

# Проектирование на программируемых логических интегральных схемах быстрых компараторов большой разрядности

В.В. Соловьев

Белорусская государственная академия связи (Минск, Республика Беларусь), valsol@mail.ru

**Аннотация** — Представлен метод проектирования на программируемых логических интегральных схемах (ПЛИС) иерархических структур двоичных компараторов высокого быстродействия и большой разрядности. Приводится методика синтеза иерархических структур компараторов. Лучшая иерархическая структура компараторов для конкретного семейства ПЛИС находится эмпирически. Предложенный метод позволяет увеличить быстродействие по сравнению с традиционным подходом для компараторов на 512 разрядов в 3.409 раз, на 1024 разряда – в 4.987 раз, на 2048 разрядов – в 27.339 раз, на 4096 разрядов – в 65.596 раз, на 8192 разряда в – 127.395 раз. Метод также позволяет для отдельных семейств в 16 раз увеличить максимальное число разрядов входных слов компараторов, которые могут быть реализованы на ПЛИС. В заключении отмечаются особенности практического использования метода, а также указываются направления дальнейшего развития данного подхода.

**Ключевые слова** — двоичный компаратор, иерархическая структура, высокое быстродействие, большая разрядность, программируемые логические интегральные схемы, FPGA, SoC.

## I. ВВЕДЕНИЕ

В настоящее время наблюдается устойчивая тенденция увеличения слов вычислительной техники. Особенно быстро разрядность слов увеличивается в системах телекоммуникации, а также в устройствах передачи и обработки информации. С другой стороны, наиболее перспективной элементной базой электронной техники на сегодняшний день являются программируемые логические интегральные схемы (ПЛИС), такие как программируемые пользователем вентильные матрицы (Field Programmable Gate Arrays – FPGAs) и системы на кристалле (System on Chips – SoCs) [1]. При проектировании сложных цифровых систем часто используются стандартные функциональные узлы большой разрядности. Отсюда возникает необходимость в разработке эффективных методов проектирования на ПЛИС стандартных функциональных узлов большой разрядности (от 512 разрядов и выше). Одним из таких трудно реализуемых стандартных функциональных узлов является компаратор двоичных чисел. Особенно остро проблема создания многоразрядных компараторов повышенного быстродействия стоит в таких областях, как криптография, телекоммуникация, распознавание образов и др.

Компаратор представляет собой комбинационную схему, которая выполняет микрооперации отношения между двумя двоичными числами  $A$  и  $B$ . Традиционный компаратор сравнивает два двоичных числа  $A$  и  $B$  разрядности  $M$  и реализует три функции:  $G$  «больше» ( $A > B$ ),  $E$  «равно» ( $A=B$ ) и  $L$  «меньше» ( $A < B$ ). Отметим, что при построении компараторов большой разрядности достаточно реализовать только две функции:  $G$  «больше» и  $E$  «равно», поскольку функция  $L$  «меньше» всегда может быть определена на основе первых двух:  $L = \bar{G} \ \& \ \bar{E}$ , где символ «&» означает логическое И.

Проблеме проектирования двоичных компараторов посвящено достаточно много работ. В [2] представлена древовидная структура компаратора, которая использует двухфазную динамическую CMOS-логику (all-N-transistor – ANT). Однако логика ANT требует 3.5 конвейерных циклов синхронизации и не годится для одноциклового функционирования. В [3] рассматривается двухфазный компаратор с одним циклом синхронизации, который использует приоритетный шифратор для определения первого неравного бита, при этом увеличение производительности составляет 16% по сравнению с [2]. В [4] предлагается двухуровневая иерархическая структура компаратора, где вместо компараторов второго уровня используется динамический мультиплексор; в результате быстродействие компаратора увеличивается на 28%. В [5] вводится дополнительная логика предвидения переноса (carry-look-ahead addition – CLA), которая реализуется с помощью Манчестерской цепочки переноса. Предлагается трехуровневая структура компаратора с двухфазовой синхронизацией, где на первом уровне используются 2-битовые блоки CLA, на втором – 4 битовые, а на третьем – 16 и 8 битовые. Метод позволяет увеличить быстродействие на 12%. В [6] предлагается проект компаратора, который может использовать как заказные, так и стандартные CMOS-ячейки. Проект компаратора имеет масштабируемую параллельно-префиксную структуру, которая позволяет переходить к сравнению следующего бита только тогда, когда сравниваемые биты равны. Метод позволяет уменьшить рассеивание динамической мощности. В [7] предлагается архитектура компаратора, специально спроектированная для статической логики с целью уменьшения энергопотребления и повышения

быстродействия при низкой активности входных данных. Проект компаратора имеет древовидную структуру со схемой предварительного кодирования. В [8] выполнена модификация проекта [7]. Структура компаратора имеет два уровня. На первом уровне находятся 8-битовые компараторы, построенные по древовидной структуре [7], а второй уровень составляет один 8-битовый компаратор. Кроме того, были изменены схемы, реализующие каждый уровень. В [9] представлен проект компаратора с логикой константной задержки (constant-delay – CD). Использование CD-логики позволяет выходам переключаться в 2 раза быстрее по сравнению с динамической логикой. Структура компаратора имеет два уровня. Первый уровень адаптирует с целью уменьшения энергопотребления древовидную структуру компаратора, специально спроектированную для статической логики. Второй уровень использует CD-логику для увеличения производительности. Предложенный проект на 20% увеличивает быстродействие и на 17% уменьшает энергопотребление по сравнению с [8].

Все рассмотренные методы предназначены для реализации компараторов на заказных микросхемах (Application Specific Integrated Circuits – ASICs) и не подходят для построения компараторов на ПЛИС. В [10] предлагаются две иерархические структуры компараторов при реализации на FPGA, которые описаны на языке Verilog и сравниваются с традиционной структурой компаратора. При этом первая архитектура позволяет увеличить быстродействие компаратора на 23%, а вторая – на 35%. В [11] рассмотрены следующие методы синтеза компараторов: параллельный, последовательный, параллельно-последовательный, а также с использованием сумматора. Для каждого метода предлагается два способа реализации: графический (схемный) и на языке AHDL (язык описания аппаратуры фирмы Altera). Экспериментальные исследования проводились с помощью пакета MAX+PLUS II при реализации 64-разрядных компараторов и сравнивались с параметризованной функцией `lpm_compare` фирмы Altera. В [12] предложен метод проектирования компараторов в виде иерархической структуры. Экспериментальные исследования проводились также с помощью пакета MAX+PLUSII фирмы Altera. Представленный метод сравнивался с последовательным и параллельным методами работы [11], а также с параметризованной функцией `lpm_compare` и методом, реализованном в компиляторе языка AHDL. На первом уровне иерархической структуры использовались компараторы на 4 разряда, которые строились четырьмя способами: с помощью функции `lpm_compare`, средствами языка AHDL, а также с помощью параллельного и последовательного метода [11]. В [13] предлагается метод проектирования на ПЛИС иерархических древовидных структур двоичных компараторов, который ориентирован на построение компараторов наименьшей стоимости. Метод позволяет уменьшить стоимость реализации

компараторов от 5.3% до 43.0%, а для компараторов большой разрядности (с длиной входных слов 1024) для отдельных семейств – в 2.225 раза.

В настоящей работе предлагается метод проектирования на ПЛИС иерархических структур компараторов большой разрядности и высокого быстродействия. Метод предназначен для синтеза компараторов на ПЛИС типа FPGA и SoC. Предлагаемый метод позволяет в широком диапазоне изменять число уровней иерархической структуры компаратора, в результате чего пользователь может выбирать компромисс между стоимостью реализации и быстродействием. Синтезируемая схема компаратора не содержит сигналов синхронизации, поэтому не требует дополнительных схем для их формирования (в отличие от рассмотренных известных методов). Высокое быстродействие в предлагаемом подходе обеспечивается отсутствием поразрядных переносов и параллельностью выполнения логических операций.

## II. МЕТОДИКА СИНТЕЗА ИЕРАРХИЧЕСКИХ СТРУКТУР КОМПАРАТОРОВ

Иерархическая двухуровневая структура компараторов показана на рис. 1. Она состоит из модулей компараторов первого уровня  $CMP_1, \dots, CMP_{N_1}$  и комбинационной схемы CL. Каждый компаратор первого уровня  $CMP_n$  представляет собой двоичный компаратор на  $M_1$  разрядов, который реализует функцию «больше»  $g_n$  и функцию «равно»  $e_n, n = \overline{1, N_1}$ .

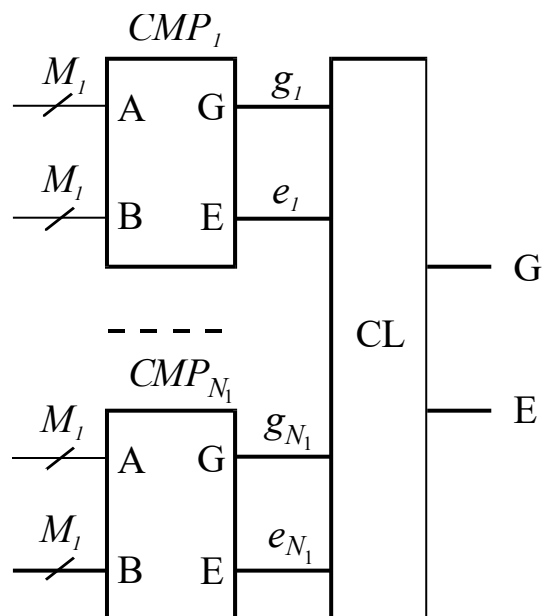


Рис. 1. Двухуровневый иерархический компаратор со структурой  $C_{M_2 \times N_1 \times M_1}$

Комбинационная схема CL на основании значений функций, сгенерированных компараторами первого уровня, вычисляет значения выходных функций «больше»  $G$  и «равно»  $E$  всей иерархической структуры с помощью следующих логических уравнений:

$$G = g_{N_1} + e_{N_1} \& g_{N_1-1} + e_{N_1} \& e_{N_1-1} \& g_{N_1-2} + \dots + e_{N_1} \& e_{N_1-1} \& \dots \& e_2 \& g_1, \quad (1)$$

$$E = e_1 \& e_2 \& \dots \& e_{N_1}. \quad (2)$$

Двухуровневую иерархическую структуру двоичного компаратора на рис. 1 будем обозначать следующим образом:  $C_{M_2 \times N_1 - M_1}$ , где  $C$  – сокращение от слова «Comparator»;  $M_1$  – длина (число разрядов) входных слов компараторов первого уровня;  $N_1$  – число компараторов первого уровня;  $M_2$  – длина входных слов двухуровневых компараторов, где  $M_2 = N_1 M_1$ . Отметим, что уравнения для функций  $G$  и  $E$  зависят только от значения числа  $N_1$  модулей компараторов первого уровня и не зависят от значения  $M_2$  (длины входных слов реализуемого компаратора), а также  $M_1$  (длины входных слов компараторов первого уровня).

Иерархическая двухуровневая структура на рис. 1, в свою очередь, может выступать в качестве компараторов первого уровня. Продолжая подобным образом далее, можно построить многоуровневую иерархическую структуру компараторов большой разрядности. Обобщенную иерархическую структуру компараторов будем обозначать с помощью следующей формулы:

$$C_{M_T \times N_{T-1} - M_{T-1} \times N_{T-2} - \dots - M_t \times N_{t-1} - \dots - M_2 \times N_1 - M_1}, \quad (3)$$

где  $M_t$  – длина слов компараторов уровня  $t$ ,  $t = \overline{1, T}$ ,  $N_{t-1}$  – число компараторов предыдущего уровня  $t-1$ , из которых строятся компараторы уровня  $t$ .

В формуле (3) иерархической структуры компаратора должны выполняться условия:

$$M_t = N_{t-1} M_{t-1} \quad \forall t = \overline{2, T}. \quad (4)$$

В общем случае иерархическую многоуровневую структуру компараторов можно представить точно так же, как на рис. 1, где на первом уровне располагаются компараторы первого уровня  $CMP_1, \dots, CMP_{N_1}$ , а комбинационная схема СЛ реализует суперпозицию функций «больше» и «равно», определяемую формулой (3). Число уровней  $T$  в формуле (3) определяет число логических уровней (глубину) иерархической структуры компаратора.

Синтез иерархических структур компараторов выполняется на основании формулы (3) при выполнении условий (4). Отметим, что для одного и того же компаратора могут быть построены различные иерархические структуры разной глубины с различным числом модулей и различной длиной входных слов на каждом уровне.

Основу логических элементов современных FPGA и SoC составляют функциональные генераторы типа LUT (Look Up Table). LUT представляет собой память

RAM небольшой разрядности, которая может быть запрограммирована на реализацию любой логической функции от небольшого числа аргументов. Поскольку число входов функциональных генераторов LUT для большинства семейств FPGA равно 4, предлагается для всех иерархических структур компараторов на первом уровне использовать компараторы на 2 бита. Это позволяет каждую функцию компараторов первого уровня реализовать на одном LUT.

Пусть  $A = (a_2, a_1)$  и  $B = (b_2, b_1)$  – входные слова 2-битового компаратора. Булева функция «равно» для компаратора на 2 разряда не поддается минимизации и имеет вид:

$$E = \bar{a}_2 \& \bar{a}_1 \& \bar{b}_2 \& \bar{b}_1 + \bar{a}_2 \& a_1 \& \bar{b}_2 \& b_1 + a_2 \& a_1 \& b_2 \& b_1 + a_2 \& \bar{a}_1 \& b_2 \& \bar{b}_1. \quad (5)$$

Булева функция «больше» для компаратора на 2 разряда после минимизации имеет следующий вид:

$$G = a_2 \& \bar{b}_2 + a_1 \& \bar{b}_2 \& \bar{b}_1 + a_2 \& a_1 \& \bar{b}_1. \quad (6)$$

Таким образом, конкретная иерархическая структура компаратора определяется формулой (3) при выполнении условий (4). Компараторы первого уровня реализуются на основании логических уравнений (5) и (6), а комбинационная схема СЛ на каждом уровне иерархической структуры определяется с помощью логических уравнений (1) и (2).

Иерархическая структура по своей природе имеет наименьшую задержку прохождения сигналов по сравнению с последовательной декомпозицией. Поэтому можно ожидать, что иерархические структуры проектируемого компаратора будут иметь малую стоимость реализации и высокое быстродействие. Логические уравнения (1) и (2), описывающие комбинационные схемы СЛ иерархической структуры, а также уравнения (5) и (6) компараторов первого уровня относительно просты. Поэтому можно ожидать, что с помощью иерархических структур можно построить двоичные компараторы достаточно большой разрядности.

**Гипотеза.** Предлагаемая методика синтеза иерархических структур двоичных компараторов позволяет строить на FPGA и SoC быстрые компараторы большой разрядности.

Открытым остается вопрос: какая формула иерархической структуры компаратора в наилучшей степени подходит для реализации компаратора определенной разрядности на конкретном семействе FPGA и SoC. Ответ на данный вопрос в предлагаемом подходе определяется эмпирически путем выполнения большого числа экспериментальных исследований.

### III. УСЛОВИЯ ПРОВЕДЕНИЯ ЭКСПЕРИМЕНТАЛЬНЫХ ИССЛЕДОВАНИЙ

Иерархические структуры двоичных компараторов исследовались на FPGA и SoC фирмы Altera с использованием пакета автоматизированного проектирования Quartus II. Проблема при синтезе компараторов большой разрядности (с длиной входных слов больше 256 разрядов) заключается в том, что компилятор выдает сообщение о невозможности синтеза из-за недостатка числа внешних выводов. Поэтому в исследуемых проектах компараторов с целью уменьшения числа используемых внешних выводов на входных шинах компараторов установлены двоичные дешифраторы. В этом случае структуру компаратора будем обозначать в виде  $DC\_c\_xxx$ , где аббревиатура «DC» указывает на наличие дешифраторов на входных шинах компараторов. Чтобы отделить логику дешифраторов от логики компараторов на выходах дешифраторов установлены буферы LCELL. Буфер LCELL гарантирует, что формируемая функция всегда будет реализована на выходе логического элемента и не будет объединяться с целью оптимизации с логикой, следующей за данным буфером.

Все проекты компараторов описывались на языке Verilog. Результаты синтеза иерархических структур компараторов сравнивались с традиционной реализацией компаратора на языке Verilog, когда функции «больше» и «равно» описываются с помощью соответствующих операций в виде:

assign G = (A>B);

assign E = (A==B).

Отметим, что получаемые при этом результаты полностью совпадают с результатами, произведенными с помощью параметризованной функции `lpm_compare` фирмы Altera.

Эксперименты проводились следующим образом. Для компараторов на 4, 8, 16, 32, 64, 128 и 256 разрядов были построены все возможные

иерархические структуры компараторов, всего (вместе с компаратором на 2 разряда) 128 проектов компараторов. Для построения компараторов на 512 разрядов из ранее построенных структур выбирались такие структуры, которые показывали лучшие (или близкие к лучшим) результаты хотя бы для одного семейства FPGA. В качестве критериев оптимизации рассматривались стоимость реализации (число используемых логических элементов) и быстродействие (максимальная задержка сигналов). Аналогично строились иерархические структуры компараторов на 1024, 2048, 4096 и 8192 разряда. В результате было построено иерархических структур компараторов на 512 разрядов – 58, на 1024 разряда – 44, на 2048 разрядов – 48, на 4096 разрядов – 5, на 8192 разрядов – 5.

Экспериментальные исследования проводились для следующих семейств FPGA и SoC фирмы Altera: MAX II, MAX V, MAX 10; Cyclone, Cyclone II, Cyclone III, Cyclone IV, Cyclone V; Arria, Arria II; Stratix, Stratix II, Stratix III, Stratix IV.

Все установки параметров синтеза были приняты по умолчанию пакета Quartus II. В качестве критерия оптимизации при нахождении иерархической структуры компаратора максимальной быстродействия рассматривалась максимальная задержка прохождения сигналов с входов на выходы проекта (с учетом задержки на входных дешифраторах) в наносекундах.

### IV. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТАЛЬНЫХ ИССЛЕДОВАНИЙ

В табл. 1 и табл. 2 представлены результаты экспериментальных исследований иерархических структур двоичных компараторов, где  $D_T$  – задержка сигналов (в наносекундах) в случае традиционной реализации компараторов на языке Verilog;  $D_H$  – задержка сигналов при использовании иерархических структур для лучшего случая;  $D_T/D_H$  – отношение соответствующих параметров; `min` и `max` – минимальное и максимальное значение параметра.

Таблица 1

Результаты исследования быстродействия иерархических структур компараторов на 512, 1024 и 2048 разрядов

Семейство	DC_c_512			DC_c_1024			DC_c_2048		
	$D_T$	$D_H$	$D_T/D_H$	$D_T$	$D_H$	$D_T/D_H$	$D_T$	$D_H$	$D_T/D_H$
MAX II	42.919	18.481	2.322	(1)	(1)	-	(1)	(1)	-
MAX V	51.961	31.930	1.627	(1)	(1)	-	(1)	(1)	-
MAX 10	47.376	20.568	2.303	(3)	23.650	-	(3)	(3)	-
Cyclone	38.035	25.477	1.493	65.046	32.013	2.032	(2)	42.177	-
Cyclone II	71.976	24.393	2.951	(3)	27.004	-	(3)	36.170	-
Cyclone III	52.037	20.399	2.551	87.810	24.038	3.653	(3)	30.922	-
Cyclone IV	66.113	19.398	3.409	105.036	22.055	4.762	(3)	30.811	-
Cyclone V	52.683	27.531	1.914	92.257	28.145	3.278	550.656	41.489	13.272
Arria	68.350	26.421	2.587	147.186	31.597	4.658	1011.508	37.490	26.981
Arria II	42.160	21.769	1.937	65.340	22.701	2.878	407.297	30.220	13.478
Stratix	32.191	26.178	1.230	54.256	28.764	1.886	106.320	37.494	2.836
Stratix II	45.637	19.202	2.377	102.240	20.503	4.987	705.011	25.788	27.339

Stratix III	36.696	17.112	2.144	79.482	19.103	4.161	527.213	21.551	24.464
Stratix IV	61.597	22.397	2.750	91.016	22.117	4.115	641.317	26.262	24.420
min	32.191	17.112	1.230	54.256	19.103	1.886	106.320	21.551	2.836
max	71.976	31.930	3.409	141.186	32.013	4.987	1011.508	42.177	27.339

Таблица 2

Результаты исследования быстродействия иерархических структур компараторов на 4096 и 8192 разряда

Семейство	DC_c_4096			DC_c_8192		
	D <sub>T</sub>	D <sub>H</sub>	D <sub>T</sub> /D <sub>H</sub>	D <sub>T</sub>	D <sub>H</sub>	D <sub>T</sub> /D <sub>H</sub>
MAX II	(1)	(1)	-	(1)	(1)	-
MAX V	(1)	(1)	-	(1)	(1)	-
MAX 10	(3)	(3)	-	(3)	(3)	-
Cyclone	(2)	48.329	-	(2)	(2)	-
Cyclone II	(3)	41.115	-	(3)	48.490	-
Cyclone III	(3)	34.980	-	(3)	39.240	-
Cyclone IV	(3)	(3)	-	(3)	(3)	-
Cyclone V	1287.112	53.600	24.013	(4)	54.128	-
Arria	2949.775	44.969	65.596	6952.727	54.576	127.395
Arria II	974.367	36.221	26.901	(4)	41.695	-
Stratix	191.316	46.125	4.148	(2)	50.704	-
Stratix II	2062.869	32.916	62.671	5018.185	42.150	119.055
Stratix III	1478.210	25.886	57.105	3911.609	32.595	120.006
Stratix IV	1734.903	32.043	54.143	4503.653	42.205	106.709
min	191.316	25.886	4.148	3911.609	32.595	106.709
max	2949.775	48.329	65.596	6952.727	54.576	127.395

- (1) – реализация невозможна из-за недостатка числа логических элементов
- (2) – реализация невозможна, поскольку попытка упаковки закончилась безуспешно
- (3) – реализация невозможна из-за недостатка длины цепи переноса или каскадирования
- (4) – ошибка пакета Quartus II: переполнение стека

Анализ результатов экспериментальных исследований компараторов на 512 разрядов показывает, что все структуры DC\_c\_512\_xxx были реализованы на всех семействах FPGA. Использование иерархических структур при построении компараторов на 512 разрядов позволяет увеличить быстродействие от 1.230 раз (для семейства Stratix) до 3.409 раз (для семейства Cyclone IV), по сравнению с традиционным подходом.

Синтез компараторов на 1024 разряда традиционным способом невозможен из-за недостатка числа логических элементов для семейств MAX II и MAX V, а также для семейств MAX 10 и Cyclone II из-за недостатка длины цепи переноса или каскадирования. В то же время использование иерархических структур позволяет строить компараторы на 1024 разряда для семейств MAX 10 и Cyclone II. Использование иерархических структур при построении компараторов на 1024 разряда позволяет увеличить быстродействие от 1.886 раз (для семейства Stratix) до 4.987 раз (для семейства Stratix II), по сравнению с традиционным подходом.

Синтез компараторов на 2048 разрядов традиционным способом невозможен также для семейств Cyclone, Cyclone III и Cyclone IV из-за

недостатка длины цепи переноса или каскадирования. В то же время использование иерархических структур позволяет строить для этих семейств компараторы на 2048 разрядов. Использование иерархических структур при построении компараторов на 2048 разрядов позволяет увеличить быстродействие от 2.836 раз (для семейства Stratix) до 27.339 раз (для семейства Stratix II), по сравнению с традиционным подходом.

Использование иерархических структур при построении компараторов на 4096 разрядов позволяет увеличить быстродействие от 4.148 раз (для семейства Stratix) до 65.596 раз (для семейства Arria), по сравнению с традиционным подходом.

Реализация компараторов на 8192 разряда традиционным способом возможна только для семейств Stratix II, Stratix III, Stratix IV и Arria. В то время как при использовании иерархических структур компараторы на 8192 разряда можно реализовать на указанных семействах, а также на семействах Cyclone II, Cyclone III, Cyclone V, Arria II и Stratix. Использование иерархических структур при построении компараторов на 8192 разряда позволяет увеличить быстродействие от 106.709 раз (для семейства Stratix IV) до 127.395 раз (для семейства Arria), по сравнению с традиционным подходом.

Отметим, что при реализации компараторов на FPGA семейства Stratix III и использовании иерархических структур задержка сигналов не превысит 17.112 нс для компараторов на 512 разрядов, 19.101 нс для компараторов на 1024 разряда, 21.551 нс для компараторов на 2048 разрядов, 25.886 нс для компараторов на 4096 разрядов, 32.595 нс для компараторов на 8192 разряда. Поскольку в исследуемых проектах на входах шин данных находятся дешифраторы, можно предположить, что быстродействие внутренних компараторов будет еще больше, чем указано выше.

Отметим также, что метод синтеза иерархических структур позволяет строить компараторы большой разрядности тогда, когда традиционный подход не работает. Это можно видеть из табл. 3, где  $M_T$  – максимальная длина входных слов компараторов, которые могут быть реализованы традиционным способом,  $M_H$  – максимальная длина входных слов компараторов, которые могут быть реализованы при использовании иерархических структур,  $M_T/M_H$  – отношение соответствующих параметров. Из табл. 3 следует, что предлагаемый подход позволяет увеличить максимальную разрядность реализуемых двоичных компараторов для семейств MAX 10, Cyclone V, Arria II и Stratix в 2 раза, для семейства Cyclone в 4 раза, для семейств Cyclone III и Cyclone IV в 8 раз, а для семейства Cyclone II в 16 раз.

Таблица 3

Максимальное число разрядов входных слов компараторов

Семейство	$M_T$	$M_H$	$M_T/M_H$
MAX II	512	512	1
MAX V	512	512	1
MAX 10	512	1024	2
Cyclone	1024	4096	4
Cyclone II	512	8192	16
Cyclone III	1024	8192	8
Cyclone IV	1024	2048	8
Cyclone V	4048	8192	2
Arria	8192	8192	1
Arria II	4096	8192	2
Stratix	4096	8192	2
Stratix II	8192	8192	1
Stratix III	8192	8192	1
Stratix IV	8192	8192	1

Таким образом, изложенная ранее гипотеза, что с помощью иерархических структур можно строить быстрые компараторы большой разрядности полностью подтвердилась.

#### ЗАКЛЮЧЕНИЕ

Представленный метод синтеза иерархических структур компараторов открывает новые возможности проектировщикам цифровой аппаратуры для разработки различных блоков цифровых систем с

использованием быстрых компараторов большой разрядности.

В рассматриваемых иерархических структурах использовались компараторы, длина входных слов которых равнялась степени 2. При необходимости построения компаратора на другое число разрядов достаточно на первом уровне установить компараторы, суммарное число разрядов слов которых равняется требуемой длине слова компаратора. В общем случае иерархические структуры компараторов могут быть реализованы на любой элементной базе, не обязательно на устройствах программируемой логики (например, на ASIC).

Дальнейшее совершенствование метода может идти по пути использования архитектурных особенностей FPGA и SoC, например, использование быстрых цепей переноса, цепей каскадирования, буферов LCELL, путем группирования логики компаратора в кластеры для реализации в одном функциональном блоке, реализации компараторов первого уровня в блоках памяти, в блоках распределенной памяти и др.

Работа выполнена при частичной финансовой поддержке Белостокского технологического университета (Польша), грант № S/WI/1/2013.

#### ЛИТЕРАТУРА

- [1] Соловьев В.В. Проектирование цифровых систем на основе программируемых логических интегральных схем. М.: Горячая линия-Телеком, 2001. 636 с.
- [2] Wang C. C., Wu C. F., Tsai K. C. 1 GHz 64-bit high-speed comparator using ANT dynamic logic with two-phase clocking // IEE Proceedings - Computers and Digital Techniques. 1998. V. 145. №. 6. P. 433-436.
- [3] Huang C. H., Wang J. S. High-performance and power-efficient CMOS comparators // IEEE Journal of Solid-State Circuits. 2003. V. 38. №. 2. P. 254-262.
- [4] Lam H. M., Tsui C. Y. A MUX-based high-performance single-cycle CMOS comparator // IEEE Transactions on Circuits and Systems II: Express Briefs. 2007. V. 54. №. 7. P. 591-595.
- [5] Perri S., Corsonello P. Fast low-cost implementation of single-clock-cycle binary comparator // IEEE Transactions on Circuits and Systems II: Express Briefs. 2008. V. 55. №. 12. P. 1239-1243.
- [6] Abdel-Hafeez S., Gordon-Ross A., Parhami B. Scalable digital CMOS comparator using a parallel prefix tree // IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2013. V. 21. №. 11. P. 1989-1998.
- [7] Chuang P., Li D., Sachdev M. A low-power high-performance single-cycle tree-based 64-bit binary comparator // IEEE Transactions on Circuits and Systems II: Express Briefs. 2012. V. 59. №. 2. P. 108-112.
- [8] Hauser A., Chichester I. High-Speed 64-Bit Binary Comparator using Two Stages // European Journal of Engineering and Innovation. 2013. V. 11. №. 2. P. 29-38.
- [9] Chuang P. I. J., Sachdev M., Gaudet V. C. A 167-ps 2.34-mW Single-Cycle 64-Bit Binary Tree Comparator With Constant-Delay Logic in 65-nm CMOS // IEEE Transactions on Circuits and Systems I: Regular Papers. 2014. V. 61. №. 1. P. 160-171.

- [10] Deb S., Chaudhury S. High-speed comparator architectures for fast binary comparison // Proc. of the Third International Conference on Emerging Applications of Information Technology (EAIT), 2012. IEEE. P. 454-457.
- [11] Соловьев В.В., Посредникова А.А. Реализация на ПЛИС компараторов большой размерности // Chip-News, Инженерная микроэлектроника. 2005. № 9. С. 20-25.
- [12] Solov'ev V.V., Posrednikova A.A. The Hierarchical Method of Synthesis of Large-Capacity Comparators with the Use of Programmable Logic Integrated Circuits // Journal of Communications Technology and Electronics. 2009. V. 54. № 3. P. 338-346.
- [13] Salauyou V., Gruszevski M. Designing of hierarchical structures for binary comparators on FPGA/SoC // Proc. of the 14th Int. Conf. Computer Information Systems and Industrial Management (CISIM). Warsaw. Poland. September 24-26. 2015. Springer. P. 386-396.

## Designing on FPGA and SoC high-performance binary comparators of a big dimensionality

V.V. Salauyou

Belarusian State Academy of Telecommunications (Minsk, Republic of Belarus), valsol@mail.ru

**Keywords** — binary comparators, hierarchical structure, big dimensionality, high-performance, programmable logic devices, FPGA, SoC.

### ABSTRACT

A design method of comparator hierarchical structures on FPGAs and SoCs with big dimensionality and high performance is offered. The method allows to change the number of levels of the comparator hierarchical structure in a wide range. As a result the user can select a compromise between the implementation cost and the performance. The synthesis circuit of the comparator is a completely combinational circuit and it does not contain clock signals therefore the comparator does not require additional circuits for creation of the clocks (unlike known methods). Besides, in the offered approach there are no cascaded carries, all computations are executed by parallel operations that provides very high-speed performance.

The two-level hierarchical structure of comparators consists of the first level comparators and the combinational circuit CL. Each first level comparator  $CMP_n$  is the binary comparator on  $M_1$  bits, which realizes functions "greater"  $g_n$  and "equal"  $e_n$ ,  $n=1, N_1$ .

The combinational circuit CL from the values of the functions generated by the first level comparators forms the values of the output functions "greater"  $G$  and "equal"  $E$  for the whole hierarchical structure.

The hierarchical two-level structure in fig. 1, in turn, can appear as comparators of the first level.

Generally the hierarchical multi-level structure of comparators can be provided as in fig. 1, where comparators  $CMP_1, \dots, CMP_{N_1}$  are at the first level and the combinational circuit CL realizes hierarchical connections of functions "greater" and "equal".

The hierarchical two-level structure can represent itself as comparators of the first level. Continuing in a similar way further, it is possible to build the multi-level hierarchical structure of comparators of the big dimensionality.

The generalized multi-level hierarchical structure of comparators we will designate by means of the formula  $C_{M_T \times N_{T-1} \dots M_t \times N_{t-1} \dots M_2 \times N_1 M_1}$ , where  $M_t$  is the width of words in bits for the comparators of the level  $t$ ,  $t=1, T$ ,  $N_{t-1}$  is the number of comparators on the previous level  $t-1$  (from which the comparators of level  $t$  are constructed). In this formula the following conditions should be satisfied  $M_t = N_{t-1} M_{t-1}$ ,  $t=2, T$ , where  $T$  is the number of logic levels (a depth) of the comparator hierarchical structure.

Note that the various hierarchical structures, with different depths, with various the numbers of units, and the various widths of binary words at each level can be constructed for the same comparator.

As the number of LUT's inputs for the majority of FPGA families it is equal 4, it is offered for all hierarchical structures of comparators on the first level to use 2-bit comparators.

The hierarchical structure by the nature has the least delay the signal propagation in comparison with sequential decomposition. Therefore it is possible hope for that hierarchical structure of the designed comparator will have small of an implementation cost and a high high-speed performance. Moreover, it is possible hope for that by means of hierarchical structures it is possible to build binary comparators of enough big dimensionality.

Hypothesis. The offered technique of the hierarchical structures synthesis of binary comparators allows to build on FPGA and SoC high-performance comparators of the big dimensionality.

Open there is a question: what the formula of the comparator hierarchical structure best approaches for an implementation of the certain dimensionality comparator on the specific family FPGA or SoC. The answer to this question in the offered approach is defined empirically by a performance of a great number of experimental researches.

The hierarchical structures of binary comparators were researched on FPGA and SoC of Altera by using Quartus II design software.

All projects of the comparators were described in language Verilog. The synthesis results of the hierarchical structures of the comparators were compared to a traditional implementation of the comparator in language Verilog. When functions “greater” also are “equal” described by means of appropriate operations in forms: assign  $G = (A > B)$  and assign  $E = (A == B)$ .

Let us mark that thus received results completely coincide with the results that formed at an implementation of the comparators by means of parametrized Altera’s function *lpm\_compare*.

The experiments were executed as follows. All possible hierarchical structures have been constructed for the comparators on 4, 8, 16, 32, 64, 128, and 256 bits, thus 128 comparators projects have been constructed (together with the comparator on 2 bits). For building of the comparators on 512 bits, the best (or close to the best) structures at least for one family FPGA were selected from the earlier constructed structures. The implementation cost (the number of used logical elements) and a high-speed performance (the maximum signal delay) were considered as criteria of an optimization. The hierarchical structures of the comparators on 1024, 2048, 4096 and 8192 bits have been constructed similarly.

The analysis of the experimental research results shows that the offered method allows to increase performance, in comparison with the traditional approach, for comparators on 512 bits by 3.409 times, on 1024 bits by 4.987 times, on 2048 bits by 27.339 times, on 4096 bits by 65.596 times, and on 8192 bits by 127.395 times. The method also allows (for separate FPGA families) by 16 times to increase the dimension of input words for comparators which can be realized on FPGA.

#### REFERENCES

- [1] Salauyou V.V. Designing of digital systems based on programmable logic devices – Moscow, Hot Line-Telecom Publ., 2001. 636 p. (In Russian)
- [2] Wang C. C., Wu C. F., Tsai K. C. 1 GHz 64-bit high-speed comparator using ANT dynamic logic with two-phase clocking. IEE Proceedings - Computers and Digital Techniques. 1998. vol. 145. no. 6. pp. 433-436.
- [3] Huang C. H., Wang J. S. High-performance and power-efficient CMOS comparators. IEEE Journal of Solid-State Circuits. 2003. vol. 38. no. 2. pp. 254-262.
- [4] Lam H. M., Tsui C. Y. A MUX-based high-performance single-cycle CMOS comparator. IEEE Transactions on Circuits and Systems II: Express Briefs. 2007. vol. 54. no. 7. pp. 591-595.
- [5] Perri S., Corsonello P. Fast low-cost implementation of single-clock-cycle binary comparator. IEEE Transactions on Circuits and Systems II: Express Briefs. 2008. vol. 55. no. 12. pp. 1239-1243.
- [6] Abdel-Hafeez S., Gordon-Ross A., Parhami B. Scalable digital CMOS comparator using a parallel prefix tree. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2013. vol. 21. no. 11. pp. 1989-1998.
- [7] Chuang P., Li D., Sachdev M. A low-power high-performance single-cycle tree-based 64-bit binary comparator. IEEE Transactions on Circuits and Systems II: Express Briefs. 2012. vol. 59. no. 2. pp. 108-112.
- [8] Hauser A., Chichester I. High-Speed 64-Bit Binary Comparator using Two Stages. European Journal of Engineering and Innovation. 2013. vol. 11. no. 2. pp. 29-38.
- [9] Chuang P. I. J., Sachdev M., Gaudet VOL. C. A 167-ps 2.34-mW Single-Cycle 64-Bit Binary Tree Comparator With Constant-Delay Logic in 65-nm CMOS. IEEE Transactions on Circuits and Systems I: Regular Papers. 2014. vol. 61. no. 1. pp. 160-171.
- [10] Deb S., Chaudhury S. High-speed comparator architectures for fast binary comparison - Proc. of the Third International Conference on Emerging Applications of Information Technology (EAIT), 2012. IEEE. pp. 454-457.
- [11] Salauyou V.V., Posrednikova A.A. A implementation on FPGA comparators of a big dimensionality. Chip-News, Engineering microelectronics. 2005. no 9. pp. 20-25. (in Russian)
- [12] Solov’ev V.V., Posrednikova A.A. The Hierarchical Method of Synthesis of Large-Capacity Comparators with the Use of Programmable Logic Integrated Circuits. Journal of Communications Technology and Electronics. 2009. vol. 54. no 3. pp. 338-346.
- [13] Salauyou V., Gruszevski M. Designing of hierarchical structures for binary comparators on FPGA/SoC - Proc. of the 14th Int. Conf. Computer Information Systems and Industrial Management (CISIM). Warsaw. Poland. September 24-26. 2015. Springer. pp. 386-396.