Eureka Technology

Understanding SD, SDIO and MMC Interface

by

Eureka Technology Inc.

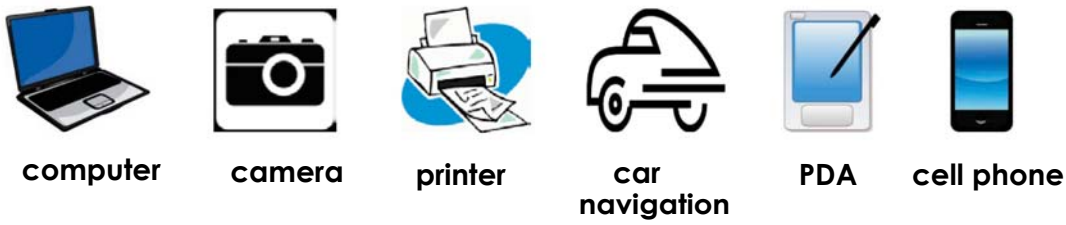
May 26th, 2011

Introduction

This white paper presents very important information for managers, engineers, and system architects who want to broaden his/her knowledge of interfacing with removable data storage devices. There are many different aspects of SD and MMC interfacees and this white paper organize them into a very easy to understand format. One must understand the different characteristics of these interface in order to harness the power of the technology and deploy them wisely into new designs and applications.

What is SD memory interface?

Secure Digital (SD) memory card first appeared at the 2000 CES trade show. Within 10 years, it has become the most popular non-volatile removable data storage media for consumer applications. SD memory belongs to the same class of removable data storage device that includes PC card, PCMCIA card, CompactFlash, SmartMedia, MultiMedia Card, Memory Stick and xD. This class of removable data storage device is not intended to replace hard drive. They deliver a fraction of the capacity of hard drive at a fraction of the speed performance in a much smaller form factor and also at a smaller unit price. SD device can typically be founded in digtial camera, video recorders, cell phone, printer, laptop computer, car navigation and other applications.



What contributes to SD Card successes?

While PCMCIA was mainly developed for notebook computers, SD card and its other competitive devices were targeted to consumer electronics such as cameras and cell phones. SD card has relatively small number of physical pins (5 signals plus clock and power) compared to the earlier PCMCIA and CompactFlash cards. Data is transferred at the moderate speed of 25 MHz (earlier version) up to 208 MHz (latest version), making it fairly inexpensive to manufacture by using matured technologies. At the same time, the synchronous data transfer protocol is very efficient, enabling SD card to delivery adequate data bandwidth requirements for most digital and video applications. The combination of small foot print, lower cost, performance and the backing by several major manufacturers as an open standard enables SD card to become the de-facto standard for consumer electronics data storage.

What is MultiMedia Card (MMC)?

MMC is closely related to SD card. The MMC standard precedes the SD standard by several years. The two standards share many common features and have the same physical and electrical specifications. The differences between the two standards are mainly on the software level commands. The similarities allow many hosts to accept both MMC and SD cards on the same socket.

What is the basic SD protocol?

SD bus transfers information serially through the bi-directional CMD and DATA pins. Each command packet consists of 48 bits of information on the CMD pin. The command packet includes the command index, argument and CRC check bits. Command is always sent by the host and the response is sent by the card, also through the CMD pin. Most response packets are also 48-bits long. The host is always the command initiator and the card is always the command target.

Not all commands have associated data transfer. Only read and write commands are accompanied by data transfer. The unit of data packet is called a block. Different block size can be defined by the host. Most data block size is 512 bytes. Data can be transferred using either 1 or all 4 data pins. Each data packet is followed by 16 bits of CRC data.

SD command processing is heavily state dependent. The card is always in a particular operating state at a given time. Only certain commands are valid in a given state. The host, through the initialization command sequence, changes the card's operating state until it is ready to transfer data.

Command and data transfer are synchronized with a clock signal generated by the host. Initially after reset, the host uses the lowest clock rate of 400 KHz to communicate with the card. Internal registers in the card provide information related to the card's capability to the host. By knowing the maximum operating frequency, data bus width and other capability information, the host increases the clock rate for optimal performance.

What are the different versions of SD and MMC specifications?

SD specification is published by SD Association. The founding members of the SD Association are Panasonic, SanDisk and Toshiba. Currently there are about 1000 member companies in the SD Association.

The first version, 1.0, was released in March 2000. This basic version supports 25 MHz bus frequency and maximum card size of 2 Gbytes. Data can be transferred in the SD mode (choice of 1-bit and 4-bit data) and the backward compatible SPI mode (1-bit only).

The second major revision of the SD interface was released in May 2006. It adds the High Capacity SD (SDHC) specification which increases the maximum card capacity to 32Gbytes. The amount of changes in this revision are not extensive. However, SDHC lays the ground work for future revision changes. The added new card sizes (4, 8 and 16 GBytes) occupy the sweet spot for most picture and video applications. Even after the introduction of the newer capacity in version 3.0, SDHC is still the dominate card size in the market place.

Version 3.0 of the SD specification adds a new capacity standard SDXC which supports up to 2 TBytes per card and it also increases data bandwidth through the introduction of the Ultra High Speed (UHS-I) specification. Using the same 4-pin interface, UHS-I supports a maximum single data rate speed of 208 MHz which delivers 104MByte/sec data rate and a maximum double data rate speed of 50 MHz for 50 Mbyte/sec data rate.

Version 4.0 is being reviewed by the SD Association and is expected to be released in 2011. Version 4.0 adds UHS-II speed version and increases the data rate even higher than version 3.0.

MMC specification was first developed in 1996 by Siemens and SanDisk. The specification was controlled by the MMC Association until 2008. Since 2008, the specification and new releases are controlled by JEDEC.

Version 2.0 was released in 1999 followed by version 3.0 in 2001. SD 1.0 specification is officially compatible with MMC version 2.11.

Four and eight bit data width were introduced in version 4.0 in 2003. 8-bit data width gives MMC twice the data bandwidth as SD running at the same speed. However, 8-bit data width is not supported by many host and 4-bit data width is still the most widely used data width for communicating with removable cards.

After JEDEC took over the MMC specification, embedded MMC (eMMC) was introduced as version 4.3. eMMC is intended for on-board application where the eMMC device is mounted on printed circuit board (PCB). Other than the mechanical form factor, eMMC and MMC shares the same features and functions.

The latest eMMC specification is version 4.41. Since version 4.4, boot mode is supported to allow the device to be used as the boot ROM. DDR data rate is also introduced. The highest bandwidth is 8-bit DDR running at 50 MHz which delivers 100 MByte/sec, giving it the same data rate as SD card running at 4 times the clock rate. The following table summarizes the data rate of different versions of both buses:

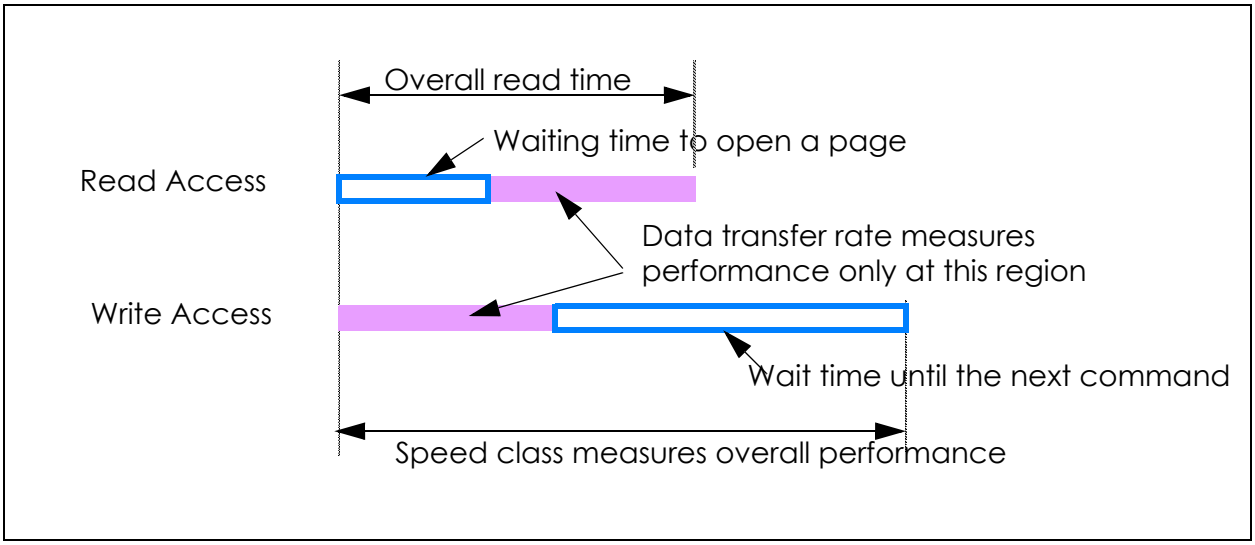
SD and MMC versions

Revision	Max Clock Frequency	Data Rate
SD 1.0 to 1.01	25 MHz	12.5 MByte/sec
SD 1.10 to 2.0	50 MHz	25 MByte/sec
SD 3.0	208 MHz	104 MByte/sec
MMC 1.0 to 3	20 MHz	2.5 MByte/sec
MMC 4.0 to 4.3	50 MHz	50 Mbyte/sec
MMC 4.4	50 Mhz	100 Mbyte/sec

Performance: Bandwidth vs. Speed Class.

The maximum data rate (104 MByte/sec for SD and 100 MByte/sec for eMMC) is not always a true measure of performance of the device. The data rate only specifies the performance of the card when it is transferring data. It does not give any indication how long the host need to wait for the initial data transfer after a command is issued. The longer is the waiting time, the lower is the usable bandwidth. For sustained data transfer, it is just as important to reduce the response time as to increase the clock rate.

Since data storage in the card is implemented by NAND Flash memory cells, the ultimate performance is dictated by the memory technology as well as the memory architecture. To read data from NAND Flash, the page needs to be opened first and there are a few tens of microseconds of wait time to open a page. Until the page is opened, no data can be read. For writing to the NAND Flash, after a page is transferred to the device, it takes a few hundred microseconds to a few milliseconds to program the data into the NAND Flash. New operation may not be allowed until the programming is completed.



There are many factors affecting performance of NAND Flash and SD cards. Single level cell (SLC) NAND Flash has less page open time and program time compared to multi level cell (MLC) NAND Flash but MLC has larger capacity and lower per bit price. Multiple channels architecture improves performance by overlapping the page open and program time of one channel with the data transfer time of another channel, thus reducing the overall down time. Other factors such as controller efficiency and error correction algorithm also affect performance.

In order to provide the user with a true measure of usable performance of the SD card, the SD specifications since version 2.0 define several speed classes that each card must belong to.

SD speed class table

Speed Class	Description	Revision
Class 0	These class cards to not meet the performance of other classes or does not specify its performance, including legacy cards before version 2.0	Version 1.0
Class 2	Card with 2 MBytes/second or more sustained bandwidth.	Version 2.0
Class 4	Card with 4 MBytes/second or more sustained bandwidth.	Version 2.0
Class 6	Card with 6 MBytes/second or more sustained bandwidth.	Version 2.0
Class 10	Card with 10 MBytes/second or more sustained bandwidth.	Version 3.0

One can see that there are big differences between the data rate defined by the clock rate and the sustained bandwidth defined by speed class. The differences are reflection of the performance of the NAND Flash storage technology, not a limitation of the SD bus protocol. By the same token, one can improve the performance of a SD card design by improving the memory design without using the highest clock rate of the SD bus.

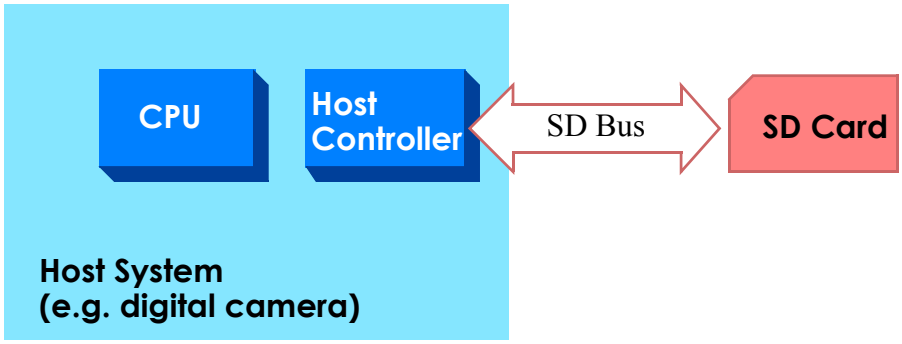
What is SDIO and SD combo?

SD bus specification defines three card types: memory only card, IO card (SDIO) and card combining memory and IO functions (SD combo). SDIO is an extension of the SD specification designed for IO only devices. New commands are defined in SDIO specification while some memory only features are removed. For example, SD memory card has a programming state which accounts for the programming state (in addition to read and write) of the NAND Flash device. SDIO device does not have programming state.

Different IO functions such as SDIO Wi-Fi or bluetooth controller can be implemented in SDIO card format. The SDIO specification supports multi-function architecture. Each function can operate independently and has its own memory space. The minimum number of function is 1 and the maximum is 8. Typical SDIO has two functions. Function 0 is the system function and function 1 is the user function.

SD combo, as the name implies, contains both SD memory and SDIO functions. It is not as commonly used as SD memory or SDIO only device.

What is SD bus architecture?



The above diagram shows a typical SD bus system with the host controller and the SD card. Even though SD specification allows multiple SD cards to be connected to the same SD bus, in practical applications, most system designs are single card only. It is fair to assume that most SD bus has a single host (initiator) and a single card (slave). Signals on the SD bus include a clock pin which is generated by the host, one bi-directional CMD (command) pin and 1 or 4 bi-directional DT (data) pins.

All SD bus transactions are initiated by the host through the CMD pin. All CMD and DT signaling are synchronized to the clock signal. Each command is a 48-bit packet and is shifted out serially by the host on the CMD pin. The following is the format of the command packet:

start bit (1 bit)	direction bit (1 bit)	command index (6 bits)	Argument (32 bits)	CRC (7 bits)	end bit (1 bit)

Some commands require responses from the SD card and some do not. Some commands require data transfer on the DT pins and some do not. If the command requires a response, the SD card shifted out the response packet serially on the CMD pin within 64 cycles from receiving the command. The response packet format is predefined for each command. Most responses are 48 bits while some are 136 bits.

If the command has associated data transfer, it is transferred serially on the DT pins. Write data is transferred from the host to the card while read data is transferred from the card to the host. Each data packet consists of a start bit, the data bits, 16 bits of CRC and an end bit. The size of the data bit is either implicit to the command or is defined by the host with an earlier command.

The SD bus specification specifies, among other things, the physical signal interface on the SD bus, the definition of each command and responses, a set of standard registers within the SD card, the internal state and status of the card, the command sequence for initializing and enabling data transfer on the bus.

The SD bus specification does not define the type of memory or IO devices within the SD or SDIO card. Even though SD memory is based on NAND Flash technology, there is nothing in the specification that prevents other types of non-volatile memory to be used, as long as the command protocols are followed. Similarly, the SDIO specification does not restrict the type of IO devices to be implemented in the SDIO card.

What is in the SD host controller?

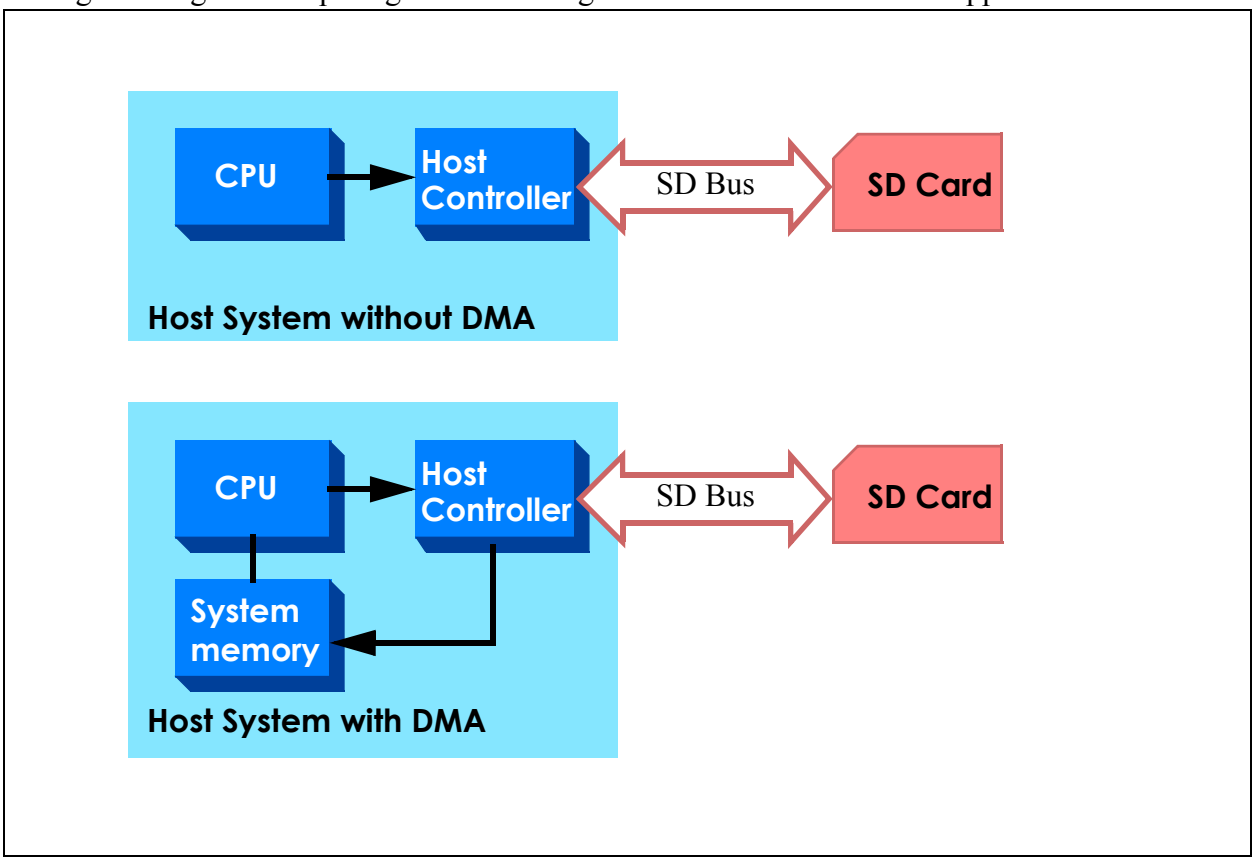
The SD host controller is the hardware logic that forms the bridge between the host CPU and the SD bus. The host specification section of the SD bus specification provides a very strict definition of the host controller design. This specification ensures a very uniform software view of the host controller as seen by the CPU and allows the same driver software to be used for different host controller hardware.

From the CPU's perspective, the host controller consists of a 256-byte register set which is mapped to the system's memory or IO space. Transaction on the SD bus is initiated by the software reading or writing to this register set. Through this register set, the host driver software can:

- Detect the insertion or removal of the SD card,
- Turns on and off power to the SD card,
- Enable, disable and control SD clock speed,
- Define command arguments and send commands to SD card,
- Receive command responses from the SD card,
- Read data from and write data to the SD card,
- Suspend, resume and terminate data transaction.
- Interrupt the CPU on different types of event such as command completion, CRC error, etc.

The host controller can also include DMA support. Without DMA, the host CPU must read or write every word of data to be transferred with the SD card. This takes up a significant portion of the CPU's processing power. With the DMA engine included in the host controller, the CPU only needs to set up the data transfer. Once data transfer starts, the DMA engine will read/write each word of the data between the SD bus and system memory, thus freeing the CPU for other tasks on the system. Interrupt can be sent to the CPU on DMA completion. DMA is a very efficient way of

transferring data but it requires the host controller to have access to the system memory. The following is a diagram comparing two host designs with and without DMA support.



For more information about the host controller design, please refer to the SD host controller core from Eureka: http://www.eurekatech.com/products/peripherals

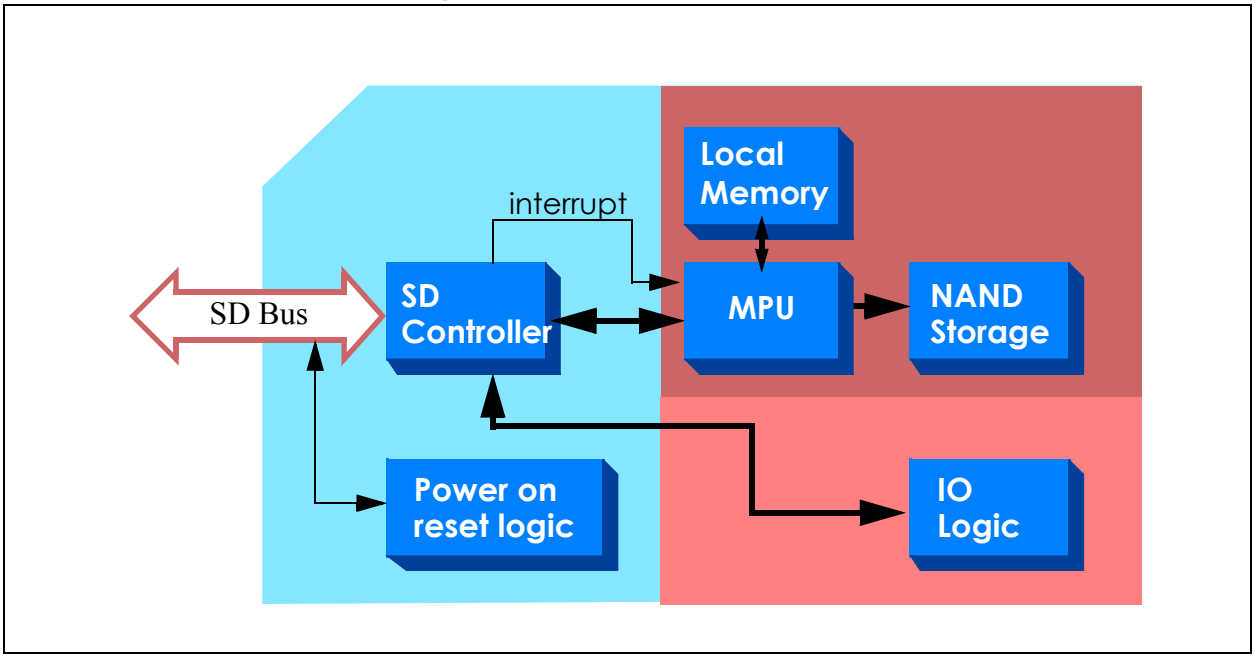
How are electrical and timing control being handled?

The host system is responsible for supplying power and timing signals to the SD card. Several card detection mechanisms are defined in the SD specification that allow hot insertion and removal of SD cards. Interrupt can be generated at card insertion. Upon card insertion interrupt, the CPU can enable power to the SD Card through the power control register in the host controller. In typical design, the power control information is forwarded from the controller logic to the power control unit of the system. Default power supply of SD card is 3.3V

The host controller is also responsible for generating clock signal to the SD card. After power is enabled, system software may enable the clock signal to the SD card at the default frequency of 400 KHz. Initialization sequence is run at this frequency until the software detects the frequency and bus width capability of the card through the card's control registers.

The host CPU can stop the clock signal and/or remove power to the card when there is no operation in order to save power. SD card cannot use phase lock loop (PLL) on its clock signal.

What is in the SD card design?



The above diagram shows a typical SD card design. It includes both SD memory and SDIO functions. For SD memory implementation, it includes the NAND Flash based storage, which is typically managed by an MPU with local memory for program execution. Typical SDIO implementation includes the SD interface and the IO function logic block. There may be MPU inside the IO logic block as well.

The SD Controller in the above diagram handles all physical and data link level functions such as command decoding, response generation, CRC, status management and pre-defined SD register set. Many SD commands can be processed directly by the SD controller module. Access to the memory space would be forwarded from the SD controller to other modules in the card.

In most NAND Flash design, access to the NAND Flash needs to go through a local processor which handles wear leveling and bad block management. The SD controller does not directly access the NAND Flash chips. It either accesses a shadow memory in the local system for data transfer or interrupts the local processor which would then process the neccessary data transfer. When interrupt is received by the MPU, it would query the SD controller module to find out the data request from the SD bus. If it is a read request, it would retrieve data from the NAND Flash and deposit read data to the SD controller module. If it is a write, it would retrieve data from the SD controller module and program it into the NAND Flash.

For SDIO device, depending on the nature of the IO logic being implemented, the SD controller core can either directly access the IO logic when there is user data transfer request from the SD bus or use the same interrupt method as described in SD memory to request service from the IO logic.

The SD specification does not dictate the internal architecture of the SD card. The designer has the freedom to choose the architecture most suitable for the application. However, it is fair to expect that the interrupt method is more commonly used in SD memory card design and the direct access method is more common for SDIO card design.

Even though SD specification is targeted to NAND Flash storage technology, many NAND Flash specific characteristics are not defined in the SD specification. For example, error correction code (ECC) which is essential for NAND Flash memory, is not defined in the SD specification. Wear leveling and bad block management are not defined either. Designer of the SD memory card should follow the requirements from the NAND Flash manufacturer in implementing such design.

What is SPI and how is it related to SD and MMC?

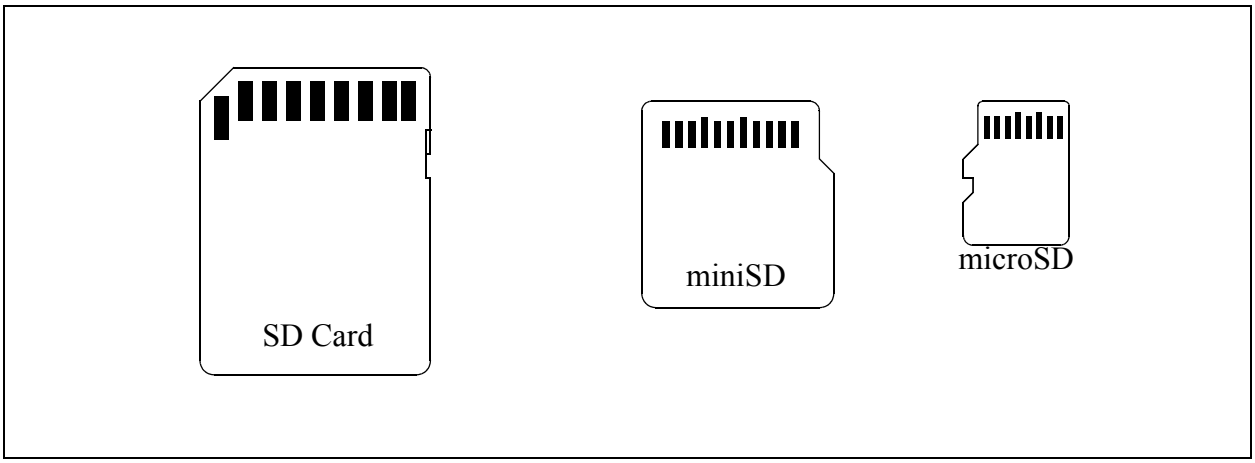
Serial Peripheral Interface (SPI) precedes SD and MMC by many years. It was originally found in microprocessors developed by Motorola and others. The hardware interface is very simple and is somewhat similar to SD bus except that signals are uni-directional and supports 1 data bit only. Both SD and MMC bus initially support SPI mode so that microprocessor equipped with SPI port can communicate with SD cards through the addition of a software driver.

SPI mode is not required for SD host because any host designed for SD bus should implement the SD bus protocol instead of lower performance SPI protocol. Starting with MMC version 4.3, SPI support is no longer required. SPI mode support is still required for SD card.

What is SD vs. miniSD, microSD, embedded SD?

SD specification allows 3 physical sizes for SD cards, regular, miniSD and microSD. All three sizes have the same logical and functional definitions. The difference is only in the physical dimension of the card. Regular size SD card is 24mm x 32mm x 1.4mm. This is the card size typically found in digital camera applications. MiniSD card is 20mm x 21.5mm x 1.4mm. MircoSD is the smallest of the three at 11mm x 15mm x 0.7mm. MicroSD is typically found in cell phone usage. Because microSD card is smaller than the other two formats in all three dimensions, many

manufacturers provide microSD to miniSD and microSD to SD adapters so that a microSD card can be used in all three formats.



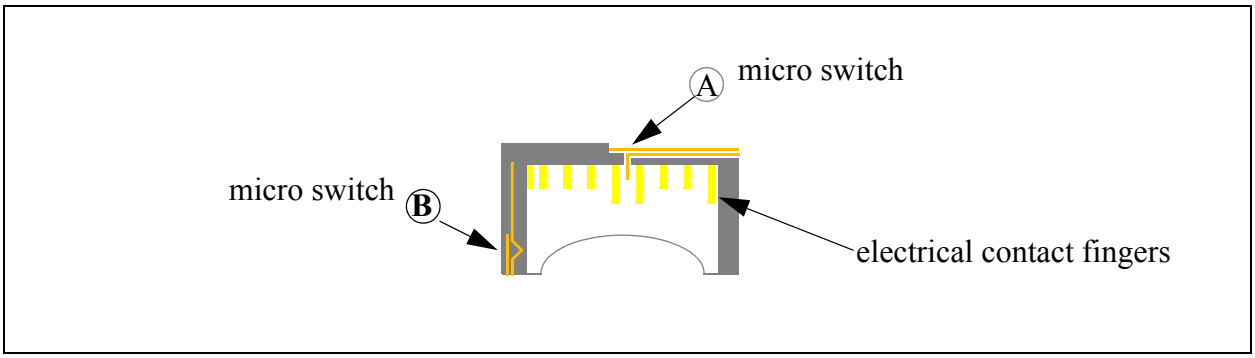
Embedded SD (eSD) is an integrated circuit in thin fine-pitched ball grid array (TFBGA) package. It is designed to be mounted on printed circuit board (PCB) and is not removable. eSD is an extension of the SD specification and there are functional and electrical differences between SD and eSD devices specific to the embedded environment.

The embedded version of MMC is eMMC. Since version 4.3, eMMC and MMC share the same specification. There are several features in the MMC specification such as boot mode which enhance the device function for embedded usage.

How to handle card insertion, removal and write protection?

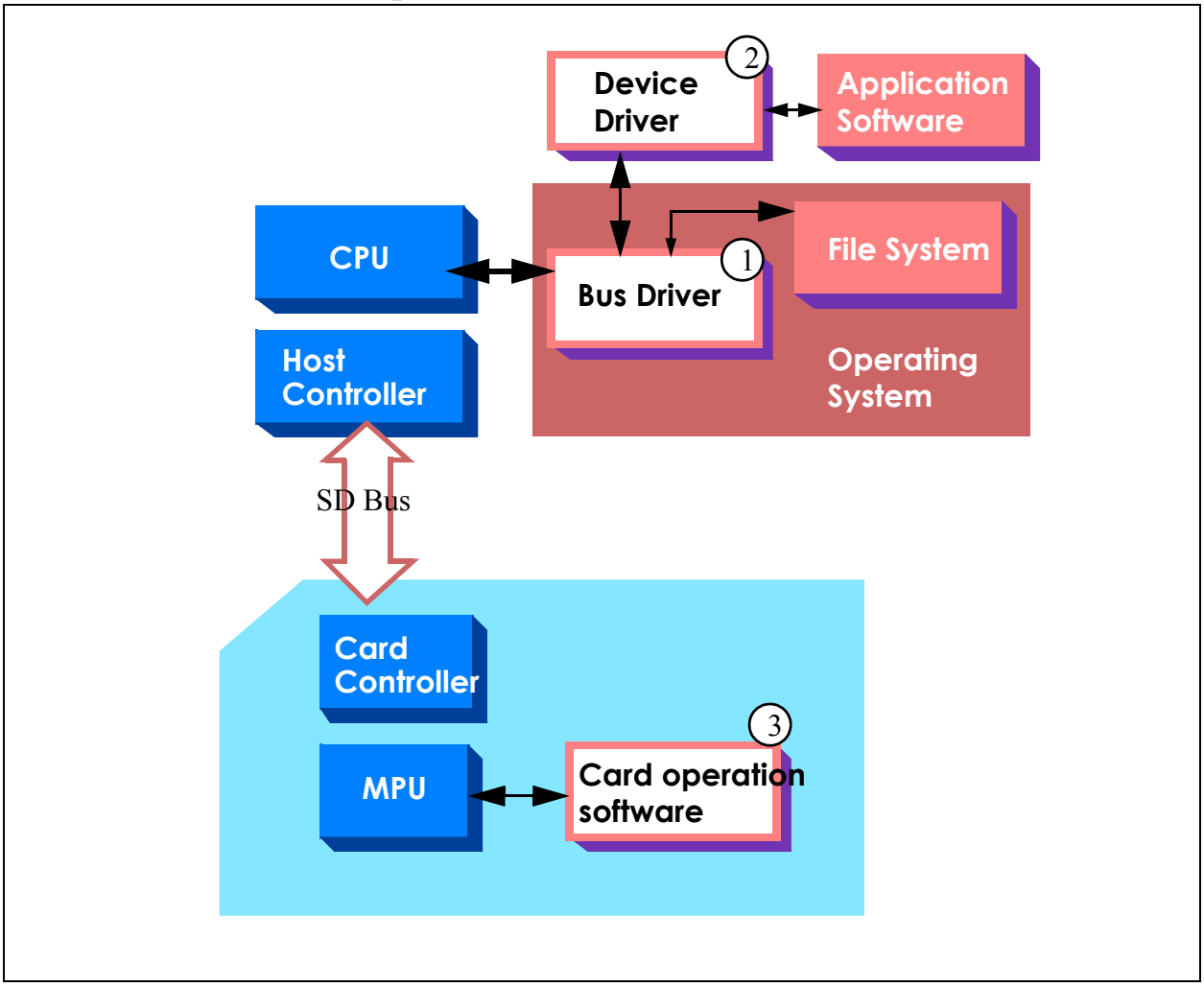
SD specification supports hot insertion/removal and write protection. There are several ways for the system to detect card insertion/removal. The most common way is by using the mechanical switches at the SD socket. Most SD socket contains two micro switches in addition to the electrical contact for the signal pins. In the following diagram, micro switch A is opened when the card is removed and closed when the card is inserted into the socket. Micro switch B lines up with the write protect dip switch on the card. If the dip switch on the card is in the read/write position, Dip switch B is closed when the card is inserted. If the dip switch on the card is in the write protect position, micro switch B is opened when the card is inserted. There are nine electrical contact fingers in the socket to match with the card. The two longer fingers in the middle are for power and

ground. This is to ensure that power and ground signals are the first to connect with the card and the last to disconnect.



The write protect dip switch on the SD card is a mechanical one only. It can be slid between the normal and write protected positions. It has no electrical connection in the card. Write protection is only detected by the host through the micro switch described above and it is the responsibility of the host software to prevent writing to a write protected card.

What are the software requirements?



Depending on the context of the discussion, SD software may mean three different things. As shown in the diagram above, it can refer to (1) the operating system module that handles the initialization and the low level signaling to the SD device, (2) the device driver that runs on top of the OS to perform application specific function of the device, or (3) the software that runs the local microprocessor inside the SD card.

In order to enable SD bus in the system, the OS must be SD aware so module 1 must exist in the system. Most operating system has such support. Operating systems such as Windows and Linux not only have the basic module 1 function but also link it to the file system. When an SD memory card is inserted, the OS automatically assigns a drive letter to the card and it can be used for any file transfer as a disk drive. In this type of application, the system developer does not need to develop any software for the host.

Software module 2 is required if the SD card is not configured as a standard SD memory device or the operating system does connect the SD device to the file system. For example, if it is an SDIO card with user specific function, the OS will only be able to detect the card existence and run

through the initialization sequence. Specific application software and device drivers will be needed to communicate with the card to perform the desire functions.

Software module 3 typically exists in SD memory card. It is a concern of the card developer and not related to the host developer. It is the software run by the by the local processor to provide the card functions. If the card is an SD memory card, this software includes the interrupt service routine to service interrupt requests from the card controller, the wear leveling and bad block management software for the NAND Flash device. If it is an SDIO device, the software may be whatever that is needed to provide the particular IO function. The architecture and software within the card is not defined by the SD specification so there is no standard to follow other than the requirements imposed by the internal modules. Typically this software module is custom built for each design.

Where is the file system?

Strictly speaking, SD memory card device does not have file system. The SD memory card presents itself as a flat memory space to the host. However, SD specification defines the file system for the host system. It is the responsibility of the host system that builds a file system on top of this flat memory space. The SD memory card is oblivious to the file system. The file system for SD memory card is based on Microsoft FAT and exFAT. The file system specification allows different host devices to exchange data through the SD memory card.

Since the file system exists in the host system only, it is perfectly acceptable for the host to bypass the file system altogether and access the SD memory space in whatever customized data format it chooses. In this case other host will not be able to access the SD memory card unless it is aware of the customized data format.

Eureka Technology Inc.
4962 El Camino Real
Los Altos, CA 94022, USA

Tel: 1 650 960 3800
Fax: 1 650 960 3805
email: info@eurekatech.com
http://www.eurekatech.com