



## APPLICATION NOTE

### ARMADA 16X Memory Controller Configuration and Tuning

#### 1. Overview

The ARMADA 16x Memory Controller Unit (MCU) supports LPDDR1, DDR2, and LVDDR2. It does not support LPDDR2. It is critical that the MCU be programmed correctly to ensure system stability.

#### 2. Programming Sequence.

The *ARMADA 16x Applications Processor Family Software Manual* (MV-S301544-00) lists the order in which the MCU registers must be programmed. The Marvell OEM Boot Module (OBM) contains sample code that sets up the MCU.

#### 3. NTIM and OBM

On bootup, the Boot ROM loads the OBM from storage memory to internal SRAM and hands control to it. The OBM reads content of the Non-Trusted Image Module (NTIM) and searches for a DDR configuration package. If one exists, the OBM uses the information in the NTIM to configure the MCU. The NTIM can be customized for specific DDR devices by changing the register values. The sequence in which these registers are arranged in the NTIM description file is the same sequence in which these registers are programmed by OBM. The following is an example section of the NTIM Descriptor file that relates to DDR configuration.

**Figure 1: Section of NTIM Description File Concerning DDR**

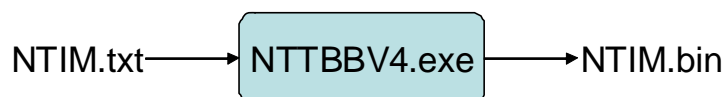
DDR Custom:	
ASPEN_SDRCTLREG13_ID: 0x0000000F	Configuration
ASPEN_ADRMAPREG0_ID: 0x000C0001	
ASPEN_SDRCFGREG0_ID: 0x00042430	
ASPEN_SDRTMGREG1_ID: 0x511D00C8	Timing
ASPEN_SDRTMGREG2_ID: 0x74780442	
ASPEN_SDRTMGREG3_ID: 0x20C84A53	
ASPEN_SDRTMGREG4_ID: 0x36E8D8D6	
ASPEN_SDRTMGREG5_ID: 0x00160131	
ASPEN_SDRCTLREG1_ID: 0x00000000	SDRAM Control
ASPEN_SDRCTLREG2_ID: 0x00080000	
ASPEN_SDRCTLREG3_ID: 0xc0000000	
ASPEN_SDRCTLREG4_ID: 0x2081C004	
ASPEN_SDRCTLREG6_ID: 0x00000001	
ASPEN_SDRCTLREG7_ID: 0x02000001	
ASPEN_PHYCTLREG11_ID: 0x00000000	Phy Control
ASPEN_PHYCTLREG14_ID: 0x80000000	
ASPEN_PHYCTLREG7_ID: 0x177C2779	
ASPEN_PHYCTLREG8_ID: 0x07700770	
ASPEN_PHYCTLREG9_ID: 0x00000011	
...	
ASPEN_USRCMDREG0_ID: 0x00000001	Start the DRAM Initialization
ASPEN_OPREAD_ID: 0x00000000	Dummy reads to allow PLL to Update properly
ASPEN_OPREAD_ID: 0x00000004	
...	
ASPEN_OPREAD_ID: 0x000000fc	
CMCC_CONFIG_ENA_ID: 0x00000000	Memory Controller Configuration for BootRom only Config_ENA → enables/disables memory configuration by bootRom MEMTEST_ENA → enables/disables memory test in the bootRom CONSUMER_ID → 0x4F424D49 is ASCII for OBM!
CMCC_MEMTEST_ENA_ID: 0x00000000	
CMCC_CONSUMER_ID: 0x4F424D49	

An NTIM binary can be generated from the NTIM description file using the latest WTPTP release package available on the Marvell extranet at:

**My Products/Cellular & Handheld Solutions/Applications Processors/ARMADA 16x (ARMADA 16x - 88AP16X)/Software > WTPTP**

Specifically, the WTPTP NTTBBV4.exe tool is used to generate the NTIM binary (see Figure 2).

**Figure 2: Generating the NTIM binary**



Alternatively, an existing NTIM binary can be modified using a hex editor. [Figure 3: "Example DDR Package within the NTIM Binary File"](#) shows the DDR information in the NTIM binary. Starting at address 0x110 (in Figure 3) is a list of ID/value MCU register pairs. For convenience, the registers IDs are shown on the right side of Figure 3.

Ensure the file size is not altered if editing the binary NTIM file..

Figure 3: Example DDR Package within the NTIM Binary File

00000000	02010300	484D4954	00000000	07200903	....HMIT....	0) ASPEN_SDRREVREG_ID // revision
00000010	536B7921	FFFFFFFF	FFFFFFFF	FFFFFFFF	Sky! yyyyyyyyyyyyyy	1) ASPEN_SDRADCREG_ID // address decode
00000020	FFFFFFFF	FFFFFFFF	064E414E	03000000	yyyyyyyyy.NAN....	2) ASPEN_SDRCFGREG0_ID // sdram config reg 0
00000030	00000000	8C030000	484D4954	494D424F	....E...HMITIMBO	3) ASPEN_SDRCFGREG1_ID // sdram config reg 1
00000040	00000000	00B002D1	78040000	00000000	....°..Nx...[.]...	4) ASPEN_SDRTMGREG1_ID // sdram timing reg 1
00000050	00000000	00000000	00000000	00000000	.....	5) ASPEN_SDRTMGREG2_ID // sdram timing reg 2
00000060	00000000	00000000	00000000	00000000	.....	6) ASPEN_SDRTMGREG3_ID // sdram timing reg 3
00000070	00000000	494D424F	4F4C534F	00000200	....IMBOLSO....	7) ASPEN_SDRTMGREG4_ID // sdram timing reg 4
00000080	00B002D1	740F0100	00000000	00000000	°.Ñt.....	8) ASPEN_SDRTMGREG5_ID // sdram timing reg 5
00000090	00000000	00000000	00000000	00000000	.....	9) ASPEN_SDRCTLREG1_ID // sdram control reg 1
000000A0	00000000	00000000	00000000	00000000	.....	A) ASPEN_SDRCTLREG2_ID // sdram control reg 2
000000B0	4F4C534F	FFFFFFFF	00000400	00000007	....OLSOyyyyy.....	B) ASPEN_SDRCTLREG3_ID // sdram control reg 3
000000C0	00000800	00000000	00000000	00000000	.....	C) ASPEN_SDRCTLREG4_ID // sdram control reg 4
000000D0	00000000	00000000	00000000	00000000	.....	D) ASPEN_SDRCTLREG5_ID // sdram control reg 5
000000E0	00000000	00000000	00000000	4854504F	.....HTPO	E) ASPEN_SDRCTLREG6_ID // sdram control reg 6
000000F0	05000000	326D7352	14000800	00000000	.....2msr.....	F) ASPEN_SDRCTLREG7_ID // sdram control reg 7
00000100	00000000	00000000	43524444	38030000	.....CRDD8.....	10) ASPEN_SDRCTLREG13_ID // sdram control reg 13
00000110	00000000	0F000000	13000000	01000C00	.....	11) ASPEN_SDRCTLREG14_ID // sdram control reg 14
00000120	02000000	30240400	04000000	C8001D51	....0\$......È..Q	12) ASPEN_SDRERRREG_ID // sdram error status
00000130	05000000	42047874	06000000	534AC820	....B..xt....SJÈ	13) ASPEN_ADRMAPREG0_ID // address map cs0
00000140	07000000	D6D8E836	08000000	31011600	....00è6....1...	14) ASPEN_ADRMAPREG1_ID // address map cs1
00000150	09000000	00000000	0A000000	08000800	.....	15) ASPEN_USRCMDREG0_ID // user initiated cmd registers
00000160	0B000000	00000000	0C000000	04C08120	.....À.....À.	16) ASPEN_SDRSTAREG_ID // sdram status register
00000170	0E000000	01000000	0F000000	01000002	.....	17) ASPEN_PHYCTLREG3_ID // phy control reg 3
00000180	1C000000	00000000	1F000000	00000080	.....€	18) ASPEN_PHYCTLREG7_ID // phy control reg 7
00000190	18000000	79277C17	19000000	70077007	....y' .....p.p.	19) ASPEN_PHYCTLREG8_ID // phy control reg 8
000001A0	1A000000	11000000	1F000000	000000A0	.....	1A) ASPEN_PHYCTLREG9_ID // phy control reg 9
000001B0	2F000000	0A000000	1F000000	00000080	/.....	1B) ASPEN_PHYCTLREG10_ID // phy control reg 10
000001C0	2F000000	0A000000	1E000000	00010020	/.....	1C) ASPEN_PHYCTLREG11_ID // phy control reg 11
000001D0	2F000000	0A000000	1F000000	00000088	/.....	1D) ASPEN_PHYCTLREG12_ID // phy control reg 12
000001E0	2F000000	0A000000	1F000000	00000088	/.....	1E) ASPEN_PHYCTLREG13_ID // phy control reg 13
000001F0	2F000000	0A000000	2F000000	0A000000	/...../.....	1F) ASPEN_PHYCTLREG14_ID // phy control reg 14
00000200	1B000000	1C311000	1E000000	08010000	....1.....	20) ASPEN_DLLCTLREG1_ID // dll control reg 1
00000210	20000000	00010000	17000000	33440020	.....3D.	21) ASPEN_TSTMODREG0_ID // test mode reg 0
00000220	15000000	01000000	2F000000	E8030000	...../.....è...	22) ASPEN_TSTMODREG1_ID // test mode reg 1
00000230	30000000	00000000	30000000	04000000	0.....0.....	23) ASPEN_MCBCTLREG1_ID // mcb control reg 1
00000240	30000000	08000000	30000000	0C000000	0.....0.....	24) ASPEN_MCBCTLREG2_ID // mcb control reg 2
00000250	30000000	10000000	30000000	14000000	0.....0.....	25) ASPEN_MCBCTLREG3_ID // mcb control reg 3
00000260	30000000	18000000	30000000	1C000000	0.....0.....	26) ASPEN_MCBCTLREG4_ID // mcb control reg 4
00000270	30000000	20000000	30000000	24000000	0... ..0...\$....	27) ASPEN_PRFCCTLREG0_ID // perf control reg 0
00000280	30000000	28000000	30000000	2C000000	0...{...0...,...	28) ASPEN_PRFCCTLREG1_ID // perf control reg 1
00000290	30000000	30000000	30000000	34000000	0...0...0...4...	29) ASPEN_PRFSTAREG_ID // perf status reg
000002A0	30000000	38000000	30000000	3C000000	0...8...0...<...	2A) ASPEN_PRFSELREG_ID // perf select reg
000002B0	30000000	40000000	30000000	44000000	0...8...0...D...	2B) ASPEN_PRFCNTREG_ID // perf count reg
000002C0	30000000	48000000	30000000	4C000000	0...H...0...L...	2C) ASPEN_SDRTMGREG6_ID // sdram timing reg 6
000002D0	30000000	50000000	30000000	54000000	0...P...0...T...	2D) ASPEN_PHYCTLREGTST_ID // phy control reg test mode
000002E0	30000000	58000000	30000000	5C000000	0...X...0...\....	2E) ASPEN_PRF_COUNTER_ID // performance counter reg
000002F0	30000000	60000000	30000000	64000000	0...h...0...d...	2F) ASPEN_OPDELAY_ID // operation delay reg
00000300	30000000	68000000	30000000	6C000000	0...h...0...l...	30) ASPEN_OPREAD_ID // operation read reg
00000310	30000000	70000000	30000000	74000000	0...p...0...t...	31) ASPEN_MCU_REGID_E_MAX
00000320	30000000	78000000	30000000	7C000000	0...x...0...l...	

NTIM and OBM are OS-agnostic. OBM can be compiled on a Windows XP machine or on a Linux machine. To compile OBM, the Software Development Suite (SDT) is needed and is available here:

#### My Products/Cellular & Handheld Solutions/Development Tools/General/C++ SDT

The following sections explain how to arrive at the register values.

## 4. Configuration Registers

SDRAM\_CNTRL\_13

MEMORY\_ADDRESS\_MAP\_REG0

SDRAM\_CONFIG\_0

These registers configure the size of the DDR device (that is, 128MB, 256MB), number of banks, number of rows and columns as well as drive strength and on-die termination. Partial self-refresh for LPDDR1 can also be configured.

The Excel spreadsheet that complements this app note can be used to calculate registers SDRAM\_CONFIG\_0 and MEMORY\_ADDRESS\_MAP\_REG0.

The specification sheet of a DDR device contains the necessary information to configure these registers.

## 5. Timing Registers

SDRAM\_TIMING\_1/2/3/4/5

The specification sheet of a DDR device contains the timing values needed for proper operation. These values typically are listed in nanoseconds (ns) or number of data clock cycles (tCK). The ARMADA 16x MCU accepts those timings in number of data clocks only, so conversion is necessary if specified in fractions of a second.

Example:

At 200 MHz clock, one data clock cycle = 5 ns

At 400 MHz clock, one data clock cycle = 2.5 ns

So if the spec requires tRTP to be a minimum of 7.5 ns, then

At 200 MHz clock, program the SDRAM\_Config0[tRTP] with 2

At 400 MHz clock, program the SDRAM\_Config0[tRTP] with 3

The Excel spreadsheet that complements this app note can be used to calculate the values for all the timing registers. The following three diagrams show the meaning of the timing fields used by the MCU in the ARMADA 16X Applications Processor.

Figure 4: Timing Diagram 1

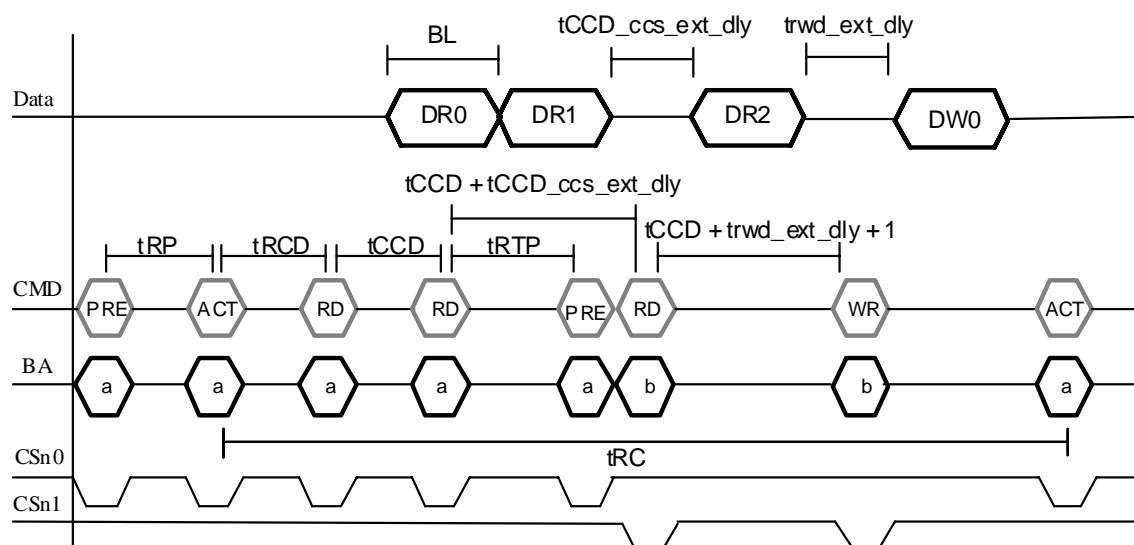
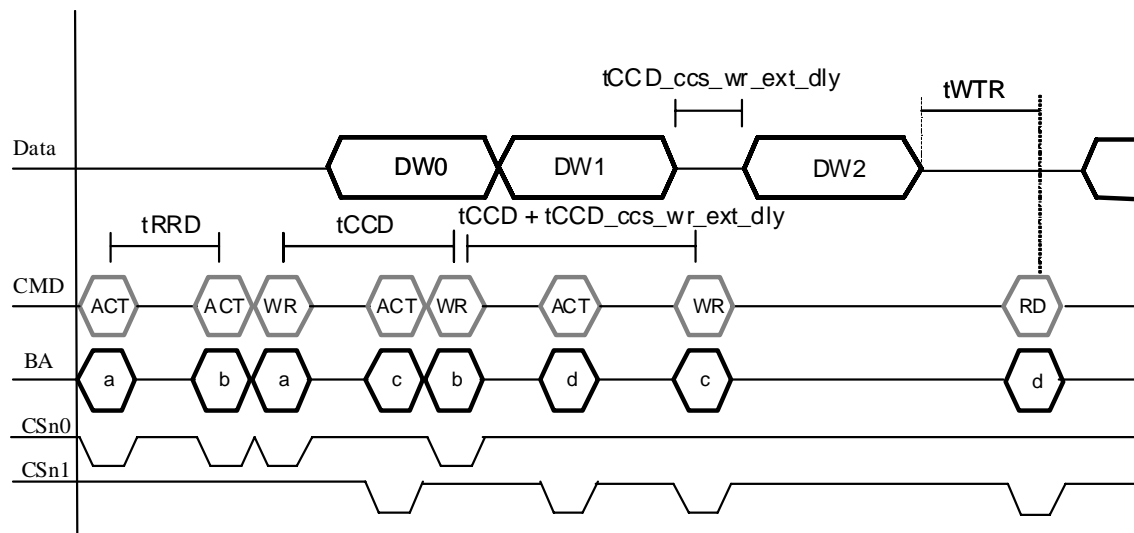
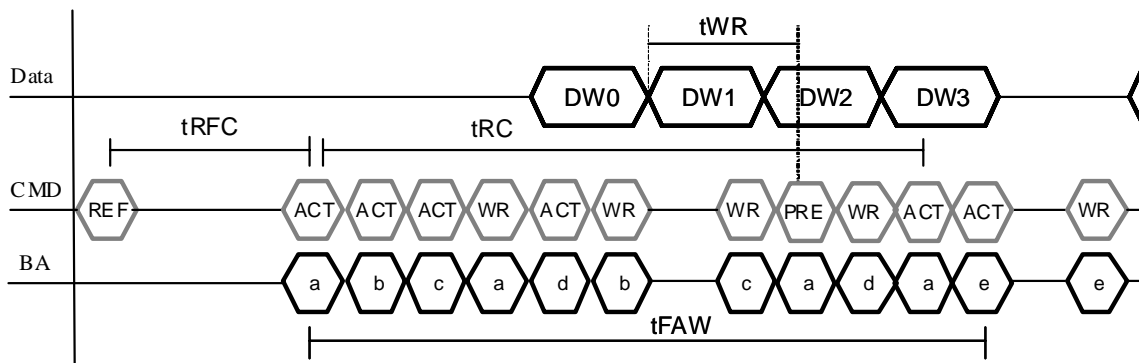


Figure 5: Timing Diagram 2



**Figure 6: Timing Diagram 3**



## 6. Control Registers

SDRAM\_CNTRL\_1/2/3/4/5/6/7

These registers control automatic power saving, automatic clock stopping (LPDDR1 only), DLL resetting (DDR2 only), SDRAM line boundary (set to line\_64), auto-precharging, Powerdown mode (DDR2 only), refresh posting enabling, fast and early write enabling, data width on the DDR device, SDRAM type (DDR1 or DDR2), CAS latency, burst length, fast/normal bank addressing, arbiter master weights for the four masters, pad termination mode for Reads, and On-die termination for Read and Write operations.

Refer to the *ARMADA 16x Applications Processor Hardware Manual* (MV-S301545-00) for details.

The Excel spreadsheet that complements this app note can be used to determine the register values for the SDRAM\_CNTRL\_4 and SDRAM\_CNTRL\_7.

## 7. PHY Registers and Tuning

PHY\_CNTRL\_3/7/8/9/10/11/12/13/14

The ARMADA 16x DDR PHY is a universal PHY (U-PHY). Adjust the driver strength and termination for DQ/DQS (data and data strobe), ADCM (Address and command), and CK (clock) to obtain optimal functionality. Tune the DDR PHY for either signal integrity (recommended) or for low power.

### 7.1 Optimal Signal Integrity Tuning

This approach adjusts for optimal and quantifiable signal integrity using an oscilloscope.

The goal is to have as steep as possible signal rise and fall times on the ARMADA 16x side with no under- or over-swing on the signal. This procedure must be verified with a high-speed capable (active or passive) scope probe.

In the simplest case, connect a 50-ohm coax cable to the scope with 50-ohm termination. On the other end, connect the cable to the signal of interest via a 500- or 1000-ohm resistor. The ARMADA 16x drive strength is set up in a similar fashion but with the scope probe on the memory side of the signal. The signal strength is increased until under- and over-swing is observed.

Each of the signals to the DDR memory can be adjusted to match the impedance of the transmission lines or traces to the DDR memory. To compensate for voltage and temperature, an external resistor is used to calibrate the driver impedance.

Each input signal from the DDR to the PHY can be terminated internally within the PHY. The impedance of the termination is adjustable to match the source impedance of the DDR.

**Note**

Since the DQ and the DQS lines provide on-die termination (ODT), no external series or parallel termination is required on those lines for point-to-point connections. This condition would apply to systems that use only one DDR memory chip. The CK, CK#, and address and control lines require parallel termination on the DDR memory side (75 ohm typically).

## 7.2 Low Power Tuning

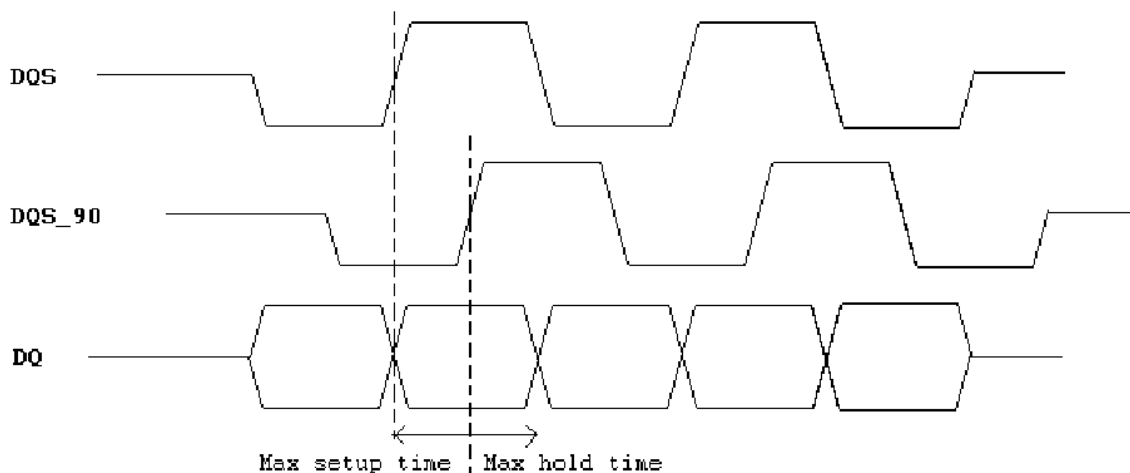
The goal here is to use minimal drive strength and thus power consumption at the possible expense of signal integrity, which can be performed without a scope. Start with a known working drive setting on the ARMADA 16x and memory side. After writing a pattern to memory, the DDR Controller is re-initialized with lower memory drive strength and the data is read back and checked for errors. If no errors occur, the drive strength can be reduced further. This procedure is repeated until Read failure occurs. When this level of drive has been reached on the memory side, the drive strength must be increased to the next higher level.

Starting again from a known working drive-strength setting on the ARMADA 16x and memory side, the drive strength on the ARMADA 16x side is reduced one step at a time. Next, a pattern is written to memory and read back to check for errors. The drive strength is reduced until errors occur. At this point, the drive strength must be increased by one step. The signal margin obtained by this tuning approach is harder to quantify than using the scope-tuning approach.

## 7.3 Read DLL

The DLL and the associated delay chain are used to delay the input data strobe (DQS) so that it aligns at the center of the data (DQ), resulting in a maximally balanced setup and hold time. Ninety-degrees phase shift is recommended but values between 45 to 135 degrees can be used.

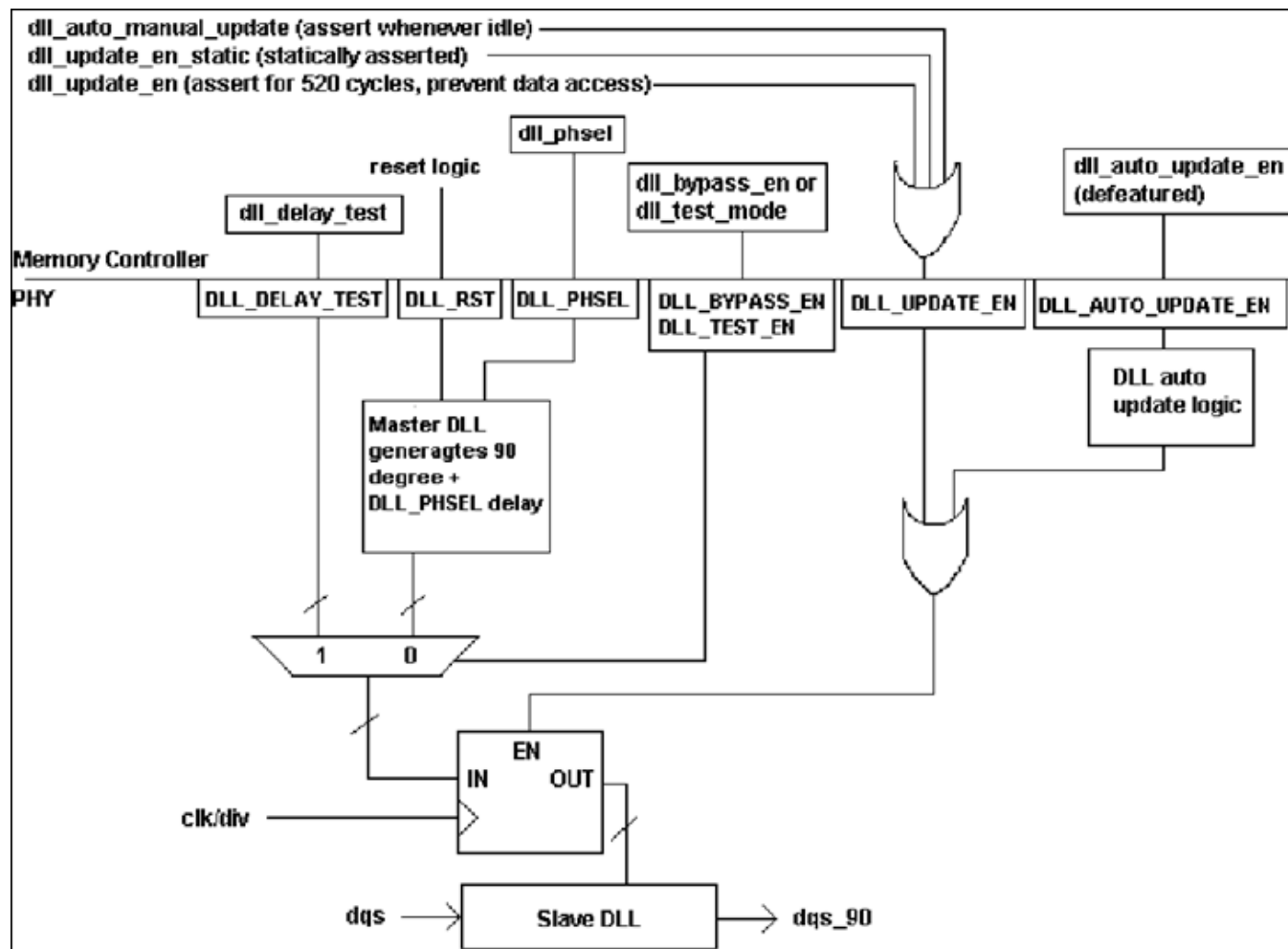
**Figure 7: Phase Shifting DQS by 90 Degrees**



To calibrate the DLLs, it is essential that the preferred DDR clock speed be set and stabilized before the calibration is enabled. The master DLL is continuously calibrating itself to determine the proper delay value for a 90-degree phase shift. The result of the master DLL calibration is provided in PHY\_CNTRL\_14[15:8]. If PHY\_CNTRL\_13[Dll\_auto\_manual\_update\_en] is set then the master DLL auto calibration delay value is used, along with the preferred phase values in PHY\_CNTRL\_13 and

PHY\_DLL\_CNTRL\_1, to control/update the two slave DQS DLLs (DQ[7:0] and DQ[15:0]). This update occurs only when the Memory Controller has been idle for 512 cycles (see Figure 8). This is the safest “auto” method, but periodic updates are not guaranteed, especially during heavy traffic.

**Figure 8: DLL Control Diagram**

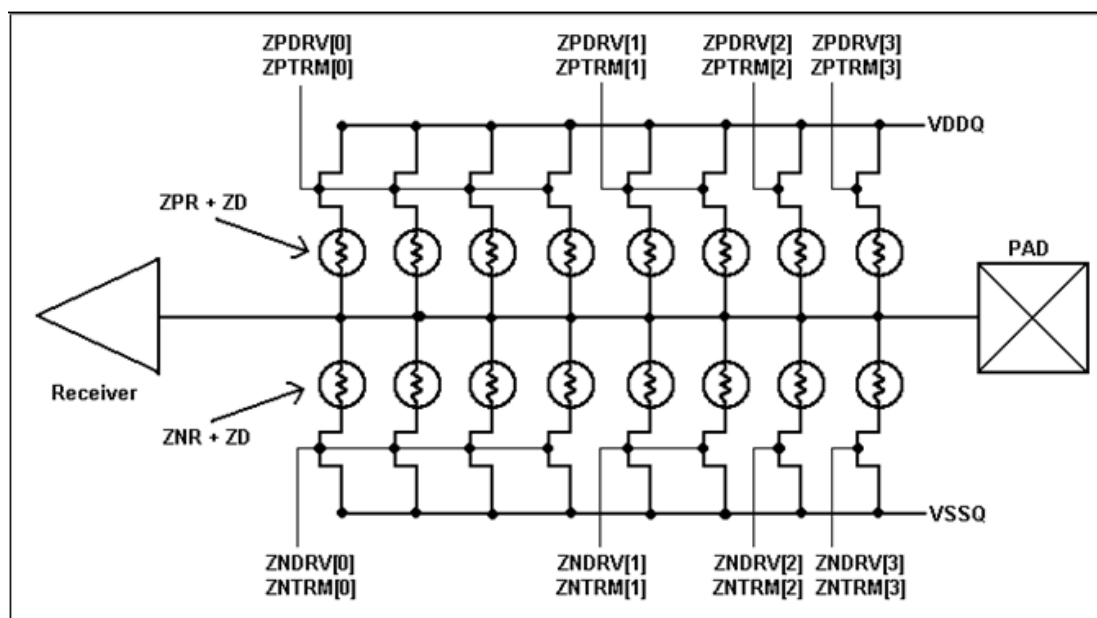


## 7.4 Drive Strength and Pad Termination Control on ARMADA 16x Devices

LPDDR1 devices typically provide drive strengths of normal, half, quarter, and eight. DDR2 devices provide drive strengths of full or reduced. This drive strength is set up in SDRAM\_CONFIG\_0[ODS] and communicated to the memory device during initialization of the DDR Memory Controller (see latest DDR setup example code).

ZPDRV and ZNDRV control the PMOS and NMOS coarse drive strength, and are used when the output enable (OE) is enabled, which means the PHY is driving this signal. During this time, only one side (either PMOS or NMOS) is driving at any given time. There are eight total mini-drivers (fingers) on each side. Bit 0 enables four fingers, Bit 1 enables two, and Bits 3 and 2 both enable one each. Thus, drive strength does not increase linearly with ZPDRV and ZNDRV.



**Figure 9: Pad Termination and Drive Control**

SDRAM\_CNTRL\_7[pad\_term\_switch\_mode] controls whether pad termination is on or off. When on, ZPTRM and ZNTRM in PHY\_CNTRL\_7 and PHY\_CNTRL\_8 control the pad termination strength. When the output enable (OE) is disabled, the drivers on both sides (ARMADA 16x and DDR device) are turned on according to ZPTRM and ZNTRM, resulting in a parallel termination. The finger control scheme is similar to ZPDRV and ZNDRV.

ZPR and ZNR are fine-tuning drivers. The higher the value, the lower the resistance of each finger (stronger drive strength). ZD is added on top of ZPR and ZNR, resulting in a range of 0 (equivalent to three-state, ZD=0 and ZPR/ZNR=0) to 16 (maximum per-finger strength, ZD=1 and ZPR/ZNR=15). Higher values are recommended for higher loads.

Data and data strobe (DQ/DQS), address/command (ADCM), and clock (CK) all have separate controls. CK and ADCM share the same Z\*DRV and Z\*TRM values. Table 1 maps the signal controls to the correct PHY registers.

**Table 1: Mapping Signal Drive and Termination Strength to PHY Registers**

	Z*DRV	Z*TRM	ZPR	ZNR	ZD
PHY_CNTRL_7	DQ/DQS	DQ/DQS	DQ/DQS	DQ/DQS	DQ/DQS
PHY_CNTRL_8	ADCM/CK	ADCM/CK	ADCM	ADCM	ADCM
PHY_CNTRL_9	--	--	CK	CK	CK

**Table 2: Example Drive Strengths**

DDR Type	ZPDRV	ZNDRV	ZPTRM	ZNTRM
Impedance Match (~50 ohms driving, no termination)	0011	0011	0	0
DDR1 (~30 ohms driving, termination on PCB)	1111	1111	0	0
DDR2 (~35 ohms driving, on-die 150 ohms termination to VDDQ and GND)	0111	0111	1100	10

## 7.5 On-Die Termination

On-Die Termination (ODT) is used with DDR2 and not with LPDDR1. For LPDDR1, the ARMADA 16x ODT must be switched off.

For LPDDR1, ODT must be off:

```
SDRAM_CONFIG_0[RTT] = 0,  
SDRAM_CNTRL_7[ODT0_switch_mode] = 0 (for Chip Select 0)  
SDRAM_CNTRL_7[ODT1_switch_mode] = 0 (for Chip Select 1)
```

For DDR2, ODT must be on:

```
SDRAM_CONFIG_0[RTT] = see Spec sheet for specific DDR2 device  
SDRAM_CNTRL_7[ODT0_switch_mode] = 0x1 or 0x2 (for Chip Select 0)  
SDRAM_CNTRL_7[ODT1_switch_mode] = 0x1 or 0x2 (for Chip Select 1)
```

The RTT value is communicated to the DDR device during initialization.

An ODT0\_switch\_mode = 0x1 means ODT is enabled for all Reads and Writes (recommended).

An ODT0\_switch\_mode = 0x2 means ODT is by SDRAM\_CNTRL\_6[ODT0\_write\_en] and SDRAM\_CNTRL\_6[ODT0\_read\_en]. This mode typically is used in a multi-chip-select system.

## 7.6 Receiver Type and Voltage Reference

This reference applies only to DQ and DQS lines, since the clock, the address and command, and mode pins are outputs only. Three options exist: internal reference, external reference, and DQSb. The external reference voltage is a good default choice. For receiver type, the CMOS receiver can be used as default for both DQS and DQ. However, the differential provides the best signal integrity at high speed. When using the differential receiver, the external reference voltage should be used for DQ and DQS# as reference voltage for DQS.

## 8. Entering and Exiting Self-Refresh Mode

Entering Self-Refresh mode:

1. Set PMUA\_MC\_SW\_SLP\_TYPE[MC\_SW\_SLP\_TYPE] = 0 to select Self-Refresh as the sleep type of the Memory Controller.
2. Set PMUA\_MC\_SLP\_REQ[MC\_SLP\_REQ\_MOH] = 1 to request entering Self-Refresh mode.
3. Poll PMUA\_MC\_SLP\_REQ[MC\_SLP\_REQ\_ACK] until this bit is set. This confirms the Memory Controller is in Self-Refresh mode.

Exiting Self-Refresh mode:

1. Set PHY\_CNTRL\_14[Phy\_sync\_en].
2. (DDR2 only) set SDRAM\_CNTRL\_1[DLL\_reset] to reset DRAM DLL (requires mode-register update).
3. (DDR2 only) set USER\_INITIATED\_COMMAND[user\_lmr0\_req] to send MRS command.
4. (DDR2 only) Clear SDRAM\_CNTRL\_1[DLL\_reset].
5. Set PHY\_CNTRL\_13[dll\_reset\_timer] = 0x8 (2048 cycles).
6. PHY\_CNTRL\_14[Phy\_DLL\_RST] to reset the PHY DLL. This blocks access to the controller for 2048 cycles.
7. Set PHY\_CNTRL\_14[update\_en\_static] and wait at least 512 DCLK cycles. This forces an update to the DQS slave DLLs.
8. Set PHY\_CNTRL\_14[update\_en\_static] again (this is a 1-to-toggle bit).
9. Set PMUA\_MC\_SW\_SLP\_TYPE[MC\_SW\_SLP\_TYPE] = 0 to select Self-Refresh as the sleep type of the Memory Controller.
10. Set PMUA\_MC\_SLP\_REQ[MC\_SLP\_REQ\_MOH] = 0 to request exist from Self-Refresh mode.

11. Poll PMUA\_MC\_SLP\_REQ[MC\_SLP\_REQ\_ACK] until this bit is cleared. This confirms the Memory Controller is out of low-power mode.

## 9. DDR Testing

The OBM does not contain test code. Marvell suggests developing code that tests for the following:

- Connectivity
- Memory retention
- Stress
- Performance

---

### Disclaimer

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose, without the express written permission of Marvell. Marvell retains the right to make changes to this document at any time, without notice. Marvell makes no warranty of any kind, expressed or implied, with regard to any information contained in this document, including, but not limited to, the implied warranties of merchantability or fitness for any particular purpose. Further, Marvell does not warrant the accuracy or completeness of the information, text, graphics, or other items contained within this document.

Marvell products are not designed for use in life-support equipment or applications that would cause a life-threatening situation if any such products failed. Do not use Marvell products in these types of equipment or applications.

With respect to the products described herein, the user or recipient, in the absence of appropriate U.S. government authorization, agrees:

- 1) Not to re-export or release any such information consisting of technology, software or source code controlled for national security reasons by the U.S. Export Control Regulations ("EAR"), to a national of EAR Country Groups D:1 or E:2;
- 2) Not to export the direct product of such technology or such software, to EAR Country Groups D:1 or E:2, if such technology or software and direct products thereof are controlled for national security reasons by the EAR; and,
- 3) In the case of technology controlled for national security reasons under the EAR where the direct product of the technology is a complete plant or component of a plant, not to export to EAR Country Groups D:1 or E:2 the direct product of the plant or major component thereof, if such direct product is controlled for national security reasons by the EAR, or is subject to controls under the U.S. Munitions List ("USML").

At all times hereunder, the recipient of any such information agrees that they shall be deemed to have manually signed this document in connection with their receipt of any such information. Copyright © 2011. Marvell International Ltd. All rights reserved. Marvell, the Marvell logo, Moving Forward Faster, Alaska, Fastwriter, Datacom Systems on Silicon, Libertas, Link Street, NetGX, PHYAdvantage, Pretera, Raising The Technology Bar, The Technology Within, Virtual Cable Tester, and Yukon are registered trademarks of Marvell. Ants, AnyVoltage, Discovery, DSP Switcher, Feroceon, GalNet, GalTis, Horizon, Marvell Makes It All Possible, RADLAN, UniMAC, and VCT are trademarks of Marvell. All other trademarks are the property of their respective owners.

Marvell Semiconductor, Inc.  
5488 Marvell Lane  
Santa Clara, CA 95054, USA  
Tel: 1.408.222.2500  
Fax: 1.408.752.9028  
Email: [commsales@marvell.com](mailto:commsales@marvell.com)  
[www.marvell.com](http://www.marvell.com)