

Изображение с растрового электронного микроскопа свидетельствует о наличии равномерной эпитаксиальной пленки толщиной примерно 3,6 мкм с четкой гетерограницей. На поверхности имеется наличие неоднородного массива квантовых точек и более крупных образований (островков), имеющих размеры порядка нескольких микрометров. Результаты рентгеновского микроанализа свидетельствуют о том, что состав выращенной эпитаксиальной пленки соответствует составу источника.

Из вышесказанного можно сделать вывод о том, что технология ионно-лучевого осаждения весьма перспективна для получения многокомпонентных полупроводниковых соединений  $A^3B^5$  и формирования на их основе полупроводниковых приборов. Дальнейшие эксперименты должны быть связаны с получением пятикомпонентных соединений, а также квантово-размерных структур.

#### **Библиографический список**

1. *Аброян, И. А.* Физические основы электронной и ионной технологии [Текст] / И. А. Аброян, А. Н. Андронов, А. И. Титов. – М. : Высшая школа, 1984.
2. *Розанов, Л. Н.* Вакуумная техника [Текст] / Л. Н. Розанов. – М. : Высшая школа, 1990.
3. *Технологический источник КЛАН-53М. Система электропитания СЕФ-53М. Система газоподдачи. Техническое описание. Инструкция по эксплуатации [Текст] / утв. Горчаров Л. А. – № 217-2009. ОО НПК «Платар». – М., 2009. – 31 с.*

УДК 004.312.46

### **СХЕМОТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ И МОДЕЛИРОВАНИЕ ВЫЧИСЛИТЕЛЬНОГО УСТРОЙСТВА С ПРОГРАММИРУЕМОЙ ЛОГИКОЙ**

**А.В. Осовский**

*Работа публикуется при поддержке РФФИ в рамках гранта 10-07-90713*

*Рассматривается схемотехническая реализация и преимущества схемы с программируемой логикой, реализующей высокоскоростную реконфигурацию наборов схем программируемой логики. Представлены результаты моделирования работы вычислительно-го устройства с программируемой логикой.*

**Ключевые слова:** *программируемая логика, вычислительные устройства с программируемой логикой, выполнение логических операций, моделирование.*

**Key words:** *programmed logic, computing devices with programmed logic, performance of logic operations, modeling.*

Развитие отечественных схемотехнических решений для информационных и телекоммуникационных систем имеет большое стратегическое значение. В настоящее время большинство схемотехнических решений в данной области, созданных зарубежными фирмами-производителями, не патентуется, а охраняется в режиме «ноу-хау». Такой подход, хотя и является отчасти оправданным коммерческий ответом на неправомерное копирование, однако в научном отношении тормозит развитие новых схемотехнических решений.

В цифровых системах обработки информации, в системах управления, телекоммуникационных системах находят все более широкое применение программируемые логические интегральные схемы (ПЛИС), которые используются для получения нестандартного набора схем, реализующих требуемые булевы функции. Исследованию ПЛИС посвящены работы многих исследователей: А.Н. Алыневского, А.П. Антонова, Р.И. Грушвицкого, В.Г. Домрачева, П.П. Мальцева, Д.А. Кнышева, В.Б. Стешенко, Е.П. Угрюмова и др. Как правило, такие схемы строятся на основе логических элементов с матричным способом соединения элементов ячеек, с возможностью перепрограммирования ячеек матрицы.

Основной недостаток ПЛИС – отсутствие возможности перепрограммирования матрицы без остановки работы системы. Даже ПЛИС таких известных производителей, как Xilinx и Altera, не позволяют при обработке массивов информации оперативно изменять алгоритм обработки, т.е. производить перенастройку «на лету». У интегральных схем Xilinx и Altera процесс перенастройки связан с остановкой системы и занимает определенное время [1].

В настоящее время при контроле и измерении параметров в реальном режиме времени в системах обработки данных, а также при проведении сложных научных экспериментов актуальной становится задача динамического реконфигурирования системы, когда в процессе работы необходима автоматическая высокоскоростная реконфигурация наборов реализуемых схем программируемой логики. Подобная высокоскоростная реконфигурация не может быть реализована на стандартных ПЛИС. Поэтому было разработано вычислительное устройство с программируемой логикой, реализующее высокоскоростную реконфигурацию для простых наборов логических комбинационных схем, т.е. реализация функции ПЛИС с перепрограммированием «на лету». Такой принцип реализован на базе матричного коммутатора [2].

Вычислительное устройство с программируемой логикой (ВУПЛ) (рис. 1) состоит из узлов выделения команд 1.1, 1.2, ..., 1.n, переключателей вертикальных шин 2.1, 2.2, ..., 2.n, узлов вертикальной настройки 3.1, 3.2, ..., 3.n, коммутирующих узлов, состоящих из элементов памяти 4.1.1, 4.1.2, ..., 4.1.n, 4.2.1, 4.2.2, ..., 4.2.n, ..., 4.m.1, 4.m.2, ..., 4.m.n и узлов ключей 5.1.1, 5.1.2, ..., 5.1.n, 5.2.1, 5.2.2, ..., 5.2.n, ..., 5.m.1, 5.m.2, ..., 5.m.n, выходных узлов 6.1, 6.2, ..., 6.m, узлов горизонтальной настройки 7.1, 7.2, ..., 7.m и местного устройства управления 8. Вычислительное устройство с программируемой логикой содержит n входов 9.1, 9.2, ..., 9.n и m выходов в виде двунаправленной шины 10.1, 10.2, ..., 10.m, вход начальной установки 11, вход передачи информации 12, вход разрешения настройки 13, синхровход 14, n шин вертикальной настройки 15.1, 15.2, ..., 15.n, m шин горизонтальной настройки 16.1, 16.2, ..., 16.m, 2n вертикальных шин 17.1, 17.2, ..., 17.n, 18.1, 18.2, ..., 18.n, и 2m промежуточных шин 19.1, 19.2, ..., 19.m, 20.1, 20.2, ..., 20.m. ВУПЛ содержит 5 внутренних шин управления: 3 шины выделения команд 47, 48, 50, шину настройки 49 и шину управления передачей информации 51.

Вычислительное устройство с программируемой логикой позволяет реализовать высокоскоростную динамическую реконфигурацию наборов реализуемых булевых схем и функционирует в режиме разовой коммутации – после настройки по образованным каналам передаются массивы данных, и параллельно выполняется их логические преобразования, до тех пор пока массив не будет полностью обработан. Далее поступает для обработки следующий массив данных или поступает заявка на перепрограммирование и перестройку матричной структуры.

Алгоритм, описывающий работу вычислительного устройства, в режиме программирования кодов логических команд содержит процедуры последовательного выделения и фиксации трех разрядов кода логической команды с целью инверсии над входными переменными, распараллеливания входных переменных и инверсии результатов операций.

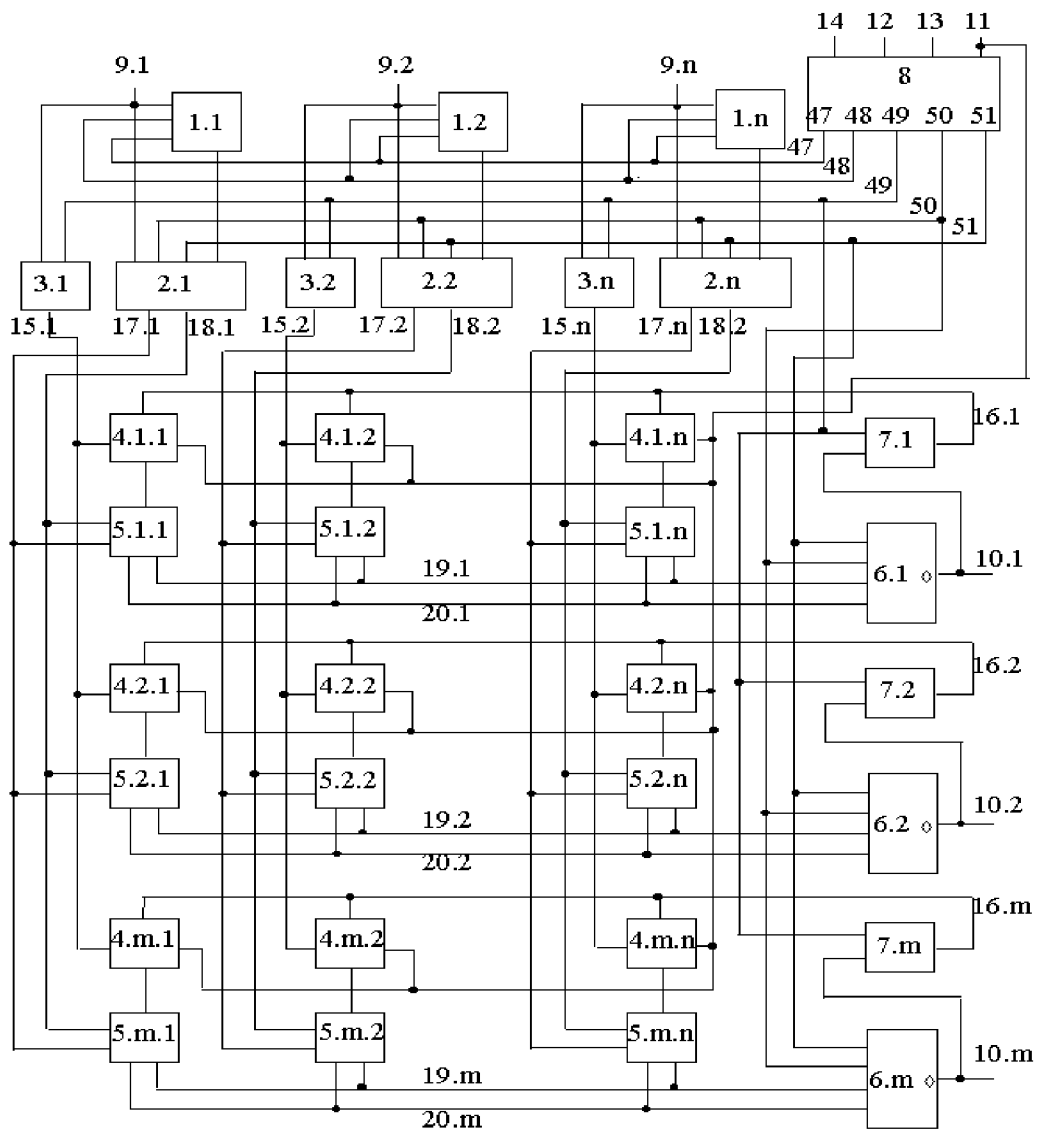


Рис. 1. Структурная схема вычислительного устройства с программируемой логикой

С целью исполнения простейших логических операций в вычислительном устройстве введены процедуры поразрядной конъюнкции с использованием свойств промежуточных шин, т.е. предусмотрены алгоритмы параллельного выполнения логических операций над массивами данных с использованием одной и двух внутренних шин устройства. Возможные наборы логических схем с высокоскоростной реконфигурацией, реализуемые на вычислительном устройстве с программируемой логикой, приведены в табл. 1, 2.

Таблица 1

Булевы схемы, реализуемые на вычислительном устройстве с программируемой логикой

	Коды команд															
	Vx <sub>1</sub>	Vx <sub>k</sub>	Vx <sub>1</sub>	Vx <sub>k</sub>	Vx <sub>1</sub>	Vx <sub>k</sub>	Vx <sub>1</sub>	Vx <sub>k</sub>	Vx <sub>1</sub>	Vx <sub>k</sub>	Vx <sub>1</sub>	Vx <sub>k</sub>	Vx <sub>1</sub>	Vx <sub>k</sub>	Vx <sub>1</sub>	Vx <sub>k</sub>
1-й разряд (D1)	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	1
2-й разряд (D2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3-й разряд (D11)	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Логическое выражение	$X \cdot Y$		$\bar{X} \cdot Y$		$X \cdot \bar{Y}$		$\bar{X} \cdot \bar{Y}$		$\overline{X \cdot Y}$		$\overline{\bar{X} \cdot Y}$		$\overline{X \cdot \bar{Y}}$		$\overline{\bar{X} \cdot \bar{Y}}$	
Функция	конъюнкция		Запрет x		Запрет y		Стрелка Пирса		Штрих Шеффера		Импликация от у к х		Импликация от х к у		дизъюнкция	

Таблица 2

Схемы «эквивалентности» и «исключающее ИЛИ», реализуемые на вычислительном устройстве с программируемой логикой

Операция			Vx <sub>1</sub>	Vx <sub>k</sub>	Vx <sub>1</sub>	Vx <sub>k</sub>	Vx <sub>1</sub>	Vx <sub>k</sub>	Vx <sub>1</sub>	Vx <sub>k</sub>
Эквивалентность	Коды команд	1-й разряд (D1)	0	0	1	0	0	1	1	1
		2-й разряд (D2)	1	1	1	1	1	1	1	1
		3-й разряд (D11)	0	0	1	1	1	1	0	0
		Логическое выражение	$X \cdot Y \vee \bar{X} \cdot \bar{Y}$		$\bar{X} \cdot Y \vee X \cdot \bar{Y}$		$\overline{X \cdot Y \vee \bar{X} \cdot \bar{Y}}$		$\overline{\bar{X} \cdot Y \vee X \cdot \bar{Y}}$	
Исключающее ИЛИ	Коды команд	1-й разряд (D1)	0	0	1	0	0	1	1	1
		2-й разряд (D2)	1	1	1	1	1	1	1	1
		3-й разряд (D11)	1	1	0	0	0	0	1	1
		Логическое выражение	$X \cdot Y \vee \bar{X} \cdot \bar{Y}$		$\bar{X} \cdot Y \vee X \cdot \bar{Y}$		$X \cdot \bar{Y} \vee \bar{X} \cdot Y$		$\overline{\bar{X} \cdot \bar{Y} \vee X \cdot Y}$	

Моделирование логической структуры вычислительного устройства с программируемой логикой проведено в программе «Electronics Workbench». Для этого функциональные схемы отдельных блоков с помощью опции Subcircuit преобразованы в отдельные подсхемы, имеющие структуру типа вход/выход. Необходимое количество подсхем каждого функционального узла соединено соответствующим образом, получена общая схема вычислительного устройства с программируемой логикой размерностью 2 × 2 программируемые ячейки (рис. 2).

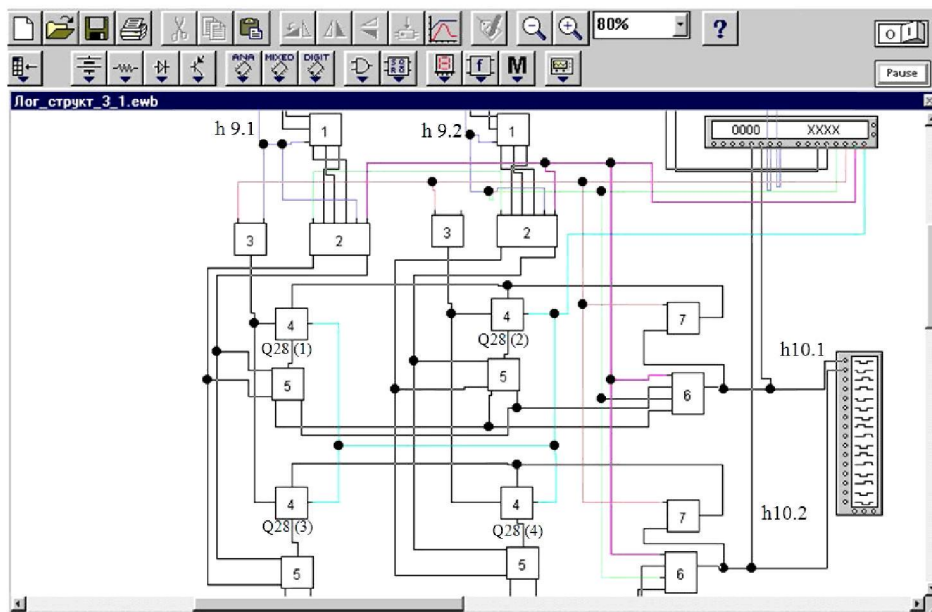


Рис. 2. Моделирование вычислительного устройства с программируемой логикой

К входам схемы подключен генератор двоичных кодов, который выдаст по тактам заданную последовательность сигналов. К контрольным точкам подключен логический анализатор, позволяющий получать временные диаграммы работы устройства в каждом такте.

Работа ВУПЛ в режиме программирования проанализирована с помощью шестнадцатиканального логического анализатора (опция Logic Analyzer). Временные диаграммы подтверждают корректность алгоритм программирования вычислительного устройства (рис. 3).

Работу вычислительного устройства с программируемой логикой в режиме выполнения логических операций над массивами данных с использованием одной и двух промежуточных шин проанализируем на схеме  $4 \times 2$  программируемых ячейки.

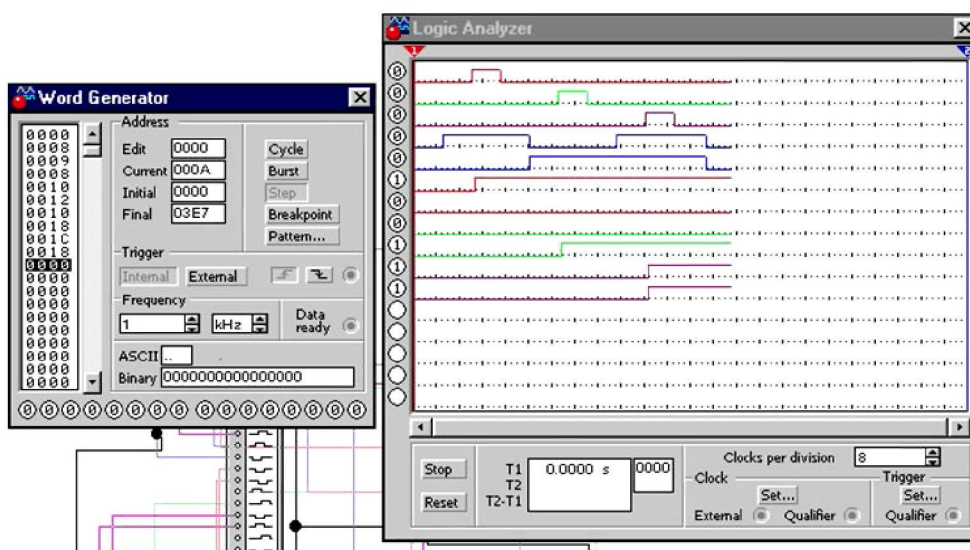


Рис. 3. Временные диаграммы режима программирования вычислительного устройства с программируемой логикой

Выполним параллельно следующие логические операции над потоками информации, поступающими в вычислительное устройство с программируемой логикой:

- 1) операцию конъюнкции над потоками, поступающими по входам 9.2 и 9.3 с выдачей результата операции на выход 10.1;
- 2) операцию «исключающее ИЛИ» над потоками, поступающими по входам 9.1 и 9.4 с выдачей результата на выход 10.2.

Для этого предварительно настроим устройство следующим образом:

- 1) в режиме фиксации каналов триггеры в элементах памяти 4.1.2, 4.1.3, 4.2.1, 4.2.4 устанавливаем в единичное состояние;
- 2) в режиме программирования по входам 9.1 ... 9.4 последовательно поданы коды логических команд (0000; 1001; 1001).

Таким образом, триггеры во всех узлах выделения команд 1.1... 1.4, в узлах выделения команд 1.1, 1.4 и в выходном узле 6.2 устанавливаются в единичное состояние, согласно кодам логических операций (см. табл. 1).

Как видно из временных диаграмм (рис. 4), над входными информационными сигналами  $h_{9,2}$  и  $h_{9,3}$  выполнена операция конъюнкции с передачей результата на выход 10.1. При этом использована одна промежуточная шина. Над входными информационными сигналами  $h_{9,1}$  и  $h_{9,4}$  выполнена операция «исключающее ИЛИ» с передачей результата на выход 10.2 с помощью двух промежуточных шин.

Таким образом, моделирование показало корректность алгоритмов функционирования вычислительного устройства с программируемой логикой.

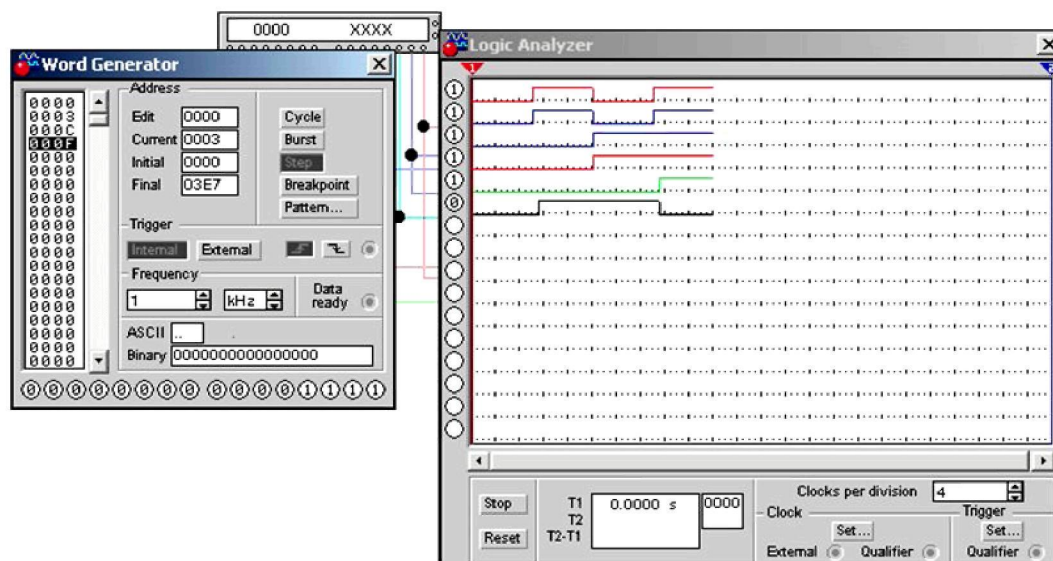


Рис. 4. Временные диаграммы выполнения логических операций над массивами данных в вычислительном устройстве с программируемой логикой

#### Библиографический список

1. *Кнышев, Д. А.* ПЛИС фирмы «Xilinx»: описание структуры основных семейств [Текст] / Д. А. Кнышев, М. О. Кузелин. – М. : Изд. дом «Додэка – XXI», 2001. – 238 с.
2. *Пат.* 2251792 Российская Федерация МПК<sup>7</sup> Н 03 К 17/04. Матричный коммутатор с программируемой логикой / Жила В. В., Осовский А. В., Кутузов Д. В. ; заявитель и патентообладатель Астрахан. гос. техн. ун-т. – № 2003113569/09 (014402) ; заявл. 08.05.2003; опубл. 10.05.05 г., Бюл. № 13.