

УДК 004.045

С. А. Чекмарев, М. Ю. Вергазов, Ф. А. Лукин, В. Х. Ханов, А. В. Шахматов

МОДЕЛИРОВАНИЕ БОРТОВОГО КОМПЬЮТЕРА НА БАЗЕ ОТКРЫТЫХ IP-БЛОКОВ ДЛЯ МАЛЫХ И СВЕРХМАЛЫХ КОСМИЧЕСКИХ АППАРАТОВ*

Представлены результаты моделирования и апробирования на аппаратном отладочном устройстве модели бортового компьютера для малых и сверхмалых космических аппаратов, созданной на основе открытых IP-блоков.

Ключевые слова: малые космические аппараты, бортовой компьютер, IP-блоки, моделирование функциональных частей.

Проектирование малых и сверхмалых космических аппаратов (СМКА) является новым перспективным направлением развития ракетно-космической техники [1]. Эксплуатационные, конструктивные, технологические и другие отличительные признаки СМКА [2] позволяют определить требования к архитектуре бортового компьютера (БК):

– архитектура должна отвечать принципам открытости, базироваться на апробированной в космической практике коммерческой (индустриальной) архитектуре, поддерживать апробированные индустриальные операционные системы и открытые стандартизированные языки программирования;

– архитектура должна соответствовать принципам унификации, являться универсальной для всех типов СМКА (по крайней мере, от мини- до наноспутников) и иметь, ориентировочно, массу на более 1–1,5 кг, потребляемую мощность не более 1,5–3 Вт;

– вычислительные ресурсы БК (производительность, объем памяти) должны быть достаточными для прямого обслуживания аппаратуры полезной нагрузки (по крайней мере, для большинства применений данного БК);

– архитектура БК должна быть реконфигурированной и позволять изменять в зависимости от назначения космического аппарата (КА) набор и характеристики функциональных узлов, а также объем памяти.

Указанным требованиям к БК удовлетворяет концепция «система на кристалле» (СнК). Конфигурируемые СнК реализуют интегральные устройства, объединяющие встроенный процессор, программируемую логику, память и прочие вспомогательные ресурсы и блоки на одном единственном кристалле. Все эти блоки соединяются между собой внутри кристалла с помощью единого интерфейса.

Среди множества процессорных ядер для реализации СнК (Nios II, Cortex-M1, MicroBlaze, PowerPC, RAD 6 000 и др.) особым образом выделяется процессор LEON3, разработанный для применения в космосе [3].

Архитектура процессора открыта, исходные IP-ядра (коммерческого и индустриального назначения) доступны разработчикам. Фирма Aeroflex Gaisler выпускает 2 версии процессоров: LEON3 – открытый

индустриальный процессор общего назначения, и LEON3-FT – сбоеустойчивый процессор, предназначенный для применения в аэрокосмической отрасли и рекомендованный Европейским аэрокосмическим агентством ESA для применения в новых проектах.

Ядро базового процессора (конвейер, контроллеры кэш-памяти и интерфейс АНВ) использует приблизительно 20 000 вентилях (реализация Virtex 4, Xilinx) и может быть реализовано как на специализированных интегральных схемах (ASIC, Application Specific Integrated Circuits), так и по технологии FPGA (Field Programmable Gate Array). В ASIC-реализации можно достигнуть тактовой частоты, превышающей 400 МГц. На FPGA в зависимости от типа ПЛИС и сложности проекта тактовая частота варьируется от 40 до 120 МГц.

Элементной базой реализации проекта БК по технологии СнК являются радиационностойкие программируемые интегральные схемы (ПЛИС) типа FPGA фирм Actel и Xilinx. Для отработки опытных образцов допускается применение ПЛИС FPGA индустриального применения, например, фирмы Altera.

Указанные обстоятельства определили выбор процессорного ядра LEON для разработки СМКА.

Постановка задачи. Архитектура бортового компьютера на базе процессора LEON3 показана на рис. 1. От типичной архитектуры компьютера на базе LEON3 она отличается лишь наличием контроллера шины SpaceWire. Впрочем, компания Aeroflex Gaisler предлагает собственный контроллер SpaceWire, но только лишь в виде закрытого коммерческого IP-блока. Поэтому в данной разработке был использован IP-блок SpaceWire Light [4]. Основное преимущество блока заключается в его полной открытости и доступности для модификации, что позволяет свободно изменять его под нужды проекта. К недостаткам же SpaceWire Light можно отнести малую начальную функциональность, но этот недостаток устраняется соответствующей доработкой IP-ядра. Представленный на рис. 1 вариант сети SpaceWire состоит из четырех узлов 4 x SpW.

Процессор LEON3 является реконфигурируемым: набор его вычислительных узлов можно изменять в процессе его синтеза.

*Работа выполнена в рамках реализации ФЦП «Научные и научно-педагогические кадры инновационной России на 2009–2013 гг.» № П11032 от 27 мая 2010 г.

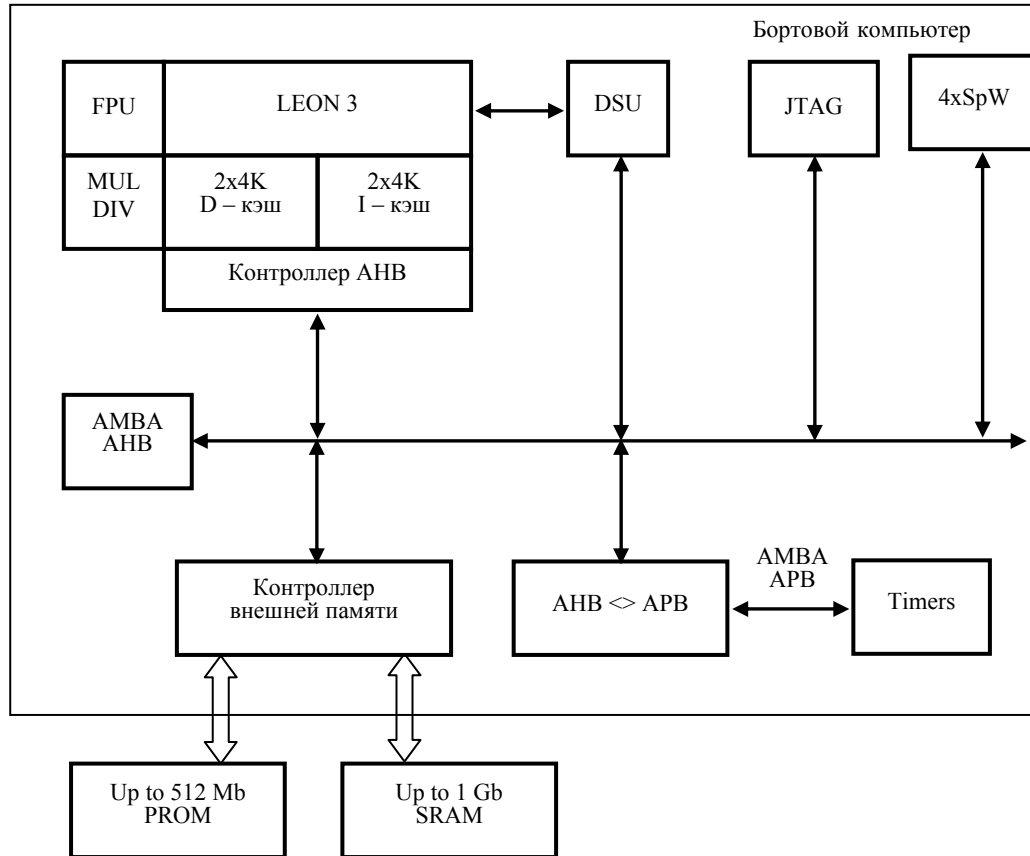


Рис. 1. Структурная схема бортового компьютера

В представленной конфигурации подключены следующие модули: FPU – арифметики с плавающей точкой; MUL/DIV – аппаратного умножения (деления); DSU – отладчик процессора фирмы Aeroflex Gaisler. Поскольку каждый блок процессора представлен отдельным IP-блоком, можно заменить отдельные элементы, не изменяя общей структуры.

Шина AMBA имеет 2 вида: скоростную AHB и низкоскоростную APB. К шине AHB подключаются высокоскоростные устройства, к шине APB – низкоскоростные.

Таймеры Timers служат для организации бортовой шкалы времени. К шине AMBA подключен контроллер внешней памяти, который поддерживает работу с постоянным запоминающим устройством (ПЗУ) и оперативным запоминающим устройством (ОЗУ). В ПЗУ хранится программный код функционирования БК для процессора LEON3 и блок начальной инициализации после включения питания. ОЗУ служит для хранения данных и кода выполняемых процессов. Размер ПЗУ и ОЗУ может варьироваться для конкретного КА, в зависимости от функциональных и программных требований.

Учитывая сложность разработки, множество начальных неопределенностей, была определена следующая стратегия (этапность) ее проведения:

– конечная цель (третий, заключительный этап) – разработка прототипного варианта бортового компьютера СМКА с учетом особенностей космической эксплуатации;

– промежуточная цель (второй этап) – разработка инженерного образца на основе открытых IP-блоков и элементной базы промышленного назначения;

– начальная цель (первый этап) – разработка компьютерной модели бортового компьютера и перенос ее (апробирование) на аппаратный прототип на базе программно-аппаратного отладочного комплекса.

В настоящий момент основные работы первого этапа завершены. Его основные результаты представлены в данной работе.

Инструменты и последовательность моделирования. Инструменты разработки можно разделить на программные для создания компьютерных моделей отдельных устройств и в целом БК и аппаратные для апробирования моделей в отладочных комплексах.

Основной средой моделирования является Mentor Graphics ModelSim. ModelSim – система цифрового моделирования проектов на основе VHDL, Verilog HDL и «смешанных» описаний, позволяющая осуществлять синтез, симуляцию, отладку и анализ цифровых устройств. Она имеет встроенные функции по анализу покрытия кода, визуализации, оптимизации скорости моделирования, трассировке сигналов и др.

Дополнительно использовались следующие среды:

– Xilinx ISE – интегрированная среда разработки, позволяющая скомпилировать vhdl-модель, синтезировать и провести трассировку проекта «система на кристалле» под все семейства ПЛИС корпорации Xilinx;

– Eclipse – свободная интегрированная среда разработки кроссплатформенных приложений; она позволяет писать программы для проектов, основанных на процессоре LEON, в том числе производить отладку процессов, выполняющихся внутри ПЛИС;

– Actel Libero IDE – интегрированная среда разработки, которая включает в себя полный набор инструментов для создания, отладки и верификации, синтеза и физической имплементации проектов, а также генерации файлов прошивки под все семейства ПЛИС корпорации Actel;

– Quartus II – интегрированная среда разработки под семейства ПЛИС фирмы Altera.

Широко применялись программные инструменты компании Aeroflex Gaisler: TSIM, GRMON, GRLIB.

GRMON – общий отладочный монитор для процессора LEON и для проектов типа «система на кристалле». GRMON включает следующие функции и инструменты: чтение (запись) всех системных регистров и памяти; встроенный дизассемблер; загрузку и выполнение приложений на процессоре LEON; дистанционный отладчик GNU (GDB); поддерживает USB, JTAG, RS232, PCI, Ethernet, SpaceWire интерфейсы.

TSIM – программа, эмулирующая архитектуру процессора LEON. Она предназначена для быстрой отладки программ, разработанных для выполнения под архитектурой SPARC v8.

GRLIB – библиотека, которая включает в себя набор открытых IP-блоков, написанных на языке VHDL (процессор LEON3, шина AMBA, интерфейсы ввода/вывода (RS-232, Ethernet и др.)), а также тесты для проверки их работоспособности [5].

В качестве операционной системы БК использовалась RTEMS (Real-Time Executive for Multiprocessor Systems) – операционная система реального времени с открытым исходным кодом, разработанная для встраиваемых систем.

Аппаратное обеспечение проекта представлено следующими устройствами:

1. Плата GR-CPIC-XC4V на базе ПЛИС Virtex4 фирмы Xilinx, входящая в комплекс GR RASTA [6] (Aeroflex Gaisler). GR-CPIC-XC4V поставляется с предустановленным БК на процессоре LEON3 в типовой конфигурации. Аппаратные ресурсы GR-CPIC-XC4V полностью соответствуют структуре БК, показанной на рис. 1.

2. DFPGA-02 на базе ПЛИС Cyclone III фирмы Altera (разработка СибГАУ). Отладочная плата DFPGA-02 предназначена для отработки вопросов практической реализации различных алгоритмов цифровой обработки сигналов. Данная плата позволяет проводить отладку IP-ядра в ПЛИС и передавать необходимую отладочную информацию непосредственно на ПК по интерфейсу USB.

3. Actel ProASIC3 Starter Kit на базе ПЛИС A3P250-PQ208 фирмы Actel. Плата для отладки проектов на базе ПЛИС семейства Actel ProASIC3. Конструкция платы позволяет получить полный доступ ко всем выводам и внутренним ресурсам ПЛИС. Под-

держивается LVDS, что позволяет проводить разработку и отладку проектов, использующих SpaceWire.

Как следует из описания используемых аппаратных отладочных устройств, только GR-CPIC-XC4V можно рассматривать в качестве полнофункционального отладчика для решения задач данного проекта. Остальные два устройства можно использовать только для апробирования отдельных моделей составных частей БК: DFPGA-02 использовался для отладки ядра процессора LEON3, а Actel ProASIC3 Starter Kit – для отработки модели контроллера SpaceWire. Но применение такого разнообразия отладчиков на разных типах ПЛИС вполне оправданно по следующим причинам: возможность параллельной отработки отдельных частей проекта разными разработчиками; изучение сред и технологий проектирования для ПЛИС, которые будут использоваться для разработки инженерного, а затем и прототипного варианта БК (ПЛИС фирмы Actel).

Важным принципом отработки моделей являлся непрерывный перенос отлаживаемых компьютерных моделей отдельных частей БК в указанные аппаратные отладчики. Данный подход (ранняя интеграция на аппаратную платформу) позволяет выявлять ошибки синтеза моделей и способствует оптимизации кода моделей устройств БК по количеству занимаемых логических элементов. Таким образом, процесс разработки моделей и их аппаратного апробирования был неразрывен.

Хотя, как уже отмечалось выше, часть работ по моделированию и апробированию в аппаратных отладчиках проводилась параллельно, с целью согласованной интеграции в целом модели БК на GR-CPIC-XC4V была определена следующая последовательность работ и отработки моделей.

1. Выполнить комплекс ознакомительных тестов с применением предустановленного варианта БК, поставляемого на отладочном устройстве GR-CPIC-XC4V. Предустановленный вариант БК включает полный набор IP-блоков, соответствующий рис. 1. Часть IP-блоков являются открытыми, часть, например SpaceWire, – закрытыми. Минимальный набор тестов должен состоять из тестов ядра процессора LEON3, тестов шины AMBA, тестов памяти, тестов контроллера SpaceWire. Отдельной большой частью работы являлась загрузка в модель БК многозадачной операционной системы RTEMS.

2. Произвести выбор открытых IP-блоков – кандидатов для включения в БК СМК; произвести их автономное моделирование и моделирование во взаимосвязи с другими элементами БК; выполнить апробирование полученных моделей с помощью имеющихся аппаратных отладочных устройств.

3. Выполнить заключительную интеграцию модели БК, состоящей полностью из открытых IP-блоков; перенести модель БК на аппаратную платформу GR-CPIC-XC4V; провести автономное тестирование моделей как составных частей БК; выполнить загрузку операционной системы; провести испытания взаимодействия БК с внешними устройствами с использованием интерфейса SpaceWire.

Результаты моделирования. Задача выполнения ознакомительных тестов БК, предустановленного в GR-CPCI-XC4V, включала следующие виды работ: проверку выполнения программ в БК; проверку загрузки операционной системы в БК; проверку выполнения программы, находящейся под управлением операционной системы; проверку внешних интерфейсов БК – UART и SpaceWire.

Опишем несколько из проведенных тестов, например, тест взаимодействия процессора LEON3 с внешней памятью. Отладчик GR-CPCI-XC4V имеет внешнюю FLASH-память. Задача состояла в загрузке с помощью GRMON в память отладчика тестовой программы, выполняемой под управлением операционной системы RTEMS. Результаты теста через UART интерфейс GR-CPCI-XC4V отправлялись на внешний компьютер.

Загрузка RTEMS обеспечивалась загрузчиком U-BOOT, скомпилированным специально под выполнение процессором LEON на отладчике GR-CPCI-XC4V. С помощью GRMON тестовая программа была размещена во внешней FLASH-памяти. Листинг команд загрузки представлен ниже:

```
grmon> flash unlock all
grmon> flash erase all
grmon> flash lo u-boot
grmon> flash lo rtems-tasks.uimg.o
grmon> run
```

Таким образом, в данном тесте была проведена проверка взаимодействия FLASH-памяти и процессора LEON3. Одновременно проведена проверка запуска на выполнение операционной системы RTEMS.

Представим результаты тестирования интерфейса SpaceWire. Задача тестирования заключалась в передаче данных по физическому кабелю между двумя портами SpaceWire, отконфигурированных под закрытый IP-блок grspw из библиотеки GRLIB. Скомпилированная тестовая программа «grspw-demo» предназначена для управления операционной системой RTEMS.

Результаты теста «grspw-demo» показаны рис 2. Контроль переданной и принятой информации осу-

ществлялся с помощью DSU-интерфейса. Таким образом, в результате выполнения программы «grspw-demo» удалось осуществить инициализацию ядра grspw, осуществить передачу данных с одного интерфейса SpaceWire на другой.

При выборе открытых IP-блоков для включения в БК в основном предполагалось использовать комплект открытых IP-блоков, входящих в библиотеку GRLIB. Так что выбор элементов для построения типового БК решался изначально. Основной задачей являлся выбор IP-блока для реализации SpaceWire.

Критерии выбора IP-ядра SpaceWire следующие: IP-ядро должно быть бесплатным, свободнораспространяемым; сходное VHDL-описание должно быть открыто и доступно для модификации; документация по IP-ядру должна быть полной и доступной; организация IP-ядра должна обеспечивать максимальную совместимость с имеющейся платформой для разработки.

В результате наложения данных критериев на совокупность доступных IP-ядер было выбрано IP-ядро SpaceWire Light.

Представим результаты моделирования работы IP-блока SpaceWire Light в ModelSim. Конфигурация БК для теста включала процессор LEON3, ОЗУ ahbram, с размещенной программой-тестом «spwamba_test», шину AMBA и IP-ядро SpaceWire Light, замкнутое само на себя. Таким образом, контролировался процесс приемопередачи сигналов. Пример моделирования представлен на рис. 3.

Здесь, например, сигналы spw_si и spw_di – входной стробирующий сигнал и сигнал данных соответственно, а spw_so и spw_do являются выходными сигналами. Сигнал r_sta_link показывает нам автомат состояния блока. Из состояния 00, что означает выключенное устройство, 01 – состояние «started», 10 – «connecting» и, наконец, 11 – «run».

Всего во время проведения данного теста моделировались ситуации передачи пакетов на скоростях 5, 10 и 20 Mbit/s, осуществлялся контроль их корректного приема.

```
This TSIM evaluation version will expire July 1, 2011

TSIM/LEON3 SPARC simulator, version 2.0.18 (evaluation version)

Copyright (C) 2001, Gaisler Research - all rights reserved.
This software may only be used with a valid license.
For latest updates, go to http://www.gaisler.com/
Comments or bug-reports to tsim@gaisler.com

serial port A on stdin/stdout
allocated 4096 K RAM memory, in 1 bank(s)
allocated 32 M SDRAM memory, in 1 bank
allocated 2048 K ROM memory
icache: 1 * 4 kbytes, 16 bytes/line (4 kbytes total)
dcache: 1 * 4 kbytes, 16 bytes/line (4 kbytes total)
tsim> _
```

Рис. 2. Выполнение программы «grspw-demo»

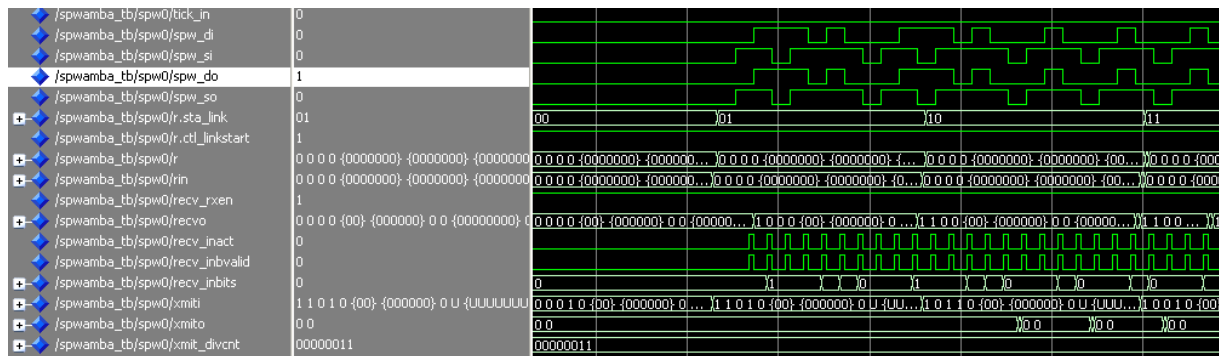


Рис. 3. Пример моделирования SpaceWire Light в ModelSim

Задача моделирования LEON3 – SpaceWire заключалась в подключении IP-ядра SpaceWire Light к шине AMBA, написании собственного SpaceWire-драйвера и создания теста проверки функционирования ядра SpaceWire, включающего, в частности, и физическую передачу данных между двумя SpaceWire-интерфейсами. Средой разработки драйвера и теста являлся Eclipse.

По завершении отладочных приготовлений была произведена загрузка разработанной тестовой программы TestSPW в память БК LEON3 и инициализация выполнения теста. Для проведения теста были использованы два SpaceWire-интерфейса, соединенные между собой соответствующим кабелем. Таким образом, одновременно можно контролировать как передачу, так и прием данных по каналу SpaceWire. Первые строки листинга результатов теста представлены ниже:

```
# ----- SPWDRIVERTEST -----
#
# Found SPWAMBA core 1 at 0x80000400, irq=10.
# Found SPWAMBA core 2 at 0x80000a00, irq=.11
# default APB register values      [OK]
# read/write access to APB registers [OK]
# reset of APB registers           [OK]
# SpaceWire link up                [OK]
# SpaceWire link down              [OK]
...
```

Таким образом, проведенный тест показал следующие результаты: определены оба IP-блока на шине AMBA; управляемо осуществлены инициализация и разрыв соединения; сформированы пакеты данных и отправлены с одного SPW-блока на другой; осуществлен контроль принимаемых и отправляемых пакетов.

Задача аппаратного апробирования проекта БКУ на аппаратной платформе DFPGA-02 на базе ПЛИС Cyclone III фирмы Altera разбивается на две части: моделирование БКУ в программе ModelSim; синтез проекта БК под ПЛИС Altera EP3C25.

Часть листинга результатов моделирования представлена ниже. В нем даны все составные части БК, включенные проект, его основные характеристики (тактовая частота, объем памяти), а также результат выполнения тестовой программы. Все включенные в проект БК IP-блоки из библиотеки GRLIB являются открытыми:

```
# LEON3 Altera EP3C25 SSRAM/DDR Demonstration
design
# GRLIB Version 1.1.0, build 4102
# Target technology: cycloneiii, memory library:
cycloneiii
# ahbctrl: AHB arbiter/multiplexer rev 1
# ahbctrl: Common I/O area at 0xffff0000, 1 Mbyte
# ahbctrl: AHB masters: 2, AHB slaves: 8
# ahbctrl: Configuration area at 0xfffff000, 4 kbyte
# ahbctrl: mst0: Gaisler Research   Leon3 SPARC V8
Processor
# ahbctrl: mst1: Gaisler Research   JTAG Debug Link
# ahbctrl: slv1: Gaisler Research   AHB/APB Bridge
# ahbctrl:   memory at 0x80000000, size 1 Mbyte
# ahbctrl: slv6: Gaisler Research   Generic AHB ROM
# ahbctrl:   memory at 0x00000000, size 1 Mbyte,
cacheable, prefetch
# ahbctrl: slv7: Gaisler Research   Single-port AHB
SRAM module
# ahbctrl:   memory at 0x40000000, size 1 Mbyte,
cacheable, prefetch
...
# leon3_0: LEON3 SPARC V8 processor rev 0
# leon3_0: icache 2*2 kbyte, dcache 2*2 kbyte
# clkgen_cycloneiii: altpII sdram/pci clock generator,
version 1
# clkgen_cycloneiii: Frequency 50000 KHz, PLL scaler
10/10
# MkProm LEON boot loader v2.0
# Copyright Gaisler Research – all right reserved
# system clock : 50.0 MHz
# baud rate : 19171 baud
# prom : 512 K, (2/2) ws (r/w)
# sram : 32 K, 1 bank(s), 0/0 ws (r/w)
#
#
# starting test.exe
# Hello world!
```

Результаты синтеза БК в среде Quartus II приведены на рис. 4. Проект содержит 10 466 логических элементов, т. е. 43 % от ресурсов ПЛИС Altera EP3C25.

Задача аппаратного апробирования IP-ядра SpaceWire Light состояла в перенесении IP-ядра SpaceWire Light на аппаратную платформу Actel ProASIC3 Starter Kit.

Разработан LVDS-интерфейс, адаптированный для IP-ядра SpaceWire Light, само ядро успешно зашито в плату, и проведено предварительное тестирование

функционирования устройства. В частности, были промоделированы и протестированы ситуации установления и поддержания соединения, разрыв и последующее восстановление соединения. В указанных случаях устройство в полной мере обрабатывает описанные стандартом ECSS-E-ST-50-12A функции.

Направление дальнейшей разработки этой ветви проекта предполагает комплексное тестирование SpaceWire, в том числе в связке с коммерческими SpaceWire-устройствами. После осуществления всех доработок и проведения соответствующих испытаний IP-блока полученные данные будут использованы для создания маршрутизирующих коммутаторов и реализации дополнительной функциональности блока (в том числе поддержка протокола удаленного доступа к памяти RMAP).

В результате проведенной работы произведен выбор открытых IP-блоков – кандидатов для включения в БК, разработаны и апробированы на аппаратных отладчиках модели всех основных узлов БК на базе открытых IP-блоков.

В целом можно считать этап моделирования бортового компьютера для малых и сверхмалых космических аппаратов завершенным с положительными ре-

зультатами, позволяющими перейти к разработке его инженерного прототипа.

Библиографические ссылки

1. Космические летательные аппараты. Введение в космическую технику / Ю. Ф. Даниев [и др.] ; под общ. ред. А. Н. Петренко. Днепропетровск : АРТ-ПРЕСС, 2007.
2. Ханов В. Х., Чекмарев С. А., Шахматов А. В. Разработка процессорного модуля бортовой системы управления космическими аппаратами // Проблемы информационной безопасности государства, общества и личности : материалы XII Всерос. науч.-практ. конф. Томск : В-Спектр, 2010. С. 234–240.
3. LEON3 GR-XC3S-1500 Template Design. Gaisler Research, 2006.
4. SpaceWire Light [Электронный ресурс]. URL: <http://www.opencores.org>.
5. GRLIB IP Core User's Manual Version 1.0.16 / J. Gaisler [et al.]. Gaisler Research, 2009. P. 727.
6. Gaisler J., Habinc S. GR-RASTA Board User Manual, GAISLER RESEARCH // PENDER ELECTRONIC DESIGN. 2009. P. 17.

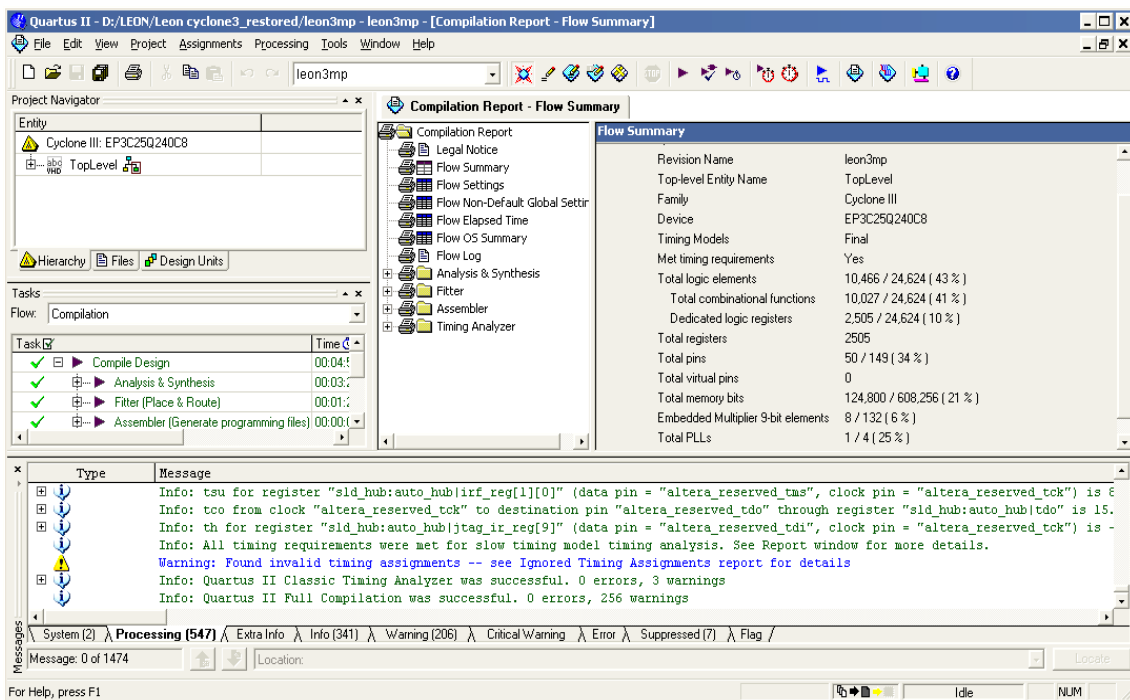


Рис. 4. Проект БК, скомпилированный в среде Quartus II

S. A. Chekmaryov, M. Yu. Vergazov, F. A. Lukin, V. Kh. Khanov, A. V. Shahmatov

SIMULATION OF ONBOARD COMPUTER ON THE BASIS OF OPEN SOURCE IP-CORES FOR SMALL AND ULTRA SMALL SPACE VEHICLES

In this article you will find the results of simulation and testing on the hardware debugging unit of the onboard computer model for small and ultra small space vehicles based on the open source IP-cores.

Keywords: small space vehicles, onboard computing unit, IP-cores, simulation of functional parts.