
ПРЕДИСЛОВИЕ

Программируемые логические интегральные схемы (ПЛИС) — удобная в освоении и применении элементная база, альтернативы которой зачастую не найти. Структурно книга разбита на семь глав и три приложения.

В главе 1 дается обзор перспективных семейств ПЛИС фирмы «Altera» и краткие сведения об особенностях их архитектуры и временных параметрах устройств. В главе 2 рассмотрена САПР MAX+PLUS II. Глава 3 посвящена языку описания аппаратуры AHDL. Язык описания аппаратуры VHDL рассмотрен в главе 4, а аппаратуры VERILOG HDL — в главе 5. В главе 6 приводятся примеры описания цифровых устройств на языках высокого уровня. В главе 7 приведены примеры реализации некоторых алгоритмов. В приложении 1 рассмотрены особенности САПР Quartus. В приложении 2 — современные интерфейсы передачи данных. В приложении 3 даны рекомендации по разработке печатных плат.

Следует заметить, что книга ни в коей мере не подменяет собой фирменную документацию, без которой проектирование устройств просто невозможно. Ее цель — помочь начинающему разработчику в выборе элементной базы и дать представление о технологии проектирования устройств на ПЛИС.

Автор выражает огромную благодарность фирме «Гамма» и лично ее директору М.А.Кузнеченкову за осуществление издания. Автор также благодарит сотрудников фирмы «Гамма» С.Н.Шипулина, И.Г.Алексеева, А.А.Кулакова и зам. директора центра «Логические системы» В.Ю.Храпова за предоставленное программное обеспечение и информацию.

Автор благодарит редакторов Издательского дома «Додэка-XXI» за чуткое и внимательное отношение к рукописи.

Огромный вклад в работу над книгой внесли к.т.н. доцент Д.А. Губанов, аспирант Ю.М. Седякин, студенты А.В. Самохин, Г.В. Шишкин, инженеры Н.Н. Анищенко, А.В. Евстифеев, Р.Б. Гаврилов и др., которым автор выражает искреннюю благодарность.

В журнале Chip News (Новости о микросхемах) был опубликован цикл статей «Школа разработки аппаратуры цифровой обработки сигналов на ПЛИС», ставших основой книги. Автор признателен главному редактору журнала А.Г. Биленко и научному редактору А.А. Осипову за полезные дискуссии, способствовавшие появлению книги.

Наконец, работа над книгой была бы в принципе невозможна без поддержки и терпения семьи.

Автор надеется, что книга найдет понимание у читателя, и будет признателен за все отклики, которые можно присылать по адресу:

107005, Москва, 2-я Бауманская улица, д. 5. Кафедра СМ-5 «Автономные информационные и управляющие системы» МГТУ им. Н.Э. Баумана.

E-mail: steshenk@sm.bmstu.ru

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	13
ГЛАВА 1. ЭЛЕМЕНТНАЯ БАЗА	19
1.1. Семейство MAX3000	19
1.2. Семейство FLEX6000	33
1.3. Семейство MAX7000	46
1.4. Семейство FLEX8000	53
1.5. Семейство MAX9000	58
1.6. Семейство FLEX10K	65
1.7. Семейство APEx20K	78
1.8. Семейство Mercury	85
1.9. Семейство ACEX	102
1.10. Конфигурационные ПЗУ	104
1.11. Программирование и реконфигурирование в системе	110
ГЛАВА 2. СИСТЕМА ПРОЕКТИРОВАНИЯ MAX+PLUS II	115
2.1. Общие сведения	115
2.2. Процедура разработки проекта	121
2.3. Редакторы MAX PLUS II	140
2.4. Процесс компиляции	154
2.5. Верификация проекта	162
ГЛАВА 3. ЯЗЫК ОПИСАНИЯ АППАРАТУРЫ AHDL	167
3.1. Общие сведения	167
3.2. Использование чисел и констант в языке AHDL	171
3.2.1. Использование чисел	171
3.2.2. Использование констант	171
3.3. Комбинационная логика	172

3.3.1.	Реализация булевых выражений и уравнений	172
3.3.2.	Объявление NODE (узел)	173
3.3.3.	Определение групп	173
3.3.4.	Реализация условной логики	174
3.3.5.	Описание дешифраторов	177
3.3.6.	Использование для переменных значений по умолчанию	180
3.3.7.	Реализация логики с активным низким уровнем	182
3.3.8.	Реализация двунаправленных выводов	183
3.4.	Последовательностная логика	184
3.4.1.	Объявление регистров	184
3.4.2.	Объявление регистровых выходов	186
3.4.3.	Создание счетчиков	186
3.5.	Цифровые автоматы с памятью (state mashine)	187
3.5.1.	Реализация цифровых автоматов (state machine)	188
3.5.2.	Установка сигналов Clock, Reset и Enable	189
3.5.3.	Задание выходных значений для состояний	189
3.5.4.	Задание переходов между состояниями	190
3.5.5.	Присвоение битов и значений в цифровом автомате	190
3.5.6.	Цифровые автоматы с синхронными выходами	191
3.5.7.	Цифровые автоматы с асинхронными выходами	193
3.5.8.	Восстановление после неправильных состояний	194
3.6.	Реализация иерархического проекта	196
3.6.1.	Использование макрофункций системы MAX+PLUS II фирмы «Altera»	196
3.6.2.	Создание и применение пользовательских макрофункций	199
3.6.3.	Определение пользовательской макрофункции	199
3.6.4.	Импорт и экспорт цифровых автоматов (state machine)	201
3.7.	Управление синтезом	203
3.7.1.	Реализация примитивов LCELL и SOFT	203
3.7.2.	Значения констант по умолчанию	205
3.7.3.	Присвоение битов и значений в цифровом формате	205
3.8.	Элементы языка AHDL	205
3.8.1.	Зарезервированные ключевые слова	205
3.8.2.	Символы	207
3.8.3.	Имена в кавычках и без кавычек	208
3.8.4.	Группы	209
3.8.5.	Числа в языке AHDL	211
3.8.6.	Булевы выражения	211
3.8.7.	Логические операторы	212
3.8.8.	Выражения с оператором NOT	213

3.8.9. Выражения с операторами AND, NAND, OR, XOR, & XNOR	213
3.8.10. Арифметические операторы	214
3.8.11. Компараторы (операторы сравнения)	215
3.8.12. Приоритеты в булевых уравнениях	216
3.8.13. Примитивы	216
3.8.14. Порты	226
3.9. Синтаксис языка AHDL	229
3.9.1. Лексические элементы	229
3.9.2. Основные конструкции языка AHDL	230
3.9.3. Синтаксис объявления названия	233
3.9.4. Синтаксис оператора включения	233
3.9.5. Синтаксис задания константы	233
3.9.6. Синтаксис прототипа функции	233
3.9.7. Синтаксис оператора вариантов	234
3.9.8. Синтаксис секции подпроекта Subdesign	236
3.9.9. Синтаксис секции переменных	236
3.9.10. Синтаксис объявления цифрового автомата	237
3.9.11. Синтаксис объявления псевдонима цифрового автомата	237
3.9.12. Синтаксис логической секции	238
3.9.13. Синтаксис булевых уравнений	238
3.9.14. Синтаксис булевых уравнений управления	238
3.9.15. Синтаксис оператора CASE	238
3.9.16. Объявление по умолчанию	238
3.9.17. Синтаксис условного оператора IF	239
3.9.18. Синтаксис встроенных (in-line) ссылок на макрофункцию или примитив	239
3.9.19. Синтаксис объявления таблицы истинности	239
3.9.20. Синтаксис порта	239
3.9.21. Синтаксис группы	240
3.9.22. Синтаксические группы и списки	241
ГЛАВА 4. ЯЗЫК ОПИСАНИЯ АППАРАТУРЫ VHDL	243
4.1. Общие сведения	243
4.2. Алфавит языка	246
4.2.1. Комментарии	247
4.2.2. Числа	247
4.2.3. Символы	248
4.2.4. Строки	248
4.3. Типы данных	248

4.3.1. Простые типы	248
4.3.2. Сложные типы	249
4.3.3. Описание простых типов	250
4.4. Операторы VHDL	261
4.4.1. Основы синтаксиса	261
4.4.2. Объекты	262
4.4.3. Атрибуты	263
4.4.4. Компоненты	263
4.4.5. Выражения	264
4.4.6. Операторы	265
4.5. Интерфейс и тело объекта	266
4.5.1. Описание простого объекта	270
4.5.2. Объявление объекта проекта F	270
4.5.3. Поведенческое описание архитектуры	270
4.5.4. Поточковая форма	271
4.5.5. Структурное описание архитектуры	273
4.6. Описание конфигурации	274
4.7. Векторные сигналы и регулярные структуры	275
4.8. Задержки сигналов и параметры настройки	277
4.9. Атрибуты сигналов и контроль запрещенных ситуаций	279
4.10. Алфавит моделирования и пакеты	280
4.11. Описание монтажного «ИЛИ» и общей шины	283
4.12. Синтезируемое подмножество VHDL	285
4.12.1. Общие сведения	285
4.12.2. Переопределенные типы (Redefined types)	286
4.12.3. Методика верификации синтезируемого описания (Verification methodology)	286
4.12.4. Моделирование элементов аппаратуры (Modeling hardware elements)	288
4.12.5. Директивы компилятора (псевдокомментарии, Pragmas)	295
4.12.6. Синтаксис синтезируемого подмножества VHDL	297
4.13. Краткое описание синтаксиса синтезируемого подмножества VHDL	343
ГЛАВА 5. ЯЗЫК ОПИСАНИЯ АППАРАТУРЫ VERILOG HDL	363
5.1. Общие сведения	363
5.2. Операторы	365
5.3. Числа в Verilog	365
5.3.1. Целые числа (Integers)	365
5.3.2. Неопределенное и высокоимпедансное состояния (x and z values)	366

5.3.3. Отрицательные числа (Negative numbers)	366
5.3.4. Подчеркивание (Underscore)	366
5.3.5. Действительные числа (Real)	366
5.3.6. Строки (Strings)	367
5.4. Цепи в Verilog (Nets)	367
5.5. Регистры (Registers)	367
5.6. Векторы (Vectors)	369
5.7. Массивы (Arrays)	370
5.8. Регистровые файлы (Memories)	370
5.9. Элементы с третьим состоянием (Tri-state)	370
5.10. Арифметические операторы (Arithmetic operators)	372
5.11. Логические операторы (Logical operators)	373
5.12. Операторы отношения (Relational operators)	374
5.13. Операторы эквивалентности (Equality)	374
5.14. Поразрядные операторы (Bitwise operators)	375
5.15. Операторы приведения (Reduction operator)	376
5.16. Операторы сдвига (Shift operator)	377
5.17. Конкатенация (объединение, Concatenation)	377
5.18. Повторение (Replication)	378
5.19. Системные директивы (System tasks)	378
5.19.1. Директивы вывода результатов моделирования (Writing to standard output)	379
5.19.2. Контроль процесса моделирования (Monitoring a simulation)	380
5.19.3. Окончание моделирования (Ending a simulation)	382
5.20. Проектирование комбинационных схем, пример проектирования мультиплексора 4 в 1	382
5.20.1. Реализация на уровне логических вентилях (Gate level implementation)	382
5.20.2. Реализация мультиплексора с помощью логических операторов (Logic statement Implementation)	384
5.20.3. Реализация с помощью оператора выбора (CASE statement implementation)	385
5.20.4. Реализация с использованием условного оператора (Conditional operator Implementation)	387
5.20.5. Тестовый модуль (The stimulus module)	387
5.21. Модули проекта (Design blocks modules)	390
5.21.1. Тестирование	392
5.22. Порты (Ports)	393
5.23. Правила соединения (Connection rules)	394

5.23.1. Входы (inputs)	394
5.23.2. Выходы (outputs)	394
5.23.3. Двухнаправленные выходы (inouts)	394
5.23.4. Соответствие портов (Port matching)	394
5.23.5. Присоединение портов (Connecting ports)	394
5.24. Базовые блоки (Basic blocks)	395
5.24.1. Инициализация (Initial block)	395
5.24.2. Конструкция Always (Always block)	395
5.25. Пример проектирования последовательностного устройства: двоичный счетчик	396
5.25.1. Поведенческая модель счетчика (Behavioural model)	400
5.26. Временной контроль (Timing Control)	402
5.26.1. Задержки (delay)	402
5.26.2. Событийный контроль (event-based control)	403
5.27. Защелкивание (triggers)	403
5.28. Список сигналов возбуждения (sensitivity list)	404
5.29. Задержка распространения в вентиле (Gate delays)	404
5.30. Операторы ветвления (Branch statements)	404
5.30.1. Оператор IF (IF statement)	404
5.30.2. Оператор выбора (CASE statement)	406
5.30.3. Оператор ветвления (Conditional operator)	406
5.31. Циклы (Looping constructs)	407
5.31.1. Цикл WHILE (WHILE LOOP)	407
5.31.2. Цикл FOR (FOR LOOP)	408
5.31.3. Цикл REPEAT (REPEAT LOOP)	408
5.31.4. Вечный цикл (FOREVER LOOP)	409
5.32. Файлы в Verilog	409
5.32.1. Открытие файла (Opening a file)	409
5.32.2. Запись в файл (Writing to a file)	410
5.32.3. Закрытие файла (Closing a file)	410
5.32.4. Инициализация регистровых файлов (памяти) (Initialising memories)	410
5.33. Задание векторов входных сигналов для моделирования (Verilog input vectors)	412
5.34. Список операторов Verilog	414
5.35. Приоритет операторов	415
5.36. Ключевые слова (keywords)	415
5.37. Директивы компилятора	416
5.38. Типы цепей (Net types)	416

ГЛАВА 6. ПРИМЕРЫ ПРОЕКТИРОВАНИЯ ЦИФРОВЫХ УСТРОЙСТВ С ИСПОЛЬЗОВАНИЕМ ЯЗЫКОВ ОПИСАНИЯ АППАРАТУРЫ VHDL И VERILOG	417
6.1. Общие сведения	417
6.2. Триггеры и регистры	418
6.2.1. Триггеры, тактируемые передним фронтом (Rising Edge Flipflop)	419
6.2.2. Триггеры, тактируемые передним фронтом, с асинхронным сбросом (Rising Edge Flipflop with Asynchronous Reset) . . .	420
6.2.3. Триггеры, тактируемые передним фронтом, с асинхронной предустановкой (Rising Edge Flipflop with Asynchronous Preset)	421
6.2.4. Триггеры, тактируемые передним фронтом, с асинхронным сбросом и предустановкой (Rising Edge Flipflop with Asynchronous Reset and Preset)	422
6.2.5. Триггеры, тактируемые передним фронтом, с синхронным сбросом (Rising Edge Flipflop with Synchronous Reset) . . .	424
6.2.6. Триггеры, тактируемые передним фронтом, с синхронной предустановкой (Rising Edge Flipflop with Synchronous Preset)	425
6.2.7. Триггеры, тактируемые передним фронтом, с асинхронным сбросом и разрешением тактового сигнала (Rising Edge Flipflop with Asynchronous Reset and Clock Enable)	426
6.2.8. Защелка с разрешением выхода (D-Latch with Data and Enable)	427
6.2.9. Защелка с входом данных с разрешением (D-Latch with Gated Asynchronous Data)	428
6.2.10. Защелка с входом разрешения (D-Latch with Gated enable) . .	429
6.2.11. Защелка с асинхронным сбросом (D-Latch with Asynchronous Reset)	430
6.3. Построение устройств потоковой обработки данных (Datapath logic)	431
6.4. Счетчики	439
6.5. Арифметические устройства	443
6.6. Конечные автоматы (Finite state machine)	449
6.7. Элементы ввода-вывода	459
6.8. Параметризация	464
6.9. Специфика проектирования устройств с учетом архитектурных особенностей ПЛИС	466
6.10. Совместное использование ресурсов	468

6.11. Дублирование регистра	473
6.12. Создание описаний с учетом особенностей архитектуры ПЛИС (Technology Specific Coding Techniques)	476

ГЛАВА 7. ПРИМЕРЫ РЕАЛИЗАЦИИ АЛГОРИТМОВ ЦОС НА ПЛИС

485

7.1. Реализация цифровых фильтров на ПЛИС семейства FLEX фирмы «Altera»	485
7.2. Реализация цифровых полиномиальных фильтров	491
7.3. Алгоритмы функционирования и структурные схемы демодуляторов	495
7.4. Реализация генератора ПСП на ПЛИС	500
7.5. Примеры описания цифровых схем на VHDL	506
7.6. Реализация нейрона на AHDL	516
7.7. Построение быстродействующих перемножителей	529

Приложение 1. Система проектирования Quartus	535
---	-----

Приложение 2. Интерфейсы передачи данных и сопряжение устройств	537
---	-----

Приложение 3. Практические рекомендации по разработке печатных плат	568
---	-----

Литература	572
-------------------------	-----

ВВЕДЕНИЕ

Идея написания этой книги назревала в течение последних двух-трех лет, когда для многих разработчиков аппаратуры центральной оперативной системы (ЦОС) стало ясно, что программируемые логические интегральные схемы (ПЛИС) — удобная в освоении и применении элементная база, альтернативы которой зачастую не найти. Последние годы характеризуются резким ростом плотности упаковки элементов на кристалле, многие ведущие производители либо начали серийное производство, либо анонсировали ПЛИС с эквивалентной емкостью более 1 миллиона логических вентилей. Цены на ПЛИС (к сожалению, только лишь в долларовом эквиваленте) неуклонно падают. Так, еще год-полтора назад ПЛИС логической емкостью 100000 вентилей стоила в Москве в зависимости от производителя, приемки, быстродействия от 1500 до 3000 у.е., сейчас такая микросхема стоит от 100 до 350 у.е., т.е. цены упали практически на порядок, и эта тенденция устойчива. Что касается ПЛИС емкостью 10000...30000 логических вентилей, то появились микросхемы стоимостью менее 10 у.е.

Такая ситуация на рынке вызвала волну вопросов, связанных с подготовкой специалистов, способных проводить разработку аппаратуры цифровой обработки сигналов на ПЛИС, владеющих основными методами проектирования, ориентирующихся в современной элементной базе и программном обеспечении. Идя навстречу многочисленным пожеланиям предприятий, заинтересованных в подготовке молодых специалистов, владеющих современными технологиями, на кафедре СМ-5 «Автономные информационные и управляющие системы» МГТУ им. Н.Э. Баумана в программу четырехсеместрового курса «Схемотехническое проектирование микроэлектронных устройств» включен семестровый раздел «Проектирование аппаратуры обработки сигналов на ПЛИС», на основе лекционных и семинарских материалов которого и выходит этот цикл статей.

Приведем известную классификацию ПЛИС [1, 2, 3] по структурному признаку, так как она дает наиболее полное представление о классе задач, пригодных для решения на той или иной ПЛИС. Следует заметить, что общепринятой оценкой логической емкости ПЛИС является число эквивалентных вентилях, определяемое как среднее число вентилях «2И-НЕ», необходимых для реализации эквивалентного проекта на ПЛИС и базовом матричном кристалле (БМК). Понятно, что эта оценка весьма условна, поскольку ПЛИС не содержат вентилях «2И-НЕ» в чистом виде, однако для проведения сравнительного анализа различных архитектур она вполне пригодна. Основными критериями такой классификации являются наличие, вид и способы коммутации элементов логических матриц. По этому признаку можно выделить следующие классы ПЛИС.

Программируемые логические матрицы — наиболее традиционный тип ПЛИС, имеющий программируемые матрицы «И» и «ИЛИ». В зарубежной литературе соответствующими этому классу аббревиатурами являются FPLA (Field Programmable Logic Array) и FPLS (Field Programmable Logic Sequencers). Примерами таких ПЛИС могут служить отечественные схемы К556РТ1, РТ2, РТ21. Недостаток такой архитектуры — слабое использование ресурсов программируемой матрицы «ИЛИ», поэтому дальнейшее развитие получили микросхемы, построенные по архитектуре программируемой матричной логики (ПМЛ — Programmable Array Logic, PAL) — это ПЛИС, имеющие программируемую матрицу «И» и фиксированную матрицу «ИЛИ». К этому классу относится большинство современных ПЛИС небольшой степени интеграции. В качестве примеров можно привести отечественные интегральные схемы (ИС) КМ1556ХП4, ХП6, ХП8, ХЛ8, ранние разработки (середина—конец 80-х годов) ПЛИС фирм «Intel», «Altera», «AMD», «Lattice» и др. Разновидностью класса ПМЛ являются ПЛИС, имеющие только одну (программируемую) матрицу «И», например схема 85С508 фирмы «Intel».

Следующий традиционный тип ПЛИС — программируемая макрологика. Она содержит единственную программируемую матрицу «И-НЕ» или «ИЛИ-НЕ», но за счет многочисленных инверсных обратных связей способна формировать сложные логические функции. К этому классу относятся, например, ПЛИС PLHS501 и PLHS502 фирмы «Signetics», имеющие матрицу «И-НЕ», а также схема XL78С800 фирмы «Exel», основанная на матрице «ИЛИ-НЕ».

Вышеперечисленные архитектуры ПЛИС содержат небольшое число ячеек, к настоящему времени морально устарели и применяются для реа-

лизации относительно простых устройств, для которых не существует готовых ИС средней степени интеграции. Естественно, для реализации алгоритмов ЦОС они не пригодны.

ИС ПМЛ (Programmable Logic Device, PLD) имеют архитектуру, весьма удобную для реализации цифровых автоматов. Развитие этой архитектуры — программируемые коммутируемые матричные блоки (ПКМБ) — это ПЛИС, содержащие несколько матричных логических блоков (МЛБ), объединенных коммутационной матрицей. Каждый МЛБ представляет собой структуру типа ПМЛ, т.е. программируемую матрицу «И», фиксированную матрицу «ИЛИ» и макроячейки. ПЛИС типа ПКМБ, как правило, имеют высокую степень интеграции (до 10000 эквивалентных вентилях, до 256 макроячеек). К этому классу относятся ПЛИС семейства MAX5000 и MAX7000 фирмы «Altera», схемы XC7000 и XC9500 фирмы «Xilinx», а также большое число микросхем других производителей (фирмы «Atmel», «Vantis», «Lucent» и др.). В зарубежной литературе они получили название Complex Programmable Logic Devices (CPLD).

Другой тип архитектуры ПЛИС — программируемые вентиляльные матрицы (ПВМ), состоящие из логических блоков (ЛБ) и коммутирующих связей — программируемых матриц соединений. Логические блоки таких ПЛИС состоят из одного или нескольких относительно простых логических элементов, в их основе лежит таблица перекодировки (ТП — Look-up table, LUT), программируемый мультиплексор, D-триггер, а также цепи управления. Таких простых элементов может быть достаточно большое количество, у современных ПЛИС емкостью до 1 миллиона вентилях число логических элементов достигает нескольких десятков тысяч. За счет такого большого числа логических элементов они содержат значительное число триггеров, также некоторые семейства ПЛИС имеют встроенные реконфигурируемые модули памяти (РМП — Embedded Array Block, EAB), что делает ПЛИС данной архитектуры весьма удобным средством реализации алгоритмов цифровой обработки сигналов, основными операциями в которых являются перемножение, умножение на константу, суммирование и задержка сигнала. Вместе с тем возможности комбинационной части таких ПЛИС ограничены, поэтому совместно с ПВМ применяют ПКМБ (CPLD) для реализации управляющих и интерфейсных схем. В зарубежной литературе такие ПЛИС получили название Field Programmable Gate Array (FPGA). К FPGA (ПВМ) классу относятся ПЛИС XC2000, XC3000, XC4000, Spartan, Virtex фирмы «Xilinx», ACT1, ACT2 фирмы

«Actel», а также семейства FLEX8000 фирмы «Altera», некоторые ПЛИС фирм «Atmel» и «Vantis».

Множество конфигурируемых логических блоков (Configurable Logic Blocks, CLB) объединяются с помощью матрицы соединений. Характерными для FPGA архитектур являются элементы ввода-вывода (Input/Output Blocks, IOBs), позволяющие реализовать двунаправленный ввод/вывод, третье состояние и т.п.

Особенностью современных ПЛИС является возможность тестирования узлов с помощью порта JTAG (B-scan), а также наличие внутреннего генератора (Osc) и схем управления последовательной конфигурацией.

Фирма «Altera» пошла по пути развития FPGA-архитектур и предложила в семействе FLEX10K так называемую двухуровневую архитектуру матрицы соединений.

Логические элементы (ЛЭ) объединяются в группы — логические блоки (ЛБ). Внутри логических блоков ЛЭ соединяются посредством локальной программируемой матрицы соединений, позволяющей соединять любой ЛЭ с любым другим. Логические блоки связаны между собой и с элементами ввода-вывода посредством глобальной программируемой матрицы соединений (ГПМС). Локальная и глобальная матрицы соединений имеют непрерывную структуру — для каждого соединения выделяется непрерывный канал.

Дальнейшее развитие архитектур идет по пути создания комбинированных архитектур, сочетающих удобство реализации алгоритмов ЦОС на базе таблиц перекодировок и реконфигурируемых модулей памяти, характерных для FPGA-структур и многоуровневых ПЛИС с удобством реализации цифровых автоматов на CPLD-архитектурах. Так, ПЛИС APEX20K фирмы «Altera» содержат в себе логические элементы всех перечисленных типов, что позволяет применять ПЛИС как основную элементную базу для «систем на кристалле» (System-On-Chip, SOC). В основе идеи SOC лежит интеграция всей электронной системы в одном кристалле (например, в случае персонального компьютера (ПК) такой чип объединяет процессор, память, и т.д.). Компоненты этих систем разрабатываются отдельно и хранятся в виде файлов параметризуемых модулей. Окончательная структура SOC-микросхемы выполняется на базе этих «виртуальных компонентов» с помощью программ систем автоматизации проектирования (САПР) электронных устройств — EDA (Electronic Design Automation). Благодаря стандартизации в одно целое можно объединять «виртуальные компоненты» от разных разработчиков.

Как известно, при выборе элементной базы систем обработки сигналов обычно руководствуются следующими критериями отбора:

- быстродействие;
- логическая емкость, достаточная для реализации алгоритма;
- схемотехнические и конструктивные параметры ПЛИС, надежность, рабочий диапазон температур, стойкость к ионизирующим излучениям и т.п.;
- стоимость владения средствами разработки, включающая как стоимость программного обеспечения, так наличие и стоимость аппаратных средств отладки;
- стоимость оборудования для программирования ПЛИС или конфигурационных ПЗУ (постоянное запоминающее устройство);
- наличие методической и технической поддержки;
- наличие и надежность российских поставщиков;
- стоимость микросхем.

В данной книге рассматриваются вопросы проектирования устройств обработки информации на базе ПЛИС фирмы «Altera».

Фирма «Altera Corporation» (101 Innovation Drive, San Jose, CA 95134, USA, www.altera.com) была основана в июне 1983 года. В настоящее время последним достижением этой фирмы является семейство APEX20K.

Кроме того, фирма «Altera» выпускает CPLD семейств MAX3000, MAX7000, MAX9000 (устаревшие серии специально не упоминаются), FPGA семейств FLEX10K, FLEX8000, FLEX6000.

Дополнительным фактором при выборе ПЛИС фирмы «Altera» является наличие достаточно развитых бесплатных версий САПР. В Табл. В.1 приведены основные характеристики пакета MAX+PLUS II BASELINE версия 9.3 фирмы «Altera», который можно бесплатно «скачать» с сайта www.altera.com или получить на CD «Altera Digital Library», на котором содержится также и полный набор документации по архитектуре и применению ПЛИС.

Кроме того, ПЛИС фирмы «Altera» выпускаются с возможностью программирования в системе непосредственно на плате. Для программирования и загрузки конфигурации устройств опубликована схема загрузочного кабеля ByteBlaster и ByteBlasterMV. Следует отметить, что новые конфигурационные ПЗУ EPC2 и EPC16 позволяют программирование с помощью этого устройства, тем самым отпадает нужда в программаторе, что, естественно, снижает стоимость владения технологией.

ПЛИС фирмы «Altera» выпускаются в коммерческом и промышленном диапазоне температур.

Таблица В.1. Основные характеристики пакета MAX+PLUS II BASELINE
версии 9.3

Поддерживаемые устройства	ACEX, EPF10K10, EPF10K10A, EPF10K20, EPF10K30, EPF10K30A, EPF10K30E (до 30000 эквивалентных вентилях), EPM9320, EPM9320A, EPF8452A, EPF8282A, MAX7000, FLEX6000, MAX5000, MAX3000A, Classic
Средства описания проекта	Схемный ввод, поддержка AHDL, средства интерфейса с САПР третьих фирм, топологический редактор, иерархическая структура проекта, наличие библиотеки параметризуемых модулей
Средства компиляции проекта	Логический синтез и трассировка, автоматическое обнаружение ошибок, поддержка мегафункций по программам MegaCore и AMPP
Средства верификации проекта	Временной анализ, функциональное и временное моделирование, анализ сигналов, возможность использования программ моделирования (симуляторов) третьих фирм

Глава 1. Элементная база

1.1. Семейство MAX3000

Летом 1999 года на рынке стали доступны ПЛИС семейства MAX3000. Их архитектура близка к архитектуре семейства MAX7000, однако имеется ряд небольших отличий. В **Табл. 1.1** приведены основные параметры ПЛИС.

Таблица 1.1. Основные параметры ПЛИС семейства MAX3000

Параметр	ЕРМ3032А	ЕРМ3064А	ЕРМ3128А	ЕРМ3256А
Логическая емкость, количество эквивалентных вентилях	600	1 250	2 500	5 000
Число макроячеек	32	64	128	256
Число логических блоков	2	4	8	16
Число программируемых пользователем выводов	34	66	96	158
Задержка распространения сигнала вход/выход, t_{PD} [нс]	4.5	4.5	5	6
Время установки глобального тактового сигнала, t_{SU} [нс]	3.0	3.0	3.2	3.7
Задержка глобального тактового сигнала до выхода, t_{CO1} [нс]	2.8	2.8	3.0	3.3
Максимальная глобальная тактовая частота, f_{CNT} [МГц]	192.3	192.3	181.8	156.3

Микросхемы семейства MAX3000 выполнены по КМОП EPROM 0.35-микронной технологии, что позволило существенно удешевить их по сравнению с семейством MAX7000S. Все ПЛИС семейства

MAX3000 поддерживают технологию программирования в системе ISP (In-System Programmability) и периферийного сканирования (boundary scan) в соответствии со стандартом IEEE Std. 1149.1 — 1990 (JTAG). Элементы ввода/вывода (ЭВВ) позволяют работать в системах с уровнями сигналов 5, 3.3 и 2.5 В. Матрица соединений имеет непрерывную структуру, что позволяет реализовать время задержки распространения сигнала до 4.5 нс. ПЛИС семейства MAX3000 имеют возможность аппаратной эмуляции выходов с открытым коллектором (open drains pin) и удовлетворяют требованиям стандарта PCI. Имеется возможность индивидуального программирования цепей сброса, установки и тактирования триггеров, входящих в макроячейку. Предусмотрен режим пониженного энергопотребления. Программируемый логический расширитель позволяет реализовать на одной макроячейке функции до 32 переменных. Имеется возможность задания бита секретности (security bit) для защиты от несанкционированного тиражирования разработки.

Реализация функции программирования в системе поддерживается с использованием стандартных средств загрузки, таких, как ByteBlasterMV, BitBlaster, MasterBlaster, а также поддерживается формат JAM.

ПЛИС семейства MAX3000 выпускаются в корпусах от 44 до 208 выводов.

На **Рис. 1.1** представлена функциональная схема ПЛИС семейства MAX3000. Основными элементами структуры ПЛИС семейства MAX3000 являются:

- логические блоки (ЛБ, Logic array blocks, LAB);
- макроячейки (МЯ, macrocells);
- логические расширители (expanders) (параллельный (parallel) и разделяемый (shareble));
- программируемая матрица соединений (ПМС, Programmable Interconnect Array, PIA);
- элементы ввода/вывода (ЭВВ, I/O control block).

ПЛИС семейства MAX3000 имеют четыре вывода, закрепленных за глобальными цепями (dedicated inputs). Это глобальные цепи синхронизации сброса и установки в третье состояние каждой макроячейки. Кроме того, эти выводы можно использовать как входы или выходы пользователя для «быстрых» сигналов, обрабатываемых в ПЛИС.



Рис.1.1. Функциональная схема ПЛИС семейства МАХ3000

Как видно из **Рис. 1.1**, в основе архитектуры ПЛИС семейства MAX3000 лежат логические блоки, состоящие из 16 макроячеек каждый. Логические блоки соединяются с помощью программируемой матрицы соединений. Каждый логический блок имеет 36 входов с ПМС.

На **Рис. 1.2** приведена структурная схема макроячейки ПЛИС семейства MAX3000.

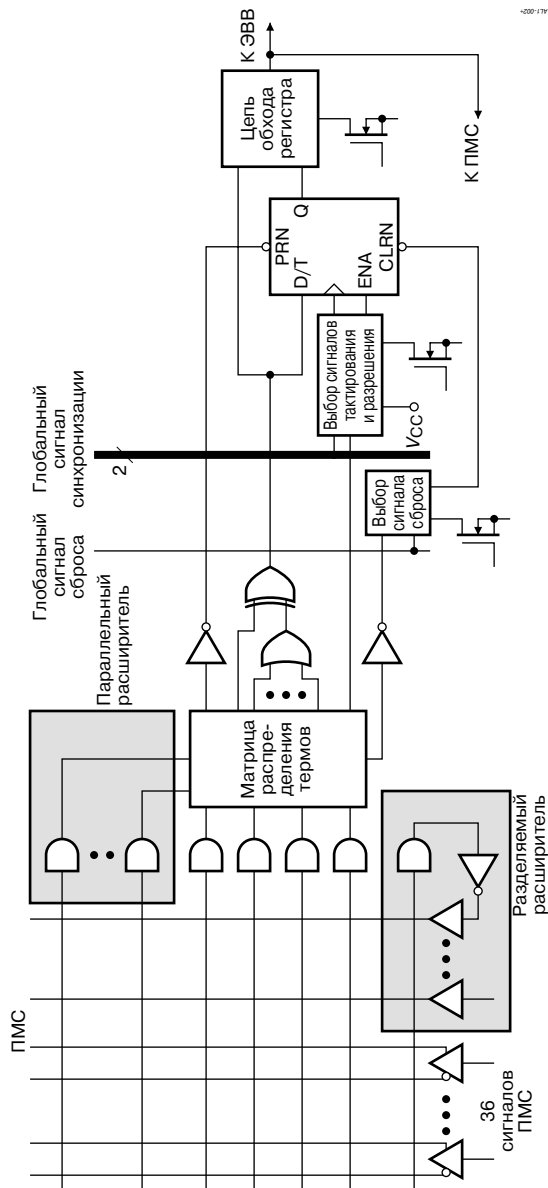


Рис. 1.2. Структурная схема макроячейки ПЛИС семейства MAX3000

Конец ознакомительного фрагмента.

Приобрести книгу можно
в интернет-магазине «Электронный универс»
(e-Univers.ru)