Introduction on DRAM Interface

Won-Joo Yun

2011. 11. 18
What is ‘Memory’?

• 1. the mental capacity or faculty of retaining and reviving facts, events, impressions, etc., or of recalling or recognizing previous experiences.

• ...

• Also called computer memory, storage.
  – a. the capacity of a computer to store information subject to recall.
  – b. the components of the computer in which such information is stored.

[dictionary.com]
What is ‘DRAM’?

• **Dynamic Random Access Memory**
  - RAM
    • Unlike electromagnetic tape or disk, it allows stored data to be accessed in any order (i.e. at random)
    • “Random” refers to the idea that any piece of data can be returned in a constant time, regardless of its physical location and whether it is related to the previous piece of data [wikipedia.com]
  - Dynamic
    • vs. static
    • needs refresh
    • the charge stored on the input capacitance will leak off over time

[3Tr Cell of 1k DRAM]
What is ‘DRAM’?

Sequential access

long time to access
different access time of locations

Random access

constant time to access
regardless of location
## Semiconductor Memory

<table>
<thead>
<tr>
<th>RAM</th>
<th>DRAM</th>
<th>1 Tr. + 1 Cap.</th>
<th>Dynamic (Need refresh)</th>
<th>Volatile</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SRAM</td>
<td>4 Tr. or 6 Tr.</td>
<td>Static</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FeRAM</td>
<td>1 Tr. + 1 Cap.</td>
<td>Almost Static</td>
<td></td>
</tr>
<tr>
<td>ROM</td>
<td>Mask ROM</td>
<td>1 Tr. (Single Poly)</td>
<td>Not Erasable</td>
<td>Non-Volatile</td>
</tr>
<tr>
<td></td>
<td>EPROM</td>
<td>1 Tr. (Dual Poly)</td>
<td>Erasable by UV</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EEPROM</td>
<td>1 Tr. (Dual Poly)</td>
<td>Electrically Erasable (by bit)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FLASH</td>
<td>1 Tr. (Dual Poly)</td>
<td>Electrically Erasable (by block)</td>
<td></td>
</tr>
</tbody>
</table>
## Memory comparison

<table>
<thead>
<tr>
<th></th>
<th>DRAM</th>
<th>SRAM</th>
<th>FLASH</th>
<th>FeRAM</th>
<th>MRAM</th>
<th>PRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mechanism for data storage</strong></td>
<td>charge and discharge of Cap.</td>
<td>switching of cross-coupled inv.</td>
<td>charge and discharge of F.G.</td>
<td>Dipole switching of Ferro-Cap</td>
<td>resistivity with magnetic polarization state</td>
<td>resistivity with chalcogenide material phase change</td>
</tr>
<tr>
<td><strong>Access Time</strong></td>
<td>&lt; 100ns</td>
<td>&lt; 50ns</td>
<td>&lt; 100ns</td>
<td>&lt; 100ns</td>
<td>&lt; 50ns</td>
<td>&lt; 100ns</td>
</tr>
<tr>
<td><strong>Write Time</strong></td>
<td>&lt; 100ns</td>
<td>&lt; 50ns</td>
<td>&lt; 10us</td>
<td>&lt; 100ns</td>
<td>&lt; 50ns</td>
<td>&lt; 500ns</td>
</tr>
<tr>
<td><strong>Erase Time</strong></td>
<td>No need</td>
<td>No need</td>
<td>~ms</td>
<td>No need</td>
<td>No need</td>
<td>No need</td>
</tr>
<tr>
<td><strong># of RD/WR operation</strong></td>
<td>R&amp;W infinite (&gt; 10^{15})</td>
<td>R&amp;W infinite (&gt; 10^{15})</td>
<td>10^8 ~ 10^{10}</td>
<td>10^{12} ~ 10^{16}</td>
<td>R&amp;W infinite (&gt; 10^{15})</td>
<td>10^9 ~ 10^{11}</td>
</tr>
<tr>
<td><strong>Data Retention Time</strong></td>
<td>need refresh</td>
<td>need not refresh</td>
<td>~ 10 years</td>
<td>~ 10 years</td>
<td>~ 10 years</td>
<td>~ 10 years</td>
</tr>
<tr>
<td><strong>Operating Current</strong></td>
<td>~ 100mA</td>
<td>~ 100mA</td>
<td>~ 10mA</td>
<td>~ 10mA</td>
<td>~ 10mA</td>
<td>~ 10mA</td>
</tr>
<tr>
<td><strong>Standby Current</strong></td>
<td>~ 200uA</td>
<td>~ 10uA</td>
<td>~ 10uA</td>
<td>~ 10uA</td>
<td>~ 10uA</td>
<td>~ 10uA</td>
</tr>
</tbody>
</table>

[Hynix]
# Memory cell structure

<table>
<thead>
<tr>
<th>Cell Structure Norm. size</th>
<th>DRAM</th>
<th>SRAM</th>
<th>NVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Tr. &amp; 1 cap. 1.0x</td>
<td>6 Tr. (4Tr. +2R) 3.0x</td>
<td>1 Tr. 0.6x</td>
<td></td>
</tr>
<tr>
<td><img src="image1" alt="DRAM Diagram" /></td>
<td><img src="image2" alt="SRAM Diagram" /></td>
<td><img src="image3" alt="NVM Diagram" /></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Application</th>
<th>Main memory Graphics</th>
<th>Buffer, Cache</th>
<th>Bios, Card Etc..</th>
</tr>
</thead>
</table>

- DRAM : Dynamic Random Access Memory
- SRAM : Static Random Access Memory
- NVM : Non-Volatile Memory, Flash

[Hynix]
Comparisons

Intel Penryn Dual Core
process: 45nm
die area: 107mm²
6MB L2 cache
⇒ 48Mb/38.5mm² = 1.25Mb/mm²

Micron DDR3 SDRAM
process: 42nm
die area: 49.2mm²
4Gb
⇒ 4Gb/43.3mm² = 92Mb/mm²

Intel-Micron (IM) Flash
process: 25nm
die area: 167mm²
64Gb
⇒ 64Gb/141mm² = 454Mb/mm²
Standard DRAM genealogy

Asynchronous

EDO

Synchronous

(D)RDRAM

XDR

XDR2

SDRAM

DDR

DDR2

DDR3

PAGE Mode

Fast PAGE Mode
# DRAM technology evolution

<table>
<thead>
<tr>
<th>Density</th>
<th>1K</th>
<th>4K</th>
<th>16K</th>
<th>64K</th>
<th>256K</th>
<th>1M</th>
<th>4M</th>
<th>16M</th>
<th>64M</th>
<th>256M</th>
<th>2G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design Rule (um)</td>
<td>10</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0.8</td>
<td>0.5</td>
<td>0.30</td>
<td>0.18</td>
<td>0.08</td>
</tr>
<tr>
<td>Chip Size (mm²)</td>
<td>10</td>
<td>13</td>
<td>26</td>
<td>30</td>
<td>35</td>
<td>50</td>
<td>70</td>
<td>110</td>
<td>140</td>
<td>160</td>
<td>200</td>
</tr>
<tr>
<td>Cell Size (um²)</td>
<td>3000</td>
<td>860</td>
<td>400</td>
<td>180</td>
<td>65</td>
<td>25</td>
<td>10</td>
<td>2.5</td>
<td>0.72</td>
<td>0.26</td>
<td>0.05</td>
</tr>
<tr>
<td>Power Supply (V)</td>
<td>20</td>
<td>12</td>
<td></td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>3.3</td>
<td>2.5</td>
<td></td>
<td>1.5</td>
</tr>
<tr>
<td>Operation Mode</td>
<td>SRAM</td>
<td>Page Mode</td>
<td>Fast Page Mode</td>
<td>EDO</td>
<td>SDR</td>
<td>DDR</td>
<td>DRD</td>
<td>DDR3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gate Oxide (nm)</td>
<td>120</td>
<td>100</td>
<td>75</td>
<td>35</td>
<td>30</td>
<td>20</td>
<td>16</td>
<td>12</td>
<td>9</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>Cell Type</td>
<td>3Tr</td>
<td>1Tr Planar Capacitor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>High ε</td>
</tr>
</tbody>
</table>

[Hynix]

2011-11-18
DRAM voltage trend

A: Determined by Cell Transistor punch through, Refresh time and SER.

B: Determined by Memory Cell Transistor Vth.

C: Determined by Burn In possibility.

D: Determined by Process margin.

E: Determined by bit line voltage and Logic speed.

[Hynix]
Lower voltage means slower device speed

Access time increases rapidly as voltage decreases.

Standby Current not good.

[Hynix]
DRAM Cell

DRAM unit cell: 1 Cell Transistor + 1 Capacitor

Invented in 1968 – R. H. Dennard/IBM
US Patent # 3,387,286
DRAM core structure

1) Memory Cell: 1T, 1C
2) X Decoder & Word Line
3) Bit Line
4) Sense Amp
5) Column Select
DRAM core operation

1) Bit line floating

2) Word line select:
   - Charge sharing

3) DRAM sensing:
   - Write recovery

4) Column select:
   - data read (or write)
Charge sharing

BL

Cell

‘1’

BL

Cell

‘0’

2011-11-18
Charge sharing

\[
V_{\text{BLP}} = \frac{1}{2} V_{\text{CC}} \quad V_{\text{CELL}} = V_{\text{CC}}, 0
\]

Stand by
\[
Q = C \times V = C_b \times V_{\text{BLP}} + C_s \times V_{\text{CELL}}
\]

Word line turn on
\[
Q = C \times V = (C_b + C_s) \times V_{\text{out}}
\]

\[
V_{\text{out}} = \frac{C_b \times V_{\text{BLP}} + C_s \times V_{\text{CELL}}}{C_b + C_s}
\]
BL SA operation

Cross-coupled sense amp

VPP = V_{CORE} + V_{TC} + \alpha

V_{CORE} = \text{High Data Level}

V_{SS} = \text{Ground}

[Hynix]
Memory I/O Interface
Why “Synchronous”? 

- **Asynchronous DRAM**
  - Page Mode DRAM
  - Fast Page Mode DRAM
  - EDO(Extended Data Out) DRAM

- **Synchronous DRAM**
  - SDRAM
  - DDR SDRAM
  - Rambus DRAM

- **Synchronous DRAM can output more data**
Conventional DRAM circuits

- **Cell Array**
  - (sub) Matrix Array, Cell (1T1C), WL, bit line (Folded)
  - Cap. - Data retention DRAM Tech. Core part.
  - SA array -- DRAM sensing, Refresh all page Cell
- **Decoder Mux, Add input**
  - Pre decoding → Decoding.
  - Redundancy, internal refresh counter,
  - Row address path, Col add path
- **Data I/O**
  - Read/write, + Data bus sense amp, block write Driver
  - Various DRAM according to Col.(add./data path) control
  - Fast page, EDO, SDRAM, DDR, DDR2,
  - PKG option x4, x8, x16
- **Control circuits**
  - Operation of Read, Write, Refresh (Timing & Selection) according to /RAS, /CAS, /WE
- **Internal bias voltage**
  - Vbb, Vpp, Vblp, Vcp, Vint, Vref
SDRAM features +

- **Pipeline**
  - In previous DRAM, column address path time determines data freq.
  - With partitioning internal path, data are outputted every clock cycle after 2 or 3 clocks

- **Clock input**
  - Up to EDO, input signals are directly controlled by /RAS, /CAS, /WE
  - Changed to command (referenced rising edge of clock) → various operation and simple spec.

- **Multi bank (2/4)**
  - Independent row access is consisted of multiple bank → increase the size of page
  - capable of continuous operation with hiding pre-charge time

- **Mode register set**
  - Programmable /CAS latency and burst length suitable for system environments (clock frequency)

- **Internal address generator**
  - Internally generates sequential column address for Burst (fast column access) operation

- **I/O Power**
  - Dedicated power of data (Vccq, Vssq) for stable operation
Pipeline

- separate signals having long access time for faster input command
Multi-bank Architecture

- Bank is a unit which can be active independently and has same data bus width as external output bus width.
- Interleaving bank operation → while one bank accessed, another active.
DRAM clock speed trends

[K-h Kim, et al. JSSC 2007]

[Hynix]
**DDR features**

- **DDR data I/O**
  - Double data rate = rising & falling edge of clock
  - Twice performance compared to SDRAM

- **DDR performance by 2n-bit pre-fetch**

- **On chip clock by DLL**
  - Frequency not limited to the access time

- **SSTL interface**
  - Input reference voltage
  - Guarantee of dout data window, termination

- **Differential input**
  - Reference by VREF

- **Data strobe by DQS**
  - Edge align, bi-directional, source synchronous

- **Differential clock**
  - CLK, /CLK

- **EMRS control**
  - Dout driver size & DLL
SDR/DDR/DDR2/DDR3 operation

Data Rate
SDR/DDR/DDR2/DDR3 operation

- **DDR** (2b pre-fetch)
- **DDR2** (4b pre-fetch)
- **DDR3** (8b pre-fetch)
SDR/DDR/DDR2/DDR3 operation

High bandwidth concept: pre-fetch
- Fetch cycle means one column cycle that is executed by read or write command issue
Memory I/O interface

- Memory clocking system
  - Source synchronous scheme
  - DLL supports
  - Impedance control

- Design trends on graphics memory
  - Low power techniques
    - Input – clock – output
  - Low cost techniques
    - Clock
  - Low jitter & high performance techniques
    - Clock – output
    - Power distribution network
Common clock scheme

- Data transfer is performed relative to a single master “clock” signal (Synchronously)
Common clock scheme

- Timing budget
Source Synchronous

- Common (master) clock is not used for data transfer
- Devices have an additional strobe pin
- Minimizing differences in routed length & layer characteristics between strobe and data signals is required

- Data / STB are synchronized at driver
- Device speed (fast, slow) is irrelevant since data & STB are supplied by the same device
- The significant issue is the accumulated skew between data & STB as the signals travel between devices
Source Synchronous

• Timing budget
  – ideal case: $t_{STB} = t_{DATA}$
    • $\rightarrow$ Maximum speed is limited only by setup + hold time
  – real: Maximum speed is limited by setup + hold + $|t_{STB} - t_{DATA}|_{max} + \ldots$

![Diagram of timing budget](image)
DLL supports SS scheme

Alignment needed

External CLK

Internal CLK (no DLL)

Internal CLK (w/ DLL)

Desired DQ

tD1 + tD2 have large P.V.T. variations

N'tCK – (tD1 + tD2)
DRAM interface on channel

[Diagram showing DRAM interface on channel with labels such as DRAM side, Controller side, Memory Core, Impedance control, Serialize/De-serialize, TX, RX, RTT, ODT control with programmed value]
TX driver (with impedance control)

Programming by OCD or Self tuning by ZQ CAL
Read data eye measurement (DDR)
On board termination resistance is integrated inside of DRAM
ODT value selection and on/off ctrl.

Resistor value should be set during initialization (EMRS(1)).

<table>
<thead>
<tr>
<th>A6</th>
<th>A2</th>
<th>Rtt (Nominal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>ODT Disabled</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>75 ohm</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>150 ohm</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>50 ohm</td>
</tr>
</tbody>
</table>

ODT turn-on period:
- tAOND = 2*tCK
- tAOFD = 2.5*tCK

[Samsung]
ODT case study @DDR2-667 writes

- For two slot population, 50ohm seems to be better than 75ohm in terms of signal integrity
- For one slot population, 150ohm seems OK
Signal integrity

To improve the bandwidth, it is critical to achieve sufficient data valid window at the receiver.

*Interface factors*

- Xtalk (including SSO & Bounce)
- ISI on Data/DQS
- Path matching
- Termination error
- Vref noise

[Hynix]
Interface for Graphics memory
GDDR3 applications

Game Consoles

Laptop / Mobile

High-End / D-T

DDR data rate (Gbps)

GDDR3

Application
Interface for Graphics memory

• Challenges for Graphics Memory
  – High-speed over 2Gbps for GDDR3, 7 Gbps for GDDR5
  – Low voltage under 1.35V for GDDR5, 1.8V for GDDR3
  – Low current consumption
  – Good quality of clock itself
  – Robust operation against various noisy environments
  – Guarantee of operation under various power down mode
Design trends

Low Power
- Reduce operating current
- Guarantee operations at low voltage
- Data output

Low Cost
- Small area
- Design for testability

Die cost down
Test cost down

High Performance
- Robust DLL
- Low jitter DLL
- Good quality of DCC
- Low SSO noise

Wide data valid window

Low heat, low voltage drop
DVS (Dynamic Voltage Swing) at mobile app.
Data Bus Inversion

2011-11-18
Clocking systems for DRAM interface

**Input**
- Input clock buffer
  - robust clock generation from poor input signal
  - support low power mode

**Clocks**
- DLL / PLL
  - delay (phase) compensation
  - wide operation range (voltage / frequency)
  - good quality of clock signal 
  → low jitter, duty-corrected clock

**Output**
- Clock control
  - output enable
- Driver
  - Impedance matching
  - Multi slew-rate
  - Data Bus Inversion

Alignment needed
Low power techniques

• **General concept**
  – Power consumption = \( V(\text{supply voltage}) \times I(\text{current}) \)

• **Input**
  – Buffer
    • in mobile: just inverters
    • in graphics: low current two-stage amps
  – Buffer with low power mode
    • guarantee of low power function in mobile applications
    • stable operation under off-terminated environments

• **Clock**
  – DLL
    • Architecture for low power consumption
    • Systematically low power operation

• **Output**
  – Data Bus Inversion DC mode
Low power in clock (DLL)

- **Architecture**
  - compact circuits and architecture
  - In digital DLL, Dual-loop → Single-loop
  - lower VDD than external VDD
    - Vperi using regulated power
    - decrease internal frequency [GDDR4]
    - minimize voltage drop
- **Smart power down control** [GDDR3]

**Synchronous DRAM Power Estimation**

\[
P_{\text{total}} = P_{\text{core}} + P_{\text{peri}} + P_{\text{output}} + P_{\text{standby}}
\]

\[
P_{\text{core}} = \alpha \left( V_{\text{core}} \cdot V_{\text{dd}} \right) / tRC (\approx 60\text{ns})
\]

\[
P_{\text{peri}} = \beta \left( V_{\text{peri}} \cdot V_{\text{dd}} \right) / tCK
\]

\[
P_{\text{output}} = \gamma(V_{\text{swing}} \cdot V_{\text{ddq}}) / tCK
\]

\[
P_{\text{standby}} = I_{\text{standby}} \cdot V_{\text{dd}}
\]

Vdd: External Power
Vcore: Internal Cell Array Power
Vperi: Internal Periphery Power
Vddq: External output Power

**Power Consumption vs. tCK**

- Proposed one
- Previous one

- **79% reduction**

VDD = 1.5V, Temp = 25 °C

[ISSCC '08]
Low power in output

- **DBI DC mode** [GDDR4]
  - data ‘0’ consumes current
  - maximum # of ‘0’ $\leq 4$

![Pseudo open drain I/O system](image)

<table>
<thead>
<tr>
<th>Case</th>
<th>DQ[1:8]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case1</td>
<td>00000000</td>
</tr>
<tr>
<td>Case2</td>
<td>00000001</td>
</tr>
<tr>
<td>Case3</td>
<td>00000011</td>
</tr>
<tr>
<td>Case4</td>
<td>00000111</td>
</tr>
<tr>
<td>Case5</td>
<td>00011111</td>
</tr>
<tr>
<td>Case6</td>
<td>00111111</td>
</tr>
<tr>
<td>Case7</td>
<td>01111111</td>
</tr>
<tr>
<td>Case8</td>
<td>11111111</td>
</tr>
<tr>
<td>Case9</td>
<td>11111111</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case</th>
<th>DQ[1:8]</th>
<th>DBI flag</th>
<th>Num. of 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case1</td>
<td>11111111</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Case2</td>
<td>11111110</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Case3</td>
<td>11111000</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Case4</td>
<td>11111000</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Case5</td>
<td>11110000</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Case6</td>
<td>00011111</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Case7</td>
<td>00111111</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Case8</td>
<td>01111111</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Case9</td>
<td>11111111</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Receiver type comparison

- Pseudo open drain (GDDR3) vs. Push-Pull (GDDR2)

- Assumption: Same channel condition for both cases.
- It doesn’t represent absolute number of ODT power difference between pseudo open drain case and push-pull case.
Low power in output

- **DBI DC mode** [GDDR4]
  - data ‘0’ consumes current
  - maximum # of ‘0’ ≤ 4

![Diagram showing pseudo open drain I/O system](image)

### Table 1: DBI DC Conditions

<table>
<thead>
<tr>
<th>Case</th>
<th>DQ[1:8]</th>
<th>DBI flag</th>
<th>Num. of 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case1</td>
<td>1111111</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Case2</td>
<td>1111110</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Case3</td>
<td>1111100</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Case4</td>
<td>1111000</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Case5</td>
<td>1110000</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Case6</td>
<td>0001111</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Case7</td>
<td>0011111</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Case8</td>
<td>0111111</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Case9</td>
<td>1111111</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

[SJ Bae. JSCC '08]
Low jitter / High performance

• **Clock**
  – DLL
    • Low jitter operation with dual-loop architecture
    • Power noise tolerant replica
    • Dual DCC for stable duty error correction
  – Dual-mode with DLL and PLL
    • DLL for phase lock, PLL for jitter reduction [ISSCC ‘09]
  – Meshed power plan

• **Output**
  – Driver
    • Data Bus Inversion AC mode : reduce SSO noise
Low jitter in output

- **DBI AC mode** [GDDR4]
  - reduce SSO noise
  - In data byte sequence, maximum # of change ≤ 4

---

**Power supply noise generation**

- Large L(dI/dt) noise!
- \(\Delta I_{DQ}\) : DC current when data is low

---

[ SJ Bae. JSCC '08 ]
GDDR5
## JEDEC GDDR SGRAM comparison

<table>
<thead>
<tr>
<th>Features</th>
<th>GDDR3 SGRAM</th>
<th>GDDR4 SGRAM</th>
<th>GDDR5 SGRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic Features</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRAM Density</td>
<td>256Mbit – 1Gbit</td>
<td>512Mbit</td>
<td>512Mbit – 2 Gbit</td>
</tr>
<tr>
<td>Data Rate</td>
<td>1000-2000 MT/s</td>
<td>1600-3200 MT/s</td>
<td>3200 – 5000 MT/s</td>
</tr>
<tr>
<td>VDD/VDDQ</td>
<td>1.8/1.8V</td>
<td>1.5/1.5V and 1.8/1.8V</td>
<td>1.5/1.5 V</td>
</tr>
<tr>
<td>VREF</td>
<td>VREF</td>
<td>VREFC, VREFD</td>
<td>Internal/External VREFD, VREFC</td>
</tr>
<tr>
<td>VREF Level</td>
<td>0.7 * VDDQ</td>
<td>0.7 * VDDQ</td>
<td>0.7 * VDDQ, 0.5 * VDDQ</td>
</tr>
<tr>
<td>External voltage regulator supply</td>
<td>No</td>
<td>No</td>
<td>VPP</td>
</tr>
<tr>
<td>Prefetch Scheme</td>
<td>4 bit</td>
<td>8 bit</td>
<td>8 bit</td>
</tr>
<tr>
<td>Burst Length</td>
<td>4, 8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>I/O Organization</td>
<td>x32</td>
<td>x32</td>
<td>x32 &amp; x16</td>
</tr>
<tr>
<td>DRAM PLL/DLL</td>
<td>DLL</td>
<td>DLL</td>
<td>PLL or DLL (optional)</td>
</tr>
<tr>
<td>Termination type</td>
<td>VDDQ</td>
<td>VDDQ</td>
<td>VDDQ</td>
</tr>
<tr>
<td>Driver Calibration (nominal)</td>
<td>ZQ Calibration (240ohm)</td>
<td>ZQ Calibration (240ohm)</td>
<td>ZQ Calibration (120ohm)</td>
</tr>
<tr>
<td>Driver Impedance (nominal)</td>
<td>PU=40 ohm PD=40ohm</td>
<td>PU=60 ohm PD=40ohm</td>
<td>PU=60 ohm PD=40ohm</td>
</tr>
<tr>
<td>Interface Training</td>
<td>No</td>
<td>Controller only</td>
<td>DRAM supported training protocol or Controller only. RD, WR, ADR and WCK2CK</td>
</tr>
<tr>
<td>On Die Thermal Sensor ODTS</td>
<td>No</td>
<td>No</td>
<td>Optional</td>
</tr>
<tr>
<td>RESET#</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Package</td>
<td>144ball FBGA (WB) 12mmx12mm</td>
<td>136ball WBGA (WB) 11-12mm x 14mm (ball-out different than GDDR3)</td>
<td>170 ball WBGA (WB/FC) 12mm x 14mm</td>
</tr>
</tbody>
</table>

[AMD(ATi)]

---

2011-11-18
# JEDEC GDDR SGRAM comparison

<table>
<thead>
<tr>
<th>Feature</th>
<th>GDDR3 SGRAM</th>
<th>GDDR4 SGRAM</th>
<th>GDDR5 SGRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Clocking</td>
<td>Source Synchronous WR &amp; RD</td>
<td>Source Synchronous WR &amp; RD</td>
<td>Source Synchronous WR / CDR RD</td>
</tr>
<tr>
<td>Data Strobe (DQS)</td>
<td>Single Ended Unidirectional DQS</td>
<td>Single Ended Unidirectional DQS</td>
<td>RDQS mode for low power/frequency mode only</td>
</tr>
<tr>
<td>Free Running Data Clock</td>
<td>No</td>
<td>No</td>
<td>Unidirectional Differential Clock for WRITE; No strobes or clock for READ</td>
</tr>
<tr>
<td>Error protection</td>
<td>No</td>
<td>No</td>
<td>Unidirectional 8bit CRC on 72 bits [(t× DQ + 1 DBi) x 8bit burst] for RD &amp; WR</td>
</tr>
<tr>
<td>DQ Termination (nominal)</td>
<td>60 and 120ohm (P2P and P22P)</td>
<td>60 and 120ohm (P2P and P22P)</td>
<td>60/120ohm (P2P)</td>
</tr>
<tr>
<td>DQ di/dt reduction</td>
<td>No</td>
<td>DBI (AC and DC)</td>
<td>DBI (DC)</td>
</tr>
<tr>
<td>DQ Preamble</td>
<td>No</td>
<td>No</td>
<td>DQ Preamble</td>
</tr>
<tr>
<td><strong>Address</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Addressing</td>
<td>1 cycle (14-16pins)</td>
<td>2 cycle (8pins)</td>
<td>DDR (8-9pins)</td>
</tr>
<tr>
<td>Address Command Clock (CK/CK#)</td>
<td>½ data rate</td>
<td>½ data rate</td>
<td>¼ data rate</td>
</tr>
<tr>
<td>Address/CMD/CTRL Topology</td>
<td>P22P/P24P</td>
<td>P2P/P22P</td>
<td>P2P/P22P</td>
</tr>
<tr>
<td>Address Termination (nominal)</td>
<td>120 and 240ohm</td>
<td>60 and 120ohm</td>
<td>60 and 120ohm</td>
</tr>
<tr>
<td>ADR di/dt reduction</td>
<td>No</td>
<td>No</td>
<td>ABI</td>
</tr>
<tr>
<td>DRAM Banks</td>
<td>4 (256Mbit) or 8 (512/1024Mbit)</td>
<td>8</td>
<td>8 (512Mbit) or 16 (1-2Gbit)</td>
</tr>
<tr>
<td>Bank Groups</td>
<td>No</td>
<td>No</td>
<td>4</td>
</tr>
</tbody>
</table>

Legend: Shared, Different, New

[AMD(ATi)]

2011-11-18
Industry signal interface trend

Q.
S.E or Diff?
GDDR5 – key elements for reliable high speed data transmission

GDDR5 combines three concepts to ensure High Performance – Stable System Operation – Low Implementation Costs

**Data eye optimization**

**Key Features**
- data / address bit inversion
- adjustable driver strengths
- adjustable voltages
- adjustable terminations

**Benefit**
- fast interface tweaking
- relaxed transmission lines
- reduced PCB costs

**Adaptive interface timing**

**Key Features**
- data training
- scalable per bit or byte according system need

**Benefit**
- no trace length matching
- reduced PCB costs
- stable system operation

**Error compensation**

**Key Features**
- error detection for read and write
- real time error detection allows fast re-send

**Benefit**
- less system margin needed
- stable system operation

[AMD(ATi), Qimonda]
Comparison GDDR3 vs. GDDR5

Synchronization issues on every pin → “combs” on PCB

No needs of “combs” on PCB → cheaper solution with higher performance

[AMD(ATi)]
Clamshell mode (x16 mode)

Graphics system designers expect GDDR5 standard to offer high flexibility in terms of frame buffer and bandwidth variation. GDDR5 supports this need for flexibility in an outstanding way with its clamshell mode. The clamshell mode allows 32 controller-I/Os to be shared between two GDDR5 components. In clamshell mode each GDDR5 DRAM’s interface is reduced to 16 I/Os. 32 controller I/Os can, therefore, be populated with two GDDR5 DRAMs, while DQ’s are single loaded and the address and command bus is shared between the two components. Operation in clam shell mode has no impact on system bandwidth.

Every GDDR5 component supports the clamshell mode. In this way, multiple frame-buffer variants can be built up using only one component type which drastically reduces the number of different inventory positions and increases flexibility in a very dynamic market environment.
Recent researches on DRAM I/F

<table>
<thead>
<tr>
<th>Ref</th>
<th>Applications</th>
<th>Conf.</th>
<th>Year</th>
<th>Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>[6]</td>
<td>GDDR3</td>
<td>ISSCC</td>
<td>2006</td>
<td>Latency control, 2.5Gbps</td>
</tr>
<tr>
<td>[5]</td>
<td>GDDR3</td>
<td>ASSCC</td>
<td>2006</td>
<td>Low power/Wide range DLL architecture, 3Gbps</td>
</tr>
<tr>
<td>[2]</td>
<td>GDDR3</td>
<td>ISSCC</td>
<td>2008</td>
<td>Dual DCC, 3Gbps</td>
</tr>
<tr>
<td>[10]</td>
<td>GDDR3</td>
<td>ISSCC</td>
<td>2008</td>
<td>Multi-slew-rate output driver, impedance control, 3Gbps</td>
</tr>
<tr>
<td>[1]</td>
<td>GDDR3</td>
<td>ISSCC</td>
<td>2009</td>
<td>Dual PLL/DLL, pseudo-rank, 3.3Gbps</td>
</tr>
<tr>
<td>[7]</td>
<td>GDDR5</td>
<td>ESSCIRC</td>
<td>2009</td>
<td>CML CDN, 5.2Gbps</td>
</tr>
<tr>
<td>[8]</td>
<td>GDDR5</td>
<td>VLSI</td>
<td>2009</td>
<td>Fast DCC, 7Gbps</td>
</tr>
<tr>
<td>[11]</td>
<td>GDDR5</td>
<td>ISSCC</td>
<td>2010</td>
<td>GDDR5 Architecture, Bank control, 7Gbps</td>
</tr>
<tr>
<td>[12]</td>
<td>GDDR5</td>
<td>VLSI</td>
<td>2010</td>
<td>Jitter and ISI reduction, 7Gbps</td>
</tr>
</tbody>
</table>
DDR4
# DDR4 Outlook

DDR4 adopts evolutionary path with High BW & reliability scheme

<table>
<thead>
<tr>
<th>Spec items</th>
<th>DDR3</th>
<th>DDR4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density / Speed</td>
<td>512Mbp<del>8Gb 1.6</del>2.1Gbps</td>
<td>2Gb<del>16Gb 1.6</del>3.2Gbps</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voltage (VDD/VDDQ/VPP)</td>
<td>1.5V/1.5V/NA (1.35V/1.35V/NA)</td>
<td>1.2V/1.2V/2.5V</td>
</tr>
<tr>
<td>Vref</td>
<td>External Vref (VDD/2)</td>
<td>Internal Vref (need training)</td>
</tr>
<tr>
<td>Data IO</td>
<td>CTT (34ohm)</td>
<td>POD (34ohm)</td>
</tr>
<tr>
<td>CMD/ADDR IO</td>
<td>CTT</td>
<td>CTT</td>
</tr>
<tr>
<td>Strobe</td>
<td>Bi-dir / diff</td>
<td>Bi-dir / diff</td>
</tr>
<tr>
<td># of banks</td>
<td>8Banks</td>
<td>16Banks (4BG)</td>
</tr>
<tr>
<td>Core architect</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Page size (X4/8/16)</td>
<td>1KB / 1KB / 2KB</td>
<td>512B / 1KB / 2KB</td>
</tr>
<tr>
<td># prefetch</td>
<td>8bits</td>
<td>8bits</td>
</tr>
<tr>
<td>Added function</td>
<td>RESET/ZQ/Dynamic ODT</td>
<td>+ CRC/DBI/Multi preamble ..</td>
</tr>
<tr>
<td>Physical</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Package type/balls</td>
<td>78 / 96 BGA</td>
<td>78 / 96 BGA</td>
</tr>
<tr>
<td>(X4,8/X16)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIMM type</td>
<td>R,LR,U,SoDIMM</td>
<td>+ ECC SoDIMM</td>
</tr>
<tr>
<td>DIMM pins</td>
<td>240 (R,LR,U) / 204 (So)</td>
<td>284 (R,LR,U) / 256 (So)</td>
</tr>
</tbody>
</table>
DDR4
Server Memory Evolution Path: Type

- **ECC UDIMM (Unbuffered DiMM)**
- **RDIMM (Registered DiMM)**
- **FBDIMM & LRDIMM (Load Reduced DiMM)**

Cost / Density / Reliability / Speed

Diagram showing the flow of data between MCH, CA, CK, RCD, DRAM, and buffer.
Buffer at the “heart” of New Technology: Load-Reduced DIMM

- Memory Buffer LRDIMM solves the Channel Speed and SI challenges by “reinforcing” all signals for command, address, control and data on the Memory-controller interface.
Potential server applications

- LRDIMMs (Load Reduced DIMMs) overcome the problem of limited chip selects in the channel as LRDIMM support “Rank Multiplication”. With rank multiplication a LRDIMM can support up to eight ranks per slot for multiple slots, even if there are only eight chip selects per channel.

  *LDIMM incorporate higher order channel addresses to perform rank multiplication, the amount of ranks per slot that can be supported depends on the number of unused addresses available.
Summary

• DRAM Introduction
• DRAM Evolutions
• Memory Interface
• Interface for graphics memory
  – GDDR3
    • Low power, low cost, low jitter / high performance
  – GDDR5
    • CDR for read (data training), external VPP, error correction, clamshell mode
• DDR4 preview
References

• Web sites and published data from Hynix, Samsung, Rambus, Elpida, Micron, AMD(ATi), Intel, nVidia, SONY, Nintendo, Microsoft, Pcwatch, JEDEC
References


Thank you
Appendix
SDRAM categorization

- by Speed / Applications
  - … / DDR1 / DDR2 / DDR3 / DDR4 / …
  - GDDR1 / GDDR2 / GDDR3 / GDDR4 / GDDR5 / GDDR5+ / GDDR6 …
  - mDDR / LPDDR2 / …

- by Density
  - … / 256Mb / 512Mb / 1Gb / 2Gb / 4 ~ 8Gb / …

- by Bus-Width
  - x4 / x8 / x16 / x32 / …
DRAM density & bus-width

- bus-width
  - # of data output pins
  - determined by applications
  - for examples,
    - PC : x64
    - Server : x64
    - Graphics card : x64 / x128 / x256 / x512 / …
    - Game consoles : x32 / x128 / …
**DRAM total density**

- DRAM device density X number of devices
- for servers
  - x4 configurations → to increase total memory capacity
  - x4 4Gb 16 devices can be used (64bit) : 4Gb X 16 = 8GB
- for laptops
  - x16 4Gb 4 devices can be used (64bit) : 4Gb X 4 = 2GB
- for PCs
  - x4 / x8 / x16 configurations
- for Graphics applications
  - x16 / x32 configurations
  - wide bus-width > total amounts of memory
# DRAM total density

## Conventional memory

<table>
<thead>
<tr>
<th>Component (bit)</th>
<th>Module (byte=x8)</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>1Gb</td>
<td>256M x4</td>
<td>256M x64</td>
</tr>
<tr>
<td>2Gb DDP</td>
<td>512M x4</td>
<td>1024M x64</td>
</tr>
<tr>
<td>1Gb</td>
<td>128M x8</td>
<td>128M x64</td>
</tr>
<tr>
<td>1Gb</td>
<td>64M x16</td>
<td>64M x64</td>
</tr>
</tbody>
</table>

## Graphics memory

<table>
<thead>
<tr>
<th>Component (bit)</th>
<th>Number</th>
<th>Bus-width</th>
<th>Total dens.</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>512Mb</td>
<td>16M x32</td>
<td>512bit/384bit</td>
<td>1GB/768MB</td>
<td>High-End</td>
</tr>
<tr>
<td>512Mb</td>
<td>16M x32</td>
<td>128bit (mirror)</td>
<td>512MB</td>
<td>XBOX 360</td>
</tr>
<tr>
<td>512Mb</td>
<td>16M x32</td>
<td>128bit</td>
<td>256MB</td>
<td>PS3</td>
</tr>
<tr>
<td>512Mb</td>
<td>16M x32</td>
<td>32bit</td>
<td>64MB</td>
<td>Nintendo Wii</td>
</tr>
</tbody>
</table>
Data bandwidth

• For example of GDDR3 on PS3
  – 700MHz/pin
  – \( \Rightarrow \) 1.4Gb/s/data channel(pin)
  – Each device has 32bit data I/O
  – \( \Rightarrow \) 1.4Gb/s \times 32 = 44.8Gb/s/component
  – 4 components configurations (32bit \times 4 = 128bit)
  – \( \Rightarrow \) 44.8Gb/s/component \times 4 = 179.2Gb/s
  – Data bandwidth is 22.4GB/s

• To increase data bandwidth
  – Clock speed
  – Wide I/O
  – More components (in other words, wide I/O in total)
Data bandwidth

• Increasing clock speed per pin
  – 700MHz → 1GHz
  – → 2.0Gb/s X 32 X 4 / 8 = 32GB/s
  – → ex) High-end graphics cards use 1.3GHz (2.6Gbps) [GDDR3]
  – → ex) 3.6 ~ 4.8Gbps [GDDR5] / up-to 7Gbps (@ES)

• Increasing I/O bits per component (wide I/O per component)
  – 32bit → 64bit
  – → 1.4Gb/s X 64 X 4 / 8 = 44.8GB/s
  – → 32bit is maximum in mass production
  – → x4 x 128 (x512) in TSV

• Increasing # of components (in other words, wide I/O in total)
  – 4 → 8
  – → 1.4Gb/s X 32 X 8 / 8 = 44.8GB/s
  – → to increase bus-width : ex) High-end graphics cards use 16 components (=512bit)
Mirror function

- To increase total density without increasing data bus-width

**Dual Rank**

- An example of 512Mb GDDR3

[Hynix]

[ISSCC '09]
XBOX 360

Xbox 360 CPU
(168mm² die)

ROP

eDRAM 10MB
(Rendering Memory)

Xbox 360 GPU
(ATI Xenos)
(170mm² die)

Shared Memory 512MB

GDDR3 512MB

GDDR3 512MB

GDDR3 512MB

GDDR3 512MB

GDDR3 512MB

GDDR3 512MB

GDDR3 512MB

GDDR3 512MB

GDDR3 512MB

Analog Chip

1GB/sec
500MB/sec W

128-bit
1.4Gbps/pm

1GB/sec

22.4GB/sec
R&W

FISB 21.00GB/sec
5.4Gbps/pm

Substrate

South Bridge

Network
USB
ETC

DVD-Drive

2.5inch HDD

Video

8 x 32b

128b

81-11-18 81

[Microsoft]
Prefetch operation

Conventional

Prefetch operation

<Simplified Block Diagram of 2n-prefetch READ>

<Simplified Block Diagram of 2n-prefetch WRITE>
DDR2/3 Architecture

- Multi bank
- Multi pre-fetch
- Cost effectiveness
- Speed bottleneck release
- Pad routing based on SI
- Power routing
- Core/Peri Bus routing
- Etc…conventional stuff
DDR2 block diagram
Simulation schematic

- $\text{Ron}_{-cs} = 30\,\text{ohm}$
- $\text{Ron}_{-dram} = 15\,\text{ohm}$
- $\text{Rs} = 20\,\text{ohm}$
- $V_{\text{term}} = 1.25\,\text{V}$
- $R_{\text{term}} = 36\,\text{ohm}$

- $L_{\text{breakout}} = 10\,\text{mm}$
- $\text{Leadin} = 63.5\,\text{mm}$
- $L_{\text{pitch}} = 10\,\text{mm}$
- $L_{\text{term}} = 5\,\text{mm}$
- $L_{\text{stub}} = 20\,\text{mm}$

Chipset schematic

Via schematic

Connector schematic

DRAM schematic

[Hynix]
Measurement vs. Simulation

<table>
<thead>
<tr>
<th></th>
<th>Meas.</th>
<th>Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rise Time</td>
<td>164 [ps]</td>
<td>160 [ps]</td>
</tr>
<tr>
<td>Fall Time</td>
<td>188 [ps]</td>
<td>182 [ps]</td>
</tr>
</tbody>
</table>

2011-11-18

[Hynix]
DRAM core speed path
Internal voltages

<table>
<thead>
<tr>
<th></th>
<th>WL-ON</th>
<th>WL-OFF</th>
<th>VBLP</th>
<th>BULK</th>
<th>V-cell</th>
<th>plate</th>
<th>I/O</th>
<th>Vperi</th>
<th>Vext vs Vint</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VCC vs VPP</td>
<td>GND vs VBB*</td>
<td>1/2VCC vs 1/3</td>
<td>GND vs VBB</td>
<td>VCC vs VCORE</td>
<td>Vss vs 1/2Vcore</td>
<td>Vref</td>
<td>Vext vs Vint</td>
<td>Interface, Reliability</td>
</tr>
<tr>
<td></td>
<td>Vt loss</td>
<td>Leak, new</td>
<td>Power, WL turn on, leak</td>
<td>Power, Reliability</td>
<td>Power, Reliability</td>
<td>Reliability</td>
<td>Interface</td>
<td>Power, Reliability</td>
<td></td>
</tr>
</tbody>
</table>
ZQ Cal

It’s ZQ Calibration for compensating PVT variation
Design trends

- **Package Level Approach**
  - MCP (Multi-Chip Package)
  - TSV (Through Silicon Via)
Receiver type comparison

Channel - P2P

Controller (RX) → PKG → VIA → PKG

Open-Drain
Current mode w/o source term.

Open-Drain
Voltage mode with source term.

Open-Drain
Current mode with source term.

Push-pull
Voltage mode

Poor

Good