

Министерство образования Российской Федерации

КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
имени А.Н. ТУПОЛЕВА

В.А. РАЙХЛИН, И.С. ВЕРШИНИН, Е.Е. ГЛЕБОВ

**ПРАКТИКУМ
ПО СИНТЕЗУ ОПЕРАЦИОННЫХ
ЛОГИКО-ЗАПОМИНАЮЩИХ СРЕД**

Учебное пособие

Казань 2002

УДК 681.3 (07)

Райхлин В.А., Вершинин И.С., Глебов Е.Е.

Практикум по синтезу операционных логико-запоминающих сред.
Учебное пособие. Казань: Изд-во Казан. гос. техн. ун-та, 2002. 59 с.

ISBN

Изучаются вопросы организации и функционирования систолических структур поиска и распознавания. Рассматривается пример построения параллельного процессора - идентификатора. Каждое занятие включает элементы синтеза и модельный эксперимент.

Для студентов специальности 220100 - Вычислительные машины, комплексы, системы и сети.

Табл. 4. Ил. 49. Библиогр.: 11 наим.

Рецензенты: кафедра радиофизики Казанского государственного университета.

д. т. н., профессор В.А. Белавин

ISBN

© Изд-во Казан. гос. техн. ун-та, 2002.

© Райхлин В.А., Вершинин И.С.,

Глебов Е.Е., 2002.

СОДЕРЖАНИЕ

Предисловие	4
Введение. ПОДХОД К СИНТЕЗУ ОПЕРАЦИОННЫХ ЛОГИКО-ЗАПОМИНАЮЩИХ СРЕД	5
Занятие 1. ПЕРЕСТРАИВАЕМАЯ СРЕДА ЧИСЛОВОГО ПОИСКА	7
Синтез среды числового поиска	7
Моделирование 1	12
Занятие 2. СРЕДА ПОИСКА МИНИМАЛЬНОГО ПОКРЫТИЯ	18
Организация операционной матрицы	18
Моделирование 2	24
Занятие 3. СРЕДА ОДНОТАКТНОГО РАСПОЗНАВАНИЯ	30
Матрица распознавания	30
Моделирование 3	34
Занятие 4. МНОГОТАКТНОЕ РАСПОЗНАВАНИЕ	39
Необходимость	39
Параллельный алгоритм	40
Занятие 5. МАТРИЧНЫЙ СПЕЦПРОЦЕССОР - ИДЕНТИФИКАТОР	45
Структура спецпроцессора	45
Моделирование 4	49
Литература	60

ПРЕДИСЛОВИЕ

Задачей данного практикума является освоение студентами принципов разработки специализированных процессорных матриц на двух уровнях иерархии представления параллельной системы: схемо- и системотехническом.

Основное внимание в пособии уделяется синтезу операционных логико-запоминающих сред различного функционального назначения как одной из разновидностей систолических структур. Системотехнический уровень представлен примером реализации на основе таких сред обрабатывающей части специализированного матричного процессора-идентификатора. В совокупности это создает достаточно цельное представление об организации процессорных матриц.

Необходимость постановки практикума на позиции моделирования объясняется трудностями создания в вузе действующих полигонов для изучения промышленных образцов параллельных процессоров. Использование программных моделей является единственно возможным выходом. В разработке одной из них (подсистемы поиска минимального покрытия) принимал участие И.Р. Саидгараев. Авторы благодарны ему за помощь.

Авторы

Введение

Подход к синтезу операционных логико-запоминающих сред

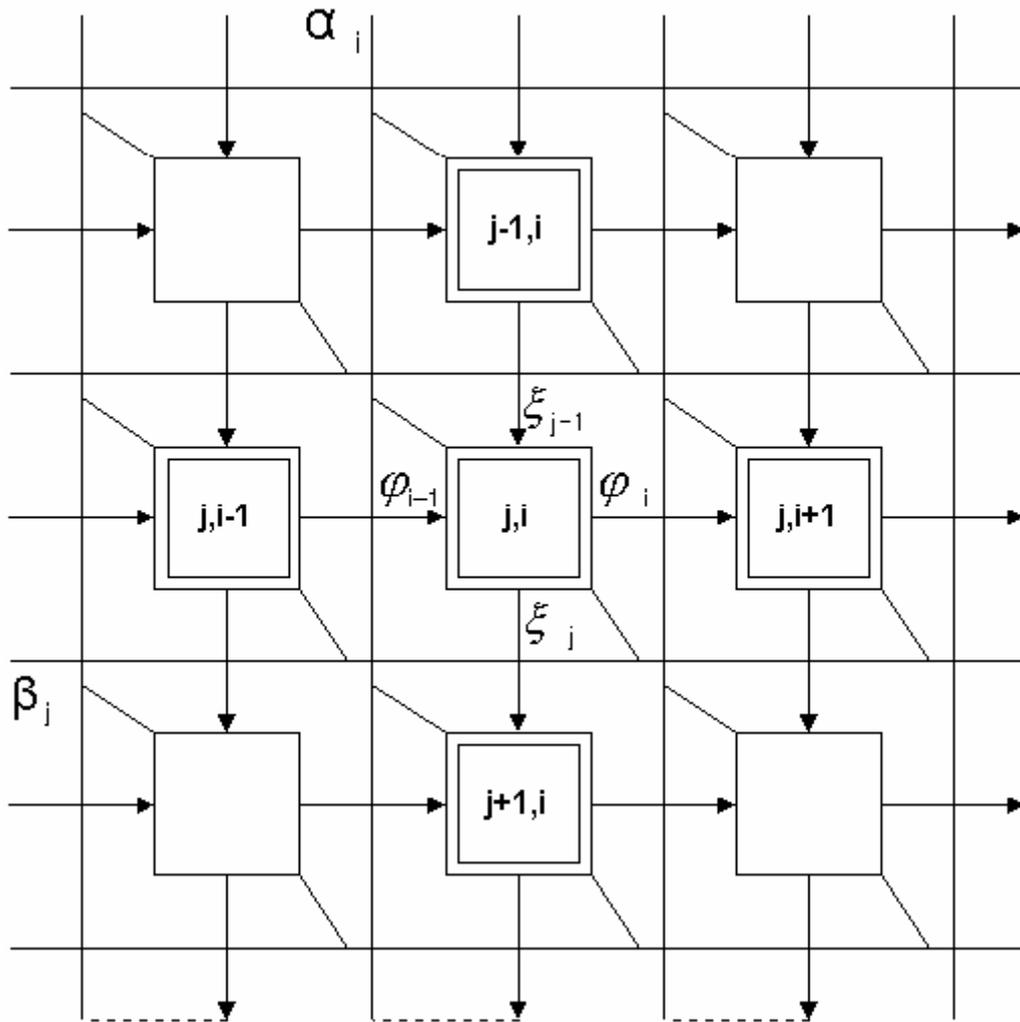


Рис. 1.1

Определим операционную ЛЗС как итеративную двумерную структуру (см. рис 1.1), которая реализует заданное множество процедур независимо от размеров среды ($i=1,2, \dots, m$; $j=1,2, \dots, n$). Элемент (j, i) среды содержит ячейку памяти для хранения 1 бита исходной информации и автоматную часть. Признаки α_i, β_j являются общими для i -го столбца и j -ой строки. Они могут выполнять функции настройки, маскирования, разрешения считывания или записи, числовых разрядов, результатов анализа содержимого строк или столбцов и т.д.

В общем случае внешние сигналы элемента суть многокомпонентные наборы. Например: $\alpha_i = (\alpha_i^1, \alpha_i^2, \dots, \alpha_i^k)$. Выходы элемента (j, i) являются

автоматными функциями входов и содержимого $a_{j,i}$ ячейки памяти. Элементы могут иметь и отдельные выходы. Допускается коммутация шин по краям среды (рис. 1.1 пунктир). "Обрамление" среды составляют группы регистров, в которые помещаются исходные данные (признаки), маски и конечные результаты. В них может выполняться и дополнительная логическая обработка информации.

Суть излагаемого подхода к синтезу операционных логико-запоминающих сред заключается в следующем. В конечном итоге задача синтеза ЛЗС для реализации заданного набора процедур сводится к определению структуры связей, схемы элемента и значений сигналов на границах среды. Основное затруднение связано с тем, что неизвестна стандартная рецептура формулировки алгоритмов выполнения процедур на основе ЛЗС. В этом смысле любой подход к синтезу ЛЗС в целом эвристичен. Но если алгоритм приемлемо сформулирован и связи определены, то дальнейший процесс синтеза хорошо формализуется с помощью метода модифицированных таблиц переходов, развитого первоначально применительно к одномерным итеративным системам.

По определению в ЛЗС выполняется последовательно-однотипная обработка элементов информации по строкам и (или) столбцам. Подобную интерпретацию допускают операции поиска числа в однородном информационном массиве по заданному набору признаков, распознавания каких-либо свойств такого массива и преобразования кодов (суммирования, сдвигов и т.д.). Поэтому достаточным условием реализуемости произвольной процедуры на основе ЛЗС является возможность формулировки алгоритма ее выполнения в терминах выделенного класса операций.

Пусть такое условие выполнено. Тогда синтезируется своя среда для каждой из непересекающихся алгоритмических компонент (операций) путем интерпретации этой компоненты применительно к использованию ЛЗС совместно с методом модифицированных таблиц переходов. После этого либо организуется единая перестраиваемая среда объединением всех составляющих,

либо создается обрабатывающий массив из нескольких специализированных сред.

На первых двух занятиях дается иллюстрация сформулированного подхода на примерах структур информационного поиска и минимизации булевых функций. Заметим, что метод модифицированных таблиц переходов позволяет описать поведение как комбинационных, так и последовательностных итеративных систем. В последнем случае на этапе реализации элемента по такой таблице в цепь обратной связи необходимо включать синхронные D – триггеры (элементы задержки на такт) по одному на каждую внутреннюю переменную. Тем самым достаточно просто решается и проблема начальной установки автомата. Граничный набор действует постоянно. Такты разделяются подачей синхроимпульсов на D – триггер.

Занятие 1

Перестраиваемая среда числового поиска

Синтез среды числового поиска

При структурной обработке информационных массивов наиболее часто встречаются задачи упорядочения элементов информации (сортировка), поиска по заданному признаку или совокупности признаков (ассоциативный поиск), выделения всех элементов, больших (меньших) заданного признака или расположенных в определенных границах и т. д. Их аппаратная реализация на основе ЛЗС может дать резкое повышение быстродействия по сравнению с программными методами.

Рассмотрим синтез перестраиваемой среды, которая реализует поиск минимальных (максимальных) чисел информационного массива, равных заданному, больших или меньших заданного, ближайшего большего (меньшего) по отношению к заданному. Поиск максимума равносителен поиску минимума среди обратных кодов чисел массива. Ближайшее большее - минимальный элемент среди всех чисел, превышающих заданное. Поэтому искомая среда является суперпозицией двух ЛЗС. Одна из них делит массив на

три подмножества чисел - равных, больших и меньших заданного. Другая отыскивает минимальный элемент массива.

Деление массива на три части выполняется по следующему алгоритму. Поиск ведется слева направо (со старших разрядов). Если на предыдущей итерации строка отнесена к подмножествам ">" или "<", то обработка i -го бита не вносит изменений в результат. Если ранее строка была включена в подмножество "=", то результат i -ой итерации определяется соотношением между α_i^1 и $a_{j,i}$ (между i -ми разрядами признака и числа в j -й строке).

Боковые входы элемента (рис. 1.2)

$$\varphi_{i-1} \in \{S_0, S_1, S_2\}.$$

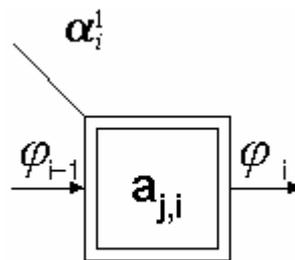


Рис. 1.2

Наборы S_0, S_1, S_2 отвечают случаям, когда левая по отношению к i -му разряду часть числа равна, больше или меньше левой части признака. Правый выход элемента определен модифицированной таблицей переходов (рис. 1.3).

$\varphi_{i-1} \backslash \alpha_i^1 a_{j,i}$	00	01	11	10	φ_i
S_0	S_0	S_1	S_0	S_2	
S_1	S_1	S_1	S_1	S_1	
S_2	S_2	S_2	S_2	S_2	

Рис. 1.3

Выполняя кодировку состояний (рис. 1.4), получаем кодированную таблицу (рис. 1.5). Из этой таблицы:

$$\varphi_i^1 = \varphi_{i-1}^1 \vee \overline{\alpha_i^1 a_{j,i}} \overline{\varphi_{i-1}^2}, \quad \varphi_i^2 = \varphi_{i-1}^2 \vee \alpha_i^1 \overline{a_{j,i}} \overline{\varphi_{i-1}^1}.$$

Граничные сигналы $\varphi_0^1 = \varphi_0^2 = 0$.

		φ^1	
		0	1
φ^2	0	s_0	s_1
	1	s_2	—

Рис. 1.4

		$\alpha_i^1 a_{j,i}$				$\varphi_i^1 \varphi_i^2$
		00	01	11	10	
$(\varphi^1 \varphi^2)_{i-1}$	00	00	10	00	01	
	01	01	01	01	01	
	11	<u>11</u>	<u>11</u>	<u>11</u>	<u>11</u>	
	10	10	10	10	10	

Рис. 1.5

Сформулируем алгоритм поиска минимума. Как и прежде, поиск ведется со старших разрядов. При анализе i -го бита j -я строка оставляется во множестве минимальных чисел, если $a_{j,i} = 0$. В случае $a_{j,i} = 1$ эта строка исключается из рассмотрения только тогда, когда для выделенного на $(i-1)$ -й итерации множества не все $a_{k,i}$ равны 1 ($1 \leq k \leq n$).

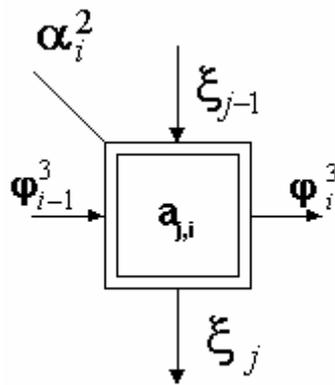


Рис. 1.6

Входы элемента (рис. 1.6) таковы. Сигнал $\varphi_{i-1}^3 = 1$, если левая часть числа принадлежит множеству минимальных левых частей массива. Иначе $\varphi_{i-1}^3 = 0$. Сигнал $\xi_{j-1} = 1$, если на подмножестве минимальных левых частей сверху от элемента (j,i) все $a_{k,i} = 1$ ($k < j$). Иначе $\xi_{j-1} = 0$. Сигнал $\alpha_i^2 = \xi_n$.

		$a_{j,i}$		φ_i^3
		0	1	
$\alpha_i^2 \varphi_{i-1}^3$	00	0	0	
	01	1	0	
	11	1	1	
	10	0	0	

Рис. 1.7

		$a_{j,i}$		ξ_j
		0	1	
$\xi_{j-1} \varphi_{i-1}^3$	00	0	0	
	01	0	0	
	11	0	1	
	10	1	1	

Рис. 1.8

Модифицированные таблицы переходов для правого и нижнего выходов (рис. 1.7, 1.8) дают:

$$\varphi_i^3 = \overline{a_{j,i}} \varphi_{i-1}^3 \vee \alpha_i^2 \varphi_{i-1}^3; \quad \xi_j = a_{j,i} \xi_{j-1} \vee \xi_{j-1} \overline{\varphi_{i-1}^3}$$

Граничные сигналы: $\varphi_0^3 = \xi_0 = 1$

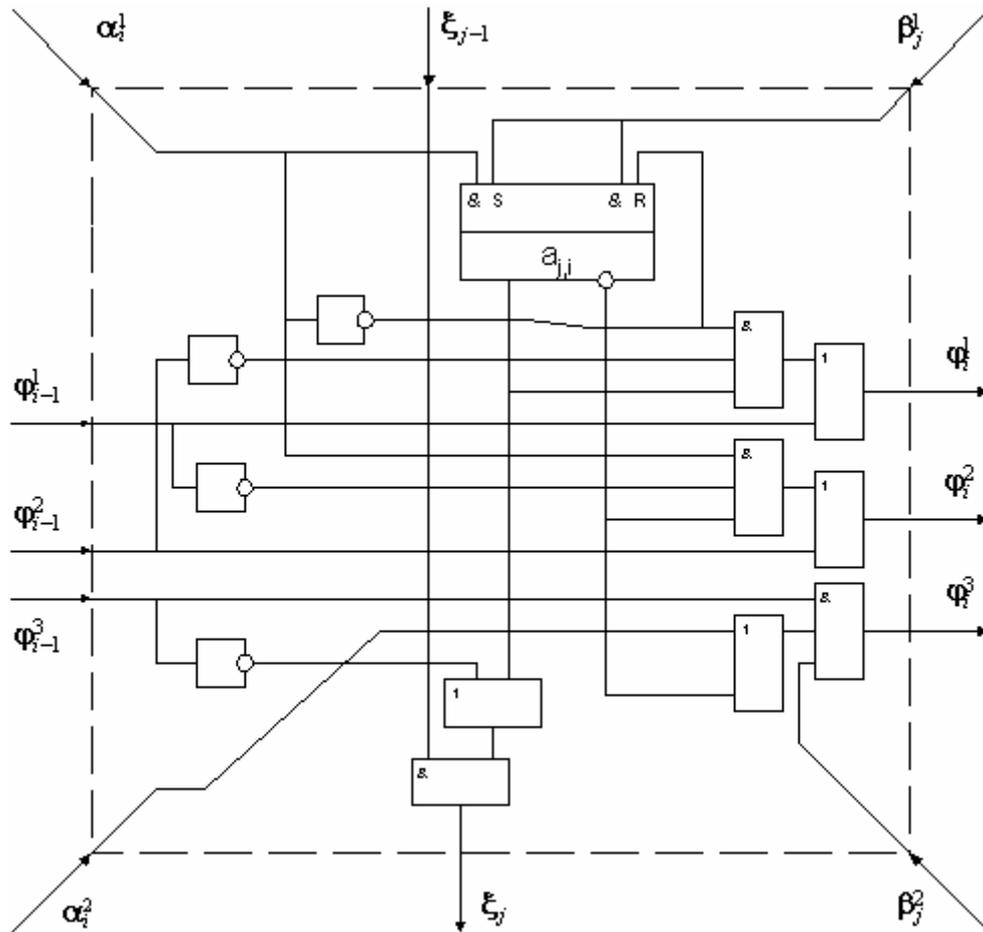


Рис. 1.9

Объединяя найденные решения, получаем схему элемента искомой среды (рис. 1.9). Значение выхода

$$\varphi_i^3 = (\overline{a_{j,i}} \vee \alpha_i^2) \varphi_{i-1}^3 \beta_j^2.$$

Сигнал $\beta_j^2 = \varphi_m^1$, если ведется поиск ближайшего большего, и $\beta_j^2 = 1$ при поиске минимума. В обоих этих случаях $\alpha_i^2 = \xi_n$. Для других видов процедур значения β_j^2 и α_i^2 безразличны.

По сигналу разрешения записи $\beta_j^1 = 1$ число $\alpha^1 = (\alpha_1^1 \dots \alpha_m^1)$ заносится поразрядно в j -ю строку. Считывание информации из l -й строки реализуется такой настройкой:

$\xi_{0,j} = 1, \varphi_{l,0}^3 = 1, \varphi_{j,0}^3 = 0$ ($j \neq l$). При этом $\xi_{n,i} = a_{l,i}$.

Так что специального сигнала разрешения считывания не требуется.

Матрица $n_1 \times m_1$ из таких элементов имеет $M = 8n_1 + 4m_1$ внешних выводов. Дополнительно необходимо учесть два вывода подключения шин "земля" и "питание". Поэтому в корпусе БИС на 40 выводов размещается до 10 элементов синтезированной среды (матрица 2×5).

Моделирование 1

Для начала моделирования необходимо запустить файл `lzs.exe`.

Моделирование состоит из двух частей:

- Демонстрационная часть
- Основная часть

В демонстрационной части показывается процесс реализации поиска на уровне элементов ЛЗС с иллюстрацией прохождения сигналов во времени через элементы среды. Входы и выходы элементов окрашиваются в синий или красный цвет, что соответствует логическим 0 или 1. Черный цвет выхода элемента означает, что данный выход еще не принял устойчивое состояние. После того, как все входы и выходы элемента определены, он изменяет свой цвет.

В основной (тестовой) части моделирование работы структуры происходит без показа протекающих в ней процессов. Программа тестирует моделируемую структуру на правильность функционирования, давая возможность студенту подтвердить на практике усвоенный ранее теоретический материал. При необходимости в процессе работы может быть вызвана нужная справка.

Общий вид окна после запуска программы представлен на рис. 1.10. С помощью меню или кнопок в панели управления можно выбрать режим тестирования или запустить демо-версию программы. Предусмотрен вызов справки и выход из программы.

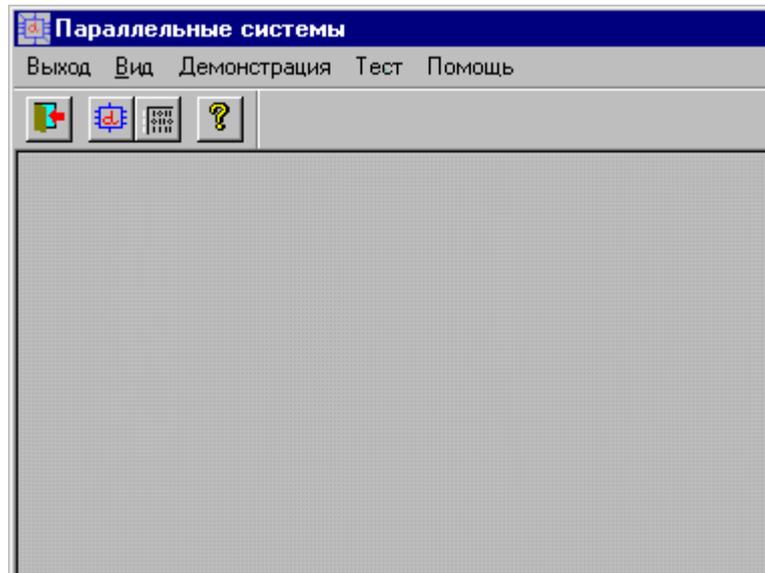


Рис. 1.10

Имеются следующие пункты меню (и соответствующие им кнопки в панели управления):

 <i>Выход</i>	Выход из программы
 <i>Демонстрация</i>	Запуск демонстрационной модели ЛЗС
 <i>Тест</i>	Формирование и тестирование модели ЛЗС, созданной студентом
 <i>Помощь</i>	Вывод на экран справки

Для запуска демонстрационной модели необходимо выбрать в меню пункт “Демонстрация” или нажать соответствующую кнопку в панели управления. После нажатия на экране появится окно демонстрации (рис.1.11). После ознакомления с демонстрационной моделью необходимо перейти к работе с тестовой версией программы.

Демонстрационная часть

Общий вид окна после запуска демонстрационной программы представлен на рис.1.11. На рисунке изображено поле элементов среды размером (4×8), окрашенных первоначально в черный цвет (до запуска демонстрации). Внутри

каждого элемента цифрой (0 или 1) обозначено его состояние. Ниже располагается ряд кнопок управления демонстрацией.

Для запуска демонстрации необходимо нажать одну из кнопок: “Поиск минимума” или “Поиск ближайшего большего”. Во время демонстрации происходит изменение цвета входов и выходов элементов, а также цвета самого элемента.

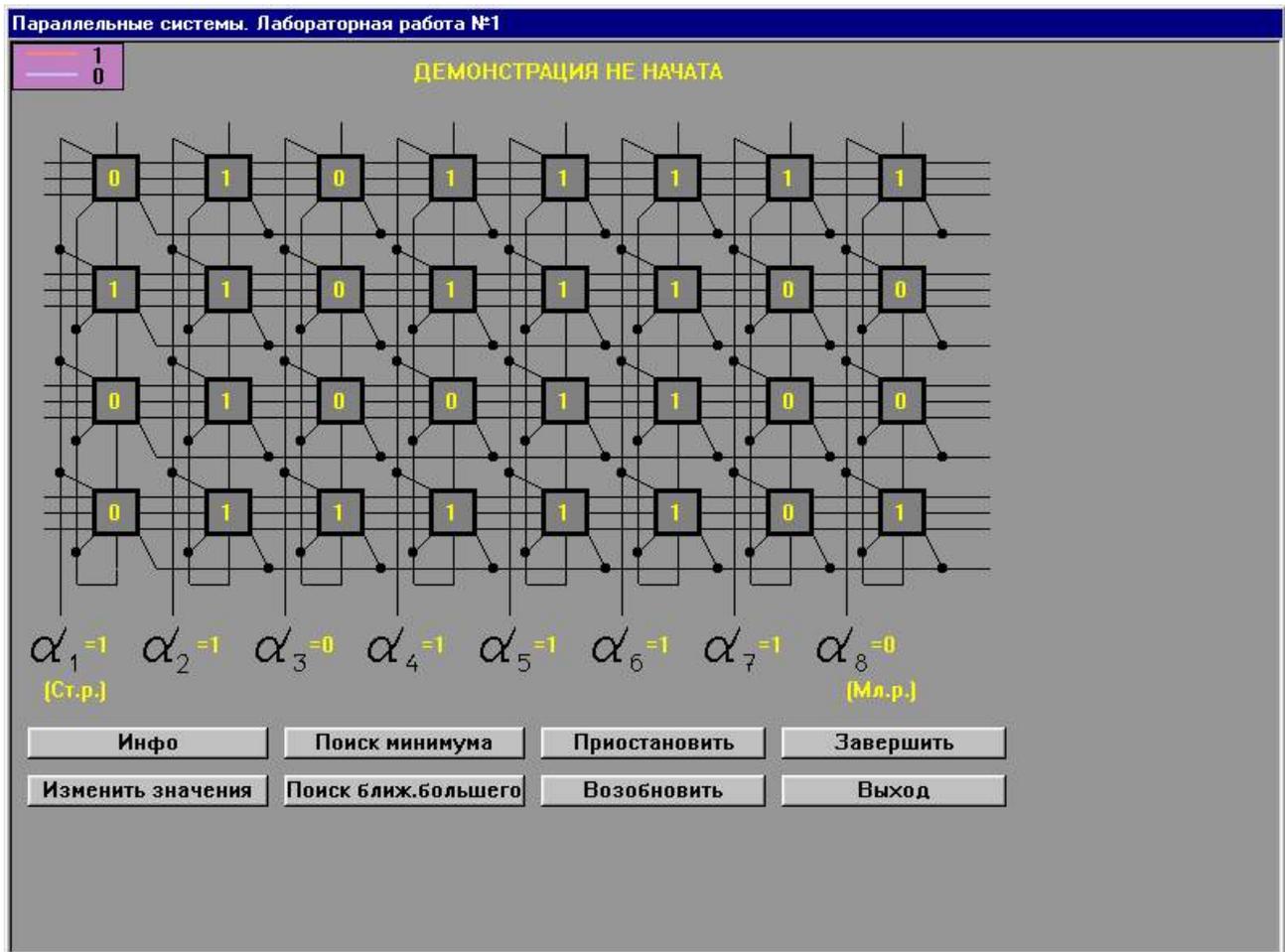


Рис. 1.11

Условное обозначение проводников при протекании по ним тока показано в левом верхнем углу. В верхней части окна отображается текущее состояние показа демонстрации. Над кнопками расположены разряды признака α (слева направо). В нижней части окна располагается ряд кнопок, с помощью которых осуществляется управление программой:

<i>Инфо</i>	Вызов справки
<i>Изменить значения</i>	Изменить состояния элементов среды
<i>Поиск минимума</i>	Запуск демонстрации поиска минимума
<i>Поиск ближ.большого</i>	Запуск демонстрации поиска ближайшего большого
<i>Приостановить</i>	Приостановка демонстрации на время
<i>Возобновить</i>	Возобновление демонстрации после приостановки
<i>Завершить</i>	Закончить показ демонстрации
<i>Выход</i>	Выход из программы

Требуется, выбирая поиск минимума или поиск ближайшего большего для различных состояний элементов среды, проверить правильность выполнения и уяснить принцип работы данной ЛЗС.

Основная (тестовая) часть

Для запуска процесса тестирования надо выбрать в меню рис. 1.10 пункт “Тест” или нажать соответствующую кнопку в панели управления. Появится запрос на размеры тестируемой матрицы (рис.1.12), где n - число строк матрицы (от 1 до 32), m - число столбцов матрицы (от 1 до 64).

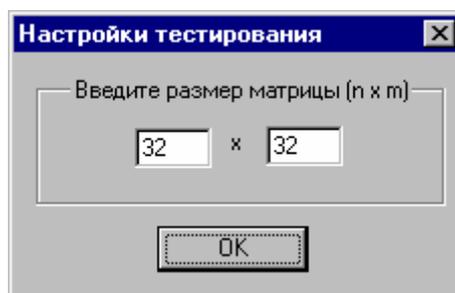


Рис. 1.12

В случае ввода недопустимого значения m или n будет выведено соответствующее предупреждение.

Далее появится запрос о способе ввода данных в матрицу (рис. 1.13).

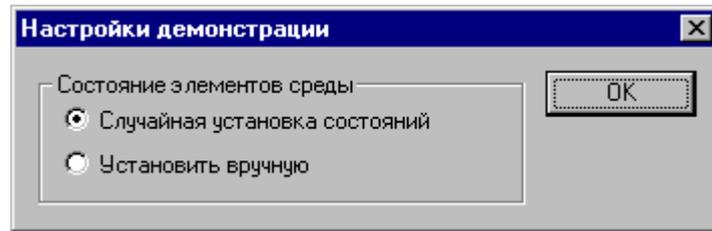


Рис. 1.13

Состояние элементов матрицы будет выбираться случайным образом или вводиться пользователем вручную.

Затем появится окно для ввода признака (рис. 1.14).

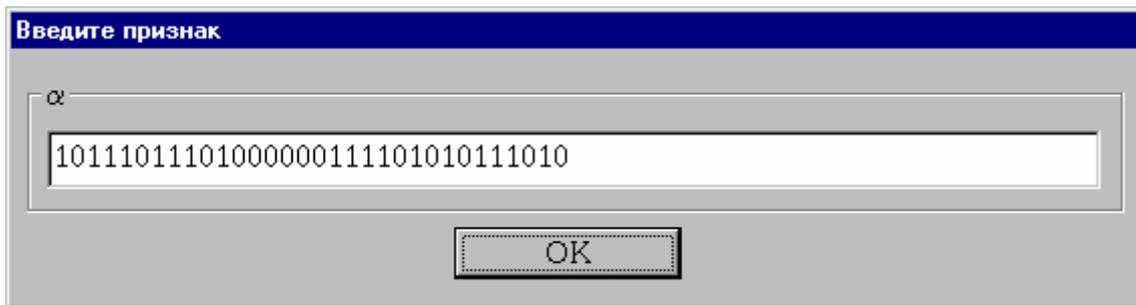


Рис. 1.14

Длина вектора-признака не должна превышать заданного количества столбцов. Ввод осуществляется со старших разрядов. Если вводится вектор меньшей длины, то недостающие младшие разряды заполняются нулями автоматически. Если введенное число окажется не двоичным, то будет дано соответствующее предупреждение.

Наконец, появится окно установки граничных сигналов (рис. 1.15). От правильности их установки зависит достоверность результатов тестирования.



Рис. 1.15

Если был выбран режим ручного ввода данных, то на экране появятся границы матрицы в соответствии с введенными значениями n и m и курсор, показывающий текущее место ввода данных. Ввод данных осуществляется нажатием клавиш “0” или “1” (рис. 1.16).

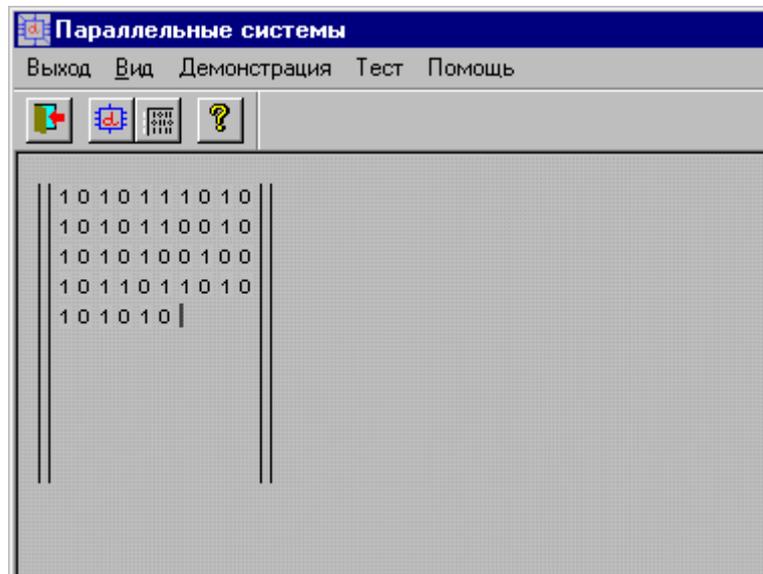


Рис. 1.16

Режим ввода данных можно прервать, нажав клавишу “Esc”. Нажатие любых других клавиш игнорируется и выдается предупреждающий звуковой сигнал.

Если был выбран режим случайного ввода данных, то они генерируются случайным образом и выдаются на экран.

По окончании ввода запускается алгоритм поиска минимума и ближайшего большего по отношению к признаку. Результаты тестирования выводятся на экран, (рис.1.17).

Строка, выделенная зеленым цветом, отвечает ближайшему большему числу массива по отношению к признаку. Строка, выделенная красным цветом, определяет минимальное значение среди всех строк матрицы.

В случае, если граничные сигналы были выставлены неправильно, выдается сообщение об ошибке (рис. 1.18).

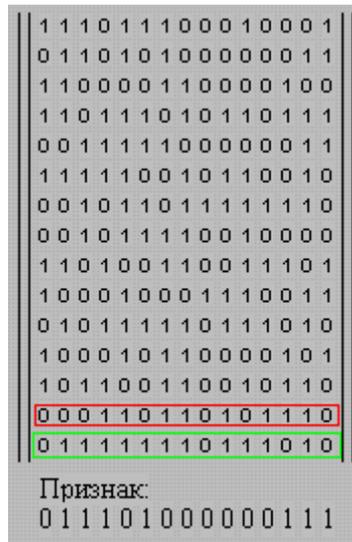


Рис. 1.17

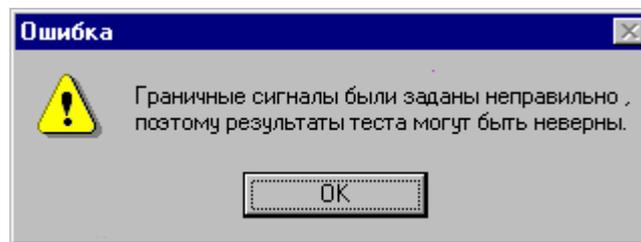


Рис. 1.18

Требуется несколько раз запустить процесс тестирования, задавая всякий раз граничные сигналы, и проверить правильность выполнения теста.

Занятие 2

Среда поиска минимального покрытия

Организация операционной матрицы

Аппаратное решение этой задачи может быть использовано не только для ускорения процесса минимизации булевых функций, но и при поиске минимального набора тестов для обнаружения неисправностей логических схем и диагностических тестов. Аналогичные задачи возникают при оптимальном кодировании микроопераций, при минимизации и кодировании автоматных таблиц.

Исходной информацией является импликантная матрица некоторой булевой функции. Строки матрицы отмечаются элементами множества импликант для этой функции. Столбцам соответствуют элементы множества M^1 наборов переменных, на которых значение функции равно 1. На пересечении строки и столбца ставится единица, если данный импликант покрывает соответствующий набор. Задача состоит в определении минимального подмножества импликант, которые в совокупности покрывают M^1 .

Приближенно эту задачу можно решить следующим образом. В исходной матрице находится столбец с минимальным числом единиц. Если таких столбцов несколько, берется любой. Из строк, которые отмечены единицами найденного столбца, выделяется строка с максимальным числом единиц. Если таких строк несколько, берется любая. Соответствующий ей импликант вводится в решение. Столбцы, отмеченные единицами этой строки, исключаются из дальнейшего рассмотрения. Процесс повторяется до тех пор, пока не будут исключены все столбцы.

Предусмотрим в исходной матрице X_a маскирование разрядов по горизонтали (маска $MX = \|m_i\|$) и вертикали (маска $MY = \|m_j\|$). Тогда применительно к использованию ЛЗС возможна такая формулировка алгоритма поиска минимального покрытия.

1. $MX := \|1\|$ (единичная матрица-строка).
2. Поиск столбца с минимальным количеством единиц. Если результат неоднозначен, берется первый слева. Пусть это будет столбец X_a^i .
3. Формирование вертикальной маски, $MY := X_a^i$.
4. Поиск строки с максимальным числом единиц. Если результат неоднозначен, берется первая сверху. Пусть это будет строка X_a^j .
5. Код (отмечающая импликанта) строки X_a^j вводится в решение.

6. Формирование горизонтальной маски, $MX := MX \& \overline{X_a^j}$ (операции выполняются поразрядно).

7. Если $MX \neq \|0\|$, идти к п.2. Иначе КОНЕЦ.

Реализация всех пунктов алгоритма, исключая 2 и 4, вполне очевидна. Для отыскания столбца с минимальным числом единиц будем зачеркивать в каждом такте по одной единице во всех незамаскированных столбцах. Такому подходу отвечает элемент, показанный на рис. 2.1.

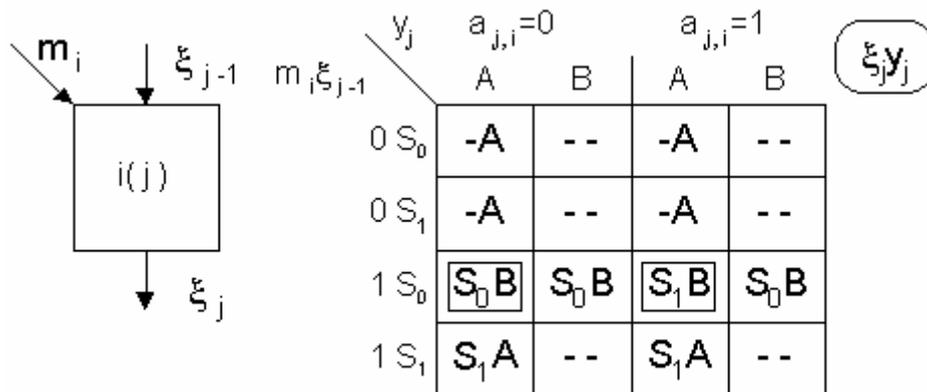


Рис. 2.1

Для однозначной индикации результата в каждом столбце применяем дополнительный $(n+1)$ -й элемент с безразличным значением $a_{n+1,i}$. В искомом столбце этот элемент "вычеркнут" (переведен в состояние В) раньше, чем в других столбцах. Одновременно прекращается поступление синхроимпульсов. Строка дополнительных элементов образует обрамление среды снизу (рис. 2.2).

Для реализации приоритетного опроса этой строки слева сигналом T_0 элементу $(n+1,i)$ придается комбинационная часть (рис. 2.3).

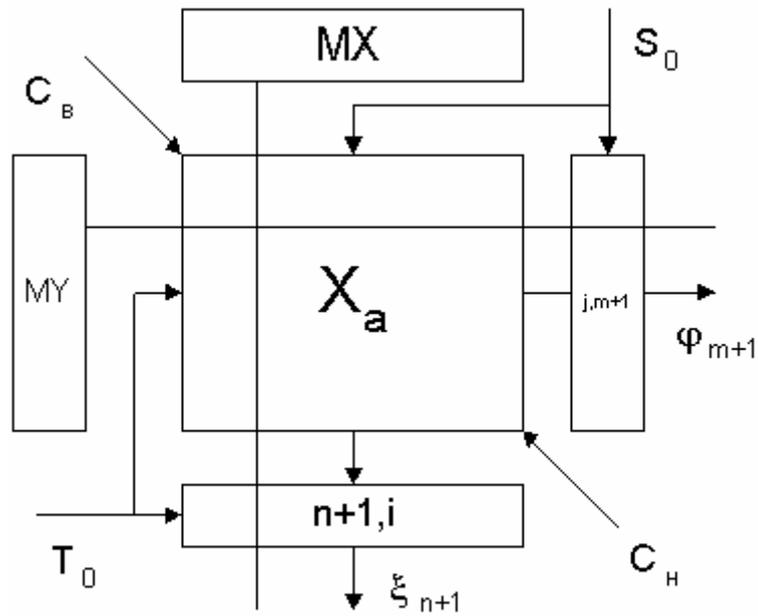


Рис. 2.2

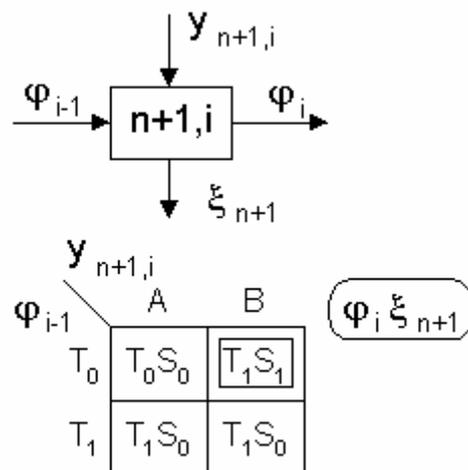


Рис. 2.3

Сигнал $\xi_{n+1}=S_1$ появляется на выходе первого слева "вычеркнутого" дополнительного элемента.

Поиск строки с максимальным числом единиц выполняется аналогично. Но теперь в каждом такте вычеркиваем по одному нулю во всех незамаскированных строках (рис. 2.4).

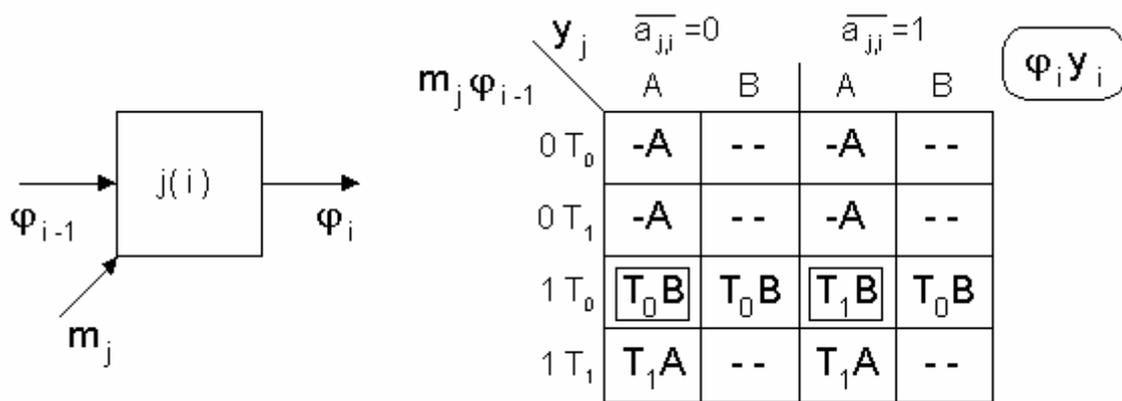


Рис. 2.4

Поступление синхроимпульсов прекращается, как только дополнительный (m+1)-й элемент хотя бы одной из строк "вычеркнут". Столбец дополнительных элементов образует обрамление среды справа (рис. 2.2). Приоритетный опрос этого столбца сверху сигналом S_0 реализует специальная комбинационная приставка (рис. 2.5) к элементу (j,m+1). Выделяется строка, на выходе которой сигнал $\phi_{m+1} = T_1$.

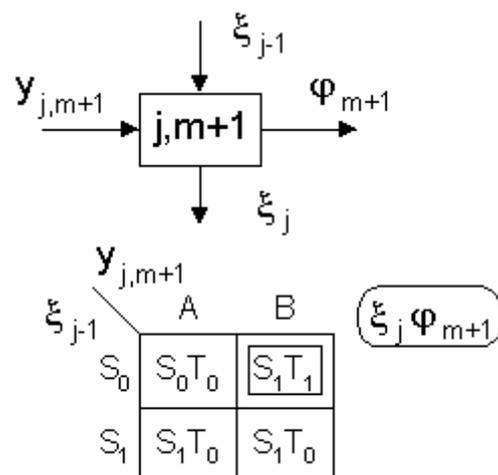


Рис. 2.5

Сигналы S_0 и T_0 по краям среды действуют постоянно. Первоначально все элементы установлены в состояние А. Матрица X_a содержит два слоя элементов ("верхний" и "нижний") со своими синхросигналами (C_B и C_H).

Наличие D-триггера в дополнительном элементе (n+1,i) необязательно, поскольку правильное значение $y_{n+1,i} = m_i \overline{\xi_n}$ получается сразу по вычеркивании нижней единицы в столбце (наличие последующих невычеркнутых нулей оставляет значение $\xi_j = \xi_{j-1} = 0$). С учетом таблицы на рис. 2.3 имеем схему (рис. 2.8), где

$$\varphi_i = \varphi_{i-1} \vee y_{n+1,i}, \quad \xi_{n+1} = y_{n+1,i} \overline{\varphi_{i-1}}.$$

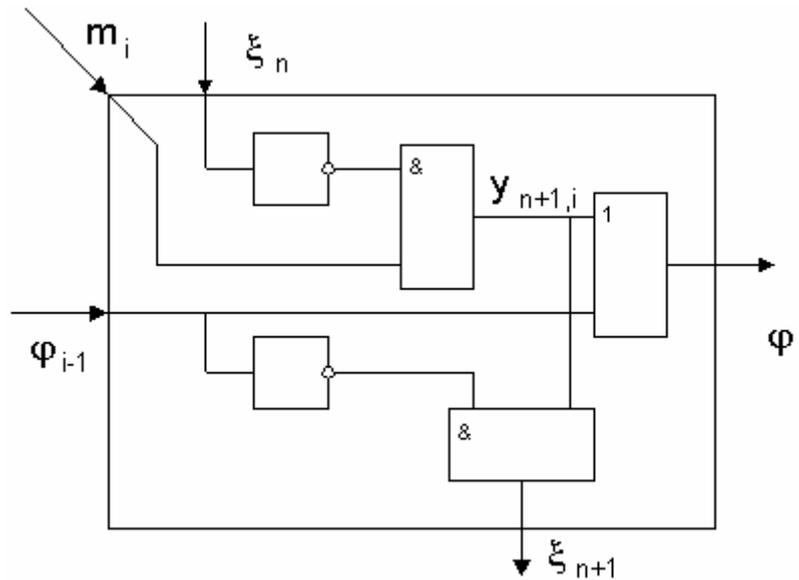


Рис. 2.8

Элемент (j,m+1) аналогичен.

Моделирование 2

Для начала моделирования необходимо запустить файл lzs2.exe.

Моделирование состоит из двух частей:

- Демонстрационная часть
- Основная часть.

В демонстрационной части показывается процесс реализации поиска на уровне элементов ЛЗС с иллюстрацией прохождения сигналов во времени через элементы среды.

В основной (тестовой) части моделирование работы структуры происходит без показа протекающих в ней процессов. Программа тестирует моделируемую структуру на правильность функционирования, давая возможность студенту подтвердить на практике усвоенный ранее теоретический материал. При необходимости в процессе работы может быть вызвана нужная справка.

Общий вид окна после запуска программы представлен на рис. 2.9. С помощью меню или кнопок в панели управления можно выбрать режим тестирования или запустить демо-версию программы. Предусмотрен вызов справки и выход из программы.

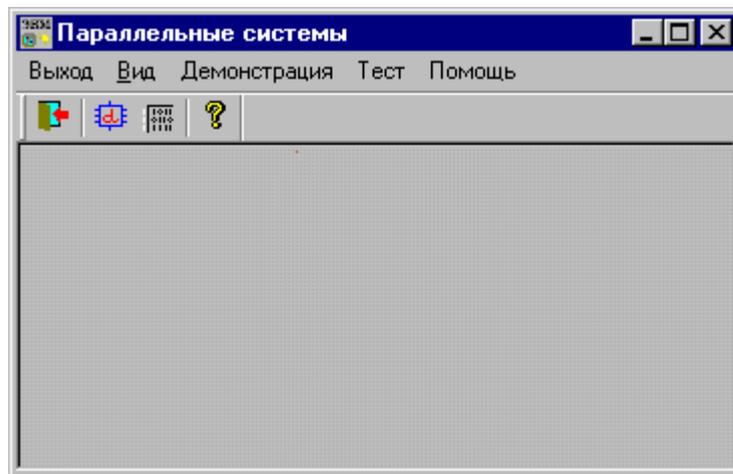


Рис. 2.9

Имеются следующие пункты меню (и соответствующие им кнопки в панели управления):

- | | | |
|---|---------------------|--|
|  | <i>Выход</i> | Выход из программы |
|  | <i>Демонстрация</i> | Запуск демонстрационной модели ЛЗС |
|  | <i>Тест</i> | Формирование и тестирование модели ЛЗС, созданной студентом. |
|  | <i>Помощь</i> | Вывод на экран справки. |

Для запуска демонстрационной модели необходимо выбрать в меню пункт “Демонстрация” или нажать соответствующую кнопку в панели управления.

После нажатия на экране появится окно демонстрации (см. рис 2.10). После ознакомления с демонстрационной моделью нужно перейти к работе с тестовой версией программы.

Демонстрационная часть

Общий вид окна после запуска демонстрационной программы представлен на рис. 2.10. В центре условно обозначено поле логических элементов ЛЭС размером (5×7). Основные элементы среды схемно представлены парой квадратов, связи - линиями, разряды маскирующих векторов - окружностями. Места соединений линий связи и выводов обозначены узлами как на обычных чертежах. Внутренние состояния триггеров каждого элемента обозначаются буквенной информацией (“А” - соответствует логическому ‘0’, “В” - соответствует логической ‘1’), а логические значения выводов - графической (логическим значениям сопоставлены определенные цвета). Дополнительные элементы обозначены квадратами с диагональной сеткой внутри.

Во время демонстрации происходят изменения цветов линий, соединяющих элементы, что означает соответствующие изменения логических уровней протекающих по ним токов. Черный цвет выхода элемента означает, что данный выход еще не принял устойчивого состояния. Два других возможных цвета (красный и синий) устанавливаются через определенное время (моделируется задержка формирования выходов внутри элемента), после того как соответствующие входы примут стационарные значения. В случае, когда с приходом тактового импульса изменяется внутреннее состояние основного элемента, выход окрашивается в черный цвет. Это происходит в связи с тем, что для основного элемента выход является функцией входов и состояния. Синий цвет выхода соответствует логическому “0”, красный - логической “1”.

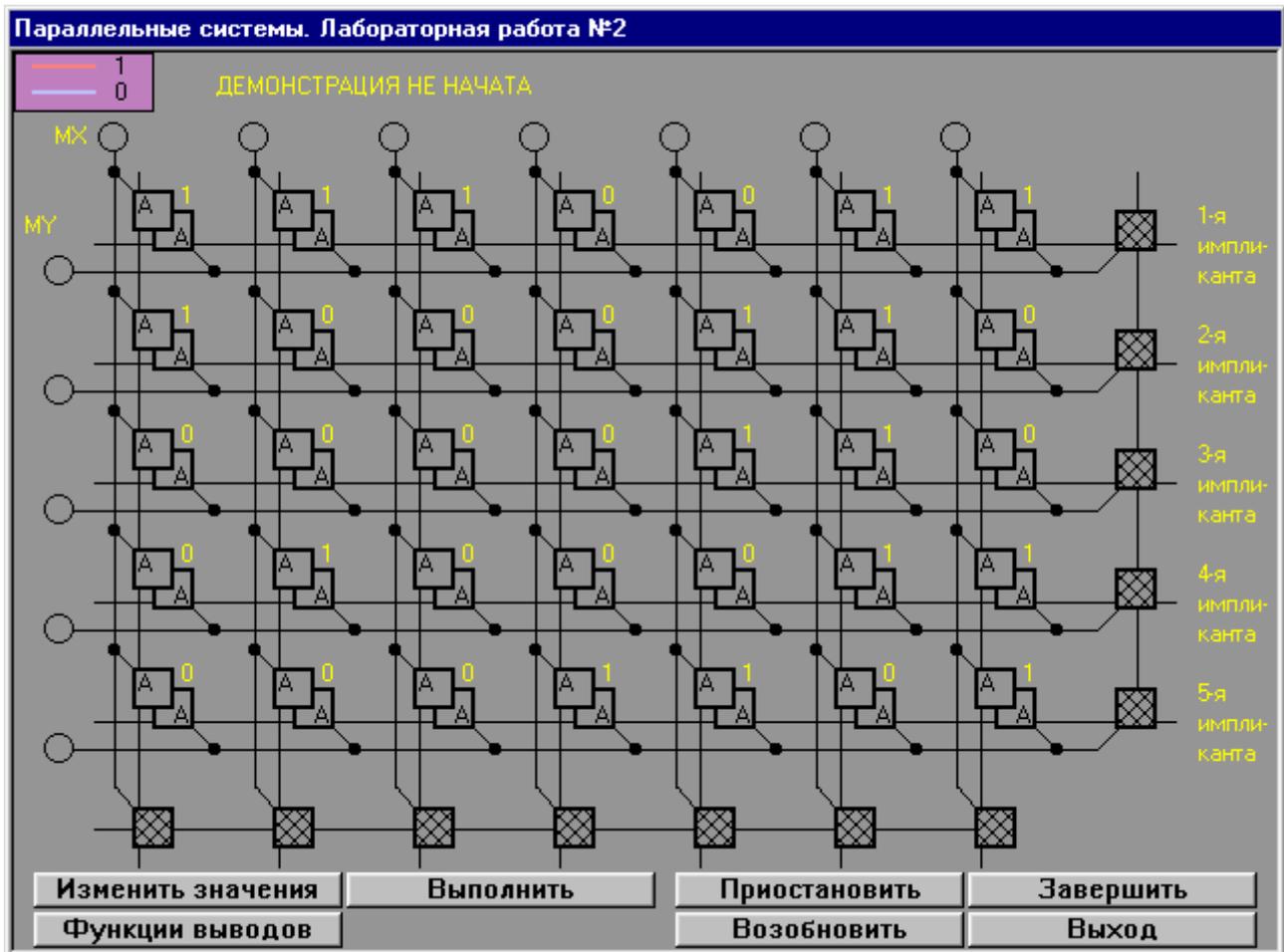


Рис. 2.10

В верхней части окна отображается текущее состояние показа демонстрации. В нижней части окна располагается ряд кнопок, с помощью которых осуществляется управление программой:

Функции выводов

Изменить значения

Выполнить

Приостановить

Возобновить

Завершить

Выход

Вызов справки

Изменить состояния элементов среды

Запуск демонстрации

Приостановка демонстрации

Возобновление демонстрации после
приостановки

Закончить показ демонстрации

Выход из программы

Требуется, запуская демонстрацию для различных состояний элементов среды, проверить правильность выполнения и уяснить принцип работы данной ЛЗС.

Основная (тестовая) часть

Для запуска процесса тестирования надо выбрать в меню рис. 2.9 пункт “Тест” или нажать соответствующую кнопку в панели управления.

Появится запрос на размеры тестируемой матрицы (рис. 2.11) m и n , где n - число строк матрицы (от 1 до 32), m - число столбцов матрицы (от 1 до 64).

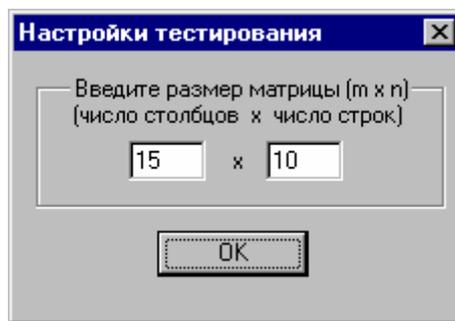


Рис. 2.11

В случае ввода недопустимого значения m или n будет выведено соответствующее предупреждение.

Далее появится запрос о способе ввода данных в матрицу (рис. 2.12): вручную или автоматически со случайным заполнением.

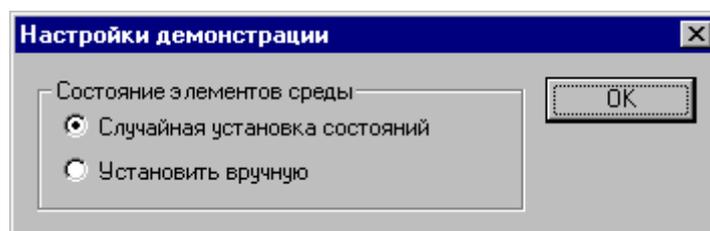


Рис. 2.12

Затем появится окно установки граничных сигналов (рис. 2.13). От правильности их установки зависит достоверность результатов тестирования.

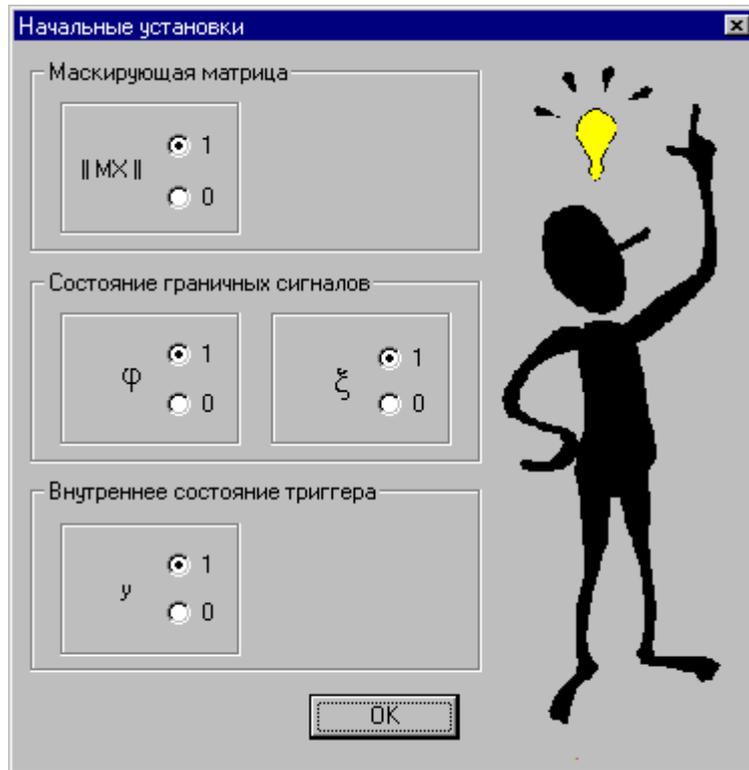


Рис. 2.13

Если был выбран режим ручного ввода данных, то на экране появятся границы матрицы в соответствии с введенными значениями n и m и курсор, показывающий текущее место ввода данных (рис. 2.14). Ввод данных осуществляется нажатием клавиш “0” или “1”. Режим ввода данных можно прервать, нажав клавишу “Esc”. Нажатие любых других клавиш игнорируется.

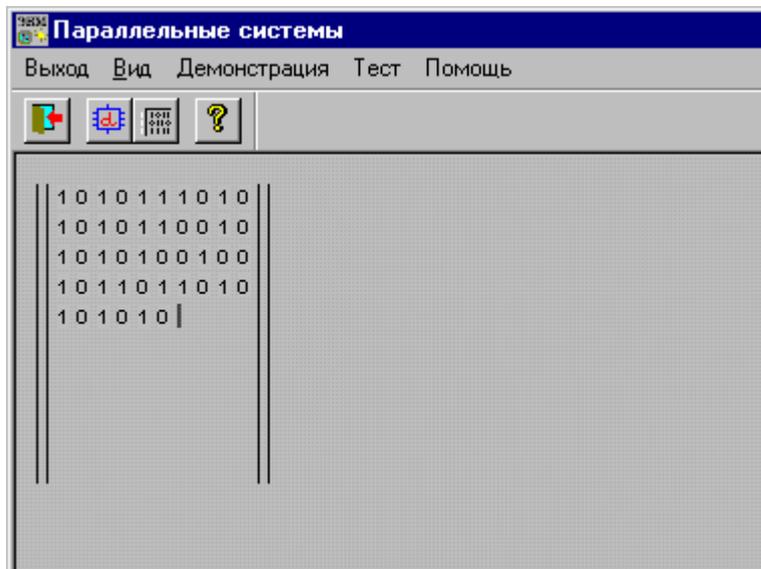


Рис. 2.14

Если был выбран режим случайного ввода данных, то они генерируются случайным образом и выдаются на экран. После этого запускается алгоритм поиска минимального покрытия, и результаты тестирования выводятся на экран (рис. 2.15). Строки, выделенные прямоугольником, образуют минимальное покрытие.

В случае, если граничные сигналы были выставлены неправильно, выдается сообщение об ошибке.

0	1	0	0	0	1	0	1	1	0	0	0	0	0	0
0	1	1	0	1	1	1	1	1	0	1	0	1	0	0
1	1	1	0	1	0	0	0	1	1	0	1	0	1	1
1	1	0	0	0	0	1	1	0	1	0	1	0	0	0
1	0	0	1	0	1	1	0	0	0	0	0	0	0	1
0	1	1	1	1	1	0	1	1	0	1	0	0	1	1
1	0	0	1	0	0	1	0	0	1	1	0	1	1	0
1	0	1	1	1	0	0	0	1	0	0	1	1	1	0
0	1	1	1	0	1	1	1	1	0	0	1	0	0	0
1	0	1	0	0	0	1	0	0	0	0	1	1	0	0
0	0	0	0	0	0	1	1	1	0	1	1	1	0	0
1	0	0	0	1	0	1	1	0	1	1	1	0	1	0
1	0	0	0	0	1	0	1	0	0	1	1	0	0	1
1	0	0	1	0	0	1	1	1	0	1	1	0	0	0
1	1	0	0	0	1	1	1	1	0	1	1	0	0	0

Рис. 2.15

Требуется несколько раз запустить процесс тестирования, задавая всякий раз граничные сигналы, и проверить правильность выполнения теста.

Занятие 3

Среда одноканального распознавания

Матрица распознавания

Базовая операционная матрица A представляет собой специализированную логико-запоминающую среду (см. рис. 3.1а), каждый элемент (j,i) которой содержит триггер для хранения элемента a_{ji} анализируемого двоичного массива (кадра) A ($j = 1 \dots k$, $i = 1 \dots l$) и дополнительную логику. По условию сигнал $z^{ji}=1$ на внешнем выходе элемента (j,i) появляется, если этот элемент отвечает правой нижней границе заданной троичной матрицы (эталона) X .

Рассмотрим случай одноктного распознавания объектов одинаковых размеров. Пусть размеры m и n матрицы X фиксированы. Боковые входы элемента рис. 3.1б определим как

$$\varphi^{j,i-1} = \|\varphi_q^{j,i-1}\|, \quad \varphi_q^{j,i-1} = a_{j,i-(n-q)}, \quad q = 1 \dots (n-1).$$

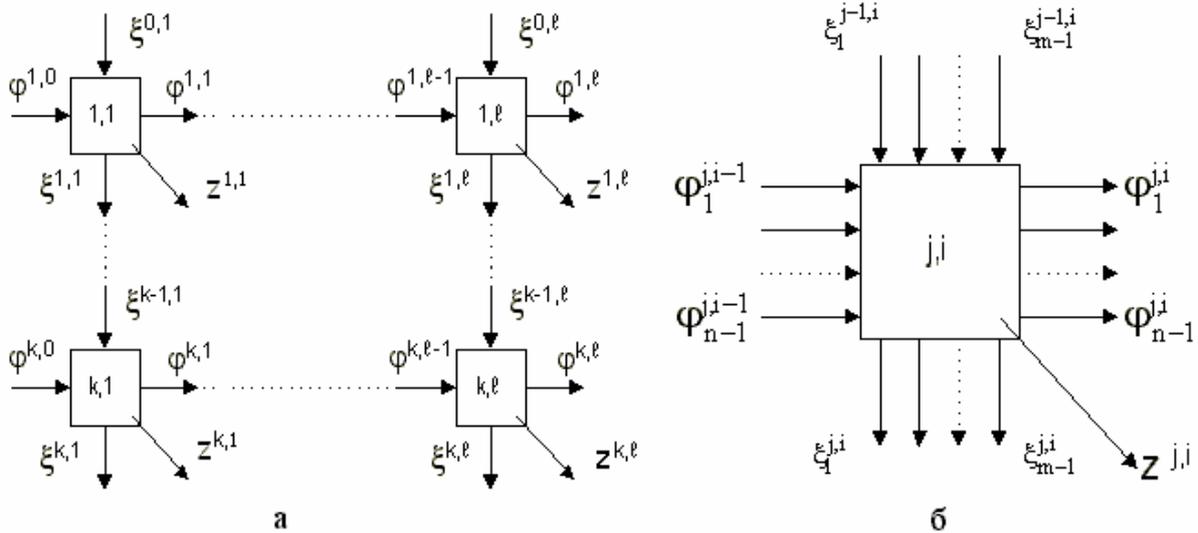


Рис. 3.1 Специализированная логико-запоминающая среда (а) и ее элемент (б)

При этом построение среды начинается с n -столбца матрицы A . Однако требуется k дополнительных $(n-1)$ -разрядных регистров для задания граничной матрицы $\|\mathbf{a}_{jq}\|, j = 1 \dots k$. Боковые выходы элемента (j,i) совпадают с боковыми входами элемента $(j,i+1)$. Поэтому

$$\varphi^{ji} = \|\varphi_q^{ji}\|, \quad \varphi_q^{ji} = \varphi_{q+1}^{j,i-1}, \quad \varphi_n^{j,i-1} = a_{ji}.$$

Обозначим $X_r = \|x_{pq}\|, p = 1 \dots r, r = 1 \dots (m-1)$. Будем говорить, что элемент (j,i) является r -границей образа X , если этот элемент отвечает правой нижней границе матрицы X_r . Введем множество вертикальных входов элемента (рис. 3.1б), так чтобы оно давало для любого допустимого значения $r \in \{1 \dots (m-1)\}$ ответ на вопрос, является ли предыдущий верхний элемент r -границей образа X :

$$\xi^{j-1,i} = \|\xi_r^{j-1,i}\|, \quad \xi_r^{j-1,i} = \bigwedge_{s=1}^r \bigwedge_{q=1}^n (\varphi_q^{j-(r-s+1),i-1})^{x_{sq}},$$

$$(\varphi_q''')^{x_{sq}} = \begin{cases} \varphi_q''', x_{sq} = 1; \\ \overline{\varphi_q'''}, x_{sq} = 0; \\ 1, x_{sq} \in \{-\}. \end{cases}$$

Соответственно вертикальные выходы

$$\xi_r^{ji} = \xi_{r-1}^{j-1,i} \bigwedge_{q=1}^n (\varphi_q^{j,i-1})^{x_{rq}}, \quad r = 1 \dots (m-1).$$

Значение $\xi_0^{j-1,i} = 1$. Граничные сигналы $\xi_r^{0,1} = 0$. Внешний выход

$$z^{ji} = \xi_m^{ji} = \xi_{m-1}^{j-1,i} \bigwedge_{q=1}^n (\varphi_q^{j,i-1})^{x_{mq}}.$$

По условию структура должна допускать последовательные опросы по множеству эталонов X неизменных размеров, что возможно только при использовании перестраиваемых модулей. В связи с этим представим выражение для $\xi_r^{j,i}$, $r = 1 \dots m$, в следующем виде:

$$\xi_r^{ji} = \xi_{r-1}^{j-1,i} \bigwedge_{q=1}^n (\varphi_q^{j,i-1} X_{rq} \vee \overline{\varphi_q^{j,i-1}} \overline{X_{rq}} \vee m_{rq}).$$

Здесь $\|X_{rq}\|$ - матрица X , в которой безразличные состояния заполнены произвольным образом, и m_{rq} - элемент матрицы масок $\|m_{rq}\|$ тех же размеров, что и матрица X :

$$m_{rq} = \begin{cases} 0, x_{rq} \in \{0,1\}; \\ 1, x_{rq} \in \{-\}. \end{cases}$$

Отсюда следует возможность реализации элемента на основе ассоциативного модуля (рис. 3.2) с выходом z типа "открытый коллектор", предназначенного для анализа строчных фрагментов разрядностью $q \leq n$. В модуле используется специальная схема RS-триггера с дополнительным управляющим входом V . Нормально $V=1$. Если же вместо триггера требуется

образовать последовательный каскад из двух инверторов, что необходимо для стыковки модулей с целью наращивания разрядности при $n > q$, то $V=0$.

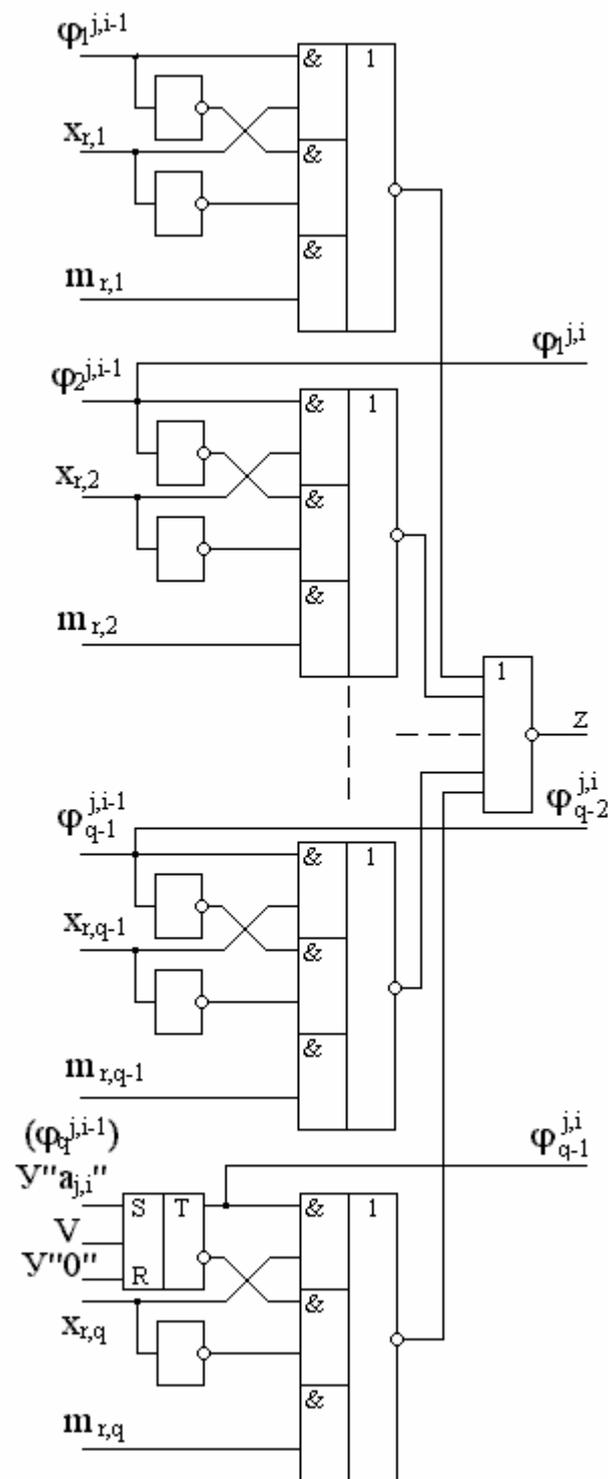


Рис. 3.2 Базовый ассоциативный модуль двумерного поиска

В табл. 3.1 приведены вычисленные параметры структуры для $l = 64$, когда $q=q_{\text{опт}}$. Здесь $q_{\text{опт}}$ - значение q , минимизирующее W при данных $N_{\text{БИС}}$, n и l ,

W - число корпусов, Q_M - горизонтальный размер фрагмента структуры, размещаемого в модуле БИС с числом сигнальных выводов $N_{\text{БИС}}$.

Таблица 3.1

Параметры базовой операционной матрицы
(случай $l = 64$, $q = q_{\text{опт}}$)

$N_{\text{БИС}}$	n	12	10	8	6	4
60	$q_{\text{опт}}$	7	10	8	6	4
	Q_M	11	7	9	12	15
	$K_{\text{БИС}}$	330	294	306	312	270
	$W_{\text{мин}}/\text{km}$	10	8	7	5	5
46	$q_{\text{опт}}$	7	6	8	6	4
	Q_M	6	7	5	7	10
	$K_{\text{БИС}}$	180	182	170	182	180
	$W_{\text{мин}}/\text{km}$	18	16	12	9	7

Как следует из таблицы 3.1, реализация матриц значительных размеров для идентификации сравнительно крупных объектов при современном уровне технологии практически затруднена. Так, уже в случае $k = l = 64$ и $n = m = 12$ требуется 7680 корпусов БИС на 60 выводов. С уменьшением количества выводов необходимое количество корпусов быстро нарастает.

Моделирование 3

Для начала моделирования необходимо запустить файл `lzs3.exe`.

Моделирование состоит из двух частей:

- Демонстрационная часть
- Основная часть.

В демонстрационной части показывается процесс распознавания на уровне элементов ЛЗС с иллюстрацией прохождения сигналов во времени через элементы среды.

Входы и выходы элементов окрашиваются в синий или красный цвет, что соответствует логическим 0 или 1. Черный цвет выхода элемента означает, что данный выход еще не принял устойчивое состояние. После того, как все входы и выходы элемента определены, он изменяет свой цвет.

В основной (тестовой) части моделируется работа структуры без показа самого процесса. Программа тестирует моделируемую структуру на правильность функционирования, давая возможность студенту подтвердить на практике усвоенный ранее теоретический материал. При необходимости в процессе работы может быть вызвана нужная справка.

Общий вид окна после запуска программы представлен на рис. 3.3. С помощью меню или кнопок в панели управления можно выбрать режим тестирования или запустить демо-версию программы. Также предусмотрен вызов справки и выход из программы.

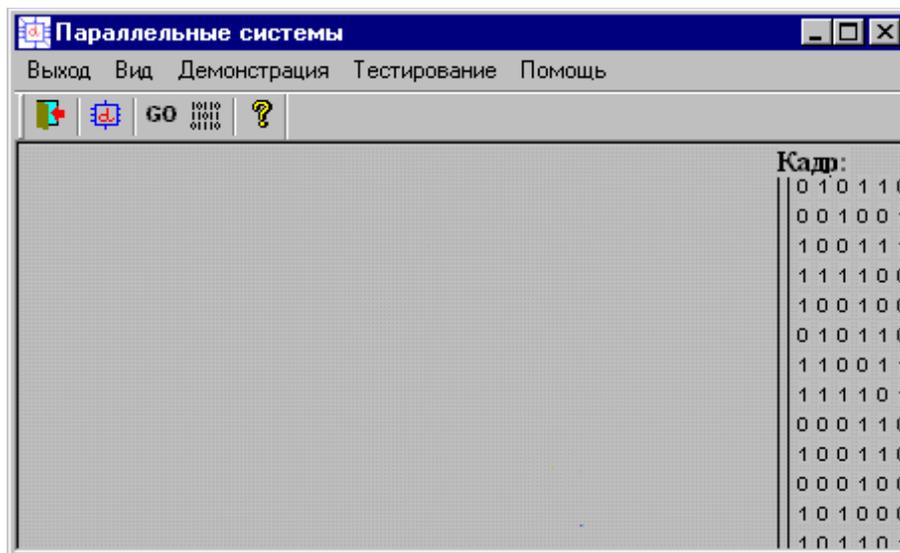


Рис. 3.3

Имеются следующие пункты меню (и соответствующие им кнопки в панели управления):

 <i>Выход</i>	Выход из программы
 <i>Демонстрация</i>	Запуск демонстрационной модели ЛЗС
 <i>Размер двоичного массива</i>	Ввод размеров двоичного массива
 <i>Тестирование</i>	Формирование и тестирование модели ЛЗС, созданной студентом
 <i>Помощь</i>	Вывод на экран справки

Для запуска демонстрационной модели необходимо выбрать в меню пункт “Демонстрация” или нажать соответствующую кнопку в панели управления. После нажатия на экране появится окно программы демонстрации (рис. 3.4). После ознакомления с демонстрационной моделью необходимо перейти к работе с тестовой версией программы.

Демонстрационная часть

Демонстрационная часть моделирует процесс распознавания эталона размером 3×3 в двоичной матрице размером 25×25 элементов. Пользователь может посмотреть процесс изменения состояния элементов среды в окне 3×3 .

Общий вид окна после запуска демонстрационной программы представлен на рис. 3.4. На рисунке изображено два поля элементов среды: поле двоичной матрицы размером (25×25) , поле просмотра размером (3×3) . Поле матрицы просмотра представлено набором элементов ЛЗС. Внутри каждого элемента цифрой (0 или 1) обозначено его состояние. Ниже располагается ряд кнопок управления демонстрацией.

Для запуска демонстрации необходимо предварительно в поле двоичной матрицы установить с помощью мыши окно просмотра и нажать кнопку "Выполнить". По желанию пользователя предварительно можно изменить состояние матрицы-эталона и двоичной матрицы. Во время демонстрации происходит изменение цвета входов и выходов элементов, а также цвета самого элемента.

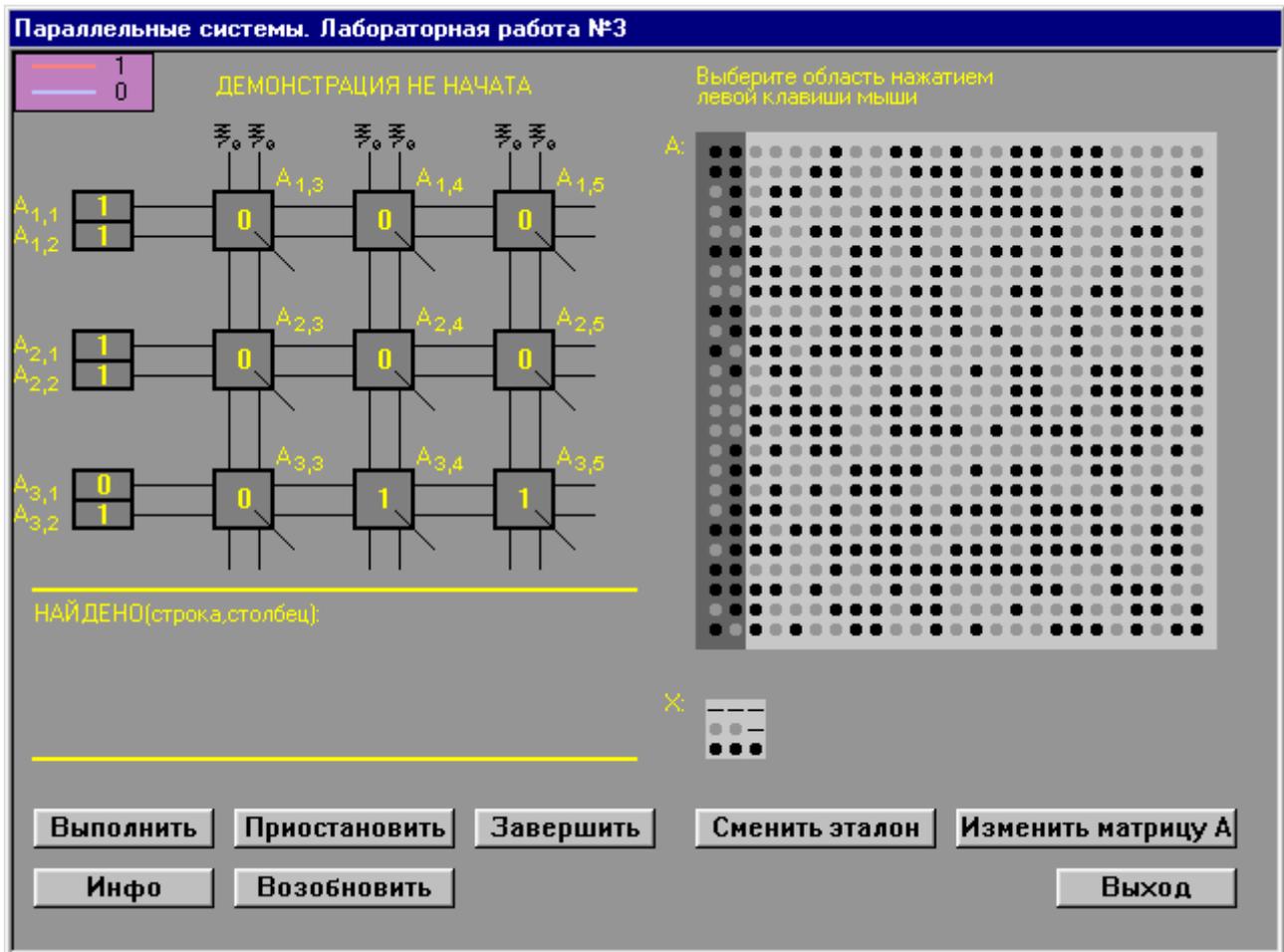


Рис. 3.4

Функциональное назначение кнопок:

<i>Приостановить</i>	Приостановка демонстрации
<i>Возобновить</i>	Возобновление демонстрации после приостановки
<i>Выполнить</i>	Запуск демонстрации
<i>Сменить эталон</i>	Смена состояний эталона
<i>Сменить матрицу A</i>	Смена состояний двоичной матрицы
<i>Завершить</i>	Закончить показ демонстрации
<i>Выход</i>	Выход из программы

В ходе демонстрации важно понять, что процесс распознавания идет по всей двоичной матрице (25×25). Но процесс поиска наблюдается только в выбранном окне(3×3).

Основная (тестовая) часть

Для запуска процесса тестирования надо выбрать в меню рис. 3.3 пункт “Тестирование” или нажать соответствующую кнопку в панели управления.

Появится запрос на размеры матрицы-эталона (рис. 3.4), где n - число строк матрицы (от 1 до 32), m - число столбцов матрицы (от 1 до 32).

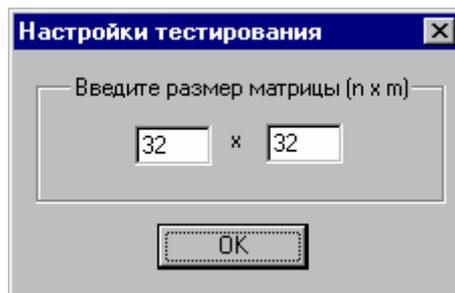


Рис. 3.4

В случае ввода недопустимого значения m или n будет выведено соответствующее предупреждение.

Далее появится запрос о способе ввода данных в матрицу (рис. 3.5).

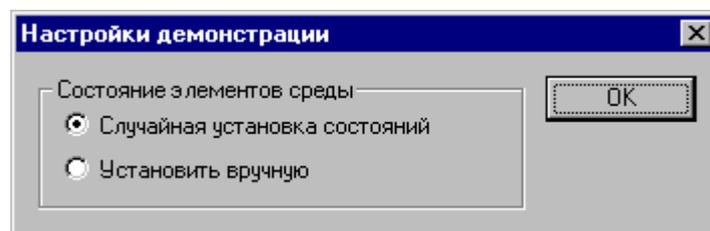


Рис. 3.5

Состояния элементов матрицы будут выбираться случайным образом или вводиться пользователем вручную.

Если данные вводятся вручную, то режим ввода можно прервать, нажав клавишу “Esc”. Нажатие любых других клавиш игнорируется. При этом выдается предупреждающий звуковой сигнал.

Если был выбран режим случайного ввода данных, то они генерируются случайным образом и выдаются на экран.

По окончании ввода появится окно результатов (рис. 3.6).

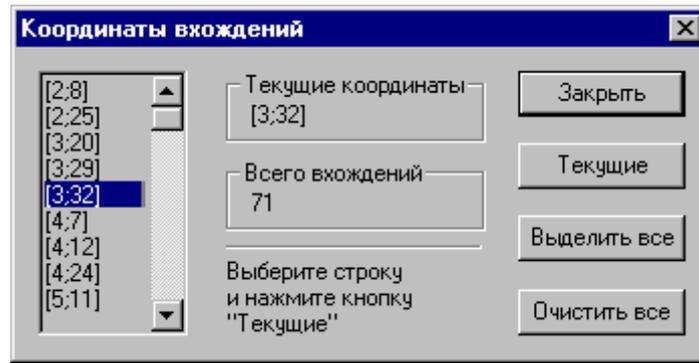


Рис 3.6

В этом окне выводится список координат всех найденных вхождений матрицы-эталона в двоичную матрицу. Кнопки "Текущие" и "Выделить все" позволяют просмотреть все найденные вхождения сразу или по одному. При нажатии на кнопку "Текущие" в двоичной матрице-кадре выделяется то вхождение, координаты которого отмечены курсором. Выделение текущих координат производится щелчком курсора мыши на нужную строку и нажатием кнопки "Текущие".

Занятие 4

Многотактное распознавание

Необходимость

Рассматриваемую задачу идентификации сформулируем следующим образом. Для данного троичного эталона (троичной матрицы)

$$X^t = \left\| x_{pq}^t \right\|; x_{pq}^t \in \{0, 1, -\}; p = 1 \dots m^t; q = 1 \dots n^t; t = 1 \dots \gamma$$

(t - номер типа объекта, γ - число типов объектов) найти координаты всех покрываемых им объектов двоичного изображения (булевой матрицы)

$$A = \left\| a_{ji} \right\|; a_{ji} \in \{0, 1\}; j = 1 \dots K; i = 1 \dots L; (K \geq m^t, L \geq n^t).$$

Однотактное аппаратное решение этой задачи (см. занятие 3) предполагает использование определенным образом организованной матрицы операционной логико-запоминающей среды на основе ассоциативных модулей

БИС специального вида. Однако реализация одноклеточных матриц практически затруднена из-за чрезмерного числа корпусов, которое быстро увеличивается с ростом размеров объектов и изображения в целом.

Выходом из затруднения является анализ изображения последовательно по кадрам и многотактно - фрагментами $(1 \times 1)n_1$ бит, $n_1 \geq 2$ - для каждого эталона. Тогда в кадре $k \times l$ бит оказывается возможным распознавание объектов различных размеров $(d \times c)n_1$, где $d = 1 \dots [k/n_1]$ и $c = 1 \dots [l/n_1]$, при разумной сложности операционной матрицы. Однако многотактная организация процедуры распознавания приводит к необходимости разработки спецпроцессора-идентификатора, работающего в комплексе с универсальной ЭВМ.

Эта ЭВМ выполняет управляющие функции: диспетчирование, прием-передача и формирование массивов данных для обработки и хранения. Спецпроцессор выполняет параллельную часть обработки. Основной вклад в объем его оборудования вносит обрабатываемая часть. Помимо операционной матрицы, она содержит многослойную буферную память с параллельным обменом данными между каждым слоем буфера и матрицей. В спецпроцессоре имеется свое устройство управления, оперативная память и так называемый координатный блок.

Существуют реальные возможности аппаратного анализа кадров размерами 128×128 . Процесс распознавания произвольных объектов может быть построен на матрицах со значением n и m от 2 до 4. При $n = m = 4$ количество корпусов $W = 2816$. Если же $n = m = 2$, то $W = 1024$. Это приемлемо.

Параллельный алгоритм

Задачей занятия является рассмотрение принципов организации спецпроцессора-идентификатора и сравнительная оценка скорости выполнения рассматриваемой процедуры идентификации на последовательной ЭВМ и комплексе: спецпроцессор - базовая ЭВМ. Решение первой части задачи будем

связывать с разработкой алгоритма многотактного распознавания. Нужную сравнительную оценку получим путем детализации (до уровня машинных команд) алгоритмов идентификации для последовательной ЭВМ и комплекса.

Размеры всех объектов будем полагать одинаковыми, а сами объекты - непересекающимися. Оценка дадим для случая "шашечного" расположения объектов (рис. 4.1а), когда количество объектов на каждом "этаже" одинаково и равно ε . В качестве варьируемых параметров возьмем размеры объектов $m \times n$, изображения в целом $K \times L$, число типов объектов γ . Анализ проведем в предположении наличия между объектами непрерывной фоновой помехи, полагая величины L , l и n кратными байту.

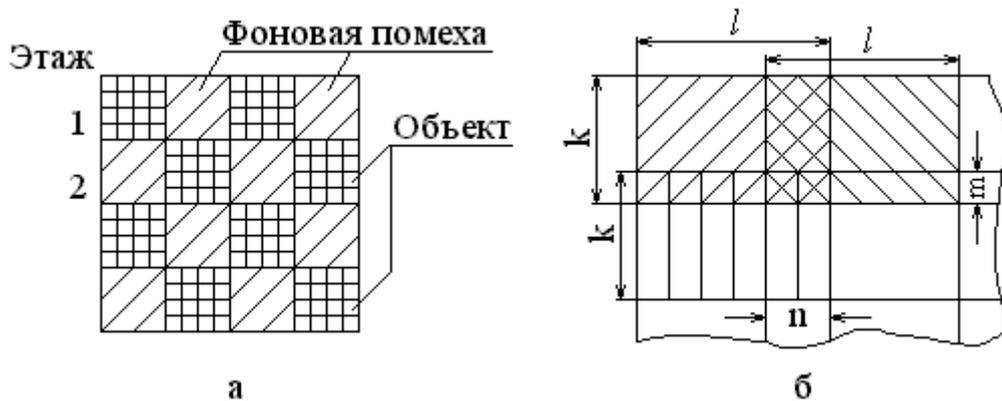


Рис. 4.1 Иллюстрация "шашечного" расположения объектов (а) и принципа формирования кадров (б)

Идентификацию на комплексе будем проводить в два этапа:

1. Формирование кадров - реализуется на базовой ЭВМ.
2. Идентификация объектов каждого кадра, последовательно по кадрам - реализуется на спецпроцессоре.

При работе с отдельным кадром $k \times l$ бит распознаются лишь те объекты, которые полностью входят в кадр. Поэтому кадры формируются с перекрытием (рис. 4.1,б): по горизонтали - на n бит, по вертикали - на m бит. Число кадров по горизонтали $Q = \lfloor (L - n) / (l - n) \rfloor$ и по вертикали $P = \lfloor (K - m) / (k - m) \rfloor$. Общее количество кадров в изображении равно PQ .

Принципы организации обрабатываемой части спецпроцессора можно уяснить из рассмотрения параллельного алгоритма многотактного распознавания объектов некоторого типа в анализируемом кадре. По условию каждый эталон последовательно и однозначно разделяется на cd непересекающихся фрагментов $\Lambda_{su} = \|\lambda_{pq}\|_{su}$ ($p, q = 1 \dots n_1; n_1 \geq 2; s = 1 \dots c; u = 1 \dots d$).

Алгоритм:

1. $w := 1, u := 1$.
2. $s := 1$.
3. Опросить кадр на Λ_{su} .
4. По результатам опроса сформировать двоичную матрицу $X' = \|x'_{ji}\|$ ($j = n_1 \dots k; i = n_1 \dots l$).
5. $s := s + 1$.
6. Если $s \leq c$, идти к шагу 7. Иначе переход к шагу 10 (для $w=1$) или к шагу 11 (при $w=2$).
7. Опросить кадр на Λ_{su} .
8. По результатам опроса сформировать матрицу $X'' = \|x''_{ji}\|$ и образовать ее покомпонентную дизъюнкцию с X' .
9. Опросить матрицу $(X' \vee X'')$ на признак-строку $\Gamma = \|1' - - - 1''\|$, где число безразличных состояний равно $(n_1 - 1)$. Переход к шагу 4.
10. По результатам опроса сформировать матрицу $Y' = \|y'_{ji}\|$. Переход к шагу 13.
11. По результатам опроса сформировать матрицу $Y'' = \|y''_{ji}\|$ и образовать ее покомпонентную дизъюнкцию с Y' .
12. Опросить матрицу $(Y' \vee Y'')$ на признак-столбец Γ^T (транспонированная матрица Γ). Переход к шагу 10.
13. $w := 2; u := u + 1$.

14. Если $u \leq d$, идти к шагу 2. Иначе к шагу 15.

15. Результат опроса принять за конечный результат.

На рис. 4.2 дана сжатая последовательная иллюстрация процесса многоступенчатого распознавания по предложенному алгоритму в случае $k = 6$; $l = 14$; $m = 4$; $n = 6$; $n_1 = 2$ ($c = 3$, $d = 2$) и

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}, \quad X = \begin{pmatrix} \Lambda_{11} & \Lambda_{21} & \Lambda_{31} \\ 1 & - & - & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & - \\ 1 & 1 & - & 1 & - & 1 \\ \Lambda_{12} & \Lambda_{22} & \Lambda_{32} \end{pmatrix}.$$

На рис. 4.2 представлены:

а - результат опроса A на $\Lambda_{11} \rightarrow X'$;

б - результат опроса A на $(\Lambda_{11} \vee \Lambda_{21}) \rightarrow (X' \vee X'')$;

в - результат опроса $(X' \vee X'')$ на $\Gamma = \|1' - 1''\|$ (опроса A на $\Lambda_{11}\Lambda_{21}$) \rightarrow новая матрица X' ;

г - результат опроса A на $(\Lambda_{11}\Lambda_{21} \vee \Lambda_{31}) \rightarrow$ новая матрица $(X' \vee X'')$;

д - результат опроса $(X' \vee X'')$ на Γ (опроса A на $\Lambda_{11}\Lambda_{21}\Lambda_{31}$) $\rightarrow X' \rightarrow Y'$;

е - результат опроса A на $(\Lambda_{12} \vee \Lambda_{22}) \rightarrow (X' \vee X'')$;

ж - результат опроса A на $\Lambda_{12}\Lambda_{22} \rightarrow X'$;

з - результат опроса A на $(\Lambda_{12}\Lambda_{22} \vee \Lambda_{32}) \rightarrow (X' \vee X'')$, $1^* = 1' \vee 1''$;

и - результат опроса A на $(\Lambda_{11}\Lambda_{21}\Lambda_{31} \vee \Lambda_{12}\Lambda_{22}\Lambda_{32}) \rightarrow (Y' \vee Y'')$;

к - результат опроса $(Y' \vee Y'')$ на Γ^T (опроса A на X) $\rightarrow Y' \rightarrow$ конечный результат распознавания.

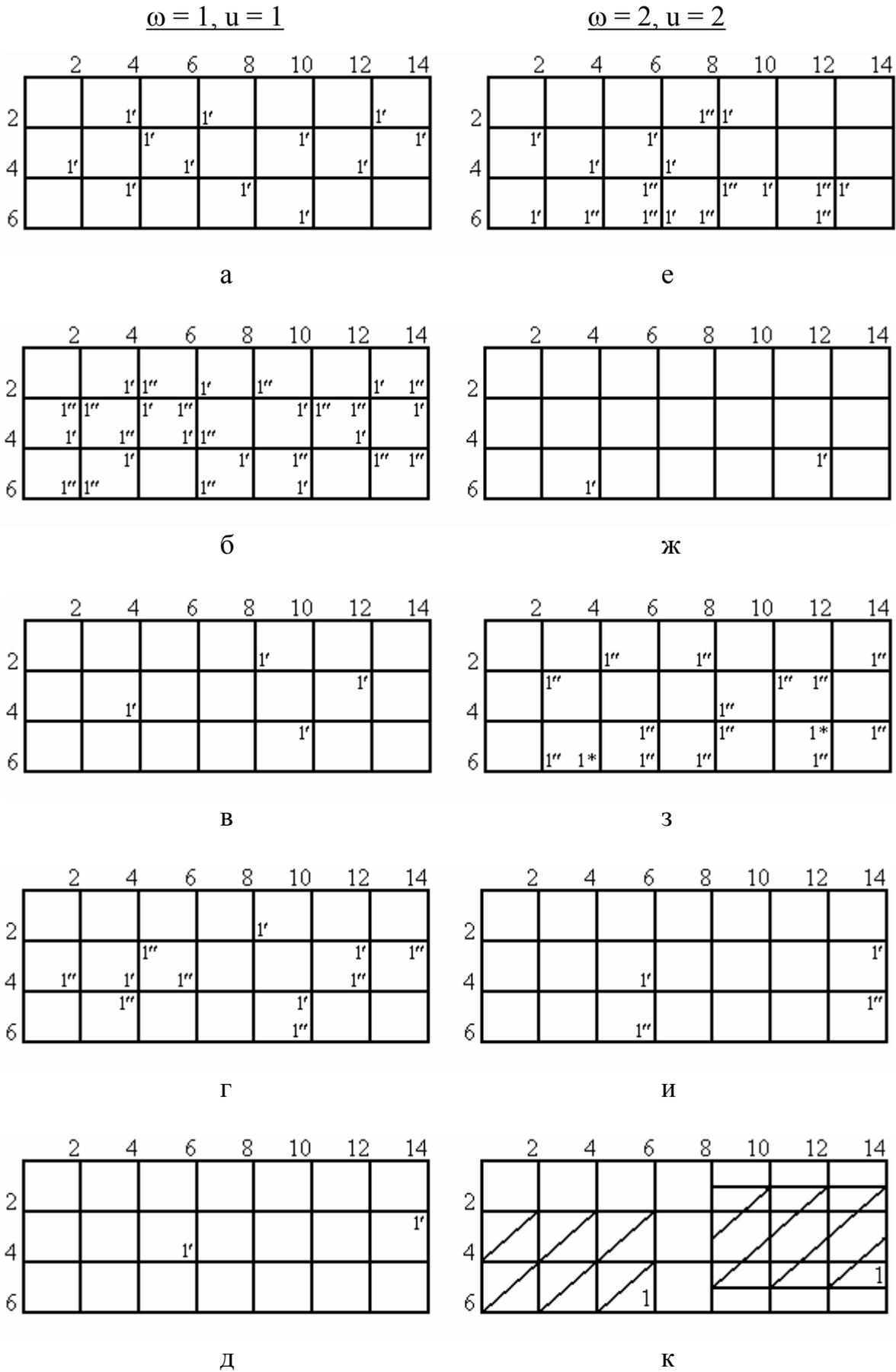


Рис. 4.2 Иллюстрация процесса многотактного распознавания для рассмотренного примера

Занятие 5

Матричный спецпроцессор - идентификатор

Структура спецпроцессора

Для реализации алгоритма необходимо образовать обрабатывающий массив (рис. 5.1,а) из буферной памяти (БФ_q - q-й слой буфера) и трех операционных матриц: базовая матрица А – структура однотактного распознавания фрагментов Λ_{su} ; В – среда формирования матриц X' и (X' v X'') и распознавания строчных фрагментов Г; С – среда формирования матриц Y' и (Y' v Y'') и распознавания столбцовых фрагментов Γ^T .

Массив функционирует следующим образом. Анализируемый кадр размещается в элементах памяти матрицы А. По сигналам ОПРОС А,В,С результаты опроса соответствующих сред – $X=||x_{ji}||$, $Y=||y_{ji}||$, $Z=||z_{ji}||$ – записываются в БФ_q. При подаче сигнала синхронизации СИ_{В,С} происходит переключение матрицы В или С в соответствии с информацией $V=||v_{ji}||$, записанной ранее в БФ_q. Одновременно формируются результаты очередного опроса на Г или Γ^T .

Сигналы УСТ В,С подаются перед началом распознавания. Они устанавливают элементы памяти сред В и С в исходное состояние. В шаге 2 алгоритма необходимо предусмотреть дополнительную установку матрицы В. По сигналу ВЫДАЧА из БФ_q считывается результат.

Матрицы X'(Y') и X''(Y'') различаются "окраской" содержащихся в них единиц (1' и 1''), что достигается использованием в средах В и С специальных элементов памяти. Значения сигналов на выходах этих элементов: 00 - если данный элемент не является границей ни одного из двух фрагментов, последовательно распознаваемых предыдущей средой; 10, 01 и 11 - если данный элемент является границей только первого из этих фрагментов, только второго и обоих фрагментов соответственно.

Согласно алгоритму, матрица X'(Y') в среде В(С) формируется по каждому нечетному синхроимпульсу, а матрица X''(Y'') - по каждому четному. Это отражено в таблице переходов рассматриваемых элементов (табл. 5.1), где

v - соответствующая компонента матрицы V ; s^k - состояние; $\varphi_1\varphi_2^k$ - выходы. При составлении таблицы учтено, что для принятой кодировки выходов в матрицах B или C распознаются фрагменты $\Phi = \|(1-)(- -)\dots(- -)(-1)\|$ либо Φ^T с двухбитными компонентами. Поэтому, если после четного синхроимпульса на выходах элемента памяти имеем 00 либо 10, то очередное значение $v = 0$.

Организация элемента (j,i) среды $B(C)$ показана на рис. 5.1,б. Входные сигналы на границе среды - нулевые. Через $\varphi_{i(j)}^\Lambda, \varphi_{i(j)}^\Pi$ обозначены левый и правый выходы элемента памяти.

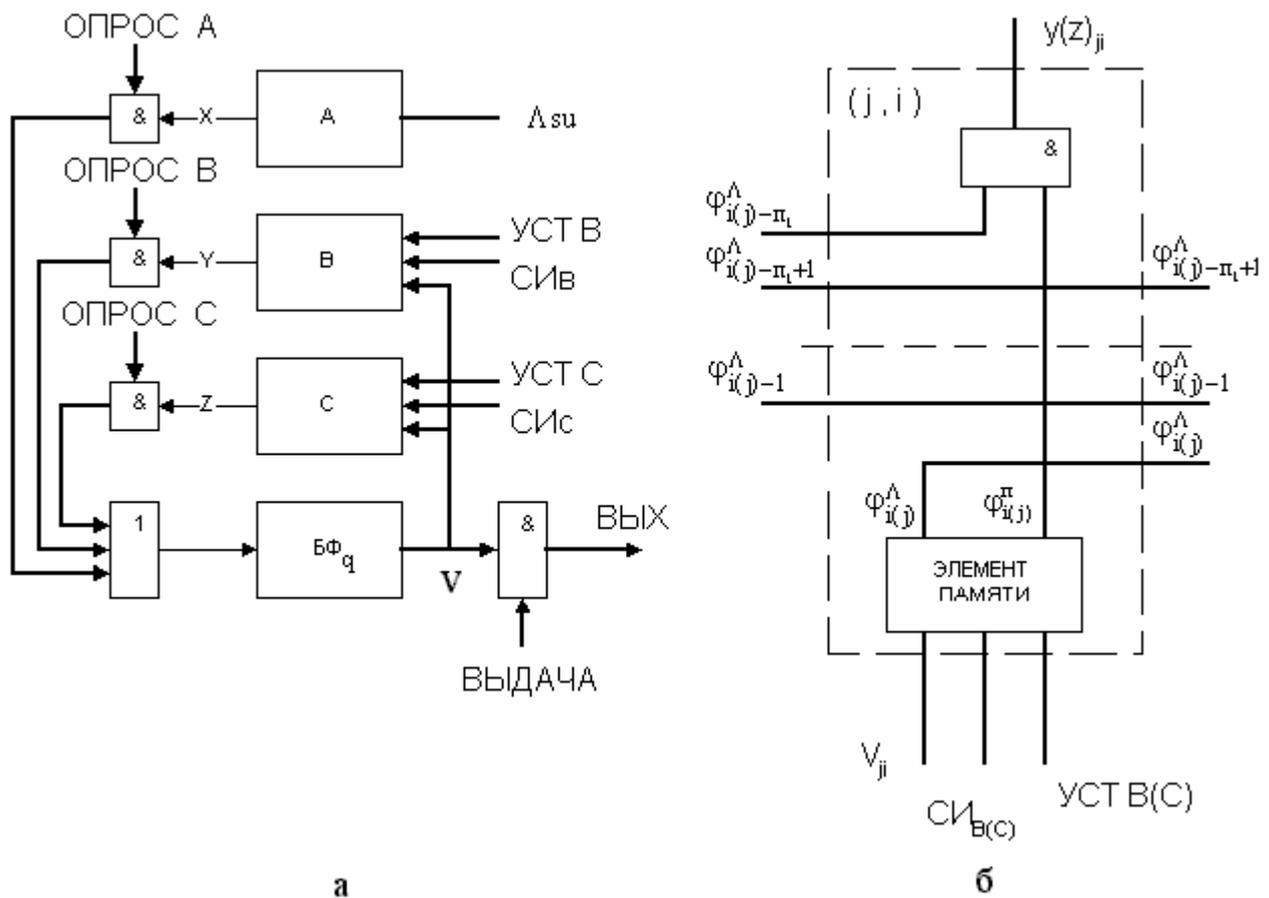


Рис. 5.1 Организация обрабатывающей части спецпроцессора (а)
и элемента среды (б)

Таблица переходов элемента памяти сред В и С

$s_{k-1} \backslash v$	0	1
1	2,00	3,10
2	2,00	1,01
3	1,10	1,11

$S^k, \varphi_1, \varphi_2^k$

Покажем теперь, каким образом проводится на комплексе (базовая ЭВМ - спецпроцессор) идентификация объектов изображения в целом по заданному набору эталонов. Исходное изображение, точнее, - его непрерывная развертка по двоичным строкам слева направо и сверху вниз, занимает байты памяти базовой ЭВМ $I = I_0 \dots (I_0 + KL/8 - 1)$. Пусть (y, x) - координаты кадра, $y = 0 \dots (P-1)$; $x = 0 \dots (Q-1)$; величина $j = 0 \dots (k-1)$ - номер строки внутри кадра. Тогда адреса крайних левых байтов кадра

$$I_{yx}^j = I_0 + y \frac{(k-m)L}{8} + x \left[\frac{l-n}{8} \right] + j \left(\frac{L}{8} \right).$$

Массив результатов формирования занимает байты памяти J с базовым адресом J_0 . При каждом (y, x, j) пересылка из одной области памяти в другую идет по $l/8$ байт. Поэтому адреса левых крайних байтов кадра (y, x) в области результата формирования

$$J_{yx}^j = J_0 + y \left(\frac{Qkl}{8} \right) + x \left(\frac{kl}{8} \right) + j \left(\frac{l}{8} \right).$$

Приведенные выражения определяют алгоритм формирования.

Сформированные кадры $\gamma = 0 \dots (PQ - 1)$ размещаются в буферной памяти спецпроцессора, по одному слою $БФ_\gamma$ на каждый кадр γ . Информация о троичных эталонах хранится в оперативной памяти спецпроцессора в байтах

$S = S_0 \dots \left(S_0 + \frac{mn}{4} \gamma - 1 \right)$. При этом каждый троичный фрагмент Λ_{su} занимает

ровно 1 байт: 4 разряда - на фрагмент двоичного эталона и еще 4 разряда - на соответствующую маску (случай $n_1=2$). Адресация байтов:

$$S_{su}^t = S_0 + \left(\frac{mn}{4}\right)t + \left(\frac{n}{2}\right)s + u; t = 0 \dots (\gamma - 1); s = 0 \dots \left(\frac{m}{2} - 1\right); u = 0 \dots \left(\frac{n}{2} - 1\right).$$

Имеется байтовый регистр, в который последовательно заносятся фрагменты Λ_{su} . В буфере выделяется рабочий слой БФ_q. Он постоянно используется для получения промежуточных результатов распознавания в виде единичных отметок в соответствующих битах рассматриваемого в данный момент кадра по каждому типу объекта (см. рис. 4.2, 5.1,а). Эти результаты передаются в координатный блок, который функционирует параллельно с обрабатывающей частью спецпроцессора (как показал анализ, совмещение имеет место при $m, n \geq 14$).

Блок служит для определения координат идентифицированных объектов. Он содержит простейшую операционную матрицу $k \times l$ бит, выполняющую поиск в строках на "≠0", с регистровым обрамлением и микропрограммное управление. Итоговая информация /t, r, j, i/ о каждом факте идентификации заносится в оперативную память спецпроцессора. Приведенные замечания вместе с рассмотренным ранее многотактным алгоритмом определяют структуру и программу работы спецпроцессора в целом.

Пусть N_n - число выполняемых машинных команд при решении рассматриваемой задачи на последовательной ЭВМ, N_ϕ и N_i - то же для комплекса на этапах формирования и идентификации; T_n , T_c , T_ϕ и T_i - соответствующие временные затраты. Сравнительные оценки быстродействия последовательной ЭВМ и комплекса могут быть получены, если известно, за какое число машинных тактов (τ_n , τ_ϕ и τ_i соответственно) выполняется в среднем одна команда в том или иной случае при неизменной длительности такта. Тогда искомая оценка определена отношением T_n/T_c , где $T_n = \tau_n N_n$, $T_c = T_\phi + T_i$, $T_\phi = \tau_\phi N_\phi$, $T_i = \tau_i N_i$. С учетом состава выполняемых команд и структурных особенностей рассматриваемых устройств для проанализированных вариантов было найдено: $\tau_n = 3,45$; $\tau_\phi = 5,25$; $\tau_i = 1,26$.

Результаты проведенных расчетов даны в таблице 5.2. Нетрудно заметить, что эффективность использования спецпроцессора растет с

уменьшением размеров объектов. Размеры изображения в целом и параметр γ слабо влияют на эффективность. При этом быстродействие повышается примерно на 2 порядка по сравнению с последовательной ЭВМ с той же длительностью такта.

Таблица 5.2

Результаты сравнительной оценки (случай $k = l = 128$)

$K = L$	$m = n$	γ	N_{π}	N_{ϕ}	$N_{и}$	T_{π}/T_c
1008	24	64	307539017	256554	8115803	91,68
1008	24	32	164419017	256554	4058203	87,80
504	24	64	77754828	64154	2028953	92,72
1008	16	64	348086506	207814	3069417	242,0

По условию обработка информации в матрицах A , B и C , как и обращение (чтение - запись) к любому слою буфера, занимают один такт спецпроцессора. В процессе сравнительного анализа быстродействия предполагалось, что в последовательной ЭВМ регистровые команды также выполняются за один такт той же длительности.

Приведенные оценки показывают достаточно высокую эффективность использования спецпроцессора для решения рассмотренной задачи идентификации. Правда, полученные оценки связывались только с производительностью. Для полноты картины необходимо, хотя бы приближенно, оценить дополнительные затраты оборудования. В связи с этим заметим, что при $k = l = 128$ и $n_1 = 2$ реализация трех операционных матриц (A , B , C) требует 2294 корпусов БИС на 60 выводов. Использование кристаллов ПЛИС на 264 сигнальных вывода позволит снизить необходимое число корпусов до 406.

Моделирование 4

Для начала моделирования необходимо запустить файл `cruid2.exe`.

Моделирование состоит из двух частей:

- Основная часть (одиночное распознавание).
- Дополнительная часть (множественное распознавание).

В основной части предполагается подробное изучение процесса идентификации объекта на кадре. Изучение может проводиться как в пошаговом режиме, так и в нормальном. В пошаговом режиме предлагается посмотреть выполнение процесса распознавания на уровне элементов ЛЗС. Здесь показывается процесс прохождения сигналов во времени через элементы среды. Входы и выходы элементов окрашиваются в синий или красный цвет, что соответствует логическим 0 или 1.

В нормальном режиме программа моделирует работу структуры распознавания, не показывая самого процесса. Программа тестирует моделируемую структуру на правильность функционирования, давая возможность студенту подтвердить на практике изученный ранее теоретический материал.

В дополнительной части демонстрируются возможности алгоритма при множественном распознавании объектов. Объекты загружаются с диска, и по их табличному описанию генерируется кадр, в котором будет проводиться распознавание. При правильном распознавании итоговый результат должен совпасть с заданной таблицей.

Общий вид окна после запуска программы представлен на рис. 5.2. В верхней части окна находятся управляющие пункты меню, а также три панели инструментов, дублирующие пункты меню для облегчения работы. В правой нижней части экрана отображаются имена загруженных кадра, объекта и маски.

Изначально в окне ничего не выводится и большинство пунктов меню недоступно. Их активизация происходит после выбора режима работы - одиночного или множественного распознавания.

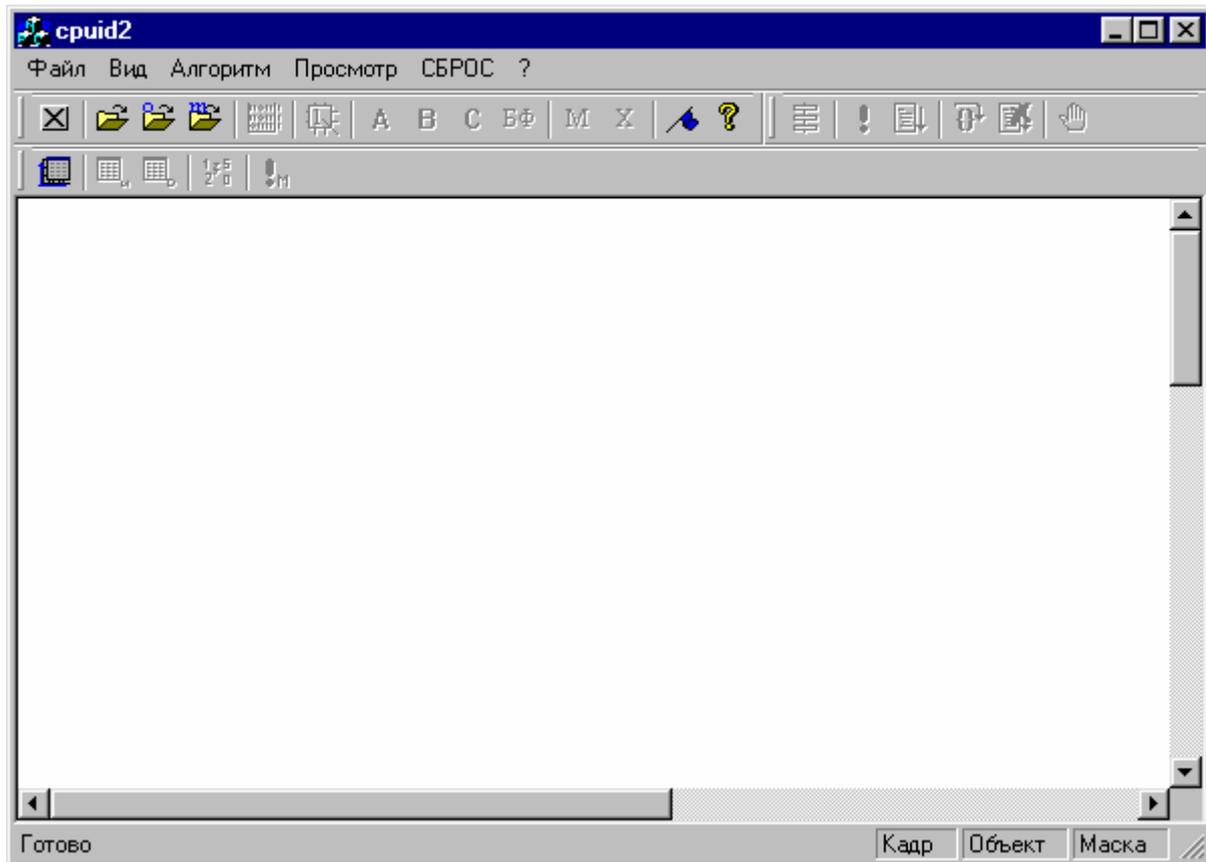


Рис. 5.2

Назначение пунктов меню (рис. 5.2):

Файл

Загрузка кадра

Загрузка объекта Загрузка соответствующих матриц с диска для однозначного распознавания

Загрузка маски

Загрузить объекты Загрузка объектов для множественного распознавания

Выход Выход из программы

Вид

Панель инструментов Убирает/показывает панель инструментов

Строка статуса Убирает/показывает строку статуса внизу окна программы.

*Панель инструментов
одиночного
распознавания* Убирает/показывает панель инструментов

*Панель инструментов
множественного
распознавания* Убирает/показывает панель инструментов

Алгоритм

<i>Последовательный алгоритм</i>	Запуск алгоритма, реализованного по методу обычного перебора значений ячеек и их сравнения
<i>Параллельный алгоритм</i>	Запуск алгоритма работы спецпроцессора-идентификатора
<i>Пошаговый алгоритм</i> <i>Точки останова</i>	То же, но в пошаговом режиме Задание точек приостановки работы в процессе выполнения алгоритма. Необходимо, если нужно просмотреть состояние матриц на каком-либо шаге алгоритма
<i>Следующий шаг</i>	Запуск следующего шага в случае, если произошел останов на заданном шаге
<i>Остановить алгоритм</i>	Прекращение работы алгоритма в пошаговом режиме
<i>Генерировать кадр</i>	Генерация кадра по табличному описанию объектов
<i>Таблица с исходными координатами</i>	Вывод на экран таблицы с заданными координатами локализаций объектов
<i>Таблица с распознанными координатами</i>	Вывод на экран таблицы с распознанными координатами локализаций объектов
<i>Запуск алгоритма множественного распознавания</i>	Запуск алгоритма множественного распознавания для сгенерированного кадра

Просмотр

<i>Показать результат</i>	Показать результат распознавания объекта на кадре
<i>Элементы матрицы</i>	Позволяет посмотреть внутренние состояния элементов матриц и значения их выводов в условном графическом изображении (рис. 5.4).
<i>Матрица А</i> <i>Матрица В</i> <i>Матрица С</i> <i>Буфер БФq</i> <i>Маска</i> <i>Объект</i>	Вывод на экран содержимого соответствующих матриц

Сброс Сброс всех результатов и обнуление загруженных матриц

Способ задания матриц:

Для режима однократного распознавания используется следующий формат матриц. Матрицы представляют собой текстовый файл формата ASCII. Программа производит последовательное чтение символов из файла и преобразует их в числовой вид. Знак перевода строки означает конец строки. Количество столбцов считается по количеству символов в строке. Для редактирования значений ячеек достаточно открыть файл в любом текстовом редакторе DOS. Данные, записанные в этом файле, выглядят так, как будто загружены в память. Поэтому следует быть осторожным при ручном редактировании файла.

Для режима многотактного распознавания, кроме текстовых объектов формата ASCII, возможна загрузка объектов черно-белых битовых матриц с расширением (*.bmp). Данные файлы могут быть созданы в любом графическом редакторе типа PaintBrush.

Размеры матриц должны удовлетворять следующим условиям (табл. 5.3).

Таблица 5.3

Матрица	Одиночное распознавание	Множественное распознавание
Кадр	128×128 (max)	1000×1000 (фикс)
Объект	(2-128)×(2-128)	20×20 (фикс)
Маска	Должна совпадать с размером объекта	

Все размеры должны быть четными.

Основная часть (одиночное распознавание)

Для начала работы в режиме одиночного распознавания необходимо загрузить кадр, объект и маску объекта. Это возможно при выборе

соответствующих пунктов меню "Файл" или при нажатии аналогичных кнопок на панели инструментов.

После загрузки вышеуказанных данных становится возможным просмотр загруженных матриц и запуск алгоритма распознавания.

Просмотр матриц осуществляется следующим образом.

При нажатии на кнопки А,В,С,БФ,М и Х на экране отображаются соответственно содержимое трех матриц А,В,С, обрабатывающей части спецпроцессора, буферной памяти, маски объекта и содержимое самого загруженного объекта.

При нажатой кнопке "Просмотр элементов" возможен просмотр матриц А,В,С как набора логических элементов (рис. 5.3).

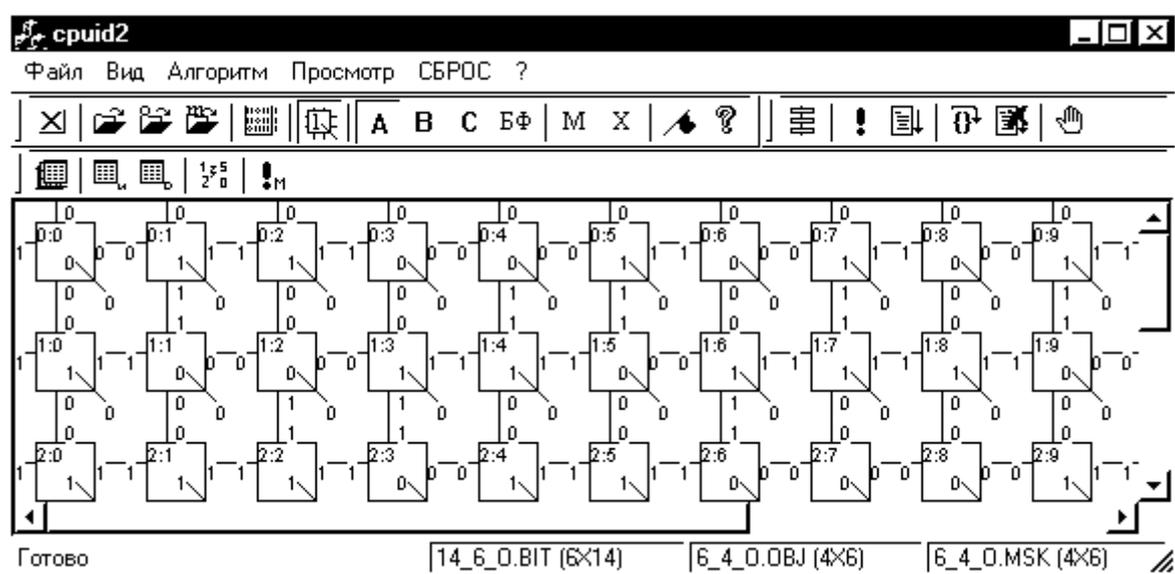


Рис. 5.3

Каждый элемент индицирует свое состояние соответствующей окраской рамки и цифрой 0 или 1 внутри. Элементы связаны между собой проводниками, которые окрашиваются в зависимости от уровня проходящих по ним сигналов: "Лог. 0" – синий цвет, "лог. 1" – красный цвет. На элементе также отображен его номер (j,i) в матрице.

Запуск алгоритма возможен как в пошаговом режиме, так и в нормальном. В нормальном режиме после окончания распознавания станет доступным пункт

меню "Просмотр результата", при выборе которого на экране отобразится матрица А с распознанными на ней объектами (рис. 5.4).

При работе в пошаговом режиме есть возможность просмотреть процесс функционирования обрабатывающей части спецпроцессора на любом шаге алгоритма. Для этого необходимо расставить контрольные точки. Диалог расстановки контрольных точек запускается при выборе соответствующего пункта меню (рис. 5.5).

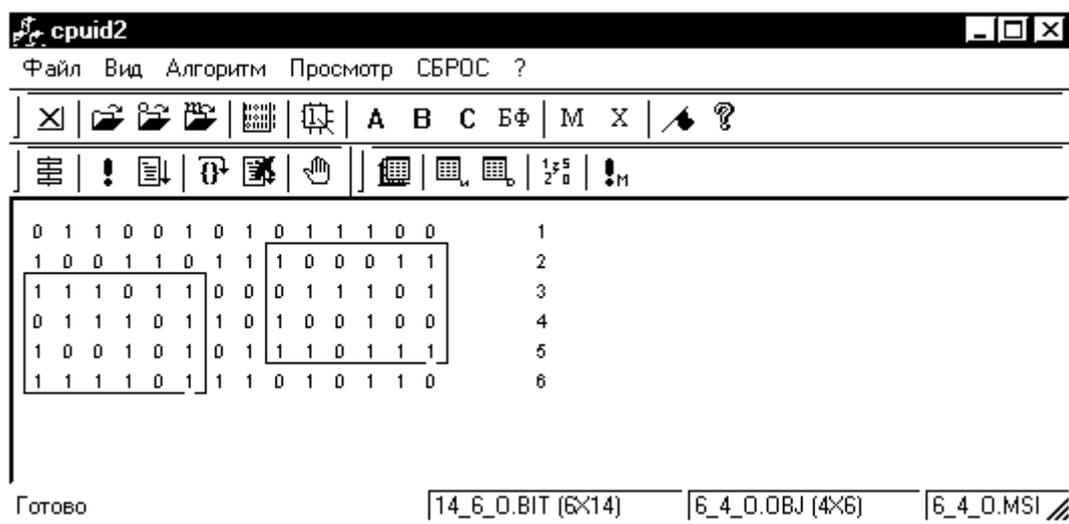


Рис. 5.4

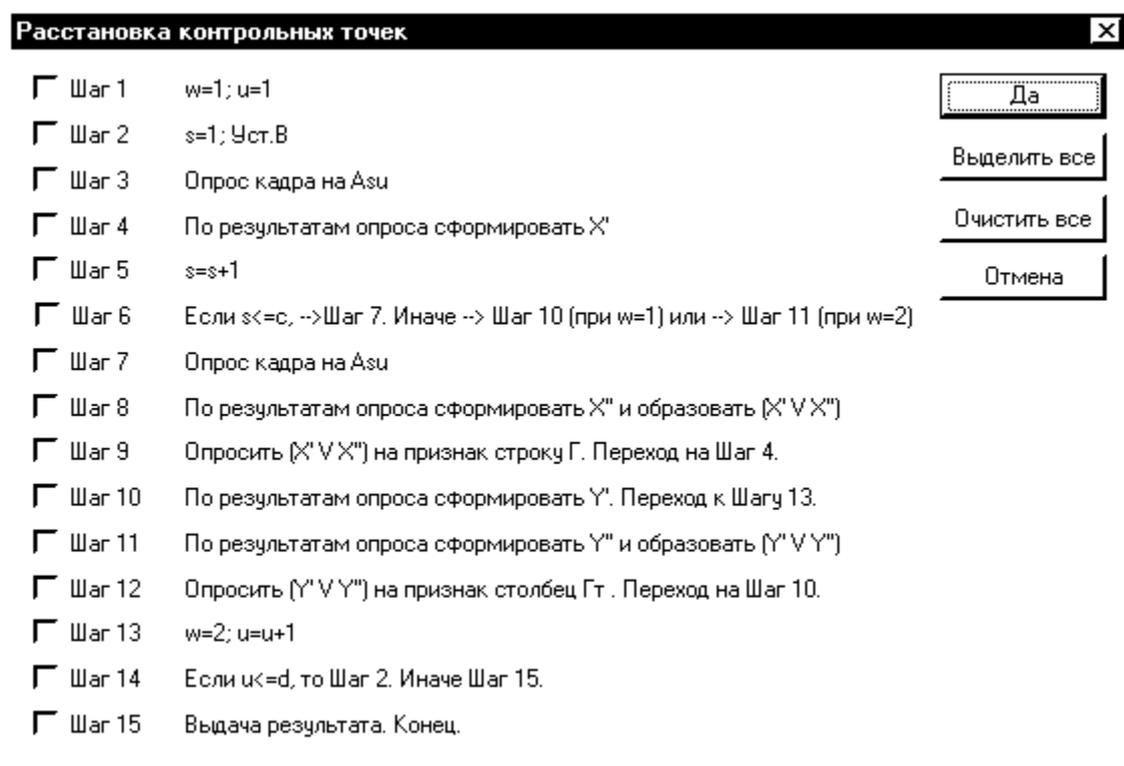


Рис. 5.5

После расстановки контрольных точек в нужных пунктах алгоритма необходимо осуществить запуск пошагового алгоритма нажатием кнопки или через пункт меню. Программа автоматически дойдет до контрольного пункта и остановится, ожидая запуска следующего шага. Для перехода к следующей контрольной точке необходимо выбрать пункт меню "Следующий шаг". Пока не выбран следующий шаг, у пользователя есть возможность просмотреть состояния матриц на данный момент.

По завершении работы алгоритма выдается сообщение о его окончании, и пользователь может посмотреть конечный результат распознавания.

Если в процессе выполнения алгоритма в пошаговом режиме возникла необходимость закончить алгоритм, то в меню нужно выбрать пункт "Остановить алгоритм".

Дополнительная часть (множественное распознавание)

Для перехода в режим множественного распознавания сначала сбрасываются все установки внутренних переменных путем выбора пункта меню "Сброс". Затем выполняется загрузка объектов, для чего необходимо выбрать пункт меню "Загрузка объектов". На экране появится диалог (рис. 5.6).

Диалог условно разделен на две половины. В левой половине располагается список известных программе файлов с объектами. В правой половине находится список объектов, которые по заданным координатам будут генерироваться в кадр.

Список известных файлов можно редактировать с помощью трех кнопок:

Новый – добавление новых файлов к списку.

Удалить – удаление выбранного файла из списка.

Запомнить список – запоминание текущего списка.

Чтобы добавить выбранный объект в список используемых файлов, необходимо нажать кнопку "Добавить>>". Выбранный файл появится в списке используемых. Этому файлу можно присвоить любое имя, под которым он

условно будет обозначен в таблице координат, и назначить ему количество локализаций в кадре.

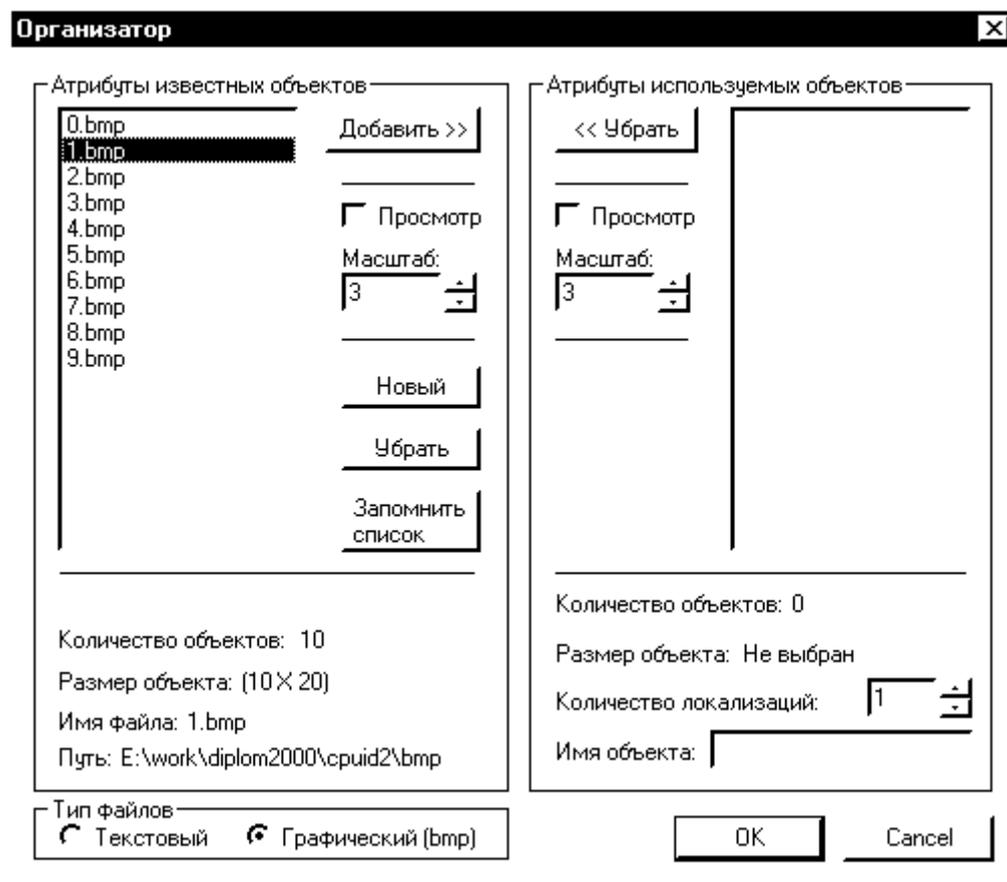


Рис. 5.6

При выборе необходимого количества объектов и нажатии кнопки ОК на экране появится диалог с таблицей координат локализаций объектов (рис. 5.7).

В этом диалоге представлены имена загруженных объектов со случайно сгенерированными координатами. Для того, чтобы исключить совпадение координат и нулевые координаты, необходимо нажать кнопку "Контроль координат". Кнопкой "Расставить по узлам" координаты вновь случайно генерируются в узлах сетки (40×40) на кадре 1000×1000. При необходимости можно вручную изменить значение координат. Для этого необходимо сделать двойной щелчок мыши по номеру объекта.

После нажатия кнопки ОК необходимо сгенерировать кадр, выбрав пункт меню "Генерировать кадр".

Процесс моделирования для каждой части занятия следует повторить несколько раз при различных исходных данных. Правильность получаемых результатов будет свидетельствовать об адекватности рассмотренного на занятии 4 алгоритма распознавания принятой структуре спецпроцессора.

Литература

1. Райхлин В.А. Асинхронные цифровые схемы и модульные структуры. - Казань: КАИ им. А.Н. Туполева. 1980.
2. Варшавский В.И. и др. Однородные структуры. - М.: Энергия. 1973.
3. Каутц У.Х. Однородные логико-запоминающие среды и большие интегральные схемы /Синтез автоматов и управление на сетях связи. - М.: Наука. 1973.
4. Райхлин В.А. Операционные логико-запоминающие среды. Вопросы применения и синтеза //Автоматика и телемеханика.1983. № 11.
5. Однородные микроэлектронные ассоциативные процессоры /Под ред. И.В. Прангишвили. - М.: Сов. радио. 1973.
6. Фет Я.И. Массовая обработка информации в специализированных однородных процессорах. - Новосибирск: Наука. 1976.
7. Райхлин В.А. Об использовании аппарата двумерного ассоциативного поиска в процессе распознавания /Проблемно-ориентированные средства повышения эффективности вычислительных систем. - Казань: КАИ им. А.Н. Туполева. 1991.
8. Райхлин В.А. Анализ производительности процессорных матриц при распознавании двоичных образов //Автометрия. 1996. № 5.
9. Ульман Дж. Вычислительные аспекты СБИС. - М.: Радио и связь. 1990.
10. Евреинов Э.В., Хорошевский В.Г. Однородные вычислительные системы. - Новосибирск: Наука. 1978.
11. Системы параллельной обработки /Под ред. Д. Ивенса. - М.: Мир. 1985.

Райхлин Вадим Абрамович
Вершинин Игорь Сергеевич
Глебов Евгений Евгеньевич

ПРАКТИКУМ
ПО ПАРАЛЛЕЛЬНЫМ СИСТЕМАМ
Учебное пособие

ЛР № 020678 от 09.12.97

Формат 60×84 1/16. Бумага офсетная. Печать офсетная.

Печ. л. 3.25. Усл. печ. л. . Усл. кр.-отт. . Уч.-изд. л. .

Тираж 100. Заказ

Издательство Казанского государственного технического университета

Типография издательства Казанского государственного технического
университета

420111 Казань, К. Маркса, 10.