Lecture 23 – eMMC Architecture and Operation

ECG 721 – Memory Circuit Design, Spring 2017
Jonathan K DeBoy
Overview

• Flash memory programming and reading
• MultiMedia Card (MMC)
• Embedded MMC (eMMC)
• Packet Formats & Operation
• Future Technologies
Flash Memory Overview

Floating Gate Memory

Delta Sigma Sensing Flash Memory

ch16_16_3_p2_video

ch17_17_1_video
Figure 16.58  FNT of electrons from the p-well to a floating gate to increase threshold voltage (showing programming).

Source: CMOS Circuit Design Layout and Simulation Page 470
Cell Programming in NAND Architecture

Figure 16.62 Programming in a Flash NAND cell.
Erasing Process

**Figure 16.59** FNT of electrons from the floating gate to p-well to decrease threshold voltage (showing erasing).

Source: CMOS Circuit Design Layout and Simulation Page 470
Reading Process – Cell Properties

**Figure 16.64** Expanded view showing erased and programmed IV curves.
Reading Process – Sensing

Switch moves to bit line when clocked comparator output is high.

$V_{DD}$

$I_{cup}$

Switch moves to bit line when clocked comparator output is high.

Bit line

$C_{bit}$

$1 \text{ V}$

Yes, output high.

No, output low.

DSM's output

$\text{clock}$

Flash memory cell.

$I_{bit}$ ($I_{prog}$ or $I_{erased}$)

Figure 17.6  Sensing a Flash memory cell using DSM.

Source: CMOS Circuit Design Layout and Simulation Page 488
MultiMedia Card (MMC)
MMC Overview

• Developed by Joint Electron Device Engineering Council (JEDEC)
• **Standard** for flash memory
  • Low pin count
  • Cheap compared to large, high performance drives (also use NAND flash)
  • Portable, easily removed
  • Non-volatile

• SD is a **format** for flash memory
Typical MMC Application System Overview

Source: JEDEC Standard No. 84-A441 Figure 3
MMC Adaptor Architecture

Source: JEDEC Standard No. 84-A441 Figure 16
MMC Architecture

Source: JEDEC Standard No. 84-A441 Figure 4
MMC Controller

Source: JEDEC Standard No. 84-A441 Figure 62
## MMC Controller Power Supply Voltages

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Min</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply voltage (NAND)</td>
<td>$V_{CC}$</td>
<td>2.7</td>
<td>3.6</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.7</td>
<td>1.95</td>
<td>V</td>
</tr>
<tr>
<td>Supply voltage (I/O)</td>
<td>$V_{CCQ}$</td>
<td>2.7</td>
<td>3.6</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.65</td>
<td>1.95</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.1</td>
<td>1.3</td>
<td>V</td>
</tr>
<tr>
<td>Supply power-up for 3.3V</td>
<td>tPRUH</td>
<td></td>
<td>35</td>
<td>ms</td>
</tr>
<tr>
<td>Supply power-up for 1.8V</td>
<td>tPRUL</td>
<td></td>
<td>25</td>
<td>ms</td>
</tr>
<tr>
<td>Supply power-up for 1.2V</td>
<td>tPRUV</td>
<td></td>
<td>20</td>
<td>ms</td>
</tr>
</tbody>
</table>

Source: JEDEC Standard No. 84-A441 Table 106
Flash Translation Layer (FTL)

- Functions
  - Logical to physical address mapping
  - Power off recovery
  - Wear leveling (erase/write counts)
- RAM tables and Flash map storage
  - Dangerous for recovery
- FTL algorithms determine physical address
  - Block/Sector/Hybrid Mapping
  - Log Block Scheme

Source: Chung et al. Figure 1
Embedded MultiMedia Card (eMMC)
eMMC Overview

• Optimized for low power and small area
• Used in many mobile platforms: phones, microcontrollers,

• Combines the flash controller, interface adapter, and memory arrays together on the same Silicon dye
• NOT removable (perfect for OS/firmware)
• Very affordable, low tier performance

• Reset signal is unique to eMMC vs other MMC applications
eMMC System Overview

Source: JEDEC Standard No. 84-B51 Figure 1
Quick Definitions for Next Slides

• **open-drain**: A logical interface operation mode. An external resistor or current source is used to pull the interface level to HIGH, the internal transistor pushes it to LOW

• **push-pull**: A logical interface operation mode, a complementary pair of transistors is used to push the interface level to HIGH or LOW
## Communication

<table>
<thead>
<tr>
<th>Card state</th>
<th>Operation mode</th>
<th>Bus mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inactive State</td>
<td>Inactive mode</td>
<td>Open-drain</td>
</tr>
<tr>
<td>Pre-Idle State</td>
<td>Boot mode</td>
<td></td>
</tr>
<tr>
<td>Pre-Boot State</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Idle State</td>
<td>Card identification mode</td>
<td></td>
</tr>
<tr>
<td>Ready State</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identification State</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stand-by State</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sleep State</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transfer State</td>
<td>Data transfer mode</td>
<td>Push-pull</td>
</tr>
<tr>
<td>Bus-Test State</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sending-data State</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receive-data State</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programming State</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disconnect State</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boot State</td>
<td>Boot mode</td>
<td></td>
</tr>
<tr>
<td>Wait-IRQ State</td>
<td>Interrupt mode</td>
<td>Open-drain</td>
</tr>
</tbody>
</table>

Source: JEDEC Standard No. 84-A441 Table 5
11 Signal Bus

- CLK: 0-200MHz
- Data Strobe: output in DDR mode
  - Data output on rise and falling edges
  - CMD and CRC status is still on rising edge
- CMD
  - Bidirectional command channel
  - Open drain for initialization and push-pull for fast command transfer
- DAT[0-7]
  - Defaults to DAT[0] after boot
  - Larger bus widths are negotiated after bootup
    - Pull-ups disconnected when lines are in use
Load Capacitance

\[ C_L = C_{\text{HOST}} + C_{\text{BUS}} + C_{\text{DEVICE}} \]

the sum of the host and bus capacitances not to exceed 20 pF

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Unit</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pull-up resistance for CMD</td>
<td>( R_{\text{CMD}} )</td>
<td>4.7</td>
<td>100(^{[2]} )</td>
<td>100(^{[2]} )</td>
<td>kΩ</td>
<td>to prevent bus floating</td>
</tr>
<tr>
<td>Pull-up resistance for DAT0–7</td>
<td>( R_{\text{DAT}} )</td>
<td>10</td>
<td>100(^{[2]} )</td>
<td>100(^{[2]} )</td>
<td>kΩ</td>
<td>to prevent bus floating</td>
</tr>
<tr>
<td>Internal pull up resistance DAT1–DAT7</td>
<td>( R_{\text{in}} )</td>
<td>10</td>
<td>150</td>
<td>kΩ</td>
<td>to prevent unconnected lines floating</td>
<td></td>
</tr>
<tr>
<td>Bus signal line capacitance</td>
<td>( C_L )</td>
<td>30</td>
<td>pF</td>
<td></td>
<td>Single Device</td>
<td></td>
</tr>
<tr>
<td>Single Device capacitance</td>
<td>( C_{\text{DEVICE}} )</td>
<td>6</td>
<td>pF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum signal line inductance</td>
<td></td>
<td>16</td>
<td>uH</td>
<td></td>
<td>( f_{\text{pp}} \leq 52 \text{ MHz} )</td>
<td></td>
</tr>
<tr>
<td>( V_{\text{REG}} ) capacitor value</td>
<td>( C_{\text{REG}} )</td>
<td>0.1</td>
<td>1</td>
<td>uF</td>
<td>To stabilize regulator output</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>when target device bus speed</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>mode is either backward-compatible, high speed DDR,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>high speed DDR, or HS200.</td>
<td></td>
</tr>
<tr>
<td>( V_{\text{M2Q}} ) capacitor value</td>
<td>( C_{\text{REG2}} )</td>
<td>1</td>
<td>uF</td>
<td></td>
<td>To stabilize regulator output</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>when target device bus speed</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>mode is HS400.</td>
<td></td>
</tr>
<tr>
<td>( V_{\text{M1Q}} ) capacitor value</td>
<td>( C_{\text{REG3}} )</td>
<td>1</td>
<td>uF</td>
<td></td>
<td>To stabilize internal regulated voltage</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( V_{\text{CQ}} ) decoupling capacitor</td>
<td>( C_{\text{CQ}} )</td>
<td>1</td>
<td>uF</td>
<td>(3), (4), (5)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NOTE 1 Recommended maximum pull-up is 30 kΩ for 1.2 V and 50 kΩ for 1.8 V interface supply voltages. A 3 V part may use the whole range up to 100 kΩ.

NOTE 2 Recommended value for \( C_{\text{REG1}}, C_{\text{REG2}}, \) and \( C_{\text{REG3}} \) might be different between eMMC device vendors. 

NOTE 3 \( C_{\text{REG}} \) is \( V_{\text{REG}} \) decoupling capacitor required for HS200/HS400 eMMC device.

NOTE 4 \( C_{\text{REG}} \) should be placed adjacent to \( V_{\text{CCD}}, V_{\text{CCQ}} \) balls (around DAT7-0) (balls). It should be located as close as possible to these ball defined in order to minimize connection parasitics.

NOTE 5 eMMC device vendor may have more specific requirements for CH1 placement. Please confirm such requirements with specific eMMC device vendor.

Source: JEDEC Standard No. 84-B51 Table 200 / Page 247
Address Spaces

- Mapped Host Address Space
  - Usable space by the host software

- Private Vendor Specific Address Space
  - Cannot be accessed by read command by host
  - Contains firmware and mapping tables for the controller
  - No host data

- Unmapped Host Address Space
  - Cannot be read by host
  - May contain old host data or copies
    - Reason why government does not like flash/SSD memory

- 32 bit addresses imply 2GB max unless multiple sectors are used (512B in size each)
Packet Formats & Operation
Multiple Read Operation

Source: JEDEC Standard No. 84-B51 Figure 2
Multiple Write Operation

Source: JEDEC Standard No. 84-B51 Figure 3
Command Token Format

Source: JEDEC Standard No. 84-B51 Figure 5
Quick Definitions for Next Slide

• **CID**: Card Identification number register
• **CSD**: Card Specific Data register
Response Token Format

Source: JEDEC Standard No. 84-B51 Figure 6
Single Bit Bus Data Packet Format for SDR

Source: JEDEC Standard No. 84-B51 Figure 7
4 Bit Bus Data Packet Format for SDR

Source: JEDEC Standard No. 84-B51 Figure 7
8 Bit Bus Data Packet Format for SDR

Source: JEDEC Standard No. 84-B51 Figure 7
Data Packet Format for DDR

8 Bits bus DDR (DAT7-DAT0 used):

DAT7: 0 (start) X b7 (odd) b7 (even) ... b7 (odd) b7 (even) ... b15 (CRC odd) ... b0 (CRC odd) b0 (CRC even) 1 (odd) X
DAT6: 0 (start) X b6 (odd) b6 (even) ... b6 (odd) b6 (even) ... b15 (CRC odd) ... b0 (CRC odd) b0 (CRC even) 1 (odd) X
DAT5: 0 (start) X b5 (odd) b5 (even) ... b5 (odd) b5 (even) ... b15 (CRC odd) ... b0 (CRC odd) b0 (CRC even) 1 (odd) X
DAT4: 0 (start) X b4 (odd) b4 (even) ... b4 (odd) b4 (even) ... b15 (CRC odd) ... b0 (CRC odd) b0 (CRC even) 1 (odd) X
DAT3: 0 (start) X b3 (odd) b3 (even) ... b3 (odd) b3 (even) ... b15 (CRC odd) ... b0 (CRC odd) b0 (CRC even) 1 (odd) X
DAT2: 0 (start) X b2 (odd) b2 (even) ... b2 (odd) b2 (even) ... b15 (CRC odd) ... b0 (CRC odd) b0 (CRC even) 1 (odd) X
DAT1: 0 (start) X b1 (odd) b1 (even) ... b1 (odd) b1 (even) ... b15 (CRC odd) ... b0 (CRC odd) b0 (CRC even) 1 (odd) X
DAT0: 0 (start) X b0 (odd) b0 (even) ... b0 (odd) b0 (even) ... b15 (CRC odd) ... b0 (CRC odd) b0 (CRC even) 1 (odd) X

4 Bits bus DDR (DAT3-DAT0 used):

CLK
LSByte (High nibble) MSByte (Low nibble)

DAT3: 0 (start) X b7 (odd) b7 (even) ... b15 (CRC odd) ... b0 (CRC odd) b0 (CRC even) 1 (odd) X
DAT2: 0 (start) X b6 (odd) b6 (even) ... b15 (CRC odd) ... b0 (CRC odd) b0 (CRC even) 1 (odd) X
DAT1: 0 (start) X b5 (odd) b5 (even) ... b15 (CRC odd) ... b0 (CRC odd) b0 (CRC even) 1 (odd) X
DAT0: 0 (start) X b4 (odd) b4 (even) ... b15 (CRC odd) ... b0 (CRC odd) b0 (CRC even) 1 (odd) X

Blocklength CRC

Notice that bytes data are not interleaved but CRC are interleaved.
Start and end bits are only valid on the rising edge. ("X": undefined)

Source: JEDEC Standard No. 84-B51 Figure 8
# Bus Speed Modes

<table>
<thead>
<tr>
<th>Mode Name</th>
<th>Data Rate</th>
<th>I/O Voltage</th>
<th>Bus Width</th>
<th>Frequency</th>
<th>Max Data Transfer (implies x8 bus width)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backwards Compatibility with legacy MMC card</td>
<td>Single</td>
<td>3 V/1.8 V/1.2 V</td>
<td>1, 4, 8</td>
<td>0-26 MHz</td>
<td>26 MB/s</td>
</tr>
<tr>
<td>High Speed SDR</td>
<td>Single</td>
<td>3 V/1.8 V/1.2 V</td>
<td>1, 4, 8</td>
<td>0-52 MHz</td>
<td>52 MB/s</td>
</tr>
<tr>
<td>High Speed DDR</td>
<td>Dual</td>
<td>3 V/1.8 V/1.2 V</td>
<td>4, 8</td>
<td>0-52 MHz</td>
<td>104 MB/s</td>
</tr>
<tr>
<td>HS200</td>
<td>Single</td>
<td>1.8 V/1.2 V</td>
<td>4, 8</td>
<td>0-200 MHz</td>
<td>200 MB/s</td>
</tr>
<tr>
<td>HS400</td>
<td>Dual</td>
<td>1.8 V/1.2 V</td>
<td>8</td>
<td>0-200 MHz</td>
<td>400 MB/s</td>
</tr>
</tbody>
</table>

Source: JEDEC Standard No. 84-B51 Table 4
Interfacing with eMMC

• Interfacing with the eMMC controller is akin to microcontroller programming
  • Many different control registers

• Bus Width Selection Example:
  • SWITCH command issued on CMD
  • Write to the BUS_WIDTH byte in Modes Segment of EXT_CSD reg

7.4 Extended CSD register (cont’d)

7.4.67 BUS_WIDTH [183]
It is set to ‘0’ (1 bit data bus) after power up and can be changed by a SWITCH command. Bus Width, Normal or DDR mode and Strobe mode (for HS400) are defined through BUS_WIDTH register.

Source: JEDEC Standard No. 84-B51 Page 203
CRC 16 Generation (Payload)

Generator polynomial \( G(x) = x^{16} + x^{12} + x^5 + 1 \)

\( M(x) = (\text{first bit}) \times x^n + (\text{second bit}) \times x^{n-1} + \ldots + (\text{last bit}) \times x^0 \)

\( \text{CRC}[15\ldots0] = \text{Remainder}[(M(x) \cdot x^{16})/G(x)] \)
Higher Performing Technologies
Universal Flash Storage (UFS) Architecture Overview

Source: JEDEC Standard No. 220C Figure 5-4 / 6-1
UFS Communication Overview

Source: JEDEC Standard No. 220C Figure 5-1
Solid State Drive (SSD)

Source: SSD Architecture and PCI Express Interface Figure 2.1 / 2.9
References


• A Survey of Flash Translation Layer – Tae-Sun Chung, et al.
  http://idke.ruc.edu.cn/people/dazhou/Papers/AsurveyFlash-JSA.pdf

• JEDEC Standard No. 84-A441
  https://www.jedec.org/standards-documents/docs/jesd84-a441

• JEDEC Standard No. 84-B51
  https://www.jedec.org/standards-documents/docs/jesd84-b51

• JEDEC Standard No. 220C
  https://www.jedec.org/standards-documents/docs/jesd220c

• SSD Architecture and PCI Express Interface – K. Eshghi & R. Micheloni
  https://link.springer.com/chapter/10.1007%2F978-94-007-5146-0_2