AN13914

i.MX 8ULP Power Consumption Measurement

Rev. 2 — 21 September 2023 Application note

Document information

Information	Content
Keywords	AN13914, i.MX 8ULP, MCIMX8ULP-EVK, 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 an NXP i.MX 8ULP EVK 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 an NXP i.MX 8ULP EVK 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.*

2 i.MX 8ULP architecture

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

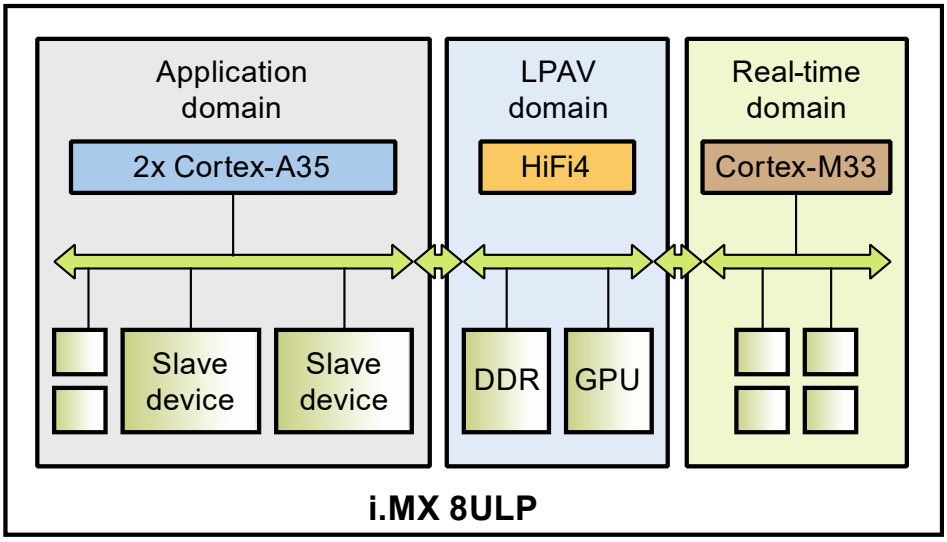

Figure 1. i.MX 8ULP architecture

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.

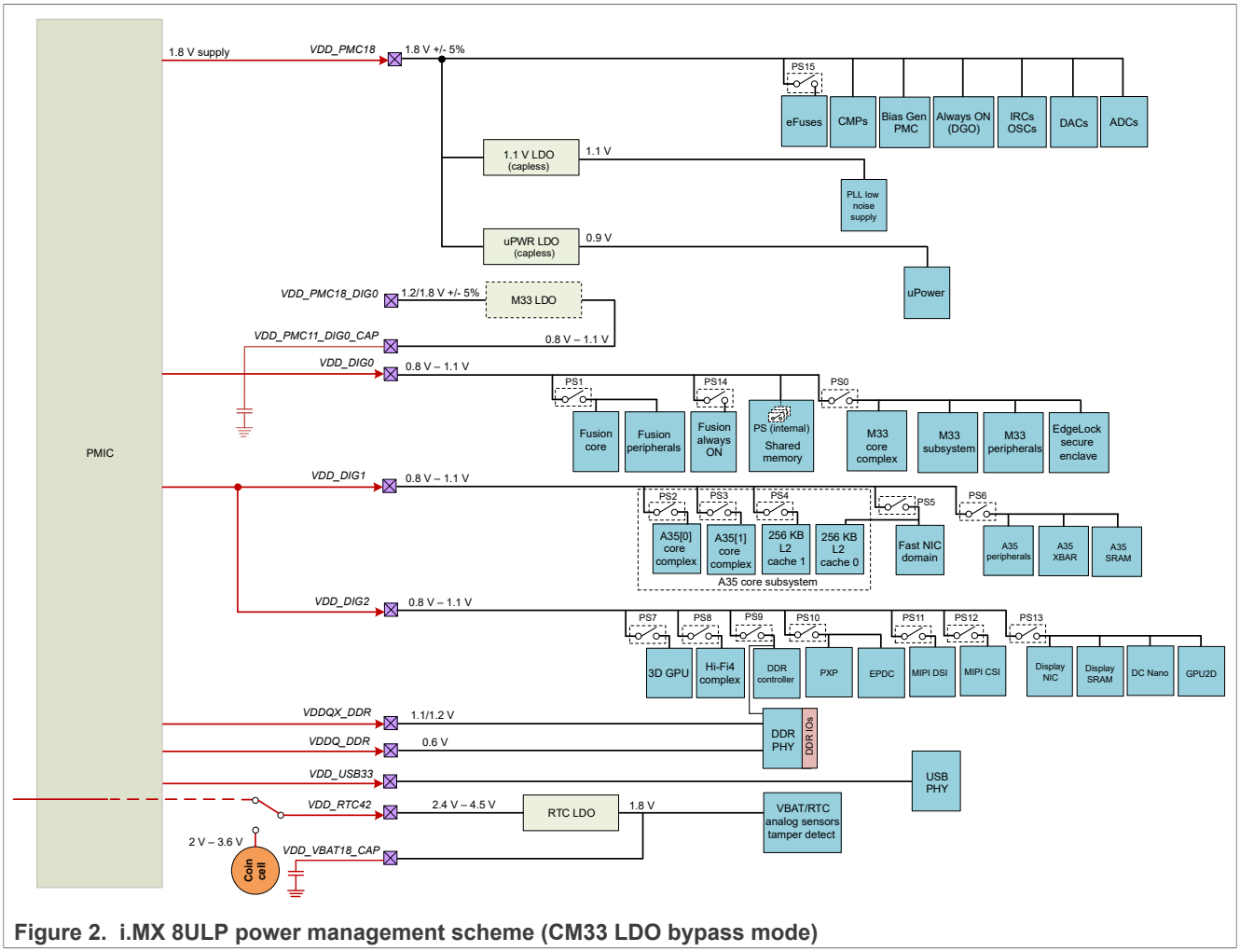

Figure 2. i.MX 8ULP power management scheme (CM33 LDO bypass mode)

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 recommended operating conditions and detailed description of the groups of pins powered by each I/O voltage supply, see i.MX 8ULP data sheet.*

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

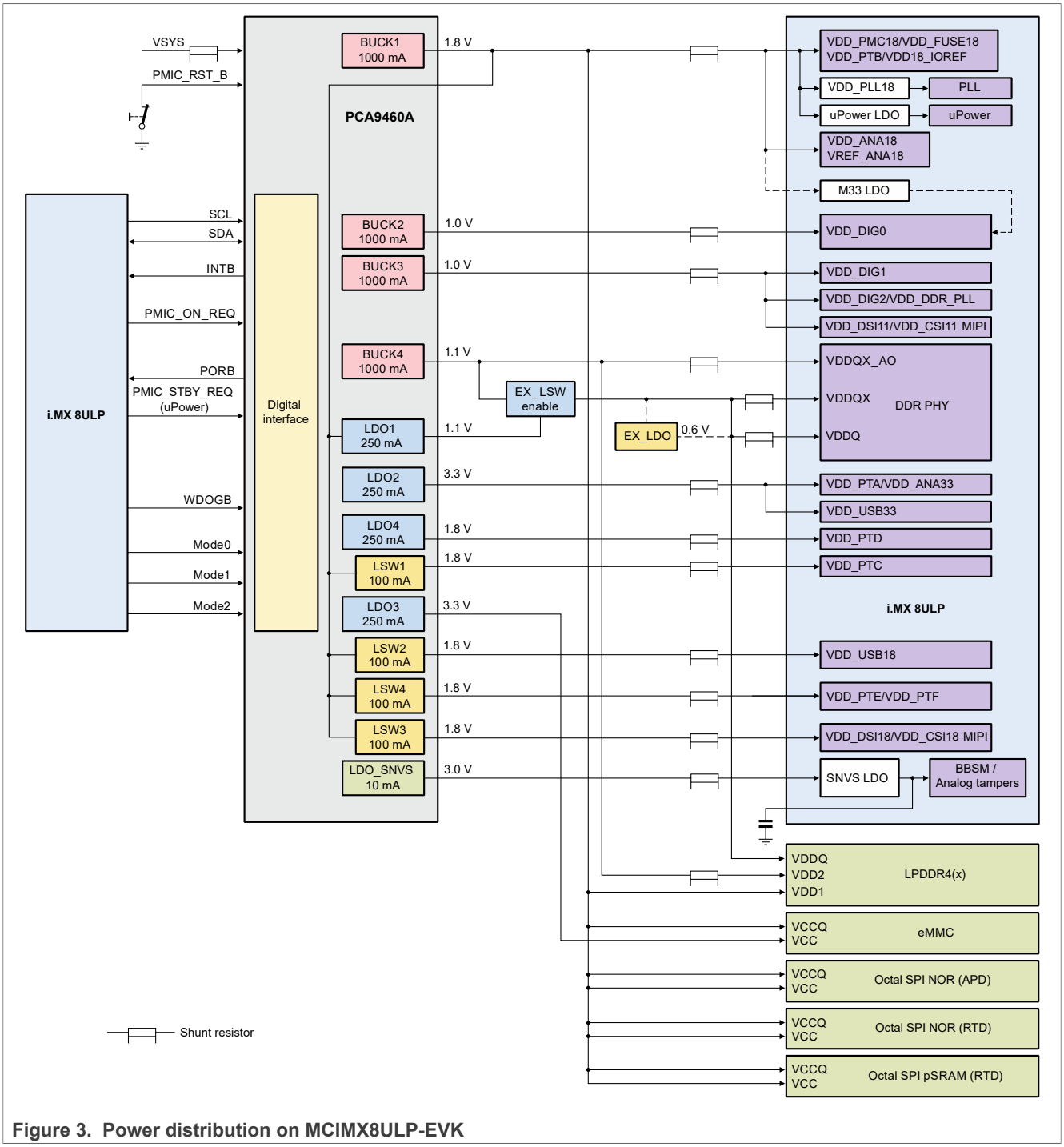

Figure 3. Power distribution on MCIMX8ULP-EVK

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

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

The measurements are taken mainly for the power supply domains given in Table 2. This 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. The MCIMX8ULP-EVK board supports CM33 LDO bypass mode, by default.
- 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-EVK 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, and it 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 as well as 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, VDD_DIG0, VDD_DIG1, and VDD_DIG2 can be changed to implement voltage and frequency scaling (VFS) during the run time of the use cases. 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	NXP MCIMX8ULP-EVK, 76882_A6 + 76883_B3 with A2 silicon, part number is PIMX8UD5 CVP08SC
System software	Linux BSP 6.1.22_2.0.0
Application software	SDK 2.13.0
	BCU 1.1.52. 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. Run the following commands to 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-mickledore
 -m imx-6.1.22-2.0.0.xml
repo sync
DISTRO=fsl-imx-xwayland MACHINE=imx8ulp-lpddr4-evk source imx-setup-
release.sh -b build-imx8ulp-evk
```

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

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

3. Run this command to build the image:

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

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

4.11 Power consumption measurement steps

Following are the steps needed to measure i.MX 8ULP power consumption on the MCIMX8ULP-EVK board:

1. Connect a micro-USB cable between the host PC and the USB micro-B port J17 on the MCIMX8ULP-EVK board.
2. Run the following command to start the monitor in the BCU path:

```
bcu.exe monitor -board=imx8ulpevkb2
```

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 measurement precision: mA/auto/µA.
6. Wait for 1 minute, and record the measurement data in BCU.

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.*
- *For all use cases (except "Audio low-bus playback" use case), the platform is booted from eMMC with the default DTB configuration (`imx8ulp-evk.dtb`) at the U-Boot stage.*
- *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-EVK board.

Table 4. MCIMX8ULP-EVK 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	54.01	mW
	Suspend mode – RTD sleep	47.43	mW
	Suspend mode – RTD deep sleep	787.00	µW
	Suspend mode – RTD power down	145.68	µW
	System idle with screen ON	448.03	mW
	System idle with screen OFF	196.30	mW
Audio/video playback use cases	Audio playback (gplay) with full frequency	427.93	mW
	Audio low-bus playback (gplay) with DDR clock at 96 MHz	243.11	mW
	Audio playback on Bluetooth (gplay)	469.04	mW
	Video playback local (gplay)	519.28	mW
	Video playback streaming (gplay)	485.77	mW
Core benchmark use cases	Dhrystone	552.62	mW
	CoreMark	517.94	mW
GPU use cases	MM07	613.40	mW
	MM06	701.94	mW
	GLMark	644.78	mW
Heavy load use cases	2-core Dhrystone + MM07 + HiFi4 + CM33 CoreMark	737.86	mW
	2-core Stream + MM07 + HiFi4 + CM33 CoreMark	687.75	mW
	GPU 2D + GPU 3D + CSI/DSI	706.73	mW
Memory use cases	memset	700.70	mW
	memcpy	592.26	mW
	Stream	607.87	mW
Storage (eMMC) use cases	DD_RD_eMMC	536.83	mW

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

Use case category	Use case	Total power (sum of average powers)	
		Value	Unit
	DD_WRT_eMMC	451.72	mW
Product use cases	Parallel E-Ink page flip	489.89	mW
	Parallel E-Ink partial screen update	495.82	mW
	Machine vision	635.65	mW
	eIQ benchmark	546.00	mW
	UAC audio playback	426.48	mW
	Always-ON display, 1 fps fresh rate	92.45	mW
System-level power estimation use cases	Battery	8.75	µW
	256 KB L2 cache	520.04	mW

5.1 Low-power mode use cases

The following use case scenarios were tested:

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

5.1.1 Suspend mode – RTD active

For this use case, Linux is in Suspend mode and RTD is set to Active mode.

Following are the required settings for this use case:

- APD is in Power-Down mode
- LPAV is in Power-Down mode with DDR in Self-Refresh mode
- RTD is in Active mode, running power mode switch demo
- All clocks and PLLs in APD and LPAV are turned off

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

1. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
2. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`).
3. Run this command to put the Linux 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	Supply domain	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	9.20	16.6	54.01	Die temperature was not measured as the CA35 core was suspended.
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	0.00	0.0		
	BUCK1_LSW3_CPU_1V8	1.80	0.00	0.0		
	BUCK1_LSW4_CPU_1V8	1.80	0.00	0.0		
	BUCK2_CPU_1V0	1.00	33.30	33.3		
	BUCK3_CPU_1V0	0.74	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	3.30	1.20	3.9		
	LDO4_CPU_1V8	1.79	0.00	0.0		
	LDO5_CPU_3V0	2.91	0.00	0.0		

5.1.2 Suspend mode – RTD sleep

For this use case, Linux is in Suspend mode and RTD is set to Sleep mode.

Following are the required settings for this use case:

- 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. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
2. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`).
3. Run this command to put the Linux 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	Supply domain	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	9.30	16.7	47.43	Die temperature was not measured as the CA35 core was suspended.
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	0.00	0.0		
	BUCK1_LSW3_CPU_1V8	1.80	0.00	0.0		

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

Power group	Supply domain	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
	BUCK1_LSW4_CPU_1V8	1.80	0.00	0.0		
	BUCK2_CPU_1V0	1.00	26.70	26.7		
	BUCK3_CPU_1V0	0.74	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	3.30	1.20	3.9		
	LDO4_CPU_1V8	1.79	0.00	0.0		
	LDO5_CPU_3V0	2.91	0.00	0.0		

5.1.3 Suspend mode – RTD deep sleep

For this use case, Linux is in Suspend mode and RTD is set to Deep-Sleep mode.

Following are the required settings for this use case:

- 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. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
2. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`).
3. Run this command to put the Linux 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	Supply domain	Average voltage (V)	Average current (µA)	Average power (µW)	Sum of average powers (µW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	BUCK1_CPU_1V8	1.80	17.47	31.46	787	Die temperature was not measured as the CA35 core was suspended.
	BUCK1_LSW1_CPU_1V8	1.80	6.44	11.62		
	BUCK1_LSW2_CPU_1V8	1.80	9.71	11.51		
	BUCK1_LSW3_CPU_1V8	1.80	0.77	1.38		
	BUCK1_LSW4_CPU_1V8	1.80	6.04	10.90		
	BUCK2_CPU_1V0	0.75	927.76	697.75		
	BUCK3_CPU_1V0	0.74	-7.96	-5.86		
	BUCK4_CPU_1V1	1.10	-1.32	-1.45		
	LDO1_CPU_1V1	0.00	0.00	0.00		

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

Power group	Supply domain	Average voltage (V)	Average current (µA)	Average power (µW)	Sum of average powers (µW)	Zone 0 die temperature (℃)
	LDO1_CPU_1V1_0V6	0.00	0.00	0.00		
	LDO2_CPU_3V3	3.30	1.48	4.89		
	LDO4_CPU_1V8	1.79	5.49	9.81		
	LDO5_CPU_3V0	2.94	3.00	8.81		

Note: *For the following three use cases, the unit of power is µW:*

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

5.1.4 Suspend mode – RTD power down

For this use case, Linux is in Suspend mode and RTD is set to Power-Down mode.

Following are the required settings for this use case:

- APD is in Power-Down mode (Linux enter suspend)
- LPAV is in Power-Down mode with DDR in Self-Refresh mode
- RTD is in Power-Down mode (CM33 enters power down with SW8 button on the EVK board as wake-up source)
- All clocks and PLLs in APD and LPAV are turned off

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

1. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
2. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`).
3. Run this command to put the system 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	Supply domain	Average voltage (V)	Average current (µA)	Average power (µW)	Sum of average powers (µW)	Zone 0 die temperature (℃)
GROUP_SOC_FULL	BUCK1_CPU_1V8	1.80	64.00	116.00	145.68	Die temperature was not measured as the CA35 core was suspended.
	BUCK1_LSW1_CPU_1V8	0.26	-1.15	-0.29		
	BUCK1_LSW2_CPU_1V8	1.81	10.24	18.49		
	BUCK1_LSW3_CPU_1V8	1.81	-0.09	-1.17		
	BUCK1_LSW4_CPU_1V8	1.81	3.11	5.61		
	BUCK2_CPU_1V0	0.73	-19.00	-13.79		
	BUCK3_CPU_1V0	0.74	-5.49	-4.04		
	BUCK4_CPU_1V1	1.10	0.00	0.00		
	LDO1_CPU_1V1	0.00	0.00	0.00		

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

Power group	Supply domain	Average voltage (V)	Average current (µA)	Average power (µW)	Sum of average powers (µW)	Zone 0 die temperature (℃)
	LDO1_CPU_1V1_0V6	0.00	0.00	0.00		
	LDO2_CPU_3V3	3.30	1.80	5.95		
	LDO4_CPU_1V8	1.79	5.67	10.12		
	LDO5_CPU_3V0	2.94	2.97	8.71		

Note: *For the following three use cases, the unit of power is µW:*

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

5.1.5 System idle with screen ON

For this use case, a 720p TV display was connected to the MCIMX8ULP-EVK board through an HDMI interface.

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 clocks and PLLs in APD and LPAV are turned off

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

1. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
2. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`).
3. Run `setup_default.sh` to put the Linux into Idle mode (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	Supply domain	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	17.00	30.6	448.03	33
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.4		
	BUCK1_LSW3_CPU_1V8	1.80	1.80	3.3		
	BUCK1_LSW4_CPU_1V8	1.80	0.90	1.6		
	BUCK2_CPU_1V0	1.00	27.20	27.2		
	BUCK3_CPU_1V0	1.09	251.00	274.6		
	BUCK4_CPU_1V1	1.08	11.90	12.8		
	LDO1_CPU_1V1	1.09	3.70	4.0		
	LDO1_CPU_1V1_0V6	1.08	74.50	80.7		
	LDO2_CPU_3V3	3.30	1.20	3.9		

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

Power group	Supply domain	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
	LDO4_CPU_1V8	1.79	0.39	0.7		
	LDO5_CPU_3V0	2.91	0.00	0.0		

5.1.6 System idle with screen OFF

For this use case, a 720p TV display was connected to the MCIMX8ULP-EVK board through the HDMI interface.

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 96 MHz (low frequency)
- Both CA35 cores are power gated if the kernel is in the lowest level of idle
- The CM33 core (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. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
2. Boot the Linux image with the DTB configuration `imx8ulp-evk-nd.dtb`.
3. Run `DDRC_96MHz_setup.sh` to put the Linux into Idle mode. For more details, 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	Supply domain	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	13.60	24.5	196.3	30
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.4		
	BUCK1_LSW3_CPU_1V8	1.80	0.00	0.0		
	BUCK1_LSW4_CPU_1V8	1.80	0.72	1.3		
	BUCK2_CPU_1V0	1.00	27.10	27.1		
	BUCK3_CPU_1V0	1.00	82.10	81.9		
	BUCK4_CPU_1V1	1.08	11.10	12.0		
	LDO1_CPU_1V1	1.10	0.64	0.7		
	LDO1_CPU_1V1_0V6	1.09	32.60	35.7		
	LDO2_CPU_3V3	3.30	1.20	3.9		
	LDO4_CPU_1V8	1.79	0.39	0.7		
	LDO5_CPU_3V0	2.91	0.00	0.0		

Note: *The BUCK4 (VDDQX_AO_DDR) power is higher than expected due to the reason that 120 Ω resistor to DDRC DDR PHY TXPD was enabled in the current board support package (BSP) release, which is not required.*

5.2 Audio/video playback use cases

The following use case scenarios were tested:

- 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

The audio file used was an MP3 file with a 128 kbit/s bit rate and a 44 kHz sample rate per second, played using the following options:

```
gplay-1.0 $audio_file
```

The CA35 cores handle the MP3 audio decoding, and the CM33 core handles the PCM playback through the I2S interface.

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 Active mode
- All clocks and PLLs in APD and LPAV are turned off

Run the use case in a loop and start logging power measurement results at the desired time interval (recommended interval is 1 minute).

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

1. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
2. Boot the Linux image with the default DTB configuration (`imx8ulp-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
```

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	Supply domain	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	16.70	30.1	427.93	32
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.4		

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

Power group	Supply domain	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
	BUCK1_LSW3_CPU_1V8	1.80	0.00	0.0		
	BUCK1_LSW4_CPU_1V8	1.80	1.00	1.8		
	BUCK2_CPU_1V0	1.00	33.90	33.8		
	BUCK3_CPU_1V0	1.09	253.10	276.9		
	BUCK4_CPU_1V1	1.08	11.80	12.7		
	LDO1_CPU_1V1	1.09	3.70	4.0		
	LDO1_CPU_1V1_0V6	1.09	50.20	54.7		
	LDO2_CPU_3V3	3.30	1.40	4.5		
	LDO4_CPU_1V8	1.79	0.50	0.9		
	LDO5_CPU_3V0	2.91	0.00	0.0		

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

For this use case, the audio file used was an MP3 file with a 128 kbit/s bit rate and a 44 kHz sample rate per second, played using the following command:

```
gplay-1.0 $audio_file
```

The CA35 cores handle the MP3 audio decoding, and CM33 core handles the PCM playback through the I2S interface.

Following are the required settings for this use case:

- The CA35 core frequency is set to 500 MHz
- The DDR frequency is set to 96 MHz
- Both CA35 cores are power gated if the kernel is in the lowest level of idle
- The CM33 core is in active status
- All clocks and PLLs in APD and LPAV are turned off

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

1. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
2. Boot the Linux image 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. It can put the system into ND mode with BUCK3 at lower voltage 1.0 V for more reduction in power supply 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: *You have to prepare your own MP3 file. To obtain results similar to the results provided in this document, ensure that audio bit rate is about 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	Supply domain	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	13.50	24.3	243.11	32
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.60	8.3		
	BUCK1_LSW3_CPU_1V8	1.80	0.00	0.0		
	BUCK1_LSW4_CPU_1V8	1.80	1.10	1.9		
	BUCK2_CPU_1V0	0.99	33.70	33.5		
	BUCK3_CPU_1V0	1.00	103.90	103.8		
	BUCK4_CPU_1V1	1.08	12.20	13.2		
	LDO1_CPU_1V1	1.09	0.73	0.8		
	LDO1_CPU_1V1_0V6	1.09	47.80	52.0		
	LDO2_CPU_3V3	3.29	1.40	4.5		
	LDO4_CPU_1V8	1.80	0.39	0.7		
	LDO5_CPU_3V0	2.92	0.00	0.0		

Note: *The BUCK4 (VDDQX_AO_DDR) power is higher than expected due to the reason that 120 Ω resistor to DDRC DDR PHY TXPD was enabled accidentally in the current board support package (BSP) release. This change will be reverted in the next BSP release.*

5.2.3 Audio playback on Bluetooth (gplay)

For this use case, the audio file used was an MP3 file with a 128 kbit/s bit rate and a 44 kHz sample rate per second, played using the following command:

```
gplay-1.0 $audio_file
```

The HiFi4 DSP handles the MP3 audio decoding, and the CM33 core handles the PCM playback through the I2S interface.

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
- The HiFi4 DSP is in Active mode
- All clocks and PLLs in APD and LPAV are turned off

Run the use case in a loop and start logging power measurement results at the desired time interval (recommended interval is 1 minute).

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

1. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
2. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`).
3. Put RTD into Active mode in PMS demo.
4. Run `setup.sh` (see Section 8 for more details).
5. Run `setup_bt_88w8987.sh` or `setup_bt_iw416.sh` (depending on the Bluetooth device attached to the board). For more information, see Section 8.

6. Run `setup_pulseaudio.sh` (see Section 8 for more details).
7. Run `gplay_audio.sh`:

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

8. 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	Supply domain	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	17.00	30.6	469.04	38
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.60	8.3		
	BUCK1_LSW3_CPU_1V8	1.80	0.00	0.0		
	BUCK1_LSW4_CPU_1V8	1.79	11.70	20.9		
	BUCK2_CPU_1V0	0.99	34.20	34.0		
	BUCK3_CPU_1V0	1.10	268.60	294.6		
	BUCK4_CPU_1V1	1.08	12.00	13.0		
	LDO1_CPU_1V1	1.09	3.70	4.0		
	LDO1_CPU_1V1_0V6	1.09	53.50	58.2		
	LDO2_CPU_3V3	3.29	1.30	4.4		
	LDO4_CPU_1V8	1.80	0.50	0.9		
	LDO5_CPU_3V0	2.92	0.00	0.0		

5.2.4 Video playback local (gplay)

For this use case, a 720p TV display was connected to the MCIMX8ULP-EVK board through the HDMI interface.

The video file used for playback was an MKV file compressed using the HEVC standard with full HD resolution at 29.97 fps. The audio encoding was AACL with a 44.1 kHz sample rate in a 2-channel configuration. The CA35 cores handle audio and video decoding, and the CM33 core handles audio PCM playback.

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

The video file was locally played using gplay with the following options:

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

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
- All clocks and PLLs in APD and LPAV are turned off

Run the use case in a loop and start logging power measurement results at the desired time interval (recommended interval is 1 minute).

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

1. Connect the display to the board through the HDMI interface.
2. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
3. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`).
4. Run `setup_video.sh` (see Section 8 for more details).
5. Put RTD into Active mode in PMS demo.
6. Run `gplay_videoplayback.sh`:

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

Note: *You have to prepare your own MP4 file. To obtain results similar to the results provided in this document, ensure that the resolution is 480p with 24 frame rate, video bit rate is about 1200 kbit/s, and encoding used is 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	Supply domain	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	17.00	30.5	519.28	34
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.4		
	BUCK1_LSW3_CPU_1V8	1.80	1.80	3.3		
	BUCK1_LSW4_CPU_1V8	1.80	0.90	1.6		
	BUCK2_CPU_1V0	1.00	34.00	33.9		
	BUCK3_CPU_1V0	1.09	306.70	335.2		
	BUCK4_CPU_1V1	1.07	12.50	13.5		
	LDO1_CPU_1V1	1.09	3.80	4.1		
	LDO1_CPU_1V1_0V6	1.08	77.50	83.9		
	LDO2_CPU_3V3	3.30	1.20	3.9		
	LDO4_CPU_1V8	1.79	0.45	0.8		
	LDO5_CPU_3V0	2.91	0.00	0.0		

5.2.5 Video playback streaming (gplay)

For this use case, a 720p TV display was connected to the MCIMX8ULP-EVK board through the HDMI interface.

The video file used for playback was an MKV file compressed using the HEVC standard with full HD resolution at 29.97 fps. The audio encoding was AACL with a 44.1 kHz sample rate in a 2-channel configuration. The CA35 cores handle audio and video decoding, and the CM33 core handles audio PCM playback.

A server was set up to host the MKV video file for streaming.

The video streaming was done using an Ethernet adapter and the player used was gplay:

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

Following are the required settings for this use case:

- The CPU frequency is set to the maximum value of 800 MHz

- The CM33 core is in Active mode
- All clocks and PLLs in APD and LPAV are turned off

Run the use case in a loop and start logging power measurement results at the desired time interval (recommended interval is 1 minute).

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

1. Prepare two boards and connect them to the same local area network.
2. Place the `480p24.mp4` file under `/home/root` in the first board and use `ifconfig` to obtain the IP address of the first board (`<ip_server>`).
3. Boot the Linux image with default DTB configuration (`imx8ulp-evk.dtb`) and boot Power Mode Switch (PMS) demo from RTD console menu on both boards.
4. Connect the HDMI display and run `setup_video_stream.sh` on the second board.
5. Put RTD into Active mode in PMS demo.
6. Run the following command on the second board:

```
gplay-1.0 http://<ip_server>/480p24.mp4
```

7. Measure the second board's power 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	Supply domain	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	17.10	30.8	485.77	34
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.4		
	BUCK1_LSW3_CPU_1V8	1.80	1.80	3.3		
	BUCK1_LSW4_CPU_1V8	1.80	1.00	1.7		
	BUCK2_CPU_1V0	1.00	34.00	34.0		
	BUCK3_CPU_1V0	1.09	276.30	302.2		
	BUCK4_CPU_1V1	1.08	12.30	13.2		
	LDO1_CPU_1V1	1.09	3.70	4.1		
	LDO1_CPU_1V1_0V6	1.08	76.30	82.6		
	LDO2_CPU_3V3	3.30	1.40	4.5		
	LDO4_CPU_1V8	1.79	0.45	0.8		
	LDO5_CPU_3V0	2.91	0.00	0.0		

5.3 Core benchmark use cases

The following use case scenarios were tested 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 performed by both CA35 cores (because Dhrystone is a single thread benchmark, two instances were started). Both the CA35 cores run the test in a loop at a frequency of 800 MHz.

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 clocks and PLLs in APD and LPAV are turned off

Run the use case in a loop and start logging power measurement results at the desired time interval (recommended interval is 1 minute).

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

1. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
2. Boot the Linux image, and boot the board to the eMMC rootfs.
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 [ "1" == "1" ]; 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	Supply domain	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	16.70	30.1	552.62	35
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.4		
	BUCK1_LSW3_CPU_1V8	1.80	0.00	0.0		
	BUCK1_LSW4_CPU_1V8	1.80	0.83	1.5		
	BUCK2_CPU_1V0	1.00	27.20	27.2		
	BUCK3_CPU_1V0	1.09	387.20	422.6		
	BUCK4_CPU_1V1	1.08	11.40	12.3		
	LDO1_CPU_1V1	1.09	3.60	3.9		
	LDO1_CPU_1V1_0V6	1.09	38.20	41.8		
	LDO2_CPU_3V3	3.30	1.20	3.9		
	LDO4_CPU_1V8	1.79	0.45	0.8		

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

Power group	Supply domain	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
	LDO5_CPU_3V0	2.91	0.00	0.0		

5.3.2 CoreMark

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

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

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 clocks and PLLs in APD and LPAV are turned off

Run the use case in a loop and start logging power measurement results at the desired time interval (recommended interval is 1 minute).

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

1. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
2. Boot the Linux image, and boot the board to the eMMC rootfs.
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. Compile `coremark` to get better performance (`coremark` is built for dual-core for i.MX 8ULP):

```
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 CoreMark benchmark running on the four CPU cores.

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

Power group	Supply domain	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	16.80	30.1	517.94	34
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.4		
	BUCK1_LSW3_CPU_1V8	1.80	0.00	0.0		
	BUCK1_LSW4_CPU_1V8	1.80	0.90	1.6		

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

Power group	Supply domain	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
	BUCK2_CPU_1V0	1.00	27.40	27.3		
	BUCK3_CPU_1V0	1.09	354.70	387.4		
	BUCK4_CPU_1V1	1.08	11.40	12.3		
	LDO1_CPU_1V1	1.09	3.60	3.9		
	LDO1_CPU_1V1_0V6	1.09	38.30	41.9		
	LDO2_CPU_3V3	3.30	1.20	3.9		
	LDO4_CPU_1V8	1.79	0.45	0.8		
	LDO5_CPU_3V0	2.91	0.00	0.0		

5.4 GPU use cases

The following GPU use case scenarios were tested:

- MM07
- MM06
- GLMark

5.4.1 MM07

MM07 is a 3D-gaming benchmark. The graphics were loaded from the eMMC into the DDR memory, were processed by GPU3D, and were copied into a display buffer in the DDR memory. It was displayed on a 720p TV display (through the HDMI interface).

Following are the required settings for this 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 clocks and PLLs in APD and LPAV are turned off

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

1. Connect the display to the board through the HMDI interface.
2. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
3. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`) and boot the board to the eMMC rootfs.
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_mm07.sh`:

```
export WL_EGL_SWAP_INTERVAL=0
cd mm07/
while true; do
   ./fm_oes2_mobile_player
done
```

7. Start 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 "MM07" use case

Power group	Supply domain	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	17.20	31.0	613.4	34
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.4		
	BUCK1_LSW3_CPU_1V8	1.80	1.80	3.3		
	BUCK1_LSW4_CPU_1V8	1.80	0.78	1.4		
	BUCK2_CPU_1V0	1.00	27.70	27.7		
	BUCK3_CPU_1V0	1.09	382.60	417.6		
	BUCK4_CPU_1V1	1.07	13.50	14.4		
	LDO1_CPU_1V1	1.08	4.30	4.6		
	LDO1_CPU_1V1_0V6	1.08	92.70	99.7		
	LDO2_CPU_3V3	3.30	1.20	3.9		
	LDO4_CPU_1V8	1.79	0.61	1.1		
	LDO5_CPU_3V0	2.91	0.00	0.0		

5.4.2 MM06

MM06 is a 3D-gaming benchmark. The graphics were loaded from the eMMC into the DDR memory, were processed by GPU3D, and were copied into a display buffer in the DDR memory. It was displayed on a 720p TV display (through the HDMI interface).

Following are the required settings for this 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 clocks and PLLs in APD and LPAV are turned off

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

1. Connect the display to the board through the HMDI interface.
2. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
3. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`) and boot the board to the eMMC rootfs.
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_mm06.sh`:

```
export WL_EGL_SWAP_INTERVAL=0
cd mm06/
while true; do
   ./fm_oes_player
done
```

7. Start 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 "MM06" use case

Power group	Supply domain	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	17.30	31.1	701.94	38
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.4		
	BUCK1_LSW3_CPU_1V8	1.80	1.80	3.3		
	BUCK1_LSW4_CPU_1V8	1.80	0.83	1.5		
	BUCK2_CPU_1V0	1.00	27.70	27.7		
	BUCK3_CPU_1V0	1.09	434.80	474.1		
	BUCK4_CPU_1V1	1.07	15.40	16.5		
	LDO1_CPU_1V1	1.08	4.60	4.9		
	LDO1_CPU_1V1_0V6	1.07	121.40	129.5		
	LDO2_CPU_3V3	3.30	1.20	3.9		
	LDO4_CPU_1V8	1.79	0.45	0.8		
	LDO5_CPU_3V0	2.91	0.00	0.0		

5.4.3 GLMark

Following are the required settings for 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 clocks and PLLs in APD and LPAV are turned off

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

1. Connect the display to the board through the HMDI interface.
2. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
3. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`) and boot the board to the eMMC rootfs.
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`:

```
while true; do
   glmark2-es2-wayland -fullscreen
done
```

7. Start 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 "GLMark" use case

Power group	Supply domain	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	17.30	31.0	644.78	36

Table 20. Power measurement results for "GLMark" use case*...continued*

Power group	Supply domain	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.4		
	BUCK1_LSW3_CPU_1V8	1.80	1.80	3.3		
	BUCK1_LSW4_CPU_1V8	1.80	0.83	1.5		
	BUCK2_CPU_1V0	1.00	27.50	27.5		
	BUCK3_CPU_1V0	1.09	399.30	435.6		
	BUCK4_CPU_1V1	1.07	15.00	16.0		
	LDO1_CPU_1V1	1.08	4.40	4.8		
	LDO1_CPU_1V1_0V6	1.07	104.20	111.8		
	LDO2_CPU_3V3	3.30	1.20	3.9		
	LDO4_CPU_1V8	1.79	0.45	0.8		
	LDO5_CPU_3V0	2.91	0.00	0.0		

5.5 Heavy load use cases

The following heavy load use case scenarios were tested:

- 2-core Dhrystone + MM07 + HiFi4 + CM33 CoreMark
- 2-core Stream + MM07 + HiFi4 + CM33 CoreMark
- GPU 2D + GPU 3D + CSI/DSI

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

5.5.1 2-core Dhrystone + MM07 + HiFi4 + CM33 CoreMark

This use case runs 2-core Dhrystone, GPU MM07, HiFi4, and CM33 CoreMark in parallel. A 720p TV display was connected to the MCIMX8ULP-EVK board through the HDMI interface.

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 Math Power busy loop stress test
- The GPU is running MM07 stress test
- Both CA35 cores are running Dhrystone benchmark
- The CM33 core is running the CoreMark benchmark
- All clocks and PLLs in APD and LPAV are turned off

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

1. Connect the display to the board through the HMDI interface.
2. Flash `flash_pms_coremark.bin` to eMMC:

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

3. Boot Power Mode Switch (PMS) with CoreMark demo from the menu on the RTD console.

4. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`) and boot the board to the eMMC rootfs.
5. Run `setup_video.sh` (see Section 8 for more details).
6. Start two Dhrystone instances, each bound on separate CPU:

```
while [ "1" == "1" ]; do
   taskset -c 0 ./dhry2 &
   taskset -c 1 ./dhry2
done
```

7. Copy `hifi4_imx8ulp-MathPower_busy_loop.bin` to `/lib/firmware/imx/dsp/` and rename it as `hifi4.bin`.
8. Run commands to load and run the firmware:

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

where <n> is HiFi4 index in remote processors.

9. Run CoreMark benchmark from the menu on the RTD console.
10. Start 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 "Dhrystone + MM07 + HiFi4 + CM33 CoreMark" use case

Power group	Supply domain	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	17.30	31.1	737.86	38
	BUCK1_LSW1_CPU_1V8	1.80	0.39	0.7		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.5		
	BUCK1_LSW3_CPU_1V8	1.80	1.80	3.3		
	BUCK1_LSW4_CPU_1V8	1.80	1.00	1.9		
	BUCK2_CPU_1V0	1.00	33.40	33.4		
	BUCK3_CPU_1V0	1.09	485.10	528.5		
	BUCK4_CPU_1V1	1.07	15.70	16.7		
	LDO1_CPU_1V1	1.08	4.60	5.0		
	LDO1_CPU_1V1_0V6	1.08	95.10	102.4		
	LDO2_CPU_3V3	3.30	1.40	4.7		
	LDO4_CPU_1V8	1.79	0.90	1.7		
	LDO5_CPU_3V0	2.91	0.00	0.0		

5.5.2 2-core Stream + MM07 + HiFi4 + CM33 CoreMark

This use case runs Stream and Suspend mode in parallel. A 720p TV display was connected to the MCIMX8ULP-EVK board through the HDMI interface.

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 Math Power busy loop stress test
- The GPU is running MM07 stress test

- Both CA35 cores are running Stream benchmark
- The CM33 core is running CoreMark benchmark
- All clocks and PLLs in APD and LPAV are turned off

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

1. Connect the display to the board through the HMDI interface.
2. Flash `flash_pms_coremark.bin` to eMMC:

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

3. Boot Power Mode Switch (PMS) with Coremark demo from the menu on the RTD console.
4. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`) and boot the board to the eMMC rootfs.
5. Run `setup_video.sh` (see Section 8 for more details).
6. Start two streams, each bound on separate CPU:

```
while [ "1" == "1" ]; do
   taskset -c 0 stream -M 200M -N 1000 &
   taskset -c 1 stream -M 200M -N 1000
done
```

7. Copy `hifi4_imx8ulp-MathPower_busy_loop.bin` to `/lib/firmware/imx/dsp/` and rename it as `hifi4.bin`.
8. Run commands to load and run the firmware:

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

where <n> is HiFi4 index in remote processors.

9. Run CoreMark benchmark from the menu on the RTD console.
10. Start 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 2-core Stream + MM07 + HiFi4 + CM33 CoreMark use case

Power group	Supply domain	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	17.20	30.9	687.75	38
	BUCK1_LSW1_CPU_1V8	1.80	0.39	0.7		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.4		
	BUCK1_LSW3_CPU_1V8	1.80	1.80	3.3		
	BUCK1_LSW4_CPU_1V8	1.80	1.00	1.8		
	BUCK2_CPU_1V0	1.00	33.40	33.3		
	BUCK3_CPU_1V0	1.09	438.30	477.9		
	BUCK4_CPU_1V1	1.07	15.70	16.8		
	LDO1_CPU_1V1	1.08	4.60	5.0		
	LDO1_CPU_1V1_0V6	1.08	95.80	103.1		
	LDO2_CPU_3V3	3.30	1.40	4.7		
	LDO4_CPU_1V8	1.79	1.00	1.7		
	LDO5_CPU_3V0	2.91	0.00	0.0		

5.5.3 GPU 2D + GPU 3D + CSI/DSI

This use case runs GPU 2D, GPU 3D, and CSI/DSI in parallel. A 720p TV display was connected to the MCIMX8ULP-EVK board through the HDMI interface.

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 GLMark benchmark
- The G2D is running a G2D test program
- MIPI-CSI/DSI is running camera image preview
- All clocks and PLLs in APD and LPAV are turned off

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

1. Connect the display to the board through the HDMI interface.
2. Connect the camera OV5640 to the board through the CSI interface.
3. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
4. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`) and boot the board to the eMMC rootfs.
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 camera capture 1080p30 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 G2D test program with the following command:

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

9. Run G3D test program with the following command:

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

10. Start 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 "GPU 2D + GPU 3D + CSI/DSI" use case

Power group	Supply domain	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	17.20	31.0	706.73	38
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.4		
	BUCK1_LSW3_CPU_1V8	1.80	3.20	5.7		
	BUCK1_LSW4_CPU_1V8	1.80	0.90	1.6		
	BUCK2_CPU_1V0	1.00	27.40	27.4		

Table 23. Power measurement results for "GPU 2D + GPU 3D + CSI/DSI" use case...*continued*

Power group	Supply domain	Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (℃)
	BUCK3_CPU_1V0	1.09	451.70	492.4		
	BUCK4_CPU_1V1	1.07	15.40	16.5		
	LDO1_CPU_1V1	1.08	4.70	5.0		
	LDO1_CPU_1V1_0V6	1.07	106.20	113.8		
	LDO2_CPU_3V3	3.30	1.20	3.9		
	LDO4_CPU_1V8	1.79	0.45	0.8		
	LDO5_CPU_3V0	2.91	0.00	0.0		

5.6 Memory use cases

The following memory-centric use case scenarios were tested:

- 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.

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 buffers is set to 1024 MB
- The CM33 core is in Sleep mode
- All clocks and PLLs in APD and LPAV are turned off

Before running the use case, the display must be turned off (see Section 8 for more details):

```
echo 1 > /sys/class/graphics/fb0/blank
```

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

1. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
2. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`) and boot the board to the eMMC rootfs.
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 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 "memset" use case

Power group	Supply domain	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	16.70	30.1	700.7	38
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.4		
	BUCK1_LSW3_CPU_1V8	1.80	0.00	0.0		
	BUCK1_LSW4_CPU_1V8	1.80	0.90	1.6		
	BUCK2_CPU_1V0	1.00	27.60	27.5		
	BUCK3_CPU_1V0	1.09	348.90	381.1		
	BUCK4_CPU_1V1	1.07	15.60	16.7		
	LDO1_CPU_1V1	1.07	3.50	3.7		
	LDO1_CPU_1V1_0V6	1.05	215.80	226.6		
	LDO2_CPU_3V3	3.30	1.20	3.9		
	LDO4_CPU_1V8	1.79	0.45	0.8		
	LDO5_CPU_3V0	2.91	0.00	0.0		

5.6.2 memcpy

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

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 buffers is set to 1024 MB
- The CM33 core is in Sleep mode
- All clocks and PLLs in APD and LPAV are turned off

Before running the use case, the display must be turned off (see Section 8 for more details):

```
echo 1 > /sys/class/graphics/fb0/blank
```

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

1. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
2. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`) and boot the board to the eMMC rootfs.
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 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 "memcpy" use case

Power group	Supply domain	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	16.70	30.1	592.26	35
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.4		
	BUCK1_LSW3_CPU_1V8	1.80	0.00	0.0		
	BUCK1_LSW4_CPU_1V8	1.80	0.83	1.5		
	BUCK2_CPU_1V0	1.00	27.50	27.5		
	BUCK3_CPU_1V0	1.09	361.30	394.5		
	BUCK4_CPU_1V1	1.07	15.70	16.7		
	LDO1_CPU_1V1	1.08	4.60	5.0		
	LDO1_CPU_1V1_0V6	1.08	96.20	103.7		
	LDO2_CPU_3V3	3.30	1.20	3.9		
	LDO4_CPU_1V8	1.79	0.45	0.8		
	LDO5_CPU_3V0	2.91	0.00	0.0		

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.

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 clocks and PLLs in APD and LPAV are turned off

Before running the use case, the display must be turned off (see Section 8 for more details):

```
echo 1 > /sys/class/graphics/fb0/blank
```

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

1. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
2. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`) and boot the board to the eMMC rootfs.
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 [ "1" == "1" ]; do
  taskset -c 0 stream -M 200M -N 1000 &
  taskset -c 0 stream -M 200M -N 1000
done
```

6. Start 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 "Stream" use case

Power group	Supply domain	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	16.70	30.1	607.87	35
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.4		
	BUCK1_LSW3_CPU_1V8	1.80	0.00	0.0		
	BUCK1_LSW4_CPU_1V8	1.80	0.90	1.6		
	BUCK2_CPU_1V0	1.00	27.10	27.0		
	BUCK3_CPU_1V0	1.09	371.70	405.9		
	BUCK4_CPU_1V1	1.07	15.60	16.7		
	LDO1_CPU_1V1	1.08	4.40	4.7		
	LDO1_CPU_1V1_0V6	1.08	100.80	108.5		
	LDO2_CPU_3V3	3.30	1.20	3.9		
	LDO4_CPU_1V8	1.79	0.45	0.8		
	LDO5_CPU_3V0	2.91	0.00	0.0		

5.7 Storage (eMMC) use cases

The following use case scenarios were tested:

- DD_RD_eMMC
- DD_WRT_eMMC

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

5.7.1 DD_RD_eMMC

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

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 clocks and PLLs in APD and LPAV are turned off

Note: *Before running the use case, the maximum amount of data that the kernel reads ahead for a single file should be set to 512 KB:*

```
echo 512 > /sys/block/<bdev>/queue/read_ahead_kb
```

Run the use case in a loop and start logging power measurement results at the desired time interval (recommended interval is 1 minute).

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

1. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.

2. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`) and boot the board to the eMMC rootfs.
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 eMMC partition and run it.
6. Start 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_RD_eMMC" use case

Power group	Supply domain	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	16.70	30.0	536.83	35
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.4		
	BUCK1_LSW3_CPU_1V8	1.80	0.00	0.0		
	BUCK1_LSW4_CPU_1V8	1.80	0.78	1.4		
	BUCK2_CPU_1V0	1.00	27.30	27.3		
	BUCK3_CPU_1V0	1.09	341.00	372.5		
	BUCK4_CPU_1V1	1.07	12.90	13.8		
	LDO1_CPU_1V1	1.09	4.10	4.4		
	LDO1_CPU_1V1_0V6	1.08	68.30	74.0		
	LDO2_CPU_3V3	3.30	1.20	3.9		
	LDO4_CPU_1V8	1.79	0.45	0.8		
	LDO5_CPU_3V0	2.91	0.00	0.0		

5.7.2 DD_WRT_eMMC

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

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 clocks and PLLs in APD and LPAV are turned off

Note: *Before running the use case, the maximum amount of data that the kernel reads ahead for a single file should be set to 512 KB:*

```
echo 512 > /sys/block/<bdev>/queue/read_ahead_kb
```

Run the use case in a loop and start logging power measurement results at the desired time interval (recommended interval is 1 minute).

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

1. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
2. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`) and boot the board to the eMMC rootfs.

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 eMMC partition and run it.
6. Start 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 "DD_WRT_eMMC" use case

Power group	Supply domain	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	16.80	30.3	451.72	35
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.4		
	BUCK1_LSW3_CPU_1V8	1.80	0.00	0.0		
	BUCK1_LSW4_CPU_1V8	1.80	0.78	1.4		
	BUCK2_CPU_1V0	1.00	27.20	27.2		
	BUCK3_CPU_1V0	1.09	274.60	300.3		
	BUCK4_CPU_1V1	1.08	12.00	12.9		
	LDO1_CPU_1V1	1.09	3.70	4.0		
	LDO1_CPU_1V1_0V6	1.09	48.90	53.2		
	LDO2_CPU_3V3	3.30	1.20	3.9		
	LDO4_CPU_1V8	1.79	5.60	9.9		
	LDO5_CPU_3V0	2.91	0.00	0.0		

5.8 Product use cases

The following use case scenarios were tested:

- Parallel E-Ink page flip
- Parallel E-Ink partial screen update
- Machine vision
- eIQ – Benchmark
- UAC audio playback
- Always-ON display, 1 fps refresh rate

Note: *Change the host and target IP addresses into scripts as required. Also, change the media file name/ locations into the script before running the script. The gstreamer commands are for both the target and the host; rest of the commands are only for target.*

Note: *The following subsections provide an example for copying the caps filter, which can be used for many use cases.*

5.8.1 Parallel E-Ink page flip

For this use case, the EPDC expansion port was connected to a parallel VB3300-FOC E-Ink panel. In this use case, a 2-page content buffer is ready (without rendering). The FBIOPAN_DISPLAY ioctl is used to flip the buffer.

Following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value)
- The CM33 core is in Sleep mode
- All clocks and PLLs in APD and LPAV are turned off

Run the use case in a loop and start logging power measurement results at the desired time interval (recommended interval is 1 minute).

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

1. Connect the EPDC panel to the board through the parallel interface.
2. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
3. Boot the Linux image with the DTB configuration `imx8ulp-evk-epdc.dtb` and boot the board to the eMMC rootfs.
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 the following command:

```
/unit_tests/Display/mxc_epdc_v2_fb_test.out -n 20
```

7. Start 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 "Parallel E-Ink page flip" use case

Power group	Supply domain	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	17.00	30.6	489.89	38
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.60	8.3		
	BUCK1_LSW3_CPU_1V8	1.80	1.80	3.2		
	BUCK1_LSW4_CPU_1V8	1.80	2.30	4.1		
	BUCK2_CPU_1V0	0.99	27.20	27.0		
	BUCK3_CPU_1V0	1.10	278.60	305.7		
	BUCK4_CPU_1V1	1.08	12.50	13.4		
	LDO1_CPU_1V1	1.09	3.90	4.2		
	LDO1_CPU_1V1_0V6	1.08	82.30	88.9		
	LDO2_CPU_3V3	3.29	1.10	3.5		
	LDO4_CPU_1V8	1.80	0.44	0.8		
	LDO5_CPU_3V0	2.92	0.00	0.0		

5.8.2 Parallel E-Ink partial screen update

For this use case, the EPDC expansion port was connected to a parallel VB3300-FOC E-Ink panel. In this use case, updates are only visible in a small rectangular portion (10% – 20%) of the screen.

Following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value)
- The CM33 core is in Sleep mode
- All clocks and PLLs in APD and LPAV are turned off

Run the use case in a loop and start logging power measurement results at the desired time interval (recommended interval is 1 minute).

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

1. Connect the EPDC panel to the board through the parallel interface.
2. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
3. Boot the Linux image with the DTB configuration `imx8ulp-evk-epdc.dtb` and boot the board to the eMMC rootfs.
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 the following command:

```
/unit_tests/Display/mxc_epdc_v2_fb_test.out -n 19
```

7. Start 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 "Parallel E-Ink partial screen update" use case

Power group	Supply domain	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	17.00	30.6	495.82	38
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.60	8.3		
	BUCK1_LSW3_CPU_1V8	1.80	1.80	3.2		
	BUCK1_LSW4_CPU_1V8	1.80	2.20	3.9		
	BUCK2_CPU_1V0	0.99	27.10	27.0		
	BUCK3_CPU_1V0	1.10	285.50	313.1		
	BUCK4_CPU_1V1	1.08	12.30	13.3		
	LDO1_CPU_1V1	1.09	3.90	4.2		
	LDO1_CPU_1V1_0V6	1.08	81.30	87.8		
	LDO2_CPU_3V3	3.29	1.10	3.6		
	LDO4_CPU_1V8	1.80	0.44	0.8		
	LDO5_CPU_3V0	2.92	0.00	0.0		

5.8.3 Machine vision

This use case detects object in video file, which is captured from OV5640.

For "Machine vision" and "eIQ benchmark" use cases, use the following commands in `/usr/bin/tensorflow-lite-2.10.0/examples` (the path may vary depending on the BSP revision) to obtain the required files:

```
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
```

Following are the required settings for this use case:

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

- Capture 1080p30 fps yuyv data from OV5640
- Use G2D to perform colorspace conversion and downsize to 480p
- Object detection is done using the file-based input
- Stress each CA35 core to simulate application stress — 100 %
- The CM33 core is in Sleep mode
- All clocks and PLLs in APD and LPAV are turned off

Run the use case in a loop and start logging power measurement results at the desired time interval (recommended interval is 1 minute).

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

1. Connect the display to the board through the HDMI interface.
2. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
3. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`) and boot the board to the eMMC rootfs.
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-2.10.0/examples` in rootfs and run `/usr/bin/tensorflow-lite-2.10.0/examples/MV.sh`. For more details, see Section 8.
8. Start 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 "Machine vision" use case

Power group	Supply domain	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	17.10	30.8	635.65	37
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.4		
	BUCK1_LSW3_CPU_1V8	1.80	1.80	3.3		
	BUCK1_LSW4_CPU_1V8	1.80	0.83	1.5		
	BUCK2_CPU_1V0	1.00	27.20	27.2		
	BUCK3_CPU_1V0	1.09	408.20	445.4		
	BUCK4_CPU_1V1	1.07	13.50	14.5		
	LDO1_CPU_1V1	1.08	4.00	4.3		
	LDO1_CPU_1V1_0V6	1.08	88.10	95.0		
	LDO2_CPU_3V3	3.30	1.20	3.9		
	LDO4_CPU_1V8	1.79	0.67	1.2		
	LDO5_CPU_3V0	2.91	0.00	0.0		

5.8.4 eIQ benchmark

This use case loop runs benchmark_model to test the eIQ benchmark.

For "Machine vision" and "eIQ benchmark" use cases, use the following commands in `/usr/bin/tensorflow-lite-2.10.0/examples` (the path may vary depending on the BSP revision) to obtain the required files:

```
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
```

Following are the required settings for this use case:

- The CPU frequency is set to 800 MHz (default value)
- Run eIQ benchmark test (standard models)
- All clocks and PLLs in APD and LPAV are turned off

Run the use case in a loop and start logging power measurement results at the desired time interval (recommended interval is 1 minute).

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

1. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
2. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`) and boot the board to the eMMC rootfs.
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-2.10.0/examples` in rootfs and run `/usr/bin/tensorflow-lite-2.10.0/examples/ML.sh 1`
6. Start power measurement 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 "eIQ benchmark" use case

Power group	Supply domain	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	17.00	30.6	546	35
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.4		
	BUCK1_LSW3_CPU_1V8	1.80	1.80	3.3		
	BUCK1_LSW4_CPU_1V8	1.80	0.90	1.6		
	BUCK2_CPU_1V0	1.00	27.80	27.8		
	BUCK3_CPU_1V0	1.09	337.90	369.2		
	BUCK4_CPU_1V1	1.08	12.20	13.1		
	LDO1_CPU_1V1	1.09	3.70	4.0		
	LDO1_CPU_1V1_0V6	1.08	76.70	83.1		
	LDO2_CPU_3V3	3.30	1.20	3.9		
	LDO4_CPU_1V8	1.79	0.45	0.8		
	LDO5_CPU_3V0	2.91	0.00	0.0		

5.8.5 UAC audio playback

Create a Linux UAC gadget device on one i.MX 8ULP, connect the device to another i.MX 8ULP as host through USB. Receive host audio through UAC and decode/playback to onboard codec.

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 clocks and PLLs in APD and LPAV are turned off

Run the use case in a loop and start logging power measurement results at the desired time interval (recommended interval is 1 minute).

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

1. Prepare a PC host with latest Ubuntu installed.
2. Connect USB0 of the MCIMX8ULP-EVK board to PC host with a Type-C USB cable.
3. Boot Power Mode Switch (PMS) demo from the menu on the RTD console.
4. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`) and boot the board to the eMMC rootfs.
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` (see Section 8 for more details).
7. Run `uac2_gadget.sh`. You should be able to see a UAC gadget as an audio card with the `aplay -l` and `arecord -l` commands, as shown in Figure 4 and Figure 5.

```
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 4. Output of "aplay -l" command

```
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 5. Output of "arecord -l" command

8. Check if UAC Gadget device is detected on PC host, as shown in Figure 6.

```
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 6. UAC gadget device detection

9. Play music in Ubuntu on PC host with the following command:

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

10. Receive music data and play on the MCIMX8ULP-EVK board with the following command:

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

11. Check if music can be heard from 3.5 mm audio jack on the MCIMX8ULP-EVK board.
12. Start power measurement 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 "UAC audio playback" use case

Power group	Supply domain	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	16.50	29.8	426.48	37
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.79	8.50	15.3		
	BUCK1_LSW3_CPU_1V8	1.80	0.00	0.0		
	BUCK1_LSW4_CPU_1V8	1.80	0.89	1.6		
	BUCK2_CPU_1V0	0.99	34.20	34.0		
	BUCK3_CPU_1V0	1.10	247.60	271.6		
	BUCK4_CPU_1V1	1.08	11.40	12.3		
	LDO1_CPU_1V1	1.09	3.60	3.9		
	LDO1_CPU_1V1_0V6	1.09	44.30	48.3		
	LDO2_CPU_3V3	3.29	2.70	8.9		
	LDO4_CPU_1V8	1.80	0.39	0.7		
	LDO5_CPU_3V0	2.92	0.00	0.0		

5.8.6 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.

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 re-initialize the LPAV domain to control MIPI-DSI and DCNano
- 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 clocks and PLLs in APD and LPAV are turned off

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

1. Connect the RK055HDMIPI4M panel to the board through the MIPI interface.
2. Flash `flash_aod_RK055AHD091.bin` to eMMC:

   ```
   uuu -b emmc flash_aod_RK055AHD091.bin
   ```

   Note: *Both RK055AHD091 and RK055MHD091 belong to RK055HDMIPI4M. The image is provided in the software associated with this application note.*
3. Boot AOD demo from the menu on RTD console.
4. Boot the Linux image with the default DTB configuration (`imx8ulp-evk.dtb`) and boot the board to the eMMC rootfs.
5. Put Linux to Suspend mode using the following command:

   ```
   echo mem > /sys/power/state
   ```
6. On RTD console, press Z to run low-power display.
7. Check if the entire screen is filled with colors where red, green, and blue colors are switched every second.
8. Start power measurement and record the results.

Table 34 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 34. Power measurement results for "Always-ON display, 1 fps refresh rate" use case

Power group	Supply domain	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	6.00	10.8	92.45	Die temperature was not measured as the CA35 core was suspended.
	BUCK1_LSW1_CPU_1V8	1.80	1.30	2.3		
	BUCK1_LSW2_CPU_1V8	1.80	0.00	0.0		
	BUCK1_LSW3_CPU_1V8	1.80	1.30	2.4		
	BUCK1_LSW4_CPU_1V8	1.80	0.00	0.0		
	BUCK2_CPU_1V0	0.99	32.30	32.1		
	BUCK3_CPU_1V0	1.00	44.80	44.8		
	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	3.29	0.00	0.0		
	LDO4_CPU_1V8	1.81	0.00	0.0		
	LDO5_CPU_3V0	2.90	0.03	0.1		

5.9 System-level power estimation use cases

The following use case scenarios were tested:

- 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. Boot the Linux image with `imx8ulp-evk.dtb` in APD.
2. Boot the PMS demo from the menu on the RTD console.
3. Press ON/OFF key for 3 seconds.
4. Measure the power and record the results.

Table 35 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 35. Power measurement results for "Battery" use case

Power group	Supply domain	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	8.78	0.00	8.75	Die temperature was not measured as the CA35 core was suspended.
	BUCK1_LSW1_CPU_1V8	0.00	1.53	0.00		
	BUCK1_LSW2_CPU_1V8	0.00	1.31	0.00		
	BUCK1_LSW3_CPU_1V8	0.00	0.73	0.00		
	BUCK1_LSW4_CPU_1V8	0.00	1.23	0.00		
	BUCK2_CPU_1V0	0.00	13.22	0.00		
	BUCK3_CPU_1V0	0.00	56.19	0.00		
	BUCK4_CPU_1V1	0.00	0.55	0.00		
	LDO1_CPU_1V1	0.00	2.66	0.00		
	LDO1_CPU_1V1_0V6	0.00	13.82	0.00		
	LDO2_CPU_3V3	0.00	1.40	0.00		
	LDO4_CPU_1V8	0.00	1.63	0.00		
	LDO5_CPU_3V0	2.94	2.98	8.75		

Note: *For the following three use cases, the unit of power is µW:*

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

5.9.2 256 KB L2 cache

Run the CoreMark benchmark on the CA35 cores with half L2 cache size.

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 clocks and PLLs in APD and LPAV are turned off

Run the use case in a loop and start logging power measurement results at the desired time interval (recommended interval is 1 minute).

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

1. Flash `flash_l2_256k_cache.bin` to eMMC:

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

2. Boot the Linux image with `imx8ulp-evk.dtb` in APD.
3. Boot the PMS demo from the menu on the RTD console.
4. Run `setup.sh`.
5. Run `coremark_loop.sh`:

```
while [ "1" == "1" ]
do
 ./coremark > /dev/null 2>&1
```

```
done
```

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

Table 36 shows the measurement results when this use case is applied to the i.MX 8ULP processor.

Table 36. Power measurement results for "256 KB L2 cache" use case

Power group	Supply domain	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	16.70	30.0	520.04	34
	BUCK1_LSW1_CPU_1V8	1.80	0.06	0.1		
	BUCK1_LSW2_CPU_1V8	1.80	4.70	8.4		
	BUCK1_LSW3_CPU_1V8	1.80	0.00	0.0		
	BUCK1_LSW4_CPU_1V8	1.80	0.78	1.4		
	BUCK2_CPU_1V0	1.00	27.30	27.3		
	BUCK3_CPU_1V0	1.09	356.40	389.2		
	BUCK4_CPU_1V1	1.08	11.50	12.4		
	LDO1_CPU_1V1	1.09	3.60	3.9		
	LDO1_CPU_1V1_0V6	1.09	38.90	42.5		
	LDO2_CPU_3V3	3.30	1.20	3.9		
	LDO4_CPU_1V8	1.79	0.45	0.8		
	LDO5_CPU_3V0	2.91	0.00	0.0		

6 Recommendations for reducing power consumption

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

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 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 Linux BSP and/or SDK. Further optimizations can be done to the individual user's system.

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

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

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 constant data is being processed (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.

It is recommended to place the i.MX 8ULP processor into the low-power mode, as much 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. The NXP BSP release software implements clock gating, by default.

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 Use of LPDDR4 instead of DDR4

The memory architecture of LPDDR4 is designed to achieve higher bandwidth and lower power consumption. It reduces power consumption by lowering the supply voltage. For DDR4, the I/O voltage is 1.2 V; whereas, for LPDDR4, the I/O voltage is 1.1 V. The DDR I/O power can be calculated from the following formula:

Imax = N x C x V x (0.5 x F)

where:

- N: Number of DDR I/O pins supplied by the power line
- C: Equivalent external capacitive load (PCB trace and via capacitance + DDR pin capacitance)
- V: DDR I/O voltage (1.1 V or 1.2 V)
- (0.5 x F): Data change rate (DDR bus frequency)

In this formula, Imax is in Amps, C is in Farads, V is in Volts, and F is in Hertz.

Note: *The 0.5 value used in the above formula is the worst-case value as DDR address and data signals are not toggling every cycle. You can make your own estimations by adjusting the 0.5 value to correspond to your usage profile.*

The formula indicates the increase in current relative to an increase in the I/O pin voltage. This additional power is consumed by both the DDR memory and the i.MX 8ULP processor, as they both drive the DDR bus. The additional power consumption adds to the increased thermal dissipation of each device. Therefore, it has a double effect on energy consumption. Thermal effect and management are explained in Section 6.11.

6.6 Lower DDR frequencies

As explained earlier, the DDR I/O bus frequency also contributes to the DDR I/O current. It is represented as “F” in the formula mentioned in Section 6.5. 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.

6.7 DDR interface optimization

Following are some ways to optimize 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 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 correct sizes for the ECC DDR regions is also important as larger regions use higher energy

6.8 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.

i.MX 8ULP uses 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).

6.9 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 WAIT/STOP state much 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 utilizes the optimal cores for the specific workloads or tasks on i.MX 8ULP for maximum efficiency. It is often easier said than done but it provides significant power savings if the system can return to the low-power state faster (see Section 6.1).

6.10 Use of SSRAM to minimize DDR accesses

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

6.11 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, then 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.

6.12 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 37).

Table 37 shows the power supply configuration to minimize power consumption in PD modes.

Table 37. 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
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)

Table 37. Power supply configuration to minimize power consumption in PD modes...*continued*

Power domain	CM33-PD/CA35-PD
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) specifically designed for use with the i.MX 8ULP processor. For more details on 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 7 shows an example power distribution using the PCA9460A PMIC for the NXP i.MX 8ULP + LPDDR4.

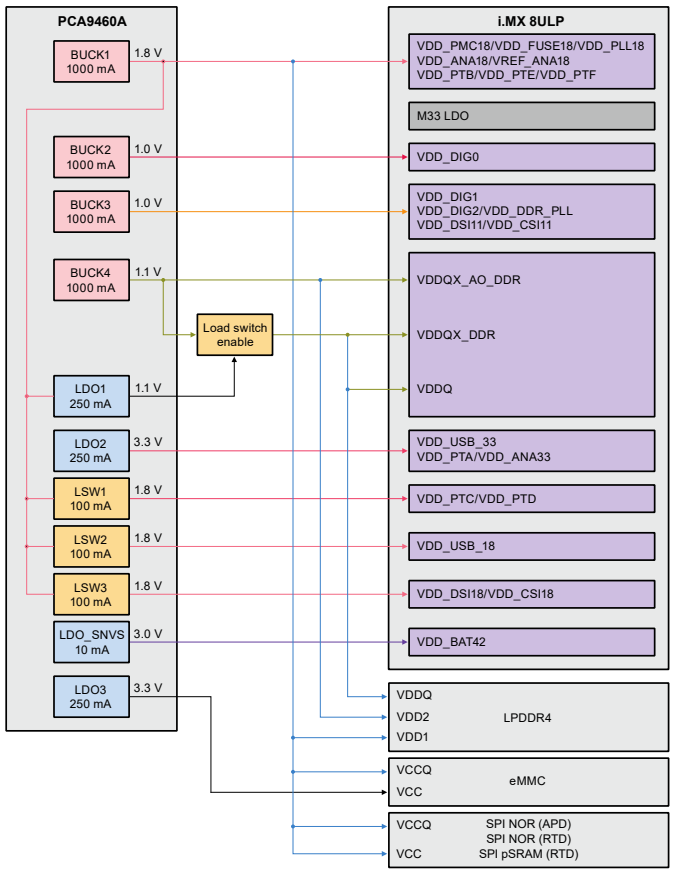

Figure 7. Recommended power distribution with LPDDR4

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

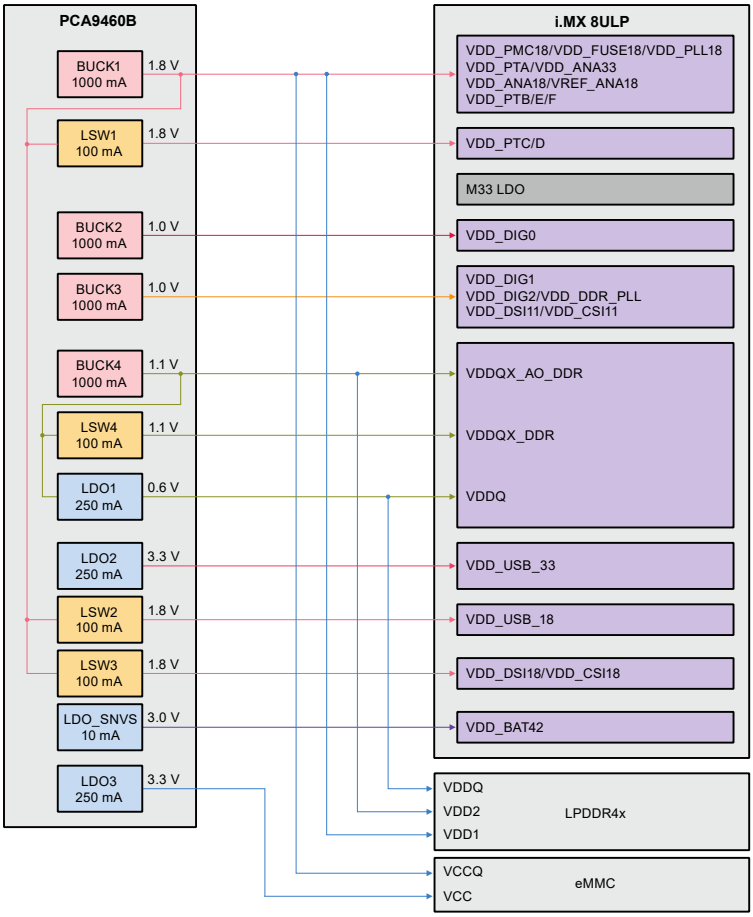

Figure 8. Recommended power distribution with LPDDR4x

8 Important commands

Some important scripts and commands are described below:

- Scripts to configure 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
echo 1 > /sys/class/graphics/fb0/blank
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
echo 1 > /sys/class/graphics/fb0/blank
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
echo 1 > /sys/class/graphics/fb0/blank
echo 0 > /sys/class/graphics/fb0/blank
```

- `setup_video_stream.sh`: This script opens the Ethernet to play the video online. It stops the Weston service 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 up
done
echo 1 > /sys/class/graphics/fb0/blank
echo 0 > /sys/class/graphics/fb0/blank
```

- `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
echo 1 /sys/class/graphics/fb0/blank
rmmod moal
#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 88W8987 Bluetooth device): This script is used to initialize host controller interface (HCI) in 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 IW416 Bluetooth device): This script is used to initialize HCI in 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_pulseaudio.sh`: This script is used to set up PulseAudio:

```
#!/bin/sh
pulseaudio --start --log-target=syslog
bluetoothctl << EOF
   power on
   agent on
   default-agent
EOF
bluetoothctl scan on # Got MAC address of remote BT device.
bluetoothctl << EOF
   pair <BT_MAC_ADDR>
   trust <BT_MAC_ADDR>
   connect <BT_MAC_ADDR>
EOF
```

– `uac2_gadget.sh`: This script is used to create a user account control (UAC) gadget device on the MCIMX8ULP-EVK 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=raw.yuv num-buffers=-1 ! \
clocksync ! queue max-size-buffers=2 max-size-time=0 ! \
rawvideoparse format=i420 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 for running 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: 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: 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 cpulimit application: Run these commands to build cpulimit application:
  1. Clone 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 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 application:

     ```
     make clean
     make
     ```

9 Related documentation

Table 38 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 38. Related documentation

Document	Description	Link / how to obtain
i.MX 8ULP Processor Reference Manual	Provides a detailed description about the i.MX 8ULP processor and its features, including memory maps, power supplies, and clocks.	Contact NXP FAE / sales representative
i.MX 8ULP Applications Processor—Consumer Products Data Sheet	Provides information about electrical characteristics, hardware design considerations, and ordering information.	

Table 38. Related documentation*...continued*

Document	Description	Link / how to obtain
i.MX 8ULP Applications Processor—Industrial Products Data Sheet		
i.MX 8ULP Hardware Developer's Guide	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.	
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 39 lists the acronyms used in this document.

Table 39. Acronyms

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

Table 39. Acronyms...*continued*

Acronym	Description
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
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
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

Table 39. Acronyms...*continued*

Acronym	Description
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
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:

Copyright 2023 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, 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 40 summarizes the revisions to this document.

Table 40. Revision history

Revision number	Release date	Description
2	21 September 2023	Updated *PCA9460 Data Sheet* link in Table 38.
		Removed AN13951 reference from the document.
1	11 August 2023	Initial public release

13 Legal information

13.1 Definitions

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

13.2 Disclaimers

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

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

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

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

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

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

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

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

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

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

Suitability for use in non-automotive qualified products — Unless this 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.

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.

13.3 Trademarks

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

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

Contents

For more information, please visit: http://www.nxp.com

Date of release: 21 September 2023
Document identifier: AN13914