SDXC-IP Reference Design on ML505/ML506 Manual

Rev1.0 Nov 11, 2010

This document describes the details of SDXC-IP reference design on ML505/ML506 Evaluation board from Xilinx. Three reference designs are provided, i.e. raw data format, FAT32 format, and exFAT format. All reference designs use same hardware structure and update only firmware on MicroBlaze.

1. Introduction

Secure Digital (SD) is a non-volatile memory card format developed by Matsushita, SanDisk, and Toshiba for use in portable devices. Today it is widely used in digital electric products for portable storage application.

There are three package sizes for SD card, i.e. Standard Size, miniSD size, and microSD size, as shown in Figure 1.

In terms of card capacity, three types of SD card are defined, i.e. Standard Capacity (SDSC) which size is up to 2 GB, High Capacity (SDHC) which size is more than 2GB and up to 32 GB, and Extended Capacity (SDXC) which size is more than 32 GB and up to 2 TB.

SD card can transfer in several bus speed mode when using 4 parallel data lines. SDXC-IP from DesignGateway can support 3 modes, i.e. Default speed mode (3.3V and up to 12.5 MB/s), High Speed mode (3.3V and up to 25 MB/s), and SDR50 (1.8V and up to 50 MB/s).
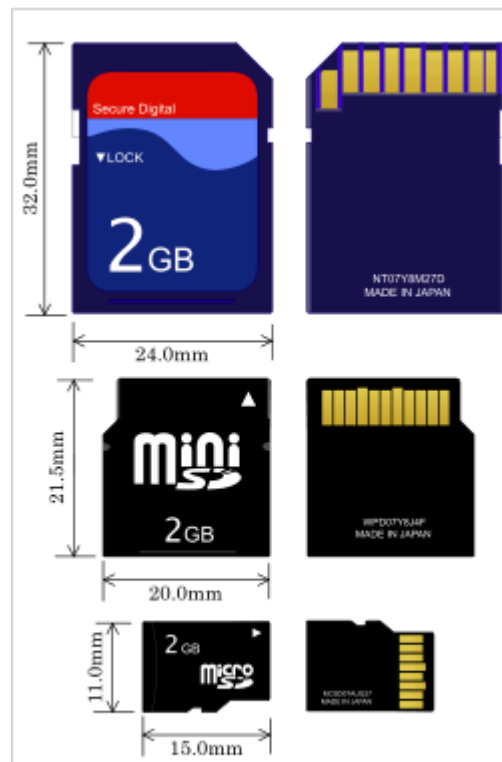
Figure 1 SD card package (From Wikipedia)

The reference design provides evaluation system which includes SDXC-IP and MicroBlaze to transfer data through DDR2. Firmware on MicroBlaze in reference design will write/read data with SDXC-IP by raw data format, FAT32 format, or exFAT format. File system can be used to compatible with other SD host such as PC. The reference system connects with SD card by using SDXC-IP Demo board which can be provided by DesignGateway. More details about reference design are follows.

2. Environment

This reference design is based on the following environment as shown in Figure 2.
- ML506/505 Platform
- ISE 11.5 / EDK 11.5
- SDXC-IP Demo board for SDXC-IP evaluation, provided by DesignGateway
- SD Card
- Serial (RS232C) communication, connect RS232C cable to P3 on ML506
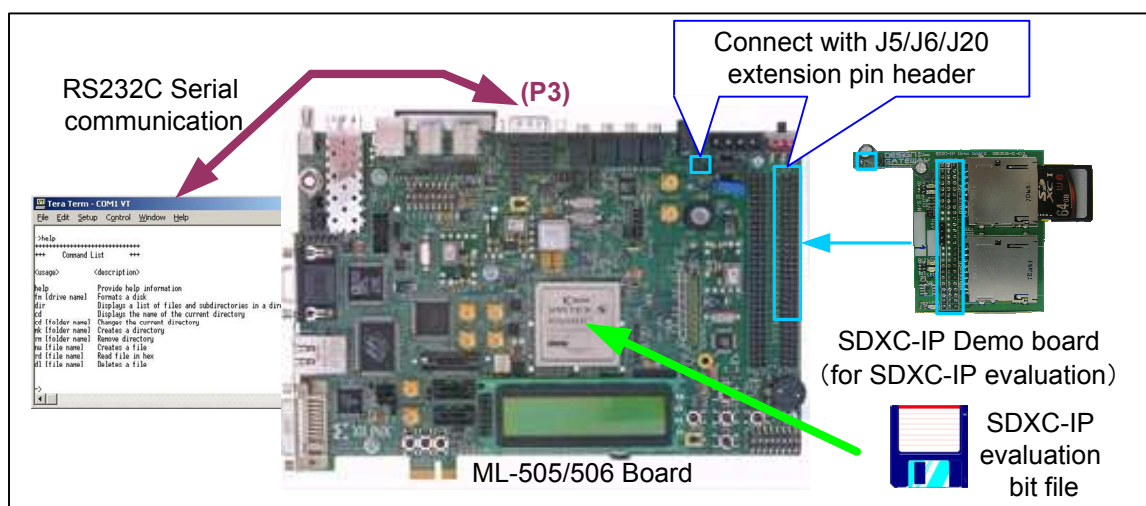 (Set baud rate=115,200 / data=8bit / Non-Parity / Stop=1bit)

Figure 2 Reference Design Environment

Refer to "SDXC_IP Demo Instruction" for operation procedure of this reference design. Because IP-Core of evaluation version has 1-hour operation limitation, IP-core will stop its operation after 1 hour expiration.

3. Hardware description
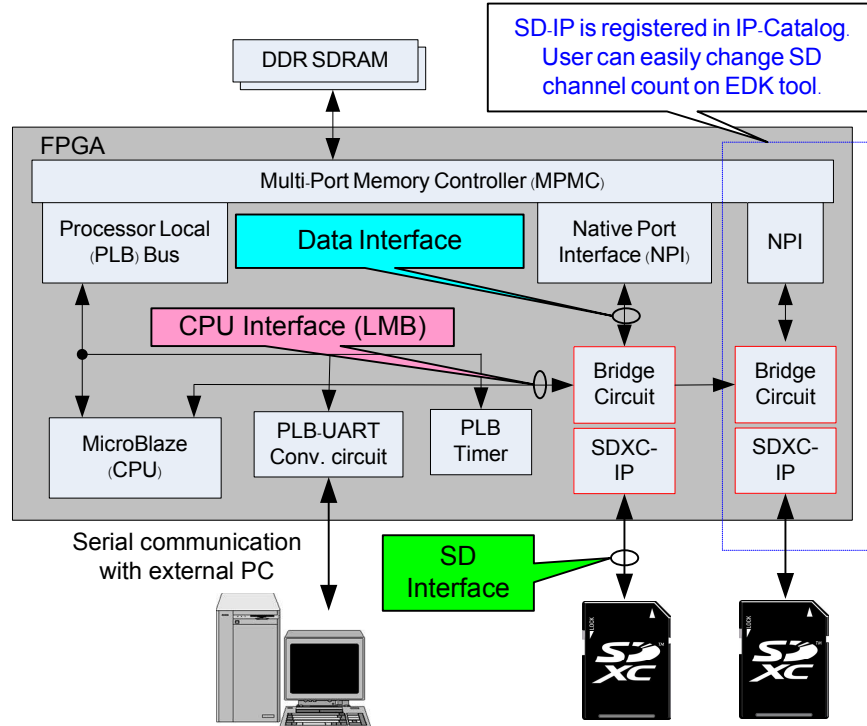
● SDXC-IP reference design on EDK

Figure 3 Hardware Block Diagram on EDK in reference design

As shown in Figure 3, this reference design is based on EDK system which consists of MicroBlaze (CPU), Multi-Port Memory Controller (MPMC), PLB UART and Timer, and SDXC-IP with Bridge Circuit.

Firmware on MicroBlaze (CPU) is designed to interface with user through UART protocol and RS-232C serial interface. User can control to start read/write SD card with setting address and transfer length value through Serial Console. MicroBlaze sends all control signals to SDXC-IP through LMB Bus (Local Memory Bus) and transfer data to SDXC-IP with DMA function through DDR and MPMC. Bridge circuit is designed to convert LMB bus and NPI bus interface to SDXC-IP interface. HDL source code of this bridge is free provided in the reference design project, stored in "pcores/npi_sd_v1_00a/hdl/vhdl/npi_sd.vhd".

In reference design, SDXC-IP combined with Bridge circuit is registered to the IP-Catalog of the Platform Studio (EDK), so user can build RAID system to use multiple SDXC-IP easily by adding SDXC-IP and add more NPI ports on MPMC.

More details about Bridge circuit to describe how to connect SDXC-IP in reference design is follows.
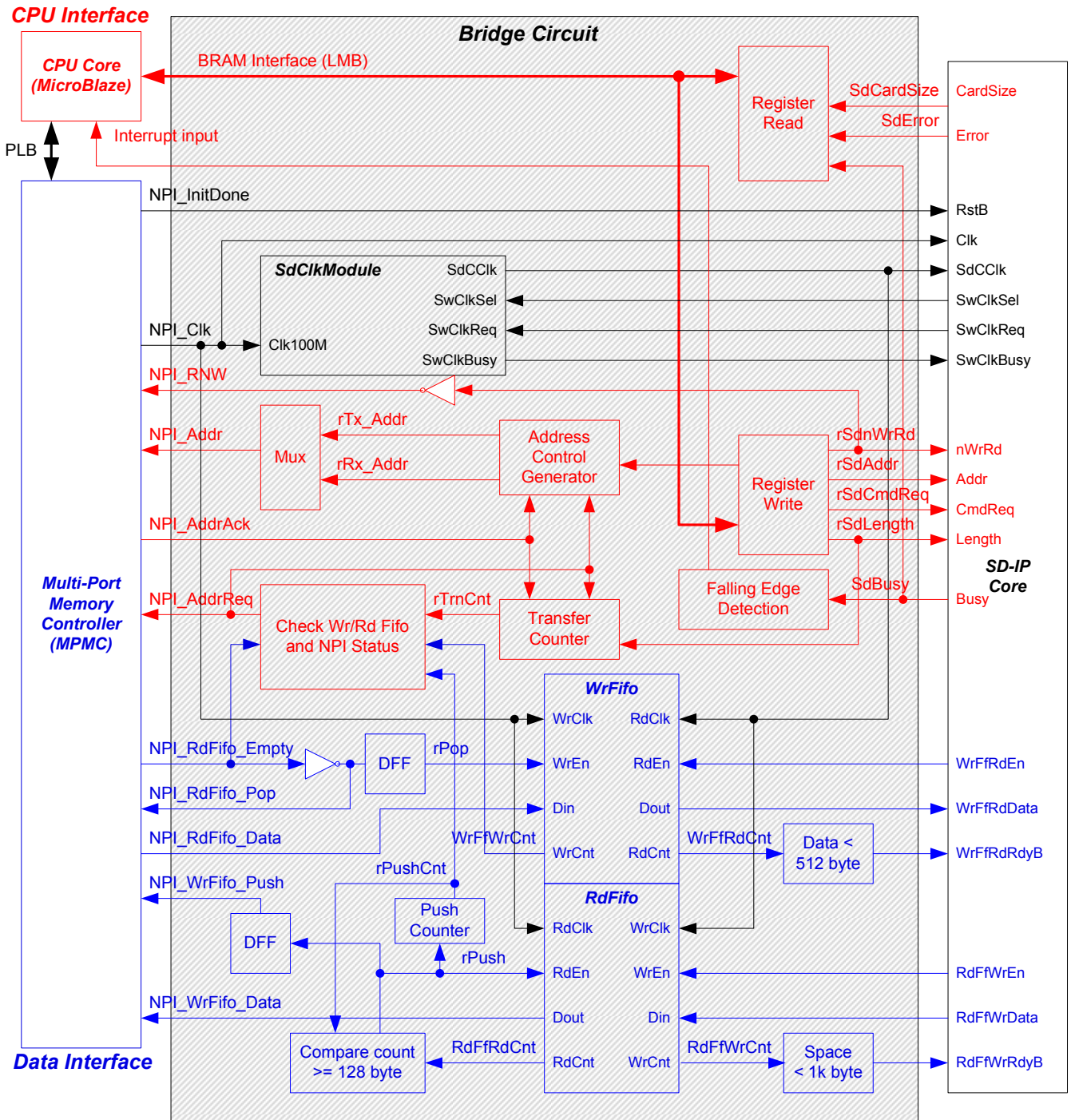
● SDXC-IP Bridge circuit to EDK system

Figure 4 Bridge Circuit Block Diagram

As shown in Figure 4, Bridge Circuit block diagram can be divided into three parts, i.e. Clock interface with SdClkModule, control signal interface with MicroBlaze and SDXC-IP, and data signal interface with MPMC and SDXC-IP.

NPI_Clk frequency in reference design is equal to 100 MHz. 100 MHz clock will be clock input to SdClkModule to generate SdCClk to SDXC-IP which can be select totally 4 different frequencies, i.e. 400 kHz, 25 MHz, 50 MHz, and 100 MHz. Inside SdClkModule, DCM is used to generate 25 MHz, 50 MHz, and 100 MHz, and simple clock divider logic is used to generate 400 kHz from 100 MHz input clock. All four clocks will be selected to be SdCClk output by using 3 BUFGMUX. "SwClkBusy" is generated by using S-R Flip Flop and "SwClkSel" will be internal latched to be clock selector signal of 3 BUFGMUX. HDL source code of SbClkModule is stored in "pcores/npi_sd_v1_00a/hdl/vhdl/SbClkModule.vhd".

For control signal with MicroBlaze and SDXC-IP, simple logic is designed for CPU write/read register through LMB Bus. MicroBlaze will set the value of SD card address (Addr), total transfer length (Length), read/write command (nWrRd), and start transfer signal (CmdReq) by writing register. Total card size (CardSize), Error status, and Busy flag can be monitored by MicroBlaze. Falling edge of "Busy" flag is applied to generate interrupt signal to MicroBlaze to notify that current transfer or initialize process is finished. Register map of MicroBlaze is shown in Table 2.

For data transfer between MicroBlaze and SDXC-IP, it's designed by using NPI port of MPMC. DDR2 is applied to be big buffer. NPI control signals to assign DDR2 address, burst size, transfer direction are also controlled by Register access from MicroBlaze and internal logic. WrFifo and RdFifo are asynchronous FIFO to convert different data size (32-bit and 4-bit) and different clock domain (NPI_Clk and SdCClk) between NPI bus data and SDXC-IP data.

● SDXC-IP Core interface

SDXC-IP Core signal interface can be divided into 4 groups, i.e. Clock and Reset signals, Command signals, Data signals, and SD card signals, as shown in Table 1. Waveforms to interface with SDXC-IP signal are shown in Figure 5 - Figure 7.

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.

Table 1 SDXC-IP Core Signal Interface Description

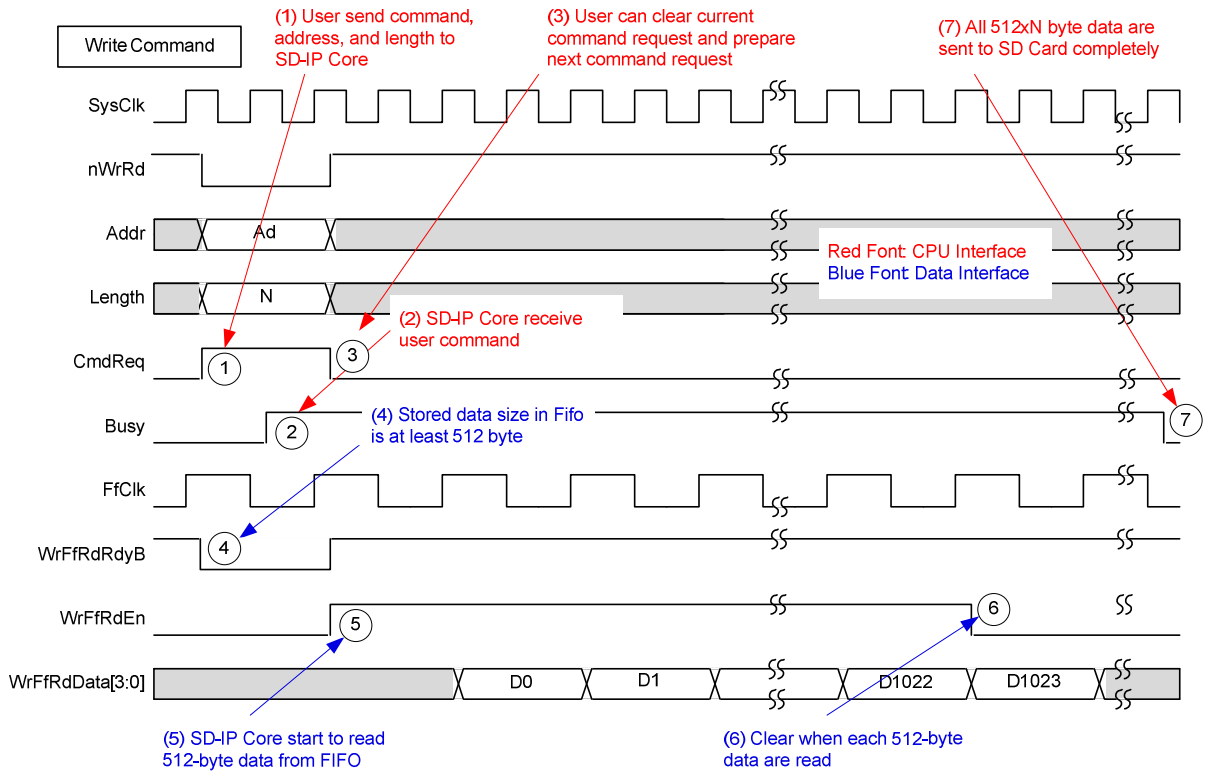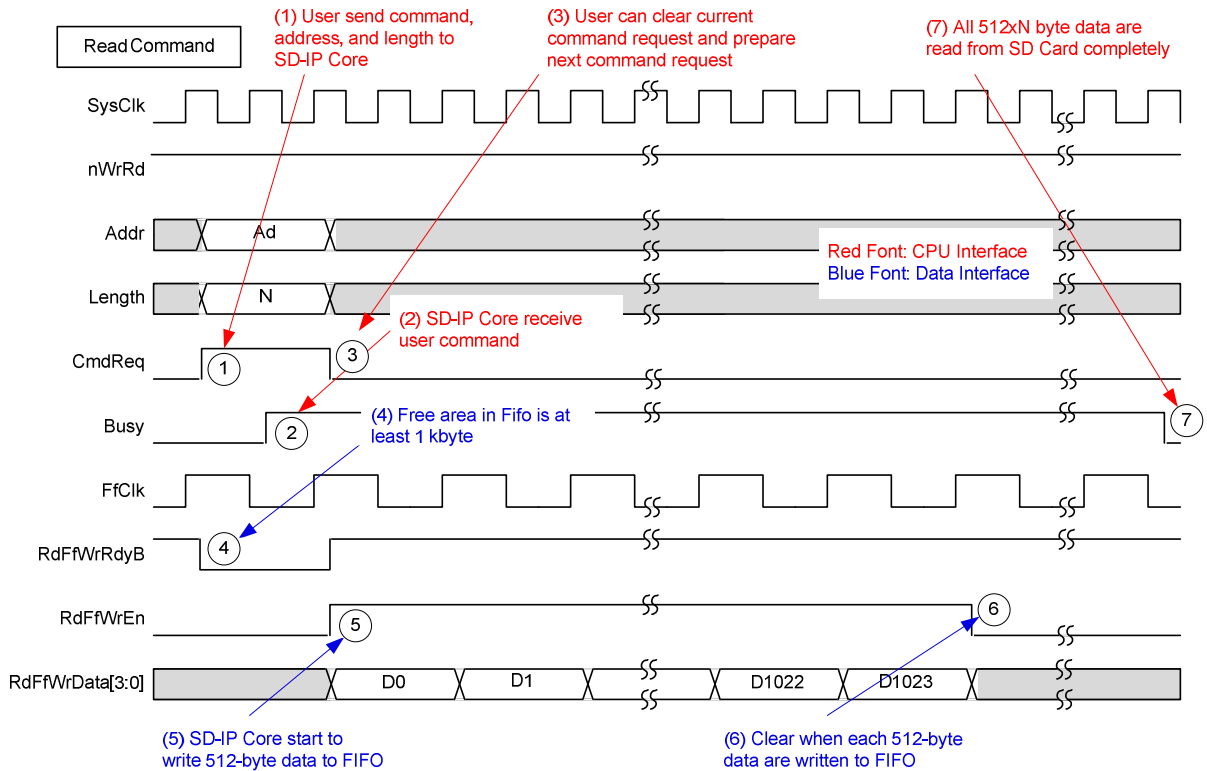| Signal	Dir	Polarity	Domain	Description
Command Interface				
Addr[31:0]	In		Clk	Start address to transfer data with SD card address in block unit (512 bytes). Valid from 0 to "CardSize – 1".
Length[31:0]	In		Clk	Total data transfer size in block unit (512 bytes). Valid from 1 to "CardSize".
nWrRd	In		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.
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.

Figure 5 Write Transfer Timing Diagram

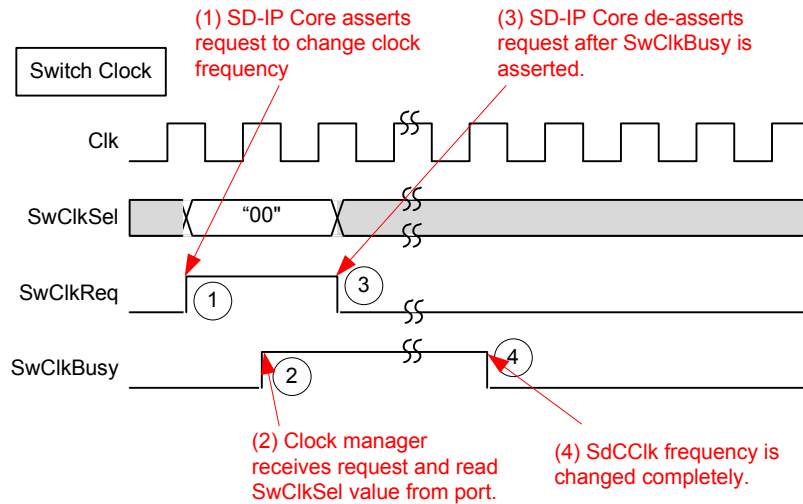Figure 6 Read Transfer Timing Diagram

(1) SD-IP Core asserts request to change clock frequency

(3) SD-IP Core de-asserts request after SwClkBusy is asserted.

(2) Clock manager receives request and read SwClkSel value from port.

(4) SdCClk frequency is changed completely.

Figure 7 Switch Clock Frequency Timing Diagram

● MicroBlaze

CPU is included for flexibility with user application by only modifying firmware on MicroBlaze without updating hardware structure. Three different firmwares are designed to demo data transfer data with SDXC card on reference design, i.e. raw data format for simple use, FAT32 format and exFAT format to compatible with many hosts such as PC. Raw data format firmware is described in more details in next topic. For FAT32 and exFAT firmware, please contact DesignGateway for more details. 1-hour timeout bit file including instruction manual of all designs are available to download.

Address Rd/Wr	Register Name (Label name in the source code)	Description (Bit order is Little-Endian expression)
BA+0x00 Rd	Status Register (STATUS)	[8]MPMC initialization completion [1]SDCardDetect: "0"=Card present, "1"=Card not present [0]Busy flag: "1"=Data transfer in progress, "0"=Idle
BA+0x04 Rd	Error Code Register (ERROR_CODE)	[3:0] Error flag signal from SDXC-IP. More details about this flag are shown in Table 1.
BA+0x08 Rd	Remained transfer Count Register (REM_CNT)	Total remained transfer size in block unit. (1block = 512bytes)
BA+0x0C Rd	Card Size Register (CARD_SIZE)	Total number of card size in block unit. (1block = 512bytes)
BA+0x00 Wr	Transmit Data Address Register (TX_ADDR)	Specify top address of transmit data address.
BA+0x04 Wr	Receive Data Address Register (RX_ADDR)	Specify top address of receive data address.
BA+0x08 Wr	Control Register (CONTROL)	[0] CmdReq Data transfer request. This bit is auto-cleared. Please set '1' to this bit for sending request to SDXC-IP [8] Reset system: '0'=No operation, '1'=Reset system. [31] Transfer direction: '0'=Write to SD Card, '1'=Read from SD Card.
BA+0x0C Wr	Target Address (TGT_ADDR)	[31:0] Start transfer address of SD Card, specified by block unit.
BA+0x10 Wr	Transfer Block Count Register (XFR_COUNT)	[31:0] Transfer block count, specified by block unit.

Table 2 Register Map between Bridge and MicroBlaze

(BA : Base Address)

4. Software description

Firmware on MicroBlaze uses C source code and user can update or modify its firmware to compatible with user system.

● Flow chart

The flow chart of Raw data format application software on MicroBlaze is shown Figure 8.

```
            ( Power Up )
                 │
                 ▼
          ┌──────────────┐
          │  Initialize  │
          │  MicroBlaze  │
          └──────────────┘
                 │
                 ▼
          ┌──────────────┐
    ┌────▶│  Send Reset  │
    │     └──────────────┘
    │            │
    │            ▼
    │     ┌──────────────┐
    │     │ Check IP Ready│
    │     └──────────────┘
    │            │
    │            ▼
    │     ┌──────────────┐
    │     │Display Card Size│◀──────────────────┐
    │     └──────────────┘                      │
    │            │                              │
    │            ▼                              │
 Refresh      ◇ Receive command ◇  Read         │
    └────────◇                 ◇────────┐       │
                  ◇           ◇         │       │
                       │ Write         ▼        │
                       ▼         ┌──────────────┐│
          ┌──────────────┐       │ Execute read ││
          │Build data pattern│   └──────────────┘│
          └──────────────┘             │         │
                 │                     ▼          │
                 ▼             ( Display Read     │
          ┌──────────────┐       Speed )          │
          │ Execute write│             │          │
          └──────────────┘             ▼          │
                 │               ◇ Verify? ◇  No  │
                 ▼              ◇         ◇───────▶│
          ( Display Write          │ Yes          │
            Speed )                ▼               │
                 │         ( Display verify        │
                 │           result )              │
                 │                │                │
                 └───────────────▶○────────────────┘
```
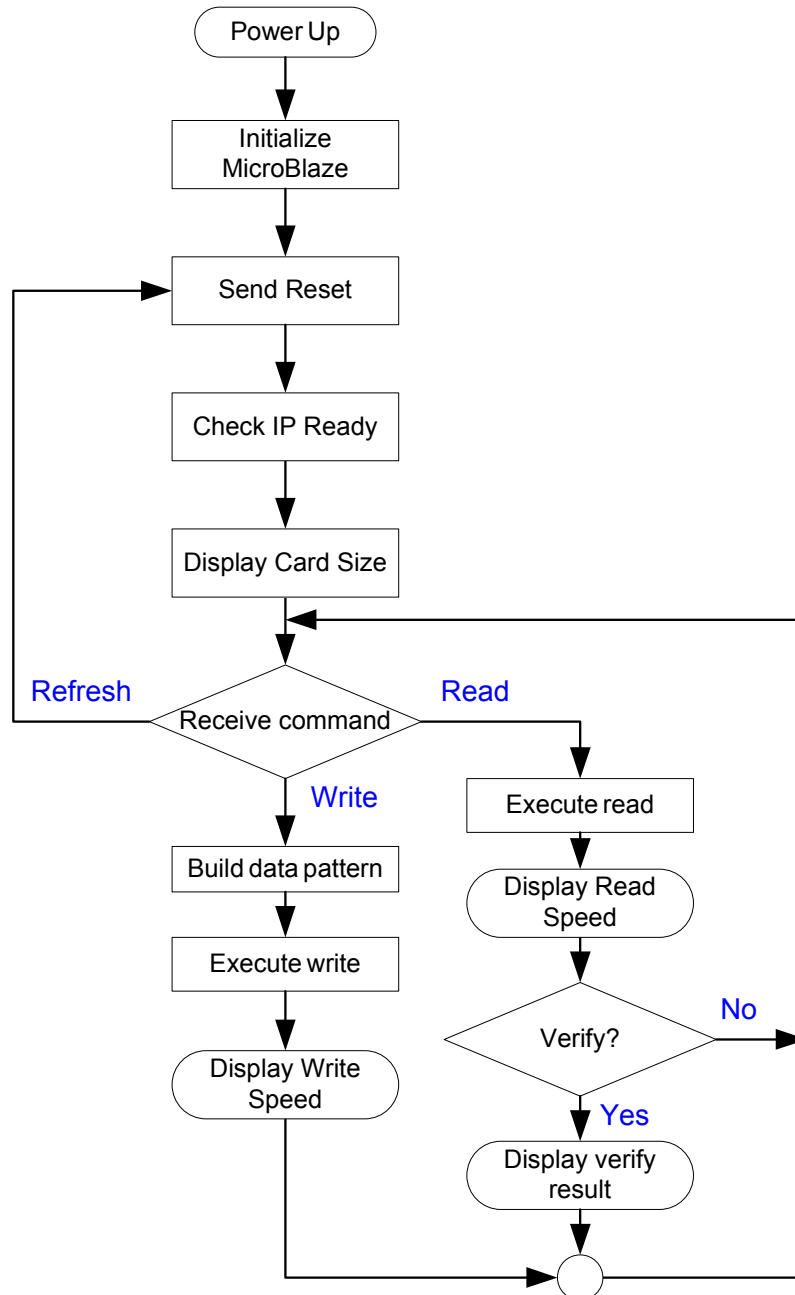
Figure 8 Flow Chart of SDXC-IP Application Firmware

After power-on system and FPGA configuration complete, firmware will set parameter to each hardware peripheral such as interrupt controller, UART controller, Timer. Next, CPU sends reset signal and monitors "Busy" flag from SDXC-IP to wait until card initialize complete. SDXC-IP starts initialize when user inserts SD card to SD slot. Relate to SDXC-IP, there are three functions, i.e. Refresh, Write, and Read SD card. 256 MB of DDR2 is half split to store write and read data at address=0x90000000-0x97FFFFFF and 0x98000000-0x9FFFFFFF in a sequence. So, "TX_ADDR" register is always set to 0x90000000 and "RX_ADDR" register is always set to 0x98000000 for this firmware.

Refresh card is used to send reset signal, check SDCardDetect, wait until SDXC-IP Idle, and display SD card size by reading "STATUS" and "CARD_SIZE" register.

For Write function, user needs to input start address, transfer length, and one of 4 test patterns by Serial Console and CPU will set these values to "TGT_ADDR" and "XFR_COUNT" register. Then, test data patterns, equal to transfer length or maximum at 128 MB size, are stored in DDR2 by CPU. Finally, CPU writes CONTROL register to send write request to SDXC-IP and wait until data transferring complete. Data transfer speed will be displayed to Serial Console as a test result.

For Read function, user needs to input start address and transfer length by Serial Console and these values will be set to "TGT_ADDR" and "XFR_COUNT" register by CPU. CPU writes "CONTROL" register to send read request to SDXC-IP and wait until data transferring complete. Data transfer speed will be displayed to Serial Console as a test result. If transfer length is not more than 128 MB, read data will be verified with expected test pattern. Verify test result will be displayed before ending this function.

After complete function, CPU is idle and waits to receive next command from user.

● Test Result

Some test results by using raw data reference design are shown in Figure 9 and Figure 10. As shown in Figure 9, SDSC Card in default speed mode shows transfer performance that should be less than 12.5 MB/s, and for high speed mode transfer performance should be less than 25 MB/s.
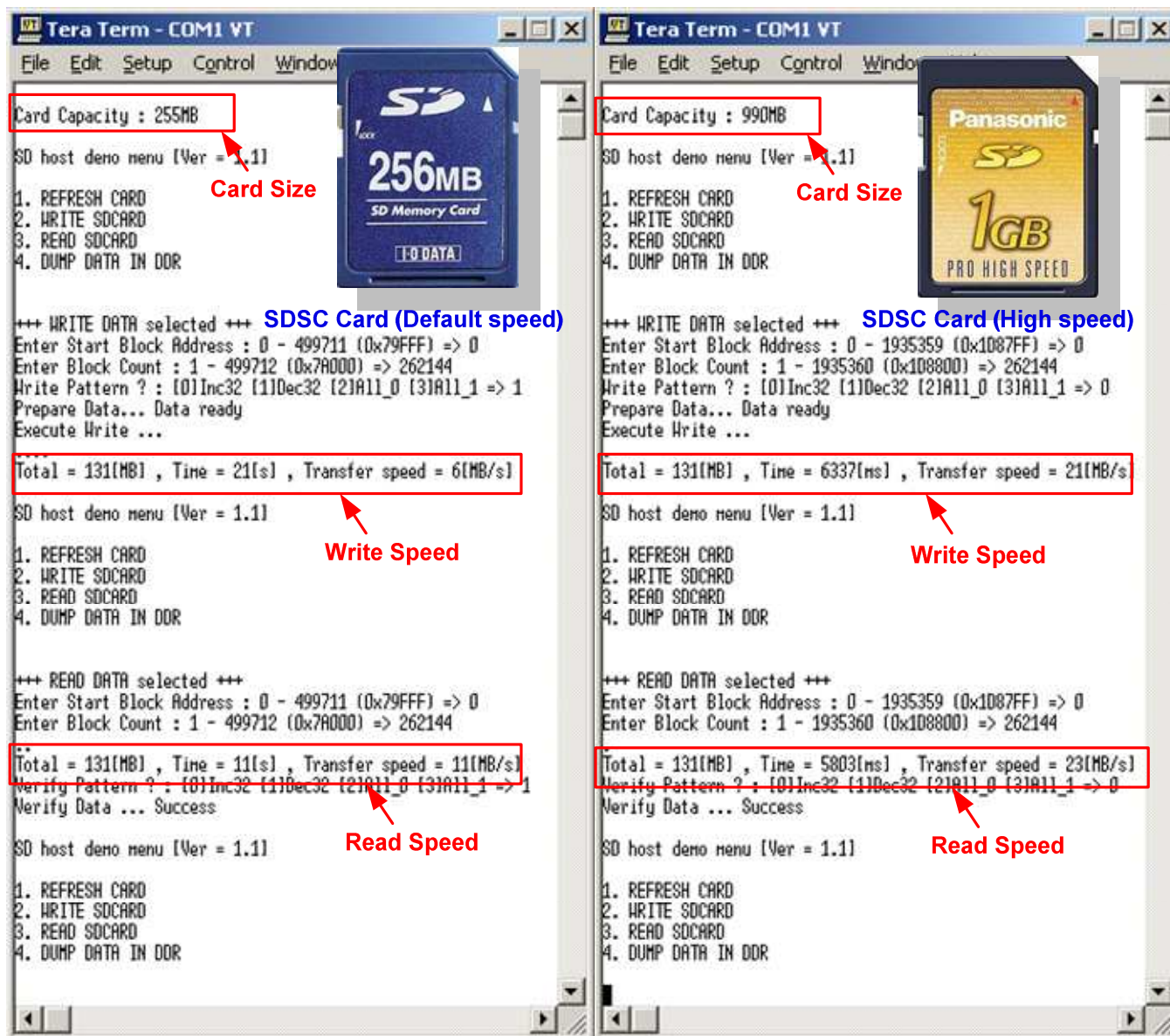

Figure 9 Test Result with SDSC Card in Default and High Speed mode

As shown in Figure 10, transfer speed of SDHC in High speed mode is less than 25 MB/s and transfer speed of SDXC in SDR50 mode is less than 50 MB/s.
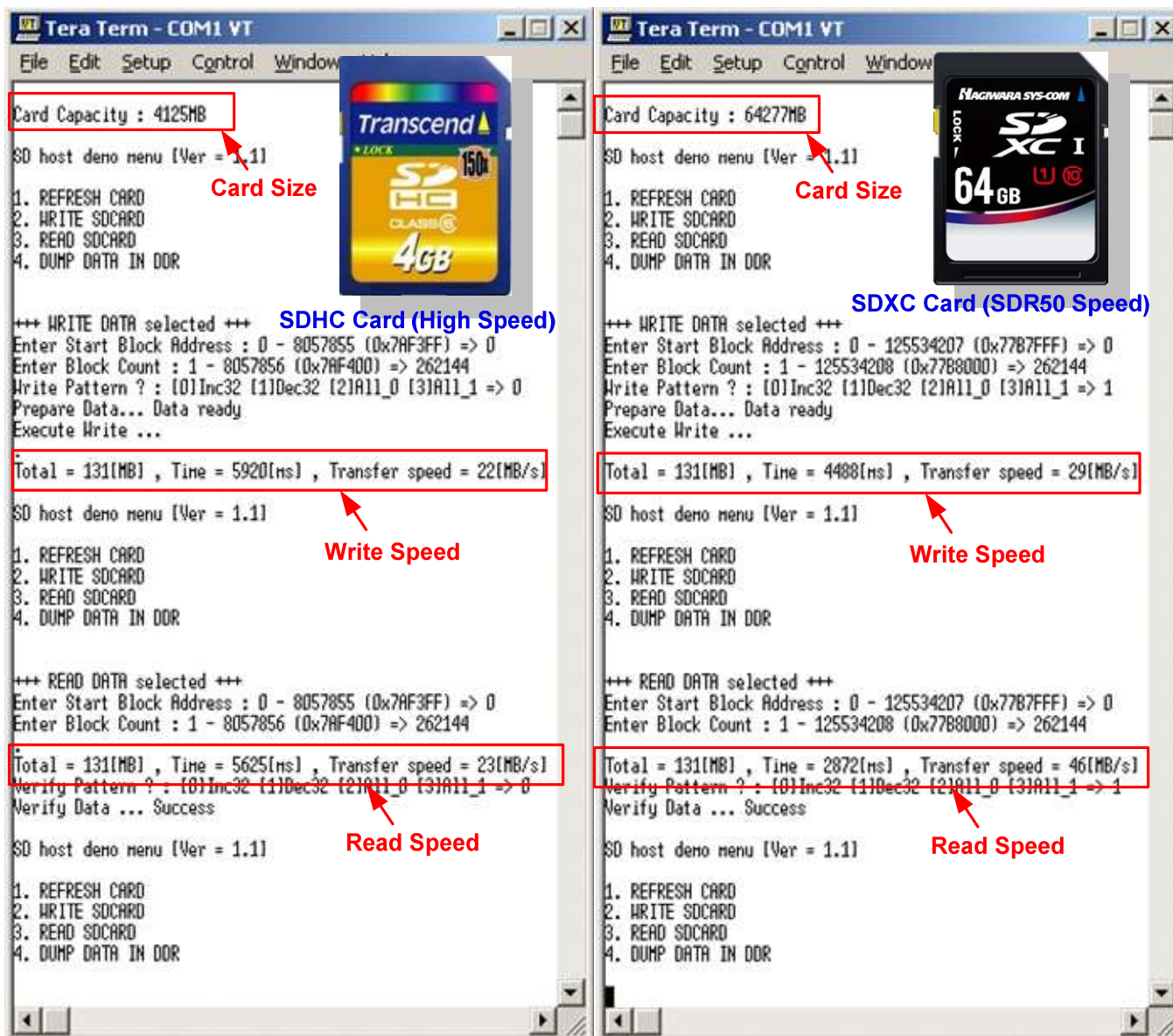

Figure 10 Test Result with SDHC and SDXC Card in High Speed and SDR50 mode

5. Revision History

Revision	Date	Description
1.0	11/Nov/2010	Initial version release

6. Contact

Design Gateway Co.,Ltd

Address: 54 BB Building 13th Fl., Room No.1302 Sukhumvit 21 Rd. (Asoke),
                 Klongtoey-Nua, Wattana, Bangkok 10110
Phone: (+66) -2-261-2277
Fax: (+66) -2-261-2290
E-mail: sales@design-gateway.com
SDXC-IP URL: http://www.dgway.com/products/IP/SDXC-IP/index-E.html

Copyright: 2010 Design Gateway Co,Ltd.