i.MX6 DDR Stress Tester

User's Guide

Document Number: IMX6
Rev. V1.0.2
12/2013

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
www.freescale.com/support

Chapter 1
Introduction

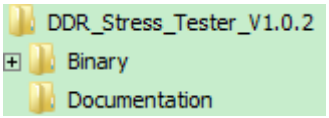
DDR_Stress_Tester is a software application for fine tuning DDR parameters and verifying DDR performance on i.MX6 boards. It is program running on PC which downloads a test image to the i.MX6 processor's IRAM through USB connection. The test image running on the target board will perform DDR calibrations and stress test. The result will be sent to PC through USB and be printed in the Command Prompt window.

The application performs write leveling, DQS gating, read/write delay calibration on the target board to match the layout of the board and archive the best DDR performance. In addition, the stress test can help the user to verify the DDR performance on their boards.

Chapter 2
Installation and Setup

2.1 Installation

There is no installation application for the DDRStressTester application. Simply create a folder where the application will reside and unzip DDR_Stress_Tester_V1.0.1.zip to the folder. There should be two sub-folders: Binary and Documentation.



Make sure the DDR_Stress_Tester.exe and the target images ddr-stress-test-mx6dq.bin, ddr-stress-test-mx6dl.bin and ddr-stress-test-mx6sl.bin are in the "Binary" directory.

2.2 System Requirements

- Minimum PC Requirements – 2.0 GHz CPU, 1GB RAM with USB connection.
- Windows® XP w/Service Pack 2 or later

Chapter 3
Running DDR Stress Tester

To run the DDR Stress Tester, perform the following steps:

1. Connect the target board to PC host

   Configure the i.MX6 target board to boot in Serial Downloader mode by setting BOOT_MODE[1:0] to 01 and power up the board. Connect a USB cable from the host computer to the USB OTG port on the MX6 target board. A “HID-compliant device” will be shown in the Device Manager as Figure 3-1:

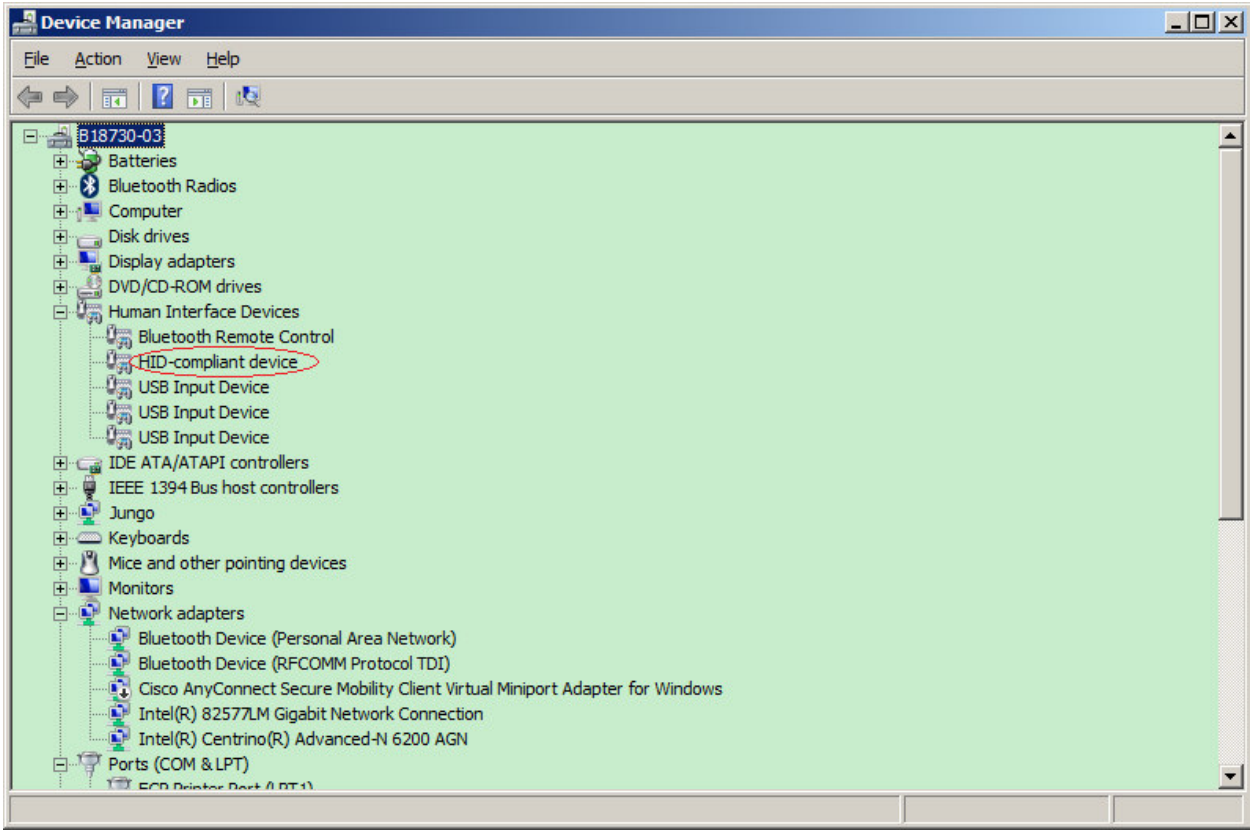


Figure 3-1 HID compliant device

2. Launch the DDR_Stress_Tester application

In Windows' Command Prompt, change to the Binary folder and then type the following command:

"DDR_ Stress_Tester -t mx6x -df mx6x_ddr_script_filename"

The DDR Stress Test version and DDR configuration will be displayed as Figure 3-2. If the DDR configuration does not match the board, please check whether the correct initialization script is used or the BOOT_CFG3[5:4] fuses are correctly set.

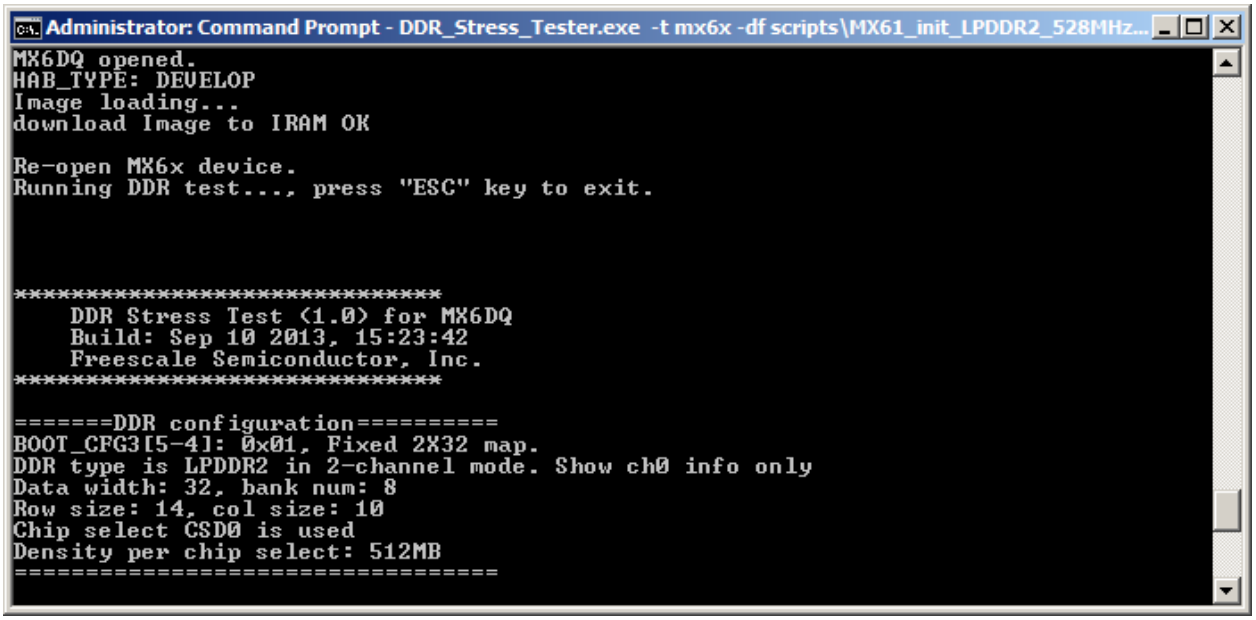


Figure 3-2 DDR configuration display window

3. Select the ARM core speed

This is for selecting the ARM core speed for running the calibrations and stress test. The maximum ARM core speed of i.MX6Q and i.MX6D is 1.2GHz, and the maximum ARM core speed of i.MX6DL, i.MX6S, and i.MX6SL is 1.0GHz.

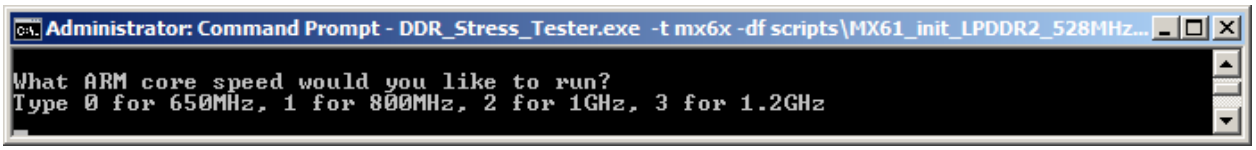


Figure 3-3 ARM core speed set up window

4. Select the DDR density

This is for selecting the size of memory to be tested. For DDR3, input the density of each chip select. For LPDDR2, input the density of each channel. If there are two chip selects per channel, input the total density for the channel. For example, if CSD0 and CSD1 are used for channel 0 and the density of each chip select is 512MB, 1GB should be selected.

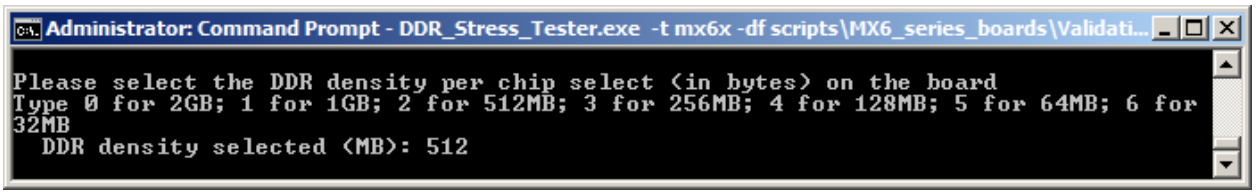


Figure 3-4 DDR density set up window for DDR3

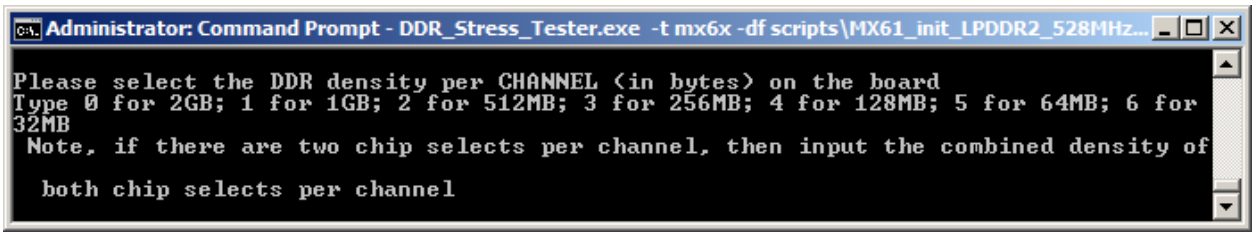


Figure 3-5 DDR density set up window for LPDDR2

5. Select chip select for DDR3

   This option will only be shown for DDR3 with 2 chip selects. If only one chip select is used, CSD0 will be used as default.

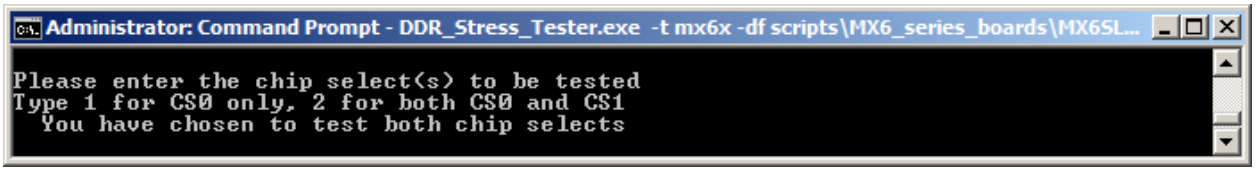


Figure 3-6 chip select set up window

6. Select the number of channels for LPDDR2

   This option will only be shown for LPDDR2 with 2 channels. If there are two channels, the user can choose to run the stress test on either channel 0 or channel 1. If it is desired to

perform calibration, the user must choose 2 channels. If it is chosen to test 1 channel here, the user must skip the calibration in the later steps.

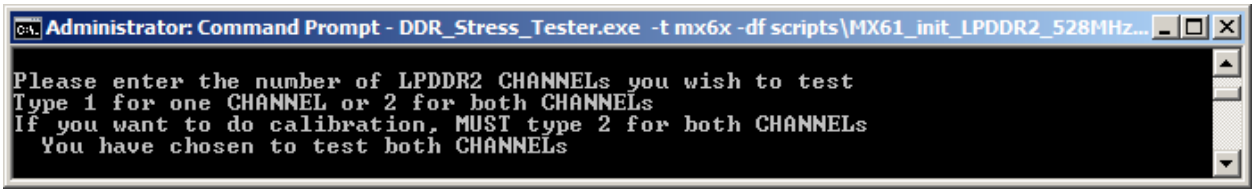


Figure 3-7 number of channel set up window for LPDDR2

7. Select DDR frequency for Calibration

   The default DDR calibration frequency is 400MHz for i.MX6DL, i.MX6S, and i.MX6SL. And, the default DDR calibration frequency for i.MX6Q and i.MX6D is 528MHz. The user can also press 'n' and input a calibration frequency which is between 350MHz and 528MHz for debugging purpose.

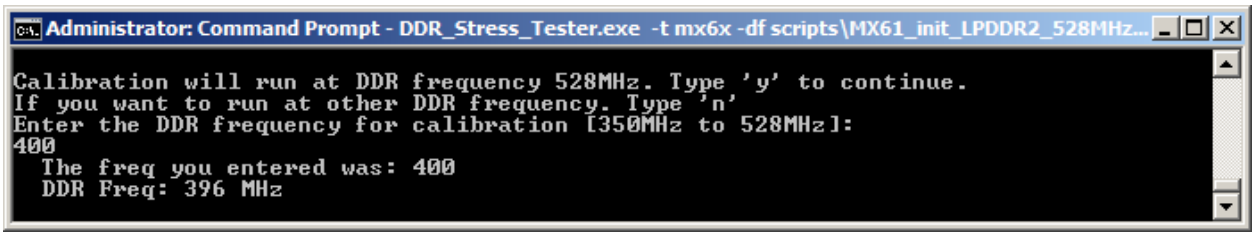


Figure 3-8 DDR calibration frequency set up window

8. Write leveling calibration

   The write leveling calibration can generate a delay between the clock and the associate DQS of up to 3 cycles as following: (WL_DL_ABS_OFFSET/256*cycle) + (WL_HC_DEL*half cycle) + (WL_CYC_DEL*cycle).

   Write leveling calibration is supported by DDR3 only. For LPDDR2, write-leveling calibration is disabled.

The DDR3 needs to be put into write leveling mode with MR1 command before write leveling calibration. After finishing write leveling calibration, it needs to restore the value of MR1 register on DDR3 to exit write leveling mode. So before write leveling calibration is executed, the user is requested to input the MR1 value. Please make sure the same MR1 value on the initialization script is entered here.

setmem /32 0x021b001c = 0x00048031 // MMDC0_MDSCR, MR1 write, CS0

The MR1 value is upper 4 hex bits written to MMDCx_MDSCR as shown above.

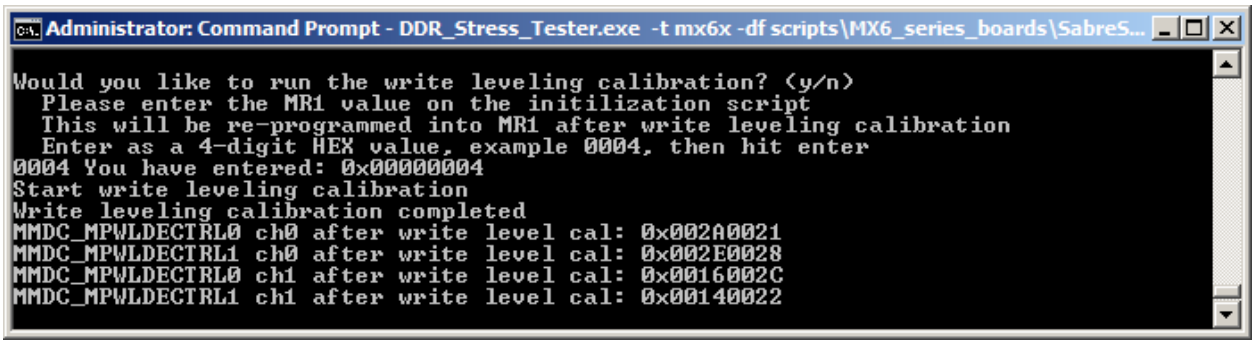


Figure 3-9 write leveling calibration window

NOTE

If write-leveling delay is larger than 0x2f, it is suggested to set the WALAT value on MMDCx_MDMISC register to 1 in the initialization script and re-run the DDR_Stress_Tester.

9. DQS gating calibration

The read DQS gating calibration is used to adjust the read DQS gating with the middle of the read DQS preamble.

The DQS gating includes a delay of up to 7 cycles (The delay is chosen according to two fields MPDGCTRL#[DG_HC_DEL#] and MPDGCTRL#[DG_DL_ABS_OFFSET#]).

DQS gating calibration will only be performed for DDR3 as DQS gating is not supported when using LPDDR2.

After running the DQS gating calibration, the following intermediate result will be displayed. The “start” and “end” show the start value and end value of DQS gating “window”. Mean is the middle point of the window. The suggested DQS gating value is calculated base on the formula *max[mean(start,end),end-0.5*tCK]*.

BYTE 0:

Start:	*HC=0x01 ABS=0x58*
End:	*HC=0x04 ABS=0x34*
Mean:	*HC=0x03 ABS=0x06*
*End-0.5*tCK:*	*HC=0x03 ABS=0x34*
Final:	*HC=0x03 ABS=0x34*

On the result, HC means 0.5 cycle, ABS represents 1/256 of a cycle. For example, HC=0x01 ABS=0x58 means that the delay is 0.5 + 88/256 cycle.

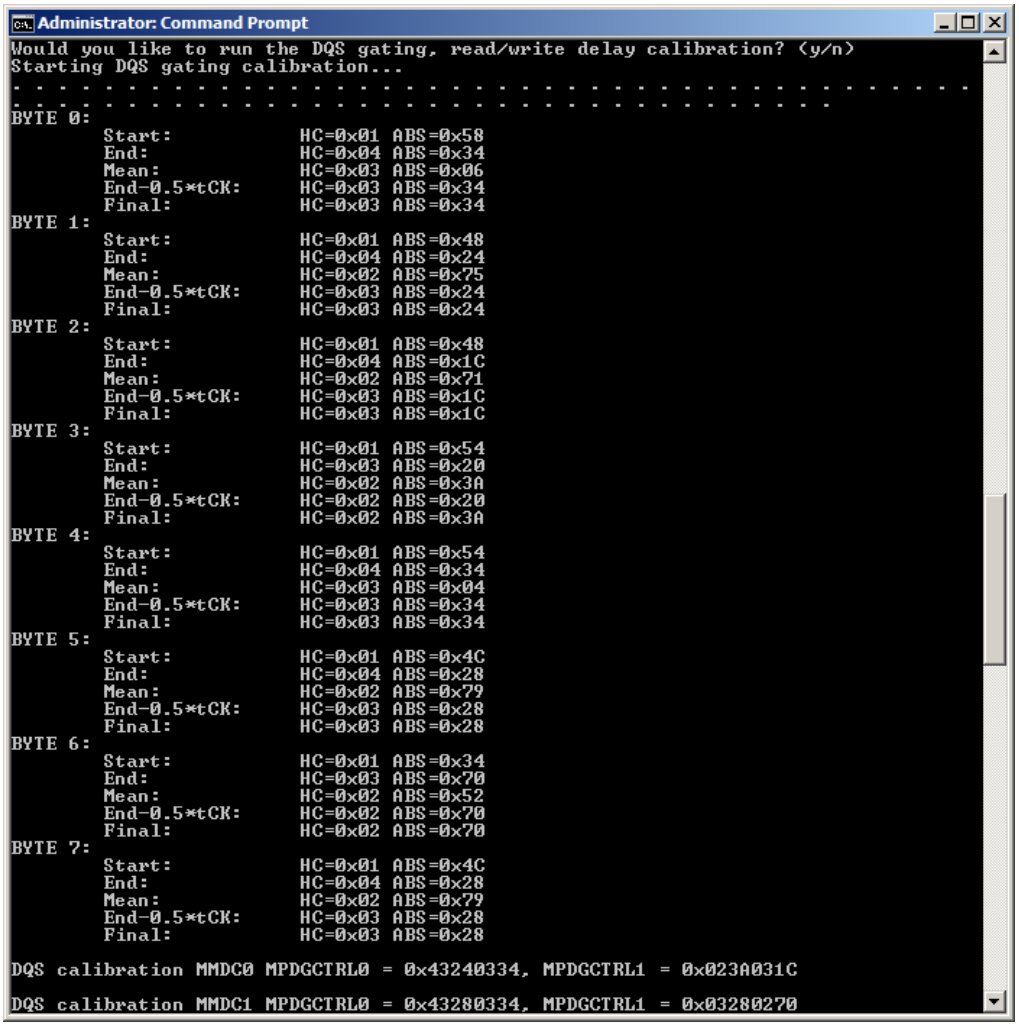


Figure 3-10 DQS calibration window

10. Read/write delay calibration

The read/write calibration is used to adjust the delay DQS between data for read and write operations. The intermediate result will be shown when running read/write delay calibration, so the user can know the window (lower and upper delay value for proper operation) for each data byte. The delay value is increased on each step and the result shows that particular byte can perform read or write properly with the delay value. For the result, the first digit on the right is byte 0, the second digit is byte 1, third digit is byte 2, and so forth. A '1' means the byte failed the read/write test with the delay value. A '0' means the byte can read/write properly.

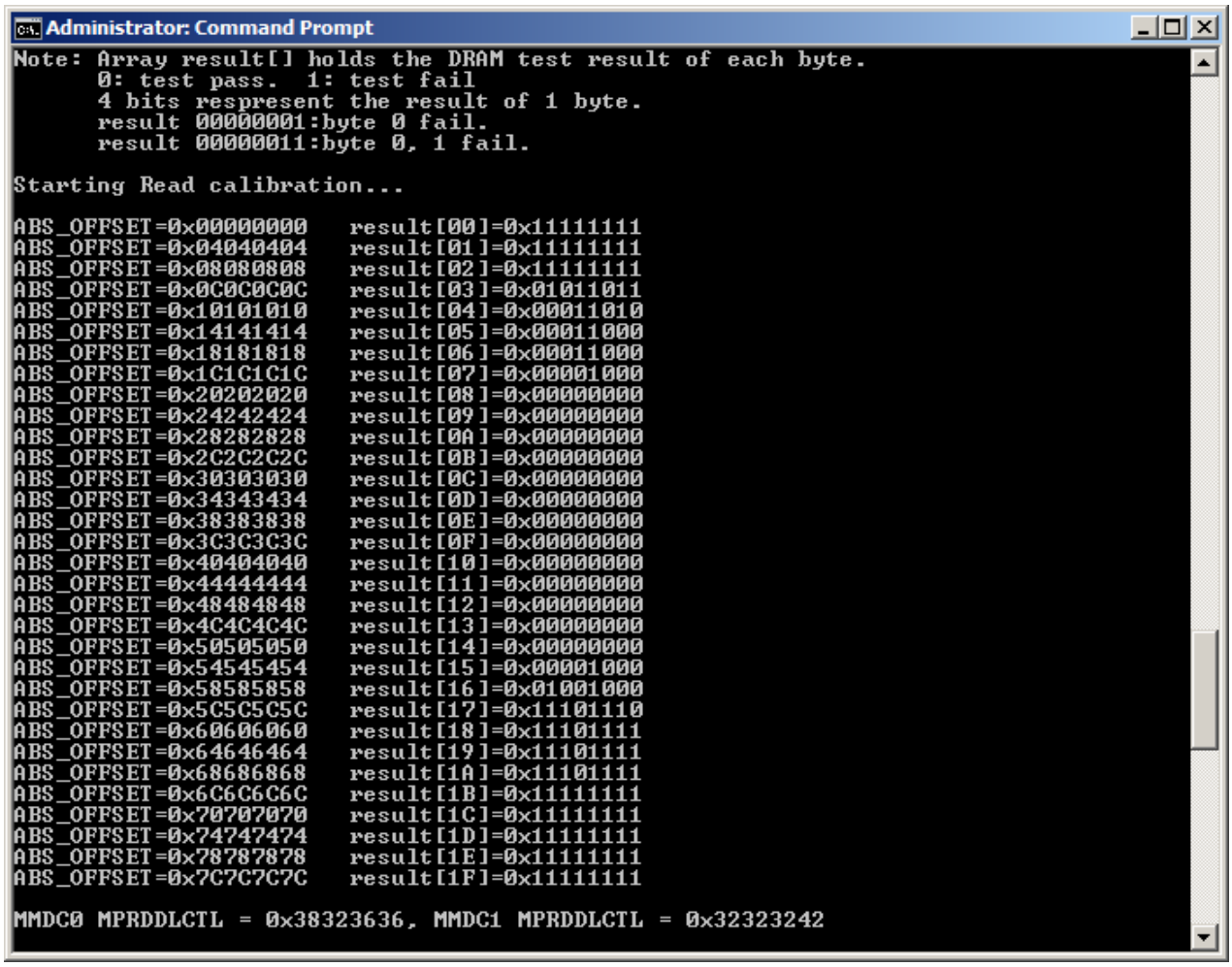


Figure 3-11 read/write delay calibration window

The average between lower and upper boundaries is calculated for the final read/write delay value. When the read/write delay calibration is completed, the tool will update the MPRDDLCTL and MPWRDLCTL registers for the DDR stress test.

11. Calibration result

After the whole calibration process completes, DQS gating, read/write delay value will be displayed and the corresponding registers will be updated for the memory stress test afterward. However, the user should update the values of MMDC registers in initialization script file according to the calibration results.

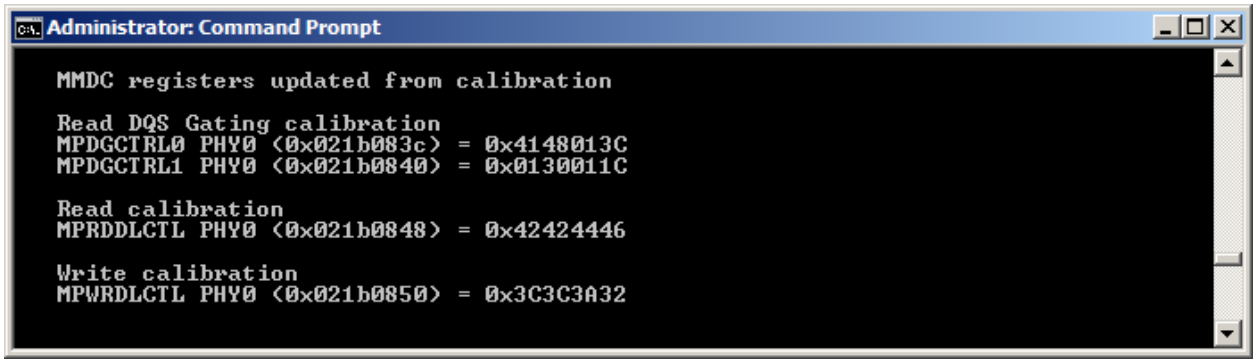


Figure 3-12 calibration result of DDR3

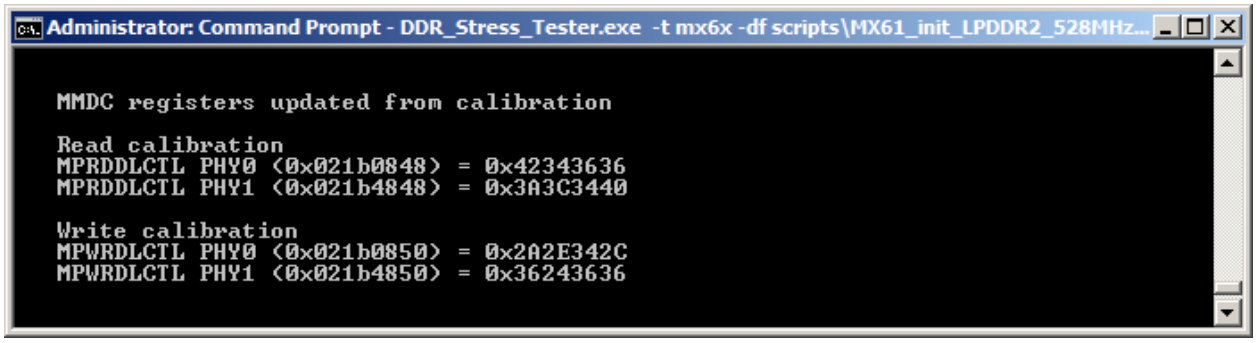


Figure 3-13 calibration result of LPDDR2

12. DDR stress test

DDR stress test will run memory test with different test patterns and go through the memory space specified in step 4. The user can input the start and end frequency which is in the range from 135MHz to 672MHz. If any failure occurs, the test will stop and show the failure data pattern.

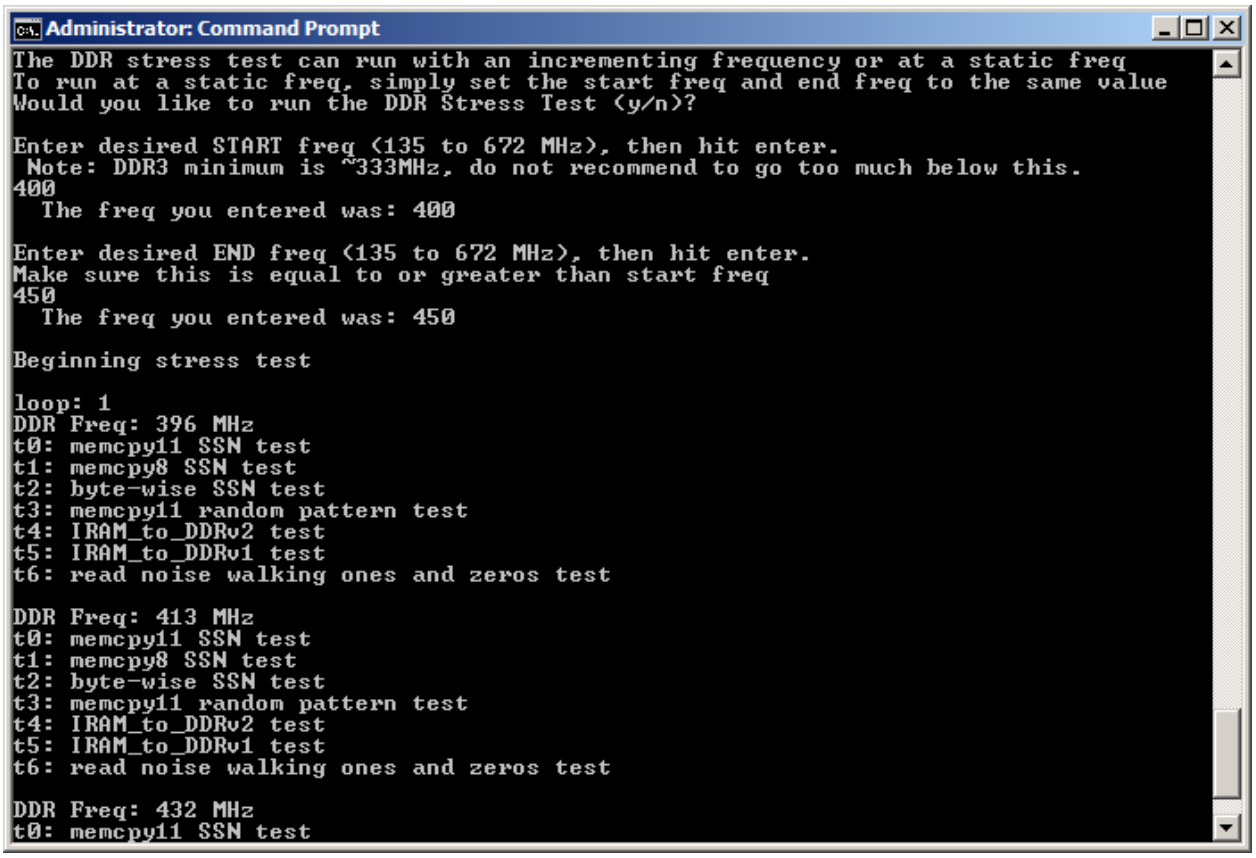


Figure 3-14 run DDR stress test

It is suggested to run the stress test at the target DDR operating frequency (528 MHz for i.MX6Q and i.MX6D, 400 MHz for i.MX6DL, i.MX6S, and i.MX6SL) for an extended period of time to verify the DDR performance on the target board. Entering the same value for START and END frequency will allow the application to perform the stress test in loop at the same frequency. It is not suggested to run the stress test at higher frequencies for verifying the stability of the DDR as the DDR signals waveform would be very different at higher frequencies and it is not meaningful.

In case the target board fails the stress test at the target operating frequency, you could try to run at the lower frequency to confirm the DDR connections and setup are correct first. Afterward, you can review the MMDC settings and DDR pins drive strength, modify the DDR initialization script and re-run the calibrations and stress test.

Chapter 4
DDR init script

When DDR_Stress_Tester runs, a DDR init scrip file (*.inc) should be selected. The script file contains parameters for DDR initialization. The users can use the script files in the release package as template to create the script file for their own boards.

The command format in script file is as below and the comment starts with "//"

setmem /32 address = value // comment

The following is a DDR script file sample:

```
//init script for i.MX6SL LPDDR2

// Enable all clocks (they are disabled by ROM code)
setmem /32  0x020c4068 =  0xffffffff
setmem /32  0x020c406c =  0xffffffff
setmem /32  0x020c4070 =  0xffffffff
setmem /32  0x020c4074 =  0xffffffff
setmem /32  0x020c4078 =  0xffffffff
setmem /32  0x020c407c =  0xffffffff
setmem /32  0x020c4080 =  0xffffffff
setmem /32  0x020c4084 =  0xffffffff

setmem /32  0x020c4018 =  0x00260324    //DDR clk to 400MHz


// DDR IOMUX configuration
//DDR IO TYPE:
setmem /32  0x020e05c0 =  0x00020000    // IOMUXC_SW_PAD_CTL_GRP_DDRMODE
setmem /32  0x020e05b4 =  0x00000000    // IOMUXC_SW_PAD_CTL_GRP_DDRPKE …
```

```
//CLOCK:
setmem /32    0x020e0338 =    0x00000028    // IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_0

//Control:
setmem /32    0x020e0300 =    0x00000030    // IOMUXC_SW_PAD_CTL_PAD_DRAM_CAS
setmem /32    0x020e031c =    0x00000030    // IOMUXC_SW_PAD_CTL_PAD_DRAM_RAS
setmem /32    0x020e0320 =    0x00000030    // IOMUXC_SW_PAD_CTL_PAD_DRAM_RESET
setmem /32    0x020e032c =    0x00000000    // IOMUXC_SW_PAD_CTL_PAD_DRAM_SDBA2 - DSE can be
configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_CTLDS
//setmem /32  0x020e033c =    0x00000008    // IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT0
//setmem /32  0x020e0340 =    0x00000008    // IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT1
setmem /32    0x020e05ac =    0x00000030    // IOMUXC_SW_PAD_CTL_GRP_ADDDS
setmem /32    0x020e05c8 =    0x00000030    // IOMUXC_SW_PAD_CTL_GRP_CTLDS

//Data Strobes:
setmem /32    0x020e05b0 =    0x00020000    // IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL
setmem /32    0x020e0344 =    0x00003030    // IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS0
setmem /32    0x020e0348 =    0x00003030    // IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS1
setmem /32    0x020e034c =    0x00003030    // IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS2
setmem /32    0x020e0350 =    0x00003030    // IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS3

//Data:
setmem /32    0x020e05d0 =    0x00080000    // IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE
setmem /32    0x020e05c4 =    0x00000030    // IOMUXC_SW_PAD_CTL_GRP_B0DS
setmem /32    0x020e05cc =    0x00000030    // IOMUXC_SW_PAD_CTL_GRP_B1DS
setmem /32    0x020e05d4 =    0x00000030    // IOMUXC_SW_PAD_CTL_GRP_B2DS
setmem /32    0x020e05d8 =    0x00000030    // IOMUXC_SW_PAD_CTL_GRP_B3DS

setmem /32    0x020e030c =    0x00000030    // IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM0
setmem /32    0x020e0310 =    0x00000030    // IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM1
setmem /32    0x020e0314 =    0x00000030    // IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM2
setmem /32    0x020e0318 =    0x00000030    // IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM3

//==========================================================================
// DDR Controller Registers
//==========================================================================

// Manufacturer:            Samsung
// Device Part Number:      K4P8G304EB-AGC1
// Clock Freq.:             400MHz
```

```
// Density per CS in Gb: 4
// Chip Selects used:       2
// Total DRAM density (Gb)          8
// Number of Banks:       8
// Row address:           14
// Column address:        10
// Data bus width         32
//==========================================================================
// MMDC0_MDSCR, set the Configuration request bit during MMDC set up
setmem /32     0x021b001c =       0x00008000          // Chan 0
setmem /32     0x021b085c =       0x1b4700c7          //LPDDR2 ZQ params

//==========================================================================
// Calibration setup.
//==========================================================================
setmem /32     0x021b0800 =       0xa1390003          // DDR_PHY_P0_MPZQHWCTRL, enable one time ZQ calibration
setmem /32      0x021b0890 =      0x00300000                  //ca bus abs delay
setmem /32     0x021b08b8 =       0x00000800          //frc_msr.

// read delays, settings recommended by design to remain constant
setmem /32     0x021b081c =       0x33333333          // DDR_PHY_P0_MPREDQBY0DL3
setmem /32     0x021b0820 =       0x33333333          // DDR_PHY_P0_MPREDQBY1DL3
setmem /32     0x021b0824 =       0x33333333          // DDR_PHY_P0_MPREDQBY2DL3
setmem /32     0x021b0828 =       0x33333333          // DDR_PHY_P0_MPREDQBY3DL3

// write delays, settings recommended by design to remain constant
setmem /32     0x021b082c =       0xf3333333          //DDR_PHY_P0 all byte 0 data & dm delayed by 3
setmem /32     0x021b0830 =       0xf3333333          //DDR_PHY_P0 all byte 0 data & dm delayed by 3
setmem /32     0x021b0834 =       0xf3333333          //DDR_PHY_P0 all byte 0 data & dm delayed by 3
setmem /32     0x021b0838 =       0xf3333333          //DDR_PHY_P0 all byte 0 data & dm delayed by 3

// Read and write data delay, per byte.
// For optimized DDR operation it is recommended to run mmdc_calibration on your board, and replace 4 delay register
assigns with resulted values
// Note:
// a. DQS gating is not relevant for LPDDR2. DSQ gating calibration section should be skipped, or the following write/read
calibration will stall
// b. The calibration code that runs for both MMDC0 & MMDC1 should be used.
setmem /32     0x021b0848 =       0x4241444a          // MPRDDLCTL PHY0
setmem /32     0x021b0850 =       0x3030312b          // MPWRDLCTL PHY0
setmem /32     0x021b083c =       0x20000000          //PHY0 dqs gating dis
```

```
setmem /32    0x021b0840 =    0x0
setmem /32    0x021b08b8 =    0x00000800        //frc_msr.
//==========================================================================
// Calibration setup end
//==========================================================================

// Channel0 - starting address 0x80000000
setmem /32    0x021b000c =    0x33374133        // MMDC0_MDCFG0
setmem /32    0x021b0004 =    0x00020024        // MMDC0_MDPDC
setmem /32    0x021b0010 =    0x00100A82        // MMDC0_MDCFG1
setmem /32    0x021b0014 =    0x00000093        // MMDC0_MDCFG2
setmem /32    0x021b0018 =    0x00001688        // MMDC0_MDMISC
setmem /32    0x021b002c =    0x0F9F26D2        // MMDC0_MDRWD
setmem /32    0x021b0030 =    0x0000020E        // MMDC0_MDOR
setmem /32    0x021b0038 =    0x00190778        // MMDC0_MDCFG3LP
setmem /32    0x021b0008 =    0x00000000        // MMDC0_MDOTC
setmem /32    0x021b0040 =    0x0000004F        // Chan0 CS0_END
setmem /32    0x021b0000 =    0xC3110000        // MMDC0_MDCTL

//==========================================================================
// LPDDR2 Mode Register Writes
//==========================================================================
// Channel 0 CS0
setmem /32    0x021b001c =    0x003F8030        // MRW: BA=0 CS=0 MR_ADDR=63 MR_OP=0 (Reset)
setmem /32    0x021b001c =    0xFF0A8030        // MRW: BA=0 CS=0 MR_ADDR=10 MR_OP=0xff (IO calibration,
calibration code)
setmem /32    0x021b001c =    0x82018030        // MRW: BA=0 CS=0 MR_ADDR=1  MR_OP=see Register
Configuration
setmem /32    0x021b001c =    0x04028030        // MRW: BA=0 CS=0 MR_ADDR=2  MR_OP=see Register
Configuration
setmem /32    0x021b001c =    0x02038030        // MRW: BA=0 CS=0 MR_ADDR=3  MR_OP=see Register
Configuration
// Channel 0 CS1
//setmem /32  0x021b001c =    0x003F8038        // MRW: BA=0 CS=1 MR_ADDR=63 MR_OP=0 (Reset)
setmem /32    0x021b001c =    0xFF0A8038        // MRW: BA=0 CS=1 MR_ADDR=10 MR_OP=0xff (IO calibration,
calibration code)
setmem /32    0x021b001c =    0x82018038        // MRW: BA=0 CS=1 MR_ADDR=1  MR_OP=see Register
Configuration
setmem /32    0x021b001c =    0x04028038        // MRW: BA=0 CS=1 MR_ADDR=2  MR_OP=see Register
Configuration
setmem /32    0x021b001c =    0x02038038        // MRW: BA=0 CS=1 MR_ADDR=3  MR_OP=see Register
```

Configuration

```
//#######################################################
//final DDR setup, before operation start:
setmem /32    0x021b0800 =    0xa1310003      // DDR_PHY_P0_MPZQHWCTRL, enable automatic ZQ calibration
setmem /32    0x021b0020 =    0x00001800      // MMDC0_MDREF
setmem /32    0x021b0818 =    0x0             // DDR_PHY_P0_MPODTCTRL
setmem /32    0x021b08b8 =    0x00000800      // DDR_PHY_P0_MPMUR0, frc_msr
setmem /32    0x021b0004 =    0x00025564      // MMDC0_MDPDC now SDCTL power down enabled
setmem /32    0x021b0404 =    0x00011006      //MMDC0_MAPSR ADOPT power down enabled
setmem /32    0x021b001c =    0x00000000      // MMDC0_MDSCR, clear this register (especially the configuration
bit as initialization is complete)…
```

NOTE

The DDR initialization script included in this release are dedicated for Freescale reference boards. Customers should fine tune the script for their own board.

Chapter 5
FAQ

- **How to configure the test to run in 16-bit, 32-bit, and 64-bit data bus mode?**
  The test will read the DDR initialization script and determine the data bus size based on the setting in the script.
- **Do I need to update the DDR initialization script after running the calibration?**
  Yes. The calibration results are stored in the MMDC registers during the test only. It is suggested to update the initialization script and re-run the test on different boards to confirm the DDR performance and make sure the MMDC register settings are correctly ported to the firmware.
- **When should I skip the calibration?**
  The calibration results depend on the PCB layout. PCBs with the same layout can use the same calibration results. So, you can skip the calibration when you are verifying the DDR performance on PCBs with same layout. Please make sure the calibration results are incorporated in the initialization script.
- **Can I perform the calibration on one channel only when using two-channel LPDDR2?**
  No. The tool supports calibration on both channels. If you want to test the DDR performance on one channel only, you can run the calibration and update the initialization script with the calibration results. Then, re-run the test and skip the calibration.

Chapter 6
Reference Documents

- i.MX 6 Series Multimedia Applications Processor Reference Manual
- AN4467 i.MX6 Series DDR Calibration
- DDR3 SDRAM JESD79-3D JEDEC Standard
- LPDDR2 JESD209-2D JEDEC Standard

Chapter 7
Revision History

Table 1 provides a revision history for this user guide.

Table 1. Document Revision History

Rev. Number	Date	Substantive Change(s)
Rev. 1	09/2013	Initial public release.
Rev. 2	10/2013	Update for 1.0.1 release
Rev. 3	12/2013	Update for 1.0.2 release