SmartDesign MSS

Simulation

Microsemi

Table of Contents

1 – Simulation

The SmartFusion Microcontroller Subsystem can be simulated using ModelSim.

MSS simulation is performed using a Bus Functional Model (BFM) strategy. Simulation can be helpful in certain situations, such as:

- Verifying the connectivity and addressing of soft peripherals in the Fabric
- Verifying the External Memory Interface configuration with your vendor's memory
- Verifying ACE behavior

This document describes the simulation support for SmartFusion MSS.

Bus Functional Model

The SmartFusion MSS Cortex M3 processor is modeled with Actel's AMBA Bus Functional Model (BFM). Refer to Actel's DirectCore AMBA BFM User's Guide (PDF) for details on the supported instructions and syntax of the BFM.

Peripherals and Behaviors

To minimize simulation time, certain peripherals in the SmartFusion MSS do not have full behavioral models. Instead they are replaced with memory models that will output a message indicating when the memory locations inside the peripheral have been accessed. This means that the peripheral signals will not toggle based on any writes to registers, or react to any signal inputs on the protocol pins. The peripherals that fall into this group include:

- UART
- SPI
- I2C
- MAC
- PDMA
- WatchDog
- Timer
- RTC

The peripherals that have full behavioral models include:

- Clock Management
- eNVM
- External Memory Controller
- ACE
- GPIO
- Fabric Interface Controller
- eFROM
- AHB Bus Matrix

The eNVM simulation model will not be initialized with data storage or initialization client data. The eSRAM and eNVM are modeled using 256 x 8 RAMs. If you are using a different size RAM your model will use the 256 x 8 RAM size.

Similarly, eFROM simulation model will not be initialized with region configuration data. You will be able to write and read to both peripherals as memory elements.

Simulation Flow

Figure 1-1 illustrates the hierarchy of a typical MSS design. The MSS component is instantiated in a top level SmartDesign component with fabric peripherals. In this scenario, generation of the MSS component will produce test.bfm and user.bfm files.

Generating the SmartDesign_Top component will produce the subsystem.bfm file.

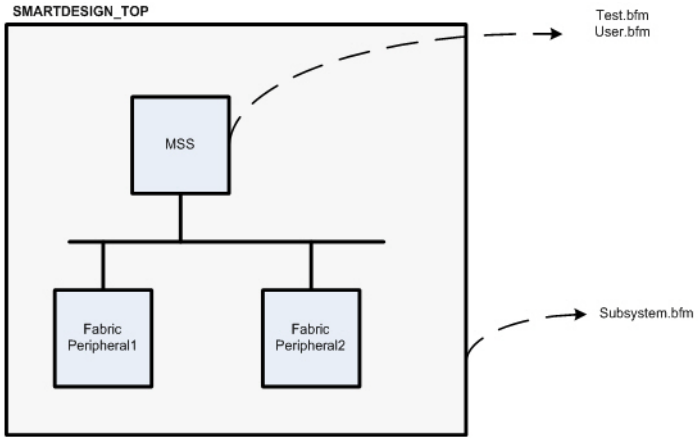

***Figure 1-1* • Typical MSS Design Hierarchy**

- **Test.bfm** - This contains the BFM commands to initialize the simulation model. The BFM commands in this file are generated based upon your MSS configuration. This file is analogous to the system boot code, as it initializes the MSS and calls your user application. Do not modify this file.
- **User.bfm** - You can customize this file to emulate CortexM3 transactions in your system. This contains an include directive to subsystem.bfm that needs to be uncommented if you have any fabric peripherals that you wish to simulate. The memory map of the fabric peripherals is specified inside subsystem.bfm, you can refer to those defines inside this BFM file. This file is analogous to your user application code.
- **Subysystem.bfm** - Contains the fabric memory map. You do not have to modify this file.

These files are automatically passed to ModelSim™ by Libero® IDE, so all you need to do is modify the user.bfm script before running ModelSim. The user.bfm script can be accessed through the File Hierarchy, below your MSS component in the Simulation Files node (as shown in Figure 1-2).

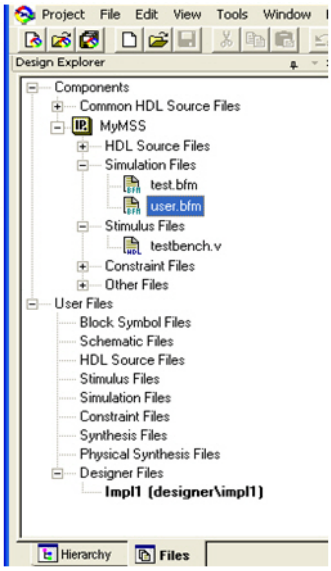

***Figure 1-2* • File Hierarchy - Simulation Files**

2 – BFM Examples

Example 1: Polling ACE Status

In the following example, the ACE status is polled for the completion of calibration and written out to one of the MSS GPIO bits.

user.bfm:

```
# register offsets
constant ADC0_STATUS 0x1000;
constant PPE_FLAGS0  0x1450;
constant PPE_SFFLAGS 0x1460;
constant GPOUT_REG   0x88;

# internal variables
int i data1 flag1;

procedure user_main;

# uncomment the following include if you have soft peripherals in the fabric
# that you want to simulate.  The subsystem.bfm file contains the memory map
# of the soft peripherals.
# include "subsystem.bfm"
# add your BFM commands below:

  // Check for ACE calibration flag
  readstore h ACE ADC0_STATUS data1;
  set flag1 data1 AND 0x8000;
  set data1 flag1 << 5;
  write w GPIO GPOUT_REG data1;

  // Wait until ADC calibration completes
  // driven onto GPIO[31]
  while flag1 == 0x8000
    readstore h ACE ADC0_STATUS data1;
    set flag1 data1 AND 0x8000;
    set data1 flag1 << 16;
    write w GPIO GPOUT_REG data1;
  endwhile
return
```

Example 2: Writing and Verifying Fabric GPIO Bits

In the following example, two soft GPIOs have been added into the Fabric. The subsystem.bfm is automatically generated by the system and contains the memory map of the soft GPIO peripherals. The labels can be referenced from within your user.bfm script.

subsystem.bfm:

```
#============================================================
# Created by Actel SmartDesign Thu Feb 18 09:51:58 2010
#
# Syntax:
# -------
#
# memmap    resource_name base_address;
#
# write     width resource_name byte_offset data;
# read      width resource_name byte_offset;
# readcheck width resource_name byte_offset data;
#
#============================================================

#------------------------------------------------------------
# Memory Map
# Define name and base address of each resource.
#------------------------------------------------------------

memmap CoreGPIO_0 0x40055000;
memmap CoreGPIO_1 0x40057000;
```

The subsystem.bfm file is automatically generated and you do not need to modify it.

user.bfm:

```
procedure user_main;

# uncomment the following include if you have soft peripherals in the fabric
# that you want to simulate.  The subsystem.bfm file contains the memory map
# of the soft peripherals.
include "subsystem.bfm"

# GPIO registers (byte wide)
# Refer Table 3-1 of CoreGPIO_HB.pdf
constant INREG0 0x90
constant OUTREG0 0xA0

print "Fabric GPIO Access Start";

write h CoreGPIO_0 OUTREG0 0x5555;
wait 4;
readcheck h CoreGPIO_0 INREG0 0x5555;
wait 4;

print "CoreGPIO_0 TEST ENDS";

wait 4;
print "CoreGPIO_1 TEST START";

write w CoreGPIO_1 OUTREG0 0xFFFFFFFF;
wait 4;
readcheck w CoreGPIO_1 INREG0 0xFFFFFFFF;
wait 4;

print "CoreGPIO_1 TEST ENDS";
print "";
return
```

A – Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call 800.262.1060
From the rest of the world, call 650.318.4460
Fax, from anywhere in the world, 650.318.8044

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues, and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

Visit the Customer Support website (www.microsemi.com/soc/support/search/default.aspx) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the website.

Website

You can browse a variety of technical and non-technical information on the SoC home page, at www.microsemi.com/soc.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech_itar@microsemi.com. Alternatively, within My Cases, select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

Microsemi Corporation (NASDAQ: MSCC) offers a comprehensive portfolio of semiconductor solutions for: aerospace, defense and security; enterprise and communications; and industrial and alternative energy markets. Products include high-performance, high-reliability analog and RF devices, mixed signal and RF integrated circuits, customizable SoCs, FPGAs, and complete subsystems. Microsemi is headquartered in Aliso Viejo, Calif. Learn more at **www.microsemi.com**.

5-02-00218-0/06.10