Using SDIO with SDHC
MC9328MX1, MC9328MXL, and MC9328MXS

1 Abstract

The MultimediaCard™ (MMC) is a 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. MMC communication is based on an advanced 7-pin serial bus designed to operate in a low voltage range at medium speed. Secure Digital Card (SD) is an evolution of the MMC with an additional two pins in the form factor that is specifically designed to meet the security, capacity, performance, and environmental requirements inherent in new audio and video consumer electronic devices. The physical form factor, pin assignment, and data transfer protocol are compatible with the MMC. The SD is composed of a memory card and an I/O card. The I/O card combines high-speed data input/output with low-power consumption for mobile electronic devices.

The MMC/SD physical specifications support two operation modes, SD transfer mode and SPI transfer mode. The Multimedia Card/Secure Digital Host Controller (SDHC) designed in the i.MX processors

Contents
1 Abstract ..................................................1
2 Modes of Operation .................................2
3 Hardware Consideration
   on using SPI Transfer Mode ......................4
4 References ...........................................8
5 Revision History .....................................8

This document applies to the following i.MX devices, collectively called i.MX throughout:
- MC9328MX1
- MC9328MXL
- MC9328MXS
integrate MMC support with SD memory and, I/O functions. The I/O function supported follows the SPI transfer mode. However, the SPI mode relies on the i.MX processor’s embedded SPI module. The disadvantage of using the SPI is the loss of performance of the SPI transfer mode versus SD transfer mode (for example a single data line and hardware Chip Select signal per card). This application note attempts to resolve the I/O and socket connections issue.

2 Modes of Operation

The two transfer modes of operation are discussed in this section.

2.1 SD Transfer Mode Operation

The Secure Memory Card bus has a single master (application), multiple slaves (cards), synchronous star topology (see Figure 1). Clock, power, and ground signals are common to all cards. Command (CMD) and data (DAT0–DAT3) signals are dedicated to each card to provide continued point-to-point connection to all cards. During the initialization process, commands are sent to each card individually, allowing the application to detect the cards and to assign logical addresses to the physical slots. Data is always sent or received—to or from each card individually. However, to simplify the handling of the card stack after the initialization process, all commands may be sent concurrently to all cards. Addressing information is provided in the command packet.

SD bus allows dynamic configuration of the number of data lines. After power-up, by default, the SD Memory Card will use only 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 hardware cost and the system.

The SD bus includes the following signals:

- **CLK**: Host to card clock signal
- **CMD**: Bidirectional Command/Response signal
- **DAT0–DAT3**: Four bi-directional data signals
- **VDD, VSS1, VSS2**: Power and ground signals
2.2 SPI Transfer Mode Operation

The SPI compatible communication mode of the SD Memory Card is designed to communicate with a SPI channel, commonly found in various microcontrollers in the market. The interface is selected during the first reset command after power-up and cannot be changed as long as the part is powered on. The SPI standard defines the physical link only, and not the complete data transfer protocol. The SD Memory Card SPI implementation uses the same command set of the SD mode. From the application point of view, the advantage of the SPI mode is the capability of using an off-the-shelf host, therefore 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 SD Memory Card SPI interface is compatible with SPI hosts available on the market. As any other SPI device the SD Memory Card SPI channel consists of the following four signals:

- CS—Host to card Chip Select signal
- CLK—Host to card clock signal
- DataIn—Host to card data signal
- DataOut—Card to host data signal

Another common SPI characteristic is byte transfers that are implemented in the card as well. All data tokens are multiples of bytes (8-bit) and always byte aligned to the Chip Select signal.
3 Hardware Consideration on using SPI Transfer Mode

Currently, the SD Host Controller in the i.MX processors does not include their own SPI module, as a result, the user is not allowed to handle SDIO-SPI transfers. Therefore, it is required to use the embedded SPI module (master) within the i.MX processor for this data communication. To allow users to SDIO-SPI transfer mode, there are several recommendations in terms of system connection and software programmability.

3.1 Hardware Connection

Table 1 summarizes the pin differences between SD mode and SPI mode. Figure 3 illustrates a recommended system connection to operate both SD and SPI transfer modes in a single SD card socket.

<table>
<thead>
<tr>
<th>Pin</th>
<th>SD 4-Bit Mode</th>
<th>SD 1-Bit Mode</th>
<th>SPI Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CD/DAT[3]</td>
<td>Data line 3</td>
<td>N/C</td>
</tr>
<tr>
<td>2</td>
<td>CMD</td>
<td>Command line</td>
<td>CMD</td>
</tr>
<tr>
<td>3</td>
<td>VSS1</td>
<td>Ground</td>
<td>VSS1</td>
</tr>
<tr>
<td>4</td>
<td>VDD</td>
<td>Supply voltage</td>
<td>VDD</td>
</tr>
<tr>
<td>5</td>
<td>CLK</td>
<td>Clock</td>
<td>CLK</td>
</tr>
<tr>
<td>6</td>
<td>VSS2</td>
<td>Ground</td>
<td>VSS2</td>
</tr>
<tr>
<td>7</td>
<td>DAT[0]</td>
<td>Data line 0</td>
<td>DATA</td>
</tr>
</tbody>
</table>
In the system circuitry shown in Figure 3, three tristate buffers are required (for example, 74LVX125) to isolate between an SPI connected device and an SDIO device. Two additional controls for a tristate buffer enable an SPI mode chip-select for an SDIO device. In an i.MX processor, there are two separate SPI modules, each with the capability to be programmed as a master. Users may select either SPI module to connect for SDIO in SPI mode.

Table 2 shows the signal selection for SPI1 or SPI2 and the MMC/SD port. Table 3 shows the I/O settings and configurations. Note: all signals are stated as GPIO and in the input state after power-up reset.

**Table 2. I/O Signal Configuration for i.MX SPI1 and SPI2 Modules**

<table>
<thead>
<tr>
<th>SPI Signal Name</th>
<th>Connect to GPIO Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPI1_SPI_RDY</td>
<td>Primary function of GPIO port C [13]</td>
</tr>
<tr>
<td>SPI1_SCLK</td>
<td>Primary function of GPIO port C [14]</td>
</tr>
<tr>
<td>SPI1_SS</td>
<td>Primary function of GPIO port C [15]</td>
</tr>
<tr>
<td>SPI1_MISO</td>
<td>Primary function of GPIO port C [16]</td>
</tr>
<tr>
<td>SPI1_MOSI</td>
<td>Primary function of GPIO port C [17]</td>
</tr>
<tr>
<td>SPI2_SCLK</td>
<td>AIN of GPIO port A [0] or AIN of port D [7]</td>
</tr>
<tr>
<td>SPI2_SS</td>
<td>AIN of GPIO port A [17] or AIN of port D [8]</td>
</tr>
<tr>
<td>SPI2_TXD</td>
<td>BIN of GPIO port D [31] OR AIN of port D [10]</td>
</tr>
</tbody>
</table>
Two scenarios are possible using SDIO in SPI mode:

- One SPI port (or the corresponding GPIO pins) is not used, and all four SPI signals (CLK, MISO, MOSI and SS) are tied directly to the SDHC’s CLK, CMD, DAT0, and DAT3 signals. In this scenario, the three tristate buffers are not required. However, a software driver is required to correctly handle the transition between SDIO-SPI mode and all other MMC/SD modes.
- The selected SPI port is shared between the SPI connected device and the SDIO. The circuitry recommended in Figure 3 is designed for this scenario. The tristate buffer is inserted to isolate between an external SPI device and the SDIO signals. A separated control is required for the tristate buffer from either the GPIO or switch.

### 3.2 Software Consideration and Programming

The i.MX processor’s SD Host Controller supports MMC/SD transfer mode. In this mode the current specification and requirements only allow communication in block transfers—that is, 512 bytes, therefore *non-block* transfers are not supported. Some SDIO devices are designed to operate in byte transfer mode. These designs require the use of the SPI modules to process this type of communication. The SPI module is byte oriented, and therefore supports any packet in the unit of byte counts. Figure 4 and Figure 5 illustrate the suggested software approach that works with the recommended system circuitry to distinguish byte transfer in SDIO. Note that these approaches assume the use of SPI compatible SDIO devices. These approaches are not intended for SDIO devices that are not SPI compliant.
Figure 4. Flow Chart for SDIO for SPI Only SDIO Card

Figure 5. Flow Chart for SDIO for SPI + 1- and 4-bit SDIO Card

Using SDIO with SDHC Application Note, Rev. 1
4 References

The following documents can be used for additional information:

2. *SD Memory Card Specifications—Part 1 Physical Layer Specification*, Ver 1.0 (MMCA: unreleased at time of this publication)
4. *MC9328MX1 i.MX Integrated Portable System Processor Reference Manual* (order number: MC9328MX1RM)
5. *MC9328MXL i.MX Integrated Portable System Processor Data Sheet* (order number: MC9328MXLRM)
6. *MC9328MXS i.MX Integrated Portable System Processor Data Sheet* (order number: MC9328MXSRM)

5 Revision History

This revision is for the purpose of applying the Freescale template and does not include technical content changes.

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. ARM and the ARM Powered Logo are registered trademarks of ARM Limited. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2005. All rights reserved.