AN13657

Using i.MX RT500 FusionF1 DSP in Low-Power Design

Rev. 0 — 6 June 2022 Application note

Document information

Information	Content
Keywords	Fusion F1 DSP, Low-power, DSP
Abstract	MIMXRT595S is a low-power microcontroller designed for low-power applications

1 Introduction

The i.MX RT500 is a family of dual-core microcontrollers for embedded applications featuring an Arm Cortex-M33 CPU combined with a Cadence Xtensa Fusion F1 Audio Digital Signal Processor CPU. Part number with prefix "MIMXRT595S" are the ones with a Fusion F1 AudioDSP inside.

The Cadence Tensilica Fusion F1 DSP is a highly optimized, highly configurable processor geared for efficient execution of data plane algorithms needed for the Internet of Things (IoT), and other applications, such as codec chips, sensor hubs, and narrowband wireless communications.

MIMXRT595S is a low-power microcontroller, that is designed for low-power applications. Those applications apply mostly in battery constraint systems, system power consumption which impacts the battery lifetime is usually a main concern.

The document introduces the means of power optimization when using MIMXRT595S Fusion F1 DSP.

2 DSP power optimization in active mode

This section provides information about changing the DSP running frequency.

2.1 Change DSP running frequency

MIMXRT595S Fusion F1 DSP can run up to 200 MHz. When VDDCore voltage is fixed, the faster DSP runs, the more power it consumes. DSP frequency is adjustable so that the DSP power consumption can be managed.

The DSP relevant clock tree in MIMXRT595S is shown as Figure 1 Through DSPCPUCLKSELA and DSPCLKSELB, there are six clock sources can be used for DSP CPU, including FRO (192 MHz or 96 MHz), external oscillator clock, low-power oscillator (1 MHz), main PLL(PFD0), DSP PLL(PFD1) and 32 K RTC clock.

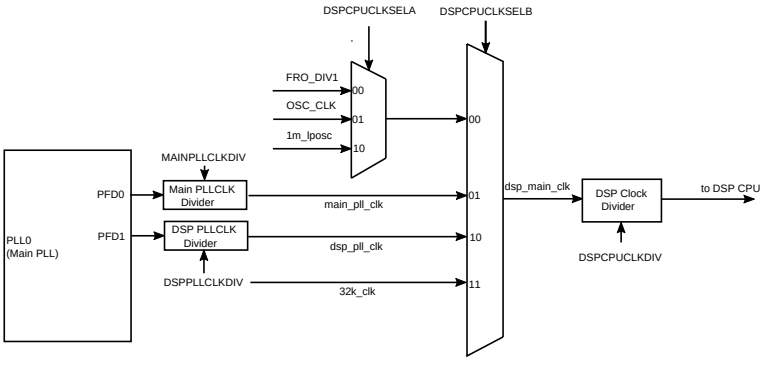

Figure 1. i.MX RT500 Clock tree for DSP

In the clock paths, there are three dividers, MAINPLLCLKDIV, DSP PLLCLKDIV, and DSPCPUCLKDIV. However, those dividers are not designed for adjusting DSP frequency when DSP is running. Instead, there are two means to adjust the clock frequency to DSP CPU, changing PFD output or switching DSP CPU clock sources.

2.1.1 Change PLL PFD

MIMXRT595S system PLL supports four PFD outputs, two of them can be used for DSP clock sources, PFD0 for main PLL and PFD1 for DSP PLL. While DSP is running, the dividers MAINPLLCLKDIV, DSPPLLCLKDIV and DSPCPUCLKDIV cannot be changed, or the DSP must be clock gated. They must be fixed before DSP starts. As DSP main clock source, the PFD output is calculated by formula.

$$PFD\ \ Output = \frac{18}{N} \times FVCO\ \ (N = 12\ \ to\ \ 35)$$

MIMXRT595 system PLL FVCO is 528 MHz, therefore the maximum PFD output frequency is 792 MHz, and minimum PFD output frequency is 271.5 MHz.

If PLL PFD1 is used as a DSP clock, DSP PLL clock divider (DSPPLLCLKDIV) is set “divide by 1” and the DSP CPU clock divider (DSPCPUCLKDIV) is set “divide by 4”, see Table 1. There are 24 configurations, DSP can run the frequency up to 198 MHz when N is 12, and 68 MHz when N is 35.

Table 1. DSP running frequency

Running frequency	configurations											
N	12	13	14	15	16	17	18	19	20	21	22	23
DSP Frequency	198	182.7692	169.7143	158.4	148.5	139.7647	132	125.0526	118.8	113.1429	108	103.3043
N	24	25	26	27	28	29	30	31	32	33	34	35
DSP Frequency	99	95.04	91.38462	88	84.85714	81.93103	79.2	76.64516	74.25	72	69.88235	67.88571

The procedures to change PLL PFD are explained in "6.5.1.28 System PLL0 PFD (SYSPLL0PFD)" of *i.MX RT500 Low-Power Crossover Processor Data Sheet with Addendum* (document IMXRT500EC), the function CLOCK_InitSysPfd in MCUXpresso SDK Builder implements the procedures. Following code are snippet for adjusting DSP running frequency:

```
/* kCLOCK_DivDspCpuClk = 4, DSP is running at 48MHz */
CLOCK_AttachClk(kFRO_DIV1_to_DSP_MAIN_CLK);

/* Enable dsp PLL clock 528 * 18 / dsp_pfd_divider. */
CLOCK_InitSysPfd(kCLOCK_Pfd1, dsp_pfd_divider);

CLOCK_AttachClk(kDSP_PLL_to_DSP_MAIN_CLK);
```

2.1.2 Switch DSP clock source

DSP clock source can be switched while DSP is running. There are six clock sources for DSP main clock. DSP clock source can be changed using the function CLOCK_AttachClk in MCUXPRESSO SDK.

2.2 Adjust VDDCore voltage

The minimum core voltage is decided by the max frequency between ARM and DSP. As claimed in MIMXRT595S data sheet, the map between ARM/DSP frequency and VDDCore voltage is listed in Table 2.

Table 2. ARM/DSP frequency and VDDCore voltage

VDDCore Voltage	Max Frequency between ARM and DSP
0.7	60 MHz

Table 2. ARM/DSP frequency and VDDCore voltage*...continued*

0.8	100 MHz
0.9	192 MHz
1.0	230 MHz

VDD Core voltage should be reviewed by checking the max frequency between ARM and DSP. The voltage should be raised **before** DSP switching to higher frequency, and the voltage should be reduced **after** DSP switching to lower frequency.

Table 3 lists the IDDCore current measurement in MIMXRT595S EVK in different DSP frequencies and VDDCore voltage.

Table 3. DSP frequency and IDDCore while DSP is running FFT

HCLK = 198 MHz, VDD Core = 1.0 V, CM33 Sleep, code execution in RAM, DSP running 1024 taps FFT in a while loop												
DSP Frequency (mHz)	198	182.7692	169.7143	158.4	148.5	139.7647	132	125.0526	118.8	113.1429	108	103.3043
IDD Core (mA)	31.71	30.2	28.9	27.77	26.74	25.85	25.07	24.36	23.72	23.14	22.61	22.13
DSP Frequency (mHz)	99	95.04	91.38462	88	84.85714	81.93103	79.2	76.64516	74.25	72	69.88235	67.88571
IDD Core (mA)	21.68	21.26	20.88	20.53	20.2	19.89	19.6	19.34	19.08	18.83	18.61	18.4
HCLK = 12 MHz, VDD Core adjustable, CM33 Sleep, code execution in RAM, DSP running 1024 taps FFT in a while loop												
DSP Frequency (mHz)	198	182.7692	169.7143	158.4	148.5	139.7647	132	125.0526	118.8	113.1429	108	103.3043
VDD Core (V)	1.0	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
IDD Core (mA)	24	19.89	18.64	17.58	16.64	15.79	15.06	14.39	13.79	13.26	12.74	12.29
DSP Frequency (mHz)	99	95.04	91.38462	88	84.85714	81.93103	79.2	76.64516	74.25	72	69.88235	67.88571
VDD Core (V)	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8
IDD Core (mA)	10.36	10.02	9.69	9.39	9.12	8.86	8.61	8.39	8.18	7.98	7.78	7.61

Analyzing the power consumption data, there are some guidelines for power optimization while DSP is running.

1. Fusion F1 uses TCM bus to access memory. If ARM is in sleep state, to reduce the HCLK frequency as much as possible.
2. DSP frequency at 182.7692 MHz and 99 MHz are the “sweet spot” in MIMXRT595S, ARM/DSP running at these frequencies can have the best performance/power ratio.

For more details, see Figure 2.

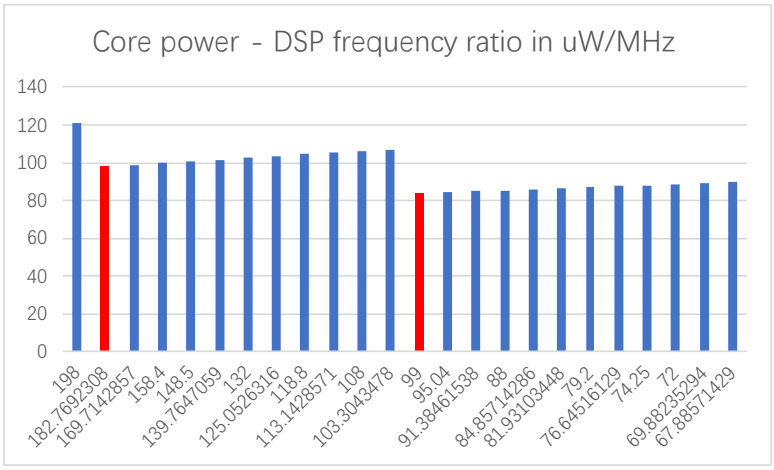

Figure 2. Core power – DSP frequency ratio in uW/MHz

Fig 2 Core power – DSP frequency ratio in uW/MHz

3 DSP power optimization in standby mode

There are two ways to stop DSP operation, one is called “DSP stall”, and another is called “DSP power down”. The difference between “DSP stall” and “DSP power down” is that “DSP stall” can be resumed without DSP reset, whereas DSP must be reset for another run after “DSP power down”.

3.1 DSP stall

DSP stall mode is controlled by SYSCTL0:DSPSTALL register. DSP operation can be suspended by setting DSPSTALL bit 0, and the operation can be resumed by clear bit0. The DSP Stall mode is supported by MCUXPRESSO SDK API DSP_Start and DSP_Stop.

As the DSP_Start and DSP_Stop are asynchronous, instruction barriers should be used after the DSPSTALL register is set.

```
DSP_Start();
__DSB();
__ISB();

DSP_Stop();
__DSB();
__ISB();
```

Similar to DSP stall mode, in DSP bare-metal programming, DSP operation can be suspended by WAITI instruction and in DSP XOS programming, the operation can be suspended by XOS thread sleep APIs. Both WAITI instruction and XOS thread sleep method can suspend DSP. The power consumption in DSP stall mode is similar to the cases when DSP operation is suspended using DSP programming methods.

As listed in Table 4, power consumption in DSP suspended mode is lower than DSP active mode, comparing with power data in Table 3.

Table 4. DSP frequency and IDDCore while DSP is suspended

HCLK = 198 MHz, VDD Core = 1.0 V, CM33 Sleep, code execution in RAM, DSP suspend												
DSP Frequency (mHz)	198	182.7692	169.7143	158.4	148.5	139.7647	132	125.0526	118.8	113.1429	108	103.3043
IDD Core (mA)	13.89	13.67	13.49	13.33	13.19	13.07	12.96	12.86	12.77	12.69	12.62	12.55
DSP Frequency (mHz)	99	95.04	91.38462	88	84.85714	81.93103	79.2	76.64516	74.25	72	69.88235	67.88571
IDD Core (mA)	12.48	12.43	12.38	12.33	12.28	12.24	12.2	12.16	12.12	12.09	12.06	12.03
HCLK = 12 MHz, VDD Core adjustable, CM33 Sleep, code execution in RAM, DSP suspend												
DSP Frequency (mHz)	198	182.7692	169.7143	158.4	148.5	139.7647	132	125.0526	118.8	113.1429	108	103.3043
VDD Core (V)	1.0	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
IDD Core (mA)	5.53	4.34	4.16	4.01	3.88	3.76	3.65	3.56	3.47	3.39	3.33	3.26
DSP Frequency (mHz)	99	95.04	91.38462	88	84.85714	81.93103	79.2	76.64516	74.25	72	69.88235	67.88571
VDD Core (V)	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8
IDD Core (mA)	2.63	2.58	2.53	2.49	2.45	2.42	2.39	2.35	2.33	2.3	2.27	2.25

It makes sense to set, even lower clock frequency to DSP while DSP is suspended for lower power consumption. Low-power oscillator or 32 kHz clocks are options.

Table 5 lists the IDDCore power consumption when 32 kHz clock or 1 MHz low-power oscillator are used as DSP clock source. There is no significant difference in between.

Table 5. IDDCore in DSP stall mode

Power consumption	IDDCore
DSP Suspended, 32 kHz clock source, VDDCore 0.8 V, code execution in SRAM, CM33 Sleep	1.62 mA
DSP Suspended, 1 MHz clock source, VDDCore 0.8 V, code execution in SRAM, CM33 Sleep	1.62 mA

3.2 DSP power down

DSP power and clock can be switched off to save more power when DSP is inactive. In MCUXPRESSO SDK, DSP_Deinit is designed for the purpose. The SRAM for the DSP data region (read/write data region, heap, stack) can be powered off, both SRAM array power and SRAM periphery power should be off. SRAM region for the DSP text data can be optionally retained or powered off. If text data is retained, the data can be reused for next DSP running without copying. In this case, the SRAM periphery power can be off and SRAM array power should be kept for the DSP text region.

If the system maximum frequency is lower, DSP clock source can be switched off and the VDDCore voltage can be reduced.

The code snippet below shows DSP power down procedures.

```
    CLOCK_EnableClock(kCLOCK_Pmc);
    DSP_Deinit();

    /* Power Down System PLL */
    POWER_EnablePD(kPDRUNCFG_PD_SYSPLL_LDO);
    POWER_EnablePD(kPDRUNCFG_PD_SYSPLL_ANA);
    POWER_EnablePD(kPDRUNCFG_PD_SYSXTAL);
    /*
        DSP_BOOT_ADDRESS 0x00400000 (Partition 28-31)
        DSP_SRAM_ADDRESS 0x00300000 (Partition 24-27)
    */
```

```
    POWER_EnablePD(kPDRUNCFG_PPD_SRAM_IF24);
    POWER_EnablePD(kPDRUNCFG_PPD_SRAM_IF25);
    POWER_EnablePD(kPDRUNCFG_PPD_SRAM_IF26);
    POWER_EnablePD(kPDRUNCFG_PPD_SRAM_IF27);

    POWER_EnablePD(kPDRUNCFG_PPD_SRAM_IF28);
    POWER_EnablePD(kPDRUNCFG_PPD_SRAM_IF29);
    POWER_EnablePD(kPDRUNCFG_PPD_SRAM_IF30);
    POWER_EnablePD(kPDRUNCFG_PPD_SRAM_IF31);

    POWER_EnablePD(kPDRUNCFG_APD_SRAM_IF24);
    POWER_EnablePD(kPDRUNCFG_APD_SRAM_IF25);
    POWER_EnablePD(kPDRUNCFG_APD_SRAM_IF26);
    POWER_EnablePD(kPDRUNCFG_APD_SRAM_IF27);
    POWER_ApplyPD();


 BOARD_SetPmicVoltageForFreq(CLOCK_GetFreq(kCLOCK_CoreSysClk),
 0);

    CLOCK_DisableClock(kCLOCK_Pmc);
```

Reinitialization is required for the next DSP run, the reinitialization is reversed operations of DSP power down. The code snippet below shows DSP reinitialization procedures.

```
    pca9420_sw1_out_t volt;
    dsp_copy_image_t text_image;
    dsp_copy_image_t data_image;
    uint32_t size;
    text_image.destAddr  = DSP_BOOT_ADDRESS;
    data_image.destAddr = DSP_SRAM_ADDRESS;
#if defined(__CC_ARM)
    size = (uint32_t)&Image$$DSP_REGION$$Length;
#elif defined(__ICCARM__)
#pragma section = "__dsp_text_section"
    text_image.srcAddr    = DSP_IMAGE_TEXT_START;
    text_image.size            = DSP_IMAGE_TEXT_SIZE;
#pragma section = "__dsp_data_section"
    data_image.srcAddr = DSP_IMAGE_DATA_START;
    data_image.size    = DSP_IMAGE_DATA_SIZE;
#elif defined(__GNUC__)
    text_image.srcAddr    = DSP_IMAGE_TEXT_START;
    text_image.size            = DSP_IMAGE_TEXT_SIZE;
    data_image.srcAddr = DSP_IMAGE_DATA_START;
    data_image.size    = DSP_IMAGE_DATA_SIZE;
#endif
    CLOCK_EnableClock(kCLOCK_Pmc);

    POWER_DisablePD(kPDRUNCFG_PPD_SRAM_IF24);
    POWER_DisablePD(kPDRUNCFG_PPD_SRAM_IF25);
    POWER_DisablePD(kPDRUNCFG_PPD_SRAM_IF26);
    POWER_DisablePD(kPDRUNCFG_PPD_SRAM_IF27);

    POWER_DisablePD(kPDRUNCFG_PPD_SRAM_IF28);
    POWER_DisablePD(kPDRUNCFG_PPD_SRAM_IF29);
    POWER_DisablePD(kPDRUNCFG_PPD_SRAM_IF30);
    POWER_DisablePD(kPDRUNCFG_PPD_SRAM_IF31);

    POWER_DisablePD(kPDRUNCFG_APD_SRAM_IF24);
    POWER_DisablePD(kPDRUNCFG_APD_SRAM_IF25);
```

```
    POWER_DisablePD(kPDRUNCFG_APD_SRAM_IF26);
    POWER_DisablePD(kPDRUNCFG_APD_SRAM_IF27);
    POWER_ApplyPD();

    POWER_DisablePD(kPDRUNCFG_PD_SYSXTAL);
    POWER_ApplyPD();
    /* Updated XTAL oscillator settling time */
    POWER_UpdateOscSettlingTime(BOARD_SYSOSC_SETTLING_US);
    /* Enable system OSC */
    CLOCK_EnableSysOscClk(true, true,
 BOARD_SYSOSC_SETTLING_US);
    /* Sets external XTAL OSC freq */
    CLOCK_SetXtalFreq(BOARD_XTAL_SYS_CLK_HZ);

    POWER_DisablePD(kPDRUNCFG_PD_SYSPLL_LDO);
    POWER_DisablePD(kPDRUNCFG_PD_SYSPLL_ANA);
    POWER_ApplyPD();

    /* Configure SysPLL0 clock source */
    CLOCK_InitSysPll(&g_sysPllConfig_BOARD_BootClockRUN);

    /* Enable dsp PLL clock 792MHz. */
    CLOCK_InitSysPfd(kCLOCK_Pfd1, dsp_pfd_divider);
    /*Let DSP run on SYS PLL PFD1 with divider 2 (198Mhz). */
    CLOCK_AttachClk(kDSP_PLL_to_DSP_MAIN_CLK);
    CLOCK_SetClkDiv(kCLOCK_DivDspCpuClk, 4);

    volt =
 BOARD_CalcVoltLevel(CLOCK_GetFreq(kCLOCK_CoreSysClk),
 CLOCK_GetFreq(kCLOCK_DspCpuClk));
    current_pmic_voltage = volt;

    /* Configure PMIC Vddcore value according to main/dsp
 clock. */

 BOARD_SetPmicVoltageForFreq(CLOCK_GetFreq(kCLOCK_CoreSysClk),
    CLOCK_GetFreq(kCLOCK_DspCpuClk));

    /* Set DSP to use vector base address 0x400000. */
    DSP_SetVecRemap(kDSP_StatVecSelAlternate, 0);

    /* Initializing DSP core */
    DSP_Init()

    /* Copy data to DSP_TCM */
    /* DSP Code is not necessary to copied if text region SRAM
 is retained */
//  DSP_CopyImage(&text_image);
    DSP_CopyImage(&data_image);

    /* Run DSP core */
    DSP_Start();
    __DSB();
    __ISB();
    CLOCK_DisableClock(kCLOCK_Pmc);
```

Note: *DSP_Init and DSP_Deinit must be called with low-power oscillator powered.*

The DSP module has an associated supply switch with the module operation release based on LPOSC clock. The module reset assert and de-assert operations require

LPOSC active (2 clock pulses minimum). LPOSC can be disabled between module reset operations. See Table 6 for IDDCore power consumption in DSP clock off mode.

Table 6. IDDCore in DSP Power down mode

Power consumption	IDDCore
DSP power down, VDDCore 0.8 V, CM33 Sleep, Code execution in SRAM	1.02 mA

3.3 Application information

There are two options for DSP standby mode. The characters of DSP power-down mode and DSP stall mode are listed below. Developer can choose the mode which is fit for the applications.

DSP power down:

1. Lower power consumption
2. The released SRAM can be used for other purpose
3. Reinitialize for next DSP run
4. ARM and DSP application state should be synchronized and has to be shared to use peripherals, for example, MU, SEMA42 should be reset. (if used)

DSP stall:

1. Higher power consumption
2. SRAM for DSP data must be kept
3. Fast DSP operation resuming

4 References

- *i.MX RT500 Low-Power Crossover Processor Data Sheet with Addendum* (document IMXRT500EC)
- *i.MX RT500 Low-Power Crossover Processor Reference Manual with Addendum* (document IMXRT500RM)
- *Xtensa XOS*Tensilica Tools
  Note: *User must log in and accepts the license to download document, the document can be found only after you install the Tensilica Tools.*
- *Fusion F1 DSP user's guide*Tensilica Tools
  Note: *User must log in and accepts the license to download document.*

Revision history

Table 7 summarizes the changes done to this document since the initial release.

Revision history

Revision number	Date	Substantive changes
0	06 June 2022	Initial release

5 Legal information

5.1 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.

5.2 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.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

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.

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.

Suitability for use in non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that 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.

5.3 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.

Contents

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

Date of release: 6 June 2022
Document identifier: AN13657