

Д. т. н. В. П. МАЛАХОВ, к. т. н. В. С. СИТНИКОВ,  
П. В. ЯДВИЧУК

Украина, Одесский национальный политехнический  
университет  
E-mail: sitnv@promel.ospu.odessa.ua

Дата поступления в редакцию  
08.01 2004 г.

Оппонент д. т. н. С. Ю. ЛУЗИН  
(ОАО "Авангард", г. С.-Петербург)

## РЕАЛИЗАЦИЯ ЭЛЕМЕНТОВ БЫСТРОДЕЙСТВУЮЩЕГО ЦИФРОВОГО ФИЛЬТРА НА ПЛИС

*Результаты исследования позволяют говорить о ПЛИС как о наиболее пригодной базе для реализации высокопроизводительных фильтров.*

Для повышения точности и качества систем управления широко используется цифровая обработка сигналов. В настоящее время разработано большое количество методов обработки сигналов, однако их применение в системах реального времени ограничено из-за невысоких вычислительных возможностей микропроцессорной техники.

Перед разработчиком ставятся сложные и противоречивые задачи — выбор структурной организации и элементной базы для компонентов системы, разрядность данных и коэффициентов, метод кодирования данных, точность вычислений и т. п. [1]. В связи с ограниченным быстродействием возникает потребность упрощения вычислительных алгоритмов обработки, которая ведет к ухудшению качества фильтрации, а значит, к ухудшению технических характеристик системы в целом. Таким образом, возникает необходимость поиска новой элементной базы для проектирования более быстродействующих и более гибких систем.

При решении задач цифровой обработки сигналов (ЦОС), например, в гидролокации, радиолокации, сейсморазведке часто используются не рекурсивные фильтры, у которых отсчеты выходного сигнала определяются только отсчетами входного сигнала [2]:

$$y(n) = \sum_{i=0}^m b_i \cdot x(n-i), \quad (1)$$

где  $x(n)$  и  $y(n)$  — отсчеты, соответственно, входного и выходного сигналов в момент времени  $n$ ;  
 $b$  — коэффициенты фильтра.

При проектировании цифрового фильтра необходимо учесть ряд факторов, которые влияют на качество разработки:

— Производительность системы  $P$ , MIPS. Она определяется как

$$P = \frac{OP}{s}, \quad (2)$$

где  $OP$  — количество простых операций, которые система может выполнять одновременно;  
 $s$  — время решения алгоритма.

— Автоматизация процесса проектирования. Со всем недавно это требование из второстепенных перешло в разряд важнейших. Оно отражает приспособленность избранного варианта реализации к этапам сквозного проектирования: начиная с разработки алгоритма и заканчивая подготовкой к производству печатных плат, настройкой монтажных роботов и паяльных станций.

— Логическая емкость. Параметр, оценивающий сложность реализуемой структуры устройства. Измеряется он количеством структурных макроячеек, используемых схемой.

— Время проектирования системы. Параметр, зависящий от квалификации разработчиков, сложности системы, наличия технической базы для тестирования и т. п.

— Стоимость системы. Этот параметр в условиях рыночной экономики все чаще выступает на первое место при выборе варианта реализации системы и элементной базы.

— Массогабаритные показатели.

Использование современных микроконтроллеров и микропроцессоров общего назначения позволило реализовать цифровые фильтры с производительностью не более 0,01 MIPS.

В последнее время стандартным подходом к реализации систем ЦОС является использование цифровых сигнальных процессоров (DSP) [1], которые содержат специализированные команды векторной арифметики, широко используемой в алгоритме (1). Достоинствами такого подхода являются простота реализации алгоритма обработки, сокращение времени проектирования, низкая стоимость реализации алгоритма. Архитектура сигнальных процессоров строится в соответствии с гарвардским типом реализации вычислительных систем. При этом решение уравнения фильтрации ведется последовательно, что не позволяет получить высокое быстродействие, хотя, в общем, производительность устройства резко возросла и составила 1—15 MIPS.

Однако для ряда задач обработки сигналов, решаемых на современном этапе развития электроники, и такие показатели производительности обработки уже не могут считаться удовлетворительными. Поиск возможных решений пошел в направлении создания цифровых устройств и систем на микросхемах с перестраиваемой архитектурой. Программируемые логические интегральные схемы (ПЛИС) позволяют реа-

лизовать параллельные вычисления с практически произвольной разрядностью коэффициентов и данных, а также разместить на одной микросхеме вычислительную, контрольную и управляющую части системы. При этом из-за особенностей архитектурной организации микросхемы ПЛИС нельзя рассматривать как замену логических микросхем традиционных серий [2].

В настоящей работе рассмотрена задача реализации быстродействующего нерекурсивного цифрового фильтра на ПЛИС.

Анализ алгоритма (1) показывает, что выходные отсчеты зависят от входных данных и коэффициентов фильтра, тогда в формуле (1) можно выделить несколько регулярных структурных единиц (задержку входного сигнала на такт, умножение отсчетов задержанного сигнала на константу, суммирование отсчетов), которые при реализации нерекурсивного фильтра будут представлены:

- многоразрядным сдвиговым регистром — для операции задержки сигнала на такт;
- векторным двоичным множителем — для операции умножения на коэффициенты фильтра;
- векторным двоичным сумматором — для операции сложения отсчетов и формирования выходного сигнала.

Следовательно, для реализации быстродействующего нерекурсивного фильтра на ПЛИС необходимо исследовать варианты описания и реализации составных элементов цифрового фильтра.

В работе исследования проведены на примере использования ПЛИС Spartan2 XC2S100-6TQ208 фирмы Xilinx и языка описания VHDL.

Регистр в составе фильтра осуществляет операцию задержки отсчетов входного сигнала на один такт  $Z^{-1}$ . Поэтому его можно рассматривать как многоразрядный элемент памяти [3, с. 79]

$$\begin{cases} Q^T = d; clk = \square; \\ Q^T = Q^{T-1}; clk = 0, clk = 1, clk = \square, \end{cases} \quad (3)$$

где  $Q^T$  — состояние выходного сигнала регистра в момент времени  $T$ ;

$d$  — сигнал на информационном входе регистра в момент времени  $T$ ;

$clk$  — сигнал на тактовом входе регистра.

В ходе исследований был проведен синтез нескольких вариантов описания регистров: на основе LPM-компонент и совокупности компонентов триггеров из библиотеки фирмы-производителя микросхем, а также на основе описания типов данных (векторных и численных). Результаты синтеза оказались одинаковыми для разных видов описания.

Описание регистра на языке VHDL с использованием численных типов данных показано в листинге 1.

```
Листинг 1
architecture RegProc of RegProc is
  subtype byte is integer range 0 to 255;
  signal clk, ClkEn : std_Logic;
  signal d,q       : byte;
  signal nrst, nset : std_logic;
begin
```

```
LReg: process(clk) begin
  if clk = '1' and clk'iyent then
    if nrst='0' then
      q<=0;
    elsif nset='1' then
      q<=byte' high;
    elsif ClkEn='1' then
      q<=d;
    end if;
  end if;
end process;
end RegProc;
```

Из отчета программы-синтезатора (листинг 2) при компиляции и синтезе регистра видно, что в микросхеме выделено 8 макроячеек (строка 3), которые сконфигурированы для выполнения функции динамического триггера (строка 4). В соответствии с определением логической емкости устройства она будет определяться количеством макроячеек, используемых схемой на кристалле ПЛИС (строка 4). Минимальная длительность тактовых импульсов в этом случае равна 2,829 нс (строка 6) или максимальная тактовая частота равна 353,482 МГц, что соответствует производительности в 353 MIPS.

Листинг 2

```
1 Macro Statistics
2 # Registers           : 1
3 8-bit register       : 1
4 Number of Slice Flip Flops: : 8
5 Clock Information:
6 Minimum input clock time : 2.829ns
```

Разрядность регистров определяется разрядностью данных, а их количество — порядком или глубиной фильтра. В листинге 3 приведено описание многоразрядного сдвигового регистра.

Листинг 3

```
architecture RegProc of RegProc is
  type TShiftReg is array (0 to TapsCount-1) of byte;
  signal ShiftReg : TShiftReg;
begin
  LReg: process(clk) begin
    if Rise(clk) then
      if res='0' then
        for i in 0 to TapsCount-1 loop
          ShiftReg(i)<=0;
        end loop;
      elsif ClkEn='1' then
        ShiftReg(0)<=d;
        for i in 1 to TapsCount-1 loop
          ShiftReg(i)<= ShiftReg(i-1);
        end loop;
      end if;end if;
    end process;
    q<=ShiftReg(TapsCount-1);
  end RegProc;
```

Отчет программы-синтезатора показывает, что максимальная тактовая частота равна 353 МГц (листинг 4). Это предельная частота. Дальнейший рост ее ограничен технологическими особенностями микросхем. Полученные функциональные блоки регистров отличаются большим быстродействием (за счет реализации синхронного обмена данными) и маленькой логической емкостью.

Листинг 4

```

1 Macro Statistics
2 # Registers           : 8
3 8-bit register       : 8
4 Number of Slice Flip Flops: : 64
5 Clock Information:
6 Minimum period: 2.829ns (Maximum Frequency:
353.482MHz)
    
```

Сумматор осуществляет операцию суммирования отсчетов входного сигнала и промежуточных вычислений [3, с. 133]

$$S = A \oplus B \oplus c; \quad \text{Cout} = AB + Ac + Bc, \quad (4)$$

где  $S$  — сумма двух двоичных разрядов;  
 $A, B$  — биты первого и второго операндов;  
 $c$  — входной перенос;  
 $\text{Cout}$  — выходной перенос.

Уравнение (4) описывает работу одноразрядного сумматора. Для увеличения разрядности используют каскадное включение одноразрядных сумматоров. Однако для получения одного бита суммы достаточно одной макроячейки микросхемы ПЛИС, т. к. генератор логических функций (LUT), входящий в состав макроячейки ПЛИС, это четырехходовая память, которая может реализовать любую функцию четырех переменных (листинг 5).

Листинг 5

```

architecture CompAdderArc of CompAdder is
signal Ta, Tb, : std_logic_vector(width-1 downto 0);
signal Tq, Tc : std_logic_vector(width-1 downto 0);
begin
Ta<=data;
Tb<=datb;
--***** Serial Carry chain adder ****
Tq(0)<= Ta(0) xor Tb(0) xor cin;
Tc(0)<=(Ta(0) and Tb(0)) or (Ta(0) and cin) or
(Tb(0) and cin);
gen_adder: for i in 1 to width-1 generate
begin
Tq(i)<= Ta(i) xor Tb(i) xor Tc(i-1);
Tc(i)<=(Ta(i) and Tb(i)) or (Ta(i) and
Tc(i-1)) or (Tb(i) and Tc(i-1));
end generate;
q<=Tq;
end CompAdderArc;
    
```

Принадлежность сумматора к регулярным структурам позволяет компактно описывать его с использованием конструкции GENERATE [4, с. 69]. В результате упаковки этой программы в кристалл ПЛИС программа-синтезатор дает отчет, приведенный в листинге 6.

Листинг 6

```

1 Macro Statistics
2 # Xors                 : 8
3 1-bit xor3            : 8
4 Number of 4 input LUTs:      8
5 Timing Summary:
6 Minimum period: 13.755ns (Maximum Frequency:
72.701MHz)
    
```

Из отчета следует, что синтезатором создан на кристалле ПЛИС массив логических элементов, которые реализуют вычисления по формуле (4). Схема

занимает на кристалле восемь макроячеек (строка 3). Максимальная тактовая частота для этой схемы может быть 72 МГц (производительность 72 MIPS).

Отметим, что для формирования логической функции выбран базис элементов ЛОГИЧЕСКОЕ ИСКЛЮЧИТЕЛЬНОЕ ИЛИ (строки 2 и 3). Таким образом, синтезатор не распознал в логических уравнениях схемы полного сумматора и синтезировал логическую схему, которая реализует алгоритм суммирования. Данная реализация не использует конструктивно имеющиеся в микросхеме быстродействующие цепи распространения переноса, что является ее недостатком. С увеличением разрядности операндов логическая емкость растет линейно, а производительность уменьшается обратно пропорционально логической емкости (рис. 1).

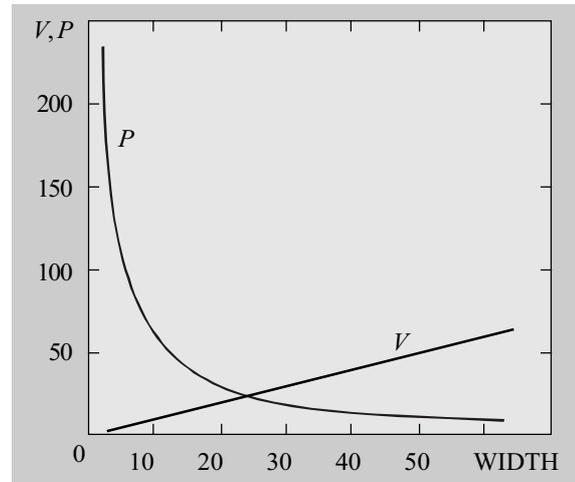


Рис. 1. Зависимость логической емкости  $V$  и производительности  $P$  сумматора от разрядности операндов  $WIDTH$

Для использования выделенных цепей распространения переноса описание сумматора должно иметь вид, показанный в листинге 7.

Листинг 7

```

signal Ta, Tb, RegOut:std_logic_vector(width-1
downto 0);
process(clk) begin
if Rise(clk) then
if res='1' then
RegOut <= (others=>'0');
Else
RegOut <= signed(Ta)+signed(Tb);
end if;
end if;
end process;
    
```

В этом случае синтезатор построил логическую схему сумматора в соответствии с уравнением (4) и использовал цепи распространения переноса (листинг 8). Как результат — максимальная тактовая частота схемы выросла примерно в три раза и составила 212 МГц (производительность 212 MIPS).

Листинг 8

```

HDL Synthesis Report
8-bit register       : 3
8-bit adder         : 1
    
```

Final Report

Number of Slice Flip Flops: 24  
 Number of 4 input LUTs: 9  
 TIMING REPORT  
 Minimum period: 4.709ns (Maximum Frequency: 212.359MHz)

При увеличении количества разрядов операндов (а следовательно, и увеличении путей распространения переносов) производительность схемы суммирования уменьшается (рис. 2).

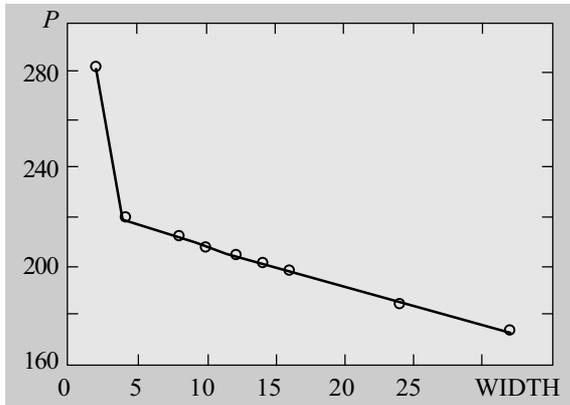


Рис. 2. Зависимость производительности сумматора  $P$  от количества разрядов операндов

Особенностью работы сумматоров в цифровых нерекурсивных фильтрах является большое количество операндов, принимающих участие в формировании выходного сигнала сумматора. Альтернативой каскадно-включенных двухоперандных сумматоров является использование сумматоров с тремя и большим количеством входов. Однако у многовходовых сумматоров резко возрастает логическая емкость, что, в свою очередь, приводит к уменьшению производительности (рис. 3). Сравнивая характеристики двухоперандных и трехоперандных сумматоров, следует отдать предпочтение каскадно-включенным двухоперандным сумматорам.

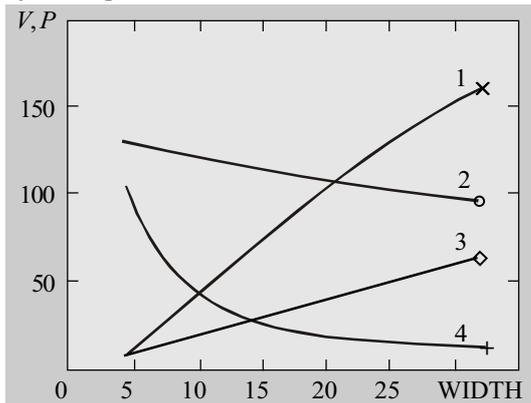


Рис. 3. Сравнительные характеристики многооперандных сумматоров:

1, 4 — соответственно емкость  $V$  и производительность  $P$  трехоперандного сумматора; 2, 3 — соответственно производительность  $P$  и емкость  $V$  каскадно-включенных двухоперандных сумматоров

Операция двоичного умножения в цифровом фильтре состоит в умножении отсчетов входного сигнала на коэффициенты фильтра и считается наиболее сложной в реализации [3, с. 147].

$$Y_n = \left( \sum_{i=0}^n x_i \cdot 2^i \right) \left( \sum_{i=0}^n y_i \cdot 2^i \right), \quad (5)$$

где  $Y_n$  — произведение;  
 $x, y$  — соответственно множимое и множитель.

При имплементации умножителей в ПЛИС наиболее часто используют структуру матричного умножителя, структуру по алгоритму Бута и структуру умножителя на константу [5].

Структурная схема устройства, реализующего матричный алгоритм умножения, приведена на рис. 4 [3, с. 148].

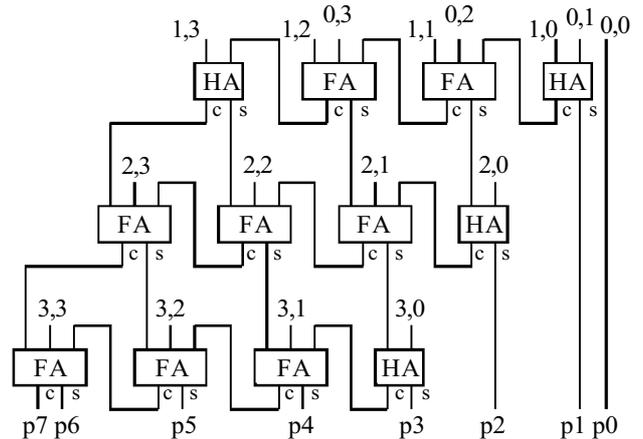


Рис. 4. Структурная схема матричного умножителя

Схема матричного умножителя состоит из каскадно-включенных сумматоров, которые формируют элементарные произведения (листинг 9).

Листинг 9

HDL Synthesis

# Registers : 13

24-bit register : 1

Number of Slice Flip Flops: 46

Number of 4 input LUTs: 158

TIMING REPORT

Minimum period: 16.088ns (Maximum Frequency: 62.158MHz)

Синтезатором сгенерирован массив сумматоров для реализации логических функций в соответствии с формулой (5) и рис. 4. При этом задействовано 158

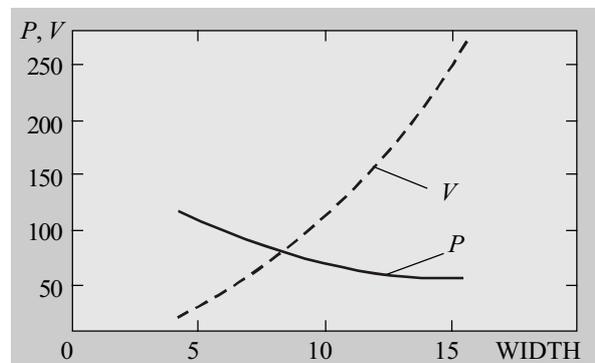


Рис. 5. Зависимость производительности  $P$  и логической емкости  $V$  матричного умножителя от количества разрядов операндов

макроблочек микросхемы. Максимальная тактовая частота схемы составляет 62 МГц (производительность 62 MIPS). Увеличение разрядности операндов приводит к резкому изменению как логической емкости, так и производительности (рис. 5).

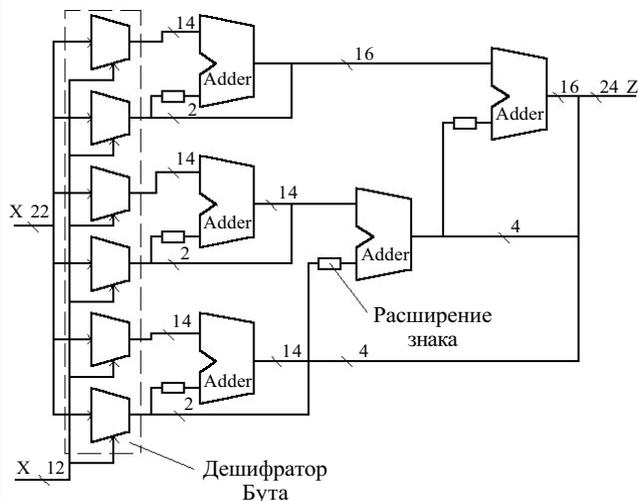


Рис. 6. Умножитель двенадцатиразрядных операндов на основе алгоритма Бута

Для уменьшения логической емкости применим алгоритм Бута, т. к. в основе его лежит идея распределенной арифметики (рис. 6). При этом быстродействие умножителя примерно равно быстродействию матричного умножителя, но величина логической емкости уменьшается вдвое (листинг 10).

Листинг 10

Final Report

Number of Slice Flip Flops: 36  
Number of 4 input LUTs: 68

TIMING REPORT

Minimum period: 13.307ns (Maximum Frequency: 65.148MHz)

В нерекурсивных фильтрах один из операндов является константой, тогда можно значительно сократить величину логической емкости и повысить производительность операции умножения за счет исполь-

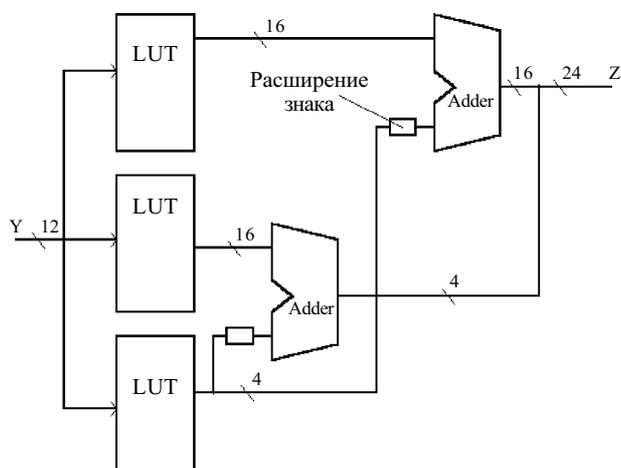


Рис. 7. Структурная схема модифицированного табличного метода умножения двенадцатиразрядных входного операнда Y и константы

зования модифицированного табличного метода умножения [5, с. 121]. При таком методе множимое разбивают на группы разрядов, каждая из которых является адресом соответствующей части таблицы произведений (рис. 7).

Из отчета о синтезе (листинг 11) следует, что при синтезе постоянное запоминающее устройство было заменено массивом мультиплексоров, которые коммутируют биты константного операнда из таблицы частичных произведений на входы сумматоров.

Листинг 11

HDL Synthesis Report

# Registers : 2  
# Multiplexers : 3  
# Adders/Subtractors : 2

Final Report

Number of Slice Flip Flops: 36  
Number of 4 input LUTs: 68

TIMING REPORT

Minimum period: 11.482ns (Maximum Frequency: 87.093MHz)

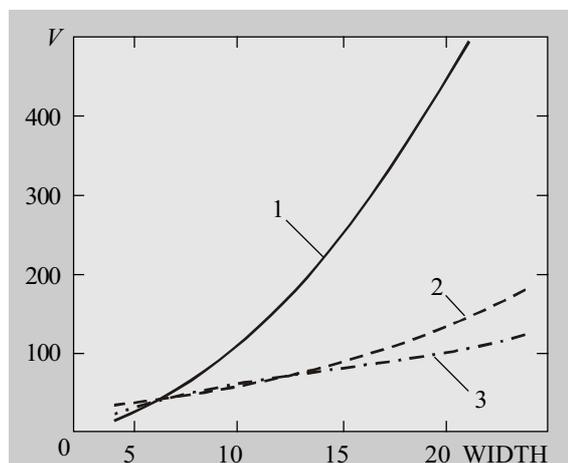


Рис. 8. Зависимость логической емкости  $V$  умножителей разных типов от количества разрядов операндов: 1 — матричный умножитель; 2 — умножитель по алгоритму Бута; 3 — табличный умножитель

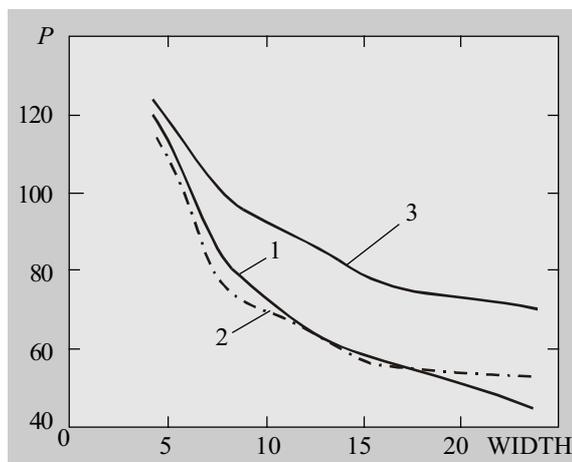


Рис. 9. Зависимость производительности  $P$  умножителей разных типов от количества разрядов операндов: 1 — матричный умножитель; 2 — умножитель по алгоритму Бута; 3 — табличный умножитель

Сравнительный анализ логической емкости и производительности при разной реализации умножителя показывает, что при разрядности больше 12 бит нецелесообразно использовать матричный умножитель, а следует отдать предпочтение табличному умножителю (рис. 8 и 9). В случае перемножения двух переменных следует использовать умножитель по алгоритму Бута.

На основе проведенных исследований можно сделать следующие выводы:

1. Теоретически достижимая производительность цифрового нерекурсивного фильтра при реализации на ПЛИС равна 350 MIPS. Такое значение производительности обусловлено отсутствием каскадируемых генераторов логических функций и может быть достигнуто лишь при наличии твердо закрепленных связей между логическими блоками ПЛИС. Дальнейший рост производительности ограничен технологическими особенностями микросхем.

2. Производительность системы на ПЛИС резко уменьшается при использовании каскадно-включенных генераторов логических функций (LUT) за счет задержек на коммутирующих элементах, в качестве которых используются полевые транзисторы.

3. Анализ вариантов реализации сумматора показал, что наиболее приемлемым является использование библиотечных функций "+", т. к. при этом синтезатором включаются имеющиеся в кристалле цепи распространения последовательного переноса. Например, производительность сумматора, описанного с использованием библиотечных функций, в 3 раза больше, чем для сумматора, описанного логическими уравнениями (соответственно 212 и 72 MIPS).

4. Производительность многооперандного сумматора экспоненциально уменьшается с ростом количества операндов и порядка фильтра. Например, при восьми операндах производительность уменьшается до 100 MIPS.

5. Анализ вариантов реализации умножителей показывает, что использование матричного метода до-

пустимо лишь при небольшой разрядности операндов (4—8 разрядов), при большей разрядности производительность умножителя экспоненциально уменьшается, а логическая емкость, наоборот, возрастает.

6. Использование алгоритма Бута при реализации двоичных умножителей позволяет уменьшить логическую емкость умножителя до 20% при практически тех же значениях производительности.

7. Применение модифицированного табличного метода умножения позволяет синтезировать умножители со значением производительности до 100 MIPS (90 MIPS при двенадцатиразрядных операндах). Логическая емкость умножителя при этом значительно уменьшается (48—60 макроячеек и 119 макроячеек при матричном методе).

\*\*\*

Таким образом, использование ПЛИС для реализации цифровых фильтров позволяет синтезировать быстродействующие фильтры, производительность которых определяется как технологическими особенностями производства микросхем, так и методом описания выбранной структуры фильтра. Возможность реализации алгоритмов параллельной обработки входных данных и высокая степень формализации проектирования позволяют говорить о ПЛИС как о наиболее пригодной базе для реализации высокопроизводительных фильтров.

#### ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

1. Солонина А. И., Улахович Д. А., Яковлев Л. А. Алгоритмы и процессы цифровой обработки сигналов.— СПб.: БХВ-Петербург, 2002.
2. Meyer-baese U. Digital signal processing with field programmable gate arrays.— Heidelberg: Springer Press, 2001.
3. Пухальский Г. И. Проектирование дискретных устройств на интегральных микросхемах.— М.: Радио и связь, 1990.
4. Шапо Ф. С., Шапо В. Ф. Введение в VHDL — язык проектирования цифровых систем.— Одесса: Астропринт, 2001.
5. Стешенко В. Б. ПЛИС фирмы ALTERA: проектирование устройств обработки сигналов.— М.: Додэка, 2000.

#### НОВЫЕ КНИГИ

### Данилов А. А. ПРЕЦИЗИОННЫЕ УСИЛИТЕЛИ НИЗКОЙ ЧАСТОТЫ.— М.: Горячая линия-Телеком, 2004.— 352 с., ил.

Рассмотрены основы звукоусиления и параметры, характеризующие усилители мощности звуковой частоты (УМЗЧ). Сформулирована задача построения звуковых усилителей, свободных от всех видов искажений. Особое внимание уделено грамотному применению отрицательной обратной связи. Для выяснения природы возникновения разного рода нелинейностей проанализированы структурные схемы и отдельные узлы УМЗЧ. Описана схема УМЗЧ с параллельным высокочастотным каналом, позволяющим решить проблему прецизионного усиления. Рассмотрены схемы защиты источников питания и монтажа мощных усилителей. Проанализированы многочисленные примеры схем бытовых и профессиональных УЗМЧ на дискретных элементах и интегральных микросхемах. Даны рекомендации по разводке и изготовлению печатных плат, измерению параметров и настройке усилителей.



Для инженерно-технических работников, занимающихся схемотехническим проектированием устройств радиоэлектроники и автоматики, студентов и радиолюбителей.

