

В.В. Соловьев

## Проектирование на ПЛИС компараторов большой размерности

Рассмотрен иерархический метод синтеза на ПЛИС компараторов большого размера, эффективность которого сравнима с двумя методами, реализованными в пакете *MAX+PLUSII* фирмы *Altera*, при проектировании компараторов на 256 разрядов, что позволит снизить их стоимость.

A hierarchical synthesis method of comparators on CPLD/FPGA is considered. The efficiency of the method can be compared with two methods implemented in Altera's CAD *MAX+PLUSII* at designing the comparator on 256 bits. The suggested method will permit of reducing the comparator's cost.

Розглянуто ієрархічний метод синтезу на ПЛИС компараторів великого розміру, ефективність якого можна зрівняти з двома методами, реалізованими в пакеті *MAX+PLUSII* фірми *Altera*, при проектуванні компараторів на 256 розрядів, що дозволить здешевити їх.

**Введение.** В настоящее время наблюдается устойчивая тенденция к возрастанию размерности слов вычислительной техники: 64-разрядные микропроцессоры стали стандартом, на пороге 128-разрядные микропроцессоры. Особенно быстро разрядность слов растет в системах телекоммуникации, а также в устройствах передачи и обработки информации. Наиболее перспективная элементная база электронной техники – программируемые логические интегральные схемы (ПЛИС) [1]. При проектировании сложная цифровая система, как правило, представляется в совокупности стандартных и оригинальных функциональных узлов [2–5]. Поэтому часто при проектировании цифровых систем возникает необходимость в разработке эффективных методов проектирования на ПЛИС стандартных функциональных узлов большого размера (от 64 разрядов и выше). Дело в том, что известные методы построения стандартных функциональных узлов ориентированы на элементы малой степени интеграции, типа вентилей, и совершенно не учитывают особенности новой элементной базы – ПЛИС.

Известно множество методов построения компараторов двоичных чисел [6]: параллельный, последовательный, параллельно-последовательный, с использованием сумматора и др. В параллельном методе проектирования компараторов логические уравнения выходных функций строятся на основе анализа каждого разряда двух сравниваемых слов, начиная со старшего бита. В результате образуются доста-

точно сложные булевы функции, плохо реализуемые на ПЛИС.

В работе [7] предложен многоуровневый параллельный компаратор для сравнения двух бинарных величин большой размерности. На первом уровне располагаются блоки компараторов небольшой размерности, выходы которых подсоединены к входам управляемых вентилях. В схему компаратора введены управляющие элементы (схемы), определяющие блоки первого уровня, сигналы которых передаются на выходы схемы. Входные сигналы для управляющих элементов вырабатываются в блоках первого уровня.

В статье рассматривается проблема проектирования на программируемых логических интегральных схемах (ПЛИС) компараторов большого размера. Предлагается метод синтеза компараторов в виде пирамидальной иерархической структуры, позволяющей строить компараторы очень большого размера. Эффективность предложенного метода сравнивалась с двумя методами, реализованными в пакете автоматизированного проектирования *MAX+PLUSII* фирмы *Altera*. Результаты экспериментальных исследований при проектировании компараторов на 256 разрядов показывают, что новый метод в зависимости от семейства ПЛИС позволяет снизить стоимость реализации (в среднем в 1,09–1,36 раза) и значительно повысить быстродействие (в среднем в 2,30–2,81 раза) компараторов.

## Постановка задачи

Компаратор – это комбинационная схема, выполняющая операции отношения между двумя бинарными числами  $A$  и  $B$  [2]. Стандартный компаратор реализует три функции:  $G$  «больше» ( $A > B$ ),  $E$  «равно» ( $A = B$ ) и  $L$  «меньше» ( $A < B$ ). Прямое построение логических уравнений выходных функций компаратора по таблице истинности даже для небольшого числа  $t$  разрядов входных слов представляет собой нетривиальную задачу. С ростом же разрядности входных слов  $A$  и  $B$  таблица истинности компаратора становится труднообозримой. Например, при  $t = 8$  число строк таблицы истинности составляет  $2^8 \cdot 2^8 = 65536$ . Поэтому для построения даже 8-разрядного компаратора прямое построение логических уравнений выходных функций по таблице истинности практически невозможно.

По способу логических преобразователей (вычислителей) современные ПЛИС можно разделить на два больших класса [1]: ПЛИС, построенные на основе двух программируемых матриц (И и ИЛИ), и ПЛИС – на основе функциональных генераторов небольшого размера (*Look Up Table – LUT*). К первому типу ПЛИС относятся стандартные (*Standard Programmable Logic Devices – SPLD*) и сложные программируемые логические устройства (*Complex Programmable Logic Devices – CPLD*), а ко второму – программируемые пользователем вентильные матрицы (*Field Programmable Gate Array – FPGA*).

Выходы программируемой матрицы И для ПЛИС первого класса называют промежуточными шинами (*product terms*). Для реализации одной элементарной конъюнкции логического выражения требуется одна промежуточная шина ПЛИС. При построении компараторов большого размера на ПЛИС со структурой двух программируемых матриц образуются очень сложные логические выражения, содержащие большое число слагаемых (элементарных конъюнкций), которые плохо поддаются минимизации. Поэтому для своей реализации выходные функции компаратора требуют большого чис-

ла промежуточных шин ПЛИС. В результате приходится прибегать к методам декомпозиции булевых функций [2, 3], что приводит к большой стоимости реализации и невысокому быстродействию компаратора. При построении компараторов на *FPGA* главная проблема заключается в небольшом числе входов функциональных генераторов *LUT*, что также приводит к необходимости декомпозиции булевых функций большого числа аргументов. Таким образом, компараторы относятся к таким функциональным узлам, которые плохо реализуются как на ПЛИС типа *PLD*, так и на ПЛИС типа *FPGA*.

С другой стороны, необходимость в эффективной реализации различных схем сравнения в современных цифровых системах весьма актуальна, особенно в системах передачи информации. Например, при наличии схемы сравнения большого размера можно не вычислять контрольную сумму повторно передаваемого блока информации, а непосредственно сравнивать его содержимое. Поэтому актуальна задача отработки методов синтеза компараторов большого размера (256 разрядов и более), позволяющих строить на ПЛИС компараторы низкой стоимости и высокого быстродействия.

## Иерархический метод синтеза компаратора

При построении компараторов большого размера достаточно реализовать только две функции, например, функцию «больше»  $G$  и функцию «равно»  $E$ , поскольку третья функция, в данном случае функция «меньше»  $L$ , всегда может быть определена на основе первых двух:  $L = \overline{G} \& \overline{E}$ , где символ  $\&$  означает логическое И.

В основу предлагаемого метода проектирования компараторов большого размера положены следующие идеи:

- компаратор большого размера строится в виде пирамидальной иерархической структуры из блоков меньшего размера;
- в качестве блоков иерархической структуры используются компараторы меньшего размера.

Например, необходимо построить компаратор на 16 разрядов из четырех блоков по четыре разряда. Двухуровневая структура такого компаратора показана на рисунке, где на первом уровне схемы располагаются блоки, представляющие собой четырехразрядные компараторы. Некоторый  $i$ -й компаратор первого уровня сравнивает определенный диапазон разрядов двоичных чисел  $A$  и  $B$  и вычисляет две логические функции: функцию «равно»  $e_i$  и функцию «больше»  $g_i$ ,  $i = \overline{1, F}$ .

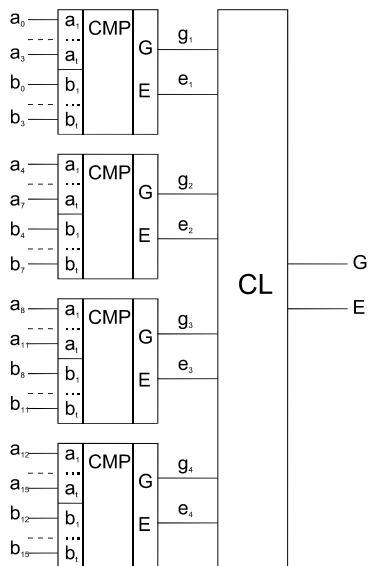


Рис. Структура компаратора на 16 разрядов из четырех блоков по 4 разряда

Выходную функцию  $G$  «больше» комбинационной схемы  $CL$  на рисунке можно построить исходя из следующих предположений. Если  $g_4 = 1$ , то, независимо от значений на выходах других блоков, имеем  $A > B$ . Если  $g_4 = 0$  и  $e_4 = 1$ , то следует проверить значение  $g_3$ . Если  $g_3 = 1$ , то  $A > B$  вне зависимости от значений на выходах других блоков. Продолжая подобные рассуждения, можно получить следующую формулу для функции больше:

$$\begin{aligned}
 G &= g_4 \\
 &+ e_4 \cdot g_3 \\
 &+ e_4 \cdot e_3 \cdot g_2 \\
 &+ e_4 \cdot e_3 \cdot e_2 \cdot g_1.
 \end{aligned}
 \tag{1}$$

Расширяя (1) на общий случай для  $F$  блоков, получим:

$$\begin{aligned}
 G &= g_F \\
 &+ e_F \cdot g_{F-1} \\
 &+ e_F \cdot e_{F-1} \cdot g_{F-2} \\
 &\dots \\
 &+ e_F \cdot e_{F-1} \cdot \dots \cdot e_2 \cdot g_1.
 \end{aligned}
 \tag{2}$$

Выходная функция  $E$  «равно» реализуется совсем просто: функция  $E$  равна единице тогда, когда все функции  $e_i$ ,  $i = \overline{1, F}$  равны единице:

$$E = e_1 \cdot \dots \cdot e_F.
 \tag{3}$$

В рассматриваемом методе для построения комбинационной схемы  $CL$  на рисунке используются логические уравнения (2) и (3).

Построенный таким образом двухуровневый компаратор в свою очередь может рассматриваться как отдельный блок для построения компараторов большего размера. Действуя далее подобным образом, можно построить компаратор достаточно большого, практически неограниченного, размера.

Отличие предлагаемого подхода от параллельно-последовательного метода [2, 4] заключается в следующем:

- секции в параллельно-последовательном методе соединяются последовательно; в предлагаемом методе блоки соединяются параллельно;
- для передачи значений сигналов между секциями в параллельно-последовательном методе используются промежуточные функции; в предлагаемом методе значения сигналов между блоками одного уровня не передаются, поэтому нет необходимости в промежуточных функциях;
- каждая секция в параллельно-последовательном методе имеет дополнительные входы для приема значений выходных функций от соседней секции; в предлагаемом методе блоки имеют меньшее число входов на величину реализуемых функций;
- выходные функции, реализуемые секциями в параллельно-последовательном методе, имеют большее число аргументов и сложнее для реализации в сравнении с выходными функциями блоков предлагаемого метода.

В отличие от [7] в предлагаемой структуре отсутствуют управляющие вентили и управля-

ющие ими схемы, а также дополнительные сигналы для управляющих схем, вырабатываемые блоками компараторов первого уровня.

### Построение иерархических структур при реализации компаратора на 256 разрядов

При использовании иерархического метода синтеза компараторов большого размера на ПЛИС возникают следующие задачи:

- определить для каждого класса ПЛИС наилучшие методы для синтеза блоков первого уровня;

- определить для каждого очередного уровня иерархической структуры и для каждого класса ПЛИС наилучшие структуры, которые будут использоваться в качестве блоков при построении следующего уровня компаратора.

Ответы на поставленные задачи получены с помощью экспериментальных исследований, т.е. эмпирически. Для этого в качестве ПЛИС рассматривались устройства фирмы *Altera* следующих классов:

- *CPLD* – семейства *MAX3000A*, *MAX7000AE* и *MAX9000*;

- *FPGA* – семейства *FLEX6000*, *FLEX10KE* и *ACEX1K*.

Из исследуемых ПЛИС исключены семейства *CLASSIC* и *FLEX8000* фирмы *Altera*, поскольку на их устройствах редко строятся компараторы на 256 разрядов.

Для проектирования компараторов использовался пакет *MAX+PLUSII* [8] фирмы *Altera* версии 10.2. Параметры логического синтеза пакета *MAX+PLUSII* устанавливались на использование всех архитектурных возможностей каждого семейства ПЛИС.

Критерии оптимизации метода синтеза – стоимость реализации  $C$ , определяемая как число требуемых для реализации макроячеек *CPLD* или логических элементов *FPGA*, и быстродействие  $D$ , определяемое как максимальная задержка прохождения сигналов с входов на выходы схемы, измеряемая в наносекундах.

В качестве наименьших блоков компараторов первого уровня были приняты компараторы на четыре и восемь разрядов, а для построения их

блоков – использовались следующие базовые методы:

- *lpm* – реализация компаратора в графическом редакторе пакета *MAX+PLUSII* с помощью параметризированной функции *lpm\_compare*;

- *ahdl* – реализация компаратора в текстовом редакторе пакета *MAX+PLUSII* средствами языка *AHDL*;

- *par* – реализация компаратора с помощью параллельного метода синтеза [6];

- *seq* – реализация компаратора с помощью последовательного метода синтеза [6].

Введем способ обозначения многоуровневой иерархической структуры компаратора. Двухуровневая структура компаратора обозначается с помощью следующей формулы:

$$M\_NxT\_method, \quad (4)$$

где *method* – один из базовых методов синтеза, используемый для построения компараторов первого уровня,  $method \in \{lpm, ahdl, par, seq\}$ ;  $T$  – разрядность блоков первого уровня;  $N$  – число блоков первого уровня;  $M$  – разрядность блока второго уровня (т.е. собственно разрядность компаратора),  $M = N \cdot T$ . При построении следующего уровня иерархической структуры выражение (4) повторяется, только вместо значения  $T\_method$  используется формула блока нижнего уровня.

Суффикс  $\_L$  в обозначении структуры указывает на наличие буферов *LCELL* [9] на выходах блоков первого уровня. Буфер *LCELL* представляет собой простейший буфер с одним входом и одним выходом. Установка такого буфера в схему означает, что данная точка цепи обязательно будет реализована на выходе макроячейки ПЛИС. Фактически с помощью буферов *LCELL* осуществляется декомпозиция сложных схем, что в отдельных случаях может привести к снижению стоимости реализации и повышению быстродействия схемы. Например, запись  $64\_4x16\_4x4\_lpm\_L$  обозначает трехуровневую иерархическую структуру компаратора на 64 разряда, состоящую из четырех блоков по 16 разрядов; в свою очередь блоки на 16 разрядов состоят из четырех блоков по четыре разряда первого уровня, реали-

зованных с помощью метода *lpm*. Кроме того, на выходах блоков первого уровня установлены буферы *LCELL*.

Очевидно, что имеется достаточно большое разнообразие иерархических структур, реализующих компаратор на 256 разрядов. Структуры отличаются друг от друга используемым методом синтеза для реализации блоков первого уровня, числом уровней, а также разрядностью блоков каждого уровня. Однако каждая иерархическая структура однозначно описывается с помощью рассмотренной выше формулы.

Задача заключается в выборе для каждого семейства ПЛИС такой структуры компаратора, которая обеспечивает наименьшую стоимость реализации, структуры компаратора, в свою очередь обеспечивающей наибольшее быстроедействие, а также нахождение для каждого семейства ПЛИС номинальной структуры.

Часто структура компаратора, обеспечивающая наименьшую стоимость реализации, имеет достаточно большую задержку и наоборот, структура, обеспечивающая наименьшую задержку, имеет большую стоимость реализации. Поэтому с практической точки зрения целесообразно выбрать структуру с низкой стоимостью, пусть и не наименьшей, и с небольшой задержкой, также не наименьшей. Такую структуру компаратора для данного семейства ПЛИС будем называть *номинальной* структурой.

Остановимся на вопросе выбора блоков для построения компараторов следующих уровней. Пусть блоки первого уровня для построения иерархических структур компараторов выбираются из четырехразрядных компараторов. Вначале выполняется синтез компараторов на четыре разряда с помощью базовых методов. Для построения иерархических структур следующего уровня (компараторов на восемь разрядов) выбираются блоки, для которых  $C = \min$  и/или  $D = \min$  хотя бы для одного семейства ПЛИС, а также номинальные структуры.

Затем анализируются компараторы на восемь разрядов, построенные с помощью базовых методов, а также двухуровневые иерархические структуры из блоков по четыре разряда.

Из построенных компараторов на восемь разрядов аналогично выбираются блоки для построения компараторов на 16 разрядов. Продолжая данный процесс, получаем ряд иерархических структур для реализации компараторов на заданное число разрядов (в нашем случае на 256 разрядов).

В процессе применения метода при реализации компаратора на 256 разрядов было построено 28 различных иерархических структур компараторов нижних уровней. Окончательные результаты приведены в табл. 1, где  $C_{min}$  – структура, при которой достигается минимальная стоимость;  $D_{min}$  – структура, при которой достигается минимальная задержка;  $Nom$  – структура, предлагаемая в качестве номинальной;  $C$  – стоимость реализации;  $D$  – задержка в наносекундах.

Т а б л и ц а 1. Лучшие иерархические структуры для построения компараторов на 256 разрядов

ПЛИС	Тип оптимизации	Структура	$C^*$	$D$
MAX3000A	$C_{min}$	256_16x16_4x4_ <i>par</i> ;	334(3)	74,8
	$D_{min}$	256_16x16L_2x8_ <i>ahdl</i> ;	436(3)	31,0
	$Nom$	256_2x128_16x8_2x4_ <i>par</i> L; 256_4x64_4x16_4x4_ <i>ahdl</i> L; 256_4x64_2x32_2x16_4x4_ <i>lpm</i> L;	342(4) 337(3) 337(3)	35,2 40,1 40,1
MAX7000AE	$C_{min}$	256_16x16_4x4_ <i>par</i> ;	334(3)	70,8
	$D_{min}$	256_16x16L_2x8_ <i>ahdl</i> ;	436(3)	28,7
	$Nom$	256_16x8_2x4_ <i>par</i> L; 256_4x64_4x16_4x4_ <i>ahdl</i> L; 256_4x64_2x32_2x16_4x4_ <i>lpm</i> L;	342(3) 337(3) 337(3)	36,9 40,1 40,1
MAX9000	$C_{min}$	256_4x64_4x16_4x4_ <i>ahdl</i> L; 256_4x64_2x32_2x16_4x4_ <i>lpm</i> L; 256_8x32_8x4_ <i>lpm</i> L; 256_8x32_2x16_4x4_ <i>ahdl</i> L;	339(6) 339(5) 339(6) 339(5)	87,7 91,5 96,8 94,1
	$D_{min}$	256_16x16L_2x8_ <i>ahdl</i> ;	469(6)	56,7
	$Nom$	256_2x128_2x64_2x32_2x16_4x4_ <i>lpm</i> L	345(6)	75,9
FLEX6000	$C_{min}$	256_2x128_2x64_2x32_4x8_ <i>ahdl</i> ;	393(3)	48,9
	$D_{min}$	256_2x128_2x64_2x32_4x8_ <i>ahdl</i> ;	393(3)	48,9
	$Nom$	256_2x128_2x64_2x32_4x8_ <i>ahdl</i> ;	393(3)	48,9
FLEX10K	$C_{min}$	256_8x32_2x16_2x8_ <i>ahdl</i> L; 256_2x128_16x8_ <i>ahdl</i> L;	384(3) 384(3)	50,8 40,7
	$D_{min}$	256_2x128_2x64_8x8_ <i>ahdl</i> ;	385(3)	22,6
	$Nom$	256_2x128_2x64_8x8_ <i>ahdl</i> ;	385(3)	22,6
ACEX	$C_{min}$	256_8x32_2x16_2x8_ <i>ahdl</i> L; 256_2x128_16x8_ <i>ahdl</i> L;	384(2) 384(2)	29,0 47,1
	$D_{min}$	256_2x128_2x64_8x8_ <i>ahdl</i> ;	385(2)	23,5
	$Nom$	256_2x128_2x64_8x8_ <i>ahdl</i> ;	385(2)	23,5

\* В случае реализации компаратора на нескольких ПЛИС в скобках указывается число микросхем.

Анализ табл. 1 позволяет сделать следующие выводы:

- для *FLEX/ACEX*-устройств лучшие иерархические структуры на первом уровне имеют восьмиразрядные блоки, построенные с помощью метода *ahdl*, с буферами *LCELL* или без них;

- для *MAX*-устройств лучшие по быстродействию иерархические структуры на первом уровне имеют восьмиразрядные блоки, построенные с помощью метода *ahdl*, без буферов *LCELL*;

- для *MAX*-устройств лучшие по быстродействию иерархические структуры на первом уровне имеют восьмиразрядные блоки, построенные с помощью метода *par* (*MAX3000A*, *MAX7000AE*), *ahdl* и *lpm* (*MAX9000*).

Поскольку лучшие иерархические структуры компараторов на первом уровне имеют четырех- и восьмиразрядные блоки, то актуально тщательное проектирование эффективных по стоимости и быстродействию блоков, с широким использованием архитектурных возможностей конкретного семейства ПЛИС.

### Оценка эффективности иерархического метода синтеза при реализации компаратора на 256 разрядов

Проведенные экспериментальные исследования не дают полного ответа на вопрос: на сколько предлагаемый метод построения иерархических структур компараторов лучше традиционных подходов. Для ответа на него в качестве традиционных методов были выбраны

методы построения компараторов, реализованные в пакете *MAX+PLUSII*: элемент *lpm\_compare* из библиотеки мегафункций графического редактора и встроенные средства языка *AHDL*, использующие параметризованные функции. Данные методы получили название соответственно *t\_lpm* и *t\_ahdl*, где *t* – число разрядов компаратора.

Заметим, что методы синтеза компараторов, реализованные в пакете *MAX+PLUSII*, разрабатывались специалистами высшей квалификации, что позволяет учитывать архитектурные особенности конкретной ПЛИС фирмы *Altera*. Более того, эта фирма не рекомендует разрабатывать собственные проекты компараторов, а использовать указанные выше мегафункции, реализованные в пакете *MAX+PLUSII*.

Параметры компараторов (стоимость реализации и быстродействие), синтезированных с помощью традиционных методов, сравнивались с соответствующими параметрами компараторов, построенных в виде различных иерархических структур. Результаты оценки эффективности иерархического метода построения компараторов на 256 разрядов приведены в табл. 2. Здесь приняты следующие обозначения:  $C_L$ ,  $C_A$ ,  $C_I$  и  $C_N$  – стоимость реализации компаратора, спроектированного с помощью метода *256\_lpm*, *256\_ahdl*, лучшей по стоимости иерархической структурой и номинальной структурой соответственно;  $D_L$ ,  $D_A$ ,  $D_I$  и  $D_N$  – быстродействие компаратора, спроектированного с помощью метода *256\_lpm*, *256\_ahdl*, лучшей по стоимо-

Таблица 2. Оценка эффективности иерархических структур компараторов на 256 разрядов

ПЛИС	256_lpm		256_ahdl		Лучшие для всех структур		Улучшение для lpm		Улучшение для ahdl		Номинальный		Улучшение для lpm		Улучшение для ahdl	
	$C_L$	$D_L$	$C_A$	$D_A$	$C_I$	$D_I$	$C_I/C_I$	$D_I/D_I$	$C_A/C_I$	$D_A/D_I$	$C_N$	$D_N$	$C_I/C_N$	$D_I/D_N$	$C_A/C_N$	$D_A/D_N$
<i>MAX3000A</i>	<sup>1)</sup>	<sup>1)</sup>	445	88,9	334	31,0	–	–	1,33	2,86	337	40,1	–	–	1,32	2,22
<i>MAX7000AE</i>	<sup>1)</sup>	<sup>1)</sup>	445	84,5	334	28,7	–	–	1,33	<b>2,94</b>	337	40,1	–	–	1,32	2,11
<i>MAX9000</i>	<sup>1)</sup>	<sup>1)</sup>	479	149,0	339	56,7	–	–	<b>1,41</b>	2,63	345	75,9	–	–	1,38	1,96
<i>mid_MAX</i>							–	–	1,36	2,81			–	–	1,34	2,10
<i>FLEX6000</i>	429	63,8	430	99,2	393	48,6	1,09	1,31	1,09	2,04	393	48,9	1,09	1,30	1,09	2,03
<i>FLEX10KE</i>	416	23,8	417	59,6	384	22,6	1,08	1,05	1,09	<b>2,64</b>	385	22,6	1,08	1,05	1,08	2,64
<i>ACEX1K</i>	418	30,1	418	52,4	384	23,5	1,09	1,28	1,09	2,23	384	29,0	1,09	1,04	1,09	1,81
<i>mid_FLEX</i>							1,09	1,21	1,09	2,30			1,09	1,13	1,09	2,16
<i>mid</i>							1,09	1,21	1,22	2,56			1,09	1,13	1,22	2,13

<sup>1)</sup> ошибка компилятора

сти иерархической структурой и номинальной структурой соответственно;  $C_L/C_I$  – отношение соответствующих величин;  $mid_{MAX}$  – среднее значение отношений для *MAX*-устройств;  $mid_{FLEX}$  – среднее значение отношений для *FLEX*-устройств;  $mid$  – среднее значение отношений для всех ПЛИС.

Однако отношения значений стоимости реализации и быстродействия для *MAX*- и *FLEX*-устройств значительно разнятся. Поэтому для более точной оценки эффективности иерархического метода введены величины  $mid_{MAX}$  и  $mid_{FLEX}$ . Отметим также, что метод *256\_lpm* не применим для построения компараторов на 256 разрядов для *MAX*-устройств из-за ошибки компилятора пакета *MAX+PLUSII*.

Таким образом, использование метода синтеза иерархических структур при проектировании 256-разрядных компараторов на *FLEX/ACEX*-устройствах позволяет на восемь–девять процентов уменьшить стоимость реализации, однако значительно повысить быстродействие, в среднем в 1,21÷2,30 раза (в отдельных случаях в 2,64 раза). Использование метода синтеза иерархических структур при проектировании 256-разрядных компараторов на *MAX*-устройствах позволяет заметно снизить стоимость реализации в среднем в 1,36 раза (в отдельных случаях в 1,41 раза) и значительно повысить быстродействие, в среднем в 2,81 раза (в отдельных случаях в 2,94 раза).

Выводы, сделанные для лучших иерархических структур, в общем случае остаются справедливыми и для номинальных структур. Отличие заключается в том, что преимущество номинальных иерархических структур не столь значительно, в среднем в 1,09÷1,34 раза по стоимости и в 1,13÷2,16 раза по быстродействию.

**Заключение.** Предложенный метод может быть применен для построения компараторов на любой элементной базе (не обязательно ПЛИС фирмы *Altera*) и с помощью любого пакета автоматизированного проектирования (не обязательно *MAX+PLUSII*).

Отметим также, что при построении компараторов большого размера использование встроенных блоков памяти *EAB* (*Embedded Array Block*) семейств *FLEX10KE* и *ACEX1K* себя не оправдало. Не смотря на то что реали-

зация логических функций с помощью блоков памяти *EAB* несколько уменьшает число используемых логических элементов, при этом значительно возрастает время задержки сигналов.

Буферы *LCELL* могут устанавливаться не только на выходах блоков первого уровня, но на выходах блоков любого уровня. Место установки буферов *LCELL* индивидуально для каждой иерархической структуры и для каждого семейства ПЛИС. В отдельных случаях это может привести к значительному уменьшению стоимости реализации и величины задержки (например, структура *256\_16x16L\_2x8\_ahdl* для семейства *MAX9000*).

Однако не существует одной лучшей иерархической структуры для построения компараторов на 256 разрядов для всех семейств ПЛИС. В то же время для каждого семейства ПЛИС можно указать одну или несколько иерархических структур, лучших по стоимости, а также структуру, лучшую по быстродействию.

Дальнейшее совершенствование метода может идти по пути разработки эффективных методов проектирования блоков компараторов первого уровня (на четыре и восемь разрядов), которые широко используют архитектурные возможности конкретного семейства ПЛИС, а также использовании быстрых цепей между функциональными блоками ПЛИС для передачи сигналов между уровнями иерархической структуры компаратора.

1. Соловьев В.В., Васильев А.Г. Программируемые логические интегральные схемы и их применение. – Минск: Белорусская наука, 1998. – 270 с.
2. Соловьев В.В. Проектирование функциональных узлов цифровых систем на программируемых логических устройствах. – Минск: ПК ООО «Бест-принт», 1996. – 252 с.
3. Соловьев В.В. Проектирование цифровых систем на основе программируемых логических интегральных схем. – М.: Горячая линия–Телеком, 2001. – 636 с.
4. Потемкин И.С. Функциональные узлы цифровой автоматики. – М.: Энергоатомиздат, 1988. – 320 с.
5. Угрюмов Е.П. Цифровая схемотехника. – Санкт-Петербург: БХВ-Петербург, 2002. – 528 с.
6. Соловьев В.В., Посредникова А.А. Реализация на ПЛИС компараторов большой размерности // ChipNews, Инженерная микроэлектроника. – 2005. – № 9. – С. 20–25.
7. Mc.Clure D.C. Multiple level parallel magnitude comparator. – United State Patent, US005400007, Mar. 21, 1995. – <http://patft.uspto.gov>
8. *MAX+PLUSII* Programmable Logic Development System. – Altera Corporation, 1991. – 334 p.
9. [http://www.altera.com/literature/ug/ug\\_low\\_level.pdf](http://www.altera.com/literature/ug/ug_low_level.pdf)

Поступила 22.10.2008

Тел. для справок: (+4885) 746-9111, (Белосток, Польша)

E-mail: [walsol@ii.pb.bialystok.pl](mailto:walsol@ii.pb.bialystok.pl)

© В.В. Соловьев, 2009