

AN13951

Optimizing Power Consumption for i.MX 8ULP

Rev. 0 — 30 May 2023

Application note

Document Information

| Information | Content |
|-------------|---|
| Keywords | AN13951, i.MX 8ULP, Power architecture, Power consumption, Software optimization |
| Abstract | This application note describes how to optimize the system level power consumption in several typical scenarios with different domain combinations. |



1 Introduction

The i.MX 8ULP family of processors features NXP advanced implementation of the dual Arm Cortex-A35 cores alongside an Arm Cortex-M33. This combined architecture enables the device to run rich operating systems, such as Linux, on the Cortex-A35 core and an RTOS, such as FreeRTOS, on the Cortex-M33 core. It also includes a Fusion DSP for low-power audio and a HiFi4 DSP for advanced audio and machine-learning applications. It targets low-power and ultra-low-power use cases and products.

The i.MX 8ULP has a complex and advanced design to cover various use cases, which divide SoC into three domains with independent and dedicated power and clock controls. This provides flexibility for users to implement different use cases by combining different domains. This application note intends to describe how to optimize the system-level power consumption in several typical scenarios with different domain combinations.

Note: This application note uses Linux and SDK code of BSP as references and examples.

2 Overview

The i.MX 8ULP SoC has three separate domains: application processor (AP), low-power audio video (LPAV), and real-time (RT) domains. The power and clock controls of these domains are separated, and the bus fabric of each domain is tightly integrated for efficient communication.

The application domain (APD) is used for high-performance computing using dual A35 cores and high-speed I/O such as USB/Ethernet/eMMC. The LPAV domain (LPAVD) is for multimedia applications including audio, video, graphics, and displays that require high-performance and large DDR memory. The real-time domain (RTD) includes low-latency M33 core, small Fusion DSP for audio/voice processing, uPower for total SoC power status control, and Sentinel for security control.

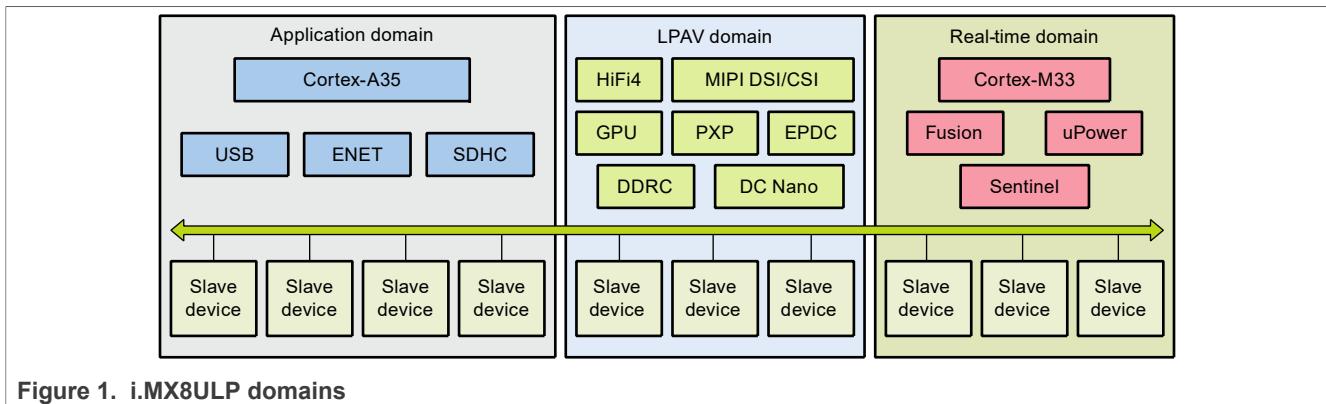


Figure 1. i.MX8ULP domains

2.1 Power architecture

Different domains have separate power supplies (power rail). [Figure 2](#) shows i.MX 8ULP power scheme. There are 18 x power switches (PS) for SoC internal IP modules. These modules can be turned on/off by software, through uPower FW API, for precise power control.

uPower is a central power controller in i.MX 8ULP. The firmware running on uPower provides the following features:

- Power mode transition controller.
- Power meter for device-power domains consumption measurement.
- Temperature sensor for device temperature measurement.
- Messaging units for communication with on-chip processors.
- I2C for communication with PMIC.

Entering/exiting low-power modes is done by calling uPower FW API in either APD or RTD software. To configure the PMIC like setting, the power rail output voltage, limitation, etc. must be done by calling uPower FW I2C or PMIC APIs.

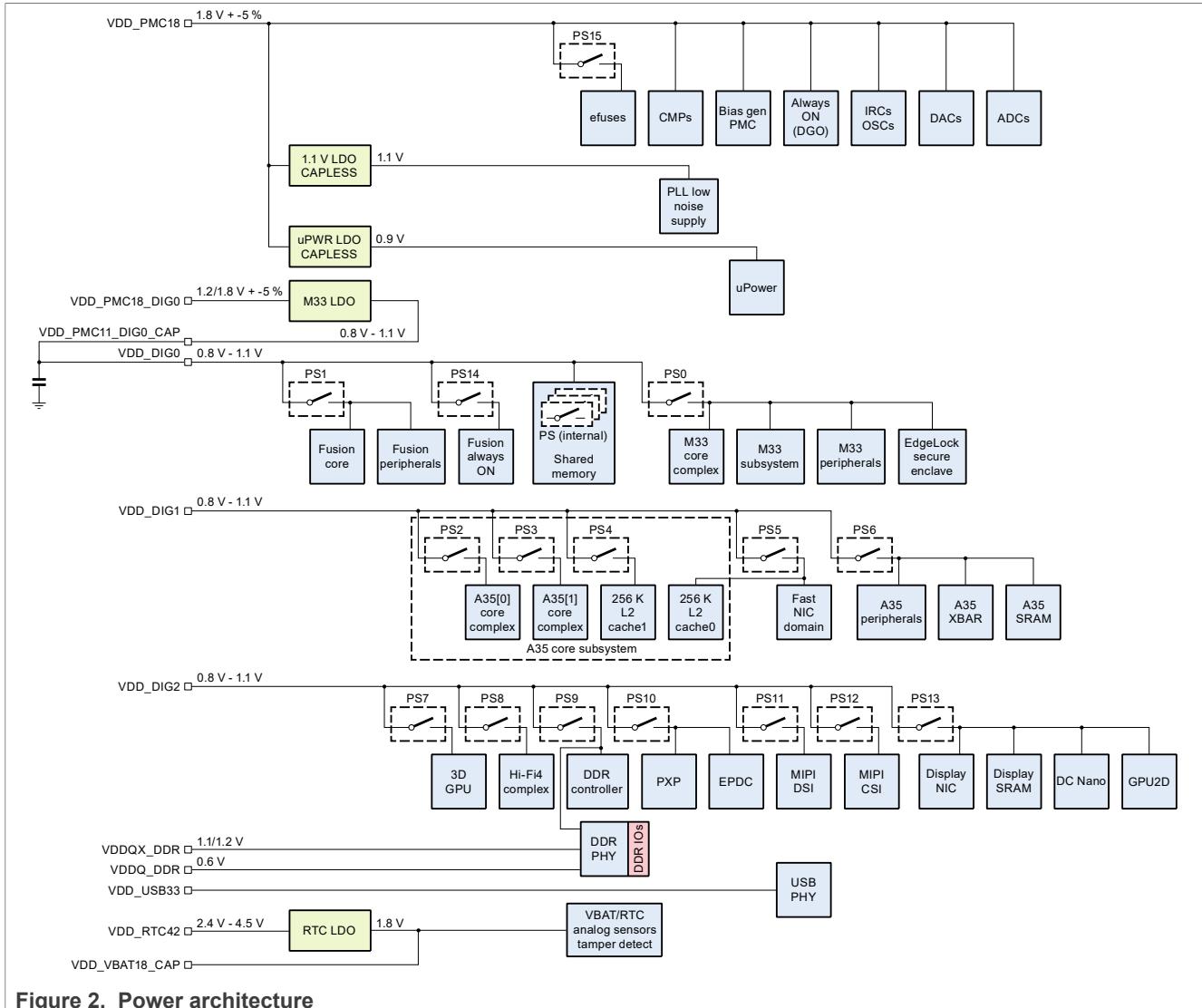


Figure 2. Power architecture

2.2 Power modes

Table 1 shows the available CA35 and CM33 power modes combination. The SoC does not support some of the combinations. For more details on each power mode, refer to the "Power Management" chapter in the *i.MX 8ULP Processor Reference Manual* (document i.MX8ULPRM).

Table 1. i.MX8ULP power modes

| CA35 | CM33 | | | | |
|------------------------|--------------------|--------------------|--------------------|------------|-----------------|
| | Active | Sleep | Deep sleep | Power down | Deep power down |
| Active | YES Scenario #1 | YES Scenario #3 | YES Scenario #3 | NO | NO |
| Partial active* | YES | YES | YES | NO | NO |

Table 1. i.MX8ULP power modes...continued

| CA35 | CM33 | | | | |
|------------------------|----------------------|--------------------|--------------------|--------------------|-----------------|
| | Active | Sleep | Deep sleep | Power down | Deep power down |
| Sleep | YES | YES | YES | NO | NO |
| Deep sleep* | YES | YES | YES | NO | NO |
| Power down | YES Scenario #2/4 | YES Scenario #2 | YES Scenario #2 | YES Scenario #2 | YES |
| Deep power down | YES | YES | YES | YES | YES |

*Linux does not support deep sleep or partial active mode for A35.

[Table 2](#) maps the Linux kernel power infrastructure to 8ULP power modes.

Table 2. Linux BSP supported power modes

| Linux power | 8ULP power modes |
|-------------|------------------|
| Run | Active |
| CPU idle | Sleep |
| Standby | N/A |
| Suspend | Power down |
| Power off | Deep power down |

According to different use cases and scenarios, user can either choose one or two or all three domains in major cases. These use-cases/scenarios can be put into following four categories:

1. All domains active – such as smart watch active.
2. RTD domain uses only – such as sensor hub and voice wake-up keyword detection in very low power.
3. APD active with LPAV – such as map navigation and E-Reader paging.
4. RTD active with LPAV – such as low-power display and Hi-Fi audio processing.

These four scenarios have been marked in [Table 1](#). The following chapters describe how to optimize the power consumption for scenario 2, 3, and 4. Active power optimizations of all domains can leverage the tips from other scenarios.

2.3 Driving modes

SoC can support different driving modes: over drive (OD), nominal drive (ND), and under drive (UD), which means SoC can run under different core voltages with corresponding bus and IP frequency. Users can select the right driving mode for their use cases and power requirement.

Default BSP boot up the SoC by putting APD/LPAV into OD mode and RTD into ND mode. Users can configure U-Boot and load specific kernel device-tree files for ND mode. The RTD domain only supports UD.

[Table 3](#) lists some key IP clocks under different modes.

Table 3. Key IP clocks under different modes

| Clock name | Over Drive (1.1 V) Frequency (MHz) | Nominal Drive (1.0 V) Frequency (MHz) |
|-------------|---------------------------------------|--|
| CM33_BUSCLK | 108 | 65 |
| DSP_CORECLK | 200 | 150 |
| FlexSPI0/1 | 400 | 150 |

Table 3. Key IP clocks under different modes...continued

| Clock name | Over Drive (1.1 V) Frequency (MHz) | Nominal Drive (1.0 V) Frequency (MHz) |
|------------------|---------------------------------------|--|
| NIC_AP_CLK | 460 | 241 |
| NIC_PER_CLK | 244 | 148 |
| uSDHC0 | 397 | 200 |
| uSDHC1 (PTE/F) | 200 | 100 |
| uSDHC2 (PTF) | 200 | 100 |
| HIFI4_CLK | 594 | 263 |
| NIC_LPAV_AXI_CLK | 316.8 | 200 |
| NIC_LPAV_AHB_CLK | 158.4 | 100 |
| DDR_CLK | 266 | 200 |
| DDR_PHY | 528 | 400 |
| GPU3D/2D | 316.8 | 200 |
| DCNano | 105 | 75 |

For more clocks, refer to the clock frequencies table in the *i.MX 8ULP Applications Processor—Industrial Products* (document IMX8ULPIEC).

3 RTD domain only

Consider `SDK_Power_mode_switch demo` as an example provided with the i.MX 8ULP SDK software release.

In this scenario, the AP and LPAV domains are in power down or deep Power-down mode, and M33 core or reset can wake them up. RTD domain can either be in active, sleep, deep sleep, or Power-down mode according to the power consumption and wake-up time requirements.

[Figure 3](#) and [Figure 4](#) show the power consumptions and wake-up time for each low-power mode.

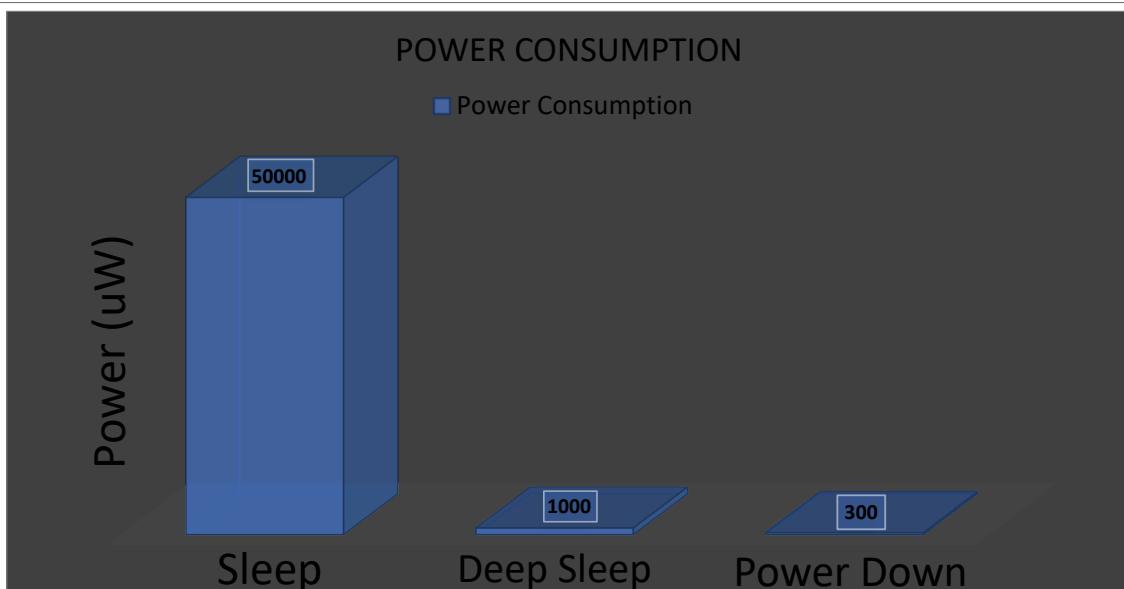


Figure 3. Power consumption in different power modes

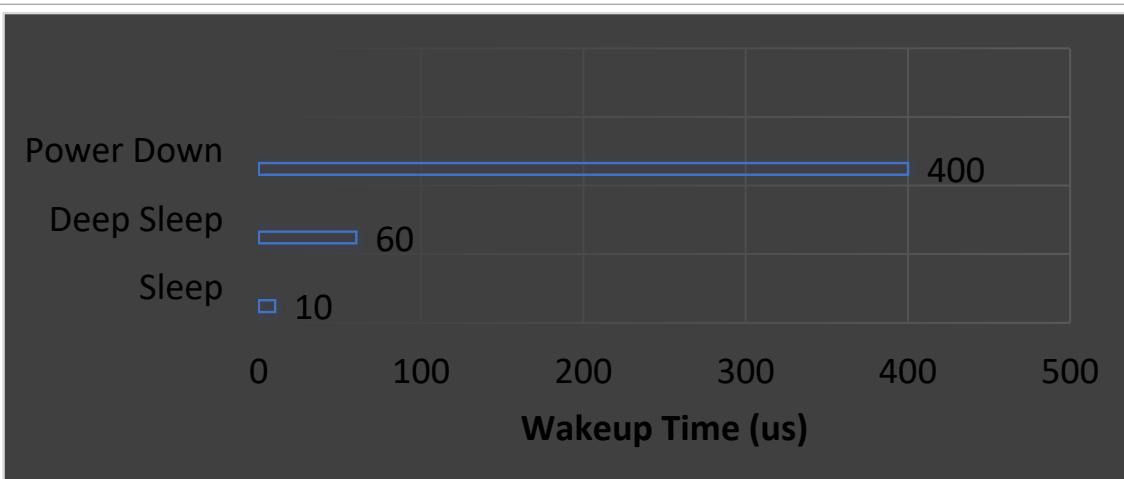


Figure 4. System wake-up time in different power modes

3.1 Select the right low-power mode

The user must select one or more right low-power modes of power saving according to the requirement. The below considerations must be taken:

- Consider the SoC power consumption, PD < 300 μ W, deep sleep < 1 mW, sleep < 50 mW
- Consider the wake-up time from low-power modes, PD > 400 μ s, deep sleep > 60 μ s, sleep > 10 μ s
- Consider the IPs used in the lowest power modes, by referring [Table 4](#).
For example:
 1. If LPI2C[3] must be functional or Async operation, but not CG/PG, use Sleep mode.
 2. If FlexSPI is required to be functional, the lowest power mode is sleep without system/bus clock gated.

Table 4. Power mode details (real-time domain)

| Modules | Power modes | Active | Sleep | Deep sleep | Power down | Deep power down |
|--------------|--------------------------|--|---|---|--|--|
| | Power state power domain | Core supply = ON, Bias = AFBB and DVS, System/Bus clocks = ON, I/O supply = ON | Core supply = ON, Bias = AFBB or ARBB, Voltage = fixed, System/Bus clock = ON (optional), I/O supply = ON | Core supply = ON, Bias = RBB Voltage/ Bias = prog, System/Bus clock = OFF, I/ O supply = ON | Core supply = ON (Mem only), Bias = RBB, Voltage/ Bias = prog, System/Bus clock = OFF, I/ O supply = ON (optional) | Core supply = OFF, Bias = RBB, Voltage/ Bias = prog, System/Bus clock = OFF, I/ O supply = ON (optional) |
| CCG0 | RTD | Functional | Functional | Functional (Limited) | PG | PG |
| PLL0 | PLL LDO | Functional | Functional | CG | PG | PG |
| PLL1 (Audio) | PLL LDO | Functional | Functional | CG | PG | PG |
| LPO (1 MHz) | RTD | Functional | Functional | Functional | PG | PG |
| SYSOSC | RTD | Functional | Functional | Functional | PG | PG |

For more details, refer to "Power mode details (real-time domain)" chapter in the *i.MX 8ULP Processor Reference Manual* (document i.MX8ULPRM).

Consider low-power voice wake-up use case as an example. The lowest power mode that a user can select is deep sleep. The mic-phone IP (MICFIL) can work under deep sleep with FRO clock on, which is not workable under Power-down mode.

3.2 Use the proper clocks

RTD domain has several clock sources, as shown in [Figure 5](#): SYSOSC, FRO, LPO, PLL0 (system PLL (SPLL)), and PLL1 (audio PLL (APLL)). Meanwhile, the RTD domain can also use the VBAT domain RTC32K/1K clock.

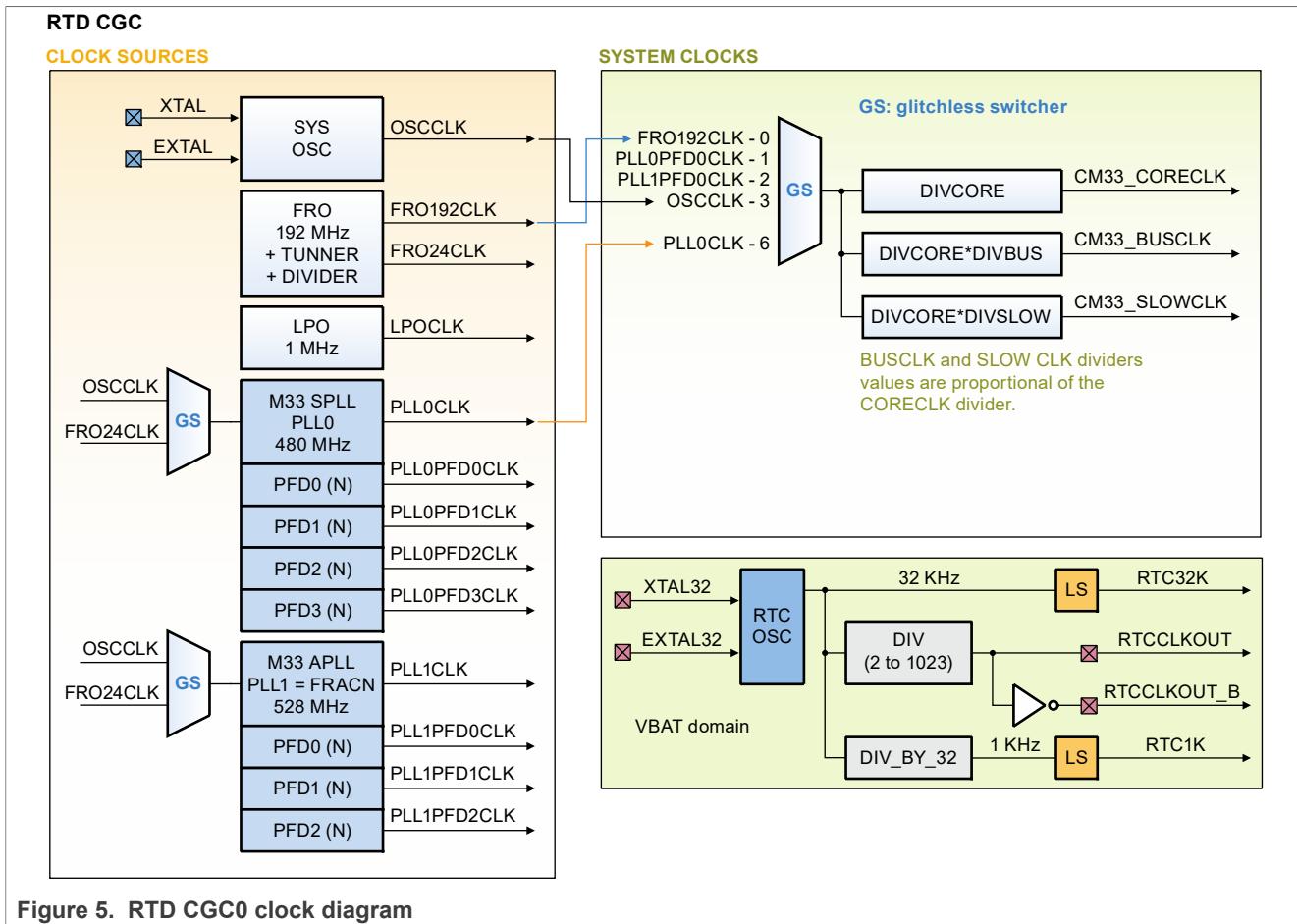


Figure 5. RTD CGC0 clock diagram

- The SYSOSC clock source is from an external onboard crystal, normal 24 MHz. PLL0/1 source and CM33 core/bus can use the SYSOSC clock source.
- The FRO is free running oscillator with a tuner, which can output 192 MHz and 24 MHz clock. FRO24 can be used for the PLL0/1 source, and FRO192 can be used for CM33 core/bus clocks.
- The LPO is fixed at 1 MHz, used by IP modules that must work in low-power modes like EWM and LPTMR.
- PLL0 is running at 480 MHz and PLL1 is 528 MHz. PLL0 is the system PLL, used by CM33 core/bus and FlexSPI. PLL1 is used by audio systems like SAI/MICFIL/MQS. They both can provide higher clock frequency for CM33 core/bus.

Since CM33 core/bus clock can be sourced from FRO or SYSOSC, it is better to avoid using PLL0/1 if a higher frequency is not required. Turning off the PLLs can save power significantly.

If PLLs are used for CM33 in active mode, they must be manually off before entering low-power modes (sleep/deep sleep/power down) to save power. This requires several steps:

1. Enable FRO or SYSOSC with *DSEN bit settings in SCR registers according to Fusion DSP usage in low-power modes¹
2. Wait for clock validity by checking the VLD bit set in the SCR register.
3. Disable the IP modules that use PLLs, or switch the clock to FRO or SYSOSC.
4. Switch the CM33 clock to FRO or SYSOSC with core/bus/slow clock DIV settings in the CGC0.CM33CLK.
5. Wait for several microseconds. To wait for the clock stable, check the CM33LOCKED bit.

¹ If FRO or SYSOSC clock is enabled by setting SOSCDSEN or FRODSEN bits, fusion DSP can optionally continue to function when M33 is in the deep sleep.

6. Disable the PLL0/1 by clearing the SCR PLLEN bit.

3.3 Power off and clock gate unused IP modes and SRAM partition

For the RTD domain, several power switches can be on/off (refer to [Section 7](#)):

- PS0: CM33 core, peripherals, and EdgeLock enclave
- PS1: Fusion DSP core
- PS14: Fusion AON
- PS15: eFuse

In SDK, user can call `UPOWER_PowerOffSwitches(upower_ps_mask_t mask)` and `UPOWER_PowerOnSwitches(upower_ps_mask_t mask)` to turn off and on the modules as needed. [Table 7](#) displays the mask parameters value.

For CM33 peripherals (IP module) which are not used, leave it as disable status (reset value), or disable it by clearing its enabled bit, like LPI2C MCR master enable bit. Make sure that the PCC clock gate control bit is cleared, for example, `PCC1.PCC_LPI2C0[CGC]` bit. In RTD domain, all IP clocks can be clock gated or ungated by PCC clock modules.

Memory partition is also a consideration to save power if those memories are not used. In SDK, user can call `UPOWER_PowerOffMemPart(uint32_t mask0, uint32_t mask1)` and `UPOWER_PowerOnMemPart(uint32_t mask0, uint32_t mask1)` to turn off and on the memory partitions as needed. [Table 8](#) displays the mask0/1 parameters value.

3.4 Entering low-power mode

Before entering the low-power modes (sleep/deep sleep/power down), several steps must be performed to ensure power consumption is low in those modes:

- General PAD settings in the SIM module
There are two types of I/O PADs inside SoC: FGPIO (PTA/B/E/F) and HGPIO (PTC/D). To save power under the low-power mode, the user should:
 - Disable the compensation function for HGPIO by clearing the `COMPE` bit in the `PTC/D_COMPCELL` registers.
 - Limit I/O operation range for FGPIO, which works within 1.8 V by setting `PTx_OPERATION_RANGE` bit in the `DGO_GP10/11` of `RTD_SEC_SIM` and `DGO_GP4/5` of `APD_SIM`. On EVK, the PTB works for 1.8 V. User should limit the PTB operation range to 1.8 V by setting `RTD_SEC_SIM[DGO_GP11] = 0x1`.
- Disable I/O pins by setting the PAD mux to analog hi-Z function
Except for the pins which are used by GPIO wake-up or module function in low-power modes, all the other PTA/B/C pins should be set to analog high-Z function to save power. Clearing the mux bits in `IOMUXC0.PCR0_PTA/B/Cx` registers can achieve this. In SDK, the user can directly assign 0 to the below array items:

```
PTA: IOMUXC0->PCR0_IOMUXCARRY0[x]
PTB: IOMUXC0->PCR0_IOMUXCARRY1[x]
PTC: IOMUXC0->PCR0_IOMUXCARRY2[x]
```

For example, `IOMUXC0->PCR0_IOMUXCARRY0[1] = 0` can disable the PTA1.

Note: Since the PMIC must be configured through I2C (PTB10/11) during the power mode transition, you cannot disable these pins.

To keep an I/O pin to act as a wake-up source, the below settings should be done for different power modes:

– Power-down mode:

1. Enable the pin bit in the `WU00_P1/PE2` registers.
2. Configure the pin mux in `IOMUXC0->PCR0_IOMUXCARRYx` to `WU00_Pxx` function. For details, refer to the I/O signal table attached in the *i.MX 8ULP Processor Reference Manual* (document i.MX8ULPRM).

- Sleep/Deep sleep mode: Setup the interrupt controller registers of the GPIO group (`GPIOx->ICR`) correctly.
- Display PLLs - Switch core/bus clocks to FRO or LPO.
- Setup PMIC to adjust power supply voltage for low-power modes
i.MX 8ULP supports adjusting of `VDD_DIG0/1/2` power rail voltage or directly power off some rails (only support switch off `LSW1` `VDD_PTC` in current EVK and SDK under power down modes) during power modes transition. Lowering the voltage in low-power modes can reduce power consumption in an effective way. Power off some rails can cut off the power directly to save power. [Table 5](#) shows the typical voltages of `VDD_DIG0/1` under different power modes (`VDD_DIG2` is tied with `DIG1` on the EVK board. It can be adjusted together with `VDD_DIG1`).

Table 5. Power supply voltage under different power modes

| Power rail | Active | Sleep | Deep sleep | Power down |
|-----------------|--------|--------|------------|------------|
| VDD_DIG0 | 1.05 V | 1.05 V | 0.73 V | 0.65 V |
| VDD_DIG1 | 1.05 V | 1.05 V | 0.73 V | 0.73 V |

To low down the voltage of power rails, the user should tell uPower how to configure the PMIC during power transition by adding items of `ps_rtd_pmic_reg_data_cfgs_t` structure into `pwr_sys_cfg->ps_rtd_pmic_reg_data_cfg[]` array. Take PCA9460 PMIC on EVK as an example below:

1. Enter Power-down mode:
 - a. Low down BUCK2 (`VDD_DIG0`) to 0.65 V.
 - b. Switch off `LSW1` for PTC I/O power supply.
2. Exit Power-down mode:
 - a. Raise up BUCK2 (`VDD_DIG0`) back to 1.0 V.
 - b. Switch on `LSW1` for PTC I/O power supply.

```
static ps_rtd_pmic_reg_data_cfgs_t rtd_pmic_reg_data_cfgs = {
    /* RTD Power Down: set BUCK2OUT_DVS0 to 0.65V */
    [0] =
    {
        .tag      = PMIC_REG_VALID_TAG,
        .power_mode = PD_RTID_PWR_MODE,
        .i2c_addr = 0x15,
        .i2c_data = 0x04,
    },
    /* RTD Power Down: off LSW1 */
    [1] =
    {
        .tag      = PMIC_REG_VALID_TAG,
        .power_mode = PD_RTID_PWR_MODE,
        .i2c_addr = 0x40,
        .i2c_data = 0x00,
    },
    /* RTD Active: set BUCK2OUT_DVS0 to 1.0V */
    [2] =
    {
        .tag      = PMIC_REG_VALID_TAG,
        .power_mode = ACT_RTID_PWR_MODE,
        .i2c_addr = 0x15,
        .i2c_data = 0x20,
    },
    /* RTD Power Down: on LSW1 */
    [3] =
    {
        .tag      = PMIC_REG_VALID_TAG,
        .power_mode = ACT_RTID_PWR_MODE,
    }
};
```

```

        .i2c_addr    = 0x40,
        .i2c_data    = 0x11,
    },
};


```

In the structure, the `power_mode` member defines the target power modes for this PMIC setting, for example, `PD_RT_D_PWR_MODE`, which means that this setting is applied when the power mode is transferred to power down. The `i2c_addr` is the register address inside PMIC, and `i2c_data` is the register value that must be configured.

For more information on register address and bits, refer to the *PCA9460, Power Management IC for i.MX 8ULP Data Sheet* (document [PCA9460DS](#)).

- Set up uPower for the power switch, memory partition switch, and PAD configuration:

```

static ps_rtd_pwr_mode_cfgs_t rtd_pwr_mode_cfgs = {
    ...
    [PD_RT_D_PWR_MODE] =
    {
        .in_reg_cfg      = IN_REG_CFG(0x00000000, 0x00000000),
        .pmic_cfg        = PMIC_CFG(0x00000023, 0x00000000),
        .pad_cfg         = PAD_CFG(0x00000003, 0x00000000, 0x00000000),
        .mon_cfg         = MON_CFG(0x00000000, 0x0, 0x0),
        .bias_cfg        = BIAS_CFG(0x00010001, 0x0001001a, 0x0001001a,
0x00000001),
        .pwrsys_lpm_cfg = PWRSYS_LPM_CFG(0),
    },
    ...
}

static ps_rtd.swt_cfgs_t rtd.swt_cfgs = {
    ...
    /* Power Down */
    [PD_RT_D_PWR_MODE] =
    {
        .swt_board[0] = SWT_BOARD(0x0, 0x00060003),
        .swt_mem[0]   = SWT_MEM(0x003fe000, 0x0, 0x003ff3ff),
        .swt_mem[1]   = SWT_MEM(0x00000000, 0x00000000, 0x00000000),
    },
    ...
}

```

For these two structures for power mode transition, refer to `lpm.c` in the `power_mode_switch` demo. User can keep those settings untouched unless additional settings are required such as, power on/off, some IP modules, and memory array. Users can turn on/off power switches by setting the `swt_board[0]`: `SWT_BOARD`(on/off bits, masks). The bits definition can be found in [Table 7](#). Power on/off memory array can be done by setting the `swt_mem[0]`: `SWT_MEM`(SRAM Ctrl array bits, SRAM peripheral bits, masks). The bits definition can be found in [Table 8](#).

For more details on power mode transition settings of uPower, refer to the *uPower Firmware User's Guide* (document [UPOWERFWUG](#)).

- Call uPower for power transition. Take entering power down mode as an example, refer to the function of `LPM_SystemPowerDown(void)` in the `SDK power_mode_switch` demo.

After the system wakes up from low-power modes, the user must recover all the registers settings before entering. For example, in IOMUXC settings, the user can use a static array variable to store the values of all PCR0 and restore them.

4 APD domain active with LPAV

Take [NXP Linux release](#) as an example operating system for APD domain.

4.1 Put RTD into sleep

Keeping the RTD domain in Sleep mode can save around 20 mW ~ 40 mW compared to active mode. Also, make sure that the unused GPIO pins are off.

4.2 Disable unused IP and pins in the Linux DTS (device tree)

Disable the device node can avoid powering up this device or ungated its clock. For example, to disable GPU3D in the device tree source (DTS):

```
&gpu3d {  
    status = "disabled";  
}
```

To prevent the power switch PS7 from turning on, disable GPU3D. If DCNano, MIPI DSI/CSI, and GPU2D are all disabled, then PLL4 does not get enabled.

To avoid enabling I/O PAD for those pins, disable unused pins in pinctrl nodes.

4.3 Use DVFS

i.MX 8ULP Linux supports voltage and frequency scaling features, formally known as DVFS on other i.MX platforms. The voltage/frequency scaling features are not dynamically implemented in the software. The user must switch using the Linux kernel sysfs. To use VFS, load the `imx8ulp-evk-nd.dtb` as a default device tree to boot up the system. Then enter low bus mode by:

```
echo 1 > /sys/devices/platform/imx8ulp-lpm/enable
```

Kernel does the following changes:

- Reduce the DDR core frequency from 528 MHz to 96 MHz.
- Reduce the APD NIC clock to 192 MHz by using FRO as a clock source instead of PLL.
- Reduce the LPAV AXI clock to 192 MHz by using FRO as a clock source instead of PLL.
- Reduce A35 cpu clock to 500 MHz.
- Low down the BUCK3 power rail (VDD_DIG1/2) voltage to 1.0 V from 1.1 V.

Exit and go back to high bus mode:

```
echo 0 > /sys/devices/platform/imx8ulp-lpm/enable
```

4.4 Use nominal drive mode (VDD_DIG1/2 1.0 V)

i.MX 8ULP SoC runs in overdrive mode by default U-Boot and kernel configurations. If high performance is not a key requirement, the user can run the SoC in nominal drive mode on boot to save power. It is a static configuration; user cannot dynamically change the voltage or frequency after boot up.

U-Boot: Build U-Boot with `imx8ulp_evk_nd_defconfig` configuration. It does the following changes:

- Low down the VDD_DIG1/2 (BUCK3) power rail to 1.0 V while bootup.
- Configure the DDR clock to 266 MHz instead of 528 MHz.

- Reduce the LPAV/APD NIC clock to 192 MHz.
- Reduce the A35 core clock to 750 MHz.

Kernel: load imx8ulp-evk-nd.dtb on boot. It reduces the GPU2D/3D clock to 200 MHz, HiFi4 DSP core clock to 260 MHz, uSDHC0 to 194 MHz, and uSDHC1/2 to 97 MHz.

5 RTD domain active with LPAV

Take the "always-on display" use case as an example, available with this application note. In this case, RTD accesses the DCNano display controller to display contents in the PSRAM. For details, refer to the code attached to this application note.

5.1 Enable LPAV domain

After Linux suspends, the AP and LPAV domain enters Power-down mode. RTD must take ownership of the LPAV domain from APD first:

- SIM_RTD_SEC.SYSCTRL0[LPAV_MASTER_CTRL] = 0 // sets the RTD to be the master domain of LPAV domain
- SIM_RTC_SEC.LPAV_MASTER_ALLOC_CTRL = 0 // allocates LPAV master IP to RTD
- SIM_RTC_SEC.LPAV_SLAVE_ALLOC_CTRL = 0 // allocates LPAV slave IP to RTD

Then, resume the VDD_DIG2 (BUCK3) core power of the LPAV domain to 1.05 V or 1.1 V to ensure all IPs in the LPAV works properly by `uPower_upwr_vtm_pmic_config()` API.

Finally, pull out the LPAV domain from Power-down mode to active mode:

```
UPOWER_PowerOnSwitches(kUPOWER_PS_AV_NIC);
```

5.2 Turn on the power switches

Turn on the IPs used later:

```
UPOWER_PowerOnMemPart((uint32_t)(kUPOWER_MP0_DCNANO_A |  
kUPOWER_MP1_FLEXSPI1 | kUPOWER_MP0_DCNANO_B |  
kUPOWER_MP0_MIPI_DSI), 0U);  
UPOWER_PowerOnSwitches((upower_ps_mask_t)(kUPOWER_PS_MIPI_DSI));
```

In the always-on display use case, the user must turn on the following to get the whole display pipeline working:

- MIPI-DSI power switch
- Memory partitions for DCNano display controller
- MIPI-DSI
- FlexSPI FIFO buffers

5.3 Configure the clocks

LPAV domain only has one PLL for clock sources. So the user must enable it and its PFD to drive IPs.

Enable PLL4 with its PFD and PFDDIV

```
const cgc_pll4_config_t g_cgcPll4Config = { .enableMode = kCGC_PllEnable,  
.div1 = 0U,  
.pfld0Div1 = 11,  
.pfld0Div2 = 0U,
```

```

.pfd1Div1 = 0U,
.pfd1Div2 = 0U,
.pfd2Div1 = 0U,
.pfd2Div2 = 0U,
.pfd3Div1 = 0U,
.pfd3Div2 = 0U,
/src = kCGC_PllSrcSysOsc,
.mult = kCGC_Pll4Mult22,
.num = 578,
.denom = 1000};

CLOCK_InitPll4(&g_cgcPll4Config); // PLL4 528Mhz
CLOCK_EnablePll4PfdClkout(kCGC_PllPfd0Clk, 32); // pll4pfd0 (528 MHz * 18) / 32
= 297 MHz
CGC_LPAV->PLL4DIV_PFD_0 = CGC_LPAV_PLL4DIV_PFD_0_DIV1(11 - 1); // pll4pfd0div1
297 / 11 = 27 MHz

```

Select the PLL4 PFD0DIV1 as the clock source for DCNano and enable its clock in PCC:

```

CLOCK_SetIpSrcDiv(kCLOCK_Dcnano, kCLOCK_Pcc5PlatIpSrcPll4Pfd0Div1, 0, 0U);
CLOCK_EnableClock(kCLOCK_Dcnano);

```

After the power switch is turned on and the clocks are ready, the user can use SDK drivers to access and control the LPAV domain IPs.

6 Related documentation/resources

Table 6 lists additional documents and resources that can be referred to for more information. Some of the documents listed below may be available only under a non-disclosure agreement (NDA). To request access to these documents, contact local field applications engineer (FAE) or sales representative.

Table 6. Related documentation/resources

| Document | Link/how to access |
|---|---|
| <i>PCA9460, Power Management IC for i.MX 8ULP Data Sheet</i> (document PCA9460DS) | PCA9460DS |
| <i>uPower Firmware User's Guide</i> (document UPOWERFWUG) | UPOWERFWUG |
| <i>i.MX 8ULP Processor Reference Manual</i> (document i.MX8 ULPRM) | Contact NXP local field applications engineer (FAE) or sales representative. |
| <i>i.MX 8ULP Applications Processor—Industrial Products</i> (document IMX8ULPIEC) | Contact NXP local field applications engineer (FAE) or sales representative. |
| MCUXpresso SDK Builder | https://mcuxpresso.nxp.com/en/welcome |

7 Appendix

Table 7 shows the name, logical number, and bit for each power switches.

Table 7. Power Switches

| Function | Logical power switch | Bit |
|-------------|----------------------|-----|
| CM33 | PS0 | 0 |
| Fusion | PS1 | 1 |
| A35[0] Core | PS2 | 2 |

Table 7. Power Switches...continued

| Function | Logical power switch | Bit |
|----------------------|----------------------|-----|
| A35[1] Core | PS3 | 3 |
| Mercury L2 Cache [1] | PS4 | 4 |
| Fast NIC / Mercury | PS5 | 5 |
| APD Periph | PS6 | 6 |
| GPU3D | PS7 | 7 |
| HiFi4 | PS8 | 8 |
| DDR Controller | PS9 | 9 |
| PXP, EPDC | PS13 | 10 |
| MIPI-DSI | PS14 | 11 |
| MIPI CSI | PS15 | 12 |
| NIC AV / Periph | PS16 | 13 |
| Fusion AO | PS17 | 14 |
| FUSE | PS18 | 15 |
| uPower | PS19 | 16 |

[Table 8](#) shows the bit and name of each memory partition controller.

Table 8. Memory partition ctrls

| SRAM_CTRL_ARRAY_0 (APD/LPAV) Mask0 | | SRAM_CTRL_ARRAY_1 (RTD) Mask1 | |
|---------------------------------------|--------------------------------|----------------------------------|--------------------------------------|
| Bit | Memories controlled | Bit | Memories controlled |
| 0 | CA35 Core 0 L1 cache | 0 | Casper RAM |
| 1 | CA35 Core 1 L1 cache | 1 | DMA0 RAM |
| 2 | L2 Cache 0 | 2 | FlexCAN RAM |
| 3 | L2 Cache 1 | 3 | FlexSPI0 FIFO, Buffer |
| 4 | L2 Cache victim/tag | 4 | FlexSPI1 FIFO, Buffer |
| 5 | CAAM Secure RAM | 5 | CM33 Cache |
| 6 | DMA1 RAM | 6 | PowerQuad RAM |
| 7 | FlexSPI2 FIFO, Buffer | 7 | ETF RAM |
| 8 | SRAM0 | 8 | Sentinel PKC, Data RAM1, Inst RAM0/1 |
| 9 | AD ROM | 9 | Sentinel ROM |
| 10 | USB0 TX/RX RAM | 10 | uPower IRAM/DRAM |
| 11 | uSDHC0 FIFO RAM | 11 | uPower ROM |
| 12 | uSDHC1 FIFO RAM | 12 | CM33 ROM |
| 13 | uSDHC2 FIFO and USB1 TX/RX RAM | 13 | SSRAM Partition 0 |
| 14 | GIC RAM | 14 | SSRAM Partition 1 |
| 15 | ENET TX FIXO | 15 | SSRAM Partition 2,3,4 |
| 16 | Reserved (Brainshift) | 16 | SSRAM Partition 5 |

Table 8. Memory partition ctrls...continued

| SRAM_CTRL_ARRAY_0 (APD/LPAV) Mask0 | | SRAM_CTRL_ARRAY_1 (RTD) Mask1 | |
|---------------------------------------|---------------------------------------|----------------------------------|-------------------------------|
| Bit | Memories controlled | Bit | Memories controlled |
| 17 | DCNano Tile2Linear and RGB Correction | 17 | SSRAM Partition 6 |
| 18 | DCNano Cursor and FIFO | 18 | SSRAM Partition 7_a (128 kB) |
| 19 | EPDC LUT | 19 | SSRAM Partition 7_b (64 kB) |
| 20 | EPDC FIFO | 20 | SSRAM Partition 7_c (64 kB) |
| 21 | DMA2 RAM | 21 | Sentinel Data RAM0, Inst RAM2 |
| 22 | GPU2D RAM Group 1 | 22 | Reserved |
| 23 | GPU2D RAM Group 2 | 23 | |
| 24 | GPU3D RAM Group 1 | 24 | |
| 25 | GPU3D RAM Group 2 | 25 | |
| 26 | HIFI4 Caches, IRAM, DRAM | 26 | |
| 27 | ISI Buffers | 27 | |
| 28 | MIPI-CSI FIFO | 28 | |
| 29 | MIPI-DSI FIFO | 29 | |
| 30 | PXP Caches, Buffers | 30 | |
| 31 | SRAM1 | 31 | |

8 Note about the source code in the document

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

Copyright YYYY NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

9 Revision history

[Table 9](#) summarizes the changes done to this document since the initial release.

Table 9. Revision history

| Revision number | Date | Substantive changes |
|-----------------|-------------|---------------------|
| 0 | 30 May 2023 | Initial release |

10 Legal information

10.1 Definitions

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

10.2 Disclaimers

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

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

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

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

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

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

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

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

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

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

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

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

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

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

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

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

10.3 Trademarks

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

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

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

EdgeLock — is a trademark of NXP B.V.

i.MX — is a trademark of NXP B.V.

Contents

| | | |
|-----------|--|-----------|
| 1 | Introduction | 2 |
| 2 | Overview | 2 |
| 2.1 | Power architecture | 2 |
| 2.2 | Power modes | 3 |
| 2.3 | Driving modes | 4 |
| 3 | RTD domain only | 5 |
| 3.1 | Select the right low-power mode | 6 |
| 3.2 | Use the proper clocks | 7 |
| 3.3 | Power off and clock gate unused IP modes and SRAM partition | 9 |
| 3.4 | Entering low-power mode | 9 |
| 4 | APD domain active with LPAV | 12 |
| 4.1 | Put RTD into sleep | 12 |
| 4.2 | Disable unused IP and pins in the Linux DTS (device tree) | 12 |
| 4.3 | Use DVFS | 12 |
| 4.4 | Use nominal drive mode (VDD_DIG1/2 1.0 V) | 12 |
| 5 | RTD domain active with LPAV | 13 |
| 5.1 | Enable LPAV domain | 13 |
| 5.2 | Turn on the power switches | 13 |
| 5.3 | Configure the clocks | 13 |
| 6 | Related documentation/resources | 14 |
| 7 | Appendix | 14 |
| 8 | Note about the source code in the document | 16 |
| 9 | Revision history | 17 |
| 10 | Legal information | 18 |

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.