External Memory Options for Automotive Applications

FTF-ACC-F1242

Steve McAslan | Senior Member of Technical Staff

J U N E . 2 0 1 5
Agenda

• The Evolution of Memory Technologies in Automotive Controllers and Processors
• The S32V230 — A Modern Automotive Architecture
• On-board Non-volatile Memory Standards
  − QuadSPI, NAND Flash, eMMC, I²C
• Off-board non-Volatile Memory Standards
  − USB, SD, MMC
• SDRAM and LP SDRAM Standards
  − DDR1..3, LPDDR1..2
• The Boot Process
• Summary
NVM Capacity

- **6801**
  - 2 k ROM
  - 128 RAM
  - 8 k ROM
  - 256 RAM
  - 512 EEPROM

- **68HC11**
  - 32 k Flash
  - 1 k RAM
  - 768 “EEPROM”

- **68HC12**
  - 256 k Flash
  - 12 k RAM
  - 4 k “EEPROM”

- **HCS12**
  - 512 k Flash
  - 32 k RAM
  - 4 k “EEPROM”

- **HCS12X**
  - 512 k Flash
  - 64 k RAM
  - 64 k “EEPROM”

- **MPC5600**
  - 6 M Flash
  - 768 k RAM

- **MPC5700**
  - 512 k Flash
  - 64 k RAM
  - 64 k “EEPROM”

Timeline:
- 1979
- 1984
- 1995
- 2000
- 2004
- 2009
- 2015
Perspective — Body, Powertrain, Chassis

Maximum RAM Capacity (kB)

<table>
<thead>
<tr>
<th>Year</th>
<th>RAM Capacity (kB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1975</td>
<td>0.125</td>
</tr>
<tr>
<td>1980</td>
<td>0.25</td>
</tr>
<tr>
<td>1985</td>
<td>0.5</td>
</tr>
<tr>
<td>1990</td>
<td>1</td>
</tr>
<tr>
<td>1995</td>
<td>2</td>
</tr>
<tr>
<td>2000</td>
<td>4</td>
</tr>
<tr>
<td>2005</td>
<td>8</td>
</tr>
<tr>
<td>2010</td>
<td>16</td>
</tr>
<tr>
<td>2015</td>
<td>32</td>
</tr>
<tr>
<td>2020</td>
<td>1024</td>
</tr>
</tbody>
</table>
Perspective — Body, Powertrain, Chassis

NVM Capacity (kB)


1 2 4 8 16 32 64 128 256 512 1024 2048 4096 8192
Perspective — Infotainment, Radio

NVM Capacity (kB)

- 8192
- 4096
- 2048
- 1024
- 512
- 256
- 128
- 64
- 32
- 16
- 8
- 4
- 2
- 1

2008 2009 2010 2011 2012 2013 2014
Perspective — Infotainment, Radio, ADAS Vision

NVM Capacity (kB)

S32V200
Perspective — Infotainment, Radio, ADAS Vision

Maximum **RAM** Capacity (kB)

<table>
<thead>
<tr>
<th>Year</th>
<th>0.125</th>
<th>0.25</th>
<th>0.5</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1024</th>
<th>2048</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008</td>
<td>0.125</td>
<td>0.25</td>
<td>0.5</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128</td>
<td>256</td>
<td>512</td>
<td>1024</td>
<td>2048</td>
</tr>
<tr>
<td>2009</td>
<td>0.125</td>
<td>0.25</td>
<td>0.5</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128</td>
<td>256</td>
<td>512</td>
<td>1024</td>
<td>2048</td>
</tr>
<tr>
<td>2010</td>
<td>0.125</td>
<td>0.25</td>
<td>0.5</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128</td>
<td>256</td>
<td>512</td>
<td>1024</td>
<td>2048</td>
</tr>
<tr>
<td>2011</td>
<td>0.125</td>
<td>0.25</td>
<td>0.5</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128</td>
<td>256</td>
<td>512</td>
<td>1024</td>
<td>2048</td>
</tr>
<tr>
<td>2012</td>
<td>0.125</td>
<td>0.25</td>
<td>0.5</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128</td>
<td>256</td>
<td>512</td>
<td>1024</td>
<td>2048</td>
</tr>
<tr>
<td>2013</td>
<td>0.125</td>
<td>0.25</td>
<td>0.5</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128</td>
<td>256</td>
<td>512</td>
<td>1024</td>
<td>2048</td>
</tr>
<tr>
<td>2014</td>
<td>0.125</td>
<td>0.25</td>
<td>0.5</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128</td>
<td>256</td>
<td>512</td>
<td>1024</td>
<td>2048</td>
</tr>
<tr>
<td>2015</td>
<td>0.125</td>
<td>0.25</td>
<td>0.5</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128</td>
<td>256</td>
<td>512</td>
<td>1024</td>
<td>2048</td>
</tr>
<tr>
<td>2016</td>
<td>0.125</td>
<td>0.25</td>
<td>0.5</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128</td>
<td>256</td>
<td>512</td>
<td>1024</td>
<td>2048</td>
</tr>
</tbody>
</table>

**S32V200**
Direction of Travel and the S32V234

• Some automotive applications have long relied on external memory
  – But typically this meant infotainment
• As higher performance applications become embedded in the car, they will also rely on external memory

• This brings massive flexibility in memory size and ratio of non-volatile vs volatile
  – At an additional system complexity

• Lower end and some specific applications will continue to rely on internal memories
A note on cache

• Neither flash nor traditional SRAM not external SDRAM is fast enough to support processors running at full speed
  – Small cache memories provide a smart buffer to avoid bottlenecks
  – A very smart buffer

• We use multiple levels of cache on-chip
  – On S32V234 two level cache is provided

• It can’t hold all the main memory and so must store addresses of the data and instructions too
  – This is why it must be very smart

• Not all memory regions are suitable for caching
  – Peripherals
  – Shared memory
On-Board Non-Volatile Memory Standards

- I²C
- SDIO/Embedded SD/eMMC
- NAND flash
- QuadSPI
• Incredibly slow
  – Up to 1 Mb/s in an optimized set up
• Small memory capacity (~256 kB)
• Simple to connect (2 pins)

• Potentially useful as a replacement for true EEPROM
eMMC

• Embedded version of SD card interface
• NAND flash technology for low cost and high density
• Same disadvantages as other NAND flash technologies
• Very large memory sizes available
• Not memory mapped
  – No XiP

• Typical existing instantiations 52 MB/s
  – eMMC v5 goes to 400 MB/s in DDR mode

• Good for mass storage at low cost
Layout:
50ohm, SD signals (SD_DATAx, SD_CMD, SD_CLK) control.
NAND Flash Overview

• Command-driven access to read/write
  – No address bus
  – > 20 I/O and control pins for a 16-bit interface
• Page-oriented, not suitable for random access or XiP
• Limited operating frequency
  – 40–50 MHz peak
• Bit errors can occur
  – Both SLC and MLC NAND flashes suffer from bit flips
  – ECC is a must — either on controller or integrated into NAND
• Very large memories available
• Higher density/lower cost than regular flash (NOR flash)
• Consider as mass storage option
Example NFC Interface
Serial Flashes

- Serial Quad flashes offer multiple advantages:
  - Low pin count as compared to parallel flashes
  - High bandwidth and low protocol overhead
  - Available in auto qual grade
  - High bandwidth enables Execute-in-Place (XiP) applications.
  - Available as KGD for SIP(System-In-Package)
  - Available now from multiple vendors
- Mass storage or XiP or data storage
Quad IO Read Example (DDR / DTR)

- Address and data samples on both rising and falling edges of SCLK
Quad SPI Features

- Dual QuadSPI architecture
  - 2 serial flashes per QuadSPI module
  - 104 MHz SDR and 80 MHz DDR
  - Programmable Sequence Engine (LUT)
  - XiP (Execute-In-Place)
  - Supports up to 4 chip selects
  - 2 x 4-bit serial flashes
  - 2 x 8-bit serial flashes
- Flexible Receive (RX) Buffering
  - Allocated to specific masters.
  - Master prioritisation (suspend and resume)
  - Pre-fetch capability
QuadSPI Parallel Access Mode

- This mode allows two identical serial flash devices to be connected and accessed in parallel forming one (virtual) flash memory with doubled readout bandwidth.

- Parallel Flash Mode is valid only for commands related to data read from the serial flash.

- Configuration:
  - The 1st device of Flash A has to be paired with the 1st device of Flash B.
  - The 2nd device of Flash A has to be paired with the 2nd device of Flash B.

- The incoming address is divided by 2 and sent to the two flashes connected in parallel.
Quad SPI Programmable Sequence Engine

- Works on a set of “instruction-operand” pair programmed by the user
- Allows the user to configure the Quad SPI module according to the serial flash connected on board thus supporting command/protocol from different vendors
- Each sequence is a sequence of instruction-operand pairs which when executed sequentially generates a valid serial flash transaction
- The LUT consists of pre-programmed sequences
- Drives the flexible I/O controller to generate serial flash patterns
Quad SPI Flexible Multi-Master Access

- Reduce read latency from serial flash
- Flexible and configurable buffers (up to 4) for multiple masters with priority
- Each buffer is associated with a Master and “datasize” representing amount of data to be fetched on every “missed” access
- Optionally, Buffer3 can be configured as “All Master” buffer. Any access from a master not associated with any other buffer is routed to Buffer3
Quad SPI: “Data learning”

- Only applicable for Quad SPI DDR mode
- Provides a known pattern on the IOs for the controller to self-calibrate
- However this is not standardized across vendors
- The QuadSPI controller provides a flexible instruction set that can emulate “Data learning” in any flash memory

Flexible instruction set sequence given to flash

<table>
<thead>
<tr>
<th>INSTR</th>
<th>ADDR</th>
<th>MODE</th>
<th>DATA_LEARN</th>
</tr>
</thead>
</table>

QSPI “Data learning” Logic

‘DATA LEARN’ pattern read

Pre-programmed location read

FLASH

DATA LEARN

Pattern read
Off-Board Non-Volatile Memory Standards

- **USB**
  - Integrated PHY means that schematic design is straightforward
    - USB host, power supply & protection requirements add complexity
  - Requires software stack to conform to USB standards
    - USB device communication is based on *pipes* (logical channels). A pipe is a connection from the host controller to a logical entity, found on a device, and named an *endpoint*. External storage/update options

- **SD/MMC**
  - Simple 4/8-bit interface
  - Command interface
Simple USB

Layout: 90 ohm differential pairs

NXP

freescale™

External Use  |  26  #FTF2015
Amphenol 101-00565-64 SD / MMC socket with card detection switch.
SDRAM and LP SDRAM Standards

• A note on terminology
• SDRAM standards
• Board considerations
A Note on Terminology

• SDRAM
  – Synchronous dynamic random access memory
  – Nowadays it is synonymous with “DDR”

• BUT
  – The first generation of SDRAM was not DDR
  – It was Single Data Rate
    ▪ SDR

• So SDR RAM can mean two different things…
A Note on Memory Technology

• The raw speed of a RAM (and Flash) cell has not increased much in the past 10 years
  – RAM c. 5 ns
  – Flash c. 10 ns

• We can operate processors only by careful design of memory banks, caches, buffers and parallelism

• This fact explains why the development of SDRAM is as it is
# Generations of SDRAM

<table>
<thead>
<tr>
<th>Generation</th>
<th>Top speed/Voltage</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDR</td>
<td>200 MHz/3.3V</td>
<td>Single data rate</td>
</tr>
<tr>
<td>DDR1</td>
<td>200 MHz/2.5V</td>
<td>Double data rate — minimum two word read/write</td>
</tr>
<tr>
<td>DDR2</td>
<td>533 MHz/1.8V</td>
<td>Minimum four word read/write, ODT</td>
</tr>
<tr>
<td>DDR3</td>
<td>1066 MHz/1.5V (1.35V)</td>
<td>Minimum eight word read/write, fly-by topology</td>
</tr>
<tr>
<td>DDR4</td>
<td>1600 MHz/1.2V</td>
<td></td>
</tr>
</tbody>
</table>
SDRAM Features

• The system is truly synchronous
• The SDRAM communicates with the processor using commands which direct a state machine controlled by the system clock
• The state machine is programmable for optimal performance and includes its own DLL clock correction system
  – Memory latencies, drive strength, burst length, etc.
• The command sequence dictates the actual bandwidth delivered by the system
  – In most cases a command is to read or write memory
• The SDRAM is not available to the system at reset because it must be configured before use
Micron DDR3 State Machine
SDRAM Architecture

• Logic and memory are tightly integrated
  – By combining multiple independent banks of DRAM it is possible to achieve very high peak and sustained bandwidth on the memory bus
    ▪ One bank can be prepared while another is delivering data
• Memory is implemented as a large matrix with >8192 bits per row
  – Typical 4 Gbit memory with 16-bit interface has:
    ▪ 32768 rows of 16384 bits in eight banks (32768*16384*8 = 4 Gbit)
    ▪ So an address consists of 15 row select lines, 10 column select lines and 3 bank select lines
      ▪ The processor must map its 32-bit address appropriately to these lines
• Due to the command/state machine nature of the memory there are losses in bandwidth due to scheduling the next access
  – In a typical multi-master system we expect utilisation in 50% - 70% range
Micron 32 M * 8 Banks * 16-bit (4 Gbit)
Hardware Notes

• The row and column addresses are multiplexed on a single address input
  – Reduces number of address lines needed and recognizes fundamental memory organisation

• Bank addresses are on separate lines in DDR3 (usually 2 or 3)

• SDRAM comes organised in 8-bit lanes and each has individual read/write masks
  – A global read or write signal determines data direction

• DRAM controller (processor) must look after refresh cycling and all other timing parameters
  – Timing parameters are normally expressed as a multiple of clk
Design Challenges

- SDRAM is very fast so tracks have to be considered as transmission lines
- DDR2/3 helps by providing On-Die-Termination (ODT)
  - This switches termination resistors on/off depending on the direction of signal travel
Pad Configuration

- For correct operation the DRAM pads must be configured to match the attached circuitry
- This is typically configured in the IOMUX or SIUL module
- Features to set include
  - ODT values (17, 20, 30, 40, 60, 120)
  - SDRAM type (DDR3, DDR2, LPDRAM2, …)
  - Timing adjust (for bit signal adjustments)
  - Drive strength (to match track impedance)
- Matching settings are available on the DRAM device

- Both the SDRAM and the SoC contain calibration capabilities to adjust drive strength and timing dynamically
DDR3 “Fly-By” Routing Topology

- DDR3 added “Fly-by” architecture
  - Address, command, control & clocks
  - Improved signal integrity enabling higher speeds
  - On module termination
During a write cycle, the skew between the clock and strobes are increased due to the fly-by topology. The write leveling will delay the strobe (and the corresponding data lanes) for each byte lane to reduce/compensate for this delay.
DDR3 Read Adjustment

- Automatic CAS to preamble calibration
- Data strobe to data skew adjustment

Recommended JEDEC’s MPR method.
Notes on DRAM Timing Specification (tRCD-tRP-CL)

• When the memory controller needs to access a different row, it must first return that bank's sense amplifiers to an idle state, ready to sense the next row. This is known as a "precharge" operation, or "closing" the row
  - tRP — time when a row access is available after a precharge command
• The active command activates an idle bank. It presents a bank address and a row address and causes a read of that row into the bank's array of all column sense amplifiers. This is also known as "opening" the row. This operation has the side effect of refreshing the dynamic (capacitive) memory storage cells of that row
  - tRCD — time from ACTIVATE to internal READ or WRITE
• CAS read latency (CL), the time between supplying a column address and receiving the corresponding data
SDRAM and refresh

• The contents of dynamic RAM must be periodically refreshed
• The refresh rate has an impact on the overall bandwidth
  – Every cell must be refreshed every 64 ms
• Higher temperatures cause RAM cells to leak faster and so refresh rates must be faster
  – At higher temperatures the theoretical bandwidth of the RAM busses are reduced
  – JEDEC specifies doubling the refresh rate above 85°C case temperature
  – ODT and other factors also derate
  – Even at high temperatures the bandwidth lost is < 1% if the system is configured optimally
• LPDRAM includes a temperature sensor which can be used to adjust its refresh strategy
ISSI LPDDR2 Refresh illustration

Regular Distributed Refresh Pattern
SDRAM Configuration

- There are typically three software tasks required before an SDRAM can be used
  - Configure the processor I/O for the board conditions (layout, speed, etc) and the connections to the DRAM (number of data and address lines, CS, etc)
  - Configure the SDRAM for the operating conditions (I/O, timing conditions)
  - Configure the processor for the SDRAM connected (SDRAM timing, memory size, etc)
DDR3 Schematic Example

DDR3 Memory - 1 GByte

Label = DDR Bytes 0x2

Label = DDR Bytes 1x3

DDR3 Memory Flyway Termination

Termination Regulator for DDR-DRAM

DDR3 DRAM0 Decoupling

DDR3 DRAM1 Decoupling

Clock terminators: Place at branch of DIMM signals to V0 and V0
Software Configuration of SDRAM
LPDDR2 vs DDR3

- LPDDR provides a lower power capability for an embedded system
- It adds some features which are not available to DDR3 but also changes the interface
- Key differences
  - Different operating voltage (1.2 V vs 1.5 V)
  - Unified command/address/bank interface
  - No ODT
  - Ability to read and write internal registers
  - Temperature sensor and self-refresh
- LPDDR2 and DDR3 are not pin or command compatible
The Boot Problem

- The classic microcontroller boot process is very simple
  - Soft- or hard-reset
  - Fetch reset vector from fixed or configurable location in internal NVM
  - Begin executing code

- Modern architectures have a problem because the NVM is not on-chip
  - External NVM may or may not support XiP
  - If it does it will be much slower than the device can operate
Typical System with NOR/NAND Flash

• After the power is turned on a dedicated boot program (stored in ROM) begins to execute on a core
  – Configures whatever clocks it needs and the peripherals it will use

• In a typical system:
  – Executes code from an external NVM
  – This program copies code to main memory (SDRAM)
  – Starts the execution from the main memory
Simplified Boot Flow Diagram

RESET

Check type

Check BOOT_CFG

Primary Boot
Download(opt.) & Authenticate

Serial Boot
D/load and Authen. Via USB/UART

Recovery Boot
D/load and Authen. Via I2C/SPI

Enter wake-up procedure

Execute Image
Boot and SDRAM

• The boot process will allow optimized configuration of the system for the SDRAM connected
• This is normally done as a series of records which configure the IO, the SDRAM controller and the SDRAM itself as required
• The boot process can then continue and have the SDRAM available as a target memory
Summary

• Modern architectures have much more flexibility in memory size and technology
• These optimize bandwidth and provide scalable performance
• Different memory technologies are optimized for different application uses
• The use case will drive the choice of solution
• Embedded flash is still the an appropriate solution for many systems
### Application Use Case Examples

<table>
<thead>
<tr>
<th></th>
<th>Very Slow</th>
<th>Slow</th>
<th>Medium</th>
<th>Fast</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass storage</td>
<td></td>
<td>NAND</td>
<td>QuadSPI</td>
<td></td>
</tr>
<tr>
<td>Data storage</td>
<td></td>
<td>QuadSPI</td>
<td>SDRAM</td>
<td></td>
</tr>
<tr>
<td>External storage</td>
<td>SD/MMC</td>
<td>USB2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Parameters</td>
<td>I²C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS/System/Code</td>
<td></td>
<td>QuadSPI</td>
<td>SDRAM</td>
<td></td>
</tr>
<tr>
<td>System variables</td>
<td></td>
<td></td>
<td>SDRAM</td>
<td>OCRAM</td>
</tr>
<tr>
<td>High bandwidth use</td>
<td></td>
<td></td>
<td></td>
<td>OCRAM</td>
</tr>
</tbody>
</table>

Underlying everything is also instruction and data cache