AN13847

Debug and Application for RT1180 Clock and Low-power Feature

Rev. 1 — 27 May 2024 Application note

Document information

Information	Content
Keywords	AN13847, RT1180, low power, wake up, debug, skill
Abstract	This application note discusses clock and low-power feature in RT1180 and introduces some debug and application skills when developing a low-power use case.

1 Introduction

The i.MX RT series of crossover MCUs are part of the EdgeVerse edge computing platform and feature Arm Cortex-M cores. It has high performance real-time functionalities and MCU usabilities at a cost-effective price. The i.MX RT1180 crossover MCU family includes a Gb Time Sensitive Networking (TSN) switch to enable real-time rich networking integration. The integration handles both time-sensitive and industrial real-time communication. The i.MX RT1180 supports multiple protocols and bridging communications between real-time Ethernet and industry 4.0 systems. This family includes a state-of-the-art EdgeLock secure enclave, a dual-core architecture with both an 800 MHz Cortex-M7, and a 240 MHz Cortex-M33 for ultimate design flexibility.

This application note discusses clock and low-power features in RT1180 and introduces some debug and application skills when developing a low-power use case.

2 RT1180 power domains

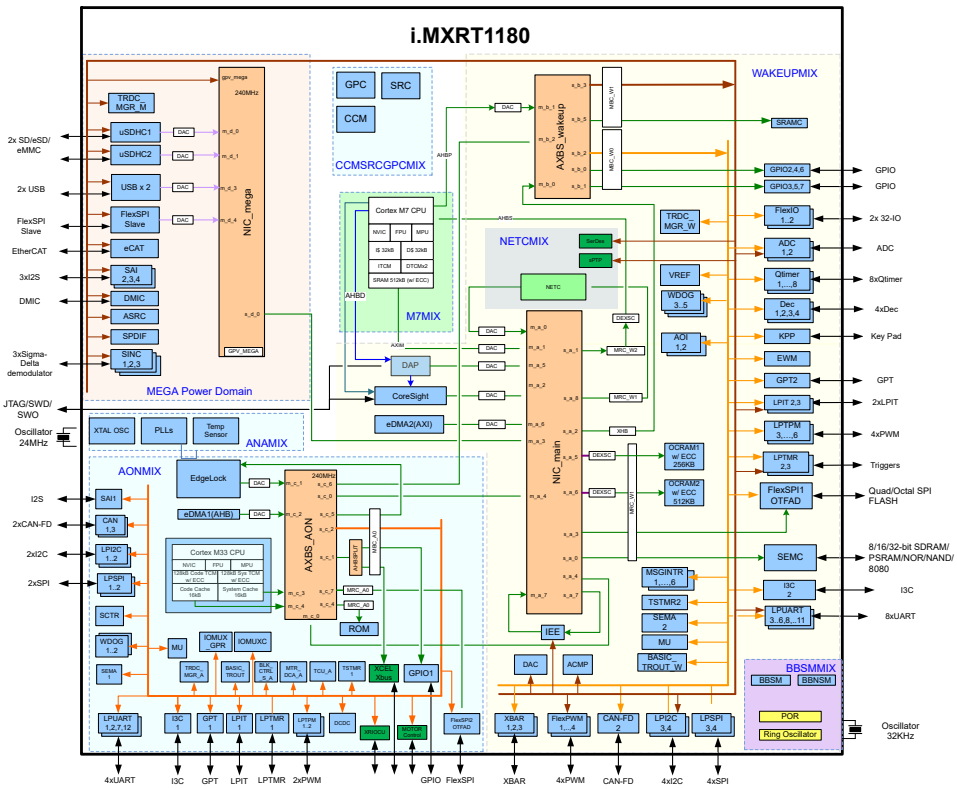

Figure 1. RT1180 power domain diagram

The peripherals of RT1180 are allocated to each power domain according to the function and attribute.

RT1180 has several power domains or power MIX. They are M7 platform MIX, CM33 platform MIX, AON MIX, WAKEUP MIX, MEGA MIX, NETC MIX, and BBSM MIX.

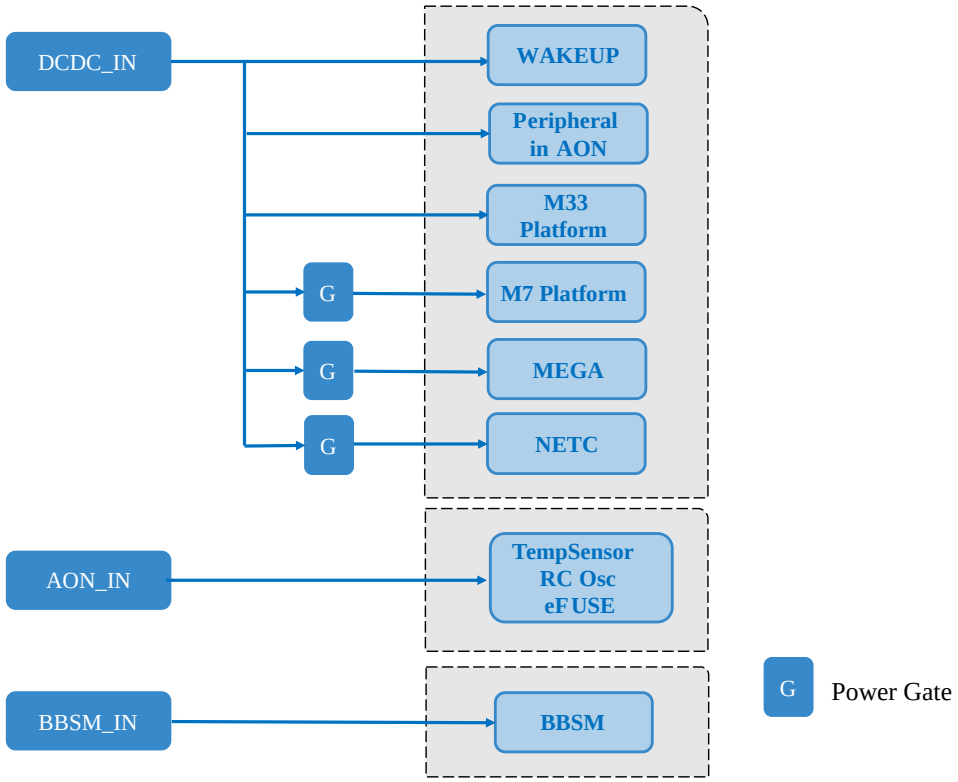

Figure 2. RT1180 power supply diagram

As shown in Figure 1, for WAKEUP MIX, peripherals in AON MIX, M33 Platform, M7 platform, MEGA MIX, and NETC MIX are powered by DCDC. But there is a minor difference. WAKEUP MIX, Peripheral in AON domain and M33 Platform are powered directly by DCDC without a power gate while M7 platform, MEGA MIX, and NETC MIX has their own power gate. For Temperature Sensor, RC OSC, and eFuse, they are powered by `AON_IN` without power gate. For BBSM domain, it is powered by `BBSM_IN`.

Table 1. Peripherals in each domain

Domain	Description
CM33	The Cortex-M33 core, cache, 256 kB TCM, and other peripherals.
CM7	The Cortex-M7 core, cache, 512 kB TCM, and other peripherals.
WAKEUP	SRAMC, GPIO2, GPIO3, GPIO4, GPIO5, GPIO6, GPIO7, FlexIO1, FlexIO2, ADC1, ADC2, Qtimer1, Qtimer2, Qtimer3, Qtimer4, Qtimer5, Qtimer6, Qtimer7, Qtimer8, DEC1, DEC,2, DEC3, DEC4, KPP, EWM, GPT2, LPIR2, LPIT3, LPTPM3, LPTPM4, LPTPM5, LPTPM6, LPTMR2, LPTMR3, Flex SPI1, SEMC, I3C2, LPUART3, LPUART4, LPUART5, LPUART6, LPUART8, LPUART9, LPUART10, LPUART11, LPSPI3, LPSPI4, LPI2C3, LPI2C4, CAN-FD2, FlexPWM1, FlexPWM2, FlexPWM3, FlexPWM4, XBAR1, XBAR2, XBAR3, VREF, WDOG3, WDOG4, WDOG5, AOI1, AOI2, OCRAM1, OCRAM2, MSGINTR1, MSGINTR2, MSGINTR3, MSGINTR4, MSGINTR5, MSGINTR6, TSTMR2, SEMA2, MU, ACMP, DAC, IEE, eDMA2
AON	SAI1, CAN1, CAN3, LPI2C1, LPI2C2, SCTR, WDOG1, WDOG2, SEMA1, LPUART1, LPUART2, LPUART7, LPUART12, I3C1, GPT1, LPIT1, LPTMR1, LPTPM1, LPTPM2, GPIO1, FlexSPI2, TSTMR, TRDC_MGR_A, MU, eDMA1
MEGA	uSDHC1, uSDHC2, USB_OTG1, USB_OTG2, FlexSPI Slave, EhterCAT, SAI2, SAI3, SAI4, DMIC, ASRC, SPDIF, SINC1, SINC2,SINC3
NETC	NETC, gPTP

3 Power state of RT1180

The power state of RT1180 falls into two parts, CPU mode and System Sleep (SS) state. Here are some examples:

- Run Mode, No SS
- Stop Mode, SS
- Suspend Mode, NO SS

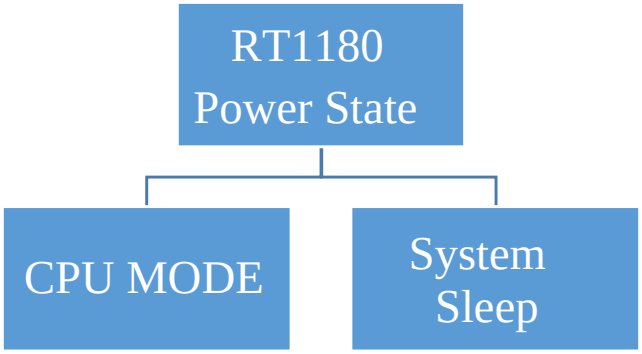

Figure 3. RT1180 power state

3.1 CPU mode

Each CPU platform has its own power mode. There are four CPU modes in RT1180: RUN, WAIT, STOP, and SUSPEND. Table 2 shows the differences.

Table 2. Difference among CPU modes

Power mode	CPU	CPU clock	CM33 power	CM7 power	On-platform peripheral	Exit time latency
RUN	RUN	ON	ON	ON	ON	—
WAIT	WFI/WFE	OFF	ON	ON[1]	ON	Extremely short
STOP	WFI/WFE	OFF	ON	ON[1]	ON	Short
SUSPEND	WFI/WFE	OFF	ON	OFF[1]	OFF	Long
NOTE	The On-platform peripheral state relates with the CPU power state. For example, there is a fast GPIO in M7 platform and usually GPIO can be a wake-up source for the whole system. But if the CM7 enters the suspend mode or other CPU-power-down mode, the fast GPIO cannot be a wake-up source, because the fast GPIO is powered down.					

[1] Configurable by SRC.

From the CPU side, the major difference is the CPU state. CPU enters WFI or WFE state and waits for the wake-up signals. Meanwhile, the clock and power of CPU also enter their own state. The exit latency or wake up time is impacted by the clock and power state. The CPU powers off under the SUSPEND mode. It means that it takes a longer wake-up latency than WAIT and STOP mode.

3.2 System Sleep (SS)

The SS mode is a low-power mode that has distinguishing settings outside CPU mode. When the system is in the Sleep mode:

- Analog modules, such as, LDO/BG, can be put into their own Standby mode .

- DCDC can be decreased into lower voltage to reduce leakage.
- DCDC module clock can be turned off.
- Almost all the system is stopped and only the wake-up source is alive.

The Sleep mode is related to the state of all CPU platforms. The system sleep controller in GPC maintains the system sleep sequence. Only when all CPU platforms send system sleep requests, the system enters the system sleep mode.

Before entering the System Sleep mode, both CPUs are under a low-power mode. It can be a WAIT, STOP, or SUSPEND mode. If a core does not enter a low-power mode, the chip cannot enter the SS mode. The low-power mode for each core can be difference, such as, CM33 STOP SS and CM7 SUSPEND SS.

In theory, WAIT mode can also enter the SS mode. But generally speaking, WAIT mode only stops the CPU, while other peripherals, such as, bus, DMA, are still working. But the Sleep mode stops the internal bus and almost all the system is stopped. Therefore, it is meaningless to use System Sleep mode in Wait mode.

4 Debug and application skills

4.1 Clock output

The clock output helps to know whether a clock or a PLL works well or not, including enable or disable status, the frequency under run mode or low-power mode. RT1180 includes two clock output pins, `CLKO1` and `CLKO2`. These pins can fan out the clock to a pin. A scope can be used to measure the information which users need. It is recommended to remain two testing pads for this function.

- `CLKO1`: `GPIO_SD_B1_00`
- `CLKO2`: `GPIO_SD_B1_01`

For `CLKO1`, the following clocks can fan out:

- `OscRc24M`
- `OscRc400M`
- `SysPll3Div2`
- `SysPll1Div2`

For `CLKO2`, the following clocks can fan out:

- `OscRc24M`
- `OscRc400M`
- `SysPll1Div5`
- `ArmPllOut`

Because the output capacity of the pin is limited, using the internal frequency divider to output after frequency division for higher frequency clocks is recommended. Take `CLKO1` output `SysPll1Div2` as an example:

1. Initialize the pin:

```
IOMUXC_SetPinMux(IOMUXC_GPIO_SD_B1_00_CCM_CLKO1,0U);
```

2. Configure the clock source and divider. `SysPll1Div2` works at a high frequency, 500 MHz. It means that a suitable frequency division factor must be configured.

```
rootCfg.mux = 3; //Select SysPll1Div2 as the clock source
rootCfg.div = 20; // Division factor is 20
CLOCK_SetRootClock(kCLOCK_Root_Cko1, &rootCfg);
```

3. If the PLL is enabled, the scope can measure a waveform.

4.2 Clock observer

RT1180 contains a clock observer. It is like an oscilloscope that can measure the frequency of the clock. It is a useful function on developing the power mode and other clock-related cases. There is an example in power mode switch:

```
void PrintSystemClocks(void);
```

This API can print the frequency of `M33_Clock_Root`, `M7_Clock_Root`, `EDGELOCK_Clock_Root`, `BUS_AON_Clock_Root`, `BUS_WAKEUP_Clock_Root`, and `BUS_AXI_Clock_Root`.

And there is an API which can get more `Clock_Root` frequency:

```
uint32_t CLOCK_GetFreqFromObs(uint32_t obsSigIndex, uint32_t obsIndex)
```

Take Audio PLL as an example:

```
CLOCK_GetFreqFromObs(CCM_OBS_PLL_AUDIO_OUT, 0);
```

Then the frequency of Audio PLL can be obtained by this API.

Note:

- *The clock observer uses 32 K as the reference clock. If the external 32 K is not used, the result from OBS is inaccurate.*
- *The OBS is only used for debug and it cannot be used by application.*
- *OBS has two indexes. To avoid problems, it is recommended to use one fixed index each for CM33 and CM7. For example, CM33 uses index 0 and CM7 always uses index 1.*

4.3 Chip enters system sleep mode or not?

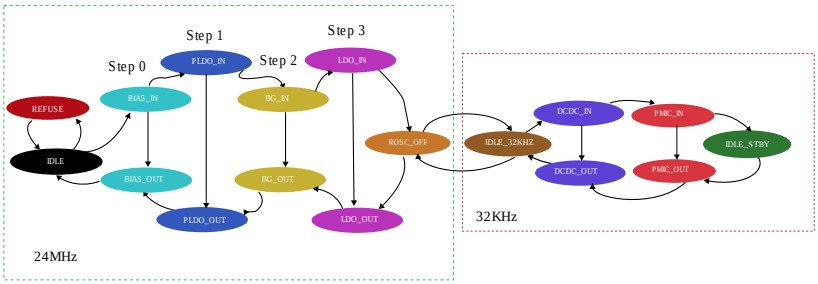

Figure 4. System sleep transition flow

If all CPUs are in low-power state and they all request system sleep mode, a system sleep sequence is triggered. There are two state machines in SSC. The first is in 24 MHz ROSC clock domain and the second is in 32 kHz clock domain. The first SSC state machine starts when CMC sends a system sleep request. It looks at all CMC system sleep status. If any CPU platform is not in sleep mode or does not allow system sleep, this system sleep request is refused. When this state machine transits to `ROSC_OFF` and there is no wake-up request, the second state machine starts. In the 24 MHz state machine, step 0 is for BIAS, step 1 for PLDO, step 2 for BANDGAP, and step 3 for LDO.

As shown in Figure 4, the last step in standby transition sequence is `PMIC_IN` in 32 kHz. It means if the PMIC enters system sleep, the whole sequence ends successfully. A `PMIC_STBY_REQ` pin indicates that the PMIC enters system sleep mode. When the system enters system sleep mode, PPC sends a signal to `PMIC_STBY_REQ`, then it outputs a high-level signal. So this PIN can be used to check whether the Chip is under system sleep mode or not.

4.4 Configure wake-up source

Configure the wake-up source before entering a low-power mode and executing WFI instruction. Following these steps, use an interrupt wake-up source to wake up chip under the low-power mode. Assuming that the handshake is completed and the peripheral is initialized, take GPIO and GPT as wake-up source as an example.

4.4.1 Configure GPIO as a wake-up source

```
/* Enable GPIO pin interrupt */
GPIO_EnableInterrupts(APP_WAKEUP_BUTTON_GPIO, 1U << APP_WAKEUP_BUTTON_GPIO_PIN);
/* Enable the Interrupt */
EnableIRQ(APP_WAKEUP_BUTTON_IRQ);
/* Mask all interrupt first */
GPC_DisableAllWakeupSource(GPC_CPU_MODE_CTRL);
/* Enable GPC interrupt */
GPC_EnableWakeupSource(APP_WAKEUP_BUTTON_IRQ);
```

Figure 5. Configure GPIO as a wake-up source

To configure GPIO as a wake-up source, perform the following steps:

1. Enable PIN interrupt.
2. Enable GPIO IRQ.
3. Disable all of wake-up sources registers in GPC.
4. Enable GPIO as a wake-up source in GPC.

4.4.2 Configure GPT timer as a wake-up source

```
/* Enable GPT Output Compare1 interrupt */
GPT_EnableInterrupts(EXAMPLE_GPT, kGPT_OutputCompare1InterruptEnable);

/* Enable at the Interrupt */
EnableIRQ(GPT_IRQ_ID);
GPC_DisableAllWakeupSource(GPC_CPU_MODE_CTRL);
/* Enable GPC interrupt */
GPC_EnableWakeupSource(GPT_IRQ_ID);
```

Figure 6. Configure GPT as a wake-up source

To configure GPIO timer as a wake-up source, perform the following steps:

1. Enable GPT Output Compare interrupt.
2. Enable GPT IRQ.
3. Disable all of wake-up source registers in GPC.
4. Enable GPT as a wake-up source in GPC.

4.5 Chip cannot enter a low-power mode

In most cases, CPU cannot enter a low-power mode because of a pending interrupt. There are a couple of methods to check whether there is a pending interrupt.

- `CM_IRQ_WAKEUP_STAT_x` in GPC register.

x means the index of CM_IRQ_WAKEUP_STAT register. There are eight registers for a GPC. That means 16 CM_IRQ_WAKEUP_STAT registers must be checked for dual-core silicon, and eight registers for single-core silicon.

- CM_NON_IRQ_WAKEUP_STAT in GPC register.
  When a debugger is connected to a silicon, it may generate a pending interrupt to block the chip from entering a low-power mode.

These registers can check whether there is a pending interrupt to block the system from entering a low-power mode. If there is a pending interrupt, perform the following steps:

1. Find which operation or peripheral generates this interrupt.
2. Disable and clean the status bit of this interrupt.

Another issue may block the system from entering a low-power mode is the handshake. If the polling method is used to check the handshake result, an uncompleted transmission may block the system.

4.6 Enter BBSM mode

BBSM mode powers down all domains except BBSM domain. BBSM domain is powered down by the external DCDC circuit. Both software and hardware are able to enter the BBSM mode.

- Software mode:

```
BBNSM->BBNSM_CTRL |= BBNSM_BBNSM_CTRL_TOSP_MASK;
```

- Hardware mode:
  Press the **ONOFF** button and hold for more than five seconds. The chip enters the BBSM mode.

4.7 Wake up from BBSM mode

To wake up the chip from BBSM mode, the peripheral which can run and generate an interrupt can be a wake-up source. For example, the WAKE UP pin which is a dedicated pin for waking up the system from BBSM mode. The WAKE UP pin generates an interrupt and wakes up the system with a high-voltage level. RTC can also be a wake-up source under BBSM mode.

The **ONOFF** button is another wake-up source to wake up chip from BBSM mode. To wake up the chip, press **ONOFF**.

4.8 Peripherals state under a low-power mode

Usually a peripheral needs two basic conditions to work normally, clock and power supply. These two conditions can help judge whether a peripheral can work under a low-power mode or not. The below lists simple check steps.

4.8.1 Clock

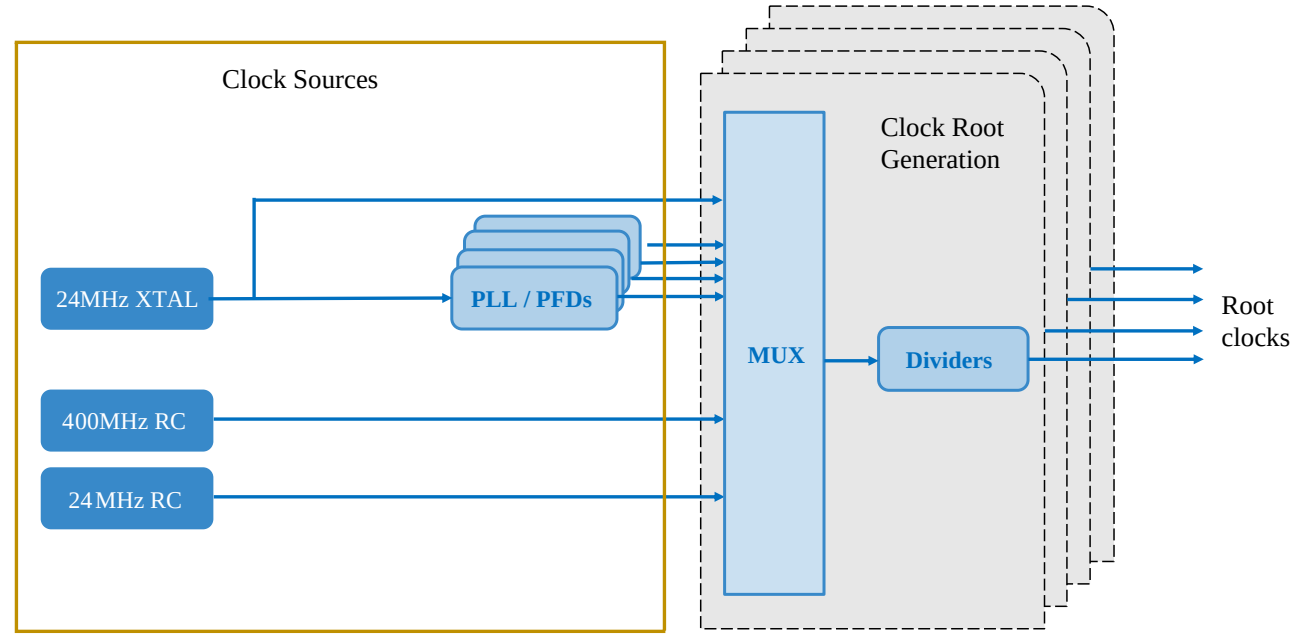

Figure 7. Clock sources and clock roots

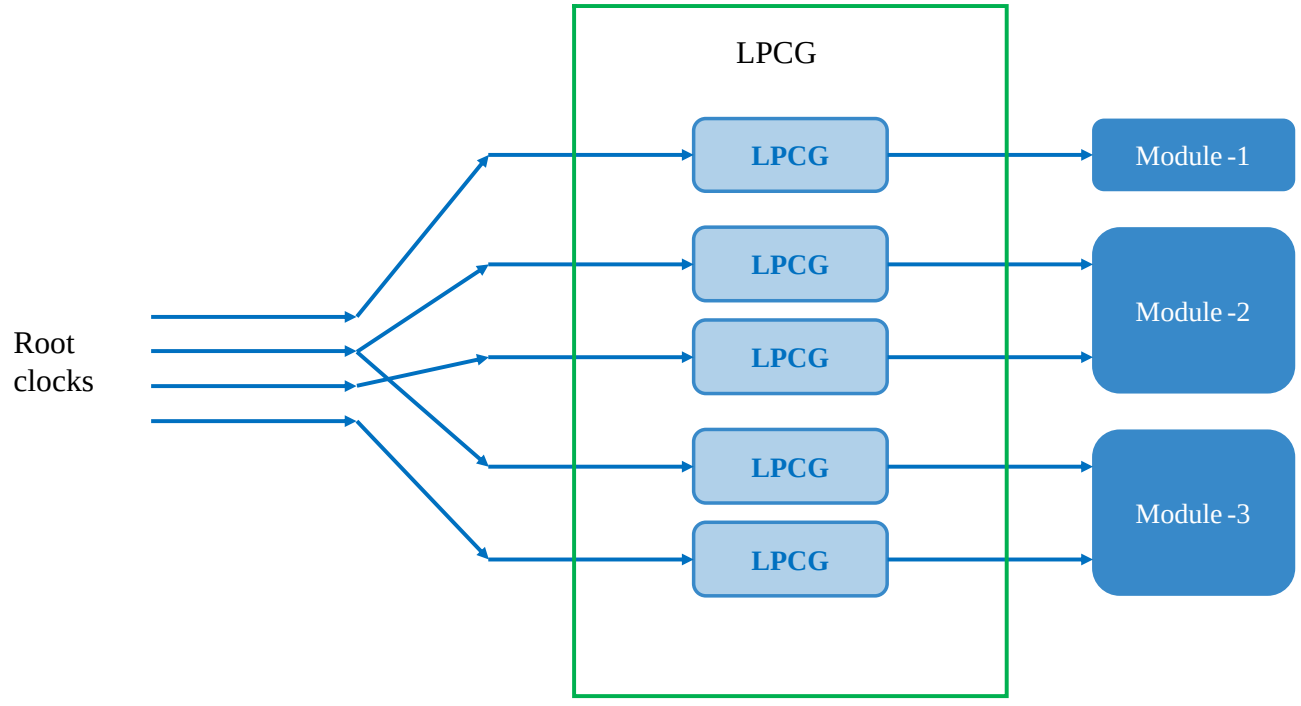

Figure 8. Clock root and LPCG

As shown in Figure 7 and Figure 8, the whole clock include three parts, clock source, clock root, and LPCG. To make sure that the clock for a peripheral is available, these three parts must be enabled and work at a suitable frequency.

4.8.1.1 Clock source

There are several clock sources in RT1180. These clock sources are controlled by CPU mode. The first step is to check whether the clock source is used for the peripheral enabled or not under the current CPU mode. If RT1180 must work under system sleep mode, shout down all clock sources to get a minimum power consumption data. If a peripheral is a wake-up source by an asynchronous interrupt, such as, LPUART, LPIIC, or LPSPI. For this case, RC24M can be alive.

Some peripherals, such as GPT, RTC, and WDOG, can use 32 K as the clock source. 32 K is an always-online clock source in CPU mode and even system sleep mode. It means that these peripherals can work under any power mode and even system sleep mode. The first step is to check whether the clock source is available or not.

4.8.1.2 Clock root

Clock root is used to select a clock source to a peripheral. All clock roots are controlled by the software. So based on step1, the second step is to enable the clock root, select a clock source for clock root, and set an appropriate frequency division factor.

4.8.1.3 LPCG

LPCG is a clock gate between clock root and a peripheral. For a peripheral which is used as a wake-up source, LPCG is always controlled by CPU power mode. The below is a list for each setting:

- The clock source is not needed in any mode, and can be turned off.
- The clock source is needed in RUN mode, but not needed in WAIT or STOP mode.
- The clock source is needed in RUN and WAIT mode, but not needed in STOP mode.
- The clock source is needed in RUN, WAIT, and STOP mode.
- The clock source is always on in any mode including SUSPEND.

The third step is to check the target CPU mode and LPCG settings.

4.8.2 Peripheral power supply

After checking the clock of a peripheral, also check the power supply. The peripheral in RT1180 belongs to different domain. It means that if a power domain or a power MIX is powered down, the peripheral under this MIX is also powered down.

4.9 Check the CPU mode

When developing a low-power case, check whether the CPU goes or went to a correct power mode.

`CPU0_CM_MODE_STAT` and `CPU1_CM_MODE_STAT` in GPC can be used to check the previous CPU mode and current CPU mode.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				Reserved				Reserved					Reserved	Reserved	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					Reserved	Reserved	Reserved	Reserved				CPU_MODE_PREVIO...		CPU_MODE_CURRENT	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

3-2 CPU_MODE_PREVIOUS	Previous CPU mode 00b - CPU was previously in RUN mode 01b - CPU was previously in WAIT mode 10b - CPU was previously in STOP mode 11b - CPU was previously in SUSPEND mode
1-0 CPU_MODE_CURRENT	Current CPU mode 00b - CPU is currently in RUN mode 01b - CPU is currently in WAIT mode 10b - CPU is currently in STOP mode 11b - CPU is currently in SUSPEND mode

Figure 9. CPU0_CM_MODE_STAT/CPU1_CM_MODE_STAT register

These registers can be accessed by both CM33 and CM7.

To get the previous and current CPU mode on M33, use the following code：

```
preCpuMode = GPC_CM_GetPreviousCpuMode(kGPC_CPU0);
curCpuMode = GPC_CM_GetCurrentCpuMode(kGPC_CPU0);
PRINTF("M33 previous CPU mode is %s\r\n", GET_CPU_MODE_NAME(preCpuMode));
PRINTF("M33 current CPU mode is %s\r\n", GET_CPU_MODE_NAME(curCpuMode));
```

To get the previous and current CPU mode on M7, use the following code：

```
preCpuMode = GPC_CM_GetPreviousCpuMode(kGPC_CPU1);
curCpuMode = GPC_CM_GetCurrentCpuMode(kGPC_CPU1);
PRINTF("M7 previous CPU mode is %s\r\n", GET_CPU_MODE_NAME(preCpuMode));
PRINTF("M7 current CPU mode is %s\r\n", GET_CPU_MODE_NAME(curCpuMode));
```

4.10 Jump to a specified address after reset

The chip wakes up from the Suspend mode through a reset. After the chip wakes up, the silicon jumps to a specified address. Both CM33 and CM7 support this function. The below takes CM7 as an example:

```
/*
    0x20000 is the vector table base address
        0x20100 is the SP after reset
  */
  *(uint32_t *)(0x20000) = 0x20100;

  /*
    The value in 0x20004 is the PC after reset
```

```
   */
     *(uint32_t *)(0x20004) = ((uint32_t)test);
     BLK_CTRL_S_AONMIX->M7_CFG &= ~BLK_CTRL_S_AONMIX_M7_CFG_INITVTOR_MASK;
     BLK_CTRL_S_AONMIX->M7_CFG |= BLK_CTRL_S_AONMIX_M7_CFG_INITVTOR(0x20000>>7);


void test()
{
  while(1)
  {
  SDK_DelayAtLeastUs(1000000U, SystemCoreClock);
  RGPIO_PortToggle(RGPIO4, 1u << 27);
  }
}
```

To achieve this function, three conditions are needed:

- Align the jump address with `0x80`. This requirement is determined by Arm core.
- Power on the memory or peripheral of the jump address or keep them in retention mode. The memory in this chapter means RAM. Once the memory is powered down, the data is lost. For the peripheral, it means FlexSPI. If the FlexSPI is powered down, the registers settings are lost. Once the chip wakes up, it fetches the instruction from the FlexSPI. But if the FlexSPI is under the uninitialized state, the chip generates a lockup reset and then re-boots from the image entry address.
- The address used in the example code must be an unused memory address.

5 Note about the source code in the document

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

Copyright 2024 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.

6 Revision history

Table 3 summarizes the revisions to this document.

Table 3. Revision history

Document ID	Release date	Description
AN13847 v.1	27 May 2024	Initial public release

Legal information

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.

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 https://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 document 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.

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.

Contents

For more information, please visit: https://www.nxp.com

Date of release: 27 May 2024
Document identifier: AN13847