NXP Semiconductors

User Guide

Document identifier: DDRTUG

Rev. 1, 21 December 2021

DDR Tool User Guide

Contents

Chapter 1 Introduction

This document introduces the DDR configuration and validation tool. It is an embedded component of Config tools for i.MX and supports the i.MX 8M family of application processors.

The DDR tool provides two main functionalities: configuration and validation.

NOTE

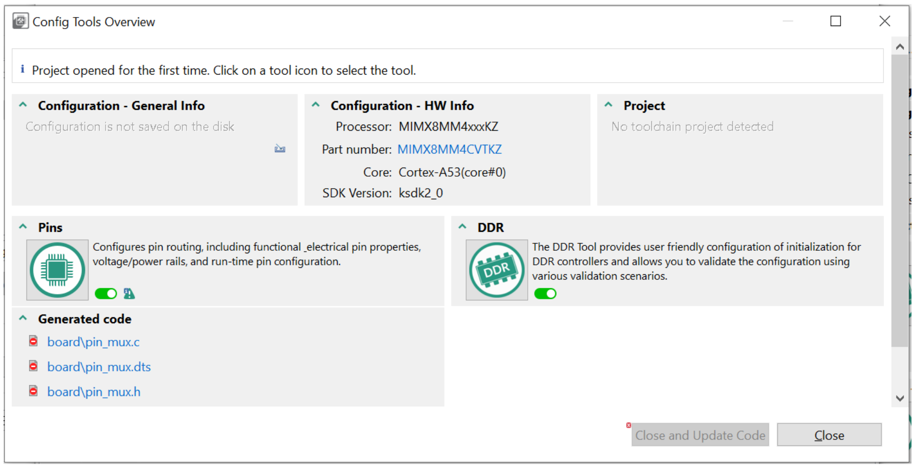
The DDR tool is provided to help customers evaluate, debug, and optimize their designs. The results, or any part thereof, provided by the tool cannot be seen as a substitute for the traditional validation and compliance methods, which you must perform to declare compliance of the designs with the respective JEDEC standards.

Chapter 2
Create a new DDR tool project

To use the DDR tool, you first must create a new project.

To create a new DDR tool project, follow these steps:

1. Open the **Config tools for i.MX**.
2. Choose **Create a new standalone configuration for processor, board, or kit** and click **Next**.
3. From **Processors**, choose one of the devices with DDR tool support and click **Finish.**
4. To open the DDR tool view, Click **DDR** tool icon.



Config Tools Overview

Project opened for the first time. Click on a tool icon to select the tool.

Configuration - General Info

Configuration is not saved on the disk

Configuration - HW Info

Processor: MIMX8MM4xxxKZ
Part number: MIMX8MM4CVTKZ
Core: Cortex-A53(core#0)
SDK Version: ksdk2_0

Project

No toolchain project detected

Pins

Configures pin routing, including functional ,electrical pin properties, voltage/power rails, and run-time pin configuration.

DDR

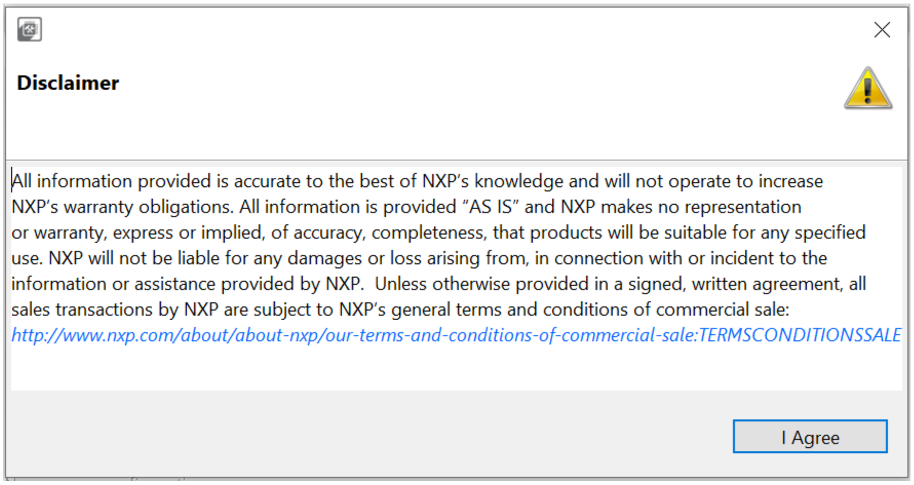
The DDR Tool provides user friendly configuration of initialization for DDR controllers and allows you to validate the configuration using various validation scenarios.

Generated code

board\pin_mux.c
board\pin_mux.dts
board\pin_mux.h

Figure 1. Config Tools overview

5. To use the DDR tool, accept the Disclaimer.



Disclaimer

All information provided is accurate to the best of NXP's knowledge and will not operate to increase NXP's warranty obligations. All information is provided "AS IS" and NXP makes no representation or warranty, express or implied, of accuracy, completeness, that products will be suitable for any specified use. NXP will not be liable for any damages or loss arising from, in connection with or incident to the information or assistance provided by NXP. Unless otherwise provided in a signed, written agreement, all sales transactions by NXP are subject to NXP's general terms and conditions of commercial sale: *http://www.nxp.com/about/about-nxp/our-terms-and-conditions-of-commercial-sale:TERMSCONDITIONSSALE*

Figure 2. Disclaimer

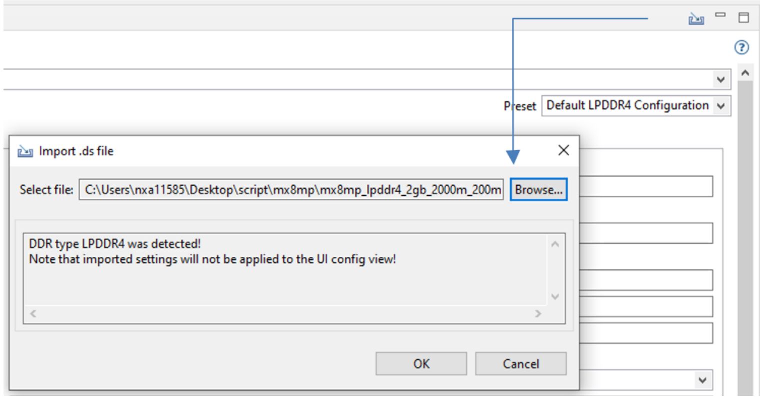
Chapter 3
DDR configuration

The DDR configuration provides a user-friendly graphical interface to configure the DDR interface and other associated subsystems. You can use it to change the DDR controller and PHY configuration when a different memory module is used to the configuration and to optimize the parameters associated with signal integrity.

3.1 Import *.ds file

Use import of *.ds file to load the initialization script provided by the **Register Programming Aid (RPA)** tool and bypass the UI configuration. To obtain the latest RPAs, refer to the following link on NPX community.

1. To import the RPA initialization script, use the **Import *.ds file** button and browse for the desired *.ds file

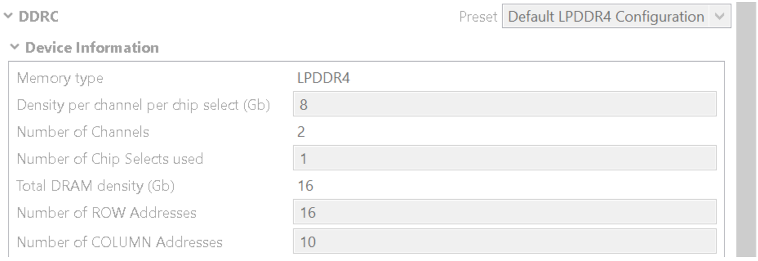


Select file: C:\Users\nxa11585\Desktop\script\mx8mp\mx8mp_lpddr4_2gb_2000m_200m

DDR type LPDDR4 was detected!
Note that imported settings will not be applied to the UI config view!

Figure 3. Import RPA script

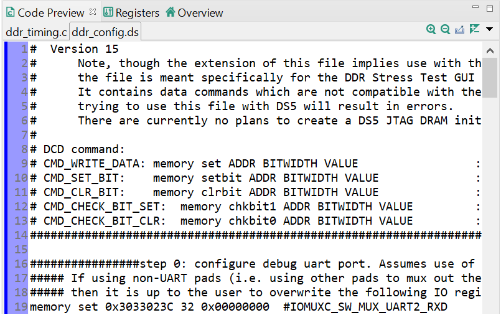
2. To load the *.ds file and disable the UI configuration interface, press OK.



Memory type	LPDDR4
Density per channel per chip select (Gb)	8
Number of Channels	2
Number of Chip Selects used	1
Total DRAM density (Gb)	16
Number of ROW Addresses	16
Number of COLUMN Addresses	10

Figure 4. Disabled UI

3. The contents of the imported *.ds file is shown in **Code Preview**, *ddr_config.ds*

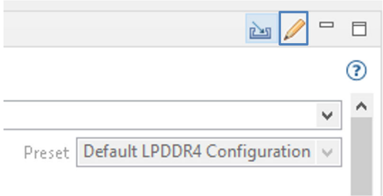


```
#  Version 15
#      Note, though the extension of this file implies use with th
#      the file is meant specifically for the DDR Stress Test GUI
#      It contains data commands which are not compatible with the
#      trying to use this file with DS5 will result in errors.
#      There are currently no plans to create a DS5 JTAG DRAM init
#
# DCD command:
# CMD_WRITE_DATA: memory set ADDR BITWIDTH VALUE                 :
# CMD_SET_BIT:    memory setbit ADDR BITWIDTH VALUE              :
# CMD_CLR_BIT:    memory clrbit ADDR BITWIDTH VALUE              :
# CMD_CHECK_BIT_SET:  memory chkbit1 ADDR BITWIDTH VALUE         :
# CMD_CHECK_BIT_CLR:  memory chkbit0 ADDR BITWIDTH VALUE         :
##################################################################

################step 0: configure debug uart port. Assumes use of
##### If using non-UART pads (i.e. using other pads to mux out the
##### then it is up to the user to overwrite the following IO regi
memory set 0x3033023C 32 0x00000000  #IOMUXC_SW_MUX_UART2_RXD
```

Figure 5. RPA script content

4. To switch back to UI configuration, press the button "**Enable manual config**".



Preset: Default LPDDR4 Configuration

Figure 6. Enable manual config

3.2 UI configuration

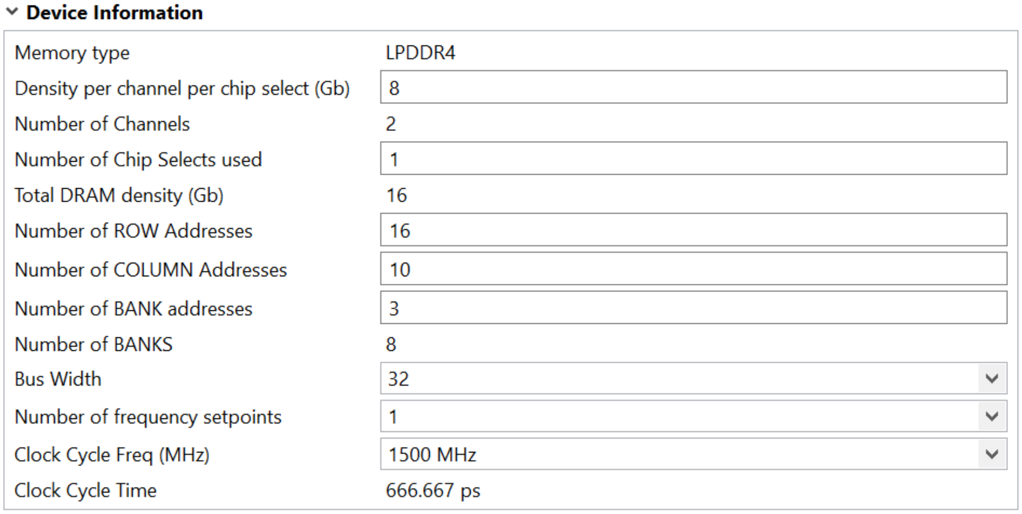
The UI configuration allows you to change manually Device Information, PHY options, or Design-specific configuration. There are two modes available, **Basic** and **Advanced.**

NOTE

Advanced mode is only recommended for experienced users.

Basic mode allows you to configure the parameters that are design-dependent.

1. Device information

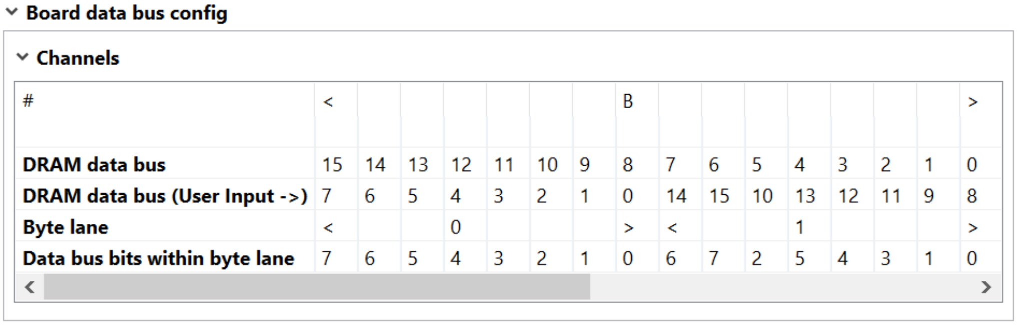


Device Information

Memory type	LPDDR4
Density per channel per chip select (Gb)	8
Number of Channels	2
Number of Chip Selects used	1
Total DRAM density (Gb)	16
Number of ROW Addresses	16
Number of COLUMN Addresses	10
Number of BANK addresses	3
Number of BANKS	8
Bus Width	32
Number of frequency setpoints	1
Clock Cycle Freq (MHz)	1500 MHz
Clock Cycle Time	666.667 ps

Figure 7. Device Information UI

2. Board data bus configuration for LPDDR4



#	<							B								>
DRAM data bus	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRAM data bus (User Input ->)	7	6	5	4	3	2	1	0	14	15	10	13	12	11	9	8
Byte lane	<			0				>	<			1				>
Data bus bits within byte lane	7	6	5	4	3	2	1	0	6	7	2	5	4	3	1	0

Figure 8. Board data bus config UI

3. UART port selection

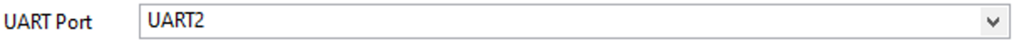

UART Port: UART2

Figure 9. UART selection UI

4. DBI selection (for LPDDR4)

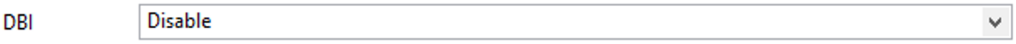

DBI: Disable

Figure 10. DBI selection

Advanced mode allows you to configure additional parameters.

1. The firmware version is the one officially supported by the BSP, but the **DDR tool** offers the possibility to select between multiple versions.

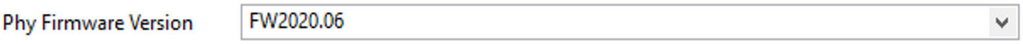

Phy Firmware Version: FW2020.06

Figure 11. Firmware version selection UI

NOTE

Use only the Firmware version for your specific SoC and BSP GA version. Not all Firmware versions are supported for each SOC.

2. PHY log level selection

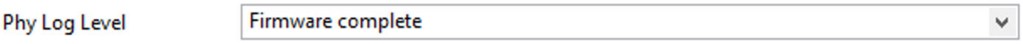
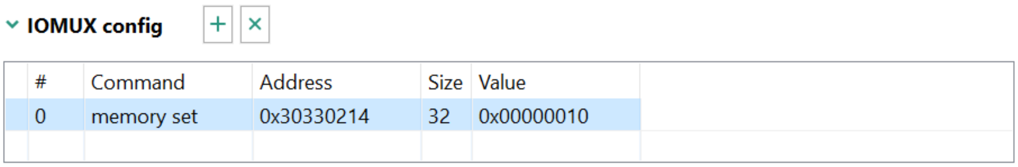

Phy Log Level: Firmware complete

Figure 12. PHY log level selection

3. IOMUX configuration

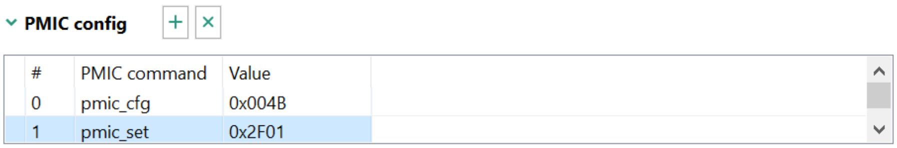


IOMUX config

#	Command	Address	Size	Value
0	memory set	0x30330214	32	0x00000010

Figure 13. IOMUX configuration UI

4. PMIC configuration sequence

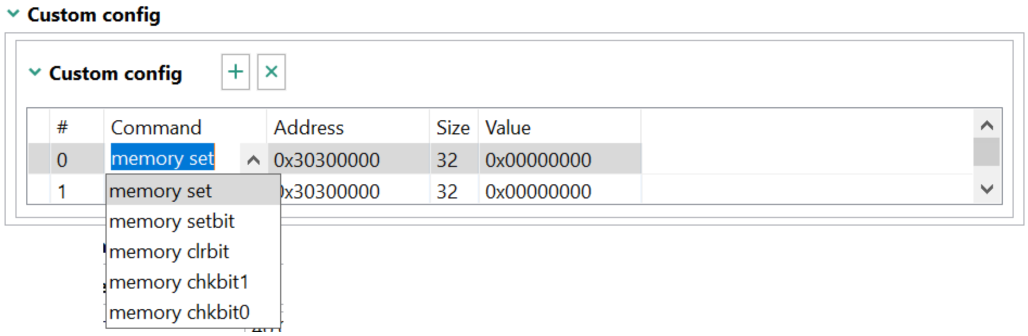


PMIC config

#	PMIC command	Value
0	pmic_cfg	0x004B
1	pmic_set	0x2F01

Figure 14. PMIC configuration UI

5. Custom configuration



Custom config

Custom config

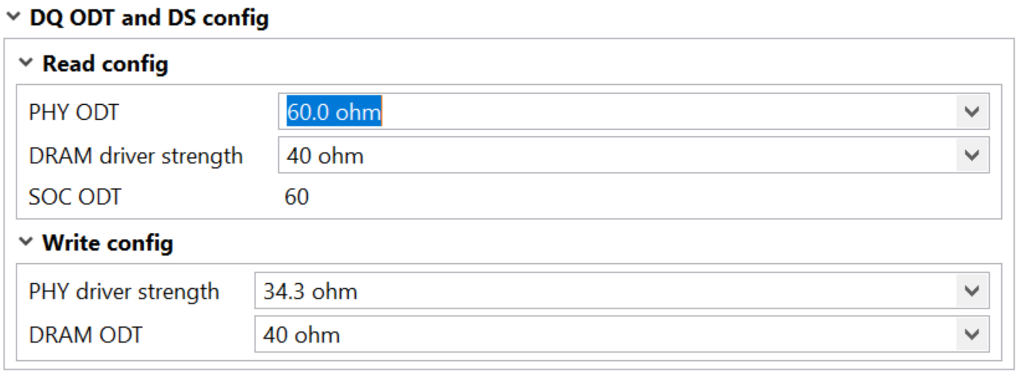
#	Command	Address	Size	Value
0	memory set	0x30300000	32	0x00000000
1	memory set	0x30300000	32	0x00000000

Figure 15. Custom configuration UI

NOTE

Any write to an incorrect address may cause unexpected behavior.

6. DQ ODT and DS configuration



DQ ODT and DS config

Read config

PHY ODT: 60.0 ohm

DRAM driver strength: 40 ohm

SOC ODT: 60

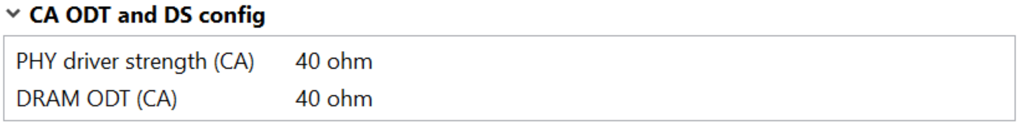
Write config

PHY driver strength: 34.3 ohm

DRAM ODT: 40 ohm

Figure 16. DQ ODT and DS configuration UI

7. CA ODT and DS configuration – Informational Only



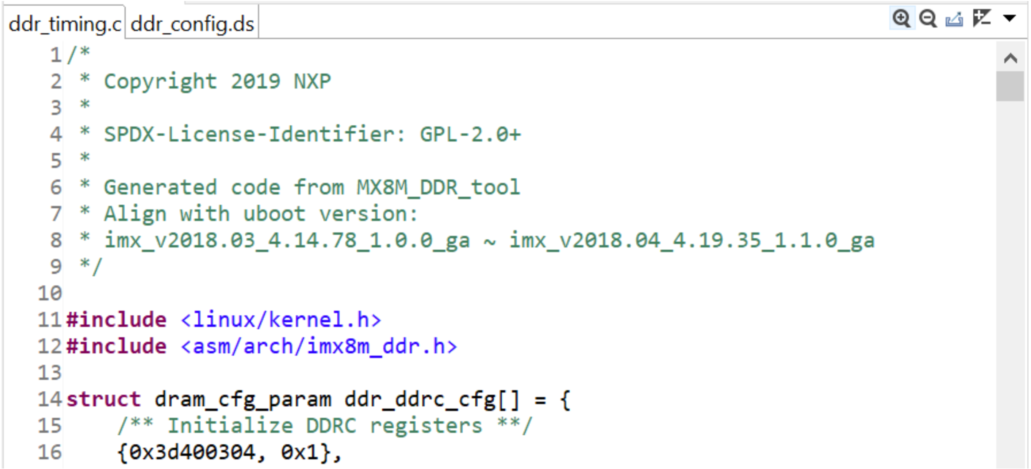
PHY driver strength (CA)	40 ohm
DRAM ODT (CA)	40 ohm

Figure 17. CA ODT and DS configuration UI

3.3 Code generation

You can generate the configuration as C code in the **Code Preview View**, which can be used by the U-Boot SPL driver.

You can trigger code generation by any change in the GUI, it is highlighted in the **Code Preview**.



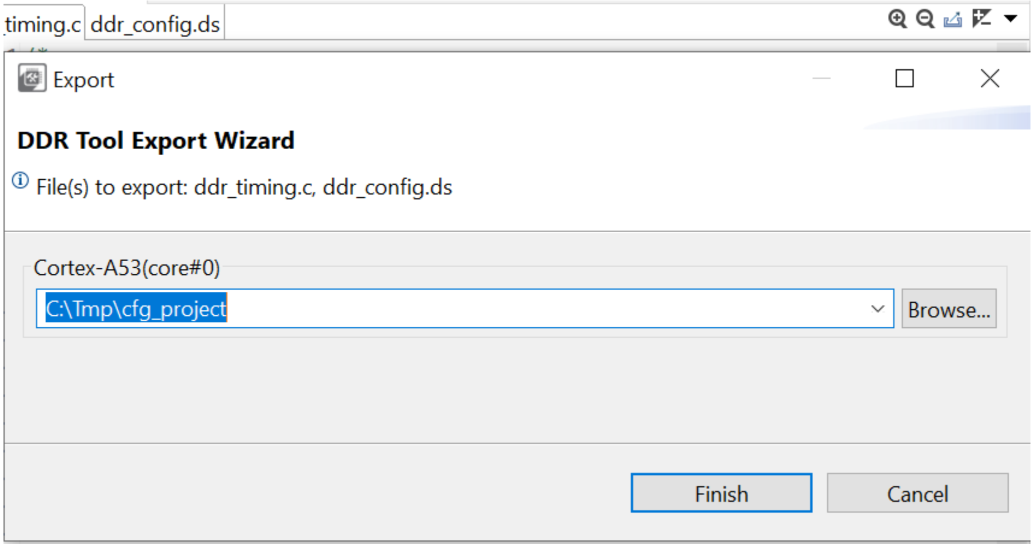
```
/*
 * Copyright 2019 NXP
 *
 * SPDX-License-Identifier: GPL-2.0+
 *
 * Generated code from MX8M_DDR_tool
 * Align with uboot version:
 * imx_v2018.03_4.14.78_1.0.0_ga ~ imx_v2018.04_4.19.35_1.1.0_ga
 */

#include <linux/kernel.h>
#include <asm/arch/imx8m_ddr.h>

struct dram_cfg_param ddr_ddrc_cfg[] = {
    /** Initialize DDRC registers **/
    {0x3d400304, 0x1},
```

Figure 18. Code Preview

You can save files from **Code Preview** on the disk by using **DDR tool Export Wizard**.



ⓘ File(s) to export: ddr_timing.c, ddr_config.ds

Cortex-A53(core#0)

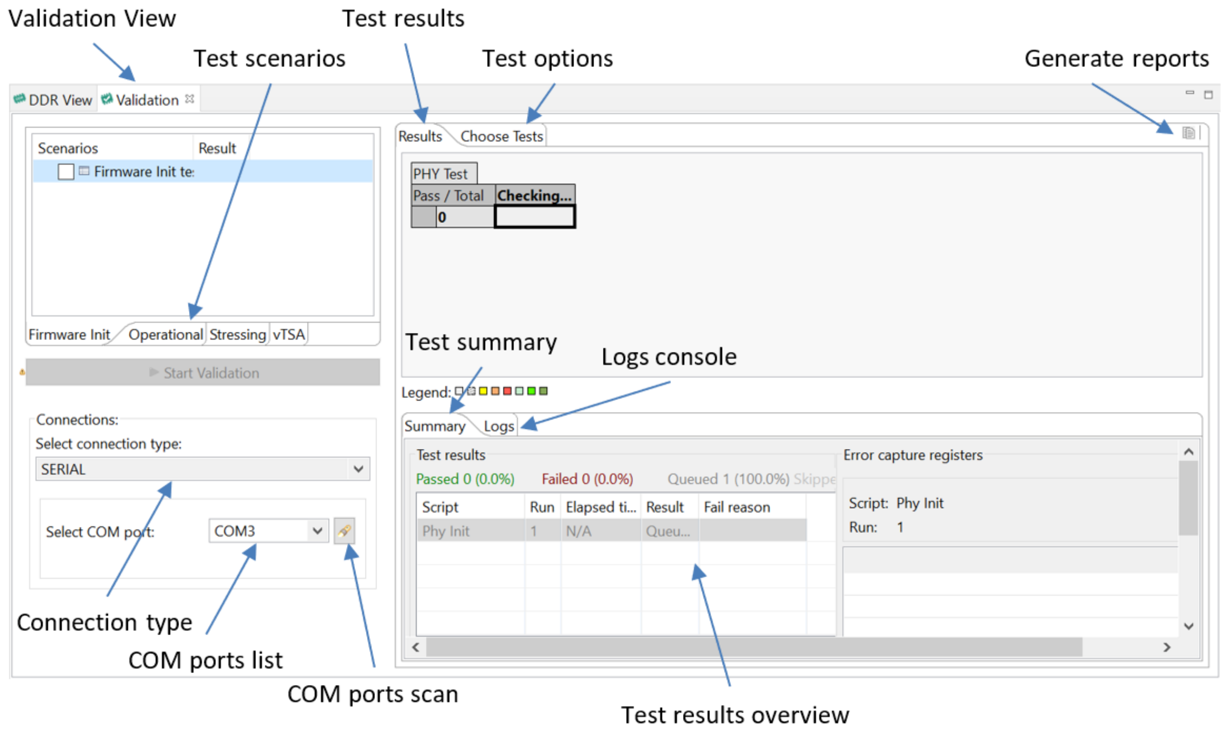
C:\Tmp\cfg_project

Figure 19. Export generated files

Chapter 4
DDR validation

DDR validation uses different scenarios to assess DDR performance by downloading a test image to the internal RAM of the processor in the Serial Download mode. Results are sent to the DDR tool via UART.

DDR evaluation can help you to assess the stability of the DDR interface on the board in a non-OS environment.



PHY Test	
Pass / Total	Checking...
0	

Connections:

Select connection type: SERIAL

Select COM port: COM3

Test results

Passed 0 (0.0%) Failed 0 (0.0%) Queued 1 (100.0%) Skippe

Script	Run	Elapsed ti...	Result	Fail reason
Phy Init	1	N/A	Queu...	

Error capture registers

Script: Phy Init

Run: 1

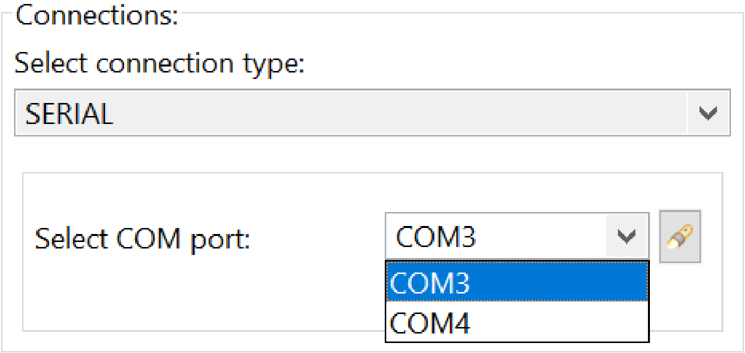
Figure 20. DDR validation

4.1 Connection

To connect to a board, you must do the following:

1. Configure the board to boot in the Serial Download mode/Manufacture mode and power up the board.
2. Connect a UART cable from the host computer to the UART of the A-core on the board.
3. Connect a USB cable from the host computer to the USB port on the board that is used by the Serial Download mode. An “HID-compliant device” or a “USB Input Device” is shown in Windows Device Manager.

After the board is connected to the host computer, you should search the UART ports by using **COM port scan**. COM port drop list is populated with all the available UART ports.



Connections:

Select connection type: SERIAL

Select COM port: COM3

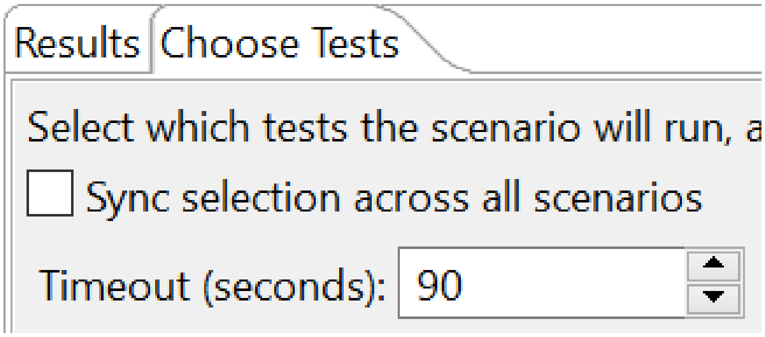
Figure 21. COM port selection

Choose the correct UART that is used as the A-core debug UART port.

4.2 Test scenarios

Once the DDR configuration and board connection are set up, you can execute different **Test scenarios**. You can customize each test by setting the parameters from **Test options**.

Depending on the test and options selected, the execution time may differ. By default a 90 seconds timeout is set, to assure that in case of an issue the test finishes. To change the default value, edit the **Timeout (seconds)** option:



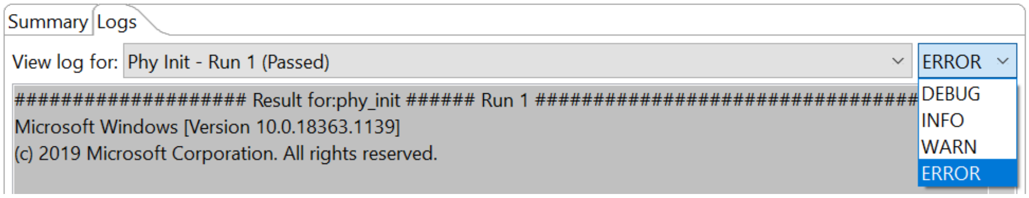
Select which tests the scenario will run, a

□ Sync selection across all scenarios

Timeout (seconds): 90

Figure 22. Timeout selection

To start test execution, press the button "**Start Validation**". You can check the status of the running test from the **Logs** console. By default, the log level is set to **ERROR**. Additional log-level options are available, with different output in the console:



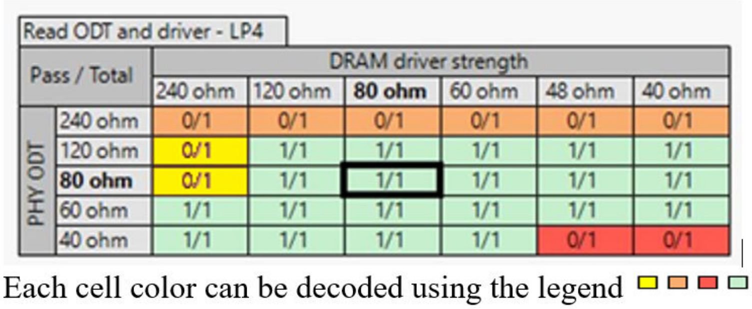
View log for: Phy Init - Run 1 (Passed)

ERROR

```
################### Result for:phy_init ###### Run 1 #################################
Microsoft Windows [Version 10.0.18363.1139]
(c) 2019 Microsoft Corporation. All rights reserved.
```

Figure 23. Log level selection

At the end of the test, the PASS/FAIL status is displayed in "**Results**". The test summary is displayed in "**Summary**".



Read ODT and driver - LP4

Pass / Total		DRAM driver strength					
		240 ohm	120 ohm	**80 ohm**	60 ohm	48 ohm	40 ohm
PHY ODT	240 ohm	0/1	0/1	0/1	0/1	0/1	0/1
	120 ohm	0/1	1/1	1/1	1/1	1/1	1/1
	80 ohm	0/1	1/1	1/1	1/1	1/1	1/1
	60 ohm	1/1	1/1	1/1	1/1	1/1	1/1
	40 ohm	1/1	1/1	1/1	1/1	0/1	0/1

Each cell color can be decoded using the legend 🟨 🟧 🟥 🟩

Figure 24. Test status

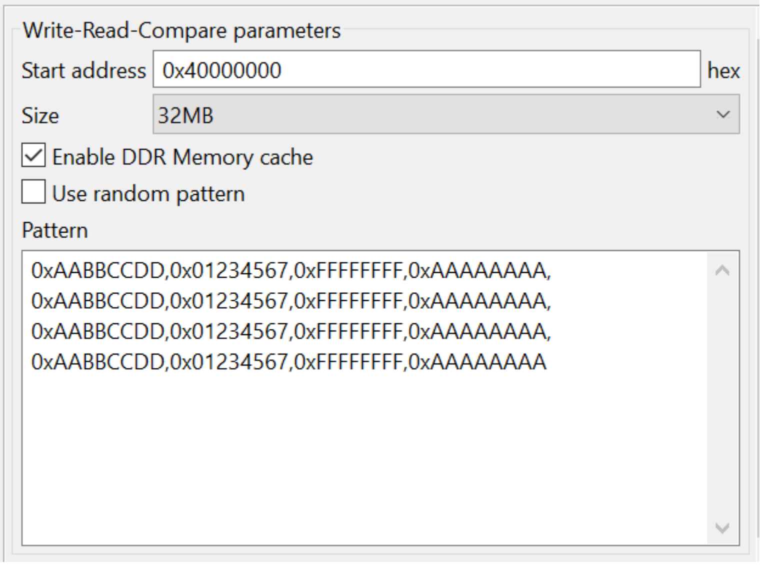
- Yellow is for *Test failed*
- Orange is for *Configuration error*
- Red is for *Target connection error or exception in the script*
- Green is for *Test passed*

The DDR tool offers several test scenarios that can be split into Inspection, Optimization, vTSA, and Stressing.

4.2.1 Inspection

Inspection shows the status of the DDR Controller and DDR PHY configuration, by executing following tests:

1. *Firmware Init* – executes the DDR PHY training to check the DDR PHY configuration.
2. *Operational* – performs basic memory access test by running **Write-Read-Compare/ Walking Ones/ Walking Zeros** tests. Such options as Start Address, Size, Enable DDR Memory cache, Access mode/Pattern option are available for each test.



Write-Read-Compare parameters

Start address: 0x40000000 hex

Size: 32MB

☑ Enable DDR Memory cache

☐ Use random pattern

Pattern

```
0xAABBCCDD,0x01234567,0xFFFFFFFF,0xAAAAAAAA,
0xAABBCCDD,0x01234567,0xFFFFFFFF,0xAAAAAAAA,
0xAABBCCDD,0x01234567,0xFFFFFFFF,0xAAAAAAAA,
0xAABBCCDD,0x01234567,0xFFFFFFFF,0xAAAAAAAA
```

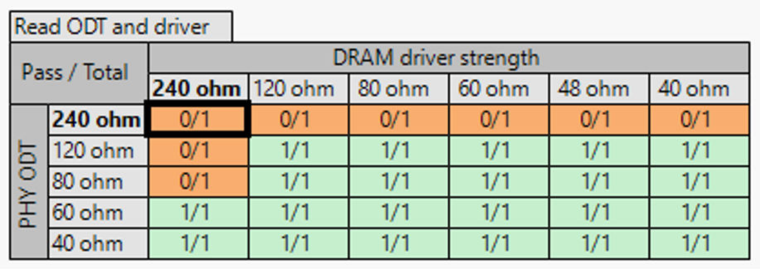
Figure 25. Test options

4.2.2 Optimization

DQ ODT and driver strength tests sweep the DQ IO configurations to create board-specific Driver Strength vs. ODT PASS/FAIL map for the Reads and the Writes.

NOTE

Optimization is not available when UI configuration is bypassed by RPA initialization script import.

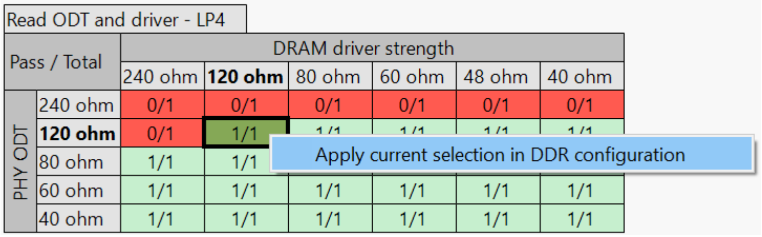


Read ODT and driver

Pass / Total		DRAM driver strength					
		240 ohm	120 ohm	80 ohm	60 ohm	48 ohm	40 ohm
PHY ODT	**240 ohm**	0/1	0/1	0/1	0/1	0/1	0/1
	120 ohm	0/1	1/1	1/1	1/1	1/1	1/1
	80 ohm	0/1	1/1	1/1	1/1	1/1	1/1
	60 ohm	1/1	1/1	1/1	1/1	1/1	1/1
	40 ohm	1/1	1/1	1/1	1/1	1/1	1/1

Figure 26. DQ ODT and DS map

For passing cells (Green cells), enable the option *Apply current selection in DDR configuration* (right click on the cell). It sets the respective Driver Strength and ODT value into the configuration for use in other scenarios.



Read ODT and driver - LP4

Pass / Total		DRAM driver strength					
		240 ohm	**120 ohm**	80 ohm	60 ohm	48 ohm	40 ohm
PHY ODT	240 ohm	0/1	0/1	0/1	0/1	0/1	0/1
	120 ohm	0/1	1/1	1/1	1/1	1/1	1/1
	80 ohm	1/1	1/1				
	60 ohm	1/1	1/1	1/1	1/1	1/1	1/1
	40 ohm	1/1	1/1	1/1	1/1	1/1	1/1

Figure 27. Apply DQ ODT and DS configuration

NOTE

You can use the Driver Strength vs. ODT map as one of the criteria when deriving optimal ODT/Driver Strength values. This map cannot serve as all-comprising output to make this determination.

NOTE

NXP strongly recommends using the default ODT and Drive strength values that are tested and validated as part of our GA BSP. To ensure that your design adheres to the board layout requirements, refer to the device i.MX 8M Hardware Developer's Guide.

4.2.3 vTSA

vTSA performs Virtual Timing Signal Analysis by running **Diag Write Margin/ Diag Read Margin** with virtual eye diagram displayed as output.

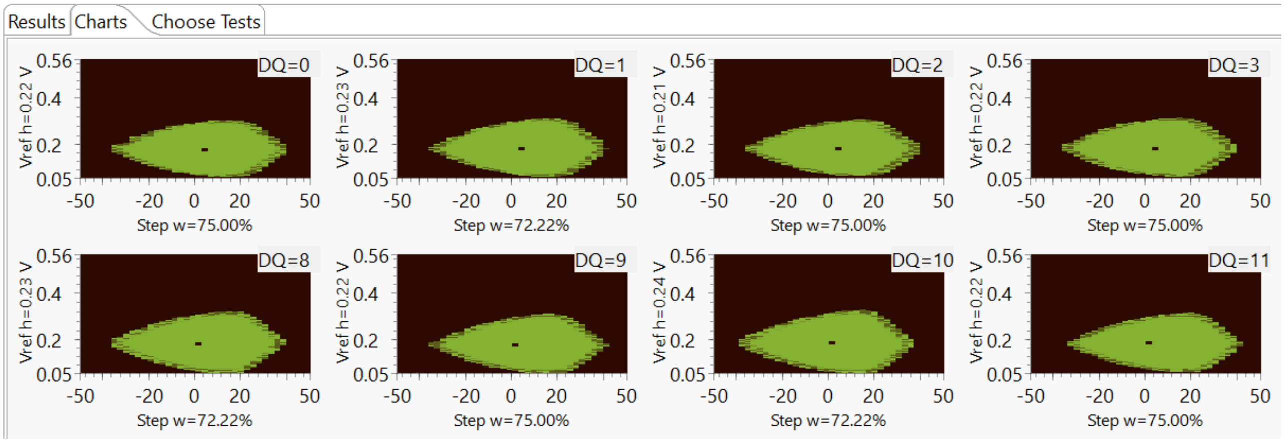


Figure 28. Diagnostic data eyes

NOTE

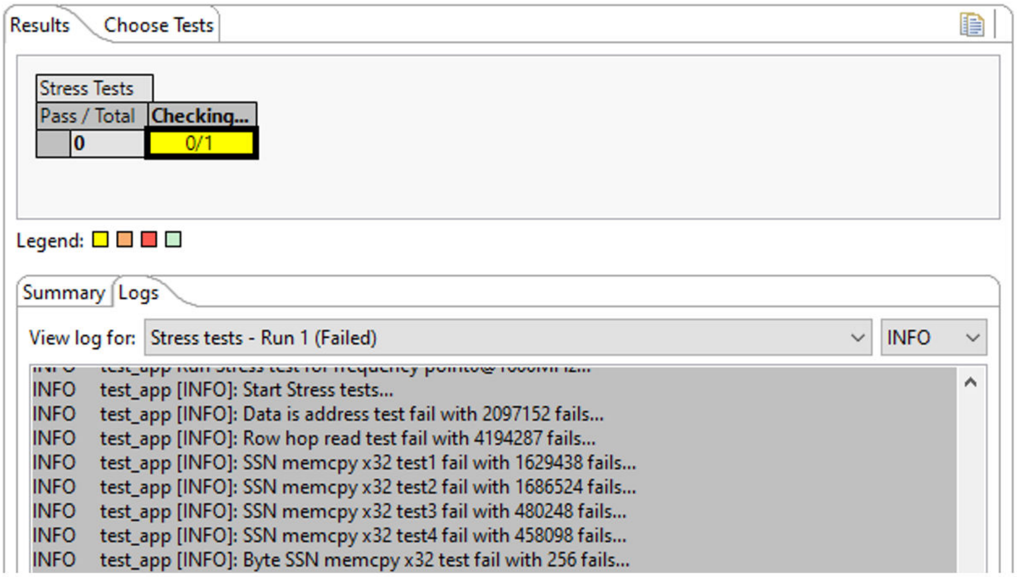
Details about vTSA are provided in FAQ section.

4.2.4 Stressing

To test the stability of the DDR configuration more extensively, you can use the *Stressing* scenario, with its suite of tests that covers different situations.

Two ways of running *Stress* tests are available:

1. **Single run** runs the test suite one time with different options selected (Size, Enable DDR Memory cache, Stop on fail). In case of failure, you can check the status of each test in the suite in the **Logs** console, with Log level set to **INFO**



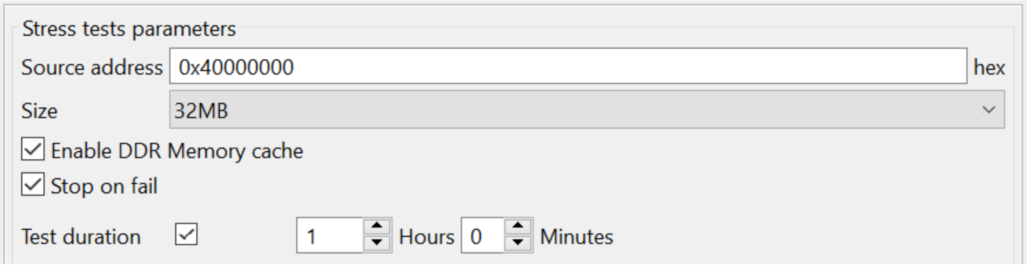
Stress Tests	
Pass / Total	Checking...
0	0/1

View log for: Stress tests - Run 1 (Failed) INFO

```
INFO    test_app Run Stress test for frequency point0@1600MHz...
INFO    test_app [INFO]: Start Stress tests...
INFO    test_app [INFO]: Data is address test fail with 2097152 fails...
INFO    test_app [INFO]: Row hop read test fail with 4194287 fails...
INFO    test_app [INFO]: SSN memcpy x32 test1 fail with 1629438 fails...
INFO    test_app [INFO]: SSN memcpy x32 test2 fail with 1686524 fails...
INFO    test_app [INFO]: SSN memcpy x32 test3 fail with 480248 fails...
INFO    test_app [INFO]: SSN memcpy x32 test4 fail with 458098 fails...
INFO    test_app [INFO]: Byte SSN memcpy x32 test fail with 256 fails...
```

Figure 29. Stress tests

2. **Test duration** runs the test suite for a selected time. This is suitable for overnight tests.



Stress tests parameters

Source address: 0x40000000 hex

Size: 32MB

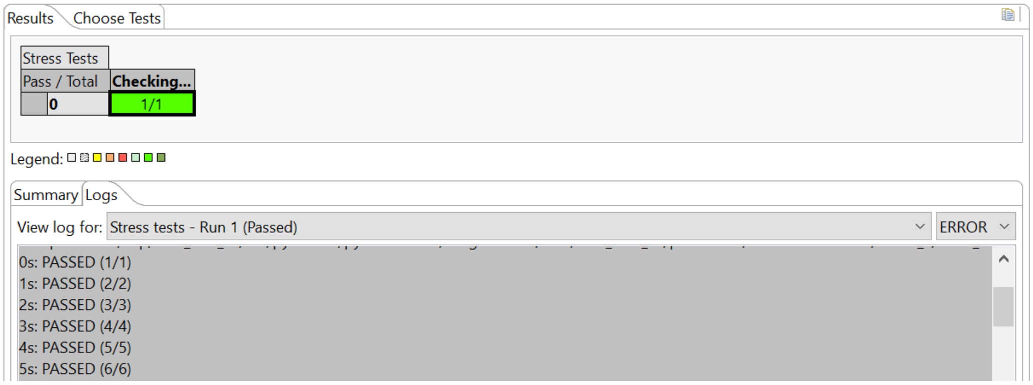
☑ Enable DDR Memory cache

☑ Stop on fail

Test duration ☑ 1 Hours 0 Minutes

Figure 30. Stress tests options

In the **Logs** console, you can monitor test execution and see the number of iterations and the duration.



Stress Tests	
Pass / Total	**Checking...**
0	1/1

Legend: □ ▨ 🟨 🟧 🟥 🟩 🟩 🟩

View log for: Stress tests - Run 1 (Passed) ERROR

```
0s: PASSED (1/1)
1s: PASSED (2/2)
2s: PASSED (3/3)
3s: PASSED (4/4)
4s: PASSED (5/5)
5s: PASSED (6/6)
```

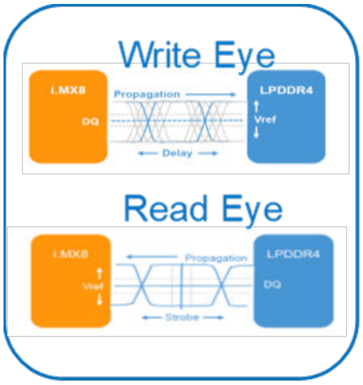
Figure 31. Stress tests results

NOTE

Make sure the **Timeout (seconds)** setting is higher than the**Test duration** setting, otherwise the test ends with timeout.

Chapter 5
FAQ

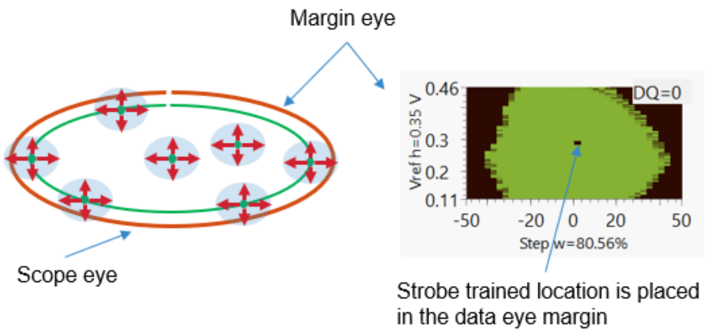
1. What does vTSA mean?
    a. vTSA is an abbreviation for Virtual Timing Signal Analysis.
    b. A “virtual” TSA uses the memory controller itself to test margins without test equipment. “Virtual” does not mean simulation!
    c. Memory controllers have the ability the alter timings, voltage references, termination settings, and so on, for both incoming and outgoing signals.
    d. “Training” is a process when the memory controller sweeps these parameters and finds the configuration with the most margin for operation.
    e. A vTSA simply logs this information for output, which provides insight into the signaling margin of the system without the need for test equipment.
    f. Initialization and calibration settings can be dumped to a file for analysis as well.
2. What is the vTSA output?
    a. Virtual Timing Signal Analysis(vTSA) provides write and read data eye diagrams virtually by running a series of write/read transactions as opposed to the hardware method of using a high-speed oscilloscope to perform manual physical TSA (pTSA) measurements.
    b. This being the case, vTSA output itself approximates the actual write/read eyes.
    c. You should expect some variation between trained values of delay lines and VREF in comparison to the vTSA report of these values.
    d. vTSA only reports the values it detects in the “widest” part of the reported eye, which may itself vary from run-to-run.
    e. The key takeaway from using this tool is to display the virtual write/read eyes to convince the user of the robustness of their design.



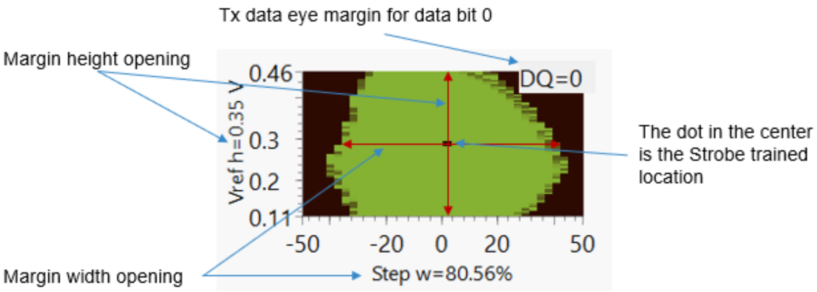
3. Need more information about vTSA?
    a. The vTSA tool is an approximation of an actual pTSA. Thus, you may note some variation between the trained delay line value and the vTSA “mid” value. It is observed and expected that this variation may be up to ~20 ps. The key takeaway from the generated eye diagrams should be focused on verifying the ample margin of the trained delay line value within the data eye.
    b. For the trained VREF value, the LPDDR4 device Mode Register 14 (MR14 which holds the trained VREF value) applies to ALL byte lanes. In other words per JEDEC, there is not an MR14 per byte lane and instead, MR14 applies to all byte lanes.

   c. For a board that follows the NXP DDR layout guidelines, there should be plenty of margin around the VREF trained value. You can find the guidelines in the respective NXP Hardware Developer Guide.

4. How a data eye margin is generated?
   a. There are several delay steps available to shift each DQ and DQS.
   b. As DQ crosses the unit interval, from zero-step delay to 1 unit internal step delay, each step is tested with a write-read-compare test to determine pass or fail.
   c. The DQ traverse of the unit interval is repeated for all available VREF steps.
   d. Delay steps generate a line, and repeating the lines at each VREF step generates data eye margin.
   e. The crossing of DQS signal with trained VREF and delay step is placed in the generated data eye margin.
   f. Each **passing** dot in the margin eye already meets the setup, hold, and voltage requirement.



5. What represents the information next to the data eye?
   a. Unit interval = 1/data rate; for example, at 3200MT/s data rate the unit interval = 312.5 ps
   b. The x-axis displays the time. It is one unit interval in percentage. -50 % to +50 %
   c. The x-axis data eye margin width opening is displayed as the percentage of one unit interval. For example: Step w=80.56 % of UI
   d. The y-axis displays the voltage.
   e. The y-axis open data eye margin height/amplitude opening is displayed in voltage. Ex: Vref h=0.35 V



6. How much margin is considered as good?
   a. To determine the required margin mask, you must do the following:
      - Optimize The DDR interface
      - Settings have been optimized, generate the worst-case data eye margin using the worst-case conditions (temperature, voltage, frequency, pattern) for a customer board DDR interface.

- Use the DDR training optimizing/centering the strobe to the eye margin

b. A green pixel in the data eye margin indicates a passing cell. It means for that green pixel the setup and hold time as well as the VIHLac/dc are satisfactory.

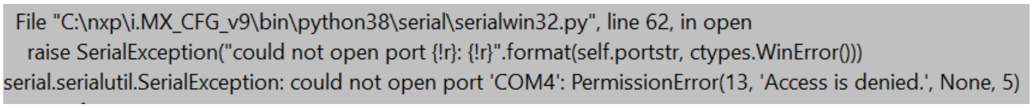
c. Any additional green pixels around the strobe location in the data eye margin are additional margin available to DDR for that DQ.

7. Why is the trained Vref sometimes not in the exact center of the eye?

   a. The VREF training must select a value that corresponds to all of the byte lanes' passing VREF window and then program this value into the LPDDR4 MR14 register. It means that for a one-byte lane, though the trained VREF value may not seem to be in the exact center of the data eye, it is selected to provide the best possible margin for this byte lane along with satisfying the other byte lanes.

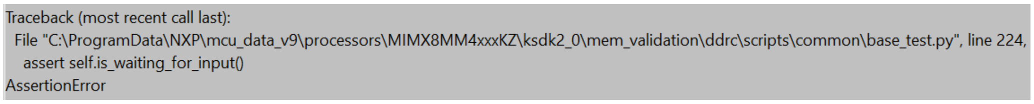
8. How to check that wrong UART is selected?

   Set the Log Level to DEBUG and check the messages from console



```
  File "C:\nxp\i.MX_CFG_v9\bin\python38\serial\serialwin32.py", line 62, in open
    raise SerialException("could not open port {!r}: {!r}".format(self.portstr, ctypes.WinError()))
serial.serialutil.SerialException: could not open port 'COM4': PermissionError(13, 'Access is denied.', None, 5)
```

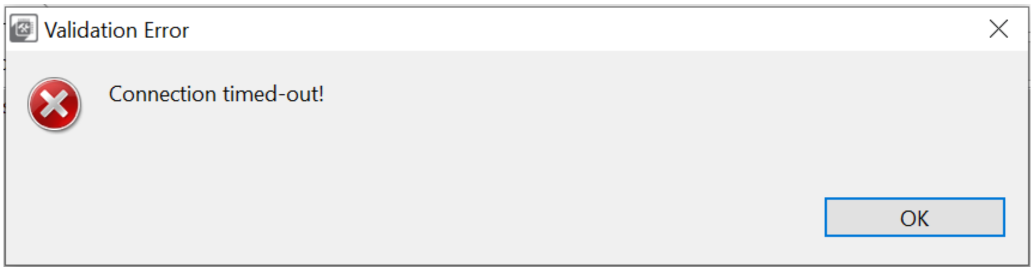
   or



```
Traceback (most recent call last):
  File "C:\ProgramData\NXP\mcu_data_v9\processors\MIMX8MM4xxxKZ\ksdk2_0\mem_validation\ddrc\scripts\common\base_test.py", line 224,
    assert self.is_waiting_for_input()
AssertionError
```

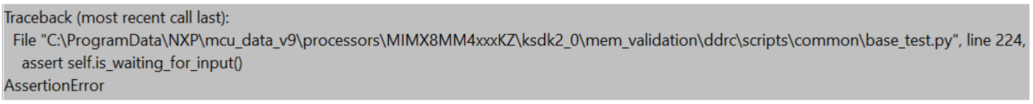
9. How to proceed in case of test timeout?

   a. In case of test timeout, the below pop-up window appears



Connection timed-out!

   b. Increase the timeout (second) option and rerun the test. If you get the below error, power off the board, unplug the UART cable, power on the board, plug in the UART cable



```
Traceback (most recent call last):
  File "C:\ProgramData\NXP\mcu_data_v9\processors\MIMX8MM4xxxKZ\ksdk2_0\mem_validation\ddrc\scripts\common\base_test.py", line 224,
    assert self.is_waiting_for_input()
AssertionError
```

Chapter 6
Revision history

Table 1. Revision history

Date	Revison number	Changes
11 August 2021	0	Initial version
21 December 2021	1	Screenshots are updated, section 4.3 is removed

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 21 December 2021
Document identifier: DDRTUG