DATASHEET

SDIO3.0 Device Controller

Arasan Chip Systems, Inc.
2010 North First Street, Suite #510
San Jose, CA 95131
Ph:408-282-1600
Fx:408-282-7800
www.arasan.com

April, 2012

DISCLAIMER

This document is written in good faith with the intent to assist the readers in the use of the product. Circuit diagrams and other information relating to Arasan Chip Systems' products are included as a means of illustrating typical applications. Although the information has been checked and is believed to be accurate, no responsibility is assumed for inaccuracies. Information contained in this document is subject to continuous improvements and developments.

Arasan Chip Systems' products are not designed, intended, authorized or warranted for use in any life support or other application where product failure could cause or contribute to personal injury or severe property damage. Any and all such uses without prior written approval of an Officer of Arasan Chip Systems Inc. will be fully at the risk of the customer. Arasan Chip Systems Inc. disclaims and excludes any and all warranties, including without limitation any and all implied warranties of merchantability, fitness for a particular purpose, title and infringement and the like and any and all warranties arising from any course or dealing or usage of trade.

This document may not be copied, reproduced, or transmitted to others in any manner. Nor may any use of information in this document be made, except for the specific purposes for which it is transmitted to the recipient, without the prior written consent of Arasan Chip Systems, Inc. This specification is subject to change at any time without notice. Arasan Chip Systems Inc. is not responsible for any errors contained herein.

In no event shall Arasan Chip Systems Inc. be liable for any direct, indirect, incidental, special, punitive, or consequential damages; or for lost of data, profits, savings or revenues of any kind; regardless of the form of action, whether based on contract; tort; negligence of Arasan Chip Systems or others; strict liability; breach of warranty; or otherwise; whether or not any remedy of buyers is held to have failed of its essential purpose and whether or not Arasan Chip Systems Inc. has been advised of the possibility of such damages.

Restricted Rights
Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor.

Copyright Notice
No part of this specification may be reproduced in any form or means, without the prior written consent of Arasan Chip Systems, Inc.

Questions or comments may be directed to:
Arasan Chip Systems, Inc.
2010 North First Street
Suite 510, San Jose, CA 95131.
Ph: 408-282-1600
Fx: 408-282-7800
Email: sales@arasan.com
http://www.arasan.com

CONTENTS

FIGURES

1 INTRODUCTION

The Arasan SDIO 3.0 Device Controller is a full / high speed card controller suitable for I/O Card applications like WLAN, Bluetooth with low power consumption for mobile devices. The controller supports SPI, 1-bit SD, 4-bit SD and 8-bit bus modes for Embedded SDIO cards. The Arasan SDIO Device Controller has an AHB interface, which allows the ARM Processor to configure the operational registers residing inside the AHB Slave core.

1.1 Overview

The SDIO 3.0 Device IP core is used to implement SDIO cards that are connected to a SD host or card reader over a standard SD bus. The flexible architecture of the SDIO Device IP core is targeted to develop a range of portable, low-power cards such as the WiFi, GPS and LTE.

The SDIO 3.0 Device IP core is fully compliant with the SD Specification Part E1 SDIO 3.0. It supports SPI, SD1 and SD4 bit transfer modes and multiple functions per card. High-speed and full-speed SD data transfers are also supported. All SDIO 3.0 features are supported including the UHS-I, SD HC, miniSDIO, embedded SDIO ATA standard function interface code and operating voltages 2.7-3.6V or 1.7-1.95V. In applications with an AHB interface, the SDIO Device controller IP is controlled by an ARM processor. The SDIO Device controller includes a bidirectional FIFO that is expandable from 4 x 32-bit to any size required. The core supports asynchronous interrupts to the SD Host for improved performance. It supports suspend/resume operation for improved performance.

The controller integrates a scatter gather DMA engine automating data transfers between the SDIO card and system memory. The SDIO Device core is available with many system bus interfaces including AHB, AXI, OCP and custom buses. The wide selection of system bus interfaces enables the core to integrate effectively SoC designs today.

1.2 Features

- Compliant with SD Specification Part E1 SDIO Specification 3.0
- Supports Asynchronous Interrupt to Host controller
- Enhanced power management using new Power State Control function
- Supports Read Wait Control, Suspend/ Resume operations for superior card performance
- High-performance UHS-I (SDR104 up to 104MB/s)
- Multiple I/O functions and one memory supported

- Host clock rate from 0 to 208 MHz
- Supports SPI, 1-bit and 4-bit SD modes
- Optional 8-bit mode for embedded SDIO
- Supports all SDIO form factors including standard, mini and micro SDIO card
- Embedded SDIO ATA interface code
- Bus Master with Scatter Gather DMA
- Dual operating voltage range 2.7V – 3.6V and 1.7V – 1.95V
- Up to 104 MB/s read/write with 4-bit data lines in SD4 mode
- Cyclic Redundancy Check (CRC7) (command), CRC16 (data) integrity
- Supports direct R/W (IO52) and extended R/W (IO53)
- Programmable through AMBA 2.0 AHB bus

1.3 Bus Topology

The SD Card defines two alternative communication protocols SD and SPI.

1. SD
    - SD 1-bit
    - SD 4-bit

2. SPI

Applications can choose either one of modes. UHS-I cards does not support SPI mode and SD 1-bit mode. Mode selection is transparent to the Host. After reset command the Card automatically moves to default mode and will expect all further communication to be in the same mode. Therefore, applications which uses only one communication mode do not have to be aware of the other.

SD

SD bus allows dynamic configuration of the number of data lines. After power up by default, the SD card will use one DAT0 for data transfer. After initialization the Host can change the bus width (number of active data lines). This feature allows easy trade off between HW cost and system performance.

8-bit bus mode can be supported only for an Embedded device. Achieving higher performance in the lower frequency is the objective of this mode. In
SD mode DAT0 is used for data transfer and busy signalling, DAT1 is used for data transfer and Interrupt signalling, DAT2 is used for data transfer and read_wait, DAT3 is used for data transfer.

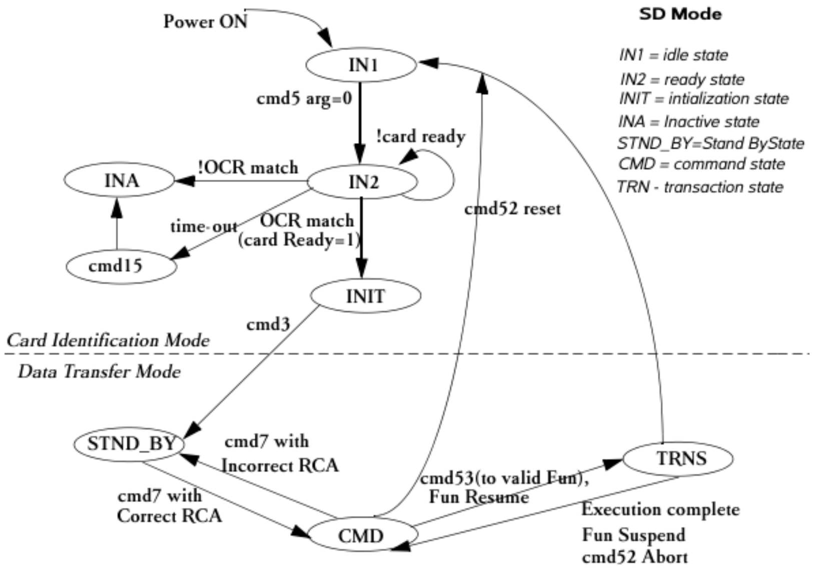

Figure 1: SDIO Card Initialization Sequence for SD Mode

SPI

The SPI mode is selected during the first reset command after power up and cannot be changed as long as the part is powered on. From the application point of view, the advantage of the SPI mode is the capability of using an off-the-shelf host, hence reducing the design-in effort to minimum. The disadvantage is the loss of performance, relatively to the SD mode which enables the wide bus option. The SPI interface uses the 8 out of the SDIO 9 signals (DAT 2 is not used, DAT1 for interrupt signalling, DAT3 for CS signalling) of the SD bus.

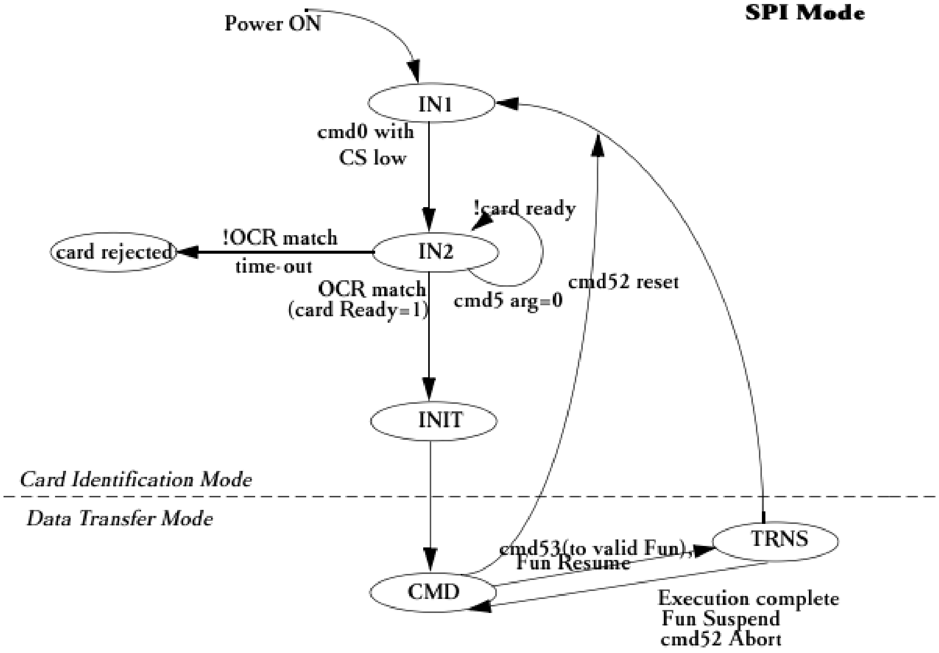

Figure 2: SDIO Card Initialization Sequence for SPI Mode

Note:

- To access the card being already in inactive state, a hard reset must be done by switching the power supply off and on.
- Card ready time-out: Time-out for Initialization process is 1 second

2 ARCHITECTURE

This section contains the architecture of SDIO Device Controller and the description of internal blocks in detail.

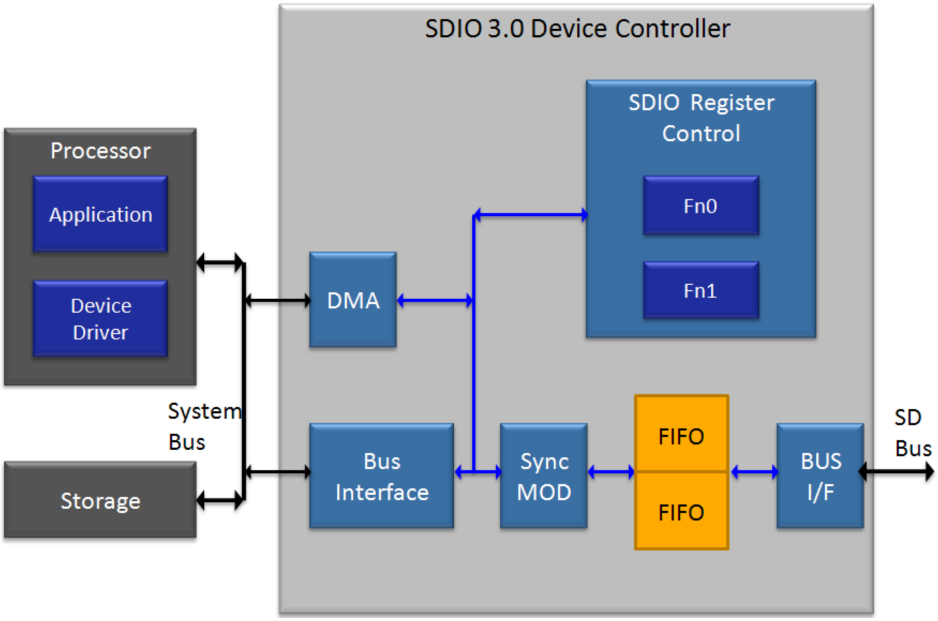

Figure 3: SDIO3.0 Device Controller Architecture

2.1 Bus Interface Unit (BIU)

The BIU communicates with the SD Host through the SD bus. SD1, SD4, SPI and 8-bit mode for embedded device are supported. The BIU houses the 16-bit CRC generator and checker for the data lines, 7-bit CRC generator and checker for the command and response lines, transmitter state machine, receiver state machine, interrupt state machine, BIU master state machine, command decoder and the response generator. The BIU bus capability is determined by bit values programmed in the R/W CCCR registers.

2.2 fn0_register

The fn0_register module contains the CIA (Common I/O Area). There are three distinct register structures supported within the CIA. They are Card Common Control Registers (CCCR), Function Basic Registers (FBR), Card Information Structure (CIS). The CIA is accessed by the Host via I/O reads and writes to function 0. The registers within the CIA also provided to enable/disable the operation of the I/O function(s), control the generation of interrupts. The registers in the CIA also provide information about the function(s) abilities and requirements.

2.3 fn_reg

The fn_reg module contains the funcion1 Host access registers. These registers are accessed by the Host through I/O reads and write to function 1. Whenever Interrupt asserted from the controller the Host should read these Registers to identify the source of interrupt. The Host can able to write these registers using cmd52 and cmd53. Host also uses these registers to send a message to ARM.

2.4 Synchronization Module (sync_mod)

The synchronization module has handshake logic to communicate with the BIU and on the other side communicates with the AHB Master and Slave Core.

2.5 FIFO Control

The FIFO control block contains one dual port FIFO for performing both read and write transactions. During write transaction (data transferred from Host to ARM), the data will be filled in to the first and second half of the FIFO alternatively. When data from first half of the FIFO is transferring to ARM, the second half of the FIFO will be filled and vice versa. The two FIFO’s are alternatively used to store data which will give maximum throughput. During a read transaction (data transferred from ARM to Host) the data from ARM will be written in to the two half of the FIFO’s alternatively. When data from one half of the FIFO is transferring to Host, the second half of the FIFO will be filled and vice-versa and thereby the throughout will be maximum.

2.6 Reg Control

The reg control block contains the function0, function1 ARM access registers. ARM Processor read/write these registers through AHB slave interface.

2.7 AHB Interface

The AHB Master is responsible for transferring data between the ARM Processor and the SDIO-AHB bridge for read and write operations using Scatter Gather DMA. AHB Slave Block houses the AHB slave interface signals. All the operational registers are in the reg control module. Reading and writing of these registers are handled by the SDIO-AHB bridge or ARM Processor. The ARM Processor should set the function ready bit in ESW Fun Ready register when it is ready to operate, indicating to the Host that all initialization has been done and Function is ready to operate.

2.8 ADMA Interface

The ADMA interface is responsible for controlling the ADMA transaction with system memory. It gives control signals to AHB Master Interface to initiate the AHB Bus transaction. This interface keeps performing ADMA fetch transaction with system memory whenever the described data length is over. It decodes the data address from the descriptor table and loads the same to AHB Master and also gets the data length from descriptor table and loads the number of bytes to AHB Master. If ADMA end bit is set in the descriptor table it will assert ADMA exhausted interrupt to ARM.

3 SIGNALS

The following section describes the direction of the pins from interfaces such as SD, AHB Master, AHB Slave, System Interface and RAM interface signals.
The Arasan SDIO Controller has four main interface groups.

1. SD Interface
2. AHB Slave Interface
3. AHB Master Interface
4. System Interface

3.1 SD Interface

Signal	DIR	Description
clk_sd	Input	SD Clock
clk_sd_inv	Input	Inverted SD Clock
SD_CMD	INOUT	SD4 bit mode: Command Line
		SD1 bit mode: Command Line
		SPI mode: Data Input
cmd_di	IN	Command Input
sel_sd_resp	OUT	Command Output
sel_resp_en	OUT	Command Output Enable
SD_DAT0	INOUT	SD4 bit mode: Data Line 0
		SD1 bit mode: Data Line
		SPI mode: Data Output
dat0_do	IN	Data0 Input
sel_tx_dat0	OUT	Data0 Output
sel_tx_dat0_en	OUT	Data0 Output Enable
SD_DAT1	INOUT	SD4 bit mode: Data Line1 or Interrupt (optional)
		SD1 bit mode: Interrupt
		SPI mode: Interrupt
dat1_irq	IN	Data1 Input
sel_tx_dat1	OUT	Data1Output
sel_tx_dat1_en	OUT	Data1 Output Enable
SD_DAT2	INOUT	SD4 bit mode: Data Line2 or Read Wait (optional)
		SD1 bit mode: Read Wait (optional)
		SPI mode: Not Used
dat2_rw	IN	Data2 Input
sel_tx_dat2	OUT	Data2 Output
sel_tx_dat2_en	OUT	Data2 Output Enable

Signal	DIR	Description
SD_DAT3	INOUT	SD4 bit mode: Data Line 3 SD1 bit mode: Not Used SPI mode: Card Select
dat3_cs	IN	Data3 Input
sel_tx_dat3	OUT	Data3 Output
sel_tx_dat3_en	OUT	Data3 Output Enable

3.2 AHB Slave Interface

Signal	DIR	Description
clk_ahb	Input	AHB System Clock
ahb_sdio_haddr[16:0]	Input	Address Bus (Byte Addresses)
ahb_sdio_hwdata[31:0]	Input	Write Data Bus
ahb_sdio_hrdata[31:0]	Output	Read Data Bus
ahb_sdio_hwrite	Input	Write or Read Direction Indication
ahb_sdio_hsize[2:0]	Input	Size (Byte, Half Word or Word)
ahb_sdio_htrans[1:0]	Input	Transfer Type
ahb_sdio_hready_glb	Input	Global Ready
ahb_sdio_hready	Output	Slave Ready
ahb_sdio_hresp[1:0]	Output	Transfer Response
int_to_arm[30:0]	Output	Interrupt to Processor

3.3 AHB Master Interface

Signal	DIR	Description
ahb_sdio_dma_hbusreq	Output	AHB Bus request
ahb_sdio_dma_hgrant	Input	AHB Bus Grant
ahb_sdio_dma_haddr[31:0]	Output	AHB Slave DMA address (byte addresses)
ahb_sdio_dma_hwdata[31:0]	Output	AHB write data
ahb_sdio_dma_hrdata[31:0]	Input	AHB read data
ahb_sdio_dma_hwrite	Output	Write / Read Direction Indication
ahb_sdio_dma_hsize[2:0]	Output	Size (byte, half word or word)
ahb_sdio_dma_hburst[2:0]	Output	Burst Size
ahb_sdio_dma_hrdyglb	Input	Global ready signal
ahb_sdio_dma_htrans[1:0]	Output	Transfer type
ahb_sdio_dma_hresp[1:0]	Input	Transfer response

3.4 System Interface

PINS	DIR	Description
pwr_on_rst_n	Input	Active Low. Asynchronous Hardware reset from the External environment
ahb_clk_wkup	Output	Active high signal to wakeup AHB clock. At power-on-reset this signal will be asserted to 1'b1 as the chip is in manual mode when manual clk enable bit is set to 1'b1 in Clock wakeup register. When auto clk enable bit in Clock wakeup register is 1'b1, SDIO controller will enable the ahb_clk_wkup depending on the activity on the bus when required
pullup_en	Output	Pull-up Enable from the Arasan SDIO-AHB bridge for Card Detection
scan_mode	Input	Active Low. If set as 1'b1 - bypass all reset pulses generated internally 1'b0 - POR will be asserted asynchronously.

4 VERIFICATION ENVIRONMENT

This section explains the test environment of SDIO controller and the block diagram shows the integration of the SDIO controller. The SDIO core has been verified in the simulation environment using the behavioral models of the surrounding environment (RTL verification). The RTL verification environment for SDIO core consists of the behavioral models to emulate the SD host and ARM processor interface.

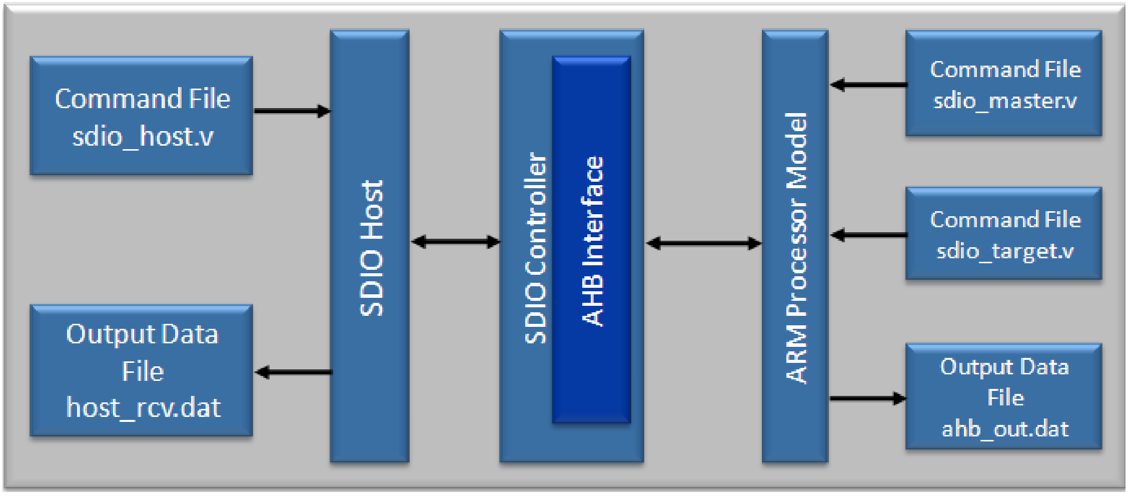

Figure 4: Verification Environment of SDIO Controller IP with AHB System Bus

5 IP DELIVERABLES

The full IP package complete with:

- Verilog HDL of the IP core
- Test environment and test scripts
- Synthesis scripts
- User Guide

6 RELATED PRODUCTS

- SD3.0/SDIO3.0/eMMC4.5 Host Controller IP
- SD3.0/eMMC4.5 Host Stack Software
- SD3.0/eMMC4.5 Hardware Development Platform (HVP)