Freescale Semiconductor
Application Note

Document Number: AN5050
Rev. 0, 05/2015

i.MX 6SoloX Power Consumption Measurement

This application note will help the user design power management systems. Through several use cases, this report illustrates the current drain measurements of the i.MX 6SoloX Applications Processors taken on the Freescale SABRE SD Platform. The reader will be enabled to choose the appropriate power supply domains for the i.MX 6SoloX chips and become familiar with the expected chip power in various scenarios.

NOTE

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 Overview of i.MX 6SoloX Voltage Supplies 2

2 Internal Power Measurement of the i.MX 6SoloX Processor 4

3 Use Cases and Measurement Results 9

4 Reducing Power Consumption................................ 18

5 Use Case Configuration and Usage Guidelines .. 25

6 Revision History 41

1 Overview of i.MX 6SoloX Voltage Supplies

The i.MX 6SoloX 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.

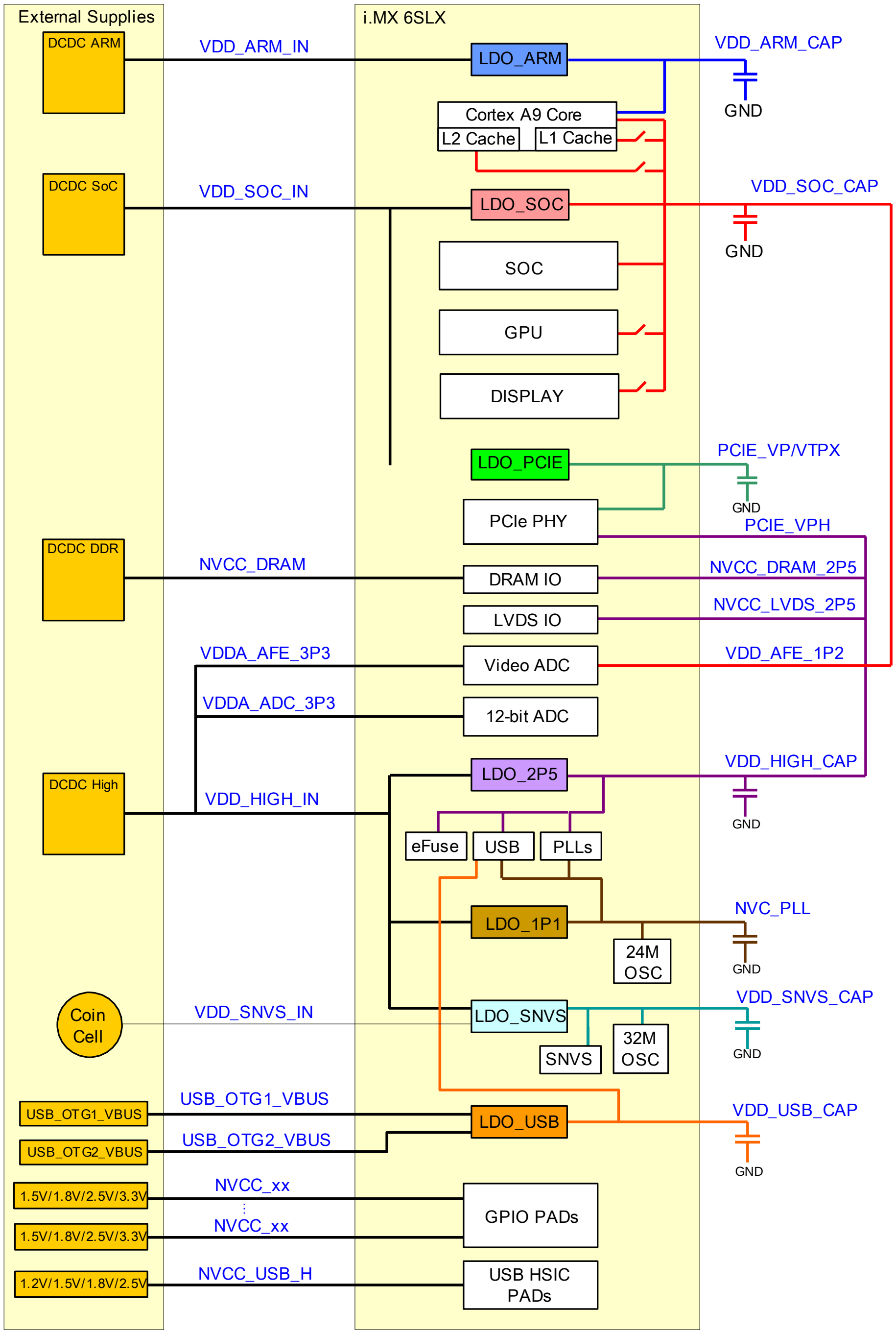

Figure 1 i.MX 6SoloX power rails

NOTE

See the i.MX 6SoloX datasheet for consumer products (IMX6SXCEC) for the recommended operating conditions of each supply rail and for a detailed description of the groups of pins each I/O voltage supply powers.

For more details regarding the i.MX 6SoloX power rails see the Power Management Unit (PMU) chapter in the *i.MX 6SoloX Applications Processor Reference Manual* (IMX6SXRM).

2 Internal Power Measurement of the i.MX 6SoloX Processor

Several use cases (described in Section 3 Use Cases and Measurement Results”) are run on the SABRE SD platform (Rev B). The measurements are taken mainly for the following power supply domains:

- VDDARM_SOC_IN platform’s supply
- VDDHIGH_IN—Source for PLLs, DDR pre-drives, PHYs, and some other circuitry

These supply domains consume the majority of the internal power of the processor. For the 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 whether or not these modules are used. 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.MX6SoloX 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 8.

NOTE

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

2.1 VDDHIGH power

The voltage VDD_HIGH domain is generated from the 2.5V LDO (LDO_2P5).

This domain powers the following circuits:

- On-chip LDOs
- Bandgap
- MLB
- eFUSE
- Analog part of the PLLs
- Pre-drivers of the DDR I/Os (NVCC_LVDS_2P5)

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

- PCIe
- 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
- 1.35 V for DDR3L
- 1.2 V for LPDDR2

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

- 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 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 Platform also includes the current of the onboard DDR3L memory devices. This board (on which the measurements were taken) includes two DDR3L devices, having a total capacity of 1 GB. The SABRE SD Platform 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/DDR3L 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 for 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.

2.3 Voltage levels and DVFS usage in measurement process

The voltage levels of all the supplies, except for VDDARM_SOC_IN, are set to the typical voltage levels as defined in i.MX 6SoloX datasheet for consumer products (IMX6SXCEC).

The VDDARM_SOC_IN supplies require special explanation. To save power, VDDARM_SOC_IN 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 in 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 CPUFREQ) driver. There are several factors that contribute to the setpoint decisions, CPU load being the most important. Other factors are CPU latency requirements, thermal restrictions, and peripheral I/O performance requirements. The voltage and frequency setpoints used for the measurements are given in Table 1.

NOTE

See the operating ranges table in the i.MX 6SoloX datasheet for consumer products (IMX6SXCEC) 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. If the measurement is done at different voltage levels, the power consumption scales with the voltage change. In real applications when DVFS is applied, 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.

2.3.2 VDDSOC voltage levels

The approximate nominal target voltage levels for VDD_SOC_IN is varies according to the VDD_SOC_CAP setpoint when LDO_SOC is bypassed. See Table 1 for the VDD_SOC_CAP settings used in the measurements. See the operating ranges table in the i.MX 6SoloX datasheet for consumer products (IMX6SXCEC) for the official operating points.

Table 1 VDDARM, VDDSOC, and VDDPU voltage levels (for reference only)

ARM Frequency	LDO State	VDDARM_SOC_IN	VDD_ARM_CAP	VDD_SOC_CAP
996 MHz	bypassed	1.250V	1.250V	1.250V
792 MHz	bypassed	1.175V	1.175V	1.175V
396 MHz	bypassed	1.175V	1.175V	1.175V

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, etc. Even when using the same board type, the measured temperature may vary due to factors, such as environment, silicon variations, and measurement error.

2.5 Hardware and software used

The software versions used for the measurements are as follows:

- Yocto rootfs (version: Oct.31 Yocto daily build), Linux Kernel version: L3.10.53-1.1.0_ga+g80bec27.
- The board used for the measurements is the Freescale SABRE SD Platform.
- 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 default voltages of the supplies.

The default input voltages are as follows:

- VDDARM_SOC_IN at 1.375
- VDD_HIGH_IN at 3.0 V

- NVCC_DRAM at 1.35 V

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

2.7 Measuring points on the Freescale SABRE SD platform

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 0.02-Ω resistors on the SD board is 1%. The measuring points for the various supply domains are as follows:

- VDDARM_SOC_IN—The chip domain current is measured on SH10 and the recommended resistance value for this measurement is 0.02 Ω.
- VDDHIGH_IN—The VDDHIGH domain current is measured on R375 and the recommended resistance value for this measurement is 0.02 Ω.
- DDR3L I/O plus Memories—The current in this domain includes the NVCC_DRAM current and the overall current of the onboard DDR3L memory devices. The current in this domain is measured on R373 and the recommended resistance value for this measurement is 0.02 Ω.

3 Use Cases and Measurement Results

The main use cases and subtypes, which form the benchmarks for the i.MX 6SoloX internal power measurements on the SABRE SD Platform, can be found the following sections.

3.1 Low-power mode use cases

3.1.1 Use case 1—Deep-Sleep mode (DSM)

This mode is called either “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.
- 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_SOC_IN is dropped to 0.975 V 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 6SoloX processor.

Table 2 Deep-Sleep mode (DSM) measurement results

Supply Domain	Voltage (V)	L3.10.53-1.1.0_ga+g80bec27	
		P (mW)	I (mA)
VDD_ARM_SOC_IN	0.972	1.021	1.050
VDD_HIGH_IN	2.973	0.743	0.25
Total Power (without DDR3L I/O + Memories)	-	1.764	-
DDR3L I/O + Memories3	1.365	22.318	16.350
Total Power	-	24.082	-

NOTE

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

3.1.2 Use case 2—System Idle mode

The use case is as follows:

- ARM is power gated if kernel is in the lowest level of idle.
- All PLLs are off if ARM is power gated.
- Operating system is on.
- LCD is turned off.
- Screen is not refreshed.

This use cases 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 6SoloX processor.

Table 3 System Idle mode measurement results

Supply Domain	L3.10.53-1.1.0_ga+g80bec27		
	Voltage (V)	P (mW)	I (mA)
VDD_ARM_SOC_IN	1.177	10.684	9.077
VDD_HIGH_IN	2.990	1.653	0.553
Total Power (without DDR3L I/O + Memories)	-	12.337	-
DDR3L I/O + Memories	1.354	22.327	16.490
Total Power	-	34.664	-

NOTE

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

3.2 Audio playback use case—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 SAI (Synchronous Audio Interface).
3. The stream, an mp3 file with bit rate 128 kbps and sampling frequency of 44100 Hz, is taken from the SD (secure digital) card.

The LCD is turned off after the timer expires. The figures are measured when LCD is off.

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

Table 4 MP3 Audio Playback measurement results

Supply Domain	L3.10.53-1.1.0_ga+g80bec27		
	Voltage (V)	P (mW)	I (mA)
VDD_ARM_SOC_IN	1.176	108.899	92.6
VDD_HIGH_IN	2.983	46.156	15.426
Total Power (without DDR3L I/O + Memories)	-	155.055	-
DDR3L I/O + Memories	1.352	48.3	35.752
Total Power	-	203.355	-

NOTE

For additional details on this use case and settings, see Section 5, “Use case configuration and usage guidelines.”

3.3 Dhrystone benchmark

Dhrystone is a synthetic benchmark 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 the L2 cache and DDR.

3.3.1 Use case 1— Dhrystone benchmark on Cortex-A9 (996 MHz), CoreMark benchmark on Cortex-M4 (227 MHz)

In this use case, the Dhrystone test is performed by two cores. The Cortex-A9 runs the test in a loop at a frequency of 996 MHz, the Cortex-M4 runs the test in a loop at a frequency of 227MHz. Table 5 shows the measurement results when this use case is applied on the i.MX 6SoloX processor.

Table 5 Dhrystone benchmark on Cortex-A9 (996 MHz), CoreMark benchmark on Cortex-M4 (227 MHz)

Supply Domain	Voltage (V)	L3.10.53-1.1.0_ga+g80bec27	
		P (mW)	I (mA)
VDD_ARM_SOC_IN	1.262	697.222	552.474
VDD_HIGH_IN	2.985	78.207	26.2
Total Power (without DDR3L I/O + Memories)	-	775.429	-
DDR3L I/O + Memories	1.348	28.575	21.198
Total Power	-	804.004	-

3.3.2 Use case 2— Dhrystone benchmark on Cortex-A9 (996 MHz), Cortex-M4 idle

In this use case, the Dhrystone test is performed by a single core. The ARM processor runs the test in a loop at a frequency of 996 MHz. The M4 core is idle. Run power of ARM is measured.

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

Table 6 Dhrystone benchmark on Cortex-A9 (996MHz), Cortex-M4 idle

Supply Domain	Voltage (V)	L3.10.53-1.1.0_ga+g80bec27	
		P (mW)	I (mA)
VDD_ARM_SOC_IN	1.261	657.990	521.8
VDD_HIGH_IN	2.985	53.775	18.015
Total Power (without DDR3L I/O + Memories)	-	711.765	-
DDR3L I/O + Memories	1.348	28.165	20.894
Total Power	-	739.93	-

3.4 Graphics use cases—3D gaming benchmark, MM07

This use case has the following features:

- XGA resolution, using MM07 (Taiji) benchmark.
- The frame rate is 3.109 fps.
- The display is of XGA resolution using LVDS.

The graphics are 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 PXP and displayed on the LCD display (through LVDS) with a refresh rate of 60 Hz.

In this use case, measurements were taken with DVFS disabled and CPU speed set to 396 MHz.

Table 7 3D gaming MM07 benchmark measurement results—DVFS disabled

Supply Domain	L3.10.53-1.1.0_ga+g80bec27		
	Voltage (V)	P (mW)	I (mA)
VDD_ARM_SOC_IN	1.181	383.824	324.999
VDD_HIGH_IN	2.983	121.286	40.659
Total Power (without DDR3L I/O + Memories)	-	505.110	-
DDR3L I/O +1 Memories	1.350	225.646	167.145
Total Power	-	730.756	-

NOTE

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

3.5 Typical max power— Dhrystone benchmark and 3D gaming benchmark (MM07, OpenGL ES2.0) on Cortex-A9 (996 MHz), CoreMark benchmark on Cortex-M4 (227 MHz)

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 LVDS display:

- 3D graphics through LVDS port with XGA resolution.

This use case has the following features:

- The graphics are 3D gaming benchmark—MM07.
- Both ARM cores are heavily loaded.
- Maximum frequencies are used for ARM, GPUs, and DDR clocks.

Table 8 Typical max power measurement results on SABRE SD platform

Supply Domain	L3.10.53-1.1.0_ga+g80bec27		
	Voltage (V)	P (mW)	I (mA)
VDD_ARM_SOC_IN	1.265	939.389	742.6
VDD_HIGH_IN	2.982	131.769	44.188
Total Power (without DDR3L I/O + Memories)	-	1071.158	-
DDR3L I/O + Memories	1.351	266.371	197.166
Total Power	-	1337.529	-

3.6 Non-multimedia use case

3.6.1 Use case 1— USB-to-eMMC file transfer

In this use case, 1 GB total of data is transferred from a USB device to an eMMC device. A data size of 1 MB is copied each time, repeatedly, 1000 times. The SDMA is used to perform the data transfer to the eMMC host controller. Table 9 shows the transfer measurement results.

Table 9 USB-to-eMMC file transfer measurement results

Supply Domain	Voltage (V)	L3.10.53-1.1.0_ga+g80bec27	
		P (mW)	I (mA)
VDD_ARM_SOC_IN	1.178	215.132	182.625
VDD_HIGH_IN	2.984	67.728	22.697
Total Power (without DDR3L I/O + Memories)	-	282.86	-
DDR3L I/O + Memories	1.350	141.904	105.114
Total Power	-	424.764	-

3.6.2 PCIe data transfer use case

In this use case, transfer data through PCIE 1G Ethernet for 100s. Table 10 shows the transfer measurement results.

Table 10 PCIe data transfer measurement results

Supply Domain	Voltage (V)	L3.10.53-1.1.0_ga+g80bec27	
		P (mW)	I (mA)
VDD_ARM_SOC_IN	1.260	645.498	512.3
VDD_HIGH_IN	2.980	135.369	45.426
Total Power (without DDR3L I/O + Memories)	-	780.867	-
DDR3L I/O + Memories	1.354	223.545	165.1
Total Power	-	1004.412	-

3.7 SNVS

In this use case, most of the power rails on boards are off except VDD_SNVS domain, VDD_SNVS is to keep RTC and other logic in the SNVS domain to remain powered on. Table 11 shows the SNVS measurement results.

Table 11 SNVS measurement results

Supply Domain	Voltage (V)	L3.10.53-1.1.0_ga+g80bec27	
		P (uW)	I (uA)
VDD_ARM_SOC_IN	0	0	0
VDD_HIGH_IN	0	0	0
SNVS_IN	3.020	87.731	29.050
Total Power (without DDR3L I/O + Memories)	-	87.731	-
DDR3L I/O + Memories	0	0	0
Total Power	-	87.731	-

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 that may help reduce system power. Some of these are already implemented in Linux BSP. Further optimizations can be done on the individual customer's system.

NOTE

> Further power optimizations are planned for future BSP releases. See the Freescale website to obtain the latest BSP release.

- Apply clock gating whenever clocks or modules are not used, by configuring CCGR registers in the Clock Controller Module (CCM).
- 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 the VDDARM and VDDSOC domains. 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 effect on system power.
- Put i.MX 6SoloX into low power modes (WAIT, STOP) whenever possible. See Chapter "Clock Controller Module (CCM)"of the i.MX 6SoloX Applications Processor Reference Manual (IMX6SXRM) 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 used greatly influences the power consumption of the DDR interface pins.
  - Use the proper output driver impedance for DDR interface pins that provides good impedance matching. Select the lowest possible drive strength that provides the required performance, in order to save current through DDR I/O pins.
  - Carefully choose onboard resistors so the least amount of current is wasted—for example, when selecting impedance matching resistors between CLK and CLK_B (when using DDR3L memories).
  - When possible, in lower performance use cases, switching to DLL Off mode allows for greatly reducing DDR frequency. This disables or reduces termination, and it reduces the drive strength. Thus, power consumption of the DDR interface pins could be significantly reduced.
  - Float i.MX 6SoloX DDR interface pins (set to high Z) when DDR memory is in Self-Refresh mode, and keep DDR_SDCKE0 and DDR_SDCKE1 at low value. If DDR_SDCKE0 and DDR_SDCKE1 are kept at low value by using external pull-down resistors, make sure there is no onboard 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 the 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 LV 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 the i.MX 6SoloX DDR interface pins are given below.

NOTE

> All the programming steps below 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 the MAPSR register (automatic power saving is enabled) to make sure that DDR is in Self-Refresh.
2. Do the following:
   a) If there is no onboard 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) If the DDR control and address bus have onboard termination resistors connected to VTT, such as in the case where SODIMM is used:
   - Option 1
     As for (2a), keep SDCKE0/1 active, this causes 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 onboard pull down resistors 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. Go into the Suspend mode.

Steps to be performed after exiting Suspend:

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

NOTE

If the system can ensure 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 the CPU is not running, or it is running from the internal RAM.

5 Use Case Configuration and Usage Guidelines

5.1 Deep-Sleep mode

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

1. Load M4 idle image to QSPI then boot up M4.
2. Boot up A9 image.
3. Enter "S" on M4 console to change M4 to low frequency mode (12 MHz).
4. Run below command to let system enter DSM mode:

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

5. Measure the power and record result.

5.2 System Idle mode

5.2.1 System Idle mode—clock configuration

Clock configuration in Table 12 is aligned with release L3.10.53-1.1.0_ga+g80bec27.

Table 12 System Idle mode clock configuration

Clock Name	Frequency (MHz)
OCRAM	24
AHB	3
CPU	0
GPU3D Core	0
GPU3D AXI	0
MMDC CH0	1

5.2.2 System Idle mode—PLL configuration

PLL configuration in Table 13 is aligned with release L3.10.53-1.1.0_ga+g80bec27.

Table 13 System Idle mode PLL configuration

PLL Name	Frequency (MHz)
PLL1—System PLL	0
PLL2—System Bus PLL	0
pll2 396m pfd	0
pll2 352m pfd	0
pll2 594m pfd	0
PLL3—OTG USB PLL	0
pll3 508m pfd	0
pll3 454m pfd	0
pll3 720m pfd	0
pll3 540m pfd	0
PLL4—Audio PLL	0
PLL5—Video PLL	0
PLL6—ENET PLL	0
PLL7—Host USB PLL	0
PLL8—MLB PLL	0

5.2.3 System Idle mode—system setup

Disconnect everything except the SD and LVDS.

1. Load M4 idle image to QSPI then boot up M4
2. Boot up A9 image with "x11=false"
3. Enter "S" on M4 console to change M4 to low frequency mode (12 MHz).
4. Run below script to let the system enter powersave governor:

   ```
   #!/bin/bash

   echo 8 > /proc/sys/kernel/printk

   ifconfig eth0 down

   ifconfig eth1 down

   echo powersave > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
   ```

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

5. Measure the power and record result

5.3 Audio playback

5.3.1 Audio playback—clock configuration

Clock configuration in Table 14 is aligned with release L3.10.53-1.1.0_ga+g80bec27.

Table 14 Audio playback clock configuration

Clock Name	Frequency (MHz)
OCRAM	24
AHB	24
CPU	396
GPU2D	0
GPU3D Core	0
GPU3D AXI	0
MMDC CH0	49.5

5.3.2 Audio playback—PLL configuration

PLL configuration in Table 15 is aligned with release L3.10.53-1.1.0_ga+g80bec27.

Table 15 Audio playback PLL configuration

PLL Name	Frequency (MHz)
PLL1—System PLL	0
PLL2—System Bus PLL	528
pll2 396m pfd	396
pll2 352m pfd	0
pll2 594m pfd	0
PLL3—OTG USB PLL	480
pll3 508m pfd	0
pll3 454m pfd	0
pll3 720m pfd	0
pll3 540m pfd	0
PLL4—Audio PLL	0
PLL5—Video PLL	0
PLL6—ENET PLL	0
PLL7—Host USB PLL	0
PLL8—MLB PLL	0

5.3.3 Audio playback—system setup

- SD boot
- Connect XGA LVDS panel

5.3.4 Audio playback—steps

1. Load M4 idle image to QSPI then boot up M4, and boot up A9 image
2. Boot board with file system located in SD card
3. Enter "S" on M4 console to change M4 to low frequency mode (12 MHz).
4. Run below script to measure at 400M:

```
#!/bin/sh
echo 8 > /proc/sys/kernel/printk
ifconfig eth0 down
ifconfig eth1 down

echo conservative >
/sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

```
echo 0 > /sys/class/graphics/fb0/blank
```

6. Play audio:

```
pacmd set-default-sink 1;while [ 1 ];
do gplay-1.0 ./128kbps_44khz_s_mp3.mp3;done;
```

7. Measure the power and record result

5.4 3D gaming

5.4.1 3D gaming—clock configuration

Clock configuration in Table 16 is aligned with release L3.10.53-1.1.0_ga+g80bec27.

Table 16 3D gaming clock configuration—CPU frequency at 396 MHz

Clock Name	Frequency (MHz)
OCRAM	198
AHB	132
CPU	396
GPU3D Core	720
GPU3D AXI	720
MMDC CH0	396

5.4.2 3D gaming—PLL configuration

PLL configuration in Table 17 is aligned with release L3.10.53-1.1.0_ga+g80bec27.

Table 17 3D gaming PLL Configuration—CPU frequency at 396 MHz

PLL Name	Frequency (MHz)
PLL1—System PLL	0
PLL2—System Bus PLL	528
pll2 396m pfd	396
pll2 352m pfd	0
pll2 594m pfd	0
PLL3—OTG USB PLL	480
pll3 508m pfd	0
pll3 454m pfd	0
pll3 720m pfd	720
pll3 540m pfd	0
PLL4—Audio PLL	0
PLL5—Video PLL	0
PLL6—ENET PLL	0
PLL7—Host USB PLL	0
PLL8—MLB PLL	0

5.4.3 3D gaming—system setup

- SD boot
- Connect XGA LVDS panel

5.4.4 3D gaming—steps

1. Load M4 idle image to QSPI then boot up M4.
2. Boot up A9 image and boot board to SD rootfs.
3. Enter "S" on M4 console to change M4 to low frequency mode (12 MHz).
4. Run below script to measure at 400M:

```
#!/bin/sh
ifconfig eth0 down
ifconfig eth1 down

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
```

5. Copy 3Dmark_es20 application MM07 folder: basemark_es2.0 to local SD.
6. Run MM07 (Taiji) app and record the fps.
7. Measure the power and record result.

5.5 Dhrystone on Cortex-A9 (996 MHz), CoreMark on Cortex-M4 (227 MHz)

5.5.1 Dhrystone on Cortex-A9 (996 MHz), CoreMark on Cortex-M4 (227 MHz)—clock configuration

Clocks configuration in Table 18 is aligned with release L3.10.53-1.1.0_ga+g80bec27.

Table 18 Dhrystone clock configuration

Clock Name	Frequency (MHz)
OCRAM	198
AHB	132
CPU	996
GPU3D Core	0
GPU3D AXI	0
MMDC CH0	396

5.5.2 Dhrystone on Cortex-A9 (996 MHz), CoreMark on Cortex-M4 (227 MHz)—PLL configuration

PLL configuration in Table 19 is aligned with release L3.10.53-1.1.0_ga+g80bec27.

Table 19 Dhrystone PLL configuration

PLL Name	Frequency (MHz)
PLL1—System PLL	996
PLL2—System Bus PLL	528
pll2 396m pfd	396
pll2 352m pfd	0
pll2 594m pfd	0
PLL3—OTG USB PLL	480
pll3 508m pfd	0
pll3 454m pfd	454
pll3 720m pfd	0
pll3 540m pfd	0
PLL4—Audio PLL	0
PLL5—Video PLL	0
PLL6—ENET PLL	0
PLL7—Host USB PLL	0
PLL8—MLB PLL	0

5.5.3 Dhrystone on Cortex-A9 (996MHz), CoreMark on Cortex-M4 (227 MHz)—system setup

- SD boot
- Connect XGA LVDS panel

5.5.4 Dhrystone on Cortex-A9 (996MHz), CoreMark on Cortex-M4 (227 MHz)—steps

1. Load M4 CoreMark image to TCM and then boot up M4.
2. Boot up A9 image and boot board to SD rootfs.
3. Run below script to measure at 1G:

```
#!/bin/sh
ifconfig eth0 down
ifconfig eth1 down

echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
echo 996000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed
echo 0 > /sys/class/graphics/fb0/blank
```

4. Run dry2 and measure

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

5. Measure the power and record result.

5.6 Dhrystone on Cortex-A9 (996 MHz), Cortex-M4 idle

5.6.1 Dhrystone on Cortex-A9 (996 MHz), Cortex-M4 idle — clock configuration

Clocks configuration in Table 20 is aligned with release L3.10.53-1.1.0_ga+g80bec27.

Table 20 Dhrystone clock configuration

Clock Name	Frequency (MHz)
OCRAM	198
AHB	132
CPU	996
GPU3D Core	0
GPU3D AXI	0
MMDC CH0	396

5.6.2 Dhrystone on Cortex-A9 (996 MHz), Cortex-M4 idle —PLL configuration

PLL configuration in Table 21 is aligned with release L3.10.53-1.1.0_ga+g80bec27.

Table 21 Dhrystone PLL configuration

PLL Name	Frequency (MHz)
PLL1—System PLL	996
PLL2—System Bus PLL	528
pll2 396m pfd	396
pll2 352m pfd	0
pll2 594m pfd	0
PLL3—OTG USB PLL	0
pll3 508m pfd	0
pll3 454m pfd	0
pll3 720m pfd	0
pll3 540m pfd	0
PLL4—Audio PLL	0
PLL5—Video PLL	0
PLL6—ENET PLL	0
PLL7—Host USB PLL	0
PLL8—MLB PLL	0

5.6.3 Dhrystone on Cortex-A9 (996 MHz), Cortex-M4 idle — system setup

- SD boot
- Connect XGA LVDS panel

5.6.4 Dhrystone on Cortex-A9 (996 MHz), Cortex-M4 idle — steps

1. Load M4 idle image to QSPI then boot up M4.
2. Boot up A9 image and boot board to SD rootfs.
3. Enter "S" on M4 console to change M4 to low frequency mode (12 MHz).
4. Run below script to measure at 1G:

```
#!/bin/sh
ifconfig eth0 down
ifconfig eth1 down

echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
echo 996000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed
echo 0 > /sys/class/graphics/fb0/blank
```

5. Run dry2 and measure

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

6. Measure the power and record result.

5.7 Max power

Dhrystone benchmark and 3D gaming benchmark (MM07, OpenGL ES2.0) on Cortex-A9 (996 MHz), CoreMark benchmark on Cortex-M4 (200 MHz).

5.7.1 Max power: Clock configuration

Clock configuration in Table 22 is aligned with release L3.10.53-1.1.0_ga+g80bec27.

Table 22 Max power clock configuration

Clock Name	Frequency (MHz)
OCRAM	198
AHB	132
CPU	996
GPU3D Core	720
GPU3D AXI	720
MMDC CH0	396

5.7.2 Max power—PLL configuration

PLL configuration in Table 23 is aligned with release L3.10.53-1.1.0_ga+g80bec27.

Table 23 Max power PLL configuration

PLL Name	Frequency (MHz)
PLL1—System PLL	996
PLL2—System Bus PLL	528
pll2 396m pfd	396
pll2 352m pfd	0
pll2 594m pfd	0
PLL3—OTG USB PLL	480
pll3 508m pfd	0
pll3 454m pfd	454
pll3 720m pfd	720
pll3 540m pfd	0
PLL4—Audio PLL	0
PLL5—Video PLL	0
PLL6—ENET PLL	0
PLL7—Host USB PLL	0
PLL8—MLB PLL	0

5.7.3 Max power—system setup

- SD boot
- Connect XGA LVDS panel

5.7.4 Max power—steps

1. Load M4 CoreMark image to TCM and then boot up M4
2. Boot up A9 image and boot borad to SD rootfs, connect to XGA LVDS display
3. Run below script to measure at 1G:

```
#!/bin/sh
ifconfig eth0 down
ifconfig eth1 down

echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
echo 996000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed
echo 0 > /sys/class/graphics/fb0/blank
```

4. Copy 3Dmark_es20 application MM07 folder: basemark_es2.0 to local SD
5. Run dry2 and measure

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

6. Run MM07(Taiji) app and record the fps
7. Measure the power and record result

5.8 USB-to-eMMC File Transfer

5.8.1 USB-to-eMMC File Transfer—clock configuration

Clock configuration in Table 24 is aligned with release L3.10.53-1.1.0_ga+g80bec27.

Table 24 USB-to-eMMC file transfer clock configuration

Clock Name	Frequency (MHz)
OCRAM	198
AHB	132
CPU	396
GPU3D Core	0
GPU3D AXI	0
MMDC CH0	396

5.8.2 USB-to-eMMC File Transfer —PLL configuration

PLL configuration in Table 25 is aligned with release L3.10.53-1.1.0_ga+g80bec27.

Table 25 USB-to-eMMC file transfer PLL configuration

PLL Name	Frequency (MHz)
PLL1—System PLL	0
PLL2—System Bus PLL	528
pll2 396m pfd	396
pll2 352m pfd	0
pll2 594m pfd	0
PLL3—OTG USB PLL	480
pll3 508m pfd	0
pll3 454m pfd	0
pll3 720m pfd	0
pll3 540m pfd	0
PLL4—Audio PLL	0
PLL5—Video PLL	0
PLL6—ENET PLL	0
PLL7—Host USB PLL	0
PLL8—MLB PLL	0

5.8.3 USB-to-eMMC File Transfer — system setup

- SD boot inserted to slot3
- Insert eMMC card to slot4 or board reworked with solder eMMC chip

5.8.4 USB-to-eMMC File Transfer — steps

1. Load M4 idle image to QSPI and then boot up M4.
2. Boot up A9 image and boot board to SD rootfs.
3. Enter "S" on M4 console to change M4 to low frequency mode (12 MHz).
4. Run below script:

```
#!/bin/sh
ifconfig eth0 down
ifconfig eth1 down

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

5. Run below command to transfer 1G data from USB to eMMC:

```
dd if=/dev/sda1 of=/dev/mmcblk0p1 bs=1M count=1000
```

6. Measure the power and record result.

5.9 PCIE data transfer

5.9.1 PCIE data transfer—clock configuration

Clock configuration in Table 26 is aligned with release L3.10.53-1.1.0_ga+g80bec27.

Table 26 PCIE data transfer clock configuration

Clock Name	Frequency (MHz)
OCRAM	198
AHB	132
CPU	396
GPU3D Core	0
GPU3D AXI	0
MMDC CH0	396

5.9.2 PCIE data transfer — PLL configuration

PLL configuration in Table 27 is aligned with release L3.10.53-1.1.0_ga+g80bec27.

Table 27 PCIE data transfer PLL configuration

PLL Name	Frequency (MHz)
PLL1—System PLL	0
PLL2—System Bus PLL	528
pll2 396m pfd	396
pll2 352m pfd	0
pll2 594m pfd	0
PLL3—OTG USB PLL	0
pll3 508m pfd	0
pll3 454m pfd	0
pll3 720m pfd	0
pll3 540m pfd	0
PLL4—Audio PLL	0
PLL5—Video PLL	0
PLL6—ENET PLL	100
PLL7—Host USB PLL	0
PLL8—MLB PLL	0

5.9.3 PCIE data transfer — system setup

- SD boot inserted to slot3
- Connect PCIE Ethernet device to board, and then connect PCIE ethernet port to PC directly (it can support 1 Gb/s as server)

5.9.4 PCIE data transfer —steps

1. Load M4 idle image to QSPI and then boot up M4
2. Boot up A9 image and boot borad to SD rootfs
3. Enter "S" on M4 console to change M4 to low frequency mode (12 MHz)
4. Run below script:

```
#!/bin/sh
ifconfig eth0 down
ifconfig eth1 down

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

5. Manually set eth2 ip be $LOCALIP, and set server ip be $SERVERIP
6. Run "`iperf -s`" on server;
7. Run "`iperf -d -c $SERVERIP -n 500M -B $LOCALIP D3-t 100`"
8. Measure the power and record result

5.10 SNVS

5.10.1 SNVS — steps

1. Load M4 idle image to QSPI and then boot up M4.
2. Boot up A9 image and boot board to SD rootfs.
3. Press SW2 (On/Off) key for about 5 seconds to let board power down.
4. Measure the power and record result.

5.11 Important commands

In Uboot 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)
- `cat /sys/kernel/debug/clk/clk_summary` — print all clks to screen.

6 Revision History

Table 28 provides a revision history for this application note.

Table 28 Revision history

Rev. Number	Date	Substantive Change(s)
Rev. 0	05/2015	Initial public release.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Document Number: AN5050
Rev. 0
5/2015