

# 6ULL Power Consumption Application Note

## 1. Introduction

This application note helps to design power management systems. Through several use cases this application note illustrates the current drain measurements of the i.MX 6ULL Applications Processors taken on the MX6ULL EVK Evaluate Kit Platform. You will be enabled to choose the appropriate power supply domains for the i.MX 6ULL chips and become familiar with the expected chip power in various scenarios.

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

## Contents

1.	Introduction .....	1
2.	Overview of i.MX 6ULL Voltage Supplies .....	2
3.	Internal Power Measurement of the i.MX 6ULL Processor	3
3.1.	VDD_HIGH_CAP power.....	3
3.2.	DDR I/O power .....	4
3.3.	Voltage levels and DVFS usage in measurement process.....	5
3.4.	Temperature measurements .....	6
3.5.	Hardware and tool used.....	6
3.6.	Board setup used for power measurements.....	6
3.7.	Measuring points on the MX6ULL EVK platform .	6
4.	Use Cases and Measurement Results.....	7
4.1.	Low-power mode use cases .....	7
4.2.	Dhrystone benchmark .....	8
4.3.	Heavy loading use case: 480P video playback.....	9
4.4.	SNVS.....	10
5.	Reducing Power Consumption .....	11
5.1.	Steps to be performed before entering Suspend (Deep-Sleep mode): .....	12
5.2.	Steps to be performed after exiting Suspend:.....	13
6.	Use Case Configuration and Usage Guidelines .....	13
6.1.	Deep-Sleep mode.....	13
6.2.	System Idle mode .....	13
6.3.	Dhrystone on Cortex-A7 .....	15
6.4.	Heavy Loading Use Case: video playback .....	16
6.5.	SNVS.....	17
6.6.	Important commands .....	17
7.	Revision History .....	18

## 2. Overview of i.MX 6ULL Voltage Supplies

The i.MX 6ULL processors have several power supply domains (voltage supply rails) and several internal power domains. The following figure shows the connectivity of these supply rails and the distribution of the internal power domains.

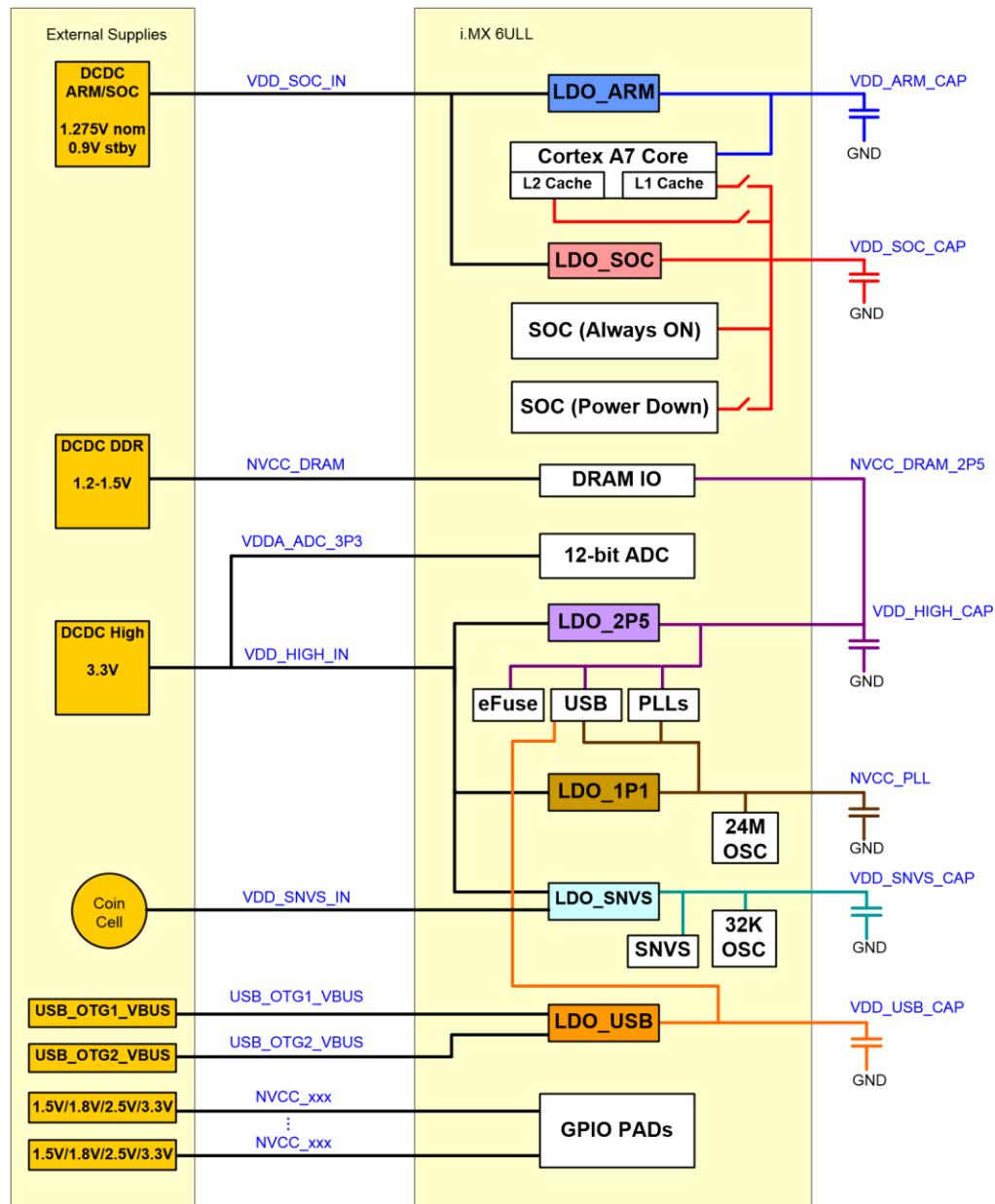


Figure 1. i.MX 6ULL power rails

**NOTE**

See the i.MX 6ULL datasheet for the recommended operating conditions of each supply rail and for a detailed description of the groups of pins that are powered by each I/O voltage supply.

For more details regarding the i.MX 6ULL power rails see the Power Management Unit (PMU) chapter in the *i.MX 6ULL Applications Processor Reference Manual* (document [IMX6ULLRM](#)).

## 3. Internal Power Measurement of the i.MX 6ULL Processor

Several use cases (described in [Section 6, Use Case Configuration and Usage Guidelines](#)) are run on the Evaluate Kit platform (Revision C2). The measurements are taken mainly for the following power supply domains:

- VDD\_SOC\_IN platform's supply
- VDD\_HIGH\_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.MX6ULL internal power analysis.

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).

### 3.1. VDD\_HIGH\_CAP power

The voltage VDD\_HIGH\_CAP domain is generated from the 2.5 V LDO (LDO\_2P5). This domain powers the following circuits:

- Bandgap
- eFUSE
- Analog part of the PLLs
- Pre-drivers of the DDR I/Os (NVCC\_DRAM\_2P5)

## 3.2. DDR I/O power

The DDR I/O is supplied from the NVCC\_DRAM which provides the power for the DDR I/O pads. The target voltage for this supply depends on the DDR interface 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)
- Load of the DDR memory devices

### NOTE

Due to the factors outlined in the previous paragraph the measurements provided in the following tables will 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 MX6ULL EVK Platform also includes the current of the onboard DDR3L memory devices. This board (on which the measurements were taken) includes one DDR3L device, with a total capacity of 512 MB. The EVK Platform does not require board-level resistor terminations. This further reduces the DDR I/O power usage.

### 3.2.1. 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 consumption 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. The default settings that are used in the Linux OS BSP release may need to be modified by the system designer to fit different systems.

### 3.3. Voltage levels and DVFS usage in measurement process

The voltage levels of all the supplies, except for VDD\_SOC\_IN, are set to the typical voltage levels as defined in i.MX 6ULL datasheet.

The VDD\_SOC\_IN supplies require special explanation. To save power, VDD\_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.

#### 3.3.1. 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 6ULL 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. 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.

#### 3.3.2. VDDSOC voltage levels

The approximate nominal target voltage levels for VDD\_SOC\_IN vary according to the VDD\_SOC\_CAP setpoint when LDO\_SOC is enabled. See the following table for the VDD\_SOC\_CAP settings used in the measurements. See the operating ranges table in the i.MX 6ULL datasheet for the official operating points.

**Table 1. VDDARM, VDDSOC voltage levels (for reference only)**

ARM Frequency	LDO State	VDD_ARM_CAP	VDD_SOC_CAP
528 MHz	Enable	1.150 V	1.150 V
396 MHz	Enable	1.000 V	1.150 V
198 MHz	Enable	0.925 V	1.150 V

### 3.4. 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.

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

### 3.5. Hardware and tool used

The hardware and tool used for the measurements are as follows:

- The board used for the measurements is the MX6ULL EVK Platform
- The measurements were performed using an Agilent 34460A 6½ Digit Multimeter

### 3.6. 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:

- VDD\_SOC\_IN at 1.30 V for a maximum of 528 MHz ARM® frequency
- VDD\_HIGH\_IN at 3.3 V
- NVCC\_DRAM at 1.35 V

By using a different setup such as a configurable and separated DC switcher for ARM, the system power can be further optimized by reducing the VDD\_SOC\_IN input voltage level to 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.

### 3.7. Measuring points on the MX6ULL EVK 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  $0.02\ \Omega$  resistors on the EVK board is 1%. The measuring points for the various supply domains are as follows:

- VDD\_SOC\_IN: The chip domain current is measured on R729 and the recommended resistance value for this measurement is  $0.02\ \Omega$ .
- VDD\_HIGH\_IN: The VDDHIGH domain current is measured on R732 and the recommended resistance value for this measurement is  $0.02\ \Omega$ .

## 4. Use Cases and Measurement Results

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

### 4.1. Low-power mode use cases

#### 4.1.1. 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 is in LDO bypass mode and ARM LDO is power gate automatically
- All PLLs (phase locked loop) and CCM (clock controller module) generated clocks are off
- CKIL (32 kHz) input is on
- All of the modules are disabled
- All analog PHYs are powered down
- External high frequency crystal and on chip oscillator are powered down (by asserting SBYOS bit in CCM)
- VDD\_SOC\_IN is reduced to 0.925 V. In this mode, no current flow is caused by external resistive loads.

The following table shows the measurement results when this use case is applied on the i.MX 6ULL processor.

**Table 2. Deep-Sleep Mode (DSM) measurement results (ARM @ 528 MHz)**

Supply Domain	Voltage (V)	Yocto L4.1.15_2.0.0_6ULL_GA_RC4-RC4; Linux Kernel Version: 4.1.15-2.0.0+gb63f3f5	
		P (mW)	I (mA)
VDD_SOC_IN	0.923	0.092	0.25
VDD_HIGH_IN	3.290	0.165	0.05
Total Power (without DDR3L I/O + Memories)	—	0.257	—
DDR3L I/O + Memories	1.340	9.648	7.2
Total Power	—	9.905	—

For additional details on this use case and settings, see [Section 6, Use Case Configuration and Usage Guidelines](#).

### 4.1.2. Use case two: system idle mode

The use case is as follows:

- ARM is power gated if kernel is in the lowest level of idle
- ANATOP will go into low power mode if lowest level of idle is entered and all PLLs are off
- 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.

The following table shows the measurement results when this use case is applied on the i.MX 6ULL processor.

**Table 3. System idle mode measurement results (ARM @ 198 MHz)**

Supply Domain	Yocto L4.1.15_2.0.0_6ULL_GA_RC4-RC4; Linux Kernel Version: 4.1.15-2.0.0+gb63f3f5		
	Voltage (V)	P (mW)	I (mA)
VDD_SOC_IN	1.299	8.314	6.4
VDD_HIGH_IN	3.292	4.938	1.5
Total Power (without DDR3L I/O + Memories)	—	13.252	—
DDR3L I/O + Memories	1.341	9.186	6.85
Total Power	—	22.438	—

For additional details on this use case and settings, see [Section 6, Use Case Configuration and Usage Guidelines](#).

### 4.2. Dhystone benchmark

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

#### 4.2.1. Use case: Dhystone benchmark on ARM® Cortex®-A7

In this use case, the Dhystone test is performed by the ARM Cortex-A7 core.

The following table shows the measurement results when this use case is applied on the i.MX 6ULL processor.

**Table 4. Dhystone benchmark measurement results on Cortex-A7 (ARM @ 528 MHz)**

Supply Domain	Yocto L4.1.15_2.0.0_6ULL_GA_RC4-RC4; Linux Kernel Version: 4.1.15-2.0.0+gb63f3f5		
	Voltage (V)	P (mW)	I (mA)
VDD_SOC_IN	1.287	202.059	157
VDD_HIGH_IN	3.292	36.212	11
Total Power (without DDR3L I/O + Memories)	—	238.271	—
DDR3L I/O + Memories	1.34	12.73	9.5
Total Power	—	251.001	—

### 4.3. Heavy loading use case: 480P video playback

The purpose of this use case is to provide the power consumption of a video playback use case to show the power consumption in high-loading conditions.

This use case has the following features:

- LCD display ON, 480x272, 60 Hz
- 480p video playback through ethernet
- ARM CPU is heavily loaded
- Maximum frequencies are used for ARM, BUS, and DDR clock.

The following table shows the measurement results when this use case is applied on the i.MX 6ULL processor.

**Table 5. Video Playback power measurement results (ARM @ 528 MHz)**

Supply Domain	Yocto L4.1.15_2.0.0_6ULL_GA_RC4-RC4; Linux Kernel Version: 4.1.15-2.0.0+gb63f3f5		
	Voltage (V)	P (mW)	I (mA)
VDD_SOC_IN	1.289	234.598	182
VDD_HIGH_IN	3.288	60.499	18.4
Total Power (without DDR3L I/O + Memories)	—	295.097	—
DDR3L I/O + Memories	1.334	107.654	80.7
Total Power	—	402.751	—

## 4.4. SNVS

In this use case, most of the power rails on the boards are off except from the VDD\_SNVS domain. VDD\_SNVS is kept on to ensure that RTC and other logic in the SNVS domain remain powered on. The following table shows the SNVS measurement results.

**Table 6. SNVS measurement results**

Supply Domain	Voltage (V)	Under SRTC mode, power consumption does not depend on software	
		P (uW)	I (uA)
VDD_SOC_IN	0	0	0
VDD_HIGH_IN	0	0	0
SNVS_IN	3.3	98.01	29.7
Total Power (without DDR3L I/O + Memories)	—	98.01	—
DDR3L I/O + Memories	0	0	0
Total Power	—	98.01	—

**The SNVS\_IN power consumption in table 6 is measured according to the following conditions:**

1. POR\_B pin setting: GPBIT[1:0] set to 2b'01;
2. BOOT\_MODE[1:0] switch to 2b'00 after system boot up;
3. Remove U2101.

**Some recommendations can be considered for SNVS\_IN power reduction:**

1. "Boot From Fuses" as system BOOT TYPE which can remove the external resistors on BOOT\_MODE0/1 PINs;
2. TAMPER0-9 PINs leave floating or only used as TAMPER function (only for MCIMX6G3Cxxxxxx);
3. Add isolation for TAMPER0-9 PINs if those are used as GPIO to prevent the SNVS\_IN leakage.

**SNVS\_IN power design need to consider:**

BOOT\_MODE0/1, TAMPER0-9, POR\_B those PINs connections.

## 5. 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 NXP 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 the i.MX 6ULL into low-power modes (WAIT, STOP) whenever possible. See Chapter "Clock Controller Module (CCM)" of the i.MX 6ULL Applications Processor Reference Manual (document [IMX6ULLRM](#)) for details.
- DDR interface optimization:
  - Use careful board routing of the DDR memories, maintaining PCB trace lengths as short as possible.
  - Use as reduced an ODT (On-Die Termination) setting as possible. The termination used greatly influences the power consumption of the DDR interface pins.
  - Use the proper output driver impedance for DDR interface pins that provide good impedance matching. Select the lowest possible drive strength that provides the required performance, in order to reduce current through DDR I/O pins.
  - 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 the i.MX 6ULL 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 6ULL DDR interface pins are shown 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.

## 5.1. 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 the 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 per 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 (this can be done through pins with GPIO capability).
      - Set the DSE (drive strength selection, in IOMUXC) for all DDR IF I/O to 0 (High Z).

3. Go into Suspend mode.

## 5.2. Steps to be performed after exiting Suspend:

1. Restore all the settings for the DDR I/O to the required values.
2. The 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.

# 6. Use Case Configuration and Usage Guidelines

## 6.1. 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. Boot up Linux OS image.
2. Run below command to let system enter DSM mode:  
`echo mem > /sys/power/state`
3. Measure the power and record result.

## 6.2. System Idle mode

### 6.2.1. System Idle mode: clock configuration

Table 7. System Idle mode clock configuration

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

## 6.2.2. System Idle mode: PLL configuration

Table 8. System Idle mode PLL configuration

PLL Name	Frequency (MHz)
PLL1—System PLL	0
PLL2—System Bus PLL	0
pI2 396m pfd	0
pI2 352m pfd	0
pI2 594m pfd	0
pI2 200m pfd	0
PLL3—OTG USB PLL	0
pI3 508m pfd	0
pI3 454m pfd	0
pI3 720m pfd	0
pI3 540m pfd	0
PLL4—Audio PLL	0
PLL5—Video PLL	0
PLL6—ENET PLL	0
PLL7—Host USB PLL	0

## 6.2.3. System Idle mode: system setup

Disconnect everything except the SD.

1. Boot up Linux image with "x11=false uart\_from\_osc " in cmdline.
2. Run the following script to let the system enter powersave governor:
3. `#!/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
```

4. Measure the power and record the result.

## 6.3. Dhystone on Cortex-A7

### 6.3.1. Dhystone on Cortex-A7: clock configuration

Table 9. Dhystone clock configuration

Clock Name	ARM Frequency @ 528 MHz
AXI	264 MHz
AHB	132 MHz
CPU	528 MHz
MMDC CH0	396 MHz

### 6.3.2. Dhystone on Cortex-A7: PLL configuration

Table 10. Dhystone PLL configuration

PLL Name	ARM Frequency @ 528 MHz
PLL1—System PLL	0
PLL2—System Bus PLL	528 MHz
pll2 396m pfd	396 MHz
pll2 352m pfd	0
pll2 594m pfd	0
pll2 200m pfd	0
PLL3—OTG USB PLL	480 MHz
pll3 508m pfd	0
pll3 454m pfd	454 MHz
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

### 6.3.3. Dhystone on Cortex-A7: system setup

- SD boot
- Connect LCD panel

### 6.3.4. Dhystone on Cortex-A7: steps

1. Boot up the Linux image and boot the board to the SD roots
2. Run the following script to measure at 528 MHz:

```
#!/bin/sh
```

```

ifconfig eth0 down
ifconfig eth1 down
echo 1 > /sys/class/graphics/fb0/blank;
echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor;
echo 528000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed;

```

3. Run dry2 and measure:
 

```
while true; do dry2; done
```
4. Measure the power and record the result.

## 6.4. Heavy Loading Use Case: video playback

480p video playback on Cortex-A7.

### 6.4.1. Video playback: clock configuration

Table 11. Maximum power clock configuration

Clock Name	ARM Frequency @ 528 MHz
AXI	264 MHz
AHB	132 MHz
CPU	528 MHz
MMDC CH0	396 MHz

### 6.4.2. Video Playback: PLL configuration

Table 12. Maximum power PLL configuration

PLL Name	ARM Frequency @ 528 MHz
PLL1—System PLL	0
PLL2—System Bus PLL	528 MHz
pll2 396m pfd	396 MHz
pll2 352m pfd	0
pll2 594m pfd	0
pll2 200m pfd	0
PLL3—OTG USB PLL	480 MHz
pll3 508m pfd	0
pll3 454m pfd	454 MHz
pll3 720m pfd	720 MHz
pll3 540m pfd	0
PLL4—Audio PLL	0
PLL5—Video PLL	649 Hz
PLL6—ENET PLL	500 MHz
PLL7—Host USB PLL	0

### 6.4.3. Video playback: system setup

- SD boot
- Connect LCD panel

### 6.4.4. Video Playback: steps

1. Boot up the Linux image and boot the board to the SD rootfs, connect to LCD display
2. Run the following script to measure at 528 MHz:

```
#!/bin/sh

echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor;
echo 528000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed
echo 0 > /sys/class/graphics/fb0/blank
```

3. Mount server where video stream locates to board:

```
mkdir /mnt/nfsmount -t nfs -o noblock <Server IP>:
<Server nfs service path> /mnt/nfs
```

4. Run video playback through the ethernet:

```
cd <video stream path>;
gplay-1.0./H264_BP13_640x480_15_1507_AAC_44_128_2_vbr.mp4 -video
-sink="imxv4l2sink sync=false" --audio-sink=fakesink
```

5. Measure the power and record the result.

## 6.5. SNVS

### 6.5.1. SNVS: steps

1. Boot up the Linux image and boot the board to the SD rootfs
2. Press the power (on/off) key for about 5 seconds so that the board powers down
3. Measure the power and record the result.

## 6.6. Important commands

In U-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)
- `cat /sys/kernel/debug/clk/clk_summary`: print all clks to screen

## 7. Revision History

**Table 13. Revision history**

Revision number	Date	Substantive changes
0	10/2016	Initial release

**How to Reach Us:**

**Home Page:**

[nxp.com](http://nxp.com)

**Web Support:**

[nxp.com/support](http://nxp.com/support)

Information in this document is provided solely to enable system and software implementers to use NXP 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. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP 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 NXP 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. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address:

[nxp.com/SalesTermsandConditions](http://nxp.com/SalesTermsandConditions).

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, and Tower, are trademarks of NXP B.V. All other product or service names are the property of their respective owners.

ARM, the ARM Powered logo, and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

© 2016 NXP B.V.

Document Number: AN5345  
Rev. 0  
10/2016

