

УДК 621.382

В. П. Маркова

(Новосибирск)

ЭЛЕКТРООПТИЧЕСКАЯ РЕАЛИЗАЦИЯ
КЛЕТЧНОГО УМНОЖИТЕЛЯ*

На базе методов параллельного микропрограммирования предложен способ преобразования двумерной структуры устройства в трехмерную. Преобразование (расслоение) исходного устройства основано на его декомпозиции на множество подустройств. Каждое подустройство реализуется на отдельном слое, а связи между ними переносятся в третье измерение. На примере оригинального алгоритма умножения целых n -разрядных чисел с фиксированной запятой построен ряд умножителей с различным количеством слоев, определены оценки сложности их реализации на электрооптических ПЛМ.

Введение. Построение оптических устройств в виде многослойных (трехмерных) структур позволит преодолеть ограничения «плоских» СВИС [1, 2] и сделать их конкурентноспособными с полупроводниковыми аналогами. В электрооптической схемотехнике [3, 4] каждый слой такой структуры — микроэлектронная СВИС, передача информации между слоями, а также ввод и вывод — осуществляются оптическим путем.

В данной работе описаны две версии предложенного (базового) алгоритма умножения двух целых n -разрядных чисел с фиксированной запятой. На их основе построены структурные схемы умножителей на электрооптических ПЛМ [5] с разным количеством слоев $k > 2p$, где p — число логических слоев, и определены оценки сложности их реализации. Здесь под логическим слоем умножителя будем понимать два физических слоя, в которых размещается электрооптическая ПЛМ. Ниже умножитель, состоящий из одного логического слоя, будем называть однослойным, все остальные — многослойными.

Расслоение исходного умножителя основано на его декомпозиции на множество подустройств. Каждое подустройство реализуется на одном логическом слое, а связи между ними переносятся в третье измерение. Данный способ расслоения так же, как и [6, 7], основан на методах параллельного микропрограммирования [8, 9] и является некоторой модификацией способа декомпозиции, предложенного в [6].

Цель расслоения — синтез ряда умножителей для реализации одного и того же алгоритма, которые отличаются количеством слоев, их топологией, а иногда и временем исполнения алгоритма.

Статья построена следующим образом. Основные понятия параллельного микропрограммирования перечислены в п. 1. В п. 2 описан базовый алгоритм умножения целых n -разрядных чисел. В п. 3 представлены однослойные электрооптические реализации предложенных и известных [10] умножителей и их оценки сложности. Варианты расслоения данных умножителей и оценки сложности их реализации на электрооптических ПЛМ даны в п. 4.

1. Параллельная микропрограмма и ее интерпретация сетями автоматов. Параллельная микропрограмма (множество параллельных микрокоманд) за-

* Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (93-012-1000).

дает алгоритм преобразования клеточного массива W , записанного в виде множества пар (клеток) (a, m) , $a \in A$, $m \in M$, где A — конечный алфавит состояний, M — конечное множество имен. В качестве микрокоманды используется операция подстановки. Для отображения повсеместности подстановок к множеству W на множестве имен определены именуемые функции $\Psi_i(m)$, $i = 1, 2, \dots, p$. Выражение $S(m) = \{(a, \Psi_i(m))\}$, в котором все именуемые функции различны, задает конфигурацию, т. е. пространственный образ подмножества клеток. Конфигурация для каждой клетки с именем m , $m \in M$, определяет подмножество соседних клеток, т. е. клеток, участвующих в операции подстановки. В нашем случае клеточный массив — k -мерная целочисленная решетка, в узлы которой вписаны символы из алфавита A , именуемые функции — сдвиги вдоль ее осей, параллельная микрокоманда Θ_i — подстановка вида

$$\Theta_i : S_{i_1}(m) * S_{i_2}(m) \rightarrow S_{i_3}(m) \text{ для всех } m \in M,$$

где $S_{i_1}(m) * S_{i_2}(m)$ и $S_{i_3}(m)$ — левая (база и контекст) и правая части подстановки. База и правая часть подстановки имеют одно и то же множество именуемых функций. Микрокоманда Θ_i применима к множеству W , если в нем найдется хотя бы одна клетка с именем m , $m \in M$, такая что $S_{i_1}(m) * S_{i_2}(m) \subseteq W$. Применение Θ_i к W означает замену состояний базовых клеток на состояния клеток правой части.

В записи конфигураций первый элемент пары может задаваться не только конкретным символом из алфавита A , но и переменной, принимающей значение из A , или некоторой функцией, для которой алфавит A является областью определения и значений. Микрокоманда, содержащая в записи конфигураций переменные и функциональные символы, называется функциональной. Применяется она так же, как и обычная, но с тем отличием, что процедуре замены состояний клеток предшествует вычисление значений стоящих в базе и правой части переменных и функциональных символов.

Множество параллельных микрокоманд $\Phi = \{\Theta_i\}$, $i = 1, 2, \dots, p$, задает параллельную микропрограмму (микропрограммное описание) алгоритма работы проектируемого устройства. В [3] показано, что архитектура устройства полностью определяется микропрограммным описанием (МО) и на абстрактном уровне задается сетью автоматов. Построение сети и функционирование ее автоматов непосредственно вытекают из правил применения микрокоманд и состоят в следующем. Множеству имен клеток M ставится в соответствие множество автоматов, алфавиту A — множество их внутренних состояний. Для автомата с именем m , $m \in M$, определяется подмножество микроопераций, содержащих имя соответствующей ей клетки в записи базовых частей. Это подмножество определяет содержимое микропрограммной памяти данного автомата. Все другие имена клеток, использованные в записи левых (правых) частей микроопераций, образуют подмножество имен автоматов, соседних с автоматом m по входам (выходам). Проведение установленных таким образом связей для подмножества автоматов, определенного по одному из критериев [11], в результате даст структурную схему устройства. В сети будем различать два типа ячеек: функциональные и ячейки памяти. Под функциональными ячейками будем понимать автоматы, реализующие хотя бы одну функциональную микрокоманду, под ячейками памяти — автоматы, которые хранят информацию и осуществляют ее сдвиг.

2. Базовый алгоритм умножения целых чисел. Архитектура известных матричных устройств умножения с фиксированной запятой соответствует классическому алгоритму умножения. Сначала вычисляется множество частичных произведений (ЧП), затем оно сворачивается в произведение исходных чисел. Увеличение скорости умножения может быть достигнуто за счет уменьшения количества ЧП и использования ускоренных схем сложения. Ускоренное сложение множества ЧП, как правило, выполняется в два этапа. На первом этапе это множество суммируется до двурядного кода, т. е. двух векторов: суммы и переноса. На втором этапе двурядный код преобразуется в

однорядный, т. е. вычисляется сумма этих векторов, которая и является результатом умножения исходных чисел.

Предложенный алгоритм умножения двух n -разрядных двоичных чисел (ниже просто алгоритм умножения целых чисел) отличается от обычного следующим. Во-первых, начиная с v -го шага алгоритма, $v < n$, генерация частичных произведений совмещена с их сложением с помощью параллельных (5,3)- и (или) (3,2)-счетчиков [12, 13]. И во-вторых, однорядный код произведения формируется в модифицированной знакоразрядной числовой системе (ЧС) [14, 15].

Параллельный (5,3)-счетчик представляет собой комбинационную схему с пятью входами и тремя выходами. Эта схема вычисляет сумму единиц j -го разряда, $j = 0, 1, \dots, n-1$, двоичных чисел X, Y, Z, V и W (рис. 1) в виде трехрядного кода: одного разряда суммы s_j и двух разрядов переноса c_{j+1}^1 и c_{j+2}^2 :

$$s_j = u_j \bmod 2, \quad (1a)$$

$$c_{j+1}^1 = \begin{cases} 0, & \text{если } u_j < 2, \\ 1 & \text{в противном случае,} \end{cases} \quad (1б)$$

$$c_{j+2}^2 = \begin{cases} 0, & \text{если } u_j < 4, \\ 1 & \text{в противном случае,} \end{cases} \quad (1в)$$

где $u_j = x_j + y_j + z_j + v_j + w_j$. Вычисление по соотношениям (1a)–(1в) называется преобразованием $5 \rightarrow 3$. Диагональное распространение переноса в преобразовании $5 \rightarrow 3$ позволяет редуцировать пять двоичных чисел в три за один такт.

Параллельный (3,2)-счетчик есть не что иное, как полный одноразрядный сумматор, функционирование которого, т. е. преобразование $3 \rightarrow 2$, чаще всего описывается логическими уравнениями вида

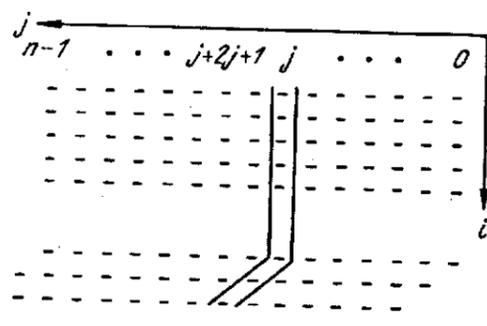
$$s_j = x_j \oplus y_j \oplus z_j, \quad (2a)$$

$$c_{j+1} = x_j y_j \vee x_j \bar{y}_j \bar{z}_j \vee \bar{x}_j y_j z_j \vee x_j y_j z_j, \quad (2б)$$

где $j = 1, 2, \dots, n$.

Модифицированная знакоразрядная ЧС относится к классу избыточных. В качестве основания она использует число два, для изображения чисел — множество $\{\bar{1}, 0, 1\}$. Введение избыточности в числовое представление делает модифицированную знакоразрядную ЧС свободной от основного недостатка двоичной ЧС — организации переноса при сложении двух чисел.

Существует несколько схем сложения в модифицированной знакоразрядной числовой системе. В базовом алгоритме умножения целых чисел используется трехэтапная схема. Пусть X и Y — два n -разрядных двоичных числа



(двоичная запись исходного числа является одним из возможных его представлений в модифицированной знакоразрядной ЧС). Тогда схема поразрядного сложения выглядит следующим образом:

$$x_j + y_j = c_{j+1}^1 + s_j^1,$$

$$s_j^1 + c_j^1 = c_{j+1}^2 + s_j^2,$$

$$s_j^2 + c_j^2 = s_j,$$

Рис. 1

Цифры суммы		Цифры переноса	
$x \backslash y$	$\bar{1} \ 0 \ 1$	$\bar{1} \ 0 \ 1$	$\bar{1} \ 0 \ 1$
$\bar{1}$	0 1 0	$\bar{1}$ $\bar{1}$ $\bar{1}$ 0	
0	1 0 $\bar{1}$	0 $\bar{1}$ 0 1	
1	0 $\bar{1}$ 0	1 0 1 1	

Цифры суммы		Цифры переноса	
$x \backslash y$	$\bar{1} \ 0 \ 1$	$\bar{1} \ 0 \ 1$	$\bar{1} \ 0 \ 1$
$\bar{1}$	0 $\bar{1}$ 0	$\bar{1}$ $\bar{1}$ 0 0	
0	$\bar{1}$ 0 1	0 0 0 0	
1	0 1 0	1 0 0 1	

где $j = 0, 1, \dots, n - 1$. Цифры промежуточных суммы и переноса на первом этапе сложения вычисляются по табл. 1, на втором — по табл. 2, цифры истинной суммы определяются арифметическим сложением. Из приведенной схемы сложения видно, что при суммировании переносы распространяются только на два разряда влево. Пример сложения приведен на рис. 2.

Пусть $X = (0, 0, \dots, 0, x_{n-1}, \dots, x_0)$ и $Y = (0, 0, \dots, 0, y_{n-1}, \dots, y_0)$ — множимое и множитель (значения n старших разрядов в векторах X и Y равны нулю). Будем различать две версии базового алгоритма вычисления их произведения $Z = (z_{2n-1}, z_{2n-2}, \dots, z_0)$. Первая версия (алгоритм А) для редукции частичных произведений использует (5,3)- и (3,2)-счетчики, вторая версия (алгоритм В) — только (3,2)-счетчики. Оба алгоритма включают три этапа. Рассмотрим алгоритм А. На первом этапе алгоритма вычисляются пять частичных произведений, на втором — генерация ЧП выполняется параллельно с их редукцией до получения двурядного кода произведения, на третьем — формируется окончательный результат.

Опишем второй этап алгоритма А. Он начинается с шестого шага и представляет собой следующую итерационную процедуру. На шаге $\nu = 2m$, $m = 3, 4, \dots, \lceil n^{-1}/2 \rceil$, выполняется преобразование 5 → 3 и вычисляется частичное произведение. На шаге $\nu = 2m + 1$ вычисляется только ЧП. Процедура завершается после формирования последнего частичного произведения, т. е. на шаге $\nu = n$. Результатом ее выполнения для четных n является множество чисел $S, C^1, C^2, ЧП_{n-1}$ и $ЧП_n$, для нечетных n — множество чисел S, C^1, C^2 и $ЧП_n$. На следующих двух шагах полученное множество редуцируется в трехрядный, а затем в двурядный код произведения.

Таким образом, алгоритм А вычисляет произведение двух двоичных n -разрядных чисел за $n + 5$ тактов, алгоритм В — за $n + 4$ такта. Высокая скорость умножения достигается за счет совмещения генерации частичных произведений с их редукцией и «почти» параллельного сложения, поддержанного модифицированной знакоразрядной ЧС.

3. Архитектура однослойного клеточного умножителя. Проектирование архитектуры и анализ клеточных умножителей выполнены методами параллельного микропрограммирования. Техника проектирования продемонстрирована на примере клеточного умножителя, выполняющего алгоритм А (умножителя А). Ниже МО, интерпретируемое двумерным клеточным умножителем А, будем называть двумерным и обозначать $\Phi(A)$, соответственно МО, интерпретируемое трехмерным клеточным умножителем \hat{A}_3 — трехмерным и обозначать $\Phi(A)$.

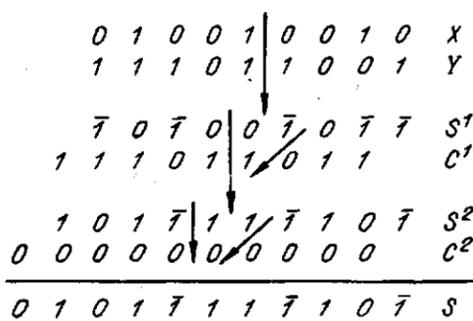


Рис. 2

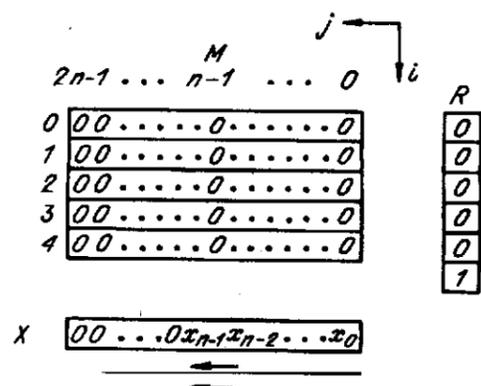


Рис. 3

3.1. Микропрограммное описание алгоритма умножения целых чисел А. Микропрограммное описание Ф алгоритма умножения А строится следующим образом. Клеточный массив с исходными данными представляет собой объединение пяти подмассивов (блоков) M, X, Y, R и P (исходные состояния подмассивов изображены на рис. 3). Массив M — поле для формирования частичных произведений и их суммирования до полу-ма. Имена клеток в перечисленных массивах принадлежат множествам имен $\{ \langle M, i, j \rangle \}$, $\{ \langle X, 0, j \rangle \}$, $\{ \langle Y, 0, j \rangle \}$, $\{ \langle R, i', 0 \rangle \}$ и $\{ \langle P, 0, j' \rangle \}$ (первая компонента в имени указывает на имя блока),

$i = 0, 1, \dots, 4, j = 0, 1, \dots, 2n - 1, i' = 1, 2, \dots, 5, j' = 0, 1, \dots, n + 5$. Каждая клетка массива может быть переменным символом x, y, z, v, w с областью определения из алфавита состояний $A_1 = \{-1, 0, 1\}$, функциональным символом f_1, f_2, \dots, f_{11} с областью значений из алфавита $A = \{-1, 0, 1, 2, 3, 4, 5\}$ и просто принимать значения из двоичного алфавита.

Микропрограмма Ф(А) состоит из тринадцати микрокоманд. Микрокоманда Θ_1 формирует частичное произведение в четвертой строке массива M:

$$\Theta_1 : \{ (\# , \langle M, 4, j \rangle) \} * \{ (x, \langle X, 0, j \rangle) (y, \langle Y, 0, 0 \rangle) (0, \langle P, 0, n + 1 \rangle) \} \rightarrow \{ (f_1, \langle M, 4, j \rangle) \},$$

где # — безразличный символ,

$$f_1 = \begin{cases} x, & \text{если } y = 1, \\ 0 & \text{в противном случае.} \end{cases}$$

Микрокоманды Θ_2 и Θ_3 осуществляют сдвиги влево и вправо множителя в массивах X и Y соответственно:

$$\Theta_2 : \{ (\# , \langle X, 0, j + 1 \rangle) \} * \{ (x, \langle X, 0, j \rangle) (0, \langle P, 0, n + 1 \rangle) \} \rightarrow \{ (x, \langle X, 0, j + 1 \rangle) \};$$

$$\Theta_3 : \{ (\# , \langle Y, 0, j - 1 \rangle) \} * \{ (y, \langle Y, 0, j \rangle) (0, \langle P, 0, n + 1 \rangle) \} \rightarrow \{ (y, \langle Y, 0, j - 1 \rangle) \}.$$

Микрокоманда Θ_4 сдвигает в сторону меньших i ЧП в массиве M. Она выполняется в течение первых пяти тактов работы алгоритма:

$$\Theta_4 : \{ (\# , \langle M, i - 1, j \rangle) \} * \{ (v, \langle M, i, j \rangle) (0, \langle P, 0, 5 \rangle) \} \rightarrow \{ (v, \langle M, i - 1, j \rangle) \}.$$

Микрокоманда Θ_5 реализует движение признака «1» по управляющему регистру R:

$$\Theta_5 : \{ (0, \langle R, i', 0 \rangle) \} * \{ (1, \langle R, i' - 1, 0 \rangle) (0, \langle P, 0, n + 1 \rangle) \} \rightarrow \{ (1, \langle R, i', 0 \rangle) \}.$$

Микрокоманда Θ_6 вычисляет преобразование $5 \rightarrow 3$ в массиве M , признаком ее выполнения служит клетка $(1, \langle R, 0, 0 \rangle)$:

$$\Theta_6 : \{(\#, \langle M, 0, j+2 \rangle)(\#, \langle M, 1, j+1 \rangle)(z, \langle M, 2, j \rangle)\} * \{(x, \langle M, 0, j \rangle) \\ (y, \langle M, 1, j \rangle)(v, \langle M, 3, j \rangle)(w, \langle M, 4, j \rangle)(0, \langle P, 0, n+1 \rangle)(1, \langle R, 0, 0 \rangle)\} \rightarrow \\ \Theta_7 : \{(1, \langle R, 0, 0 \rangle)\} * \{(1, \langle R, 1, 0 \rangle)(0, \langle P, 0, j'' \rangle)\} \rightarrow \{(0, \langle R, 0, 0 \rangle)\},$$

если $j'' < n-1, n-2m$ или $j'' < n, n=2m+1$. После выполнения преобразования $5 \rightarrow 3$ микрокоманда Θ_8 сдвигает ЧП из четвертой строки массива M в третью:

$$\Theta_8 : \{(\#, \langle M, 3, j \rangle)\} * \{(v, \langle M, 4, j \rangle)(1, \langle P, 0, j'' \rangle)(0, \langle R, 0, 0 \rangle)\} \rightarrow \{(v, \langle M, 3, j \rangle)\},$$

где $j'' \geq 5$. Микрокоманда Θ_9 реализует преобразование $3 \rightarrow 2$ в массиве M . Признаком начала выполнения микрокоманды служат клетки $(0, \langle P, 0, n+2 \rangle)$ и $(1, \langle P, 0, n+1 \rangle)$:

$$\Theta_9 : \{(\#, \langle M, 1, j+1 \rangle)(z, \langle M, 2, j \rangle)\} * \{(x, \langle M, 0, j \rangle)(y, \langle M, 1, j \rangle) \\ (0, \langle P, 0, n+2 \rangle)(1, \langle P, 0, n+1 \rangle)\} \rightarrow \{(f_5, \langle M, 1, j+1 \rangle)(f_6, \langle M, 2, j \rangle)\},$$

где функции f_5 и f_6 вычисляются по формулам (2а) и (2б). Микрокоманды Θ_{10} — Θ_{12} формируют сумму двух чисел, полученных в результате выполнения преобразования $3 \rightarrow 2$, в окончательное произведение:

$$\Theta_{10} : \{(\#, \langle M, 1, j+1 \rangle)(y, \langle M, 2, j \rangle)\} * \{(x, \langle M, 1, j \rangle)(0, \langle P, 0, n+3 \rangle) \\ (1, \langle P, 0, n+2 \rangle)\} \rightarrow \{(f_7, \langle M, 1, j+1 \rangle)(f_8, \langle M, 2, j \rangle)\}, \\ \Theta_{11} : \{(\#, \langle M, 1, j+1 \rangle)(y, \langle M, 2, j \rangle)\} * \{(x, \langle M, 1, j \rangle)(0, \langle P, 0, n+4 \rangle) \\ (1, \langle P, 0, n+3 \rangle)\} \rightarrow \{(f_9, \langle M, 1, j+1 \rangle)(f_{10}, \langle M, 2, j \rangle)\}, \\ \Theta_{12} : \{(x, \langle M, 1, j \rangle)\} * \{(y, \langle M, 2, j \rangle)(0, \langle P, 0, n+5 \rangle)(0, \langle P, 0, n+4 \rangle)\} \rightarrow \\ \{(f_{11}, \langle M, 1, j \rangle)\},$$

где f_7 и f_8, f_9 и f_{10} — функции переноса и суммы двух этапов сложения с модифицированной знакоразрядной ЧС, описанные в табл. 1 и 2, $f_{11} = x + y$. Микрокоманда Θ_{13} отсчитывает такты работы алгоритма А. Клетка $(1, \langle P, 0, n+5 \rangle)$ служит признаком завершения его работы:

$$\Theta_{13} : \{(0, \langle P, 0, j'' \rangle)\} * \{(1, \langle P, 0, j''-1 \rangle)\} \rightarrow \{(1, \langle P, 0, j'' \rangle)\}.$$

В приведенном тексте микропрограммы переменные и функциональные символы фиксируются в пределах одной микрокоманды.

Для оптической интерпретации параллельной микропрограммы $\Phi(A)$ алфавит состояний клеток в ее описании необходимо привести к двоичному. Это достигается следующими преобразованиями.

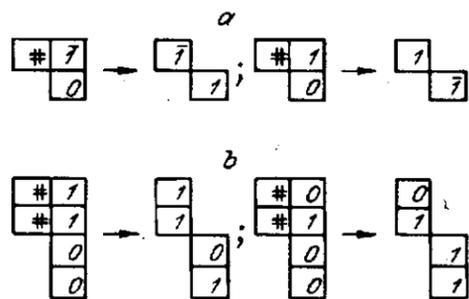


Рис. 4

1. Преобразование $5 \rightarrow 3$ записывается в виде логических уравнений аналогично преобразованию $3 \rightarrow 2$.

2. При формировании однорядного кода произведения в модифицированной знакоразрядной числовой системе разряды слагаемых представляются парой (знак, модуль), т. е. $x_i \rightarrow (\text{sign}x_i, |x_i|)$. Это влечет за собой преобразование алфавитов на всех этапах сложения в данной ЧС. На рис. 4, а, б продемонстрирован первый этап сложения двух пар разрядов непосредственно по табл. 1 и в преобразованном виде. Количество микрокоманд при этом не увеличивается.

3.2. Микропрограммное описание алгоритма умножения целых чисел В. Микропрограммное описание алгоритма умножения целых чисел В строится аналогично описанию $\Phi(A)$. Основное отличие $\Phi(B)$ от $\Phi(A)$ состоит в отсутствии микрокоманд Θ_6 и Θ_8 .

3.3. Электрооптическая реализация клеточного умножителя А. Электрооптический умножитель, интерпретирующий микропрограммное описание $\Phi(A)$, представляет собой двуслойную БИС (два физических слоя). Структурная схема умножителя А строится по процедуре, описанной в [7]. Сложность реализации умножителя будем характеризовать тремя оценками: временной, структурной и аппаратурной. Под временной оценкой сложности T будем понимать количество тактов, необходимое для вычисления произведения двух n -разрядных чисел. Структурную сложность S будем определять как сумму входов сравнения и выходов записи у функциональных ячеек (ФЯ) устройства на разряд преобразуемой информации. Аппаратурную сложность вслед за [5] будем оценивать через объем, который занимает электрооптическое устройство, т. е. $V = kA$. (В данном случае это верхняя оценка сложности.) Здесь A — площадь слоя, k — количество слоев ($k = 2$). Площадь слоя определяется следующей суммой:

$$A' = \sum_i A_{\text{яч}}^i |M_i|, \quad i = 1, \dots, l,$$

где l — количество различных типов ячеек, $A_{\text{яч}}^i$ — площадь ячейки i -го типа, $|M_i|$ — их количество. Площадь ячейки, как показано в [5], равна

$$A_{\text{яч}} = [32\gamma(u + v) + (u + v)^2 + 100] \alpha \beta,$$

где $(u + v)$ — количество входов сравнения и выходов записи ячейки, γ — количество реализуемых микроопераций, α — минимальный линейный размер окна модулятора света, β — технологический параметр ПЛМ.

Очевидно, что структурная и аппаратурная сложности учитывают топологию устройства в интегральном виде. Это затрудняет проектировщику «видеть» такие ее особенности, как степень однородности, сложность ФЯ, соотношение между локальными и глобальными связями и т. д. Поэтому для большей наглядности структурной схемы ниже из оценок S и V вынесем такие характеристики, как количество ФЯ и параметры самой сложной функциональной ячейки: площадь и количество связей.

На рис. 5 приведен фрагмент умножителя А. Полученная структура квазиоднородна. В ней четыре регистра сдвига P, X, Y, R и двумерный массив M . Каждый столбец массива M включает два типа функциональных ячеек. Первый тип ФЯ (ячейка с именем $\langle M, 2, j \rangle$) сдвигает ЧП, реализует преобразования $5 \rightarrow 3, 3 \rightarrow 2$ и $2 \rightarrow 1$. Ячейка имеет 12 входов сравнения и 5 выходов

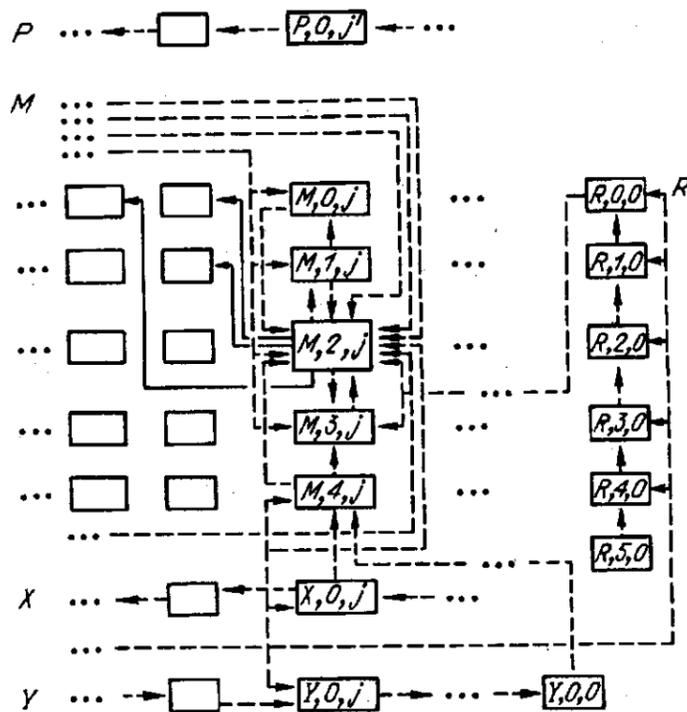


Рис. 5

записи (на схеме входы сравнения и выходы записи обозначены штриховой и сплошной линиями, вход сравнения и выход записи в ячейку не изображены). При параметрах ПЛМ $\alpha = 10$ мкм и $\beta = 1,6$ ячейка $(M, 2, j)$ занимает площадь, равную $0,317$ мм². Второй тип ФЯ (ячейка с именем $(M, 4, j)$) вычисляет только j -й разряд ЧП, у ячейки 4 соседа по входам и 1 — по выводу. Таким образом, оценки структурной и аппаратной сложностей реализации умножителя А составляют величины $S = 22$ и $V = (0,323n + 0,022)$ мм³ при расстоянии между слоями ПЛМ 0,5 мм.

В табл. 3 приведены оценки электрооптической реализации пяти плоских умножителей. Первые два умножителя (А и В) реализуют две версии предло-

Таблица 3

Оценки сложности реализации умножителя	Умножители				
	А	В	С	Д	Е
T	$n + 5$	$n + 4$			
S	22	20	$8n - 2$	$10 \frac{n}{3} + 14$	$9n + 5$
V	$0,323n$	$0,174n$	$0,036n^2$	$0,027n^2$	$0,016n^2$
Количество типов ФЯ	2	2	3	3	2
$\frac{A_{\text{яч}}}{S}$	0,317 17	0,166 15	0,035 8	0,041 10	0,016 9

женного алгоритма. Следующие три (они обозначены С, D и E) выбраны для сравнения. Умножители С, D и E построены по схеме классического алгоритма умножения и отличаются только способом суммирования частичных произведений. В умножителях С и D [10] сворачивание множества ЧП (оно во всех умножителях формируется неускоренным способом) выполняется в два этапа. Оба умножителя на первом этапе используют сумматоры с сохранением переносов, т. е. параллельные (3,2)-счетчики, на втором — сумматоры со сквозным переносом. Только умножитель С формирует двурядный код последовательно за $n - 2$ такта, а умножитель D — по дереву Уоллеса [13] за $O(\log n)$ тактов. Вычисление однорядного кода произведения оба умножителя выполняют за $O(\log n)$. В умножителе E этап формирования двурядного кода произведения отсутствует. Сложение множества ЧП выполняет сумматор многих чисел за $O(n)$ тактов [16]. Временную сложность умножителей С, D и E (в табл. 3 она отсутствует) будем характеризовать средними оценками. Они получены с помощью системы алгоритмического моделирования ALT [17], выборка составила сто пар чисел для каждого значения n . Так, например, на вычисление произведения двух 18-разрядных чисел умножитель А тратит 23 такта, умножитель В — 22 такта, умножитель С — 35 тактов, умножитель D — 45 тактов, умножитель E — 42 такта. (Очевидно, что оценки временной сложности умножителей С и D уменьшаются, если однорядный код произведения формировать в модифицированной знакоразрядной ЧС. Моделирование показало, что для $n = 18$ эта величина составляет 4 такта.)

Из табл. 3 видно, что умножители А и В по однородности не уступают умножителю E, но у них самая сложная функциональная ячейка. Ее сложность может быть уменьшена, если для построения сети использовать процедуру [11]. Однако при этом возрастают неоднородность умножителей, их структурная и аппаратная сложности. Предложенные умножители имеют меньшую временную сложность для любой разумной разрядности чисел. И кроме того, объемы, занимаемые умножителями А и В, начиная с $n \geq 20$ и $n \geq 10$, соответственно не превосходят объемы, занимаемые умножителями С, D и E (в табл. 3 в оценках аппаратной сложности умножителей члены малого порядка опущены). Очевидно, что выбор варианта умножителя остается за проектировщиком и будет определяться техническими требованиями, предъявляемыми к устройству.

4. Архитектура многослойного клеточного умножителя. Под расслоением двумерного клеточного умножителя будем понимать его преобразование в трехмерное, представляющее собой объединение нескольких двумерных клеточных структур (в принятой схемотехнике логических слоев). Проектирование многослойного умножителя основано на преобразовании (расслоении) двумерного МО в эквивалентное трехмерное.

4.1. Расслоение микропрограммного описания алгоритма А. Пусть $\Phi(A) = \{\Theta_i\}$, $i = 1, 2, \dots, p$, — двумерное микропрограммное описание. Рассматриваемый здесь способ расслоения двумерного устройства основан на декомпозиции множества микрокоманд $\Phi(A)$ в множество подмножеств $\hat{\Phi}(A) = \{\hat{\Theta}^k\}$, $k = 0, 1, \dots, t$, описывающих работу выделенных подустройств. Каждое k -е подмножество микрокоманд приписывается к k -му логическому слою трехмерной структуры. В результате для некоторых микрокоманд контекст и правая часть оказываются связанными с разными слоями проектируемого устройства. Такое расслоение имеет ограничения только на количество слоев. Каждая ячейка может иметь любое количество электрических связей внутри слоя и оптических — между слоями. Если топология какого-либо слоя не удовлетворяет техническим ограничениям проектируемого устройства, то она может быть изменена с помощью процедуры [11], иначе двумерная структура разбивается на более простые подустройства.

4.1.1. Расслоение микропрограммного описания алгоритма А. В общем случае возможно несколько вариантов разбиения устройства на подустройства, а следовательно, и расслоения МО. Ниже рассмотрим два варианта расслоения $\Phi(A)$: $\hat{\Phi}_1(A)$ и $\hat{\Phi}_2(A)$, которые описывают работу умножителей А1 и А2.

Первый вариант разбиения клеточного умножителя A . Пусть умножитель A состоит из четырех подустройств:

- 1) подустройство для хранения исходных данных;
- 2) подустройство, формирующее данные для вычисления двурядного кода произведения (это ЧП и результаты преобразований $5 \rightarrow 3$);
- 3) подустройство, выполняющее преобразования $5 \rightarrow 3$ и $3 \rightarrow 2$;
- 4) сумматор для вычисления однорядного кода произведения.

Тогда исходное множество $\Phi(A)$ может быть декомпозировано на четыре подмножества, т. е. $\hat{\Phi}_1(A) = \bigcup_k \hat{\Theta}^k$, $k = 0, 1, 2, 3$, и соответственно умножитель

A может быть представлен в виде четырехслойной структуры $\hat{A}1$ (рис. 6).

Клеточный массив с данными для умножителя $A1$ представляет собой координаты трехмерной решетки $M = \{(M, i, j, k)\}$, $i = 0, 1, \dots, 5$, $j = 0, 1, \dots, 2n$, $k = 0, 1, 2, 3$. Множество имен $\{(M, 4, j, 0)\}$, $\{(M, 3, j, 0)\}$, $\{(M, 5, j, k)\}$ и $\{(M, i, 0, k')\}$, $k' = 1, 2$, соответствуют регистрам X, Y, P и R умножителя A . Регистры P и R дублируются: регистр P расположен в каждом слое, регистр R — в первом и втором. Здесь двоичный алфавит состояний, он в отличие от умножителя A является областью определения значений переменных и функциональных символов.

Многослойный умножитель работает следующим образом. На первом такте на нулевом слое сдвигаются значения множимого и множителя (микрокоманды Θ_2 и Θ_3), на первом слое в четвертой строке формируется частичное произведение (микрокоманда Θ_1) и в нулевом столбце сдвигается признак «1» (микрокоманда Θ_5). Микрокоманда Θ_5 выполняется на втором слое, и на всех слоях начинает работать счетчик тактов (микрокоманда Θ_{13}). На втором такте умножитель $A1$, кроме перечисленных микрокоманд, сдвигает ЧП в первом слое (микрокоманда Θ_4). Сдвиг продолжается до появления клетки $(1, (M, 0, 0, 1))$ — признака начала выполнения преобразования $5 \rightarrow 3$. После формирования данного признака умножитель $A1$ вычисляет очередное ЧП на первом слое и преобразование $5 \rightarrow 3$ на втором (микрокоманда Θ_6), при этом клетки с именами $(M, 0, 0, k')$ меняют свои состояния на противоположные

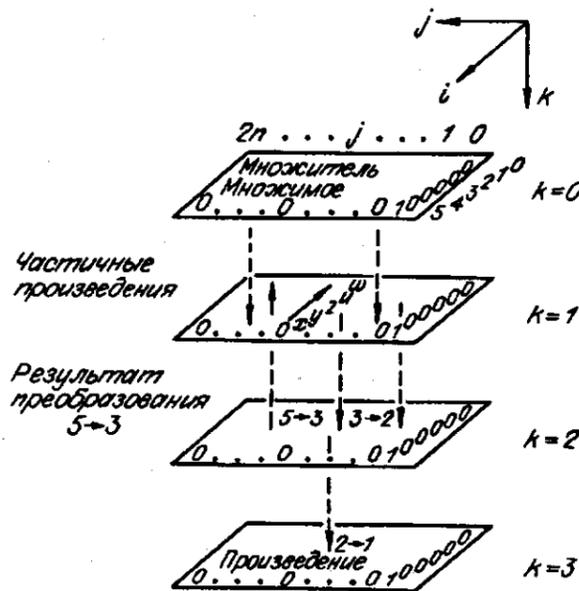


Рис. 6

(микрокоманда Θ_7). Далее умножитель $\hat{A}1$ на первом слое формирует седьмое ЧП, сдвигает шестое ЧП из четвертой строки в третью (микрокоманда Θ_8) и возвращает результат преобразования $5 \rightarrow 3$ на первый слой (дополнительная Θ_{14} микрокоманда). Таким образом, седьмой такт работы умножителя $A1$ завершается формированием очередной пятерки векторов на первом слое. Описанная процедура повторяется до тех пор, пока результат последнего преобразования $5 \rightarrow 3$ не будет передан на первый слой умножителя. Это произойдет на $(n+2)$ -м такте. После этого во втором слое реализуется преобразование $3 \rightarrow 2$ (микрокоманда Θ_9), а затем на третьем — преобразование $2 \rightarrow 1$ (микрокоманды $\Theta_{10}-\Theta_{12}$), причем данные для первого этапа преобразования $2 \rightarrow 1$ лежат во втором слое. Следовательно, умножитель $A1$ вычисляет произведение двух n -разрядных чисел за $n+6$ тактов, т. е. по сравнению с умножителем A вычислительный процесс растягивается на один такт.

В соответствии с выделенными подустройствами микрокоманды из $\Phi(A)$ по подмножествам $\hat{\Theta}^k$, $\hat{\Theta}^k \in \hat{\Phi}_1(A)$, распределяются следующим образом: $\hat{\Theta}^0 = \{\Theta_2^0, \Theta_3^0, \Theta_{13}^0\}$, $\hat{\Theta}^1 = \{\Theta_1^1, \Theta_4^1, \Theta_5^1, \Theta_7^1, \Theta_8^1, \Theta_{13}^1, \Theta_{14}^1\}$, $\hat{\Theta}^2 = \{\Theta_5^2, \Theta_6^2, \Theta_7^2, \Theta_9^2, \Theta_{13}^2\}$ и $\hat{\Theta}^3 = \{\Theta_{10}^3, \Theta_{11}^3, \Theta_{12}^3, \Theta_{13}^3\}$, где микрокоманда Θ_k^i получена подстановкой соответствующих значений координаты k в именуемые функции базы и контекста микрокоманды Θ . Поскольку умножитель $A1$ растягивает вычислительный процесс на один такт за счет подъема результата последнего преобразования $5 \rightarrow 3$ на первый слой, то в микрокомандах, реализующих преобразования $3 \rightarrow 2$ и $2 \rightarrow 1$, изменятся имена (координаты) управляющих клеток, соответствующих клеткам регистра P . В качестве примера выпишем функциональные микрокоманды из подмножества $\hat{\Theta}^2$:

$$\begin{aligned} \Theta_6^2 : \{(\#, \langle M, 1, j+2, 2 \rangle)(\#, \langle M, 2, j+1, 2 \rangle)(\#, \langle M, 3, j, 2 \rangle)\} * \{(x, \langle M, 1, j, 1 \rangle) \\ (y, \langle M, 2, j, 1 \rangle)(z, \langle M, 3, j, 1 \rangle)(u, \langle M, 4, j, 1 \rangle)(w, \langle M, 5, j, 1 \rangle) \\ (1, \langle M, 0, 0, 2 \rangle)(0, \langle M, 5, n+1, 2 \rangle)\} \rightarrow \{(f_2, \langle M, 1, j+2, 2 \rangle) \\ (f_3, \langle M, 2, j+1, 2 \rangle)(f_4, \langle M, 3, j, 2 \rangle)\}, \end{aligned}$$

где f_2, f_3 и f_4 вычисляются по формулам (1a) — (1в);

$$\begin{aligned} \Theta_9^2 : \{(\#, \langle M, 1, j+1, 2 \rangle)(\#, \langle M, 2, j+1, 2 \rangle)(\#, \langle M, 3, j, 2 \rangle)(\#, \langle M, 4, j, 2 \rangle)\} * \\ \{(x, \langle M, 1, j, 1 \rangle)(y, \langle M, 2, j, 1 \rangle)(z, \langle M, 3, j, 1 \rangle)(0, \langle M, 5, n+3, 2 \rangle) \\ (1, \langle M, 5, n+2, 2 \rangle)\} \rightarrow \{(f_4^s, \langle M, 1, j+1, 2 \rangle)(f_4^m, \langle M, 2, j+1, 2 \rangle) \\ (f_5^s, \langle M, 3, j, 2 \rangle)(f_5^m, \langle M, 4, j, 2 \rangle)\}, \end{aligned}$$

где f_4^s, f_4^m и f_5^s, f_5^m — знак и модуль функций суммы и переноса преобразования $3 \rightarrow 2$, $f_4^s = f_5^s = 0$, функции f_4^m и f_5^m вычисляются по формулам (2a) и (2б).

Второй вариант разбиения клеточного умножителя A . Этот вариант разбиения умножителя A отличается от предыдущего тем, что второе подустройство формирует данные только для преобразований $5 \rightarrow 3$. Это означает, что умножитель $A2$ не поднимает результата последнего преобразования со второго слоя на первый, а сразу же определяет двурядный код произведения. Следовательно, умножитель $A2$ вычисляет произведение двух чисел с той же скоростью, что и умножитель A . Микрокоманды из $\Phi(A)$ по подмножествам описания $\hat{\Phi}_2(A)$ распределяются аналогично описанию $\hat{\Phi}_1(A)$, причем их текст не меняется, за исключением введения соответствующих значений координаты k в именуемые функции. Допол-

нительная микрокоманда Θ_{14} от своего аналога из описания $\hat{\Phi}_1(A)$ отличается именем контекстной клетки, управляющей подъемом результата преобразования $5 \rightarrow 3$.

Данный способ расслоения двумерного МО фактически распределяет микрокоманды согласно заданному варианту разбиения между слоями трехмерного устройства. В результате такого расслоения лишь часть связей переходит в межслойное пространство, во-первых, и, во-вторых, некоторые слои устройства участвуют в вычислительном процессе в течение нескольких тактов. Поскольку мы ориентируемся на электрооптическую реализацию, такие слои будем использовать для пересвода внутрислойных глобальных связей в локальные межслойные.

Рассмотрим, например, микрокоманду Θ_6^2 . Пусть клетка с именем $\langle M, 2, j+1, 2 \rangle$ является функциональной. Из заданного варианта выделения подустройств видно, что третий слой умножителя A_2 до вычисления двурядного кода произведения не работает. Следовательно, глобальная связь управляющей клетки $(1, \langle M, 0, 0, 2 \rangle)$ с функциональными может быть преобразована в локальные, если данную управляющую клетку заменить на новые управляющие клетки, расположенные непосредственно под функциональными. Тогда микрокоманда Θ_6^2 имеет вид

$$\begin{aligned} \Theta_6^2 : \{ (\# , \langle M, 1, j+2, 2 \rangle) (\# , \langle M, 2, j+1, 2 \rangle) (\# , \langle M, 3, j, 2 \rangle) \} * \{ (x, \langle M, 1, j, 1 \rangle) \\ (y, \langle M, 2, j, 1 \rangle) (z, \langle M, 3, j, 1 \rangle) (u, \langle M, 4, j, 1 \rangle) (w, \langle M, 5, j, 1 \rangle) \\ (1, \langle M, 2, j+1, 3 \rangle) (0, \langle M, 5, n+1, 2 \rangle) \} \rightarrow \{ (f_1, \langle M, 1, j+2, 2 \rangle) \\ (f_2, \langle M, 2, j+1, 2 \rangle) (f_3, \langle M, 3, j, 2 \rangle) \}. \end{aligned}$$

Аналогичные преобразования могут быть выполнены с управляющей клеткой $(0, \langle M, 0, 0, 1 \rangle)$ в микрокомандах подъема результата преобразования $5 \rightarrow 3$ и сдвига ЧП из четвертой строки в третью первого слоя структуры. Введение новых управляющих клеток, естественно, влечет за собой соответствующие изменения в описании $\hat{\Phi}_2(A)$. Полученная таким образом параллельная микропрограмма $\hat{\Phi}_3(A)$ не будет противоречивой. Умножитель \hat{A}_3 , реализующий описание $\hat{\Phi}_3(A)$, вычисляет произведение двух чисел за $n+5$ тактов.

4.2. *Расслоение микропрограммного описания алгоритма В.* Расслоение микропрограммного описания $\Phi(B)$ выполняется аналогично описанию $\Phi(A)$. Отметим лишь особенность работы многослойного умножителя \hat{B} .

Пусть умножитель \hat{B} представляет собой четырехслойную структуру. На нулевом слое структуры расположены исходные данные, на первом — формируются частичные произведения и результаты преобразования $3 \rightarrow 2$ (но не последнего), переданные со второго слоя, на втором и третьем слоях выполняются преобразования $3 \rightarrow 2$ и $2 \rightarrow 1$ соответственно. Первые три такта умножитель \hat{B} формирует ЧП, на четвертом такте вычисляет очередное ЧП на первом слое и преобразование $3 \rightarrow 2$ — на втором. Результат преобразования возвращается на первый слой на пятом такте, и здесь же вновь сформированное ЧП (пятое) записывается на место четвертого, т. е., начиная с третьего такта, каждое четное частичное произведение стирается. Для того чтобы умножители \hat{B} и \hat{B} выполняли один и тот же алгоритм, необходимо, начиная с третьего такта работы умножителя \hat{B} , формировать ЧП через такт. Это означает, что двурядный код произведения умножитель \hat{B} вычислит за $2n-2$ такта, а истинное произведение — за $2n+1$ такт. Таким образом, расслоение умножителя \hat{B} приводит к растяжению вычислительного процесса почти вдвое, если подустройства (формирующее ЧП и реализующее преобразование $3 \rightarrow 2$) расположены в разных слоях структуры.

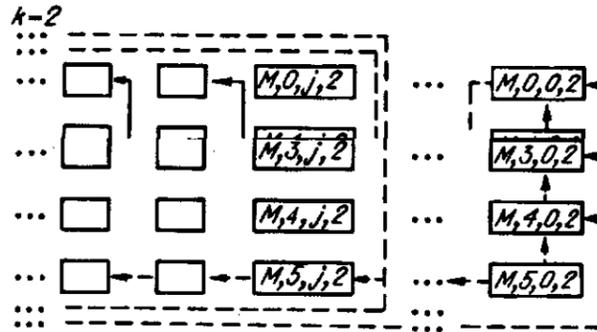


Рис. 7

Микрокоманды из $\Phi(B)$ по подмножествам описания $\hat{\Phi}(B)$ распределяются аналогично описанию $\Phi_2(A)$, однако их текст из-за уменьшения скорости вычисления произведения изменяется в большей степени по сравнению с микрокомандами из $\Phi(A)$.

4.3. Электрооптическая реализация клеточного умножителя \hat{A}_1 . Структурная схема многослойного умножителя \hat{A}_1 строится аналогично схеме однослойного умножителя A . Сложность электрооптической реализации умножителя \hat{A}_1 будем определять по наиболее сложному логическому слою в оценках, принятых в п. 3.3. Под наиболее сложным логическим слоем будем пони-

Таблица 4

Оценки сложности реализации расслоенного умножителя	Умножители			
	\hat{A}_1	\hat{A}_2	\hat{A}_3	\hat{B}
T	$n + 6$	$n + 5$	$n + 5$	$2n + 1$
$S \left(\frac{\text{эл}}{\text{оп}} \right)$ умножителя	$\frac{32}{14}$	$\frac{33}{14}$	$\frac{31}{16}$	$\frac{29}{11}$
V умножителя	$0,337n$	$0,358n$	$0,365n$	$0,184n$
Количество типов ФЯ	4	4	4	4
Наиболее сложный слой	2	2	2	3
$S \left(\frac{\text{эл}}{\text{оп}} \right)$ НС-слоя	$\frac{10}{5}$	$\frac{11}{5}$	$\frac{10}{6}$	$\frac{12}{4}$
V НС-слоя	$0,162n$	$0,178n$	$0,178n$	$0,115n$
$A_{\text{яч}}$ НС-слоя	0,153	0,169	0,169	0,112

мать слой, у которого функциональная ячейка имеет максимальную площадь.

В умножителе \hat{A}_1 наиболее сложным слоем является второй, его фрагмент приведен на рис. 7. Из рисунка видно, что полученная структура квазиоднородна. Функциональная ячейка с именем $(M, 2, j, 2)$, выполняющая преобразования $5 \rightarrow 3$ и $3 \rightarrow 2$, имеет 10 входов сравнения, 5 из них оптические (на схеме они обозначены косыми линиями), и 5 выходов записи. Ее площадь составляет $0,148 \text{ мм}^2$.

У умножителя четыре типа функциональных ячеек. Оценки временной, структурной и аппаратной сложности умножителя соответственно равны: $T = n + 6$, $S = (32/14)$ (в числителе связи электрические, в знаменателе — оптические) и $V = 0,337n \text{ мм}^3$ (табл. 4). При некотором увеличении оценок сложности многослойного ум-

ножителя по сравнению с однослойным (см. табл. 3 и 4) видно, что топология наиболее сложного (НС) слоя умножителя \hat{A}_1 проще топологии умножителя A и уменьшилась доля электрических связей (10 вместо 17).

В табл. 4 приведены оценки электрооптической реализации еще двух вариантов расслоения умножителей \hat{A} и B . Разброс по оценкам у первых трех умножителей невелик. Умножитель \hat{A}_1 имеет меньшую структурную и аппаратную сложности по сравнению с умножителями \hat{A}_2 и \hat{A}_3 , но уступает им по временной сложности на один такт. Умножители \hat{A}_2 и \hat{A}_3 по сложности второго слоя равноценны, хотя топология второго слоя умножителя \hat{A}_3 из-за замены внутрислойной глобальной связи на локальные проще. Это, в свою очередь, привело к незначительному увеличению аппаратной и структурной сложности.

Расслоение умножителя B сохраняет особенности расслоения умножителя A в смысле аппаратной и структурной сложности, но растягивает вычислительный процесс почти вдвое.

Заключение. На примере предложенных алгоритмов умножения целых n -разрядных чисел продемонстрирована техника архитектурного проектирования умножителей с различным количеством слоев. Использование описанного метода расслоения, а также возможностей электрооптической схемотехники позволяет построить спектр структурных схем умножителя, реализующего один и тот же алгоритм, и выбрать оптимальный вариант схемы с учетом технических требований на устройство.

СПИСОК ЛИТЕРАТУРЫ

1. Косцов Э. Г., Мишин А. И. Особенности построения оптических ЦВМ // Микроэлектроника.—1977.—6, вып. 2.
2. Haugen P. R., Rychnovsky S., Hutcheson L. D. Optical interconnects for high-speed computing // Opt. Eng.—1986.—25, N 10.—P. 1076.
3. Егоров В. М., Косцов Э. Г. Микроэлектронные оптические цифровые вычислительные устройства // Автометрия.—1989.—№ 3.
4. Egorov V. M., Kossov E. G. Integral optical computers // Appl. Opt.—1990.—29, N 8.—P. 1178.
5. Бандман О. Л. Электрооптическая реализация клеточных структур обработки данных // Автометрия.—1992.—№ 1.
6. Пискунов С. В. Декомпозиция алгоритмов параллельных подстановок и расслоение однородных вычислительных структур // Параллельные алгоритмы и структуры.—Новосибирск: ВЦ СО АН СССР, 1991.
7. Пискунов С. В. Построение многослойных электрооптических структур // Автометрия.—1992.—№ 4.
8. Анишев П. А., Ачасова С. М., Бандман О. Л. и др. Методы параллельного микропрограммирования.—Новосибирск: Наука, 1981.
9. Bandman O. L., Piskunov S. V. Parallel substitution algorithm as a model for distributed computations // J. New Generat. Syst.—1990.—3, N 4.—P. 3.
10. Куз С. Матричные процессоры на СБИС.—М.: Мир, 1991.
11. Маркова В. П., Пискунов С. В. Универсальная процедура построения структурных схем специализированных параллельных вычислителей // Математическое и архитектурное обеспечение параллельных вычислений.—Новосибирск: ВЦ СО АН СССР, 1989.
12. Dadda L. Some schemes for parallel multipliers // Alta Frequenza.—1965.—34, N 5.—P. 349.
13. Wallace C. S. A suggestion for a fast multipliers // IEEE Electron. Comput.—1964.—EC-34, N 2.—P. 114.
14. Avizienis A. Signal-digit number representation for fast parallel arithmetic // IRE Trans. Electron. Comput.—1961.—EC-10, N 3.—P. 45.
15. Hwang K., Louri A. Optical multiplication and division using modified signed symbolic substitution // Opt. Eng.—1989.—28, N 4.—P. 364.
16. А. с. 436350 СССР. Двоичный сумматор / Ю. Н. Корнев, С. В. Пискунов, С. Н. Сергеев.—Заявл. 14.06.71; Опубл. 15.07.74, Бюл. № 26.
17. Погудин Ю. М. АЛТ — графическая система параллельного микропрограммирования // Параллельные алгоритмы и структуры.—Новосибирск: ВЦ СО АН СССР, 1991.

Поступила в редакцию 3 марта 1993 г.