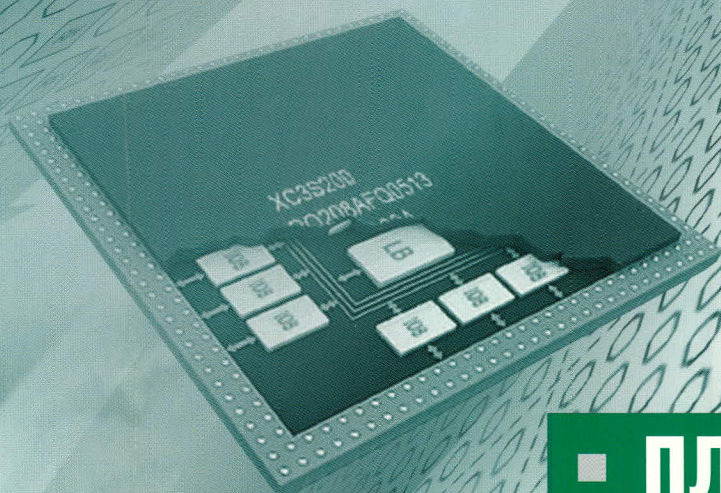


СОВРЕМЕННАЯ ЭЛЕКТРОНИКА

5
2007



■ ПЛИС

ПРОДОЛЖАЕТСЯ
ОБНОВЛЕНИЕ
ПОДПИСКИ
СТР. 80

POWER ELECTRONICS



"Благодарный партнер"
Специальный приз конкурса



ЧЛЕН
РОССИЙСКОГО
СОЮЗА ВЫСТАВОК
И ЯРМАРОК



Member

4-я Международная
специализированная выставка

Силовая Электроника

24-26 октября 2007

Москва, Конгресс-центр ЦМТ



Организаторы:



primexpo



ITE GROUP PLC

Тел.: +7 (812) 380 6000/03/07

Факс: +7 (812) 380 6001

E-mail: power@primexpo.ru

www.powerelectronics.ru

Главный редактор
Александр Майстренко

Зам. главного редактора
Татьяна Крюк

Редакционная коллегия
Роберт Алексанян, Андрей Данилов,
Григорий Маркаров

Дизайн и верстка
Татьяна Крюк, Александр Либков,
Дмитрий Юсим

Web-мастер
Дмитрий Романчук

Служба распространения
Ирина Самохина (circul@soel.ru)

Служба рекламы
advert@soel.ru

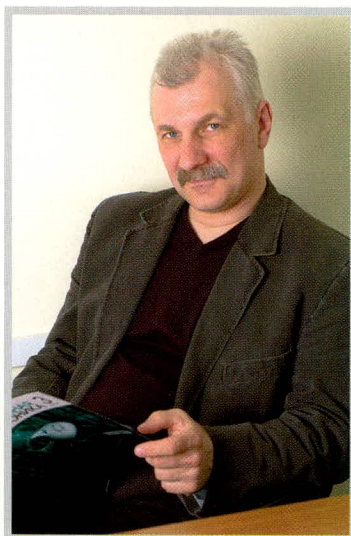
Издательство «СТА-ПРЕСС»
Директор Константин Седов
Почтовый адрес: 119313, Москва, а/я 26
Телефон: (495) 232-0087
Факс: (495) 232-1653
Сайт: www.soel.ru
E-mail: info@soel.ru

Журнал выходит 9 раз в год

Тираж 11 000 экземпляров
Журнал зарегистрирован в Федеральной
службе по надзору за соблюдением
законодательства в сфере массовых
коммуникаций и охране культурного наследия
(свидетельство ПИ № ФС77-18792
от 28 октября 2004 года)
Свидетельство № 00271-000 о внесении
в Реестр надёжных партнеров Торгово-
промышленной палаты Российской Федерации
Цена договорная

Отпечатано:
Полиграфический комплекс
«Пушкинская площадь»
www.pkpp.ru

Перепечатка материалов допускается только
с письменного разрешения редакции.
Ответственность за содержание рекламы
несут рекламодатели.
Ответственность за содержание статей
несут авторы.
Материалы, переданные редакции,
не рецензируются и не возвращаются.
© СТА-ПРЕСС, 2007



Уважаемые читатели!

Тема этого номера – программируемые логические интегральные схемы (ПЛИС).

Первоначально они были призваны заменить «цифровую рассыпку» – микросхемы низкого уровня логики, на которых, в основном, и строилась цифровая техника 30-летней давности. Сейчас ПЛИС – самостоятельный полноценный раздел полупроводниковой электроники, без которого невозможно представить её развитие. В ряде высокоскоростных приложений ПЛИС уверенно выигрывают конкуренцию с микропроцессорами и микроконтроллерами. Совместное использование микроконтроллера и ПЛИС позволяет существенно расширить и дополнить возможности каждого из устройств.

Использование ресурсов многовыводных ПЛИС позволяет расширить возможности ввода-вывода микроконтроллера. Синтез дополнительных модулей на базе ПЛИС существенно расширяет имеющуюся периферию микроконтроллера. Создаются системы реального времени, имеющие быстроедействие на порядок выше быстрогодействия микроконтроллера.

Учитывая это, производители аппаратно или программно встраивают в ПЛИС процессорные ядра. Непрерывно ужесточаются технологические нормы производства ПЛИС. Переход с технологии 90 нм на 65 нм уже состоялся, на очереди норма 45 нм, которая позволит снизить стоимость, увеличить объём, функциональные возможности и быстроедействие ПЛИС.

Бурное развитие свидетельствует о несовершенстве. Каким же будет совершенство?

*С наилучшими пожеланиями,
Александр Майстренко*

P.S. Напомню нашим читателям, что сейчас проводится продление подписки на наш журнал. Для этого необходимо всего лишь заполнить и отослать в редакцию анкету, напечатанную на стр. 80 данного номера журнала. Напомню также, что бесплатная подписка оформляется только для квалифицированных специалистов, аккуратно и полностью заполнивших анкету. Пользователи Интернета могут заполнить аналогичную анкету на нашем сайте www.soel.ru.

Market

News of the Russian Market	4
Trends in the FPGA Market and New Programmable Logic Modules	6
<i>Anton Reutov</i>	

Modern Technologies

Image Sensors (Part 3)	10
<i>Olga Gureeva</i>	

Components

LPC2000 (ARM7TDMI-S) Microcontrollers from Philips – One More Step on the Road to Creating an Ideal Platform for Embedded Applications (Part 8)	14
<i>Pavel Red'kin</i>	
Microwave Filters and Equipment on Their Basis from Pole/Zero (part 2)	20
<i>Andrey Lakhno</i>	

Projects

Asynchronous Sample-Rate Conversion between AES Audio Streams	22
<i>Gregg C. Hawkes, Reed Tidwell, John F. Snow</i>	
Stable 10-MHz Crystal Oscillator with Digital Compensation of Frequency Thermal Drift	26
<i>Dmitriy Tumaikin</i>	
Control Valve with Electric Drive	30
<i>Vladimir Vychuzhanin</i>	

Design and Simulation

Practical Course of Design of Digital Devices Based on Xilinx FPGA (Part 5)	34
<i>Valeriy Zotov</i>	
Program Package Converting PLA Design to Master Slice Layout	42
<i>Sergey Artemov</i>	
Library of Arithmetic VERILOG Descriptions in the Galois Field	46
<i>Arkadiy Polyakov, Mehdi Tajleb, Neshat Tajleb</i>	
Custom Logic VLSI Optimization in LeonardoSpectrum Synthesizer	50
<i>Pyotr Bibilo</i>	
Software Development Tools for Embedded 32-bit Systems (Part 2)	54
<i>Lyubov Samoilova, Vyacheslav Krasnov</i>	
JTAG Testing (Part 4)	56
<i>Iosif Karshenboim</i>	
Frequency Analysis of a Synthesizer for Pulsed Phase-locked Frequency Control. Part 1. Noise Filtering by Phase-lock Frequency Control Equipment (continuation)	66
<i>Yurii Nikitin</i>	

Programming

Linux OS for Systems Based on Xilinx PLA Chips (Part 1)	68
<i>Aleksei Shmatok</i>	

Events

Conference Innovations in Electronics	72
10 th Jubilee International ExpoElectronica Exhibition	78

РЫНОК

- 4 Новости российского рынка
- 6 Тенденции рынка ПЛИС (FPGA) и новинки модульных компонентов на базе программируемой логики
Антон Реутов

Современные технологии

- 10 Датчики изображения (часть 3)
Ольга Гуреева

Компоненты

- 14 Микроконтроллеры семейства LPC2000 (ARM7TDMI-S) от Philips – ещё один шаг к идеальной платформе для встраиваемых приложений (часть 8)
Павел Редькин
- 20 СВЧ-фильтры и оборудование на их основе фирмы Pole/Zero (часть 2)
Андрей Лахно

Проекты

- 22 Асинхронное преобразование частоты дискретизации аудиопотоков стандарта AES
Грег Хокс, Рид Тидвел, Джон Ф. Сноу
- 26 Высокостабильный кварцевый генератор 10 МГц с цифровой компенсацией температурного ухода частоты
Дмитрий Тумайкин
- 30 Регулирующий клапан с электроприводом
Владимир Вычужанин

Проектирование и моделирование

- 34 Практический курс сквозного проектирования цифровых устройств на основе ПЛИС фирмы Xilinx (часть 5)
Валерий Зотов
- 42 Пакет программ для перевода проекта схемы ПЛИС в базис БМК
Сергей Артёмов
- 46 Библиотека VERILOG-описаний арифметических операций в поле Галуа
Аркадий Поляков, Мехди Тайлеб, Незхат Тайлеб
- 50 Оптимизация логических схем заказных СБИС в синтезаторе LeonardoSpectrum
Пётр Бибило
- 54 Средства разработки программного обеспечения для встраиваемых 32-разрядных систем (часть 2)
Любовь Самойлова, Вячеслав Краснов
- 56 JTAG-тестирование (часть 4)
Иосиф Каршенбойм
- 66 Частотный метод анализа синтезаторной системы импульсно-фазовой автоподстройки частоты. Часть 1. Фильтрация помех структурой ФАП (продолжение)
Юрий Никитин

Программирование

- 68 ОС Linux для систем на кристаллах ПЛИС фирмы Xilinx (часть 4)
Алексей Шматов

События

- 72 Конференция «Инновации в электронике»
- 78 10-я Юбилейная международная выставка «ЭкспоЭлектроника»

Новости российского рынка

Компоненты

Операционный усилитель AS1710A с режимом «сна»

Новый одноканальный rail-to-rail-операционный усилитель AS1710A с функцией сна производства Austriamicrosystem предназначен для работы в портативных устройствах и звуковых картах.

Микросхема обеспечивает пиковый выходной ток 200 мА, работает в полосе частот до 10 МГц и имеет скорость нарастания выходного напряжения 10 В/мкс. Отличительной особенностью AS1710A является наличие входа, переводящего микросхему в режим «сна» с током потребления 1 нА. Для питания микросхемы может использоваться униполярный источник напряжения 2,7...5,5 В, ток потребления в рабочем режиме 1,6 мА, коэффициент подавления шумов питающего напряжения (PSRR) до 85 дБ.

Микросхема работает в диапазоне температур $-40...+125^{\circ}\text{C}$ и выпускается в шестивыводном корпусе SC70.

prochip.ru
Тел. 232-2522

12-битный АЦП с низким током потребления

Austriamicrosystem пополнила свою линейку АЦП двумя микросхемами. AS1524 представляет собой одноканальный АЦП с дифференциальным входом, а AS1525 АЦП – с двумя несимметричными входами и максимальной частотой выборки 150 кГц. Новые АЦП обладают ультранизким током потребления в рабочем режиме, который составляет 350 мкА, 150 квыб./с (3 В) и 2,5 мкА, 1 квыб./с (3 В). Кроме того, при отсутствии запросов микросхема автоматически переходит в режим пониженного энергопотребления с током потребления 0,2 мкА.

Совмещение высокой скорости выборки и низкого тока потребления в маленьком 8-выводном корпусе TDFN ($3 \times 3 \times 0,8$ мм) позволяет использовать микросхему в устройствах с автономным питанием 2,7...5,25 В и портативных сенсорных устройствах.

Обмен данными происходит по высокоскоростному SPI-/QSPI-/MICROWIRE-совместимому интерфейсу. Рабочая температура $-40...+125^{\circ}\text{C}$.

prochip.ru
Тел. 232-2522

AS1154/56 LVDS драйверы с пропускной способностью 800 Мб/с

Мировой лидер в разработке быстродействующих аналоговых интегральных схем Austriamicrosystem недавно анонсировала две низковольтные сигнальные ИМС. В составе ИС AS1156 имеется один, а в составе AS1154 – два LVDS-драйвера с пропускной способностью 800 Мб/с. Драйвер конвертирует два низковольтных LVTTTL/LVCMOS-сигнала в дифференциальные сигналы стандарта ANSI TIA/EIA-644 LVDS. Микросхемы имеют сквозную топологию выводов и экстремально низкую фазовую погрешность, равную $2,5 \times 10^{-10}$ с (250 пс), что даёт возможность применять ИС для высокоскоростной передачи данных.

Микросхемы выпускаются в корпусе SOIC-8, используют униполярный источник напряжения 3,0...3,6 В и работают в диапазоне температур от -40 до $+85^{\circ}\text{C}$.

prochip.ru
Тел. 232-2522

Узлы и модули

CHAMP-FX2: DSP-процессор формата VPX (VITA 46) на базе FPGA Xilinx Virtex 5 LXT

Компания Curtiss-Wright Controls Embedded Computing – производитель встраиваемых компьютерных плат и систем для военных применений – выпустила процессор цифровой обработки сигналов CHAMP-FX2, построенный на базе FPGA Xilinx Virtex 5 LXT и двухъядерного микропроцессора Freescale PowerPC/Altivec MPC8641D.

Процессор CHAMP-FX2 выпускается в формате 6U VPX (VITA 46) и содержит:

- две FPGA Xilinx Virtex 5 LX110T, LX220T или LX330T;

- 512 Мб DDR2 SDRAM и 36 Мб QDR-II+ SRAM на каждую FPGA;
- четыре четырёхполосных (x4) порта Serial RapidIO на разъёме VPX-P1;
- восьмипортовый коммутатор Serial RapidIO;
- микропроцессор MPC8641D 1 ГГц;
- 512 Мб/1 Гб ECC DDR2 SDRAM, 512 Мб Flash (8641D);
- 2xGbE, 2xRS232/RS422, 16 бит дискретного в/в;
- слот расширения XMC/PMC (VITA 42);
- датчики температуры и потребляемой мощности;
- JTAG-порт.

Плата CHAMP-FX2 выпускается в расширенном диапазоне температур $-40...+85^{\circ}\text{C}$ в исполнениях для воздушного и кондуктивного охлаждения, а также в варианте VPX-REDI (VITA 48) с защитным кожухом.

Для разработки схемотехники FPGA и программного обеспечения MPC8641D поставляется комплект разработчика Continuum FXtools и BSP для операционных систем VxWorks и Linux.

Компания Curtiss-Wright Controls Embedded Computing образована в 2004 г.

после слияния шести ведущих производителей встраиваемых плат и систем для военных применений: Dy4 Systems, VISTA Controls, Synergy Microsystems, Systran, Peritek и Primagraphics.

Представитель Curtiss-Wright CEC в России – компания AVD Systems.

www.avdsys.ru
(495) 148-9677

480-Вт источники питания для монтажа на DIN-рейку для работы трёхфазной сети

Компания XP Power расширила серию DNR промышленных источников питания AC/DC для монтажа на DIN-рейку 480-Вт одноканальными моделями DNR480T с универсальным входным напряжением 340...575 В трёхфазной сети переменного тока. Выходные напряжения 24 и 48 В регулируются в широком диапазоне: 22,5...28,5 и 47...56 В. Такой диапазон позволяет применять источники питания в различном оборудовании. Для требующих большой мощности применений могут включаться параллельно до трёх модулей.



Новости российского рынка

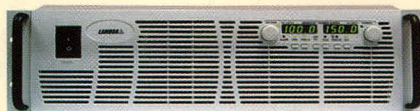


Модули серии DNR480T способны работать в широком диапазоне температур $-25...+70^{\circ}\text{C}$. Они могут выдавать в нагрузку полную мощность до температуры $+55^{\circ}\text{C}$ с последующим линейным снижением выходной мощности (коэффициент $2,5\%/^{\circ}\text{C}$) до 60% от максимального значения. Источники питания отличаются высокими энергетическими показателями качества: модель с выходным напряжением 24 В характеризуется значением КПД 89%, а 48-В модель – 90%.

prosoft.ru
Тел. (495) 234-0636

Новые мощные программируемые источники питания серии Genesys™ от Lambda

Компания Lambda начала производство мощных программируемых источников



питания популярной серии Genesys™ с выходными мощностями 10 и 15 кВт.

Новые модели работают от трёхфазных сетей переменного с частотой 47...63 Гц и напряжением 180...253, 360...440 и 432...528 В.

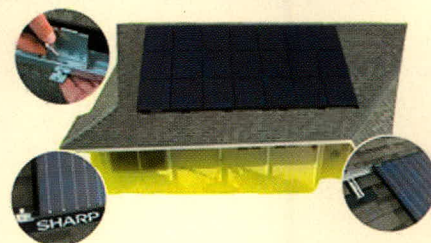
Ряд из 30 стандартных моделей серии Genesys™ включает источники питания с выходными напряжениями от 0...7,5 В (выходные токи 0...1000 А) до 0...600 В (токи 0...25 А). Можно соединить параллельно в режиме ведущий/ведомый четыре одинаковых 15-кВт блока и получить суммарную мощность 60 кВт.

Новейшей опцией в семействе Genesys™ стал интерфейс локальной сети, сертифицированной LXI Class C. Основные особенности источников питания: коэффициент мощности 0,88, встроенный интерфейс RS-232/RS485, программирование выходных параметров посредством аналоговых сигналов, USB-интерфейс для подключения к ПК, драйверы LabView™ и LabWindows™, гарантия 5 лет.

prosoft.ru
Тел. (495) 234-0636

Солнечные панели на основе собственной технологии от Sharp

Компания Sharp предлагает солнечные панели, разработанные на основе собственной технологии эффективной переработки кремния, способствующей уменьше-



нию толщины кремниевых ячеек и увеличению их КПД до 13,5%. Панели, позволяющие генерировать электричество, предназначены для применения в регионах с высокой солнечной активностью.

Линейка солнечных панелей Sharp включает несколько серий: ND (на основе поликристаллической технологии), NT (на основе монокристаллической технологии), NA (на основе комбинированной технологии).

Модель ND162E1F с ячейками площадью $155,5 \text{ мм}^2$ характеризуется номинальным напряжением 22,7 В при токе 7,14 А. В модели ND208U1F, пока ещё недоступной для жителей Европы, с мощностью 208 Вт, применяются квадратные ячейки поликристаллического кремния со стороной 156 мм. КПД составляет 12,8%. Панель NT175E1 с ячейками площадью $125,5 \text{ мм}^2$ характеризуется номинальным напряжением 24 В при максимально допустимом токе 10 А, значение КПД достигает 13,5%. NA-серия выполнена по комбинированной технологии аморфного ($\alpha\text{-Si}$) и микрокристаллического ($\mu\text{-Si}$) кремния с максимальным постоянным напряжением 600 В и мощностью 80...85 Вт.

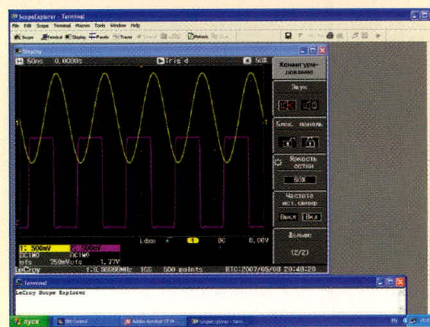
prochip.ru
Тел. 232-2522

Программное обеспечение

Дистанционное управление осциллографами WaveJet

Цифровые осциллографы LeCroy серии WaveJet при установке одной из опций (WJ-LAN/ WJ-GPIB) позволяют создать измерительную систему с возможностью удалённого управления и обработкой результатов измерений на персональном компьютере. Опции совместимы со всеми моделями осциллографов WaveJet. Пршивки, как и программное обеспечение удалённого управления и мониторинга, находятся в свободном для скачивания доступе на сайте корпорации LeCroy (<http://www.lecroy.com/tm/Library/Software>).

ScopeXplorer – бесплатное готовое приложение управления осциллографами LeCroy по интерфейсу Ethernet или GPIB



(RS-232) на базе ОС MS Windows 2000/XP, не требующее дополнительных навыков программирования.

Возможности:

- сохранение скриншотов (экранных снимков) с осциллографа на ПК;
- сохранение осциллограмм в различных форматах;

- сохранение профилей органов управления;
- отправка команд управления через терминальное окно (наглядная демонстрация возможностей удалённого управления при создании собственных инструментов на основе ActiveDSO).

ActiveDSO – бесплатный программный модуль ActiveX, позволяющий создавать собственные инструменты д/у для осциллографов LeCroy на базе Windows, например: MS Office, Internet Explorer, Visual Basic, Visual C++, Visual Java, Matlab и многих других.

www.prist.ru
Тел.: (495) 777-5591

Тенденции рынка ПЛИС (FPGA) и новинки модульных компонентов на базе программируемой логики

Антон Реутов (Москва)

В статье представлены тенденции развития рынка микросхем программируемой логики (ПЛИС), кратко обсуждены особенности новых микросхем ПЛИС и мезонинных модулей на основе программируемой логики.

Процессор – микроконтроллер – система на кристалле (SoC) – ПЛИС (FPGA). Микросхемы программируемой логики (field programmable gate array, FPGA) за 20-летнюю историю своего развития прошли путь от средств макетирования до полноправной элементной базы телекоммуникационного оборудования, систем сбора данных и потребительской электроники. В конкурентной борьбе с процессорами и микроконтроллерами за первенство в быстродействии и экономичности ПЛИС опираются на те же технологии субмикронной литографии, напряжённого кремния, медных проводников и специальных диэлектриков.

В конструкции микросхем программируемой логики реализуются технологии, подобные технологии Programmable Power в ПЛИС компании Altera, позволяющей достичь максимальной производительности одним блокам микросхемы и минимизировать энергопотребление всех остальных. Аналогичные технологии, например nanoWatt, используются для повышения энергетической эффективности микроконтроллеров. При необходимости на базе ПЛИС можно реализовать специализированную микросхему (технология HardCopy Structured ASIC компании Altera), что позволит сохранить уровень энергопотребления при повышении быстродействия.

В современных ПЛИС семейств LatticeECP2 или LatticeECP2M компании Lattice Semiconductor имеются вычислительные ядра с тактовой частотой 375 МГц, умножители, встроенная память, интерфейс DDR2 и 10-Гб

интерфейс SPI4.2. Микросхемы LatticeECP2M содержат аппаратные ресурсы, необходимые для поддержки интерфейсов PCI Express, Gigabit Ethernet и Serial RapidIO.

Компания Xilinx встраивает в свои микросхемы Virtex ядра с архитектурой PowerPC; программируемая логика компании Atmel содержит процессоры AVR. Одной из тенденций развития ПЛИС является увеличение числа разработок на основе интегрированных процессорных архитектур сторонних фирм, и в частности, ядер ARM и MIPS.

Помимо готовых аппаратных блоков (процессоров общего назначения, цифровых сигнальных процессоров, памяти), в микросхемах программируемой логики процессоры могут быть реализованы программным способом (soft-процессоры). В некоторых ПЛИС количество 16- и 32-разрядных программных процессорных ядер, оптимизированных по производительности, ресурсам логики и т.п., может достигать нескольких сотен. В качестве примеров можно назвать ядра MicroBlaze и PicoBlaze от Xilinx, процессоры семейства Nios компании Altera, ядра LatticeMico32 и LatticeMico8 компании Lattice Semiconductor.

Таким образом, программируемая логика уже применяет многопроцессорные системы, о которых сегодня говорят разработчики процессоров Intel и IBM. Многопроцессорность такого масштаба востребована в современных системах связи и обработки мультимедиа.

Возможность создания «однокристального» решения, отличающегося

меньшим энергопотреблением по сравнению с решением на основе нескольких компонентов, как правило, приветствуется разработчиками аппаратуры. Дополнительным аргументом в пользу выбора микросхем программируемой логики является увеличивающееся предложение ПЛИС, оборудованных флэш-памятью. Предполагается, что они могут быть востребованы в 44% всех разработок на основе ПЛИС. Улучшение характеристик изделий на базе одиночных ПЛИС (по сравнению с многокомпонентным решением) обеспечивается также расширением возможностей самих микросхем при переходе на новые технологии изготовления.

Повышение производительности ПЛИС становится насущной проблемой по мере роста требований к функциональным возможностям встроенных систем. Применительно к современным микросхемам программируемой логики это означает оптимизацию баланса характеристик встроенных в эти микросхемы функциональных блоков памяти, цифровой обработки сигналов, ввода/вывода и коммутируемой логики (logic fabric).

В середине 1980-х годов коммутируемая логика большинства таких микросхем базировалась на архитектуре 4-входных таблиц преобразования (4-input Look-Up Table, LUT). Современные микросхемы программируемой логики, производимые по 65-нм технологии, можно строить на основе 6-входных таблиц LUT с полностью независимыми входами, которые не являются разделяемыми ресурсами. Такова, например, архитектура ПЛИС Virtex-5 компании Xilinx, позволяющая минимизировать задержки.

Новая конструкция ПЛИС компании Xilinx даёт преимущества при арифметических операциях, а также улучшает характеристики памяти, будь то распределённая память LUT,

блочная RAM или буферная FIFO. При переходе на 65-нм технологию обеспечивается возможность работы блочной RAM с тактовой частотой 550 МГц, а размер блока достигает 36 Кбит. Увеличенные блоки могут поддерживать работу с 72-разрядными словами в двухпортовом режиме, а каскадирование двух блоков RAM позволяет увеличивать объём до 72 Кбит. Умножение 35×25 бит использует два блока DSP48 вместо четырёх блоков при 90-нм технологии.

В микросхемах Virtex-5 переход на новую топологическую норму позволил увеличить скорость передачи данных интерфейса к DDR2 SDRAM до 667 Мбит/с (с 534 Мбит/с), а число каналов – с 432 до 576 в режиме одновременно переключающихся выходов (simultaneous switching outputs, SSO).

Средний рост производительности типичных приложений при переходе с микросхем Virtex-4 (90 нм) на Virtex-5 (65 нм) составляет 30%. Однако для видеоприложений рост производительности составляет 53%, что обусловлено проведённой оптимизацией блоков DSP48.

По сравнению с предыдущим поколением изделий компании Altera, в современных 65-нм кристаллах в 1,7 раз увеличена логическая ёмкость, в 3,5 раза – встроенная память; до 260 МГц возросла тактовая частота умножителей; появились дополнительные возможности управления и синтеза синхросигналов, а также их динамической реконфигурации.

Приложение, включающее несколько задач с разными требованиями к быстродействию и работе с данными, может быть реализовано различными способами. Например – на базе быстродействующего процессора либо с использованием нескольких процессорных ядер с разными тактовыми частотами и периферией, оптимизированных для выполнения конкретных задач. С точки зрения кода приложения, разводки платы, ресурсов, затрачиваемых на разработку, энергопотребления законченной системы и возможности её работы в режиме реального времени эти способы отличаются существенно.

Конкурентным преимуществом ПЛИС является то обстоятельство, что сконфигурированная микросхема программируемой логики обеспечивает возможности параллельного выполнения нескольких задач при-

ложения. Это представляется одной из решающих предпосылок использования микросхем программируемой логики в системах связи, управления и измерений.

Новейшие ПЛИС типа Cyclone III компании Altera позволяют реализовать кодер H.264 по цене около \$20. Эти микросхемы нашли применение в устройствах шифрованной радиосвязи для военных применений, а благодаря аппаратным средствам поддержки интерфейсов LVDS, RSDS (Reduced Swing Differential Signaling/дифференциальный сигнал с уменьшенным размахом) и PPDS (Point-to-Point Differential Signaling/дифференциальная передача точка-точка) их можно использовать для работы с плоскостельными дисплеями. Относительно недорогие микросхемы Cyclone III могут выполнять функции цифровой обработки сигналов промежуточной частоты в базовых станциях.

Среди производителей ПЛИС лидируют компании Altera и Xilinx, которые контролируют 84% рынка. Оставшиеся 16% делят между собой компании Actel, Atmel, Cypress Semiconductor, Lattice Semiconductor, Quicklogic и ещё несколько фирм.

Эксперты сходятся в том, что рынок ПЛИС находится на подъёме. Но прогнозы для рынка программируемой логики разнятся, – по мнению специалистов компании In-Stat, мировые продажи микросхем программируемой логики могут вырасти в 2010 г. до 2,75 млрд. долл., а Gartner оценила объём рынка ПЛИС к этому же году в 6,72 млрд. долл.

Большая часть рынка ПЛИС – в денежном выражении – формируется малыми и средними партиями продаж. При больших объёмах разработчики идут на затраты, связанные с созданием специализированных многоядерных процессорных микросхем, но в мелкосерийном производстве средств связи, мультимедиа и многоканальных систем сбора данных у конфигурируемой логики конкурентов нет.

Наиболее ёмкими рынками для ПЛИС являются телекоммуникационное оборудование и промышленные/военные системы сбора данных. В 2005 г. эти два сегмента рынка потребили 59% от всего объёма проданных ПЛИС. Одним из основных стимулов роста рынка программируемой логики является беспроводная

связь. Развитие рынка сотовой связи в таких населённых странах, как Индия и Китай, влечёт за собой потребность в недорогом инфраструктурном оборудовании.

Программируемая логика осваивает и новые рынки. По прогнозам Gartner, к 2010 г. доля потребления на упомянутых сегментах сократится до 45% из-за роста объёмов продаж ПЛИС для потребительской электроники и оптоволоконных систем широкополосного доступа в Интернет в жилых домах (fiber-to-the-home, FTTH). В Японии, например, число подписчиков на FTTH превзошло число новых абонентов DSL. Рынок домашних телевизионных систем начал заполняться телевизорами высокой чёткости (HDTV), что влечёт за собой обновление смежной потребительской электроники и станционного оборудования поставщиков соответствующих услуг.

Модульные компоненты встроенных систем на основе ПЛИС

Производители потребительской электроники (относительно нового рынка для ПЛИС) уже используют микросхемы программируемой логики в широком спектре законченных изделий, таких как цифровые камеры, цифровые телевизоры, телевизионные приставки, игровые консоли, графические платы и приёмники GPS. Гибкость конфигурирования, высокая производительность и снижающееся энергопотребление расширили области применения программируемой логики на изделия для встроенных систем «промежуточного» уровня готовности, таких как мезонинные модули.

Производители телекоммуникационного оборудования опираются на концепцию мезонинов для удешевления и ускорения производства. Включение в конструкцию базовых плат мезонинных интерфейсов позволяет модернизировать уже выпускаемые изделия без существенной переделки. Дополняя возможности базовых плат функциональностью, обеспечиваемой мезонинными модулями (работа в сети, цифровая обработка сигналов, хранение данных, поддержка разнообразных портов ввода/вывода), поставщики встроенных систем могут быстро создавать любое специализированное изделие из доступных компонентов (Commercial Off-The-Shelf, COTS).

Первым стандартом мезонинного модуля стал PMC. Производители систем VMEbus и CompactPCI получили возможность добавлять интерфейсы локальной или глобальной сети (например, Gigabit Ethernet или T1), используя недорогой полноразмерный модуль, занимающий к тому же не целый слот, а относительно дешёвый мезонин. Стандарт PPMC предназначен для расширения спецификации PMC на телекоммуникационное оборудование. В нём предусмотрена отдельная шина для передачи данных, позволяющая передавать потоки TDM, Ethernet, ATM и Packet Over SONET (POS).

Новая мезонинная технология AdvancedMC (AMC), связанная с появлением стандарта телекоммуникационного оборудования AdvancedTCA, обеспечивает широкую полосу пропускания, предусматривает увеличение габаритов модулей (на 14% по сравнению с PMC) и мощности питания (до 60 Вт для самых больших модулей).

Компания BittWare разработала и поставляет серийно модуль B2-AMC (B2AM) в формате мезонина AdvancedMC, возможности которого определяются совместным использованием цифрового сигнального процессора (ЦСП) TigerSHARC от Analog Device и ПЛИС (FPGA) Stratix II компании Altera. Модуль предназначен для систем связи WiMAX, Software Defined Radio («программное» радио) и Super 3G. Его вычислительные возможности ограничены производительностью в 14,4 GFLOPS и 57,5 GOPS. Используемая в конструкции модуля ПЛИС типа Stratix II является основой для реализации коммуникационной архитектуры BittWare ATLANTiS, интегрирующей вычислительные ядра ЦПОС (4 TigerSHARC) с интерфейсами Serial RapidIO, PCI Express, Advanced Switching Interconnect (ASI), GigEthernet или XAUI (10 GigEthernet). Реализация архитектуры ATLANTiS на основе ПЛИС позволяет распределить ресурсы ввода/вывода между процессорами, упростить подключение каналов ввода/вывода ко всем или к одному процессору, интегрировать ПЛИС и ЦСП.

Аналогичная комбинация – микросхема Stratix II GX и 4-процессорный кластер ЦСП типа ADSP-TS201S TigerSHARC – использована компанией BittWare в семействе полноразмерных 3U CompactPCI модулей цифровой обработки сигналов GT-3U-cPCI

(GT3U), предназначенных для военных систем с кондуктивным охлаждением. Использование ПЛИС позволяет в течение всего срока жизни изделий адаптировать их к новым технологиям и требованиям заказчиков.

Реализация архитектуры ATLANTiS компании BittWare в микросхеме программируемой логики Stratix II GX FPGA обеспечивает передачу данных со скоростью 4 Гб/с и позволяет выделить значительную часть ресурсов ПЛИС для вычислительных задач. Четыре трансивера SerDes модулей GT3U благодаря архитектуре ATLANTiS могут быть сконфигурированы для поддержки протоколов Aurora, SerialLite, Serial Rapid I/O и PCI Express.

Для коммерческих приложений компания BittWare предлагает PMC-мезонин Tetra-PMC+ (TRPM), выполненный на базе ПЛИС Altera Cyclone II. Помимо четырёх 14-разрядных АЦП с частотой дискретизации 105 МГц, в этом устройстве посредством ПЛИС реализовано управление оцифровкой аналоговых сигналов и предварительной обработкой цифровых сигналов: фильтрацией, прореживанием и сдвигом по частоте (digital down conversion/DDC).

Оригинальной разработкой, опирающейся на возможности программируемой логики, является создание компанией Acromag мезонинных модулей семейств PMC-LX и PMC-SX на базе ПЛИС Virtex-4 компании Xilinx. Новые изделия предлагаются для использования в радарх и сонарах, автоматизированном испытательном оборудовании и коммуникационных системах. Особенностью конструкции модулей семейств PMC-LX и PMC-SX является возможность подключения к ним разнообразных модулей ввода/вывода AXM. Такая «вложенная» конструкция с использованием модулей расширения позволяет организовать обработку различных сигналов на базе конфигурируемой пользователем ПЛИС. Модули PMC-LX оптимизированы для поддержки высокопроизводительных ПЛИС типа LX40 или LX60 Virtex-4, а модуль PMC-SX построен на ПЛИС типа SX35 и предназначен для поддержки алгоритмов цифровой обработки сигналов.

Примером стандартного использования программируемой логики является конструкция процессорного мезонина AXA-100 в формате AdvancedMC. Он создан на базе процес-

сора Intel Core Duo, что практически удваивает вычислительную мощность систем AdvancedMC. В основе конструкции AXA-100 лежат интерфейсы XAUI (2) и PCI Express (4), реализованные на ПЛИС Xilinx Virtex-4.

ЗАКЛЮЧЕНИЕ

Использование микросхем ПЛИС в таких «полуфабрикатах» электронного оборудования, как мезонинные модули, открывает новые перспективы для микросхем программируемой логики. Производители ПЛИС-мезонинов и модулей других форматов, заинтересованные в популяризации своей продукции, подкрепляют аппаратные решения программными средствами. В качестве примера можно назвать комплект разработчика MEN Nios-CompactPCI, который позволяет подключать IP-периферию к интерфейсу soft-процессора Nios II. Компания MEN Mikro Elektronik предлагает IP-ядра для реализации RS232, HDLC и Fast Ethernet. Поддержка конфигурирования архитектуры ATLANTiS компании BittWare осуществляется программным обеспечением ATLANTiS Navigator, входящим в состав инструментария BittWorks Toolkit.

И естественно этот «дополнительный» инструментарий интегрируется со средствами поддержки разработок от тех компаний, кто собственно и производит микросхемы ПЛИС (FPGA).

Мезонины позволяют гибко конфигурировать и масштабировать конечное оборудование за счёт аналогичных свойств микросхем программируемой логики; они сформировали систему модулей, на базе которых предлагаются законченные решения высокой степени готовности начального и среднего уровней (стандарт MicroTCA). На основе систем MicroTCA можно создавать базовые станции, оборудование DSLAM (Digital Subscriber Line Access Multiplexer), маршрутизаторы уровня рабочих групп, изделия для «цифрового» дома и т.п. Весьма перспективным является использование MicroTCA в военной аппаратуре, где будут востребованы высокая готовность и надёжность таких систем, а также их относительно невысокая стоимость. Системы MicroTCA могут заинтересовать разработчиков устройств хранения данных, производительных серверов, медицинского и промышленного оборудования.



Новости мира News of the World Новости мира

Infineon потеряла в прошлом квартале \$15 млн.

Производитель чипов, немецкая компания Infineon Technologies, на днях обнародовала данные о результатах деятельности во втором квартале. Несмотря на оптимистичные прогнозы аналитиков, менеджеры Infineon Technologies в этом периоде столкнулись с серьёзными проблемами: убытки компании превысили \$15 млн., что, признаться, намного меньше отрицательных показателей прошлого года, когда Infineon потеряла \$35 млн. Парадокс, но аналитики Dow Jones News-wires предсказывали компании в этом периоде рост прибыли до уровня \$23,1 млн.

Продажи снизились на 2% и в сумме составили \$2,66 млрд., хотя в прошлом году Infineon отгрузила продукции на сумму \$2,7 млрд. Ошибочный прогноз экспертов пророчил продажи на уровне \$2,73 млрд.

Несмотря на это, президент Infineon, Вольфганг Цибарт (Wolfgang Ziebart), выразил удовлетворение деятельностью его предприятия, однако отметил недостаточный уровень делового исполнения некоторых структур. Одной из причин убытков разработчика можно назвать кризис на рынке чипов памяти DRAM, когда цены на эту продукцию упали более чем на треть из-за переизбытка. Нужно сказать, что затруднения и снижение прибылей из-за этого кризиса наблюдали и другие разработчики памяти по всему миру.

physorg.com

LG.Philips лидирует по поставкам крупных ЖК-панелей

Согласно данным аналитического агентства iSuppli, совместное детище корейских и голландских производителей LG.Philips LCD по итогам первого квартала 2007 г. вышло на первое место по поставкам крупных ЖК-панелей для телевизионных приёмников и компьютерных мониторов, обойдя тайваньскую AU Optonics. Корейский гигант Samsung Electronics, сохранив первое место по объёму полученных средств, потеснился по количеству проданных панелей на третье место рейтинга.

«LG.Philips определённо справилась с сезонным снижением спроса в первом квартале 2007 г., – говорит Света Дэш (Sweta Dash), директор группы мониторинга рынка ЖК ТВ iSuppli. – Согласно заявлениям официальных лиц, метраж отгруженных панелей снизился в указанном периоде всего на 1%. Кроме того, в пер-

вом квартале компания сократила с трёх до двух недель среднее время нахождения готовой продукции на складе, что способствовало увеличению её доходов».

Несмотря на то что второй квартал традиционно рассматривается как неудачный для производителей крупных ЖК-панелей, аналитики iSuppli предсказывают увеличение спроса на матрицы больших размеров в указанный период. В целом в 2007 г. ожидается увеличение поставок крупных ЖК-панелей на 350 млн. штук, т.е. на 24% по сравнению с уровнем 2006 г.

eetimes.com

Winbond введёт в строй производство 80-нм чипов DRAM

По итогам первого квартала 2007 г., компания Winbond Electronics, входящая в десятку крупнейших мировых производителей модулей динамической памяти DRAM, констатировала значительный, на 53%, рост объёма продаж по сравнению с аналогичным показателем прошлого года, составившим 285 млн. долл. США. Чистая прибыль компании за указанный период достигла 10,2 млн. долл. США. Однако, по отношению к четвёртому кварталу 2006 г. компания констатировала снижение количества проданной продукции на 18,8% – участь, постигшая практически всех ведущих производителей DRAM в первые три месяца текущего года.

В дополнение к приличным финансовым результатам первого квартала, компания объявила о начале второй фазы развёртывания производства на 300-мм фабрике, расположенной в одном из крупнейших тайваньских «силиконовых заповедников» Taichung Science Park (TSP). Основной задачей предприятия является ввод в строй второй производственной линии, занятой выпуском 80-нм микросхем DRAM.

Как сообщил официальный представитель компании, из 24 тыс. ежемесячно выпускаемых подложек с чипами DRAM, 16 тыс. обрабатываются согласно нормам 90-нм техпроцесса. При этом большинство из произведённых чипов представляют собой модули стандарта DDR2 ёмкостью 512 Мбит, предназначенные для основного партнёра немецкой компании Qimonda. Ожидается, что вторая фаза закончится в середине 2008 г. стартом массового производства 80-нм чипов с ожидаемым выходом до 24 тыс. подложек в месяц. Суммарная мощность обновлённого производства составит 48 тыс. подложек в месяц.

digitimes.com

TSMC отчиталась за первый квартал 2007 г.

Одна из двух ведущих тайваньских «полупроводниковых кузниц» Taiwan Semiconductor Manufacturing Co. (TSMC) объявила, что доход компании от основной деятельности в первом квартале 2007 г. составил 2 млрд. долл. США, что на 16,6% ниже, чем в первом квартале 2006 г. и на 13,4% ниже аналогичного показателя четвёртого квартала прошлого года.

Как следует из официального пресс-релиза, за указанный период чистая прибыль компании составила 567,5 млн. долл. США, что на 42,3% ниже прошлогоднего показателя и на 32,5% меньше прибыли предыдущего квартала. При этом отмечается, что на долю продуктов, произведённых согласно норм 130-нм техпроцесса и менее, приходится около 49% всей произведённой в первом квартале продукции. Доходы от реализации 90-нм продуктов составляют 22%, а 65-нм – всего 1%. Отмечается, что доходы от продукции, предназначенной для компьютерной индустрии, снизились на 21%, коммуникационных приборов – на 15% и потребительской электроники – на 12%. Доходы от продаж 65- и 90-нм продуктов остались практически неизменными с прошлого квартала, тогда как доходность от продаж 130-нм продуктов увеличилась на 1% по сравнению с четвёртым кварталом 2006 г.

Консолидированная прибыль за первый квартал 2007 г. составила 741 млн. долл. США, что составило 37,9% от чистой выручки предприятия, тогда как аналогичный показатель четвёртого квартала прошлого года составлял 46%. Рентабельность производства (gross margin, отношение валовой прибыли к выручке от продаж) понизилась за первый квартал 2007 г. на 8,1%, в основном благодаря уменьшению загрузки производственных мощностей, снижению средних отпускных цен, увеличению амортизационных отчислений и низкому вниманию к увеличивающемуся возврату отгруженных товаров в четвёртом квартале 2006 г.

Согласно прогнозу аналитиков компании, во втором квартале TSMC может достичь уровня доходов между 2,3 и 2,4 млрд. долл. США, рентабельности валовой прибыли в 42...44% и рентабельности операционной прибыли (прибыли от основной деятельности) в 32...34%.

reed-electronics.com

Датчики изображения

(часть 3)

Ольга Гуреева (Москва)

В третьей части статьи рассматриваются принципы получения изображения с помощью ПЗС и КМОП-датчиков различных архитектур. Приводится описание важнейших характеристик датчиков – разрешения и динамического диапазона.

СВЕТ И ЦВЕТ

Датчики изображения – это устройства, определяющие лишь градации серого, то есть интенсивность света от белого до совершенно чёрного.

Для того чтобы изображение стало цветным, на кремний с помощью процесса фотолитографии наносится слой цветных фильтров.

В тех датчиках, где используются микролинзы, слой фильтров помещается между линзами и светочувствительными элементами. В сканерах, где используются трёхлинейные ПЗС (т.е. рядом расположены три отдельных ПЗС, реагирующих либо на красный, либо на зелёный, либо на синий цвет) и в некоторых профессиональных камерах, где также используются три датчика, цветное изображение получается простым суммированием изображения, поступающего с трёх матриц.

Более распространёнными являются устройства с одним датчиком. В таких устройствах для получения цветного изображения используются массивы цветных фильтров (color filter arrays – CFA) (см. рис. 16).

Каждый пиксел отвечает за «свой» цвет, для чего над ним помещается соответствующий фильтр. Фотоны, прежде чем попасть на пиксел, проходят через фильтр, который пропускает свет с длинами волн только «своего»

цвета. Свет с другими длинами волн поглощается фильтром. Известно, что любой цвет в спектре можно получить смешением нескольких основных цветов. В модели RGB таких цвета три – красный, зелёный и синий.

Для различных применений разработаны различные модели массивов цветных фильтров. Тем не менее в большинстве датчиков используется массив цветных фильтров Байера (Bayer pattern) (см. рис. 17).

Эта технология была предложена компанией KODAK в семидесятых годах прошлого века и основана на результатах работ по пространственному мультиплексированию. В системе Байера фильтры расположены попеременно, в шахматном порядке, причём количество зелёных фильтров в два раза больше, чем красных и синих. Порядок расположения фильтров таков, что красные и синие фильтры расположены между зелёными. Такое соотношение объясняется строением человеческого глаза. Наше зрение наиболее чувствительно к зелёному цвету. Расположение фильтров в шахматном порядке обеспечивает цветовую равномерность изображения независимо от того, как расположен датчик во время съёмки – горизонтально или вертикально.

Информация с такого датчика считывается последовательно, построч-

но. За строчкой BGBGBG следует GRGRGR и так далее, что представляет собой последовательность RGB.

В ПЗС суммирование трёх цветов для получения цветного изображения выполняется вне датчика. Наложение цветов происходит в устройстве обработки изображения после того, как сигнал преобразован из аналоговой формы в цифровую. В КМОП-датчиках совмещение цветов может происходить непосредственно на кристалле. В любом случае первичные цвета каждого пиксела математически интерполируются с учётом цветов соседних пикселов. В действительности лишь некоторые части изображения имеют точный красный, зелёный или синий цвет, большинство из них получаются в результате совмещения этих трёх цветов.

При интерполяции обрабатывается матрица пикселов размером 3 × 3. При этом для определения цвета центрального пиксела сравниваются и учитываются значения окружающих пикселов.

Рассмотрим простейший пример, когда при трёх пикселах с синим, красным и синим цветовыми фильтрами (BRB), расположенными в ряд, требуется определить реальный цвет пиксела с красным фильтром. Если предположить, что отсутствуют средневзвешенные величины и все пикселы вносят одинаковый вклад в результирующий цвет, то цвет центрального (R) пиксела будет вычислен математически как две части синего и одна часть красного. В действительности алгоритмы даже простой линейной интерполяции намного сложнее, так как учитываются значения всех окружающих пикселов. При грубой интерполяции замечен шум от наложения цветов, в особенности на границе цветовых переходов.

РАЗРЕШЕНИЕ ДАТЧИКОВ

Сразу подчеркнём, что термин «разрешение» используется в цифровой графике некорректно. С точки зрения оптики, разрешение – это мера способности оптического устройства, в том числе и человеческого глаза, различать отдельные линии на специальных диаграммах, например, на диаграмме ISO, представленной на рис. 18.

В отличие от оптики, в компьютерной технике со времён первых мониторов разрешением принято назы-

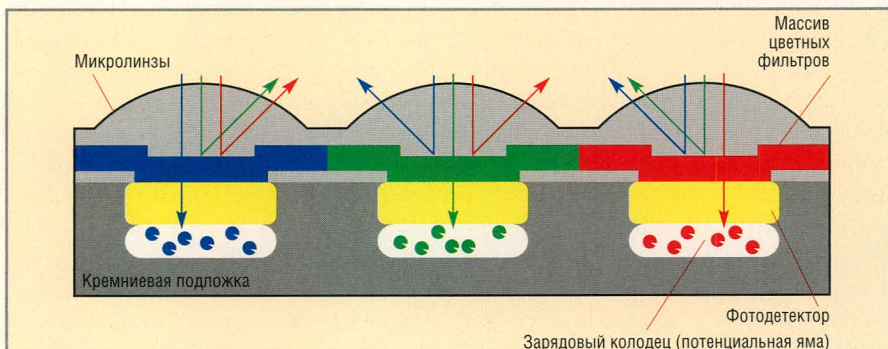


Рис. 16. Массив цветных фильтров CFA

вать количество пикселей, которое можно отобразить на экране. Поэтому в дальнейшем под разрешением датчика изображения будем понимать именно количество пикселей.

Количество пикселей, или разрешение датчика, связано с размером файла получаемого изображения и с тем, как он в дальнейшем будет обрабатываться. Существует прямая зависимость – чем больше пикселей, тем больше получаемый файл. Так, например, датчик изображения формата VGA размера 640 × 480 или 307 200 активных пикселей будет занимать в несжатом виде примерно 900 Кб (307 200 пикселей, по 3 байта R-G-B на пиксел составит 921 600 байтов, или 921 600/1024 составит 900 Кб). Датчик разрешением в 16 Мп создаст файл объёмом 48 Мб. На первый взгляд, подсчёт пикселей для определения получаемого изображения не представляет никакой сложности. Тем не менее производители датчиков в техническом описании указывают самые различные цифры, утверждая, что это и есть истинное разрешение устройства. Заметим, что общее количество пикселей – это число всех пикселей, физически существующих на датчике. При этом активными считаются только те, которые участвуют в получении изображения. Несколько процентов от общего числа пикселей являются «битыми» или «тёмными» и не участвуют в получении изображения. Они либо дефектные, либо используются датчиком для других целей. Например, могут быть предусмотрены маски для калибровки уровня темного тока или для определения формата кадра. Формат кадра – это соотношение между шириной и высотой датчика. Например, в датчике с разрешением 640 × 480 пикселей это соотношение составляет 1,34 : 1, что соответствует формату кадров многих компьютерных мониторов. Это значит, что изображения, полученные такими датчиками, будут точно укладываться в рамки экранов мониторов без предварительного кадрирования. Во многих устройствах формат кадра соответствует формату традиционной 35-миллиметровой плёнки, где соотношение сторон кадра равно 1 : 1,5. Это позволяет делать стандартные снимки.

ИНТЕРПОЛЯЦИЯ РАЗРЕШЕНИЯ

Кроме оптического разрешения (реальная способность пикселей реа-

гировать на фотоны), существует также разрешение, поддерживаемое программными средствами, использующими интерполирующие алгоритмы. Как и в случае интерполяции цветов, при интерполяции разрешения математически анализируются данные соседних пикселей. При этом в результате интерполяции создаются промежуточные, дополнительные значения соседних пикселей. Такое «внедрение» новых данных происходит равномерно, что делает переход от реальных оптических данных к интерполированным данным в большинстве случаев практически незаметным. Тем не менее временами при интерполяции могут возникнуть различные помехи, в результате чего изображение только ухудшается. Поэтому существует мнение, что интерполяция разрешения – это не способ улучшения качества изображения, а лишь метод увеличения размера файла. Таким образом, при выборе датчика изображения необходимо обращать особое внимание на то, какое разрешение указано в документации к устройству. Не стоит возлагать особых надежд на интерполированное разрешение (оно может называться как *interpolated*, так и *enhanced*), если требуется получить качественное изображение.

Существует ещё один процесс программной обработки изображения, который называется субдискретизацией (*sub-sampling*). Этот процесс является обратным интерполяции. В случае ПЗС-датчика субдискретизация удаляет данные различных пикселей после того, как данные преобразованы из аналогового в цифровой вид. В КМОП-датчиках эту операцию можно провести непосредственно на кристалле, временно прервав считывание определённых строчек или считывая данные только с избранных пикселей. Субдискретизация используется в двух случаях. Во-первых, для уплотнения данных, чтобы иметь возможность поместить большее число снимков на карте памяти или во внутренней памяти устройства (чем меньше количество пикселей, тем меньше получается размер файла). И во-вторых, для создания изображений требуемого размера для определённых целей. Субдискретизация позволяет обработать изображение таким образом, чтобы оно оптимально отображалось на мониторах (если не ста-

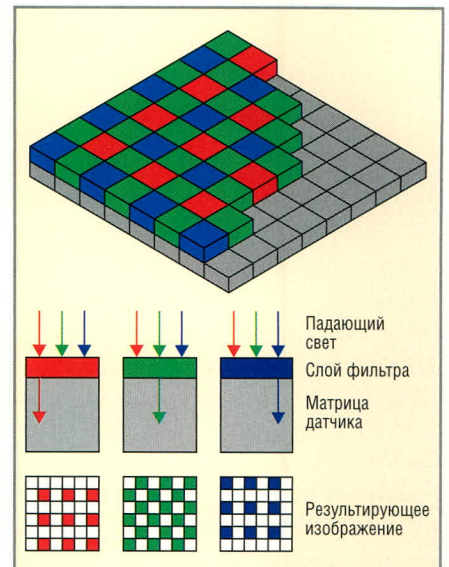


Рис. 17. Массив цветных фильтров Байера

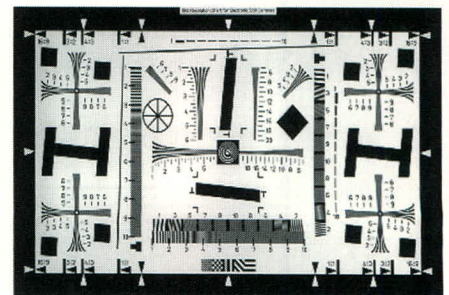


Рис. 18. Диаграмма ISO 12233

вить целью детализацию) или отправлялось быстро по электронной почте при медленном сетевом соединении.

После того как мы познакомились с основополагающими принципами работы датчиков изображения, рассмотрим более сложные ситуации, возникающие в процессе их функционирования.

КОЛЛИЗИИ ФОТОНОВ

Электроны, освобождённые в результате фотоэффекта, перемещаются по кристаллу датчика хаотически. На направление движения электронов влияет множество факторов – угол падения фотона на кристалл, длина волны фотона, структура кристаллической решетки атомов кремния, качество кремния, состав примесей, количество слоёв в матрице и т.д. Поэтому «выбивание» фотоном электрона ещё не означает захват последнего зарядовым колодцем. Электрон может быть обратно «затянут» кристаллом кремния или может переместиться в произвольном направлении. Также если фотон пересекает под определённым углом массив цветных фильтров, существует вероятность, что он попадёт не на свой, а на сосед-

ний светочувствительный элемент. Это может привести к искажению заряда соседнего пиксела. Такая помеха называется оптической перекрестной наводкой. Например, свет при некотором угле может пройти через красную область фильтра и проникнуть на соседний светочувствительный элемент с зелёным фильтром. В результате изображение искажается. Оптическую перекрестную наводку невозможно устранить на стадии обработки изображения. Она нейтрализуется с помощью создания специальных барьеров между пикселями. Так, например, компания KODAK для устранения этой помехи использует чёрные граничные участки в массиве цветных фильтров. Ряд других производителей, например Philips, используют металлические экраны, расположенные за массивом фильтров. Считается, что такой метод является более надёжным, чем чёрные границы фильтров. Тем не менее, и он не лишён недостатков. Во-первых, стоимость металла выше, чем стоимость полимера, во-вторых, металлические экраны занимают больше места и, в-третьих, увеличивается вес датчика.

Присутствие оптических помех связано с физической природой света. Уровень погружения фотона в кремний до момента преобразования в электрон зависит от длины световой волны. Если длина электромагнитной волны большая, то свет просто не задерживается в кремнии. Чем меньше длина волны, тем большей энергией обладают фотоны и тем быстрее формируются электроны. В случае, когда длина волны крайне мала, фотоны отражаются от поверхности датчика. Среди цветных фильтров Байера красный цвет обладает максимальной длиной волны, соответственно, фотоны красного цвета обладают меньшей энергией. Поэтому когда фотоны проходят через красный фильтр, они проникают глубже в кремний до момента преобразования в электроны. Это может привести к неправильной цветопередаче, то есть потере заряда на подложке или захвату электронов соседними зарядовыми колодцами. Как и в случае оптических помех, электрические помехи не поддаются корректировке на стадии обработки изображения. Для их минимизации используются специальные конструктивные решения, разрабатываемые на стадии проектирования датчиков.

РАСПЛЫВАНИЕ ИЗОБРАЖЕНИЯ

Ещё один источник помех – это расплывание изображения. Зарядовые колодцы могут накапливать лишь ограниченное количество электронов. Это количество получило название коэффициента заполнения. При сильном потоке света электроны переполняют зарядовый колодец отдельно взятого пиксела и «переливаются» в зарядовые колодцы соседних пикселей. Избыточный заряд, который перемещается от одного пиксела к другому, создаёт яркое пятно или полосы, которые приводят к расплыванию изображения. Наиболее распространённый метод устранения данного эффекта – это создание специальных барьеров и дополнительных зарядовых колодцев, предназначенных для приёма только избыточных зарядов. К сожалению, барьеры и дополнительные зарядовые колодцы занимают до 30% полезной площади пиксела, уменьшая как его чувствительность, так и квантовую эффективность. В некоторых случаях разработчики датчиков вынуждены идти на компромисс и в целях уменьшения себестоимости изделия мириться с такими недостатками, как расплывание изображения (если, конечно, подобное допускает приложение, в котором этот датчик будет использоваться).

ДИНАМИЧЕСКИЙ ДИАПАЗОН И ШУМ

Динамический диапазон датчика характеризует его способность «захватывать» оттенки изображения от самых тёмных до самых светлых тонов. Чем шире этот диапазон, тем больше оттенков изображения фиксируется датчиком.

Соотношение сигнал/шум также является важной характеристикой датчика. Это соотношение часто отражено в описании динамического диапазона датчика. То есть невозможно определить динамический диапазон датчика, не зная его уровень шума.

Интересно, что для обеспечения той же степени детализации, как и у фотоплёнки, особенно в тенях и в наиболее ярко освещённых участках изображения, динамический диапазон датчика должен быть больше, чем динамический диапазон фотоплёнки.

Датчики изображения являются электронными устройствами, поме-

хи в которых могут возникнуть по целому ряду причин, и все эти причины, в том числе описанные выше, приводят к нежелательным эффектам, искажающим изображения, т.е. к шумам.

В некоторых случаях пиксели неравномерно реагируют на свет, что приводит к образованию зон с разной чувствительностью. Тем не менее из всех шумов самую негативную роль, особенно в связи с динамическим диапазоном, играет темновой ток. Темновой ток – это ток, возникающий в электрической цепи датчика при отсутствии светового потока.

Основная причина появления темнового тока – это примеси в кремниевой подложке или повреждение кристаллической решётки кремния. В чистом кремнии темновой ток невелик, но при использовании особо чистых полупроводников значительно возрастает себестоимость готового изделия. К появлению темнового тока также приводят технологические нарушения в процессе производства датчиков, особенно на этапе ионного легирования, а также нагревание датчика. При повышении температуры устройства на 6...8°C значение темнового тока удваивается.

ТЕМПЕРАТУРНЫЕ РЕЖИМЫ

При продолжительной работе любая цифровая фотокамера нагревается, поэтому в профессиональных устройствах датчик изображения активно охлаждается. Для охлаждения используются вентилятор, элемент Пельтье либо свободная конвекция.

В датчиках, предназначенных для высокоточных научных и астрономических приложений, применяется жидкостное охлаждение. Определённый компромисс наблюдается в конструкции фотокамер с жидкокристаллическим видеоискателем. ЖК-экран таких камер крепится к корпусу с помощью откидного блока. Это решение удобно пользователям – проще наводить камеру. Но главное преимущество подобной конструкции заключается в том, что тепло, выделяемое лампами подсветки экрана, не попадает на корпус и не повышает температуру датчика изображения. Тем самым темновой ток не увеличивается. Соответствен-

но, изображение менее подвержено шумам.

Ещё один способ понизить уровень темнового тока – это технология синхронизации режимов накопления, также называемая MPP-технологией (MultiPinned Phased technology), при которой с помощью изменения напряжения «дырки» (положительные заряды) перемещаются по поверхности кремния и притягивают свободные электроны.

Также используются диодные накопители дырок (HAD – hole accumulation diode), через которые «лишние» дырки переносятся в зону отрицательного заряда.

ЛОГАРИФИЧЕСКИЙ МЕТОД ВЫЧИСЛЕНИЯ ДИНАМИЧЕСКОГО ДИАПАЗОНА ДАТЧИКА

Заметим, что различные производители вычисляют динамический диапазон датчика по-разному. Компания Philips обоснованно утверждает, что нельзя говорить о динамическом диапазоне, не упоминая при этом температуры. В противном случае нельзя будет сравнить динамические диапазоны двух разных производителей в связи с различными методами расчётов.

В связи с этим наибольшее распространение получил логарифмический метод вычисления динамического диапазона для определения битовой глубины датчика:

$$\text{Динамический диапазон (дБ)} = 20 \times \ln[\text{полная зарядная ёмкость} / (\text{темновой ток} + \text{помехи})].$$

Полная зарядная ёмкость в данной формуле определяется количеством электронов в зарядовом колодце. Помехи также выражаются через количество электронов.

Допустим, что полная зарядная ёмкость составляет 40 960 электронов, помехи – 10 электронов. Разделив первую величину на вторую, получаем 4096. Вычислив десятичный логарифм, имеем 3,61236. Соответственно, значение динамического диапазона равно 72,25 дБ.

Идея данного подхода заключается в оцифровке выходного сигнала датчика с помощью АЦП, разрядность которого может быть 8, 10, 12 и более бит, для получения такого количества оттенков серого, какое позволяет уровень сигнала. При 10-битовой

оцифровке будет 1024 оттенков серого, а при 12-битовой – 4096. Если уровень сигнала составляет всего 60 дБ, что приблизительно соответствует отношению полной зарядовой ёмкости к шуму, равному 1024 (а это 2, возведённое в десятую степень), использовать 12-битный АЦП нецелесообразно.

В случае оцифровки сигнала 60 дБ с помощью 8-битного АЦП часть оттенков будет потеряна. При 8-битовой оцифровке можно получить только 256 оттенков серого цвета. Таким образом, для сигнала с уровнем 60 дБ следует использовать 10-битовый АЦП.

Динамический диапазон также зависит от размера пиксела. Чем меньше пиксел, тем меньше его зарядовая ёмкость, тем меньше динамический диапазон. С другой стороны, увеличение размера пиксела и, следовательно, зарядовой ёмкости не всегда приводит к увеличению динамического диапазона. Большой размер пиксела увеличивает вероятность возникновения различного рода помех, поэтому для увеличения динамического диапазона требуются дополнительные усилия по их нейтрализации.

Отношение размера датчика к количеству пикселей также влияет на уровень шума.

Более плотное расположение пикселей на датчике приводит к уменьшению размера пикселей и, соответственно, к увеличению вероятности появления электрических помех, что снижает динамический диапазон. Поэтому в профессиональных камерах используются датчики большего размера, которые и стоят дороже.

На сегодняшний день производители датчиков активно разрабатывают новые архитектуры устройств, которые смогли бы обеспечить высокую чувствительность, большой динамический диапазон и низкий уровень шумов при минимальном размере пикселей.

ЛИТЕРАТУРА

1. Curtin D.P. Image sensors. www.shortcourses.com.
2. Grotta S.W. Anatomy of digital camera: Image sensors. www.extremetech.com.
3. KODAK CCD Primer *KCP-001, Charge-coupled device (CCD) Image Sensors. www.kodak.com.

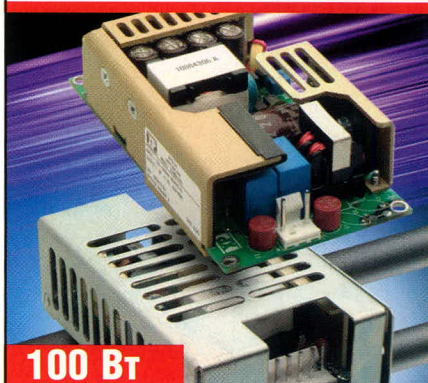
Источники питания AC/DC серии ЕСМ для промышленных и медицинских применений

- Соответствие требованиям UL/cUL 60601-1/EN60601-1
- Универсальный вход
- Предохранители в цепях нулевого и фазного проводов
- Ток утечки на землю менее 200 мкА
- Среднее время безотказной работы >250 000 часов



40/60 Вт

- Небольшие габариты 101,6 × 50,8 × 30,5 мм
- Одно-/двух-/трёхканальные модели с напряжениями 3,3...48 В
- Защита от поражения электрическим током: оборудование класса I и II
- КПД до 85%



100 Вт

- Габариты 114,3 × 63,5 × 30,5 мм
- Одно-/двух-/трёх-/четырёхканальные модели с напряжениями 3,3...48 В
- Защита от поражения электрическим током: оборудование класса I и II
- КПД до 85%



THE XP ERTS IN POWER

PROSOFT®

Тел./факс: (495) 234-0636/0640
info@prosoft.ru • www.prosoft.ru

Микроконтроллеры семейства LPC2000 (ARM7TDMI-S) от Philips – ещё один шаг к идеальной платформе для встраиваемых приложений

(часть 8)

Павел Редькин (г. Ульяновск)

В окончании статьи автор описывает работу контроллеров CAN, вспомогательных отладочных узлов процессорного ядра ARM, а также способы программирования встроенной Flash-памяти МК.

ОРГАНИЗАЦИЯ И ПРОГРАММИРОВАНИЕ FLASH-ПАМЯТИ

Загрузчик Flash-памяти

Встроенный загрузчик Flash-памяти МК LPC2000 управляет операциями начальной инициализации после сброса, а также является средством программирования Flash-памяти. Это может быть как начальное программирование чистого устройства, так и стирание, и перепрограммирование ранее уже запрограммированного устройства, а также программирование Flash-памяти прикладной пользовательской программой, выполняемое в составе приложения на базе LPC2000.

Встроенный загрузчик Flash-памяти имеет следующие основные особенности:

- внутрисистемное программирование (ISP): при внутрисистемном программировании встроенная Flash-память программируется и стирается с использованием программного обеспечения последовательного порта и загрузчика, которые резидентно находятся в памяти пользовательской системы;
- программирование в составе приложения (IAP): при программировании в составе приложения встроенная Flash-память программируется и стирается с использованием прикладного программного обеспечения пользователя.

Код загрузчика выполняется каждый раз после включения питания

или сброса устройства. Загрузчик может выполнять команды обработчика ISP или пользовательский прикладной код. Низкий уровень после сброса на выводе P0.14 расценивается логикой МК как внешний аппаратный запрос на запуск обработчика команд ISP. Если надлежащий тактовый сигнал присутствует на выводе XTAL1, когда на вывод RST поступает внешний положительный перепад, то устройству может потребоваться до 3 мс для того, чтобы опросить вывод P0.14 и принять решение о выполнении кода пользователя или о запуске обработчика команд ISP. Если на P0.14 подан внешний низкий уровень и установлен флаг переполнения сторожевого таймера, то внешний аппаратный запрос на запуск обработчика ISP логикой МК игнорируется. Если такой запрос отсутствует (на P0.14 высокий уровень), то автоматически производится поиск действительной (корректной) пользовательской программы. Если действительная пользовательская программа найдена, то ей передаётся управление. Если она не найдена, то вызывается подпрограмма автоматической установки скорости обмена (auto-baud), после чего вызывается обработчик ISP.

На вывод P0.14, находящийся в режиме высокого импеданса после сброса, необходимо подать соответствующий внешний уровень. В противном случае может произойти произвольный переход МК в режим ISP.

Критерий действительного пользовательского кода состоит в следующем. При действительной пользовательской программе зарезервированный вектор прерывания ARM, расположенный по адресу 0x0000 0014, должен содержать дополнение до двух контрольной суммы остальных векторов прерывания – так называемую действительную сигнатуру. В этом случае контрольная сумма всех векторов равна 0. Загрузчик кода отключает отображение векторов прерываний на загрузочный блок, затем вычисляет контрольную сумму векторов прерываний в секторе 0 Flash-памяти. Если вычисленная сумма соответствует заданной сигнатуре, то управление передаётся пользователю коду и в программный счётчик загружается адрес 0x0000 0000. Если сигнатура недействительна, то встроенная подпрограмма auto-baud производит синхронизацию скорости обмена с хостом через последовательный порт 0, после чего вызывается обработчик ISP.

Следует заметить, что действительная пользовательская сигнатура автоматически формируется при программировании Flash-памяти с помощью ISP-файла «прошивки» пользовательской программы.

Режим внутрисистемного программирования Flash-памяти ISP поддерживается прикладным программным обеспечением для хоста, свободно распространяемым производителем.

Некоторые команды ISP и IAP оперируют параметром «сектор» Flash-памяти и определяют номер сектора. Организация секторов Flash-памяти для устройств LPC2119/2129/2194/2292/2294 показана в табл. 85. Подпрограммы IAP, ISP и RealMonitor рас-

положены в загрузочном секторе. Команды ISP и IAP не позволяют производить операции записи/стирания/перехода (write/erase/ go) в загрузочном секторе.

Защита кода от чтения

Защита кода от чтения доступна в устройствах LPC2000 с загрузчиком версии 1.61 и выше. Защита разрешается путём программирования Flash-памяти по адресу 0x1FC (пользовательский сектор 0 Flash-памяти) значением 0x87654321 (десятичное значение 2271560481). Когда защита установлена, заблокированы порт отладки JTAG, внешняя начальная загрузка памяти и нижеперечисленные команды ISP:

- чтение памяти;
- запись в оперативную память;
- переход (Go);
- копирование оперативной памяти во Flash-память.

Защита кода от чтения не затрагивает команды IAP.

Команды ISP

Обработчик команд ISP оперирует командами, перечисленными в табл. 86. Для каждой команды обработчиком поддерживаются детальные коды возврата [1]. Обработчик команд ISP возвращает код возврата INVALID_COMMAND, когда получает неопределённую команду.

Команды IAP

Обработчик команд IAP оперирует командами, перечисленными в табл. 87. Поскольку в пользовательском приложении подпрограмму IAP нужно вызывать со словом указания в регистре R0, указывающим на память (SRAM), команда IAP включает в свой код параметры. Результат выполнения команды IAP возвращается в таблице результатов, на начало которой указывает регистр R1. Пользователь может многократно использовать командную таблицу результатов, передавая один и тот же указатель в регистрах R0 и R1. Таблица параметров должна быть достаточно большой, чтобы содержать все результаты в случае, если количество результатов превышает количество параметров. Количество параметров и результатов меняется в соответствии с командой IAP. Обработчик команды IAP возвращает код состояния INVALID_COMMAND, когда получена неопределённая команда.

ределённая команда. Подпрограмма IAP постоянно находится по адресу 0x7FFFFFF0 (код для режима Thumb).

При написании управляющей программы на языке C функцию IAP можно вызвать следующим образом.

Определить точку входа в IAP. Для режима Thumb определение точки входа будет выглядеть так:

```
#define IAP_LOCATION 0x7ffffff1

Определить структуру данных или указателей, чтобы передать таблицу команд IAP и таблицу результатов функции IAP:

unsigned long command[5];
unsigned long result[2];
```

Таблица 85. Организация Flash-памяти в устройствах LPC2119/2129/2194/2292/2294

Номер сектора	Адреса памяти и размеры сектора			
	для объёма памяти 128 К	размер сектора, К	для объёма памяти 256 К	размер сектора, К
0	0x0000 0000 – 1FFF	8	0x0000 0000 – 1FFF	8
1	0x0000 2000 – 3FFF	8	0x0000 2000 – 3FFF	8
2	0x0000 4000 – 5FFF	8	0x0000 4000 – 5FFF	8
3	0x0000 6000 – 7FFF	8	0x0000 6000 – 7FFF	8
4	0x0000 8000 – 9FFF	8	0x0000 8000 – 9FFF	8
5	0x0000 A000 – BFFF	8	0x0000 A000 – BFFF	8
6	0x0000 C000 – DFFF	8	0x0000 C000 – DFFF	8
7	0x0000 E000 – FFFF	8	0x0000 E000 – FFFF	8
8	0x0001 0000 – 1FFF	8	0x0001 0000 – FFFF	64
9	0x0001 2000 – 3FFF	8	0x0002 0000 – FFFF	64
10 (0x0A)	0x0001 4000 – 5FFF	8	0x0003 0000 – 1FFF	8
11 (0x0B)	0x0001 6000 – 7FFF	8	0x0003 2000 – 3FFF	8
12 (0x0C)	0x0001 8000 – 9FFF	8	0x0003 4000 – 5FFF	8
13 (0x0D)	0x0001 A000 – BFFF	8	0x0003 6000 – 7FFF	8
14 (0x0E)	0x0001 C000 – DFFF	8	0x0003 8000 – 9FFF	8
15 (0x0F)	0x0001 E000 – FFFF*	8	0x0003 A000 – BFFF	8
16 (0x10)			0x0003 C000 – DFFF	8
17 (0x11)			0x0003 E000 – FFFF*	8

* Загрузочный блок всегда постоянно находится на вершине доступной встроенной Flash-памяти. В случае устройства с Flash-памятью объёмом 128 Кб это 16-й сектор (сектор с табличным номером 15), а в случае устройства с Flash-памятью объёмом 256 Кб это 18-й сектор (сектор с табличным номером 17). Сектор Flash-памяти, где постоянно находится загрузочный блок, недоступен для хранения пользовательского кода.

Таблица 86. Сводная таблица команд ISP устройств LPC2000

Команда ISP	Формат команды
Разблокировка	U <Код разблокировки>
Установка скорости обмена	B <Скорость обмена> <Стоповый бит>
Эхо	A <Установка>
Запись в оперативную память	W <Начальный адрес> <Количество байтов>
Чтение памяти	R <Адрес> <Количество байтов>
Подготовить сектор(-а) к операции записи	P <Начальный номер сектора> <Конечный номер сектора>
Копировать оперативную память во Flash-память	C <Адрес Flash-памяти> <Адрес оперативной памяти> <Количество байтов>
Переход	G <Адрес> <Режим>
Стереть сектор(-а)	E <Начальный номер сектора> <Конечный номер сектора>
Заполнить пробелами сектор(-а)	I <Начальный номер сектора> <Конечный номер сектора>
Читать идентификатор устройства (ID)	J
Читать версию загрузчика кода	K
Сравнение	M <Адрес1> <Адрес2> <Количество байтов>

Таблица 87. Перечень команд IAP устройств LPC2000

Команда IAP	Код команды (дес.)
Подготовить сектор(-а) к операции записи	50
Копировать оперативную память во Flash-память	51
Стереть сектор(-а)	52
Заполнить пробелами сектор(-а)	53
Читать идентификатор устройства (ID)	54
Читать версию загрузчика кода	55
Сравнение	56

Таблица 88. Регистры приёмного фильтра CAN и центральные регистры блока CAN

Имя	Описание	Доступ	Значение после сброса	Адрес
AFMR	Регистр режима приёмного фильтра	R/W	1	0xE003 C000
SFF_sa	Регистр начального адреса стандартного индивидуального фрейма	R/W	0	0xE003 C004
SFF_GRP_sa	Регистр начального адреса стандартного группового фрейма	R/W	0	0xE003 C008
EFF_sa	Регистр начального адреса расширенного фрейма	R/W	0	0xE003 C00C
EFF_GRP_sa	Регистр начального адреса расширенного группового фрейма	R/W	0	0xE003 C010
ENDofTable	Регистр конца таблицы приёмного фильтра	R/W	0	0xE003 C014
LUTerrAd	Регистр адреса ошибки LUT	RO	0	0xE003 C018
LUTerr	Регистр ошибки LUT	RO	0	0xE003 C01C
CANTxSR	Центральный регистр состояния передачи CAN	RO	0x003F 3F00	0xE004 0000
CANRxSR	Центральный регистр состояния приёма CAN	RO	0	0xE004 0004
CANMSR	Центральный смешанный регистр состояния CAN	RO	0	0xE004 0008

Определить указатель типа функции, который передаёт два параметра и не возвращает ни одного. Обратите внимание, что IAP возвращает результат, базовый адрес которого постоянно находится в R1:

```
typedef void (*IAP)(unsigned int [], unsigned int []);
IAP iap_entry;
```

Установить указатель функции:

```
iap_entry=(IAP) IAP_LOCATION;
```

Всякий раз, когда требуется вызвать IAP, можно использовать следующую команду:

```
iap_entry (command, result);
```

Flash-память недоступна в ходе операций стирания или записи. Команды IAP, в результате которых производятся операции записи/стирания Flash-памяти, используют для их выполнения 32 байта в верхней части встроенной оперативной памяти. Пользовательская программа не

должна использовать эту область, если в приложении разрешается программирование Flash-памяти средствами IAP.

КОНТРОЛЛЕРЫ CAN И ПРИЁМНЫЕ ФИЛЬТРЫ

Блок CAN-интерфейса (Controller Area Network) МК семейства LPC2000 предназначен для того, чтобы поддерживать множество шин CAN одновременно, позволяя использовать устройство как шлюз, вентиль или маршрутизатор, подключенный к множеству шин CAN в промышленных или автомобильных приложениях.

Каждый контроллер CAN имеет регистр, структурно подобный регистру в устройствах Philips SJA1000 и PeliCAN Library block, однако 8-разрядные регистры этих устройств в LPC2000 скомбинированы в 32-разрядные слова, чтобы обеспечить доступ к ним в среде ARM. Главное эксплуатационное различие контроллера CAN LPC2000 и контроллеров вышеупомянутых устройств состоит в том, что распознавание принятых идентификаторов, называемое CAN-фильтрацией приёма, уда-

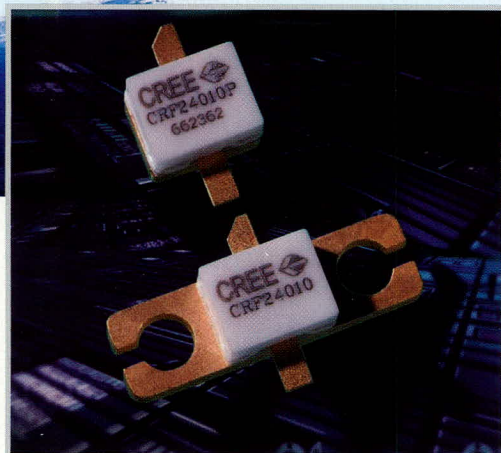
Таблица 89. Карта регистров контроллеров CAN1, CAN2, CAN3 и CAN4

Обобщённое имя	Описание	Доступ	CAN1	CAN2	CAN3	CAN4
			адрес/имя			
CANMOD	Управляет операционным режимом контроллера CAN	R/W	0xE004 4000/C1MOD	0xE004 8000/C2MOD	0xE004 C000/C3MOD	0xE005 0000/C4MOD
CANCMR	Содержит командные биты, которые затрагивают состояние контроллера CAN	WO	0xE004 4004/C1CMR	0xE004 8004/C2CMR	0xE004 C004/C3CMR	0xE005 0004/C4CMR
CANGSR	Глобальный регистр состояния контроллера и счётчиков ошибок	RO ¹	0xE004 4008/C1GSR	0xE004 8008/C2GSR	0xE004 C008/C3GSR	0xE005 0008/C4GSR
CANICR	Регистр состояния прерываний, захвата потери арбитража, захвата кода ошибки	RO	0xE004 400C/C1ICR	0xE004 800C/C2ICR	0xE004 C00C/C3ICR	0xE005 000C/C4ICR
CANIER	Регистр разрешения прерывания	R/W	0xE004 4010/C1IER	0xE004 8010/C2IER	0xE004 C010/C3IER	0xE005 0010/C4IER
CANBTR	Регистр интервала шины	R/W ²	0xE004 4014/C1BTR	0xE004 8014/C2BTR	0xE004 C014/C3BTR	0xE005 0014/C4BTR
CANEWL	Регистр предупреждения о лимите ошибок	R/W ²	0xE004 4018/C1EWL	0xE004 8018/C2EWL	0xE004 C018/C3EWL	0xE005 0018/C4EWL
CANSR	Регистр состояния	RO	0xE004 401C/C1SR	0xE004 801C/C2SR	0xE004 C01C/C3SR	0xE005 001C/C4SR
CANRFS	Регистр состояния принятого фрейма	R/W ²	0xE004 4020/C1RFS	0xE004 8020/C2RFS	0xE004 C020/C3RFS	0xE005 0020/C4RFS
CANRID	Регистр принятого идентификатора	R/W ²	0xE004 4024/C1RID	0xE004 8024/C2RID	0xE004 C024/C3RID	0xE005 0024/C4RID
CANRDA	Принятые байты данных 1–4	R/W ²	0xE004 4028/C1RDA	0xE004 8028/C2RDA	0xE004 C028/C3RDA	0xE005 0028/C4RDA
CANRDB	Принятые байты данных 5–8	R/W ²	0xE004 402C/C1RDB	0xE004 802C/C2RDB	0xE004 C02C/C3RDB	0xE005 002C/C4RDB
CANTF1	Информация о передаваемом фрейме (1)	R/W	0xE004 4030/C1TF1	0xE004 8030/C2TF1	0xE004 C030/C3TF1	0xE005 0030/C4TF1
CANTID1	Передаваемый идентификатор (1)	R/W	0xE004 4034/C1TID1	0xE004 8034/C2TID1	0xE004 C034/C3TID1	0xE005 0034/C4TID1
CANTDA1	Передаваемые байты данных 1–4 (1)	R/W	0xE004 4038/C1TDA1	0xE004 8038/C2TDA1	0xE004 C038/C3TDA1	0xE005 0038/C4TDA1
CANTDB1	Передаваемые байты данных 5–8 (1)	R/W	0xE004 403C/C1TDB1	0xE004 803C/C2TDB1	0xE004 C03C/C3TDB1	0xE005 003C/C4TDB1
CANTF2	Информация о передаваемом фрейме (2)	R/W	0xE004 4040/C1TF2	0xE004 8040/C2TF2	0xE004 C040/C3TF2	0xE005 0040/C4TF2
CANTID2	Передаваемый идентификатор (2)	R/W	0xE004 4044/C1TID2	0xE004 8044/C2TID2	0xE004 C044/C3TID2	0xE005 0044/C4TID2
CANTDA2	Передаваемые байты данных 1–4 (2)	R/W	0xE004 4048/C1TDA2	0xE004 8048/C2TDA2	0xE004 C048/C3TDA2	0xE005 0048/C4TDA2
CANTDB2	Передаваемые байты данных 5–8 (2)	R/W	0xE004 404C/C1TDB2	0xE004 804C/C2TDB2	0xE004 C04C/C3TDB2	0xE005 004C/C4TDB2
CANTF3	Информация о передаваемом фрейме (3)	R/W	0xE004 4050/C1TF3	0xE004 8050/C2TF3	0xE004 C050/C3TF3	0xE005 0050/C4TF3
CANTID3	Передаваемый идентификатор (3)	R/W	0xE004 4054/C1TID3	0xE004 8054/C2TID3	0xE004 C054/C3TID3	0xE005 0054/C4TID3
CANTDA3	Передаваемые байты данных 1–4 (3)	R/W	0xE004 4058/C1TDA3	0xE004 8058/C2TDA3	0xE004 C058/C3TDA3	0xE005 0058/C4TDA3
CANTDB3	Передаваемые байты данных 5–8 (3)	R/W	0xE004 405C/C1TDB3	0xE004 805C/C2TDB3	0xE004 C05C/C3TDB3	0xE005 005C/C4TDB3

¹В счётчики ошибок может быть произведена запись только тогда, когда установлен бит RM в регистре CANMOD.

²В эти регистры может быть произведена запись только тогда, когда установлен бит RM в регистре CANMOD.

Технологии будущего уже сегодня...



Уникальные высокотемпературные радиационно-стойкие SiC и GaN СВЧ-транзисторы

Наименование параметра / Тип прибора	SiC СВЧ-транзисторы		GaN СВЧ-транзисторы				
	CRF24010	CRF24060	CGN27015	CGN35015	CGN35050	CGN40010	CGN40045
Выходная мощность, Вт	10	60	15	15	30	10	45
Диапазон частот, ГГц	0...2,7	0...2,7	2,3...2,9	3,3...3,9	3,3...3,9	до 4	до 4
Рабочее напряжение, В	28...48	28...48	28	28	28	28	28
Типовой КПД, %	45	45	25	24	23	65	55
Типовое усиление, дБ	15	13	14,5	12	11	14,5	12
Максимальное напряжение сток-исток, В	100	120	84	84	84	84	84
Рабочая температура перехода, °С	255	255	175	175	175	175	175

- Рекордная плотность мощности СВЧ на кристалле
- Высокое напряжение питания упрощает цепи согласования
- Сверхвысокая полоса частот усиления
- Высокая эффективность
- Высокое усиление
- **Наработка на отказ (MTTF) SiC СВЧ-приборов:**
2,2 млн. часов при +225°C,
60 млн. часов при +175°C

ПРОСОФТ – официальный дистрибьютор компании Cree

ПРОСОФТ – АКТИВНЫЙ КОМПОНЕНТ ВАШЕГО БИЗНЕСА

Телефон: (495) 232-2522 • E-mail: sale@cree.ru • Web: www.xlight.ru
www.prochip.ru

реклама

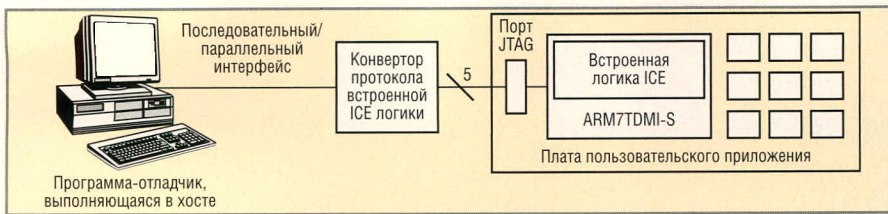


Рис. 32. Блок-схема среды отладки, использующей ICE

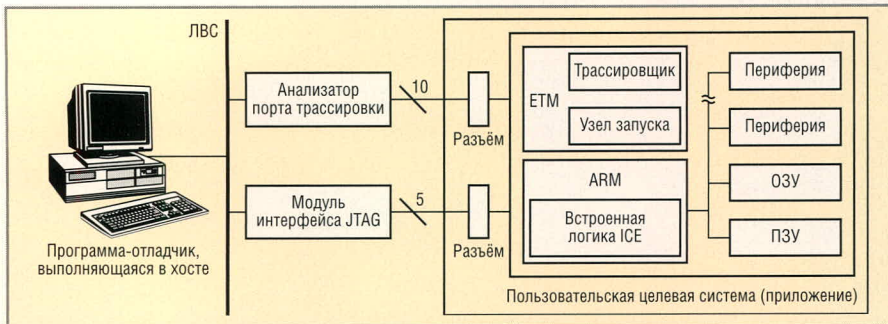


Рис. 33. Блок-схема среды отладки с использованием макроячейки ETM

лено из контроллеров CAN LPC2000 и централизованно размещено в так называемом глобальном приёмном фильтре, входящем в состав МК.

Контроллеры CAN-устройств семейства LPC2000 имеют следующие основные особенности:

- 2 или 4 (в зависимости от устройства) контроллера CAN и шины;
- скорости передачи данных до 1 Мбит/с в каждой шине;
- 32-разрядный регистр и доступ к оперативной памяти;
- CAN-интерфейс, совместимый со спецификацией 2.0B, ISO 11898-1;

- глобальный приёмный фильтр распознаёт 11- и 29-разрядные принимаемые идентификаторы для всех шин CAN;
- приёмный фильтр CAN обеспечивает автоматическое распознавание в стиле FullCAN выбранных стандартных идентификаторов.

Контроллеры CAN МК LPC2000 обслуживаются выводами RX1 – RX4 (последовательные входы) и TX1 – TX4 (последовательные выходы). Регистры блока CAN перечислены в табл. 88 и 89. Более детальные их описания, а также описания операций контроллера CAN

и глобального приёмного фильтра подробно рассмотрены в книге [1].

Встроенная логика отладки (ICE)

Устройства семейства LPC2000 имеют в своём составе аппаратный модуль логики отладки ICE, включающий в себя программный отладчик со следующими основными особенностями:

- отладчику не требуется никаких системных ресурсов, чтобы запустить сеанс отладки;
- отладчик позволяет взаимодействовать непосредственно с ядром через порт JTAG;
- команды «вставляются» непосредственно в ядро ARM7TDMI-S;
- состояние ядра ARM7TDMI-S или состояние системы может быть исследовано, сохранено или изменено в зависимости от типа «вставленной» команды;
- отладчик позволяет ускорять медленно выполняющиеся при отладке команды до более быстрого выполнения на системной скорости.

Встроенная логика отладки ICE обеспечивает поддержку отладки на кристалле. Отладка пользовательской системы требует выполнения программного обеспечения отладчика в хосте и наличия конвертора протокола ICE. Конвертор протокола ICE преобразует команды удалённого от-

Таблица 90. Выводы микроконтроллера, используемые встроенной ICE-логикой

Имя вывода	Тип вывода	Описание
TMS	Вход	Выбор тестового режима. Сигнал на выводе TMS выбирает следующее состояние в конечном автомате TAP
TCK	Вход	Тестовая синхронизация. Сигнал позволяет сдвигать (защёлкивать) входные данные на входах TMS и TDI. Этот сигнал представляет собой тактовые импульсы, по положительному фронту которых защёлкивается сигнал TMS. Сигнал TCK также определяет внутреннее состояние устройства
TDI	Вход	Тестовый вход данных. Это вход последовательных данных для сдвигового регистра
TDO	Выход	Тестовый выход данных. Это выход последовательных данных из сдвигового регистра. Данные выводятся из устройства по отрицательному фронту сигнала TCK
nTRST	Вход	Тестовый сброс. Вывод nTRST может использоваться, чтобы сбросить тестовую логику в составе встроенной ICE-логики
RTCK	Выход	Возвращаемая тестовая синхронизация. Дополнительный сигнал, который добавлен к порту JTAG. Требуется для проектов, основанных на процессорном ядре ARM7TDMI-S. Система Multi-ICE (система разработки программ ARM) использует этот сигнал для обслуживания синхронизации в случае меняющейся частоты синхронизации*

* Для уточнения подробностей рекомендуется обратиться к Multi-ICE System Design considerations Application Note 72 (ARM DAI 0072A)

Таблица 91. Описание выводов встроенной макроячейки трассировки (ETM)

Имя вывода	Тип вывода	Описание
TRACECLK	Выход	Тактирование трассировки. Сигнал тактирования трассировки обеспечивает тактовые импульсы для порта трассировки. Сигналы PIPESTAT[2:0], TRACESYNC и TRACERPKT[3:0] синхронизированы к положительному перепаду сигнала тактирования трассировки. Этот сигнал не генерируется блоком ETM, а должен быть получен из системного тактового сигнала. Частота тактовых импульсов должна быть подобрана так, чтобы обеспечивать достаточное время задержки для сигналов данных трассировки. Таким образом, поддерживается половинная скорость режима тактирования. Сигналы данных трассировки сдвигаются в соответствии с фазой сигнала TRACECLK*
PIPESTAT[2:0]	Выход	Состояние конвейера. Сигналы состояния конвейера обеспечивают индикацию из цикла в цикл стадии выполнения команды находится конвейера процессора
TRACESYNC	Выход	Синхронизация трассировки. Сигнал синхронизации трассировки используется, чтобы указать первый пакет группы пакетов трассировки, и находится в высоком уровне только для первого пакета любого адреса перехода
TRACERPKT[3:0]	Выход	Пакет трассировки. Сигналы пакета трассировки используются для вывода упакованной информации адреса и данных, связанных с состоянием конвейера. Все пакеты имеют длину восемь битов. Пакет выводится более чем за два цикла. В первом цикле выводятся биты пакета [3:0], а во втором цикле – биты пакета [7:4]
EXTINT[0]	Вход	Вход внешнего запуска (триггерный вход)

* Для получения дополнительной информации рекомендуется обратиться к документам ETM7 Technical Reference Manual (ARM DDI 0158B) и Embedded Trace Macrocell Specification (ARM IHI 0014E)

ладочного протокола в данные JTAG, которые, в свою очередь, обращаются к ядру ARM7TDMI-S пользовательской системы.

Блок-схема среды отладки приведена на рис. 32.

Отладочная архитектура ARM7TDMI-S использует встроенный порт JTAG в качестве инструмента обращения к ядру. Цепи сканирования (просмотра) подключены к ядру для заводского тестирования, они же используются в режиме отладки, чтобы фиксировать информацию на шине данных и вставлять новую информацию в ядро или в память. Выводы МК, используемые для встроенной логики отладки ICE, перечислены в табл. 90.

ВСТРОЕННАЯ МАКРОКРЕЙКА ТРАССИРОВКИ (ETM)

Встроенная макроячейка трассировки ETM, входящая в состав уст-

ройств LPC2000, может отслеживать команды, выполняемые ядром ARM. При использовании внутренней памяти устройств LPC2000 отсутствует возможность отслеживать операции, производимые ядром МК, непосредственно через выводы устройства. Однако макроячейка ETM обеспечивает возможность трассировки (мониторинга выполняемых команд) в реальном времени ядра. ETM выводит информацию о выполнении команд процессором в порт трассировки. Программный отладчик позволяет конфигурировать ETM, а использование интерфейса JTAG и дисплея (хоста) обеспечивает возможность представить информацию трассировки, получаемую ETM, в формате, удобном для восприятия пользователем.

Блок-схема среды отладки с использованием макроячейки ETM приведена на рис. 33. Описание выводов ETM приведено в табл. 91.

ЛИТЕРАТУРА

1. Редькин П.П. Микроконтроллеры ARM7. Семейство LPC2000 компании Philips: полное руководство пользователя. М.: Додэка-XXI. 2006 (готовится к выпуску).
2. LPC2119/2129/2194/2292/2294 USER MANUAL. Feb. 2004. Philips Semiconductors (www.LPC2000.com).
3. ARM 7TDMI Data Sheet. Document Number: ARM DDI 0029E. Issued: August 1995. Copyright Advanced RISC Machines Ltd. (ARM) 1995 (www.arm.com).
4. ARM7TDMI-S Technical Reference Manual (Rev 4) Copyright © ARM Limited. ARM DDI 0234A (www.arm.com).
5. ARM Instruction Set. Quick Reference Card (www.arm.com).
6. Thumb Instruction Set. Quick Reference Card (www.arm.com).
7. 80C51 Family Derivatives 8XC552/562 Overview. Philips Semiconductors, 1996 Aug 06, (www.semiconductors.philips.com/acrobat/various/8XC552_562OVERVIEW_2.pdf).



Новости мира News of the World Новости мира

Повышающий преобразователь для светодиодов

Преобразователь PR4402 фирмы Prema обеспечивает возможность работы белых светодиодов с токами до 40 мА при питании от одного батарейного элемента. Величина тока устанавливается выбором индуктивности. С PR4402 в корпусе SOT23 могут работать и белые, и цветные светодиоды в одноэлементных приложениях от 0,9 В. Работа от одного элемента позволяет избежать при использовании никель-металлгидридных аккумуляторов часто возникающего при глубоком разряде нескольких элементов изменения полярности и повреждения элемента питания.

prema.com

Сильноточные индуктивности для планарного монтажа

Индуктивности планарного монтажа SIMID 1210-H фирмы Epcos предлагаются в диапазоне от 1 мкГ до 680 мГ при величине тока нагрузки до 1,15 А. Благодаря использованию особой технологии лазерной сварки для соединения обмотки с выводными контактами создаётся теплостойкое соединение, обеспечивающее возможность работы при температуре до 150°C. Полная заливка LCP (Liquid Crystal Polymer) обеспечивает высокую механическую стабильность. Элементы не только соответствуют требованиям директивы RoHS, но и

допускают бессвинцовый профиль пайки согласно нормам JEDEC J-STD 020C.

epcos.com

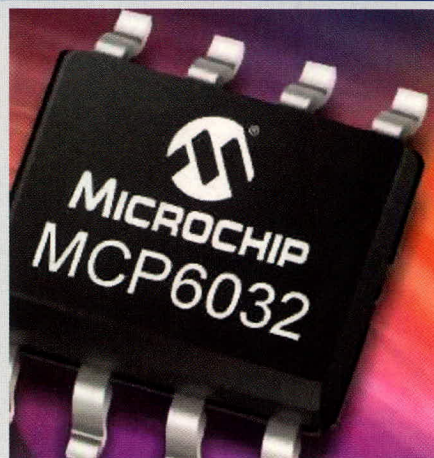
Интегрированный повышающий преобразователь с диапазоном напряжения 9...33 В

КПД преобразователя FAN5336 фирмы Fairchild Semiconductor составляет, по данным производителя, 87%. Преобразователь, синхронизированный частотой 1,5 МГц, имеет диапазон выходного напряжения 9...33 В и содержит коммутационный NFET. Элемент выдаёт пиковые значения тока до 1,5 А, дополнительными возможностями являются ограничение тока в каждом цикле коммутации и опция мягкого старта. FAN5336 предлагается в корпусах UMLP размером (3 × 3 × 0,6 мм) или MLP (3 × 3 × 0,8 мм) без содержания свинца.

fairchildsemi.com

Операционные усилители в одинарной, сдвоенной и четверённой версиях

Компания Microchip расширила свои предложения линейных компонентов операционными усилителями MCP6031, MCP6032, MCP6033 и MCP6034. Операционные усилители предлагаются в одинарной, сдвоенной и четверённой версиях и имеют ток покоя 900 нА и ширину полосы 10 кГц. Усилители предназначены



для переносных приборов и других электронных приборов в области медицинской техники, в промышленных приложениях и в бытовых приборах.

Энергосберегающая CMOS-технология и энергонезависимая память для точной компенсации в корпусе обеспечивают напряжение смещения MCP603X 150 мкВ при 25°C. Rail-to-Rail-структура входов-выходов обеспечивает большую динамику и мощность во всём диапазоне рабочих напряжений. MCP603X поддерживаются бесплатным программным обеспечением FilterLab. Оно формирует полные схемные диаграммы контуров фильтров со значениями компонентов и кривыми частотных характеристик. MCP6031, MCP6032 и MCP6033 имеются в 8-контактных корпусах MSOP и SOIC, а MCP6034 – в 14-контактных корпусах SOIC и TSSOP.

microchip.com

СВЧ-фильтры и оборудование на их основе фирмы Pole/Zero

(часть 2)

Андрей Лакно (Москва)

В статье приведён обзор СВЧ-фильтров с цифровой настройкой частоты, а также комплексно интегрированного оборудования (ICE) фирмы Pole/Zero, которое имеет широкую область применения как в военной, так и в гражданской сфере. Рассмотрены различные серии комплексно интегрированного оборудования и фильтров, приводятся их характеристики и область применения.

КОМПЛЕКСНО-ИНТЕГРИРОВАННОЕ ОБОРУДОВАНИЕ

Комплексно-интегрированное оборудование (ICE) является уникальной в своём роде разработкой Pole/Zero и представляет собой эволюцию стандартных фильтров с использованием различных элементов для достижения высокоинтегрированных решений. ICE обычно включает в себя кас-

кад настраиваемых фильтров и малошумящих усилителей для обеспечения превосходной селективности по всему диапазону настройки.

Линейка продуктов ICE разделена на пять серий: ICE1000, ICE2000, ICE3000, ICE4000, ICE5000.

Серия ICE1000

Серия ICE1000 обычно используется там, где требуется широкий диапа-

зон частот в маленьком корпусе. ICE1000 обеспечивает минимальную величину фильтрации и широкий диапазон настройки.

ICE1001 (рис. 7) представляет собой универсальный высокочастотный преселектор и содержит три фильтра Mini-Pole с диапазоном частот 1,5...4, 4...10 и 10...30 МГц.

Серия ICE1002 (рис. 8) представлена пре-/постселектором, который представляет собой быстро перестраиваемый фильтр диапазона 30 МГц...2 ГГц с высокочастотным каналом транзитной передачи.

Серия ICE1003 (рис. 9) представлена широкополосным преселектором с цифровой настройкой полосы фильтрации 20 МГц...2 ГГц и 2...3 ГГц.

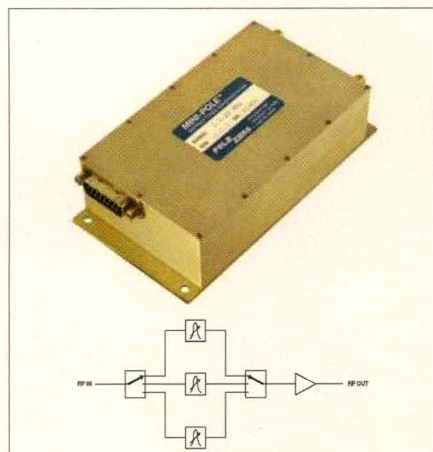


Рис. 7. Внешний вид и структурная схема преселектора серии ICE1001

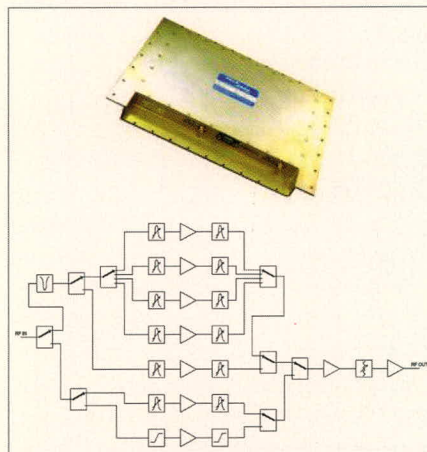


Рис. 9. Внешний вид и структурная схема широкополосного преселектора серии ICE1003

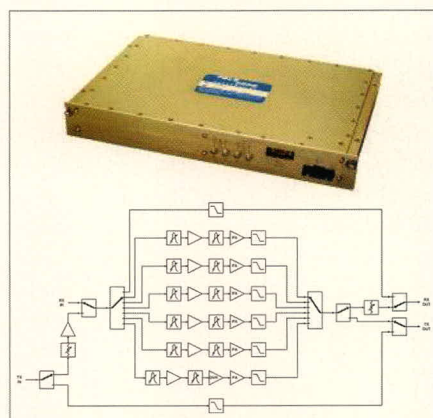


Рис. 8. Внешний вид и структурная схема пре-/постселектора серии ICE1002

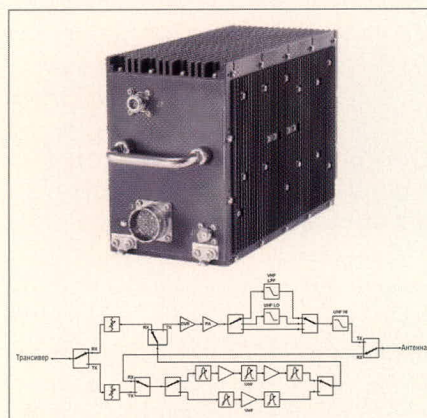


Рис. 10. Внешний вид и структурная схема приёмопередающего фильтра ICE2001

Серия ICE2000

Изделия серии ICE2000 обеспечивают среднюю селективность в диапазоне метровых и дециметровых волн. Они объединяют передающие и приёмные переключатели, а также усилитель мощности, обеспечивающий необходимый уровень передаваемого сигнала.

ICE2001 – перестраиваемый приёмопередающий фильтр, работающий в диапазоне как метровых, так и дециметровых волн, на частоте 118...400 МГц (рис. 10). Это изделие является одним из многих сертифицированных для применения в авиационной технике.

Серия ICE3000

Продукция серии ICE3000 обеспечивает превосходную селективность преимущественно путём применения последовательно соединённых усилителей и фильтров. В ICE3000 реализованы следующие возможности: регулировка коэффициента усиления приёмного тракта, низкий шум, высокое усиление в тракте передачи и другие усовершенствования. Вся продукция серии оптимизирована для применения в бортовой авиационной технике.

ICE3001 – перестраиваемый приёмопередающий фильтр для центров спутниковой связи, работающий в диапазоне частот 243...318 МГц (рис. 11).

ICE3002 – перестраиваемый принимающий фильтр для центров спутниковой связи, работающий в диапазоне частот 243...270 МГц (рис. 12). Этот элемент сертифицирован организацией JТС (Joint Interoperability Test Command), отвечающей за проверку совместимости всех сетевых решений, используемых агентствами, входящими в Министерство обороны США.

ICE3003 – перестраиваемый передающий фильтр/усилитель для центров спутниковой связи, работающий в диапазоне частот от 292 до 318 МГц (рис. 13), сертифицирован JТС.

ICE3004 – перестраиваемый приёмопередающий фильтр/усилитель УВЧ-диапазона с режимом транзитной передачи, работающий в диапазоне частот 225...400 МГц (рис. 14). Данное устройство работает напрямую с приёмопередатчиком ARC-210, оптимизирован для применения в авиационной технике, а также широко используется на наземных транспортных средствах.

Компания Pole/Zero с целью предоставления качественной продукции ввела следующую политику качества: «Компания Pole/Zero специализируется на своевременном предоставлении высококачественной продукции клиентам внутри страны и за её пределами. Удовлетворение клиента качеством продукции и постоянное её совершенствование есть личная задача каждого сотрудника».

Придерживаясь такой политики, компания Pole/Zero сфокусирована на повышении эффективности всей компании. В результате в октябре 2005 г. Pole/Zero успешно прошла сертификацию независимой организацией на соответствие стандартам группы ISO 9001:2000 и AS9100:2004, Редакция В (на управление качеством в исследованиях, проектировании, разработке, изготовлении и послепродажном обслуживании).

Вся продукция производится и тщательно проверяется на производственном комплексе в Вест Честере,

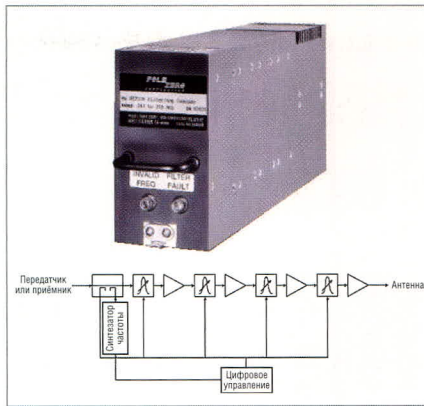


Рис. 11. Внешний вид и структурная схема перестраиваемого приёмопередающего фильтра ICE3001

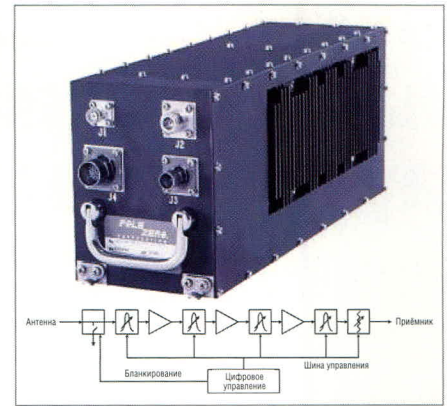


Рис. 12. Внешний вид и структурная схема перестраиваемого принимающего фильтра ICE3002

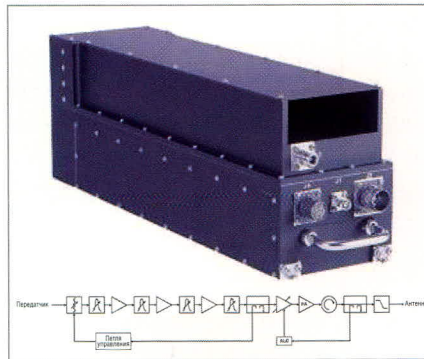


Рис. 13. Внешний вид и структурная схема перестраиваемого передающего фильтра ICE3003

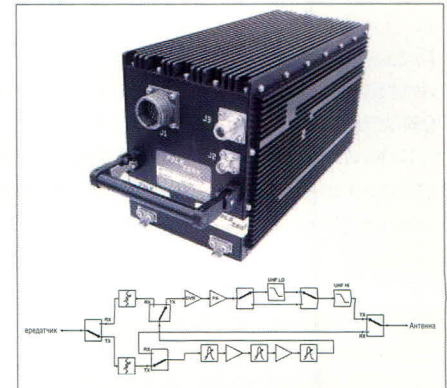


Рис. 14. Внешний вид и структурная схема перестраиваемого фильтра/усилителя ICE3004

штат Огайо, который оснащён самым современным производственным и тестовым оборудованием.

Компания Pole/Zero нацелена на новаторство, качество и абсолют-

ное удовлетворение потребностей клиентов. Более подробную информацию о продукции компании можно получить на сайте: www.polezero.com. ©

Pole/Zero СВЧ-фильтры военного и аэрокосмического применения и оборудование на их основе.

СВЧ-фильтры с цифровой настройкой

Преселекторы

Комплексно - интегрированное оборудование

реклама

АППАРАТУРА СИСТЕМ СВЯЗИ

125363, МОСКВА, ШТУРВАЛЬНАЯ УЛ., Д. 3, СТ. 1.
 ТЕЛ./ФАКС: (495) 492-4022, 492-7077, 105-5012
 E-MAIL: KLYUCHNICOVA@ESCLTD.RU

Асинхронное преобразование частоты дискретизации аудиопотоков стандарта AES

Печатается с разрешения фирмы Xilinx (www.xilinx.com)

Грег Хокс, Рид Тидвел, Джон Ф. Сноу (США)

Перевод Игоря Метина

От современного AVB-оборудования (AVB – Audio, Video, Broadcast) требуется лучшее качество изображения, более высокое разрешение, более широкая полоса пропускания, больше аудио-/видеоканалов и объединение в одном устройстве ранее реализуемых отдельно, но связанных между собой функций, таких как интерфейс HD-SDI, аудиоумножитель, аудиодемультимплексор, а также асинхронное преобразование с передискретизацией. В статье описываются области применения ПЛИС Virtex-5 Xilinx, которая представляет собой оптимальную платформу для внедрения алгоритмов передискретизации аудиопотоков стандарта AES.

ПЛИС Xilinx® идут в ногу с требованиями логической многофункциональности, поскольку включают в себя набор свойств, которые позволяют вытеснить менее функциональные, сложные и дорогие микросхемы ASSP (Application Specific Standard Product – стандартные изделия со специфической областью применения). Одна из таких функций микросхемы ASSP, а именно ASRC (Asynchronous Sample-Rate Conversion – асинхронное преобразование скорости выборки), может быть интегрирована в ПЛИС Xilinx. Эта функция может эффективно использовать возможности секций DSP48E и блочные ОЗУ для создания сложных фильтров.

Бесплатно распространяемые фирмой Xilinx руководства по применению и примеры реализованных проектов также помогают разработчи-

кам в реализации сложных алгоритмов. Примеры проектов, реализующих функции ASRC, корректно управляют синхронным преобразованием с передискретизацией, а также намного более сложными функциями ASRC, требующимися в большинстве аудио-/видеоприложений. Более простые «только синхронные» методы, предлагаемые разработчиками микросхем ASSP и поставщиками IP-ядер для ПЛИС, могут быть экономичнее по использованию ресурсов на аудиоканал, однако, при некорректном применении в асинхронном приложении, эти методы имеют одно из следующих искажений или оба сразу:

- изменения латентности от входа к выходу из-за накапливающейся задержки;
- искажения в аудиопотоке, такие как пропуск выборок или повторяющиеся выборки.

Оба эти случая представляют собой нежелательные искажения.

Понятие передискретизации

Прежде чем углубиться в теорию цифрового преобразования с передискретизацией, следует рассмотреть основные типы проблем, которые пытаются решить разработчики аудио-/видеооборудования. Существует несколько приложений, в кото-

рых можно использовать синхронное преобразование фиксированной частоты, например, 48-кГц входной сигнал, преобразованный в выходной 44,1-кГц с тем же источником синхроимпульсов или выходными синхроимпульсами, произведенными от входного генератора синхроимпульсов. Однако более вероятен асинхронный случай, когда входной и выходной генераторы синхроимпульсов полностью независимы, как в случае двух плат, обменивающихся аудиоданными между собой. Их независимые генераторы синхроимпульсов могут иметь одинаковую номинальную частоту, но отличаться на несколько импульсов на миллион (ppm).

Пример реализованного Xilinx проекта ASRC для асинхронного случая независимых входных и выходных синхроимпульсов обеспечивает две важные и сложные функции:

- автоматически и точно контролирует соотношение входа к выходу и изменения частоты дискретизации;
- адаптирует фильтрующую функцию (коэффициенты фильтра) «на ходу», для того чтобы обеспечить наиболее высокое быстродействие.

Поддержка в ПЛИС функции ASRC для цифрового аудио означает, что теперь можно значительно снизить стоимость каждого интерфейса SDI в вашей системе, а также в других системах с большим количеством каналов.

Разработанное Xilinx IP-ядро ASRC имеет высокое быстродействие и отношение сигнал-шум от входа к выходу не менее 125 дБ. Ядро также поддерживает преобразование нескольких аудиосигналов с различными входными частотами в несколько выходных аудиосигналов с различными выходными частотами. Алгоритм контроля соотношения входа к выходу подстраивается в процессе рабо-

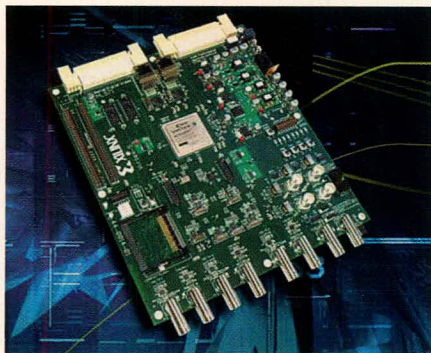


Рис. 1. Отладочно-демонстрационная плата ML571

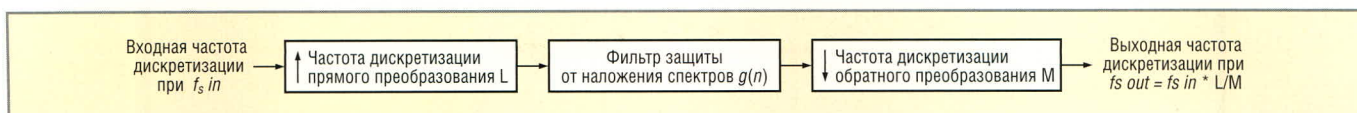


Рис. 2. Классическая схема тракта данных при изменении частоты дискретизации

ты, обеспечивая тем самым высокое быстродействие, и не требует при этом специального внимания к входным и выходным синхроимпульсам. Можно проверить всё вышесказанное об IP-ядре, запустив его на демонстрационной плате ML571 фирмы Xilinx, которая представлена на рис. 1. При этом многофункциональное и высокопроизводительное ASRC IP-ядро является бесплатным.

ТЕОРИЯ ПЕРЕДИСКРЕТИЗАЦИИ

Рисунок 2 иллюстрирует общий случай повышающего и понижающего преобразования. Отношение частот может постоянно повышаться или понижаться на величину рационального числа с дробной частью. Схема на рис. 2 демонстрирует процесс повышающего преобразования (создается больше отсчётов дискретизации и временных точек для выбора), за которым следует понижающее преобразование (выбираются отсчёты выходного потока, которые наиболее близко соответствуют отсчётам входного потока). Фильтр защиты от наложения спектров в центре тракта данных гарантирует ширину спектра вдвое меньше частоты дискретизации по критерию Найквиста, как на входе, так и на выходе.

На рис. 3 и 4 показано, что для каждой точки частоты дискретизации или фазы на выходе требуется различный набор коэффициентов вспомогательных фильтров, так как входы располагаются в различных точках относительно фазы на выходе. Вспомогательный фильтр, имеющий набор коэффициентов, которые выравниваются с положениями выходных отсчётов на входе, формируется путём интерполяции коэффициентов фильтра-прототипа. Когда вспомогательный фильтр свёртывается с соответствующей входной частотой дискретизации, формируется требуемая частота дискретизации на выходе. Этот процесс повторяется с новыми коэффициентами вспомогательного фильтра, интерполированными для каждого выходного отсчёта.

ПРИМЕР ФУНКЦИИ ASRC, РЕАЛИЗОВАННОЙ В ПЛАТЕ ML571

Простая функция, известная как кадровая синхронизация видео, является хорошим примером того, где можно использовать функцию ASRC. Видео можно сохранять в буфере кадров с некоторой частотой и удалять с немного отличающейся частотой. Этот процесс может быть полезным, если два видеоустройства не «синхронизированы» и работают с различными частотами пикселей.

В результате появляется проблема: время от времени необходимо добавить или вырезать кадр видеоданных. Человеческий глаз, возможно, не заметит добавленного или вырезанного видеокadra на телеэкране, но человеческий слух очень хорошо определяет вырезанный или добавленный фрагмент аудио. Решением проблемы является удаление видео из начального видеопотока и вставка его в конечный видеопоток с немного отличающейся частотой, а также согласование частоты аудио на выходе с новой частотой видео на выходе. Реализованный фирмой Xilinx ASRC-проект идеально подходит для этой задачи.

В качестве примера соединим две платы с SDI видео, работающим с немного отличающимися частотами, поскольку на каждой плате размещён свой генератор синхроимпульсов. Принимающая плата выполняет демультимплексирование аудиосигналов стандарта AES из видеопотока и посылает его на ASRC. Разница в частоте синхро-

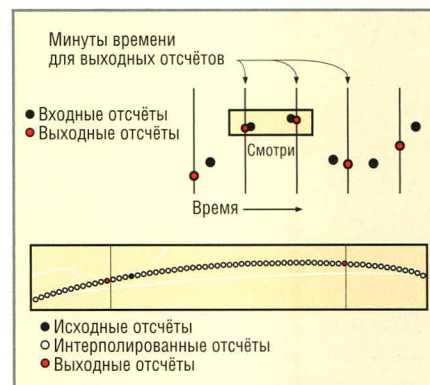


Рис. 3. Положение выходного отсчёта относительно оригинального определяет, какой из интерполированных отсчётов будет использован

импульсов между двумя платами заставляет логику синхронизации буфера кадров добавлять или вырезать видеокadры. ASRC регулирует декодирование аудио для согласования с частотой синхроимпульсов видеопотока на выходе, где оно может быть снова закодировано в SDI видеопоток на выходе.

Подробнее о синхронизации буфера кадров и методике асинхронного преобразования с фиксированной частотой дискретизации можно прочитать в XAPP514, Audio/Video Connectivity Solutions for the Broadcast Industry (www.xilinx.com/bvdocs/appnotes/xapp514.pdf).

Блок-схема и спецификация

На рис. 5 простая схема иллюстрирует два ключевых элемента, необходимых при реализации проекта ASRC. Первым элементом является

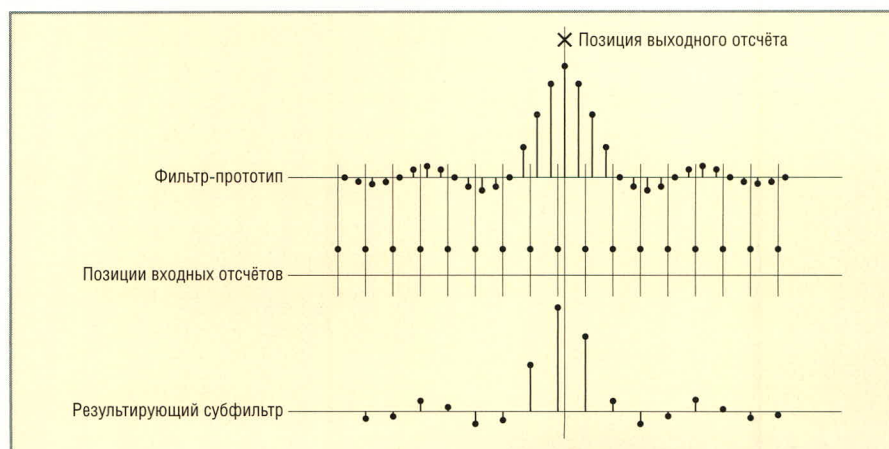


Рис. 4. Фильтр-прототип центрирован относительно выходного отсчёта

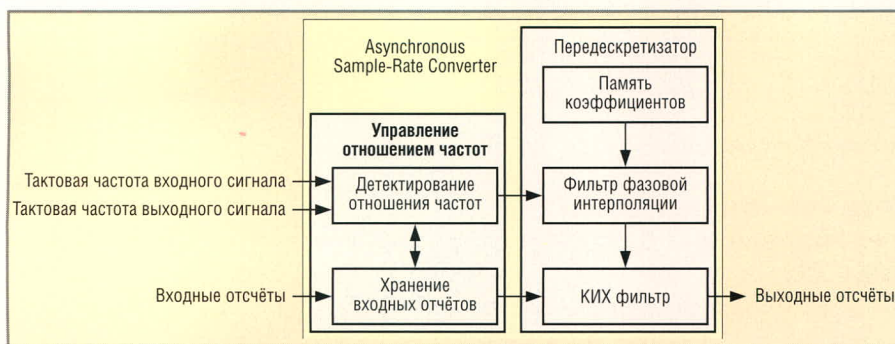


Рис. 5. Блок-схема ASRC референс-проекта

определение изменений между частотой дискретизации на входе и необходимой частотой дискретизации на выходе, он называется «управление частотой». Второй элемент в рамках «восстановления частоты» – набор прототипных фильтров, модифицированных в зависимости от статистики, сообщаемой устройством управления частотой.

Реализованный ASRC-проект преобразует стереоаудиосигнал из одной частоты дискретизации в другую. Частоты дискретизации на входе и на выходе могут быть произвольной частью друг друга или одной частоты, но базироваться на различных синхросигналах.

Выходной сигнал представляет собой вариант входного сигнала с ограниченным частотным диапазоном и изменённой частотой дискретизации. Реализованный ASRC-проект имеет следующие характеристики:

- полностью асинхронное функционирование;
- расширяемость до нескольких каналов;
- суммарное значение коэффициента нелинейных искажений + шум не более –125 дБ, типовое значение –130 дБ;
- размер аудиослова 24 бит на входе и на выходе, с внутренней математической точностью 31 бит и симметрией относительно нуля;
- автоматический контроль частоты дискретизации от входа к выходу с непрерывной модификацией фильтра;
- отношение частот при повышении частоты до 8 : 1;
- отношение частот при понижении частоты до 1 : 7,5;
- непрерывный контроль частоты дискретизации от входа к выходу с адаптивной фильтрацией;
- частоты на входе/выходе 8...92 кГц;

• мало детерминированная латентность.

Реализованный проект имеет фильтр с конечной импульсной характеристикой с интерполированным коэффициентом, реализованный на базе секций DSP48E ПЛИС Xilinx Virtex™-5 в качестве основного логического элемента и блочного ОЗУ для буферов дискретизации на входе и хранения прототипов.

ЗАКЛЮЧЕНИЕ

Необходимость поддерживать различные частоты дискретизации аудиосигнала от входа к выходу для меняющегося количества цифровых аудиоканалов и поддерживать новые функции AVB является очень трудной и актуальной задачей. Из-за огромного различия постоянно меняющихся протоколов, необходимости управления памятью, различной пропускной способности и разнообразия различных системных интерфейсов становится очевидным, что при реализации подобных проектов возникают требования быстрого действия, функциональной гибкости и при этом экономичности, которые микросхемы ASSP и ASIC не могут обеспечить. Такого рода задачи открывают возможности для ПЛИС типа Virtex-5, поскольку они способны обеспечить решения для постоянно развивающегося рынка AVB-оборудования. ©

Новости мира News of the World Новости мира

Новый оптический кабель свяжет США и страны Азии

С самого начала глобального распространения Интернета было ясно, что его кабельная основа является самым «узким местом». Так, в декабре 2006 г. землетрясение, произошедшее на морском дне недалеко от побережья Тайваня, оборвало подводные коммуникации и вызвало тяжелейшие проблемы со связью с Северо-Американским континентом. Новый пакт, подписанный 17 телекоммуникационными компаниями, призван помешать повторению подобных сценариев в будущем. Согласно этому пакту, планируется проложить подводный оптический кабель стоимостью в \$500 млн. между побережьями США и Юго-Восточной Азии по пути, идущему в обход основных разломов тихоокеанского «огненного кольца».

Проект, получивший наименование «азиатско-американские врата», соеди-

нит США с Малайзией, Сингапуром, Таиландом, Брунеем, Гон-Конгом и Филиппинами. Общая длина оптического кабеля составит 12 428 миль. Главной движущей силой проекта, призванного обеспечить «альтернативный и более безопасный канал связи региона с США», является малайзийский коммуникационный провайдер Telekom Malaysia.

Контракт на реализацию транстихоокеанского канала связи был заключён с компаниями Alcatel-Lucent и NEC. Монтаж кабеля уже начался. Первые данные по новому кабелю начнут транслироваться в декабре 2008 г.

dailytech.com

IBM: микроскоп для наноструктур

Компания IBM заявила, что исследователи из её лаборатории Almaden Research Center продемонстрировали технику ядер-

но-магнитной томографии для визуализации нанобъектов. Метод, получивший название магниторезонансной силовой микроскопии (Magnetic Resonance Force Microscopy, MRFM), сочетает в себе принципы, лежащие в основе магниторезонансной и атомной силовой микроскопии.

Демонстрации плоского изображения объектов с размерами около 90 нм являются важной вехой на пути к конечной цели – получению трёхмерного отображения объектов на атомном уровне. Ожидается, что достижение такого уровня развития технологий позволит учёным лучше понять принципы функционирования белковых структур, что будет способствовать созданию более эффективных лекарств. Полученные с помощью MRFM сведения о строении веществ на атомарном уровне смогут позитивно отразиться на целом ряде отраслей, включая и полупроводниковую индустрию.

ibm.com

Магнитодиэлектрики MICROMETALS



Снижение габаритов и потерь энергии



Высокоэффективные магнитодиэлектрические сердечники Micrometals для силовой электроники и ВЧ-техники

Применение сердечников MICROMETALS позволяет:

- Снизить стоимость индуктивных компонентов в 3...5 раз
- Снизить потери на 30...50% по сравнению с ферритами
- Оптимально распределить потери между сердечником и обмоткой
- Повысить надёжность аппаратуры
- Оптимизировать конструкцию и уменьшить габариты индуктивных компонентов

ОСНОВНЫЕ ХАРАКТЕРИСТИКИ И ДОСТОИНСТВА

- Большая номенклатура типоразмеров – от 3,4 мм до 165 мм
- Токоизолирующее полимерное покрытие до 3 слоёв
- Торoidalные, Ш- и U-образные, трубчатые, низкопрофильные сердечники, стержни, шайбы, бусины и др.
- Силовые магнитопроводы до 5 МГц
- ВЧ-сердечники для частот от 0,01 до 500 МГц
- Рабочая температура до 2000°C
- Высокая стабильность параметров

СРЕДСТВА РАЗРАБОТКИ

состоят из 4 модулей для расчёта индуктивностей:

- Фильтров постоянного тока, включая дифференциальные фильтры
- Индукторов с фиксированной зависимостью изменения индуктивности от величины тока намагничивания
- Катушек с управляемой индуктивностью 10 : 1 или 20 : 1 с использованием композиционных сердечников из феррита и магнитодиэлектрика
- Катушек индуктивности для АККМ, повышающих и понижающих DC/DC - преобразователей



CERTIFICATE OF REGISTRATION PRINCIPAL REGISTER

The Mark shown in this certificate has been registered in the United States Patent and Trademark Office to the named registrant.

The records of the United States Patent and Trademark Office show that an application for registration of the Mark shown in this Certificate was filed in the Office; that the application was examined and determined to be in compliance with the requirements of the law and with the regulations prescribed by the Director of the United States Patent and Trademark Office; and that the Applicant is entitled to registration of the Mark under the Trademark Act of 1946, as Amended.

A copy of the Mark and pertinent data from the application are part of this certificate.

This registration shall remain in force for TEN (10) years, unless terminated earlier as provided by law, and subject to compliance with the provisions of Section 8 of the Trademark Act of 1946, as Amended.



Director of the United States Patent and Trademark Office

Micrometals наносит на свои изделия запатентованную цветовую маркировку в качестве защиты от подделок. Оригинальная продукция Micrometals в компании ПРОСОФТ

PROSOFT[®]

ПРОСОФТ – АКТИВНЫЙ КОМПОНЕНТ ВАШЕГО БИЗНЕСА

Телефон: (495) 232-2522 • E-mail: info@prochip.ru • Web: www.prochip.ru

Высокостабильный кварцевый генератор 10 МГц с цифровой компенсацией температурного ухода частоты

Дмитрий Тумайкин (Ярославская обл.)

В статье приведено описание кварцевого генератора, аппаратная часть которого реализована на ПЛИС и микроконтроллере. Представлена блок-схема генератора, размещённая в ПЛИС. Даны рекомендации по выбору кварцевых резонаторов и калибровке генератора.

Одним из главных факторов, влияющих на стабильность частоты генераторов, в том числе кварцевых, являются колебания температуры окружающей среды. Температура воздействует не только на кварцевый резонатор, но и на все элементы схемы, которые также являются источниками нестабильности генерируемой частоты.

Классическим способом борьбы с дестабилизирующими факторами является помещение схемы генератора в термостат, поддерживающий температуру всех компонентов порядка 50...60°C с достаточно высокой точностью. Для вывода генератора на номинальную частоту требуется время от 30 мин до 1 ч, а иногда и более. За счёт теплоизоляции габариты такого генератора весьма значительные. Качество стабилизации частоты хорошее. Такие генераторы называют термостатированными.

Существуют генераторы термокомпенсированные, в схему которых введены элементы, компенсирующие уход частоты генератора при измене-

ниях температуры. Одним из таких генераторов является ГК-321-ГК-10М; паспортная точность поддержания частоты в диапазоне температур -10...+60°C составляет $(0,6...0,8) \times 10^{-6}$. Испытания образца этого генератора, проведённые автором, показали изменение частоты в указанном температурном диапазоне 20 Гц, т.е. уход частоты от номинального значения составил $\pm 1 \times 10^{-6}$.

В статье предлагается новый способ стабилизации частоты в термокомпенсированных кварцевых генераторах. Аппаратная часть генератора, разработанного на его основе, реализована на ПЛИС и микроконтроллере (МК).

Характеристики откалиброванного генератора приведены в таблице.

Генератор состоит из двух частей: передающей и приёмной.

ПЕРЕДАЮЩАЯ ЧАСТЬ

Передающая часть схемы генератора отслеживает окружающую температуру. В качестве датчика температуры использован температурно-зависимый кварцевый резонатор типа РКТ-206, применяемый в электронных термометрах. При комнатной температуре его резонансная частота составляет примерно 32 кГц. Кварцевый резонатор реагирует на изменение температуры с чувствительностью 2 Гц/град., причём с ростом температуры его резонансная частота понижается. Корпус резонатора – цилиндрический, диаметром 1,5 мм и высотой 6 мм. Резонатор определяет частоту вспомогательного генерато-

ра 32 кГц, которая является функцией температуры.

На рисунке 1 приведены временные диаграммы передающей части устройства. Из-за тепловой инерционности компонентов схемы генератора считывание информации о температуре производится один раз в 5 с, что позволяет отслеживать изменение температуры с дискретностью 0,1°. За это время вспомогательный генератор отсчитывает $2^{15} \times 5 = 163\,840$ импульсов. Заданный диапазон изменения температуры $t_{\text{в}} - t_{\text{н}} = 70 - (-15) = 85^\circ$. При коэффициенте преобразования 1,65 Гц/град. изменение частоты генератора составит $1,65 \times 85 = 140$ Гц за одну секунду, а за 5 с, соответственно, 700 Гц.

Очевидно, что из общего числа 163 840 импульсов информацию о температуре содержат последние 700. Поэтому в схеме предусмотрено накопление импульсов, следующих после 5 с счёта. Пятисекундный интервал времени формируется на основе кварцевого генератора 10 МГц. Некоторая нестабильность этого генератора на выходе *Osc_10MHz* при формировании 5-секундного интервала не сказывается на конечном результате: компенсация отклонения частоты от номинального значения заложена в принцип работы.

Таким образом, изменение температуры преобразуется в унарный код *Temperature_Unar_Code*, импульсы которого один раз в 5 с поступают на МК для обработки.

ПРИЁМНАЯ ЧАСТЬ

В ответ на полученную информацию о температуре микроконтроллер генерирует специальный код коррекции частоты *Serial_Correct_Code*, учитывающий нелиней-

Технические параметры генератора

Номинальная частота	10 МГц
Погрешность	$\pm 0,2 \times 10^{-6}$
Диапазон рабочей температуры	-15...+70°
Диапазон коррекции частоты (цифровая подстройка)	6 Гц
Диапазон коррекции частоты (аналоговая подстройка)	20 Гц
Напряжение питания	+5 В
Потребляемый ток	100 мА
Время готовности к работе	1 мин

ность температурной характеристики обоих кварцевых резонаторов. Эту функцию осуществляет приёмная часть устройства.

Следует отметить, что в данной схеме коррекция частоты реализуется не за счёт сжатия или растяжения периода следования импульсов основного генератора, а путём «выбивания» с определённой периодичностью избыточных импульсов, равномерно распределённых во времени.

На рисунке 2 приведены временные диаграммы приёмной части устройства. Для наглядности слева «выбит» каждый 10-й, а справа – каждый 4-й импульс. На практике «выбивание» импульса может происходить не чаще одного раза за 50 000 импульсов.

Для минимальной температуры диапазона желательно, чтобы частота генератора *Osc_10MHz* была на несколько десятков герц больше номинальной (10 МГц). Таким образом, наибольшее число импульсов, которое необходимо равномерно «выбить» из исходной частоты, составит приблизительно 200 за секунду.

ГЕНЕРАТОРЫ

Оба генератора – основной и вспомогательный – реализованы с использованием активных элементов ПЛИС. Для надёжного запуска и поддержания стабильной работы на краях температурного диапазона использован метод перевода входного логического элемента генераторов в линейный режим работы. Из ПЛИС выведены линии для подключения внешних элементов, необходимых для достижения баланса фаз и сдвига частоты.

Предусмотрена цифровая подстройка выходной частоты основного генератора 10 МГц в пределах 8 Гц при помощи перемычек. Возможен и альтернативный способ коррекции частоты путём введения подстроечного резистора.

На входе вспомогательного генератора 32 кГц установлен полевой транзистор. Его наличие призвано обеспечить высокое входное сопротивление схемы для кварцевого температурного датчика, у которого значение динамического сопротивления R_k при комнатной температуре составляет порядка 70 кОм, а с повышением температуры до конца

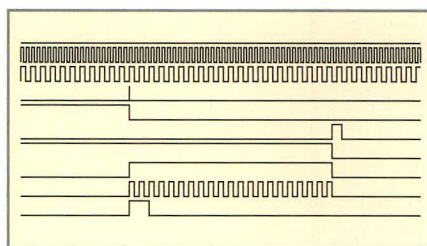


Рис. 1. Демонстрационная диаграмма работы передающей части устройства (эпюры напряжений, сверху вниз):

Osc_10MHz – частота генератора 10 МГц до стабилизации; *Osc_32_Thermo* – частота генератора 32 кГц; *End_5s* – конец 5-секундного интервала; *5s* – импульс 5 с; *Cout* – импульс переполнения счётчика по модулю 163840; *Count_32* – импульс с длительностью, равной времени полного заполнения счётчика по модулю 163840; *Overfilling_Time* – импульс выдачи температурного кода; *Temperature_Unar_Code* – унарный код температуры; *Begin_Essue* – начало выдачи температурного кода

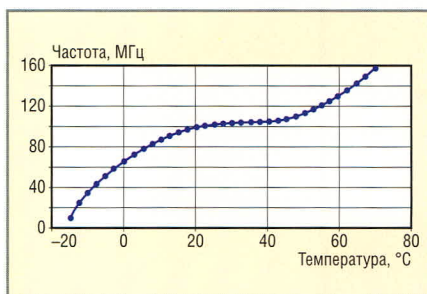


Рис. 3. Температурно-частотная характеристика кварцевого генератора 10 МГц

диапазона R_k может значительно увеличиваться. Корпуса обоих кварцевых резонаторов спаяны между собой для хорошего теплового контакта.

С ростом температуры частота основного генератора увеличивается. Испытания устройства в термостате показали, что в рабочем температурном диапазоне она изменяется на 160 Гц (см. рис. 3). Вид графика температурно-частотной характеристики (ТЧХ) зависит от типа кварцевого резонатора. На рисунке 4 приведена зависимость кода коррекции от температуры.

ВЫБОР РЕЗОНАНСНЫХ ЧАСТОТ КВАРЦЕВЫХ РЕЗОНАТОРОВ 10 МГц и 32 кГц

Принцип работы устройства определяет требования к резонансным частотам кристаллов:

1. Резонансная частота кварцевого резонатора 10 МГц при минимальной температуре рабочего

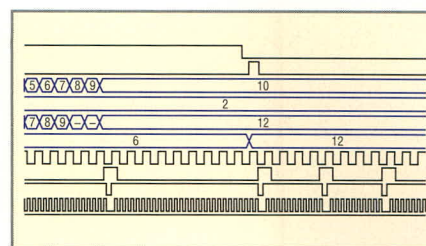


Рис. 2. Демонстрационная диаграмма работы приёмной части устройства (эпюры напряжений, сверху вниз):

Start_Enable – стартовый импульс цикла; *Clearn* – импульс сброса счётчиков и перезаписи данных в буфере-защёлке; *Correct_Bin_Code* – бинарный код коррекции; *Shift_Bin_Code* – бинарный код сдвига частоты; *Summary_Bin_Code* – суммарный бинарный код коррекции; *Final_Bin_Code* – код коррекции после буфера; *Osc_32_Thermo* – частота генератора 32 кГц; *Correct_Pulses* – импульсы коррекции частоты *Osc_10MHz*; *Correct_Avoid_Pulses* – импульсы коррекции рабочие, равные 0,5 мкс; *Out_10MHz* – вид частоты на выходе устройства

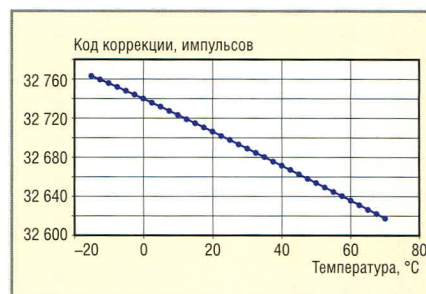


Рис. 4. Зависимость кода коррекции от температуры

диапазона не должна быть ниже 10⁷ Гц;

2. Резонансная частота температурно-зависимого кварцевого резонатора 32 кГц при минимальной температуре не должна быть выше 32 768 Гц.

Нарушение 1-го условия в сторону значительного увеличения частоты приведёт к неоправданному увеличению числа битов для обработки данных, а уменьшение – к неработоспособности устройства. Нарушение 2-го условия приведёт к аналогичным последствиям. Желательно, чтобы оба резонатора имели невысокое значение динамического сопротивления R_k и были стабильны во времени.

Если нет возможности применить кварцевый резонатор 32 кГц нужной частоты, можно изменить условия, уменьшив значение модуля пересчёта в программе ПЛИС. Частоту генератора 10 МГц до точки коррекции можно повысить подбором ёмкости в схеме.

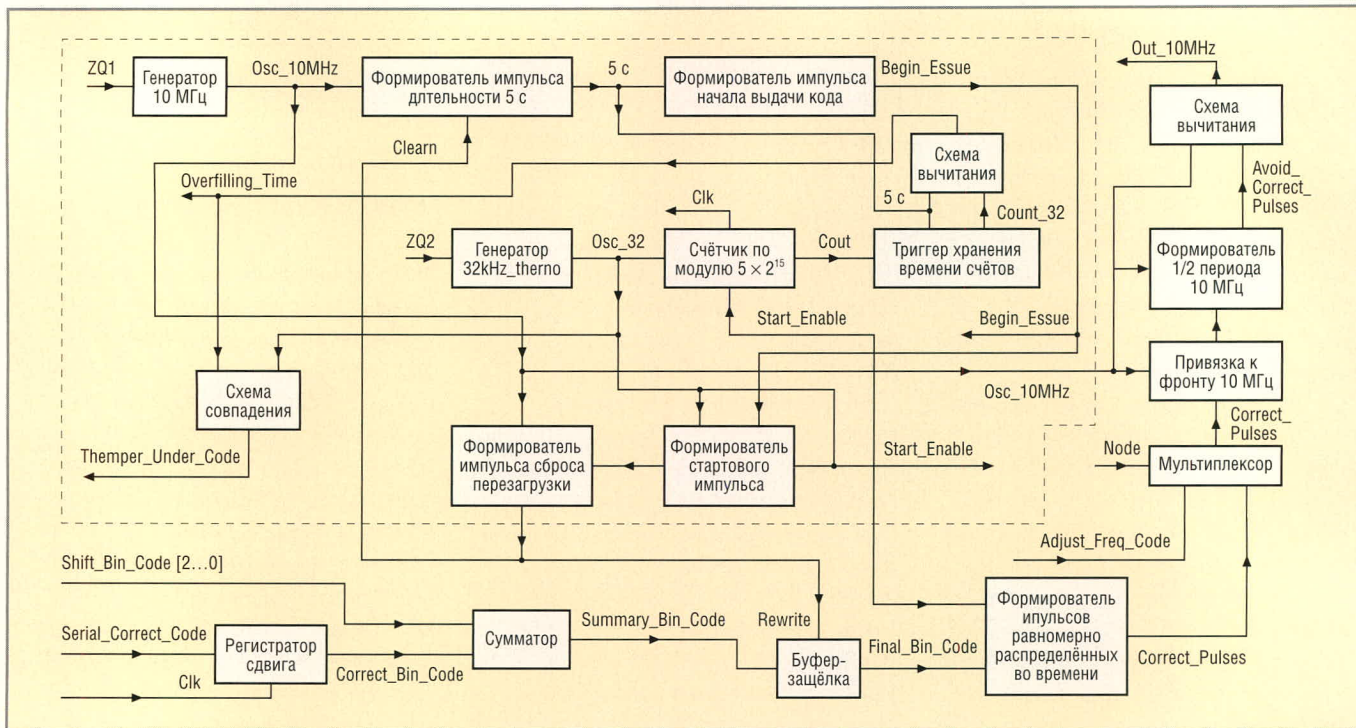


Рис. 5. Блок-схема генератора, размещённая в ПЛИС

ОПИСАНИЕ ФУНКЦИОНАЛЬНОЙ СХЕМЫ

Полная функциональная схема генератора, реализованная на базе ПЛИС, приведена на рисунке 5. Схема разделена на две части – передающую и приёмную.

Передающая часть формирует унарный код текущей температуры *Temperature_Unar_Code* и вспомогательные сигналы и содержит два генератора: основной на 10 МГц и вспомогательный на 32 кГц, реаги-

рующий на изменение температуры.

Восьмибитный последовательный код коррекции *Serial_Correct_Code*, поступающий из МК, синхронизируется внешним сигналом и преобразуется в параллельный двоичный код. Далее он поступает на один из входов сумматора. На другой вход сумматора подаётся код смещения. Выходной код сумматора проходит через буфер-защёлку, который в определённый

момент времени перезаписывает содержимое сумматора и хранит его до момента обновления.

Окончательный код коррекции *Final_Bin_Code* подаётся на формирователь импульсов, равномерно распределённых во времени. На выходе формирователя корректирующие импульсы *Correct_Pulses* следуют с периодичностью, согласованной с текущей окружающей температурой. На рисунке 6 приведена схема взаимодействия сигналов ПЛИС и МК.

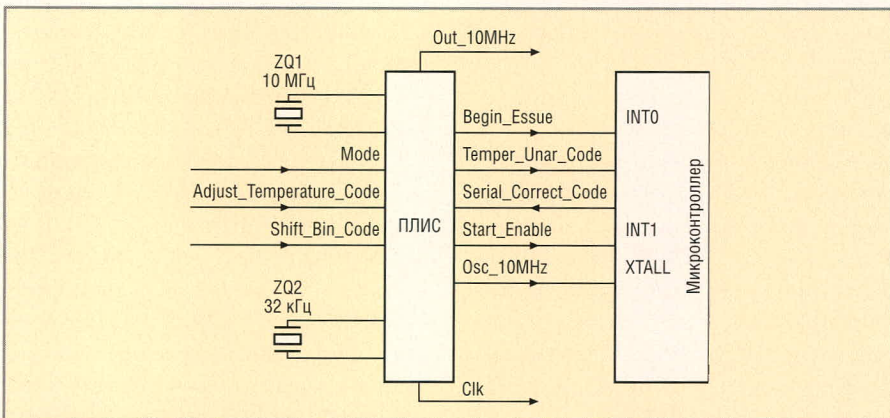


Рис. 6. Взаимодействие сигналов ПЛИС и МК

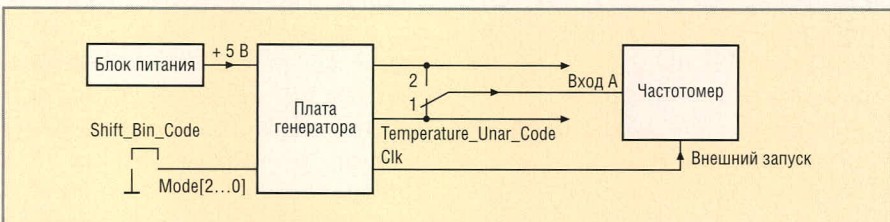


Рис. 7. Блок-схема, используемая для калибровки генератора

Для организации калибровки устройства предусмотрен мультиплексор. Режим «Работа/Калибровка» выбирается битом управления мультиплексором *Mode*. На первый вход мультиплексора поступают рабочие корректирующие импульсы *Correct_Pulses*, а второй вход мультиплексора используется для подключения импульсного генератора НЧ, задействованного в процессе калибровки. Далее импульсы коррекции привязываются к фронтам импульсов 10 МГц, и из них формируются короткие импульсы, запускающие формирователь инвертированного полупериода частоты 10 МГц.

Схема вычитания «выбивает» избыточные импульсы из исходной частоты *Osc_10MHz*, создавая тем самым стабилизированную частоту 10 МГц, обозначенную на схеме *Out_10MHz*.

КАЛИБРОВКА ГЕНЕРАТОРА

Для калибровки генератора необходимы термостат, частотомер и блок питания +5 В. Частотомер работает в режиме внешнего запуска, сигнал которого поступает от схемы генератора. На плате следует установить переключку Mode в положение калибровки (см. рис. 7). К процессу калибровки следует относиться внимательно, так как от него зависит погрешность генератора.

Плата помещается в камеру термостата, экранированные провода выводятся через специальное отверстие в камере. При калибровке задаётся текущая температура в камере термостата и выжидается время до её установления. В положении 1 переключателя S1 (см. рис. 7) по частотомеру определяется число импульсов N , которое отражает текущую температуру и является её унарным кодом *Temperature_Unar_Code*. Затем переключатель переводится в положение 2, и считывается текущее значение частоты основного генератора 10 МГц. Это значение всегда будет выше номинального на ΔF , которое является основой для

формирования кода коррекции частоты *Serial_Correct_Code* и показывает, сколько должно быть «выбито» избыточных импульсов за 1 с.

Итак, для калибровки генератора достаточно двух значений: код текущей температуры *Temperature_Unar_Code* и код коррекции частоты *Serial_Correct_Code* при этой же температуре. По данным составленной таблицы строятся графики. После завершения калибровки переключку Mode следует вернуть в рабочее положение.

Полученные данные являются основой для задания коэффициентов полинома в программе для МК. Задача МК – выдать в ответ на унарный код, полученный от ПЛИС, код коррекции температуры, учитывающий нелинейность преобразования температура-частота:

$$CUC = \sum_{i=0}^4 k_i TUC^i,$$

где CC – *Correct_Code* – код коррекции в бинарном формате; TUC – *Temperature_Unar_Code* – унарный код температуры; k_i – коэффициенты

полинома. Для этого на основании полученных в ходе калибровки данных вычисляются коэффициенты полинома 4-й степени, отражающего вид графика температурной зависимости, которые затем используются в программе.

ЗАКЛЮЧЕНИЕ

В целом испытания генератора, реализованного по данной схеме, показывают высокую стабильность в заявленном температурном диапазоне. Для применения данных генераторов в аналоговой схемотехнике в его конструкцию потребуются внести некоторые изменения – вместо выбивания импульсов воздействовать посредством ЦАП или ШИМ с выделением среднего значения (что дешевле) на схему генератора.

ЛИТЕРАТУРА

1. Альтишуллер Г.Б., Елфимов Н.Н., Шакулин В.Г. Кварцевые генераторы: Справ. пособие. М.: Радио и связь, 1984.
2. Стешенко В.В. Плис фирмы Altera. Проектирование устройств. Додэка-XXI, 2002. ©



Международный промышленный форум
РАДИОЭЛЕКТРОНИКА . ПРИБОРОСТРОЕНИЕ . АВТОМАТИЗАЦИЯ

Организаторы
ufi **FareXPO** IFE®
PROFESSIONAL EXHIBITION ORGANIZER

Техно&Ком

КРУПНЕЙШАЯ на Северо-Западе

VII международная специализированная выставка

РАДИОЭЛЕКТРОНИКА И ПРИБОРОСТРОЕНИЕ



RADEL'07
RADIO ELECTRONICS & INSTRUMENT-MAKING INDUSTRY EXHIBITION

- ЭЛЕКТРОННЫЕ КОМПОНЕНТЫ
- КОМПЛЕКТУЮЩИЕ
- ПЕЧАТНЫЕ ПЛАТЫ
- МАТЕРИАЛЫ
- КОНСТРУКТИВЫ
- ТЕХНОЛОГИИ
- ОБОРУДОВАНИЕ
- ПРИБОРЫ

20-23 ноября 2007

Санкт-Петербург
Петербургский СКК

Вас приглашают

Тел./факс: (812) 718-35-37; e-mail: radel@orticon.com; www.fareexpo.com



СПРИНГ ЭЛЕКТРОНИКС



www.platan.ru
ПЛАТАН



OSTEC



ECT



ДИПОЛЬ



ЭТЕХ



УНИВЕРСАЛПРИБОР



РЕЗОНИТ

Регулирующий клапан с электроприводом

Владимир Вычужанин (г. Одесса, Украина)

В статье описана схема управления шаговым двигателем с использованием ПЛИС и приведены результаты моделирования работы электропривода.

Для управления процессами, протекающими в теплообменниках аппаратах холодильных установок и систем кондиционирования воздуха, в качестве исполнительных устройств применяются клапаны с электроприводами. Правильный выбор клапана и обеспечение требуемой регулировочной характеристики обеспечивают необходимое качество регулирования.

Подвижные части регулирующих клапанов отличаются конструктивной, внутренней и регулировочной характеристиками. Конструктивная характеристика клапана устанавливает зависимость между перемещением подвижной части клапана и изменением площади открытого сечения для прохода среды. Внутренняя характеристика определяет зависимость между ходом

поршня и расходом среды через клапан при постоянном перепаде давлений на клапане. При мгновенном изменении положения поршня расход среды через клапан также изменяется мгновенно. Регулирующие клапаны можно представить как усилительные звенья системы автоматического регулирования. Коэффициент передачи таких звеньев описывается нелинейной функцией и при различных положениях поршня может существенно изменяться.

Электропривод регулируемого клапана, состоящий из электродвигателя и схемы управления (СУ), часто не обеспечивает необходимого качества управления процессами в теплообменниках аппаратах. Следует отметить, что современный электропривод для подобного клапана –

это пропорциональное устройство дискретного типа, управляющее исполнительным органом (поршнем клапана). Имеющиеся недостатки конструкции регулирующего клапана с электроприводом можно в значительной мере устранить путём совершенствования схемы управления, что позволяет повысить эффективность технических решений.

На первом этапе разработки необходимо выбрать тип пропорционального исполнительного устройства. В качестве такого устройства может быть использован шаговый двигатель (ШД). Благодаря возможности управления перемещением ротора на любой угол, ШД предпочтителен в качестве исполнительного устройства дискретного типа.

В качестве исполнительного элемента был выбран клапан типа ETS фирмы Danfoss. Конструкция клапана обеспечивает реверсивное движение потока и полное закрытие клапана, вне зависимости от направления потока. Использование клапана ETS позволяет изменять расход носителя и дросселировать поток при переменном положении поршня клапана. Кроме того, клапан ETS сочетает в себе лучшие свойства клапанов, применяемых в сетях, а именно: малый ход; почти линейную характеристику регулирования расхода носителя при высоких значениях относительного хода поршня клапана и высокую точность регулирования при малых значениях расхода.

Однако, электропривод, используемый фирмой Danfoss совместно с регулирующим клапаном типа ETS, обладает определёнными недостатками: большими габаритами, биполярной конфигурацией обмоток шагового двигателя и применением полношагового двухфазного управления двигателем. Разработанная схема управления (см. рис. 1) теплообменным аппаратом с использованием регулирующего клапана ETS призвана устранить недостатки электропривода фирмы Danfoss.

Положение поршня клапана изменяется посредством электропривода, управляемого импульсным сигналом. Сигнал формируется в устройстве управления 9 регулятором 10 (см. рис. 1) в соответствии с используемым законом управления (П, ПИ или ПИД). В качестве ШД для электропривода регулирующего клапана был

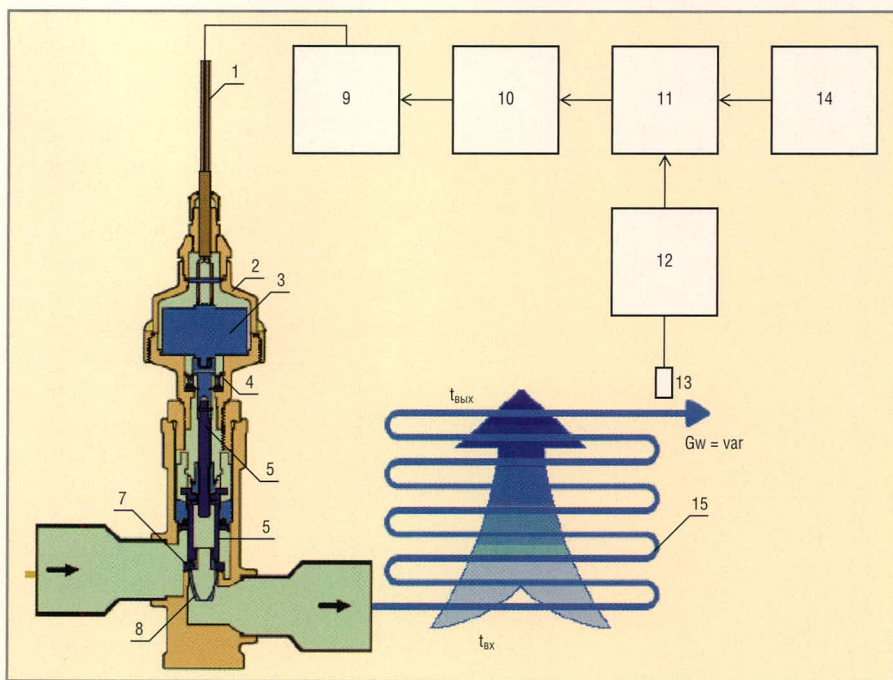


Рис. 1. Схема управления регулирующим клапаном

1 – кабель; 2 – корпус ШД; 3 – ШД; 4 – втулка; 5 – вал; 6 – поршень клапана; 7 – седло клапана; 8 – порт клапана; 9 – устройство управления ШД; 10 – регулятор; 11 – сравнивающее устройство; 12 – измерительный преобразователь; 13 – датчик расхода носителя; 14 – задающее устройство; 15 – воздухоохладитель

выбран гибридный двигатель с шагом 7,5 градусов. Это – двигатель с постоянными магнитами, но с большим числом полюсов [1, 2]. В отличие от используемой Danfoss биполярной конфигурации, предпочтение было отдано униполярной. Это объясняется тем, что в поставленной задаче от двигателя не требуется повышенный момент. Такой выбор позволяет уменьшить габариты регулирующего клапана с электроприводом, работать с меньшими потерями, а также упростить схему управления.

Частота управляющих импульсов, подаваемых на фазовые обмотки ШД, выбирается исходя из предельных механических и динамических характеристик двигателя. Первые устанавливают зависимость допустимого момента сопротивления от частоты управляющих импульсов в установленном режиме. Для различных типов ШД эта частота может достигать нескольких килогерц. Анализ динамических характеристик позволяет уточнить выбор частоты управляющих импульсов. Регулирующий клапан с шаговым приводом должен обрабатывать как единичные импульсы управления, так и последовательность импульсов с частотой, определяемой предельными динамическими характеристиками. При разработке СУ приводом с точной обработкой положения и малым моментом инерции была выбрана частота перемещения ротора 100...400 Гц. Скорость вращения ШД определялась частотой управляющих импульсов в указанном диапазоне; угол поворота вала двигателя – числом шагов (импульсов).

Схему управления фазами ШД можно реализовать на основе универсального или специализированного микроконтроллера, а также на базе ПЛИС. В нашем случае ПЛИС обладает рядом преимуществ: возможностью согласования входов и выходов с практически любыми логическими уровнями; способностью заменить несколько микросхем, включая микроконтроллер, регистры портов, интерфейс и т.п. Кроме того, использование ПЛИС ускоряет программирование, макетирование и реконфигурирование цифровой схемы. Поэтому учёт архитектурных особенностей ПЛИС и в ряде случаев их преимуществ перед микроконтроллерами позволяет реализовать на

основе ПЛИС вполне конкурентоспособные изделия.

Учитывая изложенное, схема управления ШД регулирующего клапана была реализована на ПЛИС фирмы Altera. При проектировании устройства использовалась САПР Quartus II ver.4.2. Алгоритм работы устройства учитывает последовательность включения и выключения силовых ключей в полшаговом режиме управления двигателем. Кроме повышения разрешающей способности, такой способ управления, по сравнению с полношаговым, используемым фирмой Danfoss, позволяет ослабить резонансные эффекты.

Ниже приведён фрагмент VHDL-описания работы устройства, используемого в схеме управления на ПЛИС:

```
Subdesign Main
(
    Clk, Revers, Ena: input;
    A, B, C ,D : output;
)
Variable
    FSM: machine with
states(A0,A1,A2,A3,A4,A5,A6,A7,A8);
Begin
    FSM.Clk=Clk;

    case FSM is
        when A0 =>
            if Ena == 1 then
                FSM = A1;
                A=B"0";
                B=B"0";
                C=B"0";
                D=B"0";
            end if;
        when A1 =>
            if Revers == 0 then
                A=B"1";
                B=B"0";
                C=B"0";
                D=B"0";
            else
                A=B"1";
                B=B"0";
                C=B"0";
                D=B"0";
            end if;
            if Ena == 0 then
                FSM = A0;
            else FSM = A2;
            end if;
        when A2 =>
            if Revers == 0 then
                A=B"1";
                B=B"0";
```

```
                C=B"1";
                D=B"0";
            else
                A=B"1";
                B=B"0";
                C=B"0";
                D=B"1";
            end if;
            if Ena == 0 then
                FSM = A0;
            else FSM = A3;
            end if;
        when A3 =>
            if Revers == 0 then
                A=B"0";
                B=B"0";
                C=B"1";
                D=B"0";
            else
                A=B"0";
                B=B"0";
                C=B"0";
                D=B"1";
            end if;
            if Ena == 0 then
                FSM = A0;
            else FSM = A4;
            end if;
        when A4 =>
            if Revers == 0 then
                A=B"0";
                B=B"1";
                C=B"1";
                D=B"0";
            else
                A=B"0";
                B=B"1";
                C=B"0";
                D=B"1";
            end if;
            if Ena == 0 then
                FSM = A0;
            else FSM = A5;
            end if;
```

Функциональная схема, реализующая полшаговый режим управления ШД регулирующего клапана, приведена на рисунке 2.

Схема управления ШД работает следующим образом. Заданное число шагов (импульсов) в параллельном коде подаётся на входы счётчика числа шагов *Ipm counter0* по сигналу *StepNumber[10...0]*. Максимально необходимое число шагов для конкретного устройства (2042) формируется в блоке генерации константы *Ipm constant0*. В компараторе *Ipm compare0* сравниваются количества импульсов (шагов), поступающие с выходов *Ipm constant0* и *Ipm counter0*. При числе

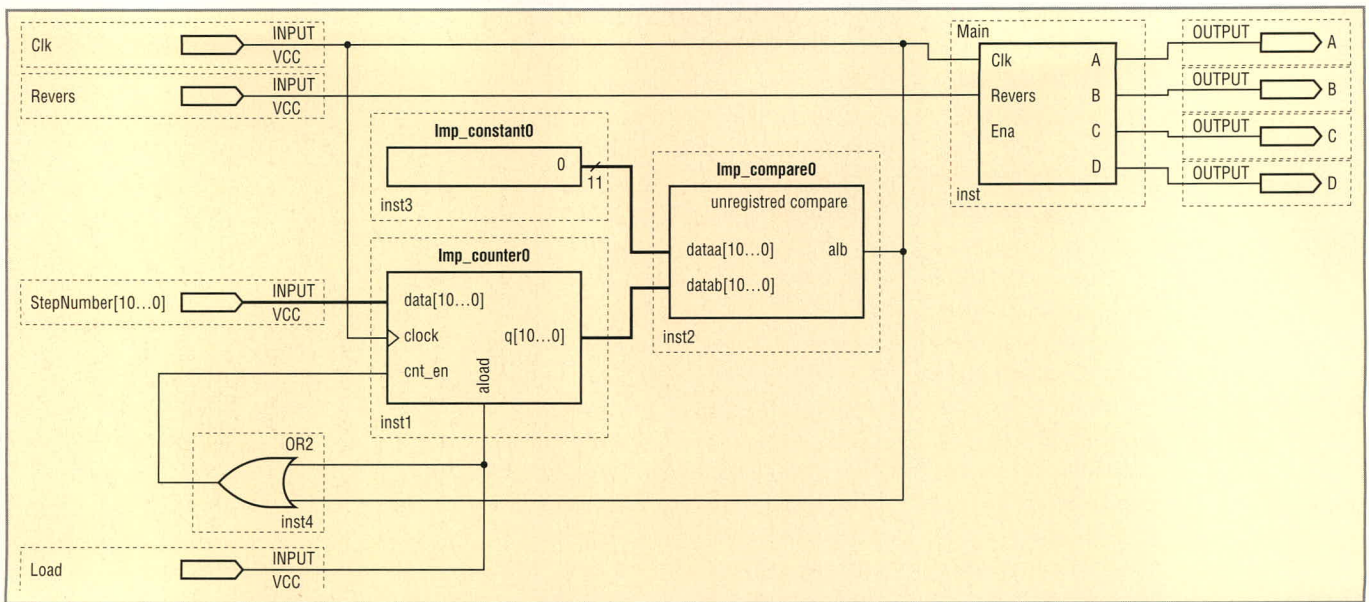


Рис. 2. Функциональная схема электропривода на основе ПЛИС, реализующая полушаговый режим ШД

Imp constant0 – блок генерации константы; *Imp counter0* – счётчик числа шагов; *Imp compare0* – компаратор; *Main* – блок формирования управляющего сигнала

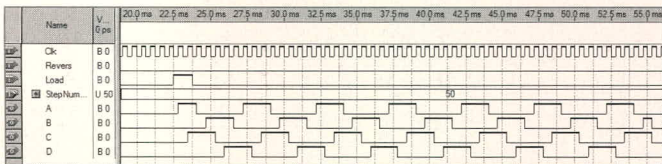


Рис. 3. Результаты моделирования при прямом направлении вращения ротора ШД

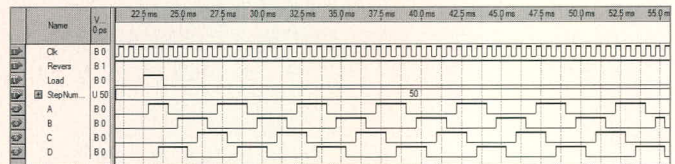


Рис. 4. Результаты моделирования при обратном направлении вращения ротора ШД

шагов менее 2042, с выхода *Imp compare0* на вход блока *Main* проходит необходимое число импульсов, соответствующее положению поршня клапана, т.е. степени открытия регулирующего клапана. Блок *Main*, формирующий управляющий сигнал, обеспечивает временной сдвиг импульсов на выходах А, В, С и D схемы. В результате реализуется полушаговый режим управления ШД. Пуск схемы осуществляется по сигналу, подаваемому на вход *Load* системой управления верхнего уровня. Прямое направление вращения двигателя задаётся «лог. 0» на входе *Revers*; для изменения направления вращения на этот вход необходимо подать «лог. 1».

После отладки компонентов схемы на основе созданных при компиляции выходных файлов, осуществлено моделирование работы СУ с помощью подсистемы (Simulator) пакета Quartus II ver.4.2. В результате компиляции создан файл для программирования и конфигурирования ПЛИС Altera.

Для реализации импульсного управления фазами ШД в полушаговом режиме может быть использована принципиальная электрическая схема устройства EPM7064LC44-5 на ПЛИС типа MAX7000S, аналогичная приведённой [3].

Технические характеристики регулирующего клапана с разработанной

схемой управления на основе униполярного ШД приведены в таблице.

Работоспособность разработанной схемы управления шаговым двигателем на базе ПЛИС проверялась путём моделирования работы устройства при частоте 200 Гц управляющих импульсов, подаваемых на фазовые обмотки. Полученные результаты (см. рис. 3 и 4) подтверждают соответствие схемы предъявляемым требованиям.

Разработанная СУ униполярным ШД на основе ПЛИС позволяет реализовать полушаговый способ управления фазовыми обмотками двигателя. В результате обеспечивается возможность точного регулирования расхода носителя в широком диапазоне. Схема управления характеризуется доступностью элементной базы, относительной простотой в реализации и настройке.

ЛИТЕРАТУРА

1. Башарин А.В., Новиков В.А., Соколовский Г.Г. Управление электроприводами. Энергоиздат, 1982.
2. Шелтинг Г., Байсе А. Электрические микромашины. Мир, 1997.
3. Вычужанин В.В. Устройства на ПЛИС для управления шаговым двигателем. Электронные компоненты. 2005. № 9, 10.

Технические характеристики регулирующего клапана

Окружающая температура	-40...60°C
Полный ход поршня	13 мм
Шаговый двигатель	Униполярный
Шаг	Полушаг
Шаговый угол	7,5°
Номинальное напряжение	12 В, -4% /+15%
Фазовый ток	400 мА, -4% /+15%
Частота перемещения ротора двигателя	100...400 Гц
Скорость перемещения	300 шар/с
Максимальное число шагов (угол)	2625 (+160°/-0°)
Полное время хода поршня	8,5 с
Начальное положение клапана	Полностью закрытое

Новости мира News of the World Новости мира

Автопроизводители США не спешат с поддержкой HD Radio

Радиовещательные корпорации, столкнувшись с растущей конкуренцией со стороны iPod и спутникового радио, пытаются удержать внимание слушателей с помощью перехода на обеспечивающий большие возможности стандарт HD Radio, однако пока что компании-автопроизводители из США не спешат с поддержкой этого новшества. В то время как Hyundai, BMW, контролируемый Ford Motor британский бренд Jaguar и некоторые другие компании занялись активным продвижением нового стандарта, их американские коллеги пока что заняли скорее выжидательную позицию.

Официальные лица двух участников «большой тройки» американских автопроизводителей, компаний General Motors и Chrysler, сообщили о том, что реализация программы поддержки HD Radio обошлась бы каждому из этих предприятий приблизительно в 150...200 млн. долл. ежегодно, при этом рентабельность такого шага пока что неочевидна. На изучение перспектив нового стандарта и принятие взвешенного решения опрошенные собираются потратить ещё около полугода. Официальных комментариев от Ford по этому вопросу пока нет.

Переход на HD Radio сулит улучшение качества звука по сравнению с традиционным аналоговым радиовещанием и обеспечивает возможность многоканальной трансляции. В настоящее время приблизительно десятая часть от более чем 12 тыс. радиостанций США развернули поддержку нового стандарта, включая станции из национального топ-100 популярности. В то же время широкое распространение новшества пока сдерживается тем, что пользователи не торопятся приобретать приёмники с поддержкой HD Radio, да и среди радиостанций ещё немало консерваторов, не желающих осваивать цифровой формат. Похоже, что потребуется какое-то время, чтобы усилиями активных приверженцев прогрессивного нововведения рынок HD Radio «дозрел» до того состояния, когда поддержка этого стандарта станет действительно широко востребованной.

yahoo.reuters.com

Toshiba: анализ пути электрона с точностью до 1 нм

Компания Toshiba сообщила о своём прорыве в области визуализации движе-

ния электронов и примесей в полупроводниках, впервые позволяющей провести анализ пути распространения зарядов с точностью до 1 нм. Это достижение, которое основано на сканирующей микроскопии сопротивления растекания (SSRM), является важным шагом на пути к созданию чипов по техпроцессу 45-нм и ниже. Toshiba представила новую методику на Международном симпозиуме по физическим основам надёжности (IRPS), который проводился с 15 по 19 апреля в Аризоне, США.

Сканирующая микроскопия сопротивления растекания (SSRM) – технология двумерного картографирования сопротивлений на перекрестных поверхностях, позволяющая анализировать распределение электронов и примесей. Высокая точность, необходимая для создания 45-нм чипов, требуется для анализа плотности электронов в канале и возможности контроля за примесями с точностью 1 нм, так как даже незначительные различия в характеристиках могут привести к увеличению уровня токов утечки и риску короткого замыкания.

Технология SSRM использует сканирующий зонд для двумерной визуализации траектории заряда в полупроводниковых устройствах. Полученные изображения показывают вариации в сопротивлении, вызванные примесями, и позволяют произвести анализ путей распространения заряда. В то же время уровень точности и повторное воспроизводство изображений высокого разрешения с помощью традиционной технологии SSRM ограничены пятью нанометрами.

Основными причинами такого ограничения точности являются ухудшение чёткости изображения из-за водяного пара, попадающего на образец из окружающей среды, а также сложность контроля стабильного контакта между образцом и зондом. Для преодоления этих барьеров инженеры Toshiba установили оборудование SSRM в вакуумной камере и стабилизировали положение зонда. Это позволило создать оптимальные условия и поднять предел разрешения до 1 нм – максимально возможного на сегодняшний день для SSRM-техники. Toshiba планирует использовать эту технологию для создания чипов по 45-нм производственному процессу.

Плотность электрического тока можно измерить с помощью сканирования поперечного сечения образца зондом с алмазным покрытием высокой проводимости.

С помощью полученных данных можно рассчитать плотность путей распространения заряженных частиц.

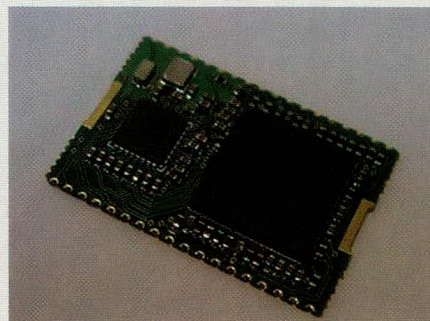
www.3dnews.ru

Новые победы беспроводной шины USB

На мероприятии Intel Developer Forum (IDF), стартовавшем в Пекине 16 апреля, ассоциация разработчиков шины USB (USB Implementers Forum, USB-IF) объявила об очередных успехах в продвижении беспроводного интерфейса USB. Несколько новых продуктов успешно прошли тестирование и сертификацию на соответствие спецификациям Certified Wireless USB. В их числе два хост-контроллера: Intel Wireless Link 1480 RTUMC480AAC (модель SL9N3) и NEC uPD720170, а также два клиентских адаптера: Alereon AL4000 и NEC uPD720180. Во второй половине года ожидается значительное расширение данного списка.

Certified Wireless USB является протоколом верхнего уровня для беспроводной связи, построенным на базе сверхширокополосной платформы Ultra Wideband (UWB). Новая технология гарантирует пропускную способность беспроводного USB-канала до 480 Мбит/с на дистанции 3 м и до 110 Мбит/с – на 10 м.

На весенней конференции IDF 2007 в китайской столице компанией Alereon были продемонстрированы прототипы цифровых фотокамер Samsung и Kodak, оснащённые чипсетом AL4000 PHY. Данный набор микросхем, который находится в стадии массового производства с января этого года, включает в себя RF-трансивер AL4100 и сигнальный процессор AL4200.



По информации USB-IF, в некоторых странах уже завершены формальности по созданию правил использования платформы Ultra Wideband (UWB). С февраля текущего года официально готовыми к беспроводной технологии можно считать страны Европейского Союза, с марта – Южную Корею.

digitimes.com

Практический курс сквозного проектирования цифровых устройств на основе ПЛИС фирмы Xilinx

(часть 5)

Валерий Зотов (Москва)

Пятая часть курса знакомит с основами языка описания аппаратуры VHDL, необходимыми для подготовки описания разрабатываемого устройства. Рассмотрен синтаксис основных операторов, используемых в составе описаний проектируемых устройств и поддерживаемых средствами синтеза САПР серии Xilinx ISE.

КРАТКИЕ СВЕДЕНИЯ ОБ ОСНОВНЫХ ЭЛЕМЕНТАХ ЯЗЫКА ОПИСАНИЯ АППАРАТУРЫ VHDL

Подробное систематизированное описание языка VHDL выходит за рамки журнальной публикации. Поэтому в настоящем и последующих разделах приводятся краткие сведения об основных понятиях, элементах и конструкциях этого языка, позволяющих быстро приступить к практической работе. Более подробную информацию о применении языка VHDL для описания цифровых устройств можно найти в [12–14].

Текст описания цифрового устройства на языке VHDL формируется из ключевых слов, идентификаторов, литералов и комментариев. Под ключевым (зарезервированным) словом понимается последовательность символов, которая имеет строго определённое назначение. Примером ключевых слов являются операторы. Полный список ключевых слов языка VHDL можно найти в документации или литературе [13].

Идентификаторы представляют собой последовательность символов, определяемую пользователем. Идентификаторы используются, в частности, в качестве имён сигналов, переменных, функций, процедур. В составе идентификаторов могут использоваться символы латинского алфавита, цифры и символ подчеркивания. При этом символ подчеркивания не может использоваться в качестве окончания идентификатора или следовать за аналогичным сим-

волом подчеркивания. Начинаться идентификатор может только с алфавитного символа.

В отличие от различных языков программирования, запись ключевых слов и идентификаторов в языке VHDL инвариантна по отношению к регистру символов. Таким образом, идентификаторы и ключевые слова, записанные с помощью строчных и заглавных букв, воспринимаются компилятором VHDL и средствами синтеза одинаково. Литералы, как правило, используются в качестве значений, которые присваиваются переменным или сигналам, а также для инициализации констант соответствующего типа. В языке VHDL различают следующие типы литералов: десятичные, базовые, символьные, строчные и строчные битовые. Десятичные литералы представляют собой числовые значения и подразделяются на целые и вещественные, которые могут быть представлены в обычной или экспоненциальной форме. Например, 1 – целый литерал, 7E3 – целый с экспонентой, 5.0 – вещественный, 1.7E-5 – вещественный с экспонентой.

Базовые литералы являются числовыми значениями с явным указанием системы счисления, которые представлены в следующем формате: <система_счисления>#<числовое_значение>#. Например, 2#1010# – базовый двоичный литерал, 10#25# – базовый десятичный литерал, 16#8F# – базовый шестнадцатеричный литерал. Символьные литералы образуют символы, заключённые в одиночные

апострофы, например, 'C', 'Z'. Строчные литералы представляют собой последовательность алфавитных символов, заключённых в апострофы. Строчные битовые литералы содержат последовательность двоичных, восьмеричных или шестнадцатеричных символов, заключённых в апострофы, перед которыми указано условное обозначение системы представления: В – двоичная, О – восьмеричная, Х – шестнадцатеричная, например, В"0011", О"5743", Х"1AFC". Комментарием считается часть строки или вся строка, которая начинается с двух следующих подряд дефисов (символов --) и заканчивается концом этой строки. Комментарии, так же как пробелы и символы перевода строки, игнорируются компилятором и средствами синтеза VHDL.

В составе языка VHDL применяется три основных класса элементов (объектов данных): сигналы, переменные и константы. Под сигналом понимается идентификатор, который соответствует некоторому физическому сигналу (физической цепи) разрабатываемого устройства. Значение, присвоенное этому идентификатору, соответствует значению физического сигнала. Кроме того, сигналы имеют ряд вспомогательных параметров, которые представлены в форме атрибутов. Атрибут сигнала указывается непосредственно после идентификатора, отделяясь от него одиночным апострофом <идентификатор_сигнала>' <идентификатор_атрибута>.

Атрибуты бывают двух видов: предопределённые и определяемые пользователем. Наиболее часто в описании проектируемого устройства используется предопределённый атрибут *event*, который имеет тип *boolean*. Данный атрибут принимает значение *true* только в том слу-

чае, если произошло изменение этого сигнала. Например, атрибут *ADR_EN'event* будет иметь значение *true* при любом переключении сигнала *ADR_EN*. Информацию о других предопределённых атрибутах можно найти в [13].

Переменная с точки зрения разработчика представляет собой идентификатор, значение которого может изменяться различными операторами. Переменные используются в функциях и процедурах для организации внутренних вычислений в описании проектируемого устройства. Константа – идентификатор, который используется для обозначения некоторого постоянного значения. Константы могут использоваться в качестве значений переменных, параметров функций и процедур. Применение констант вместо числовых значений делает текст описания более прозрачным. Если некоторое числовое значение многократно используется в описании устройства, то, применяя вместо него соответствующую константу, в случае необходимости его изменения достаточно в одном месте поменять значение этой константы.

Для каждого из элементов, перечисленных выше, в описании устройства указывается соответствующий тип, определяющий всю совокупность значений, которые может принимать данный элемент (сигнал, переменная или константа). В языке VHDL имеется только несколько предопределённых типов, среди которых наиболее часто используются типы *bit*, *bit_vector*, *integer*, *real*, *boolean*, *character*, *string*, *time*. В частности, элементы типа *bit* могут принимать значения 0 и 1; *integer* – значения целого типа; *real* – значения вещественного типа; *boolean* – значения *true* и *false*. В отличие от предопределённых типов, остальные типы, применяемые в описании цифровых устройств, относятся к группе определяемых пользователем типов. Из таких типов в описаниях устройств чаще всего используется тип *std_logic*. Тип *std_logic* целесообразно использовать для описания большинства цифровых сигналов. Сигналы, переменные и константы этого типа могут принимать следующие значения: '0' – логический ноль, '1' – логическая единица, 'Z' – состояние высокого импеданса, 'X' – неизвестное

значение, 'L' – логический ноль (слабый источник), 'H' – логическая единица (слабый источник), 'W' – неизвестное значение (слабый источник), 'U' – неинициализированное состояние, '-' – неопределённое состояние.

Кроме простых (скалярных) типов применяются составные типы данных – массивы. Массив представляет собой упорядоченную совокупность элементов одного типа, обращения к которым осуществляются с помощью индекса. Массивы в языке VHDL применяются, в первую очередь, для описания сигналов, имеющих шинную организацию. Для описания таких сигналов чаще всего используется тип массива *std_logic_vector*, элементы которого относятся к рассмотренному выше типу *std_logic*. Среди предопределённых типов, перечисленных выше, *bit_vector* также относится к типу массива.

Для осуществления основных (базовых) операций над элементами описания в языке VHDL предусмотрен набор встроенных операторов. В этот набор входят логические операторы языка *and* (логическое И), *or* (логическое ИЛИ), *not* (логическое НЕ), *nand* (логическое И-НЕ), *nor* (логическое ИЛИ-НЕ), *xor* (исключающее ИЛИ), *xnor* (исключающее ИЛИ-НЕ), операторы отношения = (равно), /= (не равно), > (больше), >= (больше или равно), < (меньше), <= (меньше или равно) и арифметические операторы + (сложения), – (вычитания), * (умножения) и / (деления). Для каждого типа элементов языка VHDL предназначена соответствующая группа встроенных операторов.

СТРУКТУРА ОПИСАНИЯ ПРОЕКТИРУЕМОГО УСТРОЙСТВА НА ЯЗЫКЕ VHDL

При описании проектируемого устройства с помощью языка VHDL оно представляется в форме соответствующего объекта, который может иметь как простую (одноуровневую), так и иерархическую (многоуровневую) структуру. Исчерпывающая информация об этом объекте указывается в форме его интерфейса и архитектуры. Интерфейс объекта описывается в виде совокупности портов, каждый из которых соответствует входу, выходу или двунаправленному входу/выходу проектируемого устройства. Для каждого порта указывается его тип и режим функционирования.

Архитектура объекта представляется в форме структуры разрабатываемого устройства или описания алгоритма его функционирования.

В общем случае структура VHDL-описания проектируемого устройства включает в себя три части, которые расположены в порядке их перечисления. В первой части приводятся ссылки на все используемые библиотеки, пакеты и элементы этих библиотек. Применение библиотек позволяет значительно сократить объём описания на языке VHDL. В библиотеках, в частности, содержится определение типов сигналов, переменных и функций, которые могут использоваться в различных проектах. Вторая часть представляет собой декларацию объекта описания. В этой части объявляется объект, который соответствует разрабатываемому устройству, приводится описание его интерфейса и настраиваемых параметров. В третьей части выполняется определение архитектуры описываемого объекта. Здесь приводится в той или иной форме информация, которая полностью описывает функционирование этого объекта. Далее последовательно рассматривается синтаксис конструкций языка VHDL, которые используются в каждой части описания разрабатываемого устройства.

ИСПОЛЬЗОВАНИЕ БИБЛИОТЕК И ПАКЕТОВ В СОСТАВЕ VHDL-ОПИСАНИЯ ПРОЕКТИРУЕМОГО УСТРОЙСТВА

Создание ссылки на требуемую библиотеку выполняется с помощью ключевого слова *library*. При этом используется следующий формат записи, используемый для указания необходимых библиотек:

```
library <название_библиотеки>;
```

Кроме ссылки на библиотеку, необходимо указать конкретный пакет данной библиотеки и элементы этого пакета, которые будут использоваться в VHDL-описании проектируемого устройства. Под пакетом в языке VHDL понимается файл, содержащий декларации типов данных, переменных, констант, сигналов и компонентов, а также определения функций и процедур, которые могут применяться в различных проектах. Для указания используемого пакета библиоте-

ки и его элементов предназначено ключевое слово *use*. Формат предложения, которое открывает доступ к пакету указанной ранее библиотеки и его элементам, имеет следующий вид:

```
use <название_библиотеки> . <название_пакета> . <идентификатор_элемента_пакета>
```

Чтобы использовать все элементы пакета, указываемого в предложении *use*, необходимо использовать следующий вариант формата этого предложения:

```
use <название_библиотеки> . <название_пакета> . all
```

В качестве примера далее приводится ссылка на наиболее часто используемую библиотеку IEEE и пакет *std_logic_1164*. Определения, содержащиеся в стандартном логическом пакете *std_logic_1164* указанной библиотеки, разработанном Институтом инженеров по электротехнике и электронике, применяются в большинстве создаваемых VHDL-описаний проектируемых устройств.

```
library ieee;
use ieee.std_logic_1164. all;
```

ДЕКЛАРАЦИИ ОБЪЕКТА В СОСТАВЕ VHDL-ОПИСАНИЯ ПРОЕКТИРУЕМОГО УСТРОЙСТВА

В общем случае синтаксис конструкции, предназначенной для декларации описываемого объекта, выглядит следующим образом:

```
entity <имя_объекта> is
generic (
    <идентификатор_параметра1> : <тип_параметра1>;
    <идентификатор_параметра2> : <тип_параметра2>;
    ...
    <идентификатор_параметраM> : <тип_параметраM>;
);
port (
    <название_порта1>: <направление_передачи_сигнала_порта1>
    <тип_порта1>;
    <название_порта2>: <направление_передачи_сигнала_порта2>
    <тип_порта2>;
    ...

```

```
<название_портаN> : <направление_передачи_сигнала_портаN>
<тип_портаN>
);
end <имя_объекта>;
```

Декларация объекта описания начинается с ключевого слова *entity*, вслед за которым указывается имя этого объекта и ключевое слово *is*. После объявления объекта в общем случае могут быть перечислены его настраиваемые параметры, которые используются в определении архитектуры разрабатываемого устройства. Применение настраиваемых параметров в VHDL-описаниях позволяет, в частности, создавать универсальные модули, которые затем при заданных значениях параметров могут использоваться в различных проектах. В форме параметров целесообразно указывать разрядность компонентов и шин, значения длительностей задержек сигналов. Например, в формируемом описании регистра (или счётчика) вместо явного указания его разрядности можно задействовать соответствующий параметр. Таким образом, впоследствии в качестве компонента можно использовать один и тот же объект для описания регистров (или счётчиков) с различной разрядностью в составе новых проектов.

Декларация настраиваемых параметров осуществляется с помощью ключевого слова *generic*, вслед за которым приводится список идентификаторов параметров с указанием их типов. Если настраиваемые параметры не применяются в составе формируемого VHDL-описания, то ключевое слово *generic* не указывается.

Далее приводится информация об интерфейсе описываемого объекта, которая начинается с ключевого слова *port*. После этого ключевого слова приводится последовательность выражений, в которых указывается название порта, режим его работы (направление передачи сигнала, ассоциированного с данным портом) и тип (соответствующий типу ассоциированного с ним сигнала). Для обозначения режима работы порта, соответствующего направлению передачи сигнала, ассоциированного с данным портом, предопределены следующие ключевые слова:

- *in* – соответствует входному порту;
- *out* – соответствует выходному порту;

- *inout* – соответствует двунаправленному (выходному/выходному) порту;
- *buffer* – соответствует выходному порту, сигнал которого можно читать в структуре описываемого объекта.

Сигналу, ассоциированному с входным портом *in* некоторого объекта, нельзя присваивать значения внутри определения архитектуры этого объекта. Значение сигнала выходного порта *out* объекта не может быть присвоено другому сигналу в составе описания архитектуры этого объекта. Сигнал, соответствующий двунаправленному порту *inout*, может использоваться в описании как входной и как выходной. Завершает декларацию объекта ключевое слово *end* с указанием названия этого объекта.

Примером декларации объекта VHDL-описания является объявление объекта *countN*, который соответствует (N+1)-разрядному двоичному счётчику с входом разрешения счёта:

```
entity countN is
generic ( N : natural );
port (
    clk: in std_logic; -- порт тактового сигнала
    ce: in std_logic; -- порт сигнала разрешения счёта
    qout: out std_logic_VECTOR(N
downto 0) -- (N+1)-разрядный порт выходного сигнала
);
end countN ;
```

ОПРЕДЕЛЕНИЕ АРХИТЕКТУРЫ ОБЪЕКТА, ПРЕДСТАВЛЯЮЩЕГО РАЗРАБАТЫВАЕМОЕ УСТРОЙСТВО

Определение архитектуры описываемого объекта выполняется в форме архитектурного тела (*architecture body*). В общем случае архитектурное тело объекта имеет следующий формат:

```
architecture <название_архитектуры> of <имя_объекта> is
[декларация используемых типов данных]
[декларация внутренних сигналов]
[декларация используемых констант]
[определения функций]
[определения процедур]
```

```
[декларация используемых компо-
нентов]
begin
<параллельный оператор1>;
<параллельный оператор2>;
...
<параллельный операторN>;
end <название_архитектуры>;
```

В квадратных скобках здесь и далее приведены необязательные элементы и параметры описания. Определение архитектуры объекта начинается с ключевого слова *architecture*, после которого указывается её название, ключевое слово *of*, имя объекта и ключевое слово *is*. В качестве названия архитектуры объекта, как правило, указывается идентификатор, производный от имени объекта. Выражения декларации типов, сигналов, переменных и констант могут располагаться в составе архитектурного тела в произвольном порядке. Далее следует ключевое слово *begin*, которое открывает блок описания структуры или поведения объекта. Последний заканчивается ключевым словом *end*, сопровождаемым названием архитектуры.

В последующих разделах приводится синтаксис конструкций языка VHDL, используемых для декларации типов данных, сигналов, констант, компонентов и определения функций и процедур. Затем рассматривается формат основных параллельных и последовательных операторов, предназначенных для описания функционирования разрабатываемого устройства.

ДЕКЛАРАЦИЯ ИСПОЛЬЗУЕМЫХ ТИПОВ ДАННЫХ, СИГНАЛОВ, КОНСТАНТ И КОМПОНЕНТОВ, ИСПОЛЬЗУЕМЫХ В СОСТАВЕ ОПРЕДЕЛЕНИЯ АРХИТЕКТУРЫ VHDL-ОПИСАНИЯ ПРОЕКТИРУЕМОГО УСТРОЙСТВА

Синтаксис выражения, предназначенного для объявления нового типа, который используется в определении архитектуры объекта, выглядит следующим образом:

```
type <название_типа> is (<список_возможных_значений>);
```

Декларация нового типа, определяемого пользователем, начинается с ключевого слова *type*, после которого вводится название декларируемого

типа, ключевое слово *is* и список всех возможных значений этого типа. Список возможных значений может указываться в виде перечисления их через запятую или с помощью ключевых слов *range...to*:

```
type <название_типа> is (<значение1>, [<значение2>, ... , [<значениеN>]);
type <название_типа> is range
<начальное_значение> to <конечное_значение>;
```

Например, в следующих выражениях объявляется тип *SYMBOL_HEX*, значениями которого являются шестнадцатеричные символы, и тип *DIGIT_DEC*, список значений которого образует набор десятичных цифр от 0 до 9:

```
type SYMBOL_HEX NAME is ('0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F');
type DIGIT_DEC is range 0 to 9;
```

Если объявляемый тип является массивом, то необходимо использовать один из следующих вариантов синтаксиса декларации:

```
type <название_типа> is array
(<начальное_значение_индекса> to
<конечное_значение_индекса>) of
<тип_элементов_массива>;
type <название_типа> is array
(<начальное_значение_индекса>
downto <конечное_значение_индекса>) of
<тип_элементов_массива>;
```

В объявлении такого типа за ключевым словом *is* следует ключевое слово *array*, после которого указывается диапазон изменения индекса массива (возрастающий или убывающий), ключевое слово *of* и обозначение типа элементов, составляющих массив. Приведённые ниже примеры наглядно демонстрируют различные варианты декларации определяемого пользователем типа, имеющего структуру массива:

```
type CPU_WORD is array (0 to 31)
of BIT;
type BUS_SIGN is array (15 downto 0) of std_logic;
```

Для объявления внутренних сигналов, применяемых в определении ар-

хитектуры описываемого объекта, необходимо использовать следующий формат декларации:

```
signal <идентификатор_сигнала>:
<тип_сигнала>;
```

Объявление внутреннего сигнала начинается с ключевого слова *signal*, вслед за которым указывается идентификатор этого сигнала и его тип. В одной строке может быть объявлено несколько сигналов одного типа. Примерами объявления внутренних сигналов различного типа являются следующие строки декларации:

```
signal SIG_NAME: BOOLEAN;
signal BUS_NAME: BIT_VECTOR (3
downto 0);
signal CLK_1, RESET, SEL:
STD_LOGIC;
```

Формат конструкции, используемой для декларации константы, выглядит следующим образом:

```
constant <идентификатор_константы>: <тип_константы> := <значение_константы >;
```

Объявление константы начинается с ключевого слова *constant*, после которого указывается имя константы и через двоеточие её тип. В большинстве случаев одновременно с объявлением константы производится определение её значения, которое отделяется от обозначения типа константы символом присвоения := (двоеточием и знаком равенства). В качестве значения константы может быть указано не только числовое значение соответствующего типа, но и простое выражение, содержащее ранее объявленные константы, например:

```
constant BUS_WIDTH: integer := 31;
constant WORD_SIZE: integer :=
BUS_WIDTH - 1;
```

Для декларации компонента, используемого в структурном описании объекта, применяется конструкция, формат которой выглядит следующим образом:

```
component <название_компонента>
generic (
<идентификатор_параметра1> : <тип_параметра1>;
```

```

        <идентификатор_параметра2> : <тип_параметра2>;
    ...
    <идентификатор_параметраМ> :
    <тип_параметраМ>
    );
    port (
    <название_порта1>: <направление_передачи_сигнала_порта1>
    <тип_порта1>;
    <название_порта2>: <направление_передачи_сигнала_порта2>
    <тип_порта2>;
    ...
    <название_портаN> : <направление_передачи_сигнала_портаN>
    <тип_портаN>
    );
    end component;
    
```

Открывает объявление компонента ключевое слово *component*, после которого указывается имя компонента. Затем в общем случае приводится ключевое слово *generic*, в списке которого перечисляются настраиваемые параметры объявляемого компонента. Список настраиваемых параметров компонента имеет тот же синтаксис, что и в объявлении объекта описания. Далее следует описание интерфейса этого компонента, которое имеет тот же формат, что и в декларации объекта. Завершает объявление компонента сочетание ключевых слов *end component*. В качестве примера далее приводится текст декларации компонента параллельного регистра данных *REG_N* с настраиваемым количеством разрядов, которое задаётся в виде значения параметра *WIDTH*.

```

component REG_N
    generic (WIDTH: integer);
    port (
    IN_DAT: in
    STD_LOGIC_VECTOR(WIDTH-1 downto 0);
    OUT_DAT: out
    STD_LOGIC_VECTOR(WIDTH-1 downto 0);
    CLK: in STD_LOGIC
    );
    end component;
    
```

ОПРЕДЕЛЕНИЕ ФУНКЦИЙ И ПРОЦЕДУР, ИСПОЛЬЗУЕМЫХ В СОСТАВЕ АРХИТЕКТУРЫ ОБЪЕКТА VHDL-ОПИСАНИЯ ПРОЕКТИРУЕМОГО УСТРОЙСТВА

Для определения функции, используемой в архитектурном теле описы-

ваемого объекта, применяется конструкция, формат которой в общем случае имеет следующий вид:

```

function <название_функции> (
    <идентификатор_параметра1>: <тип_параметра1>;
    <идентификатор_параметра2>: <тип_параметра2>;
    ...
    <идентификатор_параметраN>: <тип_параметраN>
)
return <тип_возвращаемого_значения> is
[декларация локальных типов данных]
[декларация локальных констант]
[декларация локальных переменных]
[определения вложенных функций]
[определения вложенных процедур]
begin
    последовательный оператор
1;
    последовательный оператор
2;
    ...
    последовательный оператор
M;
end <название_функции>;
    
```

Определение функции выполняется с помощью ключевого слова *function*, вслед за которым указывается её имя. После этого в скобках приводится список входных параметров определяемой функции. Для каждого из этих параметров указывается его идентификатор и соответствующий тип. Затем с помощью ключевого слова *return* задаётся тип возвращаемого значения, за которым следует ключевое слово *is*. Далее могут приводиться объявления локальных типов данных, констант и переменных, а также определения вложенных функций и процедур.

Все объявляемые локальные элементы и определяемые вложенные функции и процедуры доступны для применения только внутри определяемой функции. Завершает определение функции описание выполняемых операций (тело функции), которое приводится внутри блока *begin...end*. Это описание, как правило, представлено совокупностью последовательных операторов, которые будут рассмотрены в следующих разделах.

В качестве примера далее приводится определение функции *INCOUNT7*,

которая выполняет инкрементное изменение значения аргумента целого типа *INCNT* в пределах от 0 до 7:

```

function INCOUNT7 (INCNT: INTEGER) return INTEGER is
    variable SUM: INTEGER;
begin
    if INCNT >= 7 then
        SUM := 0;
    else
        SUM : INCNT + 1;
    end if;
    return SUM;
end INCOUNT7;
    
```

Процедуры языка VHDL по своему назначению подобны функциям. Отличия заключаются в том, что процедуры не имеют возвращаемых значений, но при этом, кроме входных аргументов, имеют ещё выходные и входные/выходные параметры. Поэтому при вызове функции её название указывается в правой части выражения присваивания, а вызов процедур аналогичен использованию операторов. Определение процедуры осуществляется с помощью конструкции, формат которой выглядит следующим образом:

```

procedure <название_процедуры>
(
    <класс_параметра1> <идентификатор_параметра1>: <режим_использования_параметра1> <тип_параметра1>;
    <класс_параметра2> <идентификатор_параметра2>: <режим_использования_параметра2> <тип_параметра2>;
    ...
    <класс_параметраN> <идентификатор_параметраN>: <режим_использования_параметраN> <тип_параметраN>
)
is
[декларация локальных типов данных]
[декларация локальных констант]
[декларация локальных переменных]
[определения вложенных функций]
[определения вложенных процедур]
begin
-- содержание процедуры
    последовательный оператор
1;
    последовательный оператор
2;
    ...
    
```

```
последовательный оператор
M;
end <название_ процедуры>;
```

В начале определения процедуры указывается ключевое слово *procedure* и её название. Затем в скобках приводится список всех параметров определяемой процедуры. Для каждого из этих параметров указывается обозначение класса, к которому он относится, его идентификатор, режим использования и соответствующее название типа данных. В качестве параметров процедуры могут использоваться переменные, константы и сигналы, поэтому перед идентификатором параметра приводится наименование класса, к которому он относится.

Режим использования параметра, указывающий направление передачи данных, определяется с помощью ключевых слов *in*, *out*, *inout*, которые соответствуют входному, выходному и входному/выходному параметру. За списком параметров следует ключевое слово *is*, после которого может располагаться блок объявления локальных типов данных, констант, переменных и определения вложенных функций и процедур, предназначенных для применения только внутри определяемой процедуры. Совокупность операций, выполняемых определяемой процедурой (содержание процедуры), описывается внутри блока *begin ... end*. В качестве примера ниже приведён текст определения процедуры маскирования восьмиразрядного сигнала:

```
procedure MASK (
signal XIN: in STD_LOGIC_VECTOR;
signal XOUT: out STD_LOGIC_VECTOR
)
is
constant MSK: STD_LOGIC_VECTOR;
variable TMP: STD_LOGIC_VECTOR;
begin
for I in 0 to 7 loop
TMP(I) := XIN(I) and MSK(I);
end loop;
XOUT <= TMP;
end;
```

Параллельно выполняемые операторы языка VHDL

В отличие от обычных (последовательно выполняемых) операторов, используемых в других языках высокого уровня, в языке VHDL присут-

ствуют операторы, которые выполняются параллельно. Применение таких операторов позволяет наиболее достоверно описать функционирование цифрового устройства. Порядок следования параллельных операторов в определении архитектуры описываемого объекта не оказывает влияния на описание его функционирования. В группу параллельно выполняемых операторов языка VHDL, которые наиболее часто используются для описания разрабатываемых устройств, входят:

- оператор процесса;
- параллельный оператор присваивания значения сигнала;
- оператор условного присваивания значения сигнала;
- оператор выборочного присваивания значения сигнала;
- оператор создания экземпляра (конкретизации) компонента;
- оператор генерации.

Под процессом в языке VHDL понимается совокупность последовательных операторов, которая выполняется «одновременно» с другими параллельными операторами при изменении одного из сигналов, связанных с этим процессом. Процесс может находиться в одном из следующих состояний: в состоянии выполнения, ожидания или приостановленном состоянии. Совокупность сигналов, оказывающих влияние на изменение состояния процесса, называют списком чувствительности данного процесса. Формат оператора процесса в общем случае имеет следующий вид:

```
[<метка>]: process (<идентификатор_сигнала1>, <идентификатор_сигнала2>, ..., <идентификатор_сигналаN>)
[декларация локальных типов данных]
[декларация локальных констант]
[декларация локальных переменных]
[определения вложенных функций]
[определения вложенных процедур]
begin
-- содержание процесса
    последовательный оператор
1;
    последовательный оператор
2;
    ...
    последовательный оператор
M;
end process [<метка>;
```

Началом описания процесса является ключевое слово *process*, после которого в скобках указывается список чувствительности этого процесса. В ряде случаев этот список может отсутствовать. Перед ключевым словом *process* можно задать необязательную метку, которая может использоваться в качестве идентификатора этого процесса. В дальнейшем если метка носит необязательный характер, то она не будет упоминаться при рассмотрении синтаксиса операторов.

После списка чувствительности может располагаться раздел, содержащий объявления типов данных, констант и переменных, а также определения функций и процедур, которые предназначены для использования только внутри этого процесса. Этот раздел не является обязательным и довольно часто отсутствует в описании процесса. Содержание процесса в виде совокупности последовательных операторов приводится после ключевого слова *begin*. В теле процесса могут присутствовать операторы ожидания *wait*, различные формы которого будут рассмотрены в следующем разделе. Завершает оператор процесса сочетание ключевых слов *end process*. Использование оператора процесса иллюстрирует следующий пример:

```
process (CLK)
begin
    SIGN_D0 <= Data_In_A &
Data_In_B;
    SIGN_D1 <= Data_In_A &
Data_In_C;
    if (SIGN_En = '0') then
        Data_OUT <=
SIGN_D0 xor SIGN_D1;
    else
        Data_OUT <=
SIGN_D0 or SIGN_D1;
end process;
```

Для присвоения значений сигналам при определении архитектуры описываемого объекта применяются параллельные операторы безусловного, условного и выборочного назначения сигнала. Формат параллельного оператора безусловного присваивания значения сигнала имеет следующий вид:

```
<идентификатор_сигнала> <=
[<вид_задержки>] <выражение, _оп-
```

```
ределяющее_значение_сигнала>
[after] [<длительность_задерж-
ки>];
```

В левой части этого оператора указывается имя сигнала, которому присваивается значение. В правой части этого оператора может быть указано некоторое конкретное значение, соответствующее типу сигнала, или выражение, которое определяет алгоритм формирования сигнала. Результат этого выражения должен иметь тот же тип, что и сигнал. В операторе присвоения значения сигналу можно также указать задержку, с которой происходит изменение состояния этого сигнала. Для этого используется ключевое слово *after*, которое должно располагаться после присваиваемого значения сигнала. При этом перед выражением, определяющим значение сигнала, можно указать вид задержки с помощью соответствующего ключевого слова.

В языке VHDL используется два вида задержек: инерционный и транспортный, которые обозначаются ключевыми словами *inertial* и *transport* соответственно. При инерционной задержке сигнал изменяет своё состояние только в том случае, если длительность нового состояния (присваиваемого значения) превышает указанное значение задержки. В случае транспортной задержки сигнал изменяет своё значение при любом соотношении длительности этого состояния и величины задержки. Если не указано ключевое слово, определяющее вид задержки, то по умолчанию задержка считается инерционной. Приведённые ниже примеры демонстрируют различные формы использования параллельного оператора назначения сигнала:

```
ADR_SIGN <= N1 and N2;
SIGN_SET <= '1' after 100 ns;
```

При использовании параллельного оператора условного назначения сигнала определённое значение присваивается сигналу только при выполнении соответствующего условия. В одном операторе может быть указано несколько значений и соответствующих им условий. Если не выполнено первое условие, то проверяется выполнение второго и т.д., пока не будет обнаружено выполненное условие. Если ни одно из указанных

условий не выполнено, то сигналу присваивается значение, которое следует за последним ключевым словом *else*.

Для записи условных выражений используются встроенные логические операторы языка VHDL и операторы отношения. Следует обратить внимание на то, что синтаксис оператора отношения «меньше или равно» (<=) аналогичен формату оператора присвоения значения сигнала. Назначение этого оператора определяется контекстом, в котором он используется. Формат параллельного оператора условного назначения сигнала выглядит следующим образом:

```
<идентификатор_сигнала> <= <выра-
жение1, _определяющее_значе-
ние_сигнала> [after] [<длитель-
ность_задержки1>] when <услов-
ное_выражение1> else
<выражение2, _определяющее_значе-
ние_сигнала> [after] [<длитель-
ность_задержки2>] when <услов-
ное_выражение2> else
...
<выражениеN-1, _определяющее_зна-
чение_сигнала> [after] [<длитель-
ность_задержкиN-1>] when <услов-
ное_выражениеN-1> else
<выражениеN, _определяющее_значе-
ние_сигнала> [after] [<длитель-
ность_задержкиN>];
```

В правой части этого оператора приводится значение или выражение, которое определяет алгоритм формирования сигнала, после которого с помощью ключевого слова *when* указывается соответствующее условие. Далее приводится ключевое слово *else*, за которым указываются альтернативные условия и соответствующие значения. При необходимости можно указать задержки изменения значения сигнала в той же форме, что и в предыдущем операторе. Примером условного назначения сигнала является выражение:

```
SIGN1 <= '1' when SIGN2 = '1'
and SIGN3 = '0' else '0';
```

Параллельный оператор выборочного назначения сигнала позволяет присвоить одно из списка возможных значений в зависимости от значения некоторого выражения (выражения выбора). Этот оператор имеет следующий формат:

```
with <выражение_выбора> select
<идентификатор_сигнала> <= <выра-
жение1, _определяющее_значе-
ние_сигнала> [after] [<длитель-
ность_задержки1>] when <значе-
ние1_выражения_выбора>,
<выражение2, _определяющее_значе-
ние_сигнала> [after] [<длитель-
ность_задержки2>] when <значе-
ние2_выражения_выбора>,
...
<выражениеN-1, _определяющее_зна-
чение_сигнала> [after] [<длитель-
ность_задержкиN-1>] when
<значениеN-1_выражения_выбора >,
<выражениеN, _определяющее_значе-
ние_сигнала> [after] [<длитель-
ность_задержкиN>] when others;
```

Началом оператора избирательного назначения сигнала является ключевое слово *with*, после которого указывается выражение выбора. Значение этого выражения выбора определяет соответствующее значение сигнала. Затем следует ключевое слово *select* и идентификатор сигнала. Далее приводятся возможные присваиваемые значения сигнала и соответствующие им значения выражения выбора, разделённые ключевым словом *when*. Необязательная информация о задержках переключения сигнала указывается в той же форме, что и в предыдущих операторах. С помощью сочетания ключевых слов *when others* в конце оператора указывается значение сигнала, которое присваивается в том случае, когда выражение выбора принимает значение, отличающееся от всех перечисленных выше. Например,

```
with BIN_VAL select
      SIG_V <=      "01111001"
when <"0001",
      "00100100"
when <"0010",
      "00110000"
when <"0011",
      "00011001"
when <"0100",
      "00010010"
when <"0101",
      "01000000"
when others;
```

Оператор конкретизации компонента выполняет функцию создания экземпляра этого компонента и определения связей с другими компонентами в структурном описании

объекта проекта. В качестве компонентов могут использоваться определённые ранее функциональные блоки разрабатываемого устройства или библиотечные элементы. Каждый компонент, используемый при определении архитектуры объекта, должен быть объявлен в блоке декларации этой архитектуры. В составе описания объекта проектируемого устройства может использоваться несколько экземпляров одного компонента. Чтобы их различать, каждому экземпляру компонента присваивается оригинальная метка, которая соответствует позиционному обозначению элемента на принципиальной схеме. Синтаксис оператора конкретизации компонента в общем случае выглядит следующим образом:

```
<метка>: <название_компонента>
port map(список_соответствия_портов_и_сигналов);
```

При формировании конкретного экземпляра компонента указывается позиционная метка – название компонента, которое определяется при создании его описания, и с помощью сочетания ключевых слов *port map* определяется список соответствия портов и сигналов (цепей). Этот список может быть представлен в виде позиционного или ключевого сопоставления портов экземпляра компонента и подключаемых сигналов. Поэтому на практике применяется два формата данного оператора. При позиционном сопоставлении подключаемые сигналы в списке перечисляются в том же порядке, что и порты в описании используемого компонента. В этом случае применяется следующий формат оператора конкретизации компонента:

```
<метка>: <название_компонента>
port map (
    <идентификатор_сигнала1>,
    <идентификатор_сигнала2>,
    ...
    <идентификатор_сигналаN>
);
```

При ключевом сопоставлении портов создаваемого экземпляра компонента и подключаемых сигналов порядок их перечисления не имеет значения. Для каждого порта компо-

нента в явной форме указывается идентификатор соответствующего сигнала. Таким образом, может применяться ещё один вариант формата оператора конкретизации компонента:

```
<метка>: <название_компонента>
port map (
    <название_порта1>
=> <идентификатор_сигнала1>,
    <название_порта2> => <идентификатор_сигнала2>,
    ...
    <название_портаN> => <идентификатор_сигналаN>
);
```

В качестве примера ниже приведены операторы создания двух экземпляров компонента *COMP_DIG*, которые демонстрируют применение двух форматов этого оператора:

```
D1: COMP_DIG
    port map(INSIGN_A,
    INSIGN_B, OUTSIGN);
D2: COMP_DIG
    port map(IPORT_A =>
    INSIGN_A,
    IPORT_B => INSIGN_B,
    OPORT_B => OUTSIGN);
```

Для компактного описания регулярных структур предназначен оператор генерации *generate*, который позволяет автоматически формировать повторяющиеся блоки описания архитектуры объекта. Данный оператор может применяться в двух вариантах: циклическом и условном. Циклический вариант оператора генерации имеет следующий формат:

```
<метка>: for <идентификатор_параметра_генерации> in <начальное_значение_параметра_генерации> to <конечное_значение_параметра_генерации> generate
    <параллельный_оператор1>;
    <параллельный_оператор2>;
    ...
    <параллельный_операторN>;
end generate;
```

В начале оператора указывается метка, за которой следует ключевое слово *for*, и идентификатор параметра генерации, диапазон изменения которого определяется с помощью ключевых слов *in* и *to*. Затем указывается ключевое слово *generate*, после которого приводится список парал-

лельных операторов, определяющих содержимое формируемых блоков. Завершает оператор сочетание ключевых слов *end generate*. Параметр генерации аналогичен параметру цикла в различных языках программирования высокого уровня.

Синтаксис условного варианта оператора генерации выглядит следующим образом:

```
<метка>: if <условное_выражение>
generate
    <параллельный_оператор1>;
    <параллельный_оператор2>;
    ...
    <параллельный_операторN>;
end generate;
```

В этом варианте после метки указывается ключевое слово *if* и условие, при котором будет выполняться генерация формируемых блоков. Содержание этих блоков определяет совокупность параллельных операторов, указанных после ключевого слова *generate*. Условное выражение в этом операторе имеет тот же формат, что и в условном операторе назначения сигнала. В качестве примера ниже приведены два варианта использования оператора генерации для формирования блока, выполняющего операции инверсии и поразрядного логического И для пятиразрядных шин:

```
G1: for J in 0 to 4 generate
    D1: INV port map
    (
    I => IN_DAT(J),
    O => OUT_DAT(J) );
end generate;
G2: if L >= 0 and L <= 4 generate
D1: AND2 port map (
    I0
=> IN_ADAT(J),
    I1 => IN_BDAT(J),
    O => OUT_ANDAT(J) );
end generate;
```

Продолжение следует

ЛИТЕРАТУРА

12. Библио П.Н. Основы языка VHDL. Солон-Р, 2000.
13. Библио П.Н. Синтез логических схем с использованием языка VHDL. Солон-Р, 2002.
14. Уэйкерли Дж.Ф. Проектирование цифровых устройств. Том 1. Постмаркет, 2002.



Пакет программ для перевода проекта схемы ПЛИС в базис БМК

Сергей Артёмов (Московская обл.)

В данной статье рассматривается программное обеспечение ConvChip, позволяющее конвертировать проект ПЛИС в базис БМК. Описываются методы перевода и сохранения временных соотношений между элементами схемы в базисах ПЛИС и БМК.

ВВЕДЕНИЕ

В последнее десятилетие проблема создания опытных образцов микросхем стоит особенно остро. Такие образцы необходимы для экспериментального исследования разработанного на ЭВМ проекта микросхемы на соответствующем стенде или непосредственно в аппаратуре. На практике первые опытные образцы микросхем часто не соответствуют предъявляемым к ним требованиям. Поэтому соответствующая модификация микросхемы выполняется на ЭВМ на базовых матричных кристаллах (БМК) с повторным выпуском опытных образцов. Этот процесс может повторяться многократно и длиться многие месяцы, что приведёт к большим финансовым затратам.

Одним из способов решения этой проблемы является отладка разработанного на ЭВМ проекта в аппаратуре типа программируемых логических интегральных схем (ПЛИС). В отличие от многомесячного изготовления опытного образца микросхемы, получение опытного образца ПЛИС с требуемыми функциями занимает несколько минут. Для этого к ЭВМ подсоединяют программатор, который конфигурирует ПЛИС под конкретный проект. Однако использование ПЛИС приводит к другой проблеме. Серийный выпуск ПЛИС стоит значительно дороже, чем выпуск аналогичной партии полупроводниковых микросхем. Особенно это чувствительно, когда необходимо выпустить партию микросхем специального назначения, т.е. с военной приёмкой. Для решения такой проблемы проект в базисе ПЛИС конвертируется в базис БМК полупроводниковых микросхем.

Современные системы автоматизированного проектирования (САПР)

таких фирм, как Cadence (www.cadence.com), Mentor Graphics (www.mentorgraphics.com), Synopsys (www.synopsys.com), Logic Vision (www.logicvision.com) и др., частично решают проблему конвертирования. Для этого проект в базисе ПЛИС импортируется в САПР, после чего динамические параметры всех примитивов в базисе ПЛИС заменяют в соответствии с базисом БМК. Фактически происходит замена базиса ПЛИС на базис БМК.

Если схема разработана на стандартных языках высокого уровня типа VHDL, Verilog 2001, SystemC, SystemVerilog, PSL, то такой подход полностью себя оправдывает [1]. Если схема имеет комбинированное описание, то автоматическое конвертирование таких схем из базиса ПЛИС в базис БМК несколько затруднено.

Существует другой подход, когда исходный проект разрабатывается в базисе БМК, а для отладки проект конвертируется в базис ПЛИС. После отладки ПЛИС в аппаратуре выпускается партия микросхем на основе БМК. Однако такой подход не всегда представляется возможным, поскольку существует большое количество проектов, изначально разработанных в базисе ПЛИС [2, 3].

Современные САПР не адаптированы под асинхронные схемы и ориентированы на разработку исключительно синхронных схем. Если схема частично содержит модули, не удовлетворяющие принципу синхронизации, то конвертирование таких проектов ПЛИС в базис БМК сильно затруднено. Такие схемы после конвертирования оказываются не работоспособными из-за разных динамических параметров базисов ПЛИС и БМК. Такие проекты отлаживаются в САПР с большим трудом.

Тем не менее, необходимость конвертирования проектов смешанных схем в настоящее время является актуальной задачей. Рассматриваемый в данной статье пакет программ ConvChip позволяет конвертировать проекты схем из базиса ПЛИС в базис БМК [4].

Для того чтобы конвертировать схему, необходимо каждому элементу в схеме ПЛИС поставить в соответствие некоторую подсхему замещения из базиса БМК. Оптимальный выбор такой подсхемы и является основной задачей конвертирования, решаемой конвертором ConvChip. При подстановке подсхем замещения из базиса БМК конечная схема часто оказывается неработоспособной. Это проявляется в том, что ответные реакции, полученные при моделировании схемы в базисе ПЛИС, не совпадают с ответными реакциями, полученными при моделировании в базисе БМК (рассматриваются только те моменты времени, когда схема находится в стабильно функционирующем состоянии). Несовпадение возникает в случае логически эквивалентных схем за счёт различий в динамических параметрах элементов, входящих в базисы ПЛИС и БМК, а также отличий архитектуры ПЛИС и полупроводниковой микросхемы.

Для того чтобы восстановить работоспособность схемы в базисе БМК, в конверторе ConvChip выполняются алгоритмы на основе следующих методов:

- метод модификации, позволяющий выполнить автоматическую коррекцию схемы без участия специалиста. Этот метод разделяется на метод перевода схемы и метод сохранения временных соотношений;
- метод диагностики, позволяющий исследовать схему, представленную одновременно в двух базисах, на предмет обнаружения причин её неработоспособности после конвертирования (рассмотрение этого метода не входит в рамки данной статьи [5]).

Метод перевода схемы

Метод перевода схемы в САПР сводится к замене базиса ПЛИС на базис БМК. Для этого исходный проект в базисе ПЛИС импортируется в САПР. После чего по имени элемента схемы в САПР осуществляется его замена на элемент в базисе БМК. Такая замена в современных САПР имеет ряд недостатков.

Если для модуля из ПЛИС существует несколько вариантов замены, различающихся по количеству элементов в них, быстродействию и флуктуационной устойчивости, то выбор в базисе БМК осуществляется без анализа выбора оптимального варианта. Например, если для счётчика с именем В существуют три варианта замены – А, В и С – в базисе БМК, то выбор осуществляется по совпадению имени, т.е. вариант В. При этом анализ вариантов А и С в САПР не выполняется. Но желательно, чтобы выбор варианта происходил не по имени В, а по критерию оптимального соотношения количества элементов, быстродействия и флуктуационной устойчивости.

Если модуль из ПЛИС не представлен одним иерархическим блоком, то он будет заменён на примитивы из базиса ПЛИС поэлементно. Между тем необходим анализ циклов на графе схемы на предмет обнаружения фрагментов, для которых существует единый модуль из базиса БМК. Например, если для некоторой группы вентилях в ПЛИС, соединённых обратными связями, существует неделимый библиотечный элемент в базисе БМК, выполняющий ту же функцию, то необходимо выполнить замену группы вентилях целиком на этот библиотечный элемент. В САПР не реализованы алгоритмы анализа таких групп вентилях. В итоге каждый вентиль переводится в базис БМК отдельно от группы. Это хуже, чем замена на один библиотечный элемент всей группы вентилях целиком.

Реализованные в конверторе ConvChip алгоритмы перевода устраняют данные недостатки.

Обозначим схему ПЛИС за ориентированный граф $G = G(A, R)$, вершины A которого обозначают элементы, их выводы и связи, а $R(a, b)$ – ребро между вершинами $a, b \in A$. Ребро $R(a, b)$ характеризует наличие связи в схеме между элементами a и b . Любому примитиву соответствует граф G' . Введём для G' следующие обозначения:

- $|G'|$ – количество вершин в G' ;
- $\|G'\|$ – быстродействие G' ;
- $\|\|G'\|\|$ – флуктуационная устойчивость G' ;
- $F(G')$ – выполняемая логическая функция G' ;
- $D(G')$ – временная задержка от входа до выхода G' ;
- $\langle D(G') \rangle$ – функция отбрасывания дробной части задержки G' .

Библиотеки базиса БМК достаточно сильно отличаются между собой. В одних набор базисных элементов ограничивается набором различного типа вентилях и простейших триггеров. В других существует широкий набор модулей, выполняющих функции счётчиков, сдвиговых регистров, сумматоров и т.д. Таким образом, существует много вариантов замены одного элемента (примитива или модуля) из схемы в базисе ПЛИС на подсхемы замещения в базисе БМК. Выбор наиболее оптимального варианта и составляет суть задачи, решаемой методом перевода схемы.

При выполнении задачи перевода осуществляется преобразование (или подстановка) $e \rightarrow \tilde{e} = \Omega(G)$. Здесь элемент \tilde{e} в базисе БМК есть образ e в базисе ПЛИС при преобразовании Ω . Фактически схема G в базисе ПЛИС отображается в схему B в базисе БМК. Таким образом, описание метода перевода схемы сводится к описанию преобразования Ω . Этот процесс состоит из следующих этапов:

1. Выбор модуля G'' из схемы в базисе ПЛИС. Если в G существует модуль G'' , т.е. $\exists G'' : G'' \subseteq G$, то необходимо проверить возможность замены G'' целиком на модуль B'' . В САПР такая замена возможна только для модулей G'' , оформленных в виде отдельных иерархических блоков;
2. Поиск двух модулей b и B'' из базиса БМК для одного модуля G' из базиса ПЛИС. Такие модули выполняют одинаковую функцию $F(b) = F(B'') = F(G')$, что и модуль G' ;
3. Поиск одного варианта подсхемы замещения B'' для модуля G'' . Если в базисе БМК существует только одна схема замещения для модуля G'' , т.е. $\exists B'' : F(B'') = F(G'')$, то она сравнивается с вариантом замены каждого примитива G'_i из модуля G'' в базисе ПЛИС на примитив B'_i в базисе БМК. При этом $B'_i = \Omega(G'_i) : G'' = \cup G'_i, B' = \cup B'_i$, где $i = 1 \dots Q$. Здесь G'' является модулем в базисе БМК, а B' не является модулем и представляет собой

подсхему, состоящую из примитивов в базисе БМК и $F(B') = F(G'')$;

4. Поиск одного варианта подсхемы замещения B'' для модуля G'' . Вариант замены G'' целиком на один модуль B'' является более предпочтительным, чем вариант замены B' на множество примитивов B'_i . Иными словами, $G'' \in G, B' = \Omega(G'')$, как правило, менее оптимален, чем $B'' = \Omega(G'')$. Поэтому если хотя бы одно из следующих трёх условий: $|B''| < |B'|, \|B''\| > \|B'\|, \|\|B''\|\| > \|\|B'\|\|$ на данном этапе выполняется, то осуществляется $B'' = \Omega(G'')$. В современных САПР выполнение таких условий не проводится;
5. При наличии для модуля G'' двух вариантов модулей b и B'' они сравниваются между собой (аналогично п. 4). Однако, в отличие от B' в п. 4, здесь модули b и B'' представлены каждый одним модулем. Такое сравнение в известных методах перевода не выполняется;
6. После сравнения по п. 5 для модуля G'' выбирается наиболее оптимальный вариант B'' или b : $B'' = \Omega(G'')$, $b = \Omega(G'')$. При этом $G = gU(UG'_i)UG''$. Здесь G'_i , где $i = 1 \dots Q$, – примитивы, которые не рассматриваются на данной итерации;
7. В G существует часть, состоящая из элементов G'_i , для которых невозможно найти B'' . Поэтому после обработки всех модулей в G осуществляется переход к п. 8 для обработки примитивов, которые не вошли ни в один из рассматриваемых модулей G'' на этапе п. 2;
8. Осуществляется поиск в G циклов $C_i \in G$;
9. Если для циклов существуют модули B'_i в базисе БМК, которые выполняют ту же функцию, что и цикл $F(B'_i) = F(C_i)$, то циклы заменяются полностью без анализа элементов внутри них;
10. При наличии схем замещения B'_i для циклов C_i схема в базисе ПЛИС представляется через множество циклов и примитивов G' , при этом $B'_i = \Omega(G'_i)$, а $B' = \Omega(G') : G = gU(UG'_i)UG''$, где $i = 1 \dots W, G' = \cup G'_i, B' = \cup B'_i$, где $i = 1 \dots Q$. Примитивы G' заменяются стандартными подсхемами замещения B'_i ;
11. При отсутствии схем замещения для модулей $G'' \in G$ и циклов $C_i \in G$, схемы в базисе ПЛИС приводятся к не иерархическому виду, и для каждого примитива G'_i выполняется следующее преобразование: $G = U(G'_i)UG''$, $B' = \cup B'_i$, где $i = 1 \dots Q$, а также $B' = \Omega(G)$,

$B'_i = \Omega(G'_i)$, $F(B'_i) = F(G'_i)$, где B'_i – схемы замещения для всех примитивов G'_i , а Q – их количество в G'' .

Итак, был рассмотрен метод перевода схемы, состоящий из одиннадцати этапов.

Метод сохранения временных соотношений

Целью данного метода является сохранение временных соотношений между элементами в базисах ПЛИС и БМК. Известные САПР реализуют алгоритмы на основе данного метода путём оптимизации фрагментов схемы по теоремам булевой алгебры с учётом установленных временных ограничений и специфики выбранного базиса. Однако такая оптимизация в САПР не всегда эффективна для смешанных схем, содержащих синхронные и асинхронные модули. Реализованные в конвертере ConvChip алгоритмы сохранения временных соотношений на практике успешно дополняют существующие инструментальные средства САПР.

Схема в базисе БМК V считается работоспособной $F(G) = F(B)$ по отношению к схеме G , когда между всеми элементами схемы в базисе ПЛИС в базисе БМК сохранены временные отношения. Таким образом, задача, решаемая методом сохранения временных соотношений, сводится к выполнению следующего требования:

$$\forall B' \in V, \forall G' \in G, D(G')/D(B') = K = \text{const} \Leftrightarrow F(G) = F(B). \quad (1)$$

Суть метода состоит в том, чтобы найти такой коэффициент K , при котором отношение временной задержки D элемента G в базисе ПЛИС и его подсхемы замещения B в базисе БМК для всех элементов схемы было бы постоянным.

Метод сохранения временных соотношений [4, 5] выполняет требование (1) максимально точно. Погрешность выполнения условия (1) зависит только от типа базиса БМК и не зависит от самого метода. Как следует из формулы (1), коэффициент K должен быть одинаков для всех элементов. Он вычисляется для наихудшего случая, т.е. среди множества отношений $D(G'_i)/D(B'_i)$, образуемых всеми элементами схемы:

$$K = \max\{D(G'_i)/D(B'_i)\}. \quad (2)$$

К каждому отношению $D(G')/D(B')$ добавляется задержка из чётного (!) числа Q инверторов I , которые не ме-

няют $F(B')$, но максимально приближают это отношение к коэффициенту K в формуле (2). Иными словами, $\forall G' \in G$ и задержками «подгоняется» отношение $D(G')/D(B')$ для выполнения (2). Причём чем меньше величина задержки инвертора $D(I)$, тем точнее для всех элементов схемы можно приблизить отношение $D(G')/D(B')$ к числу K . Теоретически $D(G')/D(B') \rightarrow K$ при $D(I) \rightarrow 0$. Следовательно:

$$\lim_{D(I) \rightarrow 0} \{D(G') - [K - D(G')/2D(I)]D(I)\} = 0. \quad (3)$$

Однако на практике $D(I)$ – величина постоянная, и для всех элементов схемы условие (2) выполнить невозможно. Поэтому задача оптимизации временных соотношений, выполняемая методом сохранения временных соотношений, записывается следующим образом:

$$|D(G')/[D(B') + Q D(I)] - \max\{D(G'_i)/D(B'_i)\}| = \min. \quad (4)$$

Однако библиотека в базисе БМК может содержать несколько вариантов инверторов I_k с разными задержками $D(I_1) \neq \dots \neq D(I_k) \neq \dots \neq D(I_M)$. Причём количество таких вариантов M зависит от типа САПР. Таким образом, соотношение (4) примет вид:

$$|D(G')/[D(B') + \sum Q_k D(I_k)] - \max\{D(G'_i)/D(B'_i)\}| = \min, \quad (5)$$

где $k = 1..M$, а сумма $\sum Q_k$ всегда является чётной.

Множество различных типов инверторов I_k из M штук путём перебора $\forall Q_k$, где $k \in [1, M]$, образуют большое количество вариантов сумм $S = \sum Q_k D(I_k)$. При этом процесс перебора всех вариантов может занимать неприемлемо много машинного времени при решении соотношения (5) в конвертере. Для сокращения объёма вычислений вводят следующие ограничения:

- а) количество рассматриваемых в формуле (5) отношений $D(G')/D(B')$,
- б) количество типов инверторов M в сумме S ,
- в) количество инверторов I_k конкретного типа k , где $k \in [1, M]$.

Рассмотрим следующие четыре случая для (а), возникающих при сокращении количества элементов для задачи (5).

1. Выбор модуля G'' из схемы в базисе ПЛИС и его замена на модуль в ба-

зисе БМК, т.е. $G'' \in G \rightarrow B'' = \Omega(G'')$. Для решения задачи (5) необходимо рассмотреть $q' + q''$ элементов, где q' и q'' – количество примитивов B' и модулей B'' в схеме в базисе БМК.

2. Для сокращения $q' + q''$ все примитивы e_k , входящие в состав модуля G'' , из рассмотрения исключаются. Поэтому задержка в виде инверторов добавляется не к отдельным элементам $e_1, e_2, \dots, e_q \in G''$, а ко всему модулю G'' . Таким образом, вместо q -отношений $D(e_1)/D(b_1), D(e_2)/D(b_2), \dots, D(e_q)/D(b_q)$, где $b_i = \Omega(e_i), \forall i \in [1, q]$, для формулы (5) рассматривается только одно отношение $D(G'')/D(B'')$.

3. Если в модуле найден цикл $C_i \in G''$, то добавление инверторов выполняется не последовательно к каждому элементу $e_i \in C_i$ а в цепь обратной связи, состоящей из элементов $a_i \in C_i$. Таким образом, отношения $D(e_1)/D(b_1), D(e_2)/D(b_2), \dots, D(e_q)/D(b_q)$, где $b_i = \Omega(e_i), \forall i \in [1, q]$ для (5), $e_i \neq a_i$ для (5) игнорируются.

4. Если модуль G'' не критичен к выполнению временных соотношений (5), то отношение $D(G'')/D(B'')$ исключается из рассмотрения. Как правило, это различные комбинационные подсхемы.

Итак, вместо $q' + q''$, рассматривается теперь количество элементов $q' + q'' - \Delta_q$, где Δ_q – элементы, которые необходимо игнорировать при выполнении какого-либо условия на данном этапе.

Рассмотрим два случая, возникающих при сокращении количества типов инверторов M (вариант б) для задачи (5).

1. Из числа M исключают инвертор I_x , задержка $D(I_x)$ которого может быть вычислена через сумму нескольких других задержек, т.е. $\exists I_x: D(I_x) = \sum Q_j^* D(I_j)$, где $j = 1..q$ при $x \in [1, M]$ и $q \leq M$. Практически поскольку $D(I_x)$ – дробная величина, то для ещё большего уменьшения количества инверторов M вводят параметр $\Delta D \in [\min(D(I_k))/4, \min(D(I_k))^*2]$, $k \in [1, M]$. Таким образом, все инверторы I_x , для которых

$$D(I_x) \in [\sum Q_j^* D(I_j) \pm \Delta D], \text{ где } j = 1..q, \quad (6)$$

исключаются из рассмотрения, т.е. $I_x \in X^*$. На практике уравнение (6) позволяет сократить M примерно в два раза. Таким образом, получается множество инверторов X^* , которые исключаются из рассмотрения.

2. Выбор набора из M инверторов из библиотеки САПР БМК. Если инвертор только один, то задача оптимизации решается по соотношениям (6). Иначе, при $M > 1$ используется выражение (5).

И наконец, рассмотрим два случая для варианта (в) – сокращение максимального количества инверторов для задачи (5).

1. При решении задачи (5) необходимо учитывать свободный объём БМК, на основе которого будет выпущена полузаказная микросхема: $|B| - |UB'_i| - |UB''_i|$, где при B'_i $i = 1..q'$, а при B''_i $i = 1..q''$. Поэтому для каждого модуля $V^k \in V$ и примитива $V^k \in V$, не входящего ни в один модуль, выделяется θ_k процентов свободного ресурса:

$$\theta_K = \left(|B| - \bigcup_{i=1}^{q'} B'_i - \bigcup_{i=1}^{q''} B''_i \right) / \left[\left(\sum_{i=1}^{q'+q''-\Delta q} \max \left\{ \frac{D(G'_i)}{D(B'_i)} \right\} - \frac{D(G'_i)}{D(B'_i)} \right) \times \right. \\ \left. \times 100\% \right] \left[\max \left\{ \frac{D(G'_i)}{D(B'_i)} \right\} - \frac{D(G_K)}{D(B_K)} \right] \quad (7).$$

Фактически чем выше в формуле (7) разница Δ между $D(G_K)/D(B_K)$ и коэффициентом K из формулы (2) (см. выражение в квадратных скобках), тем больше инверторов необходимо добавить для выполнения соотношения (5). Следовательно, чем больше Δ , тем больше θ_k . Знаменатель в формуле (7) характеризует суммарный объём всех таких Δ для элементов $q' + q'' - \Delta q$, рассматриваемых при выполнении соотношения (5). Естественно, что при удачном выборе инверторов M -типа, часть объёма ($\Delta\theta_k \in [0, \theta_k]$) остаётся невостребованной. Поэтому после выполнения задачи (5) для текущего отношения $D(G'_k)/D(B'_k)$ эта часть объёма $\Delta\theta_k$ распределяется между $\theta_1, \theta_2, \dots, \theta_x$ для не вычисленных отношений. Таким образом, θ_k накладывает ограничение на максимальное количество $\sum Q_k$ инверторов I_k , где $k = 1..M$, которые можно использовать для одного элемента при выполнении соотношения (5).

2. Чем длиннее выбирается цепочка из инверторов I_k для элемента V , тем ниже флуктуационная устойчивость этой цепочки. Следовательно, чем больше $\sum Q_k$, где $k = 1..M$, в (5) для некоторого отношения $D(G)/D(B)$, тем больше РВД появится

при флуктуационном моделировании схемы в базисе БМК. Здесь под флуктуационным моделированием понимается моделирование, выполняемое с псевдослучайным разбросом временных параметров. При флуктуационном моделировании обычно генерируется несколько вариантов схемы и несколько вариантов запаздывания входного воздействия. Для выбора конкретных значений уменьшения задержек сигналов и запаздываний входных воздействий используются генераторы случайных чисел, имеющие линейное распределение вероятности в заданном интервале.

На практике на максимальное количество инверторов накладывается дополнительное ограничение $\sum Q_k < 1$. На практике $l \in (2, 4, 6, 8, 10, 12, 14)$, причём выбор l выполняется с учётом условия, накладываемого θ_k .

После выполнения задачи (5) необходимо выполнить оптимизацию результатов. Для этого рассматривается элемент G^* из схемы в базисе ПЛИС и подсхема его замещения V^* , вычисленная по методу перевода Ω .

Если для G^* существует две подсхемы замещения – V^* и V^{**} , причём для замены выбрана V^* , т.е. $G^* \in G \rightarrow V^* = \Omega(G^*)$ и V^{**} более оптимально выполняет (5), чем V^* , то $V^* = \Omega(G^*)$ можно заменить на $V^{**} = \Omega(G^*)$. В таких ситуациях наблюдается противоречие. С одной стороны, V^* может содержать меньше элементов, чем V^{**} , или иметь более высокое быстродействие и флуктуационную устойчивость, чем V^{**} (см. условия для $|V^*|$, $|V^{**}|$, $|V^*|$ в методе перевода). С другой стороны, для V^{**} задача (5) выполняется более оптимально, т.е. V^{**} более точно позволяет приблизить $D(G^*)/D(B^{**})$ к параметру K из (2), чем отношение $D(G^*)/D(B^*)$ (см. описание параметров λ и θ_k). В таких ситуациях теоретически нельзя ответить на то, какая подсхема – V^* или V^{**} – лучше. Поэтому выбор остаётся за специалистом.

В методе сохранения временных соотношений проводится также оптимизация результатов выполнения задачи (5). Для этого анализируется следующее условие. Если полученную сумму задержек для выравнивания отношения $D(G^*)/D(B)$, где $B = \Omega(G^*)$, можно разложить на ряд слагаемых X_j , величина которых совпадает с одним из типов инверторов из множества X^* , исключённых из рассмотрения, то часть инверторов I_j , где $j \in [1, Q_j]$,

входящих в X_j , заменяется на один инвертор $I \in X^*$. Запишем это условие:

$$\sum Q_k^* D(I_k) \text{ [где } k = 1..M] = \sum Q_k^* D(I_k) \text{ [где } k = 1..Q_1] + \sum Q_k^* D(I_k) \text{ [где } k = Q_1 + 1..Q_2] + \dots + \sum Q_k^* D(I_k) \text{ [где } k = Q_i + 1..Q_q] = X_1 + X_2 + \dots + X_q \quad (8)$$

При этом необходимо иметь в виду следующие соотношения: $D(I) = X_j = \sum Q_k^* D(I_k)$, где $k = Q_{j-1}..Q_j$, $M = \sum Q_k$, где $k = 1..q$, причём $\exists X_j, j \in [1, q]$ и $\exists D(I) \in X^*$. При выполнении приведённых условий инверторы $I_k \in [Q_{j-1}, Q_j]$ заменяются на инвертор I (описание множества X^* приведено после формулы (6)).

ПОСЛЕСЛОВИЕ

Не рассмотренные в статье методы диагностики позволяют оценить поведение элементов схемы на основе результатов моделирования, полученных на внутренних связях схемы. К этой группе относятся разнообразные методы, применяемые для сравнения схемы до и после конвертирования. Целью выполнения алгоритмов, реализованных в ConvChip на основе методов диагностики, является анализ схемы на возникновение различных схемотехнических проблем. На сегодняшний день алгоритмы на основании методов диагностики, реализованные в современных САПР, не позволяют получить адекватный анализ схем, частично содержащих асинхронные модули.

ЛИТЕРАТУРА

1. Лохов А. Обзор средств функциональной верификации компании Mentor Graphics. Современная электроника. 2005. № 5.
2. Денисов А.Н., Коняхин В.В., Гаврилов С.В. Средства оперативной разработки полузаказных БИС. Актуальные проблемы твердотельной электроники и микроэлектроники. Таганрог, 2002.
3. Денисов А.Н., Коняхин В.В. Тезисы докладов IV научно-практической конф. «Проблемы обеспечения изделий авиационной и ракетно-космической отрасли высококачественной элементной базой». Сочи, 2003. МНТО РЭС им. А.С. Попова, Москва, 2003.
4. Артёмов С.А. Методы обработки схем при конвертировании в базис базовых матричных кристаллов. Оборонный комплекс научно-техническому прогрессу России. 2005. № 3.
5. Артёмов С.А. Конвертор для перевода проектов схем из САПР ПЛИС в САПР БМК. Изв. вузов. Электроника. 2006. № 2.

Библиотека VERILOG-описаний арифметических операций в поле Галуа

Аркадий Поляков, Мехди Тайлеб, Незхат Тайлеб (Москва)

В статье рассматриваются особенности библиотеки модулей высокоуровневых описаний на языке VERILOG параллельных арифметических операций (сложение, инверсия, умножение) в поле Галуа. Библиотека может быть использована при разработке систем передачи информации. Оцениваются временные и ресурсные параметры аппаратной реализации модулей на ПЛИС типа FPGA фирмы Xilinx.

ВВЕДЕНИЕ

Поля Галуа (Galois Fields – GF), названные в честь французского математика Эвариста Галуа, или конечные поля (Finite Fields), широко используются в различных областях современной информационной техники, связанных с передачей, приёмом и обработкой цифровой информации. Это, в частности, помехоустойчивое кодирование (коды Рида-Соломона), цифровая обработка сигналов, криптография, тестирование БИС и т.п. [1, 2]. От эффективности реализации арифметических операций (в первую очередь, операции умножения) в этих полях существенно зависят аппаратные и временные характеристики соответствующих информационных систем.

Таблица 1. Поле Галуа для $m = 4$ и $p = 19$

Степенное представление	Полиномиальное представление	Бинарное представление
0	0	0000
1	1	1000
α	α	0100
α^2	α^2	0010
α^3	α^3	0001
α^4	$1 + \alpha$	1100
α^5	$\alpha + \alpha^2$	0110
α^6	$\alpha^2 + \alpha^3$	0011
α^7	$1 + \alpha + \alpha^3$	1101
α^8	$1 + \alpha^2$	1010
α^9	$\alpha + \alpha^3$	0101
α^{10}	$1 + \alpha + \alpha^2$	1110
α^{11}	$\alpha + \alpha^2 + \alpha^3$	0111
α^{12}	$1 + \alpha + \alpha^2 + \alpha^3$	1111
α^{13}	$1 + \alpha^2 + \alpha^3$	1011
α^{14}	$1 + \alpha^3$	1001

Упрощённое представление о полях Галуа может быть дано следующим образом. Рассмотрим некоторое количество (начальное множество) различных чисел (символов, элементов поля). Все числа, которые могут быть получены из начального множества путём применения стандартных арифметических операций (сложение, вычитание, умножение и деление), образуют поле. Некоторые поля, как, например, множество целых чисел, являются бесконечными. В отличие от таких полей, поля Галуа являются конечными, т.е. обладают тем полезным свойством, что результатом операции (GF-операции) над одним или несколькими элементами конечного множества является другой элемент того же множества. Например, в поле Галуа, используемом для помехоустойчивого кодирования символов информации в устройстве чтения-записи DVD-дисков, всего 256 элементов – числа от 0 до 255. Операция сложения $2+2$ в таком поле даёт результат, не равный 4.

Поля Галуа характеризуются двумя параметрами: m и p . Параметр m определяет число двоичных разрядов, необходимых для двоичного представления символа множества, а также определяет число элементов множества как 2^m . Таким образом, в поле $GF(2^4)$, где $m = 4$, имеется всего 16 элементов, и для двоичного представления каждого из них достаточно четырёх двоичных разрядов.

Параметр p , или генерирующий полином, определяет порядок, в котором элементы поля следуют друг за

другом. Например, полином $p(x)$ для поля $GF(2^4)$ может быть таким: $p(x) = 1 + x + x^4$. Часто используют сокращённое обозначение полинома как двоичного числа разрядностью $m + 1$, т.е. в нашем случае, если старшие разряды слева, $p = 19$ в десятичной системе, или 10011 в двоичной, или $1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$. Обозначим корень полинома α , т.е. $\alpha^4 = \alpha + 1$.

Элементы поля $GF(2^4)$ представлены в таблице 1 в трёх формах.

1. Степенная форма: нулевой элемент равен 0, первый равен 1, второй равен α в первой степени и т.д.
2. Полиномиальная форма: $x = k_0 \times 1 + k_1 \times \alpha + k_2 \times \alpha^2 + k_3 \times \alpha^3$, где $k_0, k_1, k_2, k_3 = \{0, 1\}$ (старшие разряды справа).
3. Двоичная форма (старшие разряды в данном примере справа!).

Для разработчиков цифровых систем передачи информации была создана параметризованная (параметры m, p) библиотека VERILOG-описаний операций в поле Галуа. Язык VERILOG, наряду с языком VHDL, является общепризнанным стандартом высокоуровневого описания аппаратуры (HDL – Hardware Description Language), используемым при верификации проектов и их реализации в заданном проектировщиком логическом и конструктивном базисе [3]. Общая алгоритмическая компонента языка VERILOG – это язык Си, а не язык АДА (Паскаль), как у VHDL, а его специализированная компонента примерно такая же, как у VHDL, но выражается несколько другими средствами.

Библиотека модулей в определённой части (операция инверсии) учитывает характеристики программируемых логических интегральных схем (ПЛИС), и в частности, ПЛИС типа FPGA фирмы Xilinx (www.xilinx.com). Одной из особенностей этого типа ПЛИС является реализация логических функций с помощью т.н. таблиц решений (логических таблиц, Look Up Table – LUT) и нали-

чие блочной оперативной памяти (Block RAM).

Современные ПЛИС типа FPGA содержат до нескольких десятков тысяч LUT и сотен блоков памяти ёмкостью от 18 Кбит (VIRTEX-4) до 36 Кбит (VIRTEX-5), конфигурируемых в качестве 1-портовой или 2-портовой синхронной памяти с длиной слова от 1 до 36 разрядов.

Библиотечные модули

Операция сложения

В поле Галуа легче всего реализуются операции сложения и вычитания. Это просто m -разрядная логическая операция «исключающее ИЛИ» (XOR, сложение по модулю 2) над её аргументами – двоичными векторами. Например, в таблице 1 двоичное представление второго элемента поля $GF(2^4)$ равно 0100, и результат операции 0100 XOR 0100 равен 0000.

Ниже приведено VERILOG-описание параметризованного модуля *gfadd_m*, выполняющего операцию сложения в поле Галуа. Отметим, что в языке VERILOG символ «^» означает логическую операцию «исключающее ИЛИ» (XOR). Комментарий отделён символами //.

```
module gfadd_m(in1, in2, out1);
//описание интерфейса модуля
//gfadd_m
parameter m=8;
//значение параметра m
//по умолчанию=8
input [m-1:0] in1;
//m-разрядные входы in1, in2
input [m-1:0] in2;
output [m-1:0] out1;
//m -разрядный выход out1
assign out1 = in1^in2;
//оператор вычисления функции
//XOR и присваивания
//в переменную out1
endmodule
```

Тем из читателей, кто знаком с языком VHDL, некоторое представление о языке VERILOG даст пример описания того же модуля *gfadd_m* на языке VHDL:

```
Library IEEE;
Use IEEE.std_logic_1164.all;
entity gfadd_m is
--описание интерфейса модуля
generic (m: positive:=8);
port (in1,in2:in
std_logic_vector(m-1 downto 0);
```

```
out1: out std_logic_vector(m-1
downto 0)
);
end;
architecture beh of gfadd_m is
--описание тела модуля
begin
out1 <= in1 XOR in2;
--оператор вычисления функции
XOR и присваивания в out1
end;
```

При использовании системы автоматизированного синтеза XST САПР ISE v8.2i фирмы Xilinx и значения параметра $m = 8$ аппаратная реализация *gfadd_m* на ПЛИС типа FPGA требует всего 8 LUT.

Операция умножения

Известно большое число исследований, посвящённых методам аппаратной реализации операции умножения в этих полях [4, 5], и в частности, на базе ПЛИС типа FPGA.

Алгоритм параллельного умножения, предложенный Мастровито (Mastrovito), позволил получить оценки сложности аппаратной реализации в $m^2 - 1$ одноразрядных вентилей XOR2 плюс m^2 одноразрядных вентилей 2И (AND2). В последующих работах по развитию этого подхода эти оценки были несколько улучшены до порядка $m^{1.6}$ вентилей XOR2.

Ниже приведено VERILOG-описание интерфейса параметризованного модуля *gfmul* (обозначим его как вариант P), реализующего умножение (стандартное значение $m = 8, p = 285$) [4]:

```
module gfmul(a,b,c);
//умножение, вариант P по методу
//[4]
// GF PARAMETERS
parameter m=8;
//разрядность символа поля m
//по умолчанию=8
parameter [m:0] p=285;
//p - генерирующий
//полином=2**8+2**5+1
//Internal parameters - вспомо-
//гательные внутренние параметры
parameter [m-1:1] p1= p[m-1:1];
parameter v=m*2-2;
//PORTS - описание портов модуля
input [m-1:0] a,b;
//этот модуль предполагает, что
//старшие разряды слева
output [m-1:0] c;
```

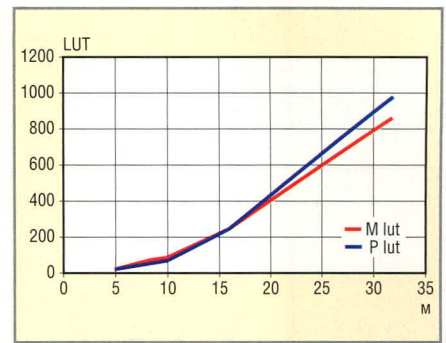


Рис. 1. Графики зависимости объёма аппаратуры умножителей M и P от параметра m

Оценки вариантов схемной реализации модуля *gfmul* на кристалле ПЛИС фирмы Xilinx типа XCV-41x60 для разных m и p представлены в таблице 2 в столбцах с префиксом P.

Синтезатор XST фирменного САПР Xilinx ISE 8.2 строит схемы, сложность которых, измеряемая в LUT, примерно пропорциональна квадрату m (см. табл. 2 и рис. 1).

Для обеспечения возможности динамического изменения генерирующего полинома p используется другой модуль – *gfmul_p(a,b,c,p)*, в котором в список сигналов введён сигнал p . Платой за эту возможность является более сложная и медленная схема. При $m = 8$ и $p = 285$ она имеет задержку 7,23 нс и требует 89 LUT, т.е. примерно в полтора раза больше, чем модуль *gfmul*.

Ниже приведён интерфейс другого параметризованного модуля *GF_MUL*, более эффективно реализующего умножение (стандартное значение $m = 5, p = 37$). В этом модуле реализован алгоритм работы [5]. Вместо параметра p используется эквивалентный ему параметр q , состоящий из нескольких элементов GF. Например, для поля ($m = 5, p = 37$), q – это вектор из четырёх значений элементов поля ($\alpha^6, \alpha^7, \alpha^8, \alpha^9$).

Таблица 2. Зависимость объёма аппаратуры и временной задержки от m для модулей *GF_MUL* и *gfmul*

m	p	M lut	M del (ns)	P lut	P del (ns)
5	37	20	2,702	20	2,07
8	301	70	3,28	53	4,76
10	1033	84	3,274	75	3,37
12	4179	145	3,847	121	4,02
16	75763	245	3,87	244	7,51
32	7115993485	866	4,49	976	23,54

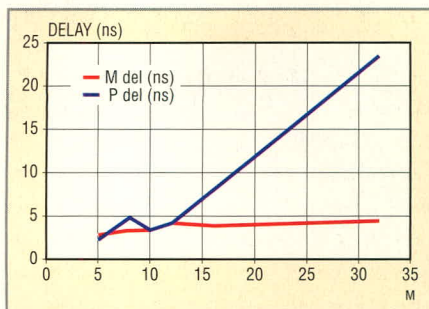


Рис. 2. Графики зависимости задержки (DELAY) умножителей M и P от параметра m

```

module GF_MUL(A,B,C);
//Parameters
parameter m=5;
parameter
q=20'b10010100101101010010;
//Ports
input [m-1:0] A, B;
//старшие разряды справа
output [m-1:0] C;
    
```

Зависимости сложности и задержки схемы модуля *GF_MUL* для различных *m* и *p* представлены в таблице 2 в столбцах с префиксом M. Видно, что если при малых *m* оба варианта умножителей (P и M) имеют примерно равные показатели, то при больших *m* (*m* > 16) временная задержка варианта M (M del) гораздо меньше (см. рис. 2).

Операция инверсии

Формулы для вычисления инверсии в поле Галуа [6] прямо вытекают из теоремы Ферма. Не вдаваясь в математику, можно сказать, что инверсия вычисляется как цепочка операций умножения и возведения в квадрат, которое, в свою очередь, реализуется как умножение величины самой на себя. Например, для GF(2⁴) инверсия (I) элемента x (I(x)) вычисляется по формуле:

$$I(x) = x^{-1} = x^2 \cdot x^{2^2} \cdot x^{2^3} \cdot x^{2^4}.$$

Текст описания параметризованного модуля *inverse_comb* для вычисления инверсии с использованием *gfmul* представлен ниже:

```

parameter m= 8;
//параметры поля Галуа
parameter p= 285;
input [m-1:0] x;
output [m-1:0] y;
wire [m-1:0] mtmp[1:m-1];
wire [m-1:0] inv[1:m-1];
wire [m-1:0] inv_rez;
    
```

```

genvar i;
gfmul #(m,p) INV_M1(x, x,
mtmp[1]);
//beta^2
assign inv[1]=mtmp[1];
generate
for (i=1; i<m-1; i=i+1)
begin :MM
gfmul #(m,p) INV_MN(mtmp[i],
mtmp[i], mtmp[i+1]);
gfmul #(m,p) INV_MK(inv[i],
mtmp[i+1], inv[i+1]);
end
endgenerate
assign y=inv[m-1];
endmodule
    
```

Если при *m* = 5 и *p* = 59 комбинационная схема *inverse_comb* имеет вполне приемлемые показатели, то при *m* = 8 и *p* = 285 количество LUT равно 362, а задержка составляет 25 нс.

Очевидно, что для таких значений *m* ресурсные и временные затраты на реализацию инверсии модулем *inverse_comb* слишком велики. Можно попытаться использовать табличные решения (ПЗУ). В блоке памяти (Block RAM) объемом 16 Кбит удастся таблично реализовать инверсию до *m* = 9, а при *m* > 9 можно использовать несколько блоков.

В модуле *ROM2_POL* предложено решение для *m* = 8 на базе 2-портовой памяти, которая позволяет одновременно работать с двумя таблицами.

```

//the first file to
//ADR= from 0 0000 0000 to 0
//1111 1111
//the second file to ADR= from
//1 0000 0000 to 1 1111 1111
module
ROM2_POL(CLKA,CLKB,WE,EN,OUT_EN,A
DRA,ADRB ,DI,DOA,DOB);
//Parameters
parameter adr_nbits=8;
parameter data_nbits=8;
parameter
file_name1="inverse_table_8_0_285
.rom";
parameter
file_name2="power_table_8_239_0_2
85.rom";
//Input Ports
input CLKA,CLKB,WE,EN,OUT_EN;
input [adr_nbits-1:0] ADRA,ADRB;
input [data_nbits-1:0] DI;
//Output Ports
output reg [data_nbits-1:0]
DOA,DOB;
    
```

```

//DOA-выход табл1-inverse_tabl
//Internal ROM duble size! 9
//bits addr!
reg [data_nbits-1:0] rom
[0:2**(adr_nbits+1)-1];
reg [data_nbits-1:0]
DOA_AUX,DOB_AUX;
initial begin
//заполнение ПЗУ из файлов
$readmemb(file_name1,rom,0,2**(ad
r_nbits)-1);
$readmemb(file_name2,rom,2**(adr_
nbits),2**(adr_nbits+1)-1);
end
//Behavioural Statements
//FIRST OUT PORT
always @(posedge CLKA)
if (EN)begin
if (WE)
//при WE=0 имеем ПЗУ
rom[{1'b0,ADRA} ]<=DI;
else
DOA_AUX<=rom[ {1'b0,ADRA} ];
if (OUT_EN)
DOA<=DOA_AUX;
end
//SECOND OUT PORT
always @(posedge CLKB)
if (EN)begin
DOB_AUX<=rom[{1'b1,ADRB}];
if (OUT_EN)
DOB<=DOB_AUX;
end
endmodule //ROM2_POL
    
```

Рассмотренная библиотека VERILOG-модулей параллельных арифметических операций в полях Галуа была использована при разработке декодера кода Рида-Соломона.

ЛИТЕРАТУРА

1. *Sylvester J.* Reed Solomon Codes. Electrobitt. January 2001.
2. *Moon T.K.* Error correction coding, mathematical methods and algorithms. John Wiley & Sons, 2005.
3. *Поляков А.К.* Языки VHDL и VERILOG в проектировании цифровой аппаратуры. Солон-П, 2003.
4. *Iliev N., Stine J., Juchimiec N.* Digital Finite – Field Multiplier for Reed-Solomon Channel codes in GF(2ⁿ) with programmable basis polynomial. ИТ VLSI LAB, 2003.
5. *Reybani-Massolem A., Hasan M.A.* Low Complexity Bit Parallel Architectures for Polynomial Basis Multiplication over GF(2^m). IEEE Transaction on Computers. 2004. V. 63. № 8.
6. *Choi S., Kim K., Lee W., Kim K.* A Finite Field Inversion Circuit for Higher-Speed communications. KERI. Korea, 2003.

Новости мира News of the World Новости мира

Toshiba перейдёт на 43 нм в течение года?

Намереваясь опередить компанию Samsung Electronics в технологической гонке, Toshiba планирует в конце 2007/начале 2008 г. запустить производство флэш-памяти типа NAND по 43-нм проектным нормам. Переход на более прецизионные нормы позволит Toshiba существенно сократить удельную стоимость микросхем, получаемых с кремниевой пластины.

По оценкам издания Nikkei business daily, при 43-нм производстве количество чипов, получаемых с одной пластины, увеличивается примерно на 40% по сравнению с 56-нм. Таким образом, Toshiba будет иметь потенциальный 40-процентный запас по снижению цен на свою продукцию, что даст ей неоспоримые преимущества перед конкурентами. Как известно, на данный момент технологическим лидером в этой отрасли является компания Samsung, которая ещё в марте начала отгрузки образцов 50-нм NAND-чипов.

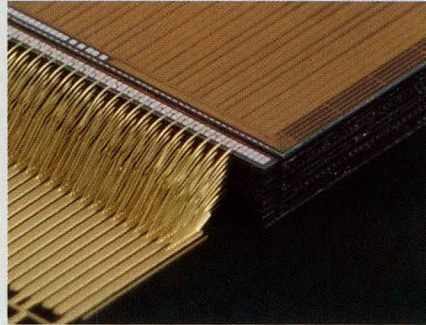
Потеснит ли Toshiba своего главного конкурента в технологической гонке? Успеет ли она вовремя и без задержек перейти на инновационное производство? Пока ответить на эти вопросы сложно, ведь сам производитель ещё даже официально не подтвердил свои планы по переходу на 43 нм. Источник утверждает, что к концу текущего года Toshiba построит завод для 43-нм производства в юго-западной префектуре Мие (Mie Prefecture), что в Японии. Этот завод станет четвёртым по производству чипов памяти в Японии.

cdrinfo.com

20 DRAM-чипов в упаковке толщиной 1,4 мм

Молодая компания Akita Elpida Memory (создана летом 2006 г. по инициативе Elpida, известного производителя DRAM-памяти) сообщила о разработке нового метода упаковки чипов. С его помощью ей удалось впервые в мире поместить 20 полупроводниковых кристаллов DRAM в одной MCP-упаковке (multi-chip package) толщиной всего 1,4 мм.

Для достижения успеха инженеры Akita Elpida разработали инновационную технологию для утонения (шлифовки) кристаллов толщиной 30 мкм перед упаковкой, метод соединения слоёв с помощью тончайших проводников, технику впрыскивания канифоли в микроскопические отверстия. Компания намерена тесно сотрудничать с производителями оборудова-



ния для дальнейшего продвижения своей разработки и коммерциализации производства.

Хочется отметить важность подобных разработок в свете непрерывного процесса миниатюризации портативной техники и растущих требований к её производительности. В этой области работает множество полупроводниковых компаний. Из недавних достижений стоит отметить разработки таких производителей, как IBM и Samsung Electronics.

www.3dnews.ru

Samsung на пути к созданию ОЗУ нового поколения

Одним из важнейших технологических достижений в полупроводниковой отрасли за последний месяц стала разработка компанией IBM методики так называемой «трёхмерной» упаковки чипов. Нечто подобное на днях анонсировала и компания Samsung Electronics.

Как сообщается в пресс-релизе, Samsung разработала метод упаковки чипов памяти, использующий технологию TSV (through silicon vias, внутрикремневые межсоединения). По заявлению компании, это позволит существенно ускорить память, уменьшить энергопотребление и габариты микросхем.

Новая упаковка называется WSP (wafer-level-processed stacked package). Она может вмещать четыре чипа DDR2 DRAM плотностью 512 Мбит (4 × 512 Мбит). Используя такие двухгигабитные структуры, Samsung может создать модули ОЗУ ёмкостью 4 Гб.

Инновационный технологический метод Samsung устраняет необходимость в относительно длинных металлических проводниках, которые соединяют между собой традиционные «двухмерные» чипы и их составные элементы, заменяя эти проводники внутрикремневыми соединениями. Межсоединения TSV представляют собой вертикальные каналы диаметром порядка единиц микрон, протравленные в кремниевой пластине с по-

мощью лазера и заполненные проводником – медью. Такие внутрикремневые соединения позволяют располагать кристаллы плотнее и создавать более тонкие упаковки. Межсоединения through-silicon vias покрыты алюминием, который играет роль экрана, в результате чего снижаются перекрестные помехи. Конкретные сроки внедрения новой разработки в массовое производство пока не называются.

www.3dnews.ru

После Intel о 450-мм пластинах заговорила TSMC

Компания TSMC (Taiwan Semiconductor Manufacturing Company) сформировала группу, заданием которой является оценка осуществимости перехода на производственный процесс с использованием кремниевых пластин диаметром 450 мм. Представители TSMC подтверждают заинтересованность в переходе на 450-мм пластины, но отмечают, что о каких-либо сроках внедрения нового производства речь пока идти не может. Напомним, что на Форуме IDF Spring 2007, который недавно завершился в Пекине, компания Intel также выразила сильную заинтересованность в 450-мм производстве.

На данный момент в полупроводниковой индустрии лидерство продолжают удерживать 200-мм фабрики, которые могут похвастаться объёмом производства в 380...390 тыс. пластин в месяц. 300-мм фабрики постепенно набирают обороты, и уже сегодня количество чипов, изготавливаемых на новых производственных линиях за месяц, эквивалентно производству 200 тыс. 200-мм пластин.

Переход на 450-мм пластины понизит удельную стоимость микросхем, но первоначальные вложения в такое производство в три раза превышают инвестиции в 300-мм производство. Поэтому, как считают эксперты, компаниям стоит задуматься о переходе именно на 300-мм пластины – это самое оптимальное решение на сегодняшний день. В 2009 г. 300-мм фабрики полностью вытеснят старые 200-мм.

По оценкам специалистов IEK (Taiwan's Industrial Economics and Knowledge Center), среди тайваньских компаний в будущем одними из первых 450-мм фабрики построят TSMC, Powerchip Semiconductor Corporation (PSC), Nanya Technology и ProMOS Technologies. Но когда будет построена первая 450-мм фабрика? На этот вопрос пока никто не может дать определённый ответ.

digitimes.com

Оптимизация логических схем заказных СБИС в синтезаторе LeonardoSpectrum

Пётр Бибило (г. Минск, Белоруссия)

В статье исследованы различные способы оптимизации логических схем, описанных на языке VHDL, посредством синтезатора LeonardoSpectrum.

Правильный выбор и умелое использование системы синтеза (или синтезатора) является одним из решающих факторов эффективного проектирования заказных СБИС (ASIC – Application-Specific Integrated Circuits).

Во-первых, синтезатор должен поддерживать язык высокого уровня, на котором описан исходный проект (т.е. языки VHDL или Verilog). Во-вторых, синтезатор должен обеспечивать применение целевой библиотеки синтеза, разработанной пользователем, и – в-третьих – допускать гибкое управление процессом синтеза.

Одним из синтезаторов, удовлетворяющих основным требованиям, является LeonardoSpectrum фирмы Mentor Graphics (далее – Leonardo).

Этот синтезатор позволяет описать собственную библиотеку логических элементов и проектировать логические схемы по исходным описаниям проектов на языках VHDL и Verilog с учётом различных технологических ограничений.

Синтез в Leonardo можно проводить несколькими способами [1].

Способ 1 – для начинающих пользователей. В окне графической оболочки указывается только исходное VHDL-описание, целевая библиотека синтеза и степень (глубина) оптимизации. Синтез может выполняться в двух режимах: *quick* (быстром) и *standard* (стандартном).

Способ 2 – управление синтезом из графической оболочки. Установки

режимов синтеза и ограничений в многочисленных окнах графической оболочки интерпретируются как соответствующие консольные команды.

Способ 3 – управление синтезом с помощью скриптов (сценариев), т.е. наборов консольных команд, собранных в отдельные текстовые файлы.

Для сравнения эффективности способов оптимизации были проведены эксперименты с различными вариантами описания исходных проектов.

В качестве исходных описаний были взяты 25 проектов: 10 проектов описаний комбинационных схем ПЛМ (программируемых логических матриц) из известной библиотеки [2] примеров логических схем, 10 проектов многоуровневой комбинационной логики, два описания таблиц микрокоманд микроконтроллеров и три алгоритмических описания (см. табл. 1). Все описания были выпол-

Таблица 1. Результаты экспериментов

N	Вид описания	Название схемы	n (входы)	m (выходы)	Эксперимент 1 $S_{\text{БМК}}$	Эксперимент 2 $S_{\text{БМК}}$	Эксперимент 3 $S_{\text{БМК}}$	Эксперимент 4 $S_{\text{БМК}}$
1	ПЛМ	b2	16	17	3585	2474	2034	2163
2		bc0	26	11	5993	5993	3760	3207
3		chkn	29	7	1461	1319	1104	1017
4		dk48	15	17	389	389	395	398
5		ibm	48	17	630	630	538	530
6		in0	15	11	2107	2067	1539	1381
7		in2	19	10	2217	1845	1405	1151
8		prom2	9	21	10671	9852	9421	7014
9		signet	39	8	796	727	668	658
10		tial	14	8	5013	4505	4533	4344
11	Многоуровневая логика	c1355	41	32	1128	1128	1080	1076
12		c432	36	7	544	633	499	401
13		c880	28	18	270	271	269	241
14		dalu	75	16	1803	1803	1254	503
15		frg1	28	3	421	266	276	210
16		frg2	143	139	3491	3604	3490	2533
17		i3	132	6	248	248	244	236
18		i8	133	81	2898	2898	2761	2128
19		too_larg	38	3	9244	9244	7771	1047
20		x4	94	71	1334	1334	1249	1240
21	Не полностью определённые функции	verg1	17	61	16388	7571	6674	10899
22		verg2	18	63	16082	–	–	15665
23	Алгоритмы	uart	n+m=26		1896	1869	1633	1632
24		watchdog	n+m=20		461	446	443	402
25		timer2	n+m=31		3119	3119	3089	3089

нены на языке VHDL. Описания *verg1*, *verg2* (см. табл. 1) представляют собой таблицы микрокоманд отечественных микроконтроллеров, описываемых на языке VHDL в качестве систем не полностью определённых (частичных) логических функций. Последние три описания являются алгоритмическими (в основном используются операторы процессов языка VHDL, операторы назначения сигналов и последовательные операторы): *uart* – универсальный асинхронный приёмопередатчик информации, представленной последовательным кодом; *watchdog* – сторожевой таймер; *timer2* – таймер-счётчик.

В качестве примера приведём VHDL-описание системы частичных логических функций *sum_sc*. (В подобной форме задавались описания *verg1*, *verg2*, содержащие более двух тысяч строк.) Подсхема *sum_sc* представляет собой сумматор для сложения пары чисел, выбираемых из множества {0, 1, 2}, т.е. неполный сумматор. Данный сумматор описывается системой частичных функций, заданной в таблице 2. Если одно из слагаемых представляет собой число 3, то значения выходов сумматора не определены. Для кодирования каждого из слагаемых необходимы две булевы переменные. Первое число задаётся в виде вектора **a** = (a(1), a(0)), второе – в виде вектора **b** = (b(1), b(0)), старшие разряды – a(1), b(1); старший разряд суммы – s(2), для представления вектора суммы **s** = (s(2), s(1), s(0)) достаточно трёх булевых переменных.

```

Листинг 1. VHDL-модель схемы sum_sc
library ieee;
use ieee.std_logic_1164.all;
entity sum_sc is
  port (a, b : in std_logic_vector (1 downto 0);
        s : out std_logic_vector (2
downto 0));
end;
architecture BEHAVIOR of sum_sc
is
begin
  s <=
"000" when a & b = "0000" else
"001" when a & b = "0001" else
"010" when a & b = "0010" else
"001" when a & b = "0100" else
"010" when a & b = "0101" else
"011" when a & b = "0110" else

```

```

"010" when a & b = "1000" else
"011" when a & b = "1001" else
"100" when a & b = "1010" else
"---";
end BEHAVIOR;

```

В качестве целевой библиотеки ASIC была выбрана библиотека проектирования базовых матричных кристаллов (БМК), состоящая из 35 элементов [3, стр.159].

Эксперимент 1 (Исследование оптимизации в режиме *quick*) и *эксперимент 2* (Исследование оптимизации в режиме *standard*) оставляем без комментариев.

Эксперимент 3. Итеративный синтез без смены целевой библиотеки. Синтез схемы от алгоритмических описаний на языке VHDL в Leonardo разбит на два этапа – высокоуровневый синтез, результатом которого является так называемое промежуточное RTL-описание (RTL – Register Transfer Level), и технологическое отображение (technology mapping) [2]. RTL-описание может быть получено повторно в Leonardo (этим он выгодно отличается от других синтезаторов) из результирующего структурного описания синтезированной логической схемы с помощью специальной команды *итнар*. Назовем это описание RTL0. Естественно, описание RTL0 функционально соответствует исходному алгоритмическому VHDL-описанию. По описанию RTL0 может быть проведён повторный синтез схемы FPGA как в синтезаторе Leonardo, так и в других синтезаторах, строящих логические схемы по VHDL-описаниям. На практике было замечено, что вновь построенная схема иногда обладает лучшими характеристиками, чем схема, построенная по исходному VHDL-описанию. Таким образом, синтезатор Leonardo позволяет уменьшить сложность схемы путём повторного (итеративного) синтеза. В эксперименте 3 для каждого исходного описания проводилось пять итераций повторного синтеза и отбирались лучшие результаты.

Эксперимент 4. Итеративный синтез с поочередной сменой целевой библиотеки синтеза. В этом эксперименте схема сначала синтезировалась в библиотеке FPGA (микросхема XC2S100 семейства SPARTAN II), затем выполнялась команда *итнар*, и полученное RTL-описание синтезирова-

лось в целевой библиотеке БМК. Всего проведено пять итераций повторного синтеза со сменой библиотек, и затем отобраны лучшие результаты.

Результаты экспериментов представлены в таблице 1. Жирным шрифтом выделены лучшие – по занимаемой схемой площади – решения. Площадь схемы подсчитывалась как сумма площадей, входящих в данную схему элементов (Leonardo позволяет подсчитать суммарную площадь всех элементов схемы), а площадь элемента выражалась в числе элементарных ячеек БМК. Элементарная ячейка БМК соответствует одному транзистору, входящему в сеть транзисторов, из которых состоит логический элемент. При оценке сложности схемы, синтезированной в библиотеке БМК, в расчёт не принималась площадь, отводимая под межсоединения. Общую площадь схемы можно получить после выполнения трудоёмкого этапа топологического проектирования. На практике установлено, что минимизация суммарной площади всех элементов схемы в общем случае приводит к меньшей площади кристалла.

Для удобства поток примеров обрабатывался единообразно с помощью соответствующих скриптов. Пример скрипта, использованного в эксперименте 2, приведён в листинге 2.

Листинг 2. Скрипт для эксперимента 2

```

clean_all;
set encoding Gray;
set modgen_select Smallest;
set asic_auto_dissolve_limit 500;
set auto_dissolve_limit 500;
read b2.vhd;
load_library bmk.syn;
set -hierarchy flatten

```

Таблица 2. Система не полностью определённых булевых функций

a(1) a(0) b(1) b(0)	s(2) s(1) s(0)
0000	000
0001	001
0010	010
0011	---
0100	001
0101	010
0110	011
0111	---
1000	010
1001	011
1010	100
1011	---
1100	---
1101	---
1110	---
1111	---

```
set effort standard
optimize -target bmk -macro
-area -effort standard -hierarchy flatten
report_area -cell_usage
```

Если требовалось получить RTL-описание (эксперименты 3 и 4) для повторного синтеза, то в скрипт (листинг 2) добавлялись две команды: *unmap* и *auto_write b2.vhd*.

В листинге 2, кроме значения *standard*, жирным шрифтом выделены важные для синтеза значения *flatten* параметра *hierarchy* и значение 500 параметра *auto_dissolve_limit*. Установка значения *flatten* параметра *hierarchy* ориентирует программу на выполнение синтеза с учётом устранения иерархии описания, т.е. схема синтезируется в виде одного блока, поскольку в этом случае имеется больше возможностей для оптимизации. Параметр *auto_dissolve_limit* задаёт число «растворяемых» элементов при оптимизации. Под «растворением» понимается преобразование структурного описания подсхемы в функциональное описание. По умолчанию значение этого параметра для ASIC равно 30. При эксперименте задаётся значение 500, тогда результирующие схемы имеют меньшую площадь и оптимизируются подсхемы большей размерности. Однако если задать значение 1000, то время синтеза может значительно возрасти. Задание значения *Gray* параметра *encoding* важно при синтезе схем с па-

мятью, – они получаются более экономичными.

В результате проведённых с использованием синтезатора Leonardo экспериментов можно сделать следующие выводы:

- Режим *standard* позволяет значительно лучше оптимизировать логические схемы, чем режим *quick*, что не является неожиданностью. Эксперименты 1 и 2 показывают, сколько можно выиграть по площади, если использовать режим *standard*. Для минимизации площади весьма важен правильный выбор значений параметров: *hierarchy*, *auto_dissolve_limit*;
- Итеративный синтез является эффективным для достаточно больших блоков комбинационной логики и особенно – для частичных функций. Для алгоритмических описаний общего вида эффективность повторного синтеза снижается, так как в таких схемах не выделяются блоки комбинационной логики. Поэтому их отдельная предварительная оптимизация вне синтезатора Leonardo может улучшить характеристики схем, например, значительно уменьшить задержку [4];
- Предварительная логическая ИЛИ минимизация матричных форм логических функций (описания ПЛМ) в классе дизъюнктивных нормальных форм значительно уменьшает сложность схем ASIC;
- Предварительный синтез в базе FPGA по сути заменяет минимизацию

цию в классе ДНФ минимизацией многоуровневых представлений функций, поэтому итеративный синтез со сменой библиотек позволяет получить лучшие результаты, чем простой итеративный синтез;

- Синтезатор Leonardo может быть использован для совместной работы с другими синтезаторами, работающими с языком VHDL, так как Leonardo позволяет сохранять промежуточные и результирующие VHDL-описания логических схем в виде RTL-описаний;
- Синтезатор Leonardo может быть использован при проектировании FPGA в системе WebPack ISE 8.1i (фирма Xilinx); синтез в этом случае осуществляется в режиме *quick*, поэтому площадь схем, синтезированных Leonardo, зачастую бывает больше площади схем, которые получает синтезатор XST, имеющийся в составе WebPack ISE.

ЛИТЕРАТУРА

1. Библио П.Н. Системы проектирования интегральных схем на основе языка VHDL. StateCAD, ModelSim, Leonardo-Spectrum. СОЛОН-Пресс, 2005.
2. <http://www1.cs.columbia.edu/~cs4861/sis/espresso-examples/ex/>.
3. Библио П.Н. Синтез логических схем с использованием языка VHDL. Солон-Р, 2002.
4. Библио П.Н., Кочанов Д.А. Оптимизационные преобразования VHDL-моделей цифровых систем. Современная электроника. 2006. № 5. С. 64–66.



Новости мира News of the World Новости мира

Оптический сенсор уменьшит количество автомобильных аварий

Немецкая компания Continental AG, один из ведущих поставщиков комплектующих для автомобильной промышленности, объявила о завершении разработки нового оптического сенсора, призванного уменьшить количество автомобильных аварий. Встроенный в зеркало заднего вида сенсор с помощью лазерного луча измеряет скорость изменения скорости приближения ближайшего к автомобилю участнику дорожного движения и автоматически отдаёт команды тормозной системе последнего.

Представленное устройство, разработанное в рамках проекта Continental APIA (Active/Passive Integration Ap-

proach), постоянно «просматривает» пространство перед автомобилем на 10 м вперёд. Если расстояние до объекта в зоне видимости начнёт уменьшаться слишком быстро, система активирует тормозную систему. Более того, если столкновение неизбежно, система может выпустить подушки безопасности, не дожидаясь удара.

Согласно заверениям компании, представленная система обладает достаточной чувствительностью для обнаружения не только автомобилей, но и велосипедистов и пешеходов. Однако пока убедительные результаты было получены при скорости автомобиля не более 35 км/час. Ожидается, что массовое оснащение автомобилей подобными сенсорами начнётся в 2008 г.

eetimes.com

Intel продемонстрировала прототип 80-ядерного ПК

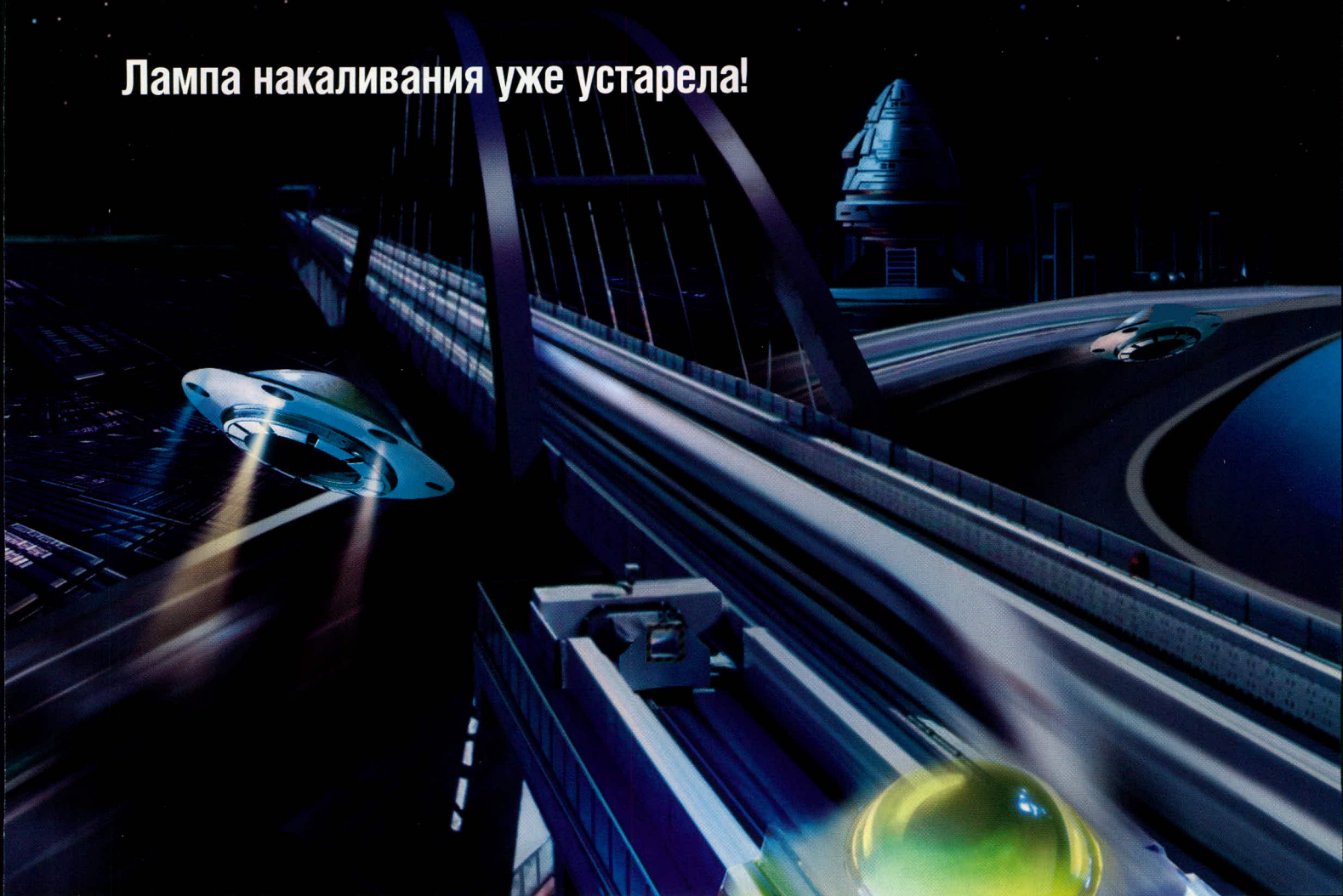
Чтобы продемонстрировать лидирующие позиции в сфере процессоростроения, Intel представила прототип компьютера с 80-ядерным процессором, получившего рабочее наименование Polaris PC, с пиковой вычислительной способностью порядка 2 терафлоп.

Процессор Polaris был разработан в рамках программы Tera-Scale Computing Research, а первую подложку с 80-ядерными процессорами генеральный директор Intel Пол Отеллини продемонстрировал в конце сентября 2006 г.

Прототип компьютера Polaris PC был разработан компанией Foxconn Electronics, присоединившейся к упомянутой программе около года назад.

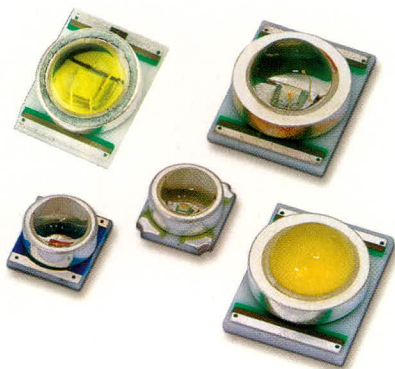
digitimes.com

Лампа накаливания уже устарела!



XLamp

Самые высокоэффективные и надежные полупроводниковые лампы Cree® XLamp™



- Весь спектр цветов: от ультрафиолета до оттенков белого
- Рекордно низкое тепловое сопротивление **8°C/Вт** (серии XR7090, XR-E7090)
- Температура кристалла **-60...+145°C** (серии XR7090, XR-E7090)
- Светоотдача до **90 Лм/Вт** (серия **XR-E7090**, белый цвет)
- Уникальный металлокерамический корпус для поверхностного монтажа: низкая себестоимость при серийном производстве
- Первичная оптика из кварцевого стекла
- Бессвинцовая технология
- Бюджетные решения для систем освещения (серия XR-C7090, 60 Лм/Вт при 350 мА)
- Высокоэффективные решения для полноцветной RGB-подсветки (серия XL4550)
- Уникальная технология кристаллов InGaN SiC®: деградация за **30 000 часов** менее **3% при +85°C!**

CREE LIGHTING

ПРОСОФТ – официальный дистрибьютор компании Cree

PROSOFT®

ПРОСОФТ – АКТИВНЫЙ КОМПОНЕНТ ВАШЕГО БИЗНЕСА

Телефон: (495) 232-2522 • E-mail: xlamp@cree.ru • Web: www.xlight.ru

Средства разработки программного обеспечения для встраиваемых 32-разрядных систем

(часть 2)

Любовь Самойлова, Вячеслав Краснов (Москва)

В статье рассмотрены наиболее часто применяемые интегрированные средства разработки программного обеспечения различных производителей и проведено их объективное сравнение с точки зрения функциональной полноты и простоты использования.

Средства разработки и отладки программ для микроконтроллеров семейства ARM представлены несколькими интегрированными средами известных фирм. Мы рассмотрим следующие:

- IAR Embedded Workbench фирмы IAR Systems (<http://www.iar.com>);
- uVision фирмы Keil Elektronik/Keil Software (<http://www.keil.com>);
- Embedded Development Environment фирмы Tasking (<http://www.tasking.com>);
- RIDE фирмы Raisonance (<http://www.raisonance.com>);
- Rowley CrossWorks фирмы Rowley Associates (<http://www.rowley.co.uk>);
- CodeMaster-ARM российской фирмы «Фитон» (<http://www.phyton.ru>, <http://www.phyton.com>).

Все перечисленные интегрированные среды предоставляют развитые средства разработки и отладки программ для микроконтроллеров семейства ARM. Файлы исходных текстов организуются в виде проектов. Каждый проект может иметь собственные настройки для компилятора, ассемблера и линкера. Все оболочки поддерживают отладку разрабатываемых программ с помощью встроенных симуляторов (программно-логических моделей микроконтроллеров), а также с помощью популярных аппаратных JTAG-отладчиков.

Для создания программ все интегрированные среды, кроме IAR Embedded Workbench, могут использовать бесплатный пакет кросс-средств GNU C Compiler for ARM. Фирма Tasking предоставляет собственные библиотеки стандартных функций Си, которые оптимизированы по разме-

ру кода и быстродействию в сравнении с библиотеками, входящими в состав пакета GNU ARM.

Настройка использования средств GNU ARM представляет определённые трудности, даже если сам GNU ARM поставляется вместе с пакетом отладочных средств. Это связано с тем, что изначально средства GNU были разработаны под ОС Linux.

Интегрированная среда для разработки и отладки программ должна удовлетворять основным требованиям, которые предъявляются к программным продуктам подобного типа с учётом особенностей разработки и отладки программ для микроконтроллеров семейства ARM. Рассмотрим эти требования и особенности.

Простота в освоении и удобство пользования

Практически каждый разработчик программ имел дело с той или иной интегрированной средой для разработки программ (оболочкой). Формальных требований и стандартов, предъявляемых к структуре, внешнему виду, группировке команд в меню интегрированных сред, нет. Однако существуют определённые неформальные правила организации функций оболочек, которых стараются придерживаться все производители. Это помогает пользователям в освоении оболочки. При наличии достаточного количества конкурирующих продуктов на рынке интегрированных сред простота использования может иметь решающее значение при выборе оболочки.

Сравниваемые интегрированные среды вполне удобны. Все основные

команды меню находятся на привычных местах, «поведение» оболочек предсказуемо. Исключение составляет оболочка Rowley CrossWorks, построенная на графической платформе Eclipse. Rowley CrossWorks перегружена графическими элементами, управление проектами слишком сложно и непрозрачно, названия многих команд меню отличаются от общепринятых, настроить горячие клавиши нельзя, локальных меню окон горячих клавиш нет вовсе.

Интегрированная среда должна быть снабжена удобной системой справки. Здесь отметим оболочку CodeMaster-ARM, которая имеет русифицированный (по желанию) интерфейс и снабжена документацией на русском языке.

Удобство настройки опций кросс-средств

Кросс-средства – это средства, с помощью которых исходный текст программы преобразуется в её исполняемый код, пригодный для «зашивки» в память микроконтроллера. К кросс-средствам относятся компиляторы языков высокого уровня (в основном используется язык Си), трансляторы с ассемблера, линкеры, библиотекарки и дополнительные утилиты, например, преобразователи выходных форматов.

Интегрированная среда может поддерживать разработку и отладку программ с использованием нескольких пакетов кросс-средств различных производителей. Если в одной и той же оболочке можно работать с несколькими пакетами кросс-средств, то это является безусловным преимуществом данной интегрированной среды, потому что при переключении с одних кросс-средств на другие не нужно изучать новую оболочку. Посмотрим, какие кросс-средства поддерживают сравниваемые оболочки. IAR Systems и

Tasking поддерживают только собственные компиляторы, которые поставляются вместе с оболочкой. Keil поддерживает три компилятора: собственный CARM, компилятор RealView и GNU ARM. Фирмы Raisonance и Rowley поддерживают только GNU ARM. CodeMaster-ARM фирмы «Фитон» поддерживает собственный компилятор CMC-ARM, компилятор GNU ARM, компилятор IAR Systems, а также может отлаживать файлы программ, созданных Keil CARM. Отметим также, что CodeMaster-ARM – единственная оболочка, которая способна отлаживать программы, созданные в других интегрированных средах, т.е. CodeMaster-ARM может использоваться только как отладчик. В остальных оболочках этого сделать нельзя, т.е. отлаживать можно только программу, созданную с помощью проекта в этой же самой среде.

Обычно кросс-средства имеют большое количество опций, которые управляют процессом трансляции (компиляции) программы. При запуске программы-компилятора опции указываются в её командной строке. Хорошая оболочка должна избавлять пользователя от изучения опций кросс-средств, предоставляя для их настройки удобные графические диалоги. У всех оболочек, кроме Rowley, с этим всё в порядке. В оболочке Rowley графического интерфейса к опциям кросс-средств нет – все настройки задаются опциями командной строки кросс-средств. Отметим удобную графическую систему для инициализации периферийных устройств, реализованную в Keil uVision.

УПРАВЛЕНИЕ РАСПРЕДЕЛЕНИЕМ ПАМЯТИ ЦЕЛЕВОГО МИКРОКОНТРОЛЛЕРА

Многие микроконтроллеры семейства ARM допускают подключение микросхем внешней памяти. Эти микросхемы могут быть различных типов: ОЗУ, ПЗУ и флэш (перезаписываемое ПЗУ), причём одновременно к микроконтроллеру может быть подключено несколько микросхем разных типов в разных адресных пространствах. Среда разработки должна предоставлять средства управления размещением кода программы и её данных в различных участках памяти с учётом типов

подключенных микросхем. Из перечисленных оболочек только Keil и CodeMaster-ARM имеют графические средства, которые позволяют наглядно указывать, по каким адресам какие сегменты программы и данных требуется разместить. У Keil возможности распределения памяти ограничены. В этой среде можно распределять ограниченное количество участков памяти, управлять размещением сегментов напрямую нельзя. Наиболее наглядный и гибкий интерфейс для распределения памяти имеет CodeMaster-ARM. У остальных оболочек, кроме Keil и CodeMaster-ARM, задавать распределение памяти можно только вручную, редактируя текстовый файл с описаниями областей памяти.

ОТЛАДКА ПРОГРАММ

Так как программы для ARM могут занимать значительный объём, оболочка должна минимизировать время, необходимое для загрузки программы в память микроконтроллера через JTAG-интерфейс. Поскольку флэш-память имеет ограниченное количество циклов стирания/записи, желательно при загрузке программы стирать и программировать только те участки памяти, которые изменились с момента предыдущей загрузки, а не стирать всю память целиком. Наиболее интеллектуальная загрузка программы реализована в отладчике оболочки CodeMaster-ARM, который осуществляет мониторинг изменений в участках памяти, программируемых во флэш. К тому же пакет кросс-средств CMC-ARM автоматически оптимизирует размещение сегментов программы таким образом, чтобы неизменяемые участки программы, т.е. библиотеки, располагались линкером до программных модулей, код которых часто модифицируется при разработке программы.

JTAG-интерфейс, предоставляемый микроконтроллерами семейства ARM, имеет всего две точки останова по коду. Отладка программы всего с двумя точками останова может представлять определённые трудности. Многие разработчики на этапе отладки размещают код программы в ОЗУ, что заодно снимает проблемы, связанные с программированием флэш-памяти. Отладчик оболочки должен поддерживать

установку дополнительных точек останова в ОЗУ. Это делается подменной машинной инструкцией по адресу точки останова специальным кодом, вызывающим останов программы.

Бывают всё же случаи, когда код программы разместить в ОЗУ нельзя, а двух точек останова для отладки недостаточно. Продвинутые отладчики, например CodeMaster-ARM, позволяют устанавливать точки останова во флэш-памяти, программируя их до загрузки программы во флэш-память.

Отладчик оболочки должен правильно и подробно отображать состояние ресурсов микроконтроллера и объектов, определённых в исходном тексте отлаживаемой программы. Как правило, для разработки программ используется язык Си, в котором можно манипулировать сложными объектами, например, структурами и указателями. В этом плане все сравниваемые оболочки предоставляют адекватный сервис, за исключением Rowley, для которой не было возможности проверить функции отладки, так как не удалось откомпилировать проект.

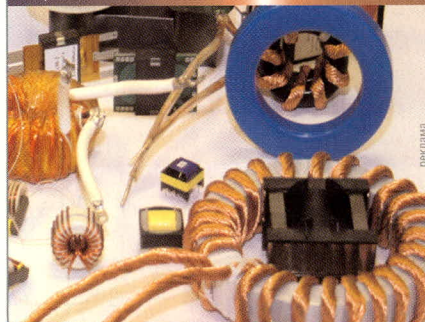


Северо-Западная лаборатория

Генеральный представитель Epcos
по ферритам в России и СНГ

ЗАО «Лэпкос» СЗЛ предлагает со склада:

- Ферритовые сердечники Epcos и ЗАО НПФ Феррокерам
 - Недорогие материалы с высокой индукцией насыщения (MPP, Kool M, High Flux, Iron Powder Magnetics и Micrometals)
 - Наномангнитные материалы для мощной силовой электроники
 - Намоточные каркасы, скобы, материалы для намотки
 - Пассивные компоненты фирмы Epcos
- ЗАО «СЗЛ» является крупнейшим в СНГ изготовителем трансформаторов и дросселей мощностью от 1 Вт до 150 Квт, работающих в диапазоне от 10 Гц до 150 МГц для силовой электроники, светотехники, телекоммуникаций. Осуществляется разработка и изготовление трансформаторов по документации или ТЗ Потребителя.



т/ф (812) 369-11-54, (812) 369-51-80
e-mail: epcos@ferrite.ru http://www.ferrite.ru

JTAG-тестирование

(часть 4)

Иосиф Каршенбойм (Санкт-Петербург)

Завершаем публикацию статьи, посвященной тестированию плат и устройств по интерфейсу JTAG. В этой части приводится описание шин и сигналов на выводах микроконтроллера Black Fin и практических методов тестирования.

ОПИСАНИЕ ШИН И СИГНАЛОВ НА ВЫВОДАХ МИКРОКОНТРОЛЛЕРА BLACKFIN

Рассмотрим назначение выводов микроконтроллера BlackFin BF537. Как видно из табл. 8, микропроцессор имеет множество шин и одиночных выводов, которым соответствуют буферы с различной нагрузочной способностью. Более подробную информацию об этом можно получить непосредственно из технического описания на микроконтроллер. Разумеется, при составлении тестов необходимо учитывать нагрузочную способность выходных буферов микросхем.

Разделим выводы микроконтроллера на несколько групп: «только вход», «только выход», «выход с третьим состоянием», «двунаправленная шина».

Сначала посмотрим, как описываются выводы «только вход». Фрагмент файла – ячейка 227 – описывает часть регистра, работающую с выводом сигнала TEST:

```
--num cell port function safe
(ccell disval rs1t )
" 227 ( BC_2 , TEST,
      input , X ) , < &
```

Вывод сигнала TEST однонаправленный и используется только как вход. Для осуществления этой функции задействована ячейка BC_2. Безопасное состояние вывода не определено – «X».

Посмотрим, как описываются выводы сигналов «только выход». Для примера возьмём два сигнала: BG_V и BGN_V. Фрагмент файла, приведённый ниже, описывает часть регистра, работающего с сигналами BG_V и BGN_V, а именно ячейки 0 и 1:

```
--num cell port function safe
(ccell disval rs1t)
```

```
" 0 ( BC_1 , BG_V , output2
    , X ) , < &
" 1 ( BC_1 , BGN_V , output2
    , X ) , < &
```

Выводы сигналов BG_V и BGN_V однонаправленные и используются только как выход. Для таких сигналов задействованы ячейки BC_1. Безопасное состояние выводов не определено – «X». Поскольку для выводов BG_V и BGN_V не предусмотрено третьего состояния, то и никакие дополнительные сигналы управления им не требуются.

Теперь посмотрим, как описываются «выходы с третьим состоянием». Для примера возьмём фрагмент файла – ячейки 15...17, где описывается часть регистра, работающего с шиной адреса:

```
--num cell port function safe
(ccell disval rs1t)
" 15 ( BC_1 , ADDR(6) ,
      output3 , X , 17 ,
          0 , Z ) , < &
" 16 ( BC_1 , ADDR(5) ,
      output3 , X , 17 ,
          0 , Z ) , < &
" 17 ( BC_1 , * ,
      control , 0 ) , < &
```

Шина адреса однонаправленная и переключается в третье состояние одним сигналом управления. Для этого в каждом разряде регистра используются ячейки только одного типа – BC_1. Ячейки типа BC_1 используют и для вывода сигналов адреса, и для записи в регистр сигнала управления. Ячейки BC_1 выполняют функцию output3 или функцию control. Для шины адреса безопасное состояние не определено – «X». Для ячеек типа BC_1 сигнал управления берётся из ячейки, соответствующей 17-му биту реги-

стра, причём состояние запрета определяет значение «0», и вся шина при этом будет находиться в состоянии «Z».

Теперь рассмотрим, как описаны ячейки, работающие с портами флагов. Порты флагов имеют независимое управление, причём выводы этих портов могут работать как на выход, так и на вход:

```
--num cell port function safe
(ccell disval rs1t)
" 47 ( BC_2 , , input
    , X ) , < &
" 48 ( BC_1 , PJ2 , output3
    , X , 49 , 0
        , Z ) , < &
" 49 ( BC_1 , * , control
    , 0 ) , < &
```

Этот фрагмент файла представляет собой ячейки 47...49, где описывается часть регистра, работающего с портом PJ. Порт PJ – набор двунаправленных сигналов с независимым для каждого сигнала управлением. Каждый вывод порта использует ячейки BC_1 и BC_2. Ячейки типа BC_1 используются для вывода сигналов данных и для записи в регистр сигнала управления. Ячейки типа BC_2 используются для ввода сигналов данных. Соответственно ячейки BC_1 выполняют функцию output3 или функцию control, а ячейки BC_2 – функцию input. Для вывода флага PJ2 безопасное состояние не определено – «X», и выходная ячейка типа BC_1 берёт сигнал управления из ячейки, соответствующей 49-му биту регистра, причём состоянию запрета соответствует значение «0», а сигнал PJ2 при этом будет находиться в состоянии «Z».

Аналогично описывается и двунаправленная шина данных. Фрагмент файла – ячейки 256...260, описывает часть регистра, работающую с шиной данных:

```
--num cell port function safe
(ccell disval rs1t )
" 256 ( BC_2 , DATA(1) ,
```

```
input , X ) , « &
" 257 ( BC_1 , DATA(1) ,
output3 , X , 260 ,
0 , Z ) , « &
" 258 ( BC_2 , DATA(0) ,
input , X ) , « &
" 259 ( BC_1 , DATA(0) ,
output3 , X , 260 ,
0 , Z ) , « &
" 260 ( BC_1 , * , control
, 0 ) «;
```

Шина данных двунаправленная и, в отличие от порта флагов с независимым управлением, переключается только одним сигналом управления. Для передачи информации в обе стороны в каждом разряде данных используются ячейки BC_1 и BC_2. Ячейки типа BC_1 используют для вывода сигналов данных и для записи в регистр сигнала управления. Ячейки типа BC_2 используют для ввода сигналов данных. Соответственно ячейки BC_1 выполняют функцию output3 или функцию control, а ячейки BC_2 – функцию input. Для шины данных безопасное состояние не определено – «X». Для ячеек типа BC_1 сигнал управления берётся из ячейки, соответствующей 260-му биту регистра, причём состоянию запрета соответствует значение «0», и вся шина при этом будет находиться в состоянии «Z».

Подведём итог

Вот общее количество выводов: 122 входа и выхода + 2 выхода типа buffer и 56 выводов linkage. Из 122 выводов 5 выводов заняты непосредственно портом JTAG.

Из 124 выводов, которые могут быть протестированы, имеем: 35 выходов + 2 выхода типа buffer + 8 входов + 94 двунаправленных вывода.

Непосредственно не проверяются следующие выводы: EMU_B, CLKIN, XTAL, CLKBUF, RTXI, RTXО.

Также не проверяются выводы, по которым подводится питание: VDD_INT, VDD_EXT, VDD_RTC, VROUT и GND.

Тестами можно охватить только 68% от общего числа выводов. Остальные выводы могут быть оценены только косвенно.

И ещё раз JTAG в картинках

Параграф о картинках уже был в предыдущих частях статьи, так зачем же мы к нему возвращаемся снова? Возвращаемся потому, что теперь мы

Таблица 8. Описание шин и выводов микроконтроллера BlackFin BF537

Название	Тип	Описание
Интерфейс памяти		
ADDR19-1	O	Шина адреса для асинхронного доступа к памяти
DATA15-0	I/O	Шина данных для синхронного и асинхронного доступа к памяти
ABE1-0/SDQM1-0	O	Сигналы разрешения выбора байта для синхронного и асинхронного доступа к памяти
BR	I	Запрос на доступ к шине
BG	O	Разрешение на доступ к шине
BGH	O	Выход ожидания разрешения на доступ к шине
Сигналы управления асинхронной памяти - Asynchronous Memory Control		
AMS3-0	O	Сигнал выбора банка
ARDY	I	Аппаратный сигнал готовности
AOE	O	Разрешение выхода
ARE	O	Разрешение чтения
AWE	O	Разрешение записи
Сигналы управления синхронной памяти - Synchronous Memory Control		
SRAS	O	Строб адреса строки
SCAS	O	Строб адреса колонки
SWE	O	Разрешение записи
SCKE	O	Разрешение тактовой частоты
CLKOUT	O	Выход тактовой частоты
SA10	O	Вывод A10
SMS	O	Выбор банка
Порты		
PF0..15	I/O	Port F: GPIO/UART1-0/Timer7-0/SPI/External DMA Request
PG0..15	I/O	Port G: GPIO/PPI/SPORT1
PH0..15	I/O	Port H: GPIO/10/100 Ethernet MAC (On ADSP-BF534, these pins are GPIO only)
PJ0..15	*	Port J: SPORT0/TWI/SPIselect/CAN
Часы реального времени - Real Time Clock		
RTXI	I	Вход резонатора для RTC
RTXO	O	Выход резонатора для RTC
Порт JTAG		
TCK	I	Синхрочастота JTAG
TDO	O	Выход данных JTAG
TDI	I	Вход данных JTAG
TMS	I	Сигнал Mode Select JTAG
TRST	I	Сигнал Reset JTAG
EMU	O	Сигнал Emulation Output
Тактирование		
CLKIN	I	Вход Clock/Crystal Input
XTAL	O	Выход Crystal Output
CLKBUF	O	Выход Buffered XTAL Output
Управление режимом работы		
RESET	I	Сброс
NMI	I	Немаскируемое прерывание
BMODE2-0	I	Сигналы управления загрузкой Boot Mode Strap 2-0
Регулятор напряжения		
VROUT0	O	Вход управления для FET
VROUT1	O	Вход управления для FET
Питание		
VDDEXT	P	Питание I/O
VDDINT	P	Внутреннее питание (регулируется от 2,25 В до 3,6 В)
VDDRTC	P	Питание для часов реального времени
GND	G	Земля

I – вход; O – выход; P – питание; G – земля

вооружены новыми знаниями и можем адекватно оценить предлагаемые «картинки». К настоящему моменту мы уже знаем, как работает узел TAP-контроллера, знаем, как именно микросхемы соединяются в

цепочку, знаем, куда и какой бит нужно посылать. А главное, мы знаем, какой эффект получим от автоматического тестирования платы, особенно если эта плата похожа на ту, которая показана на рис. 16.



Рис. 16. Фрагмент современной печатной платы с установленными на ней микросхемами

На рис. 17 приведён фрагмент программы Boundary Scan Coach, о которой мы уже упоминали. Напомним, что функциональная часть микросхемы – Core Logic – показана коричневым цветом. JTAG-регистры – зелёным цветом, выводы микросхемы – серым цветом. Голубым цветом выделен контроллер JTAG-порта. Красным цветом показан путь прохождения данных по регистру граничного сканирования. Программа Boundary Scan Coach содержит несколько частей. Наибольший интерес, с точки зрения граничного сканирования, представляет собой часть № 3 этой программы, в кото-

рой описано, как производится сканирование устройства, содержащего две микросхемы. По цепи JTAG-тестирования микросхемы включены последовательно. Данные сдвигаются из порта в первую микросхему, а из неё во вторую. Выходные данные принимаются портом. Пользователь может задавать различные режимы работы выводов, например, подавать воздействия с одной микросхемы на другую. В режиме линзы можно посмотреть состояние выводов так, как это показано на рис. 15, при описании структуры ячеек ввода-вывода.

От теории к практике!

Теперь остаётся удостовериться, что всё написанное в этой статье применимо на практике. Как проверить, что считывается на входе микросхемы? Как выдать необходимые сигналы на вывод? Для этого предлагается три пути.

Путь № 1, или классический, – быстро садимся и учим языки, при помощи которых описываются тестовые воздействия. Потом неизвестно где находим РАБОТАЮЩИЕ компиляторы этих языков. Далее понятно... С этим вариантом читатель сможет разобраться сам, если имеет достаточно времени и сил.

Путь № 2, или обычный, – с помощью C++ пишем тест сами. Например, так, как в [11, 12] или в [18].

Путь № 3, необычный, – ищем в Сети и устанавливаем Universal Scan [5].

Рассмотрим два последних пути решения проблемы. А какой из них лучше – об этом судить читателю.

Путь № 2 – ПЕРЕХОДИМ К СИМУЛЯЦИИ АППАРАТНОЙ ЧАСТИ ИНТЕРФЕЙСА JTAG

До сих пор большинство программных инструментов и обучающих программ были ориентированы только на обучение. Давайте ознакомимся с теми программами и инструментами, которые могут пригодиться в реальной жизни при работе с «железом».

Поскольку уже описана работа с интерфейсом, познакомимся с ещё одним программным симулятором его работы. На этот раз речь пойдёт о симуляции работы порта и TAP-контроллера в ModelSim. Читатели могут взять этот проект с сайта автора, на странице, посвящённой JTAG. Для чего делался этот проект? Дело в том, что изготовители FPGA постепенно «разворачиваются» навстречу пользователю микросхем. В острой конкурентной борьбе им приходится давать потребителю всё больше и больше сервиса. Когда-то встроенный логический анализатор был делом любителей-одиночек. Но, как выяснилось, инструмент этот оказался очень полезен, особенно для больших проектов и для «толстых» FPGA. А чего только не сделает производитель, чтобы продать микросхем побольше да подороже. Вот так и пришлось изготовителям вводить в состав своих программных средств встроенные логические анализаторы. Естественно, что для связи с анализаторами был применён интерфейс JTAG. Но при этом всё, что было с ним связано, поставлялось в зашифрованном виде. Однако «почин был подхвачен» массами пользователей (абсолютно ясно, как именно был подхвачен), и фирмам пришлось раскрыть адресацию и сами компоненты, при помощи которых встроенные логические анализаторы подключались к проекту пользователя. Так что теперь у разработчиков FPGA есть возможность использовать знания технологии JTAG и для отладки проектов в FPGA. Ну, а там, где FPGA, там и ModelSim, и Verilog.

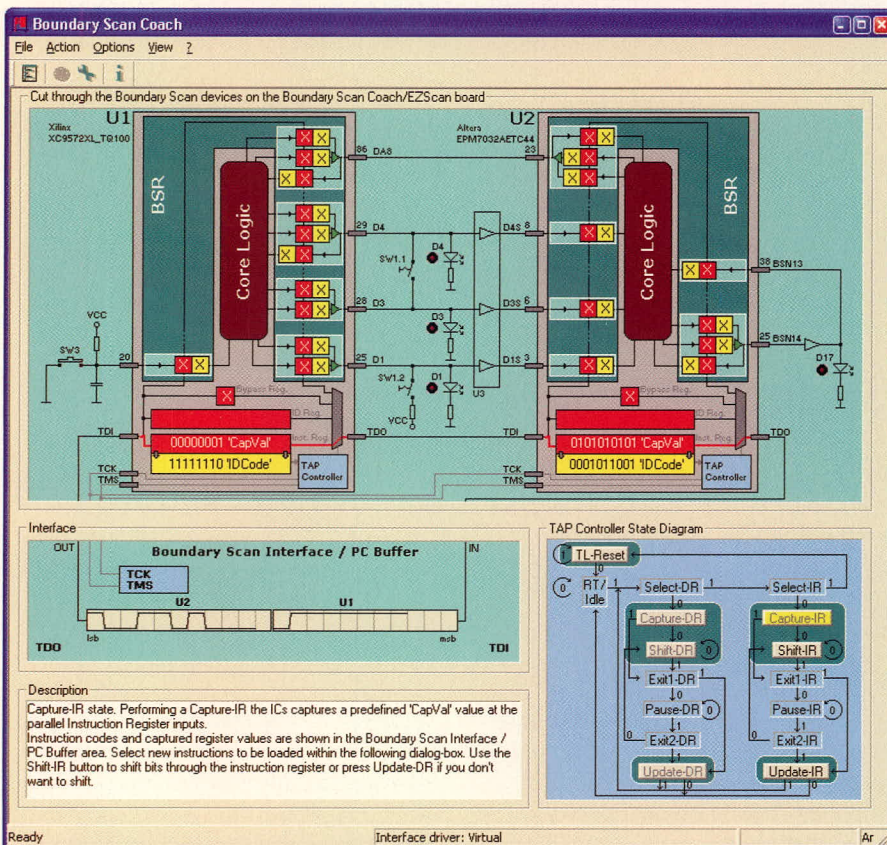


Рис. 17. Окно программы Boundary Scan Coach. Часть № 3

Итак, представим себе хост, JTAG-порт и связь с проектом пользователя.

Теперь рассмотрим рис. 18, где эти компоненты представлены в виде блок-схем.

Начнём с самого понятного разработчику – а именно с той части, которая выделена голубым цветом и представляет собой логику отладочного блока пользователя, которая находится в FPGA, там же, где и проект пользователя. С этой логикой в общих чертах всё понятно. Её можно «написать» самому, отсимулировать и затем скомпилировать. А вот JTAG-порт – это аппаратная часть микросхемы, и детального описания его работы нет. Ну а хост – это немного программ и аппаратный LPT- или USB-порт. А ведь инженер-аппаратчик привык видеть всё в виде осциллограмм или их симуляций. Вот поэтому автор предлагает воспользоваться одним небольшим проектом, сделанным им для статьи. Это симуляция описанных выше устройств, выполненная в среде ModelSim и написанная на языке Verilog. Цель проекта – помочь читателю увидеть последовательность смены состояний автомата TAP-контроллера в виде осциллограмм.

Основой проекта является именно узел TAP-контроллера. Узел TAP-контроллера вместе с регистрами и другими дополнительными узлами образуют модель JTAG-порта. Для выдачи в JTAG-порт последовательности импульсов в проекте используется блок tester. Этот узел выполнен как task.

Чтобы сформировать последовательность импульсов, выдаваемых на узел tester, можно пройтись по диаграмме автомата в «ручном режиме». Кроме этого, для того чтобы упростить формирование последовательностей импульсов, выдаваемых узлом tester, можно воспользоваться ещё одним программным инструментом, также доступным на сайте автора. Этот программный инструмент как раз и является примером решения задачи путём № 2. Программа специально написана автором на ВСВ6 для учебных проектов к статьям. Исходные тексты программы доступны на сайте автора. Одним из результатов работы этого программного инструмента как раз и будут строки для тестовой последовательности импульсов.

Поскольку программа обучающая, то она работает по сокращённой диаграмме переходов. Для упрощения

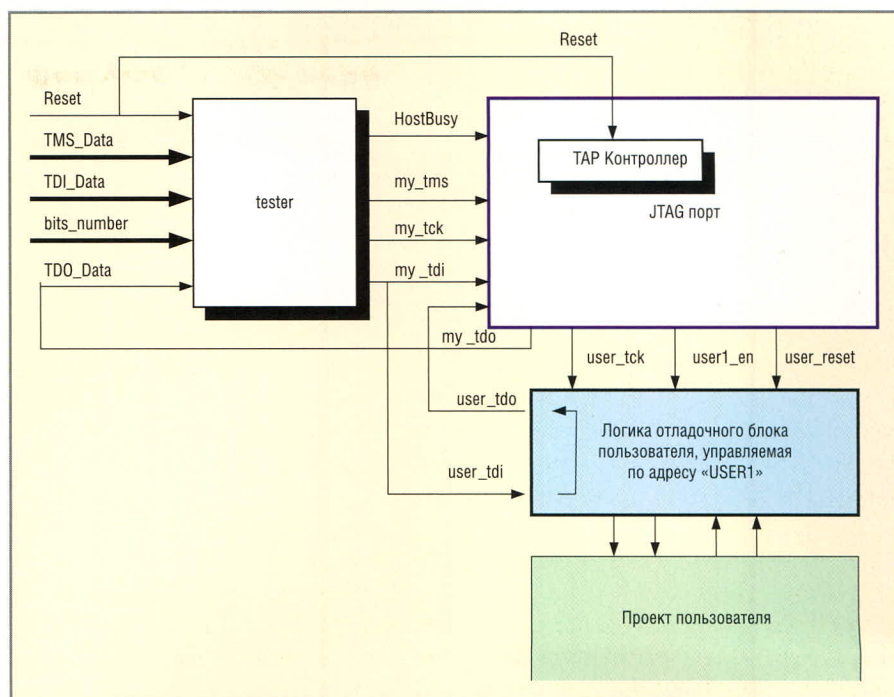


Рис. 18. Блок-схема проекта симуляции работы хоста, JTAG-порта и логики пользователя, выполненная в среде ModelSim и написанная на языке Verilog

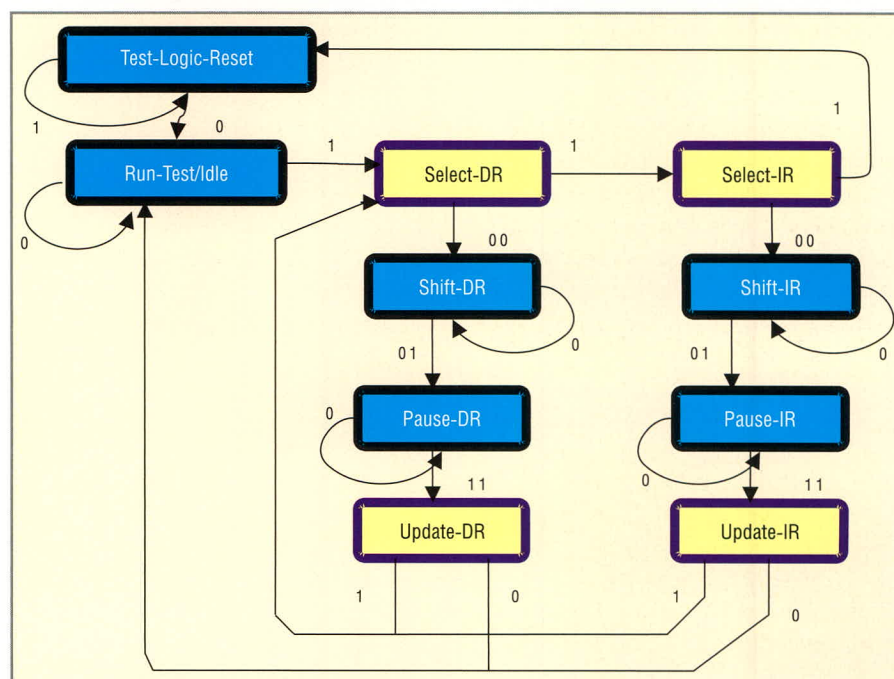


Рис. 19. Упрощённая диаграмма переходов

программы, работающей с JTAG-портом, было сокращено число состояний автомата, которыми можно пользоваться. Поэтому у нас есть возможность представить диаграмму более компактной. Такая диаграмма переходов будет выглядеть так, как показано на рис. 19. Именно этой диаграммой переходов и пользовался автор, когда разрабатывал свой программный инструмент для работы с JTAG-портом от LPT-порта хоста. Более подробно об этом можно прочитать в [18].

На рис. 20 показано основное окно программы, в котором пользователь формирует исполняемые тесты. В левой части окна расположена таблица, в которую помещаются команды и их параметры, а в правой части – органы управления, необходимые для работы с этим окном. Тесты можно исполнять как в шаговом, так и в автоматическом режиме – по таймеру. Для работы в автоматическом режиме предусмотрена возможность останова по точкам, задаваемым пользователем. Более подроб-

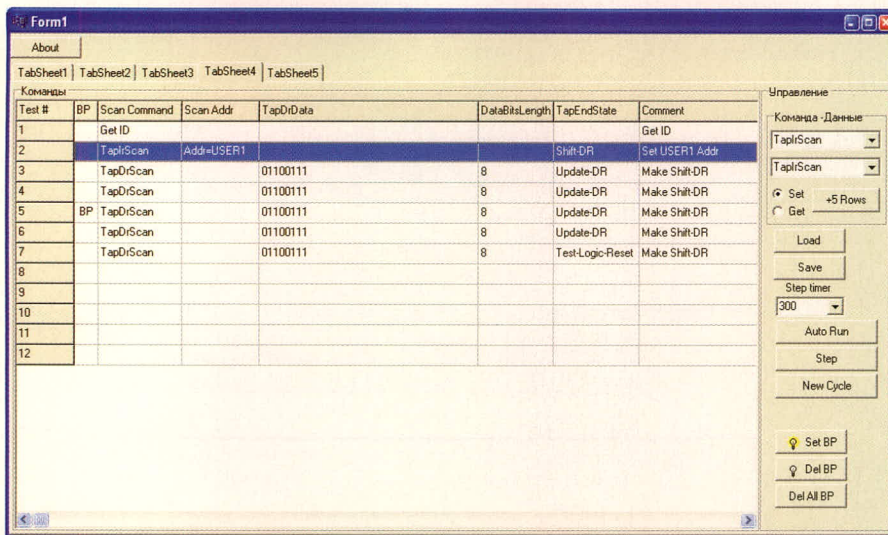


Рис. 20. Окно программы, в котором пользователь формирует исполняемые тесты

но об этой программе можно прочитать в [18].

Загрузим или сформируем тест чтения ID из микросхемы, установим мониторинг «на уровне битов» и запустим тест на выполнение. Трассировка теста распечатается в окне мониторинга даже при выключенном аппаратном адаптере.

Именно эту трассировку можно использовать при задании параметров для task. Ниже приведён фрагмент кода для симуляции режима чтения ID из микросхемы. Закомментированные строки, выделенные зелёным цветом, – это часть трассировки, взятая как результат работы программного инструмента. Строки, напечатанные чёрным цветом, – это то, что будет выполнять симулятор. Для упрощения чтение ID выполняется не за 32 такта, а только за восемь.

```
// Get ID
//переход в TAP_TLR, tms - 0x1F,
tdi - 0x00, N - 0x05
tester(32'h00FF, 32'h0000, 8,
wTDO, RST); - это сделано,
чтобы увидеть
сброс «user_reset»

//переход в TAP_SI, tms - 0x06,
tdi - 0x00, N - 0x05
tester(32'h0006, 32'h0000, 5,
wTDO, RST);

//запись команды в последнем
байте, tms - 0x00, tdi - 0x09,
N - 0x05, RdWriteTMS_SetTDI
strScanIn - 10101
tester(32'h0000, 32'h0009, 5,
wTDO, RST);
```

```
//запись последнего бита команды
и переход в TapEndState,
tms - 0x0B, tdi - 0x00,
N - 0x05, RdWriteTMS_SetTDI
strScanIn - 11111
tester(32'h000B, 32'h0000, 5,
wTDO, RST);

//cur_state - 0x03
//по переходам - 0x02
//String IrScan - 110101
//прием данных в байтах,
tms - 0x00, tdi - 0x00,
N - 0x08, RdWriteTMS_SetTDI
strScanIn - 00100111
tester(32'h0000, 32'h0000, 8,
wTDO, RST);
//прием данных в байтах,
tms - 0x00, tdi - 0x00,
N - 0x08, RdWriteTMS_SetTDI
strScanIn - 00000001
//прием данных в байтах,
tms - 0x00, tdi - 0x00,
N - 0x08, RdWriteTMS_SetTDI
strScanIn - 10000010
//прием данных в байтах,
tms - 0x00, tdi - 0x00,
N - 0x08, RdWriteTMS_SetTDI
strScanIn - 00000011
//прием последнего бита
и переход в TapEndState,
tms - 0x03, tdi - 0x00,
N - 0x03, RdWriteTMS_SetTDI
strScanIn - 110
tester(32'h0003, 32'h0000, 3,
wTDO, RST);
//cur_state - 0x08
//по переходам - 0x01
```

Теперь более подробно можно рассмотреть структуру самого JTAG-порта, блок-схема которого приведена на рис. 21.

В состав порта входят TAP-контроллер, регистр инструкций, декодер адреса инструкций, регистры Bypass, IDCODE, BoundaryScan и другие регистры. Также в состав порта входят два мультиплексора выходных данных и вспомогательный узел, формирующий сброс. На регистры и TAP-контроллер поступают тактовые импульсы TCK. Кроме того, на TAP-контроллер поступает сигнал TMS. Входные данные – TDI поступают на регистры.

Если сигнал TMS удерживать на входе более пяти тактов синхрос частоты, то TAP-контроллер выполнит сброс своей логики и будет находиться в состоянии Test Logic Reset. Импульсы TCK и сигнал TMS переведут TAP-контроллер в другие состояния. При получении инструкции в состоянии Shift_IR, TAP-контроллер пройдёт через состояние Update_IR. Во время сдвига инструкции из JTAG-порта с выхода TDO будут считываться данные. При этом полученная инструкция будет запомнена и её код поступит на дешифратор инструкций. Дешифратор инструкций выберет, какой из регистров коммутируется на выход TDO. В этот же регистр загрузятся данные при работе TAP-контроллера в состоянии Shift_DR. Результат работы симулятора при чтении ID представлен на рис. 22. Жёлтыми цифрами в верхней части рисунка обозначены периоды времени, соответствующие task'am, которые в вышеприведённом листинге показаны чёрным цветом. Жёлтым шрифтом на поле симуляции обозначены состояния TAP-контроллера. Последовательность переходов из одного состояния в другое показана жёлтыми стрелками.

Рассмотрим фазы сдвига инструкции, которые на рис. 22 показаны голубыми цифрами. Импульсы, отмеченные цифрами 1 – 5, соответствуют переходам в состояние Shift_IR. Само состояние Shift_IR обозначено цифрой 6. По окончании этого состояния автомат переходит через состояния 6 в состояние, обозначенные как 7 – 11 к состоянию Shift_DR. Состояние Shift_DR обозначено как состояние 12. По окончании работы с данными происходит переход автомата через состояние 13 в состояние Run_Test/Idle.

Голубой цифрой 14 показано формирование импульса СБРОС, кото-

рый вырабатывает счётчик числа импульсов на входе TMS. При достижении заданного уровня, а именно 5 импульсов, счётчик останавливается и выдаёт сигнал СБРОС. Этот сигнал используется логикой сопряжения JTAG-порта с логикой пользователя.

Голубой цифрой 15 показано формирование импульса разрешения выборки, который вырабатывается в соответствии с тем адресом, который был загружен в TAP-контроллер по команде IR-Scan. Часть этих сигналов также используются логикой сопряжения JTAG-порта с логикой пользователя.

Для упрощения симуляции предлагаемый автором проект не содержит «логики пользователя». К «логике пользователя» автор относит те сдвиговые регистры и регистры хранения информации, которые необходимы пользователю для работы его отладочного узла. Кроме собственно узлов, преобразующих последовательный интерфейс в параллельный, пользователю ещё понадобятся дополнительные узлы привязки асинхронного (с точки зрения системной синхрос частоты в проекте пользователя) проекта пользователя к системной синхрос частоте, на которой работает весь остальной проект пользователя.

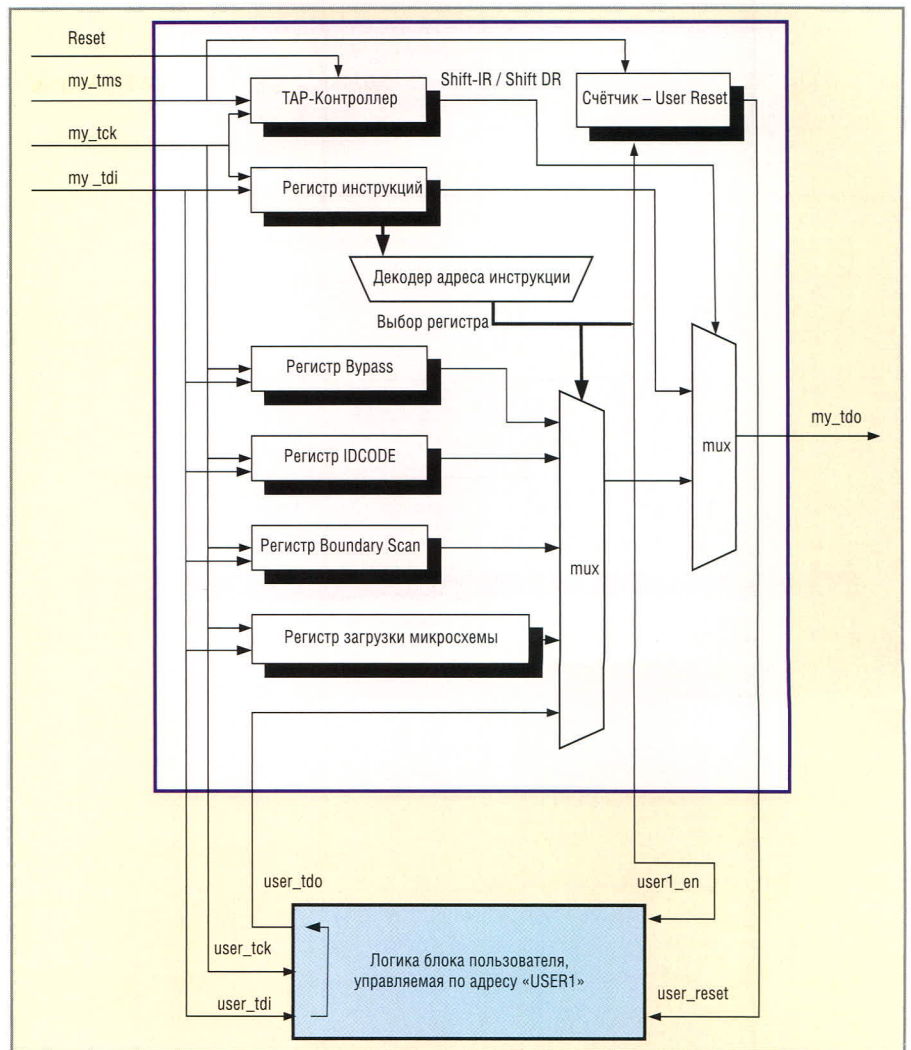


Рис. 21. Блок-схема JTAG-порта и логики пользователя

Ну и последнее, что необходимо сказать об этой диаграмме, так это то, что именно она и есть ответ на «JTAG-ребус», потому что на диаграмме «JTAG-ребус» также показано, как производилось чтение регистра ID из микросхемы.

Путь №3 – программа UNIVERSAL SCAN

Программа Universal Scan позволяет сканировать устройство пользователя, управлять выходами и читать информацию на входах, а также программировать Flash-память.

Чтобы начать тестировать плату при помощи программы Universal Scan, необходимо подключить байтбластер Altera или Xilinx к проверяемой плате и подать питание. Затем следует запустить программу Universal Scan (см. рис. 23), выбрать из меню тип байтбластера и установить его на поле. Далее нужно выбрать тип микросхемы и также установить символ микросхемы на поле. Если микросхем несколько, то последнее действие необходимо повто-

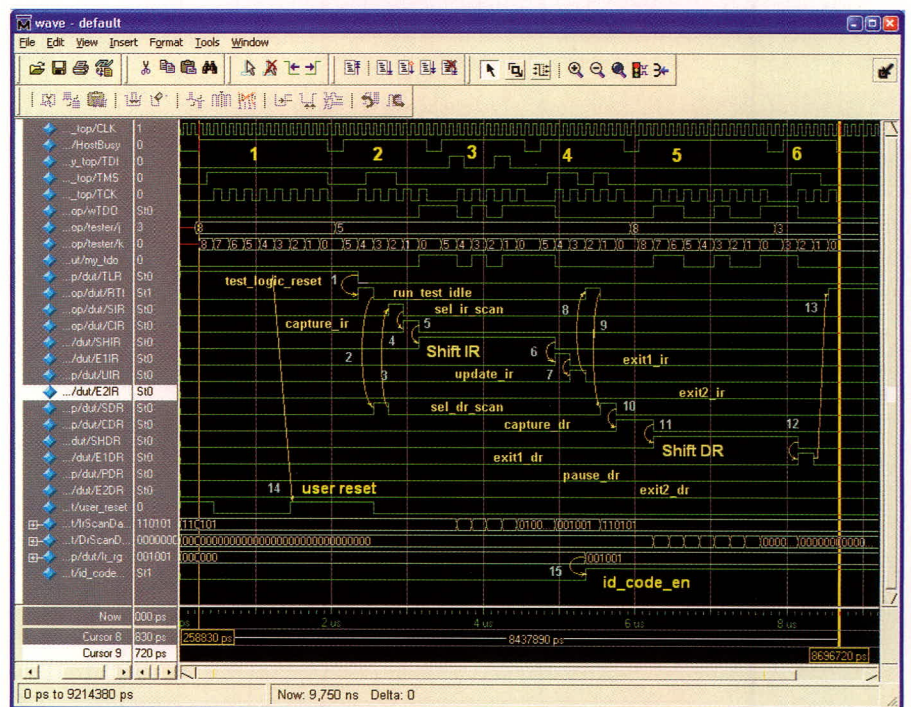


Рис. 22. Симуляция работы JTAG-порта при чтении ID из микросхемы

рить. После этого надо нажать кнопку «Сканировать». Вот, собственно, и всё. В результате сканирования выводы

микросхем раскрываются в соответствии с теми напряжениями, которые на них присутствуют.

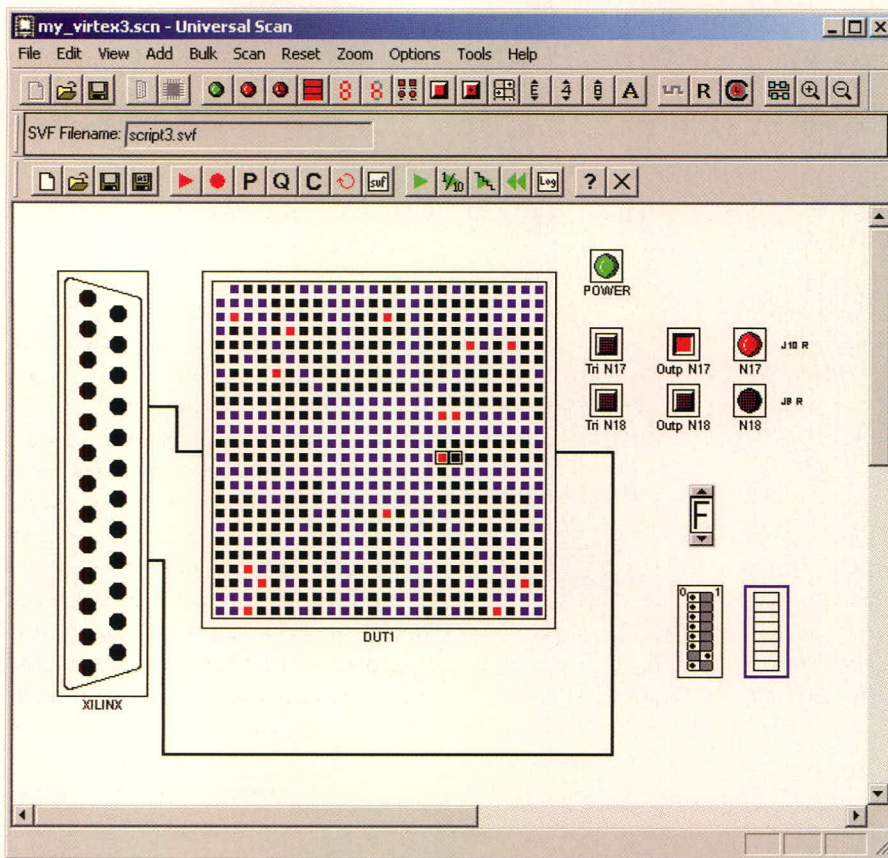


Рис. 23. Основное окно программы Universal Scan

Для указания типа микросхемы никаких особых усилий прикладывать не надо. Программа снабжена библиотекой микросхем, что позволяет иметь на рабочем поле графические символы, соответствующие выбранной микросхеме. Если же требуемой микросхемы в библиотеке нет, то программа предоставляет пользователю возможность самому сформировать необходимый символ, для чего просит предоставить ей BSDL-файл, на основе которого сама формирует изображение микросхемы. Далее начинается работа по проверке микросхемы или платы. Хотим задать воздействие на определённый вывод? Пожалуйста! Помечаем вывод маленьким квадратиком (если при-

смотреться, то он виден на рис. 23) и открываем окно управления выводом (см. рис. 24). В этом окне расположены органы для управления состоянием вывода.

Посмотрите внимательно на сигнал разрешения управлением выхода для XC2V250_FG456, сравните его с тем, что нарисовано в Datasheet на эту микросхему. Правда состоит в том, что работает цепь именно так, как показано в Universal Scan, а не в Datasheet. Почему? Да потому, что Universal Scan работает так, как написано в BSDL-файле.

Ну, а если мы хотим работать с несколькими выводами одновременно, то оказывается, гораздо проще ассоциировать с каждым из этих выводов виртуальный светодиод и виртуальные кнопки. Кнопки управляют сигналами разрешения выхода и самими сигналами выходов. Кроме того, есть наборы элементов управления – DIP-switch, цифровые переключатели и т.д. То же самое относится к элементам индикации. Кроме отдельных светодиодов, которые индицируют состояние напряжения питания или состояние одного вывода, существует множество различных виртуальных элементов индикации, таких как DIP-Leds, цифровые знакоместа и пр.

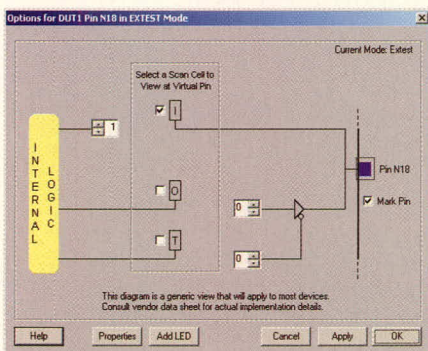


Рис. 24. Окно управления выводом программы Universal Scan

Каждый индикатор может отображать данные в прямом или в инверсном коде. Также и каждая кнопка может быть запрограммирована на работу в инверсном коде. Все эти действия выполняются с верхней панелью функциональных кнопок, показанной на рис. 24. А вот нижняя панель функциональных кнопок предназначена для того, чтобы записывать действия оператора в файл SVF-формата, а затем воспроизводить эти действия в автоматическом режиме. Тут есть полная свобода для обучению языку SVF, так как всё происходит очень наглядно: записал – отредактировал – воспроизвёл. Набор тестовых воздействий постепенно превращается в тест проверки устройства.

Программа также снабжена видеороликами, где показываются действия оператора по тестированию и программированию проверяемых изделий.

Недостатком же программы Universal Scan можно назвать лишь её «нежелание» работать по интерфейсу USB с фирменными аппаратными адаптерами Altera и Xilinx. А тот USB-адаптер, который предлагается, – в свою очередь, несовместим с программными инструментами фирм Altera и Xilinx.

ОЧЕНЬ КОРОТКО О SVF-ФАЙЛАХ

Формат SVF-файлов дан в [15]. Этот документ описывает синтаксис для файлов SVF – последовательного векторного формата. Файлы SVF используются для описания операций высокого уровня, выполняемых по интерфейсу IEEE 1149.1. Формат SVF был разработан для того, чтобы упростить использование последовательных векторов при испытаниях тестируемого изделия. Файл SVF определён как ASCII-файл, который состоит из набора инструкций SVF и данных. Максимальное число символов в строке – не более 256, хотя одна инструкция SVF может быть записана на нескольких строках. Каждая инструкция состоит из команды и связанных параметров и завершается разделителем в виде точки с запятой. Текст в SVF-файле не зависит от регистра. Комментарии могут быть вставлены в SVF-файл после восклицательного знака – «!» или пары наклонных черт вправо – «//». Символы «//» или

«!» указывают на то, что остаток строки является комментарием.

Данные, передаваемые или принимаемые по командам тестирования в пределах каждой инструкции, записываются в шестнадцатеричном виде и всегда заключаются в круглые скобки. Естественно, что данные, предназначенные для сдвига, не всегда имеют длину, кратную 16. В таком случае нули в старшем байте при анализе отбрасываются. Порядок записи бит «старший-младший» соответствует соглашениям стандарта IEEE 1149.1.

Вот фрагмент сканирования микросхемы, выполненный на основании BSDL-файла – xc2v250.bsd:

```
!----STEP 2:
SIR 6 TDI(00) TDO(11) MASK(03);
SDR 732 TDI(969B6D...DB6DB6D);
SDR 732 TDI(969B6D0...6DB6DB6D)
TDO(169B6D02B6DB...DB6DB6D)
MASK(80000...080000...0000000);
```

Что делается в этом фрагменте кода? Заремаркированная строка сообщает, что выполняется шаг тестирования № 2. Далее указывается, что длина регистра инструкций составляет 6 бит, потом описывается то, что требуется выдать по проводу TDI, то, что необходимо получить на TDO, и маска, по которой проверяются полученные данные.

Следующая строка указывает, что длина регистра данных (т.е. то, что мы называли регистром граничного сканирования) составляет 732 бит. Следующие строки описывают, как и в случае сканирования команды, то, что требуется выдать по проводу TDI, то, что необходимо получить на TDO, и маску, по которой проверяем полученные данные.

SVF-файл позволяет задавать воздействия на исследуемую микросхему, проверять результаты воздействия, формировать воздействия для группы микросхем, объединённых в цепочку, задавать временные параметры сканирования, а также программировать порты ввода-вывода.

Ну а где же «ПРО ТЕСТИРОВАНИЕ»?

И вот здесь читатель вправе спросить: «А где же про само тестирование, ведь именно это и было обещано?» И так, как же выполняются процедуры тестирования? Давайте рассмотрим их шаг за шагом. Рас-

сматривать тестирование будем на примере проверки смонтированной печатной платы (PCB). Пусть на этой плате есть несколько микросхем, объединённых в цепочку, которые имеют порт JTAG.

Первым делом определим порядок подключения микросхем в цепи JTAG, затем, используя BSDL-файлы этих микросхем, произведём следующие действия:

- выделим коды команд тестирования, длину кода для каждой команды и т.д. так, как это описывалось выше;
- выделим фрагменты, соответствующие регистру граничного сканирования и рассмотрим безопасный режим для каждой из микросхем;
- определим биты управления и их состояние для безопасного режима;
- определим, на каких входах этих микросхем входные данные будут определены и в каких ситуациях.

Более подробно рассмотрим последнее действие. Если мы тестируем изделие, значит, на него есть документация в виде схемы, а ещё лучше, если на плату есть проект для программного инструмента типа PCAD, Power PCB или любого другого аналогичного инструмента. При помощи программного инструмента можно легко выделить список связей из проекта. По списку связей и по схеме необходимо определить, какая информация будет читаться на входах микросхем. Если вход микросхемы подключен к како-

му-либо выходу, не имеющему третьего состояния, то на таком входе всегда должна присутствовать информация, определяемая состоянием указанного выхода. Если информация на тестируемом входе отличается от той, которую мы ожидаем увидеть, то это значит, что в цепи имеет место неисправность. То же самое относится к случаю, когда один выход тестируемой микросхемы, не имеющий третьего состояния, подключен к одному или к нескольким входам другой микросхемы. Состояние выхода будет всегда определено, и, соответственно, будет определено состояние тех входов, на которые поступает сигнал с тестируемого выхода. То же самое можно сказать и о входах, на которые поданы сигналы опорного напряжения или которые соединены с общим проводом. Это будет справедливо и для выходов, которые через какие-либо компоненты подключены к «плюсу» или «на землю». Примером таких цепей могут служить выходы с подключенными к ним светодиодами.

Совсем другое дело, если в проверяемой цепи присутствует выход микросхемы, имеющей третье состояние. Сначала рассмотрим цепь, состоящую из выходного буфера с третьим состоянием и подключенными к ней одним или несколькими входами. Если к этой цепи не подключены опорные резисторы, то будет иметь место следующее:

Проектирование корпусов для интегральных микросхем и полупроводниковых приборов, производство продукции из электротехнической керамики.



МВЕ
ЦЕНТР ВОЕНПОСТАВКИ

Поставка сырья и материалов, содержащих драгоценные металлы.

**КОМПЛЕКСНАЯ ПОСТАВКА ЭЛЕКТРОННЫХ КОМПОНЕНТОВ
ОТЕЧЕСТВЕННОГО И ИМПОРТНОГО ПРОИЗВОДСТВА**

- ОАО Донской завод радиодеталей (ДЗРД)
- ООО НПП «Томиллинский электронный завод»
- ЗАО «Кремний-Маркетинг»
- ОАО «Ангстрем»
- ОАО НПК «Северная заря»



www.mvs-cvp.ru

- ОАО «Уральский завод электрических соединителей» (ОАО «Завод «Исеть»)
- ОАО «Завод «Атлант»
- ОАО «Трубчевский завод «Нерусса»
- ФГУП «Карачевский завод «Электродеталь»
- ОАО «Завод «Мезон»



WAGO
INNOVATIVE CONNECTIONS



TOSHIBA



TRACO
POWER



TEMEX



VISHAY



ANALOG
DEVICES



SAMSUNG



SAMSUNG
ELECTRO-MECHANICS



OSRAM



Opto Semiconductors



TEXAS INSTRUMENTS



DALLAS
SEMICONDUCTORS



MAXIM

Поставка изделий с приёмкой "1", "5" и "9" со склада и под заказ по заводским или договорным ценам (всю информацию Вы можете получить у менеджеров)

127591, г. Москва, Дмитровское шоссе,
д.100/2; тел.: (495) 780-5384, 780-5387;
факс: (495) 788-61-05; e-mail: ec@mvs-cvp.ru

194100, г. Санкт-Петербург,
ул. Кантемировская, д.12, лит.А.;
тел.: (812) 324-40-15; факс: (812) 324-2238
e-mail: office_spb@mvs-cvp.ru

- если управление третьим состоянием выключено и выход находится в активном состоянии, то из цепи читается информация, полностью соответствующая состоянию выхода;
- если управление третьим состоянием включено и выход находится в неактивном состоянии, то из цепи читается либо информация, соответствующая состоянию подключенного к этой цепи входа, либо неопределённая информация.

Во втором случае на информацию, которая считывается из цепи, может влиять ещё незакончившийся процесс перезарядки конденсаторов, когда цепь хранит состояние, предшествующее тому, в котором производилось измерение.

Кроме того, в тестируемую цепь возможно проникновение помехи от соседних цепей или от источника питания. Процесс перезарядки конденсаторов, вызываемый последовательностью выполнения тестов, всегда определён. В том случае, когда информация определяется наводкой, результат чтения из цепи носит нерегулярный характер. Если же в цепь включен опорный резистор или имеется компонент, который хранит последний по времени уровень сигнала – кеерг, то информация, читаемая из такой цепи, становится определённой.

Итак, надо определить, в каких битах регистра сканирования будет достоверная информация при текущем состоянии схемы. Чтобы упростить определение неисправных цепей, введём такие понятия, как «выданная в регистр сканирования строка», «принятая из регистра сканирования строка», «строка маски» и «строка паттерна». Если данные, которые мы выдаём в регистр сканирования, представить как строку из набора нулей и единиц, то вот её мы и назовем «выданная в регистр сканирования строка». Принятые данные будем трактовать как «принятая из регистра сканирования строка». Тогда термин «строка маски» будет обозначать строку данных, состоящую также из нулей и единиц. На тех позициях, которые нас не интересуют в данном тесте, мы поместим нули, а там, где мы хотим увидеть и проанализировать результат чтения из регистра граничного сканирования, мы поместим единицы. Значение, которое мы хотим получить при тес-

тировании, назовём «строкой паттерна».

Тестирование производится следующим образом. Выдаём строку данных в регистр сканирования, получаем из регистра данные. Затем на принятые данные накладываем маску и полученный результат сравниваем с паттерном. Как видите, всё очень просто.

Но тогда почему же все средства тестирования, а особенно программные инструменты, так дороги? Ответ тоже прост, ибо, как говорится в одной рекламе: «Всё дело в волшебных пузырьках»... Во-первых, хорошие программы сами работают со списком связей и сами разделяют цепи на группы, в соответствии с которыми будут выданы тестирующие воздействия. Точно так же эти программы сами формируют цепочки для регистров сканирования в том случае, когда в цепи включено несколько микросхем. Ну и, наконец, эти программы сами определяют порядок выдачи тестовых воздействий.

Рекламная пауза окончена, и теперь мы более подробно продолжим рассматривать тестовые воздействия. Как нам уже известно, команды Sample-Preload и Extest переключают мультиплексор регистра граничного сканирования на чтение внешних по отношению к микросхеме данных, а команда Intest – внутренних данных, т.е. данных из ядра микросхемы (Core Logic). И ещё одно важное замечание. Команда Intest читает данные, которые уже сформированы в ядре микросхемы, поэтому здесь достаточно одного цикла чтения. Команда Sample-Preload записывает новые данные в регистр граничного сканирования, но не выводит записанные в регистр граничного сканирования данные из микросхемы. Читаются же внешние данные, поступающие на входы микросхемы. Здесь также достаточно одного цикла чтения. А вот команда Extest, как только она будет выдана в цикле записи команды, тут же выводит из микросхемы записанные в регистр граничного сканирования данные, и они появляются на выходах. Если после выдачи команды Extest мы произведём запись данных, то эти новые данные появятся на выходах только после завершения цикла чтения-записи. В этом случае данные, которые появятся на выходах микросхемы после выполнения ко-

манды Extest, могут изменить состояние проверяемого изделия. Поэтому в таком случае достоверно считанными данными будут те, которые мы получим после второго чтения.

Мы рассмотрели, какие команды и в какой регистр нужно помещать, чтобы данные с выхода одной микросхемы прочитать на входе другой. Но и это ещё не всё! Вспомните, что длина регистра граничного сканирования у микроконтроллера или у микросхемы FPGA может быть от нескольких сотен до тысячи и более бит! А теперь представьте, что у вас в цепочке находится хотя бы пять таких микросхем. Тогда для каждого теста нам надо выдать $5 \times 1000 \times 2$ циклов записи по JTAG – всего 10 000 циклов. Представим, что мы хотим выполнить тест «бегающая единица». И ещё представим, что эти пять микросхем расположены, например, в трёх физически разделённых узлах на плате. То есть цепи от одной группы микросхем никак не пересекаются с цепями от другой группы и, следовательно, не могут иметь общих неисправностей, например замыканий. Тогда для такой платы будет целесообразно «гонять» тесты по 10 000 циклов. Разделим также и тесты на три группы. Для тех микросхем, которые не относятся к определённой группе, применим режим Clamp или HighZ. Вспомним, что в режиме Clamp данные на выходах микросхемы будут зафиксированы, а в режиме HighZ выходы будут находиться в третьем состоянии. Таким образом, в те микросхемы, куда мы подавали команду Clamp, надо выдать данные только один раз в начале теста, а потом данные будут «защёлкнуты» и останутся неизменными при последующих тестах.

При такой методике тестирования суммарная длина выдаваемой цепочки сократится в три раза. Соответственно, сократится и время обработки результатов тестирования.

И последнее, что необходимо упомянуть, – это сама методика тестирования изделия. Нет нужды описывать простейшие тесты – «шахматные коды», «счётчик», «бегающая единица», – они общеизвестны.

Сколько именно тестов надо выполнить, в какой последовательности и насколько достоверным при этом будет результат – выходит за рамки данной статьи.

Технологии JTAG-тестирования ещё развиваются. Кроме технологии тестирования для дискретных сигналов, описанных в статье, появились технологии для смешанных аналого-цифровых сигналов. Такие технологии применяются в том случае, если в изделии используются стандарты сигналов типа LVDS. Также появились и технологии для сканирования аналоговых сигналов. Такие технологии призваны обеспечить тестирование, например, цепей питания. Так же, как и технологии сканирования, развиваются и технологии внутрисхемного программирования.

ЗАКЛЮЧЕНИЕ

Кроме тех программ и путей решения проблем тестирования, которые обсуждались выше, существует достаточно большое количество методов, позволяющих проверять изделия. В рамках журнальной статьи нет возможности описать или хотя бы перечислить их все. Если же сравнивать ручное и автоматическое тестирование, то разница между ними будет в том, что при автоматическом тестировании можно задать больше тестов, выбрать различные форматы работы выводов микросхемы и т.д. Для автоматического тестирования лишней час при проверке десятков тысяч цепей на каждой плате никак не сказывается на общих расходах фирмы.

Ведь сегодня простенький компьютер, занимающийся тестированием, гораздо дешевле, чем регулировщик, задействовать таких компьютеров можно довольно много, а использовать их можно «в три смены».

В последнее время в российских периодических изданиях всё больше компаний рекламируют свои возможности по изготовлению печатных плат и контрактной сборке. А это значит, что проблемы автоматического тестирования станут более актуальными и для российских разработчиков.

ЛИТЕРАТУРА

1. ADI <http://www.analog.com>.
2. ООО «Элтех» www.eltech.spb.ru.
3. Boundary Scan Coach. GOEPEL Electronic. <http://www.goepel.com>.
4. Design and Java Applets Support for an Asynchronous-Mode Learning of Digital Test. <http://www.pld.ttu.ee/applets>.
5. Universal Scan. Ricreations, Inc. <http://www.universalscan.com>.
6. Платунов А.Е., Постников Н.П., Чистяков А.Г. Механизмы граничного сканирования в неоднородных микропроцессорных системах. Chip News. http://lmt.cs.ifmo.ru/article_chip_news.html.
7. Рустинев В., Городецкий А. «Разделяй и властвуй» – принцип граничного сканирования. Chip News. http://chip-news.gaw.ru/html.cgi/archiv/01_06/stat-3.htm.

8. Каршенбойм И. Виртуальные кнопки и светодиоды, или Неизвестное обо всём известном JTAG_сканировании. Компоненты и технологии. 2005. № 6.
9. <http://www.national.com/appinfo/scan/index.html>.
10. IEEE 1149.1 (JTAG) Boundary-Scan Testing for Stratix II Devices. Ch. 9. Altera. www.altera.com.
11. Kuznetsov D. JTAG Boundary-Scan Test. Introduction. http://www.orc.ru/~dkuzn/j_intro.htm.
12. <http://jtagtools.sourceforge.net/download.html>.
13. IEEE Standard Test Access Port and Boundary-Scan Architecture. IEEE Std 1149.1-2001.
14. Boundary-Scan Test and In-System Programming Software. Corelis. http://www.corelis.com/products/Test_Software.htm.
15. Serial Vector Format Specification. ASSET InterTech, Inc. Copyright 1997-1999. Texas Instruments Inc. Copyright 1994. www.asset-intertech.com/support/svf.pdf.
16. www.amontec.com. JTAG Interface: Common Pinouts amt_ann003 (v1.1) Application Note.
17. EIA/JEP106, JEDEC Publication 106, Standard Manufacturer's Identification Code.
18. Каршенбойм И. Микропроцессор своими руками/4. Как отладить встроенный в FPGA микроконтроллер? Компоненты и технологии. 2006. № 11.



Новости мира News of the World Новости мира

Fujitsu представила 2-Мб модули FRAM

Fujitsu на днях объявила о начале производства двух новых модулей оперативной памяти типа FRAM (Ferroelectric Random Access Memory – сегнетоэлектрическая память с произвольным доступом) ёмкостью 2 Мбит в конфигурации 256x8 (MB85R2001) и 128Kx16 (MB85R2002). Напомним, что FRAM является разновидностью энергонезависимой памяти, которая используется для хранения данных сегнетоэлектрические компоненты (транзисторы и конденсаторы), обладающие переменными ферромагнитными свойствами. По основным техническим характеристикам – быстродействию, энергопотреблению и количеству циклов записи – модули FRAM намного опережают аналоги, построенные на базе флэш-памяти.

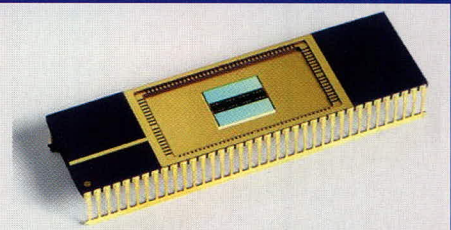
Модули FRAM способны перенести 10 млрд. циклов перезаписи и гарантируют сохранность данных в течение 10 лет.

Рабочее напряжение новинок составляет 3,0...3,6 В, среднее время доступа к ячейке 100 нс, продолжительность цикла записи/чтения 150 нс. Основными областями применения новинок станет офисное оборудование, навигационные и измерительные приборы и многофункциональные принтеры – везде, где понадобится оперативное изменение большого числа различных параметров и обеспечение их сохранности на длительный срок.

digitimes.com

Intel разрабатывает стеклянную память

Полупроводниковый гигант планирует в рамках IDF озвучить новые подробности относительно собственного типа памяти. Пока лишь известно, что компания Intel в данный момент готовит производство своей собственной памяти с изменением фазового состояния (phase change memory, PCM), которая носит обозначение PRAM.



Известно, что такая память будет энергонезависимой, подобно флэш-памяти. Скорость записи и чтения такой памяти будут на уровне типичных модулей памяти DDR, доступных на рынке уже сегодня. Но главной особенностью PRAM-памяти является материал, из которого она будет изготавливаться. Материалом служит стекло, которое позволяет обеспечивать фактически бесконечный срок службы. Известно также, что применение стекла при создании памяти даёт возможность функционирования при частом изменении сопротивления вследствие воздействия тепла, генерируемого электрическими потоками.

techinform.net

Частотный метод анализа синтезаторной системы импульсно-фазовой автоподстройки частоты. Часть 1. Фильтрация помех структурой ФАП

(продолжение)

Юрий Никитин (Санкт-Петербург)

Одним из наиболее важных свойств кольца ФАП является фильтрация помех. Рассмотрению этого вопроса посвящена данная часть статьи.

Анализ фильтрации помех кольцом ФАП

На рис. 1 приведена линеаризованная структурная схема синтезаторного квазиэлектронного кольца ИФАП, предназначенная для анализа его фильтрующих свойств.

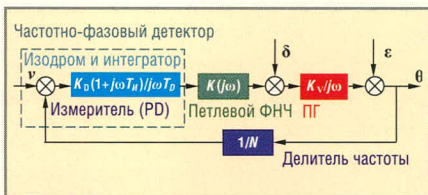


Рис. 1. Структурная схема кольца ФАП с ЧФД для анализа фильтрации помех

v – малые помехи, приходящие с опорным колебанием $\omega_{ОП} = 2\pi F_{REF}$; δ – малые помехи, воздействующие на вход управления ГУН; ϵ – малые помехи, воздействующие на выходное колебание ГУН; измеритель (Error Detector) включает в себя собственно фазовый детектор PD (Phase Detector) и, может быть, источник тока поддержания заряда CP (Charge Pump) с изодромным звеном (изодром + интегратор); T_D – постоянная времени ЧФД; T_I – постоянная времени изодромного звена; K_D – крутизна статической характеристики ЧФД [рад/А] в точке устойчивого равновесия; $K(j\omega)$ или $K(s)$ – коэффициент передачи петлевого фильтра (Loop Filter) или фильтра нижних частот (ФНЧ) – запись в частотной или в операторной форме; K_V – крутизна управляющей характеристики ГУН [Гц/В]; Θ – малые помехи в выходном колебании кольца; $1/N$ – коэффициент передачи делителя (частоты) с переменным или дробно-переменным коэффициентом деления (ДПКД или ДДПКД) в тракте приведения частоты ПГ

Запишем для коэффициента передачи разомкнутого кольца в операторной форме:

$$G_N(s) = K_D K_V K(s) \frac{1 + sT_I}{s^2 NT_D} = \frac{G_1(s)}{N}, \quad (1)$$

где $G_1(s)$ – операторный коэффициент передачи разомкнутого кольца с единичной обратной связью ($N = 1$). Для качественного (предварительного) анализа фильтрующих свойств кольца ФАП удобно считать, что петлевой ФНЧ отсутствует и $K(s) = 1$. Характеристики непрерывного (не импульсного) бесфильтрового кольца с единичной обратной связью наиболее просты в построении и наглядны при анализе; с ними удобно сравнивать характеристики более сложных колец ФАП [1–5]. Коэффициент передачи на выход системы

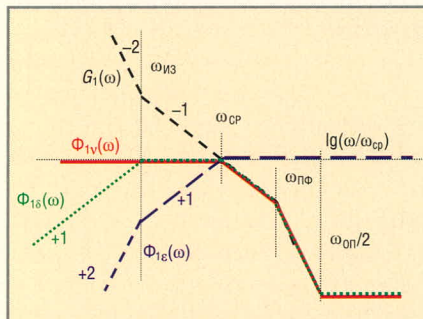


Рис. 2. ЛАХ бесфильтрового кольца ИФАП с единичной обратной связью для анализа фильтрации помех

Цифрами обозначены нормированные крутизны ЛАХ: $\pm 1 \Leftrightarrow \pm 20$ дБ/дек $\equiv \pm 6$ дБ/окт; $\pm 2 \Leftrightarrow \pm 40$ дБ/дек $\equiv \pm 12$ дБ/окт.

ФАП фазовых шумов v опорного сигнала, фазовых возмущений δ и ϵ , вызванных помехами, воздействующими на цепь управления ПГ и на собственно ПГ, можно представить в следующем виде:

$$\Phi_{1v}(s) = \frac{\Theta}{v} = \frac{G_1(s)}{1 + G_1(s)}, \quad \text{где } \delta \text{ и } \epsilon = 0, \quad (2)$$

$$\Phi_{1\epsilon}(s) = \frac{\Theta}{\epsilon} = \frac{1}{1 + G_1(s)}, \quad \text{где } \delta \text{ и } v = 0, \quad (3)$$

$$\Phi_{1\delta}(s) = \frac{\Theta}{\delta} = \frac{1}{1 + G_1(s)} \frac{K_V}{s} = \Phi_{1\epsilon}(s) \frac{K_V}{s}, \quad \text{где } \epsilon \text{ и } v = 0. \quad (4)$$

ЛАХ $G_1(s)$, $\Phi_{1v}(s)$, $\Phi_{1\epsilon}(s)$ бесфильтрового кольца ФАП с единичной обратной связью приведены на рис. 2. Из рис. 2 и формул (2) – (4) следует, что для помех, приходящих с опорным колебанием, кольцо ФАП является фильтром нижних частот с частотой среза (единичного усиления) ω_{CP} , в то время как для помех, воздействующих на ПГ (ГУН), кольцо ФАП является фильтром верхних частот. Частоты среза (излома) у обоих фильтров равны. Сказанное означает, что в области фазовых возмущений слева от частоты среза кольца ФАП (рис. 2) кольцо компенсирует дестабилизирующие воздействия на ПГ и делает это тем эффективнее, чем шире полоса прозрачности кольца (выше его частота среза ω_{CP}) и ниже частота возмущения.

Например, кольцо эффективно отслеживает изменения параметров элементов вследствие старения, тем-

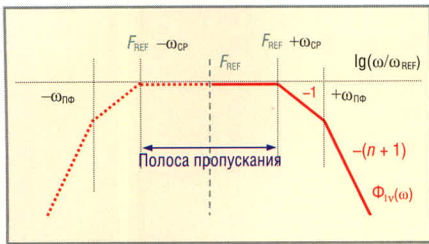


Рис. 3. ЛАХ следящего фильтра аналогового кольца ФАП с единичной обратной связью

пературные уходы ПГ; несколько хуже – воздействия механических вибраций, ещё хуже, но ещё эффективно, – воздействие помех по цепям питания. А ВЧ-наводки на ПГ кольцо пропускает на выход без ослабления – в этом случае помогает рациональная топология и экранирование. Зато низкочастотные вариации частоты опорного колебания кольцо беспрепятственно пропускает на выход. Для помех, воздействующих на вход управления ПГ (варикапы ГУН), кольцо ФАП является низкодобротным полосовым фильтром, подчёркивающим приходящие помехи в области частот от излома издромного звена $\omega_{ИЗ} = 1/T_{ИЗ}$ до частоты среза кольца $\omega_{СР}$. Правый от частоты $\omega_{СР}$ «положительный» излом на ЛАХ рис. 2 вызван наличием петлевого фильтра с коэффициентом передачи $K(s)$ и частотой излома $\omega_{ПФ}$. «Отрицательный» излом ЛАХ на частоте $\omega_{ОН}/2$ есть следствие импульсности системы ИФАП.

Отметим ещё одно важное обстоятельство. Из анализа ЛАХ коэффициента передачи $\Phi_{1V}(\omega)$ (рис. 2) можно сделать вывод, что непрерывное кольцо ФАП с единичной обратной связью является высокоэффективным следящим фильтром частоты $\omega_{ОН} = 2\pi F_{REF}$. Характеристика избирательности этого фильтра тем выше, чем круче спад ЛАХ кольца за частотой сравнения $\omega_{СР}$, т.е. чем больше звеньев содержит петлевой ФНЧ (рис. 3).

Следует добавить, что у аналогового кольца ФАП, которое используют в качестве следящего фильтра частоты F_{REF} , нет ограничений на протяжённость зоны фильтрации $\pm 0,5F_{REF}$, как у цифровых (импульсных) колец ИФАП. При проектировании такого кольца необходимо лишь соблюсти условия сохранения устойчивости вследствие набега фазы на частоту среза кольца $\omega_{СР}$ от звеньев петлевого ФНЧ.

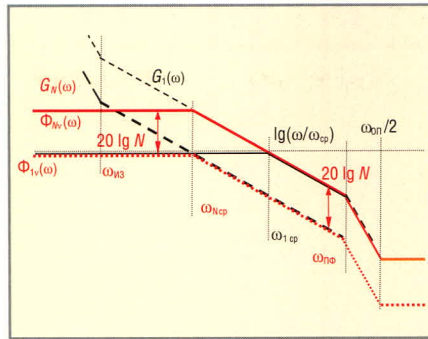


Рис. 4. ЛАХ синтезаторного кольца ИФАП для анализа фильтрации помех ν

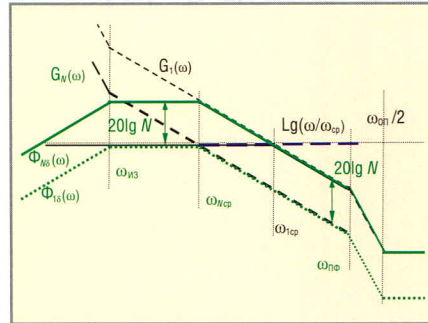


Рис. 6. ЛАХ синтезаторного кольца ИФАП для анализа фильтрации помех δ

Для синтезаторного кольца формулы (2) – (4) изменяются:

$$\Phi_{N\nu}(s) = \frac{G_1(s)}{1+G_1(s)} N = \Phi_{1\nu} N, \quad (5)$$

$$\Phi_{N\epsilon}(s) = \frac{1}{[1+G_1(s)]} \frac{1}{N} = \frac{\Phi_{1\epsilon}(s)}{N}, \quad (6)$$

$$\Phi_{N\delta}(s) = \Phi_{N\epsilon}(s) \frac{K_V}{s}. \quad (7)$$

Соответствующие формулам (5) – (7) графики показаны на рис. 4 – 6. Там же для сравнения пунктиром приведён график $G_1(\omega)$ для кольца с единичной обратной связью.

Формулы (5) – (7) и рис. 4 – 6 показывают, как изменяются фильтрующие свойства кольца при введении делителя на N в цепь обратной связи. Важный для практики вывод заключается в том, что зона эффективной компенсации помех, воздействующих на ПГ, существенно сужается. Так же существенно (в N раз) кольцо подчёркивает низкочастотные флуктуации опорного колебания и помехи, воздействующие на вход управления ПГ [6].

Анализ ЛАХ коэффициента передачи $\Phi_{N\nu}(\omega)$ (рис. 4) показывает, что импульсное кольцо ИФАП (АФАП) с неединичной обратной связью про-

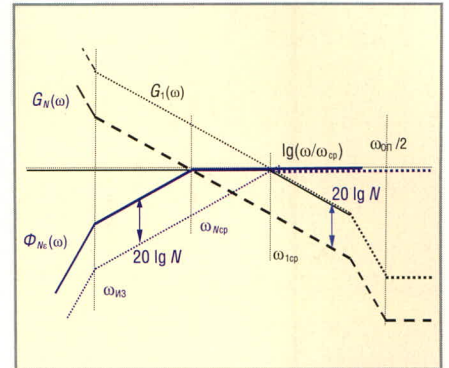


Рис. 5. ЛАХ синтезаторного кольца ИФАП для анализа фильтрации помех ϵ



Рис. 7. ЛАХ коэффициента передачи помех, приходящих с опорным сигналом на выход импульсного кольца ФАП с ДПКД на N в цепи обратной связи

должен оставаться эффективным фильтром частоты NF_{REF} . Характеристика избирательности этого фильтра тем выше, чем круче спад ЛАХ кольца за частотой сравнения $\omega_{СР}$, т.е. чем больше звеньев содержит петлевой ФНЧ (рис. 7).

Ухудшение фильтрующих свойств синтезаторного (цифрового) кольца ИФАП по сравнению с обычным аналоговым кольцом ФАП – следящим фильтром – есть плата за переход в новое качество – возможность синтеза сетки частот.

ЛИТЕРАТУРА

1. Шаширо Д.Н., Паин А.А. Основы теории синтеза частот. М.: Радио и связь, 1981.
2. Манассевич В. Синтезаторы частот (Теория и проектирование). М.: Связь, 1979.
3. Рыжков А.В., Попов В.Н. Синтезаторы частот в технике радиосвязи. М.: Радио и связь, 1991.
4. Зарецкий М.М., Мовинович М.Е. Синтезаторы частоты с кольцом фазовой автоподстройки. Л.: Энергия, 1974.
5. Макаров И.М., Менский Б.М. Линейные автоматические системы. М.: Машиностроение, 1977.
6. Карпов Л.В., Никитин Ю.А. Инженерная методика расчета астатической синтезаторной системы ФАПЧ. Известия ВУЗов. Приборостроение. 1990. № 11. С. 50–56.

ОС Linux для систем на кристаллах ПЛИС фирмы Xilinx

(часть 1)

Алексей Шматок (Москва)

В статье обсуждаются особенности разработки и реализации системного и прикладного программного обеспечения для систем на кристаллах ПЛИС, работающих под управлением ОС Linux.

Системы на кристалле (СнК, SoC) ПЛИС получили широкое распространение и используются для решения многих задач. Во всём их многообразии можно выделить СнК, в которых присутствуют микропроцессорные ядра и типовые периферийные устройства: внешняя память, сетевой адаптер, последовательный порт и т.п. Для таких систем в качестве базовой платформы для построения приложений как нельзя лучше подходит операционная система (ОС) с открытым кодом – Linux [1].

Цель данной статьи состоит в том, чтобы поделиться опытом разработки СнК на платах RMB-411 [2] и ML401 [3], дать обзор необходимых инструментальных средств разработки СнК ПЛИС Xilinx, генерации BSP, кросс-компиляции под микропроцессорную архитектуру PPC405/MicroBlaze, сборки ядра Linux, загрузчика U-boot и RootFS, а также разработки системного ПО для собственных IP-ядер (custom cores).

Конфигурация средств разработки и оборудования

На рисунке 1 показана типовая схема подключения используемого для разработки оборудования. К рабочей станции через коммутатор подключаются рабочие платы (модули ПЛИС). При помощи технологии виртуализации Linux-сервер и рабочая станция могут быть совмещены на одном физическом компьютере.

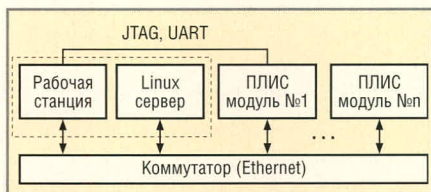


Рис. 1. Схема подключения оборудования

Для отладки применяются кабели JTAG и UART.

На рабочей станции должно быть установлено программное обеспечение для разработки систем на кристалле – Xilinx ISE, XPS. Linux-сервер обеспечивает работу служб DHCP, TFTP, NFS, а также служит основой для разработки (кросс-компиляции) приложений под архитектуру микропроцессорных ядер PPC405, MicroBlaze и т.п.

СИСТЕМА НА КРИСТАЛЛЕ ПЛИС

Минимальная тестовая конфигурация для отладочной загрузки ядра Linux должна включать:

- микропроцессорное ядро, контроллер прерываний;
- внешнюю память;
- UART;
- JTAG.

Такая конфигурация позволяет выполнить необходимый минимум: через JTAG загрузить ядро (*zImage.initrd*), создать в памяти виртуальный диск *rootfs* и подключить к системе через терминал (UART).

Желательно наличие дополнительных устройств:

- контроллера Ethernet;
- Flash-памяти;
- последовательного ПЗУ (SPROM);
- пользовательского ядра Custom Core.

Начальная конфигурация ПЛИС загружается из SPROM при включении питания. Flash-память содержит либо программный загрузчик (например, U-Boot), либо образ RAM-диска или даже часть *rootfs*.

Через сеть Ethernet производится подключение к удалённому серверу по TFTP, выполняется монтирование сетевых каталогов с NFS; возможно

монтирование *root*. В процессе разработки удобнее загружать ядро и образ RAM-диска с сервера через TFTP, а окончательный вариант записать во flash-память.

Пользовательское ядро решает прикладные задачи, при этом микропроцессорная система под управлением ОС Linux выполняет сервисные функции, служит платформой для разработки и вместе с тем обеспечивает взаимодействие с различными распространёнными интерфейсами.

На рисунке 2 схематично представлена типовая рабочая конфигурация СнК ПЛИС с микропроцессорным ядром PPC 405. Подобные конфигурации будут реализованы и для других типов микропроцессоров, например MicroBlaze, и даже для вашего собственного микропроцессорного ядра (soft core), если в таковом возникнет необходимость.

Использование единственного кристалла ПЛИС не всегда эффективно и возможно. Многие зависят от особенностей прикладного ядра и ресурсов, необходимых для его реализации. Использование кристалла большей ёмкости значительно увеличивает стоимость системы, вместе с тем определённых ресурсов самого ёмкого кристалла может быть недостаточно. Более гибким представляется решение с использованием нескольких кристаллов ПЛИС: «системного» и «прикладного». При таком подходе для реализации прикладного ядра и микропроцессорной системы подбираются адекватные задачам кристаллы ПЛИС. В системном кристалле собирается СнК под управлением Linux, что обеспечивает работу стандартных интерфейсов и протоколов. Пользовательская ПЛИС отводится для решения прикладных задач. Оба кристалла связываются локальной шиной, как показано на рисунке 3.

Также следует отметить проект [4], в котором ПЛИС используется только

для реализации прикладной задачи, а микропроцессорная система под управлением Linux собрана из отдельных микросхем.

ГЕНЕРАЦИЯ BSP для ядра LINUX и загрузчика U-BOOT

Процесс генерации BSP в EDK, как для различных версий ядра, так и для микропроцессоров PPC405/Microblaze, сводится к указанию определённых параметров SnK: используемого объёма памяти, перечислению периферийных устройств, рабочего каталога и т.п. Следует обратить внимание на возможные ошибки и несоответствия в коде BSP и в коде используемого ядра. Как правило, они устраняются, но требуют определённых усилий со стороны разработчика.

Генерация BSP для загрузчика U-Boot [5] выполняется на основе соответствующей библиотеки; её необходимо подключить в проект, провести необходимые настройки, аналогичные настройкам при генерации BSP для ядра Linux, и затем выполнить команду *Generate Libraries and BSPs*.

СРЕДСТВА РАЗРАБОТКИ для PPC405

Широкую известность получил скрипт crosstool [6], используя который, можно скомпилировать gcc под заданную микропроцессорную архитектуру.

Использование скрипта может быть следующим:

```
$ su root
$ mkdir /opt/crosstool
$ chown alex:users
/opt/crosstool/
$ su alex
$ cd ~
$ wget
http://kegel.com/crosstool/crosstool-0.43.tar.gz
$ tar -xvzf crosstool-0.43.tar.gz
$ cd crosstool-0.43
```

В файле *demo-ppc405.sh* выбираем требуемую версию gcc и glibc: *\$ sh demo-ppc405.sh*. Выполнение скрипта занимает довольно много времени, путь к GCC будет таким: */opt/crosstool/gcc-3.4.4-glibc-2.3.3/powerpc-405-linux-gnu/bin/*.

Вместо Crosstool можно использовать Embedded Linux Development Kit (ELDK) [7] – набор средств разработки для встраиваемых систем. Его

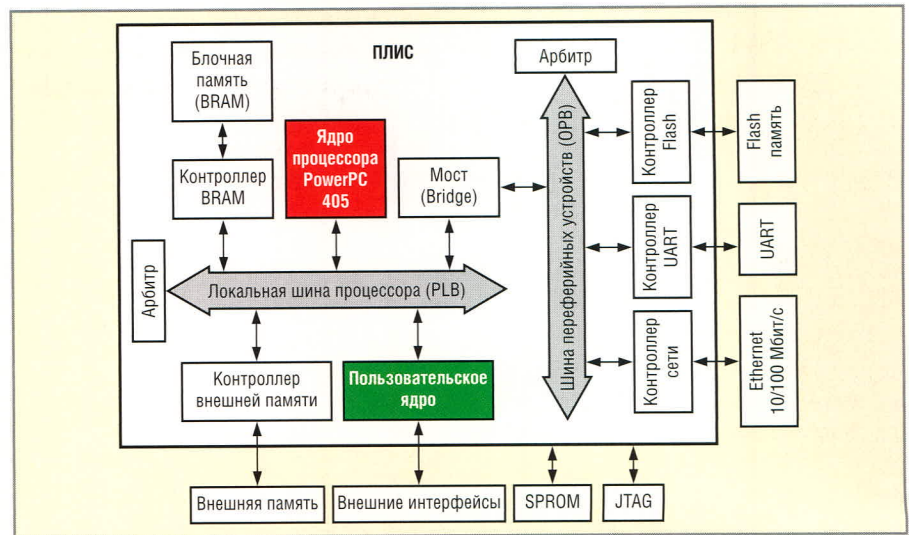


Рис. 2. Система на кристалле ПЛИС с PPC405

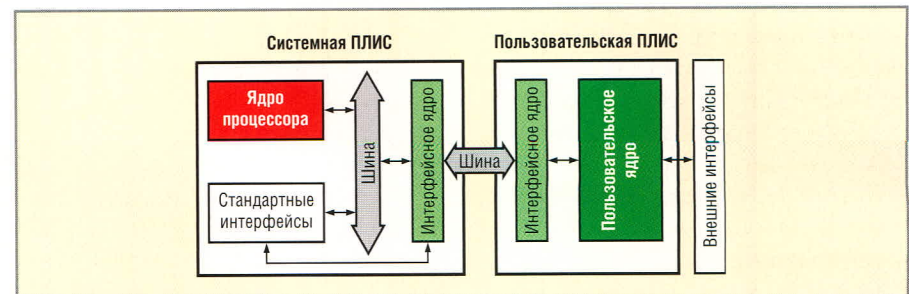


Рис. 3. Система на двух кристаллах ПЛИС

удобство заключается в том, что ELDK поставляется в виде отдельного образа инсталляционного диска, на котором имеются варианты установки как под Linux, так и под Windows, с использованием Cooperative Linux.

Ядро LINUX для PPC405/Microblaze

Основным сайтом разработки ядра Linux является www.kernel.org. Для архитектур PPC405/Microblaze развиваются отдельные ветви, основанные на определённых версиях основного ядра. Со временем изменения вносятся и в основную ветвь ядра, причём данный процесс несколько затянут и несогласован, в результате существует несколько версий ядра со своими особенностями и заплатками (patches). На этом пытаются заработать некоторые компании, предлагая свою поддержку, средства разработки и т.п. Несмотря на сложности, предпочтительно вести собственную разработку, чтобы иметь свободу действий и контролировать проект.

Итак, помимо основной ветки (www.kernel.org), версии ядра Linux 2.4/2.6 для PPC405 можно скачать следующим образом:

```
$ rsync -avz --delete
source.mvista.com::linuxppc_2_4_d
evel linuxppc_2_4_devel
$ cg-clone
git://source.mvista.com/git/linux
-xilinx-26.git
```

Для компиляции в make-файл необходимо прописать переменные ARCH и CROSS_COMPILE, далее всё как обычно. Настройка параметров ядра выполняется через вызов меню: *\$ make menuconfig*.

Объём статьи не позволяет привести список настроек; типовой конфигурационный файл, используемый для плат серии ML, можно получить с сайта Xilinx.

Настроив параметры ядра и скопировав в него BSP для вашей платы, следует выполнить компиляцию *\$ make dep && make zImage*.

Чтобы скомпилировать ядро с образом RAM-диска, содержащего rootfs, в каталог *arch/ppc/boot/images* помещается упакованный образ диска в файле *ramdisk.image.gz*, далее выполняется команда *make zImage.initrd*.

В отладочном режиме файл *zImage.initrd* может быть загружен через JTAG. Для SnK ПЛИС на базе Microblaze используется uClinux [8] –

дистрибутив, в который входит сразу несколько версий ядра (2.4, 2.6).

Средства кросс-компиляции и генерации BSP доступны по ссылке [9]. Принцип работы с uClinux аналогичный: генерация BSP, настройка параметров ядра, кросс-компиляция и загрузка.

Загрузчик U-Boot

Код загрузчика U-Boot доступен по адресу <http://sourceforge.net/projects/u-boot>. Генерация BSP для u-boot аналогична генерации BSP для ядра Linux. В случае необходимости в BSP можно добавить код инициализации, например, считывание MAC-адреса для сетевого адаптера из flash-памяти.

В каталоге *include/configs/* должен быть создан заголовочный файл рабочей платы, в котором определяются поддерживаемые команды и скрипты. Проверить полученный код U-Boot можно, загрузив образ через JTAG. В рабочем режиме код загрузчика хранится во flash-памяти.

Загрузчик/отладчик U-Boot может быть использован как самостоятельная среда для отладки и тестирования, в которой есть командный интерпретатор и поддерживается работа с различными периферийными устройствами. Загрузка Linux с использованием U-Boot может быть выполнена следующим образом: из flash-памяти считывается MAC-адрес сетевого адаптера, DHCP выделяет IP-адрес, через TFTP скачивается ядро Linux и образ RAM-диска, автоматически представляются параметры загрузки ядра, и затем выполняется его запуск.

RootFS

Наиболее простой путь создания *rootfs* заключается в использовании готового набора базовых пакетов для Linux-системы, такого как BusyBox [10], а также в ручной настройке системных файлов и скриптов.

Если существует необходимость в более мощных средствах работы с дистрибутивом Linux, используйте ScratchBox [11]. Хорошим примером проекта на его основе является Nokia Internet Tablet [12].

РАЗРАБОТКА СИСТЕМНОГО ПО ДЛЯ ПОЛЬЗОВАТЕЛЬСКИХ IP-ЯДЕР

Разработав собственное IP-ядро и подключив его к системной шине, можно быстро провести отладку и

тестирование под Linux, используя режим user mode, при этом не требуется написание драйвера, а только тестовое приложение. Типовой make-файл для компиляции таких приложений следующий:

```
# Program
PROGRAM=test-app
SRCS=$(PROGRAM).c
CC=/opt/eldk/usr/bin/ppc_4xx-gcc
TOPDIR=/opt/eldk/ppc_4xx/usr
LIBDIR=$(TOPDIR)/lib
INCDIR=$(TOPDIR)/include
CFLAGS=-I$(INCDIR)
LDFLAGS=-L$(LIBDIR)
OBJS=$(SRCS:.c=.o)
# Rules
.SUFFIXES: .c .o
.c.o:
    $(CC) -c $(CFLAGS) $<
default: all
all: $(PROGRAM)
$(PROGRAM): $(OBJS)
    $(CC) $(OBJS) -o $(PROGRAM)
$(LDFLAGS)
clean:
    rm -f $(OBJS) $(PROGRAM)
```

Само приложение должно содержать код инициализации, в котором указан физический адрес устройства (custom core) проецируется на виртуальный адрес, с которым можно работать непосредственно из приложения:

```
int core_fd=-1;
int *core_ptr=NULL;
off_t core_target=CORE_BASEADDR;
/* map custom core
to user space
*/
fd = open("/dev/mem",
O_RDWR | O_SYNC);
if (fd==-1){
/*failed*/
goto __exit_and_clean;
}
// get memory pointer
core_ptr = (int *)mmap(0,
CORE_MAP_SIZE,
PROT_READ|PROT_WRITE,
MAP_SHARED,
core_fd,
core_target &
~CORE_MAP_MASK);
if (core_ptr == MAP_FAILED){
/*failed*/
goto __exit_and_clean;
}
core_virt_addr=core_ptr+
```

```
(core_target & CORE_MAP_MASK);
// place here test code...
__exit_and_clean:
if (core_ptr!=MAP_FAILED)
munmap(core_ptr, CORE_MAP_SIZE);
if (core_fd!=-1) close(core_fd);
```

Написание драйвера займёт больше времени. Для компиляции кода можно использовать следующий make-файл:

```
# Driver
DRIVER=my_core
SRC=$(DRIVER).c
OBJ=$(DRIVER).o
DEPLOY=$(deploy)
CC=/opt/eldk/usr/bin/ppc_4xx-gcc
TOPDIR=/opt/eldk/ppc_4xx/usr
KINCDIR=/opt/linuxppc_2_4/include
LIBDIR=$(TOPDIR)/lib
INCDIR=$(TOPDIR)/include
CFLAGS=-c -O -DMODULE -D__KERNEL__
-I$(KINCDIR)
COMPILE=$(CC) $(CFLAGS) $(SRC)
#Rules
default: $(SRC)
    $(COMPILE)
clean:
    rm -f $(OBJ)
```

Для отладки лучше использовать динамическую загрузку:

```
$ insmod my_core.o
$ lsmod
$ rmmod my_core
```

При необходимости драйвер можно скомпилировать с ядром или включить в процесс генерации BSP.

Выводы и рекомендации

Для SnK ПЛИС операционная система Linux является наиболее адекватной платформой, которая обеспечивает поддержку множества периферийных устройств, различных интерфейсов и протоколов. В распоряжении разработчика имеется огромное разнообразие ПО с открытым исходным кодом, которое может быть кросс-компилировано под вашу платформу.

В качестве рекомендации для разработчиков хочется процитировать слова Дена Кегеля: «Создание средств разработки gcc / glibc для программирования встраиваемых систем бывало настолько трудным, что просто требовало железной воли, долгих дней, если не недель тяжелых усилий,

глубоких профессиональных знаний Unix и GNU, а иногда просто изворотливости».

Именно наличие «железной воли» позволяет добиваться результатов. Когда вы сталкиваетесь с ошибками компиляции, работая с кодом Linux, BSP и т.д., не отчаивайтесь – копируйте текст ошибки в поисковый запрос, и в большинстве случаев найдётся

описание того, как решается данная проблема.

Продолжение следует

ЛИТЕРАТУРА

1. ОС Linux <http://en.wikipedia.org/wiki/Linux>.
2. RMB-411 www.rosta.ru.
3. ML401 www.xilinx.com/ml401.
4. <http://www.elphel.com/3fhlo/index.html>.

5. Загрузчик U-boot <http://u-boot.sourceforge.net>.
6. <http://kegel.com/crosstool>.
7. ELDK www.denx.de.
8. uClinux <http://www.uclinux.org>.
9. Microblaze toolchain <http://www.petalogix.com/resources/downloads>.
10. BusyBox <http://www.busybox.net>.
11. ScratchBox <http://www.scratchbox.org>.
12. Nokia Internet Tablet www.maemo.org ©

Новости мира News of the World Новости мира

Sony: демонстрация сверхтонких OLED-телевизоров

На январской выставке потребительской электроники CES 2007 посетители стенда компании Sony могли полюбоваться интересными моделями OLED-телевизоров с диагоналями 11 и 27 дюймов. Тогда информации о новинках было немного, сегодня же появилась возможность восполнить этот пробел.

На японской выставке Display 2007, которая проходила в этом году с 11 по 13 апреля, компания ещё раз продемонстрировала публике свои новые телевизоры. Как стало известно, 11-дюймовая модель имеет разрешение 1024 × 600 пикселей, а 27" оснащена Full HD-матрицей (1920 × 1080 пикселей). При этом обе модели имеют поразительно высокую контрастность 1 000 000 : 1 (вероятно, речь идет о динамической контрастности), цветное пространство по шкале NTSC более 100%, яркость 600 нит, интерфейс HDMI.

Особо изумляет толщина панелей: всего 3 мм для 11" панели и 9 мм – для 27". Кроме этого, OLED-телевизоры Sony потребляют гораздо меньшую мощность, чем аналогичные плазменные и жидкокристаллические. Разработки 11-дюймовой модели уже находятся на стадии, близкой к массовому производству. О сроках

начала коммерциализации 27" телевизоров пока не сообщается.

dailytech.com

Беспилотный автомобиль Volkswagen Passat 2.0 TDI

Компания Volkswagen представила автоматизированный автомобиль Volkswagen Passat 2.0 TDI, который абсолютно не нуждается в водителе. Вместо него в корпус автомобиля вмонтировано два лазерных сенсора (один спереди, а другой сзади), которые играют роль технологичных глаз – следят за дорогой, оценивают расположение зданий, пешеходов в определённый момент времени на расстоянии до 200 м. Вся информация, собранная этими сенсорами, обрабатывается специальным компьютером, работающим под управлением ПО, которое разработала немецкая компания Ibeo Automobile Sensor.



В ноябре этот автомобиль будет соревноваться с другими роботизированными системами в безызывном шоу – гонках беспилотных автомобилей DARPA 2007, когда машинам придётся проехать самостоятельно более 60 миль.

По словам разработчиков, «коньком» проекта является совершенная лазерная технология построения моделей окружающей среды и ориентирования. Интересно отметить, что развернуть масштабное производство таких систем разработчики планируют уже в следующем году, хотя американские учёные предсказывали первые коммерческие беспилотные авто только через 25 лет.

physorg.com

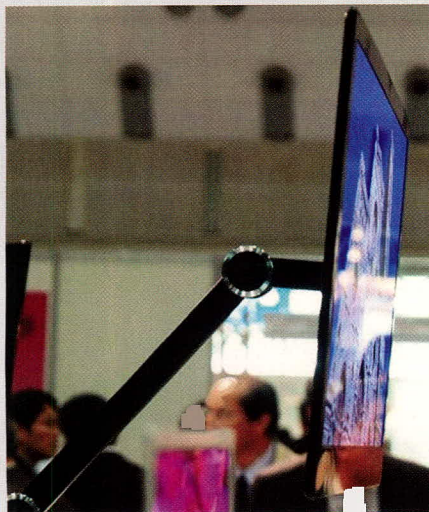
IBM: третье измерение продлит жизнь закону Мура

По мнению компании IBM, будущие суперкомпьютеры заметно выиграют в случае использования так называемой трёхмерной системы чипов. Сегодня чипы размещаются в двухмерной плоскости, а сообщение между ними организуется посредством проводных соединений, которые обладают далеко не идеальной пропускной способностью.

История такой технологии уходит в середину 90-х годов прошлого века. Инженерам потребовалось десятилетие на то, чтобы подготовить технологию к современному массовому производству. В соответствии с новой технологией, предполагается размещать чипы непосредственно друг на друге, а соединения между чипами осуществлять через каналы в кремниевом корпусе чипов. Данная технология получила название through-silicon vias, т.е. «соединения сквозь кремний». Преимущество достигается за счёт уменьшения расстояния между чипами в 1000 раз. Оно заключается в увеличении пропускной способности межчиповых соединений в 100 раз. Таким образом, по мнению сотрудников IBM, эффективность чипов повышается на 40%, что открывает путь новому поколению суперкомпьютеров.

Причём описанная технология, очевидно, не задержится в тесном кругу инженеров и исследователей. Ещё в прошлом году компания представила инновационный «многоэтажный» чип, оснащённый 80 ядрами и способный совершать более триллиона операций в секунду. Массовое же производство «многоэтажных» чипов IBM начнёт в 2008 г., представив предварительно несколько пробных вариантов уже в этом году. Более того, компания IBM не является единственной компанией, работающей в данном направлении. В то время как Intel пока только разрабатывает схожую технологию вертикального размещения чипов, компания Tru-Si уже обладает таковой.

news.bbc.co.uk



Конференция «Инновации в электронике»

Конференция «Инновации в электронике» прошла 26 апреля в рамках выставки ЭкспоЭлектроника. Конференция имела целью познакомить посетителей выставки с последними достижениями в области электроники: новыми компонентами, новым программным обеспечением, новыми приборами и оборудованием, новыми материалами и технологиями для разработки и производства электроники. Работа конференции проходила в двух секциях: «Компоненты, модули, приборы» и «Программное обеспечение». В качестве докладчиков были приглашены сотрудники представительств зарубежных компаний и российских фирм – дистрибьюторов электронных компонентов и матобеспечения: систем проектирования и операционных систем.

3M Конференция открылась докладом Хуршеда Дустбаева «Материалы компании 3M для монтажа печатных плат».

Термостойкие полиамидные ленты 3M разработаны для маскирования и защиты участков печатной платы в процессе пайки или других высокотемпературных процессов, когда требуется безостаточное удаление маскирующего материала.

Изотропные ленты 3M разработаны для различных целей монтажа электронных устройств: от экранирования от электромагнитных излучений до приклейки гибких шлейфов и печатных плат. Использование электропроводных лент значительно уменьшает вес устройства, т.к. исключает необходимость применения винтов и пр. крепёжных материалов.



Анизотропные электропроводные адгезивные ленты 3M – термоклейкие термопластичные и термополимеризуемые материалы, наполненные электропроводными частицами. При нагреве и приложении давления к плоскости ленты проводящие частицы образуют электропроводные цепи по оси Z – перпендикулярно плоскости ленты, в то же время выстраиваясь на достаточном расстоянии друг от друга по осям X и Y – в плоскости ленты, являющейся в этом случае диэлектриком.

Очень пластичные термопроводные подложки 3M объединяют в себе свойства термопроводности и огнестойкости, предназначаются для термопередачи при использовании конвекционных радиаторов и прочих пассивных компонентов. В задачах, где целью является более высокий коэффициент теплопроводности, используются силиконовые подложки. Также предлагаются подложки с неметаллическими наполнителями, служащие для поглощения электромагнитных шумов, для решения задач ЭМИ-поглощения.

Данные жидкие клеи 3M характеризуются отличной структурной адгезией. Нанесение дозированием позволяет использовать данные клеи как в автоматизированном, так и в ручном процессе. Текучесть клея способствует заполнению всех микронеровностей склеиваемых поверхностей, ультратонкий слой нанесения характеризуется малым термическим импедансом.

Светоотверждаемые адгезивы 3M – совершенно новый подход в процедуре склеивания. Отличная глубина полимеризации, её высокая скорость,

простота нанесения и полимеризация под воздействием как УФ-, так и видимого света способствуют значительному сокращению времени процесса.

Чистые эпоксидные адгезивы 3M Scotch-Weld Electronics Grade (EG) предназначены для использования при производстве сложных электронных устройств, когда важна коррозионная стойкость склеиваемых материалов и критично низкое газовыделение используемого адгезива.

Конформные покрытия 3M НОВЕК предназначены для защиты поверхностей печатных плат от внешних воздействий. Растворы характеризуются низкой вязкостью, низкой токсичностью, негорючестью, отсутствием в своём составе летучих органических соединений.



Фирма НТЦ «Модуль»
(докладчик Татьяна

Лысенко) представила доклад «Микросборка 2600ВГ2АТ – универсальная связная машина (терминал) мультиплексного канала передачи данных по ГОСТ Р 52070-2003».

В 2007 г. ЗАО НТЦ «Модуль» закончил разработку и испытания микросборки 2600ВГ2АТ – универсальной связной машины (терминала) мультиплексного канала передачи данных по ГОСТ Р 52070-2003 (MIL-STD-1553B). Данная гибридная микросборка интегрирует в своем составе логическую и приёмопередающую части абонента МКПД и предназначена для организации интерфейса управляющего вычислителя (ЦП) с резервированной информационной магистралью МКПД в авиационной, космической и специализированной аппаратуре различного функционального назначения.

Микросборка объединяет в своём составе приёмопередатчик, кодер информации МКПД, сдвоенный декодер информации МКПД, многопротокольную логику, логику управления, логику взаимодействия с ЦП и управления памятью, внутреннее статическое ОЗУ объёмом 4К 16-разрядных слов, ряд пассивных компонентов. Она выполнена в 72-выводном двустороннем планарном металлическом корпусе с

шагом между выводами 1,0 мм и имеет габариты 41,0 × 30,0 × 4,3 мм (без учёта выводов), что позволяет использовать её в аппаратуре с жёсткими требованиями по высоте компонентов.

Микросборка имеет ряд особенностей. Интегрированный интерфейс с управляющим вычислителем (ЦП) обеспечивает гибкость в выборе конфигурации и возможность прямого подключения к широкому ряду 8- и 16-разрядных микропроцессоров с минимальным использованием вспомогательной внешней логики. Микросборка содержит интегрированное внутреннее статическое ОЗУ объёмом 4К 16-разрядных слов, расширяемое до 64К 16-разрядных слов при использовании внешнего ОЗУ. Возможен выбор конфигураций интерфейса с ЦП и памятью:

- 8- или 16-разрядный буферизованный режим;
- 16-разрядный «прозрачный» режим;
- 16-разрядный режим с прямым доступом к памяти;
- поддержка режима без ожидания готовности.

Расширенные функции КШ:

- автоматическое возобновление попытки передачи сообщений;
- программируемые интервалы между сообщениями;
- автоповтор кадров;
- программируемый интервал ожидания ответного слова ОУ.

Расширенные функции ОУ:

- программируемое задание недопустимости команд;
- работа в режимах одиночного сообщения, двойной буферизации, круговой буферизации;
- настраиваемые прерывания;

- гибкие возможности буферизации данных.

Расширенные функции МТ:

- режим словного монитора;
- режим монитора избранных сообщений;
- совмещённый режим ОУ и монитора сообщений;
- запуск монитора по выбранному сообщению.

Номинал напряжений питания логической и приёмопередающей части 3,3 В, выходные логические уровни LVTTTL, входные логические уровни LVTTTL/LVCMOS/5 В TTL.

PROSOFT®
Электронные компоненты

С докладом «Системы на кристалле от компаний Cirrus Logic и RDC» выступил Андрей Архипов (фирма ПРОСОФТ).

Первым продуктом, на котором был сделан акцент, была мобильная система на кристалле EP9315 от производителя Cirrus Logic. Чип оснащён ядром ARM девятой серии и имеет в своём составе развитую периферию, включающую Ethernet, USB, IDE, PCMCIA-контроллеры, а также графический контроллер с функциями 2D-ускорения и поддержкой разрешения до 1024 × 768 точек. Описываемая система на кристалле обладает достаточно высоким быстродействием благодаря наличию раздельного кэша общим объёмом 32 Кб, системной шины с частотой 100 МГц при частоте ядра 200 МГц. Для данной системы существует много различных наборов для разработчика различных ценовых категорий. Средства разработки ПО входят в состав наборов. Принципиальные схемы и трассировка большинства отладочных плат открыты. Существуют и примеры готовых проектов

на основе отладочных плат – например, доступен проект готового NAS-сервера на основе отладочной платы EDB9315A (включая всё программное обеспечение с исходными текстами). Кроме того, для ARM-процессоров Cirrus Logic имеется отдельный интернет-ресурс (<http://arm.cirrus.com>). Он включает в себя форум, в котором участвуют как разработчики, так и технические специалисты компании Cirrus Logic. В форуме имеется множество разделов: программирование процессоров в 16- и 32-битном режиме, поддержка различных ОС, отладочные средства процессоров и т.п. Здесь же можно найти принципиальные схемы и трассировку отладочных плат, драйверы, библиотеки и примеры программ для различных ОС, а также бесплатные средства разработки программ (GCC) для процессоров Cirrus Logic под различные платформы.

Другими представленными системами на кристалле явились продукты R8610 и R8631 от компании RDC. Главное отличие этих продуктов – рабочая тактовая частота, а следовательно, и производительность. R8610 работает на частоте 150 МГц, R8631 – 300 МГц. Эти микроконтроллеры построены по 32-битной RISC-архитектуре и имеют совместимость с x86-платформами Windows и Linux, а также другими x86-совместимыми 32- и 16-разрядными операционными системами. Из встроенной периферии микроконтроллеры имеют 32-битный интерфейс PCI rev. 2.1, контроллер памяти SDRAM/ROM, контроллер Fast Ethernet 10/100 и USB 2.0 Host. Конструктивно микроконтроллеры выполнены в корпусах LQFP и LFBGA. При относительно высоком



ШИМ, UART, легко реализуются в PSoC. Кроме того, элементы, обычно не входящие в состав 8-битных микроконтроллеров, такие как фильтры, программируемые усилители, генераторы случайных чисел, также можно разместить в этих чипах. Следует отметить также гибкую систему тактирования всех узлов системы с возможностью использования как внутренних RC-генераторов, так и внешних источников частоты. Все функции, внутренние соединения, внутренняя конфигурация, и даже контакты ввода-вывода перепрограммируются пользователем. Непосредственно во время работы чипа PSoC можно динамически менять внутренние ресурсы, получая при этом функционально новую систему.

Каждый цифровой блок построен на основе трёх восьмиразрядных регистров и исходно представляет собой функционально неопределённый модуль. Разработчик может при помощи доступных процессору конфигурационных регистров задать топологию схемы цифрового блока, настроив тем самым его на выполнение той или иной функции.

Аналоговые блоки, в зависимости от типа, построены на основе операционного усилителя либо с программируемой матрицей резисторов в цепи обратной связи, либо с набором программируемых переключаемых конденсаторов. Настройка блока также осуществляется через конфигурационные регистры.

Процесс конфигурирования системы упрощается за счёт использования библиотеки модулей пользователя, которая представляет собой набор заранее настроенных и протестированных периферийных функций. Каждый библиотечный модуль подробно документирован – указываются занимаемые ресурсы, характеристики в зависимости от вариантов настроек мощности и напряжения питания, варианты размещения на массивах аналоговых и цифровых блоков. Также даны описание и примеры вызова API-функций, используемых процессорным ядром для управления модулем.

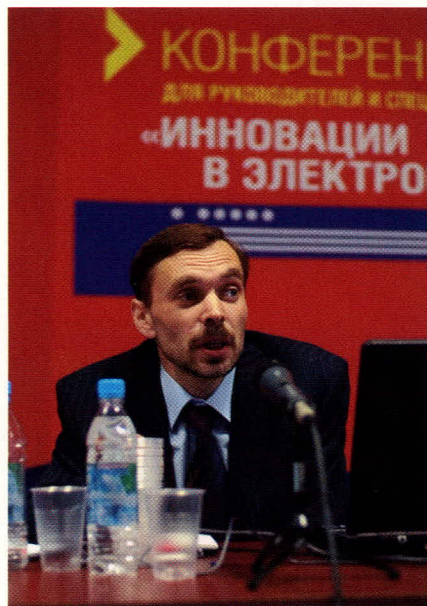
Конфигурирование системы выполняется на высоком уровне с помощью графического интерфейса интегрированной среды PSoC Designer, которая позволяет задать расположение модулей пользователя на массивах аналоговых и цифровых блоков, а также задать связи между ними.



Директор российского представительства корпорации Adtron Юрий Филатов представил доклад «Твердотельные накопители данных на основе флэш-памяти – новая продукция, перспективы развития».

Корпорация Adtron является одним из ведущих производителей твердотельных накопителей данных (SSD, флэш-диски) большой ёмкости. SSD-серии Flashpak® корпорации Adtron выполнены в стандартных для HDD форм-факторах 2.5" и 3.5" и снабжены интерфейсами IDE, SATA и SCSI. Поэтому они легко интегрируются в приложения, применяемые в оборонной и аэрокосмической отрасли, в промышленной автоматике, в медицинской промышленности, в сфере телекоммуникаций и на транспорте, а также на коммерческих предприятиях.

Во всех SSD-сериях Flashpak® используется прогрессивная технология Adtron ArrayPro™, обеспечивающая параллельный доступ и управление матрицей ячеек флэш-памяти. Технология ArrayPro™ выгодно отличает флэш-диски корпорации Adtron от низкобюджетных решений, использующих кэширование и нуждающихся в резервном электропитании, демонстрирующих существенную зависимость производительности от сферы применения. Продукты корпорации Adtron демонстрируют скорость непрерывной записи-чтения около 70 Мб/с, существенно превосходя возможности жёстких дисков с вращающимся носителем и удовлетворяя требованиям надёжности, предъявляемым при работе в тяжёлых условиях эксплуатации.



Помимо превосходной производительности и надёжности данное семейство твердотельных флэш-дисков отвечает широкому диапазону требований обеспечения защиты данных, как коммерческих предприятий, так и наиболее строгим стандартам, задаваемым военными и разведывательными службами. Средство Erasure™ Data Security Suite предоставляет возможности контроля доступа, а также высокоскоростного уничтожения данных, включая полное стирание в соответствии с современными требованиями систем защиты данных.

Сегодня корпорация Adtron представляет новые устройства серии Adtron Flashpak®, включая IDE и SATA флэш-диски моделей I25FB и A25FB соответственно. Эти продукты обладают ёмкостью до 160 Гб и являются на сегодняшний день наиболее вместительными SSD форм-фактора 2,5" на базе NAND-памяти SLC-типа. В данный момент только корпорация Adtron готова принимать заказы и отгружать потребителям твердотельные накопители с такими характеристиками.

Существенно увеличить ёмкость выпускаемых устройств позволило корпорации Adtron применение NAND-памяти SLC-типа с большей плотностью упаковки. Кроме того, экономические параметры новых флэш-дисков в сочетании с увеличенной ёмкостью существенно расширяют спектр их применения традиционными заказчиками, работающими на рынках промышленной и оборонной техники, а также позволяют применять их для оптимизации приложений с интенсивной серверной нагрузкой и систем хранения – в сфере, где обычно использовались жёсткие диски.

WIND RIVER Директор фирмы AVD Systems Алексей Демьянов сделал доклад на тему «Операционные системы Wind River VxWorks и RTLinux и современные средства разработки ПО встраиваемых микропроцессорных устройств и систем».

Компания Wind River, производитель средств разработки программного обеспечения встраиваемых микропроцессорных устройств, выпускает три группы продуктов:

- операционная система реального времени VxWorks, её средства разработки и middleware для VxWorks;

- операционная система Linux с расширениями для систем реального времени (RTLinux), её средства разработки и middleware для Linux/RTLinux;
- средства разработки для OCD (On-Chip Debugging) для VxWorks, Linux и «безОСовых» систем.

Операционные системы VxWorks и Linux поставляются в виде интегрированных пакетов, которые называются «платформами». Платформы Wind River выпускаются для четырёх основных областей применения: телекоммуникационное оборудование, промышленные контроллеры, бортовая авионика и бытовая электроника. В состав платформ входят: операционная система, среда разработки Wind River Workbench, связанное ПО (middleware) и BSP/драйверы (Board Support Packages). Все run-time-компоненты платформ поставляются в исходных текстах (source code).

Операционная система VxWorks и платформы на её основе поддерживаются для микропроцессорных архитектур PowerPC, x86/Pentium, ARM, MIPS, ColdFire, XScale, SuperH. Операционная система Linux и платформы на её основе поддерживаются для микропроцессорных архитектур PowerPC, Pentium, ARM, MIPS.

В группу средств разработки для OCD (On-Chip Debugging) входят JTAG-адаптеры, среда разработки Wind River Workbench OCD и оптимизирующий кросс-компилятор. Средства OCD поддерживают различные микропроцессорные семейства архитектур PowerPC, ColdFire, MIPS, ARM и XScale. Адаптеры Wind River поддерживают также JTAG-отладку многоядерных (multicore) микропроцессоров.

На сайте Wind River доступны для загрузки 30-дневные evaluation-вер-

сии платформ общего назначения General Purpose Platform VxWorks Edition и General Purpose Platform Linux Edition, среды разработки Wind River Workbench On-Chip Debugging и оптимизирующего кросс-компилятора Wind River Compiler.

Phyton™

О новых разработках ООО «Фирма Фитон» в области инструментальных средств для микроконтроллеров рассказал Юрий Зобнин.

В перечень выпускаемых компанией аппаратных и программных средств входят: кросс-ассемблеры и компиляторы, интегрированные среды, программные отладчики-симуляторы, внутрисхемные и JTAG-эмуляторы, универсальные программаторы, контроллеры-конструкторы. Выпускаемые в настоящее время средства обеспечивают полный цикл разработки для МК следующих семейств:

- ARM7/ARM9;
- 8051;
- PICmicro;
- 80196: Intel, НИИЭТ (г. Воронеж);
- AVR classic: Atmel;
- MSP-430: Texas Instruments;
- CoolRISC: XEMICS;
- RSC4xxx: Sensory;
- MAXQ: MAXIM/Dallas.

Новый продукт компании – пакет инструментальных средств для микроконтроллеров ARM® – CodeMaster-ARM – предназначен для обеспечения полного цикла разработки систем на базе МК с ядрами ARM7 и ARM9. В состав CodeMaster-ARM входят:

- интегрированная среда разработки СМ-ARM, объединяющая все программные компоненты системы в единый комплекс;
- многооконный редактор и менеджер проектов;

- компилятор СМ-ARM, включающий Си-компилятор, макроассемблер, линкер и утилиты;
- оптимизированный GNU GCC компилятор;
- программный симулятор ядра ARM7, ARM9;
- драйвер аппаратного JTAG-эмулятора JEM-ARM;
- программное обеспечение программатора Flash.

Пакет CodeMaster-ARM ориентирован на отладку программ на языке высокого уровня по исходному тексту. Поддерживается разработка программ на уровне ведения проектов для компиляторов языка Си СМ-ARM собственной разработки, а также компиляторов IAR Systems и GNU GCC. Встроенные многооконный редактор, менеджер проектов и большое количество сервисных возможностей существенно облегчают труд разработчика, избавляя его от рутинных операций. Поддерживается работа как в режиме проекта, так и в режиме чистого отладчика, позволяя загружать для отладки внешние исполняемые файлы, в том числе и без символьной информации.

Новые универсальные программаторы нового поколения фирмы «Фитон» ChipProg-48, ChipProg-40, ChipProg-G4, ChipProg-ISP имеют отличительные особенности:

- подключение к компьютеру через USB 2.0 совместимый порт;
- реализация программаторов на базе высокопроизводительного 32-разрядного процессора и высокоскоростной программируемой матрицы (FPGA) большого объема;
- реализация критических по времени частей алгоритма программирования на программируемой матрице аппаратно, что позволяет добиться очень высокой скорости программирования микросхем;



- аппаратный контроль каждого вывода программируемой микросхемы на наличие контакта перед программированием;
- режим автоматического распознавания присутствия микросхемы в колодке с автоматическим запуском выбранных процедур программирования;
- возможность работы неограниченного количества программаторов под управлением одного PC;
- поддержка проекта и ведение журнала программирования.



В рамках конференции был представлен обзорный доклад по средствам проектирования Mentor Graphics, который сделал директор российского дистрибьютора, ЗАО «МЕГРАТЕК», Андрей Лохов.

Mentor Graphics является одним из лидеров в области САПР электроники и предлагает широкий диапазон средств в области проектирования систем на кристалле и систем на платах, начиная от концептуального уровня проектирования и заканчивая подготовкой производства и постпроизводственным тестированием. В докладе был сделан обзор всего комплекса средств проектирования и верификации, а именно был представлен системный и RTL-уровень проектирования СБИС, физический уровень проектирования СБИС, средства верификации топологии СБИС в субмикронном диапазоне, средства повышения технологичности и выхода годных, средства тестирования и диагностики, средства проектирования систем на пе-

чатных платах (включая проектирование FPGA) и многое другое.

В целом было отмечено, что Mentor Graphics имеет технологическое лидерство по таким направлениям, как системный уровень проектирования и архитектурный синтез (Visual Elite, Vista, System Architect, Perspecta, Platform Express, BridgePoint, Catapult C), проектирование и функциональная верификация на RTL-уровне (ModelSim, Questa AFV, CodeLink, Seamless, VStation, Veloce), проектирование аналого-цифровых СБИС (ADMS), комплексная физическая верификация топологии СБИС в субмикронном диапазоне (Calibre), тестирование и диагностика (FastScan, TestKompress), проектирование систем на печатных платах (Expedition, PADS, FPGA Advantage, HyperLynx, DxSim).

В докладе также были представлены текущие успехи и дальнейшие планы по продвижению продукции Mentor Graphics на российском рынке. Было отмечено, что важной характеристикой продуктов Mentor Graphics является их «масштабируемость», т.е. способность предложить решения любого уровня – от дорогих, высокотехнологичных, до бюджетных. В частности, было предложено несколько относительно недорогих базовых решений для отечественных дизайн-центров, оптимальных по критерию стоимость/функциональные возможности.

В заключение был рассмотрен вопрос продвижения продукции Mentor Graphics в российские университеты с целью подготовки специалистов для промышленных предприятий и дизайн-центров.



Фирма «Альтернатив Солюшнс» представила доклад «Мировой рынок САПР СБИС. Решения компании Synopsys».

С уменьшением технологических норм возникали новые задачи для разработчиков САПР. На технологии 0,25 мкм основной задачей проектировщиков была оптимизация проекта по площади и мощности кристалла, с переходом на 0,18 мкм новой проблемой стала оптимизация по быстродействию кристалла в дополнение к существующим, на нормах 0,13 мкм добавилась задача анализа целостности данных, на 90 нм список пополнился задачей анализа токов утечки. Технологические нормы 65 и 45 нм только начинают осваиваться миро-

выми лидерами. Известно, что теперь основной задачей для проектировщиков является контроль проекта с точки зрения производственной пригодности, так как имеют место вариации технологического процесса, которые существенно влияют на характеристики кристаллов на данных нормах. Поэтому на данном этапе активно развиваются инструменты приборно-технологического моделирования.

Synopsys является единственной компанией – поставщиком САПР, маршрут проектирования и верификации которой сертифицирован компанией Honeywell для разработки радиационно-стойких СБИС для нужд NASA.

Инструменты Synopsys покрывают эталонный маршрут проектирования п/п фабрик TSMC, UMC, Tower Semiconductor, IBM, ST Microelectronics, Samsung, Chartered и др.

Теперь рассмотрим рынок САПР по уровням и направлениям проектирования. Данные расположены в виде таблицы на основе данных компании Gartner Dataquest, по состоянию на 2005 г. (в % по каждому направлению).

Уровни и направления проектирования	Доля рынка, %
Системный уровень	26
Логический уровень	85
Топологический уровень	44
Полнозаказное проектирование	3
Приборно-технологическое проектирование	85
Проектирование для производства	49
Топологическая верификация	15

Компания Alternative Solutions Alt-S является официальным дистрибьютором компании Synopsys и центром технической поддержки в России с 1998 г.



Фотографии Александра Либкова



10-я Юбилейная международная выставка «ЭкспоЭлектроника»

25–28 апреля 2007 г. в МВЦ «Крокус Экспо» состоялась 10-я Юбилейная международная выставка «ЭкспоЭлектроника». Организаторы выставки – международные выставочные компании «Примэкспо» и ITE Group Plc при содействии ООО «Электронинторг-С».

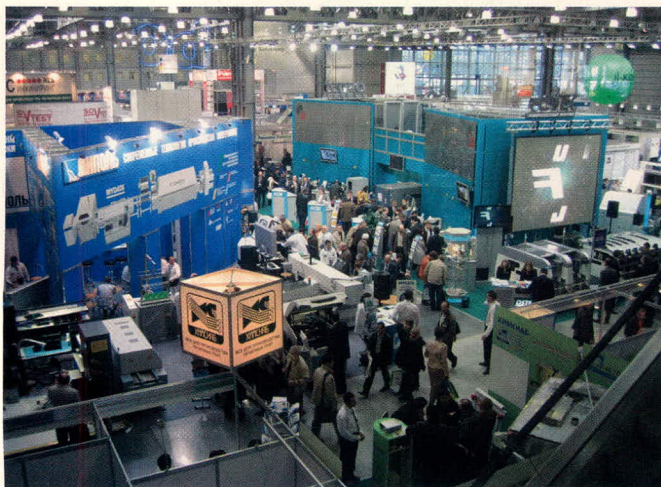
В этом году выставка «ЭкспоЭлектроника» отметила свой юбилей. За 10 лет «ЭкспоЭлектроника» стала первой по значению и масштабу отраслевой выставкой электронных компонентов и технологического оборудования в России и Восточной Европе. Выставка постоянно растёт и расширяется. Для сравнения, в 1998 г. общая площадь составляла всего 1430 кв. м, а к 2007 г. выставка увеличилась более чем в 13 раз – до 18 700 кв. м. За десять лет выставку посетили более 150 000 специалистов.

На юбилейной выставке свою продукцию представили около 500 ком-

паний из 21 страны мира. Ведущие страны – производители электронной продукции, такие как Германия, Гонконг, Корея, Тайвань и Китай, были представлены на выставке национальными стендами. Отмечается возросший интерес со стороны ведущих мировых компаний к российскому рынку, в этом году на выставке можно было встретить такие имена, как Panasonic, Analog Devices, Tektronix, Agilent Technologies, Rutronik, MICRON и многие другие. Впервые в выставке приняли участие такие бренды, как Tyco Electronics и Philips/Assembleon.

В рамках деловой программы прошла презентация «Молодая электроника. Инновационные разработки молодых ученых и их коммерческий потенциал», где свои работы представили победители в молодежном научно-инновационном конкурсе «Электроника 2006–2007» в рамках программы Национального проекта «Образование». Тема инноваций была также освещена в рамках конференции «Инновации в электронике», которую организовал журнал «Современная электроника».

Выставка прошла при официальной поддержке Министерства образования и науки Российской Федерации, Федерального фонда развития электронной техники, Управления радиоэлектронной промышленности и систем управления Федерального агентства по промышленности, ОАО «Российская электроника».



Уважаемый читатель!

Число бесплатных подписчиков журнала «Современная электроника» неуклонно растёт, соответственно растёт и число рассылаемых по почте журналов. Безусловно, нам приятно осознавать рост читательского интереса к нашему журналу. Но вместе с тем, всё больше подписчиков сообщают нам о фактах пропажи журнала на почте или из почтового ящика.

Редакция гарантирует только отправку журнала бесплатному подписчику, но не может гарантировать его доставку.

Риск пропажи журнала можно уменьшить. Во-первых, можно обратиться в отдел доставки вашего почтового отделения и оформить получение журнала до востребования. Во-вторых, можно оформить платную подписку на журнал, и в этом случае почта будет нести ответственность за его доставку.

Оформить платную подписку можно в почтовом отделении через агентства «Роспечать» и «Книга-Сервис». Подписаться можно как на 6 месяцев, так и на год.

Преимущества платной подписки:

- подписаться может любой желающий, тогда как бесплатная подписка оформляется только для специалистов в области электроники. Поступающие в редакцию подписные анкеты тщательно обрабатываются, и часть их отсеивается;
- журнал будет гарантированно доставлен, тогда как при бесплатной подписке редакция гарантирует только отправку, но не доставку журнала.

«Роспечать»

по каталогу агентства «Роспечать»

Подписные индексы:

- на полугодие – 72419,
- на год – 81872

Тел.: (495) 101-2550

Факс: (495) 785-1470

«Книга-Сервис»

по объединённому каталогу «Пресса России»

Подписные индексы:

- на полугодие – 43588
- на год – 43589

Тел.: (495) 124-7110

Факс: (495) 719-0822

Оформить платную подписку можно также через альтернативные подписные агентства. Вы можете воспользоваться их услугами, сравнить цены и удобство условий подписки на наше издание.

«Вся пресса»

Тел.: (495) 906-0735/0726

E-mail: allpress@sovintel.ru

«Артос-ГАЛ»

Тел.: (495) 160-5848; 109-0647/0648

«Интер-Почта-2003»

Тел./факс: (495) 500-0060; 580-9580

Красносельское агентство «Союзпечать»

Тел.: (495) 707-1288/1289



Уважаемые читатели!

Редакция журнала «Современная электроника» проводит актуализацию информации о подписчиках журнала.

Для получения бесплатной подписки на журнал «Современная электроника» с №7 2007 г. заполните данную анкету и отправьте её по факсу (495) 232-1653 или по адресу: 119313 Москва, а/я 26.

Анкету можно также заполнить на web-сайте журнала «Современная электроника» <http://www.soel.ru/>.

Обращаем ваше внимание, что редакция оформляет бесплатную подписку только для квалифицированных специалистов, аккуратно и полностью заполнивших анкету.

Для гарантированного получения журнала «Современная электроника» вы можете оформить платную подписку (информация на сайте www.soel.ru).

Поля, отмеченные *, обязательны для заполнения. Можно отмечать несколько пунктов в одном разделе анкеты.

 /

Укажите в этом поле ваш идентификационный номер из двух чисел, который напечатан на адресной наклейке конверта, в котором вы получаете журнал, – это ускорит обработку анкеты

Фамилия, имя, отчество* _____

Организация* _____

Должность* _____

Телефон* _____

E-mail* _____

Отдел _____

Факс* _____

Сайт* _____

Адрес предприятия* _____

Почтовый индекс, город* _____

Район, область* _____

Адрес* _____

Почтовый адрес для доставки журнала «Современная электроника», если он отличается от адреса предприятия:

Почтовый индекс, город: _____

Район, область: _____

Адрес: _____

Тип вашей должности:

- Руководитель/менеджер высшего звена
 Руководитель отдела, группы, участка, ...
 Менеджер по закупкам/снабжению
 Технический руководитель проекта

- Инженер-разработчик
 Инженер по технической поддержке/обслуживанию
 Преподаватель, научный сотрудник, аспирант
 Другой _____

Область деятельности вашей организации:

- Телекоммуникации, связь
 Автомобильная промышленность
 Авиация, космонавтика, ВПК
 Бытовая электроника
 Системы безопасности
 Торговое оборудование
 Энергетика
 Вычислительная техника
 Приборостроение

- Медицина
 Системы отображения информации
 Промышленная автоматизация
 Силовая электроника
 Транспорт
 Автоматизация зданий, умный дом
 Светотехника
 Другое _____

Вид деятельности вашей организации:

- Производство мелкосерийное
 Производство крупносерийное
 Торговля оптовая
 Торговля розничная

- Научные исследования
 Опытно-конструкторские разработки
 Образование
 Другое _____

Вы рекомендуете, принимаете решение о применении или закупаете следующие компоненты:

- Цифровые микросхемы и микропроцессоры
 Аналого-цифровые компоненты
 Пассивные и дискретные компоненты
 Аналоговые компоненты
 Силовые компоненты
 СВЧ-компоненты

- Электромеханические компоненты
 Изделия полупроводниковой светотехники
 Оптоэлектронные компоненты
 Устройства отображения информации
 Компоненты для телекоммуникаций и связи
 Другое _____

Количество сотрудников в вашей организации:

- До 10 чел.
 10 – 50 чел.
 50 – 100 чел.
 Более 100 чел.
 Более 1000 чел.



ЭЛЕКТРОНИКА
КОМПОНЕНТЫ • ОБОРУДОВАНИЕ • ТЕХНОЛОГИИ

ChipEXPO -2007

ОКТАБРЬ 3-5

ВЕДУЩАЯ РОССИЙСКАЯ ВЫСТАВКА
РОССИЯ • МОСКВА • ЭКСПОЦЕНТР

3-5

ОКТАБРЯ

ПРИ ПОДДЕРЖКЕ:



Министерство промышленности
и энергетики Российской Федерации
Министерство экономического развития
и торговли Российской Федерации
Федеральное агентство по промышленности
Департамент науки и промышленной
политики города Москвы
Московская торгово-промышленная палата

ИНФОРМАЦИОННАЯ
ПОДДЕРЖКА



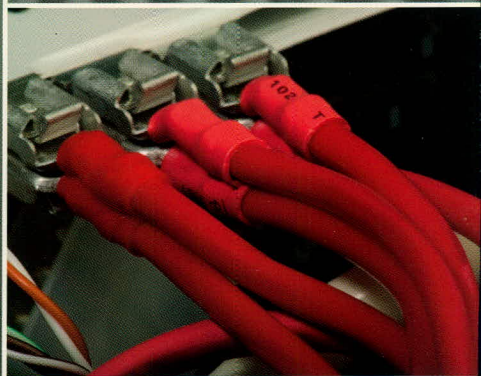
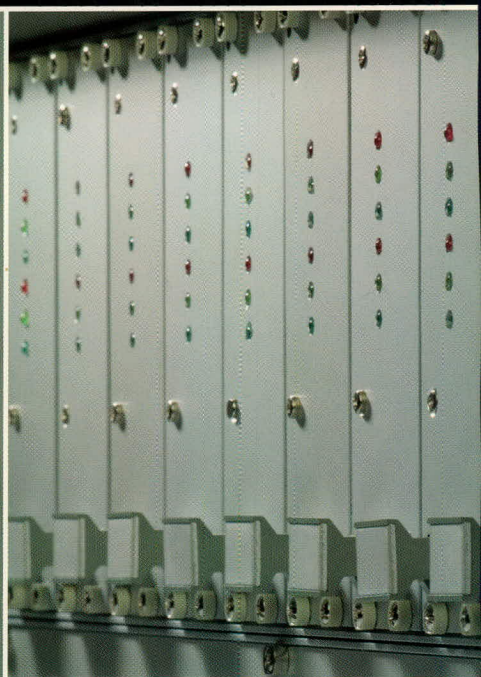
ОРГАНИЗАТОР ВЫСТАВКИ

ЗАО «ЧипЭкспо», Россия,
111141, Москва, ул. Перовская 19/2, стр. 3,
тел./факс: (095) 368-1039, e-mail: info@chipexpo.ru

www.chipexpo.ru

Вычислительные системы на базе ПЛИС Xilinx Virtex II/4/5

**Готовые решения —
здесь и сейчас**



ROSTA

ООО НПО «Роста»

ул. Живописная, 3/1, офис 17
Москва, 123103, Россия
телефон: +7 495 947 9017
факс: +7 495 947 9018
эл. почта: info@rosta.ru
сайт: www.rosta.ru