AN14913

PCIe Power Management using ASPM

Rev. 2.0 — 1 April 2026 Application note

Document information

Information	Content
Keywords	AN14913, ASPM, i.MX 95, PCIe, power management
Abstract	This application note provides a detailed view of PCIe ASPM link power management on i.MX 95.

1 Introduction

This application note provides a detailed view of PCIe ASPM link power management on i.MX 95. It also provides a step-by-step guide to exercise PCIe ASPM on the i.MX Linux BSP. The board used in the test is i.MX 95 19 mm x 19 mm EVK (IMX95LPD5EVK-19).

1.1 PCIe power management overview

PCIe power management (PM) aims at reducing power consumption in PCIe devices. It does that by putting PCIe devices to low power states when they are idle and bringing them back to full power state whenever required. PCIe devices such as graphic cards and storage devices can consume significant power, especially when active. Power management allows them to scale down their power consumption during idle or low-activity periods.

In PCIe, power management is exercised at two levels: Link and Device.

1. **Link** – This power management conserves power on the serial link within a PCIe fabric when there is no active data transfer. It also conserves power even when the device is fully powered on. ASPM is responsible for managing the power consumption of a PCIe link. Link states such as L0, L0s, L1, L1 substates (L1ss), L2, and L3 fall under this category.
   - L0 – It is also referred to as L0 Active. It is a normal operational state in which data is sent and received. In this state, the link is fully powered and functional, allowing seamless communication between the Root Complex (RC) and Endpoint (EP).
   - L0s – The ‘s’ in the name means ‘standby’. The link logically remains at L0 but physical signaling is reduced. No data transfer occurs in this state, minimal activity to keep the link going.
   - L1 – This state provides greater power savings than L0s but with a cost of greater recovery latency. Also known as L1.0, it results in the deactivation of the ‘Link and Transaction’ layer within each device.
   - L2 – In this link state, power is aggressively conserved, with most of the Transmitter and Receiver shut off. Main power and clock are not guaranteed but Aux power is available. L2 support is optional as it is dependent upon the presence of Auxiliary power on the board. Since, i.MX 95 does not support Auxiliary power, therefore, the PCIe components cannot go to L2 link state.
   - L3 – When no power is present whatsoever, the component is said to be in the L3 state.

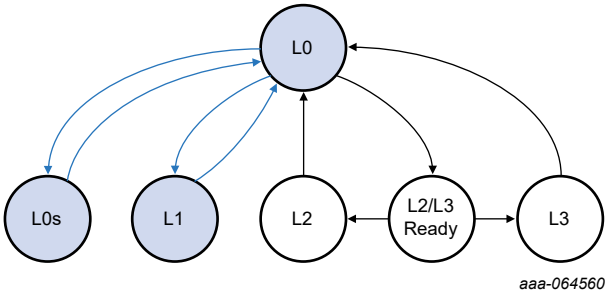


Figure 1. PCIe link power management states

Section 1.3 covers details about ASPM L1 substates state that goes a few steps beyond L1 state in terms of power savings.

2. **Device** – Device power management involves putting the entire device into low-power states when not in use. PCIe power management is responsible for managing the power states of the entire device. Device states such as D0, D1, D2, D3Hot, and D3Cold fall under this category.

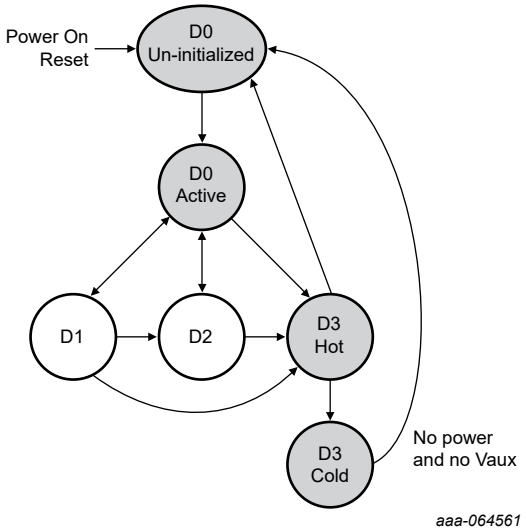


Figure 2. PCIe device power management states

1.2 i.MX 95 overview

The i.MX 95 is a high-performance applications processor designed for next-generation edge computing applications across automotive, industrial, medical, and IoT domains.

- It has six Arm Cortex-A55 application cores (up to 2.0 GHz).
- A high-performance Arm Cortex-M7 and a low-power Cortex-M33 for real-time and safety-critical tasks.
- PCIe capabilities:
  - 2x PCIe Gen 3.0 interfaces, each supporting x1 lane.
  - Operable in RC and EP mode.

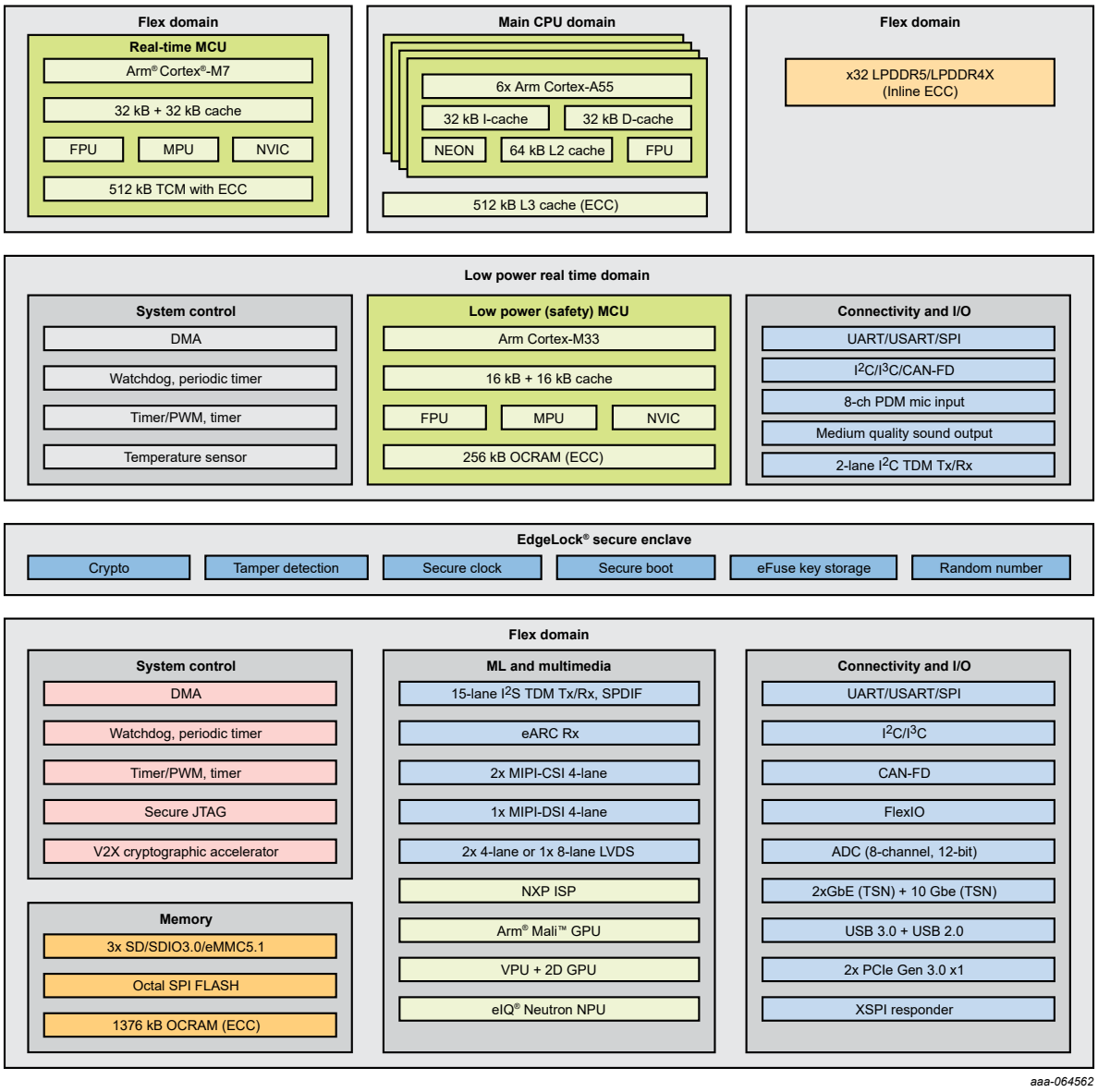


Figure 3. i.MX 95 applications processor block diagram

1.3 Link power management using PCIe ASPM

Active State Power Management (ASPM) is a feature in PCIe that allows link power conservation even when the software has not put the device in a low-power state. Hardware solely handles the transitions into and out of ASPM once the software has enabled ASPM in the `Active State PM Control` field of the `Link control` register. In Section 1.1, the link states along with the L0 and L1(aka L1.0) states have been defined. The transition from L0 to L1 is initiated when the link is idle (no pending transactions) and both the link partners support ASPM L1. Both upstream and downstream ports detect a link idle condition and either of them can initiate the L1 entry by sending an `Electrical Idle Ordered Set` to the other link partner. L1 results in the deactivation of the 'Link and Transaction' layer within each device, meaning that data can no longer be transferred between the link partners. L1 state can optionally have substates–L1.1 and L1.2. This section covers the L1 ASPM substates that go a step further and give a finer power control. Figure 4 depicts the link state transitions to/from L1 state.

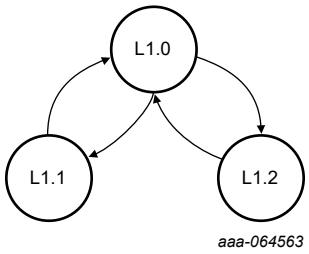


Figure 4. Link states transition

L1.1

- In the L1.1 link state, the PLL, RX/TX circuits are OFF but Link common mode voltages are maintained to allow faster recovery from low-power state.
- Power savings are moderate.
- A bidirectional open-drain clock request signal, that is, CLKREQ# is used for entering and exiting from this state.
- Detecting Electrical Idle does not require the Upstream and Downstream ports to be enabled.

L1.2

- In the L1.2 link state, the PLL, RX/TX circuits are OFF with no requirement to maintain the Link common mode voltages.
- Power savings are more than L1.1 but higher exit latency than L1.1.
- A bidirectional open-drain clock request signal, that is, CLKREQ# is used for entering and exiting from this state.
- Detecting Electrical Idle does not require the Upstream and Downstream ports to be enabled.

After reaching L1.0 state from L0, the controller is capable of automatically moving to deeper Powersaving L1 substates (L1.1 and L1.2), if the device supports it. This transition happens using the CLKREQ# signal.

It is important to understand the role of the CLKREQ# signal in ASPM. CLKREQ# is a sideband signal used in PCIe to manage the reference clock supplied to the PCIe subsystem. It enables dynamic clock management, which is critical for mobile and low-power platforms. The PCIe device can deassert this signal to indicate the platform that it no longer needs the reference clock. The platform can then turn off the clock to save power. When the PCIe device must resume the activity, it asserts the signal to request the clock back from the platform. In L1.2, for example, the reference clock is cut off, so for the device to request the clock back, this signal is used. When the device wants to exit the L1.2 low power state, it asserts the CLKREQ# low.

As CLKREQ# is a bidirectional open-drain signal, both host and device can potentially drive this signal as depicted in Figure 5.

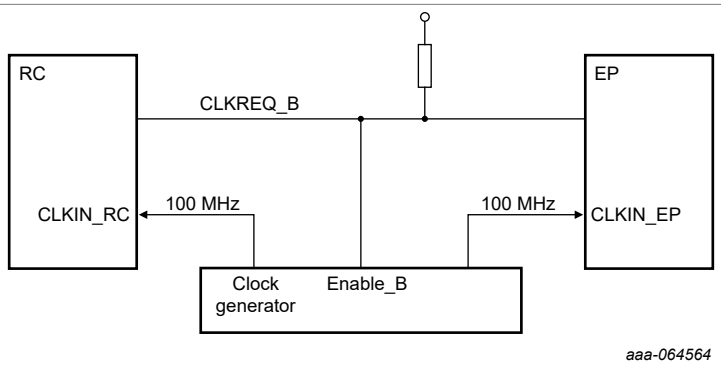


Figure 5. CLKREQ# signal driven by both RC and EP

1.3.1 PCIe device entry and exit to/from ASPM L1 substates

The description below is based on the following setup.

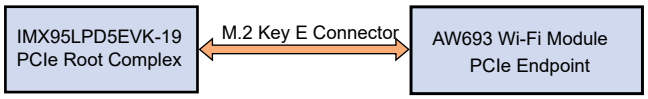


Figure 6. PCIe test setup: EVK as RC with AW693 Wi-Fi module EP

The PCIe device (if supported) should advertise its support for ASPM L1.1 or L1.2 by setting `ASPM L1.1` and `ASPM L1.2` bits of L1 PM `Substates Extended Capability` Register. These bits setting must be performed at the downstream port before it is done at the upstream port.

To verify if the PCIe device has enabled ASPM, you can check the `Link capability` register of the device. Alternatively, you can also check the output of `lspci -vvv` command on the Linux console of the RC.

Example output:

```
        DevSta: CorrErr+ NonFatalErr- FatalErr- UnsupReq- AuxPwr+ TransPend-
        LnkCap: Port #0, Speed 8GT/s, Width x4, ASPM L0s L1, Exit Latency L0s <1us, L1 <8us
                ClockPM+ Surprise- LLActRep- BwNot- ASPMOptComp+
        LnkCtl: ASPM L0s L1 Enabled; RCB 64 bytes, LnkDisable- CommClk+
                ExtSynch- ClockPM+ AutWidDis- BWInt- AutBWInt-
        LnkSta: Speed 8GT/s, Width x1 (downgraded)
                TrErr- Train- SlotClk+ DLActive- BWMgmt- ABWMgmt-
        DevCap2: Completion Timeout: Range ABCD, TimeoutDis+ NROPrPrP- LTR+
                 10BitTagComp- 10BitTagReq- OBFF Not Supported, ExtFmt- EETLPPrefix-
                 EmergencyPowerReduction Not Supported, EmergencyPowerReductionInit-
                 FRS- TPHComp- ExtTPHComp-
                 AtomicOpsCap: 32bit- 64bit- 128bitCAS-
        DevCtl2: Completion Timeout: 50us to 50ms, TimeoutDis-
                 AtomicOpsCtl: ReqEn-
                 IDOReq- IDOCompl- LTR- EmergencyPowerReductionReq-
                 10BitTagReq- OBFF Disabled, EETLPPrefixBlk-
        LnkCap2: Supported Link Speeds: 2.5-8GT/s, Crosslink- Retimer- 2Retimers- DRS-
        LnkCtl2: Target Link Speed: 8GT/s, EnterCompliance- SpeedDis-
                 Transmit Margin: Normal Operating Range, EnterModifiedCompliance- ComplianceSOS-
                 Compliance Preset/De-emphasis: -6dB de-emphasis, 0dB preshoot
        LnkSta2: Current De-emphasis Level: -6dB, EqualizationComplete+ EqualizationPhase1+
                 EqualizationPhase2+ EqualizationPhase3+ LinkEqualizationRequest-
                 Retimer- 2Retimers- CrosslinkRes: unsupported
Capabilities: [b0] MSI-X: Enable+ Count=16 Masked-
        Vector table: BAR=7 offset=ffffff8
        PBA: BAR=0 offset=00002100
Capabilities: [100 v2] Advanced Error Reporting
        UESta:  DLP- SDES- TLP- FCP- CmpltTO- CmpltAbrt- UnxCmplt- RxOF- MalfTLP-
                ECRC- UnsupReq- ACSViol- UncorrIntErr- BlockedTLP- AtomicOpBlocked- TLPBlockedErr-
                PoisonTLPBlocked- DMWrReqBlocked- IDECheck- MisIDETLP- PCRC_CHECK- TLPXlatBlocked-
        UEMsk:  DLP- SDES- TLP- FCP- CmpltTO- CmpltAbrt- UnxCmplt- RxOF- MalfTLP-
                ECRC- UnsupReq- ACSViol- UncorrIntErr+ BlockedTLP- AtomicOpBlocked- TLPBlockedErr-
                PoisonTLPBlocked- DMWrReqBlocked- IDECheck- MisIDETLP- PCRC_CHECK- TLPXlatBlocked-
        UESvrt: DLP+ SDES+ TLP- FCP+ CmpltTO- CmpltAbrt- UnxCmplt- RxOF+ MalfTLP+
                ECRC- UnsupReq- ACSViol- UncorrIntErr+ BlockedTLP- AtomicOpBlocked- TLPBlockedErr-
                PoisonTLPBlocked- DMWrReqBlocked- IDECheck- MisIDETLP- PCRC_CHECK- TLPXlatBlocked-
        CESta:  RxErr- BadTLP- BadDLLP+ Rollover- Timeout- AdvNonFatalErr- CorrIntErr- HeaderOF-
        CEMsk:  RxErr- BadTLP- BadDLLP- Rollover- Timeout- AdvNonFatalErr+ CorrIntErr+ HeaderOF+
        AERCap: First Error Pointer: 00, ECRCGenCap+ ECRCGenEn- ECRCChkCap+ ECRCChkEn-
                MultHdrRecCap- MultHdrRecEn- TLPPfxPres- HdrLogCap-
        HeaderLog: 00000000 00000000 00000000 00000000
Capabilities: [158 v1] Secondary PCI Express
        LnkCtl3: LnkEquIntrruptEn+ PerformEqu+
        LaneErrStat: LaneErr at lane: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Capabilities: [178 v1] Latency Tolerance Reporting
        Max snoop latency: 0ns
        Max no snoop latency: 0ns
Capabilities: [180 v1] L1 PM Substates
        L1SubCap: PCI-PM_L1.2+ PCI-PM_L1.1+ ASPM_L1.2+ ASPM_L1.1+ L1_PM_Substates+
                  PortCommonModeRestoreTime=10us PortTPowerOnTime=10us
        L1SubCtl1: PCI-PM_L1.2- PCI-PM_L1.1- ASPM_L1.2- ASPM_L1.1-
                   T_CommonMode=0us LTR1.2_Threshold=26016ns
```

Figure 7. `lspci` output for an endpoint

As shown in Figure 7, it is observed that L1 ASPM and L1 substates are enabled for this device.

Assuming that the PCIe link is in L1 (L1.0) state and both the link partners support L1 substates. The DLLP layer exchanges power management messages to prepare them for the transition to L1.1/L1.2.

As a part of the protocol, these messages are exchanged between RC and EP. At the end of the message exchanges, the PCIe controller moves to a state machine [L1.1/L1.2] in which clock can be cut off.

The following describes the message exchange sequence for L0 to L1 substates entry and L1 substates exit to L0.

- L0 to L1 ASPM substates entry:
  1. The downstream component sends the `PM_Active_State_Request_L1` DLLP repeatedly to the Upstream component. After receiving this request, the upstream component can either accept by sending the `PM_Request_Ack` DLLP or reject by sending the `PM_Active_State_Nak` TLP.

2. Upon getting the PM_Request_Ack DLLP, the downstream port stops sending the request DLLP and disables the DLLP and TLP transmission, places its transmit lanes into Electrical Idle state.
3. When the upstream component receives an Electrical Idle ordered set on its Receive lanes, it stops sending the PM_Request_Ack DLLP. It disables DLLP and TLP transmission, placing its transmit lanes into Electrical Idle state.
4. At this point, the link is in the L1 state.
5. If the PCIe device supports L1 substates and advertises it in its link capabilities register, it can deassert CLKREQ# to indicate that it does not need the clock. The system then turns off the clock. The link goes to L1 substates (L1.1 or L1.2).
6. When both link partners support L1.1 and L1.2, the L1 PM Substates Protocol dynamically selects the entry state (L1.1 or L1.2) based on their advertised capabilities. It decides the entry state basis on the lowest common denominator of what both partners support. In practice, if both partners support L1.2 and the system policy permits, the link enters L1.2 for maximum power savings. However, if only L1.1 is allowed or only one partner supports L1.1, then L1.1 is used instead.

- L1 substates to L0 exit:
  1. EP asserts the CLKREQ# signal to indicate the upstream component that it needs reference clock.
  2. The system on board detects the asserted CLKREQ# signal and turns on the reference clock using its power management logic.
  3. Once the clock is stable, the device follows the standard L1 exit protocol, where the LTSSM transitions the link through the Recovery state using TS1 and TS2 ordered sets. This exchange of TS1/TS2 ordered sets is essential to re-establish bit and symbol lock, perform lane-to-lane deskew, and ensure both the transmitter and receiver are fully synchronized before resuming normal operation. After the retraining is successful, the link goes to L0.

2 Enabling PCIe ASPM on i.MX 95 19 mm x 19 mm EVK

This section describes the hardware, software, and configuration steps required to enable and validate PCIe ASPM functionality on the i.MX 95 19 mm × 19 mm EVK.

2.1 Hardware setup

The setup includes the following hardware components:

1. i.MX 95 19 mm x 19 mm EVK
2. AW693 M.2 Wi-Fi module - i.MX 95 19 mm x 19 mm EVK has an M.2 PCIe slot to which AW693 is connected

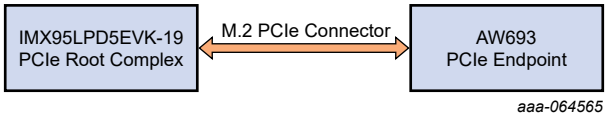


Figure 8. EVK and Wi-Fi module connection

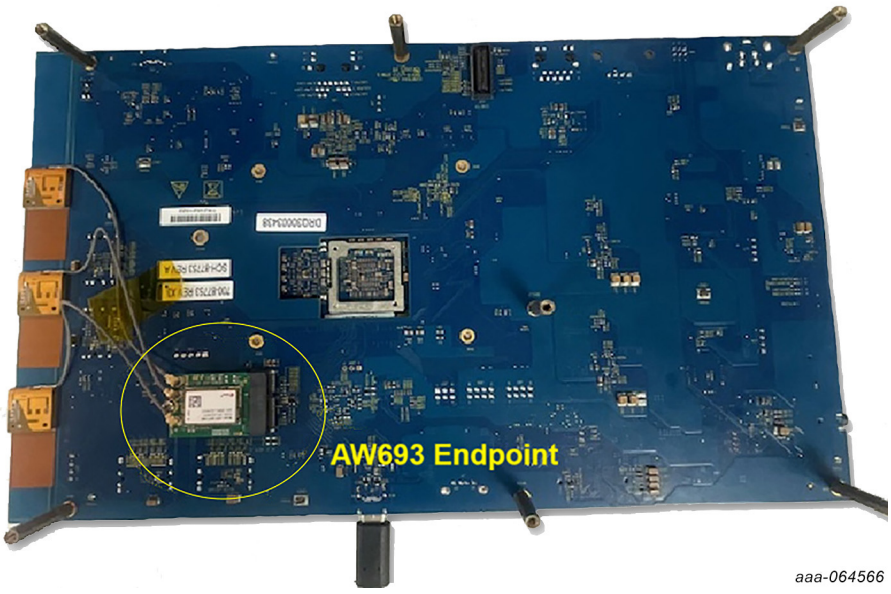


Figure 9. AW693 as EP connected to i.MX 95 19 mm x 19 mm EVK through M.2 Key-E slot

2.2 Software setup

This section describes:

- Software components needed
- Linux kernel configurations
- Preparation and deployment of the build on i.MX 95 19 mm x 19 mm EVK

The setup requires the following software components:

- Linux BSP version: 6.6.52
- DTB: imx95-19x19-evk.dtb

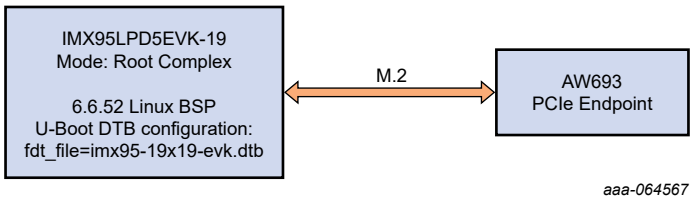


Figure 10. Software stack used on i.MX 95 19 mm x 19 mm EVK

Preparation and deployment of the build on i.MX 95 19 mm x 19 mm EVK

1. Obtain the NXP Linux BSP sources
   The following commands show how to download the i.MX Yocto Project BSP recipe layers. For this example, a directory named `imx-yocto-bsp` is created for the project. You can use any name for the directory.

```
$ mkdir imx-yocto-bsp
$ cp imx-yocto-bsp
# Initialize the Yocto Project from manifest
$ repo init -u https://github.com/nxp-imx/imx-manifest -b imx-linux-scarthgap
 -m imx-6.6.52-2.2.0.xml
```

```
$ repo sync
```

For further details on building an image for a specific release, refer to the *i.MX Yocto Project Users's Guide* document (see Section 4). A document for each Linux release is available on the Embedded Linux for i.MX Applications Processors webpage.

2. Set up the build configuration
   i.MX provides a script, `imx-setup-release.sh`, to simplify the setup for i.MX machines. To use the script, provide the target machine name and the desired graphical backend. The script sets up a directory and the configuration files for the specified machine and backend.

```
$ MACHINE=imx95-19x19-lpddr5-evk DISTRO=fsl-imx-xwayland source ./imx-setup-release.sh
 -b bld-xwayland
```

   To enable PCIe ASPM in the Linux kernel, ensure that the following CONFIGs are enabled in the `linux-imx defconfig`:

```
CONFIG_PCIEASPM=y
CONFIG_PCIEASPM_POWER_SUPERSAVE=y
```

   To start building the kernel (only), run the following command.

```
$ bitbake linux-imx
```

3. Build the image
   To start building the SD card image, run the following commands.

```
$ bitbake -f -c compile linux-imx
$ bitbake -f -c deploy linux-imx
```

4. Deployment on board
   The image is available under the `<build_dir>/tmp/deploy/images/imx95-19x19-lpddr5-evk` directory. Image can be deployed on the board in the following ways:
   a. Using UUU tool
      Connect USB cable on USB1 connector (J29) on board.

```
$ uuu -lsusb
$ uuu -b emmc_all imx-image-core-imx95-19x19-lpddr5-
evk.rootfs-20250319110032.wic
```

   b. Write the image on an SD card for each board:
      The image is available under `<build_dir>/tmp/deploy/images/ imx95-19x19-lpddr5-evk`. It is necessary to write the image on an SD card for each board. The `lsblk` command can be used to find the SD card drive. To write the image, use the below command:

```
$ bzcat imx-image-core-imx95-19x19-lpddr5-evk.rootfs-20250319110032.wic |
 sudo dd of=/dev/sd<drive> bs=1M && sync
```

   c. Replace the kernel image on board
      Replace the image in bootpartition.

```
$ root@imx95evk# cd /run/media/boot-mmcblk0p1
$ root@imx95evk# scp <user>@<ip>:<dir>/linux-imx/linux imx/arch/arm64/
boot/Image run/media/boot-mmcblk0p1/
```

   d. To observe the effects of the new kernel, reboot the i.MX 95 19 mm x 19 mm EVK.

2.3 Methodology

1. Ensure that the AW693 M.2 module is connected to the i.MX 95 19 mm x 19 mm EVK Key E interface.
2. On the EVK, make sure that `imx95-19x19-evk-adv7535-ap1302.dtb` is used:

```
=> print fdt_file
fdt_file= imx95-19x19-evk-adv7535-ap1302.dtb
```

3. Boot the EVK until Linux console and execute the following command:

```
$  echo "powersupersave" > /sys/module/pcie_aspm/parameters/policy
```

4. Writing to this file instructs the kernel to change the ASPM policy to the one with the most aggressive Powersaving. This step completes the software-side configuration. The kernel driver and the PCIe controller hardware state machine handle the rest. That means if both the link partner supports L1 ASPM and there are no pending transactions, the link automatically transitions to L1 substates.

2.4 PCIe ASPM configuration in Linux kernel

Figure 11 and Figure 12 show the L1 substates Entry and Exit flows to explain what happens inside the Linux kernel after enabling ASPM.

L0 to L1 substates entry

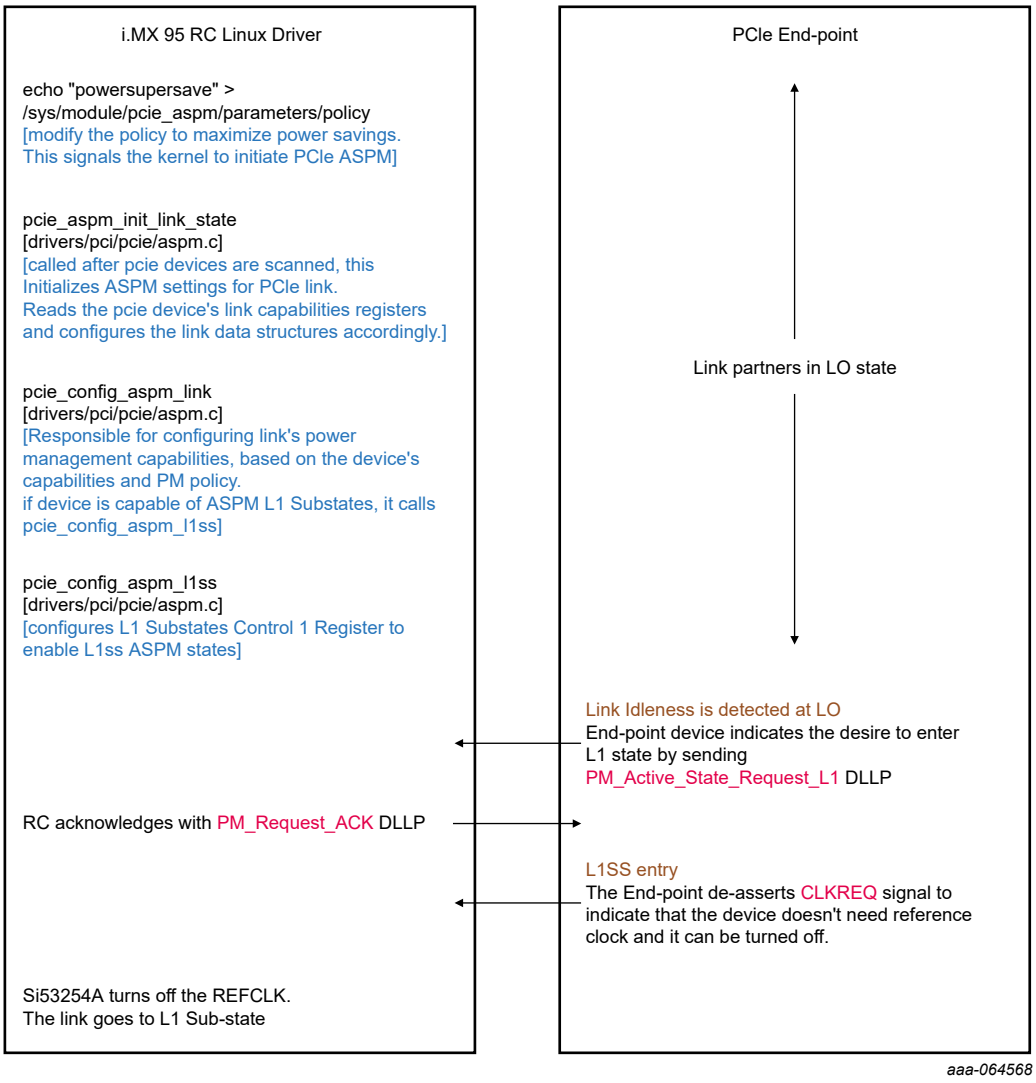


Figure 11. L0 to L1 substates entry flow in Linux kernel

L1 substates to L0 exit

The exit from L1 substates to L0 can be initiated by either Upstream or Downstream port. Figure 12 shows how downstream port initiates the exit.

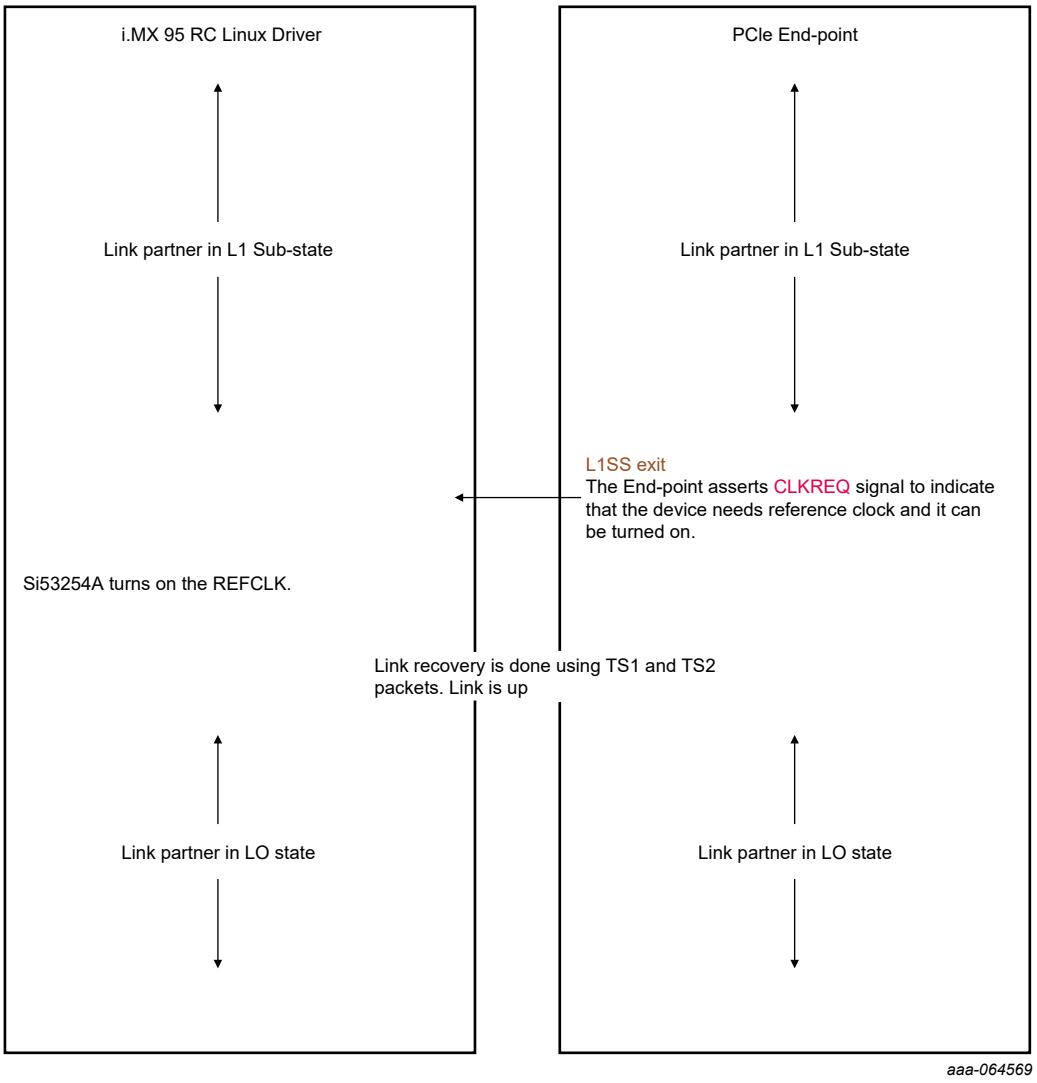
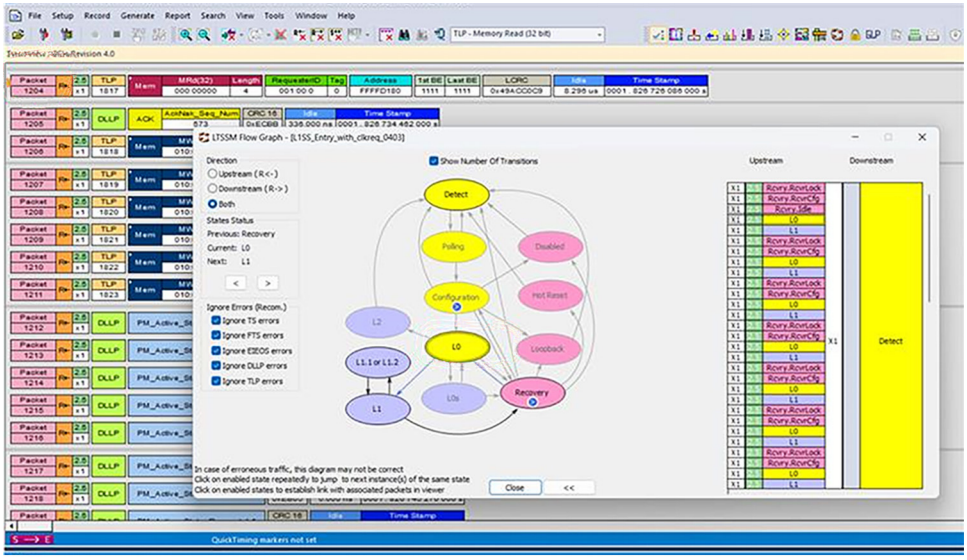


Figure 12. Exit flow from L1 substates to L0 in Linux kernel

2.5 PCIe analyzer traces

This section covers the PCIe analyzer traces that are captured during verification of L1 ASPM. These traces are captured via a Teledyne Lecroy PCIe analyzer. It has proven to be useful for the TLP and DLLP level system troubleshooting/debugging. These traces help to verify if the link indeed goes to L1 substates.

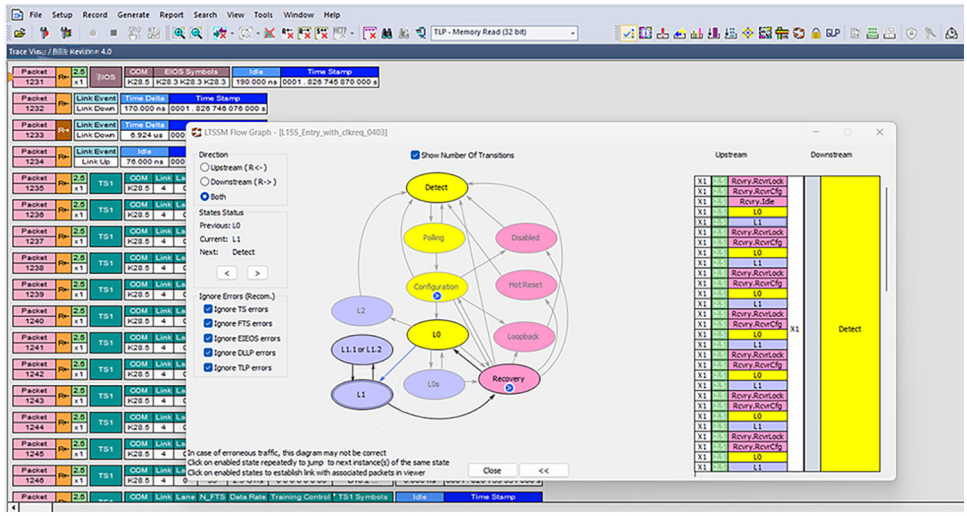
- LTSSM flow depicting L0 state at packet 1204



aaa-064570

Figure 13. LTSSM trace depicting L0 state at packet 1204

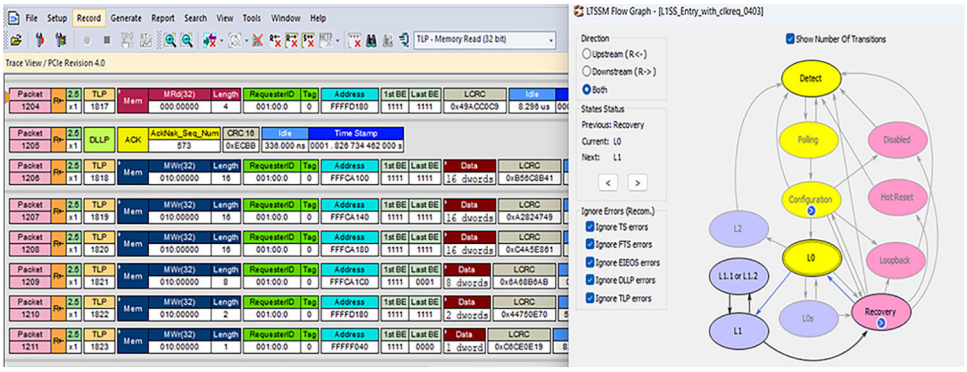
- LTSSM flow depicting L1 state at packet 1231



aaa-064571

Figure 14. LTSSM trace depicting L1 state at packet 1231

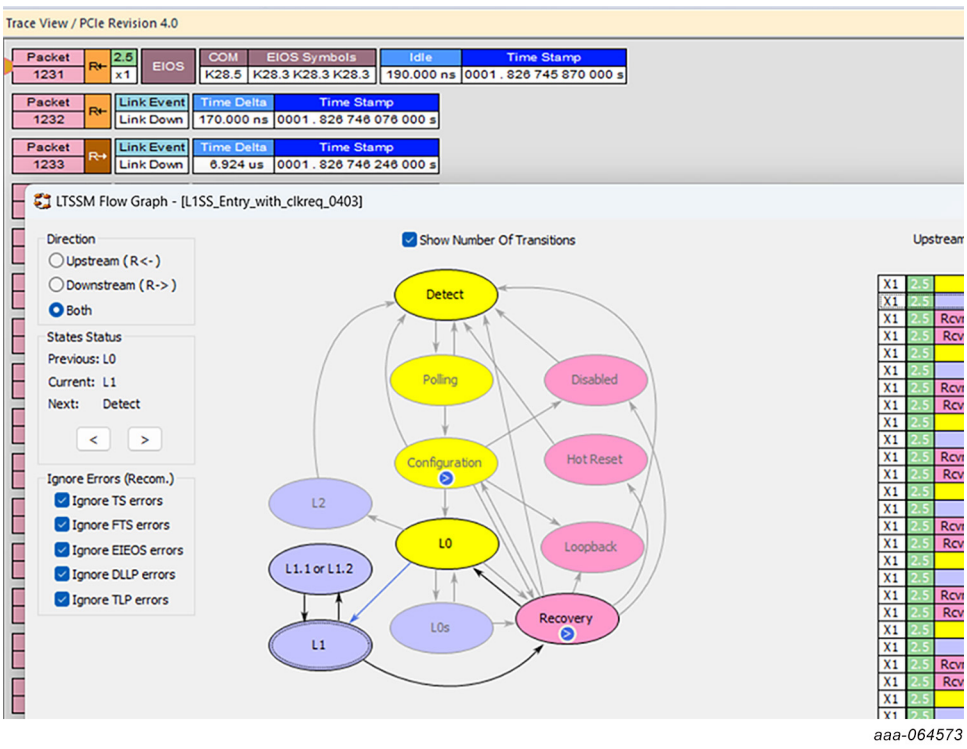
- L0 state depicting memory TLP transactions on the analyzer



aaa-064572

Figure 15. Memory TLP transactions observed in L0 state

- L1 state depicted at packet 1231



aaa-064573

Figure 16. L1 state depicted at packet 1231 on analyzer

- Power management active state request sent from downstream to upstream

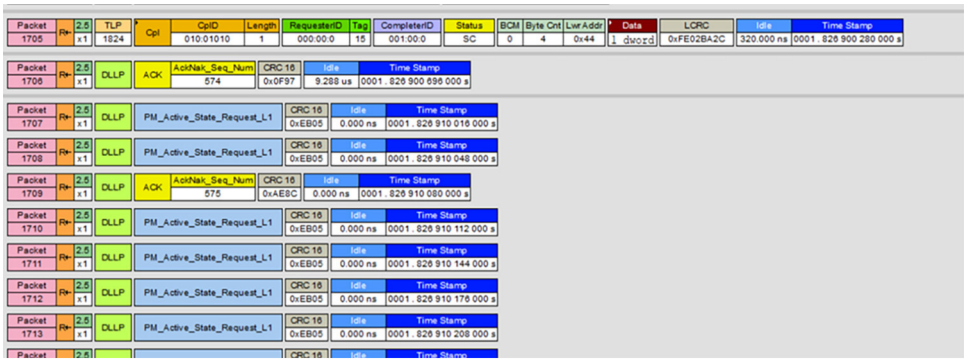


Figure 17. PM active state request sent from downstream to upstream

- CLKREQ# signal deasserted for L1.1/L1.2 at packet 2822

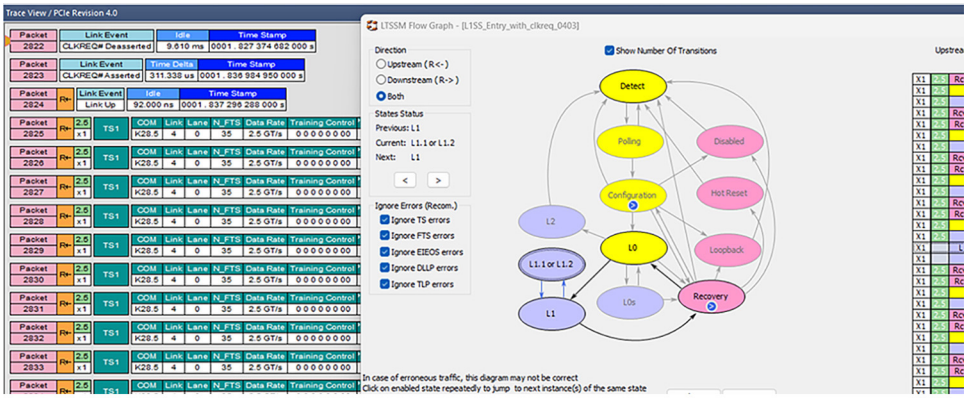


Figure 18. CLKREQ# deasserted for L1.1/L1.2 at packet 2822

3 Acronyms

Table 1 lists the acronyms used in this document.

Table 1. Acronyms

Acronym	Description
ASPM	Active State Power Management
BSP	Board Support Package
DLLP	Data Link Layer Packet
EP	Endpoint
EVK	Evaluation Kit
IOT	Internet of Things
LTSSM	Link Training and Status State Machine
PCIe	Peripheral Component Interconnect Express
PLL	Phase Locked Loop
PM	Power Management

Table 1. Acronyms...*continued*

Acronym	Description
RC	Root Complex
TLP	Transaction Layer Packet
UUU	Universal Update Utility

4 References

Section 4 lists the references and additional information for this release.

Table 2. References

Reference	Link
i.MX Yocto Project User's Guide	UG10164
Embedded Linux for i.MX Applications Processors - Linux Current Quarterly Release and Documentation	IMXLINUX

5 Note about the source code in the document

Example code shown in this document has the following copyright and GPL-2.0 or (at your option) any later version license:

Copyright 2026 NXP

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, see <https://www.gnu.org/licenses/>.

6 Revision history

Table 3 summarizes the revisions to this document.

Table 3. Revision history

Document ID	Release date	Description
AN14913 v.2.0	1 April 2026	Initial public release
AN14913 v.1.0	9 March 2026	Initial NDA release

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.

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 https://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.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

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.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell 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.

Contents