XAPP797 (v2.0) February 7, 2014

# Throughput Performance Measurement

Author: Sanjay Kulkarni, Prasad Gutti, Pankaj Kumbhare

## Summary

This application note demonstrates measurement of the SPI bandwidth by using the SPI flash memory in the Dual and Enhanced Quad modes of the AXI Quad SPI IP core for 1 MB of data. Results were obtained using the Xilinx KC705 evaluation board with Numonyx SPI memory. Measurement can be performed on other boards with a few modifications to the example software files.

### Structure

The reference design systems are built using the Xilinx Vivado® IP integrator provided with Vivado Design Suite: System Edition 2013.4. IP integrator is used to create systems by instantiating the processor, interconnect, interrupt controller, peripheral IP cores, memory controller, and UARTs. The design also includes software built using the Xilinx Software Development Kit (SDK). The software runs on a MicroBlaze™ processor subsystem and implements control, status, and monitoring functions. Complete IP integrator and SDK project files are provided in the reference design zip file to allow examining and rebuilding the design, or as a template for starting a new design.

## Introduction

The AXI Quad SPI core supports Legacy, Enhanced and XIP modes. These three modes are further sub-categorized into three SPI modes, Standard, Dual and Quad mode. Standard mode commands use a single line (IO1), dual mode uses two lines (IO0, IO1) and Quad mode uses four lines (IO0, IO1, IO2, IO3) to exchange data. Legacy mode supports applications that are based on an earlier version of the core (v1_00a). Enhanced mode supports the AXI4 Memory Mapped Interface which provides support for the fixed-burst capability of the transmit and receive FIFOs. Enhanced mode reduces the AXI interface time required to fill or read the DTR or DRR FIFO. These FIFOs are configurable at compile time and can be either 16 or 256 elements deep.

The main focus of this application note is to measure the SPI bandwidth in all modes, where the core is configured in Quad SPI mode with a SPI clock rate of 80 MHz. Table 1 shows the AXI4 interface used for the different modes.

*Table 1:* **AXI Quad SPI Core Configuration Mode AXI4 Interfaces**

| Mode | AXI4-Lite Interface | AXI4 Interface |
|---|---|---|
| Legacy Mode | YES | - |
| Enhanced Mode | - | YES |
| XIP Mode | YES | YES |

One of the three SPI modes is selected based on the type of SPI slave used. Table 2 shows the SPI modes as well as the supported SPI clock frequency and the appropriate I/O interface.

*Table 2:* **SPI Mode, SCK Ratio and I/O Interfaces**

| SPI Modes | SPI Clock Division Ratio wrt. EXT_SPI_CLK | I/O Interface (CS and SCK Always Present) |
|---|---|---|
| Standard | 2, 4, 8, 16, 16xn Where n = 1 … 128 | IO0, IO1 |
| Dual | 2 | IO0, IO1 |
| Quad | 2 | IO0, IO1, IO2, IO3 |

For the highest possible bandwidth on the SPI side, the core should be used in Quad mode where the data transactions use all four lines. The SPI bandwidth is best utilized in this mode because the supported commands are Fast Read Quad Output (`0x6B` h), Fast Read Quad I/O (`0xEB` h), and Quad Input Page Program (`0x32` h), all of which support writing or reading of the SPI flash on four I/O lines.

## Hardware Requirements

The hardware board(s) and other equipment required for these systems are:

- Xilinx KC705 evaluation board (Rev C, D or 1.1)

The software tool requirements for building and downloading this reference system are:

- Vivado IP integrator 2013.4
- Vivado Design Suite: System Edition 2013.4
- SDK 2013.4

## System Throughput Measurement

To examine the system throughput for a 1 MB read from the SPI flash, the core must be configured in Quad mode for optimal performance. Winbond and Numonyx SPI memories now support Dual and Quad mode as the preferred mode over standard mode. Dual mode commands improve the SPI bandwidth by approximately 2x, while the Quad mode commands improve the SPI bandwidth by 4x compared with SPI Standard mode.

With the AXI Quad SPI v3.1 core, the SPI bandwidth is used as lossless bandwidth. This means that as long as data is present in the Transmit data FIFO, the SPI clock is continuous. Use of efficient software coding techniques such as refilling the DTR FIFO when half empty or polling the DTR Occupancy register for empty space ensures that no idle cycles occur between the SPI transactions. These techniques are applicable to Legacy as well as Enhanced Quad modes. The time taken to write or read 1 MB of data is less in Enhanced Quad mode as compared to Legacy Quad mode due to the addition of burst capability. In all of these tests, an 80 MHz SPI clock is used while targeting the Numonyx memory on the KC705 board.

## System Design Block Diagram

The AXI Quad SPI throughput measurement system design is shown in Figure 1.

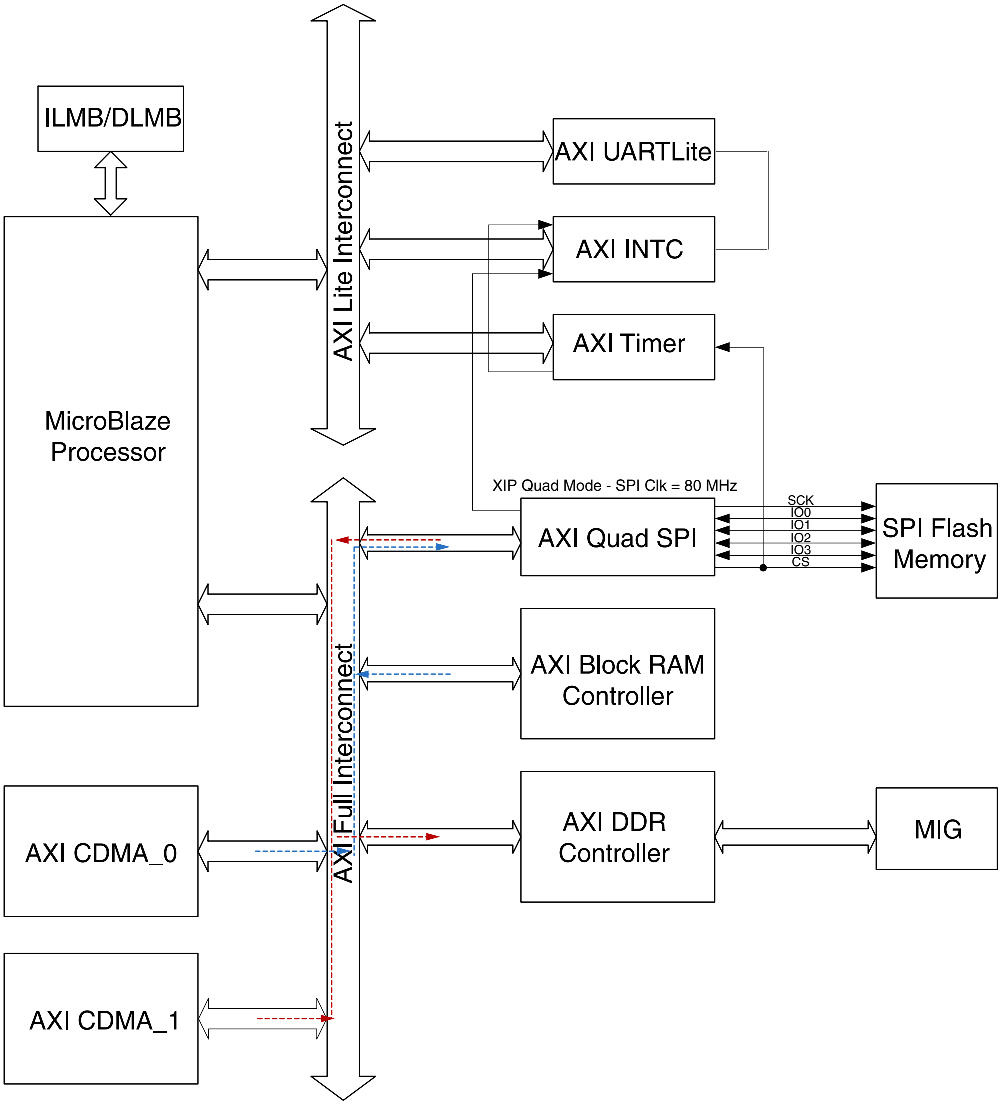


*Figure 1:* **Throughput Measurement System Design - Enhanced Quad Mode**

## Scope

This exercise measures the throughput of the AXI Quad SPI core. Throughput is measured as time taken by the system to write and to read 1 MB of data into and from the SPI flash in Enhanced mode. Table 3 shows the cores, versions and addresses for the system used to demonstrate the performance measurement.

*Table 3:* **AXI Quad SPI - Performance Measurement System Cores and Addresses**

| IP | Version | Base Address | High Address |
|---|---|---|---|
| axi_quad_spi | 3.1 | 0xC4000000 | 0xC400FFFF |
| proc_sys_reset | 5.0 | N/A | N/A |
| axi_intc | 4.1 | 0x41200000 | 0x4120FFFF |
| lmb_bram_if_cntlr | 4.0 | 0x00000000 | 0x00001FFF |
| microblaze (Standalone OS) | 9.2 | N/A | N/A |
| mdm | 3.0 | 0x41400000 | 0x4140FFFF |
| clock_generator | 5.1 | N/A | N/A |
| axi_bram_ctrl | 3.0 | 0xC0000000 | 0xC001FFFF |
| axi_interconnect - lite | 2.1 | N/A | N/A |
| axi_interconnect - full interface | 2.1 | N/A | N/A |
| axi_uartlite | 2.0 | 0x40600000 | 0x4060FFFF |
| mig_7series_0 | 2.0 | 0x80000000 | 0xBFFFFFFF |
| axi_cdma_0 | 4.1 | 0x44A00000 | 0x44A0FFFF |
| axi_cdma_1 | 4.1 | 0x44A10000 | 0x44A1FFFF |
| axi_timer | 2.0 | 0x41C00000 | 0x41C0FFFF |
| util_reduced_logic | 1.00 | N/A | N/A |
| xlconcat_0 | 1.0 | N/A | N/A |

## Throughput Measurement Software Flowchart

The software flowchart in Figure 2 shows the logic for writing 1 MB of data into the SPI flash for measurement of the system throughput with an AXI4 memory-mapped interface.

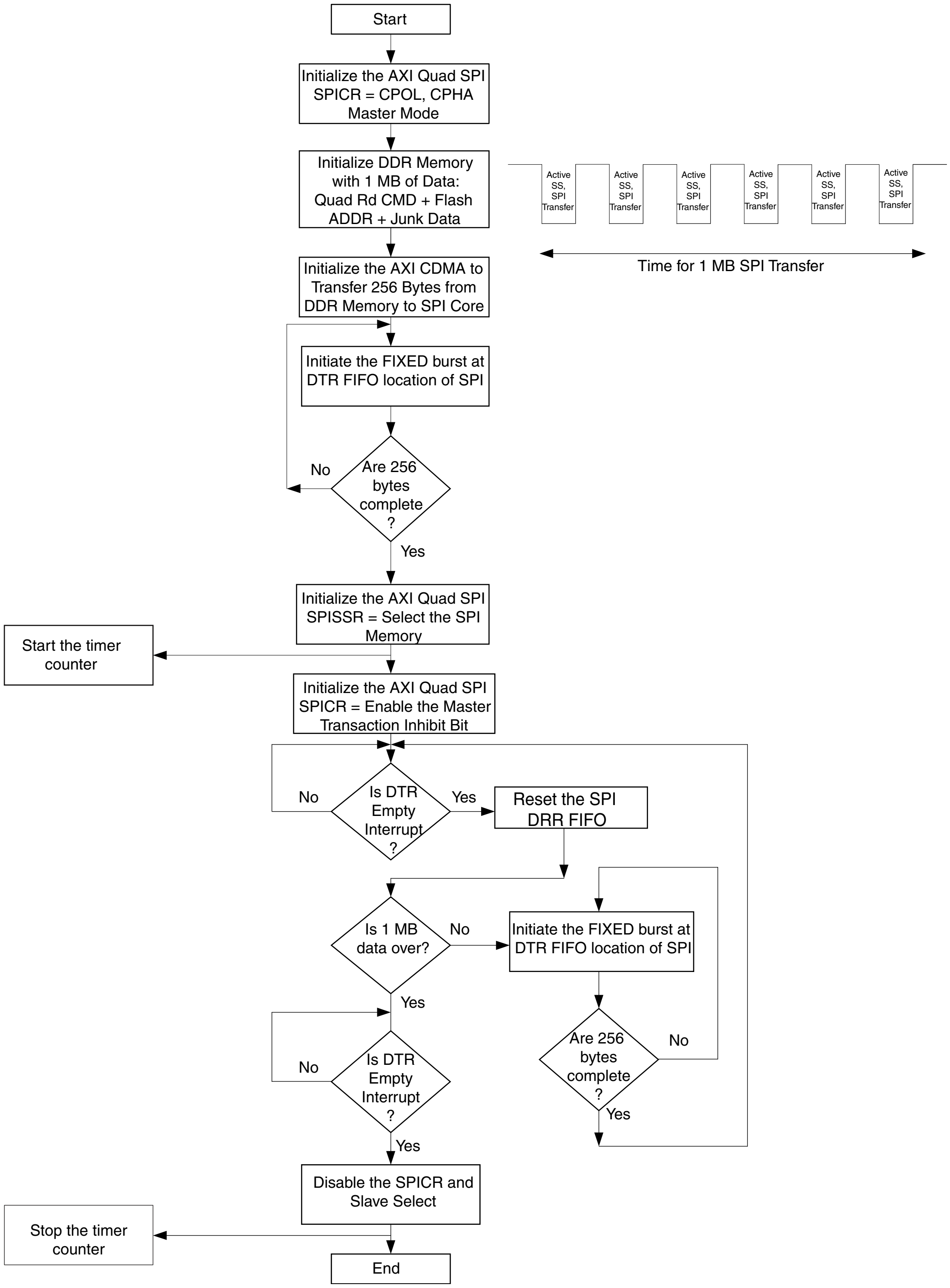

Start
Initialize the AXI Quad SPI SPICR = CPOL, CPHA Master Mode
Initialize DDR Memory with 1 MB of Data: Quad Rd CMD + Flash ADDR + Junk Data
Initialize the AXI CDMA to Transfer 256 Bytes from DDR Memory to SPI Core
Initiate the FIXED burst at DTR FIFO location of SPI
No
Are 256 bytes complete ?
Yes
Initialize the AXI Quad SPI SPISSR = Select the SPI Memory
Start the timer counter
Initialize the AXI Quad SPI SPICR = Enable the Master Transaction Inhibit Bit
No
Is DTR Empty Interrupt ?
Yes
Reset the SPI DRR FIFO
Is 1 MB data over?
No
Initiate the FIXED burst at DTR FIFO location of SPI
Are 256 bytes complete ?
No
Yes
Yes
No
Is DTR Empty Interrupt ?
Yes
Disable the SPICR and Slave Select
Stop the timer counter
End
Active SS, SPI Transfer
Active SS, SPI Transfer
Active SS, SPI Transfer
Active SS, SPI Transfer
Active SS, SPI Transfer
Active SS, SPI Transfer
Time for 1 MB SPI Transfer

*Figure 2:* **Software Flowchart - Enhanced Quad Mode Write Transaction**

Figure 3 shows the software flowchart for reading 1 MB of data from the SPI flash.

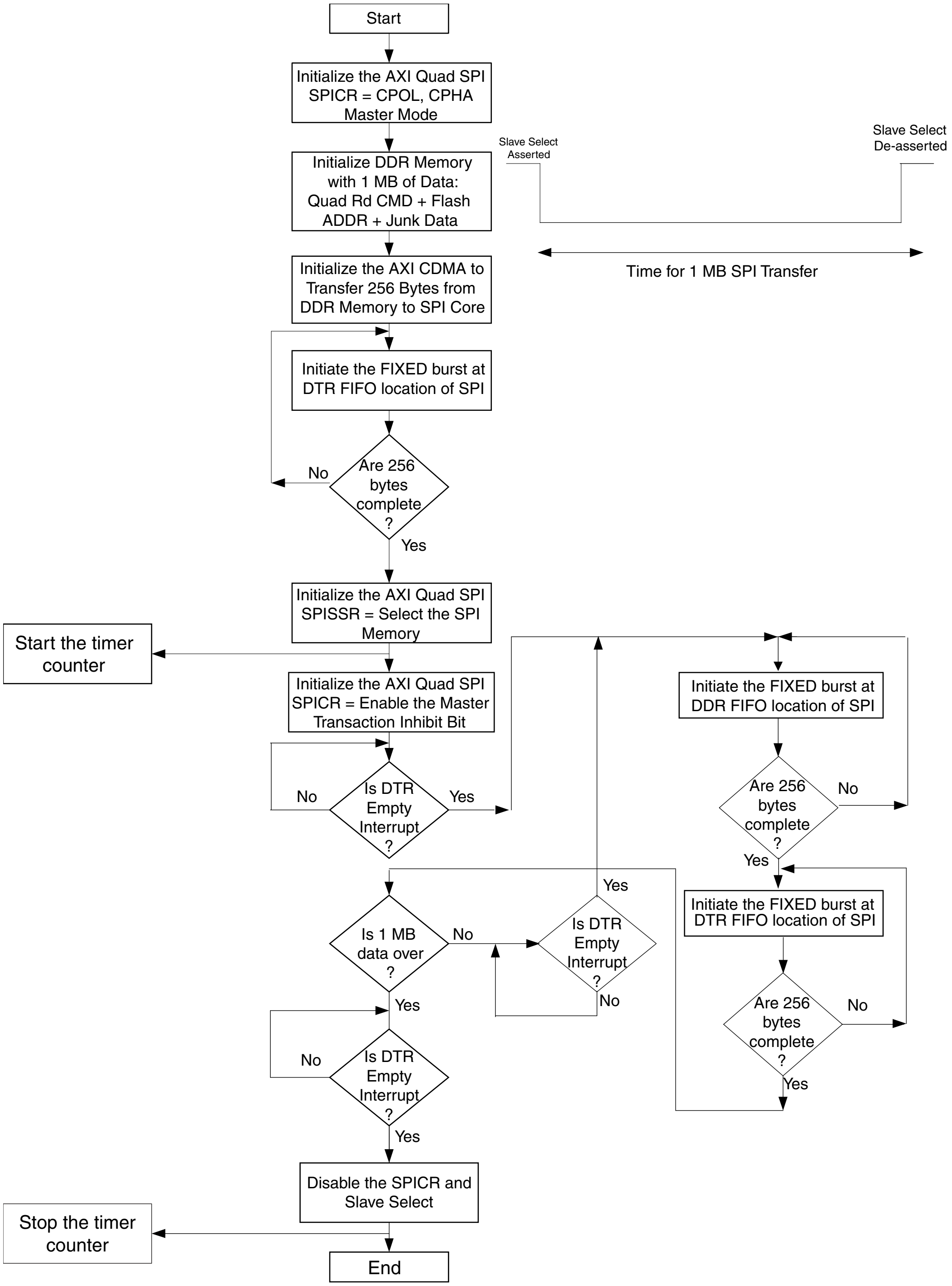

Start
Initialize the AXI Quad SPI SPICR = CPOL, CPHA Master Mode
Initialize DDR Memory with 1 MB of Data: Quad Rd CMD + Flash ADDR + Junk Data
Initialize the AXI CDMA to Transfer 256 Bytes from DDR Memory to SPI Core
Initiate the FIXED burst at DTR FIFO location of SPI
Are 256 bytes complete ?
No
Yes
Initialize the AXI Quad SPI SPISSR = Select the SPI Memory
Start the timer counter
Initialize the AXI Quad SPI SPICR = Enable the Master Transaction Inhibit Bit
Is DTR Empty Interrupt ?
No
Yes
Is 1 MB data over ?
No
Yes
Is DTR Empty Interrupt ?
Yes
No
Is DTR Empty Interrupt ?
No
Yes
Initiate the FIXED burst at DDR FIFO location of SPI
Are 256 bytes complete ?
No
Yes
Initiate the FIXED burst at DTR FIFO location of SPI
Are 256 bytes complete ?
No
Yes
Disable the SPICR and Slave Select
Stop the timer counter
End
Slave Select Asserted
Slave Select De-asserted
Time for 1 MB SPI Transfer

*Figure 3:* **Software Flowchart - Enhanced Quad Mode Read Transaction**

### Write and Read Process Differences

All of the SPI flash devices have an 8-bit interface mode. Only after sending the command, address bytes, and dummy bytes does the actual data transmission start. This data is in 8-bit packet mode. Flash write is restricted to a single page while read has no such restriction. The SPI flash continues to send data to the host while the chip select is asserted and the SPI clock is active. This means that the SPI devices are comparatively slower for the write as opposed to the read process.

## Procedure for Throughput Measurement in Enhanced Quad Mode

1. To connect to the XMD prompt, from the build directory, type:

   ```
   xmd
   ```

2. Configure the FPGA with `system_enh_quad_md.bit` through a JTAG cable using this command at the XMD prompt:

   `fpga -f system_enh_quad_md.bit` (ensure this is the Enhanced Quad mode bit file)

3. To connect to the processor running on the FPGA type:

   ```
   connect mb mdm
   ```

4. Reset and stop the FPGA using these commands at the XMD prompt:

   ```
   rst
   stop
   ```

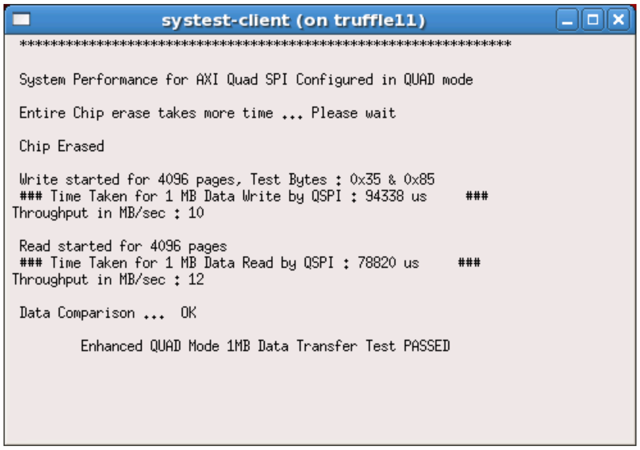
5. Get the ELF file for 1 MB data to write and read from the SPI flash. This file is the output of the software example provided with the reference design ZIP file.
6. To observe the results, open hyperTerminal and configure it to 9600 baud with default configuration. Make sure that the UART cable is connected to the board and the PC.
7. Download the ELF file into memory (block RAM or DDR) and run using these commands:

   ```
   dow kc705_enh_quad_perf.elf
   run
   ```

8. When the test has been completed and displays SUCCESS, PASSED, or FAILED on the hyperTerminal, stop the processor by typing:

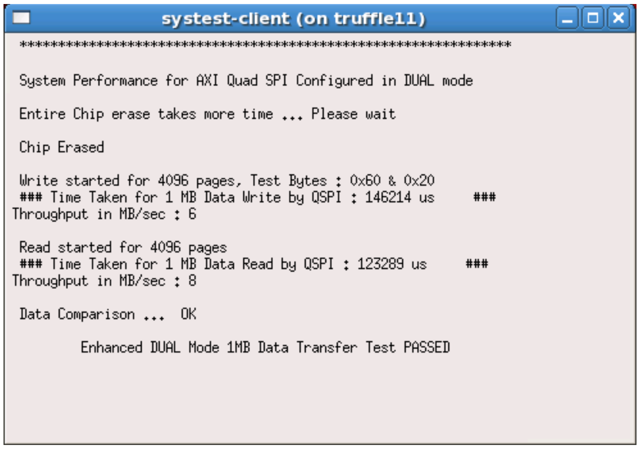
   ```
   stop
   ```

Sample output screens are shown in Figure 4 and Figure 5.



```
 ****************************************************************

 System Performance for AXI Quad SPI Configured in QUAD mode

 Entire Chip erase takes more time ... Please wait

 Chip Erased

 Write started for 4096 pages, Test Bytes : 0x35 & 0x85
 ### Time Taken for 1 MB Data Write by QSPI : 94338 us     ###
Throughput in MB/sec : 10

 Read started for 4096 pages
 ### Time Taken for 1 MB Data Read by QSPI : 78820 us     ###
Throughput in MB/sec : 12

 Data Comparison ...  OK

         Enhanced QUAD Mode 1MB Data Transfer Test PASSED
```

*Figure 4:* **Sample output for Enhanced Quad Mode**



```
 ****************************************************************

 System Performance for AXI Quad SPI Configured in DUAL mode

 Entire Chip erase takes more time ... Please wait

 Chip Erased

 Write started for 4096 pages, Test Bytes : 0x60 & 0x20
 ### Time Taken for 1 MB Data Write by QSPI : 146214 us     ###
Throughput in MB/sec : 6

 Read started for 4096 pages
 ### Time Taken for 1 MB Data Read by QSPI : 123289 us     ###
Throughput in MB/sec : 8

 Data Comparison ...  OK

        Enhanced DUAL Mode 1MB Data Transfer Test PASSED
```

*Figure 5:* **Sample output for Enhanced Dual Mode**

## Hardware System Provided with the Application

This section describes the throughput measurement hardware system provided in this application note.

1. The reference design zip file provides the files for the previously outlined Procedure for Throughput Measurement in Enhanced Quad Mode, page 7. Download the bit file to the FPGA and run `application.elf`.
2. The `xapp797_kc705_enh_quad_md.zip` file contains these folders:
   a. `project_1`: This folder contains complete system files including software for measuring throughput for a 1 MB data transfer.
   b. `ready_for_download`: This folder contains the `system_enh_quad_md.bit` and the `kc705_enh_quad_perf.elf` files. The `system_enh_quad_md.bit` file should be downloaded using the `fpga` XMD command. The `kc705_enh_quad_perf.elf` file should be downloaded using the `dow` XMD command.
3. The `xapp797_kc705_enh_dual_md.zip` file contains these folders:
   a. `project_1`: This folder contains complete system files, including software for measuring throughput for a 1 MB data transfer.
   b. `ready_for_download`: This folder contains the `system_enh_dual_md.bit` and the `kc705_enh_dual_perf.elf` files. The `system_enh_dual_md.bit` file should be downloaded using the `fpga` XMD command. The `kc705_enh_dual_perf.elf` file should be downloaded using the `dow` XMD command.

***Note:*** The write process to SPI flash is slower compared to the read process. This is reflected in the results. The existing results are based on and limited to the system provided in the application note. The speed can vary with other system designs.

## Reference Design Details

Table 4 shows the reference design checklist for this application note.

*Table 4:* **Reference Design**

| Parameters | Description |
|---|---|
| **General** | |
| Developer Name | Sanjay Kulkarni, Prasad Gutti |
| Target devices (stepping level, ES, production, speed grades) | Kintex-7 XC7K325T-2FFG900 |
| Source code provided | Yes |
| Source code format | VHDL/Verilog (Some cores are encrypted) |
| Design uses code/IP from existing Xilinx application note/reference designs, CORE Generator™ software, or third party | Reference designs provided for SDK and cores generated from Vivado IP catalog |
| **Simulation** | |
| Functional simulation performed | N/A |
| Timing simulation performed | N/A |
| Test bench used for functional and timing simulations | N/A |
| Test bench format | N/A |
| Simulator software/version used | N/A |
| SPICE/IBIS simulations | N/A |
| **Implementation** | |
| Synthesis software tools/version used | |
| Implementation software tools/versions used | Vivado Design Suite: System Edition 2013.4 |
| Static timing analysis performed | Yes (Passing timing in PAR/TRCE) |
| **Hardware Verification** | |
| Hardware verified | Yes |
| Hardware platform used for verification | Kintex-7 FPGA KC705 evaluation kit |

The design files for the reference design can be downloaded from https://secure.xilinx.com/webreg/clickthrough.do?cid=314897 and https://secure.xilinx.com/webreg/clickthrough.do?cid=314898, registration required.

## Device Utilization and Performance

Tables 5 and Table 6 show the device resource usage for both setups in this Application Note.

*Table 5:* **Device Utilization in Enhanced Quad Mode**

| Device | Speed Grade | Package | Occupied Slices | Slice LUTs | I/Os | RAMB36 | RAMB18 |
|---|---|---|---|---|---|---|---|
| XC7K325 | -2 | TFFG900 | 36352<br>(5%) | 19287<br>(19%) | 124<br>(21%) | 58<br>(13%) | N/A |

### Throughput Measurement System Utilization in Enhanced Quad Mode

*Table 6:* **Module Level Utilization**

| IP Core | Instance Name | Slices | Slice Registers | LUTs | LUTRAM | Block RAM/ FIFO | DSP Slices | BUFG | BUFR |
|---|---|---|---|---|---|---|---|---|---|
| axi_qspi | axi_quad_spi_0 | 1355 | 693 | 645 | 96 | 0 | 0 | 0 | 0 |
| axi_intc | microblaze_0_intc | 396 | 192 | 204 | 32 | 0 | 0 | 0 | 0 |
| microblaze | microblaze_0 | 3259 | 1524 | 1610 | 192 | 10 | 0 | 0 | 0 |
| mig_7series_0 | DDR3_SDRAM | 23988 | 10501 | 13021 | 2325 | 0 | 0 | 2 | 0 |
| axi_bram_ctrl | axi_bram_ctrl_0 | 680 | 315 | 363 | 2 | 32 | 0 | 0 | 0 |
| axi_cdma | axi_cdma_0 | 1888 | 1020 | 868 | 81 | 0 | 0 | 0 | 0 |
| axi_cdma | axi_cdma_1 | 1887 | 1020 | 867 | 81 | 0 | 0 | 0 | 0 |
| axi_timer | axi_timer_0 | 490 | 222 | 268 | 0 | 0 | 0 | 0 | 0 |
| axi_uartlite | RS232_Uart_1 | 185 | 88 | 96 | 10 | 0 | 0 | 0 | 0 |

## Assumptions

This application note assumes some general knowledge of the Vivado IP integrator. See *Vivado Design Suite User Guide: Designing IP Subsystems Using IP Integrator* (UG994) [Ref 3] for more information about the Vivado IP integrator.

## References

For a glossary of technical terms used in Xilinx documentation, see:

www.xilinx.com/company/terms.htm.

1. *LogiCORE IP AXI Quad SPI Product Guide* (PG153)
2. *Flash Memory Bootloading Using SPI with Spartan-3A DSP 1800A Starter Platform* (XAPP1053)
3. *Vivado Design Suite User Guide: Designing IP Subsystems Using IP Integrator* (UG994)

## Revision History

The following table shows the revision history for this document.

| Date | Version | Description of Revisions |
|---|---|---|
| 02/20/2013 | 1.0 | Initial Xilinx release |
| 02/07/2014 | 2.0 | Added support for Vivado Design Suite and IP integrator<br>Update References section to refer to current documentation |

## Notice of Disclaimer

The information disclosed to you hereunder (the “Materials”) is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of the Limited Warranties which can be viewed at http://www.xilinx.com/warranty.htm; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in Critical Applications: http://www.xilinx.com/warranty.htm#critapps.