standard	4885580	76,33	24322121	***66,21**	5053415	***59,18**	1,00	1,00	69,33	-
	dec_8b10b	area	quick	55047	5,60	56977	5,28	*55047	5,60	1,00	1,00	1,04	0,94
		area	standard	***54388**	5,77	*56140	5,54	*55047	5,60	0,99	1,03	1,02	0,99
		delay	remap	62468	*4,36	64577	***4,36**	62468	***4,36**	0,99	1,03	1,02	0,99
	enc_8b10b	area	quick	81859	7,76	99486	7,23	*81859	*7,60	1,00	1,02	1,22	0,95
		area	standard	***81016**	*7,47	*95937	6,69	*81859	*7,60	0,99	0,98	1,17	0,88
		delay	standard	85168	7,57	116103	***6,30**	85530	8,08	1,00	1,00	1,32	0,85
	Main	area	quick	1047578	44,17	*1524785	40,59	*1052433	*44,28	1,00	1,00	1,45	0,92
		area	standard	***1017636**	*44,00	1695656	40,34	*1052433	*44,28	0,97	0,99	1,61	0,91
		delay	standard	1093786	45,59	6715547	***37,06**	1125146	53,11	1,00	0,99	4,73	0,76
	timer2	area	remap	579472	*11,60	–	–	579472	*11,60	1,00	1,00	–	–
		area	quick	***533052**	11,91	655672	12,46	***533052**	11,91	1,00	1,00	1,23	1,05
		area	standard	534268	11,96	*611473	12,01	***533052**	11,91	1,00	1,00	1,15	1,01
		delay	standard	554462	12,17	755019	***11,01**	556454	11,77	1,00	1,00	1,87	1,01
	Uart	area	quick	409656	9,55	477821	9,17	***409656**	9,55	1,00	1,00	1,17	0,96
		area	standard	***380456**	9,55	*457861	9,08	***409656**	9,55	0,93	1,00	1,12	0,95
		delay	remap	430938	*8,12	3599764	8,51	430938	*8,12	0,93	1,00	1,12	0,95
		delay	quick	420581	8,27	569389	***7,61**	420581	8,27	1,00	1,00	8,35	1,05
	vga80x40	area	quick	***170290**	13,82	*171920	13,82	***170290**	13,82	1,00	1,00	1,01	1,00
		area	standard	172455	13,82	174989	13,82	***170290**	13,82	1,01	1,00	1,03	1,00
		delay	quick	198737	*12,50	203162	***12,50**	198737	***12,50**	1,00	1,00	1,04	1,00
		delay	standard	196371	*12,50	201756	***12,50**	198737	***12,50**	1,00	1,00	1,02	1,00
	Watchdog	area	quick	***105657**	13,06	109496	12,30	105657	13,06	1,00	1,00	1,04	0,94
		area	standard	***105657**	13,06	*108732	12,30	***105629**	13,15	1,00	0,99	1,03	0,94
		delay	quick	108804	*10,88	112789	***10,12**	108804	*10,88	1,00	1,00	1,04	0,94
		delay	standard	109100	*10,88	112409	***10,12**	108760	10,98	1,00	1,00	1,04	0,93
Количество лучших решений				**19**	**3**	**0**	**21**	**5**	**4**				

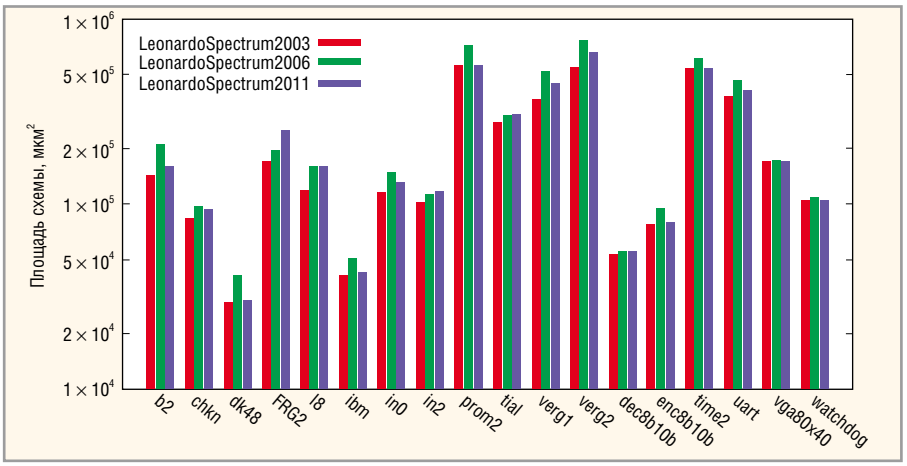

Рис. 1. Сравнение площадей схем, полученных в различных версиях синтезатора LeonardoSpectrum

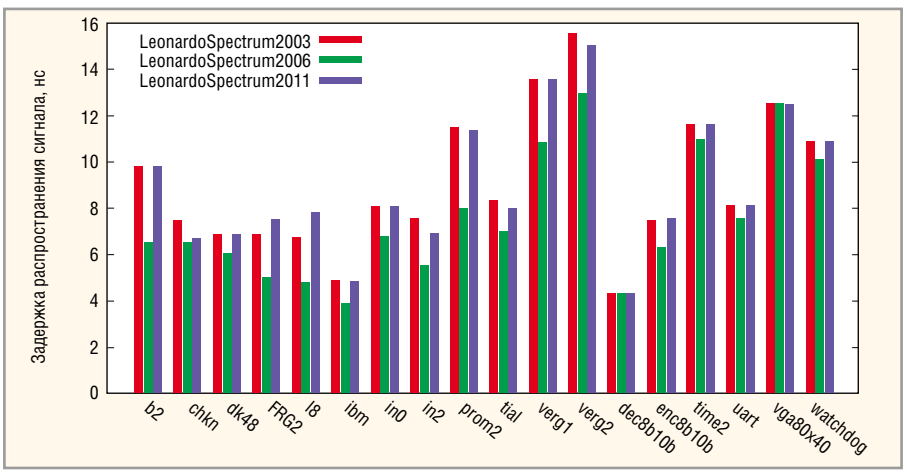

Рис. 2. Сравнение задержек, полученных в различных версиях синтезатора LeonardoSpectrum

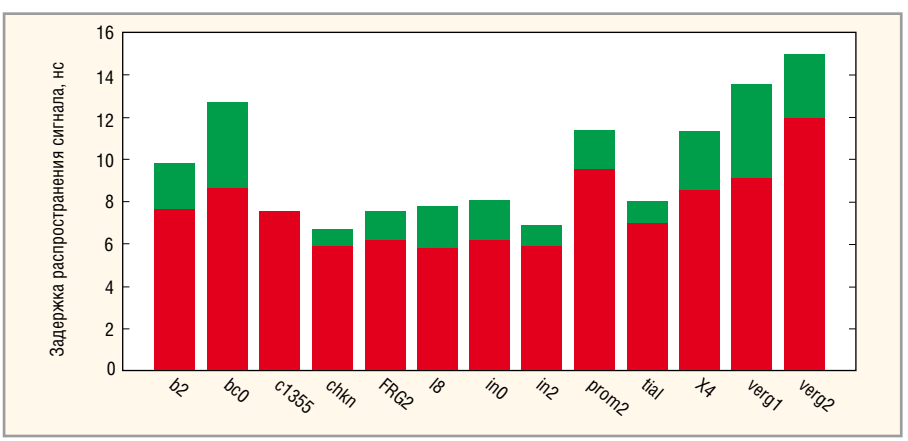

Рис. 3. Диаграмма уменьшения задержек в эксперименте 2, полученных в версии LeonardoSpectrum-2011

Проанализировав таблицу 2, можно сделать следующие выводы. Для версии L2006 выбор критерия оптимизации *Area* приводил к схеме меньшей площади, а выбор критерия *Delay* – к схеме с меньшей задержкой. Например, для первого примера b2 выбор режима <Area, Standard> позволил реализовать схему с площадью S_{L2006} = 211588 и задержкой τ_{L2006} = 11,10 (см. первую строку таблицы 2). В режиме <Delay, Quick> результат уже другой: S_{L2006} = = 471711 и τ_{L2006} = 6,53 (см. вторую строку таблицы 2). Звёздочкой в первой и второй строках отмечены меньшие значения площади и задержки. Для этого примера выбор режимов <Delay, Standard>, <Delay, Remap> и др. приводит к схемам с большей задержкой. Однако лучшее (по площади) решение для примера b2 обеспечила версия L2003, поэтому значение S_{L2003} = 141581 отмечено звёздочкой и выделено жирным шрифтом для трёх версий. Наименьшее значение задержки для данной схемы обеспечила версия L2006, поэтому значение τ_{L2006} = 6,53 выделено жирным шрифтом для трёх версий.

Для версий программы L2003, L2006 и L2011 критерий оптимизации *Auto* приводил к тем же результатам, что критерий *Area*, поэтому критерий *Auto* далее не упоминается.

Таким образом, версия L2003 позволяет получать схемы наименьшей площади при тех же режимах синтеза, что в версиях L2006 и L2011. Версия L2006 реализует схемы с наименьшей задержкой. Сравнения версий LeonardoSpectrum приведены на диаграммах: площади схем показаны на рисунке 1, задержки – на рисунке 2. Видно, что версия L2011 позволяет получать компромиссные схемы. Если же требуется реализовать схему с меньшей задержкой, то следует воспользоваться версией L2006, понимая при этом, что проигрыш по площади может быть весьма значительным.

Вернёмся к первому примеру b2 (см. табл. 2). Здесь версия L2006 позволяет уменьшить задержку до τ_{L2006} = 6,53 нс и получить площадь S_{L2006} = 471711, версия L2011 позволяет уменьшить площадь до S_{L2011} = 158801 и получить задержку τ_{L2011} = 11,36 нс.

Эксперимент 2 проводился с целью изучения эффективности повторного синтеза для уменьшения задержки схемы. Для этого в скрипт (см. листинг) перед командой *optimize* добавлена команда *set input2output 7.73*, устанавливающая требуемое значение задержки (т.е. 7,73 нс). Каждый раз оно устанавливалось на 10% меньше достигнутого, и синтез выполнялся снова. Однако на втором либо третьем проходе, как правило, уменьшение задержки не происходило, а увеличивалась площадь схемы.

Пользователь может устанавливать требуемое значение задержки, но программа часто синтезирует схему, задержка которой лишь приближена к желаемому значению. Для примера b2 площадь S_{L2011} = 313892, а задержка схемы составляет τ_{L2011} = 9,83 нс. Требование уменьшения задержки до τ = 6,16 нс привело к синтезу схемы с площадью S_{L2011} = 414399 и задержкой τ_{L2011} = 7,70 нс. Таким образом, уменьшение задержки схемы на 27% увеличило её площадь на 25%. На рисунке 3 показана диаграмма уменьшения задержек схем в эксперименте 2 для версии L2011 (зелёным цветом отмечено относительное уменьшение задержки).

Таким образом, после последовательного выполнения двух требований уменьшения задержки получаемой схе-

мы на 10%, задержка исходной схемы снижалась примерно на 18%, при этом минимальное уменьшение задержки составляло 2%, а максимальное – 40%.

Следует отметить, что требование получения схемы с малой задержкой при проектировании в целевой библиотеке заказных СБИС часто вообще не может быть выполнено – схемные реализации с требуемыми характеристиками задержек просто не существуют, поэтому никакие ухищрения и варьирование параметров синтеза не помогают.

Эксперимент 3 был посвящён сравнению двух форм представления систем логических функций: дизъюнктивных нормальных форм (ДНФ) и бинарных диаграмм решений (Binary Decision Diagram, BDD) [6]. На потоке из 62 практических примеров было установлено, что наиболее эффективной формой представления систем функций при синтезе схем из библиотечных элементов являются логические уравнения, соответствующие представлениям систем функций в виде BDD.

Диаграммы решений строятся на основе разложения Шеннона. Разложением Шеннона полностью определённой булевой (логической) функции $f(x_1,...,x_n)$ по переменной x_i называется представление $f(x_1,...,x_n)$ в виде:

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

Функции $f(x_1,..., x_{i-1}, 0, x_{i+1},..., x_n)$ и $f(x_1,..., x_{i-1}, 1, x_{i+1},..., x_n)$ в (1) являются коэффициентами разложения, они получаются из функции $f(x_1,...,x_n)$ подстановкой вместо переменной x_i константы 0 или 1, соответственно. Диаграмма задаёт в виде графа последовательность разложений Шеннона исходной функции и получаемых коэффициентов разложения. Минимизация сложности BDD основана на том, что в процессе разложения системы функций могут появляться одинаковые коэффициенты разложения не только у одной, но и у нескольких (либо даже у всех) функций, входящих в систему. Выделение одинаковых подфункций (коэффициентов разложения) приводит к сокращению аппаратной сложности и, соответственно, площади схемы. Минимизация BDD осуществлялась с помощью программы TIE_BDD [7].

Приведём пример многоуровневого разложения, соответствующего BDD. Обозначим через $<x_1, x_2, x_3, x_4, x_5, x_6>$ первую из $n!$ перестановок переменных, по которой проведём разложение Шеннона для системы ДНФ функций:

$f^1 = x_1x_2\bar{x}_4x_5\bar{x}_6 \vee \bar{x}_1x_4\bar{x}_5x_6 \vee x_2\bar{x}_3x_5;$

$f^2 = \bar{x}_1\bar{x}_4x_5x_6 \vee \bar{x}_1\bar{x}_3x_5 \vee x_1x_2x_3x_5\bar{x}_6 \vee \vee x_1\bar{x}_2x_4\bar{x}_5x_6;$

$f^3 = x_1\bar{x}_2\bar{x}_3x_6 \vee x_1\bar{x}_2x_4x_6 \vee x_1\bar{x}_3x_4x_6 \vee \vee \bar{x}_1x_2\bar{x}_4x_5\bar{x}_6 \vee x_1\bar{x}_2x_5 \vee x_2\bar{x}_3x_5.$

Построенные для данной системы функций коэффициенты разложения Шеннона дают следующее многоуровневое представление системы функций:

$f^1 = \bar{x}_1\psi^1 \vee x_1\psi^2; f^2 = \bar{x}_1\varphi^2 \vee x_1\psi^3;$

$f^3 = \bar{x}_1\psi^2 \vee x_1\psi^4;$

$\psi^1 = \bar{x}_2\varphi^1 \vee x_2\varphi^1; \psi^2 = x_2\varphi^2; \psi^3 = \bar{x}_2s^1 \vee x_2\varphi^3;$

$\psi^4 = \bar{x}_2\varphi^4 \vee x_2\varphi^5; \varphi^1 = \bar{x}_3s^2 \vee x_3s^1;$

$\varphi^2 = \bar{x}_3\lambda^3 \vee x_3s^3; \varphi^3 = x_3\lambda^4; \varphi^4 = \bar{x}_3\lambda^2 \vee x_3s^2;$

$\varphi^5 = \bar{x}_3s^2; s^1 = x_4\lambda^1; s^2 = \bar{x}_4\lambda^3 \vee x_4\lambda^2; s^3 = \bar{x}_4\lambda^4;$

$\lambda^1 = x_5\omega^1; \lambda^2 = \bar{x}_5\omega^1 \vee x_5; \lambda^3 = x_5; \lambda^4 = x_5\omega^2;$

$\omega^1 = x_6; \omega^2 = \bar{x}_6.$

Основной проблемой при построении многоуровневых представлений меньшей сложности является выбор перестановки переменных, которая приводит к возможно меньшему числу логических выражений приведённого выше вида. В экспериментах программа TIE_BDD строила BDD по 5000 случайно выбираемым перестановкам переменных и выбирала из рассмотренных вариантов диаграммы наименьшей сложности.

Эксперименты на практических примерах [2] комбинационных схем показали, что в подавляющем большинстве случаев (41 пример из 62) использование многоуровневых представлений вместо ДНФ позволяло синтезировать в версии L2011 схемы меньшей площади, чем по исходным описаниям функций, соответствующих ДНФ. Напомним, что при синтезе функциональные описания комбинационных схем в виде ДНФ и в виде разложений Шеннона были представлены на языке VHDL. Некоторая информация об алгоритмах оптимизации, применяемых синтезатором LeonardoSpectrum, содержится в [5, с. 122]. Таким образом, судя по результатам эксперимента 3, комбинационные схемы в библиотечном базисе лучше синтезировать по минимизированным BDD, а не по минимизированным системам ДНФ, представляющим логические функции.

Возможность уменьшения площади комбинационных схем имеется и при повторном синтезе описания, полученного с помощью команды *unmap*. Такие эксперименты с повторным синтезом, выполнением команды *unmap* и сменой целевых библиотек синтеза описаны в [8]. Стили кодирования (Encoding Style) не имеют значения для комбинационной логики, однако для схем с триггерами они позволяют варьировать параметры площади и быстродействия. С другими управляющими параметрами синтеза LeonardoSpectrum можно ознакомиться в [1].

Выводы

На испытанном потоке проектов схем наилучшие решения по площади давала версия L2003, по задержке – версия L2006. Новые версии программы L2011 ориентированы на уменьшение площади схем, при этом задержки, естественно, увеличиваются. В версиях L2003 и L2006 установка критерия оптимизации «Delay» приводила к схемам с минимальной задержкой, однако для более поздних версий L2011 установка критерия «Delay» практически не влияла на получение схемы с меньшей задержкой; минимум задержки приходился на другие режимы, в основном, <Area, Standard>.

При общем совершенствовании программы LeonardoSpectrum в старых версиях синтезатора можно реализовать лучшие решения. Поэтому в ответственных случаях целесообразно выполнять синтез как в новых, так и в старых версиях программы, и выбирать лучшие решения по тому или иному критерию.

Литература

1. *Бибило П.Н.* Системы проектирования интегральных схем на основе языка VHDL. StateCAD, ModelSim, LeonardoSpectrum. СОЛОН-Пресс. 2005.
2. http://www1.cs.columbia.edu/~cs4861/sis/espresso-examples/ex.
3. http://opencores.org/project,8b10b_encdec,overview.
4. *Бибило П.Н., Кириенко Н.А.* Оценка энергопотребления логических КМОП-схем по их переключательной активности. Микроэлектроника. № 1. 2012. С. 65–77.
5. LeonardoSpectrum User's Manual, Software Release 2011a. May 2011.
6. *Кнут Д.Э.* Искусство программирования. Том 4. А. Комбинаторные алгоритмы. Часть 1. ИД Вильямс. 2013.
7. *Бибило П.Н., Леончик П.В.* Алгоритм построения диаграммы двоичного выбора для системы полностью определённых булевых функций. Управляющие системы и машины. № 6. 2009. С. 42–49.
8. *Бибило П.Н., Романов В.И.* Логическое проектирование дискретных устройств с использованием продукционно-фреймовой модели представления знаний. Ленанд. 2014.