SD Host vs. SPI Comparison

This report is intended to provide a comparison between a utilizing a generic SPI Host and a full-featured SD/SDIO/MMC Host for integrating SD Card / device functionality into an Altera NIOS-based platform using uClinux as an OS. Engineers and system integrators can benefit from learning of some of the advantages and drawbacks with the various approaches available in integrating an SD Host into their platform.

SD Protocol Overview

The SD/SDIO Protocol (current spec 2.0) is a high-speed serial protocol used primarily for interfacing with SD (SecureDigital) Flash memory cards. SDIO, a specification based on SD, can be used to interface with devices such as Wireless ICs. This paper will mainly focus on the SD protocol and interfacing with flash memory. It should also be noted that the SD Protocol is largely based on the MultiMediaCard (MMC) format, although some mechanical differences exist. For the purpose of this report, they can be treated equally.

The first revision of the SD Protocol supported only up to 4GB of data in a flash memory card. With the advent of SDHC (2.0), capacities of up to 32 GB (2TB theoretical) can be supported. On a physical level, SDHC is equivalent to SD 1.0. The changes mostly exist in the block/MAC-level of the protocol.

The table below outlines the various signals used when interfacing with an SD device. Different operating modes utilize the signals in different ways.

SD Mode (1 and 4-bit)			**SPI Mode**		
Name	Type	Description	Name	Type	Description
CMD	Bidir.	Command/Response	DI	Input	Data In
CLK	Input	Clock	SCLK	Input	Clock
DAT[0]	Bidir.	Data Line 0	DO	Output	Data Out
DAT[1]	Bidir.	Data Line 1	RSV	-	-
DAT[2]	Bidir.	Data Line 2	RSV	-	-
DAT[3]	Bidir.	Data Line 3	CS	Input	Chip-select

There are 3 fundamental modes that the SD Physical layer can operate in:

1. 4-bit SD DAT Mode
2. 1-bit SD DAT Mode
3. SPI-Mode

All modes of transport only dictate the physical layer operation; the block layer of the interface remains the same regardless of transport mode used. All modes of operation can be clocked up to 25MHz in standard-speed mode, or 50 MHz in high-speed mode.

1 and 4-bit SD DAT Mode

The 1 and 4-bit SD DAT Modes differ only by the number of data lines used. The 4-bit mode allows the host to utilize all four of the SD Data lines, thereby increasing throughput significantly. In SD Mode, all command/response tokens are sent over the CMD line. The DAT lines are reserved for data blocks. All command/response tokens are required to be protected with a 7-bit CRC, and the Data blocks with a 16-bit CRC (both based on CCITT polynomials).

In SD Mode, the DAT signals are also used for three additional functions. 1) To implement wait states from the device to the host, 2) To indicate successful reception of data blocks on write operations, and 2) To signal interrupts from the device to the host. Wait states are useful when writing large data blocks to an SD Device for example. In the event that the write-buffers on the card are full, the device can indicate it is busy through special signaling on the DAT line. This eliminates the need for un-necessary polling. On reception of data blocks that are being written to the device, the device will calculate and verify the CRC of the data in real-time, and report the status on the DAT lines as well. This allows the host to terminate or re-issue a write block transaction if the device reports a CRC error.

SPI Mode

SPI Mode utilizes the popular Serial Peripheral Interface Bus (SPI). This bus contains clock, chip-select, data-in and data-out signals. Most microcontrollers/processors come with a host SPI port, or a variant, allowing for easy interfacing. In addition, several low-end microcontrollers also support a SPI port due to its low resource requirements.

With SPI Mode, all command/response tokens are transmitted over the Data-in / Data-out pins. There is no separate line for command/response. Once command/response tokens have been transmitted, data blocks are also transmitted on the same lines. CRC protection is also specified using SPI, however, it is disabled by default and uses a weak CRC protection scheme since most SPI Hosts do not have built in CRC generation.

Test Setup

Hardware

The Nios II Evaluation Kit from Altera (aka: NEEK) is the platform used for evaluating performance of an SD Host versus SPI Host. The kit features 32 MB DDR RAM, a Cyclone III FPGA, and SD Slot. Unfortunately, the NEEK only supports 1-bit SD DAT Mode since all 4 Data signals are not connected to the FPGA.

The FPGA Projects used were:

- FPS-Tech SD Host Evaluation project; used for testing the FPS-Tech SD Host controller. Datasheet can be found here.
- neek_ocm_spi test project located on the NIOS Wiki; used for testing the performance in SPI Mode. Project uses Altera SPI Host controller. Datasheet can be found here.

SD Host Overview

A brief description of the SD Hosts used will be outlined in this section. The FPS-Tech SD Host controller is a high-performance SD Host designed specifically for Altera NIOS-based platforms. The host has a number of features that dramatically improve performance in uClinux environments. Due to the relatively low clock speeds that the NIOS can run at, the SD Host tries to offload as much work as possible to free up the processor for other tasks. Some of the features of the host include:

- Full support for 1 and 4-bit DAT Modes
- Integrated card-detect and write-protect pins with filters
- Open-source uClinux driver, available on public upstream repository
- Powerful integrated DMA engine backpressure and pre-fetching support to tackle high-latency memory sub-systems (such as DDR)
- Support for SDHC flash cards
- Deep read and write FIFOs to ensure maximal physical-layer transfer rate
- DMA engine designed specifically to maximize Altera Avalon's interconnect architecture features
- Integrated 7-bit and 16-bit CCITT CRC generation
- Detailed profiling registers included to identify any potential bottlenecks in system
- External busy signal output for busy indication

The SPI host controller used was a standard Altera SPI Host. The host allows for different data byte sizes, but does not have any DMA or CRC capabilities since it is a standard SPI Host. The Altera SPI host is also supported by the upstream uClinux distribution through an open-source driver.

Software

uClinux 2.6.26 and 2.6.27 were used as the kernel revisions for benchmarking the test setups. A modified version of the disk IO benchmarking utility 'bonnie' (information located here) was used to measure read and write throughput. The utility was modified to remove character based read/write tests since they are more a measure of the processor performance. The test was run on various file sizes, although performance maintained relatively constant across different test cases.

In addition to measuring read/write performance, interrupt statistics and profiling registers were also monitored to obtain additional information about the tests. This test is as much of a software driver test as it is a SD Host test since driver interaction with the kernel can heavily influence the results. Since most system integrators will be considering uClinux to interface with SD memory, the system consisting of the driver and SD Host will be treated as one.

Test Setup Summary

The test-setup for the SPI-based host is given in the table below:

Test Setup	
Date	9/14/2008
Hardware Platform	NEEK
CPU / MEM Configuration	100 MHz CPU / 66.5 MHz, 32 MB DDR
FPGA Project	neek_ocm_spi
SD Controller	
Type	Altera Standard SPI Controller
Version	Quartus 8.0
PHY Interface Mode / Speed	1-bit SPI Mode / 15.0 MHz
Linux Kernel / Driver	
Kernel Version	2.6.27-rc6
Driver Configuration	No bounce buffer
SD Card	
Brand	Ultra SecureDigital Card
Size	1 GB

The test setup for the FPS-Tech SD Host is given in the table below:

Test Setup	
Date	7/9/2008
Hardware Platform	NEEK
CPU / MEM Configuration	87.5MHz, 32 MB DDR
FPGA Project	uclinux_sdio_test
SD Controller	
Type	FPS-Tech SD/SDIO/MMC Host
Version	1.1
PHY Interface Mode / Speed	21MHz / 1-bit
Linux Kernel / Driver	
Kernel Version	2.6.26-rc9
Driver Configuration	No bounce buffer, max_blk_count = 8, max_seg_size = max_req_size = 4096
SD Card	
Brand	Ultra Secure Digital Card
Size	1 GB

Benchmark Results

SPI Host Results

Test Configuration	
File Size	50 MB
Write Stats	
Block Output	75 KB/s, 3.2% CPU
Driver Interrupts	*<Independent Read/write interrupts not available>*
XFER_LEN / BUSY_LEN / BUS_LEN	*<Stats not available for SPI Core>*
Read Stats	
Block Input	69 KB/s, 1.1% CPU
Driver Interrupts	110,591,051 *(read+write)*
XFER_LEN / BUSY_LEN / BUS_LEN	24 minutes for read+write

Test Configuration	
File Size	250 MB
Write Stats	
Block Output	74 KB/s, 3.3% CPU
Driver Interrupts	*<Independent Read/write interrupts not available>*
XFER_LEN / BUSY_LEN / BUS_LEN	*<Stats not available for SPI Core>*
Read Stats	
Block Input	69 KB/s, 1.1% CPU
Driver Interrupts	553,209,715 *(read+write)*
XFER_LEN / BUSY_LEN / BUS_LEN	1h 58m for read+write

FPS-Tech SD Host Results

Test Configuration	
File Size	50 MB
Write Stats	
Block Output	1464 KB/s, 39% CPU
Driver Interrupts	53,157
XFER_LEN / BUSY_LEN / BUS_LEN	22 /3/0
Read Stats	
Block Input	1737 KB/s, 18.4% CPU
Driver Interrupts	26,031
XFER_LEN / BUSY_LEN / BUS_LEN	21 /2/0

Test Configuration	
File Size	250 MB
Write Stats	
Block Output	1410 KB/s, 39% CPU
Driver Interrupts	271,021
XFER_LEN / BUSY_LEN / BUS_LEN	120/19/2
Read Stats	
Block Input	1739 KB/s, 18.5% CPU
Driver Interrupts	130,188
XFER_LEN / BUSY_LEN / BUS_LEN	109/12/0

Note*: The XFER_LEN/BUSY_LEN/BUS_LEN numbers are all given in 'seconds' units. For details on these parameters, please refer to the FPS-Tech SD Host datasheet.*

Summary and Conclusions

Using the results available in the benchmarking and protocol information sections, one should be able to determine which SD Host (SPI or full-featured SD-host) suits their application best. While it is clear that the FPS-Tech SD-host provides a very large increase in throughput, this may or may not be the sole requirement of the application at hand.

The table below summarizes the advantages/disadvantages of both approaches is given below.

Item	SPI Host	FPS-Tech SD Host
Logic Footprint	Very small logic footprint inside the FPGA (<200 LEs) due to simple logic.	Larger footprint; around 2,000 LEs are required to implement the full 4-bit SD Host
Memory Footprint	Requires no on-chip memory inside FPGA	Required 1 M9K of on-chip memory for read/write FIFO
Throughput	Very low performance due to byte-level interaction between driver and host	Highest throughput possible due to integrated DMA engine, block pre-fetching schemes and full 4-bit DAT interface
Driver Size	Slightly more complicated driver due to byte-level interaction between driver and host	Simple driver since most functionality is offloaded onto SD Host IP.
Cost	SPI Host and driver are free from Altera	Driver is free open-source, but SD Host IP requires a one-time fee (no royalty)
Protocol Usage	SPI Host utilizes the simpler SPI mode of the SD protocol	Full 4-bit DAT in SD Mode utilized, with optional 1-bit mode if desired
Processor/OS Overhead	Significant processor overhead incurred due to large number of interrupts (byte-level processing)	Very-low processor overhead since transactions are handled on a multi block-basis by the Host IP
Driver Availability	Driver is integrated into upstream distribution, open-source, supported by NIOS Community	Driver is integrated into upstream distribution, open-source, supported by FPS-Tech
Device-side interrupts	SPI Mode does not support Device-side interrupts	Full support for generation of interrupts from device side
Busy Indication	SPI Mode does not support wait state signaling from device side	Full support for wait states from device, eliminating need for polling
CRC Generation/Verification	SPI Mode supports CRC, but is disabled by default. The Altera SPI Host does not support hardware generation of CRCs	Generation and verification of both 7 and 16- bit CCITT CRCs are performed automatically

In conclusion, both host options offers their own distinct advantages that system integrators need to consider when choosing the right host. The SPI Mode host is a free, simple and effective way to implement SD functionality into a system if performance is not a requirement. In the event that throughput and processor overhead are of concern, going with the full-featured SD Host by FPS-Tech will offer numerous advantages. If the target application requires as many CPU cycles as possible to be dedicated for other applications, the FPS-Tech SD Host will allow this since the bulk of the SD Host interfacing is offloaded into hardware.