AN14506

i.MX 91 Power Consumption Measurement

Rev. 1.1 — 6 January 2025 Application note

Document information

Information	Content
Keywords	AN14506, i.MX 91, i.MX 91-EVK board, power consumption, i.MX 91 power domains, use cases and measurement results
Abstract	This application note describes how to measure the current drain of the i.MX 91 application processor on the NXP i.MX 91 EVK board, through different use cases.

1 Introduction

The purpose of this application note is to help system designers to create power-optimized systems. It describes how to measure the current drain of the i.MX 91 application processor on an NXP IMX91LP4EVK-11 (also referred to as the i.MX 91 EVK board) through different use cases. Users can choose the appropriate power supply domains for the i.MX 91 processor and become familiar with the expected processor power consumption in various scenarios.

Note:

1. *Some use case binaries can be found in AN14506SW.zip. Copy the binaries (Linux app) to the board, then use "`chmod +x *`" to grant execute permissions.*
2. *The reported results are not guaranteed because the data presented in this application note is based on empirical measurements performed on a small sample size.*
3. *In this document, IMX91LP4EVK-11 board is also referred to as the i.MX 91 EVK board.*

This document also describes the power domains and power modes supported by the i.MX 91 processor and suggestions to help reduce system power consumption.

2 i.MX 91 power architecture

The power architecture of the chip is established with the presumption that the most affordable systems are built for the scenario in which the PMIC is used to supply all the power rails to the processor.

Figure 1 shows the power architecture diagram for the entire SoC.

Note: *Figure 1 shows only the power supplies and does not show the capacitors that can be required for internal LDO regulators.*

The values in Table 1 are for reference purpose only. For actual values, refer to the i.MX 91 Applications Processors Data Sheet (see Section 8).

From an architectural standpoint, most SoC digital and analog logic can be power gated in Low-power mode through an external PMIC supply or an internal power switch.

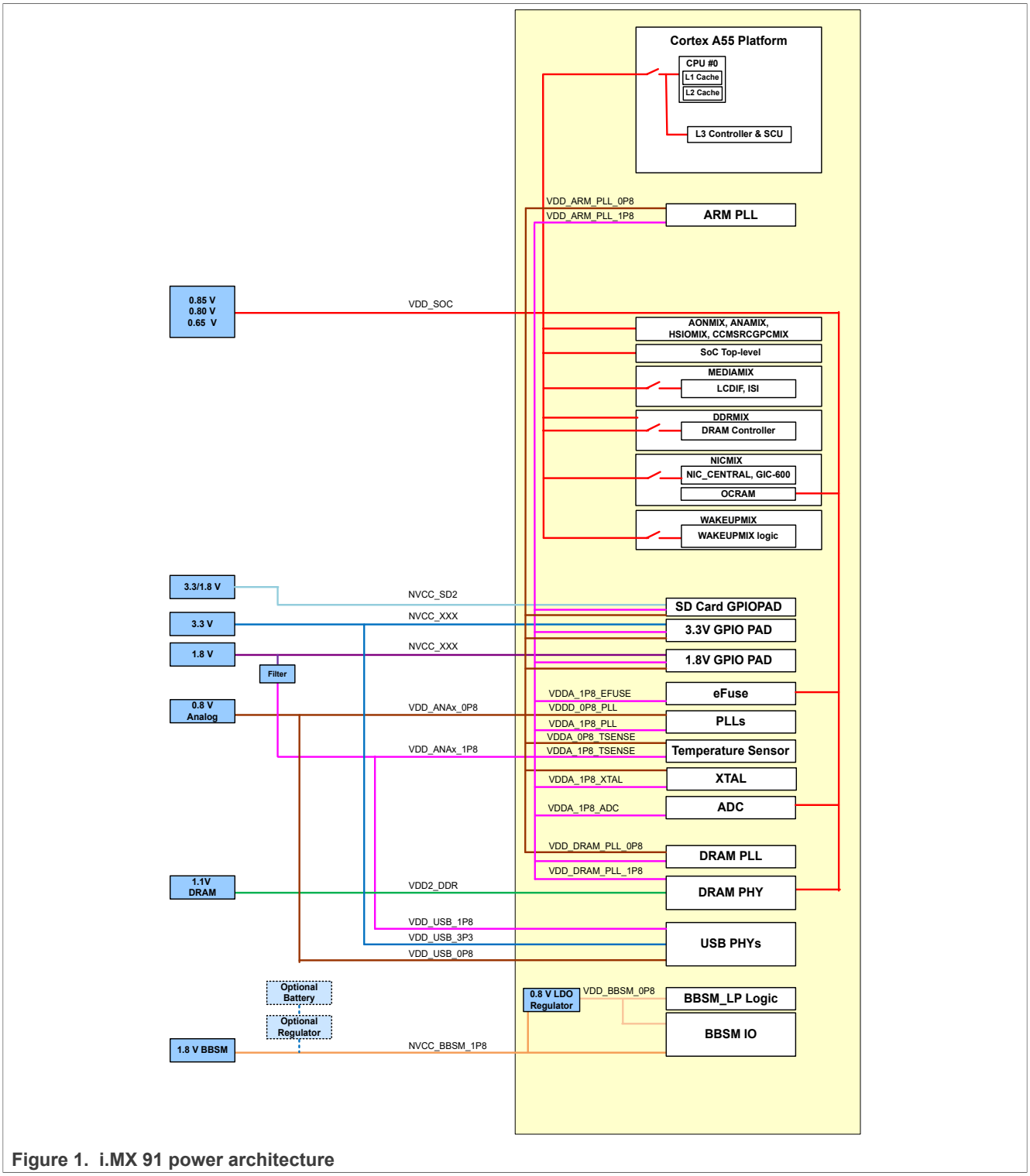

Figure 1. i.MX 91 power architecture

Table 1 summarizes the operating condition for all the external power rails.

Note: *Table 1 does not include I/O for capacitors required to support internal LDO regulators. The values in this table are for reference purpose only. For actual values, refer i.MX 91 Applications Processors Data Sheet.*

Table 1. External power supply

Power rail	Vmin (V)	Vtyp (V)	Vmax (V)	Description
VDD_SOC	0.80	0.85	0.90	Power supply for SoC, Nominal drive mode
	0.76	0.80	0.84	Power supply for SoC, Low-drive mode
	0.61	0.65	0.69	Power supply for SoC, Suspend mode
VDD_ANAx_0P8 VDD_USB_0P8	0.76	0.80	0.90	Digital supply for PLLs, temperature sensor, LVCMOS I/O, and USB PHYs
VDD_ANAx_1P8 VDD_USB_1P8	1.71	1.80	1.89	1.8 V supply for PLLs, eFuse, temperature sensor, LVCMOS voltage detect reference, ADC, 24 MHz XTAL, and USB PHYs
VDD_USB_3P3	3.07	3.30	3.45	3.3 V supply for USB PHY (V_{max} consistent with V_{max} supported by NVCC GPIO supplies)
VDD2_DDR	1.06	1.10	1.14	Voltage tolerances as per LPDDR4 PHY
NVCC_BBSM_1P8	1.65	1.80	1.95	I/O supply for GPIO in BBSM bank
NVCC_AON NVCC_SD2 NVCC_GPIO NVCC_WAKEUP	1.65	1.80	1.95	Power supply for GPIO when it is in 1.8 V mode
	3.00	3.30	3.45	Power supply for GPIO when it is in 3.3 V mode

3 i.MX 91 power overview

This section describes the power domains and modes for the i.MX 91 processor.

3.1 i.MX 91 power domains overview

To optimize the power consumption in low-power modes, i.MX 91 has multiple power domains. As a result, most SoC digital and analog logics are power gated with an internal power switch or an external supply from PMIC in Low-power mode.

Note: *The values in Table 2 are for reference purpose only. For actual values, refer to the i.MX 91 Applications Processors Data Sheet. The link is listed in Section 8.*

Table 2. SoC power domains

Domain	Power Supply	Type	Physical Domains
PDCPU	VDD_SOC with switch	Digital	Cortex-A55 CPU and SS.
SOC	VDD_SOC	Digital	CCMSRCGPCMIX, AONMIX, ANAMIX (digital portion), SocTop-level
MEDIA_LOGIC	VDD_SOC with switch	Digital	MEDIAMIX (excluding the always-on PHYs)
WAKEUP	VDD_SOC with switch	Digital	WAKEUPMIX
DRAM_LOGIC	VDD_SOC with switch	Digital	DDRC (controller/PLL-related logic, but not PHY-related logic)
NIC	VDD_SOC with switch	Digital	NICMIX
BBSM_LOGIC	VDD_BBSM_0P8	Digital	BBSMMIX
ANALOG_0P8	VDDA_0P8	Analog	ANAMIX (analog 0.8 portion)
ANALOG_1P8	VDDA_1P8	Analog	ANAMIX (analog 1.8 portion)
USB_PHY_3P3	VDD_USB_3P3	Analog	USBPHY (HSIOMIX)
USB_PHY_1P8	VDD_USB_1P8	Analog	USBPHY (HSIOMIX)
USB_PHY_0P8	VDD_USB_0P8	Digital	USBPHY (HSIOMIX)
DRAM_IO	VDD2_DDR	IO	DDRMIX
ANALOG_DRAM	VDDA_DRAM	Analog	DDRMIX
NVCC_<XXX>	NVCC_<XXX>	IO	1.8V/3.3V GPIO
BBSM_IO	NVCC_BBSM_1P8	IO	BBSMIO Pads

Note:

- *For the recommended operating conditions of each supply rail, and a detailed description of the groups of pins powered by each I/O voltage supply, refer i.MX 91 Applications Processors Data Sheet.*
- *For more details regarding the i.MX 91 power architecture, refer to the i.MX 91 Applications Processors Reference Manual (document IMX91RM).*

3.2 i.MX 91 power mode overview

The i.MX 91 supports the following power modes:

- **Run mode**: In this mode, all external power rails are on, the Cortex-A55 is active and running; other internal modules can be on/off based on the application.
- **Idle mode**: This is the mode in which the Cortex-A55 core automatically enters when there is no thread running and all high-speed devices are not active. The Cortex-A55 can be put into power gated state, DRAM and the bus clock are reduced. Most of the internal logic is clock gated, but still remains powered. As compared with the Run mode, all the power rails from the power management remain the same. Most of the modules still remain in their state as in Run mode, so the interrupt response in this mode is very small.
- **Suspend mode**: This mode defined as the most power-saving mode. in this mode, all the clocks and the unnecessary power supplies are off and all power gateable portions of the SoC are power gated. Also, the Cortex-A55 CPU is fully power gated. Further, all internal digital logic, the analog circuits that can be powered down are off, and all PHYs are power gated. VDD_SOC (and related digital supply) voltage is reduced to the "Suspend mode" voltage. As compared to Idle mode, the exit time from this mode is longer, but the power consumption is also much lower.
- **BBSM mode**: This mode is also called the RTC mode. In this mode, only the power for the BBSM domain remains on to keep RTC and BBSM logic alive.
- **Off mode**: In this mode, all power rails are off.

3.2.1 Low-power modes

Table 3 defines the state of each module in the Idle mode, Suspend mode, and BBSM mode.

Note: *The values in Table 3 are for reference purpose only. For actual values, refer i.MX 91 Applications Processors Data Sheet. See Section 8.*

Table 3. Low-power mode definition

Parameter	IDLE	SUSPEND	BBSM
CCM LPM mode	WAIT	STOP	N/A
Arm Cortex-A55 CPU0	OFF	OFF	OFF
MEDIA	OFF	OFF	OFF
DRAM controller and PHY	ON	OFF	OFF
WAKEUPMIX	ON	ON	OFF
NICMIX	ON	OFF	OFF
ARM_PLL	OFF	OFF	OFF
DRAM_PLL	OFF	OFF	OFF
SYSTEM_PLL1	ON	OFF	OFF
XTAL	ON	ON	OFF
RTC	ON	ON	ON
External DRAM device	Self-Refresh [1]	Self-Refresh [2]	OFF
USB PHY	In Low Power State	OFF	OFF
DRAM clock	200 MHz	OFF	OFF
AXI clock	133 MHz	OFF	OFF
Module clocks	ON as needed	OFF	OFF
EdgeLockSecure Enclave	ON	ON	OFF
GPIO Wakeup	Yes	Yes	OFF
RTC Wakeup	Yes	Yes	Yes
USB remote wakeup	Yes	No [3][4]	No
Other wakeup source [5]	Yes	No	No

[1] Automatic enter self-refresh when there is no DRAM access.
[2] Put into self-refresh mode by software before entering low power mode.
[3] Turn off externally by PMIC when PMIC_STBY_REQ signal is asserted.
[4] USB remote wakeup can be "Yes" if required.
[5] Remote wakeup can be supported if the USB PHY power is ON in this mode.

4 i.MX 91 processor power measurement

This document provides details of several use cases run by NXP on the NXP i.MX 91-EVK board to measure i.MX 91 power. These use cases are described under Section 5.

4.1 Hardware and software requirements

Table 4 provides details of the hardware and software used during the power measurement.

Table 4. Hardware and software used

Category	Description
Hardware	NXP IMX91LP4EVK-11 board, 91079 A1 + 91080 A1
Software	Linux kernel version: L6.6.36
	Yocto rootfs
	BCU tool is available at bcu version: 1.1.92

Note: *For the Software used, the measurements are performed using the onboard measurement circuitry and BCU software tool. These measurements are taken at room temperature without thermal forcing equipment.*

4.2 Building the i.MX Yocto Project

To build the i.MX Yocto Project, perform the steps as follows:

1. To download and build the i.MX Yocto Project community BSP recipe layers, run the following commands:

```
repo init -u https://github.com/nxp-imx/imx-manifest \
-b imx-linux-scarthgap -m imx-6.6.36-2.1.0.xml
repo sync
DISTRO=fsl-imx-xwayland MACHINE=imx91-11x11-lpddr4-evk source \
imx-setup-release.sh -b build-imx91-11x11-lpddr4-evk
```

   For more information on the i.MX Yocto Project, refer to *i.MX Yocto Project User's Guide*.
2. To build, run the following command:

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

   The build image can be found in `build-imx91-11x11-lpddr4-evk/tmp/deploy/image`.

4.3 Power consumption measurement

To measure the i.MX 91 power consumption, the steps are as follows:

1. Connect a Type-C USB cable between the host PC and the J1401 USB port on the i.MX 91-EVK board.
2. To start the monitor in the BCU path, run the following command:

```
bcu monitor -board=imx91evk11
```

3. Run the related use cases. These use cases are described in Section 5.
4. To reset the value, press "**3**" once the use case starts.
5. To switch measurement precision: mA/auto/uA, press "**4**". This step is optional.
6. Wait for 1 minute and record the data in the BCU.

The measurements are taken mainly for the power supply domains shown in Table 5. This table also provides a mapping between the power rails in BCU software, and the power supply domains in the i.MX 91 processor. For more information, download *BCU.pdf*.

Table 5. Measured power supply domains

Power groups	Power supply domains	Description
GROUP_SOC_FULL	nvcc_1p8	Power supply for NVCC_WAKEUP rail
	nvcc_3p3	Power supply for NVCC_AON and NVCC_GPIO rails
	nvcc_bbsm_1p8	I/O supply for GPIO in BBSM bank
	nvcc_sd2	Power supply for GPIO in NVCC_SD2 rail
	vdd2_ddr	Power supply for DRAM PHY
	vdd_ana_0p8	Power supply for PLLs, temperature sensor, LVCMOS I/O, MIPI, and USB PHYs
	vdd_ana_1p8	1.8 V power supply for PLLs, eFuse, temperature sensor, LVCMOS voltage detect reference, ADC, 24 MHz XTAL, LVDS, MIPI, and USB PHYs
	vdd_soc	Power supply for VDD_SOC
	vdd_usb_3p3	Power supply for USB PHYs
	vddq_ddr	Power supply for DRAM PHY

Note: *As the i.MX 91 EVK board reuses the PCB design of i.MX 93 EVK, VDDQ_DDR is still retained. However, it is actually used as VDD2_DDR to power the DDR PHY. The actual power consumption of VDD2_DDR should be the sum of VDD2_DDR and VDDQ_DDR.*

5 Use cases and measurement results

The main use cases and subcases that form the benchmarks for the i.MX 91 internal power measurements on the EVK platform are described in the following sections.

Note:

- *Before running a use case,* `<configuration_script>.sh` *must be run to configure the environment, see Section 7.*
- *The boot mode for all use cases is Single Boot (eMMC) mode.*
- *The current sample resistors on the power path create a drop in the voltage on each power rail.*

Table 6 summarizes the power measurement results of various use cases performed on the i.MX 91-EVK board.

Table 6. i.MX 91-EVK power summary report

Use cases category	Use cases	Total power (sum of average powers in GROUP_SOC_FULL) (mW)
Core benchmark use cases	Dhrystone	419.54
	CoreMark	390.57
Memory use cases	Memset	641.74
	Memcpy	495.78
	Stream	550.81
Audio playback use cases	Audio playback (gplay)	301.30
	Audio low-bus playback (aplay)	197.11
Storage use cases	DD_WRITE_eMMC	318.67
	DD_READ_eMMC	415.72
	DD_WRITE_SD	425.15
	DD_READ_SD	442.48
Low-power mode use cases	System Idle with display in ND mode	332.21
	System Idle with display in LD mode (DDR to half speed)	248.48
	System Idle with display in LD mode [DDR to lowest speed with Software Fast Frequency Change (SWFFC)]	225.03
	System Idle without display in ND mode with DDRC auto clock gating	225.31
	System Idle without display in LD mode with DDRC auto clock gating (DDR to half speed)	167.75
	System Idle without display in LD mode with DDRC auto clock gating (DDR to lowest speed with SWFFC)	154.00
	System in DSM	8.18

Table 6. i.MX 91-EVK power summary report*...continued*

Use cases category	Use cases	Total power (sum of average powers in GROUP_SOC_FULL) (mW)
	Battery	0.109

5.1 Core benchmark use cases

The following use cases scenarios have been tested with the Cortex A55 core:

- Dhrystone
- CoreMark

5.1.1 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.

When the Dhrystone use case is running, the state of the system is as follows:

1. The CPU frequency is set to the maximum value of 1.4 GHz.
2. The DDR data rate is set to 2400 MT/s.
3. The display is OFF.

To measure the power consumption of Dhrystone, the steps are as follows:

1. Boot the Linux image with `imx91-11x11-evk.dtb`.
2. Run `setup.sh`. See Section 7.1.
3. Run `dhrystone_loop.sh`.

```
while true; do
    ./dhry2
done
```

4. Measure the power and record the results.

Table 7 shows the measurement results when this use case is applied to the i.MX 91 processor.

Table 7. Measurement results for i.MX 91-11x11-EVK_Dhrystone_loop (average value)

Rail label		Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (°C)
GROUP_DRAM	lpd4x_vdd1	1.8	1.91	3.43	16.38	32
	lpd4x_vdd2	1.1	11.78	12.92		
	lpd4x_vddq	1.1	0.03	0.03		
GROUP_SOC_FULL	nvcc_1p8	1.8	0.35	0.63	419.54	
	nvcc_3p3	3.3	2.47	8.14		
	nvcc_bbsm_1p8	1.81	0.12	0.21		
	nvcc_sd2	3.3	0.94	3.09		
	vdd2_ddr	1.1	13.7	15.05		
	vdd_ana_0p8	0.79	18.72	14.8		
	vdd_ana_1p8	1.8	8.81	15.83		
	vdd_soc	0.84	417.22	350.87		
	vdd_usb_3p3	3.3	0.06	0.19		
	vddq_ddr	1.1	9.78	10.73		

5.1.2 CoreMark

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

When the use case is running, the state of the system is as follows:

1. The CPU frequency is set to the maximum value of 1.4 GHz.
2. The DDR data rate is set to 2400 MT/s.
3. The display is OFF.

To measure the power consumption of CoreMark, the steps are as follows:

1. Boot the Linux image with `imx91-11x11-evk.dtb`.
2. Run the `setup.sh`. See Section 7.1.
3. Run `coremark_loop.sh`:

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

4. Measure the power and record the results.

Table 8 shows the measurement results when this use case is applied to the i.MX 91 processor.

Table 8. Measurement results for i.MX 91-11x11-EVK_CoreMark_loop (average value)

Rail label		Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (°C)
GROUP_DRAM	lpd4x_vdd1	1.80	1.96	3.52	16.93	32
	lpd4x_vdd2	1.10	12.15	13.33		
	lpd4x_vddq	1.10	0.07	0.08		
GROUP_SOC_FULL	nvcc_1p8	1.80	0.21	0.37	390.57	
	nvcc_3p3	3.30	2.44	8.06		
	nvcc_bbsm_1p8	1.81	0.12	0.22		
	nvcc_sd2	3.30	0.93	3.09		
	vdd2_ddr	1.10	13.83	15.20		
	vdd_ana_0p8	0.79	18.73	14.81		
	vdd_ana_1p8	1.80	8.82	15.85		
	vdd_soc	0.84	382.93	322.26		
	vdd_usb_3p3	3.30	0.05	0.16		
	vddq_ddr	1.10	9.62	10.56		

5.2 Memory use cases

The following memory-centric use case scenarios have been tested:

- Memset
- Memcpy
- Stream

The memset and memcpy are part of a `perf-bench`, which is a general framework for benchmark suites.

5.2.1 Memset

The 'Memset' use case is for evaluating the performance of a simple memory set in various ways. When the 'memset' use case is running, the state of the system is as follows:

1. The CPU frequency is set to the maximum value of 1.4 GHz.
2. The DDR data rate is set to 2400 MT/s.
3. The size for the memory buffers is set to 1024 MB.

To measure the power consumption of the memset, the steps are as follows:

1. Boot the Linux image with `imx91-11x11-evk.dtb`.
2. Run `setup.sh`. See Section 7.1.
3. 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
```

4. Measure the power and record the results.

Table 9 shows the measurement results when this use case is applied to the i.MX 91 processor.

Table 9. Measurement results for i.MX 91-11x11-EVK_memset_loop (average value)

Rail label		Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (°C)
GROUP_DRAM	lpd4x_vdd1	1.8	5.21	9.38	54.36	36
	lpd4x_vdd2	1.09	41.07	44.95		
	lpd4x_vddq	1.1	0.03	0.03		
GROUP_SOC_FULL	nvcc_1p8	1.8	0.24	0.43	641.74	
	nvcc_3p3	3.3	2.47	8.13		
	nvcc_bbsm_1p8	1.81	0.12	0.22		
	nvcc_sd2	3.3	0.94	3.09		
	vdd2_ddr	1.09	101.31	110.86		
	vdd_ana_0p8	0.79	18.8	14.86		
	vdd_ana_1p8	1.8	8.83	15.86		
	vdd_soc	0.84	486.83	409.19		
	vdd_usb_3p3	3.3	0.04	0.15		
	vddq_ddr	1.09	72.27	78.95		

5.2.2 memcpy

The 'memcpy' use case evaluates the performance of a simple memory copy in various ways. When this use case is running, the state of the system is as follows:

1. The CPU frequency is set to the maximum value of 1.4 GHz.
2. The DDR data rate is set to 2400 MT/s.
3. The size for the memory buffers is set to 1024 MB.

To measure the power consumption of 'memcpy', the steps are as follows:

1. Boot the Linux image with `imx91-11x11-evk.dtb`.
2. Run `setup.sh`. See Section 7.1.
3. 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
```

4. Measure the power and record the results.

Table 10 shows the measurement results when this use case is applied to the i.MX 91 processor.

Table 10. Measurement results for i.MX 91-11x11-EVK_memcpy_loop (average value)

Rail label		Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (°C)
GROUP_DRAM	lpd4x_vdd1	1.8	5.38	9.68	67.50	34
	lpd4x_vdd2	1.09	46.3	50.65		
	lpd4x_vddq	1.1	6.54	7.18		
GROUP_SOC_FULL	nvcc_1p8	1.8	0.27	0.49	495.78	
	nvcc_3p3	3.3	2.44	8.06		
	nvcc_bbsm_1p8	1.81	0.12	0.22		
	nvcc_sd2	3.31	0.94	3.09		
	vdd2_ddr	1.1	37.74	41.44		
	vdd_ana_0p8	0.79	18.76	14.83		
	vdd_ana_1p8	1.8	8.79	15.79		
	vdd_soc	0.84	457.4	384.53		
	vdd_usb_3p3	3.3	0.05	0.18		
	vddq_ddr	1.1	24.78	27.16		

5.2.3 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.

When the use case is running, the state of the system is as follows:

1. The CPU frequency is set to the maximum value of 1.4 GHz.
2. The DDR data rate is set to 2400 MT/s.

3. All phases, such as Copy, Scale, Add, and Triad, are included.

To measure the power consumption of the stream, the steps are as follows:

1. Boot the Linux image with `imx91-11x11-evk.dtb`.
2. Run `setup.sh`. See Section 7.1.
3. Run `streamcpy_loop.sh`:

```
while true; do
   stream -M 200M -N 1000
done
```

4. Measure the power and record the results.

Table 11 shows the measurement results when this use case is applied to the i.MX 91 processor.

Table 11. Measurement results for i.MX 91-11x11-EVK_stream_loop (average value)

Rail label		Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (°C)
GROUP_DRAM	lpd4x_vdd1	1.8	5.36	9.64	74.62	35
	lpd4x_vdd2	1.09	47.44	51.89		
	lpd4x_vddq	1.1	11.96	13.1		
GROUP_SOC_FULL	nvcc_1p8	1.8	0.24	0.43	550.81	
	nvcc_3p3	3.3	2.45	8.09		
	nvcc_bbsm_1p8	1.81	0.12	0.22		
	nvcc_sd2	3.3	0.94	3.09		
	vdd2_ddr	1.1	46.22	50.71		
	vdd_ana_0p8	0.79	18.78	14.85		
	vdd_ana_1p8	1.8	8.82	15.84		
	vdd_soc	0.84	505.61	424.88		
	vdd_usb_3p3	3.3	0.05	0.17		
	vddq_ddr	1.1	29.7	32.53		

5.3 Audio playback use cases

The following audio use case scenarios have been tested:

- Audio playback (gplay)
- Audio low-bus playback (aplay)

5.3.1 Audio full frequency

For this use case, the audio file is an MP3 file with a 128 kbit/s bit rate and a 44 kHz sample rate. CA55 handles audio decoding, I^2S, and audio codec.

When the use case is running, the state of the system is as follows:

1. The CPU frequency is set to the maximum value of 1.4 GHz.
2. The DDR data rate is set to 2400 MT/s.
3. The A55 core handles the audio decoding, I2S, and audio codec.
4. mp3 file with a 128 kbit/s bit rate and a 44 kHz sample rate/s .

To measure the power consumption of the audio playback, use the steps below:

1. Boot the Linux image with `imx91-11x11-evk.dtb`.
2. Run `setup.sh`. See Section 7.1.
3. Run `gplay_audio.sh`.

```
gplay-1.0 Mpeg1L3_44kHz_128kbps_s_Ed_Rush_Sabotage_mplayer.mp3
```

4. Measure the power and record the results.

Note: *Prepare your own MP3 file. To obtain similar results in this document, ensure that the audio bit rate is about 128 kbit/s. For audio playback with Gstreamer, you must enable the pipewire at the first boot after downloading the image:*

If there is no sound, use the following command to query and set the sound card.

```
systemctl --user --now enable pipewire wireplumber pipewire-pulse
      wpctl status          #show audio card status and ID
      wpctl inspect $ID     #show audio card profile
      wpctl set-default $ID #set default audio source with the wm8962
```

Table 12 shows the measurement results when this use case is applied to the i.MX 91 processor.

Table 12. Measurement results for i.MX 91-11x11-EVK_gplay_audio-default (average value)

Rail label		Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (°C)
GROUP_DRAM	lpd4x_vdd1	1.8	2.09	3.76	19.17	33
	lpd4x_vdd2	1.1	13.49	14.79		
	lpd4x_vddq	1.1	0.57	0.62		
GROUP_SOC_FULL	nvcc_1p8	1.8	0.16	0.29	301.30	
	nvcc_3p3	3.3	4.91	16.18		
	nvcc_bbsm_1p8	1.81	0.12	0.22		
	nvcc_sd2	3.3	0.94	3.09		

Table 12. Measurement results for i.MX 91-11x11-EVK_gplay_audio-default (average value)*...continued*

Rail label		Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (°C)
	vdd2_ddr	1.1	14.1	15.5		
	vdd_ana_0p8	0.79	22.84	18.02		
	vdd_ana_1p8	1.8	9.99	17.94		
	vdd_soc	0.84	259.85	218.94		
	vdd_usb_3p3	3.3	0.05	0.17		
	vddq_ddr	1.1	9.99	10.96		

5.3.2 Audio low-bus playback (aplay)

For this use case, the audio file is a WAV file with a 24 bit and a 32 kHz sample rate. Arm Cortex-A55 processor (CA55) handles audio decoding, I^2S, and audio codec.

When this use case is running, the state of the system is as follows:

1. The CPU frequency is set to 0.9 GHz.
2. The DDR data rate is set to 625 MT/s.

To measure the power consumption of the audio low-bus playback, the steps are as follows:

1. Boot the Linux image with `imx91-11x11-evk-ld.dtb`.
2. Run `DDRC_625MTS_setup.sh` (625 MT/s data rate). See Section 7.4.
3. Run `aplay_audio.sh`:

```
while true; do
   aplay -D hw:wm8962audio audio32k24b2c.wav
done
```

4. Measure the power and record the results.

Note: *Prepare your own MP3 file. To obtain similar results in this document, ensure that the audio bit rate is about 128 kbit/s.*

Table 13 shows the measurement results when this use case is applied to the i.MX 91 processor.

Table 13. Measurement results for i.MX 91-11x11-EVK_audio_low_power_50 MHz (average value)

Rail label		Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (°C)
GROUP_DRAM	lpd4x_vdd1	1.8	2.1	3.77	20.44	30
	lpd4x_vdd2	1.1	13.17	14.45		
	lpd4x_vddq	1.1	2.02	2.22		
GROUP_SOC_FULL	nvcc_1p8	1.8	0.22	0.39	197.11	
	nvcc_3p3	3.3	4.93	16.28		
	nvcc_bbsm_1p8	1.81	0.12	0.21		
	nvcc_sd2	3.3	0.93	3.09		
	vdd2_ddr	1.1	11.29	12.41		
	vdd_ana_0p8	0.79	22.19	17.51		
	vdd_ana_1p8	1.8	5.82	10.47		
	vdd_soc	0.8	161.11	128.53		
	vdd_usb_3p3	3.3	0.05	0.15		
	vddq_ddr	1.1	7.36	8.08		

5.4 Storage use cases

The tested use case scenarios for storage are as follows:

- DD_WRITE_eMMC
- DD_READ_eMMC
- DD_WRITE_SD
- DD_READ_SD

5.4.1 DD_WRITE_eMMC

When the `DD_WRITE_eMMC` use case is running, the state of the system is as follows:

1. Boot the Linux image with `imx91-11x11-evk.dtb`.
2. Run `setup.sh`. Refer Section 7.1.
3. Copy `dd_write.sh` on the eMMC partition and run. Refer Section 7.6.
4. Measure the power and record the results.

To measure the power consumption of DD_WRITE_eMMC, the steps are as follows:

1. The CPU frequency is set to the maximum value of 1.4 GHz.
2. The DDR data rate is set to 2400 MT/s.
3. The maximum amount of data the kernel reads ahead for a single file is set to 512 kB.

Table 14 shows the measurement results when this use case applies to the i.MX 91 processor.

Table 14. Measurement results for i.MX 91-11x11-EVK_DD_WRITE_eMMC (average value)

Rail label		Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (°C)
GROUP_DRAM	lpd4x_vdd1	1.8	2.15	3.87	19.89	32
	lpd4x_vdd2	1.1	14.17	15.54		
	lpd4x_vddq	1.1	0.43	0.48		
GROUP_SOC_FULL	nvcc_1p8	1.8	7.01	12.61	318.67	
	nvcc_3p3	3.3	2.35	7.74		
	nvcc_bbsm_1p8	1.81	0.12	0.21		
	nvcc_sd2	3.3	0.94	3.09		
	vdd2_ddr	1.1	14.78	16.24		
	vdd_ana_0p8	0.79	18.8	14.86		
	vdd_ana_1p8	1.79	8.81	15.81		
	vdd_soc	0.84	280.67	236.45		
	vdd_usb_3p3	3.3	0.05	0.18		
	vddq_ddr	1.1	10.46	11.47		

5.4.2 DD_READ_eMMC

The state of the system, when the use case is running, is as follows:

1. The CPU frequency is set to the maximum value of 1.4 GHz.
2. The DDR data rate is set to 2400 MT/s.
3. The maximum amount of data the kernel reads ahead for a single file is set to 512 kB.

To measure the power consumption of DD_READ_eMMC, the steps are as follows:

1. Boot the Linux image with `imx91-11x11-evk.dtb`.
2. Run `setup.sh`. Refer Section 7.1.
3. Make sure the file `dd_obs_testfile` exists and rename it to `dd_ibs_testfile`.
4. Copy `dd_read.sh` on the eMMC partition and run. Refer Section 7.5.
5. Measure the power and record the results.

Table 15 shows the measurement results when this use case applies to the i.MX 91 processor.

Table 15. Measurement results for i.MX 91-11x11-EVK_DD_READ_eMMC (average value)

Rail label		Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (°C)
GROUP_DRAM	lpd4x_vdd1	1.80	3.60	6.47	45.06	34
	lpd4x_vdd2	1.10	29.85	32.69		
	lpd4x_vddq	1.10	5.38	5.90		
GROUP_SOC_FULL	nvcc_1p8	1.80	7.20	12.93	415.72	
	nvcc_3p3	3.30	2.40	7.91		
	nvcc_bbsm_1p8	1.81	0.12	0.22		
	nvcc_sd2	3.31	0.94	3.10		
	vdd2_ddr	1.10	23.12	25.41		
	vdd_ana_0p8	0.79	18.92	14.96		
	vdd_ana_1p8	1.79	8.81	15.80		
	vdd_soc	0.84	378.74	318.76		
	vdd_usb_3p3	3.30	0.05	0.16		
	vddq_ddr	1.10	15.03	16.48		

5.4.3 DD_WRITE_SD

When the use case is running, the state of the system is as follows:

1. The CPU frequency is set to the maximum value of 1.4 GHz.
2. The DDR data rate is set to 2400 MT/s.
3. The maximum amount of data the kernel reads ahead for a single file is set to 512 kB.

To measure the power consumption of DD_WRITE_SD, the steps are as follows:

1. Boot the Linux image with `imx91-11x11-evk.dtb`.
2. Run `setup.sh`. Refer Section 7.1.
3. Copy `dd_write.sh` on the SD partition and run. Refer Section 7.6.
4. Measure the power and record the results.

Table 16 shows the measurement results when this use case applies to the i.MX 91 processor.

Table 16. Measurement results for i.MX 91-11x11-EVK_DD_WRITE_SD10 (average value)

Rail label		Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (°C)
GROUP_DRAM	lpd4x_vdd1	1.8	2.38	4.28	21.64	34
	lpd4x_vdd2	1.1	15.1	16.56		
	lpd4x_vddq	1.1	0.73	0.8		
GROUP_SOC_FULL	nvcc_1p8	1.8	0.41	0.74	425.15	
	nvcc_3p3	3.29	28.53	94		
	nvcc_bbsm_1p8	1.81	0.12	0.22		
	nvcc_sd2	1.79	14.52	25.93		
	vdd2_ddr	1.1	15.17	16.67		
	vdd_ana_0p8	0.79	18.84	14.89		
	vdd_ana_1p8	1.8	8.82	15.85		
	vdd_soc	0.84	291.23	245.3		
	vdd_usb_3p3	3.3	0.04	0.12		
	vddq_ddr	1.1	10.42	11.43		

5.4.4 DD_READ_SD

When the use case is running, the state of the system is as follows:

1. The CPU frequency is set to the maximum value of 1.4 GHz.
2. The DDR data rate is set to 2400 MT/s.
3. The maximum amount of data the kernel reads ahead for a single file is set to 512 kB.

To measure the power consumption of DD_READ_SD, the steps are as follows:

1. Boot the Linux image with `imx91-11x11-evk.dtb`.
2. Run `setup.sh`. Section 7.1.
3. Make sure the file `dd_obs_testfile` exists and rename it to `dd_ibs_testfile`.
4. Copy `dd_read.sh` on the SD partition and run. Refer Section 7.5.
5. Measure the power and record the results.

Table 17 shows the measurement results when this use case applies to the i.MX 91 processor.

Table 17. Measurement results for i.MX 91-11x11-EVK_DD_READ_SD10 (average value)

Rail label		Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (°C)
GROUP_DRAM	lpd4x_vdd1	1.8	2.66	4.78	28.88	35
	lpd4x_vdd2	1.1	19.72	21.62		
	lpd4x_vddq	1.1	2.26	2.48		
GROUP_SOC_FULL	nvcc_1p8	1.8	0.34	0.61	442.48	
	nvcc_3p3	3.29	28.54	94.02		
	nvcc_bbsm_1p8	1.81	0.12	0.22		
	nvcc_sd2	1.79	14.88	26.58		
	vdd2_ddr	1.1	17.51	19.25		
	vdd_ana_0p8	0.79	18.89	14.93		
	vdd_ana_1p8	1.8	8.8	15.81		
	vdd_soc	0.84	306.02	257.74		
	vdd_usb_3p3	3.3	0.04	0.14		
	vddq_ddr	1.1	12.01	13.18		

5.5 Low-power mode use cases

The following Low-power mode use case scenarios have been tested:

- System Idle with display in Normal Drive mode
- System Idle with display in Low Drive mode (DDR to half speed)
- System Idle with display in Low Drive mode (DDR to lowest speed with SWFFC)
- System Idle without display in Normal Drive mode with DDRC auto clock gating
- System Idle without display in Low Drive mode with DDRC auto clock gating (DDR to half speed)
- System Idle without display in Low Drive mode with DDRC auto clock gating (DDR to lowest speed with SWFFC)
- System in Deep Sleep Mode
- Battery

5.5.1 System Idle with display in Normal Drive (ND) mode

When the use case is running, the state of the system is as follows:

1. The CPU default frequency is set to 1.4 GHz.
2. The DDR data rate is set to 2400 MT/s.

To measure the power consumption for the system Idle with display on ND mode with DDRC auto clock gating, the steps are as follows:

1. Connect the TM050RDH03-41 Parallel LCD Display to the board on J1001.
2. Boot the Linux image with `imx91-11x11-evk-tianma-wvga-panel.dtb`.
3. Run `setup_video.sh`. Refer Section 7.2.
4. The default mode is the ND mode.
5. Measure the power and record the results.

Table 18 shows the measurement results when this use case is applied to the i.MX 91 processor.

Table 18. Measurement results for i.MX 91-11x11-EVK_System_idle_w_display_on_ND_mode_DDRC_auto_clock_gating (average value)

Rail label		Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (°C)
GROUP_DRAM	lpd4x_vdd1	1.8	2.1	3.78	20.43	33
	lpd4x_vdd2	1.1	14.03	15.39		
	lpd4x_vddq	1.1	1.15	1.26		
GROUP_SOC_FULL	nvcc_1p8	1.8	0.25	0.46	332.21	
	nvcc_3p3	3.3	15.78	52		
	nvcc_bbsm_1p8	1.81	0.12	0.21		
	nvcc_sd2	3.31	0.94	3.09		
	vdd2_ddr	1.1	14.44	15.88		
	vdd_ana_0p8	0.79	22.41	17.68		
	vdd_ana_1p8	1.8	9.88	17.74		
	vdd_soc	0.84	254.04	214.18		
	vdd_usb_3p3	3.3	0.05	0.17		
	vddq_ddr	1.1	9.84	10.8		

5.5.2 System Idle with display in Low Drive (LD) mode (DDR to half speed)

When the `System Idle with display in LD mode (DDR to half speed)` use case is running, the state of the system is as follows:

1. The CPU default frequency is set to 0.9 GHz.
2. The DDR frequency is set to 1200 MT/s.

To measure the power consumption for the system Idle with display on LD mode with DDRC auto clock gating, DDR to half speed, the steps are as follows:

1. Connect the KD50G21-40NT-A1 display to the board on J1001.
2. Boot the Linux image with `imx91-11x11-evk-tianma-wvga-panel-ld.dtb`.
3. Run `setup_video.sh`. Refer Section 7.2.
4. To put the system into the LD mode (DDR to half speed), run the following command:

```
echo 2 > /sys/devices/platform/imx93-lpm/mode
```

5. Measure the power and record the results.

Table 19 shows the measurement results when this use case is applied to the i.MX 91 processor.

Table 19. Measurement results for i.MX 91-11x11-EVK_System_idle_w_display_on_LD_mode_half_speed_DDR (average value)

Rail label		Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (°C)
GROUP_DRAM	lpd4x_vdd1	1.8	2.11	3.81	21.57	32
	lpd4x_vdd2	1.1	13.92	15.27		
	lpd4x_vddq	1.1	2.27	2.5		
GROUP_SOC_FULL	nvcc_1p8	1.8	0.24	0.43	248.48	
	nvcc_3p3	3.3	15.71	51.8		
	nvcc_bbsm_1p8	1.81	0.12	0.22		
	nvcc_sd2	3.31	0.93	3.09		
	vdd2_ddr	1.1	12.8	14.07		
	vdd_ana_0p8	0.79	21.78	17.19		
	vdd_ana_1p8	1.8	7.56	13.59		
	vdd_soc	0.8	174.21	138.99		
	vdd_usb_3p3	3.3	0.04	0.14		
	vddq_ddr	1.1	8.18	8.97		

5.5.3 System Idle with display in LD mode (DDR to lowest speed with SWFFC)

When the use case is running, the state of the system is as follows:

1. The CPU default frequency is set to 0.9 GHz.
2. The DDR frequency is set to 625 MT/s.

To measure the power consumption for the system Idle with display on LD mode (DDR to lowest speed with SWFFC), the steps are as follows:

1. Connect the TM050RDH03-41 Parallel LCD Display to the board on J1001.
2. Boot the Linux image with `imx91-11x11-evk-tianma-wvga-panel-ld.dtb`.
3. Run `setup_video.sh`. Refer Section 7.2.
4. To put the system into the LD mode, DDR to the lowest speed with SWFFC, run the following command:

```
echo 3 > /sys/devices/platform/imx93-lpm/mode
```

5. Measure the power and record the result.

Table 20 shows the measurement results when this use case is applied to the i.MX 91 processor.

Table 20. Measurement results for i.MX 91-11x11-EVK_System_idle_w_display_on_LD_mode_lowest_speed_DDR_SWFFC (average value)

Rail label		Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (°C)
GROUP_DRAM	lpd4x_vdd1	1.8	2.19	3.94	23.67	31
	lpd4x_vdd2	1.1	13.95	15.3		
	lpd4x_vddq	1.1	4.03	4.42		
GROUP_SOC_FULL	nvcc_1p8	1.8	0.23	0.42	225.03	
	nvcc_3p3	3.3	15.69	51.72		
	nvcc_bbsm_1p8	1.81	0.12	0.21		
	nvcc_sd2	3.31	0.94	3.09		
	vdd2_ddr	1.1	11.74	12.91		
	vdd_ana_0p8	0.79	21.77	17.18		
	vdd_ana_1p8	1.8	5.72	10.29		
	vdd_soc	0.8	152.12	121.41		
	vdd_usb_3p3	3.3	0.05	0.16		
	vddq_ddr	1.1	6.96	7.63		

5.5.4 System Idle without display in ND mode with DDRC auto clock gating

When the use case is running, the state of the system is as follows:

- The CPU default frequency is set to 1.4 GHz.
- The DDR data rate is set to 2400 MT/s.

To measure the power consumption of a system Idle without display on ND mode with DDRC auto clock gating, the steps are as follows:

1. Boot the Linux image with `imx91-11x11-evk.dtb`.
2. Run `setup.sh`. Refer Section 7.1.
3. The default mode is the ND mode.
4. To enable `auto_clk_gating`, run the following command:

```
echo 256 > /sys/devices/platform/imx93-lpm/auto_clk_gating
```

   Where `256` implies the duration for DDR Idle to enter self; the unit is the clock cycle.
5. Measure the power and record the results.

Table 21 shows the measurement results when this use case is applied to the i.MX 91 processor.

Table 21. Measurement results for i.MX 91-11x11-EVK_system_idle_w/o_display_on_ND_mode_DDRC_auto_clock_gating (average value)

Rail label		Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (°C)
GROUP_DRAM	lpd4x_vdd1	1.8	0.4	0.73	4.88	29
	lpd4x_vdd2	1.1	3.79	4.16		
	lpd4x_vddq	1.1	-0.01	-0.01		
GROUP_SOC_FULL	nvcc_1p8	1.8	0.2	0.36	225.31	
	nvcc_3p3	3.3	2.42	7.98		
	nvcc_bbsm_1p8	1.81	0.12	0.21		
	nvcc_sd2	3.3	0.94	3.09		
	vdd2_ddr	1.1	13.57	14.91		
	vdd_ana_0p8	0.79	18.68	14.77		
	vdd_ana_1p8	1.8	8.78	15.76		
	vdd_soc	0.84	186.98	157.67		
	vdd_usb_3p3	3.3	0.05	0.18		
	vddq_ddr	1.1	9.46	10.38		

5.5.5 System Idle without display in LD mode with DDRC auto clock gating (DDR to half speed)

When the System Idle without display in LD mode with DDRC auto clock gating (DDR to half speed) use case is running, the state of the system is as follows:

- The CPU default frequency is set to 0.9 GHz.
- The DDR data rate is set to 1200 MT/s.

To measure the power consumption for the system Idle without display on LD mode with DDRC auto clock gating (DDR to half speed), the steps are as follows:

1. Boot the Linux image with `imx91-11x11-evk-ld.dtb`.
2. Run `setup.sh`. Refer Section 7.1.
3. To put the system into the LD mode (DDR to half speed), run the following command:

```
echo 2 > /sys/devices/platform/imx93-lpm/mode
```

4. To enable `auto_clk_gating`, run the following command:

```
echo 256 > /sys/devices/platform/imx93-lpm/auto_clk_gating
```

   Where `256` implies the duration for DDR Idle to enter self; the unit is the clock cycle.
5. Measure the power and record the results.

Table 22 shows the measurement results when this use case is applied to the i.MX 91 processor.

Table 22. Measurement results for i.MX 91-11x11-EVK_system_idle_w/o_display_on_LD_mode_DDRC_auto_clock_gating_half_speed_DDR (average value)

Rail label		Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (°C)
GROUP_DRAM	lpd4x_vdd1	1.8	0.35	0.62	4.88	28
	lpd4x_vdd2	1.1	3.83	4.2		
	lpd4x_vddq	1.1	0.05	0.05		
GROUP_SOC_FULL	nvcc_1p8	1.8	0.25	0.45	167.75	
	nvcc_3p3	3.3	2.43	8.01		
	nvcc_bbsm_1p8	1.81	0.12	0.21		
	nvcc_sd2	3.3	0.93	3.09		
	vdd2_ddr	1.1	11.11	12.21		
	vdd_ana_0p8	0.79	18.06	14.28		
	vdd_ana_1p8	1.8	6.48	11.64		
	vdd_soc	0.8	136.68	109.05		
	vdd_usb_3p3	3.3	0.05	0.15		
	vddq_ddr	1.1	7.9	8.66		

5.5.6 System Idle without display in LD mode with DDRC auto clock gating (DDR to lowest speed with SWFFC)

When the System Idle without display in LD mode with DDRC auto clock gating (DDR to lowest speed with SWFFC) use case is running, the state of the system is as follows:

- The CPU default frequency is set to 0.9 GHz.
- The DDR data rate is set to 625 MT/s.

To measure the power consumption for the system Idle without display on LD mode with DDRC auto clock gating (DDR to lowest speed with SWFFC), the steps are as follows:

1. Boot the Linux image by using `imx91-11x11-evk-ld.dtb`.
2. Run `setup.sh`. Refer Section 7.1.
3. To put the system into the LD mode, DDR to the lowest speed with SWFFC, run the following command:

```
echo 3 > /sys/devices/platform/imx93-lpm/mode
```

4. To enable `auto_clk_gating`, run the following command:

```
echo 256 > /sys/devices/platform/imx93-lpm/auto_clk_gating
```

   Where `256` implies the duration for DDR Idle to enter self; the unit is the clock cycle.
5. Measure the power and record the results.

Table 23 shows the measurement results when this use case is applied to the i.MX 91 processor.

Table 23. Measurement results for i.MX 91-11x11-EVK_system_idle_w/o_display_on_LD_mode_DDRC_auto_clock_gating_lowest_speed_DDR_SWFCC (average value)

Rail label		Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (°C)
GROUP_DRAM	lpd4x_vdd1	1.8	0.4	0.71	4.82	28
	lpd4x_vdd2	1.1	3.6	3.95		
	lpd4x_vddq	1.1	0.14	0.15		
GROUP_SOC_ FULL	nvcc_1p8	1.8	0.23	0.42	154.00	
	nvcc_3p3	3.3	2.53	8.35		
	nvcc_bbsm_ 1p8	1.81	0.12	0.21		
	nvcc_sd2	3.3	0.93	3.09		
	vdd2_ddr	1.1	9.07	9.97		
	vdd_ana_0p8	0.79	18.05	14.27		
	vdd_ana_1p8	1.8	4.62	8.3		
	vdd_soc	0.8	128.24	102.35		
	vdd_usb_3p3	3.3	0.05	0.17		
	vddq_ddr	1.1	6.28	6.89		

5.5.7 System in DSM

The System in DSM use case is based on the Suspend mode, which implies the following:

- CA55 cluster is OFF
- MEDIAMIX is OFF
- NICMIX is OFF
- WAKEUPMIX is OFF
- PLL is OFF
- 24 M OSC is OFF
- PMIC is in STBY mode
- The DDR is in the Retention mode.

To measure the power consumption of the system in the DSM, the steps are as follows:

1. Boot the Linux image with `imx91-11x11-evk-DSM.dtb`.
2. To put the system into the Suspend (Deep sleep) mode, run the following command:

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

3. Measure the power and record the results.

Table 24 shows the measurement results when this use case is applied to the i.MX 91 processor.

Table 24. Measurement results for i.MX 91-11x11-EVK_DSM (average value)

Rail label		Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (mW)	Zone 0 die temperature (°C)
GROUP_DRAM	lpd4x_vdd1	1.8	0.32	0.57	1.38	Die temperature cannot be measured as the CA55 core is in suspended state.
	lpd4x_vdd2	1.1	0.76	0.84		
	lpd4x_vddq	1.1	-0.02	-0.03		
GROUP_SOC_FULL	nvcc_1p8	1.8	0.49	0.88	8.18	
	nvcc_3p3	3.3	0.5	1.66		
	nvcc_bbsm_1p8	1.81	0.1	0.18		
	nvcc_sd2	3.31	0.24	0.81		
	vdd2_ddr	1.1	0.1	0.11		
	vdd_ana_0p8	0.8	0.59	0.47		
	vdd_ana_1p8	1.8	0.52	0.93		
	vdd_soc	0.65	4.46	2.9		
	vdd_usb_3p3	3.3	0.06	0.19		
	vddq_ddr	1.1	0.04	0.05		

5.5.8 Battery

BBSM mode of the i.MX 91 processor is a Low-power mode where only the power for the BBSM domain remains on.

When the use case is running, the state of the system is as follows:

- All power supplies except NVCC_BBSM_1P8 are off externally.
- 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 CA55 and LPAV are turned off.

To configure and run the use case, follow the steps below:

1. Boot the Linux image with `imx91-11x11-evk.dtb` in CA55.
2. Press the ON/OFF key for 5 seconds.
3. Measure the power and record the results.

Table 25 shows the measurement results when this use case is applied to the i.MX 91 processor.

Table 25. Measurement results for i.MX 91-11x11-EVK_Battery (average value)

Rail label		Average voltage (V)	Average current (mA)	Average power (mW)	Sum of average powers (µW)	Zone 0 die temperature (°C)
GROUP_DRAM	lpd4x_vdd1	0	0	0	0.00	Die temperature cannot be measured as the CA55 core has been suspended.
	lpd4x_vdd2	0	0	0		
	lpd4x_vddq	0	0	0		
GROUP_SOC_FULL	nvcc_1p8	0.00	-0.02	0.00	109	
	nvcc_3p3	0.00	-0.02	0.00		
	nvcc_bbsm_1p8	1.81	0.06	0.11		
	nvcc_sd2	0.00	0.00	0.00		
	vdd2_ddr	0.00	0.00	0.00		
	vdd_ana_0p8	0.00	-0.01	0.00		
	vdd_ana_1p8	0.00	-0.01	0.00		
	vdd_soc	0.00	0.00	0.00		
	vdd_usb_3p3	0.00	0.00	0.00		
	vddq_ddr	0.00	0.00	0.00		

6 Reducing power consumption

The overall system power consumption depends on the software optimization and the system hardware implementation. The following list of suggestions can help reduce system power consumption. Some of these suggestions are already implemented in the Linux BSP and/or SDK. The system of each individual user can undergo further optimizations.

Note: *Further power optimizations are planned in future software releases. To obtain the latest software releases, refer i.MX Software and Development Tools.*

- Apply clock gating by configuring registers in the CCM, whenever clocks or modules are not used.
- For run modes, use the slowest frequency that can still meet the application requirements.
- Minimize the number of operating PLLs. Enabled PLLs can consume a few milliamps of current.
- Applying voltage and frequency scaling (VFS) for the Arm cores and scaling the frequencies of the AXI, AHB, and IPG bus clocks can significantly reduce power consumption. However, the operation frequency reduction causes longer access times to the DDR, which increases the power consumption of the DDR I/O and memories. Consider this trade-off for each mode to quantify the overall effect on system power.
- Put the SoC into Low-power modes whenever possible, as long as it can still support the application requirements. Consider the following example:
  - Put the system into Suspend mode when it can enter deep sleep.
  - Power off the CA55 cores and other domains for low-load use cases.
- For each operating mode, use the lowest voltage (with the power supply tolerance) that can still meet the requirements of voltage specifications in the data sheet.
- DDR interface optimization:
  - Use careful board routing of the DDR memories, maintaining PCB trace lengths as short as possible.
  - Use the proper 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.
  - Use of LPDDR4 memory offerings in the latest process technology can significantly reduce the power consumption of the DDR devices and the DDR I/O.

The following sections provide more details for system optimization. These sections are not an exhaustive list of features that can provide power reduction. However, these features are the easiest and the most common ones to use.

- Run fast and idle
- Clock gating
- DDRC auto clock gating
- PLL reduction
- Core VFS and system bus scaling
- Lower DDR frequencies
- DDR interface optimization
- Power gating of PHYs
- Distribution of workloads
- Use OCRAM to minimize DDR access
- Thermal management to reduce leakage

6.1 Run fast and idle

NXP testing and various research have shown that for most customer use cases, the best power/energy management protocol is to run the cores at maximum speeds for the workload and then drop to the lowest power mode as soon as possible. This strategy cannot provide optimal energy savings for the use cases where constant data is being processed, for example, low-latency audio playback. However, this strategy does work for other standard workloads. Consider this trade-off for each application to quantify the overall effect on the system power/energy consumption.

Users must place the i.MX 91 into the Low-power mode as far as possible.

6.2 Clock gating

The CCM inside the i.MX 91 provides a programmable method to disable the clock sources for modules when the modules are not used. To reduce energy waste, always configure the CCM registers. Driving any inactive signal, whether on the SoC or the PCB, is simply charging and discharging the line and the load capacitance of this signal. The NXP BSP-released software implements clock gating by default.

6.3 DDRC auto clock gating

When the bus is idle after the number of cycles configured in the `ssi_idle_strap` field in the DDR BLK_CTRL module, the DDRC does auto clock gating to save power. This feature can be used to balance DDR subsystem performance and power significantly. The number of idle cycles before clock gating can be adjusted dynamically based on the actual use case to fine-tune the power saving.

In the i.MX 91, `auto_clk_gating` parameter is used to enable the DDRC auto clock gating. Therefore, power is saved when there is no access to the DDR after the programmed idle count expires. "`Write 0`" disables the auto clock gating and the "`write non-zero`" value sets the `ssi_idle_strip` to this non-zero value and enables the auto clock gating. A value < 256 has some significant side effort for DDR performance, so a value >=256 is suggested when the user wants to enable it. When the auto clock gating is enabled, a high-resolution display can flicker at lower DDR frequency. It is recommended not to adjust the `auto_clk_gating` parameter when the display is running.

6.4 PLL reduction

Each PLL block consumes significant energy when active. Each application has unique requirements, but, if possible, reduce the number of operating PLLs. The CCM within the i.MX 91 provides Root Clock mux and programmable control to each PLL either by direct control mode or CPU Low-power mode. As a result, the Root Clocks source is allowed to modify to limit the PLL source and reduce the number of active PLLs when operating. Ensure that the application considers the PLL relock time when transitioning back to full operation.

6.5 Core VFS and system bus scaling

Applying VFS for the Arm cores and scaling (not dynamic) the frequencies of the NOC, AXI, AHB, and IPG system bus clocks can significantly reduce the power consumption of the VDD_SOC domains. However, the operation of system frequency reduction causes longer access times to the DDR, which can increase the energy consumption for specific use cases. Consider this trade-off for each mode to quantify the overall effect on the system power consumption.

6.6 Lowering DDR frequencies

As explained previously, the DDR I/O bus frequency also contributes to the DDR I/O current. Software interfaces allow for the use of the Hardware Fast Frequency Change (HWFFC) or Software Fast Frequency

Change (SWFFC) technology of DDRC. Using these techniques allow significant reduction of power consumption by lowering the DDR frequency.

6.7 DDR interface optimization

To optimize the DDR interface, the suggestions are as follows:

- Employ careful board routing of the DDR memories, maintaining the PCB trace lengths as short as possible. Longer trace lengths and more vias create more PCB capacitance for the signal, resulting in more energy wastage along the signal path.
- Keep the on-die termination (ODT) value as low as possible. The termination used greatly influences the power consumption of the DDR interface pins. To ensure the ODT variance does not reduce the bus signal integrity, simulate the DDR interface.
- Use an appropriate output driver impedance for the DDR interface pins that provide 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 must 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.
- Sizing DDR memory is important. If you select 4 GB memory when only 2 GB is used, you are wasting the refresh current for the unused 2 GB of DDR.
- Sizing of ECC DDR regions is important as they use more energy for this feature.

6.8 Power gating of PHYs

The PHYs of unused modules often get overlooked when searching for power savings. Many PHYs contain local PLLs or clocking circuits and voltage references, which consume power even when not in use. As a result, high-speed PHYs like Ethernet, MIPI, and USB get affected.

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. By spreading the workload, the system can return to the "Idle" state sooner.

System designers must ensure that the design uses the optimal cores for the specific workloads or tasks, on the i.MX 91 for maximum efficiency. While often easier said than done, it does provide significant power savings if the system can return to the low-power state faster (run fast and idle).

6.10 Use OCRAM to minimize DDR access

To achieve significant power savings, load highly accessed code into the On-Chip RAM (OCRAM). As a result, both the i.MX 91 and the DDR memory current consumption reduces. Another advantage of using the OCRAM is a performance increase since DDR memory access time delays this code.

6.11 Thermal management to reduce leakage

Thermal management is also a key element of power reduction. As temperature increases, so does the SoC gate leakage current for each gate within the device. Millions of high-gate leakages add up when looking for the lowest power consumption. As explained earlier, with any power savings, the temperature of the SoC reduces, and the lifetime reliability of the device improves.

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

7 Configuration_script and important commands

Before running a use case, the <configuration_script>.sh script must be run to configure the environment. A detailed description of these scripts is as follows:

7.1 setup.sh

The CPU frequency is set to the maximum value of 1.4 GHz to achieve the best performance. Disable the Ethernet, stop the Weston service, and blank the display. Set 512 kB as the maximum amount of data the kernel reads ahead for a single file.

```
#!/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
```

7.2 setup_video.sh

The CPU frequency is set to the maximum value of 1.4 GHz to achieve the best performance. Disable the Ethernet and awaken the display. Set 512 kB as the maximum amount of data the kernel reads ahead for a single file.

```
#!/bin/bash
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
```

7.3 setup_video_stream.sh

The CPU frequency is set to the maximum value of 1.4 GHz to achieve the best performance. To play the video online, open the Ethernet and awaken the display. Set 512 kB as the maximum amount of data the kernel reads ahead for a single file.

```
#!/bin/bash
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
```

7.4 DDRC_625MTS_setup.sh

: After running the shell scripts below, the DDR frequency switches to Low-bus mode 312.5 MHz (data rate is 625 MT/s). The CPU frequency is set to the minimum value of 1400 MHz. DDR VFS aims at saving power. Disable the Ethernet, stop the Weston service, and blank the display.

```
#!/bin/bash
systemctl stop weston.service
if [ -f /sys/class/graphics/fb0/blank ];
then echo 1 > /sys/class/graphics/fb0/blank
fi
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 3 > /sys/devices/platform/imx93-lpm/mode
```

7.5 dd_read.sh

This script is used to run the `dd read` command on the memory device.

```
#!/bin/bash
# Since we're dealing with dd, abort if any errors occur
set -e
TEST_FILE=${1:-dd_ibs_testfile}
if [ $EUID -ne 0 ]; then
echo "NOTE: Kernel cache will not be cleared between tests without sudo. This
 will
 likely cause inaccurate results." 1>&2 ;fi
count=$COUNT conv=fsync > /dev/null 2>&1
# Header
PRINTF_FORMAT="%8s : %s\n"
printf "$PRINTF_FORMAT" 'block size' 'transfer rate'
while true
BLOCK_SIZE=4096
do
# Clear kernel cache to ensure more accurate test
[ $EUID -eq 0 ] && [ -e /proc/sys/vm/drop_caches ] && echo 3 > /proc/sys/vm/
drop_caches
# Read test file out to /dev/null with specified block size
DD_RESULT=$(dd if=$TEST_FILE of=/dev/null bs=$BLOCK_SIZE 2>&1 1>/dev/null)
# Extract transfer rate
TRANSFER_RATE=$(echo $DD_RESULT | \grep --only-matching -E '[0-9.]+ ([MGk]?B|
bytes)/
s(ec)?')
printf "$PRINTF_FORMAT" "$BLOCK_SIZE" "$TRANSFER_RATE"
done
```

7.6 dd_write.sh

This script is used to run the `dd write` command on the memory device.

```
#!/bin/bash
# Since we're dealing with dd, abort if any errors occur
set -e
TEST_FILE=${1:-dd_obs_testfile}
TEST_FILE_EXISTS=0
if [ -e "$TEST_FILE" ]; then TEST_FILE_EXISTS=1; fi
TEST_FILE_SIZE=1024000000
if [ $EUID -ne 0 ]; then
echo "NOTE: Kernel cache will not be cleared between tests without sudo. This
 will
 likely cause inaccurate results." 1>&2
fi
# Header
PRINTF_FORMAT="%8s: %s\n"
printf "$PRINTF_FORMAT" 'block size' 'transfer rate'
while true
BLOCK_SIZE=4096
do
# Calculate number of segments required to copy
COUNT=$(($TEST_FILE_SIZE / $BLOCK_SIZE))
if [ $COUNT -le 0 ]; then
echo "Block size of $BLOCK_SIZE estimated to require $COUNT blocks, aborting
 further
 tests."
break
fi
# Clear kernel cache to ensure more accurate test
[ $EUID -eq 0 ] && [ -e /proc/sys/vm/drop_caches ] && echo 3 > /proc/sys/vm/
drop_caches
# Create a test file with the specified block size
DD_RESULT=$(dd if=/dev/zero of=$TEST_FILE bs=$BLOCK_SIZE count=$COUNT conv=fsync
 2>&1
 1>/dev/null)
# Extract the transfer rate from dd's STDERR output
TRANSFER_RATE=$(echo $DD_RESULT | \grep --only-matching -E '[0-9.]+ ([MGk]?B|
bytes)/
s(ec)?')
# Output the result
printf "$PRINTF_FORMAT" "$BLOCK_SIZE" "$TRANSFER_RATE"
done
```

8 Related documentation

For the list of current documents, refer to i.MX 91/Documentation. Additional literature would be published as and when new NXP products become available.

- For actual values of SoC power domains, refer to:
  - i.MX 91 Application Processors Data Sheet for Industrial Products IMX91IEC
  - i.MX 91 Application Processors Data Sheet for Consumer Products IMX91CEC
- For more details regarding the i.MX 91 power architecture, refer to the document:
  - i.MX 91 Applications Processors Reference Manual (document IMX91RM)
- For more information on the i.MX Yocto Project, refer *i.MX Yocto Project User's Guide*
- For detailed information of power design, refer to *AN14012*, "i.MX *93 to i.MX 91 Design Compatibility Guide*".

Note: *Some of the documents listed above might be available only under a non-disclosure agreement (NDA). To request access to such documents, contact your local Field Applications Engineer (FAE) or Sales Representative.*

9 Acronyms

Table 26 lists the acronyms used in this document.

Table 26. Acronyms

Acronym	Description
CA55	Arm Cortex-A55 processor
ADC	Analog-to-digital converter
AHB	Arm AMBA high-performance bus
APLL	Auxiliary phase-locked loop clock generator
Arm	Advanced RISC machine processor architecture
AXI	Arm Advanced eXtensible Interface
BSP	Board support package
BBSM	Battery-backed security module
CCM	Clock controller module
DAC	Digital-to-analog converter
DDR	Dual data rate DRAM
DMA	i.MX 91 direct memory access controller
DRAM	Dynamic random-access memory
ECC	Error Correction Code
EVK	Evaluation kit
FIRC	FAST internal reference clock
GND	Ground
GPIO	General-purpose input/output
HWFFC	Hardware Fast Frequency Change
I/Os	Inputs/Outputs
LDO	Low drop-out regulator
LPAV	Low-power audio/video domain
LPDDR4	Low-power DDR4 SDRAM with 1.1 V I/O supply
LVD	Low-voltage detector
MIPI-CSI	MIPI - Camera serial interface controller
MIPI-DSI	MIPI - Display serial interface controller
ND	Nominal drive
OD	Over drive
ODT	On-die termination
OTP	One-time programmable
PCB	Printed-circuit board
PLL	Phase-locked loop clock generator
PMC	Power management controller

Table 26. Acronyms...*continued*

Acronym	Description
PMIC	Power management-integrated circuit
RAM	Random access memory
ROM	Read-only memory
RTC	Real-time clock
SDK	Software development kit
SWFFC	Software Fast Frequency Change
SoC	System on chip
SRAM	On-chip static random access memory
SRTC	Secure real-time clock
UART	Universal asynchronous receiver/transmitter
USB	Universal serial bus
USB 2.0	USB version 2.0 peripheral
USB OTG	USB on-the-go
VFS	Voltage and frequency scaling

10 Note about the source code in the document

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

Copyright 2024-2025 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 must be 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

11 Revision history

Table 27 summarizes revisions to this document.

Table 27. Document revision history

Document ID	Release date	Description
AN14506 v.1.1	6 January 2025	AN software link updated; other minor updates
AN14506 v.1.0	24 December 2024	Initial public release

Legal information

Definitions

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

Disclaimers

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Trademarks

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

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

Contents