Optimally Configuring DDR for Custom Boards

AMF-NET-T1021

Jacques Landry
FAE, Digital Networking - Freescale
Session Outline

AMF-NET-T1021
Optimally Configuring DDR for Custom Boards

• An overview of QorIQ processor's memory controller capabilities, configuration and testing for your board. Learn how to use QCS Configuration and DDRv tools to generate a customized configuration, run memory tests, and validate functionality on your board in a matter of hours. Will include a demo of these tools running memory tests on a QorIQ processor board.
Agenda

- Introduction and Industry Trends
- Memory Organization and Operation
- Features and Capabilities

- Overview and Demo of DDR Tools
  - DDR configuration using QorIQ Configuration Suite (QCS)
  - DDR validation using QorIQ Optimization Suite (QOS) DDRv plug-in to QCS
DDR SDRAM Memories
Introduction and Industry Trends
Introduction

• The current industry mainstream DRAM product is DDR3/3L. It is expected for this trend to continue till 2015 when the pricing cross-over is expected to occur.

• Almost all Freescale networking devices offer and support DDR3/3L.

• DDR4 has been introduced and DRAM vendors are expected to ramp production in 2014.

• The first Freescale device with DDR3L/DDR4 support is expected by end of 2013 (QorIQ T1040 family) followed by LS102x family products shortly after in Q1 2014.
DDR3 and DDR4 – Major Vendors

- Supported by all major memory vendors
## DRAM Migration Roadmap

<table>
<thead>
<tr>
<th>Year</th>
<th>DDR</th>
<th>DDR2</th>
<th>DDR3</th>
<th>DDR4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011</td>
<td>7%</td>
<td>23%</td>
<td>70%</td>
<td>0%</td>
</tr>
<tr>
<td>2012</td>
<td>5%</td>
<td>18%</td>
<td>75%</td>
<td>2%</td>
</tr>
<tr>
<td>2013</td>
<td>2%</td>
<td>13%</td>
<td>75%</td>
<td>10%</td>
</tr>
<tr>
<td>2014</td>
<td>1%</td>
<td>9%</td>
<td>70%</td>
<td>20%</td>
</tr>
<tr>
<td>2015</td>
<td>1%</td>
<td>7%</td>
<td>45%</td>
<td>47%</td>
</tr>
</tbody>
</table>
DDR3L vs DDR4
DRAM differences
## DDR SDRAM Highlights and Comparison

<table>
<thead>
<tr>
<th>Feature/Category</th>
<th>DDR3L</th>
<th>DDR4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package</td>
<td>BGA only</td>
<td>BGA only</td>
</tr>
<tr>
<td>Densities</td>
<td>512Mb - 8Gb</td>
<td>2Gb - 16Gb</td>
</tr>
<tr>
<td>Voltage</td>
<td>1.35V Core 1.35V I/O</td>
<td>1.2V Core 1.2V I/O</td>
</tr>
<tr>
<td>Data I/O</td>
<td>Center Tab Termination (CTT)</td>
<td>Pseudo Open Drain (POD) CTT</td>
</tr>
<tr>
<td>Data I/O CMD, ADDR I/O</td>
<td>CTT</td>
<td>CTT</td>
</tr>
<tr>
<td>Internal Memory Banks</td>
<td>8</td>
<td>16 for x4/x8 8 for x16</td>
</tr>
<tr>
<td>Data Rate</td>
<td>800–1866 Mbps</td>
<td>1600–3200 Mbps</td>
</tr>
<tr>
<td>VREF</td>
<td>VREFCA &amp; VREFDQ external</td>
<td>VREFCA external VREFDQ internal</td>
</tr>
<tr>
<td>Data Strobes/Prefetch/Burst Length/Burst Type</td>
<td>Differential/8-bits/BC4, BL8/Fixed, OTF</td>
<td>Same as DDR3L</td>
</tr>
<tr>
<td>Additive/read/write Latency</td>
<td>0, CL-1, CL-2/ AL+CL/ AL +CWL</td>
<td>Same as DDR3L</td>
</tr>
</tbody>
</table>
### DDR SDRAM Highlights and Comparison (continued)

<table>
<thead>
<tr>
<th>Feature/Category</th>
<th>DDR3L</th>
<th>DDR4</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRC Data Bus</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Boundary Scan/Connectivity test (TEN pin)</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Bank Grouping</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Data Bus Inversion (DBI_n pin)</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Write Leveling / ZQ</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ACT_n new pin &amp; command</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Low power Auto self-refresh</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
DDR3/DDR3L/DDR4

Power Saving

- DDR3 DRAM provides 25% power savings over DDR2

- DDR3L DRAM provides 20% to 27% power saving over DDR3

- DDR4 DRAM provides 37% power saving over DDR3L
**DDR3L vs. DDR4 DRAM pinouts**

- **DDR4 Pins added**
  - VDDQ (2) : 1.2V pins to DRAM
  - VPP : 2.5V external voltage source for DRAM internal word line driver
  - Bank Group (2): pins to identify the bank groups
  - DBI_n: Data Bus Inversion
  - ACT_n: Active command
  - PAR: Parity error signal for address bus
  - Alert_n: Both, Parity error on C\A and CRC error on data bus
  - TEN: Connectivity test mode

- **DDR3 Pins eliminated**
  - VREFDQ
  - Bank Address (1): one less BA pin
  - VDD (1), VSS (3), VSSQ (1)
Basic DDR SDRAM Structure
Memory Organization and Operation
Single Transistor Memory Cell

- **Access Transistor**
- **Row (word) line**
- **Column (bit) line**
- **Vcc/2**
- **Storage Capacitor**
- **Parasitic Line Capacitance**

- "1" => Vcc
- "0" => Gnd

- "precharged" to Vcc/2
Memory Arrays

ROW ADDRESS DECODER

SENSE AMPS & WRITE DRIVERS

Row Buffer

COLUMN ADDRESS DECODER
Internal Memory Banks

- Multiple arrays organized into banks
- Multiple banks per memory device
  - DDR3 – 8 banks, and 3 bank address (BA) bits
  - DDR4 - 16 banks with 4 banks in each of 4 sub bank groups
  - Can have one active row in each bank at any given time
- Concurrency
  - Can be opening or closing a row in one bank while accessing another bank

<table>
<thead>
<tr>
<th>Bank 0</th>
<th>Bank 1</th>
<th>Bank 2</th>
<th>Bank 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Row 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Row 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Row 3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Row …</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Row Buffers</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Memory Access

- A requested row is **ACTIVATED** and made accessible through the bank’s row buffer.

- **READ** and/or **WRITE** are issued to the active row.

- The row is **PRECHARGED** and is no longer accessible through the bank’s row buffer.
DDR2-533 Read Timing Example

- **Trcd (ACTTORW)** = 4 clk
- **Tck** = 3.75 ns
- **Tccd** = 2 clk
- **Trtp (RD_TO_PRE)** = 2 clk
- **Trp (PRETOACT)** = 4 clk
- **CASLAT** = 4 clk

**Mem Clk**

**/CS**

**/RAS**

**/CAS**

**/WE**

**Address**

**DQS**

**DQ**
Example – DDR2/3 SDRAM

- Micron MT47H32M8
- 32M x 8 (8M x 8 x 4 banks)
- 256 Mb total
- 13-bit row address
  - 8K rows
- 10-bit column address
  - 1K bits/row (8K total when you take into account the x8 width)
- 2-bit bank address
- Data bus: DQ, DQS, /DQS, DM
- ADD bus: A, BA, /CS, /RAS, /CAS, /WE, ODT, CKE, CK, /CK
Example – DDR2/3 DIMM

- Micron MT9HTF3272A
- 9 each 32M x 8 memory devices
- 32M x 72 overall
- 256 MB total, single “rank”
- 9 “byte lanes”

Two Signal Bus

- 1- Address, command, control, and clock signals are shared among all 9 DRAM devices
- 2- Data, strobe, data mask not shared
DRAM Module Type

**UDIMM**: Unbuffered Desktop standard

**SODIMM**: Notebook standard

**MiniDIMM**: Computing and Networking

**VLP MiniDIMM**: Computing and Networking

**RDIMM**: Registered Server standard

**VLP RDIMM**: Very Low Profile Computing and Networking
Fly-By Routing Topology

- **Introduction of “fly-by” architecture**
  - Address, command, control & clocks
  - Data bus (not illustrated below) remains unchanged, i.e., direct 1-to-1 connection between the Controller bus lanes and the individual DDR devices.
  - Improved signal integrity...enabling higher speeds
  - On module termination
Fly-By Routing Improved SI

DDR2 Matched Tree Routing

DDR3 Fly By Routing
What Is Write Leveling?

- During a write cycle, the skew between the clock and strobes is 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.
What Is Write Leveling?

Source @ Mem CTRL

Destination 0

Destination 1

Destination 2

... 

Destination 8

Write Leveling for lane 0

Write Leveling for lane 1

Write Leveling for lane 2

Write Leveling for lane 8
Write Adjustment

- Write leveling sequence during the initialization process will determine the appropriate delays to each strobe/data byte lane and add this delay for every write cycle.

- Write leveling used to add delay to each strobe/data line.
Read Adjustment

- Instead of JEDEC’s MPR method, Freescale controllers use a proprietary method of read adjust method. Auto CPO will provide the expected arrival time of preamble for each strobe line of each byte lane during the read cycle to adjust for the delays caused by the fly-by topology.
  - Automatic CAS to preamble calibration
  - Data strobe to data skew adjustment

![Diagram of Freescale Chip and Data Lanes]
CLK_ADJ - Clock Adjust

- CLK_ADJ defines the timing of the address and command signals relative to the DDR clock.

![Diagram showing CLK_ADJ settings](image-url)
Power-up

DDR Reset

Asserted at least 200us

Chip selects enabled and DDR clocks begin

DRAMs Initialized

ZQ Calibration

ZQCL Issued (512 clocks)
Also DLL lock time is occurring

Write Leveling

Automatically handled
By the controller

Read Adjust

Automatic CAS-to-Preamble
(aka Read Leveling)....
Plus Data-to-Strobe adjustment

Init Complete

Ready for User accesses

Controller Started

Mem_EN = 1

Stable CLKS

CKE = HIGH

DDR CTRL INIT

Need at least 500us from reset de-assertion to the controller being enabled.

Timed loop may be needed.
Register Configuration

- Two general type of registers to be configured in the memory controller
- First register type is set to the DRAM related parameter values that are provided via SPD or DRAM datasheet
- Second register type is the non-SPD values that are set based on customer’s application. For example:
  - On-die-termination (ODT) settings for DRAM and controller
  - Driver impedance setting for DRAM and controller
  - Clock adjust, write data delay, Cast to preamble override (CPO)
  - 2T or 3T timing
  - Burst type selection (fixed or on-the-fly burst chop mode)
  - Write-leveling start value (WRLVL_START)
- Freescale’s Processor Expert QorIQ Configuration Suite includes a DDR configuration tool for many devices. For other devices, Freescale support resources can help generate or analyze DDR settings.
QorIQ Layerscape Family DDR Controllers
Features and Capabilities DDR3L and DDR4
Common DDR3L & DDR4 Controller Features

- Supports most JEDEC standard x8, x16 DDR3L & DDR4 devices
- Memory device densities from 1Gb – through 8Gb
- Data rates up to: 1600 MT/s DDR3L and DDR4
- Devices with 12-16 row address bits, 8-11 column address bits, 2-3 logical bank address bits
- Data mask signals for sub-doubleword writes
- Up to four physical banks (ranks / chip selects)
- Physical bank (rank) sizes up to 8GB, total memory up to 32GB per controller
- Physical bank interleaving between 2 or 4 chip selects
- Memory controller interleaving when more than 2 controllers are available
- Un-buffered or registered DIMMs
Common DDR3L /4 Controller Features (continued)

• Up to 32 open pages (DDR3L only), 64 open pages for DDR4
  - Open row table
  - Amount of time rows stay open is programmable
• Auto-precharge, globally or by chip select
• Self-refresh
• Up to 8 posted refreshes
• Automatic or software-controlled memory device initialization
• ECC: 1-bit error correction, 2-bit error detection, detection of all errors within a nibble
• ECC error injection
• Read-modify-write for sub-doubleword writes when using ECC
• Automatic data initialization for ECC
• Dynamic power management
Common DDR3L /4 Controller Features (continued)

- Partial array self refresh
- Address and command parity for Registered DIMM (DDR3 only)
- Independent driver impedance setting for data, address/command, and clock
- Synchronous and Asynchronous clock-in option
- Write-leveling
- Automatic CPO
- Asynchronous RESET
- Automatic ZQ calibration
- Mirrored DIMM supported
DDR4 only Controller Features

- Internal DQa Vref supply & calibration, both controller & DRAM
- Data write CRC (not available in LS1)
- Data Inversion bus
- Address bus parity error
- 16 banks for more concurrency
- Connectivity test mode
- ODT park and buffer disable
- DRAM mode register readout capability
- Low power auto self refresh
- Pseudo open drain (POD) driver and termination
- Command Address latency (CAL)
DDDR4 Output Driver / Termination

- Center tap termination is used in DDR3 receiver
- POD termination or pull up is used in DDR4 receiver
- Push-Pull driver in DDR3 and POD driver in DDR4
- Less power is consumed using POD driver & termination.
Multiport Muxing in Layerscape LS102x memory controller

- DDR4 support up to 16Gb vs. 8Gb in DDR3
- DDR4 uses A0-A13 for column accesses (i.e. MA[14] & MA[15] not used for column access)
- DDR4 has 4 banks within each group (i.e. MBA[2] not used)

<table>
<thead>
<tr>
<th>DDR3</th>
<th>DDR4</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRAS</td>
<td>MRAS/ MA[16]</td>
</tr>
<tr>
<td>MCAS</td>
<td>MCAS/ MA[15]</td>
</tr>
<tr>
<td>MWE</td>
<td>MWE/ MA[14]</td>
</tr>
<tr>
<td>MA[15]</td>
<td>ACT_n</td>
</tr>
<tr>
<td>MA[14]</td>
<td>BG1</td>
</tr>
<tr>
<td>MBA[2]</td>
<td>BG0</td>
</tr>
<tr>
<td>MDM[0-8]</td>
<td>MDM / DBI</td>
</tr>
<tr>
<td>MAPAR_ERR</td>
<td>Alert_n</td>
</tr>
<tr>
<td>MAPAR_OUT</td>
<td>PAR</td>
</tr>
</tbody>
</table>
New pin: ACT\textsubscript{n}

- ACT\textsubscript{n} is a single pin for Active command input

- When ACT\textsubscript{n} is low:
  - ACT Command is asserted
  - WE/CAS/RAS pins will be treated as address pins (A14:A16)

- When ACT\textsubscript{n} is high
  - WE/CAS/RAS pins will be treated as command pins
New pin: DBI_n

- Active low input/output for data bus inversion mode
- As an input to DRAM, a low on DBI_n indicates that the DRAM inverts write data received on the DQ inputs.
- As an output from the DRAM, a low on DBI_n indicates that the DRAM has inverted the data on its DQ outputs.
- Maximum of half of the bits driven low including DBI_n pin
- Available only on x8 and x16 DRAM
- Fewer bits driven low means less noise, better data eye and lower power consumption.
Data Bus Inversion - DBI

- If more than 4-bits of a byte lane are low, invert the data and drive the DBI_n pin low
- If 4 or less bits of a byte lane are low, do not invert the data and drive the DBI_n pin high

- **Controller**
  - DQ0: 0 1 0 0
  - DQ1: 1 1 0 0
  - DQ2: 0 0 0 0
  - DQ3: 0 1 1 0
  - DQ4: 0 1 0 0
  - DQ5: 1 0 1 0
  - DQ6: 1 1 1 0
  - DQ7: 0 0 1 0
  - DBI_n: 0 1 1 0
  - # low bits: 5 3 4 8

- **Data Bus**
  - 0 1 0 0
  - 0 0 0 0
  - 0 0 1 0
  - 0 1 1 0
  - 1 0 1 1
  - 1 1 0 1
  - 1 1 1 1
  - 1 1 0 1
  - # low bits: 4 3 4 1

- **Memory**
  - 0 1 0 0
  - 1 1 0 0
  - 0 0 0 0
  - 0 1 1 0
  - 1 0 1 0
  - 1 1 1 0
  - 0 0 1 0
  - # low bits: 5 3 4 8
New pin: B\(\text{Gn Bank Group Address}\)

- Different timing within a group and between groups
  - Active to active
  - Write to read
  - CAS to CAS
- Controller to maintain

Timing requirements for both
Within a group (Long) and
Between groups (short)
New pin: Parity

- C/A Parity signal (PAR) covers ACT_n, RAS_n, CAS_n, WE_n and the address bus. Control signals CKE, ODT, CS_n are not included.

- Even parity, i.e. valid parity is defined as an even number of ones across the inputs used for parity computation combined with the parity signal. The parity bit is chosen so that the total number of ‘1’s in the transmitted signal, including the parity bit is even.

- Commands must be qualified by CS_n.

- Alert_n used to flag error to memory controller.
Data Write CRC

- Example data mapping with CRC for 8-bit, 4-bit and 16-bit devices
- Note: not the same as ECC
Data Write CRC (continued)

A x16 device is treated as two x8 devices. x16 device will have two identical CRC trees implemented. CRC(0-7) covers data bits d(0-71). CRC(8-15) covers data bits d(72-143).

<table>
<thead>
<tr>
<th>DQ0</th>
<th>DQ1</th>
<th>DQ2</th>
<th>DQ3</th>
<th>DQ4</th>
<th>DQ5</th>
<th>DQ6</th>
<th>DQ7</th>
<th>DBIL_n</th>
<th>DQ8</th>
<th>DQ9</th>
<th>DQ10</th>
<th>DQ11</th>
<th>DQ12</th>
<th>DQ13</th>
<th>DQ14</th>
<th>DQ15</th>
<th>DBIU_n</th>
</tr>
</thead>
<tbody>
<tr>
<td>d0</td>
<td>d1</td>
<td>d2</td>
<td>d3</td>
<td>d4</td>
<td>d5</td>
<td>d6</td>
<td>d7</td>
<td>CRC0</td>
<td>d8</td>
<td>d9</td>
<td>d10</td>
<td>d11</td>
<td>d12</td>
<td>d13</td>
<td>d14</td>
<td>d15</td>
<td></td>
</tr>
<tr>
<td>d16</td>
<td>d17</td>
<td>d18</td>
<td>d19</td>
<td>d20</td>
<td>d21</td>
<td>d22</td>
<td>d23</td>
<td>CRC1</td>
<td>d32</td>
<td>d33</td>
<td>d34</td>
<td>d35</td>
<td>d36</td>
<td>d37</td>
<td>d38</td>
<td>d39</td>
<td></td>
</tr>
<tr>
<td>d40</td>
<td>d41</td>
<td>d42</td>
<td>d43</td>
<td>d44</td>
<td>d45</td>
<td>d46</td>
<td>d47</td>
<td>CRC2</td>
<td>d48</td>
<td>d49</td>
<td>d50</td>
<td>d51</td>
<td>d52</td>
<td>d53</td>
<td>d54</td>
<td>d55</td>
<td></td>
</tr>
<tr>
<td>d56</td>
<td>d57</td>
<td>d58</td>
<td>d59</td>
<td>d60</td>
<td>d61</td>
<td>d62</td>
<td>d63</td>
<td>CRC3</td>
<td>d64</td>
<td>d65</td>
<td>d66</td>
<td>d67</td>
<td>d68</td>
<td>d69</td>
<td>d70</td>
<td>d71</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CRC4</td>
<td></td>
<td></td>
<td>CRC5</td>
<td></td>
<td>CRC6</td>
<td></td>
<td>CRC7</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CRC8</td>
<td></td>
<td></td>
<td>CRC9</td>
<td></td>
<td>CRC10</td>
<td></td>
<td>CRC11</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CRC12</td>
<td></td>
<td></td>
<td>CRC13</td>
<td></td>
<td>CRC14</td>
<td></td>
<td>CRC15</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CRC16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CRC values are calculated as follows:

- CRC0: d0, d1, d2, d3, d4, d5, d6, d7
- CRC1: d8, d9, d10, d11, d12, d13, d14, d15
- CRC2: d16, d17, d18, d19, d20, d21, d22, d23
- CRC3: d24, d25, d26, d27, d28, d29, d30, d31
- CRC4: d32, d33, d34, d35, d36, d37, d38, d39
- CRC5: d40, d41, d42, d43, d44, d45, d46, d47
- CRC6: d48, d49, d50, d51, d52, d53, d54, d55
- CRC7: d56, d57, d58, d59, d60, d61, d62, d63
- CRC8: d64, d65, d66, d67, d68, d69, d70, d71

**CRC Calculation Formula:**

The CRC calculation for each input data bit is performed using the polynomial:

\[ x^8 + x^7 + x^6 + x^4 + x^2 + 1 \]

The CRC values are generated by feeding the input data bits through this polynomial and performing modulo-2 division. The resulting remainder is the CRC value.

**Example:**

- For CRC0: d0, d1, d2, d3, d4, d5, d6, d7
  - CRC0 = \( x^7 + x^5 + x^4 + x^2 + 1 \) evaluated at each data bit
  - Result: CRC0 = \( d0 \times x^7 + d1 \times x^6 + d2 \times x^5 + d3 \times x^4 + d4 \times x^3 + d5 \times x^2 + d6 \times x^1 + d7 \times x^0 \)

The CRC values are compared against the expected values to ensure data integrity.
New pin Alert_n

- Alert_n – Active low output signal that indicates an error event for both the C/A Parity Mode and the CRC Data Mode.
- CRC Data mode. Not ECC. The DRAM device generates a checksum per byte lane for both READ and WRITE data and returns the checksum to the controller. Based on the checksum, the controller can decide if the data or the returned CRC was transmitted in error and take appropriate measures, details TBD.
Low power auto self-refresh

• While DRAM is in self-refresh mode, four refresh mode options available:
  − Manual mode, normal temperature (0 – 85C)
  − Manual mode, extended temperature (0 – 95C)
  − Manual mode, reduced temperature (0 – 45C)
  − Automatic mode: automatically switches between modes based on temperature sensor measurements

• Power savings by reducing refresh rate when possible
Command Address Latency (CAL)

- DDR4 supports Command Address Latency, CAL, function as a power savings feature. CAL is the delay in clock cycles between CS_n and CMD/ADDR. CAL gives the DRAM time to enable the CMD/ADDR receivers before a command is issued. Once the command and the address are latched the receivers can be disabled.

![Diagram of Command Address Latency](image-url)

**Figure 34 — Definition of CAL**
**DRAM Receiver data eye mask and BER**

- Bit error rate (similar to serdes) is defined for DRAM receiver measurement.
- DRAM receiver data mask is defined for random and deterministic Jitter as data rates approaching 3GT/s.
- For LS1 (i.e. data rates of 1600MT/s or less) we will continue with the conventional setup and hold time measurements.
Summary

• DDR3/3L is mainstream now
• DDR4 is expected to start gaining market share by 2014
• Next generation QorIQ Layerscape and QorIQ T Series devices families support DDR3L & DDR4
• DDR4 low power consumption is suitable for next generation devices
• Follow JEDEC recommended topologies for discrete parts
• Using QCS and DDRv tool, configuration and initialization of memory controller can be easily achieved
Useful References

- **Books:**

- **Freescale Application Notes:**
  - AN2582 Hardware and Layout Design Considerations for DDR Memory Interfaces
  - AN2910 Hardware and Layout Design Considerations for DDR2 Memory Interfaces
  - AN2583 Programming the PowerQUICCIII / PowerQUICCII Pro DDR SDRAM Controller
  - AN3369 PowerQUICC DDR2 SDRAM Controller Register Setting Considerations
  - AN3939 PQ & QorIQ Interleaving
  - AN3940 Layout Design Considerations for DDR3 Memory Interface
  - AN4039 PowerQUICC DDR3 SDRAM Controller Register Setting Considerations

- **Micron Application Notes:**
  - TN-46-05 General DDR SDRAM Functionality
  - TN-47-02 DDR2 Offers New Features and Functionality
  - TN-47-01 DDR2 Design Guide
  - TN-41-07 DDR3 Power-Up, Initialization, and Reset
  - TN-41-08 DDR3 Design Guide

- **JEDEC Specifications:**
  - JESD79E Double Data Rate (DDR) SDRAM Specification
  - JESD79-2F DDR2 SDRAM Specification
  - JESD79-3D DDR3 SDRAM Specification

- **Tools**
  - QorIQ Configuration Suite
  - QorIQ Optimization Suite
DDR Configuration and Validation Tools
QorIQ Configuration Suite – Now Available!

• **QorIQ Configuration Suite v3.0 is NOW AVAILABLE!!!**
  - Supports all QorIQ and Qorivva devices
  - Works with Eclipse 3.5, Eclipse 3.6, Eclipse 3.7 development tools
    - Pure Java solution for maximum choice of host system support
    - Add-in to CodeWarrior Development Studio for PA, v10.1 or later
  - Available from [www.freescale.com/QCS](http://www.freescale.com/QCS) – FREE DOWNLOAD*

• **Includes the following configuration tools all designed to collaborate on consistent configuration:**
  - PBL tool to define the Reset Control Word bit values and PBI data for the pre-boot
  - BOOTROM generator for those QorIQ without RCW functionality
  - DDR configuration supports setting the controller to a working state for any DDR
  - Data path graphical view helps to define data path configuration for the DPAA.
  - Hardware Device Tree editor supports references, synchronous GUI and XML editing, node validation based on specification bindings
  - Packaged as a separate product with installer and wizard functionality

* Must be a QorIQ customer or under QorIQ NDA for download permission

Installing Processor Expert for QorIQ

- You need CodeWarrior for PA 10.1 or later

  **OR**, you download an Eclipse version for free

  **OR**, you use an existing Eclipse workbench you have installed (Wind River, QNX, GNU, etc.)

- Processor Expert for QorIQ Configuration Suite installs using the Eclipse updater’s “Add new software…” capability

- The Configuration Suite is 100% pure Java so it should run on any Eclipse 3.6.1 or later host environment (Windows, Linux, Solaris, Mac OS, 32-bit/64-bit, …)
DDR Configuration
Creating New DDR Configuration
Create New QCS project

[Image of Eclipse IDE showing the process of creating a new QorIQ Configuration Project]
Select device and DDR component

1. Choose the SoC you would like to use:
   - P2010
   - P2020
   - P2040
   - P2041

2. Choose what you want to configure:
   - Components to be selected:
     - Components
       - BootROM Configuration
       - DDR Memory Controller Configuration
       - Device Tree Editor

The DDR Memory Controller configuration tool supports specific settings for a custom DDR based on the manufacturer's data sheet and includes optional clock, bus, and DMA settings.
Get DRAM information – P2020RDB-PCA

From back of RDB box

From DRAM datasheet

P2020RDB-PCA
CPU: P2020NSE2MHC 1200MHz
DDR3: MICRON MT41J128M16HA-15ED 1333MHz

DDR3 SDRAM
MT41J512M4 – 64 Meg x 4 x 8 Banks
MT41J256M8 – 32 Meg x 8 x 8 Banks
MT41J128M16 – 16 Meg x 16 x 8 Banks
How about rest of the timing parameters?

- Tool automatically computes tRCD, tRP, and CL!
- User can change these values if required.
• From memory data sheet:
  - Maximum speed rating
  - Capacity
QCS project explorer
Review DDR registers values
Review DDR registers values – contd.

<table>
<thead>
<tr>
<th>Reg. name</th>
<th>Init. value</th>
<th>After reset</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDR1_CS0 BNDS</td>
<td>0000003F</td>
<td>00000000</td>
</tr>
<tr>
<td>DDR1_CS1 BNDS</td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>DDR1_CS2 BNDS</td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>DDR1_CS3 BNDS</td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>DDR1_CS0.getConfig</td>
<td>80014202</td>
<td>00000000</td>
</tr>
<tr>
<td>DDR1_CS1getConfig</td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>DDR1_CS2getConfig</td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>DDR1_CS3getConfig</td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>DDR1_TIMING_CFG_3</td>
<td>00030000</td>
<td>00000000</td>
</tr>
<tr>
<td>DDR1_TIMING_CFG_0</td>
<td>00330104</td>
<td>00111015</td>
</tr>
<tr>
<td>DDR1_TIMING_CFG_1</td>
<td>66668846</td>
<td>00000000</td>
</tr>
<tr>
<td>DDR1_TIMING_CFG_2</td>
<td>0FA8D0CC</td>
<td>00000000</td>
</tr>
<tr>
<td>DDR1_SDRAM_CFG</td>
<td>47000008</td>
<td>03000000</td>
</tr>
<tr>
<td>DDR1_SDRAM_CFG_2</td>
<td>24401050</td>
<td>00000000</td>
</tr>
<tr>
<td>DDR1_SDRAM_MODE</td>
<td>00061421</td>
<td>00000000</td>
</tr>
<tr>
<td>DDR1_SDRAM_MODE_2</td>
<td>00000000</td>
<td>00000000</td>
</tr>
</tbody>
</table>
Generate DDR configuration
# DDR Controller 1 Registers

# DDR_SDRAM_CFG
mem [0xFF702110] = 0x47000006

# CS0_BNDS
mem [0xFF702000] = 0x3F

# CS0_CONFIG
mem [0xFF702080] = 0x80014202

# CS0_CONFIG_2
mem [0xFF7020C0] = 0x00

# TIMING_CFG_3
mem [0xFF702100] = 0x00030000

# TIMING_CFG_0
mem [0xFF702104] = 0x00330104

# TIMING_CFG_1
mem [0xFF702108] = 0x666B8846

# TIMING_CFG_2
mem [0xFF70210C] = 0x0FA8D0CC

# DDR_SDRAM_CFG_2
mem [0xFF702114] = 0x24401050

# DDR_SDRAM_MODE
mem [0xFF702118] = 0x00061421

#define PEX_CONFIG_DDR1_INIT_EXT_ADDR 0x00000000
#define PEX_CONFIG_DDR1_TIMING_4 0x00220001
#define PEX_CONFIG_DDR1_TIMING_5 0x02401400
#define PEX_CONFIG_DDR1_2Q_CNTL 0x59300600
#define PEX_CONFIG_DDR1_WRLVL_CNTL 0x653F014
#define PEX_CONFIG_DDR1_RCW_1 0x00000000
#define PEX_CONFIG_DDR1_RCW_2 0x00000000

/* DDR Controller 1 configuration global structures */
s1_ddr_cfg_regcs_t ddr_cfg_regcs_0 = {
  .cs[0].bnds = PEX_CONFIG_DDR1_CS0_BNDS,
  .cs[1].bnds = PEX_CONFIG_DDR1_CS1_BNDS,
  .cs[2].bnds = PEX_CONFIG_DDR1_CS2_BNDS,
  .cs[3].bnds = PEX_CONFIG_DDR1_CS3_BNDS,
  .cs[0].config = PEX_CONFIG_DDR1_CS0_CONFIG,
  .cs[1].config = PEX_CONFIG_DDR1_CS1_CONFIG,
  .cs[2].config = PEX_CONFIG_DDR1_CS2_CONFIG,
Steps to adapt DDR configuration file in CodeWarrior

• Open the CW config file you want to adapt

D:\Program Files\Freescale\CW PA v10.1\PA\PA_Support\Initialization_Files\QorIQ_P4\P4080DS_init_core0.cfg

• Replace DDR1 config section with the one from

D:\Profiles\b08844\workspace\p4080\Generated_Code\ddrCtrl_1.cfg

• Use this new config file with your stationary project
DDR Validation Tool
License file:
<QCS Install directory>/eclipse/Optimization/license.dat
DDRv Connection Setup

Choose validation mode:
- In depth

Connection settings:
- System: P2020; USBTAP id:
- Hardware configurations
DDRv Basic Connection Test

- Run basic test to confirm target connection
Configure DDR scenarios and tests

1. Validation
   - Centering the clock

2. Choose tests
   - BIST Write Read Compare: 0
   - Read Write Compare: 3
   - Walking Ones: 0
   - Walking Zeros: 0

3. Start Validation
Centering of the clock results

- Click “cell” to choose Write level start and CLK_ADJ values.
• Click “cell” to choose optimized ODT value.
• Click “cell” to choose optimized ODT value.
Centering of the clock - after ODT optimization

- Centering of clock scenario was re-run after finding the right ODT values

<table>
<thead>
<tr>
<th>WRVL_START</th>
<th>CLK_ADJ</th>
<th>0 clocks</th>
<th>1/8 clocks</th>
<th>1/4 clocks</th>
<th>3/8 clocks</th>
<th>1/2 clocks</th>
<th>5/8 clocks</th>
<th>3/4 clocks</th>
<th>7/8 clocks</th>
<th>1 clocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 clock delay</td>
<td>0/3</td>
<td>3/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
</tr>
<tr>
<td>1/8 clock delay</td>
<td>0/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
</tr>
<tr>
<td>1/4 clock delay</td>
<td>0/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
</tr>
<tr>
<td>1/2 clock delay</td>
<td>0/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>0/3</td>
</tr>
<tr>
<td>5/8 clock delay</td>
<td>0/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>0/3</td>
</tr>
<tr>
<td>3/4 clock delay</td>
<td>0/3</td>
<td>0/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>0/3</td>
</tr>
<tr>
<td>7/8 clock delay</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>2/3</td>
<td>3/3</td>
<td>0/3</td>
</tr>
<tr>
<td>1 clock delay</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>0/3</td>
</tr>
<tr>
<td>9/8 clock delay</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>0/3</td>
</tr>
<tr>
<td>5/4 clock delay</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>0/3</td>
</tr>
<tr>
<td>11/8 clock delay</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
</tr>
<tr>
<td>3/2 clock delay</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
</tr>
<tr>
<td>13/8 clock delay</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
</tr>
<tr>
<td>7/4 clock delay</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
</tr>
<tr>
<td>15/8 clock delay</td>
<td>0/3</td>
<td>3/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
</tr>
<tr>
<td>2 clock delay</td>
<td>0/3</td>
<td>3/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
</tr>
<tr>
<td>17/8 clock delay</td>
<td>0/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
</tr>
<tr>
<td>9/4 clock delay</td>
<td>0/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
</tr>
<tr>
<td>19/8 clock delay</td>
<td>0/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
</tr>
<tr>
<td>5/2 clock delay</td>
<td>0/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
</tr>
</tbody>
</table>
Generate optimized DDR configuration

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
<td>DDR_Controller_1</td>
<td>DDR_Controller_1</td>
</tr>
<tr>
<td>Memory type</td>
<td>DDR 3</td>
<td></td>
</tr>
<tr>
<td>DDR Bus Clock</td>
<td>400 MHz</td>
<td>DDR Data Rate: 800 MT/s</td>
</tr>
<tr>
<td>Type of DIMM</td>
<td>Unbuffered DIMMs</td>
<td></td>
</tr>
<tr>
<td>Bus mode</td>
<td>64-bit bus</td>
<td></td>
</tr>
</tbody>
</table>

SDRAM Control Configuration
- Control Configuration 1
- Control Configuration 2

SDRAM Timing Configurations
- Auto-adjust chip select address: yes

Chip Select 0
- Memory Bounds
  - Start Address: 0
  - Size: 1 GB
- Configuration
  - Auto Precharge Always: no
  - Internal Banks Number: 8 internal banks
  - Number of row bits: 14 row bits
  - Number of column bits: 10 column bits
  - ODT for writes configuration: Assert ODT only during writes to C...
  - ODT for reads configuration: Never assert ODT for reads
  - Partial array self refresh: Full Array

Chip Select 1: Disabled
Chip Select 2: Disabled
Chip Select 3: Disabled

Pricing $995
License file:
<QCS Install directory>/eclipse/Optimization/license.dat
2020RDB-PCA: Import DDR configuration from existing system running uboot

- At uboot prompt

  => md ffe02000
  - ffe02000: 0000003f 00000000 00000000 00000000 ....?........
  - ffe02080: 08014202 00000000 00000000 00000000 .B........
  - ffe02100: 00030000 00110104 616b8846 0fa8c8cc ....Ok.F....
  - ffe02110: c7000008 24401040 00441421 00000000 ...$@.@.D!....
  - ffe02120: 00000000 0c300100 deadbeef 00000000 .....O........
  - ffe02130: 03000000 00000000 00000000 00000000 ................
  - ffe02160: 00220001 02401421 00000000 00000000 .'..@.........
  - ffe02170: 89080600 8675f608 00000000 00000000 .....U........

  => md ffe02b00
  - ffe02b00: 00000000 00000000 00000000 00000000 ................
  - ffe02b10: 00000000 00000000 00000000 00000000 ................
  - ffe02b20: 5dc07777 77000000 00000000 00000000 ].www........

- Save content to a file.
Processor Expert for QorIQ … For More Info

• Freescale’s Processor Expert landing page
  - http://www.freescale.com/ProcessorExpert

• QorIQ Configuration Suite
  - http://www.freescale.com/QCS

• QorIQ Optimization Suite
  - http://www.freescale.com/QOS

• Freescale Component Store – purchasing embedded software
  - http://www.freescale.com/webapp/sps/site/homepage.jsp?code=BEAN_STORE_MAIN&tid=SWnT
Pricing & Availability

• Part numbers: CWA-QIQ-OPTP-FL (floating license) & CWA-QIQ-OPTP-NL (node locked)
• Price: $999 Annual Subscription
• License Duration: 1 year
• Support & Maintenance: Included
• Availability
  – Scenarios Tool – Now
  – DDRv – Now
AMERICAS | APRIL 8-11, 2014
Gaylord Texan Resort & Convention Center | Dallas

Come to FTF for the training and collaboration, leave with the knowledge and inspiration to make the world a smarter place.

Registration opens December 2, 2013

more info at www.freescale.com/FTF