NXP Semiconductors
Application Notes

Document Number: AN13636
Rev. 0, 07/2022

S32Gx DRAM Content Retention

by: NXP Semiconductors

1. Introduction

S32G is a family of high-performance vehicle network processors targeting automotive applications for gateways, domain and safety controllers and high compute.

The DDR SubSystem inside the chip offers a set of functionalities which can be applied during low power mode for saving power.

The DDR SubSystem (controller, PHY and IOs) supports power saving feature called DDR IO retention which :

- Stops DDR transactions
- Configures DRAM in self-refresh
- Configures DDR Subsystem in IO retention

This application note summarizes the DDR SubSystem power states, the entry-exit sequence implementation of the DDR self-refresh and DDR IO retention modes. This mode can be used for optimal realization of DDR Subsystem power-saving and ensuring DRAM content retention in device Standby mode.

Contents

2. S32Gx DDR IO power state

The DDR IO retention mode is a non-operational low power mode, where entry allows for DRAM contents to be maintained through SOC Standby mode. During DDR IO retention mode, the DDR SubSystem state:

- The DDR PHY enters in internal deep sleep state which is the lowest implemented power state of the DDR PHY
- The DRAM memory should be put into self-refresh
- The VDD and VDD_DDR0 power supply rails (among other) are powered down in this state
- The state of the CKE and MEMRESET pins will be preserved with their last values when VDD is shut off. VDD_IO_DDR0 must remain asserted for the DDR pins to retain their state.

NOTE

- For device power specifications, refer to device Data Sheet.
- For Standby mode implications, refer to device Reference Manual and Application Note “AN12952”.

3. S32Gx DDR IO retention flow

The DDR IO retention mode flow chart mentions several preconditions for configuring DDR PLL and also refers to some DDR modes which are described as follows –

User has to configure operating DRAM frequency in DDR PLL during initialization at Power-on-Reset phase and cannot be changed during Standby entry/exit time with DDR in IO retention mode.

DDR IO retention flow:

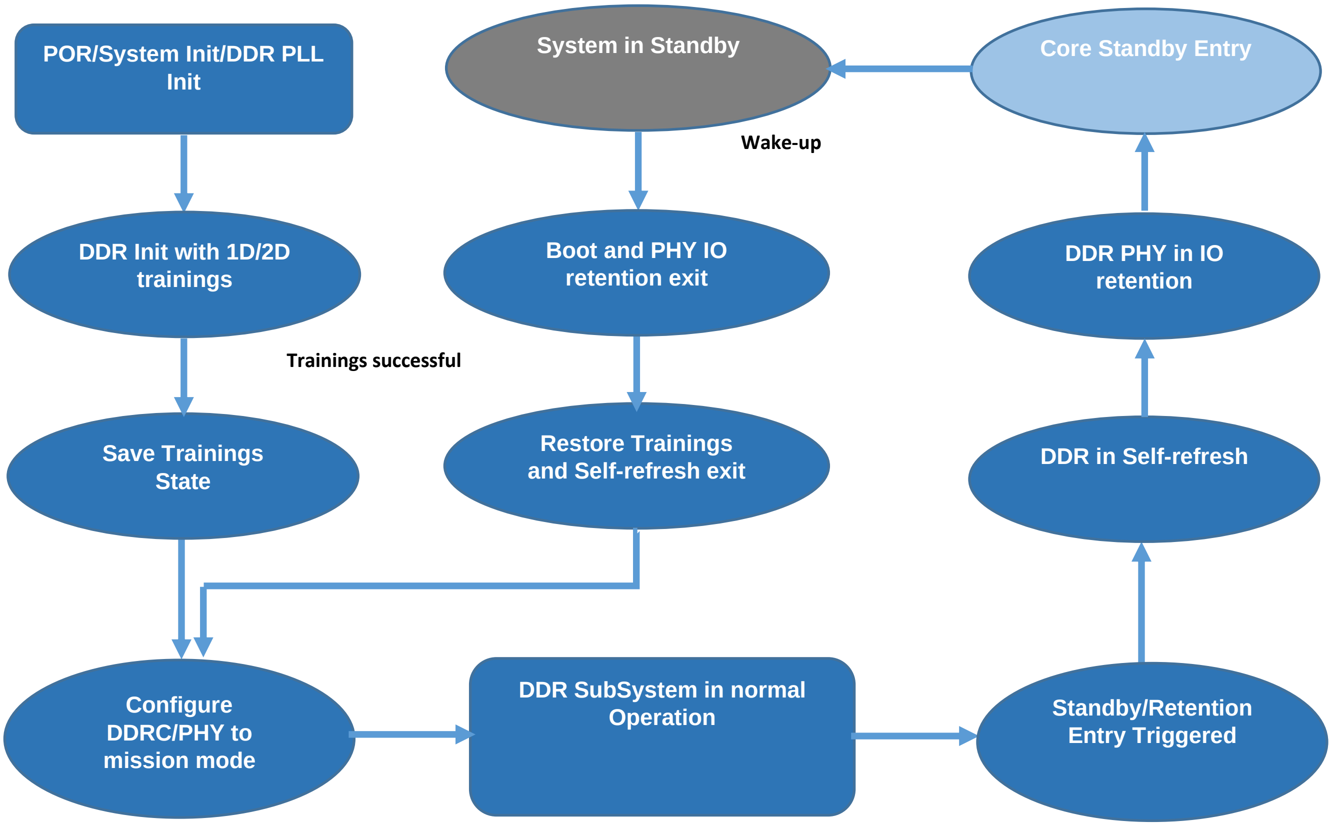
POR/System Init/DDR PLL Init
System in Standby
Core Standby Entry
Wake-up
DDR Init with 1D/2D trainings
Boot and PHY IO retention exit
DDR PHY in IO retention
Trainings successful
Save Trainings State
Restore Trainings and Self-refresh exit
DDR in Self-refresh
Configure DDRC/PHY to mission mode
DDR SubSystem in normal Operation
Standby/Retention Entry Triggered

Figure 1. **DDR IO retention flow**

The above flow diagram dictates about some modes, refer to the below explanation about different modes:

- Normal mode – In this mode the DDR SubSystem module is functional after initialization and calibrations for read/write data from DRAM.
- Mission mode – During mission mode enable, AXI ports are enabled to perform read/write transactions after DDR initialization and trainings.
- Self-refresh mode – When self-refresh is enabled, the DDR controller put the DRAM into self-refresh and controller does not send refresh commands to DRAM memory.
- IO retention mode – When the IO retention is enabled, the DDR PHY module keep the few DDR IO pins active and the CKE and MEMRESET preserved.
- Standby mode – This is the mode when the maximum power saving is entered.

4. DDR IO retention mode implementation

4.1. Pre-condition

- The DDR SubSystem should be initialized successfully with 1D/2D trainings operations after Power-on-Reset
- The Standby RAM must be initialized during normal operations before storing the trainings state
- The static DDR init code (DDR controller and PHY configurations) must be present in non-volatile memory (external flash) after standby exit as the Standby RAM has insufficient memory to store the complete DDR initialization code
- The trainings state must be saved into Standby RAM after DDR initialization after training and prior to the Standby mode entry.

4.2. Entry Procedure

1. Block AXI port from accepting transactions
   a. Set UMCTL2_MP.PCTRL_n[PORT_EN] to 0 to block AXI ports
   b. Poll for UMCTL2_MP.PSTAT[RD_PORT_BUSY_n] = 0 for idle AXI ports
2. Disable scrubber, if enabled
   a. Set UMCTL2_MP.SBRCTL[SCRUB_EN] to 0 to disable scrubber
   b. Poll for UMCTL2_MP.SBRSTAT[SCRUB_BUSY] = 0 for no outstanding read commands
3. Place DRAM in Self Refresh
   a. Set UMCTL2_REGS.PWRCTL[SELFREF_SW] to 1 to move to self-refresh
   b. Poll UMCTL2_REGS.STAT[SELFREF_TYP] (LPDDR4/DDR3L) and STAT[SELFREF_STATE] (LPDDR4 Only) = 2 for self-refresh entry
4. Transition DDR PHY into power saving mode
   a. Set UMCTL2_REGS.DFIMISC to 0
   b. Set UMCTL2_REGS.SWCTL[SW_DONE] to 0
   c. Set UMCTL2_REGS.DFIMISC[DFI_FREQUENCY] to 0x1F
   d. Set UMCTL2_REGS.DFIMISC[DFI_INIT_START] to 1
   e. Poll UMCTL2_REGS.DFISTAT[DFI_INIT_COMPLETE] = 0
   f. Set UMCTL2_REGS.DFIMISC[DFI_FREQUENCY] to 0x1F
   g. Set UMCTL2_REGS.DFIMISC[DFI_INIT_START] to 0
   h. Poll UMCTL2_REGS.DFISTAT[DFI_INIT_COMPLETE] = 1
   i. Set UMCTL2_REGS.SWCTL[SW_DONE] to 1
   j. Poll UMCTL2_REGS.SWSTAT[SW_DONE] = 1
5. Enable IO retention mode holding current state on CKE and MEMRESET
   a. Set DDR_GPR.DDR_RET_CONTROL[DDR_RET_CONTROL] to 0
   b. Set DDR_GPR.DDR_CONFIG_0[MEMORY_RET] to 1

6. Put the whole device in Standby mode by turning off the Core VDD. The VDD_DDR0 and the VDD_IO_DDR0 must remain active during Standby mode.

NOTE

> The complete IO retention mode entry procedure is also provided in the DDR Tool generated DDR initalization code as a function/routine "*ddrss_to_io_retention_mode(void)*" in "*ddr_lp.c*" that can be used by user directly. The routine covers Entry Procedure step #1 to step #5.

4.3. Exit procedure

The device is kept in Standby mode until a wake-up event occurs. A wake-up event the device has to prepare Standby exit and restore Core VDD. The VDD_DDR0 and VDD_IO_DDR0 supplies were remain active in order to keep the DRAM memory in self-refresh mode.

Resets during IO retention mode will lead to the exit of this mode and a reset of the PHY.

1. Set DDR_GPR.DDR_RET_CONTROL[DDR_RET_CONTROL] = 1
2. Restore DDR controller configurations
    a. Load controller registers, refer to Configuration Routines
    b. Set UMCTL2_REGS.INIT0[SKIP_DRAM_INIT] to 3 to skip the DRAM init routine
    c. Set UMCTL2_REGS.PWRCTL[SELFREF_SW] to 1 to maintain self-refresh
    d. Set DDR SS Register REG_GRP0[AXI_PARITY_EN] and REG_GRP0[AXI_PARITY_TYPE] to desired parity setting
    e. Set DDR SS Register REG_GRP0[DFI1_ENABLED] to 1 (LPDDR4 Only)
    f. Set MC_RGM Register RGM_PRST_0[PERIPH_3_RST] to deassert reset to controller and AXI ports
    g. Set UMCTL2_REGS.DBG1 = 0
    h. Set UMCTL2_REGS.RFSHCTL3[DIS_AUTO_REFRESH] to 1
    i. Set UMCTL2_REGS.PWRCTL[POWERDOWN_EN] to 0
    j. Set UMCTL2_REGS.PWRCTL[SELFREF_EN] to 0
    k. Set UMCTL2_REGS.PWRCTL[EN_DFI_DRAM_CLK_DISABLE] to 0
    l. Set UMCTL2_REGS.SWCTL[SW_DONE] to 0
    m. Poll UMCTL2_REGS.SWSTAT[SW_DONE_ACK] = 0
    n. Set UMCTL2_REGS.DFIMISC[DFI_INIT_COMPLETE_EN] to 0
    o. Set UMCTL2_REGS.SWCTL[SW_DONE] to 1
    p. Poll UMCTL2_REGS.SWSTAT[SW_DONE_ACK] =1
3. Restore DDR PHY configurations
    a. Reload the PHY configurations, refer to Configuration Routines steps 2, 3, 4 and 5
    b. Initialize DDR PHY to mission mode
        i. Set UMCTL2_REGS.SWCTL[SW_DONE] to 0
        ii. Poll UMCTL2_REGS.SWSTAT[SW_DONE_ACT] = 0
        iii. Set UMCTL2_REGS.DFIMISC[DFI_INIT_STAT] to 1
        iv. Set UMCTL2_REGS.SWCTL[SW_DONE] to 1

- v. Poll UMCTL2_REGS.SWSTAT[SW_DONE_ACK] = 1
- vi. Poll UMCTL2_REGS.DFISTAT[DFI_INIT_COMPLETE] = 1
- vii. Set UMCTL2_REGS.SWCTL[SW_DONE] to 0
- viii. Poll UMCTL2_REGS.SWSTAT[SW_DONE_ACK] = 0
- ix. Set UMCTL2_REGS.DFI_INIT_START to 0
- x. Set UMCTL2_REGS.DFIMISC[DFI_INIT_COMPLETE_EN] to 1
- xi. Set UMCTL2_REGS.SWCTL[SW_DONE] = 1
- xii. Poll UMCTL2_REGS.SWSTAT[SW_DONE_ACK] = 1

4. Exit DDR controller from self-refresh
   - a. Set UMCTL2_REGS.PWRCTL[SELFREF_SW] to 0
   - b. Poll UMCTL2_REGS.STAT[SELFREF_TYPE] = 0
   - c. Poll UMCTL2_REGS.STAT[OPERATING_MODE] = 1
   - d. Set UMCTL2_REGS.RFSHCTL3[DIS_AUTO_REFRESH] to 0
   - e. Set UMCTL2_REGS.PWRCTL[POWERDOWN_EN] to 1
   - f. Set UMCTL2_REGS.PWRCTL[SELFREF_EN] to 1
   - g. Set UMCTL2_REGS.PWRCTL[EN_DFI_DRAM_CLK_DISABLE] to 1
5. Allow AXI ports to accept transactions
   - a. Set UMCTL2_MP. PCTRL_n[PORT_EN] to 1 to block AXI ports

NOTE

The complete IO retention mode exit procedure is also provided in the DDR Tool generated DDR initalization code as a function/routine "*ddrss_to_normal_mode(uintptr_t csr_array)*" in "*ddr_lp.c*" that can be used by user directly. The routine covers Exit Procedure step #2 to step #5.

4.4. Configuration routines

There are many registers in DDR SubSystem that need to be configured in a sequence during initialization. These registers are configured in DDR Tool input parameters based on DRAM configurations and operating DDR frequency. The DDR init code is generated using DDR Tool firmware init test which can be used in user application.

The Configuration Routines in the DDR init code are described below:

1. Refer to DDR Tool generated code *ddrc_init_cfg* function in *ddr_init* routine

```
/* DDR controller configurations */
ddrc_init_cfg();
```

2. Refer to DDR Tool generated code routines *dq_swap_cfg*, *phy_cfg* and other PHY configurations

```
/* Apply DQ swapping settings */
load_register_cfg(config->dq_swap_cfg_size, config->dq_swap_cfg);
/* Initialize phy cfg module */
load_register_cfg(config->phy_cfg_size, config->phy_cfg);
/* Configure PLL optimal settings */
```

```
set_optimal_pll(config);
```

3. Refer to DDR Tool generated code to restore a saved trainings state from StandbyRAM
   ```
   /* Restore saved trainings state from StandbyRAM */
   load_csr();
   ```
4. Refer to DDR Tool config pie_cfg
   ```
   /*  Load pie image after training has executed */
   load_register_cfg(config->pie_cfg_size, config->pie_cfg);
   ```
5. Refer to DDR Tool generated code under routine "*post_train_setup()*" for CalBusy
   ```
   /*
    * CalBusy.0 = 1, indicates the calibrator is actively calibrating.
    * Wait Calibrating done.
    */
   do {
           tmp32 = readl(DDR_PHYA_MASTER0_CALBUSY);
   } while ((tmp32 and 0x1) != 0);
   ```
6. Refer to DDR Tool generated code to store trained state in StandbyRAM
   ```
   /* Saved trainings state in StandbyRAM  before self-refresh entry*/
   store_csr();
   ```

5. Trainings state retention with DDR PHY registers

There are some DDR PHY registers which should be retained during IO retention mode for saving trainings state to avoid 1D/2D trainings because if trainings are somehow triggered then the DRAM contents will be lost. The registers values should be saved in StandbyRAM to retain during standby/low power mode.

Please refer to the DDR Tool generated code array variable "csr_to_store[]" to retain the DDR PHY registers offset which are given in the generated code "ddr_lp_csr.c".

There are a total of 338 registers (32-bit) those need to be saved to keep 1D and 2D trainings state.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Suitability for use in automotive applications — This NXP product has been qualified for use in automotive applications. If this product is used by customer in the development of, or for incorporation into, products or services (a) used in safety critical applications or (b) in which failure could lead to death, personal injury, or severe physical or environmental damage (such products and services hereinafter referred to as “Critical Applications”), then customer makes the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, safety, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. As such, customer assumes all risk related to use of any products in Critical Applications and NXP and its suppliers shall not be liable for any such use by customer. Accordingly, customer will indemnify and hold NXP harmless from any claims, liabilities, damages and associated costs and expenses (including attorneys' fees) that NXP may incur related to customer's incorporation of any product in a Critical Application.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer’s applications and products. Customer’s responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer’s applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μVision, Versatile — are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

Airfast — is a trademark of NXP B.V.

Altivec — is a trademark of NXP B.V.

CodeWarrior — is a trademark of NXP B.V.

ColdFire — is a trademark of NXP B.V.

ColdFire+ — is a trademark of NXP B.V.

CoolFlux — is a trademark of NXP B.V.

CoolFlux DSP — is a trademark of NXP B.V.

DESFire — is a trademark of NXP B.V.

EdgeLock — is a trademark of NXP B.V.

EdgeScale — is a trademark of NXP B.V.

EdgeVerse — is a trademark of NXP B.V.

eIQ — is a trademark of NXP B.V.

Embrace — is a trademark of NXP B.V.

Freescale — is a trademark of NXP B.V.

GreenChip — is a trademark of NXP B.V.

HITAG — is a trademark of NXP B.V.

ICODE and I-CODE — are trademarks of NXP B.V.

Immersiv3D — is a trademark of NXP B.V.

I2C-bus — logo is a trademark of NXP B.V.

JCOP — is a trademark of NXP B.V.

Kinetis — is a trademark of NXP B.V.

Layerscape — is a trademark of NXP B.V.

MagniV — is a trademark of NXP B.V.

Mantis — is a trademark of NXP B.V.

MCCI — is a trademark of NXP B.V.

MIFARE — is a trademark of NXP B.V.

MIFARE Classic — is a trademark of NXP B.V.

MIFARE FleX — is a trademark of NXP B.V.

MIFARE4Mobile — is a trademark of NXP B.V.

MIFARE Plus — is a trademark of NXP B.V.

MIFARE Ultralight — is a trademark of NXP B.V.

MiGLO — is a trademark of NXP B.V.

MOBILEGT — is a trademark of NXP B.V.

NTAG — is a trademark of NXP B.V.

NXP SECURE CONNECTIONS FOR A SMARTER WORLD — is a trademark of NXP B.V.

PEG — is a trademark of NXP B.V.

Plus X — is a trademark of NXP B.V.

POR — is a trademark of NXP B.V.

PowerQUICC — is a trademark of NXP B.V.

Processor Expert — is a trademark of NXP B.V.

QorIQ — is a trademark of NXP B.V.

QorIQ Qonverge — is a trademark of NXP B.V.

RoadLink — wordmark and logo are trademarks of NXP B.V.

SafeAssure — is a trademark of NXP B.V.

SafeAssure — logo is a trademark of NXP B.V.

SmartLX — is a trademark of NXP B.V.

SmartMX — is a trademark of NXP B.V.

StarCore — is a trademark of NXP B.V.

Symphony — is a trademark of NXP B.V.

Synopsys & Designware — are registered trademarks of Synopsys, Inc.

Synopsys — Portions Copyright © 2021 Synopsys, Inc. Used with permission. All rights reserved.

Tower — is a trademark of NXP B.V.

TriMedia — is a trademark of NXP B.V.

UCODE — is a trademark of NXP B.V.

VortiQa — is a trademark of NXP B.V.

Vybrid — is a trademark of NXP B.V.

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

© NXP B.V. 2022. All rights reserved.

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