

ReEncoded LdGray

The Design Warrior's Guide to FPGA's

Clive «Max» Maxfield

Клайв Максфилд

Проектирование на ПЛИС
Архитектура, средства
и методы

Курс молодого бойца

Перевод с английского

УДК 004.312
ББК 32.844-1
М17

Максфилд К.

М17 Проектирование на ПЛИС. Курс молодого бойца. — М.: Издательский дом «Додэка-XXI», 2007. — 408 с.: илл. (Серия «Программируемые системы»).

ISBN 978-5-94120-147-1

Эта книга является не только пособием по проектированию устройств на основе ПЛИС (FPGA), но и содержит поистине энциклопедические сведения. Кроме архитектурных особенностей последних поколений микросхем ПЛИС, здесь рассматриваются различные методы и средства проектирования. Проводится обзор и анализ схемотехнических подходов к проектированию (которые всё ещё находят применение), HDL-моделирования и логического синтеза, а так же современных технологий проектирования, основанных на использовании языка C/C++. Рассматриваются специализированные вопросы, такие как совместное проектирование программно-аппаратных систем и разработка систем цифровой обработки сигналов (ЦОС). Обсуждаются и технические новинки, например программируемые пользователем массивы узлов (FPNA).

Написанная в непринуждённом, увлекательном стиле, книга будет хорошим пособием и для начинающих, и для опытных инженеров, разрабатывающих устройства на основе ПЛИС. Книга послужит весьма ценным источником информации и для специалистов, разрабатывающих устройства на основе заказных микросхем и переходящих на использование ПЛИС. А также, несомненно, привлечет внимание широкого круга читателей, в том числе технических аналитиков, студентов и продавцов технической продукции.

УДК 004.312
ББК 32.844-1

Максфилд Клайв

**Проектирование на ПЛИС. Архитектура, средства и методы
Курс молодого бойца**

Переводчик *В. М. Барская*

Главный редактор *В. М. Халикеев*

Ответственный редактор *И. А. Корабельникова*

Научный редактор *В. Н. Немцев*

Технический редактор *А. И. Михалченков*

График *А. Н. Клочков, И. С. Кайнова*

Верстальщик *И. С. Кайнова*

Подписано в печать 15.03.2007. Формат 70x100/16. Бумага офсетная № 1.
Гарнитура «NewtonС». Печать офсетная. Объем 25,5 п.л. Усл. печ. л. 33,1 + вкл. 0,125
Тираж 2000 экз. Код FPGA. Заказ № 625.

Издательский дом «Додэка-XXI»
ОКП 95 3000

105318 Москва, а/я 70.

Тел./факс: (495) 366-24-29, 366-09-22

E-mail: boor@dodccfaru; red@dodcca.ru

Отпечатано с готовых диапозитивов в ОАО «Московская типография № 6»
115088 Москва, Южнопортовая ул., 24

Настоящее издание «Проектирование на ПЛИС: архитектура, средства и методы. Курс молодого бойца» Клайва Максфилда выполнено по договору с Elsevier Inc., 200 Wheeler Road, 6th Floor, Burlington, MA01803, USA.

ISBN 978-5-94120-147-1 (рус.)

ISBN 0-7506-7604-3 (англ.)

© Издательский дом «Додэка-XXI», 2007

® Серия «Программируемые системы»

ОГЛАВЛЕНИЕ

Благодарности	15
Предисловие	16
Глава 1. Введение	18
Что такое ПЛИС	18
Чем интересны ПЛИС	18
Как можно использовать ПЛИС	20
Что есть в этой книге	21
Чего нет в этой книге	22
Кому предназначается эта книга	23
Глава 2. Основные понятия	24
Особенность микросхем ПЛИС	24
Простая программируемая функция	24
Метод плавких перемычек	25
Метод наращиваемых перемычек	26
Устройства, программируемые фотошаблоном	28
ППЗУ	29
СППЗУ	30
ЭСППЗУ	32
Flash-технология	32
Статическое ОЗУ	33
Заключение	34
Глава 3. История развития ПЛИС	35
Родственные технологии	35
Транзисторы	35
Интегральные микросхемы	36
Статическое и динамическое ОЗУ, микропроцессоры	37
Простые и сложные ПЛУ	37
ППЗУ	38
Программируемые логические матрицы (ПЛМ)	41
Программируемые матрицы PAL и GAL	43
Дополнительные программируемые опции	43
Сложные ПЛУ	44
Программные средства ABEL, CUPL, PALASM, JEDEC и другие	46
Заказные интегральные схемы — ASIC	47
Заказные интегральные микросхемы	48
Микроматрица и микромозаика	49
Вентильные матрицы	49
Схемы на стандартных элементах	51
Структурированные специализированные микросхемы и структурированные ASIC	52
ПЛИС	53
ПЛИС-платформа	56
Гибриды вида ПЛИС — заказные интегральные схемы	57
Как создают ПЛИС	58

Глава 4. Архитектура ПЛИС	60
Предупреждение	60
Небольшое введение	60
Нарастиваемые перемычки, статическое ОЗУ и прочее... ..	61
Устройства на основе статического ОЗУ	61
Защита интеллектуальной собственности в устройствах на основе статического ОЗУ	62
Устройства на основе нарастиваемых перемычек	63
Устройства на основе СППЗУ	65
Устройства на основе ЭСППЗУ и Flash	65
Гибридные устройства на ячейках Flash и статического ОЗУ	66
Выводы	67
Мелко-, средне- и крупномодульные архитектуры	67
Логические блоки на мультиплексорах и таблицах соответствия	69
Устройства на основе мультиплексоров	69
Устройства на основе таблиц соответствия	69
Мультиплексоры или таблицы соответствия?	71
3-, 4-, 5- или 6-входные таблицы соответствия	71
Таблицы соответствия, распределенное ОЗУ, сдвиговые регистры	72
Конфигурируемые логические блоки, блоки логических массивов, секции	73
Логические ячейки фирмы Xilinx	73
Логические элементы компании Altera	74
Секции и логические ячейки	74
Конфигурируемые логические блоки CLB и блоки логических массивов LAB	75
Распределенное ОЗУ и сдвиговые регистры	76
Схемы ускоренного переноса	76
Встроенные блоки ОЗУ	77
Встроенные умножители, сумматоры, блоки умножения с накоплением и др.	77
Аппаратные и программные встроенные микропроцессорные ядра	79
Аппаратные микропроцессорные ядра	79
Программные микропроцессорные ядра	81
Дерево синхронизации и диспетчеры синхронизации	81
Дерево синхронизации	81
Диспетчер синхронизации	82
Ввод/вывод общего назначения	85
Конфигурируемые стандарты ввода/вывода	86
Согласование ввода/вывода	86
Напряжение ядра и напряжение ввода/вывода	87
Гигабитные приёмопередатчики	87
IP — блоки аппаратной, программной и микропрограммной интеллектуальной собственности	88
Системный вентиль и реальный вентиль	90
Возраст ПЛИС	92
Глава 5. Программирование или конфигурирование ПЛИС	93
Ласковые слова	93
Конфигурационные файлы и прочее	93
Конфигурационные ячейки	93
ПЛИС на нарастиваемых перемычках	95
ПЛИС на ячейках статического ОЗУ	95
Ловкость рук и никакого мошенства	96
Программирование встроенных блоков ОЗУ, распределенного ОЗУ и других ОЗУ	97
Мультипрограммирование конфигурационных цепочек	98
Быстрая реинициализация устройства	98
Конфигурационный порт	98
Последовательная загрузка, ПЛИС в режиме ведущий	99
Параллельная загрузка, ПЛИС в режиме ведущий	100
Параллельная загрузка, ПЛИС в режиме ведомый	102
Последовательная загрузка, ПЛИС в режиме ведомый	102

JTAG-порт	103
Встроенные процессоры	104
Глава 6. Ведущие производители	106
Введение	106
Поставщики FPGA и FPAА	106
Поставщики устройств FPNA	107
Поставщики САПР для полного цикла разработки ПЛИС	107
Специалисты по ПЛИС и независимые разработчики САПР	108
Консультанты по разработке ПЛИС и их средства проектирования	109
Открытые, недорогие и бесплатные системы проектирования	109
Глава 7. Различные подходы к проектированию систем на основе ПЛИС и заказных микросхем	111
Введение	111
Подходы к проектированию	111
Конвейеры и уровни логики	112
Что такое конвейер	112
Конвейер в электронных системах	112
Логические уровни	114
Метод асинхронного проектирования	115
Асинхронные элементы	115
Комбинационные петли	115
Элементы задержки	115
Анализ систем синхронизации	115
Зоны синхронизации	115
Выравнивание тактовых сигналов	116
Стробирование и разрешение тактовых сигналов	116
ФАПЧ и схема согласования тактовых сигналов	116
Достоверность передачи данных через разные зоны синхронизации	117
Регистры и защелки	117
Защелки	117
Триггеры с входами установки и сброса	117
Общий сброс и исходное состояние	117
Разделение ресурсов или разделение по времени	117
Использовать или потерять!	118
Подождите, ещё не вечер!	118
Кодирование конечных автоматов	119
Методики тестирования	119
Глава 8. Схемотехническое проектирование	120
Давным-давно	120
Развитие САПР электронных систем	121
Начальный этап проектирования. Логическое моделирование	121
Завершающий этап проектирования. Компоновка	125
САПР электронных систем	126
Простой метод схемотехнического проектирования заказных ИС	126
Простой метод схемотехнического проектирования ПЛИС	128
Сопоставление	129
Компоновка	129
Размещение и разводка	130
Временной анализ и повторное моделирование	131
Одноуровневые и иерархические принципиальные схемы	132
Одноуровневые принципиальные схемы	132
Иерархические принципиальные схемы	133
Современная последовательность схемотехнического проектирования ПЛИС	134
Глава 9. Проектирование на основе языков описания аппаратных средств	136
Закат схемотехнического проектирования	136
История развития HDL-проектирования	136

Уровни абстракции	136
Простые HDL-методы проектирования заказных микросхем	138
Простые HDL-методы проектирования ПЛИС	140
Архитектурное проектирование ПЛИС	141
Логический и физический синтез	141
Средства графического ввода продолжают жить	142
Типы языков HDL	143
Verilog HDL	144
VHDL и VITAL	146
Проектирование с применением нескольких языков	148
UDL/I	148
Superlog и SystemVerilog	148
SystemC	149
Информация к размышлению	150
Ужасы проектирования	150
Последовательные и параллельные мультиплексоры	150
Остерегайтесь зашёлков!	151
Рационально используйте константы	152
Распределение ресурсов	152
Последнее, но не менее важное	153
Глава 10. Виртуальное макетирование ПЛИС	154
Общие сведения о виртуальном макетировании	154
Применение виртуальных прототипов при проектировании заказных микросхем	154
Виртуальные прототипы на уровне вентилях. Быстрый и грубый синтез	155
Виртуальные прототипы на уровне вентилях. Синтез, оптимизированный по быстродействию	156
Виртуальные прототипы на уровне кластеров	157
Виртуальные прототипы на основе RTL	158
Виртуальные прототипы ПЛИС	160
Интерактивная правка	161
Поэтапные размещение и разводка	162
Виртуальные прототипы ПЛИС уровня регистровых передач	163
Глава 11. Проектирование на основе C/C++ и других языков	164
Проблемы использования HDL-языков	164
C, C++ и другие версии	166
Проектирование на языке SystemC	167
Что такое SystemC и его история	167
SystemC 1.0	168
SystemC 2.0	168
Уровни абстракции	170
Альтернативные методы проектирования на основе языка SystemC	170
Принять или не принять	172
Проектирование на расширенном C/C++	173
Что такое расширенный C/C++	173
Расширенный стандарт языка C/C++	175
Проектирование на немодифицированном C/C++	176
Уровни абстракций синтеза	179
Системы мультиязычного проектирования и тестирования	180
Глава 12. Проектирование средств цифровой обработки сигналов	182
Введение в ЦОС	182
Альтернативы реализации ЦОС	182
Выбрать любое устройство, но мне его не показывать	182
Оценки системного уровня и алгоритмическая верификация	183
Работа программного обеспечения на ядре ЦСП	184
Специализированное аппаратное обеспечение ЦОС	185
Другие встраиваемые в ПЛИС ресурсы ЦОС	187

Разработка ПЛИС для ЦОС	188
Специализированные языки	188
Системы проектирования и моделирования на системном уровне	190
Модели с фиксированной и плавающей точкой	190
Перевод из системного/алгоритмического уровня к RTL. Ручной способ	191
Перевод из системного/алгоритмического уровня к RTL. Автоматический способ	192
Перевод из системного/алгоритмического уровня в код C/C++ и другие	194
Блоки интеллектуальной собственности	195
Не забудьте о средствах тестирования!	196
Смешанные системы проектирования: ЦОС и VHDL/Verilog	196
Глава 13. Проектирование устройств со встроенными микропроцессорами	198
Введение	198
Аппаратные и программные ядра	200
Аппаратные ядра	200
Программные микропроцессорные ядра	201
Разделение устройства на аппаратные и программные компоненты	203
Аппаратное и программное мировоззрение	204
ПЛИС как среда проектирования	205
Контроль состояния устройства	206
Некоторые альтернативные методы совместной проверки	207
RTL-описания: VHDL или Verilog	208
C/C++, SystemC и т. д.	208
Реальные микросхемы в аппаратных устройствах моделирования	209
Эмулятор машинных команд	209
Очень интересная среда проектирования	211
Глава 14. Модульное и пошаговое проектирование	213
Работа с одним целым	213
Разбиение на меньшие блоки	214
Модульное проектирование	215
Пошаговое проектирование	216
Отрицательные стороны	216
Всегда есть выход	217
Глава 15. Быстродействующие схемы и некоторые соображения по поводу печатных плат	218
Предисловие	218
Когда мы были молодыми	218
Время перемен	219
FPGA Xchange	221
Другие элементы проектирования	221
Быстродействующие схемы	221
Анализ искажений сигналов	222
SPICE и IBIS	222
Стартовая мощность потребления	223
Применение внутренней согласованной нагрузки	223
Параллельная и последовательная передача данных	224
Глава 16. Отслеживание состояния внутренних точек схемы в ПЛИС	225
Отсутствие наглядности	225
Мультиплексирование	226
Специальные цепи отладки	227
Виртуальный логический анализатор	227
VirtualWires	228
Проблема	229
Решение VirtualWire	230
Глава 17. Блоки интеллектуальной собственности	232
Источники блоков интеллектуальной собственности	232
Ручная работа	232

Незашифрованные RTL-описания	232
Шифрованные RTL-описания	233
Таблица соединений до размещения и разводки	233
Таблица соединений после размещения и разводки	234
Генераторы ядер IP	234
Разное	235
Глава 18. Переход от заказных микросхем к ПЛИС и наоборот	236
Сценарии проектирования	236
Только ПЛИС	236
От ПЛИС к ПЛИС	236
От ПЛИС к заказной микросхеме	237
От заказной микросхемы к ПЛИС	239
Глава 19. Средства моделирования, синтеза, верификации и реализации	241
Введение	241
Типы моделирования	241
Событийное моделирование	241
Краткий обзор систем событийного моделирования	243
Логические величины и системы логических величин	245
Моделирование с использованием разных языков	245
Альтернативные форматы описания задержек	246
Системы циклового моделирования	251
Выбор оптимальной системы логического моделирования!	251
Средства синтеза. Логический/HDL и физический синтез	253
Технология логического/HDL-синтеза	253
Технология физического синтеза	253
Коррекция временных параметров, репликация и повторный синтез	254
Выбор оптимального средства синтеза!	256
Временной анализ. Статический и динамический	256
Статический временной анализ	256
Статистический статический временной анализ	257
Динамический временной анализ	258
Общая верификация	259
Верификация блоков интеллектуальной собственности	259
Среда верификации и разработка тестов	261
Анализ результатов моделирования	262
Формальная верификация	262
Особенности формальной верификации	263
Что такое формальная верификация, и чем она хороша	263
Термины и определения	265
Альтернативные методы описания утверждений/свойств	266
Статическая и динамическая формальная верификация	268
Обзор других языков	268
Разное	270
Преобразование из HDL в C	270
Кодовое покрытие	271
Анализ производительности	272
Глава 20. Выбор правильного устройства	273
Такой широкий выбор	273
Главное, чтобы инструмент был	273
Технология изготовления	274
Основные ресурсы и корпус	275
Интерфейсы ввода/вывода общего назначения	276
Встроенные умножители, блоки ОЗУ и т. д.	276
Встроенные микропроцессорные ядра	276
Возможности гигабитного ввода/вывода	277

Блоки интеллектуальной собственности	277
Скоростные показатели	278
На оптимистической ноте	278
Глава 21. Гигабитные приёмопередатчики	279
Введение	279
Дифференциальные пары	280
Многообразие стандартов	282
8- и 10-битное кодирование	283
Погружение в приёмопередатчики	284
Соединение нескольких приёмопередающих блоков	286
Конфигурируемые параметры	286
Определение разделителей	287
Амплитуда выходного сигнала	287
Внутрикристальные согласующие резисторы	288
Внесение предискажений	288
Компенсация	289
Восстановление синхронизации, флуктуация и глазковые диаграммы	289
Восстановление синхронизации	289
Флуктуация и глазковые диаграммы	291
Глава 22. Системы с перестраиваемой архитектурой	293
Динамически реконфигурируемая логика	293
Динамически реконфигурируемые внутренние соединения	294
Системы с перестраиваемой архитектурой	294
Глава 23. FPNA — программируемый пользователем массив узлов	298
Введение	298
Мелко-, средне- и крупномодульные архитектуры	298
Алгоритмическая оценка	300
Технология picoAggra компании picoChip	301
Идеальные приложения для picoAggra: беспроводные базовые станции	301
Среда проектирования picoAggra	302
Технология адаптивных вычислительных машин компании QuickSilver	303
Конфигурирование состава узлов	305
Разновидности узлов	305
Пространственная и временная сегментация	306
Создание и выполнение приложений на АВМ	307
Подождите, ещё не вечер	308
Это кристалл, Джим, но не такой, каким мы его знаем!	308
Глава 24. Средства проектирования независимых разработчиков	309
Введение	309
ParaCore Architect	309
Реализация модулей для обработки данных с плавающей точкой	310
Реализация функций БПФ	310
Web-интерфейс	312
Язык проектирования системы Confluence	312
Простой пример	313
Но подождите, есть кое-что ещё	315
Бесплатная копия	316
У вас есть средство?	316
Глава 25. Проектирование с использованием открытого программного обеспечения	317
Как открыть магазин по продаже устройств на основе ПЛИС без крупных вложений	317
Базовая платформа: Linux	317
Приобретение ОС Linux	320
Среда верификации	320
Icarus Verilog	320
Dinotrace и GTKWave	321

Covered	321
Verilator	321
Python	322
Формальная верификация	322
Верификация модели	323
Автоматизированная формулировка логических выводов	323
В чём заключается проблема?	324
Доступ к общим блокам интеллектуальной собственности	324
OpenCores	325
OVL	325
Средства синтеза и реализации	325
Макетная плата	325
Другие необходимые средства и утилиты	326
Глава 26. Перспективы развития ПЛИС	327
Ужасы проектирования	327
Архитектуры и технологии нового поколения	328
Устройства с миллиардами транзисторов	328
Сверхбыстрые устройства ввода/вывода	328
Сверхскоростное конфигурирование	328
Увеличение количества аппаратных блоков интеллектуальной собственности	330
Аналоговые и комбинированные устройства	330
ASMBL и другие архитектуры	331
Уровень детализации	331
Включение ядер ПЛИС в заказные микросхемы	332
Включение ядер FPNA в заказные микросхемы и ПЛИС и наоборот	332
Устройства на основе магнитного ОЗУ	332
Не забывайте о средствах проектирования	332
Ожидание неожиданного	333
Приложение А. Целостность сигнала	334
Вступление	334
Ёмкостное и индуктивное взаимодействие	334
Эффекты на уровне кристалла	336
Резистивно-ёмкостные эффекты на уровне кристалла	336
Увеличение ёмкостной связи боковых стенок	337
Выбросы, вызванные перекрестными наводками	338
Задержки распространения сигнала	339
Воздействие нескольких проводников-«агрессоров»	340
И не надо забывать про эффект Миллера	341
Эффекты на уровне печатной платы	342
Эффекты индуктивно-ёмкостного характера на уровне печатной платы	342
Разные способы представления проблемы	342
Эффекты взаимной ёмкости и индуктивности	342
Эффект Миллера наоборот	343
Эффекты линии передачи	344
Как можно облегчить себе жизнь	344
Приложение Б. Эффекты задержки в технологии глубокого субмикрона	345
Введение	345
Развитие спецификаций задержки	345
Набор определений	346
Крутизна импульса	346
Входной порог срабатывания	347
Внутренние и внешние задержки	347
Задержки вывод-вывод и точка-точка	348
Зависимость от состояния и зависимость от крутизны	349
Альтернативные модели внутренних соединений	349

Модель с сосредоточенной нагрузкой	349
Распределённая RC-модель	350
Чистая LC-модель	350
RLC-модель	351
Эффекты задержки глубокого субмикрона	351
Зависимость задержки вывод-вывод от пути прохождения сигнала	351
Зависимость задержки вывод-вывод от порога срабатывания	352
Зависимость задержки вывод-вывод от крутизны фронта входного сигнала	353
Зависимость задержки вывод-вывод от состояния вентиля	354
Зависимость выходной характеристики от пути прохождения сигнала	355
Зависимость выходной характеристики от крутизны входного сигнала	356
Зависимость выходной характеристики вентиля от его состояния	356
Зависимость порога срабатывания вентиля от его состояния	357
Зависимость величины паразитных элементов вентиля от его состояния	357
Влияние изменения нескольких сигналов на задержку вывод-вывод	358
Влияние изменения нескольких сигналов на выходную характеристику	359
Паразитное отражение	360
Выводы	360
Приложение В. Линейный сдвиговый регистр с обратной связью	361
Ауроборос	361
Реализация вида «многие к одному»	361
Множество отводов	363
Реализация вида «один ко многим»	365
Инициализация LFSR	366
Очереди FIFO	367
Модификация LFSR для формирования последовательностей длиной $2n$	368
Доступ к предыдущему значению	369
Шифрование и дешифровка	370
Контроль с помощью циклических избыточных кодов	371
Сжатие данных	372
Встроенное самотестирование	373
Генераторы псевдослучайных последовательностей	374
И последнее, но не менее важное	375
Об авторе	376
Словарь	377
Предметный указатель	397

Моей жене Джине, которая подобно радуге, украшает и услаждает мою жизнь.

Также моим приемному сыну Джозефу и внукам Уиллоу, Гайгу, Кигану и Карму, восторг которых будет беспределен при виде своих имен в книге для взрослых!

Благодарности

Я давно хотел написать книгу по программируемым логическим интегральным схемам, и был несказанно рад, когда мой издатель, Кэррол Льюис из научного отдела компании Elsevier (как мне сообщили, это самый крупный мировой англоязычный издательский дом) предоставил мне такую возможность.

Была только одна небольшая проблема. Дело в том, что почти в течение 10 лет я работал все дни напролет, а вечера и выходные посвящал написанию книг. При этом меня посещали мысли о том, что было бы совсем неплохо проводить хотя бы какое-то время в кругу семьи и друзей. Поэтому я страшно обрадовался, когда представители компаний Mentor Graphics и Xilinx предложили свои услуги в качестве спонсоров и, тем самым, дали мне возможность работать над книгой днем и иметь свободные вечера и выходные.

Дальше — больше. Будучи по профессии инженером, я ненавижу иметь дело с книгой, которая по своему содержанию претендует быть сугубо технической, но каким-то непостижимым образом «ухитряется» превратиться в глашатая маркетинга, что меня совершенно не интересует. Поэтому моему восторгу не было предела, когда оба моих спонсора дали понять, что моя книга не должна быть ориентирована ни на компанию Mentor, ни на компанию Xilinx и может содержать любую информацию в любой форме, которая, по моему мнению, будет уместной, причем я должен делать это без страха и экивоков.

Эта книга никогда не увидела бы свет, если бы не помощь множества людей. Число их так велико, что не представляется возможным назвать всех. Тем не менее, я выражаю свою глубокую признательность всем представителям компаний Mentor и Xilinx, которые уделили мне массу времени и предоставили кучу информации. Я также благодарен Гэри Смиту и Дайу Надамуни (компания Gartner DataQuest) и Ричарду Гоэрингу (компания EETimes), у которых находилось время читать мои животрепещущие электронные письма, как правило, содержащие просьбу ответить на «всего лишь еще один маленький вопрос...».

Не могу не упомянуть представителей компаний O-In, AccelChip, Actel, Aldec, Altera, Altium, Axis, Cadence, Carbon, Celoxica, Elanix, In Time, Magma, picoChip, Quick-Logic, QuickSilver, Synopsys, Synplicity, The Math Works, Hier Design и Verisity, которые оказали мне неоценимую поддержку, а также Тома Хокинса (компания Launchbird Design Systems), который предельно ответственно относился к своим обязанностям по обеспечению меня потрясающими обзорами конструкторских утилит с открытым кодом. Хочу также поблагодарить доктора Эрика Богатина (компания GigaTest Labs) за любезно предоставленную мне информацию по эффектам интеграции сигналов на печатных платах.

И, конечно, самые искренние слова благодарности в адрес моего издателя Кэррола Льюиса за возможность включить в эту книгу Приложение В («Designus Maximus Unleashed», ISBN 0-7506-9089-5) и Приложение С («Vebop to the Boogie», второе издание (ISBN 0-7506-7543-8) из книг, авторами которых я являюсь.

Прошу прощения, если кого-то забыл упомянуть (неприменно сообщите мне об этом, и я обязательно исправлюсь в следующем издании).

Предисловие

Как автор, я беру на себя смелость утверждать, что эта книга может быть отнесена к разряду любопытного и нестандартного произведения, написанного в жанре технической прозы. Подобное утверждение преследует вполне определенную цель: привлечь к книге внимание широкого круга читателей. С одной стороны, это могут быть опытные инженеры, которые либо уже занимаются разработками систем с применением ПЛИС (FPGA) — программируемых логических интегральных схем, либо планируют заняться этим в ближайшем будущем. Именно им адресованы главы по созданию устройств на основе ПЛИС. В них рассматриваются стадии проектирования, инструментарий и методики с огромным количеством замысловатых технических тонкостей, которые способен оценить только инженер. С другой стороны, такие главы, как «Основные понятия», доступны читателям самого разного уровня подготовки.

Подобная «двуликость» продиктована огромной популярностью программируемых логических интегральных схем, особенно среди тех, кто никогда не имел с ними дело. Первые ПЛИС содержали сравнительно небольшое количество эквивалентов логических вентилях, к тому же производительность микросхем была невысокая. Поэтому раньше серьезные устройства создавались на базе *заказных интегральных схем (ASIC — application specific integrated circuit и/или ASSP — application specific standard parts)*. Но проектирование и изготовление заказных интегральных схем было довольно трудоемким и дорогостоящим занятием, к тому же модифицировать схему, которая представляла собой «идею, уже воплощенную в кремнии» практически невозможно без создания новой версии интегральной схемы.

Результаты сравнительного анализа свидетельствуют о том, что создание ПЛИС намного дешевле, чем заказных аналогов (ASIC или ASSP). Кроме того, в ПЛИС проще вносить изменения и сроки выхода этих устройств на рынок значительно сокращаются. Особого внимания заслуживает и тот факт, что современные программируемые логические интегральные схемы (ПЛИС — FPGA) содержат миллионы эквивалентов логических вентилях, встроенные процессоры и современные высокоскоростные интерфейсы. Все это позволяет использовать ПЛИС в приложениях, которые до настоящего времени могли быть реализованы только с помощью заказных микросхем (ASIC и ASSP).

Разработка ПЛИС со встроенными микропроцессорами требует совместных усилий разработчиков как аппаратного, так и программного обеспечения. Часто программисты плохо представляют себе тонкости, связанные с аппаратным обеспечением устройства. Поэтому книга будет интересна не только разработчикам аппаратных средств, но и той части славного братства программистов, которые занимаются разработкой встроенных приложений для таких устройств.

Еще одна группа читателей, которая может быть частью целевой аудитории данной книги, — это студенты колледжей и университетов по специальности «Электроника», продавцы и все люди, работающие в компаниях, чья сфера деятельности связана с САПР электронных устройств и программируемыми логическими интегральными схемами; а также аналитики и редакторы соответствующих журналов. Многие из этих читателей оценят доступность технического материала во вступительной части и изящество стиля изложения в приложениях.

И последнее, но очень важное замечание: я стремился написать книгу, которую мне самому было бы интересно читать. Честно говоря, сам я редко читаю техническую литературу, так как это нудное и утомительное занятие. Именно поэтому в своей книге я попытался сочетать сложные темы и базовые понятия, «как это появилось» и «почему это надо делать именно так, а не иначе», с россыпью удивительных по своей красоте мелочей. Одно из преимуществ такого изложения состоит в том, что, перечитывая страницы этой книги на закате лет, когда мой разум начнет бродить по закоулкам памяти, я буду приятно удивлен, и это придаст мне силы (всегда приятно знать, что впереди тебя ждет что-то удивительное).

Клайв Максфилд (Макс), июнь 2003 — январь 2004.

ВВЕДЕНИЕ

Что такое ПЛИС

FPGA (field programmable gate arrays), или ПЛИС (программируемые логические интегральные схемы), представляют собой цифровые интегральные микросхемы (ИС), состоящие из программируемых логических блоков и программируемых соединений между этими блоками. Возможность конфигурировать эти устройства позволяет инженерам-разработчикам решать множество различных задач.

В зависимости от способа изготовления ПЛИС могут программироваться либо один раз, либо многократно. Устройства, которые могут программироваться только один раз, называются *однократно программируемыми*.

Словосочетание «field programmable», содержащееся в расшифровке аббревиатуры FPGA, означает, что программирование FPGA-устройств выполняется на месте, «в полевых условиях» (в отличие от устройств, внутренняя функциональность которых жестко прописана производителем). Оно может также означать, что FPGA-устройства (ПЛИС) конфигурируются в лабораторных условиях, или свидетельствовать о том, что речь идет о возможности модификации функций устройства, встроенного в электронную систему, которая уже как-то используется. Если устройство может быть запрограммировано, оставаясь в составе системы более высокого уровня, оно называется *внутрисистемно программируемым*.

Чем интересны ПЛИС

Существует множество различных типов цифровых микросхем, в том числе и такие как «*рассыпная логика*» (небольшие компоненты, содержащие несколько простых фиксированных логических функций), устройства памяти и микропроцессоры. В данном случае интерес представляют *программируемые логические устройства (ПЛУ), специализированные заказные интегральные микросхемы (ASIC — application specific integrated circuit, специализированная интегральная схема и ASSP — application specific standard parts, специализированная стандартная схема*¹⁾ и, естественно, ПЛИС. Причем термин ПЛУ объединяет два типа устройств: простые программируемые логические устройства (*простые ПЛУ*) и сложные программируемые логические устройства (*сложные ПЛУ*)²⁾.

¹⁾ В советской (российской) технической литературе ASIC принято называть «заказными интегральными схемами (микросхемами)» хотя сегодня часто ASIC называют «специализированными интегральными схемами». Часто делается уточнение и микросхемы делят на «заказные» и «полузаказные». — *Прим. ред.*

²⁾ Более подробная информация о ПЛУ, и заказных микросхемах (ASIC и ASSP) содержится в главах 2 и 3. (Если не указано иное, примечание сделано автором.)

Внутренняя архитектура ПЛУ определена производителем, таким образом, что они могут быть сконфигурированы (перепрограммированы) «на месте» для выполнения самых различных функций. В отличие от ПЛИС эти устройства содержат меньшее количество вентилях и используются для решения небольших и достаточно простых задач.

Вместе с тем, существуют заказные интегральные схемы ASIC и ASSP, которые содержат сотни миллионов логических вентилях и могут выполнять невероятно большие и сложные функций. В основе ASIC и ASSP лежат одни и те же конструкторские решения, и у них одна и та же технология производства. Оба типа устройств разрабатываются для использования в составе специальных приложений, но при этом ASIC разрабатываются и производятся по заказу специализированных компаний, а ASSP предназначаются массовому пользователю¹⁾.

Несмотря на то, что предлагаемые пользователю заказные микросхемы отличаются высокой степенью интеграции, уровнем сложности решаемых задач и производительностью, их разработка и производство довольно длительный и дорогостоящий процесс. К тому же, окончательный вариант схемы «замораживается в кремнии», и для ее модификации требуется создание новой версии.

Таким образом, ПЛИС занимают промежуточное положение между ПЛУ и заказными интегральными схемами. С одной стороны, их функциональность может быть задана непосредственно на месте в соответствии с требованиями заказчика-пользователя. С другой стороны, они могут содержать миллионы логических вентилях и, следовательно, реализовывать чрезвычайно большие и сложные функции, которые изначально могли быть реализованы только с помощью заказных интегральных схем.

Что касается стоимости ПЛИС, то она намного ниже стоимости заказных интегральных схем (хотя окончательная версия заказной микросхемы при массовом производстве оказывается более дешевой). К тому же, в случае использования ПЛИС внесение изменений в устройство не вызывает особых затруднений и существенно сокращаются сроки выхода таких устройств на рынок. Все это делает ПЛИС привлекательными не только для крупных разработчиков, но и для небольших новаторских конструкторских бюро, которые благодаря способности ПЛИС создать «рай в шалаше» остаются жизнеспособными. Другими словами, «аппаратные» или «программные» идеи отдельных инженеров или небольших групп инженеров могут быть реализованы в виде испытательных стендов на основе ПЛИС без больших единовременных затрат на проектирование или закупку дорогостоящей оснастки, необходимой для разработки заказных микросхем. Именно этим объясняется тот факт, что в 2003 году, было начато почти 450000 разработок, предусматривающих использование ПЛИС, всего 5000 разработок с использованием заказных микросхем ASSP и только от 1500 до 4000 разработок с использованием заказных микросхем ASIC, причем эти цифры стремительно падают из года в год.



Трудно назвать эти цифры достоверными, потому что пока нет четкого представления о том, что следует называть началом разработки. В случаях с заказными микросхемами, например, не ясно, какие разработки надо учитывать: только те, которые были доведены до

В ПЛИС понятие «логический вентиль» приобретает иной оттенок. Эта тема подробно будет рассмотрена в главе 4.

¹⁾ В дальнейшем термин «заказная интегральная схема» (заказная микросхема) будет использоваться и по отношению к ASIC и по отношению к ASSP, при условии, что в тексте не указано иное или такая интерпретация не противоречит контексту.

конца, или и те, которые были закрыты на полпути. Ситуация еще больше усложняется, когда речь заходит о реконфигурируемости ПЛИС. В качестве убедительного доказательства существующего положения дел можно привести слова представителя поставщиков ПЛИС, который, порекомендовав мне использовать данные аналитиков на Web-сайте ПЛИС производителей, заметил: «Но там приводятся не очень-то точные данные». А на мой вопрос о причине появления таких данных, он, усмехаясь, ответил: «Главным образом потому, что именно такие данные они получают от нас». Эти цифры весьма относительны, поскольку зависят от того, кто проводил опрос. Не удивительно, что поставщики ПЛИС стремятся занижить число разработок с применением заказных микросхем по сравнению с оценкой из других источников.

Как можно использовать ПЛИС

Первые ПЛИС появились в середине 80-х годов. В то время они использовались преимущественно для создания *связующей логики*, для реализации конечных автоматов средней сложности и для решения некоторых задач обработки данных. По мере усложнения и увеличения размеров ПЛИС начинают пользоваться большим спросом. В начале 90-х прошлого века самый большой объем продаж отмечался в области сетей и телекоммуникаций, в которых предусматривались обработка и передача больших потоков информации. К концу 90-х спрос на ПЛИС резко возрос в потребительской, автомобильной и производственной сферах.

ПЛИС, как правило, использовались для создания прототипов заказных микросхем или для создания испытательных стендов, на которых проверяется физическая реализуемость новых алгоритмов. Однако благодаря низким затратам на производство и малым срокам выхода на рынок эти микросхемы всё чаще используются как конечный продукт. У некоторых крупных поставщиков ПЛИС есть устройства, которые составляют прямую конкуренцию заказным микросхемам.

К началу второго тысячелетия появились высокопроизводительные ПЛИС, которые содержат миллионы вентиляей. Некоторые из них содержат встроенные микропроцессорные ядра, высокоскоростные интерфейсы ввода/вывода и другие устройства. Современные ПЛИС находят применение практически в любой сфере, включая устройства связи и программируемые радиостанции. ПЛИС применяют в радиолокации, обработке изображений и в других приложениях *цифровой обработки сигналов (ЦОС)*. ПЛИС используют повсюду, в том числе и в *однокристалльных системах*¹⁾, содержащих программные и аппаратные модули.

Если быть более точным, в настоящее время ПЛИС заполняют четыре крупных сегмента рынка: заказные интегральные схемы, цифровая обработка сигналов, системы на основе встраиваемых микроконтроллеров и микросхемы, обеспечивающие физический уровень передачи данных. Кроме того, с появлением ПЛИС возник новый сектор рынка — *системы с перестраиваемой архитектурой*, или *reconfigurable computing (RC)*.

¹⁾ Хотя понятие «однокристалльной системы» подразумевает целостную электронную систему в одном устройстве, в реальности для их функционирования требуются дополнительные комплектующие. Поэтому более корректным будет название «однокристалльная подсистема» или «часть однокристалльной системы».

Термин «связующая логика» подразумевает сравнительно небольшое количество простых логических элементов, которые используются для соединения («склеивания»), и в качестве интерфейса, между большими логическими блоками, функциями или устройствами.

- **Заказные интегральные схемы.** Как уже отмечалось, современные ПЛИС используются для создания устройств такого уровня, который до этого могли обеспечить только заказные микросхемы.
- **Цифровая обработка сигналов.** Высокоскоростная цифровая обработка сигналов традиционно производилась с помощью специально разработанных микропроцессоров, называемых *цифровые сигнальные процессоры (ЦСП)* или *digital signal processors (DSP)*. Однако современные ПЛИС содержат встроенные умножители, схемы арифметического переноса и большой объем оперативной памяти внутри кристалла. Все это в сочетании с высокой степенью параллелизма ПЛИС обеспечивает превосходство ПЛИС над самыми быстрыми сигнальными процессорами в 500 и более раз.
- **Встраиваемые микроконтроллеры.** Несложные задачи управления обычно выполняются встраиваемыми процессорами специального назначения, которые называются *микроконтроллерами*. Эти недорогие устройства содержат встроенную программу, память команд, таймеры, интерфейсы ввода/вывода, расположенные рядом с ядром на одном кристалле. Цены на ПЛИС падают, к тому же, даже самые простые из них можно использовать для реализации программного микропроцессорного ядра с необходимыми функциями ввода/вывода. В результате ПЛИС становятся все более привлекательными устройствами для реализации функций микроконтроллеров.
- **Физический уровень передачи данных.** ПЛИС уже давно используются в качестве связующей логики, выполняющей функцию интерфейса между микросхемами, реализующими физический уровень передачи данных, и высшими уровнями сетевых протоколов. Тот факт, что современные ПЛИС могут содержать множество высокоскоростных передатчиков, означает, что сетевые и коммуникационные функции могут быть реализованы в одном устройстве.
- **Системы с перестраиваемой архитектурой.** Можно использовать «аппаратное ускорение» программных алгоритмов, основываясь на таких свойствах программируемых логических интегральных схем (ПЛИС), как параллелизм и перенастраиваемость. В настоящее время различные компании заняты созданием огромных перенастраиваемых вычислительных машин на основе ПЛИС. Такие системы могут использоваться для выполнения широкого спектра задач — от моделирования аппаратуры до криптографического анализа или создания новых лекарств.

Что есть в этой книге

Каждый, кто имеет дело с разработкой электронных устройств и использует системы автоматизированного проектирования электронных устройств (САПР электронных устройств или EDA — *electronic design automation*) знает, что и электронные компоненты, и системы проектирования с каждым годом становятся сложнее. Не являются исключением и ПЛИС, а также связанные с ними системы проектирования.

Когда ПЛИС только заявили о себе, где-то в середине 80-х, все было относительно просто. Первые устройства содержали всего несколько тысяч логических вентилях, точнее их эквивалентов. Процесс проектирования электронных компонентов в основном сводился к вводу описания электрической схемы, был понятным и доступным для освоения и использования. По сравнению с ними современные ПЛИС представляют собой невероятно сложные устройства, а средств и методов проектирования стало больше, чем может понадобиться пользователю.

Изложение материала в книге начинается с рассмотрения основных понятий и архитектурных изысков ПЛИС. Затем дань отдается множеству средств, методов и этапов проектирования, которые могут быть использованы инженерами в зависимости от поставленной цели. При этом книга позволяет заглянуть во «внутренний мир» ПЛИС. Кроме того, в ней рассматривается интегрирование устройств на основе ПЛИС, а также содержится информация по гигабитным интерфейсам, которые появились совсем недавно.

И последнее, что необходимо иметь в виду. Книга буквально нашпигована аббревиатурами, которые шутки ради можно назвать сокращениями из трех букв или акронимами. Если в разговоре со специалистом кто-то начнет называть вещи не своими именами, он сразу окажется чужаком, т. е. лицом, не принадлежащим к данному кругу (одним из «тех», но не из «наших»). Поэтому при вводе нового акронима, или аббревиатуры, с большим количеством символов, в книге дается его «расшифровка».

Чего нет в этой книге

В этой книге не отдается предпочтение какому-либо поставщику ПЛИС или какому-то конкретному виду ПЛИС. Объясняется это тем, что новые функциональные возможности и чипы появляются настолько быстро, что вся содержащаяся в книге информация может оказаться устаревшей прежде, чем она станет бестселлером (иногда, прежде чем автор успеет закончить данное высказывание).

По этой же причине в книге не приводятся названия отдельных фирм-поставщиков *САПР электронных устройств* и не содержатся ссылки на их продукты, потому что фирмы-поставщики постоянно трансформируются, меняя при этом свои названия или названия своих продуктов. Ввиду интенсивного развития данной сферы полностью теряют смысл такие выражения, как: «Средство А обладает данной характеристикой, а средство Б не обладает». Поскольку не исключено, что спустя несколько месяцев средство Б будет пользоваться повышенным спросом, а средство А будет вынуждено покинуть рынок.

Поэтому данная книга посвящена главным образом рассмотрению различных особенностей ПЛИС и многообразию средств и методов проектирования. При этом читателю предоставляется возможность самому выбрать поставщика микросхем, чьи ПЛИС поддерживают требуемую архитектуру, и поставщика САПР, чьи инструментальные средства обеспечат требуемую функциональность (полезные Web-ссылки представлены в гл. 6).

Кому предназначается эта книга

Книга рассчитана на широкий круг читателей, в который могут входить:

- специалисты небольших консалтинговых фирм;
- специалисты по разработке аппаратного и программного обеспечения крупных компаний;
- разработчики заказных микросхем, только еще начинающие работать с ПЛИС;
- разработчики устройств цифровой обработки сигналов, начинающие использовать ПЛИС в своей работе;
- студенты колледжей и университетов;
- молодежь из отделов сбыта, маркетинга и других подразделений компаний, сфера деятельности которых связана с САПР электронных устройств и ПЛИС;
- аналитики и редакторы журналов.