DESIGN GATEWAY COMPANY LIMITED

Design Gateway Co.,Ltd

54 BB Building 13th Fl., Room No.1302 Sukhumvit 21 Rd. (Asoke), Klongtoey-Nua, Wattana, Bangkok 10110, Thailand
Phone: (+66) 02-261-2277
Fax: (+66) 02-261-2290
E-mail: sales@design-gateway.com
URL: www.design-gateway.com

SDXC-IP Core Facts

Provided with Core	
Documentation	User Guide, Design Guide
Design File Formats	NGC, VHDL
Constraints Files	User constraint file
Verification	VHDL Test Bench
Instantiation Templates	VHDL
Reference Designs & Application Notes	EDK Project, See Reference Design manual
Additional Items	Demo on ML505/ML506
Simulation Tool Used	
ModelSim version 6.5d	
Support	
Support Provided by Design Gateway Co.,Ltd	

Features

- Auto detect SD card type
- Support SDSC (Standard Capacity), SDHC (High Capacity), and SDXC (Extended Capacity).
- Support Bus speed mode: Default speed (12.5 MB/s), High Speed (25 MB/s), and SDR50 (50 MB/s)
- Auto SD clock suspension for power saving
- Auto checking error
  - Cyclic Redundancy Check (CRC16) for read data transfer
  - CRC status check for write data transfer
  - Timeout when no response returned from SD card
  - Timeout when SD card is not ready in time during initialization process
- High performance for data transfer by 4-bit SD data transfer mode
- Simple connection for command signal and FIFO interface for data signal.
- No global clock and clock management inside it to reduce clock resource for RAID system.
- Compliant with SD Specifications Part 1 Physical Layer version 3.01
- Reference designs available on ML505/ML506 with two data formats, i.e.
  - High performance write/read transfer by raw data format (no file system).
  - Write/Read transfer in file system format (FAT32, and recently released for exFAT).

Table 1: Example Implementation Statistics for Xilinx® FPGAs

Family	Example Device	Fmax (MHz)	Slice LUT-FF	Slice LUTS	Slices[1]	IOB[2]	BUFG	BRAM	MULT/ DSP48/E	DCM / CMT	Design Tools
Spartan-6	XC6SLX45-3FGG484	256	733	706	260	134	-	-	-	-	ISE11.5
Virtex-5	XC5VLX50T-1FF1136	344	734	646	249	134	-	-	-	-	ISE11.5
Virtex-6	XC6VLX240T-1FF1136	370	713	699	242	134	-	-	-	-	ISE11.5

Notes:

1) Actual slice count dependent on percentage of unrelated logic – see Mapping Report File for details

2) Assuming all core I/Os and clocks are routed off-chip

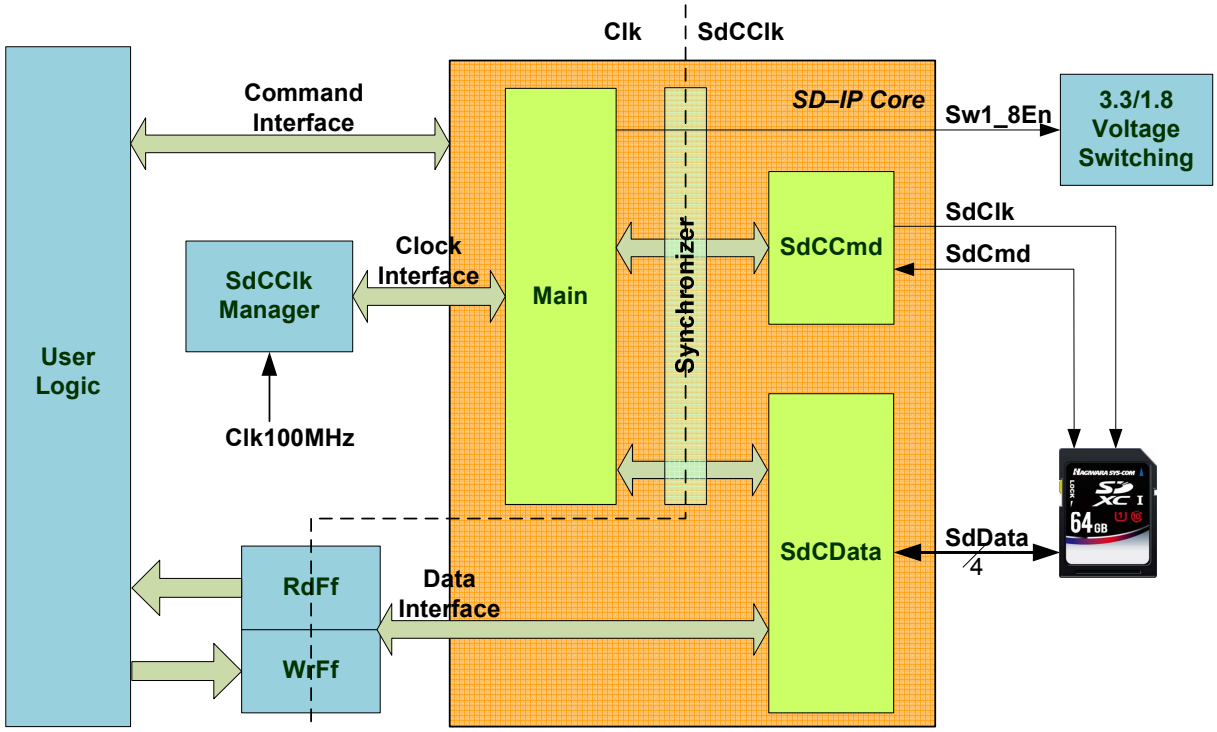

Figure 1: SDXC-IP Block Diagram

Applications

SDXC-IP core can be applied in many storage applications with compact size. High performance with high capacity can be designed by using RAID system. For example application,

- Data processing in portable product
- Video/Audio/Image processing in consumer product
- Data acquisition in test equipment

General Description

SDXC-IP core is designed to help FPGA user easily write/read data with SD card. This IP-Core implements both low-level and high-level layer of SD protocol to initialize SD Card, write, and read SD card to specified physical address from user. Simple command interface is synchronized with user clock (Clk) and Data interface can be directly connected with FIFO by SdCClk. Two clock sources are required for SDXC-IP, i.e. Clk which is system clock of User Logic and SdCClk for SD card which frequency can be adjusted by SDXC-IP core. External clock manager is required to interface with SDXC-IP core to adjust SD Card clock frequency and only one clock manager is used to reduce clock resource in case of SD-RAID system. The example of clock manager is also provided in reference design.

To interface with SDHC/SDXC in SDR50 speed mode, external voltage switching is required to switch voltage of both FPGA circuit and SD card from 3.3 V to 1.8 V, controlled by “Sw1_8En” signal from SDXC-IP.

Functional Description

SDXC-IP consists of three modules, i.e. Main (Clk domain), SdCCmd and SdCData (SdCClk domain), as shown in Figure 1. The details of each module are described as below.

SDXC-IP Core

Main module

Main module is designed to receive write/read command with specified address and length from user and then decode and convert to SD protocol command. After that, Main module sends control signals to SdCCmd and SdCData to start transferring data between RdFf/WrFf and SD card. At 1st time power-on system with SD card insertion, Main module will be busy state to run initialize process with SD card and get that SD card information and provide them to user. During card initialization, Main will request at most 2 times to clock manager to change SD card clock frequency to optimize initialize time usage and achieve the best transfer performance for that SD card. Error detection for 1-sec timeout during SD initialization is included in this module.

SdCCmd module

This module is designed to send command code and receive command response with SD card during initialization, write, or read SD card. CRC7 is auto-inserted by this module after serializing all command code. Error detection when no response returned from SD card within 64 Clock is included in this module.

SdCData module

This module is designed to read/write data from/to two FIFOs (RdFf and WrFf). 512-byte burst size is used to transfer data between FIFO and SD card. In SD write direction, 512-byte data will be read from WrFf when WrFfRdyB is '0'. In SD read direction, 512-byte data will be written to RdFf when RdFfRdyB is '0'. Error detections are included to check CRC status error returned in write-direction and CRC16 value in data packet returned in read-direction.

Rd/WrFf

Both FIFOs are asychronous FIFO which can be easily generated by using CORE Generator in ISE software from Xilinx. To connect with SDXC-IP, data width of FIFO at SDXC-IP side should be equal to 4-bit. "RdFfWrRdyB" and "WrFfRdRdyB" of SDXC-IP can be generated from data counter of FIFO. The example of RdFf and WrFf are shown in reference design.

SdCClk Manager

This module is designed to generate clock for SD card (SdCClk) which should be changed its frequency during operation. Four different frequencies are applied to support a variety of SD card bus speed mode, i.e. 400 kHz during initialize process, 25 MHz for Default speed mode, 50 MHz for High speed mode, and 100 MHz for SDR50 mode. By fedding 100 MHz clock input with DCM and BUFGMUX, clock frequency of SdCClk can be adjusted.

Core I/O Signals

Descriptions of all signal I/O are provided in Table 2.

Table 2: Core I/O Signals.

Signal	Dir	Polarity	Domain	Description
Clock and Reset Interface				
RstB	In	Low		Assert '0' to reset SDXC-IP from unrecoverable error or re-initialize SD card.
Clk	In			Clock system for command interface with user logic. The frequency should be at least 20 MHz.
SdCClk	In			Clock input for SD card interface. Clock frequency of this clock is specified from SwClkSel signal, i.e. "00"- 400 kHz, "01"- 25 MHz, "10"- 50 MHz, and "11"- 100 MHz.
SwClkSel[1:0]	Out		Clk	Frequency value control signal for SdCClk. Timing diagram of this signal is shown in Figure 4.
SwClkReq	Out	High	Clk	Request signal to change SdCClk frequency. Timing diagram of this signal is shown in Figure 4.
SwClkBusy	In	High	Clk	Busy and acknowledge flag to check that clock frequency switching is ready, busy, or completed. Timing diagram of this signal is shown in Figure 4.
Command Interface				
Addr[31:0]	In		Clk	Start address to transfer data with SD card address in block unit (512 bytes). Valid from 0 to "CardSize – 1".
Length[31:0]	In		Clk	Total data transfer size in block unit (512 bytes). Valid from 1 to "CardSize".
nWrRd	In		Clk	Transfer direction (0: write data to SD card, 1: read data from SD card).
CmdReq	In	High	Clk	Request signal to start transfer data
Busy	Out	High	Clk	Busy and acknowledge signal after receiving CmdReq ('0': SDXC-IP Idle, '1': SDXC-IP Busy)
CardSize[31:0]	Out		Clk	Total number of SD card size in block unit (512 bytes)
Error[3:0]	Out	High	Clk	Flag for indicate SD card error [0]: 1-sec timeout during SD card intialization. Assert when SD card is not ready within 1 sec after power-on and card insertion detect. Auto-clear if card intialization complete. [1]: Card response timeout. Assert when no response from SD card within 64 clock after sending command. This error can be cleared only by RstB signal or re-insert SD card. [2]: CRC status error. Assert when CRC status returned from SD card after write data is not correct. This error will cancel current write operation and auto-clear when user sending new write command transfer. [3]: CRC data error. Assert when CRC data in received data packet from SD card is not correct. This error will cancel current read operation and auto-clear when user sending new read command transfer.
Data Interface				
RdFfWrData[3:0]	Out		SdCClk	4-bit data output from SD card to RdFf
RdFfWrEn	Out	High	SdCClk	Data write enable to RdFf
RdFfWrRdyB	In	Low	SdCClk	RdFf is ready to receive new 512-byte data. Set to '0' when available space in FIFO is more than 512 bytes.
WrFfRdData[3:0]	In		SdCClk	4-bit data input from WrFf to SD card
WrFfRdEn	Out	High	SdCClk	Data read enable to WrFf
WrFfRdRdyB	In	Low	SdCClk	WrFf is ready to read all 512-byte data. Set to '0' when FIFO stores at least 512 bytes data.

Signal	Dir	Polarity	Domain	Description
SD Card Interface				
VoltChDelay[7:0]	In			Constant value to set delay time to wait voltage switching circuit from 3.3V to 1.8V. Step size is 1.3 ms. Valid from 1 (1.3 ms) – 255 (331.5 ms).
Sw1_8En	Out			Voltage supply value select for FPGA and SD card interface. '0': 3.3 V, '1': 1.8V. After assert to '1', this signal can be cleared to '0' only when card re-insertion or power-off/on system and cannot be cleared by RstB signal. Following SD specification, SD card cannot switch voltage back from 3.3V to 1.8V when runs in 1.8V mode.
SdClk	Out			Clock output to SD card.
SdCmd	In/Out			Command and response from/to SD card. Internal or external pull-up circuit is required to connect between FPGA pin and SD card.
SdData[3:0]	In/Out			4-bit data bus from/to SD card. Internal or external pull-up circuit are required to connect between FPGA pin and SD card.
SdCardDetect	In	Low		SD card detection signal. This signal isasserted to '0' when card is inserted. Internal or external pull-up circuit is required to connect between FPGA pin and SD card.

Timing diagram

Figure 2 and Figure 3 show timing diagram of both command and data interface for write and read transfer consecutively. Each timing diagram is divided into 2 paths, i.e. command (CMD) interface, synchronized with Clk signal and data interface, synchronized with SdcClk signal. CmdReq is asserted to SDXC-IP core to request write/read transfer with SD card. During asserting CmdReq, nWrRd, Addr, and Length signals need to hold same value until Busy from SDXC-IP is asserted to acknowledge current request. CmdReq and other signals can be de-asserted after detecting Busy flag assertion.

In write transfer as shown in Figure 2, Busy is cleared after total data blocks are transferred from WrFf to SD card. Data is transferred in 512-byte burst size and WrFfRdRdyB is checked before starting each burst-write transfer. WrFfRdEn signal is asserted for 1024 clock in each burst transfer. Similar to read timing diagram of typical FIFO, WrFfRdData is valid after WrFfRdEn is asserted about 1 clock period.

In read transfer as shown in Figure 3, Busy is cleared after total data blocks are transferred from SD card to RdFf. Data is transferred in 512-byte burst size and RrFfWrRdyB is checked before starting each burst-read transfer. RdFfWrEn signal is asserted for 1024 clock in each burst transfer. Similar to write timing diagram of typical FIFO, RdFfWrEn and RdFfWrData are asserted simultaneously.

Figure 4 shows timing diagram between SDXC-IP core and SdCClk manager during to change SdCClk frequency. "SwClkReq" is asserted with "SwClkSel" signal and de-asserted after "SwClkBusy" is asserted. SDXC-IP will wait and run next function after "SwClkBusy" is de-asserted to confirm that SdCClk clock frequency is changed completely.

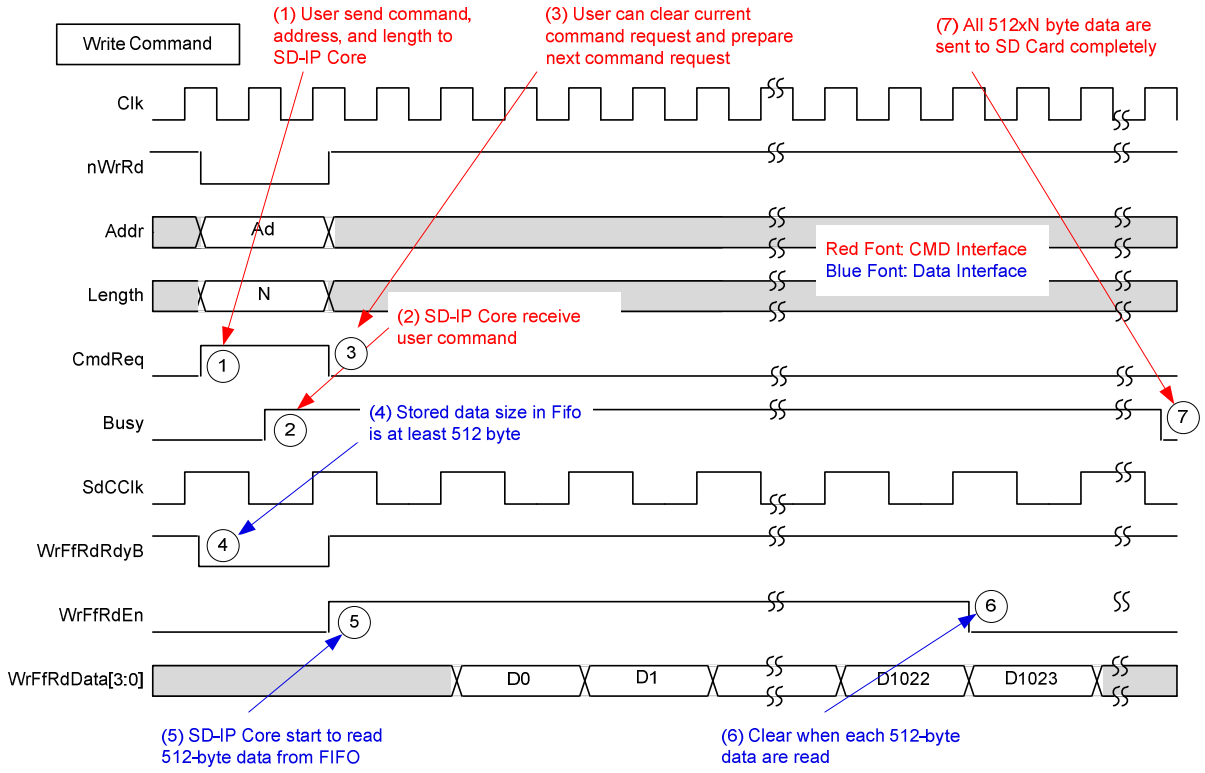

Figure 2: Write Transfer Timing diagram

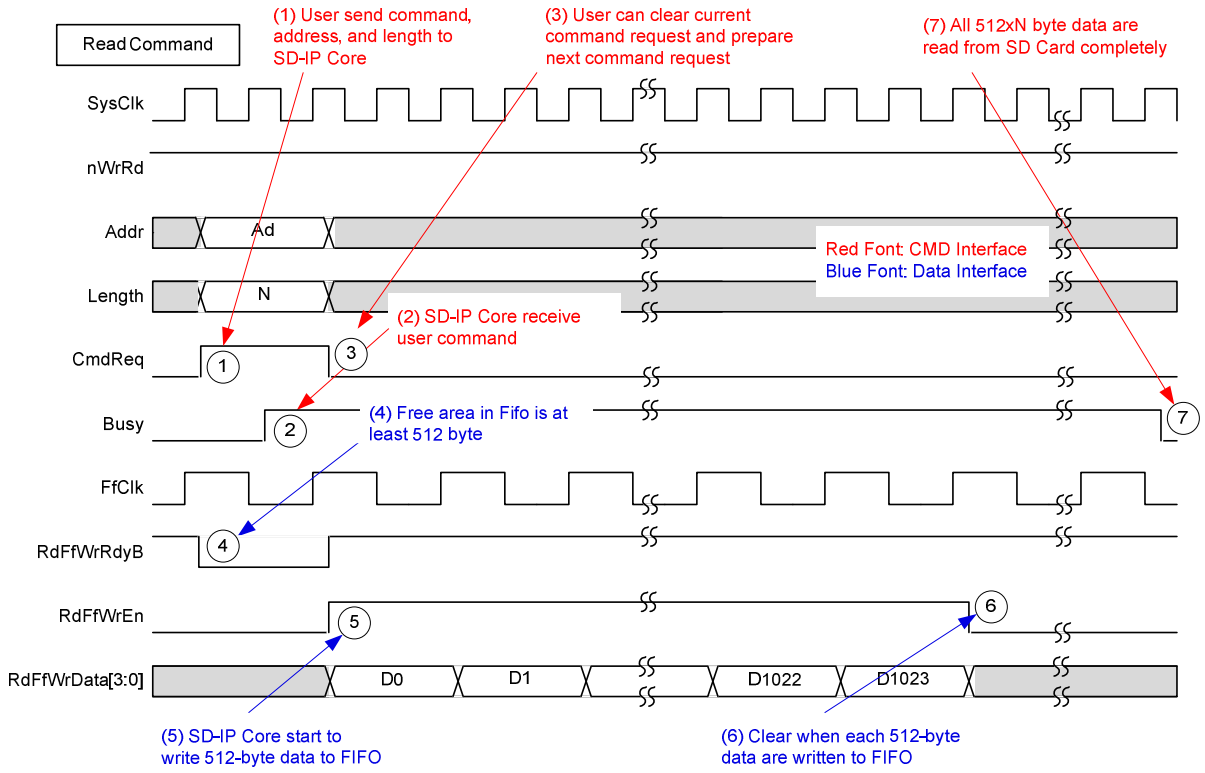

Figure 3: Read Transfer Timing diagram

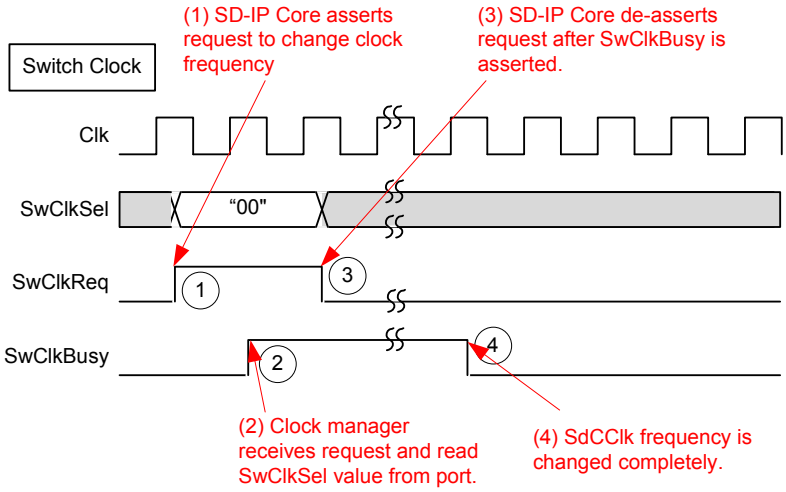

Figure 4: Switch clock frequency Timing diagram

Verification Methods

The SDXC-IP core functionality was verified by simulation and also proved on real board design by using ML506 evaluation board.

Recommended Design Experience

The user must be familiar with HDL design methodology as well as instantiation of Xilinx netlists in a hierarchical design environment.

Ordering Information

This product is available directly from Design Gateway company. Please contact Design Gateway for pricing and additional information about this product using the contact information on the front page of this datasheet.

Revision History

Revision	Date	Description
1.0	Jun-15-2009	New release
1.1	Jul-19-2010	Update to support SDXC card
1.2	Nov-11-2010	Update to SDXC-IP