

УДК 004.3

А.А. Баркалов¹ (д-р техн. наук, проф.),

И.Я. Зеленёва² (канд. техн. наук, доц.),

Е.Р. Татолов² (магистрант),

¹Университет Зеленогурский, Польша

²Донецкий национальный технический университет

A.Barkalov@iie.uz.zgora.pl, tatolov@bk.ru

АНАЛИЗ ЭФФЕКТИВНОСТИ МЕТОДОВ КОДИРОВАНИЯ СОСТОЯНИЙ ПРИ СИНТЕЗЕ АВТОМАТОВ МИЛИ НА FPGA

Рассмотрен подход к исследованию эффективности реализации автоматов Мили в базе микросхем FPGA фирмы Xilinx с использованием средства синтеза XST. Представлены поддерживаемые XST методы кодирования состояний и проанализировано их влияние на синтез тестовых автоматов из набора IWLS LGSynth93. Показаны общие результаты исследований, выделены и сформулированы особенности использования методов для оптимизации аппаратных затрат и повышения быстродействия логической схемы автомата.

Ключевые слова: автомат Мили, кодирование состояний, XST, аппаратные затраты, быстродействие, KISS2, FPGA.

Введение

Автоматы Мили находят широкое применение при проектировании цифровых систем различной сложности, выступая, как правило, в роли устройств управления [1, 2]. Основная оптимизация характеристик автомата возможна на этапах кодирования внутренних состояний и построения логической схемы, где могут решаться задачи снижения уровня аппаратной избыточности, повышения быстродействия, уменьшения потребляемой мощности [3]. Поскольку базис реализации существенно влияет на характеристики логической схемы автомата, то и эффективность оптимизационных методов преимущественно зависит от особенностей целевой цифровой платформы [1, 4].

Микросхемы FPGA (Field-Programmable Gate Array) фирмы Xilinx являются современным базисом построения цифровых систем разного функционального назначения [5–9]. В этом случае процесс проектирования осуществляется с использованием САПР Xilinx ISE Design Suite [10], одной из компонент которой является средство синтеза XST (Xilinx Synthesis Technology) [11]. Назначение XST состоит в преобразовании описания системы на одном из языков описания аппаратуры (HDL от Hardware

Description Language) в набор микросхемно-независимых функциональных и запоминающих элементов из библиотеки Xilinx. После осуществления синтеза может быть получена информация о необходимом для его реализации количестве LUT-элементов (Look-Up Table) и триггеров, а также предполагаемая максимальная частота функционирования.

Проблемы эффективной реализации автоматов Мили в базисе FPGA рассмотрены в работах многих отечественных и зарубежных исследователей. Например, в работах [12–15] предлагаются способы специфического кодирования состояний, структуры имплементации на встроенных блоках памяти, методы многоуровневой реализации схем автоматов. Однако, в настоящее время отсутствуют результаты анализа эффективности непосредственно поддерживаемых XST алгоритмов кодирования состояний, которые могут применяться к автоматам, заданным согласно специальным техникам описания [11]. Такой анализ необходим для разработки новых методов кодирования состояний, дающих лучшие результаты, чем стандартные методы, поддерживаемые средством синтеза XST.

Целью работы является комплексное сравнительно-статистическое исследование возможностей использования реализуемых XST методов кодирования состояний для уменьшения аппаратурных затрат и повышения быстродействия логической схемы автомата Мили.

Методика проведения исследований

Средство синтеза XST поддерживает способы спецификации автоматов с использованием одного, двух и трех процессов, а также позволяет управлять общим критерием оптимизации (быстродействие, аппаратурные затраты) и алгоритмом кодирования состояний (автоматическое, унитарное, кодирование Грея, компактное, кодирование Джонсона, последовательное, скоростное). Кроме того, возможен выбор способа реализации логической схемы (LUT-элементы, встроенные блоки памяти) [11].

При автоматическом кодировании состояний, которое применяется по умолчанию, XST реализует собственный алгоритм выбора наилучшего метода кодирования, в зависимости от цели оптимизации и конфигурации автомата. Для исследования эффективности как данного алгоритма, так и остальных методов, был использован набор тестовых автоматов IWLS LGSynth93, представленных в формате KISS2 (рис. 1) [16, 17]. Преобразование автоматов из формата KISS2 в двухпроцессные VHDL-модели было выполнено с помощью программы KISS2 Converter, а их синтез проведен системой XST 11.3 для микросхемы XC5VLX30.

Разработанная авторами программа KISS2 Converter позволяет осуществлять автоматизированный анализ и обработку KISS2-файлов,

формирование на их основе автоматов Мили и Мура, генерацию XST-ориентированных VHDL-моделей, допускающих управление алгоритмом кодирования состояний и способом реализации логической схемы.

Для каждого тестового автомата была собрана статистика использования LUT-элементов и расчетное значение максимальной частоты функционирования при всех поддерживаемых способах кодирования состояний. При этом логическая схема автомата реализовывалась на LUT-элементах, а целью оптимизации являлся параметр быстродействия.

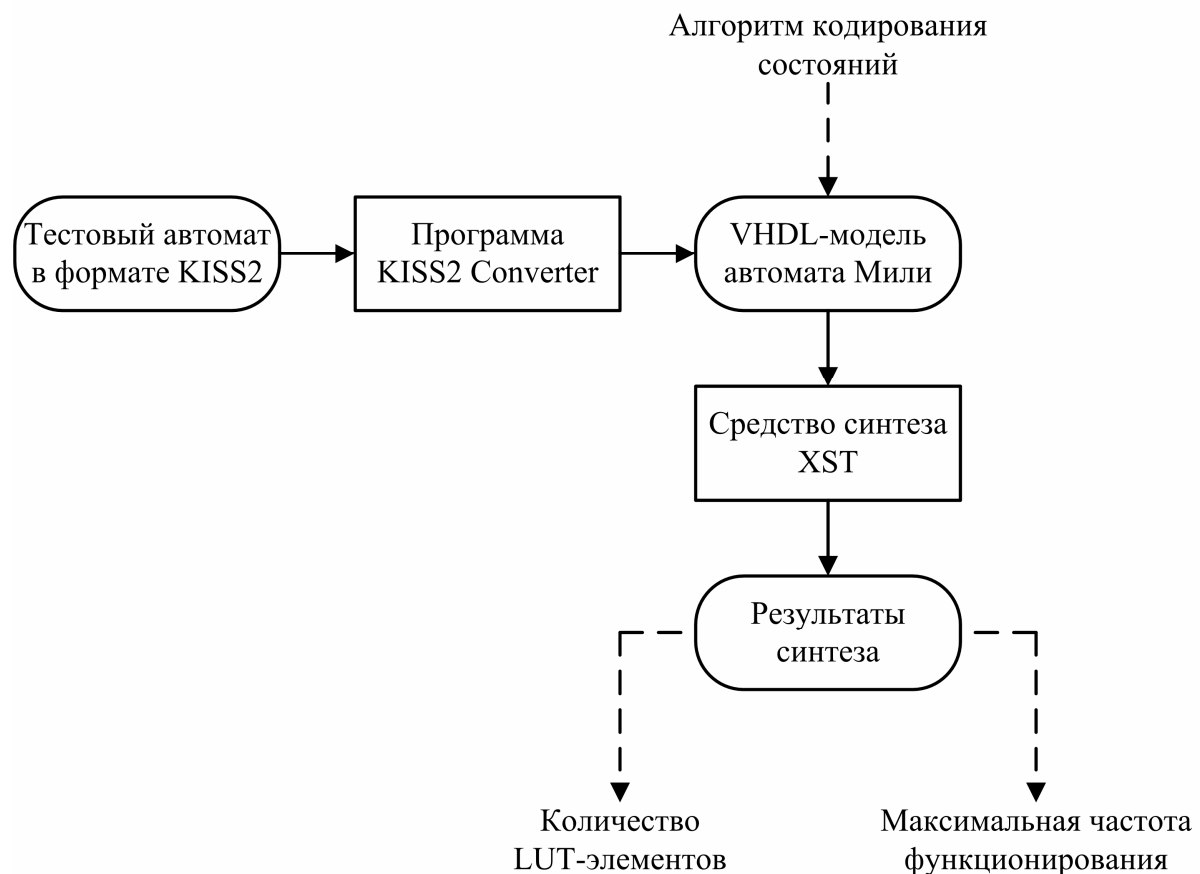


Рисунок 1 – Методика исследования эффективности алгоритмов кодирования состояний

Анализ эффективности методов кодирования

В рамках исследований было проведено более трехсот экспериментов, результаты которых отражены в табл. 1. В этой таблице методы кодирования обозначены следующим образом: «auto» – автоматическое, «one-hot» – унитарное, «compact» – компактное, «sequential» – последовательное, «gray» – кодирование Грея, «johnson» – кодирование Джонсона, «speed1» – скоростное.

Таблица 1 – Результаты исследований на тестовых автоматах

Автомат	<i>auto</i>		<i>one-hot</i>		<i>compact</i>		<i>sequential</i>		<i>gray</i>		<i>johnson</i>		<i>speed1</i>	
	LUT	MHz	LUT	MHz	LUT	MHz	LUT	MHz	LUT	MHz	LUT	MHz	LUT	MHz
bbara	11	639	9	966	13	635	13	639	19	589	24	545	13	962
bsse	29	559	29	559	29	582	29	538	31	538	36	408	38	556
bbtas	5	962	8	966	5	966	5	962	5	955	5	962	9	966
beecount	7	952	19	639	7	952	7	952	7	948	21	625	30	583
cse	49	480	52	477	46	463	50	487	46	454	71	434	72	453
dk14	8	945	29	522	8	945	8	945	8	945	19	623	40	512
dk15	7	1062	19	737	7	1062	7	1062	7	1062	7	1062	19	659
dk16	46	556	46	556	15	625	19	506	27	554	86	355	70	399
dk17	6	952	14	669	6	952	6	952	6	952	7	895	27	571
dk27	5	900	8	906	5	897	5	959	5	955	6	899	10	903
dk512	17	730	17	730	7	899	7	895	7	899	21	437	19	790
ex1	64	586	64	586	74	447	67	478	66	406	106	340	72	605
ex4	15	962	15	962	16	626	15	598	14	748	33	546	15	962
ex6	29	553	30	580	20	621	23	615	22	616	36	426	31	598
keyb	56	384	56	384	65	358	71	382	66	447	62	435	85	374
kirkman	51	874	84	1058	53	569	48	880	51	874	112	451	84	1058
lion	3	1084	5	962	3	1080	3	1080	3	1084	3	1084	5	962
mark1	27	726	27	726	19	708	22	622	18	623	27	574	29	959
mc	5	1071	8	1071	5	1071	6	1071	5	1071	5	1071	8	1071
opus	22	596	22	754	21	628	26	585	22	596	26	576	26	671
planet	100	888	100	888	138	389	145	417	149	375	192	346	106	637
planet1	100	888	100	888	138	389	145	417	149	375	192	346	106	637
pma	73	554	73	554	115	438	108	367	112	375	121	405	88	559
s1	77	550	77	550	75	447	89	328	105	368	114	361	81	552
s1488	140	425	140	425	141	432	130	394	147	433	192	334	162	458
s1494	124	412	124	412	143	442	135	383	145	383	192	333	152	462
s208	28	559	28	559	13	669	12	716	15	639	29	483	50	386
s27	4	962	21	636	4	962	7	679	4	962	12	664	21	631
s298	362	406	362	406	330	313	264	311	274	314	716	244	399	397
s386	26	577	31	586	28	581	28	558	29	429	43	422	36	441
s420	28	559	28	559	14	629	12	716	15	639	29	483	36	510
s510	42	900	42	900	39	448	53	440	50	427	123	388	42	900
s820	63	429	63	429	85	395	92	441	93	438	98	366	93	399
s832	63	429	63	429	73	412	77	431	87	394	97	335	108	444
sand	99	569	99	569	121	426	125	421	125	438	189	306	103	490
scf	179	676	179	676	202	338	205	349	197	389	337	327	180	561
shiftreg	0	1584	9	1080	0	1584	4	959	4	959	5	902	4	903
sse	29	559	29	559	28	543	37	548	32	540	44	394	36	612
styr	118	430	118	430	127	369	138	363	138	353	181	323	161	454
tav	6	1556	6	1556	6	911	6	911	5	914	5	914	6	1556
tbk	55	406	179	360	71	465	129	342	137	290	295	276	444	342

Для каждого тестового автомата и алгоритма кодирования приведены значения количества LUT-элементов («LUT»), необходимых для реализации, а также показатели максимальной частоты функционирования логической схемы в МГц («MHz»).

Тестовые автоматы «donfile», «modulo12», «s1a» и «s8» обладают константными выходными сигналами, что в процессе синтеза привело к реализациям в виде подключенных к выходным линиям уровней нуля или единицы.

С точки зрения уменьшения аппаратных затрат (рис. 2) и повышения быстродействия (рис. 3), лучшие результаты показало автоматическое кодирование, которое привело к оптимальным

реализациям в 58,54% и 39,02% случаев соответственно. Среди методов аппаратурной минимизации следует отметить компактное кодирование, продемонстрировавшее эффективность в 46,34%. Скоростное кодирование привело к максимальному быстрдействию в 36,59% экспериментов, что всего на 2,43% хуже показателя автоматического кодирования, однако не обеспечило ни одного аппаратурно-оптимального решения (0,00%).

Лучшие результаты по одновременному достижению минимальной аппаратурной реализации и максимального быстрдействия (двухцелевая оптимизация) (рис. 4) показали методы автоматического и компактного кодирования (29,27%). Широко используемый метод унитарного кодирования имеет сбалансированные результаты по аппаратурным затратам (31,71%) и быстрдействию (34,15%), но всего в 12,20% случаев приводит к одновременной оптимизации обоих факторов. Таким образом, получены данные для сравнения вновь разрабатываемых алгоритмов кодирования состояний с известными подходами.

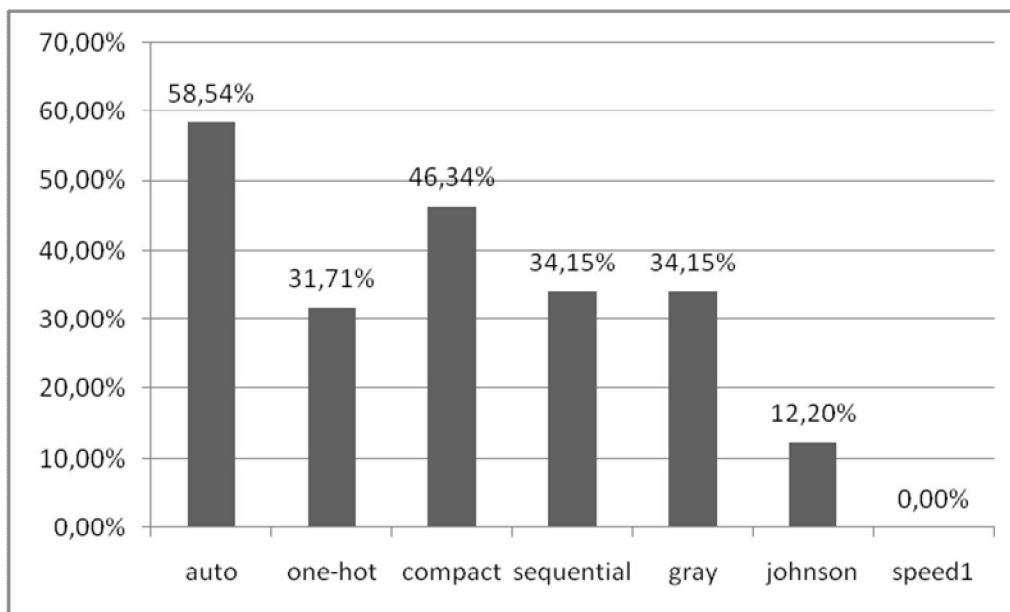


Рисунок 2 – Эффективность методов при оптимизации аппаратурных затрат

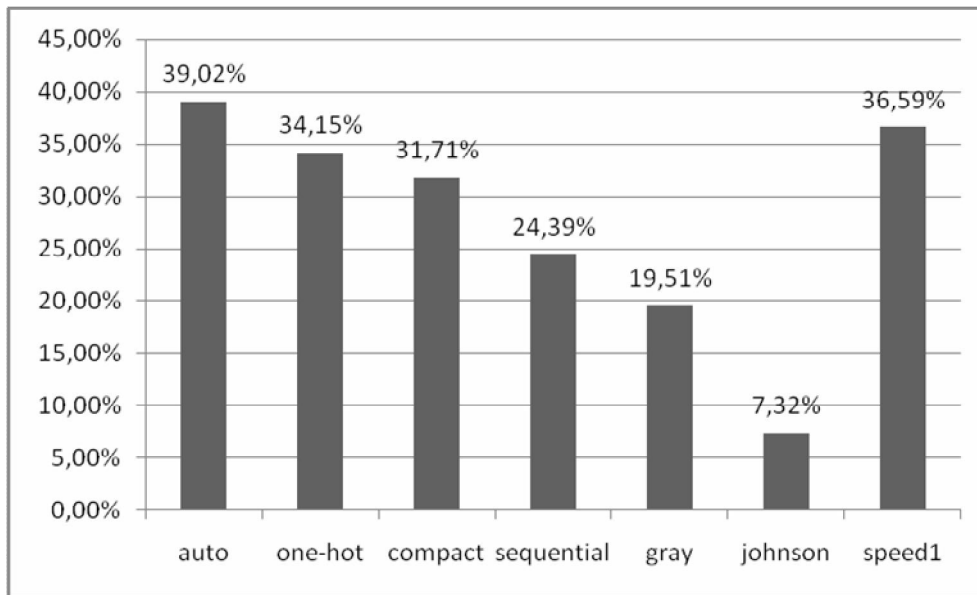


Рисунок 3 – Эффективность методов при оптимизации быстродействия

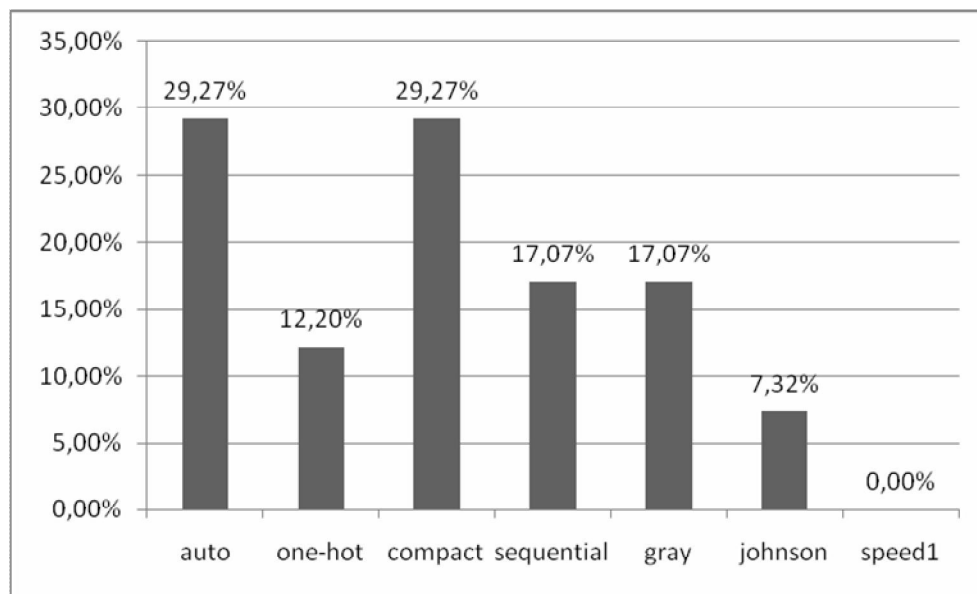


Рисунок 4 – Эффективность методов при двухцелевой оптимизации

Заключение

Проведенные в работе эксперименты показали, что используемый средством синтеза XST по умолчанию алгоритм выбора оптимального кодирования состояний, при скорости функционирования, как целевом параметре, и LUT-ориентированной реализации логической схемы, приводит к неэффективным результатам в 60,98% (быстродействие) и 41,46% (аппаратурные затраты) случаев, а двухцелевая оптимизация достигается менее чем для трети автоматов (29,27%).

Методы компактного и скоростного кодирования, при оптимизации аппаратурных затрат и быстродействия соответственно, в наибольшей степени приближаются к показателям автоматического кодирования. Кроме того, результаты двухцелевой оптимизации при компактном и автоматическом кодировании полностью совпадают.

Дальнейшие направления исследований связаны с изучением способов реализации автоматов Мили с использованием как макроячеек базиса CPLD, так и встроенных блоков памяти FPGA. Кроме того, планируется выполнить аналогичный анализ эффективности методов кодирования состояний при синтезе автоматов Мура.

Список литературы

1. Баркалов А.А. Синтез микропрограммных автоматов на заказных и программируемых СБИС / А.А. Баркалов, Л.А. Титаренко. – Донецк: ДонНТУ, Технопарк ДонНТУ УНИТЕХ, 2009. – 336 с.
2. Баранов С.И. Синтез микропрограммных автоматов (граф-схемы и автоматы) / С.И. Баранов. – Л.: Энергия, 1979. – 232 с.
3. Barkalov A.A. Synthesis of operational and control automata / A.A. Barkalov, L.A. Titarenko. – Donetsk: DonNTU, TechPark DonNTU UNITECH, 2009. – 256 p.
4. Adamski M. Design of digital systems and devices / M. Adamski, A. Barkalov, M. Wegrzyn. – Springer-Verlag, 2011. – 365 p.
5. Grout I. Digital systems design with FPGAs / I. Grout. – Elsevier, 2008. – 724 p.
6. Zeidman B. Designing with FPGAs and CPLDs / B. Zeidman. – Elsevier, 2002. – 224 p.
7. Грушвицкий Р.И. Проектирование систем на микросхемах программируемой логики / Р.И. Грушвицкий, А.Х. Мурсаев, Е.П. Угрюмов. – СПб.: БХВ-Петербург, 2002. – 608 с.
8. Максфилд К. Проектирование на ПЛИС. Курс молодого бойца / К. Максфилд. – М.: Издательский дом «Додэка-XXI», 2007. – 408 с.
9. FPGA, CPLD and EPP solutions from Xilinx, Inc. [Электронный ресурс]. – Режим доступа: <http://www.xilinx.com>.
10. Xilinx ISE Design Suite Software [Электронный ресурс]. – Режим доступа: <http://www.xilinx.com/products/design-tools/ise-design-suite>.
11. XST User Guide, v. 11.3 [Электронный ресурс]. – Режим доступа: http://www.xilinx.com/support/documentation/sw_manuals/xilinx11/xst.pdf.
12. Kubatova H. FEL-Code: FSM internal state encoding method / H. Kubatova, M. Becvar // Proceedings of 5th International Workshop on Boolean Problems. – Freiberg, 2002. – P. 109-114.
13. Sklyarov V. Synthesis and implementation of RAM-based finite state machines in FPGAs / V. Sklyarov // Proceedings of Conference on Field Programmable Logic. – Villach, 2000. – P. 718-728.
14. Bukowiec A. State machines synthesis and implementation into FPGAs with multiple encoding of states / A. Bukowiec, A. Barkalov, L. Titarenko // Radioelectronics and Informatics. – 2008. – № 4. – P. 43-48.
15. Selvaraj H. FSM implementation in embedded memory blocks of programmable logic devices using functional decomposition / H. Selvaraj, M. Rawski, T. Luba //

Proceedings of International Conference on Information Technology: Coding and Computing.
– Las Vegas, 2002. – P. 355-360.

16. International workshop on logic synthesis benchmark suite (LGSynth93).
[Электронный ресурс]. – Режим доступа: <http://www-lab13.kuee.kyoto-u.ac.jp/eda/benchmark/mcnc/benchmarks/LGSynth93/LGSynth93.tar>.

17. Yang S. Logic synthesis and optimization benchmarks user guide. Version 3.0.
Technical Report / S. Yang. – North Carolina, 1991. – 44 p.

Надійшла до редакції 21.10.2011.

Рецензент: канд. техн. наук, доц. Цололо С.О.

О.О. Баркалов, І.Я. Зеленцова, Є.Р. Татолов
Донецький національний технічний університет

Аналіз ефективності методів кодування станів при синтезі автоматів Мілі на FPGA. Розглянуто підхід до дослідження ефективності реалізації автоматів Мілі в базисі мікросхем FPGA фірми Xilinx з використанням засобу синтезу XST. Представлено підтримувані XST методи кодування станів і проаналізовано їх вплив на синтез тестових автоматів з набору IWLS LGSynth93. Показано загальні результати досліджень, виділено та сформульовано особливості використання методів для оптимізації апаратних витрат і підвищення швидкодії логічної схеми автомату.

Ключові слова: автомат Мілі, кодування станів, XST, апаратні витрати, швидкодія, KISS2, FPGA.

A. Barkalov, I. Zelenyova, E. Tatolov
Donetsk National Technical University

Analysis of Efficiency of State Assignment Methods for Mealy FSM Synthesis on FPGA. Approach to research of Mealy FSM implementation efficiency on Xilinx FPGA using synthesis tool XST is considered. State assignment methods supported by XST are pointed out and their impact on IWLS LGSynth93 benchmarks synthesis is analyzed. General results of research are shown. Peculiarities of using methods for optimization FSM circuit hardware amount and speed are marked.

Keywords: Mealy FSM, state assignment, XST, hardware amount, speed, KISS2, FPGA.