NXP Semiconductors
Application Note

Document Number: AN12410
Rev. 0, 04/2019

i.MX 8M Mini Power Consumption Measurement

1. Introduction

This application note helps you to design power management systems. It illustrates the current drain measurements of the i.MX 8M Mini application processors taken on the NXP EVK platform through several use cases. You may choose the appropriate power supply domains for the i.MX 8M Mini processors and become familiar with the expected processor power consumption in various scenarios.

Because the data presented in this application note is based on empirical measurements taken on a small sample size, the presented results are not guaranteed.

Contents

1. Introduction 1
2. Overview of i.MX 8M mini voltage supplies 2
3. Internal power measurement of i.MX 8M mini processor . 6
   3.1. DDR I/O power 6
   3.2. Voltage levels in the measurement process 7
   3.3. Temperature measurements 8
   3.4. Hardware and software used 9
   3.5. Measuring points on EVK platform 9
4. Use cases and measurement results 10
   4.1. Low-power mode use cases 10
   4.2. Audio_Playback, M4 idle 12
   4.3. Core benchmark 16
   4.4. GPU 18
   4.5. Heavy-load use cases 19
   4.6. Memory 22
   4.7. Storage – SD3.0 card 24
   4.8. Storage – eMMC 25
   4.9. Storage – USB 2.0 26
5. Reducing power consumption 28
   5.1. Steps to be performed before entering Suspend (Deep-sleep) mode 29
   5.2. Steps to be performed after exiting Suspend mode 29
6. Use-case configuration and usage guideline 30
   6.1. Suspend mode 30
   6.2. System idle mode 30
   6.3. Audio_Playback 32
   6.4. C-Ray 36
   6.5. Coremark 38
   6.6. GPU 39
   6.7. Heavy-load use cases 44
   6.8. Memory 47
   6.9. Storage – SD3.0 Card 48
   6.10. Storage – eMMC 50
   6.11. Storage – USB2.0 50
   6.12. Important commands 51

2. Overview of i.MX 8M mini voltage supplies

The i.MX 8M Mini processors have several power supply domains (voltage supply rails) and several internal power domains. *Figure 1* shows the connectivity of these supply rails and the distribution of the internal power domains.

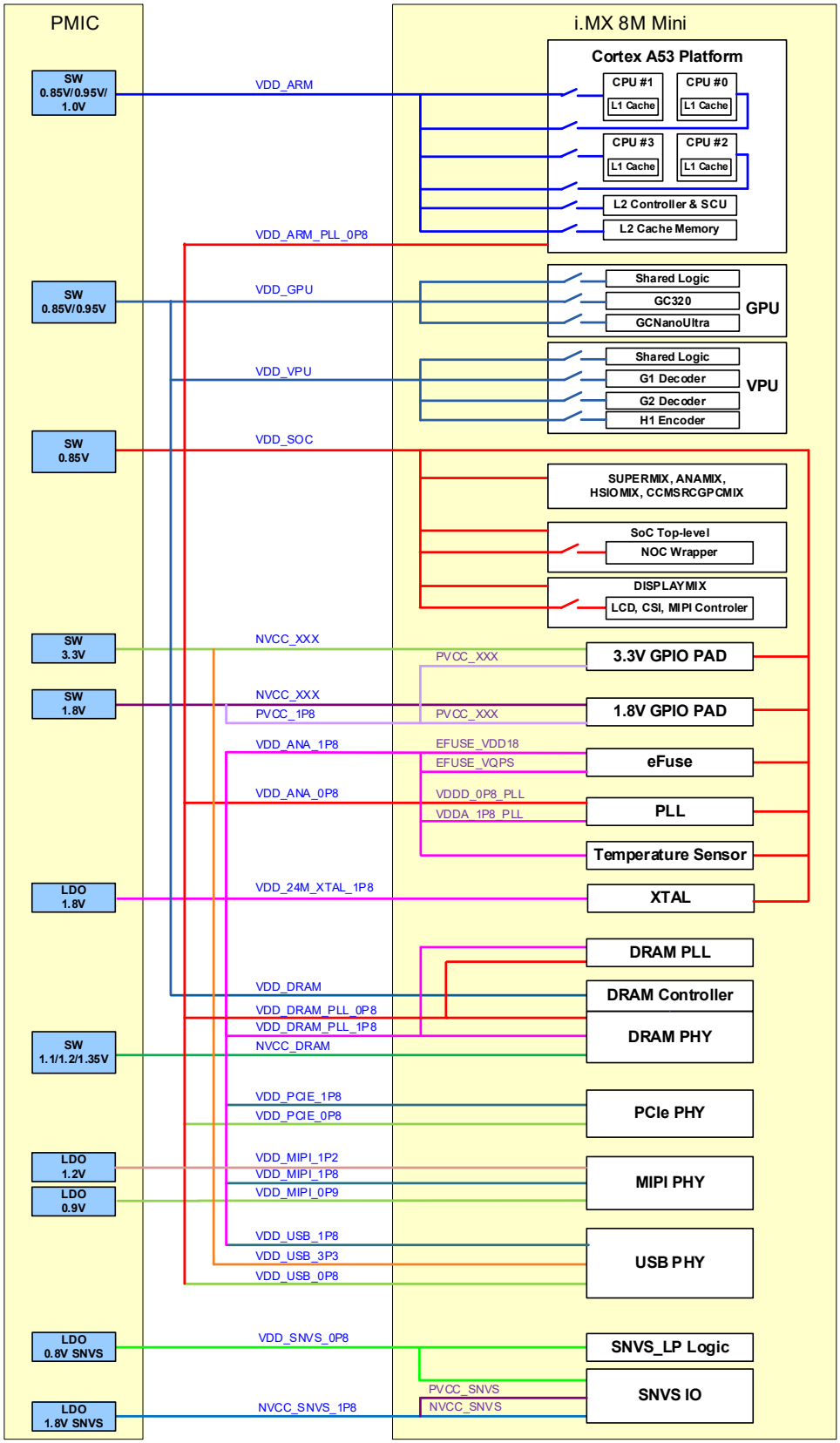

Figure 1. i.MX 8M mini power rails

NOTE

For the recommended operating conditions of each supply rail and for a detailed description of the groups of pins that are powered by each I/O voltage supply, see *i.MX 8M Mini datasheet for consumer products*. For more information about the i.MX 8M Mini power rails, see Chapter "Power Management Unit (PMU)" in the *i.MX 8M M.ini Applications Processor Reference Manual.*

Figure 2 is a snippet from the i.MX 8M mini LPDDR4 EVK board schematic showing the power distribution.

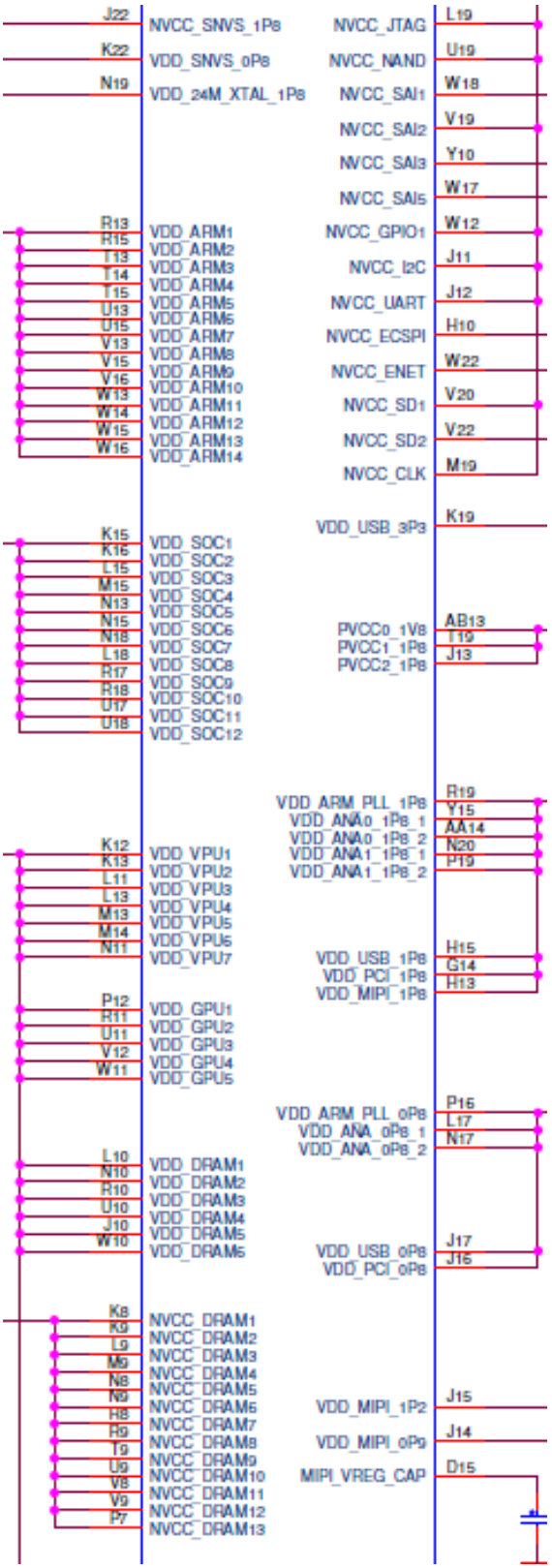

Figure 2. i.MX 8M mini power schematic

3. Internal power measurement of i.MX 8M mini processor

Several use cases (described in *Section 6 Use-case configuration and usage guideline*) are running on the EVK platform. The measurements are taken for these power supply domains:

- VDD_ARM: Arm® Cortex®-A53 Mini cores supply
- VDD_SOC: SoC logic supply
- VDD_GPU_VPU_DRAM: GPU,VPU, DRAM controller and PHY digital logic, and PLL power supply
- NVCC_DRAM: DRAM IO power supply (including an external DDR device)

These supply domains consume the majority of the processor's internal power. For relevant use cases, the power of additional supply domains is added. However, the power of these supply domains does not depend on specific use cases, but on whether these modules are used or not. The power consumption of the SNVS is comparatively negligible (except for the Suspend mode).

The NVCC_* power consumption depends primarily on the board-level configuration and the components. Therefore, it is not included in the i.MX 8M Mini internal power analysis.

Table 2 through *Table 28* provide the power consumption of these supplies (in different use cases).

NOTE

Unless stated, otherwise, all measurements were taken on a typical process silicon, at a room temperature (approximately 26 °C).

3.1. DDR I/O power

The DDR I/O is supplied from the NVCC_DRAM, which provides the power for the DDR I/O pads. The target voltage for this supply depends on the DDR interface used. The target voltages for the different DDR interfaces are:

- 1.35 V for DDR3L
- 1.2 V for DDR4
- 1.1 V for LPDDR4

The power consumption of the NVCC_DRAM supply is affected by various factors, including:

- The amount of activity on the DDR interface
- On-Die Termination (ODT): enabled/disabled, termination value, which is used for the DDR controller and the DDR memories
- Board termination for the DDR control and the address bus
- Configuration of the DDR pads (such as the drive strength)
- Board layout
- Load of the DDR memory devices

NOTE

Due to the factors specified in the previous paragraph, the measurements provided in the following tables vary from one system to another. The provided data is for guidance only and should not be treated as a specification.

The measured current on the EVK platform also includes the current of the on-board LPDDR4 memory device.

3.2. Voltage levels in the measurement process

The voltage levels of all the supplies (except for VDD_ARM, VDD_GPU, and VDD_VPU) are set to the typical voltage levels, as defined in the *i.MX 8M Mini Data Sheet for Consumer Products*.

The VDD_ARM, VDD_GPU, and VDD_VPU supplies require special explanation. To save power, these power voltages are changed during the run time of the use cases. The voltage levels of these supplies can be changed to stand-by voltage levels in low-power modes.

3.2.1. VDD_ARM/VDD_GPU/VDD_VPU voltage levels

The target voltage levels of the VDD_ARM, VDD_GPU, VDD_VPU may vary for different modes according to the use cases. The modes are the nominal mode and the overdrive mode. There are several factors that contribute to the mode decisions, with the module load being the most important. The other factors are module latency requirements, thermal restrictions, and peripheral I/O performance requirements. *Table 1* lists the voltage levels used for the measurements.

Table 1. VDD_ARM/VDD_GPU/VDD_VPU voltage levels (for reference only)

Power rail	V_{min} (V)	V_{typ} (V)	V_{max} (V)	Description
VDD_ARM	0.805	0.850	0.950	Nominal mode
	0.900	0.950	1.000	Overdrive mode
	0.950	1.000	1.050	1.0 V mode
VDD_GPU	0.805	0.850	0.900	Nominal mode
	0.855	0.900	1.000	Overdrive mode
VDD_VPU	0.805	0.850	0.900	Power supply for VPU, 450/450 MHz
	0.900	0.950	1.000	Power supply for VPU, 700/750 MHz
VDD_SOC	0.805	0.850	0.900	Power supply for SoC logic
VDD_DRAM	0.805	0.850	0.900	Power supply for DDRC, 0.85 V supports up to 1.0 GHz (DDR clock)
	0.855	0.900	0.950	Power supply for DDRC, 0.9 V supports up to 1.2 GHz (DDR clock)
	0.900	0.950	1.000	Power supply for DDRC, 0.95 V supports up to 1.5 GHz (DDR clock)

NOTE

For the official operating points, see the operating ranges table in the *i.MX 8M Mini Data Sheet for Consumer Products.*

VDD_GPU is combined with VDD_VPU, VDD_DRAM and VDD_DRAM_PLL_0P8 on the EVK board, due to design limitation.

The BD71847MWV PMIC does not support 0.950 V for VDD_GPU, VDD_VPU, and VDD_DRAM. For this PMIC, 0.975 V typical is acceptable and supported.

VDD_SOC is combined with VDD_ARM_PLL_0P8, VDD_ANA_0P8, VDD_USB_0P8, and VDD_PCI_0P8 on the EVK board, due to design limitation.

Most of the measurements are performed using these voltage levels and the power data that appears in this document is in accordance with these values. If the measurement is done at different voltage levels, the power consumption scales change with the voltage. In real applications, the software (in conjunction with the hardware) automatically adjusts the voltage and frequency values based on the use-case requirements.

The voltage used for the power calculation is the average voltage between those setpoints. It depends on the amount of time spent at each setpoint.

3.2.2. VDD_SOC voltage levels

For the official operating points, see the operating ranges table in the *i.MX 8M Mini Data Sheet for Consumer Products*. For the voltage levels used for the measurements, refer to *Table 1*.

3.3. Temperature measurements

In some use cases, the die temperature is measured. The temperature measurements were done using the on-chip temperature sensor. When measuring the temperature, it is recommended to wait until the temperature stabilizes.

NOTE

The measured temperatures are for reference only and vary on different systems due to the differences in the board, enclosure, and heat spreading techniques. When using the same board type, the measured temperature may vary due to factors such as environment, silicon variations, and measurement errors.

3.4. Hardware and software used

The software versions used for the measurements are:

- Yocto rootfs, Linux® Kernel version: *L 4.14.78 i.MX8MMini GA*.
- The board used for the measurements is the i.MX 8M Mini Rev.C LPDDR4 EVK platform.
- The measurements were performed using the 34470A 6½ digital multimeter.

3.5. Measuring points on EVK platform

To measure the power consumption, do the rework first. Split the connection between the PMIC and CPU, and then solder a 0.025 Ω sensor resistor in series. The power data is obtained by measuring the average voltage drop over the measurement points and dividing it by the resistor value to determine the average current. The tolerance of the 0.025 Ω resistors you use should be 1 % or less. The measuring points for the various supply domains are as follows:

- VDD_ARM: A53 Arm complex current for low-power measurements. The resistance value is 0.025 Ω.
- VDD_SOC: Chip domain current for SOC. The recommended resistance value for this measurement is 0.025 Ω.
- VDD_GPU_VPU_DRAM: Chip domain current for GPU, VPU and DRAM. The recommended resistance value for this measurement is 0.025 Ω.
- LPDDR4 I/O plus memory: Current in this domain includes the NVCC_DRAM current and the overall current of the on-board LPDDR4 memory device. The recommended resistance value for this measurement is 0.025 Ω.

4. Use cases and measurement results

The main use cases and subtypes that form the benchmarks for the i.MX 8M Mini internal power measurements on the EVK platform are described in the following sections.

A 1080p TV display was used only for the GPU and video playback use cases.

NOTE

For all use cases, platform is booted from an SD card with the default dtb configuration. In the U-Boot stage, check whether the default `dtb` file is `fsl-imx8mm-evk.dtb`:

4.1. Low-power mode use cases

These use-case scenarios were tested:

- Suspend mode
- IDLE_DEFAULT
- IDLE_DDRC_25MHz

4.1.1. Suspend mode

This mode is called either **Dormant mode** or **Deep sleep mode** in the Linux BSP. This is the lowest possible power state where the external supplies are still on.

The use case is as follows:

- The Arm platform is power-gated.
- The L2 Cache peripherals are power-gated.
- The Arm Cortex-M4 is in the reset status.
- All PLL (Phase-Locked Loop) and CCM (Clock Controller Module) generated clocks are OFF.
- The CKIL (32 kHz) input is on.
- All modules are disabled.
- The external high-frequency crystal and the on-chip oscillator are powered down (by asserting the SBYOS bit in the CCM).

Table 2 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 2. Suspend mode

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM[1]	0.000038	**0.019**	0.000
VDD_SOC	0.806	**9.903**	7.980
VDD_GPU_VPU_DRAM	0.000051	**0.123**	0.000
NVCC_DRAM	1.102	**0.965**	1.062
Total power	—	—	**9.042**

1. The VDD_ARM power consumption is caused by board design limitation. This power rail can be powered off in this power mode.

4.1.2. IDLE_DEFAULT

For this use case, a MIPI-DSI-to-HDMI card adapter was connected to the EVK DSI port and no display was attached to the MIPI-DSI card port.

The use case is as follows:

- The CPU frequency is set to 1200 MHz (default).
- The Arm Cortex-A53 core is power-gated if the kernel is in the lowest level of idle.
- The Arm L2 cache and PLAT are powered on.
- The Arm Cortex-M4 is in the reset status.
- All the unused PLLs are OFF and the unused clocks are gated.
- The VPU, GPU, and DISPMIX are in low-power mode.
- The operating system is on.
- The DDRC frequency is set to 750 MHz (default).

Table 3 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 3. IDLE_DEFAULT

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	0.854	**1.969**	1.682
VDD_SOC	0.852	**186.935**	159.327
VDD_GPU_VPU_DRAM	0.976	**402.484**	392.904
NVCC_DRAM	1.100	**57.316**	63.047
Total power	—	—	**616.961**

1. The die temperature was not logged because it impacts the default governor (conservative) and sets the CPU clock frequency to 1.2 GHz.

For more details about this use case and settings, see *Section 6 Use-case configuration and usage guideline*.

4.1.3. IDLE_DDRC_25MHz

For this use case, a MIPI-DSI-to-HDMI card adapter was connected to the EVK DSI port with no display attached to it.

After booting up the platform and logging in, follow the procedure in *Section 6 Use-case configuration and usage guideline* to change the DDRC clock from 750 MHz (default) to 25 MHz.

The use case is as follows:

- The CPU frequency governor is set to `powersave` (The CPU frequency is set to the minimum value).
- The Arm Cortex-A53 core is power-gated if the kernel is in the lowest level of idle.
- The Arm L2 cache and PLAT are powered on.
- The Arm Cortex-M4 is in the reset status.
- All the unused PLLs are OFF and the unused clocks are gated.
- The operating system is on.
- The DDRC frequency is set to 25 MHz.

Table 4 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 4. IDLE_DDRC_25MHz

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	0.854	**1.103**	0.942
VDD_SOC	0.853	**93.089**	79.391
VDD_GPU_VPU_DRAM	0.975	**23.296**	22.707
NVCC_DRAM	1.100	**32.888**	36.167
Total power	—	—	**139.206**

1. The die temperature was not logged because it impacts the default governor (conservative) and sets the CPU clock frequency to 1.2 GHz.

For more details about this use case and settings, see *Section 6 Use-case configuration and usage guideline*.

4.2. Audio_Playback, M4 idle

These use-case scenarios were tested:

- Audio_Playback(gplay)
- Audio_Playback(gplay)_DDRC_25MHz
- Audio+Video_Playback(gplay)
- Audio+Video_Stream(gplay)

4.2.1. Audio_Playback(gplay)

The audio file used was an mp3 file with a 128-kbps bitrate and a 44 kHz sample rate/s, played using the following options:

The use case is as follows:

- The CPU frequency governor is set to **performance** (The CPU frequency is set to the maximum value).
- The Arm L2 cache and PLAT are powered on.
- The Arm Cortex-M4 is in the reset status.
- The VPU, GPU, and DISPMIX are in low power mode.
- All the unused PLLs are OFF and the unused clocks are gated.
- The operating system is on.

Table 5 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 5. Audio_Playback(gplay)

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.003	**32.210**	32.316
VDD_SOC	0.853	**197.492**	168.432
VDD_GPU_VPU_DRAM	0.976	**405.538**	395.688
NVCC_DRAM	1.100	**69.737**	76.714
Total power	—	—	**673.150**

1. The die temperature was approximately **45 °C** (avg.). The ambient temperature was approximately **26 °C**. `cat/sys/class/thermal/thermal_zone0/temp` was used to log the temperature while the audio file was playing.

For more details about this use case and settings, see *Section 6 Use-case configuration and usage guideline*.

4.2.2. Audio_Playback(gplay)_DDRC_25MHz

For this use case, DDRC clock frequency was set to 25 MHz, as specified in *Section 6 Use-case configuration and usage guideline*.

The audio file used was an mp3 file with a 128-kbps bitrate and a 44kHz sample rate/s, played using the following options.

```
gplay-1.0 $audio_file
```

The use case is as follows:

- The CPU frequency governor is set to `powersave` (CPU frequency is set to minimum value).
- The Arm Cortex-A53 core is power-gated if the kernel is in the lowest level of idle.
- The Arm L2 cache and PLAT are powered on.
- The Arm Cortex-M4 is in the reset status.
- The VPU, GPU, and DISPMIX are in low power mode.

- All the unused PLLs are OFF and the unused clocks are gated.
- The operating system is on.
- The DDRC frequency is set to 25 MHz.

Table 6 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 6. Audio_Playback(gplay)_DDRC_25MHz

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	0.855	**56.831**	48.500
VDD_SOC	0.853	**99.593**	84.978
VDD_GPU_VPU_DRAM	0.975	**28.393**	27.690
NVCC_DRAM	1.100	**94.974**	104.471
Total power	—	—	**265.600**

1. The die temperature was approximately **42 °C** (avg.). The ambient temperature was approximately **26 °C**. `cat/sys/class/thermal/thermal_zone0/temp` was used to log the temperature while the audio file was playing.

For more details about this use case and settings, see *Section 6 Use-case configuration and usage guideline*.

4.2.3. Audio+Video_Playback(gplay)

For this use case, the MIPI-DSI port was connected to a 1080p TV display.

The video file used for playback was a mkv file format compressed using the HEVC standard with full HD resolution at 29.97 fps and the audio encoding was AACL with a 44.1-kHz sample rate in a 2-channel configuration.

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

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

The use case is as follows:

- The CPU frequency governor is set to **performance** (The CPU frequency is set to the maximum value).
- The Arm L2 cache and PLAT are powered on.
- The Arm Cortex-M4 is in the reset status.
- All the unused PLLs are OFF and the unused clocks are gated.
- The operating system is on.

Table 7 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 7. Audio+Videp_Playback(gplay)

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.003	**91.168**	91.437
VDD_SOC	0.853	**264.185**	225.245
VDD_GPU_VPU_DRAM	0.976	**546.114**	553.069
NVCC_DRAM	1.101	**231.704**	255.108
Total power	—	—	**1104.859**

1. The die temperature was approximately **50 °C** (avg.). The ambient temperature was approximately **26 °C**. *cat/sys/class/thermal/thermal_zone0/temp* was used to log the temperature while the audio file was playing.

For more details about this use case and settings, see *Section 6 Use-case configuration and usage guideline*.

4.2.4. Audio+Video_Stream(gplay)

For this use case, MIPI-DSI port was connected to a 1080p TV display.

The video file used for playback was mkv file format compressed using the HEVC standard with full HD resolution at 29.97 fps and the audio encoding was AACL in a 2-channel configuration with 44.1 kHz samples/s.

A server was setup to host the mkv video file for streaming.

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

```
gplay-1.0 $FILE
```

The use case is as follows:

- The CPU frequency governor is set to **performance** (The CPU frequency is set to the maximum value).
- The Arm L2 cache and PLAT are powered on.
- The Arm Cortex-M4 is in the reset status.
- All the unused PLLs are OFF and the unused clocks are gated.
- The operating system is on.

Table 8 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 8. Audio+Videp_Stream(gplay)

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.003	**97.163**	97.436
VDD_SOC	0.853	**269.894**	230.118
VDD_GPU_VPU_DRAM	0.976	**544.207**	531.248
NVCC_DRAM	1.101	**231.831**	255.251
Total power	—	—	**1114.052**

1. The die temperature was approximately **49 °C** (avg.). The ambient temperature was approximately **26 °C**. `cat /sys/class/thermal/thermal_zone0/temp` was used to log the temperature while the audio file was playing.
2. For video streaming Ethernet adapter was used.

For more details about this use case and settings, see *Section 6 Use-case configuration and usage guideline*.

4.3. Core benchmark

These use-case scenarios were tested:

- C-Ray
- Coremark

4.3.1. C-Ray

- C-Ray is an extremely simple ray-tracer, which is not representative of any real-world ray-tracing application. In fact, it is essentially a floating-point benchmark that runs from the L1-cache. Therefore, it is not as synthetic and meaningless as Whetstone because you can actually using the software to do simple ray-tracing.
- This is a multi-thread benchmark and the default test scene involves only a small amount of data, so that on most systems the CPU does not have to access the main RAM to run the test.
- The CPU frequency governor is set to **performance** (The CPU frequency is set to the maximum value).
- The DDRC frequency is set to 750 MHz.

Table 9 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 9. C-Ray

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.004	**1144.423**	1149.102
VDD_SOC	0.852	**201.123**	171.411
VDD_GPU_VPU_DRAM	0.977	**412.179**	402.529
NVCC_DRAM	1.100	**57.360**	63.088
Total power	—	—	**1786.129**

1. The die temperature was approximately **57 °C** (avg.). The ambient temperature was approximately **26 °C**. *cat /sys/class/thermal/thermal_zone0/temp* was used to log the temperature while the bench mark is running.

4.3.2. 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.

For the best performance, compile as follows:

```
-O2 -DMULTITHREAD=4 -DUSE_PTHREAD -lpthread -O3 -funroll-all-loops
--param max-inline-insns-auto=550 -ftracer -falign-jumps=16 -ftree-loop-im -fivopts
-ftree-loop-ivcanon -fvect-cost-model -fvariable-expansion-in-unroller
--param max-unrolled-insns=999999 --param max-average-unrolled-insns=99999999
--param iv-max-considered-uses=9999999 --param iv-consider-all-candidates-bound=99999
--param iv-always-prune-cand-set-bound=999999   -fmodulo-sched
-fmodulo-sched-allow-regmoves -fgcse-lm -fgcse-sm -fgcse-las -funsafe-loop-optimizations
-freschedule-modulo-scheduled-loops  -ftree-vectorize -DPERFORMANCE_RUN=1   -lrt
```

- The CPU frequency governor is set to **performance** (The CPU frequency is set to the maximum value).
- The DDRC frequency is set to 750 MHz.

Table 10 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 10. Coremark

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.005	**953.832**	958.524
VDD_SOC	0.852	**198.468**	169.134
VDD_GPU_VPU_DRAM	0.976	**410.289**	400.638
NVCC_DRAM	1.100	**57.204**	62.920
Total power	—	—	**1591.216**

1. The die temperature was approximately **51 °C** (avg.). The ambient temperature was approximately **26 °C**. *cat/sys/class/thermal/thermal_zone0/temp* was used to log the temperature while the bench mark is running.

4.4. GPU

These use-case scenarios were tested:

- MM07
- MM06
- GPU_Kanzi
- GPU_GLmark

MM07 and MM06 are 3D-gaming benchmarks. The graphics are loaded from the SD card into the DDR (Double Data Rate) memory, processed by GPU3D, and copied to a display buffer in the DDR memory. It is displayed on the1080p display (through MIPI-DSI).

- The CPU frequency governor is set to **performance** (The CPU frequency is set to the maximum value).
- The DDRC frequency is set to 750 MHz.

4.4.1. MM07

Table 11 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 11. GPU_MM07

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.003	**42.057**	42.190
VDD_SOC	0.852	**251.545**	214.218
VDD_GPU_VPU_DRAM	0.976	**629.058**	613.781
NVCC_DRAM	1.102	**224.968**	247.860
Total power	—	—	**1118.049**

1. The die temperature was approximately **50 °C** (avg.). The ambient temperature was approximately **26 °C**. `cat/sys/class/thermal/thermal_zone0/temp` was used to log the temperature while the benchmark is running.

4.4.2. MM06

Table 12 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 12. GPU_MM06

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.004	**70.456**	70.721
VDD_SOC	0.851	**276.058**	235.020
VDD_GPU_VPU_DRAM	0.973	**773.266**	752.356
NVCC_DRAM	1.101	**357.077**	393.177
Total power	—	—	**1451.273**

1. The die temperature was approximately **50 °C** (avg.). The ambient temperature was approximately **26 °C**. `cat/sys/class/thermal/thermal_zone0/temp` was used to log the temperature while the benchmark is running.

4.4.3. GPU_Kanzi

Table 13 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 13. GPU_Kanzi

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.003	**135.703**	136.078
VDD_SOC	0.851	**274.515**	233.715
VDD_GPU_VPU_DRAM	0.975	**634.001**	618.341
NVCC_DRAM	1.102	**318.083**	350.645
Total power	—	—	**1338.779**

1. The die temperature was approximately **54 °C** (avg.). The ambient temperature was approximately **26 °C**. `cat/sys/class/thermal/thermal_zone0/temp` was used to log the temperature while the benchmark is running.

4.4.4. GPU_GLmark

Table 14 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 14. GPU_GLmark

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.004	**64.990**	65.247
VDD_SOC	0.851	**268.040**	228.215
VDD_GPU_VPU_DRAM	0.975	**670.220**	653.427
NVCC_DRAM	1.101	**287.991**	317.077
Total power	—	—	**1263.966**

1. The die temperature was approximately **53 °C** (avg.). The ambient temperature was approximately **26 °C**. `cat/sys/class/thermal/thermal_zone0/temp` was used to log the temperature while the benchmark is running.

4.5. Heavy-load use cases

These use-case scenarios were tested:

- VPU
- 4-core Dhryst + VPU + Taiji
- 4-core Memtest + VPU + Taiji
- 4-core Streamcpy + VPU + Taiji
- Coremark + Kanzi

The purpose of these use cases is to provide the power consumption for heavy-load use cases to show the power consumption in extreme conditions.

4.5.1. VPU

This use case has the following features:

- 1080p display ON, 1920 x 1080, 44.1 kHz.
- 1080p file decoding via Hantro decoder.
- (`g2dec -P -Ers -ibs -b -N200 -X HEVC_1920x1080_29.97fps_AACLC_44.1Khz_2ch.mkv`)
- The CPU frequency governor is set to **performance** (The CPU frequency is set to the maximum value).
- The DDRC frequency is set to 750 MHz.

Table 15 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 15. VPU

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.006	**478.117**	480.773
VDD_SOC	0.851	**247.712**	210.923
VDD_GPU_VPU_DRAM	0.976	**505.336**	493.043
NVCC_DRAM	1.100	**191.353**	210.548
Total power	—	—	**1395.287**

1. The die temperature was approximately **56 °C** (avg.). The ambient temperature was approximately **26 °C**. `cat/sys/class/thermal/thermal_zone0/temp` was used to log the temperature while the benchmark is running.

4.5.2. 4-core Dhryst + VPU + Taiji

This use case runs 4.3.1, 4.5.1, and 4.4.1 in parallel.

- The CPU frequency governor is set to **performance** (The CPU frequency is set to the maximum value).
- The DDRC frequency is set to 750 MHz.

Table 16 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 16. 4-core Dhryst + VPU + Taiji

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.002	**1095.964**	1098.667
VDD_SOC	0.851	**276.425**	235.375
VDD_GPU_VPU_DRAM	0.975	**703.264**	685.462
NVCC_DRAM	1.102	**230.678**	254.240
Total power	—	—	**2273.744**

1. The die temperature was approximately **70 °C** (avg.). The ambient temperature was approximately **26 °C**. `cat/sys/class/thermal/thermal_zone0/temp` was used to log the temperature while the benchmark is running.

4.5.3. 4-core Memtest + VPU + Taiji

This use case runs memtester (an effective user-space tester for stress-test the memory subsystem), 4.5.1, and 4.4.1 in parallel.

- The CPU frequency governor is set to **performance** (The CPU frequency is set to the maximum value).
- The DDRC frequency is set to 750 MHz.

Table 17 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 17. 4-core Memtest + VPU + Taiji

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.006	**885.786**	891.031
VDD_SOC	0.851	**302.892**	257.863
VDD_GPU_VPU_DRAM	0.974	**746.186**	726.709
NVCC_DRAM	1.104	**416.140**	459.390
Total power	—	—	**2334.995**

1. The die temperature was approximately **72 °C** (avg.). The ambient temperature was approximately **26 °C**. `cat/sys/class/thermal/thermal_zone0/temp` was used to log the temperature while the benchmark is running.

4.5.4. 4-core Streamcpy + VPU + Taiji

This use case run streamcpy (4.6.3), 4.5.1, and 4.4.1 in parallel.

- The CPU frequency governor is set to **performance** (The CPU frequency is set to the maximum value).
- The DDRC frequency is set to 750 MHz.

Table 18 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 18. 4-core Streamycpy + VPU + Taiji mode measurement results

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.005	**863.126**	867.779
VDD_SOC	0.851	**300.985**	256.270
VDD_GPU_VPU_DRAM	0.974	**761.450**	741.418
NVCC_DRAM	1.103	**446.948**	493.044
Total power	—	—	**2358.510**

1. The die temperature was approximately **71 °C** (avg.). The ambient temperature was approximately **26 °C**.
2. `cat/sys/class/thermal/thermal_zone0/temp` was used to log the temperature while the benchmark is running.

4.5.5. Coremark + Kanzi

This use case run Coremark (4.2.3) and 4.4.3 (GPU_Kanzi) in parallel.

- The CPU frequency governor is set to **performance** (The CPU frequency is set to the maximum value).
- The DDRC frequency is set to 750 MHz.

Table 19 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 19. Coremark + Kanzi mode measurement results

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.004	**978.255**	982.091
VDD_SOC	0.852	**282.274**	240.376
VDD_GPU_VPU_DRAM	0.975	**647.717**	631.412
NVCC_DRAM	1.102	**278.099**	306.531
Total power	—	—	**2160.410**

1. The die temperature was approximately **70 °C** (avg.). The ambient temperature was approximately **26 °C**. `cat/sys/class/thermal/thermal_zone0/temp` was used to log the temperature while the benchmark is running.

4.6. Memory

These use-case scenarios were tested:

- Memset
- Memcpy
- Stream

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

4.6.1. Memset

Suite for evaluating the performance of a simple memory set in various ways.

- The size of the memory buffers is set to 1024 MB.
- The CPU frequency governor is set to **performance** (The CPU frequency is set to the maximum value).
- The DDRC frequency is set to 750 MHz.

Table 20 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 20. Memset

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.004	**323.568**	325.021
VDD_SOC	0.852	**262.694**	223.754
VDD_GPU_VPU_DRAM	0.977	**575.507**	562.491
NVCC_DRAM	1.102	**431.141**	475.217
Total power	—	—	**1586.483**

1. The die temperature was approximately **56 °C** (avg.). The ambient temperature was approximately **26 °C**. *cat/sys/class/thermal/thermal_zone0/temp* was used to log the temperature while the benchmark is running.

4.6.2. Memcpy

Memcpy is a suite for evaluating the performance of a simple memory copy in various ways.

- The size of the memory buffers is set to 1024 MB.
- The CPU frequency governor is set to **performance** (The CPU frequency is set to the maximum value).
- The DDRC frequency is set to 750 MHz.

Table 21 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 21. Memcpy

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.004	**276.013**	277.133
VDD_SOC	0.852	**237.372**	202.236
VDD_GPU_VPU_DRAM	0.977	**479.761**	468.696
NVCC_DRAM	1.101	**242.210**	266.583
Total power	—	—	**1214.648**

1. The die temperature was approximately **50 °C** (avg.). The ambient temperature was approximately **26 °C**. *cat/sys/class/thermal/thermal_zone0/temp* was used to log the temperature while the benchmark is running.

4.6.3. Stream

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

- The stream array size is set to 102400000 elements.
- All phases are included (Copy, Scale, Add, and Triad).
- The CPU frequency governor is set to **performance** (The CPU frequency is set to the maximum value).
- The DDRC frequency is set to 750 MHz.

Table 22 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 22. Stream

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.005	**709.700**	713.160
VDD_SOC	0.852	**252.025**	214.683
VDD_GPU_VPU_DRAM	0.977	**515.806**	504.055
NVCC_DRAM	1.102	**382.437**	421.424
Total power	—	—	**1853.323**

1. The die temperature was approximately **60 °C** (avg.). The ambient temperature was approximately **26 °C**. `cat/sys/class/thermal/thermal_zone0/temp` was used to log the temperature while the benchmark is running.

4.7. Storage – SD3.0 card

These use-case scenarios were tested:

- DD_RD_SDCARD
- DD_WRT_SDCARD

4.7.1. DD_RD_SDCARD

- The CPU frequency governor is set to **performance** (The CPU frequency is set to the maximum value).
- The DDRC frequency is set to 750 MHz.

Table 23 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 23. DD_RD_SDCARD

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.004	**102.908**	103.324
VDD_SOC	0.852	**226.495**	192.990
VDD_GPU_VPU_DRAM	0.976	**416.789**	406.979
NVCC_DRAM	1.100	**142.935**	157.259
Total power	—	—	**860.552**

1. The die temperature was approximately **47 °C** (avg.). The ambient temperature was approximately **26 °C**. `cat/sys/class/thermal/thermal_zone0/temp` was used to log the temperature while the benchmark is running.

4.7.2. DD_WRT_SDCARD

- Set the maximum amount of data that the kernel reads ahead for a single file to 512 KB.
- (`echo 512 > /sys/block/<bdev>/queue/read_ahead_kb`)
- The CPU frequency governor is set to **performance** (The CPU frequency is set to the maximum value).
- The DDRC frequency is set to 750 MHz.

Table 24 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 24. DD_WRT_SDCARD

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.005	**159.110**	159.955
VDD_SOC	0.852	**223.900**	190.772
VDD_GPU_VPU_DRAM	0.976	**411.407**	401.714
NVCC_DRAM	1.100	**119.350**	131.284
Total power	—	—	**883.726**

1. The die temperature was approximately **48 °C** (avg.). The ambient temperature was approximately **26 °C**. *cat/sys/class/thermal/thermal_zone0/temp* was used to log the temperature while the benchmark is running.

4.8. Storage – eMMC

These use-case scenarios were tested:

- DD_RD_eMMC
- DD_WRT_eMMC

4.8.1. DD_RD_eMMC

- Set the maximum amount of data that the kernel reads ahead for a single file to 512 KB.
- (*echo 512 > /sys/block/<bdev>/queue/read_ahead_kb*)
- The CPU frequency governor is set to **performance** (The CPU frequency is set to the maximum value).
- The DDRC frequency is set to 750 MHz.

Table 25 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 25. DD_RD_eMMC mode measurement results

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.004	**189.449**	190.248
VDD_SOC	0.852	**246.808**	210.276
VDD_GPU_VPU_DRAM	0.976	**440.167**	429.806
NVCC_DRAM	1.100	**196.853**	216.483
Total power	—	—	**1046.812**

1. The die temperature was approximately **45 °C** (avg.). The ambient temperature was approximately **26 °C**. *cat/sys/class/thermal/thermal_zone0/temp* was used to log the temperature while the benchmark is running.

4.8.2. DD_WRT_eMMC

- Set the maximum amount of data that the kernel reads ahead for a single file to 512 KB.
- (`echo 512 > /sys/block/<bdev>/queue/read_ahead_kb`)
- The CPU frequency governor is set to **performance** (The CPU frequency is set to the maximum value).
- The DDRC frequency is set to 750 MHz.

Table 26 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 26. DD_WRT_eMMC

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.007	**170.256**	171.376
VDD_SOC	0.852	**228.416**	194.630
VDD_GPU_VPU_DRAM	0.976	**416.032**	406.202
NVCC_DRAM	1.100	**110.160**	121.186
Total power	—	—	**893.394**

1. The die temperature was approximately **47 °C** (avg.). The ambient temperature was approximately **26 °C**. `cat/sys/class/thermal/thermal_zone0/temp` was used to log the temperature while the benchmark is running.

4.9. Storage – USB 2.0

These use-case scenarios were tested:

- DD_RD_USB2.0
- DD_WRT_ USB2.0

4.9.1. DD_RD_USB2.0

- Set the maximum amount of data that the kernel reads ahead for a single file to 512 KB.
- (`echo 512 > /sys/block/<bdev>/queue/read_ahead_kb`)
- The CPU frequency governor is set to **performance** (The CPU frequency is set to the maximum value).
- The DDRC frequency is set to 750 MHz.

Table 27 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 27. DD_RD_USB2.0

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.004	**91.803**	92.125
VDD_SOC	0.852	**245.348**	209.024
VDD_GPU_VPU_DRAM	0.976	**411.645**	401.860
NVCC_DRAM	1.100	**129.611**	142.581
Total power	—	—	**845.590**

1. The die temperature was approximately **44 °C** (avg.). The ambient temperature was approximately **26 °C**. *cat/sys/class/thermal/thermal_zone0/temp* was used to log the temperature while the benchmark is running.

4.9.2. DD_WRT_USB2.0

- Set the maximum amount of data that the kernel reads ahead for a single file to 512 KB.
- (*echo 512 > /sys/block/<bdev>/queue/read_ahead_kb*)
- The CPU frequency governor is set to **performance** (The CPU frequency is set to the maximum value).
- The DDRC frequency is set to 750 MHz.

Table 28 shows the measurement results when this use case is applied on the i.MX 8M Mini processor.

Table 28. DD_WRT_USB2.0

Supply mode	Voltage (V)	L4.14.98-MM8	
		I (mA)	P (mW)
VDD_ARM	1.004	**130.808**	131.271
VDD_SOC	0.852	**246.330**	209.839
VDD_GPU_VPU_DRAM	0.976	**412.293**	402.533
NVCC_DRAM	1.100	**105.401**	115.936
Total power	—	—	**859.579**

1. The die temperature was approximately **46 °C** (avg.). The ambient temperature was approximately **26 °C**. *cat/sys/class/thermal/thermal_zone0/temp* was used to log the temperature while the benchmark is running.

5. Reducing power consumption

The overall system power consumption depends on both the software optimization and how the system hardware is implemented. The following is a list of suggestions that may help to reduce the system power consumption:

- Apply the clock gating whenever the clocks or modules are not used by configuring the CCGR registers in the Clock Controller Module (CCM).
- Reduce the number of operating PLLs: Applicable mainly in the Audio_Playback or Idle modes.

Core DVFS and system bus scaling: Applying the DVFS for ARM and scaling the frequencies of the NOC, AXI, AHB, and IPG bus clocks can significantly reduce the power consumption of the VDD_ARM and VDD_SOC domains. However, due to reduced operation frequency, the accesses to the DDR take longer, which increases the power consumption of the DDR I/O and memory. This trade-off must be taken into account for each mode to quantify the overall effect on the system power consumption.

- Put the i.MX 8M Mini into the low-power modes (STOP) whenever possible. See Chapter "Clock Controller Module (CCM)" in the *i.MX 8M Mini Applications Processor Reference Manual* for details.
- DDR interface optimization:
  - Employ careful board routing of the DDR memories, maintaining the PCB trace lengths as short as possible.
  - Use as reduced an ODT (On-Die Termination) setting as possible. The termination used greatly influences the power consumption of the DDR interface pins.
  - Use a proper 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.
  - 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.

The various steps are shown below.

NOTE

All the programming steps below are performed in the Arm trusted firmware from the internal RAM.

5.1. Steps to be performed before entering Suspend (Deep-sleep) mode

1. Read the DBGCAM register in DDRC to make sure that the explicit transaction command queue is empty. Wait until the AXI port is idle.
2. Do the following:
    a) Put the DDR into self-refresh.
    b) Transition the DDR PHY into LP3/IO retention state by using the DFI frequency operation.
    c) Set the PwrOkIn signal in SRC to 0. This enables the data retention feature on the CKE and MEMRESET.
    d) Gate the DDRC's CORE clock and APB clock.
    e) Enable DDRMIX ISO to power-gate the DDRC and PHY.
3. Enter Suspend mode.

5.2. Steps to be performed after exiting Suspend mode

1. Restore all the settings for the DDRC and PHY to the required values.
2. The system proceeds to the Run mode.

6. Use-case configuration and usage guideline

NOTE

Before running a use case, *<configuration_script>.sh* must be run to configure the environment. These are: *setup.sh*, *setup_default.sh*, *setup_video.sh*, *setup_video_stream.sh*, *DDRC_25MHz_setup.sh* (see *Section 6.12 Important commands* for details).

6.1. Suspend mode

In this use case, all clocks and PLLs are turned off, except for the 32 kHz clock which is used to wake up the system:

1. Boot up the Linux image.
2. Run this command to put the system into the DSM mode:

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

3. Measure the power and record the result.

6.2. System idle mode

NOTE

No display was connected to the platform.

6.2.1. IDLE_DEFAULT

6.2.1.1. Clock configuration

The clock configuration in *Table 29* is aligned with release L4.14.98.

Table 29. IDLE_DEFAULT clock configuration

Clock name	Frequency (MHz)
NOC	150
AXI	24
AHB	20
CPU	1200
DDRC	750

6.2.1.2. PLL configuration

The PLL configuration in *Table 30* is aligned with release L4.14.98.

Table 30. IDLE_DEFAULT PLL configuration

Clock root	Source selected	Frequency (MHz)
ARM_A53_CLK	PLL_ARM_MAIN_CLK	1200
NOC_CLK	SYS3_PLL_750M	150
MAIN_AXI_CLK	OSC_24M	24
DISP_AXI_CLK	SYS1_PLL_800M	OFF
ENET_AXI_CLK	SYS1_PLL_266M	OFF
NAND_USDHC_BUS_CLK	SYS1_PLL_266M	OFF
AHB_CLK_ROOT	SYS1_PLL_133M	20
IPG_CLK	AHB_CLK_ROOT	10

6.2.1.3. System setup

Disconnect everything except for the SD card.

Make sure there are no displays connected to the platform.

1. Boot up the Linux OS.
2. Run *setup_default.sh* (*Section 6.12 Important commands*) to put the system to system idle mode.
3. Measure the power and record the result.

6.2.2. IDLE_DDRC_25MHz

6.2.2.1. Clock configuration

The clock configuration in *Table 31* is aligned with release L4.14.98.

Table 31. IDLE_DDRC_25MHz clock configuration

Clock name	Frequency (MHz)
NOC	150
AXI	24
AHB	20
CPU	1200
DDRC	750

6.2.2.2. PLL configuration

The PLL configuration in *Table 32* is aligned with release L4.14.98.

Table 32. IDLE_DDRC_25MHz PLL configuration

Clock root	Source selected	Frequency (MHz)
ARM_A53_CLK	PLL_ARM_MAIN_CLK	1200
NOC_CLK	SYS3_PLL_750M	150
MAIN_AXI_CLK	OSC_24M	24
DISP_AXI_CLK	SYS1_PLL_800M	OFF
ENET_AXI_CLK	SYS1_PLL_266M	OFF
NAND_USDHC_BUS_CLK	SYS1_PLL_266M	OFF
AHB_CLK_ROOT	SYS1_PLL_133M	20
IPG_CLK	AHB_CLK_ROOT	10

6.2.2.3. System setup

Disconnect everything except for the SD card.

Make sure there are no displays connected to the platform.

1. Boot up the Linux OS.
2. Run `DDRC_25MHz_setup.sh` to put the system into system idle mode and set the DDRC frequency to 25 MHz (*Section 6.12 Important commands*).
3. Measure the power and record the result.

6.3. Audio_Playback

6.3.1. Clock configuration

The clock configuration in *Table 33* to *Table 35* is aligned with release L4.14.98.

Table 33. Audio_Playback(gplay) clock configuration

Clock name	Frequency (MHz)
NOC	150
AXI	24
AHB	20
CPU	1800
DDRC	750

Table 34. Audio_Playback_DDRC_25MHz(gplay) clock configuration

Clock name	Frequency (MHz)
NOC	150
AXI	24
AHB	20
CPU	1200
DDRC	25

Table 35. Audio+Video_Playback(gplay) clock configuration

Clock name	Frequency (MHz)
NOC	750
AXI	333
AHB	133
CPU	1800
DDRC	750

6.3.2. PLL configuration

The PLL configuration in *Table 36* to *Table 38* is aligned with release L4.14.98.

Table 36. Audio_Playback(gplay) PLL configuration

Clock root	Source selected	Frequency (MHz)
ARM_A53_CLK	PLL_ARM_MAIN_CLK	1800
NOC_CLK	SYS_PLL3_OUT	750
MAIN_AXI_CLK	24M_OSC	24
DISP_AXI_CLK	SYS1_PLL_800M	800
ENET_AXI_CLK	SYS1_PLL_266M	266
NAND_USDHC_BUS_CLK	SYS1_PLL_266M	266
AHB_CLK_ROOT	SYS1_PLL_133M	20
IPG_CLK	AHB_ROOT_CLK	10
DRAM_CLK	PLL_DRAM_MAIN_CLK	750
PCIE_CTRL_CLK	SYS2_PLL_250M	OFF
LCDIF_PIXEL_CLK	24M OSC	—
SAIx_CLK	AUDIO_PLL_OUT	36.8
ENETx_REF_CLK	SYS2_PLL_125M	125
ENETx_TIME_CLK	SYS2_PLL_100M	100
ENET_PHY_REF_CLK	24M OSC	25
NAND_CLK	SYS1_PLL_400M	OFF
QSPI_CLK	SYS1_PLL_100M	OFF
USHDCx_CLK	SYS1_PLL_400M	392
I2Cx_CLK	24M OSC	24
UARTx_CLK	24M OSC	24
ECSPIx_CLK	SYS2_PLL_200M	OFF
PWMx_CLK	24M OSC	24
GPTx_CLK	24M OSC	24
TRACE_CLK	24M OSC	OFF
WDOG_CLK	24M OSC	OFF

Table 37. Audio_Playback(gplay)_DDRC_25MHz PLL configuration

Clock root	Source selected	Frequency (MHz)
ARM_A53_CLK	PLL_ARM_MAIN_CLK	1200
NOC_CLK	SYS3_PLL_750M	150
MAIN_AXI_CLK	OSC_24M	24
DISP_AXI_CLK	SYS1_PLL_800M	OFF
ENET_AXI_CLK	SYS1_PLL_266M	OFF
NAND_USDHC_BUS_CLK	SYS1_PLL_266M	OFF
AHB_CLK_ROOT	SYS1_PLL_133M	20
IPG_CLK	AHB_CLK_ROOT	10

Table 38. Audio+Video_Playback(gplay) PLL configuration

Clock root	Source selected	Frequency (MHz)
ARM_A53_CLK	PLL_ARM_MAIN_CLK	1800
NOC_CLK	SYS1_PLL_800M	750
MAIN_AXI_CLK	SYS2_PLL_333M	333
DISP_AXI_CLK	SYS1_PLL_800M	800
ENET_AXI_CLK	SYS1_PLL_266M	266
NAND_USDHC_BUS_CLK	SYS1_PLL_266M	266
AHB_CLK_ROOT	SYS1_PLL_133M	133
IPG_CLK	AHB_ROOT_CLK	66
DRAM_CLK	PLL_DRAM_MAIN_CLK	750
PCIE_CTRL_CLK	SYS2_PLL_250M	OFF
LCDIF_PIXEL_CLK	24M OSC	—
SAIx_CLK	AUDIO_PLL_OUT	36.8
ENETx_REF_CLK	SYS2_PLL_125M	125
ENETx_TIME_CLK	SYS2_PLL_100M	100
ENET_PHY_REF_CLK	24M OSC	24
NAND_CLK	SYS1_PLL_400M	OFF
QSPI_CLK	SYS1_PLL_100M	OFF
USHDCx_CLK	SYS1_PLL_400M	392
I2Cx_CLK	24M OSC	24
UARTx_CLK	24M OSC	24
ECSPIx_CLK	SYS2_PLL_200M	OFF
PWMx_CLK	24M OSC	24
GPTx_CLK	24M OSC	24
TRACE_CLK	24M OSC	OFF
WDOG_CLK	24M OSC	OFF

6.3.3. Audio_Playback(gplay)

The audio file used was an mp3 file with a 128-kbps bitrate and a 44-kHz sample rate/s, played using the following options:

1. Boot up the Linux OS.
2. Run *setup.sh* (*Section 6.12 Important commands*).
3. Run *gplay_audio.sh* and measure:

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

4. Start the die temperature recording (*Section 6.12 Important commands*).
5. Measure the power and record the result.

6.3.4. Audio_Playback(gplay)_DDRC_25MHz

The audio file used was an mp3 file with a 128-kbps bitrate and a 44-kHz sample rate/s, played using the following options:

1. Boot up the Linux OS.
2. Run *DDRC_25MHz_setup.sh* to set DDRC frequency at 25 MHz (*Section 6.12 Important commands*).
3. Run *gplay_audio.sh* and measure:

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

4. Start the die temperature recording (*Section 6.12 Important commands*).
5. Measure the power and record the result.

6.3.5. Audio+Video_Playback(gplay)

For this use case, the MIPI-DSI port was connected to a 1080p TV display.

The video file used for playback is .mkv file format compressed with HEVC standard with full HD resolution at 29.97 fps and the audio encoding is AACL with a 44.1 kHz samples/s and 2-channel configuration.

Video file was locally played using gplay-1.0, with the following options:

1. Boot up the Linux OS.
2. Run *setup_video.sh* to put the system into system idle mode (*Section 6.12 Important commands*).
3. Run *gplay_videoplayback.sh* and measure:

```
path=`pwd`
FILE=HEVC_1920x1080_29.97fps_AACLC_44.1Khz_2ch.mkv
gplay-1.0 $path/$FILE
```

4. Start the die temperature recording (*Section 6.12 Important commands*).
5. Measure the power and record the result.

6.3.6. Audio+Video_Stream(gplay)

For this use case, the MIPI-DSI port was connected to a 1080p TV display.

The video file used for playback is mkv file format compressed using the HEVC standard with full HD resolution at 29.97 fps and the audio encoding is AACL in 2-channels configuration with 44.1 kHz samples/s.

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 is gplay-1.0 with the following options:

1. Boot up the Linux OS and connect the board to the network.
2. Run *setup_video_stream.sh* (*Section 6.12 Important commands*).
3. Run *gplay_video_stream.sh* and measure:

```
video=HEVC_1920x1080_29.97fps_AACLC_44.1Khz_2ch.mkv
FILE=http://134.27.109.125/$video
gplay-1.0 $FILE
```

4. Start the die temperature recording (*Section 6.12 Important commands*).
5. Measure the power and record the result.

6.4. C-Ray

NOTE

No display was connected to the platform.

6.4.1. Clock configuration

The clock configuration in *Table 39* is aligned with release L4.14.98.

Table 39. C-Ray clock configuration

Clock name	Frequency (MHz)
NOC	750
AXI	333
AHB	133
CPU	1800
DDRC	750

6.4.2. PLL configuration

The PLL configuration in *Table 40* is aligned with release L4.14.98.

Table 40. C-Ray PLL configuration

Clock root	Source selected	Frequency (MHz)
ARM_A53_CLK	PLL_ARM_MAIN_CLK	1800
NOC_CLK	SYS_PLL3_OUT	750
MAIN_AXI_CLK	SYS2_PLL_333M	333
DISP_AXI_CLK	SYS1_PLL_800M	800
ENET_AXI_CLK	SYS1_PLL_266M	266
NAND_USDHC_BUS_CLK	SYS1_PLL_266M	266
AHB_CLK_ROOT	SYS1_PLL_133M	133
IPG_CLK	AHB_ROOT_CLK	66
DRAM_CLK	PLL_DRAM_MAIN_CLK	750
PCIE_CTRL_CLK	SYS2_PLL_250M	OFF
LCDIF_PIXEL_CLK	24M OSC	—
SAIx_CLK	AUDIO_PLL_OUT	36.8
ENETx_REF_CLK	SYS2_PLL_125M	125
ENETx_TIME_CLK	SYS2_PLL_100M	100
ENET_PHY_REF_CLK	24M OSC	24
NAND_CLK	SYS1_PLL_400M	OFF
QSPI_CLK	SYS1_PLL_100M	OFF
USHDCx_CLK	SYS1_PLL_400M	392
I2Cx_CLK	24M OSC	25
UARTx_CLK	24M OSC	25
ECSPIx_CLK	SYS2_PLL_200M	OFF
PWMx_CLK	24M OSC	24
GPTx_CLK	24M OSC	24
TRACE_CLK	24M OSC	OFF
WDOG_CLK	24M OSC	OFF

6.4.3. Steps

1. Boot up the Linux image and boot the board to the SD rootfs.
2. Run `setup.sh` (*Section 6.12 Important commands*).
3. Run `c-ray_loop.sh` and measure:

```
while true; do
cat scene | ./c-ray-mt -t 4 > foo.ppm
done
```

4. Start the die temperature recording (*Section 6.12 Important commands*).
5. Measure the power and record the result.

6.5. Coremark

NOTE

No display was connected to the platform.

6.5.1. Clock configuration

The clock configuration in *Table 41* is aligned with release L4.14.98.

Table 41. Coremark clock configuration

Clock name	Frequency (MHz)
NOC	750
AXI	333
AHB	133
CPU	1800
DDRC	750

6.5.2. PLL configuration

The PLL configuration in *Table 42* is aligned with release L4.14.98.

Table 42. Coremark PLL configuration

Clock root	Source selected	Frequency (MHz)
ARM_A53_CLK	PLL_ARM_MAIN_CLK	1800
NOC_CLK	SYS_PLL3_OUT	750
MAIN_AXI_CLK	SYS2_PLL_333M	333
DISP_AXI_CLK	SYS1_PLL_800M	800
ENET_AXI_CLK	SYS1_PLL_266M	266
NAND_USDHC_BUS_CLK	SYS1_PLL_266M	266
AHB_CLK_ROOT	SYS1_PLL_133M	133
IPG_CLK	AHB_ROOT_CLK	66
DRAM_CLK	PLL_DRAM_MAIN_CLK	750
PCIE_CTRL_CLK	SYS2_PLL_250M	OFF
LCDIF_PIXEL_CLK	24M OSC	—
SAIx_CLK	AUDIO_PLL_OUT	36.8
ENETx_REF_CLK	SYS2_PLL_125M	125
ENETx_TIME_CLK	SYS2_PLL_100M	100
ENET_PHY_REF_CLK	24M OSC	24
NAND_CLK	SYS1_PLL_400M	OFF
QSPI_CLK	SYS1_PLL_100M	OFF
USHDCx_CLK	SYS1_PLL_400M	392
I2Cx_CLK	24M OSC	25
UARTx_CLK	24M OSC	25
ECSPIx_CLK	SYS2_PLL_200M	OFF
PWMx_CLK	24M OSC	24
GPTx_CLK	24M OSC	24
TRACE_CLK	24M OSC	OFF
WDOG_CLK	24M OSC	OFF

6.5.3. Steps

1. Boot up the Linux image and boot the board to the SD rootfs.
2. Run *setup.sh* (*Section 6.12 Important commands*).
3. Run *c-ray_loop.sh* and measure:

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

4. Start the die temperature recording (*Section 6.12 Important commands*).
5. Measure the power and record the result.

6.6. GPU

Two benchmarks were used for GPU power measurements. A 1080p TV display was connected to the MIPI-DSI port. Before running any benchmark, the governor must be set to **performance** (See *Section 6.12 Important commands* for details):

```
cpufreq-set -g performance
```

After setting the governor, run the respective GPU benchmark in a loop and start the power measurements and temperature logging at the desired time interval.

6.6.1. MM07

1. Run *setup_video.sh* (*Section 6.12 Important commands*).
2. Run *gpu_mm07.sh*:

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

3. Start the die temperature recording (*Section 6.12 Important commands*).
4. Start power measurement and record the result.

6.6.1.1. Clock configuration

The clock configuration in *Table 43* is aligned with release L4.14.98.

Table 43. MM07 clock configuration

Clock name	Frequency (MHz)
NOC	750
AXI	333
AHB	133
CPU	1800
DDRC	750

6.6.1.2. PLL configuration

The PLL configuration in *Table 44* is aligned with release L4.14.98.

Table 44. MM07 PLL configuration

Clock root	Source selected	Frequency (MHz)
ARM_A53_CLK	PLL_ARM_MAIN_CLK	1800
NOC_CLK	SYS_PLL3_OUT	750
MAIN_AXI_CLK	SYS2_PLL_333M	333
DISP_AXI_CLK	SYS1_PLL_800M	800
ENET_AXI_CLK	SYS1_PLL_266M	266
NAND_USDHC_BUS_CLK	SYS1_PLL_266M	266
AHB_CLK_ROOT	SYS1_PLL_133M	133
IPG_CLK	AHB_ROOT_CLK	66
DRAM_CLK	PLL_DRAM_MAIN_CLK	750
PCIE_CTRL_CLK	SYS2_PLL_250M	OFF
LCDIF_PIXEL_CLK	24M OSC	—
SAIx_CLK	AUDIO_PLL_OUT	36.8
ENETx_REF_CLK	SYS2_PLL_125M	125
ENETx_TIME_CLK	SYS2_PLL_100M	100
ENET_PHY_REF_CLK	24M OSC	24
NAND_CLK	SYS1_PLL_400M	OFF
QSPI_CLK	SYS1_PLL_100M	OFF
USHDCx_CLK	SYS1_PLL_400M	392
I2Cx_CLK	24M OSC	25
UARTx_CLK	24M OSC	25
ECSPIx_CLK	SYS2_PLL_200M	OFF
PWMx_CLK	24M OSC	24
GPTx_CLK	24M OSC	24
TRACE_CLK	24M OSC	OFF
WDOG_CLK	24M OSC	OFF

6.6.2. MM06

1. Run *setup_video.sh* (*Section 6.12 Important commands*).
2. Run *gpu_mm06.sh*:

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

3. Start the die temperature recording (*Section 6.12 Important commands*).
4. Start power measurement and record the result.

6.6.2.1. Clock configuration

The clock configuration in *Table 45* is aligned with release L4.14.98.

Table 45. MM06 clock configuration

Clock name	Frequency (MHz)
NOC	750
AXI	333
AHB	133
CPU	1800
DDRC	750

6.6.2.2. PLL configuration

The PLL configuration in *Table 46* is aligned with release L4.14.98.

Table 46. MM06 PLL configuration

Clock root	Source selected	Frequency (MHz)
ARM_A53_CLK	PLL_ARM_MAIN_CLK	1800
NOC_CLK	SYS_PLL3_OUT	750
MAIN_AXI_CLK	SYS2_PLL_333M	333
DISP_AXI_CLK	SYS1_PLL_800M	800
ENET_AXI_CLK	SYS1_PLL_266M	266
NAND_USDHC_BUS_CLK	SYS1_PLL_266M	266
AHB_CLK_ROOT	SYS1_PLL_133M	133
IPG_CLK	AHB_ROOT_CLK	66
DRAM_CLK	PLL_DRAM_MAIN_CLK	750
PCIE_CTRL_CLK	SYS2_PLL_250M	OFF
LCDIF_PIXEL_CLK	24M OSC	—
SAIx_CLK	AUDIO_PLL_OUT	36.8
ENETx_REF_CLK	SYS2_PLL_125M	125
ENETx_TIME_CLK	SYS2_PLL_100M	100
ENET_PHY_REF_CLK	24M OSC	24
NAND_CLK	SYS1_PLL_400M	OFF
QSPI_CLK	SYS1_PLL_100M	OFF
USHDCx_CLK	SYS1_PLL_400M	392
I2Cx_CLK	24M OSC	25
UARTx_CLK	24M OSC	25
ECSPIx_CLK	SYS2_PLL_200M	OFF
PWMx_CLK	24M OSC	24
GPTx_CLK	24M OSC	24
TRACE_CLK	24M OSC	OFF
WDOG_CLK	24M OSC	OFF

6.6.3. GPU_Kanzi

1. Run *setup_video.sh* (*Section 6.12 Important commands*).
2. Run *gpu_kanzi.sh*:

```
kanzi_dir=`pwd`"/Kanzi/KPA_1_0_1_137/linux-aarch64"
cd $kanzi_dir
./kanzi.sh
```

   Where *kanzi.sh*:

```
export LD_LIBRARY_PATH="$PWD"
while true;do
  ./kpa.exe
done
```

3. Start the die temperature recording (*Section 6.12 Important commands*).
4. Start power measurement and record the result.

6.6.3.1. Clock configuration

The clock configuration in *Table 47* is aligned with release L4.14.98.

Table 47. GPU_Kanzi clock configuration

Clock name	Frequency (MHz)
NOC	750
AXI	333
AHB	133
CPU	1800
DDRC	750

6.6.3.2. PLL configuration

The PLL configuration in *Table 48* is aligned with release L4.14.98.

Table 48. GPU_Kanzi PLL configuration

Clock root	Source selected	Frequency (MHz)
ARM_A53_CLK	PLL_ARM_MAIN_CLK	1800
NOC_CLK	SYS_PLL3_OUT	750
MAIN_AXI_CLK	SYS2_PLL_333M	333
DISP_AXI_CLK	SYS1_PLL_800M	800
ENET_AXI_CLK	SYS1_PLL_266M	266
NAND_USDHC_BUS_CLK	SYS1_PLL_266M	266
AHB_CLK_ROOT	SYS1_PLL_133M	133
IPG_CLK	AHB_ROOT_CLK	66
DRAM_CLK	PLL_DRAM_MAIN_CLK	750
PCIE_CTRL_CLK	SYS2_PLL_250M	ON
LCDIF_PIXEL_CLK	24M OSC	—
SAIx_CLK	AUDIO_PLL_OUT	36.8
ENETx_REF_CLK	SYS2_PLL_125M	125
ENETx_TIME_CLK	SYS2_PLL_100M	100
ENET_PHY_REF_CLK	24M OSC	24
NAND_CLK	SYS1_PLL_400M	ON

Clock root	Source selected	Frequency (MHz)
QSPI_CLK	SYS1_PLL_100M	ON
USHDCx_CLK	SYS1_PLL_400M	392
I2Cx_CLK	24M OSC	25
UARTx_CLK	24M OSC	25
ECSPIx_CLK	SYS2_PLL_200M	OFF
PWMx_CLK	24M OSC	24
GPTx_CLK	24M OSC	24
TRACE_CLK	24M OSC	ON
WDOG_CLK	24M OSC	ON

6.6.4. GPU_GLmark

1. Run *setup_video.sh* (*Section 6.12 Important commands*).
2. Run *gpu_glmark.sh*:

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

3. Start the die temperature recording (*Section 6.12 Important commands*).
4. Start power measurement and record the result.

6.6.4.1. Clock configuration

The clock configuration in *Table 49* is aligned with release L4.14.98.

Table 49. GPU_GLmark clock configuration

Clock name	Frequency (MHz)
NOC	750
AXI	333
AHB	133
CPU	1800
DDRC	750

6.6.4.2. PLL configuration

The PLL configuration in *Table 50* is aligned with release L4.14.98.

Table 50. GPU_GLmark PLL configuration

Clock root	Source selected	Frequency (MHz)
ARM_A53_CLK	PLL_ARM_MAIN_CLK	1800
NOC_CLK	SYS_PLL3_OUT	750
MAIN_AXI_CLK	SYS2_PLL_333M	333
DISP_AXI_CLK	SYS1_PLL_800M	800
ENET_AXI_CLK	SYS1_PLL_266M	266
NAND_USDHC_BUS_CLK	SYS1_PLL_266M	266
AHB_CLK_ROOT	SYS1_PLL_133M	133
IPG_CLK	AHB_ROOT_CLK	66
DRAM_CLK	PLL_DRAM_MAIN_CLK	750
PCIE_CTRL_CLK	SYS2_PLL_250M	OFF

Clock root	Source selected	Frequency (MHz)
LCDIF_PIXEL_CLK	24M OSC	—
SAIx_CLK	AUDIO_PLL_OUT	36.8
ENETx_REF_CLK	SYS2_PLL_125M	125
ENETx_TIME_CLK	SYS2_PLL_100M	100
ENET_PHY_REF_CLK	24M OSC	24
NAND_CLK	SYS1_PLL_400M	OFF
QSPI_CLK	SYS1_PLL_100M	OFF
USHDCx_CLK	SYS1_PLL_400M	392
I2Cx_CLK	24M OSC	25
UARTx_CLK	24M OSC	25
ECSPIx_CLK	SYS2_PLL_200M	OFF
PWMx_CLK	24M OSC	24
GPTx_CLK	24M OSC	24
TRACE_CLK	24M OSC	OFF
WDOG_CLK	24M OSC	OFF

6.7. Heavy-load use cases

Four use cases were used for power measurements. A 1080p TV display was connected to the MIPI-DSI port. Before running any benchmark, the governor must be set to **performance** (see *Section 6.12 Important commands* for details):

```
cpufreq-set -g performance
```

After setting the governor, run the respective use case in a loop and start power measurements and temperature logging at the desired time interval (recommended is 1 minute) according to *Section 6.12 Important commands*.

6.7.1. VPU

A 1080p TV display was connected to the MIPI-DSI port. Before running any benchmark, the governor must be set to **performance** (see *Section 6.12 Important commands* for details):

```
cpufreq-set -g performance
```

After setting the governor, run the respective use case in a loop and start power measurements and temperature logging at the desired time interval (recommended is 1 minute) according to *Section 6.12 Important commands*.

1. Run *setup_video.sh* (*Section 6.12 Important commands*).
2. Run *vpu_g2dec.sh*:

```
EXT_LOOP=10000
FILE=HEVC_1920x1080_29.97fps_AACLC_44.1Khz_2ch.mkv
export LD_LIBRARY_PATH=/unit_tests/VPU/hantro/:$LD_LIBRARY_PATH
while true;do
cnt=1
 while [ $cnt -le $EXT_LOOP ]
 do
    /unit_tests/VPU/hantro/g2dec -P -Ers -ibs -N200 -X -b $FILE
   cnt=$(($cnt+1));
 done
done
```

3. Start the die temperature recording (*Section 6.12 Important commands*).
4. Start power measurement and record data.

6.7.2. 4-core Dhryst + VPU + Taiji

A 1080p TV display was connected to the MIPI-DSI port. Before running any benchmark, the governor must be set to **performance** (see *Section 6.12 Important commands* for details):

```
cpufreq-set -g performance
```

After setting the governor, run the respective use case in a loop and start power measurements and temperature logging at the desired time interval (recommended is 1 minute) according to *Section 6.12 Important commands*.

1. Run *setup_video.sh* (*Section 6.12 Important commands*).
2. Start four Dhrystone, each bind on separate CPU:

```
while [ "1" == "1" ]
do
sudo taskset -c 0 ./Dhrystone/gcc_dry2 &
sudo taskset -c 1 ./Dhrystone/gcc_dry2 &
sudo taskset -c 2 ./Dhrystone/gcc_dry2 &
sudo taskset -c 3 ./Dhrystone/gcc_dry2
done
```

3. Start Taiji use case in a loop (*Section 6.6.1 MM07*).
4. Start VPU use case (*Section 6.7.1 VPU*).
5. Start the die temperature recording (*Section 6.12 Important commands*).
6. Start power measurement and record data.

6.7.3. 4-core Memtest + VPU + Taiji

A 1080p TV display was connected to the MIPI-DSI port. Before running any benchmark, the governor must be set to **performance** (see *Section 6.12 Important commands* for details):

```
cpufreq-set -g performance
```

After setting the governor, run the respective use case in a loop and start power measurements and temperature logging at the desired time interval (recommended is 1 minute) according to *Section 6.12 Important commands*.

1. Run *setup_video.sh* (*Section 6.12 Important commands*).
2. Start four memtesters, each bind on separate CPU:

```
while [ "1" == "1" ]
do
sudo taskset -c 0 memtester 200M &
sudo taskset -c 1 memtester 200M &
sudo taskset -c 2 memtester 200M &
sudo taskset -c 3 memtester 200M
done
```

3. Start Taiji use case in a loop (*Section 6.6.1 MM07*).
4. Start VPU use case (*Section 6.7.1 VPU*).
5. Start the die temperature recording (*Section 6.12 Important commands*).

6. Start power measurement and record data.

6.7.4. 4-core Streamcpy + VPU + Taiji

A 1080p TV display was connected to the MIPI-DSI port. Before running any benchmark, the governor must be set to **performance** (see *Section 6.12 Important commands* for details):

```
cpufreq-set -g performance
```

After setting the governor, run the respective use case in a loop and start power measurements and temperature logging at the desired time interval (recommended is 1 minute) according to *Section 6.12 Important commands*.

1. Run *setup_video.sh* (*Section 6.12 Important commands*) .
2. Start four streams, each bind on a separate CPU.

```
while [ "1" == "1" ]
do
sudo taskset -c 0 stream -M 200M -N 1000 &
sudo taskset -c 1 stream -M 200M -N 1000 &
sudo taskset -c 2 stream -M 200M -N 1000 &
sudo taskset -c 3 stream -M 200M -N 1000
done
```

3. Start Taiji use case in a loop (*Section 6.6.1 MM07*).
4. Start VPU use case (*Section 6.7.1 VPU*).
5. Start the die temperature recording (*Section 6.12 Important commands*).
6. Start power measurement and record data.

6.7.5. Coremark + Kanzi

A 1080p TV display was connected to the MIPI-DSI port. Before running any benchmark, the governor must be set to **performance** (see *Section 6.12 Important commands* for details):

```
cpufreq-set -g performance
```

After setting the governor, run the respective use case in a loop and start power measurements and temperature logging at the desired time interval (recommended is one minute) according to *Section 6.12 Important commands*.

1. Run *setup_video.sh* (*Section 6.12 Important commands*) .
2. Start Coremark use case in a loop. (*Section 6.5 Coremark*)
3. Start GPU_Kanzi use case (*Section 6.6.3 GPU_Kanzi*).
4. Start the die temperature recording (*Section 6.12 Important commands*).
5. Start power measurement and record data.

6.8. Memory

NOTE

No display was connected to the platform.

Three use cases were used for power measurements. Before running any benchmark, the governor must be set to **performance** and the display must be turned off (see *Section 6.12 Important commands* for details):

```
cpufreq-set -g performance
echo 1 > /sys/class/graphics/fb0/blank
```

After setting the governor, run the respective use case in a loop and start power measurements and temperature logging at the desired time interval (recommended is one minute) according to *Section 6.12 Important commands*.

6.8.1. Memset

1. Run *setup.sh* (*Section 6.12 Important commands*).
2. Run *memset_loop.sh*:

```
while true; do
    perf bench -f simple mem memset -l10000 -s 1024MB
done
```

3. Start the die temperature recording (*Section 6.12 Important commands*).
4. Start power measurement and record data.

6.8.2. Memcpy

1. Run *setup.sh* (*Section 6.12 Important commands*).
2. Run *memcpy_loop.sh*:

```
while true; do
    perf bench -f simple mem memcpy -l10000 -s 1024MB
done
```

3. Start the die temperature recording (*Section 6.12 Important commands*).
4. Start power measurement and record data.

6.8.3. Stream

Make sure stream libraries are added to *LD_LIBRARY_PATH*

1. Run *setup.sh* (*Section 6.12 Important commands*).
2. Run *streamcpy_loop.sh*:

```
export LD_LIBRARY_PATH=`pwd`:$LD_LIBRARY_PATH
while true; do
    ./stream
done
```

3. Start the die temperature recording (*Section 6.12 Important commands*).
4. Start power measurement and record data.

6.9. Storage – SD3.0 Card

An SD card was used to run the benchmarks.

6.9.1. DD_RD_SDCARD

1. Run *setup.sh* (*Section 6.12 Important commands*).
2. Copy *dd_read_SD10.sh* on the SD card partition and run it (see below).
3. Start the die temperature recording (*Section 6.12 Important commands*).
4. Start power measurement and record data.

```
#!/bin/bash

# Since we're dealing with dd, abort if any errors occur
set -e

TEST_FILE=${1:-dd_ibs_testfile}
#if [ -e "$TEST_FILE" ]; then TEST_FILE_EXISTS=$?; fi
#TEST_FILE_SIZE=3221225472
#134217728

# Exit if file exists
#if [ -e $TEST_FILE ]; then
#  echo "Test file $TEST_FILE exists, aborting."
#  exit 1
#fi
#TEST_FILE_EXISTS=1

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

# Create test file
#echo 'Generating test file...'
#BLOCK_SIZE=65536
#COUNT=$(($TEST_FILE_SIZE / $BLOCK_SIZE))
#dd if=/dev/urandom of=$TEST_FILE bs=$BLOCK_SIZE count=$COUNT conv=fsync > /dev/null 2>&1

# Header
PRINTF_FORMAT="%8s : %s\n"
printf "$PRINTF_FORMAT" 'block size' 'transfer rate'

# Block sizes of 512b 1K 2K 4K 8K 16K 32K 64K 128K 256K 512K 1M 2M 4M 8M 16M 32M 64M
for BLOCK_SIZE in 512 1024 2048 4096 8192 16384 32768 65536 131072 262144 524288 1048576
2097152 4194304 8388608 16777216 33554432 67108864
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
```

6.9.2. DD_WRT_SDCARD

1. Run *setup.sh* (*Section 6.12 Important commands*).
2. Copy *dd_write_SD10.sh* on the SD card partition and run it (see below).
3. Start the die temperature recording (*Section 6.12 Important commands*).
4. Start power measurement and record data.

```
#!/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=3221225472
#134217728

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'

# Block sizes of 512b 1K 2K 4K 8K 16K 32K 64K 128K 256K 512K 1M 2M 4M 8M 16M 32M 64M
for BLOCK_SIZE in 512 1024 2048 4096 8192 16384 32768 65536 131072 262144 524288 1048576
2097152 4194304 8388608 16777216 33554432 67108864
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)?')

  ## Clean up the test file if we created one
  #if [ $TEST_FILE_EXISTS -ne 0 ]; then rm $TEST_FILE; fi

  # Output the result
```

```
  printf "$PRINTF_FORMAT" "$BLOCK_SIZE" "$TRANSFER_RATE"
done
```

6.10. Storage – eMMC

A partition was created on eMMC and benchmarks were run on it.

6.10.1. DD_RD_eMMC

1. Run `setup.sh` (*Section 6.12 Important commands*).
2. Copy `dd_read_SD10.sh` on eMMC partition and run.
3. Start the die temperature recording (*Section 6.12 Important commands*).
4. Start power measurement and record data.

6.10.2. DD_WRT_eMMC

1. Run `setup.sh` (*Section 6.12 Important commands*).
2. Copy `dd_write_SD10.sh` on eMMC partition and run.
3. Start the die temperature recording (*Section 6.12 Important commands*).
4. Start power measurement and record data.

6.11. Storage – USB2.0

A USB 2.0 was used to run the benchmarks.

6.11.1. DD_RD_USB2.0

1. Run `setup.sh` (*Section 6.12 Important commands*).
2. Copy `dd_read_SD10.sh` on USB 2.0 partition and run.
3. Start the die temperature recording (*Section 6.12 Important commands*).
4. Start power measurement and record data.

6.11.2. DD_WRT_USB2.0

1. Run `setup.sh` (*Section 6.12 Important commands*).
2. Copy `dd_write_SD10.sh` on USB 2.0 partition and run.
3. Start the die temperature recording (*Section 6.12 Important commands*).
4. Start power measurement and record data.

6.12. Important commands

1. Before running a use case, `<configuration_script>.sh` must be run to configure the environment. These are: `setup.sh`, `setup_default.sh`, `setup_video.sh`, `setup_video_stream.sh`, `DDRC_25MHz_setup.sh` (see below).

- `setup.sh`: The CPU frequency is set to the maximum value 1800 MHz to achieve the best performance. Disable the Ethernet, stop the Weston service, and blank the display. Set the maximum amount of data that the kernel reads ahead for a single file to 512 KB.

```
#!/bin/bash
systemctl stop weston.service
cpufreq-set -g performance
echo 1 > /sys/class/graphics/fb0/blank
partitions=`lsblk | awk '{print $1}'|grep  'blk\|sd'|cut -d'-' -f2`
for partition in $partitions; do
        echo 512 > /sys/block/$partition/queue/read_ahead_kb
done
eth_int=`ifconfig -a|grep 'eth'|awk {'print $1'}`
for eth in $eth_int;do
 ifconfig $eth down
done
echo 0 > /sys/devices/platform/busfreq/enable
```

- `setup_default.sh`: The CPU frequency is the default value 1200 MHz. Disable the Ethernet, stop the Weston service, and blank the display. Set the maximum amount of data that the kernel reads ahead for a single file to 512 KB.

```
#!/bin/bash
systemctl stop weston.service
echo 1 > /sys/class/graphics/fb0/blank
partitions=`lsblk | awk '{print $1}'|grep  'blk\|sd'|cut -d'-' -f2`
for partition in $partitions; do
        echo 512 > /sys/block/$partition/queue/read_ahead_kb
done
eth_int=`ifconfig -a|grep 'eth'|awk {'print $1'}`
for eth in $eth_int;do
 ifconfig $eth down
done
echo 0 > /sys/devices/platform/busfreq/enable
```

- `setup_video.sh`: The CPU frequency is set to the maximum value 1800 MHz to achieve the best performance. Disable the Ethernet, stop the Weston service, and awake the display. Set the maximum amount of data that the kernel reads ahead for a single file to 512 KB.

```
#!/bin/bash
export WL_EGL_SWAP_INTERVAL=0
cpufreq-set -g performance
partitions=`lsblk | awk '{print $1}'|grep  'blk\|sd'|cut -d'-' -f2`
for partition in $partitions; do
        echo 512 > /sys/block/$partition/queue/read_ahead_kb
done
eth_int=`ifconfig -a|grep 'eth'|awk {'print $1'}`
for eth in $eth_int;do
 ifconfig $eth down
done
echo 1 > /sys/class/graphics/fb0/blank
echo 0 > /sys/class/graphics/fb0/blank
echo 0 > /sys/devices/platform/busfreq/enable
```

- `setup_video_stream.sh`: The CPU frequency is set to the maximum value 1800 MHz to achieve the best performance. Open the Ethernet to play the video online. Stop the Weston service and awake the display. Set the maximum amount of data that the kernel reads ahead for a single file to 512 KB.

```
#!/bin/bash
export WL_EGL_SWAP_INTERVAL=0
cpufreq-set -g performance
partitions=`lsblk | awk '{print $1}'|grep  'blk\|sd'|cut -d'-' -f2`
for partition in $partitions; do
        echo 512 > /sys/block/$partition/queue/read_ahead_kb
done
eth_int=`ifconfig -a|grep 'eth'|awk {'print $1'}`
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_25MHz_setup.sh`: After running below shell scripts, you will see logs that the DDRC frequency switches between high bus mode 750 MHz and low bus mode 25 MHz, due to DDR DVFS. CPU frequency is set to the minimum value 1200 MHz. Both of DDR DVFS and CPU powersave are aimed at saving power. Disable the Ethernet, stop the Weston service, and blank the display.)

```
#!/bin/bash
systemctl stop weston.service
echo 1 > /sys/class/graphics/fb0/blank
cpufreq-set -g powersave
rmmod qca9377
#echo 8 > /proc/sys/kernel/printk;
eth_int=`ifconfig -a|grep 'eth\|can\|sit'|awk {'print $1'}`
for eth in $eth_int;do
 ifconfig $eth down
done
sleep 5
```

2. In the U-boot console:

- `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, which are called kernel command lines.

3. In the Linux OS console:

- `cat /proc/cmdline`: displays the command line.
- `cat /sys/devices/virtual/thermal/thermal_zone0/temp`: prints the temperature to the screen (the chip should be calibrated).

NOTE

The die temperature value was logged (written) externally, not on the SD card, for not impacting power consumption.

- `cat /sys/kernel/debug/clk/clk_summary`: prints all clocks to the screen.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number: AN12410
Rev. 0
04/2019