SmartFusion cSoC: Achieving Maximum Throughput From FPGA Master to MSS Peripherals

Table of Contents

Introduction

The SmartFusion® customizable system-on-chip (cSoC) FPGA devices integrate FPGA technology with hardened ARM® Cortex™-M3 processor based microcontroller subsystem (MSS) and programmable high-performance analog blocks built on a low power flash semiconductor process. The MSS consists of hardened blocks such as 100MHz ARM Cortex-M3 processor, peripheral DMA (PDMA), embedded nonvolatile memory (eNVM), embedded SRAM (eSRAM), embedded FlashROM (eFROM), external memory controller (EMC), watchdog timer, the Philips Inter-Integrated Circuit (I^2C), serial peripheral interface (SPI), 10/100 Ethernet controller, real-time counter (RTC), GPIO block, fabric interface controller (FIC), in-application programming (IAP) and system registers. The programmable analog block contains analog computing engine (ACE) and analog front-end (AFE) consisting of ADCs, DACs, active bipolar prescalers (ABPS), comparators, current monitors and temperature monitors.

This application note describes how to use the FIC to maximize data transfer between FPGA fabric logic and MSS peripherals. The FIC connects the FPGA fabric logic to MSS. It performs an AHB to AHB or AHB to APB bridging functions between the AHB bus matrix and AHB or APB bus, in the FPGA fabric. The FIC is hard block and hence consumes no logic in the FPGA fabric. It provides two bus interfaces between MSS and FPGA fabric. The first is mastered by MSS and has slaves in the FPGA fabric and the second has a master in the FPGA fabric and slaves in the MSS.

Refer to the *SmartFusion Microcontroller Subsystem User's Guide* for more details on fabric interface controller.

Overview

This application note explains how to use FIC to connect the FPGA fabric logic to the MSS peripherals and how to maximize the data transfer between MSS peripherals and FPGA fabric logic. Figure 1 shows the interface between MSS and FPGA fabric.

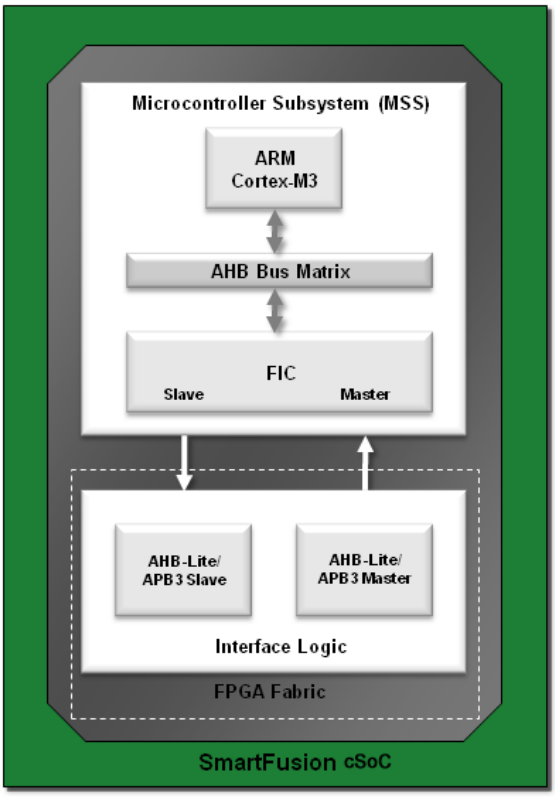

***Figure 1* • FPGA Fabric to FIC Connection**

Design Description

The FPGA fabric logic can be connected to the MSS using AHB or APB bus interface. The FPGA fabric interface controller acts as AHB to AHB-Lite or AHB to APB bridge to interface the AHB communication matrix with FPGA fabric logic.

Refer to *Connecting User Logic to the SmartFusion Microcontroller Subsystem* application note for more details on how to connect the user logic in the FPGA fabric to microcontroller subsystem.

In some scenarios, the FPGA fabric logic needs to access the MSS peripherals like eSRAM or external memory controller (EMC) with very high throughput. In such cases the FPGA fabric logic should be connected to the FIC using AHB-Lite interface. Figure 2 on page 3 shows the FIC configuration with Interface type as AHB-Lite. The FIC now acts as AHB to AHB-Lite bridge without any bus conversion, except for signal mapping. Whereas, in APB interface, the bus conversion from AHB to APB and vice versa are needs to be done.

The FIC configurator provides the options to configure AHB-Lite as Master Interface or Slave Interface or both as shown in Figure 2 on page 3. The AHB-Lite configuration provides ‘Use Bypass Mode’ option to enable or disable the address and data pipelining between FPGA fabric logic and MSS.

In bypass mode (non-pipelined mode/"Use Bypass Mode" option checked), it is possible to achieve zero wait state access between FPGA master and zero-wait state capable MSS slave if there is no other master accessing that slave.

However, in Non-pipelined mode, the setup time requirement of FIC interface is bigger, which may lower overall frequency of operation. Also in Non-pipelined mode, the frequency ratio between MSS clock and FPGA fabric clock is restricted to 1:1.

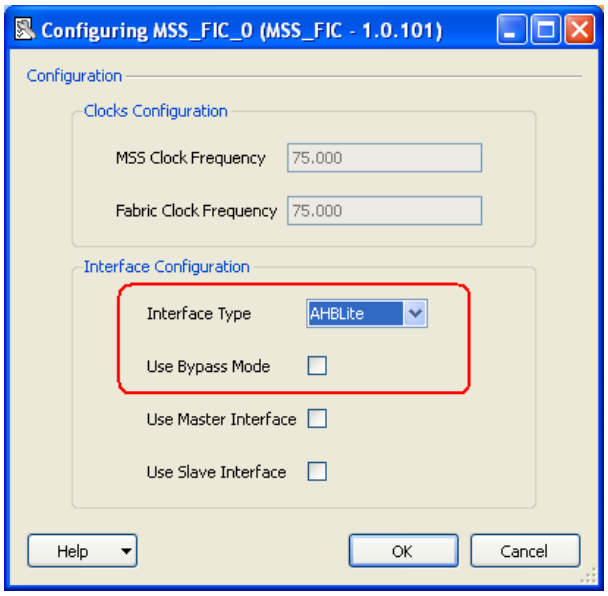

***Figure 2* • FIC Configuration with AHB-Lite Interface**

In Pipelined mode (**Use Bypass Mode** option unchecked/default mode), the interface between AHB bus matrix and FPGA has registered signals that reduce setup requirements. This may improve overall system frequency but these registers introduce a bubble in AHB transaction pipe. It results in inserting a wait-state for each transaction even if Master and Slave are capable of zero-wait state access.

You have to analyze the critical paths between the FIC and the logic in the FPGA fabric, when the **Use Bypass Mode** is enabled. You need to make sure that all the timing requirements have been met between FIC and FPGA fabric logic.

The following sections analyze the write data throughput in Bypass mode and Pipelined mode with example design.

Fabric Master Accessing eSRAM with Pipelined Mode

The example design consists of AHB master in FPGA fabric that writes 32-bit data to 256 locations of eSRAM starting from the address 0x20007FFC. The AHB master logic is connected to the slave interface of the FIC with Pipelined mode. Figure 3 on page 4 shows the FIC configuration with Interface type as AHB-Lite slave and the unchecked **Use Bypass Mode** option.

As mentioned earlier, it is not possible to achieve zero wait transaction due to registered signals in FIC interface. HREADYOUT signal goes low for every transaction as shown in Figure 4 on page 5.

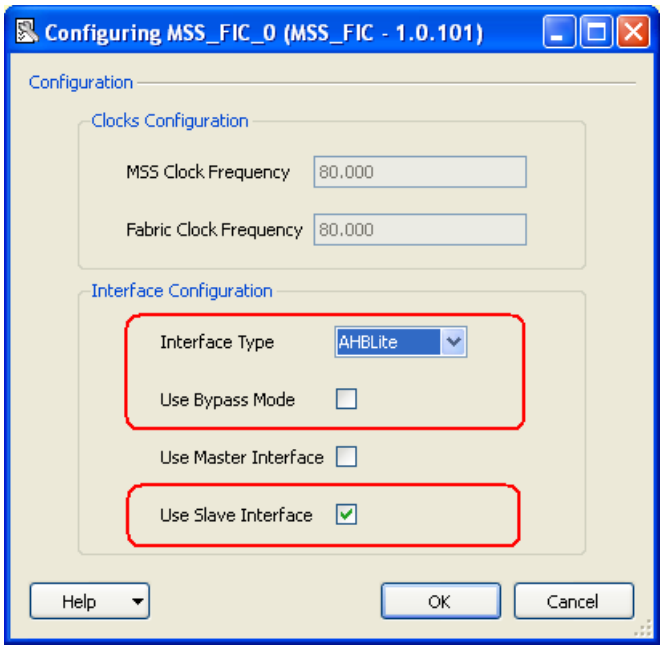

***Figure 3* • FIC Configuration for Pipelined Mode**

It takes multiple clock cycles to complete a single write. Figure 4 shows the AHB master write transactions for single data whereas, Figure 5 on page 6 shows write transactions for 256 data.

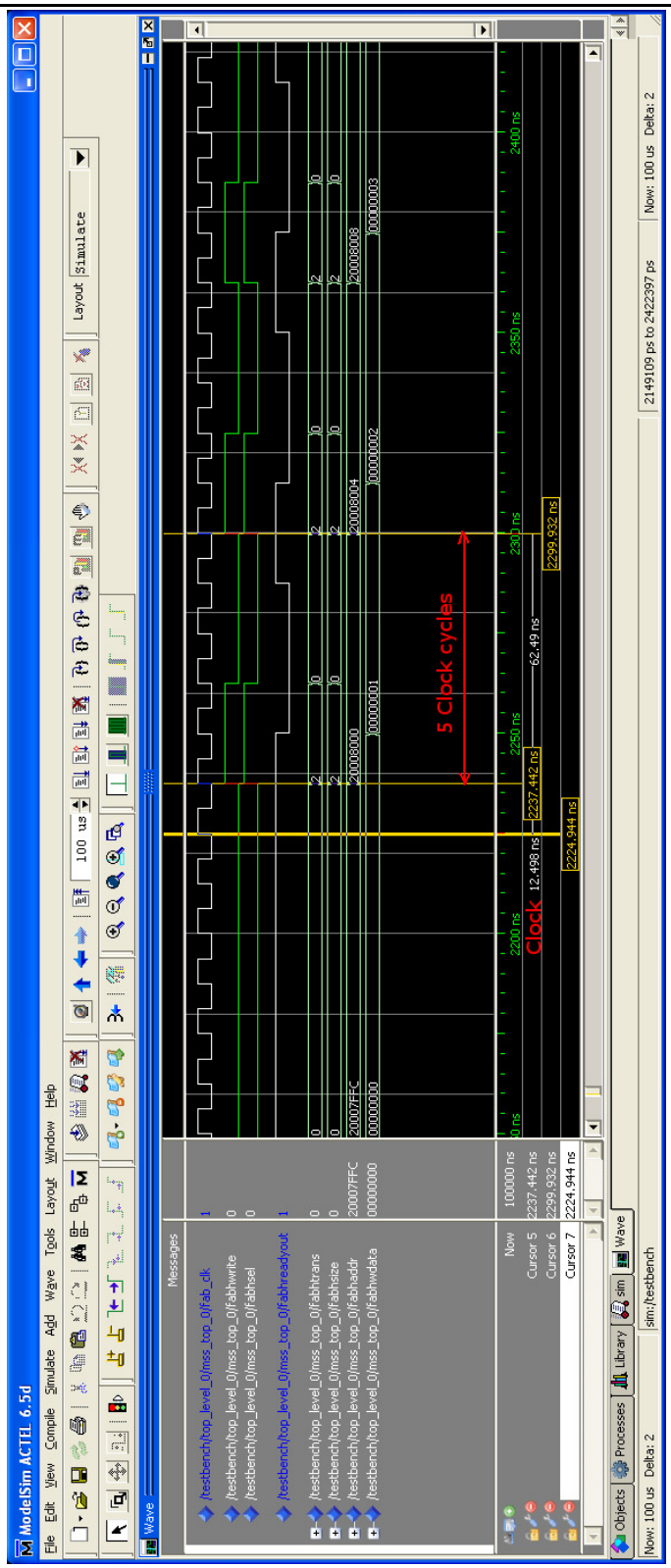

***Figure 4* • AHB Master Write Transactions for Single Data**

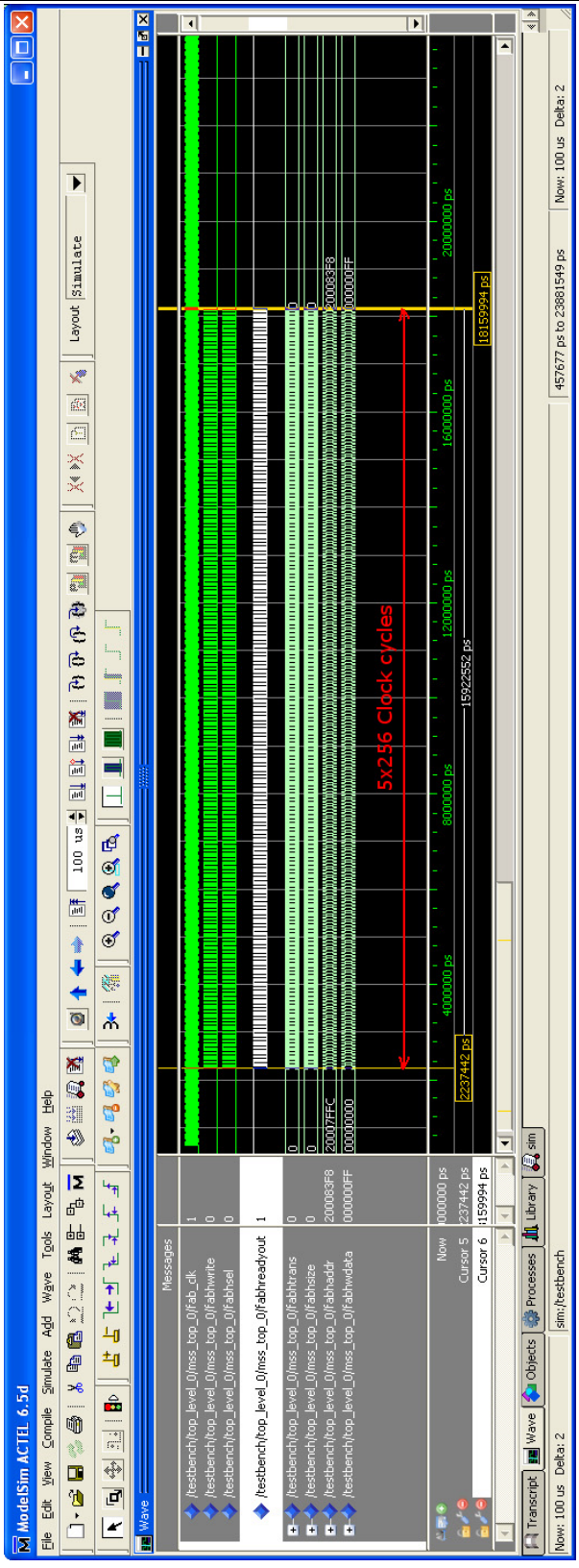

***Figure 5* • AHB Master Write Transactions for 256 Data**

Figure 6 and Figure 7 show the timing paths between MSS and fabric logic. These paths are in FAB (fabric clock) domain.

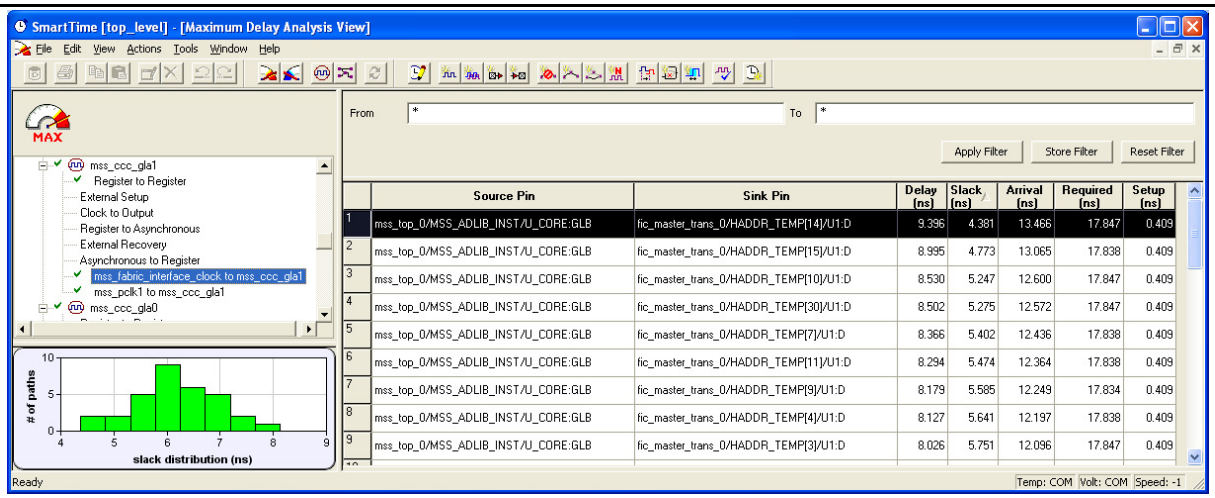

***Figure 6* • Maximum Delay Analysis View**

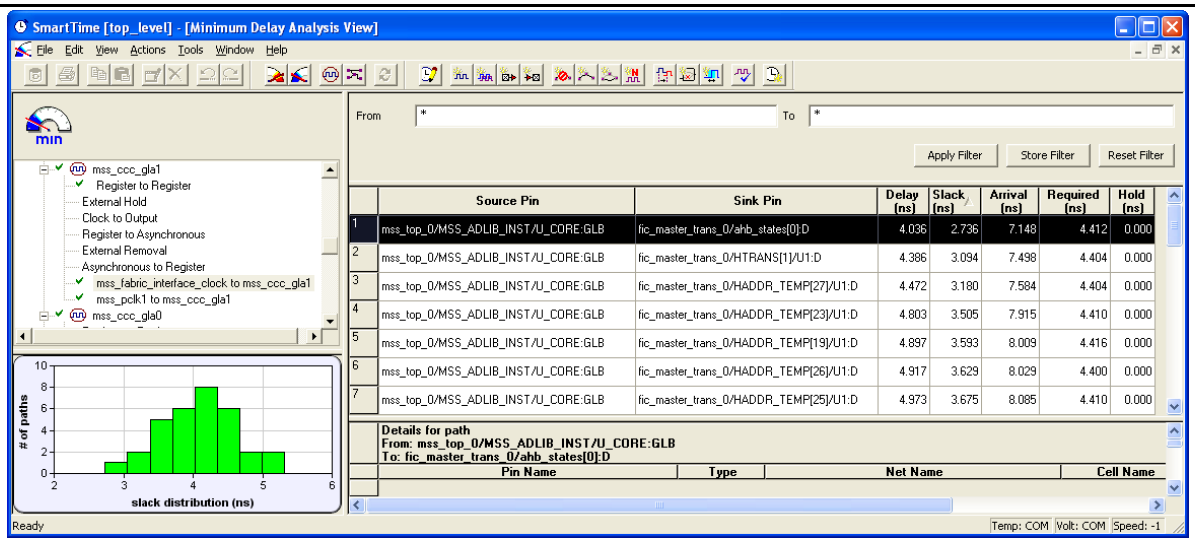

***Figure 7* • Minimum Delay Analysis View**

Fabric Master Accessing eSRAM with Bypass Mode

The example design consists of AHB master in FPGA fabric that writes 32-bit data to 256 locations of eSRAM starting from the address 0x20007FFC. The AHB master logic is connected to the slave interface of FIC with Bypass mode. Figure 8 shows the FIC configuration with interface type as AHB-Lite slave and the **Use Bypass Mode** option selected. As mentioned, it is possible to achieve zero wait state access between the FPGA fabric logic and MSS since there is no register stage between the FPGA fabric and AHB bus matrix. In this configuration, the HREADYOUT signal from AHB bus matrix is always high since no other master is accessing eSRAM and each write takes single clock cycle.

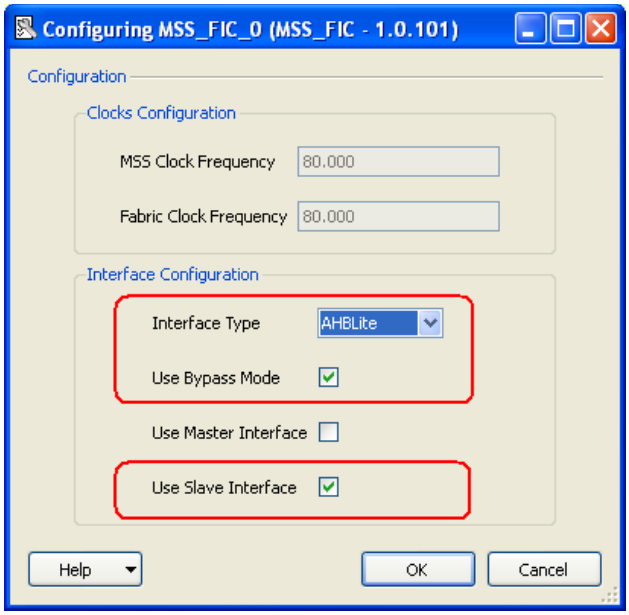

***Figure 8* • FIC Configuration for Bypass Mode**

Figure 9 shows the AHB master write transactions for single data, whereas Figure 10 on page 10 shows write transactions for 256 data.

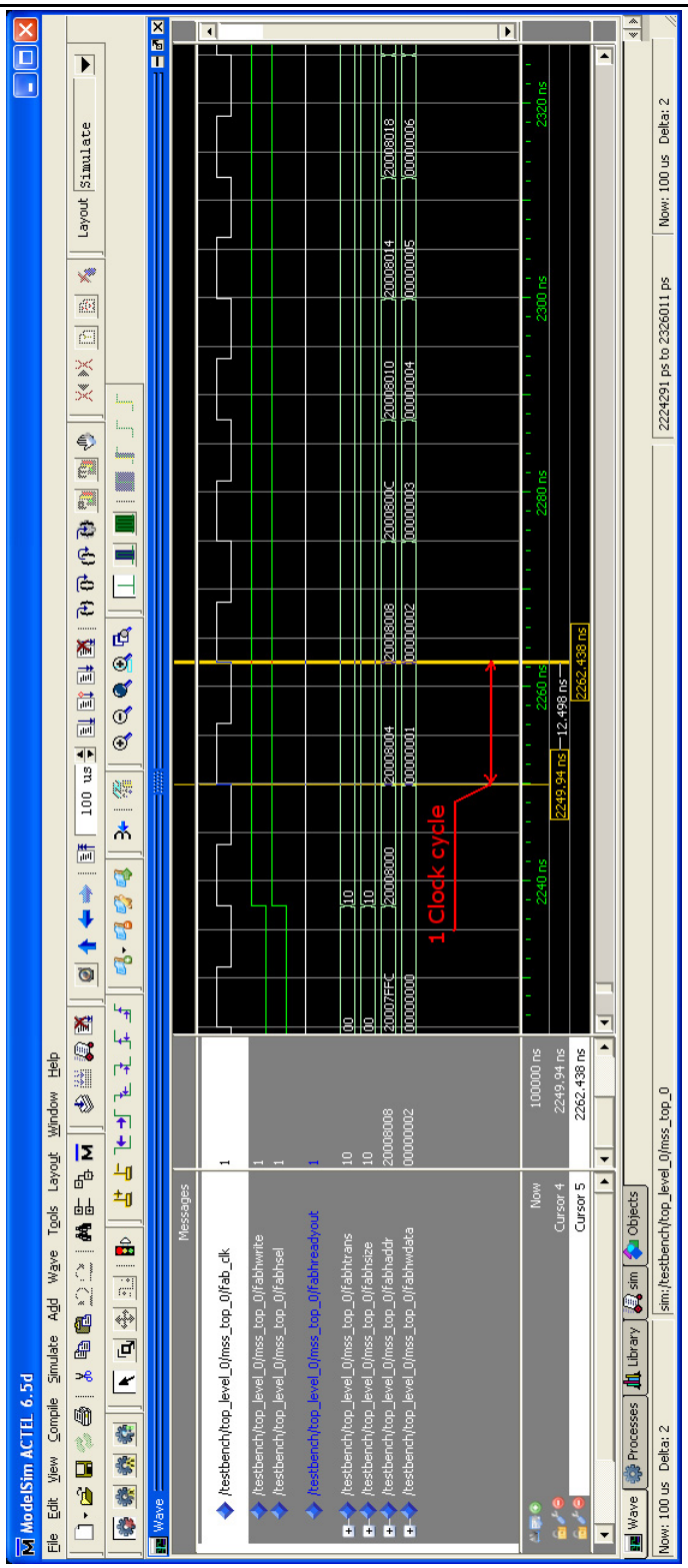

***Figure 9* • AHB Master Write Transactions for Single Data**

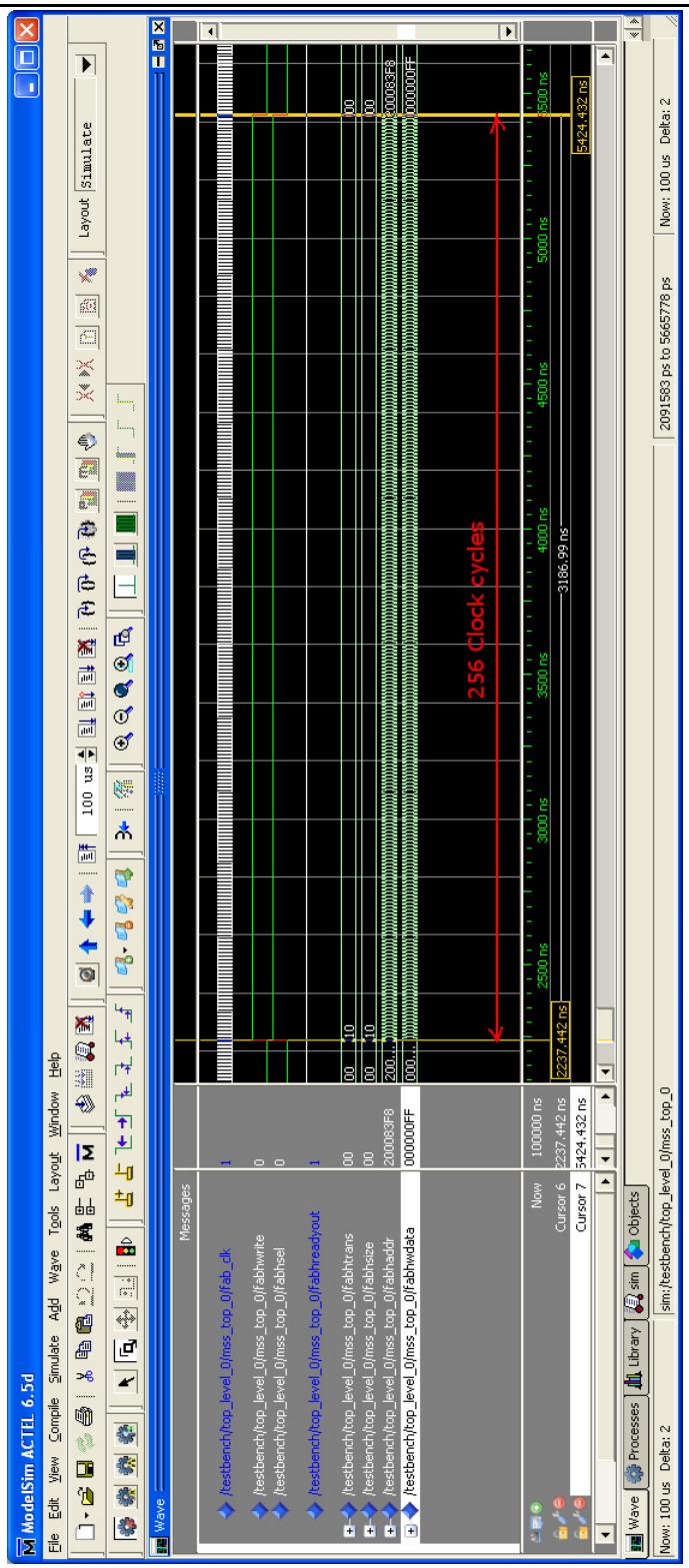

Figure 10 • **FIC Configuration for Bypass Mode**

Figure 11 on page 11 and Figure 12 on page 11 show the timing paths between MSS and fabric logic. These paths are in FAB (fabric clock) domain.

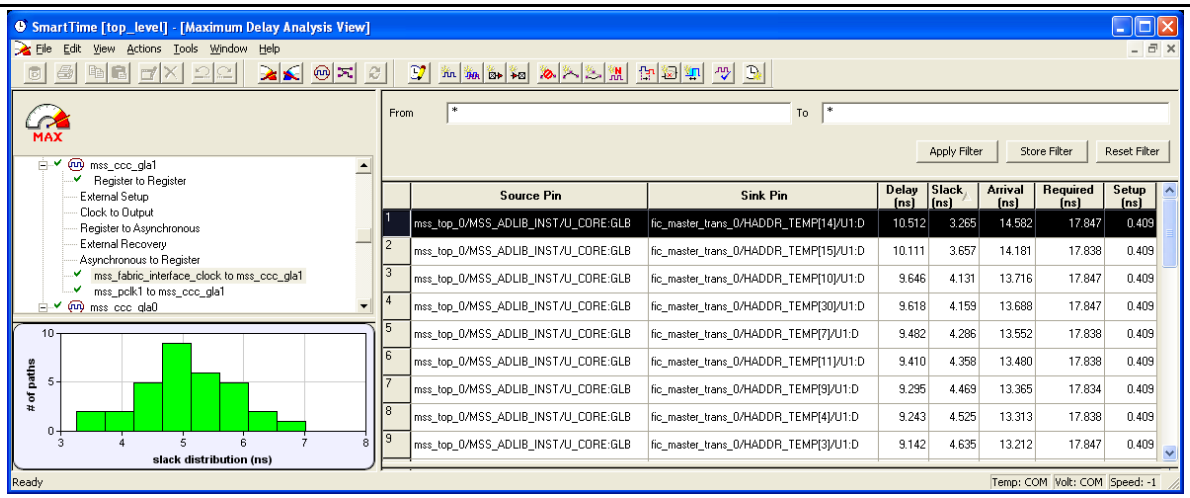

Figure 11 • **Maximum Delay Analysis View**

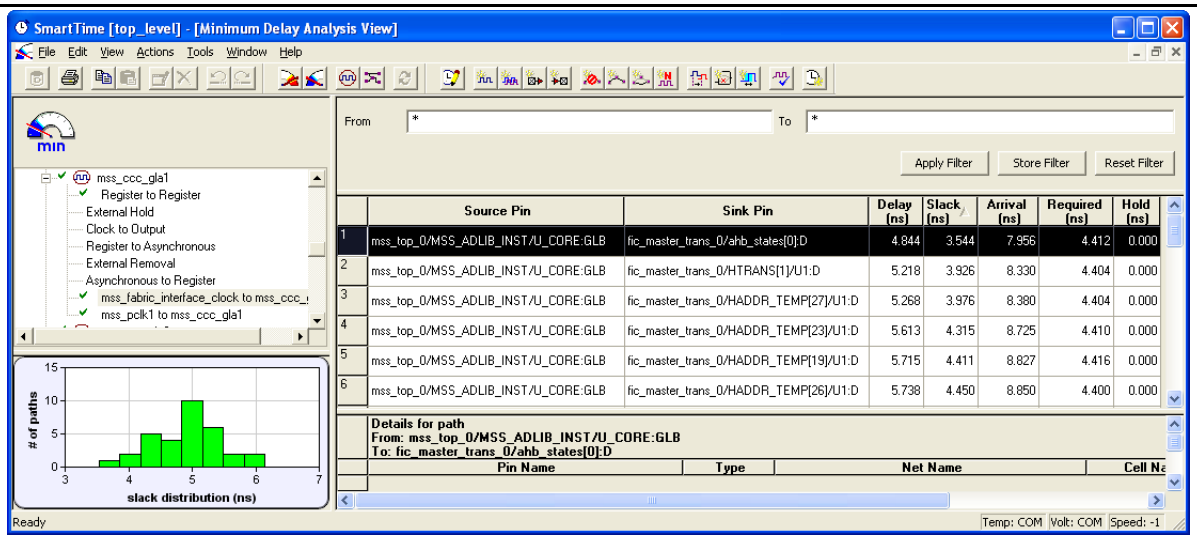

Figure 12 • **Minimum Delay Analysis View**

Table 1 shows the result of the write data throughput analysis in pipelined mode and bypass mode.

Table 1 • **Summary of Write Data Throughput Analysis in Pipelined Mode and Bypass Mode**

Mode	No. of clock cycles required for single data transfer	No. of clock cycles required for 256 data transfer
Pipelined	5	1280
Bypass	1	256

The result shows that the Pipelined mode takes 1024 extra clock cycles to write 32-bit data to 256 locations of eSRAM compared to the bypass mode. Libero® System-on-Chip (SoC) projects are provided in the design files attached with this design example (refer to "Appendix A – Design Files" on page 12).

Conclusion

This application note describes the functionality of **Use Bypass Mode** in the FIC to maximize the data transfer between FPGA fabric logic and MSS peripherals. It also analyzes the write data throughput in Bypass mode and Pipelined mode with example design.

Appendix A – Design Files

You can download the design files from the Microsemi SoC Products Group website: www.microsemi.com/soc/download/rsc/?f=A2F_AC363_DF.

The design file consists of Libero SoC projects. Refer to the Readme.txt file included in the design file for directory structure and description.

List of Changes

The following table lists critical changes that were made in the current version of the chapter.

Revision*	Changes	Page
Revision 4 (February 2012)	Removed ".zip" extension in the Design files link (SAR 36763).	12
Revision 3 (January 2012)	Replaced "Libero IDE" with "Libero SoC" (SAR 35792)	11, 12
Revision 2 (March 2011)	Figure 1 has been replaced with a new figure (SAR 30743)	2
Revision 1 (January 2011)	Modified second row contents of Table 1.	11

*Note: *The revision number is located in the part number after the hyphen. The part number is displayed at the bottom of the last page of the document. The digits following the slash indicate the month and year of publication.*

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**.

51900231-4/02.12