AN5168

Application note

How to configure DDR on STM32MP1 MPUs

Introduction

This document describes the procedure and steps needed to configure the DDR subsystem (DDRSS) on STM32MP1 series MPUs. In this application note, the STM32MP13x devices belong to STM32MP131, STM32MP133 and STM32MP135 lines, the STM32MP15x belong to STM32MP151, STM32MP153, and STM32MP157 lines.

The DDRSS configuration is achieved by programming multiple parameters and settings in the DDR controller (DDRCTRL), the PHY interface (DDRPHYC), and the SDRAM mode registers. These parameters are determined according to the: DDR type, DDR size, SDRAM topology, runtime frequency, and the SDRAM device datasheet parameters. All these parameters must be programmed during the initialization sequence.

The STM32CubeMX DDR test suite uses intuitive panels and menus. These hide the complexity associated with correct parameter determination and initialization launching (refer to [6]). The configuration requires very few inputs from the user to set up DDRSS quickly to run. Some advanced user modes and specific features may be used if they are important for the application.

During the bring-up phase, the user can run extensive test suites. These tests can be launched using STM32CubeMX and are used to verify the robustness of the DDR configuration.

References

[1]	STM32MP1 series reference manuals and datasheets
[2]	JEDEC JESD79-3F DDR3 SDRAM standard
[3]	JEDEC JESD209-2F LPDDR2 SDRAM standard
[4]	JEDEC JESD209-3C LPDDR3 SDRAM standard
[5]	STM32MP1 series DDR memory routing guidelines (AN5122)
[6]	STM32CubeMX for STM32 configuration and initialization C code generation (UM1718)
[7]	DDR-PHY interface (DFI) (http://www.ddr-phy.org)

1 General information

This document applies to the STM32MP1 series Arm®-based MPUs.

Note: Arm is a registered trademark of Arm limited (or its subsidiaries) in the US and/or elsewhere.

2 DDR subsystem initialization and configuration

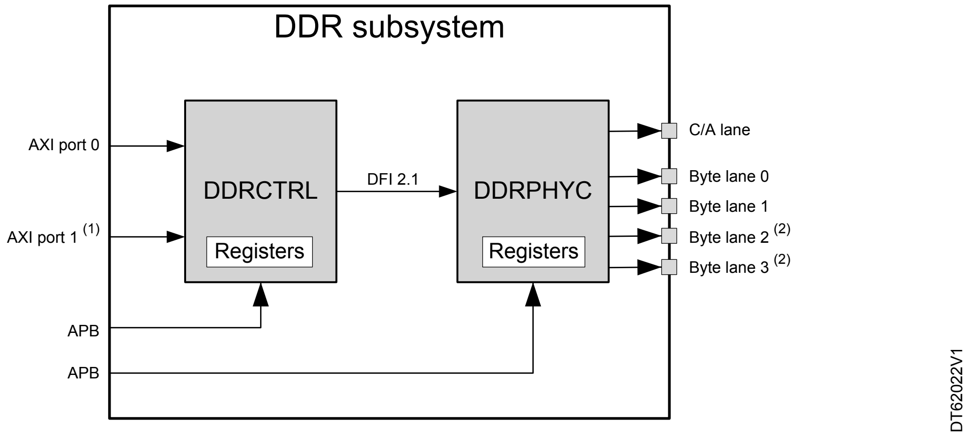
The DDR subsystem includes DDRCTRL and DDRPHYC (see the figure below).

DDRCTRL supports the DDR command scheduling during normal operation with scheduling of commands and refreshes.

DDRPHYC is a DDR PHY with DFI interface [7] to DDRCTL and a byte lane architecture, suitable to interface DDR3/3L and LPDDR2/3 up to 533 MHz.

DDRPHYC fully supports the DDR initialization with several PHY tuning options (built-in). DDRPHYC includes a BIST engine used to support software driven tuning.

Figure 1. DDR subsystem



1. *AXI port 1 is present only on STM32MP15x*
2. *The number of byte lanes depends on:*
   - *The product: STM32MP15x has a 32-bit interface. STM32MP13x has only a 16-bit interface.*
   - *The package options: refer to the product datasheet, the STM32MP13x packages are 16 bits.*

2.1 DDRSS and SDRAM initialization

Power, clock, and reset are internally sequenced by the device to respect the SDRAM power-up sequence.

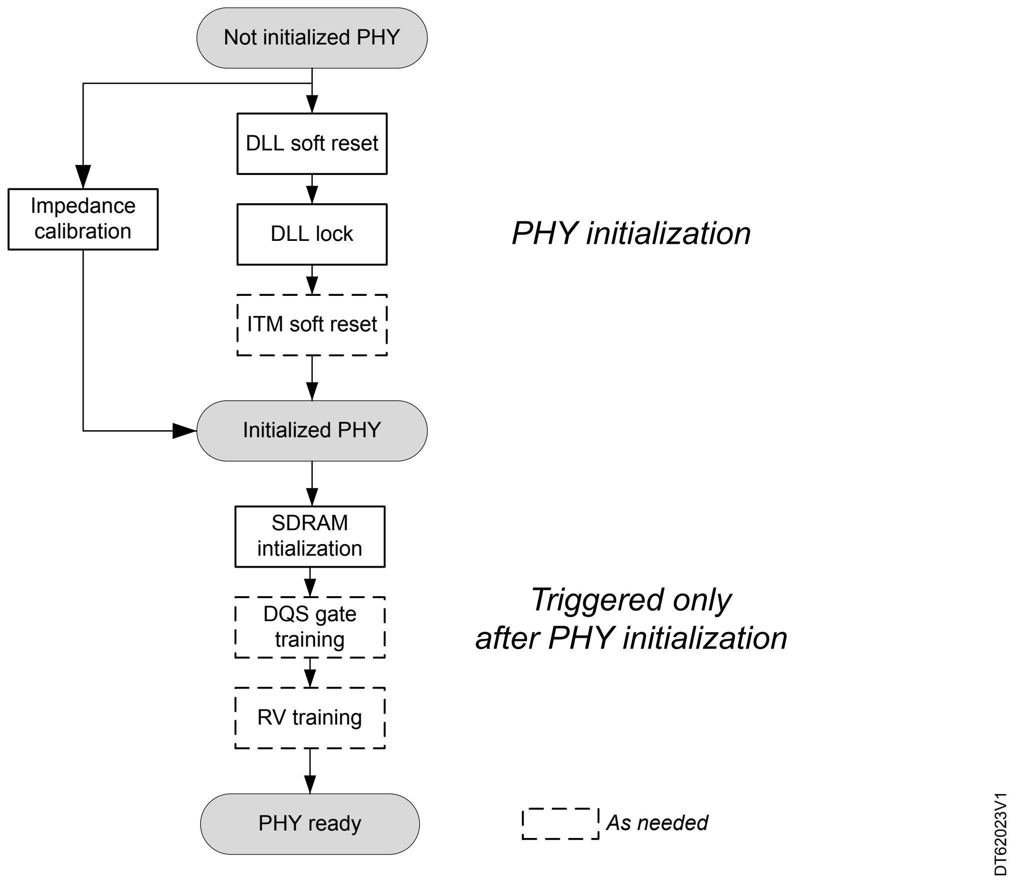
The PHY initialization sequence shown in Figure 2 is controlled by the DDRPHYC physical utility block (PUB). This PUB-based initialization sequence is launched after DDRPHYC is released from reset and sequenced according to the DDRPHYC.PIR register.

The initialization sequence includes the following steps:

1. DDRCTRL and DDRPHYC initialization including the reset of the internal timing modules (ITM) and the DLL reset and lock
2. DDRPHYC I/O initial calibration (ZCAL), launched in parallel with the DLL lock
3. SDRAM initialization with mode register write and calibration commands
4. Built-in DQS gate training (DQSTRN) and read valid training (RVTRN)

Note:
- *DDRPHYC I/O is set in SSTL mode for all SDRAMs.*
- *The built-in DQS gate training and read valid training are run at every initialization as recommended.*

Figure 2. **DDRPHYC initialization sequence**



2.2 DDRCTRL configuration

Most DDRCTRL registers are static and loaded before DDRCTRL is released from reset and AXI ports are enabled. This is supported by the DDR software driver.

The DDR subsystem is ready after both DDRPHYC and DDCTRL have been initialized, enabling the AXI port is the last step.

The configuration of DDRCTRL and DDRPHYC registers is determined with minimal information provided by menu/option selection from the STM32CubeMX DDR panel.

The DDRSS configuration is then saved and restored to DDRCTRL and DDRPHYC registers for subsequent reset and normal operations.

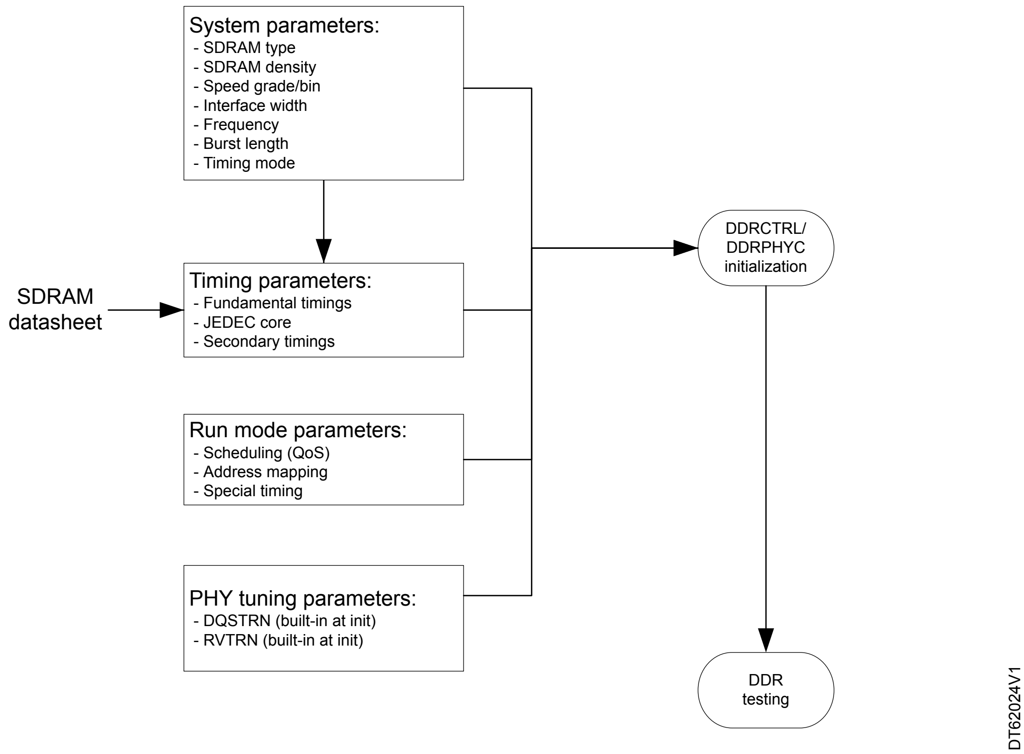
2.3 DDR configuration

The following steps are needed to configure the DDR:

1. Determine and program suitable values for DDRCTRL and DDRPHYC registers.
2. Launch the DDRCTRL and DDRPHYC initialization sequence.
3. Launch the DDR tuning with parameters save and restore (optional).
4. Launch the DDR testing (optional).

The overall configuration flow and parameters are highlighted in the figure below.

Figure 3. DDR initialization sequence



The configuration parameters are described according to types in Section 3: Configuration parameters.

2.4 DDR PHY tuning

The built-in DQS gate train (DQTRN) and read valid (training (RVTRN) are executed at DDRSS startup, by setting PIR[8] = 1. They are normally sufficient to guarantee correct operation at frequency ≤ 533 MHz. The PHY tunings (DQ bit deskew and Eye centering), as described in earlier versions of this application note, are no more required.

2.5 DDR testing

After DDR is configured and PHY is tuned for optimized timing, the configuration robustness must be checked by running an extensive sequence of tests launched from STM32CubeMX.

It is important to run all tests and levels proposed by STM32CubeMX during the bring-up phase before using the DDR.

The DDR testing and test flow with failure diagnostic and corrections are described in Section 8: DDR testing with STM32CubeMX.

The DDR subsystem bring-up is complete after the DDR configuration and successful stress-testing. The configuration parameters are saved for normal run mode initialization and DDR run mode.

3 Configuration parameters

The DDR configuration parameters, whether applicable to DDRCTRL or DDRPHYC, are classified into the groups listed below:

- **System parameters**: DDR type (DDR3/LPDDR2/3), DDR bus width (16-bit/32-bit), clock frequency, and density. The burst length and timing mode are determined by the system configuration and set by STM32CubeMX, presenting only a few required inputs to the user in the DDR configuration panel. STM32MP15x, used with 16-bit DDR is half populated. STM32MP15x, used with 32-bit, or STM32MP13x used with 16-bit are fully populated.
- **Timings parameters**, determined according to the DDR clock frequency and the SDRAM datasheet. This group includes the following parameters:
  - Fundamentals timings
  - JEDEC Core timings
  - Secondary timings
- **Run mode and special parameters**, related to performance scheduling, refresh timings, and address mappings. These parameters are selected from predefined sets as proposed to the user in the configuration panel. Some parameters and options are prompted by the STM32CubeMX DDR configuration (for example to set specific modes or extend temperature support).
- **PHY tuning parameters**, related to PHY timings determined during the initialization either using some PHY built-in sequence for DQSTRN and RVTRN.

3.1 System parameters

Most parameters are extracted from the STM32CubeMX pinout and clock configuration. The user options in the DDR panel allow the selection of the parameters listed below:

- DDR type (DDR3/3L, LPDDR2, or LPDDR3)
- Bus width (16-bit or 32-bit)
- DDR density (1, 2, 4, 8 Gbit/s)

The DDR interface ballout is updated accordingly by STM32CubeMX that highlights the physical balls to be connected to the SDRAM.

The DDR topology option is determined as follows:

- DDR3 32-bit is made with dual BGA 16 bits connected in fly-by topology with RTT termination.
- DDR3 16-bit is made with single BGA 16 bits connected in point to point.
- LPDDR2/3 are 16 bits or 32 bits (mostly 32) and connected in point to point.

3.2 Timing parameters

The timing parameters are established by STM32CubeMX based on the DDR frequency and the chosen SDRAM datasheet from the configuration panel menu. This process ensures accurate timing for the SDRAM interface.

These parameters are loaded into the DDRCTRL and DDRPHYC registers. DDRCTRL uses timing parameters for command scheduling. DDRPHYC uses several parameters during the SDRAM initialization sequence, mode register loading and to control the built-in DQS gate training (DQSTRN) sequence.

The DDR timings are classified in various groups detailed in the next sections.

3.2.1 Fundamental timings

The fundamental timings are reference to the standard timing triplet, for example, with DDR-1066.

The usual values are 8-8-8, 7-7-7 or 6-6-6.

3.2.2 JEDEC core timings

JEDEC core timings are specific to SDRAM type (refer to Section 5, Section 6, and Section 7 for more details on the specific SDRAM type). These timings are conservatively retrieved from typical datasheets or JEDEC tables and may be overridden according to the actual datasheet.

These parameters are used by DDRCTRL to constraint the command scheduling.

The two following timing modes are proposed during the configuration:

- Optimized timing mode where timings are directly computed according to the SDRAM datasheet and the clock frequency.
- Relaxed timing mode with all the critical core timings incremented by one clock. The objective is to improve timing margins in case of incertitude on exact datasheet parameters.

3.2.3 Secondary timing

DDRCTRL has a significant number of registers and bit fields to store for programming its timing behavior.

Several SDRAM specific secondary parameters (for example ODT timing position) are made programmable. Therefore, they are configured with default or predefined values that must not be modified.

The secondary parameters configured by STM32CubeMX are listed below:

- Refresh control parameters
- ZQ control
- ODT control
- Initialization timings (default values used for each DDR type)

3.3 Run mode parameters

These parameters are used to control the DDRCTRL features and optimization techniques listed in the next sections.

3.3.1 Low-power mode settings

The low-power features are controlled by DDRCTRL (for example, automatic self-refresh entry, see [1] for more details). The low-power modes are also controlled from the RCC (reset and clock control) or from the software (software self-reset). Refer to [1] for more details.

3.3.2 Address mapping

DDRCTRL features a flexible address mapper to convert the AXI bus address to SDRAM rows, banks, and columns.

The two following configurations are predetermined by STM32CubeMX:

- B/R/C (banks/rows/columns) with lowest power
- R/B/C (rows/banks/columns), the most conventional

Note: The DDRCTRL flexible address mapper allows the mapping of individual R/B/C bits into almost any sequence. The determination of the corresponding ADDRMAP registers is nontrivial. Refer to the DDRCTRL section in [1] for more details.

3.3.3 QoS (quality of service) and scheduling

DDRCTRL is scheduling the DDR commands to maximize their use by grouping read/write commands and scheduling bank activations from incoming commands posted in a 16-entry read-command buffer and a 16-entry write-command buffer. These buffers are made of content addressable memory (CAM).

DDRCTRL uses an advanced multitier arbitration scheme for an optimal DDR use and low latency according to programmed settings.

DDRCTRL features a rich set or arbitration policies and optimization techniques, such as page open/close policy, traffic mapping into classes and timeouts or scheduler and port arbitration control.

The command arbitration is determined according to traffic classes and timeout settings for some traffic classes.

DDRCTRL maps the incoming AXI port transactions into traffic classes with priorities and/or timeout, programmable for each AXI port.

For read, there are up to three classes listed below:

- HPR: allocated store with high priority, suitable to latency sensitive traffic
- VPR: highest priority on timeout expiration, suitable for latency critical traffic
- LPR: lower priority, suitable for best effort traffic

For write, there are two classes listed below:

- VPW: highest priority on timeout expiration, suitable for latency critical traffic
- NPW: normal priority, suitable for best effort traffic

STM32CubeMX proposes predefined configurations aligned with port allocation and QoS settings of the AXI interconnect (refer to [1] for more details).

STM32MP15x and STM32MP13x use a common QoS strategy except that:

- STM32MP15x feature 2 AXI ports and STM32MP13x feature a single AXI port with all controller assigned to it.
- STM32MP15x feature dual read queue per port supporting three read traffic classes (H/V/LPR) and STM32MP13x feature a single read queue per port supporting only two traffic classes.

The settings described in Table 1, Table 2 and Table 3 are applicable to STM32MP15x. For STM32MP13x settings are converted to single port and two classes(V/LPR) as proposed by the DDR configuration tool classes. For more details, refer to [1] .

The fixed number of QoS values per controller is used by default. The AXI interconnect matrix (AXIM) maps all controllers to port 0, except LTDC to port 1 and controller port QoS according to the table below.

Table 1. Bus controllers AXI port assignment and QoS value for STM32MP15x

Bus controller	Read/write	Used port	Default QoS	Traffic class	Read/write QoS region
CPU	Read/write	0	12	Latency is sensitive	2
LTDC	Read	1	11	Latency is critical	1
ETH	Read/write	0	7	Latency is critical Best effort	
MCU(1)	Read/write	0	6		
USBH	Read/write	0	5		
SDMMC1	Read/write	0	4		
SDMMC2	Read/write	0	4		
GPU	Read/write	0	3	Best effort	0
DAP	Read/write	0	2		

1. *MCU is referring to the bus controllers DMA1/2, OTG, and SDMMC3.*

The DDRCTRL QoS and scheduling parameters are set as follows:

- Two AXI port configurations are recommended for x16 and x32 for improved RT traffic especially LTDC mapped to port 1
- QoS values (0 to 15) are mapped to classes for all ports read as follows:
  – Region 2 with 12 ≤ QoS ≤ 15 are mapped to HPR.
  – Region 1 with 4 ≤ QoS ≤ 11 are mapped to VPR.
  – Region 0 with 0 ≤ QoS ≤ 3 are mapped to LPR.
- QoS values (0 to 15) are mapped to classes for all ports write as follows:
  – Region 2 with 12 ≤ QoS ≤ 15 are mapped to VPW.
  – Region 1 with 4 ≤ QoS ≤ 11 are mapped to VPW.
  – Region 0 with 0 ≤ QoS ≤ 3 are mapped to NPW.

The QoS model is detailed as follows:

- HPR is allocated to the high-priority read queue and is suitable for latency-sensitive traffic, with bounded bandwidth.
- VPR/VPW are variable priority, suitable for latency critical, or real time critical traffic. VPR or VPW timeout expiration is preempting HPR.
- LPR/NPW are the best effort traffic.
- Read is preferred to write.
- Read CAM has an allocation for HPR (three entries) and VPR/LPR (13 entries).
- DDRCTRL features an antistarvation, controlled by max_starve/run_length for each queue: HPR and LPR for read, WR for write.

- VPR/VPW time out can be specified per port and per region according to the r/wqos_map_timeout parameters, with the value in clock cycle coded in hexadecimal. The lower this value is, the faster VPR/VPW goes into the 'expired-VPR/VPW' state, whether time out occurs when the transaction is in the AXI port queues or in the CAM store.
- Port aging may be used to prevent port starvation in case the head of the line blocks at AXI level. As this is not expected, port aging is not used with the proposed configuration.
- Intelligent precharge policy may be used instead of an open page policy for power saving.

Parameters and DDRCTRL registers are determined as shown in the tables below.

Note: QoS settings and scheduling control parameters are static and must be set during the configuration before DDRCTRL is out of reset.

Table 2. QoS and scheduling parameters for STM32MP15x

Scheduler parameter	Value	Description
sched.rdwr_idle_gap	0x0	Immediate switch between RD/WR transaction store
sched.lpr_num_entries	0xC	HPR: 3 entries, LPR: 13 entries
sched.pageclose	0x0	Opens the page policy.
sched.prefer_write	0x0	Always prefers read to write.
sched.force_low_pri_n	0x1	0: force all to low priority (leave to 1)
sched1.pageclose_timer	0x0	Not available with open page policy
perfhpr1.hpr_xact_run_length	0x1	Runs length for HPR.
perfhpr1.hpr_max_starve	0x1	Starvation control for HPR set to 1 clock
perflpr1.lpr_xact_run_length	0x8	Runs length for LPR.
perflpr1.lpr_max_starve	0x200	Starvation control for LPR set to 512 clock (~1 µs)
perfwr1.w_xact_run_length	0x8	Runs length for write.
perfwr1.w_max_starve	0x400	Starvation control for WR set to 1024 clock (~2 µs)

Table 3. QoS settings per AXI port for STM32MP15x

AXI port parameter	Port 0	Port 1(1)	Description
pcfgr.rdwr_ordered_en	0x1	0x1	Preserves the read/write orderings at the same address for coherency.
pcfgr.rd_port_pagematch_en	0x0	0x0	Disables the page match enable for read.
pcfgr.rd_port_aging_en	0x0	0x0	Enables the AXI read port aging.
pcfgr.rd_port_priority	0x0	0x0	AXI read port aging initial value to prevent port starvation
pcfgw.wr_port_pagematch_en	0x0	0x0	Disables the page match enable for write.
pcfgw.wr_port_aging_en	0x0	0x0	Enables the AXI write port aging.
pcfgw.wr_port_priority	0x0	0x0	AXI write port aging initial value to prevent port starvation
pctrl.port_en	0x1	0x1	Enables the AXI port.
pcfgqos0.rqos_map_region2(2)	0x2	0x2	Region 2 set to HPR
pcfgqos0.rqos_map_region1	0x1	0x1	Region 1 set to LPR or VPR
pcfgqos0.rqos_map_region0	0x0	0x0	Region 0 set to LPR
pcfgqos0.rqos_map_level2(2)	0xB	0xB	Region 2 for rqos > val
pcfgqos0.rqos_map_level1	0xA	0x3	Region 0 for rqos ≤ val
pcfgqos1.rqos_map_timeoutr	0x80	0x80	Timeout for red queue to go to expired VPR set to 128 clocks
pcfgqos1.rqos_map_timeoutb	0x100	0x40	Timeout for blue queue to go to expired VPR set to 256 clocks
pcfgwqos0.wqos_map_region2	0x1	0x1	Region 2 set to NPW or VPW
pcfgwqos0.wqos_map_region1	0x1	0x1	Region 1 set to NPW or VPW
pcfgwqos0.wqos_map_region0	0x0	0x0	Region 0 set to NPW
pcfgwqos0.wqos_map_level2	0xC	0xB	Region 2 for wqos > val
pcfgwqos0.wqos_map_level1	0xA	0x3	Region 0 for wqos ≤ val
pcfgwqos1.wqos_map_timeout2	0x100	0x100	Timeout for write in QoS region 2 to go to expired VPW
pcfgwqos1.wqos_map_timeout1	0x200	0x200	Timeout for write in QoS region 0 and 1 to go to expired VPW

1. *Not applicable to STM32MP13x because of the single port.*
2. *Not applicable to STM32MP13x because of single read queue.*

The QoS strategy is simplified for STM32MP13x with a single AXI port and a single read queue. As a consequence:

- All controllers are assigned to port 0 in Table 1. Bus controllers AXI port assignment and QoS value for STM32MP15x.
- The registers associated to port 1 do not exist in Table 4. DDRCTRL scheduling and performance control registers.
- QoS region 2 for read does not exist.

Table 4. DDRCTRL scheduling and performance control registers

Register	Address	Value	Description
SCHED	0x250	0x00000C01	Scheduler control register
SCHED1	0x254	0x00000000	Scheduler control register 1
PERFHPR1	0x25C	0x01000001	High-priority read CAM register 1
PERFLPR1	0x264	0x08000200	Low-priority read CAM register 1
PERFWR1	0x26C	0x08000400	Write CAM register 1
PCFGR_0	0x404	0x00010000	Port 0 configuration read register
PCFGW_0	0x408	0x00000000	Port 0 configuration write register
PCTRL_0	0x490	0x00000001	Port 0 control register (managed by the driver)
PCFGQOS0_0	0x494	0x02100B0A	Port 0 read QoS configuration register 0
PCFGQOS1_0	0x498	0x00800100	Port 0 read QoS configuration register 1
PCFGWQOS0_0	0x49C	0x01100C0A	Port 0 write QoS configuration register 0
PCFGWQOS1_0	0x4A0	0x01000200	Port 0 write QoS configuration register 1
PCFGR_1 (1)	0x4B4	0x00010000	Port 1 configuration read register
PCFGW_1 (1)	0x4B8	0x00000000	Port 1 configuration write register
PCTRL_1 (1)	0x540	0x00000001	Port 1 control register (managed by the driver)
PCFGQOS0_1 (1)	0x544	0x02100B03	Port 1 read QoS configuration register 0
PCFGQOS1_1 (1)	0x548	0x00800040	Port 1 read QoS configuration register 1
PCFGWQOS0_1 (1)	0x54C	0x01100B03	Port 1 write QoS configuration register 0
PCFGWQOS1_1 (1)	0x550	0x01000200	Port 1 write QoS configuration register 1

1. *Register or bit field not applicable to STM32MP13x*

*Note: The AXI port parameter **rdwr_ordered_en** is set to:*
- *1 for STM32MP15x*
- *0 for STM32MP13x*

3.3.4 DDRCTRL refresh controller

The DDRCTRL refresh controller is programmed according to the clock frequency and the SDRAM refresh requirement (JEDEC). The refresh parameters are determined by STM32CubeMX according to the JEDEC timing (t_{REFI} and t_{RFC}) and programmed to DDRCTRL.RFSHTMG.

The periodic autorefresh is used by default. However, DDRCTRL also supports the following optional features (refer to [1] for more details):
- Burst refresh with speculative burst refresh timeout
- Per-bank refresh for LPDDR2/3
- Refresh command posting by software

Note:
- *The LPDDR2/3 temperature derating with mode register 4 (MR4) periodic polling for autorefresh rate adjustment is enabled by default.*
- *ASR/SRT support can be set during the DDR3 configuration.*

3.4 PHY tuning

It is expected that the DQ and DQS from all byte lanes arrive at DDRPHYC well aligned. The recommendations provided in the application note AN5122 [5] help to ensure matched delay paths on the PCB. Consequently, the fine step delays, provisioned in the DDRPHYC, do not need to be modified and can be left at their default values.

The hardware built-in DQSTRN and RVTRN are launched at startup to determine a suitable window for capturing the DQ read data on DQS/DQS# differential strobe between the preamble and postamble at quarter bit resolution (refer to [1] , section DDRPHYC for more details).

The DQSTRN and RVTRN are robust and are expected not to fail; nevertheless the training results are visible in DDRPHYC registers:

- DXnGSR0 and DXnGSR1 with pass/fail, intermittent errors, and passing window size information
- DXnDQSTRN with SL and PS values obtained from DQSTRN
- DXnGCR with RVSL values obtained from RVTRN

During system initialization or in the event of a suspected failure, it is possible to examine these values. These values must be uniform, with no more than a one-quarter period discrepancy (for SL/PS) between lanes or between multiple startup or multiple boards.

Note: The receiver DLL phase should be left at nominal position (90°) after DQSTRN and RVTRN. The receiver DLL position is configured in the register DDRPHYC_DXnDLLCR.

4 Configuration sequence and parameters

The DDRSS subsystem set-up sequence is the following:

1. Set up power and clock.
2. Release the APB domain reset.
3. Load DDRCTRL registers (with SDRAM init skipped).
4. Release the DDRCTRL reset and DDRPHYC reset.
5. Load DDRPHYC registers
6. Launch the complete PHY and DDR initialization by PUB, according to DDRPHYC.PIR, with the steps listed below:
   a. Initialize DDRPHYC.
   b. Reset DRAM for DDR3.
   c. Reset DDL.
   d. Reset ITM.
   e. Initialize SDRAM.
   f. Calibrate the impedance (driver and ODT at PHY and SDRAM).
   g. Indicate that DFI init complete and wait the DDRCTRL normal operating mode.
   h. Perform the DQSTRN and RVTRN built-in.
   i. Enable the two AXI ports.

Then the tests can be executed during the bring-up.

5 DDR3/3L configuration

5.1 General considerations about DDR3/3L

DDR3 and DDR3L configurations are identical, except for power supply and V_{REF} as noted below:

- For DDR3: V_{DD} = 1.5 V and V_{REF} = 0.75 V
- For DDR3L: V_{DD} = 1.35 V and V_{REF} = 0.675 V

They are both referred as DDR3 in this document. Refer to [2] for more details.

DDR3 is available in BGA with a 16-bit interface. The STM32CubeMX DDR panel is supporting the configurations listed in Table 5. DDR3 density and topologies.

DDR3 supported topologies are the following:

- 16 bits: single BGA in p2p, with density from 1 to 8 Gbit/s
- 32 bits: two BGA in fly-by topology, each die with a density from 1 to 4 Gbit/s

Table 5. DDR3 density and topologies

Total DDR density (Mbytes)	16-bit interface	32-bit interface
128	1 device x16 (1 Gbit)	Not available
256	1 device x16 (2 Gbit/s)	2 device x16 (1 Gbit)
512	1 device x16 (4 Gbit/s)	2 device x16 (2 Gbit/s)
1000	1 device x16 (8 Gbit/s)	2 device x16 (4 Gbit/s)

The DDR frequency is set by the STM32CubeMX clock configuration. It is also used by the DDR tool with the following constraints:

- 300 MHz ≤ frequency ≤ 533 MHz (300 MHz is the DDR3 lower limit with DLL on)
- DDR3 DLL off with frequency ≤ 125 MHz

DDR-1066 has three speed bins, G/F/E, according to the STM32CubeMX DDR panel menu selection (respectively CONFIG0/CONFIG1/CONFIG2).

The JEDEC speed grades and speed bins are used to select the closest match to DDR-1066 at frequency ≤ 533 MHz, and to get all the other JEDEC timing parameters. The user only needs to input the CONFIG0/1/2 according to the table below.

Table 6. DDR3 datasheet index value according to speed grade/speed bin

DDR3 speed bin	DDR3 speed grade				
-	DDR-1066	DDR-1333	DDR-1600	DDR-1866	DDR-2133
N	-	-	-	-	CONFIG1
M	-	-	-	CONFIG1	CONFIG1
L	-	-	-	CONFIG1	CONFIG2
K	-	-	CONFIG1	CONFIG1	CONFIG2
J	-	CONFIG0	CONFIG1	CONFIG2	-
H	-	CONFIG1	CONFIG2	-	-
G	CONFIG0	CONFIG1	CONFIG2	-	-
F	CONFIG1	CONFIG2	-	-	-
E	CONFIG2	-	-	-	-

By default, the CONFIG0 is selected with the conservative timings corresponding to the 8-8-8 fundamental triplet timing at 533 MHz.

The CL/CWL parameters are determined according to the frequency with the datasheet selector (0.1 or 2) information according to the table below.

Table 7. CL/CWL versus frequency and datasheet index

CONFIGx	Frequency range (MHz)			
-	f ≤ 125	300 ≤ f ≤ 333	333 < f ≤ 400	400 < f ≤ 533
CONFIG0	CL = 6, CWL = 6			CL = 8, CWL = 6
CONFIG1				CL = 7, CWL = 6
CONFIG2				CL = 6, CWL = 6

Several timing parameters may have a lower value than DDR-1066 and may be overridden in the DDR configuration tool advanced parameters. However, this has a marginal impact on performances. Propose default parameters according to bin selection is usually sufficient.

5.2 DDR3 power-up sequence, initialization, and ZQ calibration

The DDR3 power-up sequence is specified by JEDEC with a voltage ramping, V_{DD}/V_{DDQ}, generated from a single supply with a ramp < 200 ms.

The DDR3 initialization sequence is specified as follows:

1. Optionally maintain RESET# low for a minimum of either 200 µs (power-up initialization) or 100 ns (power-on initialization). DDRPHYC drives RESET# low from the beginning of the reset assertion. Therefore, this step may be skipped if enough time has already expired to satisfy the RESET# low time.
2. After RESET# is deasserted, wait a minimum of 500 µs with CKE low.
3. Apply NOP and drive CKE high.
4. Wait a minimum of t_{XPR}.

Caution: DDRPHYC, DDR mode and DDRCTRL registers must be consistently programmed. Mismatches between the register fields may cause transaction failures. Secure that all register fields are consistently programmed before starting any SDRAM transaction.

5. Issue a load mode register 2 (MR2) command.
6. Issue a load mode register 3 (MR3) command.
7. Issue a load mode register (MR1) command to set parameters and enable DLL.
8. Issue a load mode register (MR0) command to set parameters and reset DLL.
9. Issue a ZQ calibration command.
10. Wait 512 SDRAM clock cycles for the DLL to lock (t_{DLLK}) and ZQ calibration (t_{ZQinit}) to finish.

This wait time is relative to step 8, when the DLL reset command is issued onto the SDRAM command bus.

The DDR3 initialization is fully controlled by DDRPHYC. The PTR0/1/2 registers define the initialization timings parameters that are adjusted to clock frequency. The complete initialization is triggered by the PIR register.

5.3 Frequency range restriction

The clock frequency must respect the following conditions:

- When DLL on, 300 MHz ≤ f ≤ 533 MHz.
- When DLL off, f < 125 MHz.

As DLL is on by default, DLL off mode cannot be used directly and is not supported.

5.4 On-die-terminations (ODTs)

ODTs are essential to improve the signal integrity. They can be used with all DDR3 topologies (16-bit and 32-bit interfaces).

ODTs are applicable to DQ/DQS byte lanes as described below:

- The local PHY ODTs are dynamically switched during read operations. The ODTs impedance is programmable via ZQ0CR1.ZPROG[7:4] (see Table 8). The DQ/DQS and A/C output impedance is programmable via ZQ0CR1.ZPROG[3:0] (see Table 9).
  By default ZPROG[7:0] = 0x38 setting the DQ/DQS and A/C output impedance to 53 Ω and the DQ/DQS ODT impedance to 80 Ω. These values are a good compromise between signal levels, signal reflection, and power optimizations
- The SDRAM ODTs are dynamically switched during write operations by the ODT pin. DDR is set up with RTT_WR (also named dynamic ODT) and controlled by MR2[1:0] as follows:
  – MR2[1:0] = 0b00: disabled
  – MR2[1:0] = 0b01: 60 Ω
  – MR2[1:0] = 0b10: 120 Ω

Note: The DDR3 RTT_NOM mode is not used and not needed with single rank.

The DQ/DQS output impedance on the DDR side is set by MR1[1] as follows:

- MR1[1] = 0: 40 Ω
- MR1[1] = 1: 34 Ω

The nominal configuration for DDR3 is DQ/DQS output impedance to 40 Ω and ODT impedance to 60 Ω (set by the MR2 mode register).

Table 8. ODT impedance versus ZPROG bits (RZQ = 240 Ω ± 1%)

ZPROG[7:4]	ODT impedance (Ω)
0x0	-
0x1	120
0x2	96
0x3	80
0x4	69
0x5	60
0x6	52
0x7	46
0x8	40
0x9	37
0xA	34
0xB	32
0xC	30
0xD	28
0xE	26.5
0xF	25

Table 9. DQ/DQS and A/C output impedance versus ZPROG bits (RZQ = 240 Ω ± 1 %)

ZPROG[3:0]	DQ/DQS and A/C impedance (Ω)
0x0 t0 0x4	-
0x5	80
0x6	69
0x7	60
0x8	53
0x9	48
0xA	44
0xB	40
0xC	37
0xD	34
0xE	32
0xF	30

The ODT turn on/off timings are controlled as follows:

- DDRPHYC side: DXnGCR.RTTOAL and DXnGCR.RTTOH parameters
  ODTs are turned on 2.25 cycles before read and turned off one cycle after the last byte.
- DDR side: via ODT pin by DDRCTRL according to ODTCFG.WR_ODT_DELAY and ODTCFG.WR_ODT_HOLD
  ODTs are placed WL-2 cycles ahead of the write burst and asserted during six cycles for BL = 8.

Note: The PHY ODTs are activated by internal signal TE = 1 and ODT on/off switching may cause significant current switching and potentially affecting PDN that may disturb CK/CK# and induce jitter. Higher ODTs values reduce the current switching and may be used to reduce the supply noise in case of issue.

5.5 Command/address lanes (C/A) terminations

The RTT terminations are used for C/A bus with fly-by topology in 32-bit and eventually in 16-bit configuration. Refer to [5] for more details.

The RTT termination presence does not impact DDR configuration.

5.6 Impedance calibration (ZCAL)

The DDR3 signal impedance needs to be accurate both on local PHY and DDR PHY. This is accomplished by the impedance calibration (ZCAL), that is the calibration of both output driver and ODT impedances.

5.6.1 PHY

DDRPHYC features a ZCAL engine to adjust the SSTL I/O impedance to programmed values, relying on the external RZQ = 240 Ω +/- 1%.

ZCAL is automatically triggered during the initialization.

ZCAL can also be launched later by software. In addition, DDRCTRL is supporting the DFI controller PHY update that can be used to issue ZCAL at regular time intervals (or self-refresh exit).

The PHY impedances are programmed with ZPROG as detailed in Section 5.4: On-die-terminations (ODTs).

5.6.2 SDRAM

The four impedances are calibrated as detailed below, relying on the external RZQ = 240 Ω ± 1%:

- Driver pull-up or pull-down: two possible values, 34 Ω (default) and 40 Ω
- ODT pull-up or pull-down: multiple possible values (default = 60 Ω)

The ZQCL or ZQCS commands trigger the SDRAM calibrations as follows:

- The ZQCL command is used for initial calibration during the power-up initialization sequence and launched by DDRPHYC. This command may be issued also later by DDRCTRL at self-refresh exit, depending on the system environment. ZQCL takes 256 clocks.
- The ZQCS command may be used for periodic calibrations to account for voltage and temperature variations. This command is issued by DDRCTRL at regular time intervals. ZQCS takes 64 clocks.

5.7 DLL on/off mode

DLL is on by default with DDR3. The DLL-OFF mode operation is restrictive and with low interest.

However, DLL may be turned off during the DDR3 power-down mode according to MR0[12] for power saving, as detailed below:

- When MR0[12] = 0 (DDR3 slow-exit mode), t_{XSDLL} is applied on power-down exit (PDE). t_{XSDLL} is a 512 clock delay from PDE to any command that requires the DLL to be locked.
- When MR0[12] = 1 (DDR3 fast-exit mode), only t_{XS} is applied on PDE.

By default, the DDR tool uses the fast-exit mode (MR0[12] = 1) for lowest latency impact in case of automatic power-down entry.

5.8 DDRPHYC built in DQSTRN

The DQS gate training (DQSTRN), is launched by default during the initialization. DQSTRN uses the DDR array and read/write operation (row/bank/column zero by default).

DQSTRN using MPR may be supported by DDRPHYC but this is not proposed as an option.

DQSTRN is fully controllable by the PUB registers: launching, row/bank/column, pattern and status report (see [1] for more details). Built-in DQS gate training is run on every startup with an update of the DXnDQSTR registers. The training status is reported in the DXnGSR0/1 registers.

5.9 DDRPHYC built in RVTRN

The Read valid training (RVTRN) is deemed not needed for DDR3, therefore RVTRN is not launched.

Note: DDR3 has lower t_{DQSCK} spread because of DLLs characteristics.

5.10 DDR3 configuration example

This example shows the DDR3 configuration on a board with a dual 4-Gbit DDR3L.

The clock frequency is defined by the STM32CubeMX clock configuration (528 MHz in this example).

The DDR type and width are defined in the STM32CubeMX pinout and configuration.

The values for these examples are the following:

- DDR type: DDR3 or DDR3L
- Width: 32 bits
- Density: 4 Gbits

The other parameters are set as follows in the DDR parameters window according to the scroll window menus and tick boxing:

- Speed bin grade: DDR3-1066 / 8-8-8
- Impedance during read: R_{ON} 40 Ω and ODT = 80 Ω (default)
- Impedance during write: R_{ON} 53 Ω and ODT = 60 Ω (default)
- Address mapping configuration: R/B/C
- Relax timing mode: (tick box)
- Temperature case over 85°C (tick box)
- Datasheet selection: 0
- Sched/Qos option: 2

Note: The PHY byte lanes are enabled according to the interface width as follows:

- *All byte lanes on in 32-bit mode*
- *Byte lanes 2/3 off in 16-bit mode*

The relaxed timing mode is used to increment the core timings by 1 to improve timing margins. This must be done in case of suspicious failures.

The register values for the above configuration are detailed in the tables below.

Table 10. DDRCTRL and DDRPHYC timings register values

Peripheral	Register	Address	Value
DDRCTRL	RFSHTMG	0x064	0x0040008B
	DRAMTMG0	0x100	0x121B1214
	DRAMTMG1	0x104	0x000A041C
	DRAMTMG2	0x108	0x0608090F
	DRAMTMG3	0x10C	0x0050400C
	DRAMTMG4	0x110	0x08040608
	DRAMTMG5	0x114	0x06060403
	DRAMTMG6	0x118	0x02020002
	DRAMTMG7	0x11C	0x00000202
	DRAMTMG8	0x120	0x00001005
	DRAMTMG14	0x138	0x000000A0
	ODTCFG	0x240	0x06000600
DDRPHYC	PTR0	0x018	0x0022AA5B
	PTR1	0x01C	0x04841104
	PTR2	0x020	0x042DA068
	DTPR0	0x034	0x38D488D0
	DTPR1	0x038	0x098B00D8
	DTPR2	0x03C	0x10023600
	MR0	0x040	0x00000840
	MR1	0x044	0x00000000
	MR2	0x048	0x00000248

Table 11. DDRCTRL and DDRPHYC constant register values versus DDR3

Peripheral	Register	Address	Value
DDRCTRL	MSTR	0x000	0x00040401
	MRCTRL0	0x010	0x00000010
	MRCTRL1	0x014	0x00000000
	DERATEEN	0x020	0x00000000
	DERATEINT	0x024	0x00800000
	PWRCTL	0x030	0x00000000
	PWRTMG	0x034	0x00400010
	HWLPCTL	0x038	0x00000000
	RFSHCTL0	0x050	0x00210000
	RFSHCTL3	0x060	0x00000000
	CRCPARCTL0	0x0C0	0x00000000
	ZQCTL0	0x180	0xC2000040
	DFITMG0	0x190	0x02060105
	DFITMG1	0x194	0x00000202
	DFILPCFG0	0x198	0x07000000
	DFIUPD0	0x1A0	0xC0400003
	DFIUPD1	0x1A4	0x00000000
	DFIUPD2	0x1A8	0x00000000
	DFIPHYMSTR	0x1C4	0x00000000
	ODTMAP	0x244	0x00000001
	DBG0	0x300	0x00000000
	DBG1	0x304	0x00000000
	DBGCMD	0x30C	0x00000000
	POISONCFG	0x36C	0x00000000
	PCCFG	0x400	0x00000010
DDRPHYC	PGCR	0x008	0x01442E02
	ACIOCR	0x024	0x10400812
	DXCCR	0x028	0x00000C40
	DSGCR	0x02C	0xF200001F
	DCR	0x030	0x0000000B
	MR3	0x04C	0x00000000
	ODTCR	0x050	0x00010000
	ZQ0CR1	0x184	0x00000038
	DX0GCR	0x1C0	0x0000CE81
	DX1GCR	0x200	0x0000CE81
	DX2GCR	0x240	0x0000CE81
	DX3GCR	0x280	0x0000CE81

Table 12. DDRCTRL address map register values for DDR3

Register	Address	Value
ADDRMAP1	0x204	0x00080808
ADDRMAP2	0x208	0x00000000
ADDRMAP3	0x20C	0x00000000
ADDRMAP4	0x210	0x00001F1F
ADDRMAP5	0x214	0x07070707
ADDRMAP6	0x218	0x0F0F0707
ADDRMAP9	0x224	0x00000000
ADDRMAP10	0x228	0x00000000
ADDRMAP11	0x22C	0x00000000

Table 13. DDRCTRL QoS scheduling register values for DDR3

Register	Address	Value
SCHED	0x250	0x00000C01
SCHED1	0x254	0x00000000
PERFHPR1	0x25C	0x01000001
PERFLPR1	0x264	0x08000200
PERFWR1	0x26C	0x08000400
PCFGR_0	0x404	0x00010000
PCFGW_0	0x408	0x00000000
PCFGQOS0_0	0x494	0x02100C03
PCFGQOS1_0	0x498	0x00800100
PCFGWQOS0_0	0x49C	0x01100C03
PCFGWQOS1_0	0x4A0	0x01000200
PCFGR_1	0x4B4	0x00010000
PCFGW_1	0x4B8	0x00000000
PCFGQOS0_1	0x544	0x02100C03
PCFGQOS1_1	0x548	0x00800040
PCFGWQOS0_1	0x54C	0x01100C03
PCFGWQOS1_1	0x550	0x01000200

The low-power features are the following:

- Power down: automatic power-down option (advanced user)
- Self-refresh: can be controlled by software (SSR) or hardware (HSR) or can be automatic (ASR) (see [1] , section RCC for more details)

The extended temperature range supports the ASR/SRT modes (STM32CubeMX option). The DDR3 MR2 is used to control self-refresh to support an extended temperature range. In case of extended temperature, the auto self-refresh must be set with ASR = 1 (when DDR3 supports this optional ASR feature). If AST is not supported, then SRT may be used to indicate operating temperature for subsequent self-refresh intervals.

6 LPDDR2 configuration

6.1 General considerations about LPDDR2

LPDDR2 is available in x32 and x16 in multiple package ballouts (BGA and PoP). The DDR tool is supporting the configurations listed in the table below.

The supported LPPDDR2-S4 configurations with density ≥ 1 Gbit have eight banks.

The LPDDR2 supported topologies are the following:

- 16 bits: single BGA in p2p with density from one to four Gbit/s (single die)
- 32 bits: single BGA in p2p with density from one to four Gbit/s (single die)

Refer to [3] for more details.

Table 14. **LPDDR2 density and topologies**

Total DDR density (Mbytes)	16-bit interface	32-bit interface
128	1 device x16 (1 Gbit)	1 device x32 (1 Gbit)
256	1 device x16 (2 Gbit/s)	1 device x32(2 Gbit/s)
512	1 device x16 (4 Gbit/s)	1 device x32(4 Gbit/s)

Caution: The configurations listed in the table above are LPDDR2-S4 with eight banks. The DDR tool does not support lower density LPDDR2-S2 with four banks.

LPDDR2 features the following:

- No ODT, no DLL, and no reset pin
- CMOS I/O (DDRPHYC is configured with I/Os set in SSTL mode)
- 10-533 MHz continuous range operation
- Specific low-power features: PASR, temperature controlled self-refresh
- Per bank refresh (advanced user)
- MRR MR4 polling and refresh T derating (set by default)
- Deep power down (DPD) mode[As the LPDDR2 content is lost during DPD, this mode has very limited interest and is not discussed.]

Note: As the LPDDR2 content is lost during DPD, this mode has very limited interest and is not discussed.

6.2 LPDDR2 power-up, initialization, and ZQ calibration

LPDDR2 needs two power supplies (V_{DD1} = 1.8 V, V_{DD2} = 1.2 V) and V_{REF} = 0. 6 V.

LPDDR2 power-up sequence and the following voltage ramping must be observed:

- < 20 ms ramp
- $V_{DD1} > V_{DD2}$ - 200 mV
- (V_{DD1}, V_{DD2}) > (V_{DDQ} - 200 mV, V_{DDCA} - 200 mV)
- CKE low
- all inputs and V_{REF} within the supply range

6.2.1 LPDDR2 initialization sequence

The LPDDR2 initialization sequence is specified as follows:

- Wait a minimum of 100 ns (t_{INIT1}) with CKE driven low.
- Apply NOP and set CKE high.
- Wait a minimum of 200 µs (t_{INIT3}).
- Issue a RESET command.
- Wait a minimum of 1 µs + 10 µs (t_{INIT4} + t_{INIT5}).
- Issue a ZQ calibration command.
- Wait a minimum of 1 µs (t_{ZQINIT}).
- Issue a write mode register to MR1.
- Issue a write mode register to MR2.
- Issue a write mode register to MR3.

LPDDR2 initialization is fully controlled by DDRPHYC. The PTR0/1/2 registers define the initialization timings parameters that are adjusted to clock frequency. The complete initialization is triggered by the PIR register.

6.2.2 LPDDR2 specific settings

The LPDDR2 specifics settings are listed below:

- Extended clock frequency range: 10 MHz to 533 MHz
- No DLL at LPDDR2: faster exit from low-power modes (PDN and self-refresh) and less power
- I/O: CMOS mode, no termination at DDRPHYC, no ODT, Z_{OUT} = 48 Ω (LPDDR2) and Z_{OUT} = 53 Ω (DDRPHYC) recommended at LPDDR2 for optimal signaling
- DQS/DQS# pull-up/pull-down for DQS glitch filtering set to a highest value of 688 Ω
- When frequency > 200 MHz, DDRPHYC DLLs are on. When frequency ≤ 200 MHz, DDRPHYC DLLs are in bypass to reduce the power dissipation.
- ZQ calibration, ZQCL at SRX, ZQCS at regular interval (option for advanced user)
- Low-power options: ASR and precharge power-down can be set by DDRCTRL.
- DQSTRN gate extension is used to compensate t_{DQSCK} variations from lacking DLLs.
- MR4 polling (temperature) is enabled by default.

Note: DDRPHYC I/O must be in SSTL mode when frequency > 200 MHz.

6.3 Frequency range restriction

The clock frequency must respect the following conditions:

- 10 MHz ≤ CLK ≤ 100 MHz (DDRPHYC DLLs in bypass and bp200 = 1)
- 100 MHz < CLK ≤ 200 MHz (DDRPHYC DLLs in bypass and bp200 = 0)
- 200 MHz < CLK ≤ 533 MHz (DDRPHYC DLLs are on)

6.4 Output impedance

The LPDDR2 I/O impedance is set according to the MR3 register as listed in the table below.

Table 15. **LPDDR2 output impedance versus MR3 bits (RZQ = 240 Ω ± 1%)**

MR3[3:0]	LPDDR2 Z_{OUT} (Ω)
0x0	Reserved
0x1	34.3
0x2	40 (default)
0x3	48 (recommended)
0x4	60
0x5	68.6
0x6	80
0x7	120
Others	Reserved

For the signal integrity (SI) and to reduce overshoot with an unterminated interface, Z_{OUT} = 48 Ω is recommended.
The DDRPHYC I/O impedance is set according to DDRPHYC_ZQ0CR1.ZPROG as listed in the table below.

Table 16. **Output impedance versus ZPROG bits (RZQ = 240 Ω ± 1%)**

ZPROG[3:0]	PHY Z_{OUT}(Ω)
0x0 to 0x4	-
0x5	80
0x6	69
0x7	60
0x8	53 (recommended)
0x9	48
0xA	44
0xB	40 (default)
0xC	37
0xD	34
0xE	32
0xF	30

6.5 Command/address lanes (C/A)

The LPDDR2 command and address are encoded to C/A [9:0] lines and use the DDR signaling.

6.6 Impedance calibration (ZCAL)

Both DDRPHYC and LPDDR2 Z_{OUT} are calibrated against the external R = 240 Ω on the board.

ZCAL is always launched during the DDR initialization. ZCAL may be triggered at SRX and at regular intervals later (advanced user). ZCAL is supporting the calibration of the output driver (Z_{OUT}).

ODTs are not applicable to LPDDR2.

6.6.1 At local PHY

The process called ZCAL starts by itself when the system is starting up, but it can also be started manually using software.

Driver pull-up and pull-down according to ZPROG value in Table 16.

6.6.2 At DDR

The LPDDR2 impedance calibration is initiated with the mode register (MR10) command.

There are four ZQ calibration commands and related timings listed below:

- t_{ZQINIT} corresponds to the initialization calibration.
- $t_{ZQRESET}$ is used for resetting ZQ to its default setting. See the mode register 10 (MR10) for a description on the command codes.
- The ZQCL command is for long calibration. This command is used to perform the initial calibration during the power-up initialization sequence. ZQCL is launched by DDRPHYC and may also be issued later by DDRCTRL depending on the system environment. ZQCL takes 256 clocks.
- The ZQCS command is for short calibration. This command may be used to perform periodic calibrations to account for voltage and temperature variations. ZQCS is issued by DDRCTRL at regular time intervals and takes 64 clocks.

6.7 DDRPHYC built in DQSTRN

DDRPHYC supports the hardware built-in DQSTRN that is launched by default during the initialization.

DQSTRN uses the DDR array and read/write operation (R/B/C zero by default). DQSTRN is fully controlled by the DDRPHYC registers: launching, R/B/C usage, pattern, status reporting. Built-in DQS gate and RV training are run with default settings. They update DXnGCR and DXnDQSTR registers. The status is reported in DXnGSR0/1 registers.

Note: *At low frequency, fewer steps can be determined by DQSTRN. There may even be a unique valid step with the system latency (SL) and phase shift (PS) predetermined values and DQSTRN cannot be applied. In this case, the DXnDQSTR and DXnDQTR registers are directly set with predefined values or left at default (SL = 0 and PS = 2 for 360° shift).*

6.8 DDRPHYC built in RVTRN

The Read valid training (RVTRN) is run on every startup with an update of the DXnGCR register. The training status is reported in the DXnGSR0/1 registers.

6.9 LPDDR2 configuration example

This example shows the LPDDR2 configuration on a board with a 4-Gbit LPDDR2.

The clock frequency is defined by the STM32CubeMX clock configuration (528 MHz in this example).

The DDR type and width are defined in the STM32CubeMX pinout and configuration.

Values for this example are the following:

- DDR type: LPDDR2
- Width: 32 bits
- Density: 4 Gbits

The other parameters are set in the DDR parameters window according to the scroll window menus and tick boxing:

- Impedance during read: Ron 48 Ω (default)
- Impedance during write: Ron 53 Ω (default)
- Burst length: 8
- Address mapping configuration: R/B/C
- Relax timing mode: (tick box)

JEDEC timing parameters from a typical datasheet are proposed and listed by the STM32CubeMX DDR configuration tool. They may be overridden in case of discrepancy with the LPDDR2 datasheet.

Note: *There are no bins with LPDDR2. A single set of timing parameters is selected (CONFIG0). RL/WL are automatically selected according to the frequency.*

The PHY byte lanes are configured according to interface width as follows:

- 16-bit mode: byte lanes 0 /1 on and byte lane 2/3 off
- 32-bit mode: all four byte lanes on

The flexible address mapping of the system address to DDR B/R/C is determined according to the density. The user can select one of two following predefined options:

- R/B/C (row/bank/column, default) that uses the bank interleaving for slightly better performance with higher power
- B/R/C (bank/row/column, default) that may improve power but worse performance in case of bank conflicts

The relaxed timing mode may be used to increment the core timings by 1 to improve timing margins. This must be done in case of suspicious failures.

The values of the AXI port mapping, the QoS settings, and the DDRCTRL scheduler timeout are selectable from a predefined sched/QoS parameter set within STM32CubeMX.

QoS type 2 is used by default and is applicable to most use cases with the following features:

- queue antistarvation support
- three read traffic class (HPR/VPR/LPR)
- low timeout for VPR expiration

7 LPDDR3 configuration

LPDDR3 is supported like LPDDR2 with frequency ≤ 533 MHz and without ODTs.

LPDDR3 has a few restrictions versus LPDDR2: BL8 only and fewer Z_{OUT} impedance settings.

Note: BL8 is mandatory for LPDDR3.

Refer to [4] for more details.

LPDDR3 has slight timing differences and RL/WL restrictions versus frequency. Furthermore, due to the higher frequency support, LPDDR3 has some extended mode register values and several RL/WL restrictions at lower frequency (for example RL=3/WL=1 support is optional).

LPDDR3 features the following:

- No DLL and no reset pin
- CMOS I/O (DDRPHYC is configured with I/Os set in SSTL mode)
- 10-533 MHz continuous range operation
- Specific low-power features: PASR, temperature controlled self-refresh
- Per bank refresh (advanced user)
- MRR MR4 polling and refresh T derating (set by default)
- Deep power down (DPD) mode (Note that as the LPDDR2 content is lost during DPD, this mode has very limited interest and is not discussed.

Note: Per JEDEC209-3C, LPDDR3 is available in 1-,2-, 4-, 6-, and 8-Gbit density and above in x16 and x32. However, the existing LPDDR3 is a short subset of possible configurations. The most common configuration, 8-Gbit density x32, is used as an example in the configuration tool.

8 DDR testing with STM32CubeMX

STM32CubeMX is supporting an extensive DDR testing suite, launching selected tests either to verify the DDR configuration robustness or to catch potential elusive errors. The tests are described in Section 8.1. Different causes and possible corrective actions are suggested for the failed tests in Section 8.2 and Section 8.3. An overall test flow is presented in Section 8.4.

The software including the tests may be downloaded to SRAM and the execution is controlled from STM32CubeMX.

For more information, refer to [6] .

8.1 Tests description

Tests are classified in the three following types:

- **Basic tests**: These simple and running fast tests are intended to capture the major configuration or hardware issues showing off immediately.
- **Intensive tests**: These tests use extensive coverage of data and address patterns for noise and high SSO conditions, high throughput traffic, or interleaved read/write. Depending on the parameters, the test runtime may be long. An intensive test can be deployed progressively, with test trial before launching long and exhaustive test sequences.
- **Stress tests**: These tests are intensive and executed with stretched conditions (such as a small frequency increase 10-20 MHz), with a skew of parameters (for example a fine step delay increase) or with specific frequency selective patterns.

These tests are intended to catch low-margin timings of a configuration that may cause elusive errors and eventual crashes later during runtime.

A stress test campaign must always be done during the system bring-up. Stress tests may also be run in case of suspicious failure. Any test and its skewed parameter must be directed to pinpoint the observed failure (for example, when errors are related to specific bit or byte).

All the available tests are detailed in the table below.

Table 17. Tests list

Test #	Test name	Type	Description
1	Simple databus	Basic	Verifies that each data bus signal can be driven high at the given address.
2	Databus walking 0	Basic	Verifies that each data bus signal can be driven low.
3	Databus walking 1	Basic	Verifies that each data bus signal can be driven high.
4	Address bus	Basic	Verifies each address bus line in a memory region, by performing a walking 1 test on the relevant address bits and checking for aliasing.
5	Mem device	Intensive	Performs read/write over an entire memory region. Each data bit is written and read back with the values 0 and 1.
6	Simultaneous switching output	Intensive	Stresses the data bus over an address range by doing simultaneous switching output. Writes a pseudo-random value and read it back.
7	Noise	Intensive	Verifies read/write while forcing switching of all data bus lines.
8	Noise burst	Intensive	Verifies read/write while forcing switching of all data bus lines (test based on 8-word bursts).
9	Random	Intensive	Verifies read/write with a pseudo-random value on one region.
10	Frequency selective pattern	Intensive with stress	Stresses the data bus by performing successive write 8-word burst operations using mostly zero/one patterns and frequency divider patterns (F/1, F/2, F/4) for 16 and 32 data bus width.
11	Block sequential	Intensive	Well known user-space memory tester adapted.
12	Checkerboard		
13	Bit spread		
14	Bit flip		
15	Walking ones		
16	Walking zeroes		
17	Infinite read	Basic	Performs an infinite read for a specific pattern (for debug and lab usage only, not visible).
18	infinite write	Basic	Performs an infinite write access to DDR (for debug and lab usage only, not visible).
Any	Overclocking	Intensive with stress	Runs level1 intensive tests with DDR clock increase by ~5% (up to 30 MHz).
Any	DQS timings margins check	Intensive with stress	Runs level1 intensive tests with stepping of fine step DQ and DQS delays.

8.2 Failure classification

Failure types are detailed below:

- **Catastrophic failures**: failures that happen with basic tests, generally resulting from a major configuration or hardware issue. The cause can be more or less obvious from the test report.
- **Sporadic failures**: failures that are not caught by intensive tests and not frequent. It is essential to identify failure commonalities and patterns to pinpoint their root cause. For example, they may affect a particular bit or byte lane or they may be read or write.

8.3 Possible actions on test failure

The most common action is to modify the configuration. The goal is to correct a wrong setting or to add more margin to a parameter and then restart the test sequence.

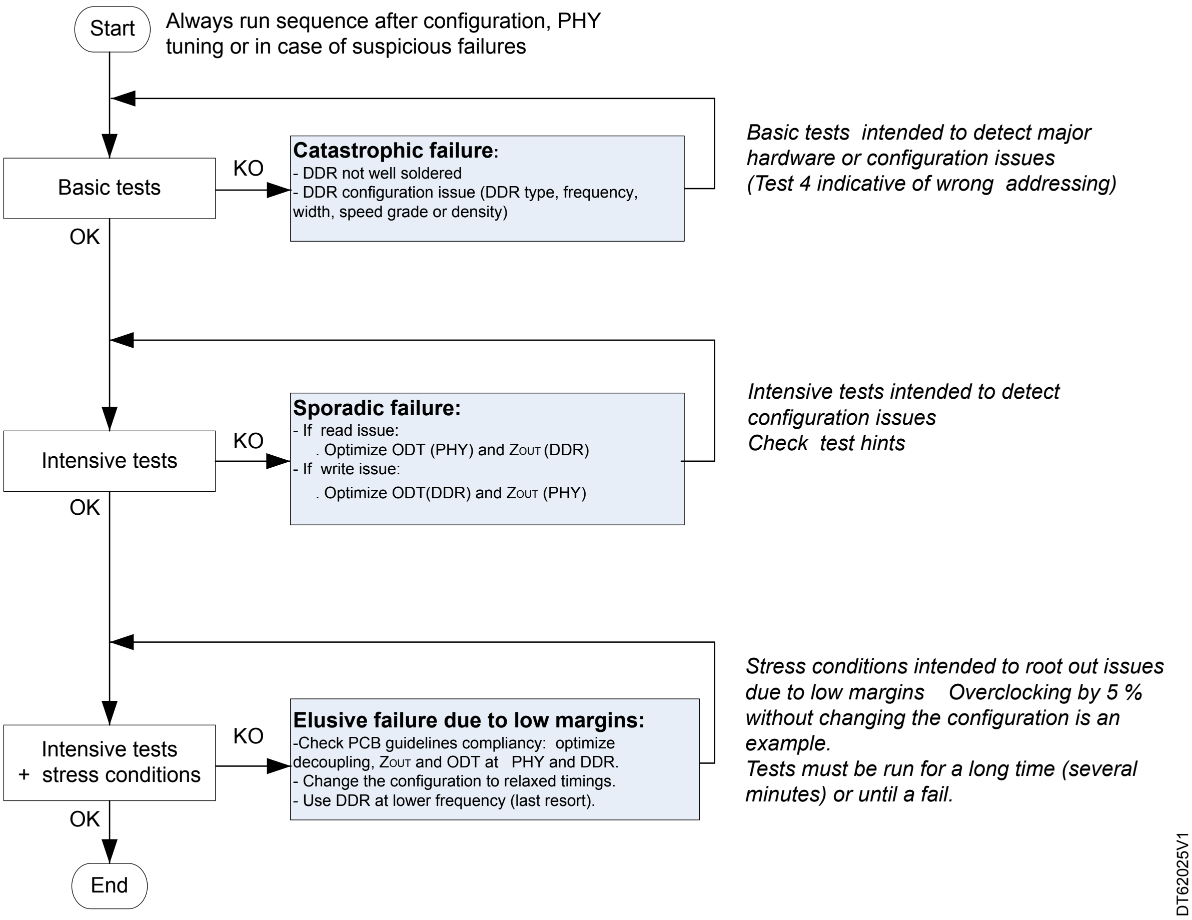
If a failure is related to PHY timing margins or the environmental conditions:

- the configuration settings must be changed by applying this reconfiguration within STM32CubeMX
- use the "Reload DDR registers" feature to run again a DDR initialization with the new register value.

8.4 Test flow

The overall flow is described in the figure below.

Figure 4. DDR test flow diagram



Note: Physical parameters listed above can be changed from the STM32CubeMX configuration panel.

9 Advanced user configuration

The table below lists some features and modes that may be changed by an advanced user. They come with a brief description of their relevance and applicability to DDR types. The register to use to modify the default proposed by STM32CubeMX can be found in [6] .

Note: The ASR/SRT option is supported by STM32CubeMX. The temperature derating with MR4 polling is set by default. STM32CubeMX supports the nonstandard impedance and ODT.

Table 18. Advanced user feature and parameters

Feature	DDR3	LPDDR2/3	Description
Per-bank-refresh	N	Y	$t_{RFCpb} < t_{RFCab}$ Per bank refresh is faster than all bank refresh. Provides more predictable DDR system latency with LPDDR2/3.
Burst-refresh	Y	Y	Option to group auto refresh commands up to 9 Postpones regular refresh to improve the scheduling of the refresh commands.
Nonstandard BL	N	Y	LPDDR2 only. The LPDDR2 burst may be 4, 8 or 16.
Burst chop	Y	N	Burst chop: BC4 Used to terminate a burst earlier (MSTR.BURSTCHOP bit)
Flexible address mapping	Y	Y	Extends the short pick list with RBC/BRC Almost any order of bits can be programmed for row/bank/column bit interleaving but no an easy way to anticipate which setting is better. Tedious to determine the register bits. The register settings need to be verified with the address boundary checking test program.
Custom QoS settings	Y	Y	Default QoS and port mapping set at AXI interconnect level A pick list of standard QoS settings is proposed.
ASR/SRT	Y	N	Applicable when T_{CASE} > 85°C to double refresh rate SRT direct software and ASR automatically based on DDR sensor SRT/ASR are exclusive and controlled by the DDR3 mode register.
Temperature derating	N	Y	Applicable when T_{CASE} > 85 °C to derate the refresh rate DDRCTRL supports the automatic polling of MR4. The polling rate can be set according to the estimated T gradient. Use DERATEEN/DERATEINT register control. Set by default for LPDDR2/3 with ~20 ms @400 MHz interval.
Nonstandard ZQ and ODT values	Y	N	By default DDR3 ZQ = 40 and ODT = 60 DDR3 ZQ may be set as 34 or 40 ohm. ODT may be off, 120, 60 or 40. The DDR impedance is changed by the mode register. DDRPHYC ZQ and ODT are controlled by ZQ0CR0/1 registers ZPROG.
Nonstandard ODT scheme	Y	N	By default DDR RTT_nom off and RTT_wr = 60. DDPHY dynamic ODT. It is not recommended to change these settings.
Fast/Slow PD exit	Y	N	Current MR0[12] = 1. Fast exit mode (DDR3 DLL on during PD)
Auto ZCAL	Y	Y	By default, ZQCL after SRX and ZQCS at regular intervals are both disabled. These two features may be enabled in case of significant environmental (T) change that may cause ZQ being out of range (> +/-10%) during the mission mode or during the self-refresh periods.
ZCAL on SRX	Y	Y	For DDRPHYC, ZCAL on self-refresh exit and ZCAL at regular interval may be managed by software but it is not required in usual conditions. Regular interval ZCAL may increase the worse case system latency to DDR.
Low power modes	Y	Y	ASR1/HSR1 mode DDRCTRL low-power counters can be used to transition to power-down (with or without clock stop) and to self-refresh after some idle time. The transition to DPD is also supportable with LPDDR2/3 but not used because LPDDR2/3 content. Automatic self-refresh ASR is applicable to DDR3. However, given the heuristic approach with ASR, it may not bring significant power saving versus the software supported self-refresh (SSR).
DDRPHYC custom settings	Y	Y	DDRPHYC settings may be modified: • ZCAL override • Specific features and modes: DQS active gate (DQS gate closed on last DQS falling) • Drift compensation • DQS gate extension, fixed latency/ no bubble

Revision history

Table 19. Document revision history

Date	Version	Changes
6-Feb-2019	1	Initial release.
19-Oct-2021	2	Document remastered due to change of DDR 'tuning strategy'.
01-Mar-2023	3	Updated QoS (quality of service) and scheduling
22-Feb-2024	4	Updated Section 5.3: Frequency range restriction
20-May-2024	5	Updated: • Section 6.9: LPDDR2 configuration example • Section 6.2.2: LPDDR2 specific settings • Section 6.4: Output impedance

Contents

List of tables

List of figures