eMMC v4.41 and v4.5
Architecture for High Speed
Functions and Features

Victor Tsai
Micron Technology, Inc.
Agenda

- e·MMC Market Trend
- e·MMC Versions
- e·MMC v4.41 New Features for High-Performance Mobile Handset Platform
- e·MMC v4.5 Preview
- In Conclusion
eMMC Market Trend
eMMC Share of Total Flash Market (% of total Gbytes)

Source: Micron Marketing
e·MMC Density Trend

Source: Micron Marketing
Mobile Handset Booting Architecture

Source: Micron Marketing
eMMC Versions
• e·MMC v4.41
  - JEDEC document JESD84-A441, published in March 2010
  - Replaces e·MMC v4.4
  - Incorporates all e·MMC v4.4 features, plus new features

• e·MMC v4.4
  - JEDEC document JESD84-A44, published in March 2009
  - Considered to be obsolete
  - Replaced by e·MMC v4.41
e·MMC v4.41 New Features for High-Performance Mobile Handset Platform
New Features for Demand Paging

- eMMC v4.41 specification introduces 2 new features for Demand Paging code execution strategies:
  - Background Operation
  - High Priority Interrupt (HPI)

- The features are complementary in improving Write Throughput performance as well as Paging request latencies on the non-volatile memory solution
Background Operation

- e-MMC may perform various internal background operations necessary for internal maintenance purposes during run-time, independent from the normal operations initiated by the Host.
- In order to reduce latencies during time-critical operations such as Read and Write, and minimize uncontrolled power consumption by e-MMC during Idle time, this feature gives the Host the capability to delay Device background operations until the Host explicitly initiates Device background operation in a controlled manner.
- How it works
  - Device indicates to the Host the criticality of its need for internal background maintenance operation by the value in the BKOPS_STATUS (ECSD[246]) register.
  - When Host can allocate an extended period of inactivity, it initiates Device background operation by writing any value to BKOP_START (ECSD[164]) register.
    - i.e. CMD6 with argument = (0x 03A4 FF00)
  - Host may use high priority interrupt command to stop the background operation.
Background Operation Benefits

- Most time-consuming operations are due to e-MMC internal Flash Data management (Garbage collection, wear leveling, bad block management, ...)
- By enabling the **Background Operation** feature:
  - The OS can check the e-MMC internal state and ask to perform time consuming operations in background -- i.e. when the user applications are not accessing the non-volatile memory
  - The e-MMC usage is optimized to be in the best internal state during Foreground operations
  - The number of Foreground time consuming operations is reduced and average throughput improved
High Priority Interrupt (HPI)

- During the execution of a large multiple-block Write or Erase operation, BUSY time can be long and unpredictable.
- Due to this limitation, it is difficult to utilize e·MMC in system use cases such as Demand Paging, where data must be retrieved from the e·MMC with minimal latency.
- e·MMC 4.41 introduced a mechanism to interrupt a busy condition in a controlled manner within a well-defined timeout, without compromising data integrity.
- How it works
  - During a large multiple-block Write operation, data transfer can be terminated by a Stop command (CMD12), but there may still be an extended period of BUSY time necessary to complete the Write operation on the data already transferred.
  - HPI interrupts this process and returns the Device to Trans state such that the Host can send a Read command to retrieve high-priority data.
  - The CORRECTLY_PRG_SECTORS_NUM field (EXT_CSD bytes [245:242]) indicates the number of 512B sectors successfully written to the e·MMC when the Write operation is interrupted by HPI -- Host can restart the Write operation with the correct address offset based on this information.
HPI Function

**e·MMC v4.4**

System

- **Write**
- **Read**

CMD

- Write CMD25
- stop CMD12
- Read CMD18

DAT

- DATA

Device Busy

Host may have to wait long time to issue the read command

**e·MMC v4.41**

System

- **Write**
- **Read**

CMD

- Write CMD25
- stop CMD12
- Read CMD18
- Write CMD25

DAT

- DATA-0
- Device Busy
- DATA
- DATA-1
- Device Busy
- DATA-2

Arg[0]="1" Host can interrupt the long busy

Host can restart the rest of the write operation if needed
HPI Benefits

- Time consuming operations can dramatically reduce Paging efficiency, resulting in OS freeze
- By enabling the **HPI feature**:
  - The OS is able to suspend on-going operations (generated by user applications or background operation) to execute first the higher priority Paging operations
  - The suspended operation can be resumed by the e-MMC media driver as soon as the Paging operation is terminated
  - Paging performances are homogeneous even during Foreground time consuming operations
Combined Effects of Background Operation and HPI

- By enabling both **Background Operations and HPI features**:
  - Average eMMC Write throughput is improved for better user experience
  - Homogeneous paging latencies allows to configure easier the Paging Algorithm and avoids System freezing
  - On-going background operations can be postponed to process Foreground requests from user

Minimized number of time consuming operations
Reduced time to interrupt a time consuming operation
DDR Mode (52Mhz Max.)

- Transferred user data are sampled on both clock edge (DDR), but CMD line remains sampled clock rising edge (SDR)
  - The rising edge of the clock always capture odd numbered byte.
  - The falling edge of the clock always capture even numbered byte.
  - Block length is fixed as 512B.
  - Two CRC16 are computed per data line
    - (one for even bits and the other for odd bits)

Figure 35 — Stop transmission during data transfer from the host

Only data payload (+CRC if present) are double data rate
Start bit, End bit and idle conditions remains full cycle
Security and Data Integrity

eMMC v4.41/v4.4 specification introduces several new features that address the security and data integrity needs of high-performance handset platforms

- Addition of hardware RESET pin and signal definition
- Partition features to enable segregation of data
- Replay Protected Memory Block (RPMB)
- Multiple Write Protection definitions
- Secure Trim, Secure Erase
- Data Reliability definition
- Enhanced Reliable Write definition
Different Types of RESET

- **Hard Reset**
  - Power Cycle or triggered by H/W RESET (RSTN) signal
  - Write protection is removed in memory regions protected by Power-On WP

- **Soft Reset**
  - CMD0 arg=0x00000000 -- device moves to Idle state
  - CMD0 arg=0xF0F0F0F0 -- device moves to Pre-Idle state
  - Write protection in memory regions protected by Power-On WP is maintained
Partition Feature

- Host can configure partition in e-MMC device using extended CSD register
  - Independent addressable space
  - Partition size is multiple of a WP group size
  - Configuration is one time program
- Each partition can be configured with two types of attribute
  - Enhanced attribute (faster read/write access, better data integrity)
  - Default attribute (normal mass storage memory)
  - User may create enhanced area in the user area
Host configures max 4 partitions allowing system to separate code from user storage area
Use Case Example

**eMMC Physical Layout**

- Boot partition 1, 2
- User Area (Default)

**Partitioning**

- Partition 1 (Enhanced)
- Partition 2 (Enhanced)
- Partition 3 (Enhanced)
- User area (High density)

**Code Layout**

- XLOAD
- EBOOT
- IPL
- Logo
- MBR
- ULDR
- NK
- OS (Image FS)
- EXTFAT

Copying image data

*Flash Storage Summits 2010*
Replay Protected Memory Block (RPMB)

- This function provides means for the system to store data to the specific memory area in an authenticated and replay protected manner.
- RPMB operation is a separate self-contained security command protocol that has its own command opcodes (message types) and well-defined data structure.
- This feature is designed to fulfill the security requirements below:
  - EICTA CCIG Doc Ref: Eicta Doc: 04cc100
  - GSMA Doc Ref: Security Principles Related to Handset Theft 3.0.0
RPMB Requirements (Device Side)

- The Replay Protected Memory Block (RPMB) is defined as a separate partition in the e-MMC memory space
  - Partition size = multiples of 128KByte
- Secure storage of Authentication Key
  - An Authentication Key is written to the RPMB at host system manufacturing time and is used as shared secret to authenticate subsequent RPMB transactions between the Host and Device
- Transaction Authentication
  - Transactions (messages) are authenticated by the Message Authentication Code (MAC) which is a hash value generated by the Authentication Key, a random number provided by the Host and the message itself using HMAC SHA-256
Write Protect Feature

- **Permanent Write Protect**
  - Once the Permanent Write Protect is set, the protected memory region becomes read-only

- **Power-On Write Protect (Volatile Write Protect)**
  - Once the Power-on Write Protect is set, it is persistent until the next power cycle or H/W RESET
  - Host needs to re-set the Power-On Write Protect to memory regions that it wants to apply this type of write protection each time after power cycle or H/W RESET

- For boot partitions, Write Protect is applied to an entire boot partition

- For User Area Partition and General Partitions, Write Protect is applied to memory regions defined by multiples of Write Protect Group Size
Use Case Example

Partitioning

Boot partition 1, 2

Partition 1 (Enhanced)

Partition 2 (Enhanced)

Partition 3 (Enhanced)

User area (High density)

Code Layout

XLOAD

EBOOT

IPL

Logo

MBR

ULDR

NK

OS (Image FS)

EXTFAT

WP settings

Permanent WP

Power-On WP (4MB)

Power-On WP (4MB)

Power-On WP (4MB)

Power-On WP (4MB)

Power-On WP (4MB)

Power-On WP (8MB)

Power-On WP (256MB)

Unprotected

* Assuming minimum write protect group size is 4MB
Secure Trim/Secure Erase

• Secure Erase
  – When a Secure Erase command is sent, data in the specified memory addresses must be purged from the physical memory array
  – “Logical” memory erase is not acceptable

• Secure Trim
  – For cases where smaller amounts of data might be spread through multiple erase groups, a force garbage collect command is added
  – This allows the same function as Secure Erase to be performed on write blocks (Sectors) instead of erase groups
Secure Erase

- Secure Erase command sequences
  - CMD35 – Specify start address of erase groups
  - CMD36 - Specify end address of erase groups
  - CMD38 (with arg=0x80000000) – Erase operation

![Diagram showing Secure Erase command sequences]
Secure Trim

- Secure Trim command sequences

- **Step 1:**
  - CMD35 – Specify start address of write blocks to be erased
  - CMD36 - Specify end address of write blocks to be erased
  - CMD38 (with arg=0x80000001) - Keep write block address to be erased.
  - Host can repeat Step 1 sequence until all memory blocks to be erased is identified

- **Step 2:**
  - CMD35
  - CMD36
  - CMD38 (with arg=0x80008000) - Erase operation for the write blocks.
Secure Bad Block Management

- Allow the user to specify that bad blocks cannot contain any user data when they are retired.
- When blocks are discarded, all “good” bits must be purged before discarding.
- ECSD register [134] need to be set to execute this feature.

<table>
<thead>
<tr>
<th>Name</th>
<th>Field</th>
<th>Size (Bytes)</th>
<th>Cell Type</th>
<th>ECSD Bytes</th>
<th>ECSD Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bad block management mode</td>
<td>SEC_BAD_BLK_MGMNT</td>
<td>1</td>
<td>R/W</td>
<td>134</td>
<td>0h</td>
</tr>
</tbody>
</table>
Data Reliability

- Data Reliability is defined as followed
  - High data reliability: once a Device indicates to the Host that a write has successfully completed, the data that was written, along with all previous data written, cannot be corrupted by other operations that are host initiated, controller initiated or accidental (such as power interruption)
  - Normal data reliability: there is some risk that previously written data may be corrupted for unforeseen events such as power interruption

- Device indicates implementation of this feature in the ECSD registers
  - WR_REL_PARAM [166]: to indicate whether the Device supports this feature
  - WR_REL_SET [167]: to set High Data Reliability per partition

- Performance implication
  - Write performance may be impacted when high data reliability is set
Enhanced Reliable Write

- All blocks are 512B (sector) in length; each sector being modified by the write is atomic.
- If a power loss occurs during a Reliable Write, sectors may either contain old data or new data; all sectors being modified by the write operation may be in one of the following states:
  - All sectors contain new data
  - All sectors contain old data
  - Some sectors contain new data and some sectors contain old data
- REL_WR_SEC_C parameter has no meaning
- Device defines which type (legacy or new) is supported.
eMMC v4.5 Preview
eMMC v4.5 Primary Objectives

- Embedded-only specification
- Performance improvement/optimization
- Clarification of v4.41 functions and features
Some New Features under Consideration

- **Extending Partition Attributes**
  - Adding attribute registers to clearly define and distinguish the behaviors of individual partitions

- **Data Tag**
  - Providing information on the type and access frequency of the data being written

- **Real-Time Clock**
  - Adding a capability for the e·MMC device to receive real-time clock information from the Host such that certain time-sensitive operations internal to the e·MMC device may be improved

- **Power-Off Notification**
  - Adding a capability for the Host to notify the e·MMC of an impending power shutdown

- **Dynamic Device Capacity**
  - Extending the useful life of the e·MMC device by adding the capability of Host-initiated reduction of Device storage capacity in order to free up spare memory space to enable the e·MMC device to continue to function

- **Discard Command**
  - A variant of the TRIM command that is more memory technology friendly
In Conclusion
• e·MMC has established to be the dominant standard of managed, embedded mass-storage solution for Mobile

• New e·MMC v4.41 features address many advanced requirements in high-performance handset architecture

• v4.5 Preview – a peek into the future of the e·MMC Standard
Thank You