AN14486

Measuring Power Consumption on MCIMX8ULP-EVK9 Board

Rev. 1.0 — 24 December 2024 Application note

Document information

Information	Content
Keywords	AN14486, i.MX 8ULP, MCIMX8ULP-EVK9, i.MX 8ULP power measurement, i.MX 8ULP power domains
Abstract	This application note describes how to measure the current drain of the NXP i.MX 8ULP application processor on the MCIMX8ULP-EVK9 board, through different use cases.

1 Introduction

This application note is intended to help system designers to create power-optimized systems. It describes how to measure the current drain of the NXP i.MX 8ULP application processor on the MCIMX8ULP-EVK9 board, through different use cases. Users can choose the appropriate power supply domains for the i.MX 8ULP processor and become familiar with the expected processor power consumption in these various scenarios.

Note: *The measurements are done on a small sample size; therefore, the results presented in this document are not guaranteed.*

Note: *The difference between this application note and i.MX 8ULP Power Consumption Measurement (document AN13914) is that this one has been tested on the MCIMX8ULP-EVK9. Moreover, the DDR rework is done on this board: LPDDR4X has been used instead of LPDDR4. In such a situation, the voltage of LDO1_CPU_1V1_0V6 becomes 0.6 V and the total power becomes low.*

2 i.MX 8ULP architecture

Figure 1 shows the high-level architecture diagram of the i.MX 8ULP processor.

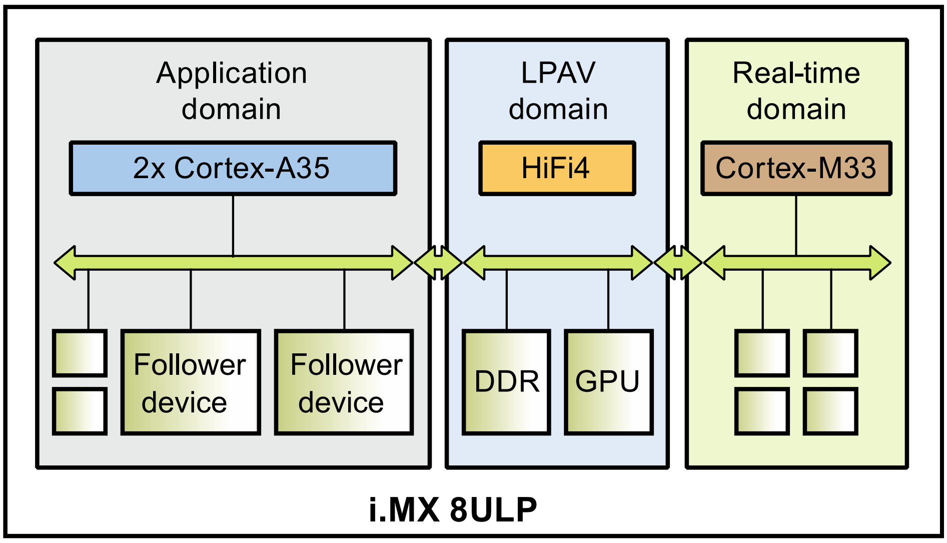


Figure 1. i.MX 8ULP architecture

The i.MX 8ULP architecture contains three domains:

- Real-time processor domain (RTD) or Arm Cortex-M33 domain
- Application processor domain (APD) or Arm Cortex-A35 domain
- Low-power audio/video (LPAV) domain

Note: *For a detailed architecture diagram of the i.MX 8ULP processor, see i.MX 8ULP Processor Reference Manual (IMX8ULPRM).*

The i.MX 8ULP architecture supports both PMIC and non-PMIC use cases; therefore, helping to minimize system cost. When the system is using the RT domain, then the power supply to the AP or LPAV domain can be shut down to save power.

3 i.MX 8ULP power overview

The i.MX 8ULP processor has several power domains, each containing multiple power supplies. Table 1 lists the different i.MX 8ULP power domains and components in each power domain.

Table 1. i.MX 8ULP power domains

Power domain	Components
Real-time processor domain (RTD)	• Arm Cortex-M33 platform • Fusion F1 DSP • Security subsystem • Power management • Multiple peripherals • System-level components • Three GPIO ports: A, B, and C
Application processor domain (APD)	• One or two Arm Cortex-A35 core platform • Multiple peripherals • Three GPIO ports: D, E, and F
Low-power audio/video (LPAV) domain	• HiFi4 DSP • Graphics Processing Unit 3D (GPU3D) • Graphics Processing Unit 2D (GPU2D) • LPDDR3/LPDDR4/LPDDR4x interface • MIPI-DSI interface • MIPI-CSI interface
DGO (always-ON) domain	• Reset and system mode control logic • Low-leakage Wake-Up Unit (WUU) • Analog comparators • Low-power timers
VBAT domain	• Real-time clock (RTC) • Battery-Backed Security Module (BBSM) • Battery-Backed Non-Secure Module (BBNSM)

In general, these domains are independent of each other. Multiple power modes are available in the real-time and application processor domains to optimize power consumption to the demands of the application at a given time. These modes optimize power consumption by reducing clock frequencies, reducing voltages, gating clocks, and gating power supplies.

Figure 2 describes a use case of the i.MX 8ULP base power management scheme with an external PMIC and CM33 LDO bypassed. It also shows the connections of the i.MX 8ULP power supplies and the distribution of the internal power domains.

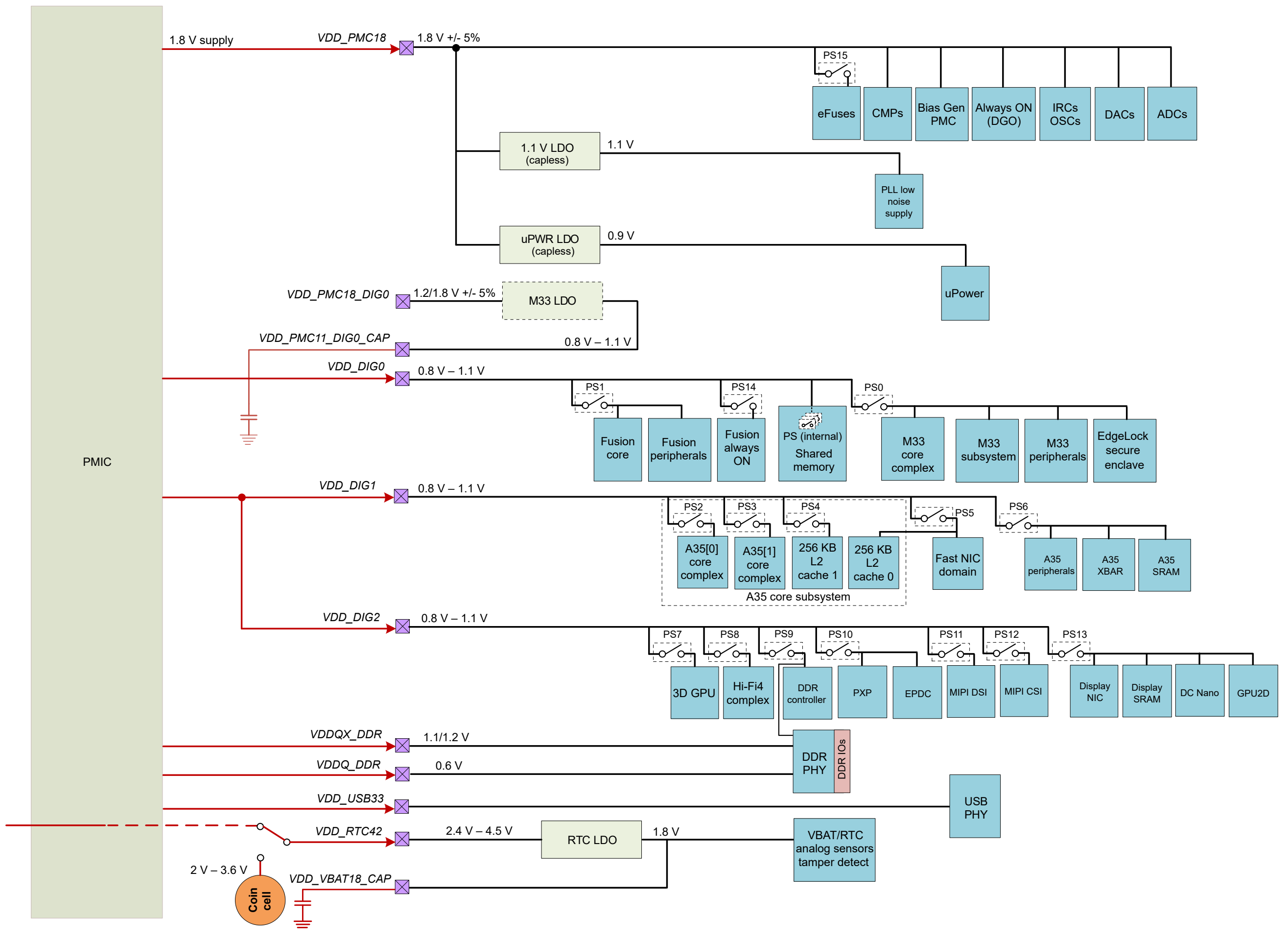


Figure 2. i.MX 8ULP power management scheme (CM33 LDO bypass mode)

Note:

- *For information on the i.MX 8ULP base power management scheme and its other use cases, see i.MX 8ULP Processor Reference Manual (IMX8ULPRM).*
- *For details on recommended operating conditions and groups of pins powered by each I/O voltage supply, see i.MX 8ULP data sheet.*

Figure 3 describes the power management scheme used on the MCIMX8ULP-EVK9 board. In this power management scheme, an external PMIC powers the i.MX 8ULP processor with its M33 LDO set to bypass mode.

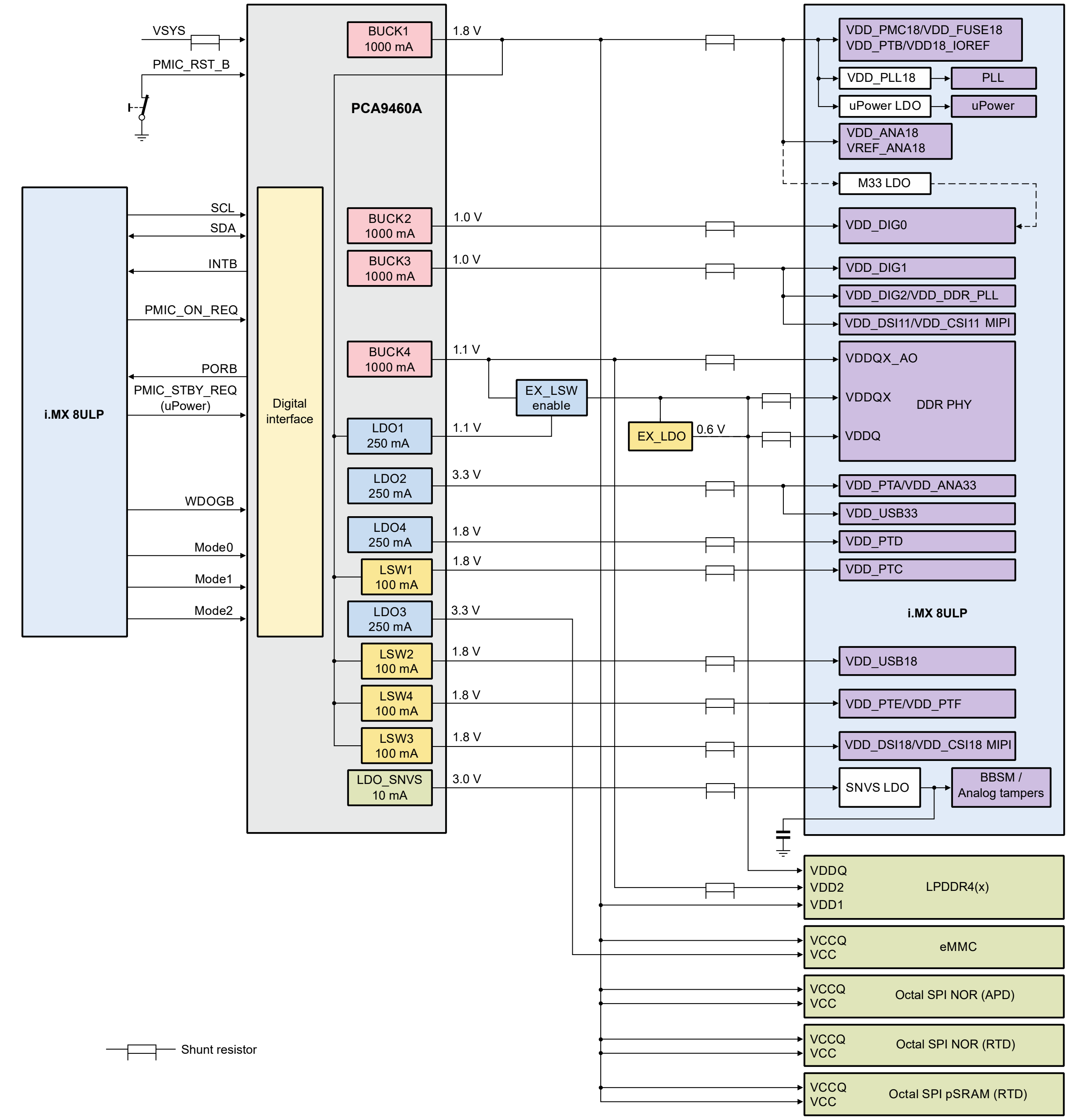


Figure 3. Power distribution on MCIMX8ULP-EVK9

4 i.MX 8ULP power measurement on MCIMX8ULP-EVK9 board

This document provides details of several use cases run by NXP on the NXP MCIMX8ULP-EVK9 board to measure i.MX 8ULP power consumption. These use cases are described under Section 5.

The measurements are taken mainly for the power supply domains given in Table 2. The table also provides a mapping between the power rails in BCU software and power supply domains in the i.MX 8ULP processor. For more information, download "BCU.pdf" from https://github.com/nxp-imx/bcu/releases.

Table 2. Measured power supply domains

Power groups	Power rail in BCU	Power supply domain	Description
GROUP_SOC_FULL	BUCK1_CPU_1V8	VDD_PMC18 + VDD_PLL18 + VDD_FUSE18 + VDD18_IOREF + VDD_PTB + VDD_ANA18 + VREF_ANA18	PMC, PLL, fuse, port B I/O interface power supply and analog power supply
	BUCK1_LSW1_CPU_1V8	VDD_PTC	Port C I/O interface power supply
	BUCK1_LSW2_CPU_1V8	VDD_USB_18	USB power supply
	BUCK1_LSW3_CPU_1V8	VDD_DSI18 + VDD_CSI18	MIPI DSI/CSI I/O power supply
	BUCK1_LSW4_CPU_1V8	VDD_PTE + VDD_PTF	Port E and Port F I/O interface power supply
	BUCK2_CPU_1V0	VDD_DIG0	RTD core power supply
	BUCK3_CPU_1V0	VDD_DIG1 + VDD_DIG2 + VDD_DDR_PLL + VDD_DSI11 + VDD_CSI11	APD core power supply, LPAV domain core supply, DRAM PHY PLL power supply and DSI core supply
	BUCK4_CPU_1V1	VDDQX_AO_DDR	Always-ON DRAM I/O interface power supply
	LDO1_CPU_1V1	VDDQX_DDR	DRAM I/O interface pre-driver power supply
	LDO1_CPU_1V1_0V6	VDDQ_DDR	DRAM I/O interface power supply
	LDO2_CPU_3v3	VDD_PTA + VDD_ANA33 + VDD_USB_33	Port A I/O interface power supply, analog power supply, and USB I/O power supply
	LDO4_CPU_1V8	VDD_PTD	Port D I/O interface power supply
	LDO5_CPU_3V0	VDD_VBAT42	VBAT domain power supply
GROUP_PLATFORM	VSYS_5V0_4V2	VSYS on PMIC	Total power consumption of the SOM board. It includes i.MX 8ULP, DRAM, eMMC, SPI NOR flash, and pSRAM.

Low-power mode measurements in this document apply to multiple power supplies. These measurements primarily contain measurements from the dominant power supply in each active domain:

- For the real-time processor domain (RTD), the dominant supply is `VDD_DIG0`. The dominant supply depends on the internal LDO configurations:
  - In CM33 LDO enable mode, a constant voltage is applied to `VDD_PMC18_DIG0` and the internal LDO provides a lower voltage from `VDD_PMC11_DIG0_CAP` to `VDD_DIG0` under software control. In this case, `LDO_EN` should be tied to `VDD_PMC18`.

  - In CM33 LDO bypass mode, the internal LDO is disabled and an external variable voltage is usually provided by an external Power Management IC (PMIC). In this case, `LDO_EN` should be tied to the ground and each of `VDD_PMC18_DIG0` and `VDD_PMC11_DIG0_CAP` should be connected to the ground through a 10 kΩ resistor. By default, the MCIMX8ULP-EVK9 board supports CM33 LDO bypass mode.
- For the application processor domain (APD), the dominant supply is `VDD_DIG1`.
- For the low-power audio/video (LPAV) domain, the dominant supply is `VDD_DIG2`, which should be tied to `VDD_DIG1` at the board level. `VDD_DSI11` must be tied to `VDD_DIG2` at the board level, even if not used. `VDD_CSI11` should be tied to `VDD_DIG2` at the board level, if used.

The subsections that follow describe various considerations related to i.MX 8ULP power measurement on the MCIMX8ULP-EVK9 board.

4.1 DGO (always-ON) domain power supplies

The DGO domain uses the `VDD_PMC18` power supply, which supplies power to the following circuits:

- CMP0/1
- PMC
- LPTMR
- WUU0/1
- SYSOSC/ LPOSC
- uPower
- Multiple other chip-level functions

4.2 Real-time processor domain (RTD) power supplies

The real-time processor domain uses the following power supplies:

- `VDD_PMC18_DIG0`: Supplies power to CM33 LDO:
  - In the CM33 LDO enable mode, the LDO output `VDD_PMC11_DIG0_CAP` is connected to an external filter capacitor. The LDO output is routed back into `VDD_DIG0` to supply the internal logic. For decoupling and bulk capacitor requirements, see *i.MX 8ULP Hardware Developer's Guide* (IMX8ULPHDG).
  - In the CM33 LDO bypass mode, each of `VDD_PMC18_DIG0` and `VDD_PMC11_DIG0_CAP` should be connected to the ground through a 10 kΩ resistor.
- `VDD_PMC18`: Supplies power to the following circuits:
  - uPower LDO
  - DGO (always-ON) domain logic and the Power Management Controller (PMC)
  - Low noise LDO for PLL
- `VDD_PTA`: Supplies power to the I/Os on port A (signals named PTA*n*).
- `VDD_PTB`: Supplies power to the I/Os on port B (signals named PTB*n*).
- `VDD_PTC`: Supplies power to the I/Os on port C (signals named PTC*n*).

Power consumption of `VDD_PTA`, `VDD_PTB`, and `VDD_PTC` is completely application-dependent and therefore, measurement results related to these supplies in this document are only for reference purposes.

For an equation that can be used to estimate GPIO segment power based on the activity of the individual I/O signals, see the "Absolute maximum ratings" table in i.MX 8ULP data sheet.

4.3 Application processor domain (APD) power supplies

The application processor domain uses the following power supplies:

- `VDD_DIG1`: Supplies power to the CA35 core application processor domain logic.

- VDD_USB33 and VDD_USB18: Supplies power to the USB PHY. VDD_USB33 and VDD_USB18 must be powered even if not used.
- VDD_PTD: Supplies power to the I/Os on port D (signals named PTD*n*).
- VDD_PTE: Supplies power to the I/Os on port E (signals named PTE*n*).
- VDD_PTF: Supplies power to the I/Os on port F (signals named PTF*n*).

For an equation that can be used to estimate GPIO segment power based on the activity of the individual I/O signals, see the "Maximum Supply Currents" table in i.MX 8ULP data sheet.

4.4 Low-power audio/video domain (LPAV) power supplies

The LPAV domain uses the following power supplies:

- VDD_DIG2: Supplies power to the low-power audio/video domain logic and the DRAM controller.
- VDD_DSI18 and VDD_DSI11: Supplies power to the MIPI-DSI interface. VDD_DSI11 must be tied to VDD_DIG2 at the board level, even if not used.
- VDD_CSI18 and VDD_CSI11: Supplies power to the MIPI-CSI interface. VDD_CSI11 should be tied to VDD_DIG2 at the board level, if used.
- VDD_DDR_PLL, VDDQX_DDR, VDDQ_DDR, and VDDQX_AO_DDR: Supplies power to the LPDDR3/LPDDR4/LPDDR4x PHY:
  - VDD_DDR_PLL must always be connected to VDD_DIG2 at the board level.
  - VDDQ_DDR is the 1.2 V / 1.1 V / 0.6 V supply for the LPDDR3/LPDDR4/LPDDR4x interface I/Os.
  - VDDQX_AO_DDR is the 1.2 V / 1.1 V / 1.1 V supply for the LPDDR3/LPDDR4/LPDDR4x interface always-ON I/Os (DDR_CKE/ DDR_RAM_RST_B).

4.5 VBAT domain power supplies

VDD_VBAT42 supplies power to the VBAT domain. In most applications, this supply is provided by a battery. An internal LDO regulates the output to the 1.8 V used by the internal logic on the VBAT domain. VDD_VBAT18_CAP is connected to an external capacitor. For decoupling and bulk capacitor requirements, see *i.MX 8ULP Hardware Developer’s Guide* (IMX8ULPHDG).

4.6 Analog and other supplies

The following supplies are used for analog and chip-level functions:

- VDD_ANA33: It is a 1.8 V or 3.3 V supply for analog functions. The voltage level of this supply should match that of VDD_PTA.
- VDD_ANA18: It is a 1.8 V supply for analog functions. VDD_ANA18 should be tied to VDD_PMC18 at the board level.
- VDD_PLL18: It is a 1.8 V supply for the analog portions of the PLLs.
- VDD_fuse18: It is a 1.8 V supply for the fuse IP. VDD_fuse18 should be tied to VDD_PMC18 at the board level.
- VREFH_ANA18: It is the 1.8 V voltage reference for the high end of the ADC range.
- VDD18_IOREF_1 and VDD18_IOREF_2: Each of these supplies is a 1.8 V reference supply used by the I/Os.

4.7 Voltage levels and VFS usage in measurement process

The voltage level of each supply is set to the typical voltage level as defined in *i.MX 8ULP Hardware Developer’s Guide* (IMX8ULPHDG), unless otherwise specified.

To minimize power consumption in each power mode, voltage and frequency scaling (VFS) can be performed by modifying `VDD_DIG0`, `VDD_DIG1`, and `VDD_DIG2`, while running a use case. For voltage specifications of each power mode, see the "Recommended operating conditions" table in i.MX 8ULP data sheet.

4.8 Temperature

The power measurements described in this document were performed at room temperature (approximately 25 °C), unless otherwise specified.

4.9 Hardware and software requirements

Table 3 provides details of the hardware and software used during the power measurement.

Table 3. Hardware and software used

Category	Description
Hardware	NXPMCIMX8ULP-EVK9, BB: 52007+ SOM: 52006 Bwith A2 silicon, part number is PIMX8UD7VK08SC, with LPDDR4X rework
System software	Linux BSP 6.6.23_2.0.0
Application software	SDK 2.16.0
	BCU 1.1.92. For more details on BCU, see https://github.com/nxp-imx/bcu.

4.10 Build i.MX Yocto Project

To build i.MX Yocto Project, perform these steps:

1. Download and build the i.MX Yocto Project Community BSP recipe layers:
   Note: *For more information on i.MX Yocto Project, see i.MX Yocto Project User's Guide.*

```
repo init -u https://github.com/nxp-imx/imx-manifest -b \
imx-linux-scarthgap -m imx-6.6.23-2.0.0.xml
repo sync
DISTRO=fsl-imx-xwayland MACHINE=imx8ulp-9x9-lpddr4-evk source \
imx-setup-release.sh -b build-imx8ulp-9x9-lpddr4-evk
```

2. For some audio or video cases, `gstreamer1.0-libav` is needed. Put the following commands at the end of the `build-imx8ulp-9x9-lpddr4-evk/conf/local.conf` file:

```
LICENSE_FLAGS_ACCEPTED += "commercial"
IMAGE_INSTALL:append = "gstreamer1.0-libav"
PACKAGECONFIG:append_pn-gstreamer1.0-libav = "x264"
```

3. Build the image:

```
bitbake imx-image-full
```

   The built image can be found in `build-imx8ulp-9x9-lpddr4-evk/tmp/deploy/images`.

4.11 Power consumption measurement steps

The following are the steps needed to measure i.MX 8ULP power consumption on the MCIMX8ULP-EVK9 board:

1. Connect a micro-USB cable between the host PC and the USB micro-B port J17 on the MCIMX8ULP-EVK9 board.

2. Start the monitor in the BCU path:

```
bcu.exe monitor -board=imx8ulpevk9
```

3. Run the related use cases, according to Section 5.
4. After starting the use case, press '3' to reset the value.
5. Optionally, press '4' to change the measurement precision: mA/auto/μA.
6. Wait for 1 minute, and record the measurement data in BCU.

4.12 Software package

Some use case binaries related to i.MX 8ULP power measurement can be found in the software package: AN14486SW.zip. Copy the binaries (Linux app) to the board, then use "chmod +x *" to grant the execution permissions.

5 Use cases and measurement results

The subsections below describe the main use cases and subcases that form the benchmarks for measurement of the i.MX 8ULP internal powers on i.MX 8ULP platforms.

Note:

- *Before running a use case, you must run the required configuration scripts to configure the environment. For more details, see Section 8.*
- *The boot mode for all use cases is Single Boot (eMMC) mode.*

Table 4 summarizes the power measurement results of various use cases performed on the MCIMX8ULP-EVK9 board.

Table 4. MCIMX8ULP-EVK9 power summary report

Use case category	Use case	Total power (sum of average powers)	
		Value	Unit
Low-power mode use cases	Suspend mode – RTD active	54.30	mW
	Suspend mode – RTD sleep	48.34	mW
	Suspend mode – RTD deep sleep	0.70	mW
	Suspend mode – RTD power down	59.43	μW
	System idle with screen ON	360.49	mW
	System idle with screen OFF	153.01	mW
Audio/video playback use cases	Audio playback (gplay) with full frequency	350.30	mW
	Audio low-bus playback (gplay) with DDR clock at 96 MHz	178.58	mW
	Audio playback on Bluetooth (gplay)	381.37	mW
	Video playback local (gplay)	484.66	mW
	Video playback streaming (gplay)	490.73	mW
Core benchmark use cases	Dhrystone	493.69	mW
	CoreMark	457.06	mW
GPU use cases	GLMark	580.39	mW
	vkmark	596.16	mW

Table 4. MCIMX8ULP-EVK9 power summary report*...continued*

Use case category	Use case	Total power (sum of average powers)	
		Value	Unit
Heavy load use cases	2-core Dhrystone + GLMark + HiFi4 + CM33 CoreMark	607.11	mW
	2-core Stream + GLMark + HiFi4 + CM33 CoreMark	612.47	mW
	GPU2D + GPU3D + CSI/DSI	590.64	mW
Memory use cases	memset	517.40	mW
	memcpy	486.63	mW
	Stream	571.35	mW
Storage (eMMC) use cases	DD_RD_eMMC	454.27	mW
	DD_WRT_eMMC	392.33	mW
Product use cases	Machine vision	547.16	mW
	eIQ benchmark	424.43	mW
	UAC audio playback	374.43	mW
	Always-ON display, 1 fps fresh rate	90.21	mW
System-level power estimation use cases	Battery	9.15	µW
	256 KB L2 cache	461.20	mW

5.1 Low-power mode use cases

This section explains the following low-power mode use case scenarios:

- Suspend mode (four subcases)
- System idle with screen ON
- System idle with screen OFF

5.1.1 Suspend mode – RTD active

The following are the required settings for this use case:

- The Linux kernel is in Suspend mode.
- APD is in Power-Down mode.
- LPAV is in Power-Down mode with DDR in Self-Refresh mode.
- RTD is in Active mode, running the Power Mode Switch (PMS) demo.
- All clocks and PLLs in APD and LPAV are turned off.

To configure and run the use case, perform these steps:

1. Power on the MCIMX8ULP-EVK9 board by turning on the ON/OFF switch (SW10).
2. Ensure that:
   - On the RTD console, the PMS demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
3. Put the Linux kernel into the Suspend (Power-Down) mode:

```
echo mem > /sys/power/state
```

4. From the menu on the RTD console, press 'A' to put RTD into Active mode (RTD is in Active mode by default).
5. Measure the power and record the results.

Table 5 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 5. Power measurement results for "Suspend mode – RTD active" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	8.74	15.75	54.30	Die temperature was not measured as the CA35 core was suspended.
	buck1_lsw1_cpu_1v8	1.81	0.06	0.11		
	buck1_lsw2_cpu_1v8	1.80	0.01	0.02		
	buck1_lsw3_cpu_1v8	1.80	0.00	0.00		
	buck1_lsw4_cpu_1v8	1.81	0.02	0.04		
	buck2_cpu_1v0	1.00	33.76	33.76		
	buck3_cpu_1v0	0.73	0.00	0.00		
	buck4_cpu_1v1	1.10	0.00	0.00		
	ldo1_cpu_1v1	0.00	0.00	0.00		
	ldo1_cpu_1v1_0v6	0.00	-0.01	0.00		
	ldo2_cpu_3v3	3.30	1.38	4.57		
	ldo4_cpu_1v8	1.81	0.01	0.01		
	ldo5_cpu_3v0	2.92	0.01	0.03		

5.1.2 Suspend mode – RTD sleep

The following are the required settings for this use case:

- The Linux kernel is in Suspend mode.
- APD is in Power-Down mode.
- LPAV is in Power-Down mode with DDR in Self-Refresh mode.
- RTD is in Sleep mode.
- All clocks and PLLs in APD and LPAV are turned off.

To configure and run the use case, perform these steps:

1. Power on the MCIMX8ULP-EVK9 board by turning on the ON/OFF switch (SW10).
2. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
3. Put the Linux kernel into the Suspend (Power-Down) mode:

```
echo mem > /sys/power/state
```

4. From the menu on the RTD console, first press 'D', and then press 'S' to put RTD into Sleep mode.
5. Measure the power and record the results.

Table 6 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 6. Power measurement results for "Suspend mode – RTD sleep" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	BUCK1_CPU_1V8	1.8	8.7	15.68	48.34	Die temperature was not measured as the CA35 core was suspended.
	BUCK1_LSW1_CPU_1V8	1.81	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.8	0.01	0.02		
	BUCK1_LSW3_CPU_1V8	1.8	0	0		
	BUCK1_LSW4_CPU_1V8	1.81	0	0		
	BUCK2_CPU_1V0	1	27.89	27.9		
	BUCK3_CPU_1V0	0.73	0.05	0.03		
	BUCK4_CPU_1V1	1.1	0	0		
	LDO1_CPU_1V1	0	0	0		
	LDO1_CPU_1V1_0V6	0	-0.03	0		
	LDO2_CPU_3V3	3.31	1.38	4.57		
	LDO4_CPU_1V8	1.81	0.01	0.01		
	LDO5_CPU_3V0	2.92	0.01	0.03		

5.1.3 Suspend mode – RTD deep sleep

The following are the required settings for this use case:

- The Linux kernel is in Suspend mode.
- APD is in Power-Down mode.
- LPAV is in Power-Down mode with DDR in Self-Refresh mode.
- RTD is in Deep-Sleep mode.
- All clocks and PLLs in APD and LPAV are turned off.

To configure and run the use case, perform these steps:

1. Power on the MCIMX8ULP-EVK9 board by turning on the ON/OFF switch (SW10).
2. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).

3. Put the Linux kernel into the Suspend (Power-Down) mode:

   ```
   echo mem > /sys/power/state
   ```

4. From the menu on the RTD console, first press 'E', and then press 'S' to put RTD into Deep-Sleep mode.
5. Measure the power and record the results.

Table 7 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 7. Power measurement results for "Suspend mode – RTD deep sleep" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	BUCK1_CPU_1V8	1.80	0.00	0.0	0.70	Die temperature was not measured as the CA35 core was suspended.
	BUCK1_LSW1_CPU_1V8	1.81	0.01	0.01		
	BUCK1_LSW2_CPU_1V8	1.81	0.01	0.03		
	BUCK1_LSW3_CPU_1V8	1.81	0.00	0.0		
	BUCK1_LSW4_CPU_1V8	1.81	0.00	0.01		
	BUCK2_CPU_1V0	0.75	0.91	0.68		
	BUCK3_CPU_1V0	0.73	0.00	0.0		
	BUCK4_CPU_1V1	1.10	0.00	0.0		
	LDO1_CPU_1V1	0.00	0.00	0.0		
	LDO1_CPU_1V1_0V6	0.00	0.00	0.0		
	LDO2_CPU_3V3	3.31	0.00	0.01		
	LDO4_CPU_1V8	1.81	0.01	0.01		
	LDO5_CPU_3V0	2.94	0.00	0.01		

5.1.4 Suspend mode – RTD power down

The following are the required settings for this use case:

- The Linux kernel is in Suspend mode.
- APD is in Power-Down mode.
- LPAV is in Power-Down mode with DDR in Self-Refresh mode.
- RTD is in Power-Down mode (use SW8 push button (wake-up source) on the MCIMX8ULP-EVK9 board to power down CM33 cores).
- All clocks and PLLs in APD and LPAV are turned off.

To configure and run the use case, perform these steps:

1. Power on the MCIMX8ULP-EVK9 board by turning on the ON/OFF switch (SW10).
2. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.

- On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).

3. Put the Linux kernel into the Suspend (Power-Down) mode:

```
echo mem > /sys/power/state
```

4. From the menu on the RTD console, first press 'F', and then press 'S' to put RTD into Power-Down mode.
5. Measure the power and record the results.

Table 8 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 8. Power measurement results for "Suspend mode – RTD power down" use case

Power group	**Power rail**	**Average voltage (V)**	**Average current (mA)**	**Average power (mW)**	**Sum of average powers (µW)**	**Zone 0 die temperature** (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	0.04	0.06	59.43	Die temperature was not measured as the CA35 core was suspended.
	buck1_lsw1_cpu_1v8	0.24	0.00	0.00		
	buck1_lsw2_cpu_1v8	1.80	0.01	0.02		
	buck1_lsw3_cpu_1v8	1.81	0.00	0.00		
	buck1_lsw4_cpu_1v8	1.81	0.01	0.01		
	buck2_cpu_1v0	0.72	-0.03	-0.02		
	buck3_cpu_1v0	0.73	-0.05	-0.04		
	buck4_cpu_1v1	1.10	0.00	0.00		
	ldo1_cpu_1v1	0.00	0.00	0.00		
	ldo1_cpu_1v1_0v6	0.00	-0.02	0.00		
	ldo2_cpu_3v3	3.31	0.00	0.00		
	ldo4_cpu_1v8	1.81	0.01	0.01		
	ldo5_cpu_3v0	2.95	0.00	0.01		

Note: *The unit of power is µW for these use cases:*

- *Suspend mode – RTD power down*
- *Battery*

5.1.5 System idle with screen ON

The following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value).
- The DDR frequency is set to 528 MHz (1056 MT/s).
- The Arm Cortex-M33 core (RTD) is in Sleep mode.
- All the unused PLLs are OFF, and the unused clocks are gated.
- A 720p TV display is connected to the MCIMX8ULP-EVK9 board through an HDMI interface.

To configure and run the use case, perform these steps:

1. Power on the MCIMX8ULP-EVK9 board by turning on the ON/OFF switch (SW10).
2. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
3. To put the Linux kernel into Idle mode, run `setup_default.sh` (see Section 8 for more details).
4. From the menu on the RTD console, first press 'D', and then press 'S' to put RTD into Sleep mode.

5. Measure the power and record the results.

Table 9 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 9. Power measurement results for "System idle with screen ON" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	16.54	29.80	360.49	32
	buck1_lsw1_cpu_1v8	1.80	0.06	0.11		
	buck1_lsw2_cpu_1v8	1.80	4.54	8.16		
	buck1_lsw3_cpu_1v8	1.80	1.78	3.20		
	buck1_lsw4_cpu_1v8	1.80	0.68	1.23		
	buck2_cpu_1v0	1.00	28.49	28.50		
	buck3_cpu_1v0	1.09	237.82	259.09		
	buck4_cpu_1v1	1.10	0.47	0.52		
	ldo1_cpu_1v1	1.10	3.59	3.94		
	ldo1_cpu_1v1_0v6	0.60	34.72	20.70		
	ldo2_cpu_3v3	3.30	1.39	4.58		
	ldo4_cpu_1v8	1.80	0.37	0.66		
	ldo5_cpu_3v0	2.92	0.01	0.03		

5.1.6 System idle with screen OFF

The following are the required settings for this use case:

- The CPU frequency is set to 650 MHz.
- The DDR frequency is set to 96 MHz (low frequency).
- If the Linux kernel is in the lowest level of idle, both CA35 cores are power gated.
- The CM33 core (RTD) is in Sleep mode.
- All the unused PLLs are OFF, and the unused clocks are gated.
- A 720p TV display is connected to the MCIMX8ULP-EVK9 board through the HDMI interface.

To configure and run the use case, perform these steps:

1. Power on the MCIMX8ULP-EVK9 board by turning on the ON/OFF switch (SW10).
2. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the DTB configuration `imx8ulp-evk-nd.dtb`.
3. To put the Linux kernel into Idle mode, run `DDRC_96MHz_setup.sh` (see Section 8).
4. From the menu on the RTD console, first press 'D', and then press 'S' to put RTD into Sleep mode.
5. Measure the power and record the results.

Table 10 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 10. Power measurement results for "System idle with screen OFF" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	13.01	23.44	153.01	30
	buck1_lsw1_cpu_1v8	1.80	0.03	0.05		
	buck1_lsw2_cpu_1v8	1.80	4.54	8.16		
	buck1_lsw3_cpu_1v8	1.80	-0.01	-0.02		
	buck1_lsw4_cpu_1v8	1.80	0.87	1.57		
	buck2_cpu_1v0	1.00	28.38	28.39		
	buck3_cpu_1v0	0.99	77.63	77.09		
	buck4_cpu_1v1	1.10	0.09	0.10		
	ldo1_cpu_1v1	1.10	0.64	0.70		
	ldo1_cpu_1v1_0v6	0.60	13.87	8.29		
	ldo2_cpu_3v3	3.30	1.39	4.58		
	ldo4_cpu_1v8	1.80	0.36	0.64		
	ldo5_cpu_3v0	2.92	0.01	0.03		

5.2 Audio/video playback use cases

This section explains the following use case scenarios related to audio/video playback:

- Audio playback (gplay) with full frequency
- Audio low-bus playback (gplay) with DDR clock at 96 MHz
- Audio playback on Bluetooth (gplay)
- Video playback local (gplay)
- Video playback streaming (gplay)

5.2.1 Audio playback (gplay) with full frequency

For this use case, you need an audio file with the following specifications:

- File type: MP3
- Bit rate: 128 kbit/s
- Sample rate: 44 kHz per second

The audio file is played using the following command:

```
gplay-1.0 $audio_file
```

The following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value).

- The DDR frequency is set to 528 MHz (1056 MT/s).
- The CA35 cores are in Active mode, and they handle MP3 audio decoding.
- The CM33 core is in Active mode, and it handles audio PCM playback through the I2S interface.
- All the unused PLLs are OFF, and the unused clocks are gated.

To configure and run the use case, perform these steps:

Note: *Run the use case in a loop and log power measurement results at the desired time interval (the recommended interval is 1 minute).*

1. Power on the MCIMX8ULP-EVK9 board by turning on the ON/OFF switch (SW10).
2. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
3. Run `setup.sh` (see Section 8 for more details).
4. Run `gplay_audio.sh`:

   ```
   audio_file='Mpeg1L3_44kHz_128kbps_s_Ed_Rush_Sabotage_mplayer.mp3'
   gplay-1.0 $audio_file
   ```

   Note: *To run the above command, prepare your own MP3 file. To achieve results similar to the results provided in Table 11, the audio bit rate should be approximately 128 kbit/s.*
5. Measure the power and record the results.

Table 11 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 11. Power measurement results for "Audio playback (gplay) with full frequency" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	16.41	29.56	350.30	32
	buck1_lsw1_cpu_1v8	1.80	0.04	0.07		
	buck1_lsw2_cpu_1v8	1.80	4.55	8.18		
	buck1_lsw3_cpu_1v8	1.80	0.01	0.01		
	buck1_lsw4_cpu_1v8	1.80	0.87	1.57		
	buck2_cpu_1v0	1.00	28.43	28.44		
	buck3_cpu_1v0	1.09	239.63	261.10		
	buck4_cpu_1v1	1.10	0.49	0.53		
	ldo1_cpu_1v1	1.10	3.52	3.86		
	ldo1_cpu_1v1_0v6	0.60	19.59	11.70		
	ldo2_cpu_3v3	3.30	1.38	4.57		
	ldo4_cpu_1v8	1.80	0.37	0.66		
	ldo5_cpu_3v0	2.92	0.01	0.03		

5.2.2 Audio low-bus playback (gplay) with DDR clock at 96 MHz

For this use case, you need an audio file with the following specifications:

- File type: MP3
- Bit rate: 128 kbit/s
- Sample rate: 44 kHz per second

The audio file is played using the following command:

```
gplay-1.0 $audio_file
```

The following are the required settings for this use case:

- The CA35 core frequency is set to 650 MHz.
- The DDR frequency is set to 96 MHz.
- If the Linux kernel is in the lowest level of idle, the CA35 cores are power gated. These cores handle MP3 audio decoding.
- The CM33 core is in Active mode, and it handles audio PCM playback through the I2S interface.
- All the unused PLLs are OFF, and the unused clocks are gated.

To configure and run the use case, perform these steps:

1. Power on the MCIMX8ULP-EVK9 board by turning on the ON/OFF switch (SW10).
2. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the DTB configuration `imx8ulp-evk-nd.dtb`.
     Note: *The `imx8ulp-evk-nd.dtb` configuration puts GPU3D/2D, HiFi4, and uSDHC0/1/2 on low power for Nominal Drive (ND) mode. Therefore, the system is set to ND mode, and BUCK3 operates at a lower voltage (1.0 V) for improved power consumption.*
3. Run `DDRC_96MHz_setup.sh` (see Section 8 for more details).
4. Run `gplay_audio.sh`

   ```
   audio_file='Mpeg1L3_44kHz_128kbps_s_Ed_Rush_Sabotage_mplayer.mp3'
   gplay-1.0 $audio_file
   ```

   Note: *To run the above command, prepare your own MP3 file. To achieve results similar to the results provided in Table 12, the audio bit rate should be approximately 128 kbit/s.*
5. Measure the power and record the results.

Table 12 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 12. Power measurement results for "Audio low-bus playback (gplay) with DDR clock at 96 MHz" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	13.31	23.99	178.58	29
	buck1_lsw1_cpu_1v8	1.80	0.04	0.08		
	buck1_lsw2_cpu_1v8	1.80	4.55	8.17		
	buck1_lsw3_cpu_1v8	1.80	0.00	0.01		

Table 12. Power measurement results for "Audio low-bus playback (gplay) with DDR clock at 96 MHz" use case*...continued*

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
	buck1_lsw4_cpu_1v8	1.80	1.96	3.54		
	buck2_cpu_1v0	1.00	34.69	34.69		
	buck3_cpu_1v0	0.99	92.88	92.20		
	buck4_cpu_1v1	1.10	0.11	0.12		
	ldo1_cpu_1v1	1.10	0.68	0.75		
	ldo1_cpu_1v1_0v6	0.60	16.34	9.76		
	ldo2_cpu_3v3	3.30	1.39	4.58		
	ldo4_cpu_1v8	1.80	0.38	0.68		
	ldo5_cpu_3v0	2.92	0.01	0.03		

5.2.3 Audio playback on Bluetooth (gplay)

For this use case, you need an audio file with the following specifications:

- File type: MP3
- Bit rate: 128 kbit/s
- Sample rate: 44 kHz per second

The audio file is played using the following command:

```
gplay-1.0 $audio_file
```

The following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value).
- The CM33 core is in Active mode, and it handles audio PCM playback through the I2S interface.
- The HiFi4 DSP is in Active mode, and it handles MP3 audio decoding.
- A Bluetooth speaker is used to play the audio stream.
- All the unused PLLs are OFF, and the unused clocks are gated.

To configure and run the use case, perform these steps:

Note: *Run the use case in a loop and log power measurement results at the desired time interval (the recommended interval is 1 minute).*

1. Power on the MCIMX8ULP-EVK9 board by turning on the ON/OFF switch (SW10).
2. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
3. From the menu on the RTD console, press 'A' to put RTD into Active mode in the PMS demo.
4. Run `setup.sh` (see Section 8 for more details).
5. Prepare a Bluetooth speaker.

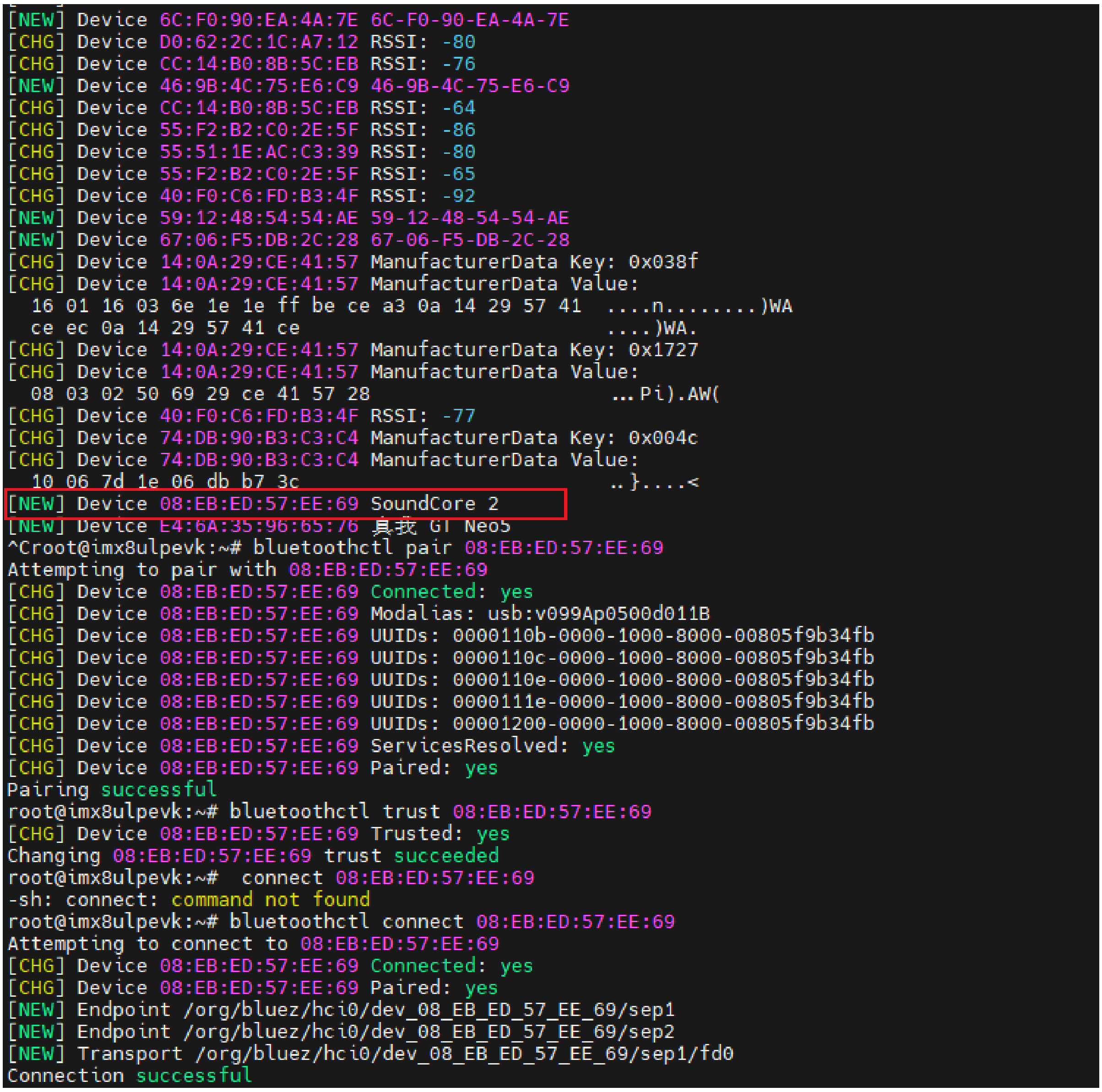
6. Run the following command to enable the audio service:

```
systemctl --user --now enable pipewire wireplumber pipewire-pulse
```

7. To search the speaker MAC address, run the following commands:

```
modprobe moal mod_para=nxp/wifi_mod_para.conf
modprobe btnxpuart
hciconfig hci0 up
bluetoothctl
[bluetooth]# default-agent
[bluetooth]# agent on
[bluetooth]# scan on[1]
[bluetooth]# pair xx:xx:xx:xx:xx:xx
[BT dev]# connect xx:xx:xx:xx:xx:xx
[BT dev]# quit
```

[1] Copy the MAC address of the Bluetooth speaker (using the key combination CTRL+C) when it appears on the terminal. In the current example, the Bluetooth speaker is named as *SoundCore 2*, and its MAC address is 08:EB:ED:57:EE:69 (see Figure 4).



```
[NEW] Device 6C:F0:90:EA:4A:7E 6C-F0-90-EA-4A-7E
[CHG] Device D0:62:2C:1C:A7:12 RSSI: -80
[CHG] Device CC:14:B0:8B:5C:EB RSSI: -76
[NEW] Device 46:9B:4C:75:E6:C9 46-9B-4C-75-E6-C9
[CHG] Device CC:14:B0:8B:5C:EB RSSI: -64
[CHG] Device 55:F2:B2:C0:2E:5F RSSI: -86
[CHG] Device 55:51:1E:AC:C3:39 RSSI: -80
[CHG] Device 55:F2:B2:C0:2E:5F RSSI: -65
[CHG] Device 40:F0:C6:FD:B3:4F RSSI: -92
[NEW] Device 59:12:48:54:54:AE 59-12-48-54-54-AE
[NEW] Device 67:06:F5:DB:2C:28 67-06-F5-DB-2C-28
[CHG] Device 14:0A:29:CE:41:57 ManufacturerData Key: 0x038f
[CHG] Device 14:0A:29:CE:41:57 ManufacturerData Value:
  16 01 16 03 6e 1e 1e ff be ce a3 0a 14 29 57 41  ....n.........)WA
  ce ec 0a 14 29 57 41 ce                          ....)WA.
[CHG] Device 14:0A:29:CE:41:57 ManufacturerData Key: 0x1727
[CHG] Device 14:0A:29:CE:41:57 ManufacturerData Value:
  08 03 02 50 69 29 ce 41 57 28                    ...Pi).AW(
[CHG] Device 40:F0:C6:FD:B3:4F RSSI: -77
[CHG] Device 74:DB:90:B3:C3:C4 ManufacturerData Key: 0x004c
[CHG] Device 74:DB:90:B3:C3:C4 ManufacturerData Value:
  10 06 7d 1e 06 db b7 3c                          ..}....<
[NEW] Device 08:EB:ED:57:EE:69 SoundCore 2
[NEW] Device E4:6A:35:96:65:76 真我 GT Neo5
^Croot@imx8ulpevk:~# bluetoothctl pair 08:EB:ED:57:EE:69
Attempting to pair with 08:EB:ED:57:EE:69
[CHG] Device 08:EB:ED:57:EE:69 Connected: yes
[CHG] Device 08:EB:ED:57:EE:69 Modalias: usb:v099Ap0500d011B
[CHG] Device 08:EB:ED:57:EE:69 UUIDs: 0000110b-0000-1000-8000-00805f9b34fb
[CHG] Device 08:EB:ED:57:EE:69 UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device 08:EB:ED:57:EE:69 UUIDs: 0000110e-0000-1000-8000-00805f9b34fb
[CHG] Device 08:EB:ED:57:EE:69 UUIDs: 0000111e-0000-1000-8000-00805f9b34fb
[CHG] Device 08:EB:ED:57:EE:69 UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device 08:EB:ED:57:EE:69 ServicesResolved: yes
[CHG] Device 08:EB:ED:57:EE:69 Paired: yes
Pairing successful
root@imx8ulpevk:~# bluetoothctl trust 08:EB:ED:57:EE:69
[CHG] Device 08:EB:ED:57:EE:69 Trusted: yes
Changing 08:EB:ED:57:EE:69 trust succeeded
root@imx8ulpevk:~#  connect 08:EB:ED:57:EE:69
-sh: connect: command not found
root@imx8ulpevk:~# bluetoothctl connect 08:EB:ED:57:EE:69
Attempting to connect to 08:EB:ED:57:EE:69
[CHG] Device 08:EB:ED:57:EE:69 Connected: yes
[CHG] Device 08:EB:ED:57:EE:69 Paired: yes
[NEW] Endpoint /org/bluez/hci0/dev_08_EB_ED_57_EE_69/sep1
[NEW] Endpoint /org/bluez/hci0/dev_08_EB_ED_57_EE_69/sep2
[NEW] Transport /org/bluez/hci0/dev_08_EB_ED_57_EE_69/sep1/fd0
Connection successful
```

Figure 4. Copy Bluetooth MAC address

8. Run `gplay_audio.sh`:

```
audio_file='Mpeg1L3_44kHz_128kbps_s_Ed_Rush_Sabotage_mplayer.mp3'
gplay-1.0 $audio_file
```

9. Measure the power and record the results.

Table 13 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 13. Power measurement results for "Audio playback on Bluetooth (gplay)" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	16.47	29.68	381.37	32
	buck1_lsw1_cpu_1v8	1.80	0.09	0.16		
	buck1_lsw2_cpu_1v8	1.80	4.54	8.16		
	buck1_lsw3_cpu_1v8	1.80	0.00	0.00		
	buck1_lsw4_cpu_1v8	1.80	7.54	13.54		
	buck2_cpu_1v0	1.00	35.30	35.30		
	buck3_cpu_1v0	1.09	249.25	271.52		
	buck4_cpu_1v1	1.10	0.48	0.53		
	ldo1_cpu_1v1	1.10	3.56	3.90		
	ldo1_cpu_1v1_0v6	0.60	22.30	13.31		
	ldo2_cpu_3v3	3.30	1.39	4.58		
	ldo4_cpu_1v8	1.80	0.37	0.66		
	ldo5_cpu_3v0	2.92	0.01	0.03		

5.2.4 Video playback local (gplay)

For this use case, you need a video file with the following specifications:

- File type: MKV
- Video compression standard: HEVC
- Display resolution: Full HD resolution at 29.97 fps
- Audio encoding: AACL in a 2-channel configuration
- Audio sample rate: 44.1 kHz per second

The video file is played locally using the following command:

Note: *The i.MX 8ULP processor has no hardware decoder; therefore, a software decoder is used with it.*

```
gplay-1.0 $path/$FILE
```

The following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value).
- The CA35 cores are in Active mode, and they handle audio decoding and video decoding.
- The CM33 core is in Active mode, and it handles audio PCM playback through the I2S interface.
- All the unused PLLs are OFF, and the unused clocks are gated.
- A 720p TV display is connected to the MCIMX8ULP-EVK9 board through the HDMI interface.

To configure and run the use case, perform these steps:

Note: *Run the use case in a loop and log power measurement results at the desired time interval (the recommended interval is 1 minute).*

1. Connect the display to the MCIMX8ULP-EVK9 board through the HDMI interface.
2. Power on the board by turning on the ON/OFF switch (SW10).
3. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
4. Run `setup_video.sh` (see Section 8 for more details).
5. From the menu on the RTD console, press 'A' to put RTD into Active mode in the PMS demo.
6. Run `gplay_videoplayback.sh`:

   ```
   gplay-1.0 ./480p24.mp4
   ```

   Note: *To run the above command, prepare your own MP4 file. To achieve results similar to the results provided in Table 14, use the following settings:*
   - *Resolution: 480p with 24 frame rate*
   - *Video bit rate: Approximately 1200 kbit/s*
   - *Encoding: H.264 format*
7. Measure the power and record the results.

Table 14 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 14. Power measurement results for "Video playback local (gplay)" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	16.95	30.54	484.66	35
	buck1_lsw1_cpu_1v8	1.80	0.06	0.10		
	buck1_lsw2_cpu_1v8	1.80	4.56	8.21		
	buck1_lsw3_cpu_1v8	1.80	1.76	3.18		
	buck1_lsw4_cpu_1v8	1.80	0.89	1.60		
	buck2_cpu_1v0	1.00	34.35	34.36		
	buck3_cpu_1v0	1.09	341.39	371.33		
	buck4_cpu_1v1	1.10	0.48	0.53		
	ldo1_cpu_1v1	1.09	4.02	4.40		
	ldo1_cpu_1v1_0v6	0.60	42.22	25.15		
	ldo2_cpu_3v3	3.30	1.39	4.58		
	ldo4_cpu_1v8	1.80	0.36	0.66		
	ldo5_cpu_3v0	2.93	0.01	0.03		

5.2.5 Video playback streaming (gplay)

For this use case, you need a video file with the following specifications:

- File type: MKV
- Video compression standard: HEVC
- Display resolution: Full HD resolution at 29.97 fps
- Audio encoding: AACL in a 2-channel configuration
- Audio sample rate: 44.1 kHz per second

The video streaming is done with an Ethernet adapter using the following command:

```
gplay-1.0 $network_path/$FILE
```

The following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value).
- The CA35 cores are in Active mode, and they handle audio decoding and video decoding.
- The CM33 core is in Active mode, and it handles audio PCM playback through the I2S interface.
- All the unused PLLs are OFF, and the unused clocks are gated.
- A server is set up to host the MKV video file for streaming.
- A 720p TV display is connected to the MCIMX8ULP-EVK9 board through the HDMI interface.

To configure and run the use case, perform these steps:

Note: *Run the use case in a loop and log power measurement results at the desired time interval (the recommended interval is 1 minute).*

1. Connect your PC and the MCIMX8ULP-EVK9 board to the same local network.
2. On the server PC:
   a. Download `node.js` from https://nodejs.org.
   b. Install http-server using the following command:
   ```
   npm install http-server -g
   ```
   c. Enter the target folder containing the target video (`480p24.mp4`) in the terminal.
   d. Run the following command:
   ```
   http-server -c-1
   ```
   Now, you can use the <ip_server>.
3. On the board:
   a. Power on the board by turning on the ON/OFF switch (SW10).
   b. Ensure that:
      - On the RTD console, the Power Mode Switch (PMS) demo is booted.
      - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
   c. Connect the HDMI display and run `setup_video_stream.sh`.
   d. From the menu on the RTD console, press 'A' to put RTD into Active mode in the PMS demo.
   e. Run the following command on the board:
   ```
   gplay-1.0 http://<ip_server>/480p24.mp4
   ```
   Note: *To run the above command, prepare your own MP4 file. To achieve results similar to the results provided in Table 15, use the following settings:*
   - *Resolution: 480p with 24 frame rate*

- *Video bit rate: Approximately 1200 kbit/s*
- *Encoding: H.264 format*

f. Measure the power of the board and record the results.

Table 15 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 15. Power measurement results for "Video playback streaming (gplay)" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	17.12	30.85	490.73	36
	buck1_lsw1_cpu_1v8	1.80	0.06	0.11		
	buck1_lsw2_cpu_1v8	1.80	4.57	8.22		
	buck1_lsw3_cpu_1v8	1.80	1.77	3.19		
	buck1_lsw4_cpu_1v8	1.80	0.88	1.59		
	buck2_cpu_1v0	1.00	34.08	34.09		
	buck3_cpu_1v0	1.09	347.14	377.48		
	buck4_cpu_1v1	1.10	0.49	0.53		
	ldo1_cpu_1v1	1.09	4.02	4.40		
	ldo1_cpu_1v1_0v6	0.60	41.98	25.00		
	ldo2_cpu_3v3	3.30	1.39	4.58		
	ldo4_cpu_1v8	1.80	0.37	0.66		
	ldo5_cpu_3v0	2.93	0.01	0.03		

5.3 Core benchmark use cases

The section explains how to test the following use case scenarios with the CA35 cores:

- Dhrystone
- CoreMark

5.3.1 Dhrystone

Dhrystone is a synthetic benchmark used to measure the integer computational performance of processors and compilers. The small size of the Dhrystone benchmark enables it to fit into the L1 cache and minimizes access to the L2 cache and DDR.

In this use case, the Dhrystone test is run on both CA35 cores (because Dhrystone is a single thread benchmark, two instances are started). Each CA35 core runs the test in a loop at a frequency of 800 MHz.

The following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value).
- The DDR frequency is set to 528 MHz (1056 MT/s).
- The CM33 core is in Sleep mode.
- The display is OFF.
- All the unused PLLs are OFF, and the unused clocks are gated.

To configure and run the use case, perform these steps:

Note: *Run the use case in a loop and log power measurement results at the desired time interval (the recommended interval is 1 minute).*

1. Power on the MCIMX8ULP-EVK9 board by turning on the ON/OFF switch (SW10).
2. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
3. From the menu on the RTD console, first press 'D', and then press 'S' to put RTD into Sleep mode.
4. Run `setup.sh` (see Section 8 for more details).
5. Run `dhrystone_loop.sh`:

```
while true; do
   taskset -c 0 ./dhry2 &
   taskset -c 1 ./dhry2
done
```

6. Measure the power and record the results.

Table 16 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 16. Power measurement results for "Dhrystone" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	16.16	29.12	493.69	36
	buck1_lsw1_cpu_1v8	1.80	0.06	0.11		
	buck1_lsw2_cpu_1v8	1.80	4.54	8.16		
	buck1_lsw3_cpu_1v8	1.80	0.00	0.00		
	buck1_lsw4_cpu_1v8	1.80	0.67	1.21		
	buck2_cpu_1v0	1.00	28.39	28.40		
	buck3_cpu_1v0	1.09	373.23	405.86		
	buck4_cpu_1v1	1.10	0.48	0.53		
	ldo1_cpu_1v1	1.10	3.50	3.84		
	ldo1_cpu_1v1_0v6	0.60	18.45	11.02		
	ldo2_cpu_3v3	3.30	1.39	4.58		
	ldo4_cpu_1v8	1.80	0.47	0.85		
	ldo5_cpu_3v0	2.92	0.01	0.03		

5.3.2 CoreMark

CoreMark is a modern and sophisticated benchmark that allows you to measure the processor performance accurately. It is intended to replace the older Dhrystone benchmark. Arm recommends using CoreMark over Dhrystone.

Note: *No display is connected to the platform.*

The following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value).
- The DDR frequency is set to 528 MHz (1056 MT/s).
- The CM33 core is in Sleep mode.
- The display is OFF.
- All the unused PLLs are OFF, and the unused clocks are gated.

To configure and run the use case, perform these steps:

Note: *Run the use case in a loop and log power measurement results at the desired time interval (the recommended interval is 1 minute).*

1. Power on the MCIMX8ULP-EVK9 board by turning on the ON/OFF switch (SW10).
2. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
3. From the menu on the RTD console, first press 'D', and then press 'S' to put RTD into Sleep mode.
4. Run `setup.sh` (see Section 8 for more details).
5. To get better performance, compile the `coremark` application (in i.MX 8ULP, the `coremark` application is built for dual cores):

   ```
   make XCFLAGS="-DMULTITHREAD=2 -DUSE_PTHREAD -pthread"
   ```

6. Run `coremark_loop.sh`:

   ```
   while true; do
      ./coremark
   done
   ```

7. Measure the power and record the results.

Table 17 shows the measurement results when this use case is applied to the i.MX 8ULP processor with the CoreMark benchmark running on the four CPU cores.

Table 17. Power measurement results for "CoreMark" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	16.19	29.17	457.06	35
	buck1_lsw1_cpu_1v8	1.80	0.05	0.09		
	buck1_lsw2_cpu_1v8	1.80	4.55	8.18		
	buck1_lsw3_cpu_1v8	1.80	-0.01	-0.01		
	buck1_lsw4_cpu_1v8	1.80	0.69	1.25		
	buck2_cpu_1v0	1.00	28.44	28.45		
	buck3_cpu_1v0	1.09	339.69	369.64		
	buck4_cpu_1v1	1.10	0.47	0.52		
	ldo1_cpu_1v1	1.10	3.49	3.83		

Table 17. Power measurement results for "CoreMark" use case*...continued*

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
	ldo1_cpu_1v1_0v6	0.60	18.39	10.99		
	ldo2_cpu_3v3	3.30	1.29	4.27		
	ldo4_cpu_1v8	1.80	0.36	0.66		
	ldo5_cpu_3v0	2.92	0.01	0.03		

5.4 GPU use cases

This section explains the following GPU use case scenarios:

- GLMark
- vkmark

5.4.1 GLMark

The following are the required settings for the GLmark use case:

- The CPU frequency is set to 800 MHz (default value).
- The DDR frequency is 528 MHz (1056 MT/s).
- The CM33 core is in Sleep mode.
- All the unused PLLs are OFF, and the unused clocks are gated.

To configure and run the use case, perform these steps:

1. Connect the display to the MCIMX8ULP-EVK9 board through the HDMI interface.
2. Power on the board by turning on the ON/OFF switch (SW10).
3. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
4. From the menu on the RTD console, first press 'D', and then press 'S' to put RTD into Sleep mode.
5. Run `setup_video.sh` (see Section 8 for more details).
6. Run `gpu_glmark.sh`:

```
glmark2-es2-wayland -b bump -fullscreen --run-forever
```

7. Start the power measurement and record the results.

Table 18 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 18. Power measurement results for "GLMark" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_ SOC_FULL	buck1_cpu_1v8	1.80	16.83	30.32	580.39	38
	buck1_lsw1_ cpu_1v8	1.80	0.06	0.10		
	buck1_lsw2_ cpu_1v8	1.80	4.53	8.15		
	buck1_lsw3_ cpu_1v8	1.80	1.76	3.16		
	buck1_lsw4_ cpu_1v8	1.80	0.73	1.31		
	buck2_cpu_1v0	1.00	34.10	34.11		
	buck3_cpu_1v0	1.09	420.22	456.56		
	buck4_cpu_1v1	1.10	0.46	0.50		
	ldo1_cpu_1v1	1.09	4.52	4.94		
	ldo1_cpu_1v1_ 0v6	0.59	60.48	35.95		
	ldo2_cpu_3v3	3.30	1.39	4.59		
	ldo4_cpu_1v8	1.80	0.37	0.67		
	ldo5_cpu_3v0	2.93	0.01	0.03		

5.4.2 vkmark

The following are the required settings for the vkmark use case:

- The CPU frequency is set to 800 MHz (default value).
- The DDR frequency is 528 MHz (1056 MT/s).
- The CM33 core is in Sleep mode.
- All the unused PLLs are OFF, and the unused clocks are gated.

To configure and run the use case, perform these steps:

1. Connect the display to the MCIMX8ULP-EVK9 board through the HDMI interface.
2. Power on the board by turning on the ON/OFF switch (SW10).
3. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
4. From the menu on the RTD console, first press 'D', and then press 'S' to put RTD into Sleep mode.
5. Run `setup_video.sh` (see Section 8 for more details).
6. Run `gpu_vkmark.sh`:
   ```
   vkmark -b clear --fullscreen --run-forever
   ```
7. Start the power measurement and record the results.

Table 19 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 19. Power measurement results for "vkmark_clear" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	16.48	29.70	596.16	37
	buck1_lsw1_cpu_1v8	1.80	0.06	0.11		
	buck1_lsw2_cpu_1v8	1.80	4.54	8.17		
	buck1_lsw3_cpu_1v8	1.80	1.77	3.19		
	buck1_lsw4_cpu_1v8	1.80	0.71	1.28		
	buck2_cpu_1v0	1.00	35.26	35.27		
	buck3_cpu_1v0	1.09	411.52	447.20		
	buck4_cpu_1v1	1.10	0.48	0.53		
	ldo1_cpu_1v1	1.09	3.87	4.23		
	ldo1_cpu_1v1_0v6	0.59	103.62	61.21		
	ldo2_cpu_3v3	3.30	1.39	4.59		
	ldo4_cpu_1v8	1.80	0.37	0.66		
	ldo5_cpu_3v0	2.93	0.01	0.03		

5.5 Heavy load use cases

This section explains the following heavy load use case scenarios:

- 2-core Dhrystone + GLMark + HiFi4 + CM33 CoreMark
- 2-core Stream + GLMark + HiFi4 + CM33 CoreMark
- GPU2D + GPU3D + CSI/DSI

The purpose of these use case scenarios is to show the power consumption in extreme conditions. For these use cases, a 720p TV display is connected to the MCIMX8ULP-EVK9 board through the HDMI interface.

5.5.1 2-core Dhrystone + GLMark + HiFi4 + CM33 CoreMark

This use case runs a 2-core Dhrystone, GPU GLMark, HiFi4, and CM33 CoreMark in parallel. A 720p TV display is connected to the MCIMX8ULP-EVK9 board through the HDMI interface.

The following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value).
- The CM33 core frequency is set to 160 MHz.
- The DDR frequency is set to 528 MHz (1056 MT/s).
- The HiFi4 is running the Math Power busy loop stress test.
- The GPU is running the GLMark_Bump stress test.
- Both CA35 cores are running the Dhrystone benchmark.

- The CM33 core is running the CoreMark benchmark.
- All the unused PLLs are OFF, and the unused clocks are gated.

To configure and run the use case, perform these steps:

1. Connect the display to the MCIMX8ULP-EVK9 board through the HDMI interface.
2. Change the SW5[8:1] settings on the board to 01xx_xxxx (Serial Download mode).
3. Download the `flash_pms_coremark.bin` binary image to eMMC:
   Note: *The `flash_pms_coremark.bin` image is different from the default binary image (`flash.bin`). Therefore, for other use cases that require the default binary image, you must download and use the `flash_singleboot_default.bin` image.*

   ```
   uuu -b emmc flash_pms_coremark.bin
   ```

4. Ensure that the board is configured to the default boot setting (SW5[8:1] = 1000_xx00 (Single Boot - eMMC mode)).
5. Power on the board by turning on the ON/OFF switch (SW10).
6. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted with CoreMark.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
7. Run `setup_video.sh` (see Section 8 for more details).
8. Start two Dhrystone instances, each bound on separate CPU:

   ```
   while true; do
      taskset -c 0 ./dhry2 &
      taskset -c 1 ./dhry2
   done
   ```

9. Copy `hifi4_imx8ulp-MathPower_busy_loop.bin` to `/lib/firmware/imx/dsp/` and rename it as `hifi4.bin`.
10. Use this command to load and run the firmware:

    ```
    echo start > /sys/class/remoteproc/remoteproc<n>/state
    ```

    where <n> is the HiFi4 (DSP) remote processor index.
11. Use this command to check the name of the HiFi4 (DSP) remote processor:

    ```
    cat /sys/class/remoteproc/remoteproc<n>/name
    ```

12. Press 'K' to run the CoreMark benchmark from the menu on the RTD console.
13. Run `gpu_glmark.sh`:

    ```
    glmark2-es2-wayland -b bump -fullscreen --run-forever
    ```

14. Start the power measurement and record the results.

Table 20 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 20. Power measurement results for "2-core Dhrystone + GLMark_Bump + HiFi4 + CM33 CoreMark" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	16.75	30.18	607.11	39
	buck1_lsw1_cpu_1v8	1.81	0.06	0.12		
	buck1_lsw2_cpu_1v8	1.80	4.57	8.2		
	buck1_lsw3_cpu_1v8	1.80	1.77	3.19		
	buck1_lsw4_cpu_1v8	1.80	0.79	1.42		
	buck2_cpu_1v0	1.00	38.41	38.42		
	buck3_cpu_1v0	1.09	447.33	485.73		
	buck4_cpu_1v1	1.10	0.48	0.52		
	ldo1_cpu_1v1	1.09	4.11	4.50		
	ldo1_cpu_1v1_0v6	0.60	49.37	29.38		
	ldo2_cpu_3v3	3.30	1.39	4.58		
	ldo4_cpu_1v8	1.80	0.46	0.83		
	ldo5_cpu_3v0	2.93	0.01	0.03		

5.5.2 2-core Stream + GLMark + HiFi4 + CM33 CoreMark

This use case runs 2-core Stream, GPU GLMark, HiFi4, and CM33 CoreMark in parallel. A 720p TV display is connected to the MCIMX8ULP-EVK9 board through the HDMI interface.

The following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value).
- The CM33 core frequency is set to 160 MHz.
- The DDR frequency is set to 528 MHz (1056 MT/s).
- The HiFi4 is running the Math Power busy loop stress test.
- The GPU is running the GLMark_Bump stress test.
- Both CA35 cores are running the Stream benchmark.
- The CM33 core is running the CoreMark benchmark.
- All the unused PLLs are OFF, and the unused clocks are gated.

To configure and run the use case, perform these steps:

1. Connect the display to the MCIMX8ULP-EVK9 board through the HDMI interface.
2. Change the SW5[8:1] settings on the board to 01xx_xxxx (Serial Download mode).
3. Download the `flash_pms_coremark.bin` binary image to eMMC:

Note: *The `flash_pms_coremark.bin` image is different from the default binary image (`flash.bin`). Therefore, for other use cases that require the default binary image, you must download and use the `flash_singleboot_default.bin` image.*

```
uuu -b emmc flash_pms_coremark.bin
```

4. Ensure that the board is configured to the default boot setting (SW5[8:1] = 1000_xx00 (Single Boot - eMMC mode)).
5. Power on the board by turning on the ON/OFF switch (SW10).
6. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted with Coremark.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
7. Run `setup_video.sh` (see Section 8 for more details).
8. Start two streams, each bound on separate CPU:

```
while true; do
    stream -P 2 -M 40M -N 5
done
```

9. Copy `hifi4_imx8ulp-MathPower_busy_loop.bin` to `/lib/firmware/imx/dsp/` and rename it as `hifi4.bin`.
10. Use this command to load and run the firmware:

```
echo start > /sys/class/remoteproc/remoteproc<n>/state
```

    where <n> is the HiFi4 (DSP) remote processor index.

11. Use this command to check the name of the HiFi4 (DSP) remote processor:

```
cat /sys/class/remoteproc/remoteproc<n>/name
```

12. Press 'K' to run the CoreMark benchmark from the menu on the RTD console.
13. Run `gpu_glmark.sh`:

```
glmark2-es2-wayland -b bump -fullscreen --run-forever
```

14. Start the power measurement and record the results.

Table 21 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 21. Power measurement results for "2-core Stream + GLMark_Bump + HiFi4 + CM33 CoreMark" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	16.86	30.39	612.47	39
	buck1_lsw1_cpu_1v8	1.81	0.08	0.15		
	buck1_lsw2_cpu_1v8	1.80	4.56	8.20		
	buck1_lsw3_cpu_1v8	1.80	1.77	3.18		
	buck1_lsw4_cpu_1v8	1.80	0.83	1.50		
	buck2_cpu_1v0	1.00	38.59	38.60		
	buck3_cpu_1v0	1.09	446.78	485.25		
	buck4_cpu_1v1	1.10	0.49	0.54		
	ldo1_cpu_1v1	1.09	4.64	5.07		
	ldo1_cpu_1v1_0v6	0.59	57.71	34.32		
	ldo2_cpu_3v3	3.30	1.39	4.59		
	ldo4_cpu_1v8	1.80	0.37	0.66		
	ldo5_cpu_3v0	2.93	0.01	0.03		

5.5.3 GPU2D + GPU3D + CSI/DSI

This use case runs the GPU2D, GPU3D, and CSI/DSI in parallel. A 720p TV display is connected to the MCIMX8ULP-EVK9 board through the HDMI interface.

The following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value).
- The DDR frequency is set to 528 MHz (1056 MT/s).
- The CM33 core is in Sleep mode.
- The GPU is running the GLMark benchmark.
- The G2D is running a G2D test program.
- MIPI-CSI/DSI is running camera image preview.
- All the unused PLLs are OFF, and the unused clocks are gated.

To configure and run the use case, perform these steps:

1. Connect the display to the MCIMX8ULP-EVK9 board through the HDMI interface.
2. Connect the OV5640 camera to the board through the CSI interface.
3. Power on the board by turning on the ON/OFF switch (SW10).
4. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.

- On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).

5. From the menu on the RTD console, first press 'D', and then press 'S' to put RTD into Sleep mode.
6. Run `setup_video.sh` (see Section 8 for more details).
7. Run a camera capture 1080p, 30 fps with the following command:

```
gst-launch-1.0 -v v4l2src device=/dev/video0 ! \
"video/x-raw,format=YUY2,width=1920,height=1080,framerate=30/1" ! \
queue ! waylandsink
```

8. Run the G2D test program with the following command:

```
while true; do
   /opt/g2d_samples/g2d_basic_test
done
```

9. Run the G3D test program with the following command:

```
while true; do
   glmark2-es2-wayland -s 350x350
done
```

10. Start the power measurement and record the results.

Table 22 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 22. Power measurement results for "GPU2D + GPU3D + CSI/DSI" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	16.66	30.01	590.64	38
	buck1_lsw1_cpu_1v8	1.80	0.06	0.10		
	buck1_lsw2_cpu_1v8	1.80	4.55	8.18		
	buck1_lsw3_cpu_1v8	1.80	2.97	5.34		
	buck1_lsw4_cpu_1v8	1.80	0.73	1.31		
	buck2_cpu_1v0	1.00	28.62	28.64		
	buck3_cpu_1v0	1.09	436.77	474.39		
	buck4_cpu_1v1	1.10	0.49	0.54		
	ldo1_cpu_1v1	1.09	4.65	5.08		
	ldo1_cpu_1v1_0v6	0.59	53.41	31.77		
	ldo2_cpu_3v3	3.30	1.39	4.59		
	ldo4_cpu_1v8	1.80	0.37	0.66		
	ldo5_cpu_3v0	2.93	0.01	0.03		

5.6 Memory use cases

This section explains the following memory-centric use case scenarios:

- memset
- memcpy
- Stream

memset and memcpy are part of a perf-bench (a general framework for benchmark suites).

5.6.1 memset

The purpose of this use case is to evaluate the performance of a simple memory set in various ways.

The following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value).
- The DDR frequency is set to 528 MHz (1056 MT/s).
- The size of the memory buffer is set to 1024 MB.
- The CM33 core is in Sleep mode.
- All the unused PLLs are OFF, and the unused clocks are gated.

To configure and run the use case, perform these steps:

1. Power on the MCIMX8ULP-EVK9 board by turning on the ON/OFF switch (SW10).
2. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
3. From the menu on the RTD console, first press 'D', and then press 'S' to put RTD into Sleep mode.
4. Run `setup.sh` (see Section 8 for more details).
5. Run `memset_loop.sh`:

```
while true; do
   buff_size=`cat /proc/meminfo | grep CmaFree | awk '{print$2}'`
   perf bench -f simple mem memset -l 20000 -s ${buff_size}KB
done
```

6. Start the power measurement and record the results.

Table 23 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 23. Power measurement results for "memset" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	16.06	28.93	517.40	35
	buck1_lsw1_cpu_1v8	1.80	0.09	0.15		
	buck1_lsw2_cpu_1v8	1.80	4.55	8.18		
	buck1_lsw3_cpu_1v8	1.80	0.00	0.00		
	buck1_lsw4_cpu_1v8	1.80	0.70	1.27		
	buck2_cpu_1v0	1.00	28.20	28.21		
	buck3_cpu_1v0	1.09	338.31	368.04		

Table 23. Power measurement results for "memset" use case*...continued*

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
	buck4_cpu_1v1	1.10	0.48	0.53		
	ldo1_cpu_1v1	1.09	3.53	3.85		
	ldo1_cpu_1v1_0v6	0.59	123.79	72.97		
	ldo2_cpu_3v3	3.30	1.39	4.58		
	ldo4_cpu_1v8	1.80	0.37	0.67		
	ldo5_cpu_3v0	2.93	0.01	0.03		

5.6.2 memcpy

The purpose of this use case is to evaluate the performance of a simple memory copy in various ways.

The following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value).
- The DDR frequency is set to 528 MHz (1056 MT/s).
- The size of the memory buffer is set to 1024 MB.
- The CM33 core is in Sleep mode.
- All the unused PLLs are OFF, and the unused clocks are gated.

To configure and run the use case, perform these steps:

1. Power on the MCIMX8ULP-EVK9 board by turning on the ON/OFF switch (SW10).
2. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
3. From the menu on the RTD console, first press 'D', and then press 'S' to put RTD into Sleep mode.
4. Run `setup.sh` (see Section 8 for more details).
5. Run `memcpy_loop.sh`:

```
while true; do
   buff_size=`cat /proc/meminfo | grep CmaFree | awk '{print$2}'`
   perf bench -f simple mem memcpy -l 20000 -s ${buff_size}KB
done
```

6. Start the power measurement and record the results.

Table 24 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 24. Power measurement results for "memcpy" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	16.13	29.07	486.63	35
	buck1_lsw1_cpu_1v8	1.80	0.08	0.14		

Table 24. Power measurement results for "memcpy" use case*...continued*

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
	buck1_lsw2_cpu_1v8	1.80	4.52	8.13		
	buck1_lsw3_cpu_1v8	1.8	0.01	0.02		
	buck1_lsw4_cpu_1v8	1.80	0.72	1.30		
	buck2_cpu_1v0	1.00	28.51	28.53		
	buck3_cpu_1v0	1.09	348.56	379.20		
	buck4_cpu_1v1	1.10	0.49	0.54		
	ldo1_cpu_1v1	1.09	4.58	5.01		
	ldo1_cpu_1v1_0v6	0.60	49.43	29.42		
	ldo2_cpu_3v3	3.30	1.39	4.58		
	ldo4_cpu_1v8	1.80	0.37	0.66		
	ldo5_cpu_3v0	2.92	0.01	0.03		

5.6.3 Stream

The Stream benchmark is a simple synthetic benchmark program that measures the sustainable memory bandwidth (in MB/s) and the corresponding computation rate for simple vector kernels.

The following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value).
- The DDR frequency is set to 528 MHz (1056 MT/s).
- All phases are included (Copy, Scale, Add, and Triad).
- The CM33 core is in Sleep mode.
- All the unused PLLs are OFF, and the unused clocks are gated.

To configure and run the use case, perform these steps:

1. Power on the MCIMX8ULP-EVK9 board by turning on the ON/OFF switch (SW10).
2. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
3. From the menu on the RTD console, first press 'D', and then press 'S' to put RTD into Sleep mode.
4. Run `setup.sh` (see Section 8 for more details).
5. Run `streamcpy_loop.sh`:

```
while true; do
  stream -P 2 -M 40M -N
done
```

6. Start the power measurement and record the results.

Table 25 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 25. Power measurement results for "Stream" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	16.16	29.11	571.35	38
	buck1_lsw1_cpu_1v8	1.80	0.09	0.17		
	buck1_lsw2_cpu_1v8	1.80	4.54	8.17		
	buck1_lsw3_cpu_1v8	1.80	0.01	0.02		
	buck1_lsw4_cpu_1v8	1.80	0.70	1.26		
	buck2_cpu_1v0	1.00	28.42	28.44		
	buck3_cpu_1v0	1.09	424.33	461.01		
	buck4_cpu_1v1	1.10	0.47	0.51		
	ldo1_cpu_1v1	1.09	4.69	5.12		
	ldo1_cpu_1v1_0v6	0.59	54.24	32.27		
	ldo2_cpu_3v3	3.30	1.39	4.58		
	ldo4_cpu_1v8	1.80	0.36	0.65		
	ldo5_cpu_3v0	2.93	0.01	0.03		

5.7 Storage (eMMC) use cases

This section explains the following storage-related use case scenarios:

- DD_RD_eMMC
- DD_WRT_eMMC

Note: *Create a partition on the eMMC memory and run the benchmarks on it.*

5.7.1 DD_RD_eMMC

This use case reads the data on the eMMC partition using the `dd` command.

The following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value).
- The DDR frequency is set to 528 MHz (1056 MT/s).
- The CM33 core is in Sleep mode.
- All the unused PLLs are OFF, and the unused clocks are gated.

To configure and run the use case, perform these steps:

Note: *Run the use case in a loop and log power measurement results at the desired time interval (the recommended interval is 1 minute).*

1. Power on the MCIMX8ULP-EVK9 board by turning on the ON/OFF switch (SW10).
2. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.

- On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).

3. From the menu on the RTD console, first press 'D', and then press 'S' to put RTD into Sleep mode.
4. Run `setup.sh` (see Section 8 for more details).
5. Copy `dd_read_emmc.sh` on the eMMC partition and run it.
6. Start the power measurement and record the results.

Table 26 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 26. Power measurement results for "DD_RD_eMMC" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	16.11	29.03	454.27	37
	buck1_lsw1_cpu_1v8	1.80	0.06	0.11		
	buck1_lsw2_cpu_1v8	1.80	4.54	8.15		
	buck1_lsw3_cpu_1v8	1.80	0.01	0.01		
	buck1_lsw4_cpu_1v8	1.80	0.74	1.33		
	buck2_cpu_1v0	1.00	28.36	28.36		
	buck3_cpu_1v0	1.09	328.96	357.98		
	buck4_cpu_1v1	1.10	0.49	0.53		
	ldo1_cpu_1v1	1.10	4.01	4.40		
	ldo1_cpu_1v1_0v6	0.60	31.98	19.07		
	ldo2_cpu_3v3	3.30	1.39	4.58		
	ldo4_cpu_1v8	1.80	0.39	0.70		
	ldo5_cpu_3v0	2.93	0.01	0.03		

5.7.2 DD_WRT_eMMC

This use case writes the data on the eMMC partition using the `dd` command.

The following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value).
- The DDR frequency is set to 528 MHz (1056 MT/s).
- The CM33 core is in Sleep mode.
- All the unused PLLs are OFF, and the unused clocks are gated.

Note: *This command overwrites data at 10 GB offset on user space area. Use this command carefully to avoid any damage to your rootfs.*

To configure and run the use case, perform these steps:

Note: *Run the use case in a loop and log power measurement results at the desired time interval (the recommended interval is 1 minute).*

1. Power on the MCIMX8ULP-EVK9 board by turning on the ON/OFF switch (SW10).
2. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
3. From the menu on the RTD console, first press 'D', and then press 'S' to put RTD into Sleep mode.
4. Run `setup.sh` (see Section 8 for more details).
5. Copy `dd_write_emmc.sh` on the eMMC partition and run it.
6. Start the power measurement and record the results.

Table 27 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 27. Power measurement results for "DD_WRT_eMMC" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_ SOC_FULL	buck1_cpu_1v8	1.80	16.40	29.55	392.33	35
	buck1_lsw1_ cpu_1v8	1.80	0.06	0.11		
	buck1_lsw2_ cpu_1v8	1.80	4.53	8.15		
	buck1_lsw3_ cpu_1v8	1.80	0.00	0.00		
	buck1_lsw4_ cpu_1v8	1.80	0.72	1.29		
	buck2_cpu_1v0	1.00	28.39	28.40		
	buck3_cpu_1v0	1.09	266.98	290.73		
	buck4_cpu_1v1	1.10	0.48	0.53		
	ldo1_cpu_1v1	1.10	3.61	3.96		
	ldo1_cpu_1v1_ 0v6	0.60	24.10	14.38		
	ldo2_cpu_3v3	3.30	1.39	4.58		
	ldo4_cpu_1v8	1.79	5.93	10.62		
	ldo5_cpu_3v0	2.93	0.01	0.03		

5.8 Product use cases

This section explains the following use case scenarios:

- Machine vision
- eIQ – Benchmark
- UAC audio playback
- Always-ON display, 1 fps refresh rate

5.8.1 Machine vision

This use case detects an object in a video file, which is captured from the OV5640 camera.

To obtain the required files for the "Machine vision" use case, use the following commands in `/usr/bin/tensorflow-lite-2.15.0/examples` (the path varies with BSP revision):

```
wget https://github.com/google-coral/test_data/raw/master/
ssd_mobilenet_v2_coco_quant_postprocess.tflite
wget https://github.com/google-coral/test_data/raw/master/coco_labels.txt
```

The following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value).
- Capture 1080p, 30 fps YUY2 data from OV5640.
- Use G2D to perform color space conversion and downsize to 480p.
- Object detection is done using the file-based input.
- To simulate application stress, stress each CA35 core — 100%.
- The CM33 core is in Sleep mode.
- All the unused PLLs are OFF, and the unused clocks are gated.

To configure and run the use case, perform these steps:

Note: *Run the use case in a loop and log power measurement results at the desired time interval (the recommended interval is 1 minute).*

1. Connect the display to the MCIMX8ULP-EVK9 board through the HDMI interface.
2. Power on the board by turning on the ON/OFF switch (SW10).
3. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
4. From the menu on the RTD console, first press 'D', and then press 'S' to put RTD into Sleep mode.
5. Run `setup.sh` (see Section 8 for more details).
6. Run object detection using the file-based input
7. Copy `MV.sh` to `/usr/bin/tensorflow-lite-2.15.0/examples` in rootfs and run `/usr/bin/tensorflow-lite-2.15.0/examples/MV.sh`. For more details, see Section 8.
   Note: *To record the test video in this shell file, use the below command:*
   ```
   gst-launch-1.0 v4l2src device=/dev/video0 !  videoconvert ! 'video/x-
   raw,format=YUY2,width=640,height=480,framerate=(fraction)30/1'  ! filesink
    location=yuv.raw
   ```
8. Start the power measurement and record the results.

Table 28 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 28. Power measurement results for "Machine vision" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	16.42	29.59	547.16	37
	buck1_lsw1_cpu_1v8	1.80	0.04	0.07		
	buck1_lsw2_cpu_1v8	1.80	4.54	8.17		
	buck1_lsw3_cpu_1v8	1.80	2.55	4.58		

Table 28. Power measurement results for "Machine vision" use case...*continued*

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
	buck1_lsw4_ cpu_1v8	1.80	0.68	1.23		
	buck2_cpu_1v0	1.00	28.67	28.69		
	buck3_cpu_1v0	1.09	400.19	435.01		
	buck4_cpu_1v1	1.10	0.49	0.53		
	ldo1_cpu_1v1	1.09	4.32	4.73		
	ldo1_cpu_1v1_ 0v6	0.60	45.49	27.08		
	ldo2_cpu_3v3	3.30	1.30	4.28		
	ldo4_cpu_1v8	1.80	1.77	3.16		
	ldo5_cpu_3v0	2.93	0.01	0.03		

5.8.2 eIQ benchmark

This use case runs the `benchmark_model` application in a loop to test the eIQ benchmark.

The following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value).
- Run the eIQ benchmark test (standard models).
- All the unused PLLs are OFF, and the unused clocks are gated.

To configure and run the use case, perform these steps:

Note: *Run the use case in a loop and log power measurement results at the desired time interval (the recommended interval is 1 minute).*

1. Power on the MCIMX8ULP-EVK9 board by turning on the ON/OFF switch (SW10).
2. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
3. From the menu on the RTD console, first press 'D', and then press 'S' to put RTD into Sleep mode.
4. Run `setup.sh` (see Section 8 for more details).
5. Copy `ML.sh` to `/usr/bin/tensorflow-lite-2.15.0/examples` in rootfs and run `/usr/bin/tensorflow-lite-2.15.0/examples/ML.sh 1`
6. Start the power measurement and record the results.

Table 29 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 29. Power measurement results for "eIQ benchmark" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_ SOC_FULL	buck1_cpu_1v8	1.80	16.16	29.12	424.43	34
	buck1_lsw1_ cpu_1v8	1.80	0.06	0.10		

Table 29. Power measurement results for "eIQ benchmark" use case*...continued*

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
	buck1_lsw2_cpu_1v8	1.80	4.54	8.17		
	buck1_lsw3_cpu_1v8	1.80	0.00	0.00		
	buck1_lsw4_cpu_1v8	1.80	0.71	1.28		
	buck2_cpu_1v0	1.00	28.23	28.24		
	buck3_cpu_1v0	1.09	306.98	334.20		
	buck4_cpu_1v1	1.10	0.48	0.53		
	ldo1_cpu_1v1	1.10	3.54	3.88		
	ldo1_cpu_1v1_0v6	0.60	23.37	13.95		
	ldo2_cpu_3v3	3.30	1.29	4.28		
	ldo4_cpu_1v8	1.80	0.37	0.66		
	ldo5_cpu_3v0	2.93	0.01	0.03		

5.8.3 UAC audio playback

This use case involves creating a Linux UAC gadget device on an i.MX 8ULP processor, connecting the device to an Ubuntu PC (host) through the USB interface, receiving host audio through UAC, and decoding/playing the audio using the onboard codec.

The following are the required settings for this use case:

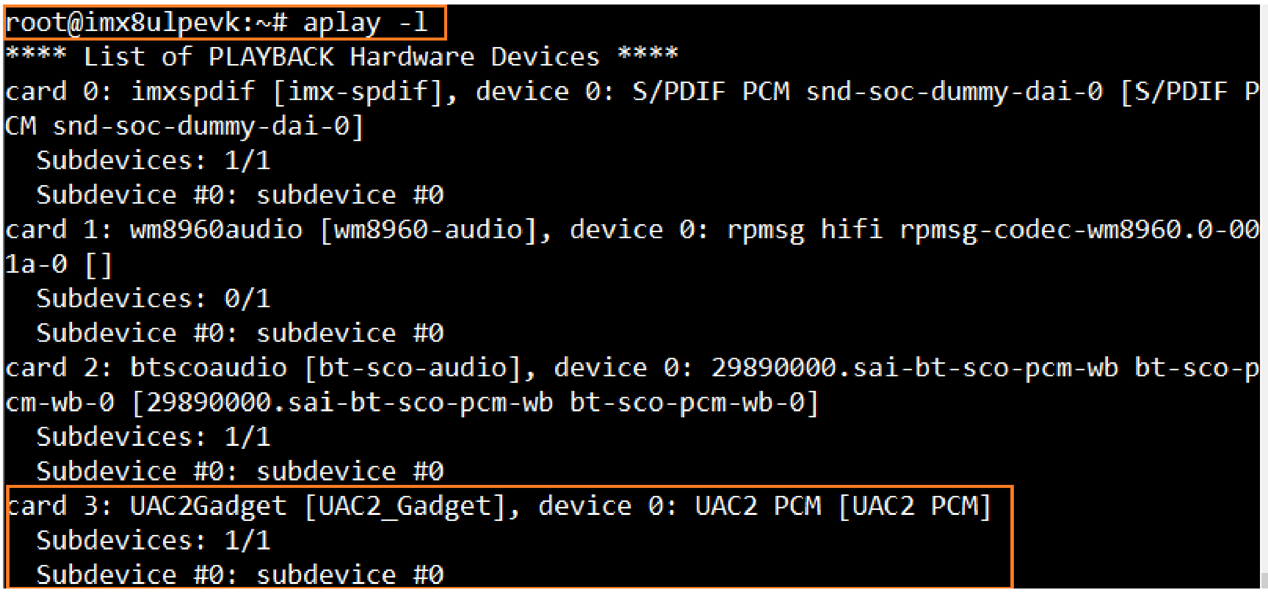
- The CPU frequency is set to 800 MHz (default value).
- The DDR frequency is set to 528 MHz (1056 MT/s).
- The CM33 core is in Sleep mode.
- All the unused PLLs are OFF, and the unused clocks are gated.

To configure and run the use case, perform these steps:

Note: *Run the use case in a loop and log power measurement results at the desired time interval (the recommended interval is 1 minute).*

1. Prepare a PC host with the latest Ubuntu installed.
2. Connect USB0 of the MCIMX8ULP-EVK9 board to the Ubuntu PC with a Type-C USB cable.
3. Power on the board by turning on the ON/OFF switch (SW10).
4. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
5. From the menu on the RTD console, first press 'D', and then press 'S' to put RTD into Sleep mode.
6. Run `setup.sh` on the board (see Section 8 for more details).
7. Run `uac2_gadget.sh` on the board (see Section 8 for more details).
8. To ensure that the Ubuntu PC can play audio with the onboard codec (WM8960), check if the WM8960 device is detected on the Ubuntu PC, by running the `aplay -l` command, as shown in Figure 5.

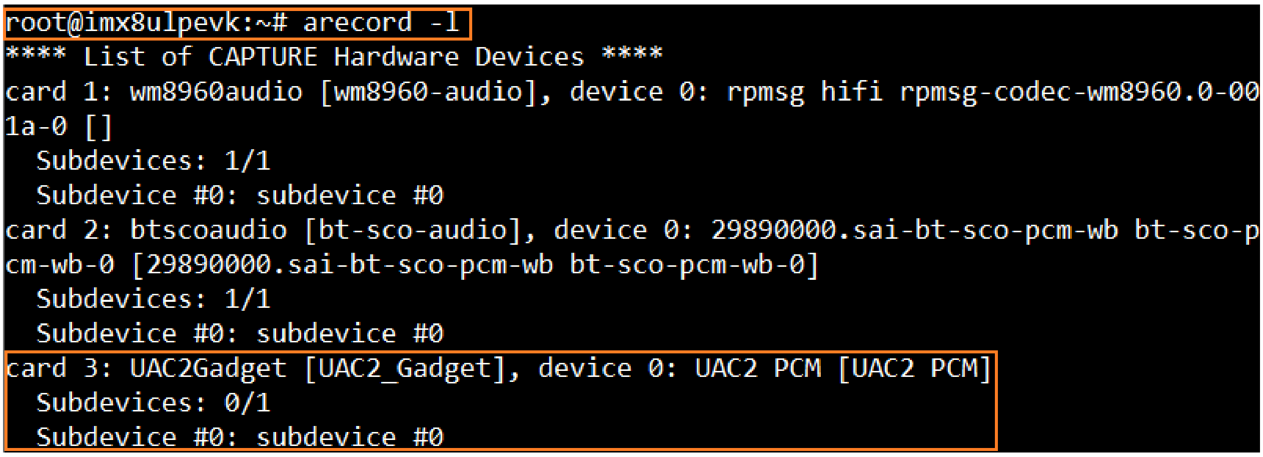
Note: *If the UAC gadget device is not listed after running the* `aplay -l` *command, check the* `uac2_gadget.sh` *script file to ensure that* `UDC` *in the last command (*`echo ci_hdrc.0 > UDC #8mm 8mn 8ulp`*) corresponds to the name of the UDC gadget available at the current path.*



```
root@imx8ulpevk:~# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: imxspdif [imx-spdif], device 0: S/PDIF PCM snd-soc-dummy-dai-0 [S/PDIF P
CM snd-soc-dummy-dai-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: wm8960audio [wm8960-audio], device 0: rpmsg hifi rpmsg-codec-wm8960.0-00
1a-0 []
  Subdevices: 0/1
  Subdevice #0: subdevice #0
card 2: btscoaudio [bt-sco-audio], device 0: 29890000.sai-bt-sco-pcm-wb bt-sco-p
cm-wb-0 [29890000.sai-bt-sco-pcm-wb bt-sco-pcm-wb-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 3: UAC2Gadget [UAC2_Gadget], device 0: UAC2 PCM [UAC2 PCM]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

Figure 5. Output of "aplay -l" command on MCIMX8ULP-EVK9 board

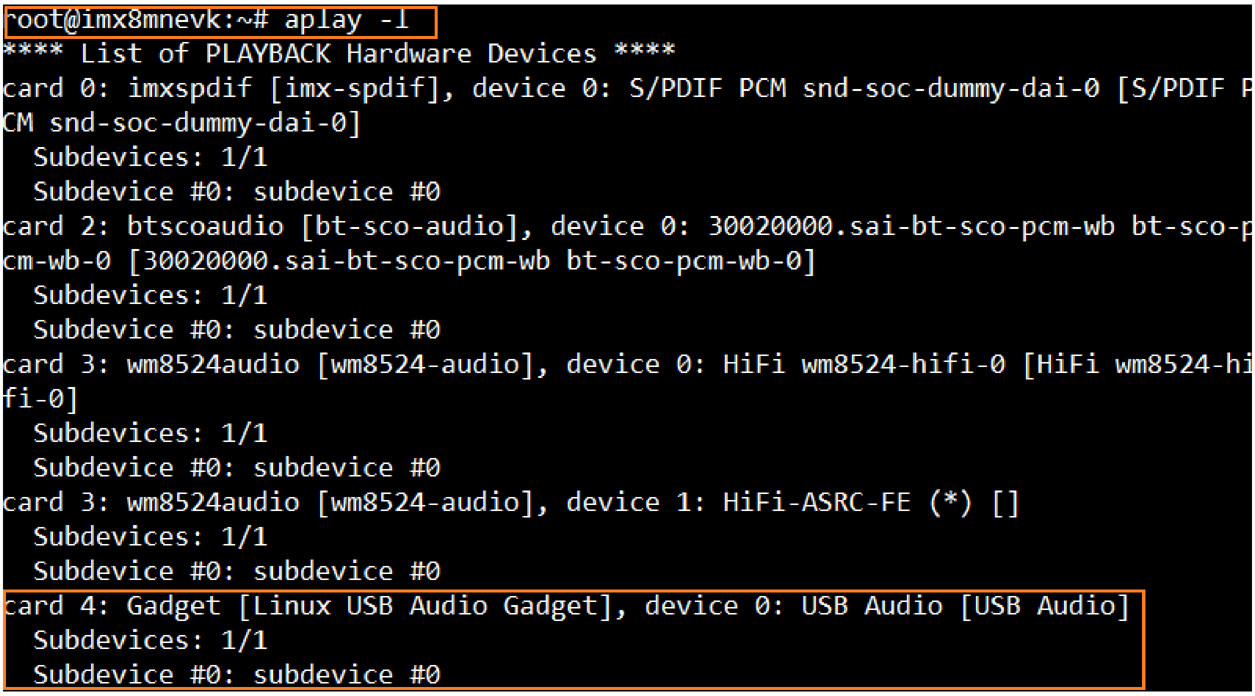
9. To ensure that the MCIMX8ULP-EVK9 board can record audio from the UAC gadget device, check if the UAC gadget device is detected on the board, by running the `arecord -l` command, as shown in Figure 6.
Note: *If the WM8960 audio device is not listed after running the* `arecord -l` *command, check the* `uac2_gadget.sh` *script file to ensure that* `UDC` *in the last command (*`echo ci_hdrc.0 > UDC #8mm 8mn 8ulp`*) corresponds to the name of the UDC gadget available at the current path.*



```
root@imx8ulpevk:~# arecord -l
**** List of CAPTURE Hardware Devices ****
card 1: wm8960audio [wm8960-audio], device 0: rpmsg hifi rpmsg-codec-wm8960.0-00
1a-0 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 2: btscoaudio [bt-sco-audio], device 0: 29890000.sai-bt-sco-pcm-wb bt-sco-p
cm-wb-0 [29890000.sai-bt-sco-pcm-wb bt-sco-pcm-wb-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 3: UAC2Gadget [UAC2_Gadget], device 0: UAC2 PCM [UAC2 PCM]
  Subdevices: 0/1
  Subdevice #0: subdevice #0
```

Figure 6. Output of "arecord -l" command on MCIMX8ULP-EVK9 board

10. To ensure that the Ubuntu PC can play audio to the UAC gadget device, check if the UAC gadget device is detected on the Ubuntu PC, as shown in Figure 7.



```
root@imx8mnevk:~# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: imxspdif [imx-spdif], device 0: S/PDIF PCM snd-soc-dummy-dai-0 [S/PDIF P
CM snd-soc-dummy-dai-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 2: btscoaudio [bt-sco-audio], device 0: 30020000.sai-bt-sco-pcm-wb bt-sco-p
cm-wb-0 [30020000.sai-bt-sco-pcm-wb bt-sco-pcm-wb-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 3: wm8524audio [wm8524-audio], device 0: HiFi wm8524-hifi-0 [HiFi wm8524-hi
fi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 3: wm8524audio [wm8524-audio], device 1: HiFi-ASRC-FE (*) []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 4: Gadget [Linux USB Audio Gadget], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

Figure 7. UAC gadget device detection on Ubuntu PC

11. Use this command to play music on the Ubuntu PC:

```
aplay -Dhw:4,0 audio48k16b2c.wav -vv
```

Note: *Replace '4' in `-Dhw:4,0` in the above command with your UAC gadget device index, which you got after running the `aplay -l` command on the Ubuntu PC.*

12. Use this command to receive music data and play on the MCIMX8ULP-EVK9 board:

```
arecord -Dhw:3,0 -c2 -r48000 -fS16_LE | aplay -Dhw:1,0
```

Note: *Replace '3' in `-Dhw:3,0` in the above command with your UAC gadget device index, which you got after running the `arecord -l` command on the MCIMX8ULP-EVK9 board.*

Note: *Replace '1' in `-Dhw:1,0` in the above command with your WM8960 audio card index, which you got after running the `aplay -l` command on the MCIMX8ULP-EVK9 board.*

13. Check if music can be heard from the 3.5 mm audio jack on the MCIMX8ULP-EVK9 board.
14. Start the power measurement and record the results.

Table 30 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 30. Power measurement results for "UAC audio playback" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	16.19	29.17	374.43	32
	buck1_lsw1_cpu_1v8	1.80	0.05	0.09		
	buck1_lsw2_cpu_1v8	1.79	8.17	14.66		
	buck1_lsw3_cpu_1v8	1.80	0.00	0.00		

Table 30. Power measurement results for "UAC audio playback" use case*...continued*

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
	buck1_lsw4_cpu_1v8	1.80	0.70	1.27		
	buck2_cpu_1v0	1.00	35.52	35.51		
	buck3_cpu_1v0	1.09	243.78	265.57		
	buck4_cpu_1v1	1.10	0.48	0.52		
	ldo1_cpu_1v1	1.10	3.51	3.85		
	ldo1_cpu_1v1_0v6	0.60	22.79	13.61		
	ldo2_cpu_3v3	3.30	2.88	9.50		
	ldo4_cpu_1v8	1.80	0.37	0.66		
	ldo5_cpu_3v0	2.92	0.01	0.03		

5.8.4 Always-ON display, 1 fps refresh rate

This use case shows the CM33 core waking up from Sleep mode every second to update the buffer for display.

The following are the required settings for this use case:

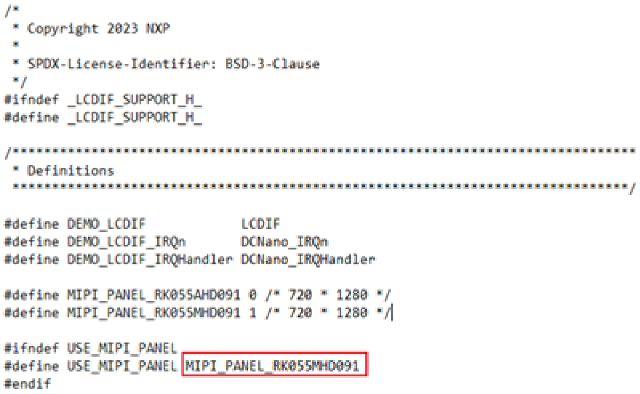
- APD domain is in Suspend (Power-Down) mode.
- DDR is in Retention mode.
- The CM33 core reinitializes the LPAV domain to control MIPI-DSI and DCNano.
- A frame buffer is allocated in pSRAM through FlexSPI.
- DCNano fetches display content from pSRAM.
- The CM33 core wakes up from Sleep mode every second to update the buffer for display.
- All the unused PLLs are OFF, and the unused clocks are gated.

To configure and run the use case, perform these steps:

1. Connect the RK055HDMIPI4M panel to the MCIMX8ULP-EVK9 board through the MIPI interface.
2. Download the latest CM33 SDK (SDK 2.14.1 for power measurement results provided in this document) from https://mcuxpresso.nxp.com/en.
3. Build the "low_power_display" release project, and then build the `flash.bin` image using the `m33_image` image, which can be found in `{CURRENT REPO}\armgcc\release\sdk20-app.bin` or `{CURRENT REPO}\iar\Release\sdk20-app.bin`.
   Note: *For details on how to build the `flash.bin` image, refer to Section 6, Step 5 from Getting Started with MCUXpresso SDK for EVK-MIMX8ULP and EVK9-MIMX8ULP in `SDK_2_xx_x_EVK-MIMX8ULP/docs`.*
4. Change SW5[8:1] settings on the board to 01xx_xxxx (Serial Download mode).
5. Download the `flash.bin` binary image to eMMC:
   Note: *The new `flash.bin` image is different from the default `flash.bin` image. Therefore, for other use cases that require the default binary image, you must download and use the `flash_singleboot_default.bin` image.*

   ```
   uuu -b emmc flash.bin
   ```

***Note:** Both RK055AHD091 and RK055MHD091 belong to RK055HDMIPI4M. If the image fails to display, modify the `USE_MIPI_PANEL` value from `MIPI_PANEL_RK055MHD091` to `MIPI_PANEL_RK055AHD091` in the `lcdif_support.h` file:*



```
/*
 * Copyright 2023 NXP
 *
 * SPDX-License-Identifier: BSD-3-Clause
 */
#ifndef _LCDIF_SUPPORT_H_
#define _LCDIF_SUPPORT_H_

/*******************************************************************************
 * Definitions
 ******************************************************************************/

#define DEMO_LCDIF            LCDIF
#define DEMO_LCDIF_IRQn       DCNano_IRQn
#define DEMO_LCDIF_IRQHandler DCNano_IRQHandler

#define MIPI_PANEL_RK055AHD091 0 /* 720 * 1280 */
#define MIPI_PANEL_RK055MHD091 1 /* 720 * 1280 */

#ifndef USE_MIPI_PANEL
#define USE_MIPI_PANEL MIPI_PANEL_RK055MHD091
#endif
```

6. Ensure that the board is configured to the default boot setting (SW5[8:1] = 1000_xx00 (Single Boot - eMMC mode)).
7. Power on the board by turning on the ON/OFF switch (SW10).
8. Ensure that:
   - On the RTD console, the Always-ON Display (AOD) demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
9. Put the Linux kernel into the Suspend (Power-Down) mode:

```
echo mem > /sys/power/state
```

10. On the RTD console, press 'Z' to run the low-power display.
11. Check if the entire screen is filled with colors where red, green, and blue colors are switched every second.
12. Start the power measurement and record the results.

Table 31 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 31. Power measurement results for "Always-ON display, 1 fps refresh rate" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	5.59	10.08	90.21	Die temperature was not measured as the CA35 core was suspended.
	buck1_lsw1_cpu_1v8	1.80	1.16	2.09		
	buck1_lsw2_cpu_1v8	1.80	0.00	0.01		
	buck1_lsw3_cpu_1v8	1.80	1.29	2.33		
	buck1_lsw4_cpu_1v8	1.80	0.00	0.00		
	buck2_cpu_1v0	1.00	32.98	32.98		
	buck3_cpu_1v0	0.99	42.90	42.64		
	buck4_cpu_1v1	1.10	0.00	0.00		
	ldo1_cpu_1v1	0.00	0.01	0.00		
	ldo1_cpu_1v1_0v6	0.00	-0.02	0.00		
	ldo2_cpu_3v3	3.31	0.01	0.02		
	ldo4_cpu_1v8	1.81	0.00	0.01		
	ldo5_cpu_3v0	2.90	0.02	0.05		

5.9 System-level power estimation use cases

This section explains the following use case scenarios:

- Battery
- 256 KB L2 cache

5.9.1 Battery

VBAT mode of the i.MX 8ULP processor is a low-power mode having only the VBAT domain powered. VBAT mode is a chip-level state with the following conditions:

- All power supplies except VDD_VBAT42 are OFF externally.
- VDD_VBAT42 is ON and within the voltage range specified in the i.MX 8ULP data sheet.
- The secure real-time clock (SRTC) is maintained and is running.
- Tamper logic is retained.
- SNVS is at 1.8 V DGO (VBAT input: 3 V).
- All clocks and PLLs in APD and LPAV are turned off.

In VBAT mode, the application is OFF, and a battery retains the SRTC and the tamper logic.

To configure and run the use case, perform these steps:

1. Power on the MCIMX8ULP-EVK9 board by turning on the ON/OFF switch (SW10).
2. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
3. Press the ON/OFF push button (SW6) for 3 seconds to turn off the i.MX 8ULP processor.
4. Measure the power and record the results.

Table 32 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 32. Power measurement results for "Battery" use case

Power group	Power rail	Average voltage (V)	Average current (µA)	Average power (µW)	Sum of average powers (µW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	0.00	-0.01	0.00	9.15	Die temperature was not measured as the CA35 core was suspended.
	buck1_lsw1_cpu_1v8	0.00	0.00	0.00		
	buck1_lsw2_cpu_1v8	0.00	0.00	0.00		
	buck1_lsw3_cpu_1v8	0.00	0.00	0.00		
	buck1_lsw4_cpu_1v8	0.00	0.00	0.00		
	buck2_cpu_1v0	0.00	-0.02	0.00		
	buck3_cpu_1v0	0.00	-0.02	0.00		
	buck4_cpu_1v1	0.00	0.00	0.00		
	ldo1_cpu_1v1	0.00	0.00	0.00		
	ldo1_cpu_1v1_0v6	0.00	-0.01	0.00		
	ldo2_cpu_3v3	0.00	0.00	0.00		
	ldo4_cpu_1v8	0.00	0.00	0.00		
	ldo5_cpu_3v0	2.95	0.00	0.01		

Note: *The unit of power is µW for these use cases:*

- *Suspend mode – RTD power down*
- *Battery*

5.9.2 256 KB L2 cache

This use case requires running the CoreMark benchmark on the CA35 cores with half the L2 cache size.

The following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value).
- The DDR frequency is set to 528 MHz (1056 MT/s).
- The size of the Arm L2 cache is set to 256 KB (the default size is 512 KB).
- RTD is in Sleep mode.
- All the unused PLLs are OFF, and the unused clocks are gated.

To configure and run the use case, perform these steps:

Note: *Run the use case in a loop and log power measurement results at the desired time interval (the recommended interval is 1 minute).*

1. Change the SW5[8:1] settings on the MCIMX8ULP-EVK9 board to 01xx_xxxx (Serial Download mode).
2. Download the `flash_l2_256k_cache.bin` binary image to eMMC:
   Note: *The `flash_l2_256k_cache.bin` image is different from the default binary image (`flash.bin`). Therefore, for other use cases that require the default binary image, you must download and use the `flash_singleboot_default.bin` image.*

   ```
   uuu -b emmc flash_l2_256k_cache.bin
   ```

3. Power on the board by turning on the ON/OFF switch (SW10).
4. Ensure that:
   - On the RTD console, the Power Mode Switch (PMS) demo is booted.
   - On the APD console, the Linux image is booted with the default DTB configuration (`imx8ulp-9x9-evk.dtb`).
5. Run `setup.sh`.
6. Run `coremark_loop.sh`:

   ```
   while true;
   do
    ./coremark > /dev/null 2>&1
   done
   ```

7. From the menu on the RTD console, first press 'D', and then press 'S' to put RTD into Sleep mode.
8. Measure the power and record the results.

Table 33 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 33. Power measurement results for "256 KB L2 cache" use case

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	buck1_cpu_1v8	1.80	16.14	29.08	461.20	34
	buck1_lsw1_cpu_1v8	1.80	0.05	0.09		
	buck1_lsw2_cpu_1v8	1.80	4.53	8.14		
	buck1_lsw3_cpu_1v8	1.80	0.00	0.00		
	buck1_lsw4_cpu_1v8	1.80	0.70	1.27		
	buck2_cpu_1v0	1.00	28.21	28.22		
	buck3_cpu_1v0	1.09	343.51	373.76		
	buck4_cpu_1v1	1.10	0.49	0.53		
	ldo1_cpu_1v1	1.10	3.49	3.83		
	ldo1_cpu_1v1_0v6	0.60	18.43	11.01		
	ldo2_cpu_3v3	3.30	1.39	4.58		
	ldo4_cpu_1v8	1.80	0.37	0.66		

Table 33. Power measurement results for "256 KB L2 cache" use case*...continued*

Power group	Power rail	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
	ldo5_cpu_3v0	2.92	0.01	0.03		

6 Recommendations for reducing power consumption

The overall system power consumption depends on both the software optimization and the system hardware implementation.

The following are some recommendations that may help reduce system power consumption:

- Apply clock gating by configuring registers in the Peripheral Clock Controller (PCC) module, whenever clocks or modules are not used.
- For active modes, use the slowest frequency that can support application requirements.
- Reduce the number of active PLLs whenever possible. Enabled PLLs can consume a few milliamperes of current.
- Core VFS and system bus scaling: Applying VFS on the Arm cores and scaling the frequencies of the AXI, AHB, and IPG bus clocks can significantly reduce the power consumption. However, due to reduced operating frequency, accessing the DDR takes more time. It increases the power consumption of the DDR I/O and memories. The trade-off between the two must be considered for each mode, to quantify the overall effect on system power.
- Put NXP i.MX 8ULP into low-power modes whenever possible and into the lowest power mode that supports the requirements of the current application.
- For each operating mode, use the lowest voltage (with the power supply tolerance) that meets the requirements of voltage specifications in the i.MX 8ULP data sheet.
- DDR interface optimization:
  - Use careful board routing of the DDR memories, maintaining PCB trace lengths as short as possible.
  - Use an appropriate output driver impedance for DDR interface pins that provides good impedance matching. To save current through DDR I/O pins, select the lowest possible drive strength that provides the required performance.
  - Set the i.MX 8ULP DDR interface pins high-Z when DDR memory is in Self-Refresh mode. Turn OFF `VDDQ_DDR`, `VDDQX_DDR`, and the I/O supply for LPDDR3/LPDDR4/LPDDR4x (VDDQ) when DDR memory is in Self-Refresh mode.
  - Use of LPDDR3/LPDDR4/LPDDR4x memory offerings in the latest process technology can significantly reduce the power consumption of the DDR devices and the DDR I/O.

Some of these recommendations are already implemented in the Linux BSP and/or SDK. Further optimizations can be done on the systems of individual users.

Note: *Further power optimizations are planned in future software releases. To obtain the latest software releases, visit https://www.nxp.com/imxsw.*

Note: *For details on how to optimize i.MX 8ULP through software, see Optimizing Power Consumption for i.MX 8ULP (AN13951).*

The subsections below describe some common system optimization techniques that can be used to reduce system power consumption:

- Section 6.1 "Run fast and idle"
- Section 6.2 "Clock gating"
- Section 6.3 "PLL reduction"
- Section 6.4 "Core VFS and system bus scaling"

- Section 6.5 "Lower DDR frequencies"
- Section 6.6 "DDR interface optimization"
- Section 6.7 "Power gating of PHYs"
- Section 6.8 "Distribution of workloads"
- Section 6.9 "DDR access time reduction with SSRAM"
- Section 6.10 "Thermal management to reduce leakage"
- Section 6.11 "Nominal drive mode"

6.1 Run fast and idle

NXP testing and various research have shown that for most customer use cases, the best power/energy management strategy is to run the cores at maximum speeds for the workload and then drop to the lowest power mode as soon as possible. Although this strategy may not provide optimal energy savings for some use cases where data is processed continuously (for example, low latency audio playback), this strategy works for other normal workloads. For each application, the trade-off between the speed and power consumption must be considered to quantify the overall effect on the system power/energy consumption.

You are recommended to keep the i.MX 8ULP processor in the low-power mode as far as possible.

6.2 Clock gating

The Clock Generation and Control (CGC) and Peripheral Clock Controller (PCC) modules inside the i.MX 8ULP provide a programmable method to disable clock sources to modules not being used. Always configure the CGC and PCC registers to apply proper clock gating. It is one of the simplest methods to reduce energy wastage. Driving any unused signal on the SoC or the PCB is simply charging and discharging the line and load capacitance of the signal. By default, the NXP BSP release software implements clock gating.

6.3 PLL reduction

Each PLL block consumes significant energy when active. Each application has unique requirements; however, if possible, reduce the number of active PLLs. The Clock Generation and Control (CGC) module within i.MX 8ULP provides programmable control for clock root selection. It may allow common PLL root clocks within the application and reduce the number of active PLLs when operating.

When entering low-power states (Partial Active / Sleep / Deep-Sleep) or low-power operating states (Audio Playback), reduce the number of active PLLs. It lowers the power requirements for these states. Ensure that the application considers the PLL re-lock time when transitioning back to full operation.

6.4 Core VFS and system bus scaling

Power consumption of the VDD_DIG1 and VDD_DIG2 domains can significantly be reduced by applying voltage and frequency scaling (VFS) to the Arm cores and by scaling (not dynamic) the frequencies of the NOC, AXI, AHB, and IPG system bus clocks. However, the operation of system frequency reduction causes longer access time to the DDR, which may increase energy consumption for some specific use cases. The trade-off between the speed and power consumption must be considered for each mode to quantify the overall effect on system power consumption.

6.5 Lower DDR frequencies

As explained earlier, the DDR I/O bus frequency also contributes to the DDR I/O current. It must be balanced with the system operating requirements. For lower-power operating states, reduction of the DDR bus frequency can provide additional power savings.

6.6 DDR interface optimization

The following are some ways to optimize the DDR interface:

- Employ careful board routing of the DDR memories, maintaining the PCB trace lengths as short as possible. Longer trace lengths and additional vias create additional PCB capacitance for the signal, resulting in more energy wastage along the signal path.
- Keep on-die termination (ODT) values as small as possible. The termination used greatly influences the power consumption of the DDR interface pins. The DDR interface should be simulated to ensure that the ODT variance does not reduce the bus signal integrity.
- Use an appropriate output driver impedance for the DDR interface pins that provides good impedance matching. Select the lowest possible drive strength that provides the required performance to reduce the current flowing through the DDR I/O pins. Remember that simulation should be done to ensure signal integrity.
- The use of the DDR memory offerings in the latest process technology can significantly reduce the power consumption of the DDR devices and the DDR I/O
- Choosing the correct DDR memory size is important. If you select a 4 GB memory when only 2 GB is used, then you waste the refresh current for the unused 2 GB of DDR.
- Choosing the correct sizes for the ECC DDR regions is also important as larger regions use higher energy.

6.7 Power gating of PHYs

The PHYs of unused modules are often overlooked when looking to save power. However, several PHYs contain local PLLs (or clocking circuits) and voltage references that consume power even when not in use. It is applicable to high-speed PHYs, such as Ethernet, MIPI, HDMI, PCIe, and USB.

The i.MX 8ULP processor uses the uPower controller to manage the power gating within the SoC. For more details, see "MicroPower Controller Subsystem (uPower Controller)" chapter in *i.MX 8ULP Processor Reference Manual* (IMX8ULPRM).

6.8 Distribution of workloads

The concept of distributed workloads is to review the system requirements and determine which SoC block is best suited for each task. Ideally, the system returns to the WAIT/STOP state sooner after spreading the workload. It applies to multicore distributions and to functions that might be suited for HiFi4 DSP, ML Engine, or graphics cores.

System designers should ensure that the design uses the optimal cores for the specific workloads or tasks on i.MX 8ULP for maximum efficiency. If the system can return to the low-power state faster (see Section 6.1), significant power can be saved.

6.9 DDR access time reduction with SSRAM

Significant power savings can be achieved by loading the frequently accessed code into the synchronous static random access memory (SSRAM). It reduces current consumption for both the i.MX 8ULP processor and the DDR memory. Another advantage of using the SSRAM is the performance increase, as this code is delayed by the DDR memory access time.

The i.MX BSP ensures that all commonly accessed low-power routines are located in the Arm Trusted Firmware (ATF) of the internal SSRAM.

6.10 Thermal management to reduce leakage

Thermal management is also a key element of power reduction. With an increase in temperature, the SoC gate leakage current increases for each gate within the device. Millions of high-gate leakages add up when looking for the lowest power consumption. As explained earlier, any power savings that can be achieved also reduce the temperature of the SoC and improve the lifetime reliability of the device.

As each system is unique, the system designer should ensure that the operating temperature of the SoC is as low as possible to reduce power loss due to the leakage current. If it cannot be achieved through software control, the designer should include a heat sink or other thermal management methods to remove the heat from the SoC.

See *i.MX 8ULP Hardware Developer’s Guide* (IMX8ULPHDG) for thermal guidelines.

6.11 Nominal drive mode

The NXP supplied Linux BSP GA release configures the system to run in Over Drive mode (ODM), by default. For some specific user applications, this Over Drive mode and the associated performance may not be required. In such cases, you can use Nominal Drive mode (NDM) as defined in the i.MX 8ULP data sheet.

7 Controlling i.MX 8ULP power supplies in lowest power modes

In Standby state, many applications are expected to use the CM33-PD/CA35-PD power mode combinations to minimize power consumption. These combinations are the lowest power combinations on the i.MX 8ULP.

To minimize power consumption, the I/O voltage supplies `VDD_PTA`, `VDD_PTB`, `VDD_PTC`, `VDD_PTD`, `VDD_PTE`, and `VDD_PTF` must remain powered. To avoid leakage current, only `VDD_PTC` and `VDD_PTD` can be optionally turned off (see Table 34).

Table 34 shows the power supply configuration to minimize power consumption in PD modes.

Table 34. Power supply configuration to minimize power consumption in PD modes

Power domain	CM33-PD/CA35-PD
VDD_VBAT42	ON
VDD_FUSE18	ON (internal power gate)
VDDQ_DDR	OFF
VDDQX_DDR	OFF
VDDQX_AO_DDR	ON
VDD_DIG0	ON
VDD_DIG1	ON
VDD_DIG2	ON
VDD_PMC18_DIG0	• ON (CM33 LDO enable mode) • OFF (CM33 LDO bypass mode)
VDD_DDR_PLL	ON
VDD_PMC18	ON
VDD_ANA18	ON
VDD_PTA	ON
VDD_PTB	ON
VDD_PTC	ON (optional OFF)
VDD_PTD	ON (optional OFF)
VDD_PTE	ON

Table 34. Power supply configuration to minimize power consumption in PD modes*...continued*

Power domain	CM33-PD/CA35-PD
VDD_PTF	ON
VDD18_IOREF_1/VDD18_IOREF_2	ON
VDD_DSI18	ON if used
VDD_CSI18	ON if used
VDD_DSI11	ON (internal power gate)
VDD_CSI11	ON if used (internal power gate)
VDD_USB0_33/VDD_USB1_33	ON
VDD_USB0_18/VDD_USB1_18	ON
VDD_ANA33	ON
VREFH_ANA18	ON

7.1 Power distribution using PCA9460 PMIC

The NXP PCA9460 processor is a power management integrated circuit (PMIC) that is designed for use with the i.MX 8ULP processor. For more details on the PCA9460, see the PCA9460 product summary page on the NXP website.

PCA9460 has the following three variants:

- PCA9460A: Targets NXP i.MX 8ULP + LPDDR4
- PCA9460B: Targets NXP i.MX 8ULP + LPDDR4x
- PCA9460C: Targets NXP i.MX 8ULP + LPDDR3

For each PCA9460 variant, the default output of each of BUCK4 and LDO1 is different. For more details, see *PCA9460 Data Sheet*.

Figure 8 shows an example power distribution using the PCA9460A PMIC for the NXP i.MX 8ULP + LPDDR4.

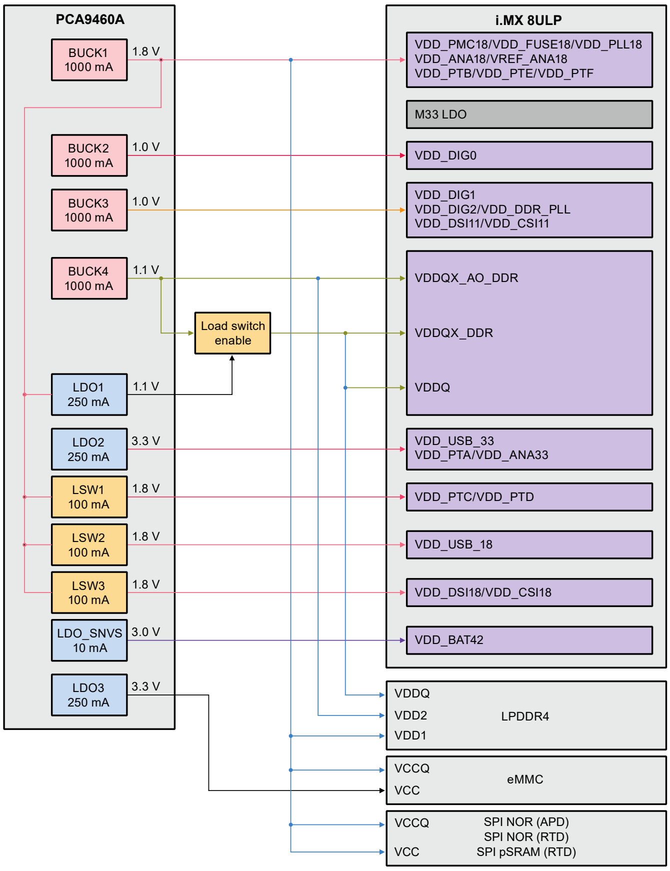


Figure 8. Recommended power distribution with LPDDR4

Figure 9 shows an example power distribution using the PCA9460B PMIC for the NXP i.MX 8ULP + LPDDR4x.

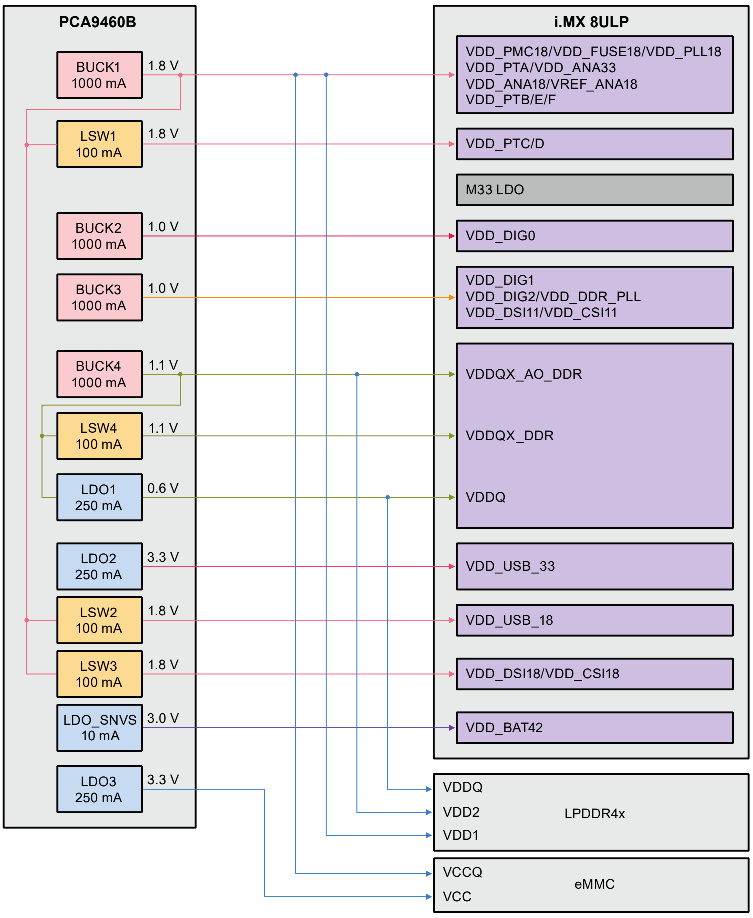


Figure 9. Recommended power distribution with LPDDR4x

8 Important commands

Some important scripts and commands are described below:

- Scripts to configure the environment: Before running a use case, you must configure the environment by running the required configuration scripts, such as `setup.sh`, `setup_default.sh`, `setup_video.sh`, `setup_video_stream.sh`, and `DDRC_96MHz_setup.sh`. Some important configuration scripts are described below:
  - `setup.sh`: This script disables the Ethernet, stops the Weston service, and makes the display blank. It sets the maximum amount of data that the kernel reads ahead for a single file to 512 KB. The script is provided below:

```
#!/bin/bash
systemctl stop weston.service
for i in {0..9}; do
  device="/sys/class/graphics/fb${i}"
  if [ -e "$device" ]; then
    echo 1 > /sys/class/graphics/fb${i}/blank
  fi
done
partitions=`lsblk |awk '$1 !~/-/{print $1}' |grep 'blk\|sd'`
for partition in $partitions; do
   echo 512 > /sys/block/$partition/queue/read_ahead_kb
```

```
done
eth_int=`ifconfig -a | grep 'eth[0-9]'|awk {'print substr($1, 0, 4)'}`
for eth in $eth_int; do
   ifconfig $eth down
done
```

– `setup_default.sh`: This script disables the Ethernet and makes the display blank. It sets the maximum amount of data that the kernel reads ahead for a single file to 512 KB. The script is provided below:

```
#!/bin/bash
for i in {0..9}; do
  device="/sys/class/graphics/fb${i}"
  if [ -e "$device" ]; then
    echo 1 > /sys/class/graphics/fb${i}/blank
  fi
done
partitions=`lsblk |awk '$1 !~/-/{print $1}' |grep 'blk\|sd'`
for partition in $partitions; do
   echo 512 > /sys/block/$partition/queue/read_ahead_kb
done
eth_int=`ifconfig -a | grep 'eth[0-9]'|awk {'print substr($1, 0, 4)'}`
for eth in $eth_int; do
   ifconfig $eth down
done
```

– `setup_video.sh`: This script disables the Ethernet and awakes the display. It sets the maximum amount of data that the kernel reads ahead for a single file to 512 KB. The script is provided below:

```
#!/bin/bash
export WL_EGL_SWAP_INTERVAL=0
partitions=`lsblk |awk '$1 !~/-/{print $1}' |grep 'blk\|sd'`
for partition in $partitions; do
   echo 512 > /sys/block/$partition/queue/read_ahead_kb
done
eth_int=`ifconfig -a | grep 'eth[0-9]'|awk {'print substr($1, 0, 4)'}`
for eth in $eth_int; do
   ifconfig $eth down
done
device="/sys/class/graphics/fb0"
if [ -e "$device" ]; then
  echo 1 > /sys/class/graphics/fb0/blank
  echo 0 > /sys/class/graphics/fb0/blank
fi
```

– `setup_video_stream.sh`: This script opens the Ethernet to play the video online. It awakes the display. It sets the maximum amount of data that the kernel reads ahead for a single file to 512 KB. The script is provided below:

```
#!/bin/bash
export WL_EGL_SWAP_INTERVAL=0
partitions=`lsblk |awk '$1 !~/-/{print $1}' |grep 'blk\|sd'`
for partition in $partitions; do
   echo 512 > /sys/block/$partition/queue/read_ahead_kb
done
eth_int=`ifconfig -a | grep 'eth[0-9]'|awk {'print substr($1, 0, 4)'}`
for eth in $eth_int; do
   ifconfig $eth up
done
device="/sys/class/graphics/fb0"
if [ -e "$device" ]; then
  echo 1 > /sys/class/graphics/fb0/blank
  echo 0 > /sys/class/graphics/fb0/blank
fi
```

- `DDRC_96MHz_setup.sh`: This script disables the Ethernet, stops the Weston service, and makes the display blank. The script is provided below:

```
#!/bin/bash
systemctl stop weston.service
for i in {0..9}; do
  device="/sys/class/graphics/fb${i}"
  if [ -e "$device" ]; then
    echo 1 > /sys/class/graphics/fb${i}/blank
  fi
done
#echo 8 > /proc/sys/kernel/printk;
eth_int=`ifconfig -a | grep 'eth[0-9]'|awk {'print substr($1, 0, 4)'}`
for eth in $eth_int; do
   ifconfig $eth down
done
echo 1 > /sys/devices/platform/imx8ulp-lpm/enable
```

  When you run the script, you see logs indicating that the DDR frequency switches between 528 MHz (High-Bus mode) and 96 MHz (Low-Bus mode). It happens due to the voltage and frequency scaling (VFS) of DDR. DDR VFS is done to save power.

- `dd_write_emmc.sh`: This script is used to run the `dd write` command on eMMC. The script is provided below:

```
#!/bin/bash
emmc_index=$(ls /dev/mmc*rpmb | head -1 | \
awk -F '/dev/mmcblk' '{print $2}' | \
awk -F 'rpmb' '{print $1}')

RC=1
while true; do
   time -p dd if=/dev/zero of=/dev/mmcblk${emmc_index} \
   bs=4096 seek=2621440 count=1024000 conv=fsync  || \
   { echo "TFAIL:dd write emmc fail!";exit $RC; }
done
```

- `dd_read_emmc.sh`: This script is used to run the `dd read` command on eMMC. The script is provided below:

```
#!/bin/bash
emmc_index=$(ls /dev/mmc*rpmb | head -1 | \
awk -F '/dev/mmcblk' '{print $2}' | \
awk -F 'rpmb' '{print $1}')

RC=1
while true; do
   time -p dd if=/dev/mmcblk${emmc_index} \
   of=/dev/null bs=4096 count=102400 || \
   { echo "TFAIL:dd read emmc fail!";exit $RC; }
done
```

- `setup_bt_88w8987.sh` (for the 88W8987 Bluetooth device): This script is used to initialize the host controller interface (HCI) in the kernel. The script is provided below:

```
#!/bin/bash
modprobe moal mod_para=nxp/wifi_mod_para.conf
sleep 5
hciattach /dev/ttyLP2 any 115200 flow
```

```
sleep 1
hciconfig hci0 up
hcitool -i hci0 cmd 0x3f 0x0009 0xc0 0xc6 0x2d 0x00
killall hciattach
hciattach /dev/ttyLP2 any -s 3000000 3000000 flow
sleep 1 hciconfig hci0 up
hciconfig hci0 piscan
hciconfig hci0 noencrypt
hciconfig -a
sleep 1
```

– `setup_bt_iw416.sh` (for the IW416 Bluetooth device): This script is used to initialize the host controller interface (HCI) in the Linux kernel. The script is provided below:

```
#!/bin/bash
modprobe moal mod_para=nxp/wifi_mod_para.conf
modprobe btnxpuart
hciconfig hci0 up
hciconfig hci0 piscan
hciconfig hci0 noencrypt
hciconfig -a
sleep 1
```

– `setup_pulseaudio.sh`: This script is used to set up PulseAudio:

```
#!/bin/sh
pulseaudio --start --log-target=syslog
bluetoothctl power on
bluetoothctl agent on
bluetoothctl default-agent
bluetoothctl scan on # Got MAC address of remote BT device.
```

– `uac2_gadget.sh`: This script is used to create a user account control (UAC) gadget device on the MCIMX8ULP-EVK9 board:

```
#!/bin/bash
modprobe libcomposite

# Create gadget
mkdir /sys/kernel/config/usb_gadget/g1
cd /sys/kernel/config/usb_gadget/g1
echo 0x0200 > bcdUSB
echo 0x0101 > idProduct
echo 0x1d6b > idVendor
mkdir strings/0x409
echo "1234567890" > strings/0x409/serialnumber
echo "Microsoft Applied Sciences" > strings/0x409/manufacturer
echo "Test UAC2 Gadget" > strings/0x409/product

# Create config
mkdir configs/c.1
echo 120 > configs/c.1/MaxPower
mkdir configs/c.1/strings/0x409
echo "Conf 1" > configs/c.1/strings/0x409/configuration

# Create function
mkdir functions/uac2.0
echo 48000 > functions/uac2.0/c_srate
echo 48000 > functions/uac2.0/p_srate
echo 2 > functions/uac2.0/c_ssize
echo 2 > functions/uac2.0/p_ssize
echo 0x3 > functions/uac2.0/c_chmask
echo 0x3 > functions/uac2.0/p_chmask

### config UAC2 parameters here
ln -s functions/uac2.0 configs/c.1

# UDC should correspond to udc name
# Activate
echo ci_hdrc.0 > UDC #8mm 8mn 8ulp
```

  - `MV.sh`: This script starts a machine vision example:

```
#!/bin/bash
gst-launch-1.0 filesrc location=yuv.raw num-buffers=-1 ! \
clocksync ! queue max-size-buffers=2 max-size-time=0 ! \
rawvideoparse format=yuy2 width=640 height=480 framerate=30/1 ! \
tee name=t t. ! queue name=thread-nn max-size-buffers=2 leaky=2 ! \
imxvideoconvert_g2d ! video/x-raw,width=300,height=300,format=RGBA ! \
videoconvert ! video/x-raw,format=RGB ! tensor_converter ! \
tensor_filter framework=tensorflow-lite \
model=ssd_mobilenet_v2_coco_quant_postprocess.tflite \
custom=NumThreads:2 ! tensor_decoder mode=bounding_boxes \
option1=tf-ssd option2=coco_labels.txt option3=0:1:2:3,50 option4=640:480 \
option5=300:300 ! mix. t. ! queue name=thread-img max-size-buffers=2  ! \
imxcompositor_g2d name=mix sink_0::zorder=2 sink_1::zorder=1 ! \
queue name=thread-display max-size-buffers=2 ! waylandsink
```

  - `ML.sh`: This script is used to run the product use cases described under Section 5.8. The script starts a machine learning example:

```
#!/bin/bash
TENSORFLOW_EXAMPLES_DIR=$(find /usr/bin/tensorflow-lite-*/examples | head -
n1)
if [ ! -d "$TENSORFLOW_EXAMPLES_DIR" ]; then
  echo "TFAIL: not found or no dirctory of tensorflow-lite example!"
fi
cd $TENSORFLOW_EXAMPLES_DIR
while true; do
  ./benchmark_model --graph=mobilenet_v1_1.0_224_quant.tflite || break;
done
```

- U-Boot console commands: The following are some frequently used U-Boot console commands:
  - `printenv`: Displays the environment variables.
  - `setenv`: Updates the environment variables:
    - `setenv <name> <value> ...`: Sets the environment variable “name” to “value ...”.
    - `setenv <name>`: Deletes the environment variable “name”.
  - `saveenv`: Saves the updates to the environment variables.
  - `bootargs`: Passes to the kernel what are called kernel command lines.
- Linux OS console commands: The following are some frequently used Linux OS console commands:
  - `cat /proc/cmdline`: Displays the command line.
  - `cat /sys/devices/virtual/thermal/thermal_zone0/temp`: Prints the temperature to the screen (the SoC should be calibrated).

    Note: *The die temperature value was logged (written) externally (not on the SD card) to avoid impacting power consumption.*
  - `cat /sys/kernel/debug/clk/clk_summary`: Prints all clocks to the screen.

- Script to get current CPU usage: Following is a script to get the current CPU usage:

```
#!/bin/bash

# by Paul Colby (http://colby.id.au), no rights reserved

PREV_TOTAL=0
PREV_IDLE=0

While true; do:
# Get the total CPU statistics, discarding the 'cpu ' prefix
CPU=(`sed -n 's/^cpu\s//p' /proc/stat`)
# Get the idle CPU time
IDLE=${CPU[3]}

# Calculate the total CPU time
TOTAL=0 for VALUE in "${CPU[@]}"; do
   let "TOTAL=$TOTAL+$VALUE"
done

# Calculate the CPU usage since we last checked
let "DIFF_IDLE=$IDLE-$PREV_IDLE"
let "DIFF_TOTAL=$TOTAL-$PREV_TOTAL"
let "DIFF_USAGE=(1000*($DIFF_TOTAL-$DIFF_IDLE)/$DIFF_TOTAL)/10"
echo -en "\rCPU: $DIFF_USAGE% \b\b"

# Remember the total and idle CPU times for the next check
PREV_TOTAL="$TOTAL"
PREV_IDLE="$IDLE"

# Wait before checking again
sleep 1 done
```

- Commands to build the cpulimit application: Run these commands to build the `cpulimit` application:
  1. Clone the cpulimit repo:

     ```
     git clone https://github.com/opsengine/cpulimit.git
     ```

  2. Download aarch64 toolchain from https://developer.arm.com/downloads/-/arm-gnu-toolchain-downloads.
  3. Decompress the toolchain package and set up aarch64 toolchain:

     ```
     $ export CROSS_COMPILE=<path to toolchain>/bin/aarch64-linux-gnu-
     $ export CC=${CROSS_COMPILE}gcc
     $ export LD=${CROSS_COMPILE}ld
     ```

  4. Build the application:

     ```
     make clean
     make
     ```

9 Related documentation

Table 35 lists and explains the additional documents and resources that you can refer to for more information on the board. Some of the documents listed below may be available only under a non-disclosure agreement (NDA). To request access to these documents, contact your local field applications engineer (FAE) or sales representative.

Table 35. Related documentation

Document	Description	Link / how to obtain
i.MX 8ULP Processor Reference Manual (IMX8ULPRM)	Provides a detailed description about the i.MX 8ULP processor and its features, including memory maps, power supplies, and clocks.	IMX8ULPRM.pdf
i.MX 8ULP Applications Processor—Industrial Products Data Sheet (IMX8ULPIEC)	Provides information about electrical characteristics, hardware design considerations, and ordering information.	IMX8ULPIEC.pdf
i.MX 8ULP Applications Processor—Commercial Products Data Sheet (IMX8ULPCEC)		IMX8ULPCEC.pdf
i.MX 8ULP Hardware Developer’s Guide (IMX8ULPHDG)	Provides information about board layout recommendations and design checklists to ensure first-pass success and to avoid board bring-up problems. It is intended to help hardware engineers design and test their i.MX 8ULP processor-based designs.	IMX8ULPHDG.pdf
Optimizing Power Consumption for i.MX 8ULP (AN13951)	Describes how to optimize system-level power consumption in different scenarios with i.MX 8ULP.	AN13951.pdf
PCA9460 Data Sheet	Provides details about NXP PCA9460, which is a power management integrated circuit (PMIC) supporting ultra-low-power applications. PCA9460 is designed to be used with the i.MX 8ULP processor, which enables such key applications.	PCA9460.pdf
i.MX Yocto Project User's Guide	Describes how to build an image for an i.MX board by using a Yocto Project build environment. It describes the i.MX release layer and i.MX-specific usage.	IMX_YOCTO_PROJECT_USERS_GUIDE.pdf
Board Control Utilities Release Notes (BCU.pdf)	Provides release information about BCU software.	GitHub

10 Acronyms

Table 36 lists the acronyms used in this document.

Table 36. Acronyms

Acronym	Description
CA35	Arm Cortex-A35 core
CM33	Arm Cortex-M33 core
ADC	Analog-to-digital converter
AHB	Arm AMBA high-performance bus
AOD	Always-ON Display
APD	Application processor domain
ATF	Arm Trusted Firmware
AXI	Arm Advanced eXtensible Interface
BBNSM	Battery-Backed Non-Secure Module

Table 36. Acronyms*...continued*

Acronym	Description
BBSM	Battery-Backed Security Module
BSP	Board support package
CGC	Clock Generation and Control
CMP	i.MX 8ULP analog comparator module
DAC	Digital-to-analog converter
DDR	Dual data rate DRAM
DGO	Designator for always-ON power domain
DRAM	Dynamic random-access memory
EPDC	Electrophoretic Display Controller
EVK	Evaluation kit
fps	Frames per second
GND	Ground
GPIO	General-purpose input/output
GPU	Graphics Processing Unit
GPU2D	Graphics Processing Unit 2D
GPU3D	Graphics Processing Unit 3D
HCI	Host controller interface
HEVC	High Efficiency Video Coding standard
High-Z	High-impedance
I2S	Inter-IC sound bus
I/O	Inputs/output
LDO	Low drop-out regulator
LPAV	Low-power audio/video domain
LPDDR3	Low-power DDR3 SDRAM
LPDDR4	Low-power DDR4 SDRAM with 1.1 V I/O supply
LPDDR4x	Low-power DDR4 SDRAM with 0.6 V I/O supply
LPTMR	i.MX 8ULP Low-Power Timer
MAC	Medium access control
MIPI-CSI	MIPI - Camera Serial Interface controller
MIPI-DSI	MIPI - Display Serial Interface controller
MT/s	Megatransfers per second
ND	Nominal Drive
ODM	Over Drive mode
ODT	On-die termination
PCB	Printed circuit board
PCC	Peripheral Clock Controller

Table 36. Acronyms*...continued*

Acronym	Description
PCM	Pulse-code modulation
PLL	Phase-locked loop clock generator
PMC	Power Management Controller
PMIC	Power management integrated circuit
PMS	Power Mode Switch
PTA	Signals associated with processor port A
PTB	Signals associated with processor port B
PTC	Signals associated with processor port C
PTD	Signals associated with processor port D
PTE	Signals associated with processor port E
PTF	Signals associated with processor port F
RAM	Random access memory
RTC	Real-time clock
RTD	Real-time processor domain
SDK	Software Development Kit
SoC	System-on-Chip
SRAM	On-chip static random access memory
SRTC	Secure real-time clock
SSRAM	Synchronous static random access memory
UAC	User account control
USB	Universal Serial Bus
VFS	Voltage and frequency scaling
WUU	Wake-Up Unit

11 Note about the source code in the document

Example code shown in this document has the following copyright and BSD-3-Clause license:

INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

12 Revision history

Table 37 summarizes the revisions to this document.

Table 37. Revision history

Document ID	Release date	Description
AN14486 v.1.0	24 December 2024	Initial public 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.

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 document 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.

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.

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 and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) 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.

Bluetooth — the Bluetooth wordmark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by NXP Semiconductors is under license.

eIQ — is a trademark of NXP B.V.

Contents

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