Freescale Semiconductor
Application Note

Document Number: AN3049
Rev. 0, 10/2005

Setup and Use of the Multimedia Card/Secure Digital Host Controller

MC9328MX1, MC9328MXL, and MC9328MX21

by: Jason Castillo

1 Abstract

This document describes the Multimedia Card (MMC) and the Secure Digital (SD) modules on Freescale Semiconductor's i.MX application processors. This document gives an overview of the similarities and differences of both the MMC and SD protocols and cards including configuration of the MMC/SD module. The document also describes the interface of the MMC/SD modules, specifically the card identification mode and the system clock controller, along with example code. System considerations at various data transfer rates are included.

This document applies to the following devices, collectively called i.MX throughout:

- MC9328MX1
- MC9328MXL
- MC9328MX21

Contents

2 MMC/SD Module Overview

This section provides an overview of both the Multimedia Card (MMC) and Secure Digital Card (SD) modules. The MMC/SD module includes the following features:

- Supports up to 10 cards (including one SD card)
- Password protection for cards
- Built-in programmable frequency counter for MMC/SD bus
- Maskable hardware interrupt for card detection (insertion/removal), SD I/O interrupt, internal status, and FIFO status
- Contains an integrated 32 × 16-bit FIFO
- Supports plug-and-play (Pop)
- Supports many SD functions including multiple I/O and combined I/O and memory
- Supports up to seven I/O functions plus one memory on a single SD I/O card
- Card can interrupt MMC/SD module
- Support single or multiple block access, or stream access to the card for read, write, or erase operations
- Supports SD I/O ReadWait and interrupt detection during 1- or 4-bit access

2.1 Multimedia Card (MMC) Protocol

The MMC is low cost data storage and communication medium implemented as a hardware card with a simple control unit and a compact, easy-to-implement interface that is designed to cover a wide variety of applications. It is based on an advanced 7-pin serial bus designed to operate over a voltage range of 2.0 to 3.6 V. Note that the maximum operating voltage range for the i.MX is 3.3 V. Therefore the i.MX-based design is limited to a supply range of 2.0 to 3.3 V.

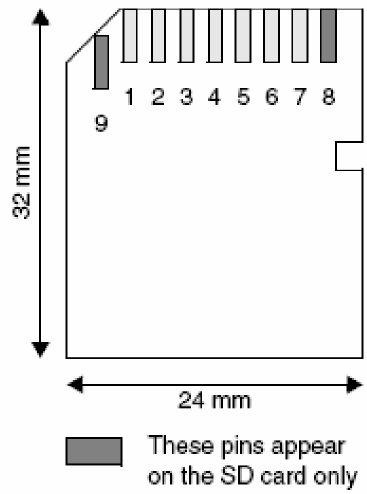
2.2 Secure Digital Card (SD) Protocol

The SD is an extended version of the MMC with two additional pins. The additional pins are designed to meet the security, capacity, performance, and environmental requirements inherent in new audio and video consumer electronic devices. The physical form factor and data transfer protocol are compatible with the MMC. The SD is composed of a memory card and an I/O card.

3 MMC/SD Module Configuration

The MMC card has 7 pins and the SD card has 9 pins. The pins are used to communicate with other functions within the card. The pin assignment and form factor are shown in Table 1.

Table 1. MMC/SD Card Pin Assignment

<table>
<tr><th rowspan="2">Form Factor and Pinout</th><th rowspan="2">Pin Number</th><th rowspan="2">MMC Card</th><th colspan="2">SD Card</th></tr>
<tr><th>1-Bit Mode</th><th>4-Bit Mode</th></tr>
<tr><td rowspan="9">
</td><td>1</td><td>Reserved</td><td>Card Detect</td><td>Data Line DAT [3]</td></tr>
<tr><td>2</td><td colspan="3">Command / Response (CMD)</td></tr>
<tr><td>3</td><td colspan="3">Supply Voltage Ground (Vss1)</td></tr>
<tr><td>4</td><td colspan="3">Supply Voltage (Vdd)</td></tr>
<tr><td>5</td><td colspan="3">Clock (CLK)</td></tr>
<tr><td>6</td><td colspan="3">Supply Voltage Ground (Vss2)</td></tr>
<tr><td>7</td><td colspan="3">Data Line DAT [0]</td></tr>
<tr><td>8</td><td></td><td>Interrupt (IRQ)</td><td>Data Line DAT [1] or Interrupt (IRQ)</td></tr>
<tr><td>9</td><td></td><td>ReadWait (RW)</td><td>Data Line DAT [2] or ReadWait (RW)</td></tr>
</table>

Each card has a set of information registers that hold the operating parameters and other card conditions. See the description in Table 2. Details of each register can be found in card registers chapter in both the MultiMediaCard System Specification and the SD Memory Card Specification.

Table 2. MMC/SD Card Registers

MMC or SD	Identifier	Register Name	Description	Size (Bits)
Both	CID	Card Identification Number	Each card has a unique CID.	128[1]
Both	RCA	Relative Card Address	Assigned by the MMC/SD module during initialization.	16
Both	DSR	Driver Stage	Configures the card's output drivers. Use is optional, not required.	16
Both	CSD	Card Specific Data	Contains information on the card's operating conditions.	128[1]
Both	OCR	Operation Conditions	Indicates the card's operating voltage range. Detects restricted cards and indicates whether power-up is complete. Use is optional, not required.	32
Both	CSR	Card Status Register	Contains card's error and status information. Sent to the MMC/SD module in response format R1.	32
SD only	SCS	SD Card Status	Contains status information proprietary to the SD card (such as protection, card type, and bus width).	512
SD only	SCR	SD Configuration Register	Contains additional configuration information only applicable to the SD card.	64

[1] There can be fewer bits for the SD I/O Card, depending on implementation.

3.1 Pin Configuration

The MMC/SD module uses six I/O pins to communicate with external MMC/SD cards.

- **SD_CMD**—Bidirectional command/response signal between the MMC/SD module and the card. Open-drain for initialization state and push-pull for fast command transfers.
- **SD_Clk**—MMC/SD module to card clock signal (output).
- **SD_DAT [3:0]**—Four bidirectional data signals. When in push-pull mode, one card or the MMC/SD module can drive each line at a time.

These six I/O pins are multiplexed with other functions on the i.MX devices and must be configured for MMC/SD module operation. Explanation of the pin configuration can be found in the reference manual. Example 1 is code sample of the pin configuration. The data direction register (DDIR) is not necessarily required to initialize, however, it is highly recommended, especially when working with multiple modules.

Example 1. Pin Configuration for MMC/SD Card on the i.MX1/L

```
/**************** Initialize ****************/
/*  Set up GPIO for MMC/SD                   */
/*******************************************/
void MMC_SD_Port_Initialize()
{

    // SD_CMD - Primary function of GPIO Port B [13]
    GIUS_B  &= ~0x00002000;     // CLEAR BIT 13
    GPR_B   &= ~0x00002000;     // CLEAR BIT 13
    PUEN_B  |=  0x00002000;     // SET BIT 13

    // SD_CLK - Primary function of GPIO Port B [12]
    GIUS_B  &= ~0x00001000;     // CLEAR BIT 12
    GPR_B   &= ~0x00001000;     // CLEAR BIT 12

    // SD_DAT[3] - Primary function of GPIO Port B [11]
    GIUS_B  &= ~0x00000800;     // CLEAR BIT 11
    GPR_B   &= ~0x00000800;     // CLEAR BIT 11

    // SD_DAT[2] - Primary function of GPIO Port B [10]
    GIUS_B  &= ~0x00000400;     // CLEAR BIT 10
    GPR_B   &= ~0x00000400;     // CLEAR BIT 10
    PUEN_B  |=  0x00000400;     // SET BIT 10

    // SD_DAT[1] - Primary function of GPIO Port B [9]
    GIUS_B  &= ~0x00000200;     // CLEAR BIT 9
    GPR_B   &= ~0x00000200;     // CLEAR BIT 9
    PUEN_B  |=  0x00000200;     // SET BIT 9

    // SD_DAT[0] - Primary function of GPIO Port B [8]
    GIUS_B  &= ~0x00000100;     // CLEAR BIT 8
    GPR_B   &= ~0x00000100;     // CLEAR BIT 8
    PUEN_B  |=  0x00000100;     // SET BIT 8

    DDIR_B |= 0x00003F00;           // Initialize SDHC's port

    GPCR &= 0xFFFFF3FF;     // DS_SLOW = 26 MHz / 15 pF
```

```
//   GPCR  |= 0x00000400;      // DS_SLOW = 26 MHz / 30 pF
//   GPCR  |= 0x00000800;      // DS_SLOW = 26 MHz / 45 pF
//   GPCR  |= 0x00000C00;      // DS_SLOW = 26 MHz / greater than 45 pF

    /******************************************/
    /* Include the AIPI Module Configuration */
    /******************************************/
    PSR0_2 &= ~0x00000010;
    PSR1_2 |=  0x00000010;
    PAR_2  |=  0x00000010;
}
```

Notice in the code sample that the last five registers are included within the pin configuration. The DDIR register for the i.MX1/L is set to 0x00003F00 to enable the SDHC ports to output. Because the i.MX21 can have two SDHC ports, set both PTB_DDIR and PTE_DDIR registers to 0x000003F0 and 0x00FC0000 respectively. The GPCR is the peripheral control register which is used to control the driving strength. When using the MMC/SD module the driving strength must be set to 15 pF; otherwise overshoot and ringing problems can occur on the SD_CLK. Setting the driving strength on the i.MX21 is explained in the driving strength section of reference [6]. The next three registers (PSR0_2, PSR1_2, and PAR_2) are used to allow the IP bus peripheral to send 32 bits and whether it can be accessed in user mode.

4 Interfacing with the MMC/SD Module

This section describes the different functional aspects of the MMC/SD module.

4.1 Card Identification Modes

The MMC/SD module is in identification mode after reset and during the search for new cards on the bus. During the identification mode, the MMC/SD module performs the following steps which are explained in more detail in the next sections:

1. Detects the cards
2. Resets all cards that are in the card identification state
3. Validates operation voltage range
4. Identifies the cards
5. Asks each card (separately, on its own SD_CMD line) to publish its relative card address (RCA)

4.1.1 Card Detect

Card detection occurs automatically through the SD_DAT [3] pin. Only the SD card has this feature because the MMC does not have the SD_DAT [3] pin. The detection circuitry is sampled by the i.MX system clock (HCLK) so the state of the MMC/SD module does not matter. After the card is detected, the user must mask the card detection interrupt (INT_MASK |= 0x40) to avoid misleading interrupt generation during card access as the SD_DAT lines change.

4.1.2 Reset

When the i.MX is powered up, all cards (including the cards having been in inactive state) are in idle state. Sending the command GO_IDLE_STATE (CMD0) also sets all cards into idle state. After the cards are reset through the hardware or software (CMD0):

- All cards' output bus drivers are in high-impedance state
- The cards are initialized with a relative card address (RCA) of 0x0001
- The cards are configured with a default driver stage register setting (lowest speed, highest driving current capability).

In identification mode, low frequency is required for MMC cards (0 – 400 kHz).

4.1.3 Voltage Validation

All valid cards should be able to establish communication with the processor. If the card is not compatible with the valid V_{DD} range, the card will not be able to complete the identification mode.

SD_APP_OP_COND (ACMD41) is a special command for the SD cards to identify and reject cards that are not within the V_{DD} range. Because SD_APP_OP_COND is an application command, APP_CMD (CMD55) must precede ACMD41.

MMC cards do not respond to CMD55. To recognize the MMC card, first send CMD55 and confirm whether a time-out has occurred. If a time-out has occurred, then check for the MMC card by sending the SEND_OP_COND (CMD1) command and check again for a time-out. If a time-out occurs after CMD1 then the following possibilities may have occurred:

1. No card present or incorrectly installed
2. Card does not meet the valid V_{DD} range
3. Frequency is greater than 400 kHz

4.1.4 Card Registry

After validating the voltage for the card(s) and activating the bus, the module requests the card(s) to send their operation conditions before going to the identification state. To request the operation conditions for the card(s), the user must send the commands CMD55 followed by ACMD41 for the SD cards. For the MMC cards the user must send the command CMD1. Afterwards the user must send CMD2 to each card to get its unique card identification (CID) number. The last step is to send the SEND_RELATIVE_ADDR (CMD3) command to publish a new relative card address. After the new RCA has been published the state changes to standby state and the output switches from open-drain to push-pull. The RCA is used in the data transfer mode.

4.2 System Clock Controller

Two clock stages are used in the MMC/SD module to maximize the power-saving capabilities. The input clock operates around 20–100 MHz and passes through a prescaler that adjusts the inner clock under 20 MHz, which is the maximum operation frequency of the module. This clock is then passed through a user-programmable clock divider. The divider is used to program the final MMC/SD bus clock for different card applications. The resulting clock is used throughout the module. Figure 1 illustrates the clock controller module.

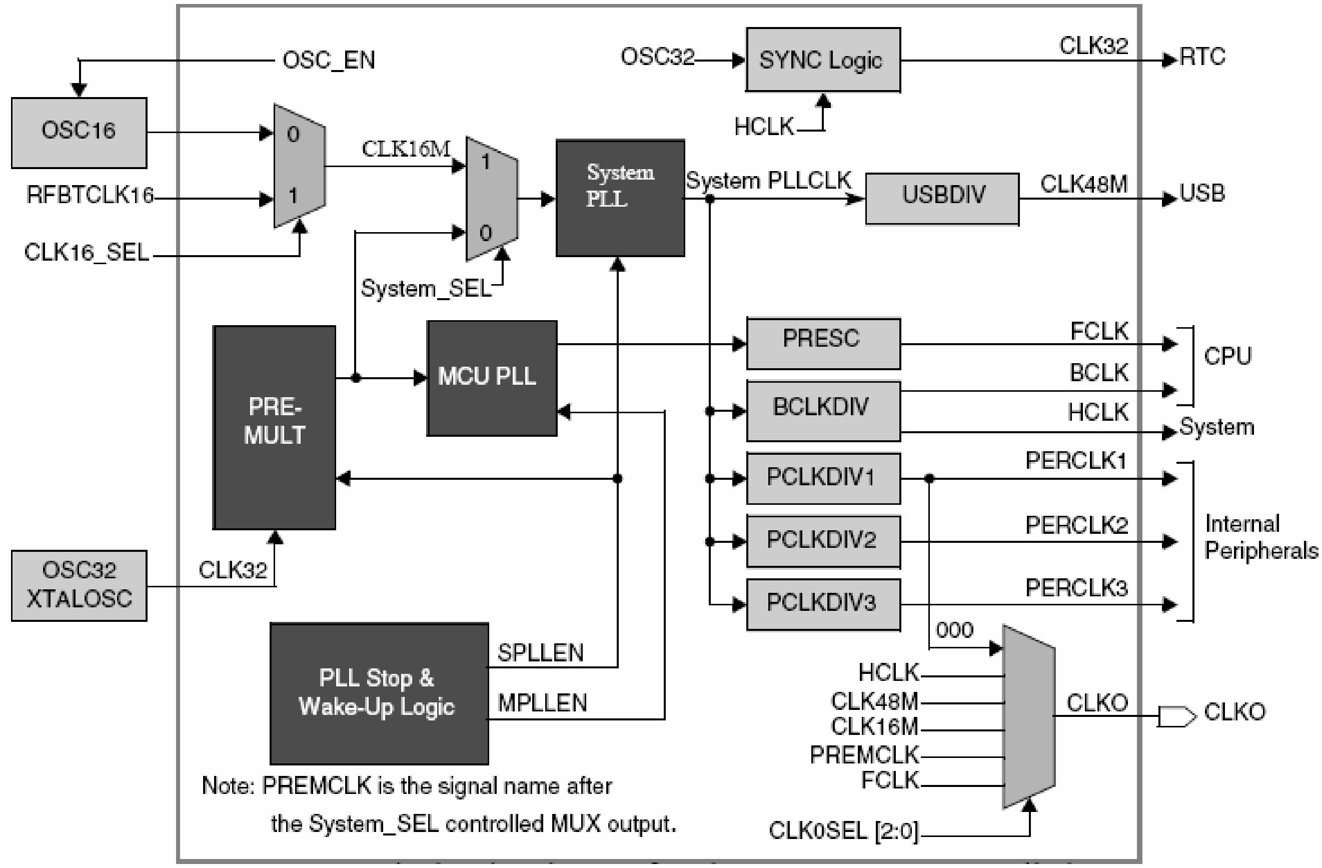


Figure 1. i.MX1/L Clock Controller Module (MMC/SD uses PERCLK2 Signal)

To set the clock controller to desire frequency, you must configure the clock controller register: CSCR, PCDR, MPCTL0, MPCTL1, SPCTL0, and SPCTL1. Example 2 is a code sample to configure the clock with an output frequency of 406.25 kHz. For the i.MX21 there is an extra gate used for each peripheral clock to conserve power when other peripherals are not in use. The extra gate is not shown in the code example.

Example 2. Clock Controller Initialization for i.MX1/L

```
//----------------------------- Set up premultiplier PLL ------------------------------//
/*  CLKO_SEL[31:29]      =    HCLK (CLKO pin)
 *  USB_DIV[28:26]       =    000
 *  SD_CNT[25:24         =    00
 *  SPLL_RESTART[22]     =    RESTARTS SYSTEM PLL AT NEW FREQUENCY (0)
 *  MPLL_RESTART[21]     =    RESTARTS MCU PLL AT NEW FREQUENCY (0)
 *  CLK16_SEL[18]        =    0
 *  OSC_EN[17]           =    0
 *  SYSTEM_SEL[16]       =    CLOCK SOURCE IS THE INTERNAL PREMULTIPLIER
 *  PRESC[15]            =    PRESCALER DIVIDES BY 2
 *  BLCK_DIV[13:10]      =    SYSTEM PLLCLK DIVIDED BY 1
 *  SPEN[1]              =    SYSTEM PLL ENABLED
 *  MPEN[0]              =    MCU PLL ENABLED
 */
```

```
    CSCR = 0x20008003;
    PCDR = 0x0;             /* PERCLK2 feeding SD = Sys PLL(16.384) divided by 1 */
//-----------------------------------------------------------------------------------//
//------------------------------- Set up Digital PLL ---------------------------------//
// NOTE: Digital PLL (DPLL) consists of both the MCU PLL and System PLL
// Input Frequency = 16.384 MHz
// MFI = 5; MFN = 46; MFD = 82; PD = 13
// Actual Tolerance = 0.000323%
// Actual Frequency = 13.000042 MHz
    MPCTL0 = 0x3452142e;
    SPCTL0 = 0x3452142e;
    MPCTL1 = 0;
    SPCTL1 = 0x00008000;
    CSCR  |= 0x600000;
    // CLK_RATE = 0x0:  CLK_DIV = CLK_20M     = 13      MHz
    // CLK_RATE = 0x1:  CLK_DIV = 13/2   MHz  = 6.5     MHz
    // CLK_RATE = 0x2:  CLK_DIV = 13/4   MHz  = 3.25    MHz
    // CLK_RATE = 0x3:  CLK_DIV = 13/8   MHz  = 1.625   MHz
    // CLK_RATE = 0x4:  CLK_DIV = 13/16  MHz  = 812.5   KHz
    // CLK_RATE = 0x5:  CLK_DIV = 13/32  MHz  = 406.25  kHz
    // CLK_RATE = 0x6:  CLK_DIV = 13/64  MHz  = 203.125 kHz
    // CLK_RATE = 0x7:  CLK_DIV = 13/128 MHz  = 101.563 kHz
    CLK_RATE = 0x5;
```

4.3 Data Transfer Mode

After the card identification mode is set, it is then switched over to the transfer state where it waits for either a write or read command. This section provides brief information about four different data transfer modes:

- Block write
- Block read
- Stream write and read

4.3.1 Block Write

Blocks of data are written by sending data from the host to the MMC/SD card. Writing occurs during the data transfer mode. Commands for writing are as follows:

- CMD24 WRITE_BLOCK
- CMD25 WRITE_MULTIPLE_BLOCK
- CMD26 PROGRAM_CID
- CMD27 PROGRAM_CSD
- CMD42 LOCK_UNLOCK
- CMD56 GEN_CMD

After a write command is implemented, the SD_CLK automatically stops and confirms whether the FIFO is empty. When the FIFO is empty the SD_CLK starts and data is sent to the FIFO. The FIFO is 512 bytes which is equivalent to the MMC/SD module the block size requirements. When the FIFO is full, the SD_CLK stops automatically and sends the data to the MMC/SD card. The process of fetching another block of data then repeats. Figure 2 is a screenshot of both the SD_CLK and the SD_DAT [0] pins running on the i.MX1 using DMA transfer.

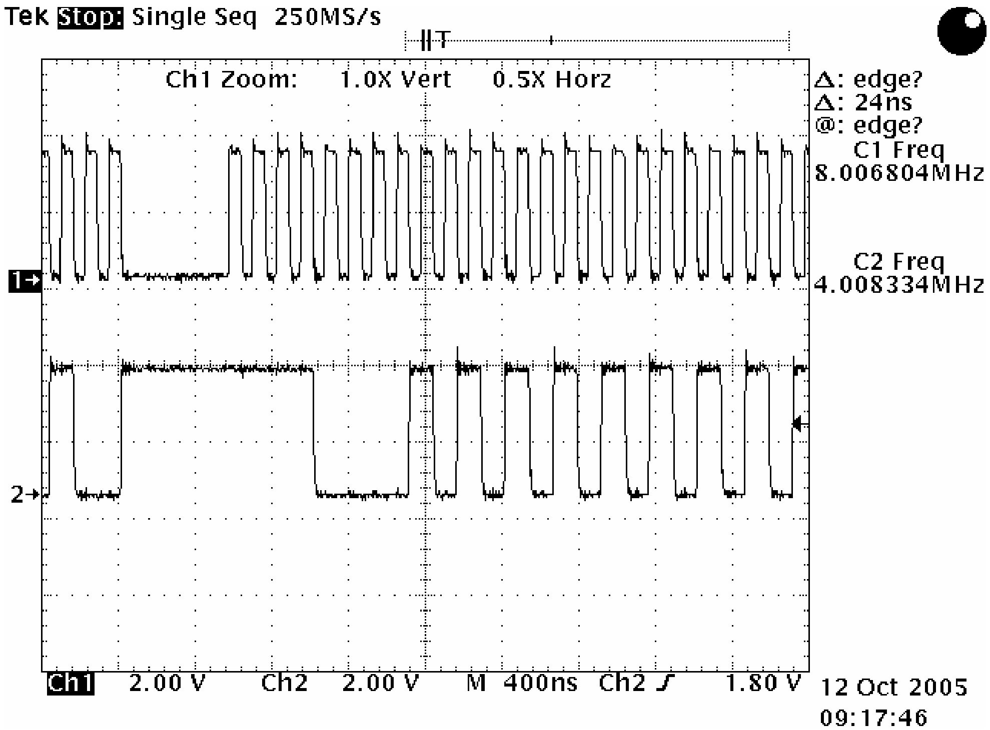


Figure 2. SD_CLK (1) and SD_DAT [0] (2) Multiple Block Write

4.3.2 Block Read

Blocks of data are read by sending data from the MMC/SD card to the host. Reading the data in the MMC/SD card occurs during the data transfer mode. Commands for reading data are as follows:

- CMD17 READ_SINGLE_BLOCK
- CMD18 READ_MULTIPLE_BLOCK
- CMD30 SEND_WRITE_PROT
- CMD56 GEN_CMD
- ACMD51 SEND_SCR

After a read command is implemented the SD_CLK starts automatically and the FIFO fills with data from the MMC/SD card. After the FIFO is full the SD_CLK stops. During this point the data from the FIFO automatically is loaded into memory (RAM). When all the data has been loaded into memory the SD_CLK starts again to get the next block of data from the MMC/SD card. Figure 3 is a screenshot of both the SD_CLK and SD_DAT [0] pins of the i.MX1 using DMA transfer.

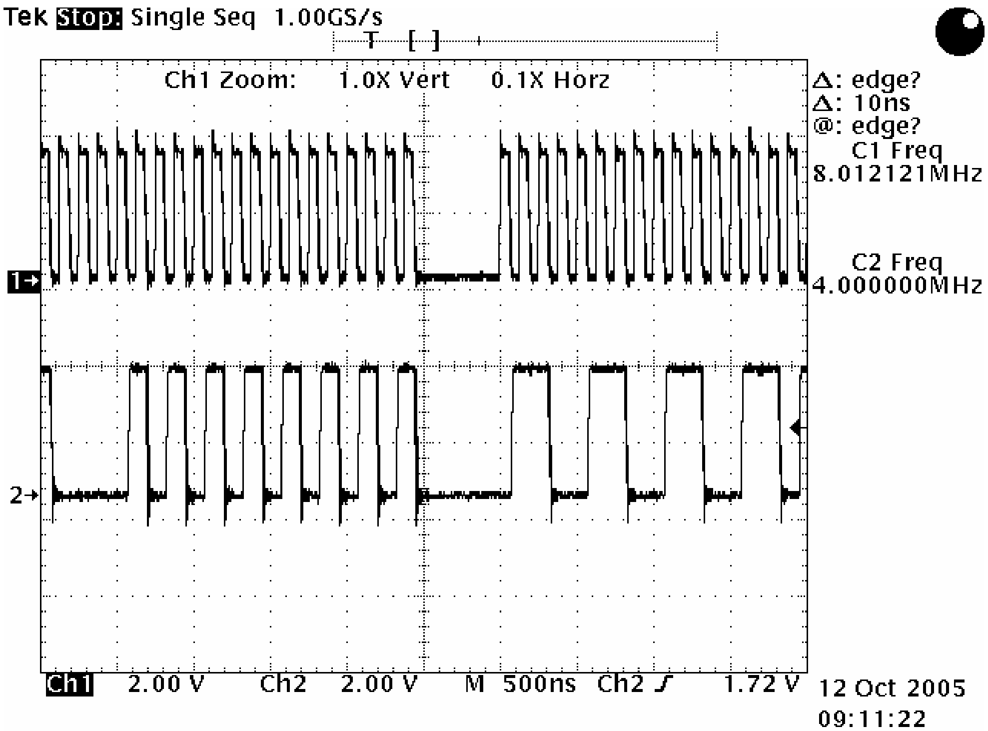


Figure 3. SD_CLK (1) and SD_DAT [0] (2) Multiple Block Read

Both screenshots were captured using a low driving strength. If the driving strength (DS_SLOW bits) is set to the maximum drive strength, it causes overshoot on the SD_CLK signal.

4.3.3 Stream Write and Read

The Stream Mode is not supported in any of the three processors.

5 Special Considerations

The MMC/SD module varies in speed during the data transfer mode. Table 3 and Table 4 describe the various speeds of the MMC/SD module on the i.MX1 and i.MXL processors. The i.MX21 is referenced in [6].

Table 3. i.MX1 Frequency Range

Card Type	Method	Size (Block Size)	Minimum	Maximum
MMC	Polling	Single	4 MHz	20 MHz
MMC	Polling	Multiple	4 MHz	20 MHz
MMC	DMA	Single	4 MHz	20 MHz
MMC	DMA	Multiple	4 MHz	20 MHz
SD (1-Bit Mode)	Polling	Single	4 MHz	25 MHz
SD (4-Bit Mode)	Polling	Single	8 MHz	25 MHz
SD	Polling	Multiple	–	–

Table 3. i.MX1 Frequency Range (continued)

Card Type	Method	Size (Block Size)	Minimum	Maximum
SD	DMA	Single	4 MHz	25 MHz
SD	DMA	Multiple	4 MHz	25 MHz

Table 4. i.MXL Frequency Range

Card Type	Method	Size (Block Size)	Minimum	Maximum
MMC	Polling	Single	4 MHz	20 MHz
MMC	Polling	Multiple	4 MHz	20 MHz
MMC	DMA	Single	4 MHz	20 MHz
MMC	DMA	Multiple	4 MHz	20 MHz
SD (1 Bit Mode)	Polling	Single	4 MHz	20 MHz
SD (4 Bit Mode)	Polling	Single	8 MHz	20 MHz
SD	Polling	Multiple	–	–
SD	DMA	Single	4 MHz	20 MHz
SD	DMA	Multiple	4 MHz	20 MHz

The setting for the data transfer for both polling and DMA is described under the MMC/SD host controller chapter which is referenced on [3] for the i.MX1 and [4] for the i.MXL. The frequency ranges have been tested without any other loads—that is, LCD module or any other modules within the processor, in use and without using any interrupts. Two things to be aware of during data transfer mode: polling mode and DMA transfer. Polling mode does not work when using multiple blocks for SD cards. After writing data, the program holds when checking the FIFO full status. If using SD cards for multiple blocks transfer, then DMA is the best solution. When using DMA to transfer data the priority is important. If the DMA is not set to a high priority then data can get lost and cause a hold and wait for data transfer to complete.

6 References

The following reference documents are used in conjunction with this document for configuration of the MMC/SD module.

6.1 Freescale Semiconductor Documents

The following i.MX technical reference manuals may be found at the Freescale Semiconductor Inc. World Wide Web site at http://www.freescale.com/imx. These documents may be downloaded directly from the World Wide Web site, or printed versions may be ordered.

[1]: *MC9328MX1 Data Sheet* (order number MC9328MX1)

[2]: *MC9328MXL Data Sheet* (order number MC9328MXL)

[3]: *MC9328MX1 Reference Manual* (order number MC9328MX1RM)

[4]: *MC9328MXL Reference Manual* (order number MC9328MXLRM)

[5]: *M9328MX1_L_ADS_V2_0 ADS Schematic and Orcad File*

[6]: *AN2906 MMC/SD Host Controller Addendum Application Note* (order number AN2906)

6.2 Standard Documents

The following standard documentations were used as reference for this application note and can be purchased at their respective Web sites.

MultiMediaCard System Specification version 3.1 at http://www.mmca.org/home.

SD Memory Card Specification version 1.0 and SD I/O Specification version 1.0 at http://www.sdcard.org/.

NOTES

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

***For Literature Requests Only*:**
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-521-6274 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: AN3049
Rev. 0
10/2005