Freescale Semiconductor
Application Note

Document Number: AN4509
Rev. 0, 10/2012

i.MX 6Dual/6Quad Power Consumption Measurement

by *Freescale Semiconductor, Inc.*

This application note helps the user design power management systems. Through several use cases, this report illustrates current drain measurements of the i.MX 6Dual/6Quad system-on-chip (SoC), taken on the SABRE SD Freescale board. The reader will be enabled to choose the appropriate power supply domains for the i.MX 6Dual/6Quad SoC and become familiar with the expected SoC power in different scenarios.

Contents

NOTE

Since the data presented in this Application Note is based on empirical measurements on a small sample size, the results presented are not guaranteed.

1 Overview of i.MX 6Dual/6Quad Voltage Supplies

The i.MX 6Dual/6Quad SoC has 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.

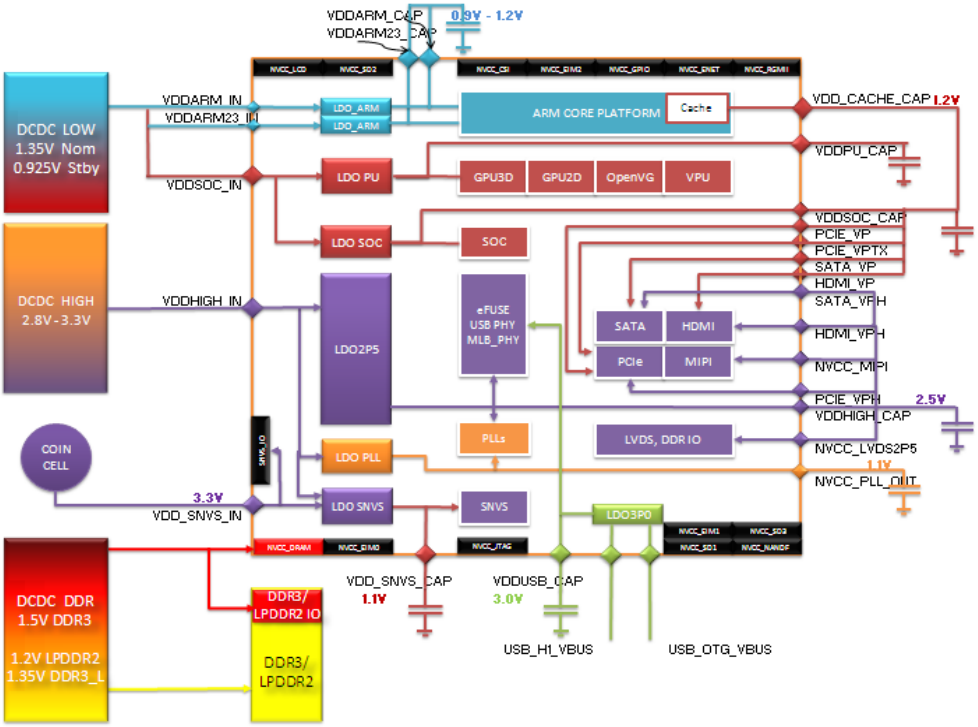

Figure 1. i.MX 6Dual/6Quad Power Rails

NOTE

See the i.MX 6Dual/6Quad datasheet, for the recommended operating conditions of each supply rail and for a detailed description of the groups of I/Os (pins) each I/O voltage supply powers.

For more details regarding the i.MX 6Dual/6Quad power rails, see Power Management Unit (PMU) chapter in the *i.MX 6Dual/6Quad Applications Processors Reference Manual (IMX6DQRM).*

2 Internal Power Measurement of the i.MX 6Dual/6Quad Processor

Several use cases (described in Section 3, “Use Cases and Measurement Results”) are run on the SABRE SD board. The measurements are taken mainly for the following power supply domains—VDDARM_IN, which is the ARM platform’s supply, VDDSOC_IN, which is the peripheral supply, VDDHIGH_IN,

which is the source of PLLs, DDR pre-drives, PHYs, and some other circuitries. These supply domains consume the majority of the internal power of the processor. For the relevant use cases, the power of additional supply domains are added. However, the power of these supply domains does not depend on specific use cases, but whether these modules are used or not. The power consumption of SNVS is comparatively negligible except in Deep Sleep Mode.

The NVCC_* power consumption depends primarily on the board level configuration and the components. Therefore, it is not included in the i.MX 6Dual/6Quad internal power analysis. The power of NVCC_DRAM is added for reference.

The power consumption for these supplies, in different use cases, is provided in Table 2 through Table 16.

NOTE

Unless stated otherwise, all the measurements are done on typical process silicon, at room temperature (26 °C approximately).

2.1 VDDHIGH Power

The voltage VDDHIGH domain is generated from the 2.5V LDO (LDO2.5).

This domain powers the following circuits: On-chip LDOs, Bandgap, MLB, eFUSE, Analog part of the PLLs.

It may also power the following domains (depends on board connectivity):

- Pre-drivers of the DDR I/Os
- SATA, PCIe, MIPI, and HDMI PHYs
- LVDS bridge
- Differential input buffers of the DDR I/O

2.2 DDR I/O Power

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

- 1.5 V for DDR3
- 1.2 V for LPDDR2
- 1.35 V for DDR3_L

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

- Amount of activity of the DDR interface
- On-die termination (ODT)—Enabled/disabled, termination value, which is used for the DDR controller and DDR memories
- Board termination for DDR control and address bus
- Configuration of the DDR pads (such as, drive strength)
- Board layout

- Load of the DDR memory devices

NOTE

- Due to the above mentioned reasons, the measurements provided in the following tables would vary from one system to another. The data provided is for guidance only and should not be treated as a specification.
- The measured current on the Freescale SABRE SD board also includes the current of the on-board DDR3 memory devices. This board (on which the measurements were taken) includes four DDR3 devices, having a total capacity of 1GB. For power optimized systems that use LPDDR2 memories, the power consumed by the DDR I/O and DDR memories would be significantly lower. The SD board utilizes a “T” topology for board memory routing that does not require board level resistor terminations. This further reduces the DDR I/O power usage.

2.2.1 On-Die Termination (ODT) Settings

On-Die Termination (ODT) is a feature of the DDR3 SDRAM that allows the DRAM to turn on/off termination resistance for each DQ, DQS, DQS#, and DM signal. The ODT feature is designed to improve signal integrity of the memory channel by allowing the DRAM controller to independently turn on/off termination resistance for any or all DRAM devices.

Using weaker ODT settings can greatly reduce the power of the DDR I/O. The required ODT settings are system dependent and may vary among different board designs. These settings should be carefully selected to balance between power optimization while ensuring that JEDEC requirements for the DDR parameters are still met. Thus, the default settings that are used in the Linux BSP release may need to be modified by the system designer to fit different systems.

NOTE

There are two major differences in this version of the document and the previous version (Rev. A), which are as follows:

- Updated DDR Memory Die Revision: The DDR devices on SABRE SD Rev B4 are of type MT41K128M16JT-125:K (Die Rev K) which is newer, while for Rev. A of the power app note, the devices which were used on the board were MT41J128M16HA-15E:D (Die Rev. D). In some cases, the newer die consumes ~50% less power than the previous one.
- Optimized ODT settings: The ODT (On-Die Termination) that was used for measurements on RevA of the this document is 120 Ω for the memory and 60 Ω for the i.MX 6Dual/6Quad DDR I/O.
  For Rev. 0 of this document, the ODT values were optimized for further power reduction. Some of the measurements were taken using 120 Ω for the memory and 120 Ω for the i.MX 6Dual/6Quad DDR I/O and in some of the measurements the ODT for the i.MX 6Dual/6Quad DDR I/O was

disabled. The ODT values that were used are indicated in the tables with the measurements results.

2.3 Voltage Levels and DVFS Usage in Measurement Process

The voltage levels of all the supplies, except for VDDARM and VDDSOC, are set to the typical voltage levels as defined in i.MX 6Dual/6Quad datasheet.

The VDDARM and VDDSOC supplies require special explanations. To save power, VDDARM voltage is changed using DVFS (dynamic voltage and frequency scaling), during the run time of the use cases. The voltage levels of these supplies can be changed to standby voltage levels during low power modes.

2.3.1 VDDARM Voltage Levels

The target voltage levels for VDDARM can vary according to the DVFS setpoint used, which is selected by the DVFS (also named as CPUFREQ) driver. There are several factors that contribute to the set-point decisions, CPU load being the most important. Other factors are CPU latency requirements, thermal restrictions, peripheral I/O performance requirements. The voltage presented below are on VDDARM_CAP. The voltage and frequency setpoints used for the measurements are given in Table 1.

NOTE

See the operating table in the i.MX 6Dual/6Quad datasheet for the official operating points.

Most of the measurements are performed using these voltage levels and the power data that appears in this document is according to these values. In case the measurement is done at different voltage levels, the results scale accordingly. In real applications when DVFS is applied, the frequency and voltage values are automatically adjusted according to 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.

2.3.2 VDDSOC Voltage Levels

The approximate nominal target voltage levels for VDDSOC_IN is 1.425 V when LDO_SOC is used, and 1.25 V when LDO_SOC is bypassed. See Table 1 for the VDDSOC_CAP and VDDPU_CAP settings used in the measurements. See the Operating Ranges table in i.MX 6Dual/6Quad datasheet for the official operating points.

Table 1. VDDARM, VDDSOC, and VDDPU Voltage Levels (for Reference Only)

VPU Frequency	ARM Frequency	LDO State	VDDARM_IN	VDDARM_CAP	VDDSOC_IN	VDDSOC_CAP/ VDDPU_CAP
0 to 352 MHz	996 MHz	Enabled	1.425 V	1.250 V	1.425 V	1.250 V
264->352 MHz	792 MHz	Enabled		1.150 V		1.250 V
0 to 264 MHz		Enabled		1.150 V		1.175 V
264->352 MHz	396 MHz	Enabled		0.950 V		1.250 V
<264 MHz		Enabled		0.950 V		1.175 V

2.4 Temperature Measurements

In some of the use cases, the die temperature is measured. The temperature measurements were taken using the on-chip thermal sensor, on a thermally calibrated part. While measuring temperature, it is recommended to wait until the temperature stabilizes.

NOTE

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

For more details on thermal aspects, see the application note *AN4579, Thermal Management Guidelines for the i.MX 6Dual/6Quad,* available on the Freescale Extranet.

2.5 Hardware and Software Used

The software versions used for the measurement are as follows:

- For most Linux use cases, Gnome (Linux BSP version GA1209) based on Linux kernel version 3.0.35 was used. For other use cases, ER1203 and ER1205, based on Linux kernel version 3.0.15, were used.
- Android version used: Android R13.4-GA based on:
  - Google Ice Cream Sandwich 4.0.4 r1.1 release
  - Linux kernel version 3.0.35
- The board used for the measurements is the Freescale SABRE SD RevB4 board.
- The measurements were performed using Agilent 34401A 6 ½ Digit Multimeter.

2.6 Board Setup used for Power Measurements

The power measurements are taken using the following input voltages of the supplies.

- VDDARM_IN and VDDSOC_IN at 1.425 V
- VDDHIGH_IN at 2.78 V
- NVCC_DRAM at 1.5 V

Also, the on-chip LDOs are used which is the recommended settings for simplified and cost effective system. The ARM voltage scaling is done through configuring LDO_ARM.

Thus, by using a different setup, such as configurable and separated DC switcher for ARM, the system power may be further optimized by reducing VDDARM_IN input voltage level and match it to the desired operating point. Such setup would likely result in a higher system cost, so there is a trade off between cost versus system power.

2.7 Measuring Points on the Freescale SABRE SD Board

The power data is obtained by measuring the average voltage drop over the measurement points, and dividing it by the resistor value to get the average current. The tolerance of the 0.02 Ω resistors on the SD board is 1%. The measuring points for the various supply domains are as follows:

- VDDSOC—The SOC domain current is measured on R21 and the recommended resistance value for this measurement is 0.02 Ω.
- VDDCORE—The ARM domain current is measured on R27 and the recommended resistance value for this measurement is 0.02 Ω.
- VDDHIGH—The VDDHIGH domain current is measured on SH17 and the recommended resistance value for this measurement is 0.1 Ω.
- DDR3 I/O plus Memories—The current in this domain includes the NVCC_DRAM current and the overall current of the on-board DDR3 memory devices. The current in this domain is measured on R25 and the recommended resistance value for this measurement is 0.02 Ω.

3 Use Cases and Measurement Results

3.1 Use Cases

The main use cases and subtypes, which form the benchmarks for i.MX 6Dual/6Quad internal power measurements, are as follows:

- Low power mode
  - Deep Sleep mode
  - System Idle mode
  - User Idle mode
- Audio playback
  - MP3 Audio Playback
- Video playback
  - H.264 Video Playback, 1080p on HDMI LCD
  - H.264 Video Playback, 1080p on LVDS LCD
- Dhrystone benchmark
  - Quad core Dhrystone benchmark
  - Dual core Dhrystone benchmark
  - Single core Dhrystone benchmark
- Graphics
  - 3D gaming benchmark, MM06
  - 3D gaming benchmark, MM07
- Typical Max Power
  - Graphics plus dual 1080p video playback
- Non-Multimedia
  - USB to eMMC file transfer

NOTE

All measurements in this Application Note are taken at room temperature of 26 °C unless otherwise noted.

3.2 Low Power Mode Use Cases

3.2.1 Use Case 1— Deep Sleep Mode (DSM)

This mode is named as Dormant mode or Suspend to RAM, in the Linux BSP. This is the lowest possible power state where external supplies are still on.

The use case is as follows:

- ARM platform is power gated.
- L1 Cache periphery is power gated.
- PU regulator is disabled (means that GPUs and VPU are power gated).
- SoC regulator is bypassed.
- All PLLs (phase locked loop) and CCM (clock controller module) generated clocks are off.
- CKIL (32 KHz) input is on.
- All the modules are disabled.
- Well bias is applied.
- All analog PHYs are powered down.
- External high frequency crystal and on chip oscillator are powered down (by asserting SBYOS bit in CCM).
- VDDARM_IN and VDDSOC_IN are dropped to 0.9V by asserting the PMIC_STBY_REQ.

In this mode, no current flow is caused by external resistive loads.

Table 2 shows the measurement results when this use case is applied on the i.MX 6Dual/6Quad processor.

Table 2. DSM Measurement Results

Supply Domain	Voltage (V)	Linux GA1209		Android - R13.4-GA	
		P (mW)	I (mA)	P (mW)	I (mA)
VDDARM_IN	0.9	0	0	0.13	0.15
VDDSOC_IN	0.9	2.43	2.7	4.19	4.65
VDDHIGH_IN	2.78	1.4	0.5[1]	2.92	1.05
Total Power (without DDR3 I/O + Memories)		**3.83**		**7.24**	
DDR3 I/O + Memories[2]	1.5	23.25	15.5	24.6	16.4
Total Power		27.08		31.84	

1 In case wake-up from USB is enabled, STOP_MODE_CONFIG bit in Miscellaneous Control Register in CCM should be set, and the VDDHIGH_IN current would be 1.8 mA. There is no impact on the current of the other power rails mentioned here.

2 The current in this domain includes the NVCC_DRAM current and I/O and memories current of the on-board DDR3 devices. The current for the i.MX 6Dual/6Quad DDR I/O (NVCC_DRAM supply) can be reduced to nearly zero by floating all DDR pins and maintaining CKE0/1 driven low.

NOTE

For additional details on this use case and settings, see Section 5, "Use Cases Configuration and Usage Guidelines."

3.2.2 Use Case 2—System Idle Mode

The use case is as follows:

- ARM is in WFI mode most of the time.
- Some PLLs are on.
- Operating system is on.
- LCD is turned off.
- Screen is not refreshed.

This use case simulates the situation when the device is left idle for some time and the display is turned off after the timer expires.

Table 3 shows the measurement results when this use case is applied on the i.MX 6Dual/6Quad processor.

Table 3. System Idle Mode Measurement Results

Supply Domain	Voltage (V)	Linux - GA1209	
		P (mW)	I (mA)
VDDARM_IN	1.37	16.57	12.1
VDDSOC_IN	1.37	72.06	52.6
VDDHIGH_IN	2.78	139	50
Total Power (without DDR3 I/O + Memories)		**88.63**	
DDR3 I/O + Memories[1]	1.5	40.35	26.9
Total Power		128.98[2]	

1 The ODT settings are—120 Ω for the memory and for the i.MX 6Dual6/Quad DDR I/O the ODT is disabled.

2 The measured die temperature is 35 °C.

NOTE

For additional details on this use case and settings, see Section 5, "Use Cases Configuration and Usage Guidelines."

3.2.3 Use Case 3—User Idle Mode

The use case is as follows:

- ARM is in WFI mode most of the time.
- Some PLLs are on.
- Operating system and LCD are on, but cores are almost not in operation
- The XGA screen refresh is done by IPU through LVDS.

The use case simulates the situation when the device is left idle and no application is performed on the screen (like reading from the screen).

Table 4 shows the measurement results when this use case is applied on the i.MX 6Dual/6Quad processor.

Table 4. User Idle Mode Measurement Results

Supply Domain	Voltage (V)	Linux - GA1209		Android - R13.4-GA	
		P (mW)	I (mA)	P (mW)	I (mA)
VDDARM_IN	1.37	49.46	36.1	13.97	10.2
VDDSOC_IN	1.37	412.37	301	433.61	316.5
VDDHIGH_IN	2.78	166.8	60	179.31	64.5
Total Power (without DDR3 I/O + Memories)		**628.63**		**626.89**	
DDR3 I/O +[1] Memories	1.5	151.5	101	143.7	95.8
Total Power		780.13[2]		770.59[3]	

1 The ODT settings are—120 Ω for the memory and for the i.MX 6Dual6/Quad DDR I/O the ODT is disabled.
2 The measured die temperature is 40 °C
3 The measured die temperature is 36 °C

NOTE

For additional details on this use case and settings, see Section 5, "Use Cases Configuration and Usage Guidelines."

3.3 Audio Playback Use Cases

3.3.1 Use Case 1—MP3 Audio Playback

The use case procedure is as follows:

1. MP3 (MPEG-1 audio layer 3) decoding is done by ARM.
2. Audio playback is run through SSI (serial synchronous interface).
3. The stream 128 Kbps_44 kHz_s_mp3.mp3 is taken from the SD (secure digital) card.

The LCD is turned off after the timer expires.

Table 5 shows the measurement results when this use case is applied on the i.MX 6Dual/6Quad processor.

Table 5. MP3 Audio Playback Measurement Results

Supply Domain	Voltage (V)	Linux - GA1209	
		P (mW)	I (mA)
VDDARM_IN	1.42	39.05	27.5
VDDSOC_IN	1.42	122.12	86
VDDHIGH_IN	2.78	128.71	46.3
Total Power (without DDR3 I/O + Memories)		**289.88**	
DDR3 I/O + Memories	1.5	69.75	46.5[1]
Total Power		359.63[2]	

[1] The ODT settings are—120 Ω for the memory and for the i.MX 6Dual6/Quad DDR I/O the ODT is disabled.
[2] The measured die temperature is 36 °C.

NOTE

For additional details on this use case and settings, see Section 5, “Use Cases Configuration and Usage Guidelines.”

3.4 Video Playback Use Cases

3.4.1 Use Case 1—H.264 Video Playback, 1080p on HDMI LCD

A video from the movie Avatar was taken for this use case. This use case has the following features:

- The video source is H.264, 1080p resolution, 30 FPS, 3.6 Mbps bitrate
- The audio source is AAC 44.1 Kbps stereo.
- VDOA module is used to reduce DDR bus load.
- The display is 1080p resolution using HDMI.

The video/audio stream is loaded from the SD card into the DDR (double data rate) memory and then demuxed by Cortex-A9. The demuxed video signal is decoded by the VPU. It is then taken by IPU and displayed on the LCD display (through HDMI) with a refresh rate of 60 Hz. In parallel, the demuxed audio signal is decoded using Cortex-A9 and is played back through the SSI.

Table 6 shows the measurement results when this use case is applied on the i.MX 6Dual/6Quad processor.

Table 6. 1080P Video Playback Measurement Results

Supply Domain	Voltage (V)	Linux - GA1209	
		P (mW)	I (mA)
VDDARM_IN	1.42	58.36	41.1
VDDSOC_IN	1.42	609.75	429.4
VDDHIGH_IN	2.78	199.05	71.6
Total Power (without DDR3 I/O + Memories)		**867.16**	
DDR3 I/O + Memories	1.5	442.95 304.5	295.3[1] 202.7[2]
Total Power		1310.11[3] 1171.5	

1 The ODT settings are—120 Ω for the memory and 120 Ω for the i.MX 6Dual6/Quad DDR I/O.

2 The ODT settings are—120 Ω for the memory and for the i.MX 6Dual6/Quad DDR I/O the ODT is disabled.

3 The measured die temperature is 46 °C.

NOTE

For additional details on this use case and settings, see Section 5, “Use Cases Configuration and Usage Guidelines.”

3.4.2 Use Case 2—H.264 Video Playback, 1080p on LVDS LCD

The same Avatar video used for the previous use case was used for this use case as well. This use case has the following features:

- The video source is H.264, 1080p resolution, 30 FPS, 3.6 Mbps bitrate
- The audio source is AAC 44.1Kbps stereo.
- VDOA module is used to reduce DDR bus load.
- The display is 1080p resolution using LVDS.

The video/audio stream is loaded from the SD card into the DDR (double data rate) memory and then demuxed by Cortex-A9. The demuxed video signal is decoded by the VPU. It is then taken by IPU and displayed on the LCD display (through LVDS) with a refresh rate of 60 Hz. In parallel, the demuxed audio signal is decoded using Cortex-A9 and is played back through the SSI.

Table 7 shows the measurement results when this use case is applied on the i.MX 6Dual/6Quad processor.

Table 7. 1080P Video Playback Measurement Results on LVDS LCD

Supply Domain	Voltage (V)	Linux - GA1209		Android - R13.4-GA	
		P (mW)	I (mA)	P (mW)	I (mA)
VDDARM_IN	1.42	51.83	36.5	56.52	39.8
VDDSOC_IN	1.42	557.78	392.8	639.99	450.7
VDDHIGH_IN	2.78	166.8	60	185.43	66.7
Total Power (without DDR3 I/O + Memories)		**776.41**		**881.94**	
DDR3 I/O + Memories	1.5	447 297.15	298[1] 198.1[2]	311.25	207.5[3]
Total Power		1223.41[4] 1073.56		1193.19[5]	

1 The ODT settings are—120 Ω for the memory and 120 Ω for the i.MX 6Dual6/Quad DDR I/O.
2 The ODT settings are—120 Ω for the memory and for the i.MX 6Dual6/Quad DDR I/O the ODT is disabled.
3 The ODT settings are—120 Ω for the memory and for the i.MX 6Dual6/Quad DDR I/O the ODT is disabled.
4 The measured die temperature is 44 °C.
5 The measured die temperature is 41 °C.

NOTE

For additional details on this use case and settings, see Section 5, “Use Cases Configuration and Usage Guidelines.”

3.5 Dhrystone Benchmark

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

3.5.1 Use Case 1—Quad Core Dhrystone Benchmark

In this use case, Dhrystone test is performed by all four cores. The ARM processor runs the test in a loop at a frequency of 1 GHz.

Table 8 shows the measurement results when this use case is applied on the i.MX 6Dual/6Quad processor.

Table 8. Quad Core Dhrystone Benchmark Measurement Results

Supply Domain	Voltage (V)	Linux - GA1209	
		P (mW)	I (mA)
VDDARM_IN	1.42	2272	1600
VDDSOC_IN	1.42	451.56	318
VDDHIGH_IN	2.78	140.95	50.7
Total Power (without DDR3 I/O + Memories)		**2864.51**	
DDR3 I/O + Memories	1.5	40.2	26.8[1]
Total Power		2904.71[2]	

1 The ODT settings are—120 Ω for the memory and for the i.MX 6Dual6/Quad DDR I/O the ODT is disabled.

2 The measured die temperature is 63 °C.

3.5.2 Use Case 2—Dual Core Dhrystone Benchmark

In this use case, Dhrystone test is performed by two cores. The ARM processor runs the test in a loop at a frequency of 1 GHz.

Table 9 shows the measurement results when this use case is applied on the i.MX 6Dual/6Quad processor.

Table 9. Dual core Dhrystone benchmark Measurement Results

Supply Domain	Voltage (V)	Linux - GA1209	
		P (mW)	I (mA)
VDDARM_IN	1.42	1221.2	860
VDDSOC_IN	1.42	404.7	285
VDDHIGH_IN	2.78	139	50
Total Power (without DDR3 I/O + Memories)		**1764.9**	
DDR3 I/O + Memories	1.5	40.65	27.1[1]
Total Power		1805.55[2]	

[1] The ODT settings are—120 Ω for the memory and for the i.MX 6Dual6/Quad DDR I/O the ODT is disabled..

[2] The measured die temperature is 44 °C.

3.5.3 Use Case 3—Single Core Dhrystone Benchmark

In this use case, Dhrystone test is performed by a single core. The ARM processor runs the test in a loop at a frequency of 1 GHz. Run power of ARM is monitored.

Table 10 shows the measurement results when this use case is applied on the i.MX 6Dual/6Quad processor.

Table 10. Single Core Dhrystone Benchmark Measurement Results

Supply Domain	Voltage (V)	Linux- ER1203	
		P (mW)	I (mA)
VDDARM_IN at 1 GHz	1.37	706.9	516

3.6 Graphics Use Cases

3.6.1 Use Case 1—3D Gaming Benchmark, MM06

This use case has the following features:

- VGA resolution, using MM06 (Samurai) benchmark.
- The frame rate is 175 FPS.
- The display is of XGA resolution using LVDS

The graphics is loaded from the SD card into the DDR (double data rate) memory, processed by the GPU3D, then copied to display buffer in the DDR. It is then processed by the IPU and displayed on the LCD display (through LVDS) with a refresh rate of 60 Hz.

Table 11. 3D Gaming MM06 Benchmark Measurement Results

Supply Domain	Voltage (V)	Linux - GA1209	
		P (mW)	I (mA)
VDDARM_IN	1.37	219.2	160
VDDSOC_IN	1.37	1272.32	928.7
VDDHIGH_IN	2.78	202.94	73
Total Power (without DDR3 I/O + Memories)		**1694.46**	
DDR3 I/O + Memories	1.5	807 591	538[1] 394[2]
Total Power		2501.46[1,3] 2285.46[2]	

1 The ODT settings are—120 Ω for the DDR memory and 120 Ω for the i.MX 6Dual6/Quad DDR I/O.

2 The ODT settings are—120 Ω for the DDR memory and for the i.MX 6Dual6/Quad DDR I/O the ODT is disabled.

3 The measured die temperature is 41 °C.

In the above use case, measurements were taken with DVFS disabled and CPU speed set to 396 MHz. In another example, we enable DVFS and then take the measurements. In this case, frame rate is 260 FPS. The results with these settings are shown in Table 12.

Table 12. 3D Gaming MM06 Benchmark Measurement Results

Supply Domain	Voltage (V)	Linux - GA1209	
		P (mW)	I (mA)
VDDARM_IN	1.37	350.72	256
VDDSOC_IN	1.37	1565.91	1143
VDDHIGH_IN	2.78	190.71	68.6
Total Power (without DDR3 I/O + Memories)		**2107.34**	
DDR3 I/O + Memories	1.5	757.5	505[1]
Total Power		2864.84	

1 The ODT settings are 120 Ω for the DDR memory and for the i.MX 6Dual6/Quad DDR I/O the ODT is disabled.

NOTE

For additional details on this use case and settings, see Section 5, "Use Cases Configuration and Usage Guidelines."

3.6.2 Use Case 2— 3D Gaming Benchmark, MM07

This use case has the following features:

- The graphics used is of VGA resolution, using the MM07 (Taji Girl) benchmark.
- The frame rate is 24 FPS
- The display is of XGA resolution using LVDS

The graphics is loaded from the SD card into the DDR (double data rate) memory, processed by the GPU3D, then copied to display buffer in the DDR. It is then taken by IPU and displayed on the LCD display (through LVDS) with a refresh rate of 60 Hz.

Table 13. 3D Gaming MM07 Benchmark Measurement Results

Supply Domain	Voltage (V)	Linux - GA1209	
		P (mW)	I (mA)
VDDARM_IN	1.37	220.57	161
VDDSOC_IN	1.37	1126.55	822.3
VDDHIGH_IN	2.78	195.99	70.5
Total Power (without DDR3 I/O + Memories)		**1543.11**	
DDR3 I/O + Memories	1.5	567.6 390.75	378.4[1] 260.5[2]
Total Power		2110.71[1,3] 1933.86[2]	

1 The ODT settings are—120 Ω for the DDR memory and 120 Ω for the i.MX 6Dual6/Quad DDR I/O.

2 The ODT settings are—120 Ω for the DDR memory and for the i.MX 6Dual6/Quad DDR I/O the ODT is disabled.

3 The measured die temperature is 41 °C.

In the above use case, measurements were taken with DVFS disabled and CPU speed set to 396 MHz. In another example, we enable DVFS and then take the measurements. In this case, frame rate is 44.2 FPS. The results with these settings are shown in Table 14.

Table 14. 3D Gaming MM07 Benchmark Measurement Results

Supply Domain	Voltage (V)	Linux - GA1209	
		P (mW)	I (mA)
VDDARM_IN	1.37	458.95	335
VDDSOC_IN	1.37	1628.93	1189
VDDHIGH_IN	2.78	183.48	66
Total Power (without DDR3 I/O + Memories)		**2271.36**	
DDR3 I/O + Memories	1.5	493.5	329[1]
Total Power		2764.86[2]	

1 The ODT settings are 120 Ω for the DDR memory and for the i.MX6Dual6/Quad DDR I/O the ODT is disabled.

2 The measured die temperature is 41 °C.

NOTE

For additional details on this use case and settings, see Section 5, "Use Cases Configuration and Usage Guidelines."

3.7 Typical Max Power

3.7.1 Graphics Plus Dual 1080p Video Playback

The purpose of this use case is to provide the power consumption of a very intensive use case, which is highly atypical, but perhaps could be relevant for some systems when planning to work under extreme conditions.

This use case is running concurrently on three displays:

- Two 1080p video playback: One through HDMI and one through IPU parallel port
- 3D graphics through LVDS port with XGA resolution

This use case has the following features:

- The video source is H.264, 1080p resolution, 15 FPS, 10 Mbps bitrate (no audio).
- The graphics is 3D gaming benchmark—MM06.
- All four ARM cores are heavily loaded.
- Maximum frequencies are used for ARM, IPU, VPU, GPUs, and DDR clocks.

The video stream is loaded from the SD card into the DDR memory. The video input is decoded twice by the VPU (but not displayed on the screen). In addition, two Cortex-A9 cores are used to perform concurrent software decoding of the input video stream. The software decoding is needed since the existing BSP version still does not support this. VPU decoding is done here as a background activity to consume power.

The decoded streams are then taken by IPU1 and IPU2 and displayed on the two LCD displays (through HDMI and through the parallel interface) with a refresh rate of 60 Hz. The GPU3D is used to render the graphics. Then, the graphics is displayed by IPU2 through LVDS on XGA display. One of Cortex-A9 is running Dhrystone pattern in a loop in the background.

Table 15. Typical Max Power Measurement Results—on SD Board

Supply Domain	Voltage (V)	Linux - ER1205 - on SD Board[1]	
		P (mW)	I (mA)
VDDARM_IN	1.37	2068.7	1510 (1625 max[2])
VDDSOC_IN	1.37	1555	1135 (1250 max[2])
VDDHIGH_IN	2.78	236.3	85
Total Power (without DDR3 I/O + Memories)		**3860**	
DDR3 I/O +[3] Memories	1.5	1995	1330 (1390 max[2])
Total Power		5855	

1 The current also depends on the silicon temperature, which depends on the heat dissipation in the system. The measured die temperature for this use case is approximately 80 degrees celsius.

2 This is a maximum current measured over a small period of time to present the sustained peak current for the supply in this measurement. Still, there would be variations from part to part under different PVT conditions.

3 The ODT (On-Die Termination) that was used for measurements is 120 Ω for the memory and 60 Ω for the i.MX 6Dual6/Quad DDR I/O. The DDR I/O power may be further reduced by using optimized ODT settings of the i.MX 6Dual6/Quad DDR I/O and the DDR memory I/O. Optimization needs to be done per system.

For additional reference, see results below, taken on internal Freescale validation board. On this board, NVCC_DRAM is separated from DDR memory devices, so i.MX6 DDR I/O is measured here separately.

Table 16. Typical Max Power Measurement Results on Freescale Board

Supply Domain	Voltage (V)	Linux - ER1204 - on Freescale Board[1]	
		P (mW)	I (mA)
VDDARM_IN	1.37	1897.5	1385 (1450 max[2])
VDDSOC_IN	1.37	1541.3	1125 (1250 max[2])
VDDHIGH_IN	2.78	233.5	84
Total Power (without DDR3 I/O)		**3672.3**	
MX6Q/D DDR3 I/O[3]	1.5	1237.5	825
Total Power		4909.8	

1 This board has eight DDR devices, having a total of 2GB. In addition, the DDR control and address lines have on board termination, which consumes additional DDR I/O current.

2 This is a maximum current measured over a small period of time to present the sustained peak current for the supply in this measurement. Still, there would be variations from part to part under different PVT conditions.

3 The ODT (On Die Termination) that was used for measurements is 120 Ω for the memory and 60 Ω for the i.MX 6Dual6/Quad DDR I/O. The DDR I/O power may be further reduced by using optimized ODT settings of the i.MX 6Dual6/Quad DDR I/O and the DDR memory I/O. Optimization needs to be done per system.

3.8 Non-Multimedia Use Cases

3.8.1 USB to eMMC File Transfer

The use case is as follows:

In this use-case, a total 1GB of data is being transferred from USB device to eMMC device. A data size of 1 MB is copied each time, repeatedly for 1000 times. The SDMA is being used to perform the data transfer to the eMMC host controller.

Table 17. USB to eMMC File Transfer Measurement Results

Supply Domain	Voltage (V)	Linux - GA1209	
		P (mW)	I (mA)
VDDARM_IN	1.37	206.46	150.7
VDDSOC_IN	1.37	392.51	286.5
VDDHIGH_IN	2.78	156.24	56.2
Total Power (without DDR3 I/O + Memories)		**755.2**	
DDR3 I/O + Memories	1.5	222.45	148.3[1]
Total Power		977.65	

1 The ODT settings are—120 Ω for the memory and for the i.MX 6Dual6/Quad DDR I/O the ODT is disabled.

4 Reducing Power Consumption

The overall system power consumption depends on both software optimization and how the system hardware is implemented. Below is a list of suggestions which may help to reduce system power. Part of this is already implemented in Linux BSP. Further optimizations can be done on individual customer's system.

NOTE

Further power optimizations are planned to be implemented in future BSP releases. See Freescale website to obtain the latest BSP release.

- Apply clock gating whenever clocks or modules are not used, by configuring CCGR registers in the CCM (Clock Controller Module).
- Reduce the number of operating PLLs—Applicable mainly in Audio Playback mode or Idle modes.
- Core DVFS and system bus scaling—Applying DVFS for ARM and scaling the frequencies of the AXI, AHB, and IPG bus clocks can significantly reduce the power consumption of VDDGP domain and VCC domain, respectively. However, due to the reduced operation frequency, the accesses to the DDR take longer, which increases the power consumption of the DDR I/O and memories. This trade off needs to be taken into account for each mode, to quantify the overall affect on system power.
- Put i.MX 6Dual/6Quad into low power modes (WAIT, STOP) whenever possible. See the CCM chapter of i.MX 6Dual/6Quad reference manual for details.
- DDR interface optimization:
  - Use careful board routing of the DDR memories, maintaining PCB trace lengths as short as possible.
  - Use a reduced ODT (On-Die Termination) setting, as possible. The termination which is used greatly influences the power consumption of the DDR interface pins.
  - Use proper output driver impedance for DDR interface pins which provides good impedance matching. Select the lowest possible drive strength, which still provides the required performance, in order to save current through DDR I/O pins.
  - Carefully choose on-board resistors so the least amount of current is wasted, for example, when selecting impedance matching resistors between CLK and CLK_B (when using DDR3 memories).
  - When possible, at lower performance use cases, switch to DLL off mode, which allows to greatly reduce DDR frequency and thus disable or reduce termination and reduce the drive strength, which significantly reduces the power consumption of the DDR interface pins.
  - Float i.MX 6Dual/6Quad DDR interface pins (set to high Z) when DDR memory is in self refresh mode, keeping DDR_SDCKE0 and DDR_SDCKE1 at low value, if done using external pull-down, need to make sure there is no on board termination on these pins during this mode.
  - If possible (depending on system stability), configure DDR input pins to CMOS mode, instead of differential mode. This can be done by clearing DDR_INPUT bit in the corresponding registers in IOMUXC. This setting is mostly recommended when operating at low frequencies, such as in DLL off mode.

— Use of DDR3L memory devices, operating at low I/O voltage, can further reduce the I/O power by 20%.

— Use of 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 involved in floating of i.MX 6Dual/6Quad DDR interface pins are given below.

NOTE

All the above programming steps are performed when the code is running from the internal RAM rather than from the DDR memory. The code is non-cacheable.

Steps to be performed before entering suspend (Deep Sleep mode):

1. Read the power saving status in MMDC in MAPSR register, since automatic power saving is enabled, to make sure that DDR is in self-refresh.
2. Do the following:
   a) In case, there is no on board termination for DDR control and address bus, set the DSE (drive strength selection, in IOMUXC) for all DDR IF I/O to 0 (High Z), except for CKE0 and CKE1.
   b) In case, the DDR control and address bus have on board termination resistors connected to VTT, such as in the case where SODIMM is used:

   — Option 1
   – As for (a), keep SDCKE0/1 active, this will cause some extra current from the pins sharing the same DSE control in IOMUXC_SW_PAD_CTL_GRP_CTLDS register. The pins are DRAM_CS0, DRAM_CS1, DRAM_SDBA2, DRAM_SDCKE0, DRAM_SDCKE1, and DRAM_SDWE.

   — Option 2 (requires on board pull down resistor on DRARM_SDCKE0/1 pins)
   – Set the Supply of the termination resistor to be floated (can be done through some pins with GPIO capability on it).
   – Set the DSE (drive strength selection, in IOMUXC) for all DDR IF I/O to 0 (High Z).
3. Get into the suspend mode.

Steps to be performed after exiting suspend:

1. Restore all the settings for the DDR I/O to the required value.
2. System proceeds to run mode.

NOTE

In case system can make sure there are no masters accessing the DDR, the following may be applied to other scenarios besides Deep Sleep mode.

DDR pins can be floated in the same manner, even when suspend is not entered and DDR can be manually put into self refresh, to save power. This happens when CPU is not running, or it is running from the internal RAM.

5 Use Cases Configuration and Usage Guidelines

5.1 HDMI 1080P Play Back (VPM3)

5.1.1 Clocks Configuration

Clocks configuration in Table 18 is aligned with release ER1209RC1.

Table 18. Clocks Configuration

Clock Name	Frequency (MHz)
AXI	264
AHB	132
CPU	Varies between DVFS points
GPU2D	off
GPU3D Core	off
GPU3D Shader	off
VPU	off
IPU1	264
IPU2	off
MMDC CH0	528
MMDC CH1	off

5.1.2 PLLs Configuration

PLLs configuration in Table 19 is aligned with release ER1209RC1.

Table 19. PLLs Configuration

PLL Name	Frequency (MHz)
PLL1—System PLL	996
PLL2—System Bus PLL	528
pll2 396m pfd	396
pll2 352m pfd	452
pll2 594m pfd	off
PLL3—OTG USB PLL	480
pll3 508m pfd	off
pll3 454m pfd	off
pll3 720m pfd	off
pll3 540m pfd	540

Table 19. PLLs Configuration (continued)

PLL Name	Frequency (MHz)
PLL4—Audio PLL	off
PLL5—Video PLL	891
PLL6—MLB PLL	off
PLL7—Host USB PLL	off
PLL8—ENET PLL	off

5.1.3 System Setup

1. Disconnect LVDS.
2. Input video used for measuring is `fsl-clip-1080p.264`.

5.1.4 Steps

1. Power on the board and in the serial console press any key to stop autoboot.
2. `setenv bootargs 'console=ttymxc3,115200 vmalloc=256M enable_wait_mode=on'`
3. `setenv bootargs_base 'bootargs ${bootargs} fec_mac=${ethaddr} ${hdmi_mode}'`
4. `setenv hdmi_mode 'video=mxcfb0:dev=hdmi,1920x1080M@60,if=RGB24'`
5. `saveenv`
6. Restart the board.
7. Run the script below to set the system into right work point.

```
#!/bin/sh
echo 1 > /sys/class/graphics/fb0/blank
echo 1 > /sys/class/graphics/fb1/blank
echo 1 > /sys/class/graphics/fb2/blank
echo 1 > /sys/class/graphics/fb3/blank
echo 1 > /sys/class/graphics/fb4/blank
ifconfig eth0 down
echo 1 > /sys/devices/platform/imx_busfreq.0/enable
echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
echo 396000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed
echo 0 > /sys/class/graphics/fb0/blank
```

8. Run `test - /unit_tests/mxc_vpu_test. out -D "-f 2 -y 1 -I <path to your video>"`
9. Use case is running, measurements can be taken now.

5.2 Deep Sleep Mode (LPM6)

5.2.1 Clocks and PLLs Configuration

In this use case all clocks and PLLs are turned off except 32 KHz clock which is for system wake up.

5.2.2 Steps

1. `echo mem > /sys/power/state`
2. Use case is running, measurements can be taken now.

5.3 User Idle Mode (LPM3)

5.3.1 Clocks Configuration

Clocks configuration in Table 20 is aligned with release ER1209RC1.

Table 20. Clocks Configuration

Clock Name	Frequency (MHz)
AXI	264
AHB	176
CPU	198
GPU2D	off
GPU3D Core	off
GPU3D Shader	off
VPU	off
IPU1	off
IPU2	264
MMDC CH0	528
MMDC CH1	off

5.3.2 PLLs Configuration

PLLs configuration in Table 21 is aligned with release ER1209RC1.

Table 21. PLLs Configuration

PLL Name	Frequency (MHz)
PLL1—System PLL	396
PLL2—System Bus PLL	528
pll2 396m pfd	396
pll2 352m pfd	452
pll2 594m pfd	off
PLL3—OTG USB PLL	480
pll3 508m pfd	off
pll3 454m pfd	off
pll3 720m pfd	off
pll3 540m pfd	off
PLL4—Audio PLL	off
PLL5—Video PLL	off
PLL6—MLB PLL	off
PLL7—Host USB PLL	off
PLL8—ENET PLL	off

5.3.3 System Setup

Disconnect everything except the SD and LVDS.

5.3.4 Steps

1. Power on the board and in the serial console press any key to stop autoboot
2. `setenv bootargs_base 'setenv bootargs ${bootargs} fec_mac=${ethaddr} ${lvds_mode}'`
3. `savenv`
4. Run:

```
// blank display
echo 1 > /sys/class/graphics/fb1/blank
echo 1 > /sys/class/graphics/fb2/blank
echo 1 > /sys/class/graphics/fb3/blank
echo 1 > /sys/class/graphics/fb4/blank
// stop the fec
```

```
ifconfig eth0 down
echo 0 > /sys/class/graphics/fb0/blank
//enable bus freq adjustment
```

5. Run `echo 1 > /sys/devices/platform/imx_busfreq.0/enable`
6. Use case is running, measurements can be taken now.

5.4 System Idle Mode (LPM2)

5.4.1 Clocks Configuration

Clocks configuration in Table 22 is aligned with release GA1209.

Table 22. Clocks Configuration

Clock Name	Frequency (MHz)
AXI	264
AHB	176
CPU	198
GPU2D	off
GPU3D Core	off
GPU3D Shader	off
VPU	off
IPU1	off
IPU2	off
MMDC CH0	off
MMDC CH1	off

5.4.2 PLLs Configuration

PLLs configuration in Table 23 is aligned with release GA1209.

Table 23. PLLs Configuration

PLL Name	Frequency (MHz)
PLL1—System PLL	396
PLL2—System Bus PLL	528
pll2 396m pfd	396
pll2 352m pfd	off
pll2 594m pfd	off
PLL3—OTG USB PLL	480
pll3 508m pfd	off
pll3 454m pfd	off
pll3 720m pfd	off
pll3 540m pfd	540
PLL4—Audio PLL	off
PLL5—Video PLL	off
PLL6—MLB PLL	off
PLL7—Host USB PLL	off
PLL8—ENET PLL	off

5.4.3 System Setup

Disconnect everything except the SD and LVDS.

5.4.4 Steps

1. `//disable dvfs`
   - `echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor`
2. `//set cpu freq at 200M`
   - `echo 198000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed`
3. Run the script `/uart_off.sh` (see below).

```
echo 1 > /sys/class/graphics/fb0/blank
echo 1 > /sys/class/graphics/fb1/blank
echo 1 > /sys/class/graphics/fb2/blank
echo 1 > /sys/class/graphics/fb3/blank
```

```
echo 1 > /sys/class/graphics/fb4/blank
```

4. Run the script `uart_off.sh` (see below)
5. Use case is running, measurements can be taken now.

```
uart_off.sh:
        echo "disabling UART"
        /unit_tests/memtool 0x20c407c=0x1

        sleep 5
        /unit_tests/memtool 0x20c8010=0x80010000
        sleep 300
        /unit_tests/memtool 0x20c8010=0x80003000

        sleep 2
        /unit_tests/memtool 0x20c407c=0xf000001
        echo "uart is back ON"
```

5.5 Audio Playback

5.5.1 Clocks Configuration

Clocks configuration in Table 24 is aligned with release GA1209.

Table 24. Clocks Configuration

Clock Name	Frequency (MHz)
AXI	50
AHB	25
CPU	24 MHz from XTAL
GPU2D	off
GPU3D Core	off
GPU3D Shader	off
VPU	off
IPU1	off
IPU2	off
MMDC CH0	132
MMDC CH1	off

5.5.2 PLLs Configuration

PLLs configuration in Table 25 is aligned with release GA1209.

Table 25. PLLs Configuration

PLL Name	Frequency (MHz)
PLL1—System PLL	off
PLL2—System Bus PLL	528
pll2 396m pfd	off
pll2 352m pfd	off
pll2 594m pfd	off
PLL3—OTG USB PLL	480
pll3 508m pfd	508
pll3 454m pfd	off
pll3 720m pfd	off
pll3 540m pfd	off
PLL4—Audio PLL	176
PLL5—Video PLL	off
PLL6—MLB PLL	off
PLL7—Host USB PLL	off
PLL8—ENET PLL	off

5.5.3 System Setup

- SD boot
- Connect XGA LVDS panel

5.5.4 Steps

1. Add enable_wait_mode=on in kernel command line
2. Boot system to SD rootfs with LVDS, run below to enable busfreq scaling

```
#!/bin/sh
echo 1 > /sys/class/graphics/fb0/blank
echo 1 > /sys/class/graphics/fb1/blank
echo 1 > /sys/class/graphics/fb2/blank
echo 1 > /sys/class/graphics/fb3/blank
ifconfig eth0 down
```

If it doesn't run into low busfreq automatically, run:

```
echo 1 > /sys/devices/platform/imx_busfreq.0/enable
echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

```
echo 996000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed
echo 198000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed
echo interactive > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

3. `gplay 128kbps_44khz_s_mp3.mp3`
4. Measure SoC and Arm data, and record
5. Enable dvfs and remeasure the power
6. `echo 1 > /sys/devices/platform/imx_busfreq.0/enable`
7. Get the dvfs status before and after with:

```
cat /sys/devices/system/cpu/cpu0/cpufreq/stats/time_in_state
#system should in interactive gonvernor
```

5.6 XGA 1080p Playback

5.6.1 Clocks Configuration

Clocks configuration in Table 26 is aligned with release GA1209.

Table 26. Clocks Configuration

Clock Name	Frequency (MHz)
AXI	partial on/off
AHB	132
CPU	396
GPU2D	off
GPU3D Core	off
GPU3D Shader	off
VPU	264
IPU1	off
IPU2	264
MMDC CH0	528
MMDC CH1	off

5.6.2 PLLs Configuration

PLLs configuration in Table 27 is aligned with release GA1209.

Table 27. PLLs Configuration

PLL Name	Frequency (MHz)
PLL1—System PLL	off
PLL2—System Bus PLL	528
pll2 396m pfd	396
pll2 352m pfd	off
pll2 594m pfd	off
PLL3—OTG USB PLL	480
pll3 508m pfd	off
pll3 454m pfd	off
pll3 720m pfd	off
pll3 540m pfd	off
PLL4—Audio PLL	off
PLL5—Video PLL	off
PLL6—MLB PLL	off
PLL7—Host USB PLL	off
PLL8—ENET PLL	off

5.6.3 System Setup

- SD boot
- Connect XGA LVDS panel

5.6.4 Steps

1. Boot board and run below scripts

```
#!/bin/sh
echo 1 > /sys/class/graphics/fb0/blank
echo 1 > /sys/class/graphics/fb1/blank
echo 1 > /sys/class/graphics/fb2/blank
echo 1 > /sys/class/graphics/fb3/blank
echo 1 > /sys/class/graphics/fb4/blank
ifconfig eth0 down
echo 1 > /sys/devices/platform/imx_busfreq.0/enable
echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
echo 198000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed
```

2. Run: `/unit_tests/mxc_vpu_test.out -D "-f 2 -y 1 -i "`
3. Measure the power and record result
4. Dump clock before and after and during playback (only once), and record them `./clocks.sh`
5. Enable dvfs and remeasure the power
6. `echo 1 > /sys/devices/platform/imx_dvfscore.0/enable`
7. Get the dvfs status before and after with

```
cat /sys/devices/system/cpu/cpu0/cpufreq/stats/time_in_state
```

5.7 3D Gaming

5.7.1 Clocks Configuration

Clocks configuration in Table 28 is aligned with release GA1209.

Table 28. Clocks Configuration

Clock Name	Frequency (MHz)
AXI	264
AHB	132
CPU	396
GPU2D	off
GPU3D Core	594
GPU3D Shader	528
VPU	off
IPU1	off
IPU2	264

Table 28. Clocks Configuration (continued)

Clock Name	Frequency (MHz)
MMDC CH0	528
MMDC CH1	off

5.7.2 PLLs Configuration

PLLs configuration in Table 29 is aligned with release GA1209.

Table 29. PLLs Configuration

PLL Name	Frequency (MHz)
PLL1—System PLL	996
PLL2—System Bus PLL	528
pll2 396m pfd	396
pll2 352m pfd	452
pll2 594m pfd	594
PLL3—OTG USB PLL	480
pll3 508m pfd	off
pll3 454m pfd	off
pll3 720m pfd	off
pll3 540m pfd	off
PLL4—Audio PLL	off
PLL5—Video PLL	off
PLL6—MLB PLL	off
PLL7—Host USB PLL	off
PLL8—ENET PLL	off

5.7.3 System Setup

- SD boot
- Connect XGA LVDS panel

5.7.4 Steps

1. Add `enable_wait_mode=on` to kernel command line
2. Boot board to SD rootfs, disable Ethernet (ifconfig eth0 down), connect to XGA LVDS dispaly.
3. Run script, below, to measure at 200M:

```
#!/bin/sh
echo 1 > /sys/class/graphics/fb1/blank
echo 1 > /sys/class/graphics/fb2/blank
echo 1 > /sys/class/graphics/fb3/blank
ifconfig eth0 down
echo 1 > /sys/devices/platform/imx_busfreq.0/enable
echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
echo 198000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed
echo 0 > /sys/class/graphics/fb0/blank
```

4. Run 3Dmark_es11 applicaiton, you can copy it from `10.192.225.222/rootfs/wb/utils/Graphics/imx61_rootfs/test/3DMarkMobile`
5. Test samurai record the FPS and with mmdc program to get bus loading
6. Measure the power and record result
7. Enable dvfs and remeasure the power
8. `echo 1 > /sys/devices/platform/imx_busfreq.0/enable`
9. Get the dvfs status before and after with:

```
cat /sys/devices/system/cpu/cpu0/cpufreq/stats/time_in_state
```

10. Run script, below, to test 1G

```
#!/bin/sh
echo 1 > /sys/class/graphics/fb1/blank
echo 1 > /sys/class/graphics/fb2/blank
echo 1 > /sys/class/graphics/fb3/blank
ifconfig eth0 down
echo 1 > /sys/devices/platform/imx_busfreq.0/enable
echo performance > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
echo 0 > /sys/class/graphics/fb0/blank
```

5.8 Dhrystone

5.8.1 Clocks Configuration

Cocks configuration in Table 30 is aligned with release GA1209.

Table 30. Clocks Configuration

Clock Name	Frequency (MHz)
AXI	off
AHB	132
CPU	996
GPU2D	off
GPU3D Core	off
GPU3D Shader	off
VPU	off
IPU1	off
IPU2	off
MMDC CH0	528
MMDC CH1	off

5.8.2 PLLs Configuration

PLLs configuration in Table 31 is aligned with release GA1209.

Table 31. PLLs Configuration

PLL Name	Frequency (MHz)
PLL1—System PLL	996
PLL2—System Bus PLL	528
pll2 396m pfd	396
pll2 352m pfd	off
pll2 594m pfd	off
PLL3—OTG USB PLL	480
pll3 508m pfd	off
pll3 454m pfd	off
pll3 720m pfd	off
pll3 540m pfd	540
PLL4—Audio PLL	off
PLL5—Video PLL	off

Table 31. PLLs Configuration (continued)

PLL Name	Frequency (MHz)
PLL6—MLB PLL	off
PLL7—Host USB PLL	off
PLL8—ENET PLL	off

5.8.3 System Setup

- SD boot
- Connect XGA LVDS panel

5.8.4 Steps

1. Boot board to SD rootfs
2. Run:

```
dry2

#measure the power for 1 core
```

3. Run below to measure the power for 4 core

```
while true; do dry2 & ; done
```

4. Change the kernel command line to max_cpu=2
5. Run script:

```
while true; do dry2 & ; done
#measure the power for 2 core
#for 1.2G add arm_freq=1200 to kernel command line
```

5.9 Max Power

5.9.1 Clocks Configuration

Clocks configuration in Table 32 is aligned with release ER1205.

Table 32. Clocks Configuration

Clock Name	Frequency (MHz)
AXI	264
AHB	132
CPU	996
GPU2D	off
GPU3D Core	528
GPU3D Shader	594
VPU	264

Table 32. Clocks Configuration (continued)

Clock Name	Frequency (MHz)
IPU1	264
IPU2	264
MMDC CH0	528
MMDC CH1	off

5.9.2 PLLs Configuration

PLLs configuration in Table 33 is aligned with release ER1205.

Table 33. PLLs Configuration

PLL Name	Frequency (MHz)
PLL1—System PLL	996
PLL2—System Bus PLL	528
pll2 396m pfd	396
pll2 352m pfd	452
pll2 594m pfd	594
PLL3—OTG USB PLL	480
pll3 508m pfd	off
pll3 454m pfd	off
pll3 720m pfd	off
pll3 540m pfd	540
PLL4—Audio PLL	off
PLL5—Video PLL	891
PLL6—MLB PLL	off
PLL7—Host USB PLL	off
PLL8—ENET PLL	24

5.9.3 System Setup

- SD boot
- Connect HDMI daughter card, and XGA LVDS panel
- Connect TV to each HDMI interface

5.9.4 Steps

1. Rebuild kernel
   — Enable `sii_902x` with `CONFIG_FB_MXC_SII902X=y`

— For SD board

```
    add HDMI mode to sabre SD board

diff --git a/arch/arm/mach-mx6/board-mx6q_arm2.c
b/arch/arm/mach-mx6/board-mx6q_arm2.c
index deecf7a..149ef9a 100644
--- a/arch/arm/mach-mx6/board-mx6q_arm2.c
+++ b/arch/arm/mach-mx6/board-mx6q_arm2.c
@@ -1443,7 +1443,7 @@ static struct fsl_mxc_hdmi_platform_data hdmi_data = {
 };

 static struct fsl_mxc_hdmi_core_platform_data hdmi_core_data = {
-       .ipu_id         = 0,
+       .ipu_id         = 1,
        .disp_id        = 0,
 };

@@ -1454,11 +1454,11 @@ static struct fsl_mxc_lcd_platform_data lcdif_data = {
 };

 static struct fsl_mxc_ldb_platform_data ldb_data = {
-       .ipu_id         = 1,
-       .disp_id        = 0,
+       .ipu_id         = 0,
+       .disp_id        = 1,
        .ext_ref        = 1,
        .mode           = LDB_SEP0,
-       .sec_ipu_id     = 0,
+       .sec_ipu_id     = 1,
        .sec_disp_id    = 1,
 };

diff --git a/arch/arm/mach-mx6/board-mx6q_sabresd.c
b/arch/arm/mach-mx6/board-mx6q_sabresd.c
index e749bc4..5f720f3 100644
--- a/arch/arm/mach-mx6/board-mx6q_sabresd.c
+++ b/arch/arm/mach-mx6/board-mx6q_sabresd.c
@@ -758,6 +758,42 @@ static struct fsl_mxc_lightsensor_platform_data ls_data = {
        .rext = 499,    /* calibration: 499K->700K */
 };
```

```
+static int sii902x_get_pins(void)
+{
+       /* Sii902x HDMI controller */
+       gpio_request(SABRESD_DISP0_PWR_EN, "disp0-pwr");
+       gpio_direction_output(SABRESD_DISP0_PWR_EN, 1);
+       gpio_request(SABRESD_DISP0_RST_B, "disp0-reset");
+       gpio_direction_output(SABRESD_DISP0_RST_B, 0);
+       gpio_request(SABRESD_DISP0_RD, "disp0-detect");
+       gpio_direction_input(SABRESD_DISP0_RD);
+       return 1;
+}
+
+static void sii902x_put_pins(void)
+{
+       gpio_free(SABRESD_DISP0_RST_B);
+       gpio_free(SABRESD_DISP0_RD);
+       gpio_free(SABRESD_DISP0_PWR_EN);
+}
+
+static void sii902x_hdmi_reset(void)
+{
+       gpio_set_value(SABRESD_DISP0_RST_B, 0);
+       msleep(10);
+       gpio_set_value(SABRESD_DISP0_RST_B, 1);
+       msleep(10);
+}
+
+static struct fsl_mxc_lcd_platform_data sii902x_hdmi_data = {
+       .ipu_id = 0,
+       .disp_id = 0,
+       .reset = sii902x_hdmi_reset,
+       .get_pins = sii902x_get_pins,
+       .put_pins = sii902x_put_pins,
+};
+
+
 static struct i2c_board_info mxc_i2c0_board_info[] __initdata = {
        {
                I2C_BOARD_INFO("wm89**", 0x1a),
```

```
@@ -787,6 +823,10 @@ static struct i2c_board_info mxc_i2c1_board_info[] __initdata =
{
        {
                I2C_BOARD_INFO("max11801", 0x48),
                .platform_data = (void *)&max11801_mode,
+       }, {
+               I2C_BOARD_INFO("sii902x", 0x39),
+               .platform_data = &sii902x_hdmi_data,
+               .irq = gpio_to_irq(SABRESD_DISP0_RD),
        },
 };

@@ -1268,7 +1308,7 @@ static struct fsl_mxc_hdmi_platform_data hdmi_data = {
 };

 static struct fsl_mxc_hdmi_core_platform_data hdmi_core_data = {
-       .ipu_id = 0,
+       .ipu_id = 1,
        .disp_id = 0,
 };

@@ -1279,8 +1319,8 @@ static struct fsl_mxc_lcd_platform_data lcdif_data = {
 };
 static struct fsl_mxc_ldb_platform_data ldb_data = {
-       .ipu_id = 1,
-       .disp_id = 0,
+       .ipu_id = 0,
+       .disp_id = 1,
        .ext_ref = 1,
        .mode = LDB_SEP0,
        .sec_ipu_id = 1,
@@ -1375,6 +1415,7 @@ static struct platform_device sabresd_vmmc_reg_devices = {
        },
 };

+
 static int __init imx6q_init_audio(void)
 {
        if (board_is_mx6_reva()) {
@@ -1745,6 +1786,10 @@ static void __init mx6_sabresd_board_init(void)
```

```
        gpio_request(SABRESD_ALS_INT, "als-int");
        gpio_direction_input(SABRESD_ALS_INT);

+       /* enable DISP0 power */
+       gpio_request(SABRESD_DISP0_PWR_EN, "disp0-pwr");
+       gpio_direction_output(SABRESD_DISP0_PWR_EN, 1);
+
        imx6q_add_hdmi_soc();
        imx6q_add_hdmi_soc_dai();
```

2. Set boot command line as below

```
console=ttymxc3,115200 disable_mipi_dsi
video=mxcfb0:dev=sii902x_hdmi,1920x1080M@60,if=RGB24
video=mxcfb1:dev=hdmi,1920x1080M@24,if=RGB24 video=mxcfb2:dev=ldb,LDB-XGA,if=RGB666
ldb=sep0 root=/dev/mmcblk0p1 rootwait
```

3. Run attached scripts:
   a) Run 2 VPU decoders to 1080P HDMI display
   b) Run 3 instances of dhrystone
   c) Run 1 instance of gpu mm06(samurai) and display to LVDS

```
#power_measure1.sh &
#power_measure2.sh &
#power_measure1.sh
```

4. Measure
   — The cpu usage is 1 core that is for VPU hardware decode and GPUs. Other 3 cores are busy with software program.
   – power_measure1.sh

```
#!/bin/sh -x
while [true ]; do
dry2 &
dry2 &
dry2
done
```

   – #!/bin/sh -x

```
echo 0 > /sys/class/graphics/fb0/blank
echo 0 > /sys/class/graphics/fb4/blank
echo 0 > /sys/class/graphics/fb2/blank
a_stream_path=/mnt/nfs/test_stream/video/H264_ML_1920x1080_10Mbps_15fps_noaudi
o.h264
while true
do
/unit_tests/mxc_vpu_test.out -D "-f 2 -i ${a_stream_path}" &
/unit_tests/mxc_vpu_test.out -D "-f 2 -i ${a_stream_path} -x 19 "
done
```

   – power_measure3.sh

```
#!/bin/sh
modprobe galcore
export FB_FRAMEBUFFER_0=/dev/fb4
cd /mnt/nfs/util/Graphics/imx61_rootfs/test/3DMarkMobile/fsl_imx_linux/
/mnt/nfs/util/Graphics/imx61_rootfs/test/3DMarkMobile/fsl_imx_linux/fm_oes_pla
yer
```

5.10 Important Commands

In Boot Console

- `printenv` – display environment variables.
- `setenv` – update environment variables.
  - `setenv <name> <value> ...`
    - Set environment variable 'name' to 'value ...'
  - `setenv <name>`
    - Delete environment variable 'name'
- `saveenv` – save updates to environment variables.
- `bootargs` – pass to the kernel, which are called kernel command lines.

In Linux Console

- `cat /proc/cmdline` – displays command line
- `cat /sys/devices/virtual/thermal/thermal_zone0/temp` – print temperature to screen (chip should be calibrated)
- In order to print to screen clocks configuration, use the `clocks.sh` script

clocks.sh

```
#!/bin/bash

saved_path=$PWD

if ! mount|grep -sq '/sys/kernel/debug'; then
    mount -t debugfs none /sys/kernel/debug
fi

printf "%-24s %-20s %3s %9s\n" "clock" "parent" "use" "flags" "rate"

for foo in $(find /sys/kernel/debug/clock -type d); do
    if [ "$foo" = '/sys/kernel/debug/clock' ]; then
        continue
    fi

    cd $foo
```

```
    ec="$(cat usecount)"
    rate="$(cat rate)"
    flag="$(cat flags)"

    clk="$(basename $foo)"
    cd ..
    parent="$(basename $PWD)"

    if [ "$parent" = 'clock' ]; then
    parent="   ---"
    fi

    printf "%-24s %-24s %2d %2d %10d\n" "$clk" "$parent" "$ec" "$flag" "$rate"
    cd $saved_path
done
```

6 Revision History

Table 34 provides a revision history for this application note.

Table 34. Document Revision History

Rev. Number	Date	Substantive Change(s)
Rev. 0	10/2012	Initial public release.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number: AN4509
Rev. 0
10/2012