

ПЕРСПЕКТИВНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

УДК 621.578

Э. Г. Косцов, С. В. Пискунов

(Новосибирск)

КОМПЬЮТЕРНОЕ ПРОЕКТИРОВАНИЕ  
ДВУХСЛОЙНОЙ ВЫЧИСЛИТЕЛЬНОЙ МАТРИЦЫ  
С ОПТИЧЕСКИМИ МЕЖСОЕДИНЕНИЯМИ

Предложена методика компьютерного проектирования двухслойной вычислительной матрицы (чипа) с оптическими межсоединениями. Каждый слой матрицы является перестраиваемой клеточной структурой с локальными связями между клетками. Логические сигналы между слоями передаются по оптическим каналам связи. Конструкция матрицы допускает возможность наращивания ее размеров за счет оптическойстыковки чипов. Путем имитационного моделирования получены оценки сложности рассматриваемой матрицы и показан способ организации на ее основе массовых конвейерных вычислений.

**Введение.** Экспоненциальный закон повышения производительности микроэлектронных систем, эмпирически установленный Гордоном – Муром, достаточно хорошо выполняется на протяжении почти 40 лет: вычислительные мощности однокристальных микропроцессорных систем каждые 3 года увеличиваются в 4 раза. Этому закону подчиняется и прогресс технологий микроэлектроники: ежегодное уменьшение размеров транзисторов на 13 % – от 20 мкм в 1960 году до 0,18 мкм в 1999 году, плотность их размещения при этом удваивается за каждые 1,8 года. Однако его дальнейшее выполнение не является очевидным.

Последующее развитие микроэлектроники связано с решением проблемы соединений в первую очередь на поверхности чипов. Соединения в логических микросхемах занимают до 70 % всей площади кристалла, а их проектирование требует больших усилий. По мере роста быстродействия транзисторов на перезарядку соединений уходит до 60–70 % энергии. Решение проблемы можно искать на пути использования в интегральных схемах оптических каналов связи.

В работах [1–3] было показано, что введение в структуру интегральной логической схемы новой компоненты – тонкопленочного электрооптического модулятора света – позволяет сокращать площадь, занимаемую соединениями, до 5–7 % и уменьшать более чем на два порядка число активных элементов по сравнению с обычной микросхемой с такими же функциональными возможностями. Предложенная схема имела многослойную структуру: в

каждом слое межэлементные связи выполнялись так же, как в обычных микролазерных схемах, а передача логических сигналов между слоями осуществлялась за счет модуляции света, проходящего через слои нормально к их поверхности.

Из специфики распространения дифрагированного светового пучка следует, что для исключения потерь информации расстояние между источником света (окном модулятора) и фотоприемником не должно превышать  $\sqrt{\lambda}b$ , где  $\lambda$  – длина световой волны;  $b$  – размер окна модулятора. Следовательно, при росте числа слоев, на которых размещены логические схемы, возрастает величина  $b$  и уменьшается плотность размещения элементов.

Поэтому одной из важнейших задач при разработке оптических логических устройств является не только сокращение длины соединений, но и уменьшение числа слоев и расстояния между ними.

Для решения этой задачи в работе [4] был предложен настраиваемый клеточный автомат, соединяющий в себе вычислительную среду [5] и клеточный автомат с окрестностью Марголуса [6]. В работе [7] показана возможность его применения в качестве логического базиса для построения семейства многослойных вычислительных матриц, использующих элементную базу, предложенную в [1–3]. Ее основу составляет оптический ключ, содержащий модулятор и фотоприемник. Эти элементы расположены в разных слоях один под другим и обеспечивают связь слоев между собой. Показана принципиальная возможность построения матрицы с минимальным числом слоев, равным двум. При этом первый слой матрицы состоит из идентичных регистров сдвига, в которых оптические входы и выходы ячеек памяти связаны световыми сигналами с оптическими входами и выходами схем совпадений, находящихся во втором слое. Схемы совпадений образуют матрицу идентичных схем. Схема совпадений и соответствующий ей участок регистра сдвига выполняют функции клетки настраиваемого клеточного автомата. Такой подход обеспечивает перевод большей части логических связей в межслойные связи и исключает логические связи между регистрами сдвига и схемами совпадений в каждом из слоев.

Двухслойная матрица выполняет преобразование входных данных в конвейерном режиме и, следовательно, может имитировать работу множества однотипных цифровых схем. Однако в отличие от других матриц семейства, которые настраиваются на имитацию работы цифровой схемы с помощью дополнительных слоев, в двухслойной матрице настройка выполняется путем специального маскирования оптических элементов.

В данной работе предложена методика компьютерного проектирования такой матрицы и процедура ее настройки, получены оценки производительности и технологичности, рассмотрены возможные применения. Матрица строится в три этапа с использованием системы моделирования распределенных в пространстве вычислений WinALT [8], которая имеет развитые средства графического конструирования, отладки и выполнения клеточных моделей на ЭВМ. Она основана на модели распределенных вычислений (алгоритме параллельных подстановок – АПП) [9, 10].

На каждом из последующих этапов происходит постепенная детализация структуры матрицы с учетом специфики физической реализации устройства. На заключительном этапе в модели присутствуют в виде клеток образы всех оптических компонент матрицы, а электронные компоненты заданы через выполняемые ими логические функции.

Необходимые понятия, связанные с визуальным представлением клеточного автомата, заимствуются из [7].

**Имитационная модель двухслойной матрицы.** Процесс конструирования модели двухслойной логической матрицы на каждом этапе состоит из двух процедур: а) построение модели матрицы и б) ее настройка на реализацию заданного цифрового устройства.

Достоверность процедур построения матрицы для обеспечения корректного перехода с этапа на этап обеспечивается с помощью машинного моделирования, которое выполняется с использованием системы WinALT.

Представляя достаточно детально модели клеточных автоматов, из которых шаг за шагом «выкристаллизовывается» матрица, мы ставим следующие задачи:

- показать конвейерность матрицы (возникновение в матрице виртуальных цифровых устройств);
- продемонстрировать компьютерную технологию реализации в электрооптической матрице высокопроизводительных устройств;
- получить информацию о логической структуре настроенной матрицы для последующего обоснования ее технических возможностей.

Опишем этапы конструирования на примере матрицы, реализующей полный сумматор.

**Первый этап. Построение модели матрицы.** Как и в [7], графически автомат изображается в виде плоскости, разделенной на клетки (рис. 1, a). Разбиение автомата на блоки размером  $2 \times 2$  клетки, ограниченные сплошными линиями, назовем  $E$ -разбиением, а разбиение автомата на блоки, ограниченные штриховыми линиями, –  $O$ -разбиением. Каждая клетка может находиться в трех состояниях: белом, сером и черном. Исходное состояние автомата – все клетки белые.

**Отображение цифровой схемы в клеточный автомат.** Цифровая схема представлена совокупностью серых клеток. Сигналы, преобразуемые в цифровой схеме, изображаются черными клетками. На рис. 1, a показан клеточный автомат **S1**, содержащий образ полного сумматора. Состояния клеток каждого блока автомата образуют некоторую картинку – образ блока.

В рассматриваемом автомате блок из  $E$ -разбиения может быть настроен на выполнение не более чем двух любых команд, блоки из  $O$ -разбиения команд не выполняют. На рис. 1, b представлена настройка автомата **S1**: указаны номера команд, на выполнение которых настроены блоки. Эти команды показаны на рис. 1, c. Они состоят из левых и правых частей, разделенных стрелкой. Команда, номер которой указан в блоке на рис. 1, b первым, может

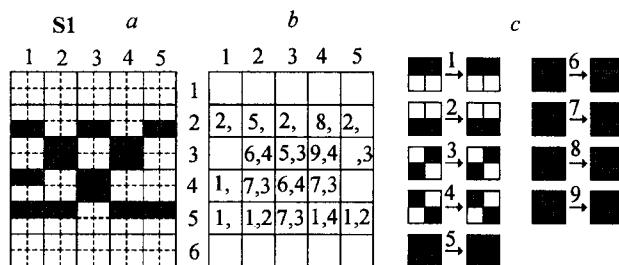


Рис. 1. Графическое представление клеточного автомата: *a* – автомат **S1**, содержащий образ полного сумматора; *b* – настройка автомата **S1**; *c* – состав команд

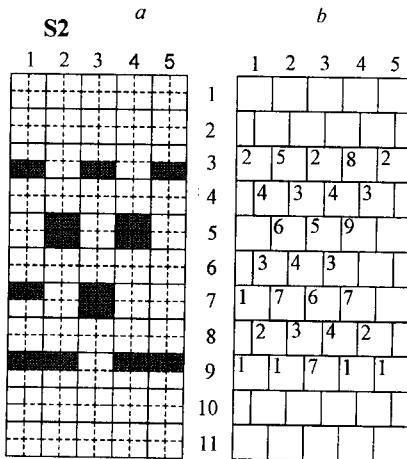
быть выполнена для исходного расположения образа схемы и для образа, сдвинутого на один ряд блоков вверх (вниз) по отношению к его исходному расположению. Эти сдвиги формируют  $E$ -группу сдвигов. Команда, номер которой указан в блоке на рис. 1,  $b$  вторым, может быть выполнена для образа схемы, сдвинутого по отношению к исходному образу на один ряд клеток вниз и один столбец клеток вправо. Эти сдвиги формируют  $O$ -группу сдвигов. Команда применима, если ее левая часть совпадает с образом блока, который настроен на ее реализацию. Выполнение команды состоит в том, что образом блока становится ее правая часть. Команды с номерами 5–7 имитируют функцию полусумматора, с номерами 7–9 – функцию ИЛИ, остальные команды имитируют передачу сигналов в образе схемы. До начала работы клеточного автомата в него вводятся исходные данные. Ввод заключается в том, что часть или все входные клетки (это серые клетки первого столбца клеток автомата) образа цифровой схемы заменяются черными клетками.

Машинная модель автомата  $S1$  построена; она сходна с моделью  $S2$ , получаемой из  $S1$  на втором этапе. Далее описывается детально только модель  $S2$ , так как она непосредственно используется на заключительном этапе конструирования матрицы.

**Второй этап. Построение модели матрицы.** Выполняется модификация клеточного автомата, полученного на первом этапе, таким образом, чтобы отдельный блок настраивался на выполнение только одной команды. Для этого число блоков в клеточном автомате удваивается, группа  $O$ -сдвигов заменяется перестановкой верхнего и нижнего рядов клеток в каждом блоке  $O$ -разбиения, а сдвиги из  $E$ -группы осуществляются на два ряда блоков вверх и вниз.

**Отображение цифровой схемы в клеточный автомат.** В результате модификации клеточного автомата  $S1$  получается новый автомат  $S2$ , показанный на рис. 2,  $a, b$ . Отметим, что список команд остается прежним. Все описанные построения были выполнены на ЭВМ. Графическое представление на экране монитора модели клеточного автомата  $S2$  приведено на рис. 3.

Опишем структуру и функционирование модели автомата  $S2$ . В левом верхнем углу графического окна **fsumsh3.3do\*** в штриховой рамке показан образ автомата  $S2$ . Этот образ содержится в двухслойном клеточном массиве **int8::ffieldw**. Для удобства на рис. 3 представлена послойная развертка массива. Автомат показан в тот момент, когда выполнено чередование верхнего и нижнего рядов клеток в каждом блоке  $O$ -разбиения. Номера команд подстановок кодируются цветом, но при желании числовый номер той или иной команды можно увидеть в статусной строке. Остальные графические объекты окна можно разделить на три группы. Первую группу образуют два нижних ряда шаблонов. Эти шаблоны задают левые



Rис. 2. Графическое представление клеточного автомата, полученного в результате «растяжения» автомата  $S1$ :  $a$  – автомат  $S2$ , содержащий образ полного сумматора;  $b$  – настройка автомата  $S2$

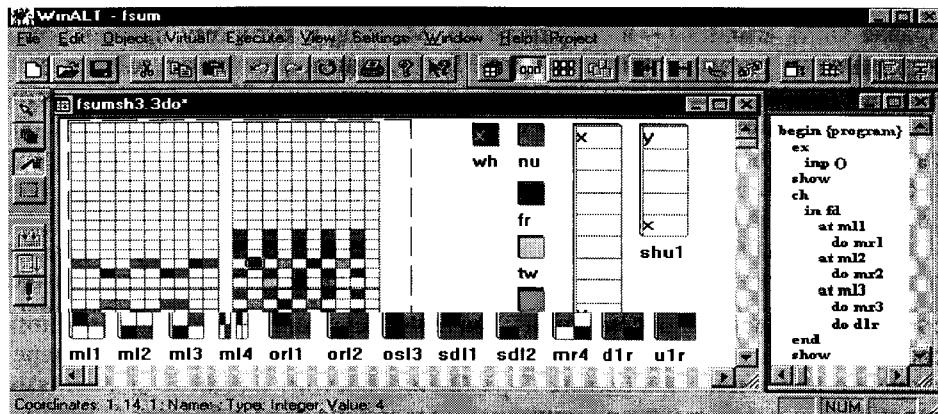


Рис. 3. Компьютерная модель автомата S2

и правые части команд подстановок, изображенных на рис. 1. Каждый шаблон состоит из двух слоев. В первом слое содержится графический образ либо левой, либо правой части команды подстановки. Если графический объект изображает левую часть команды, тогда в левой верхней клетке второго слоя указывается номер этой команды, а остальные клетки пусты: в них вписан символ типа **void**. Если графический объект изображает правую часть команды, тогда все клетки второго слоя пусты. В качестве примера в графическом окне показана развертка шаблона с именем **ml4**, поставленного в соответствие левой части команды с номером 4. Вторую группу образуют шаблоны с именами **wh**, **nu**, **fr**, **tw**, **th**, **fo**. Шаблон **wh** символизирует часы, в которых переменная **x** циклически принимает значения, заданные остальными шаблонами. Третью группу образуют шаблоны, обеспечивающие выполнение сдвигов. Сдвиг на два ряда блоков вверх обеспечивается с помощью шаблона **shu1**, на четыре ряда блоков вниз – с помощью шаблона **shd1**, перестановка рядов клеток осуществляется с помощью шаблона **shp1**.

Моделирующая программа выполняет пошаговую имитацию работы клеточного автомата. Команды подстановок выполняются на каждом шаге. Кроме того, на первом шаге (**x = nu**) производится ввод данных и перестановка рядов клеток. На втором шаге (**x = fr**) вновь выполняется перестановка рядов клеток, на третьем шаге (**x = tw**) – сдвиг на два ряда блоков вверх, на четвертом шаге (**x = th**) – сдвиг на четыре ряда блоков вниз, на пятом шаге (**x = to**) – сдвиг на два ряда блоков вверх. Эти шаги повторяются многократно. В системе WinALT многие операторы, используемые для записи моделирующих программ, такие же, как и в [4]. Оператор **ex** задает итеративное повторение шагов моделирования. Оператор **ch** задает один шаг выполнения блока операторов. На рис. 3 в окне справа показан фрагмент моделирующей программы, содержащий блок операторов, которые описывают команды подстановок, изображенные на рис. 1, с. Отдельная команда подстановки задается парой операторов **at - do**. Левая часть команды указывается в операторе **at**,

правая часть команды – в операторе **do**. Массив, к которому применяются подстановки, указывается в операторе **in**. В данном фрагменте, вместо полного имени **int::ffieldw**, использован алиас (сионим) **fd**. Команда подстановки выполняется, если первый слой шаблона левой части совпадает с образом блока в первом слое массива **fd**, а цвет левой верхней клетки второго слоя шаблона совпадает с цветом аналогично расположенной клетки во втором слое образа блока.

Процедуры ввода, вывода данных, сдвига блоков клеток и перестановки рядов клеток записываются в системе WinALT на паскалеводобном языке. Ввод данных в первый слой клеточного массива **fd** на каждом первом шаге состоит в замене в первом столбце серых клеток черными в соответствии с очередной строкой полной булевой таблицы от трех переменных. Имя **inp** этой процедуры можно увидеть в записи фрагмента моделирующей программы. Вывод данных на очередном первом шаге состоит в замене черных клеток серыми в последнем столбце первого слоя. Процедуры сдвига блоков состоят в присвоении переменной **у** значения переменной **x** при применении соответствующего шаблона к столбцам клеток массива **fd**. Процедура перестановки рядов клеток в блоках *O*-разбиения состоит в обмене значениями переменных **x** и **m**, **у** и **n** при применении шаблона **shp1** к блокам *O*-разбиения. Чтобы шаблон применялся только к этим блокам, они помечаются.

На рис. 3 пометками служат серые клетки самого верхнего ряда второго слоя и далее через ряд. Такую же пометку (с совпадающим кодом цвета) имеет левая верхняя клетка второго слоя шаблона **shp1**. Назовем построенную модель **M1**. Она позволяет перед переходом к последнему этапу убедиться в правильности функционирования клеточного автомата. Кроме того, можно убедиться в его конвейерности. Действительно, легко видеть, что в модели автомата одновременно присутствуют четыре комплекта данных (111, 110, 101, 100) на разных стадиях переработки. Нужно только знать, что при вводе единица изображается черной клеткой, нуль образа не имеет. Первый разряд комплекта данных поступает на верхний вход сумматора, второй разряд – на средний вход и третий разряд – на нижний вход. Далее, на очередном первом шаге будет считан результат, а на входы сумматора поступит новый комплект данных.

**Третий этап. Построение модели матрицы.** Выполняется замена графического образа автомата, полученного на втором этапе, графическим образом клеточного автомата – имитатора двухслойной электрооптической матрицы. Смысль замены заключается в переходе от троичной кодировки состояний клеток автомата, полученного на втором этапе, к двоичной кодировке состояний клеток автомата-имитатора и в специализации клеток автомата-имитатора в соответствии со спецификой электрооптической элементной базы [1–3]. Отметим, что будет использоваться еще одно, особое, состояние клетки: в этом состоянии клетка не участвует в преобразовании информации. Такое состояние будем называть состоянием маскирования и изображать серым цветом.

Построим первый слой автомата-имитатора. Исходным материалом для его построения служат графический образ автомата, полученный на втором этапе, и графический образ ячейки памяти 3 с двумя модуляторами 1, 2 на выходе и двумя фотодетекторами 4, 5 на входе (рис. 4, *a*, *b*). Белая клетка изображает единичное состояние ячейки памяти, открытый модулятор и открытый фотодетектор, черная клетка – нулевое состояние ячейки памяти, закрытый модулятор и закрытый фотодетектор. Фактически мы используем пара-

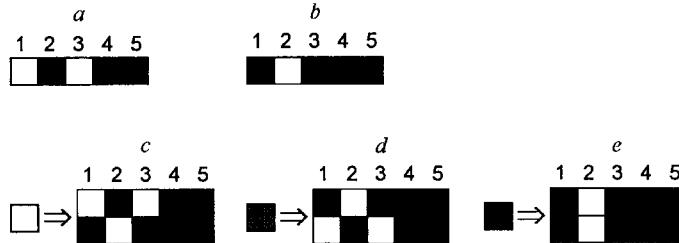


Рис. 4. Графический образ ячейки памяти: *a* – в состоянии 1, *b* – в состоянии 0. Правила двоичной кодировки клеток: *c* – белой, *d* – серой, *e* – черной

фазное кодирование логического сигнала в каждой ячейке матрицы, что при реальной физической реализации устройства позволяет исключить влияние фонового сигнала и уменьшить требования к идентичности параметров модуляторов света и фотоприемников, т. е. поднять технологичность устройства. Первый слой автомата получается в результате замены белых, серых и черных клеток группами клеток по правилам, показанным на рис. 4, *c–e*. Правила составлены с использованием кодов, изображенных на рис. 4, *a, b*. Блок имеет размеры  $10 \times 4$  клетки.

Далее под первым слоем помещается второй слой. Этот слой составлен из блоков такого же размера, как и блоки первого слоя. Расположение блоков в слое совпадает с расположением блоков в таблице настройки автомата. В отдельном блоке черные клетки в столбцах с номерами 1, 2, 6, 7 изображают закрытые фотодетекторы, остальные столбцы замаскированы: в них расположены серые клетки.

Таким образом, автомат-имитатор электрооптической матрицы построен.

*Отображение цифровой схемы в клеточный автомат.* Опишем его настройку на реализацию заданной цифровой схемы. Настройка выполняется с помощью масок команд. Для того чтобы их построить, первоначально нужно выполнить перекодировку клеток команд. Правила перекодировки показаны на рис. 5, *a–c*. Суть правил состоит в том, что в коде (см. рис. 4, *c–e*), поставленном в соответствие клетке, выделяется фрагмент, содержащий два первых столбца, и в этом фрагменте изображение открытого модулятора заменяется пустой клеткой (на рисунке она выделена штриховым контуром), а изображение закрытого модулятора – маской (клеткой серого цвета). Затем код команды отображается в блок размером  $10 \times 4$  клетки. Механизм отображения покажем на примере. На рис. 5, *d* изображен блок с кодом левой части

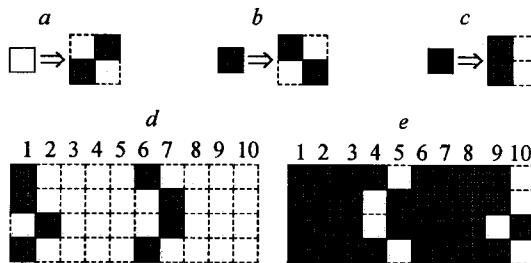


Рис. 5. Маски: *a* – белой, *b* – серой, *c* – черной клеток, *d* – левой части команды с номером 1, *e* – правой части этой команды

команды с номером 1, на рис. 5, e – блок с кодом правой части той же команды. В первом блоке столбцам с номерами 1, 2 и 6, 7 поставлены в соответствие коды первого и второго столбцов левой части команды, во втором блоке столбцам с номерами 4, 5 и 9, 10 – коды первого и второго столбцов правой части команды, остальные столбцы составлены в первом блоке из пустых клеток, во втором блоке из серых клеток. Полученный код назовем маской команды. Для остальных команд их маски получаются аналогично. Настройка первого слоя состоит в том, что клетки маски левой части команды накладываются на клетки того блока, которому в соответствии с таблицей настройки приписан номер этой команды. При этом клетка, расположенная под пустой клеткой, не меняется, а клетка, расположенная под серой клеткой, заменяется серой клеткой. Совершенно аналогично выполняется настройка второго слоя, но только на клетки блока накладывается маска правой части команды и серая клетка, расположенная под белой клеткой, заменяется черной клеткой, изображающей закрытый модулятор.

Графическое изображение на экране монитора слоев автомата-имитатора электрооптической матрицы, синтезированных по автомату S2, дано на рис. 6. Для удобства восприятия показана послойная развертка клеточного массива **int8::EO\_structure**. Сделаем одно замечание. В первом слое все клетки, не вошедшие в площадь, занимаемую образом цифровой схемы, кроме клеток, изображающих ячейки памяти, замаскированы. Во втором слое замаскированы все клетки с такими же координатами (для исключения этих клеток из преобразования информации). Клетки, изображающие ячейки памяти в первом слое, оставлены для того, чтобы обеспечить сдвиги образа цифровой схемы относительно масок команд, эквивалентные сдвигам блок-

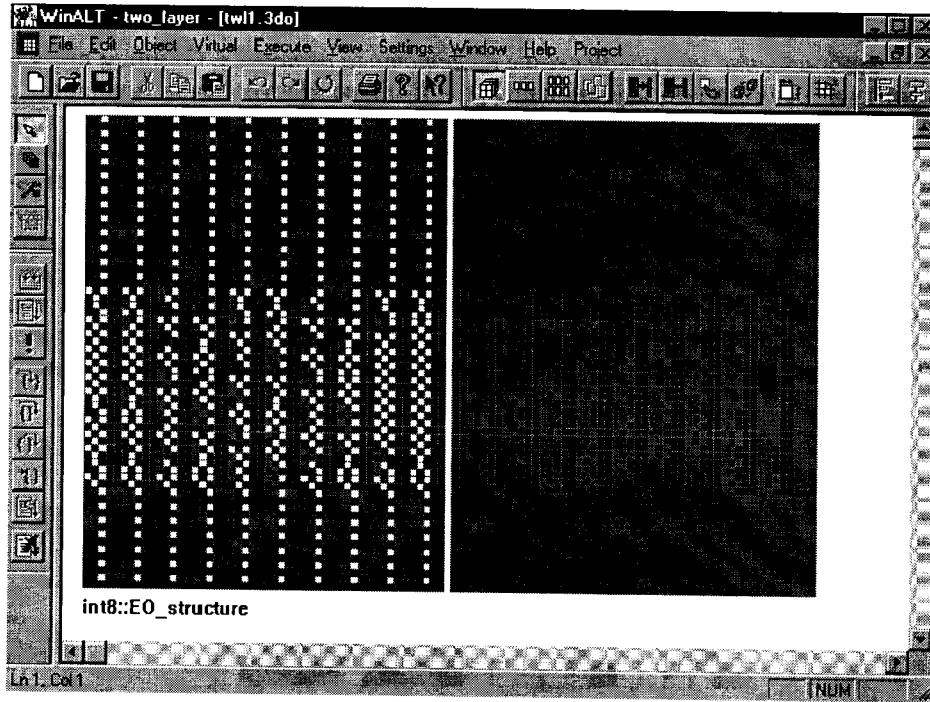


Рис. 6. Изображение на экране монитора слоев автомата-имитатора электрооптической матрицы

ков и перестановке рядов клеток в модели **M1**. С учетом сделанного замечания из сопоставления рис. 2 и рис. 6 легко видеть, что на рис. 6 представлена кодировка автомата, изображенного на рис. 2. С другой стороны, мы покажем, что имитируемая двухслойная матрица эквивалентна некоторой разновидности четырехслойной матрицы, физическое функционирование которой описано в [7]. Действительно, модификация клеточного автомата, которую мы выполнили на втором этапе, позволяет составить первый (четвертый) слой просто убрать, а модуляторы второго слоя, расположенные под закрытыми окнами, выключить из работы. Такие действия фактически означают настройку двухслойной матрицы на реализацию заданной цифровой схемы с помощью масок команд, что и продемонстрировано на примере цифровой схемы полного сумматора.

Из изложенного выше следует вывод, что моделирующая программа автомата-имитатора двухслойной матрицы должна описывать способ преобразования оптической информации, предложенный в [1–3]. Такая программа была построена с использованием только двух команд подстановок (**P1**, **P2**) и трех функций. Команды подстановок выполняются клетками, расположенными в разных слоях, и описывают взаимодействие модуляторов и фотоприемников. Функции реализуются клетками, расположенными в одном слое, и описывают управление ячейками памяти и схему совпадения. Функция, описывающая схему совпадения, представлена далее под именем **AND\_OR**.

Команда **P1** выполняется в столбцах клеток с номерами  $5i$ ,  $5i+1$ ,  $i=0,1,2,\dots$ , первого и второго слоев и заменяет во втором слое черную клетку белой, если в первом слое над этой клеткой расположена белая клетка. Команда **P1** имитирует прохождение света через открытый модулятор первого слоя на фотодетектор второго слоя и перевод его в низкоомное состояние, причем свет падает сверху вниз.

В столбцах клеток с номерами  $5i+3$ ,  $5i+4$ ,  $i=0,1,2,\dots$ , обоих слоев выполняется симметричная команда **P2**, заменяющая в первом слое черную клетку белой, если под нею во втором слое расположена белая клетка. Эта подстановка также имитирует прохождение света через модулятор на фотодетектор, но в другом направлении – снизу вверх.

Клетки, изображающие ячейки памяти, расположены в столбцах с номерами  $5i+2$ ,  $i=0,1,2,\dots$ , первого слоя. Пометим ячейку памяти **З** (см. рис. 4) переменной **m**, модулятор 1 переменной **x**, модулятор 2 переменной **y**, фотодетектор 4 переменной **z**, фотодетектор 5 переменной **t**. Функция **record\_mem** превращает черную клетку, изображающую ячейку памяти **m** в нулевом состоянии, в белую клетку, если клетка **z** белая. Симметрично эта функция превращает белую клетку, изображающую ячейку памяти **m** в единичном состоянии, в черную клетку, если клетка **t** белая. Функция **record\_mod** превращает клетку **x** в белую, а клетку **y** – в черную, если клетка **m** белая, и наоборот, если клетка **m** черная. Вместе эти функции описывают управление ячейкой памяти по оптическим входам (фотодетекторам) и состояние оптических выходов (модуляторов света) ячейки памяти в зависимости от ее собственного состояния.

Несколько более сложна единственная логическая функция **AND\_OR**, выполняемая клетками блоков второго слоя. Назначение функции **AND\_OR** состоит в регистрации применимых команд подстановок. Пронумеруем столбцы любого блока второго слоя числами 1, 2, ..., 10. Пометим клетки первого ряда столбцов с номерами 1, 2 переменными **x1**, **x2**, второго ряда – **x3**, **x4**, третьего ряда – **x5**, **x6**, четвертого ряда – **x7**, **x8**. Аналогично помечаются клетки рядов столбцов с номерами 6, 7, но только переменными **x9–x16**. Такая же разметка выполняется для клеток столбцов с номерами 4, 5 и 9, 10, но только переменная **xi** заменяется переменной **yi** для  $i=1, \dots, 10$ . Функция **AND\_OR** записана ниже на языке системы WinALT с очевидными сокращениями:

```
function AND_OR;
begin
  if (((x1 = ph_detector_open) or (x2 = ph_detector_open)) and
      ...
      ((x15 = ph_detector_open) or (x16 = ph_detector_open))) then
    begin
      if (y1 <> mask) then y1 := modulator_open; end {if}
      ...
      if (y16 <> mask) then y16 := modulator_open; end {if}
    end
  end {if}
end {function AND_OR}
```

Запись функции соответствует сложившейся нотации языков высокого уровня и, как мы полагаем, не требует особых пояснений. Отметим только, что константа **ph\_detector\_open** подобрана так, что оператор сравнения в круглых скобках истинен, если клетка, помеченная переменной **xi**, белая. Выполнение оператора присваивания означает превращение незамаскированной клетки, помеченной переменной **yi**, в белую клетку. Легко видеть, что функция **AND\_OR**, действительно, отражает механизм регистрации применимых команд подстановок, описанный в [7].

Процедуры сдвига и перестановки рядов клеток выполняются в первом слое только для рядов клеток, содержащих ячейки памяти. Процедура сдвига, совместно с функциями **record\_mem** и **record\_mod**, обеспечивает сдвиг образа схемы на восемь рядов клеток вверх и 16 рядов клеток вниз. Эта процедура-функция реализует группу *E*-сдвигов. Нужно только помнить, что теперь блок имеет высоту в четыре ряда клеток. Процедура перестановки, совместно с функциями **record\_mem** и **record\_mod**, выполняет в каждом блоке перестановку первого и второго рядов клеток с третьим и четвертым рядами клеток и наоборот.

Структура моделирующей программы модели **M2** идентична структуре программы модели **M1**, т. е. программы обеспечивают одно и то же чередование преобразований и сдвигов информации в соответствующих клеточных автоматах. Отличаются указанные программы только последовательностью элементарных преобразований информации.

Преобразование информации, аналогичное замене образа левой части команды подстановки образом правой части этой же команды в программе модели **M1**, в программе модели **M2** состоит из следующих шагов: выполнение команды **P1**, вычисление функции **AND\_OR**, выполнение команды **P2**, вычисление функции **record\_mem**, вычисление функции **record\_mod**.

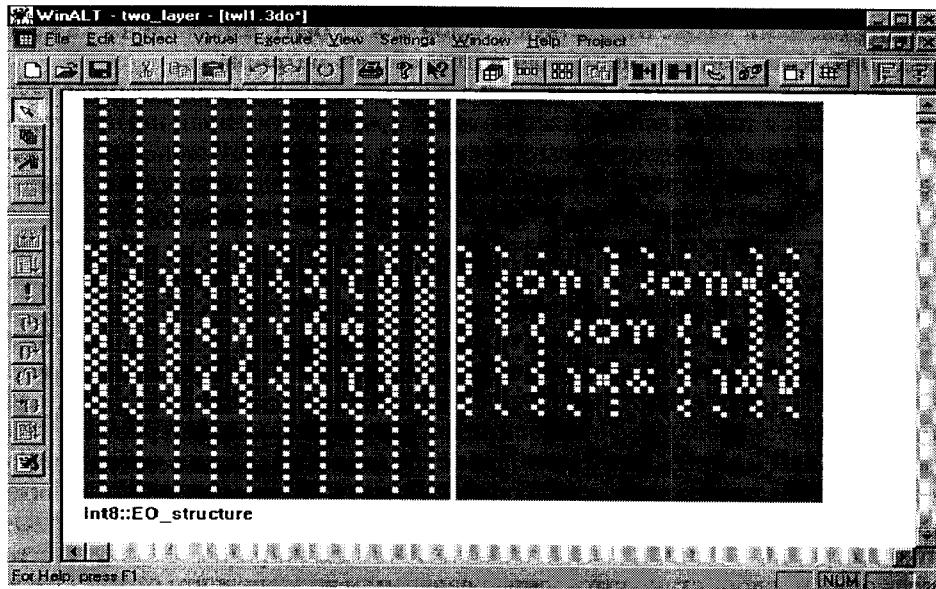


Рис. 7. Образ шага вычислений в автомате-имитаторе

Отождествим каждое конкретное состояние автомата модели **M1** с его кодом в автомате модели **M2**. В этом случае сравнение выполнения моделей **M1** и **M2** на ЭВМ подтверждает их эквивалентность: для одного и того же комплекта входных данных они дают один и тот же результат. Например, автоматы, изображенные на рис. 3, 7, находятся в одинаковой фазе переработки одних и тех же комплектов данных. На экране монитора (см. рис. 7) зафиксирован результат выполнения функции **AND\_OR**: во втором слое открыты модуляторы, кодирующие правые части тех команд, левые части которых совпадали с расположеннымими под ними образами блоков. Далее подстановка **P2** откроет фотодетекторы первого слоя, затем функция **record\_mem** изменит состояния ячеек памяти, а функция **record\_mod** – состояния модуляторов. В итоге позиции сигналов в образах схем на рис. 3, 7 совпадут.

Отметим, что поскольку при построении моделей **M1** и **M2** нигде не использовались ни размеры слоев, ни какие-то особенности цифровой схемы, кроме той, что информация в цифровой схеме всегда «течет» слева направо, то описанная нами процедура пригодна для отображения любой цифровой схемы такого типа в клеточный автомат-имитатор двухслойной электрооптической матрицы.

#### **Преимущества двухслойной электрооптической матрицы для построения вычислительных устройств массовой обработки информации.**

1. *Высокая технологичность.* Действительно, изображение модели на экране показывает, что первый слой матрицы составлен из одинаковых вертикально расположенных двоичных регистров сдвига, ячейки памяти которых имеют оптические входы и выходы (фотодетекторы и модуляторы света соответственно). Топология слоя проста: самая длинная связь в регистре – это связь  $i$ -й ячейки с  $i+16$ -й ячейкой, логические связи в слое между регистрами отсутствуют. Второй слой состоит из итеративно повторяющейся электрооптической схемы, описываемой функцией **AND\_OR**. Логические связи

между схемами также отсутствуют. Слои могут быть изготовлены каждый отдельно и соединены вместе только при сборке матрицы.

Такая матрица путем настройки превращается в заданную цифровую схему, причем только запись масок команд предполагает наличие некоторой заключительной технологической настройки, при которой выполняется выключение из работы тех или иных модуляторов в каждом из слоев. Однако нетрудно показать, что можно избежать выполнения процедуры маскирования в обоих слоях и проводить ее только в каком-то одном слое. Действительно, если убрать проверку совпадения переменной  $u_i$  с маской в функции  $AND\_OR$  и при построении второго слоя сделать клетки в столбцах с номерами  $5i + 3$  и  $5i + 4$ , изображающие модуляторы, черными, то можно получить второй вариант матрицы. В этой матрице выполняется маскирование только в первом слое тех же самых модуляторов, что и в первом варианте, и к ним добавляется маскирование фотодетекторов, расположенных над ранее маскировавшимися модуляторами во втором слое первого варианта матрицы. Совершенно аналогично можно получить матрицу с маскированием модуляторов и фотодетекторов только во втором слое. Существование таких вариантов настройки – свидетельство технологичности матрицы, потому что теперь один из слоев становится универсальным и не зависящим от того, какая цифровая схема отображается в матрицу. Такое свойство матрицы – прямое следствие ее двухслойной конструкции. Далее, для определенности, будем считать, что настройка выполняется только во втором слое.

Если исходная матрица строится как матрица с постоянной настройкой, то на последнем этапе модуляторы и фотоприемники можно просто исключить («выжечь»). Но если ввести во второй слой некоторое дополнительное оборудование, например соединить каждый модулятор и фотоприемник с управляющей ячейкой постоянной памяти с оптической записью и стиранием информации, то можно получить перепрограммируемую светом матрицу.

Итак, матрица собирается из двух слоев: один – неизменный, другой – настраиваемый на выполнение заданного набора команд подстановок.

Дополнительным достоинством матрицы является и то, что при ее физической реализации есть выбор: можно использовать как пассивные источники света – модуляторы, так и активные источники, например микролазеры с вертикальным излучением, опираясь на достаточно разработанную технологию smart-pixel arrays [11], которая способна обеспечивать обмен информацией между двумя поверхностями по оптическим каналам связи со скоростью более 1 Гбит/с с поверхности подложки в 1 см.<sup>2</sup>

2. *Высокие технические характеристики матрицы.* Действительно, при массовых вычислениях она превосходит цифровую схему, выполненную, например, в виде микроэлектронной СБИС, не только по технологичности, но и по меньшему расходу оборудования (аппаратурная сложность) и по большему быстродействию (временная сложность). Объясняется это тем, что в электрооптической матрице, как это видно на рис. 3, 7, одновременно преобразуется несколько комплектов входных данных. Следовательно, в ней может идти в конвейерном режиме имитация не одной цифровой схемы, а целого множества таких схем. Будем называть их виртуальными. Мощность этого множества равна числу пар регистров в первом слое матрицы.

Это означает, что аппаратурная сложность имитации виртуальной цифровой схемы равна суммарному количеству оборудования, заключенному в десяти столбцах клеток каждого из слоев матрицы. Примем сложность непосредственной реализации цифровой схемы, приведенную к отдельному вен-

тилю, за единицу. Пусть эта сложность при отображении цифровой схемы в матрицу увеличивается в  $k$  ( $k > 1$ ) раз и пусть в цифровой схеме содержится  $L$  вентилей, а в столбце шаблонов помещается порядка  $\sqrt{L}$ , тогда при  $L > k\sqrt{L}$  сложность имитации виртуальной цифровой схемы получается меньше сложности непосредственной реализации этой же схемы. Отметим, что ширина полос, составленных из отрезков электронных регистров сдвига и окаймляющих электрооптическую часть матрицы сверху и снизу, одна и та же для любых цифровых схем.

Относительно временной сложности отметим следующее: после входления конвейера в стационарный режим очередной выходной вектор получается через время, равное времени выполнения пяти шагов преобразования информации в слоях матрицы плюс время записи нового состояния в ячейку памяти регистра сдвига независимо от размеров и сложности имитируемой цифровой схемы. Рассматриваемая матрица является микроконвейером в общем случае с большим числом ступеней. Это значит, что время его входления в стационарный режим может быть достаточно большим. Поэтому, чтобы компенсировать это время, представляется целесообразным использовать предложенную матрицу в первую очередь для построения специпроцессоров, реализующих потоки однотипных преобразований массивов чисел (например, потоки коротких сортировок и малоточечных преобразований Фурье, матричные операции и т. п.).

Отметим, что полная совокупность команд, на выполнение которых могут быть настроены блоки матрицы, описывает 64 булевых функции с двумя входами и двумя выходами, что позволяет компактно отображать, практически не изменяя логический базис исходных цифровых схем, большинство логических схем, используемых в вычислительной технике. Отметим также, что матрицу можно применять не только для имитации цифровых схем, но и для непосредственной имитации реальных физических процессов, например диффузионных, в тех случаях, когда для этого используются клеточные автоматы с окрестностью Марголуса.

Возможность выполнения рассматриваемого устройства в виде двухслойной конструкции позволяет при реальном его изготовлении разместить логическую матрицу только на одной подложке (создать микрочип), в том числе и на одной ее стороне, с разделением функциональных плоскостей слоем диэлектрика толщиной не более 1 мкм. В этом случае размер окна модулятора, допускающего выполнение законов геометрической оптики при осуществлении проекции одной решетки на другую, может быть порядка 1 мкм и менее (это самый большой элемент микросхемы). Полагая, что число модуляторов света в одном блоке размером  $2 \times 2$  клетки равно 32, оценим размер такого блока: он равен 120 мкм<sup>2</sup>. Тогда матрица размером  $64 \times 64$  блока будет иметь площадь 0,5 мм<sup>2</sup>. Такая матрица может быть, например, настроена на выполнение функций 12-разрядного сумматора со схемой ускоренного переноса, что превратит ее в устройство, реализующее 64 таких виртуальных сумматора.

Из результатов моделирования видно, что первый слой состоит из логически, а значит, электрически не связанных друг с другом «вертикальных» регистров сдвига (электрические связи есть только между разрядами регистра), а второй – из логически изолированных друг от друга схем совпадения, не связанных электрически в любом направлении на поверхности. Поскольку устройство структурно однородно и в нем используется принцип локаль-

ности связей, то достаточно легко осуществить оптическуюстыковку двух таких микросхем по краевым элементам и таким образом наращивать размеры устройства в двух направлениях. Такаястыковка осуществляется за счет того, что краевые элементы (модуляторы и фотоприемники) располагаются таким образом, чтобы выход левой стороны модуляторов одного микрочипа находился над входом правой стороны фотоприемников другого микрочипа, верхняя сторона одного – над нижней стороной другого микрочипа и т. д.

**Заключение.** В данной работе описана методика проектирования двухслойной вычислительной матрицы и представлена процедура ее настройки на выполнение заданной функции. Матрица может выполнять большинство известных логических функций. Поэтому ее можно рассматривать в качестве универсального вычислительного устройства. Она может быть реализована в интегральном исполнении (в виде чипа) и размещена на одной подложке в двух плоскостях. Верхний и нижний слои матрицы представляют собой множество логически несвязанных однородных схем (клеток). Логические сигналы между слоями передаются по оптическим каналам связи. Такая конструкция чипа дает возможность итеративного наращивания их числа в двух направлениях. Путем компьютерного моделирования подтверждена возможность организации массовых конвейерных вычислений и получения высокой производительности: промежуток времени между очередными результатами на выходе матрицы равен шести тактам ее работы независимо от размеров и сложности имитируемой цифровой схемы.

#### СПИСОК ЛИТЕРАТУРЫ

1. Косцов Э. Г., Мишин А. И. Особенности построения оптических цифровых вычислительных машин // Микроэлектроника. 1977. 6. С. 139.
2. Egorov V. M., Kostsov E. G. Integral optical digital computers // Appl. Opt. 1990. 29(8). P. 1178.
3. Егоров В. М., Косцов Э. Г. Перспективы создания оптических цифровых высокопроизводительных вычислительных устройств // Автометрия. 1985. № 1. С. 114.
4. Пискунов С. В. Многослойная универсальная клеточная структура и ее электрооптическая реализация // Автометрия. 1995. № 3. С. 14.
5. Еvreinov Э. В., Прангишвили И. В. Цифровые автоматы с настраиваемой структурой. М.: Энергия, 1974.
6. Тоффоли Т., Марголус Н. Машины клеточных автоматов. М.: Мир, 1991.
7. Косцов Э. Г., Пискунов С. В. Трехмерные электрооптические матрицы для параллельной обработки информации // ЖНиПФ. 1998. 43, № 5. С. 26.
8. Beletkov D. T., Ostapkevich M. B., Piskunov S. V., Zhileev I. V. WinALT, a software tool for fine-grain algorithms and structures synthesis and simulation // Lecture Notes in Computer Sci. Berlin: Springer-Verlag, 1999. 1662. P. 491.
9. Анишев П. А., Ачасова С. М., Бандман О. Л. и др. Методы параллельного микропрограммирования. Новосибирск: Наука, 1981.
10. Achasova S. M., Bandman O. L., Markova V. P., Piskunov S. V. Parallel substitution algorithm. Theory and Application. Singapore: World Sci., 1994.
11. Sherif S. S., Griebel S. K., Au A. et al. Field-programmable smart-pixel arrays: design, VLSI implementation, and applications // Appl. Opt. 1999. 38, N 5. P. 838.

Институт автоматики и электрометрии СО РАН,  
E-mail: Kostsov@iae.nsk.su

Поступила в редакцию  
29 ноября 1999 г.