Peripheral reinitialization after waking-up from Power Down mode for LPC55S69

The documentation is only valid for the LPC55xx and LPC55Sxx families. In power down mode, some of peripherals for LPC55xx are power off, which means that the peripherals lose it's power in power-down mode, so it is required to reinitialize the peripherals after waking-up from power down mode. The DOC lists the peripherals which lose power in power down mode and are required to initialize.

1. Power modes description

There are 4 low power modes, they are sleep mode, deep-sleep mode, power-down mode, and deep power down mode. For the sleep mode, deep-sleep mode, all peripherals are powered, so all the peripheral registers can retain their values, the peripheral external logic can retain also, for example GPIO output logic can keep it's original state in the sleep and deep sleep modes.

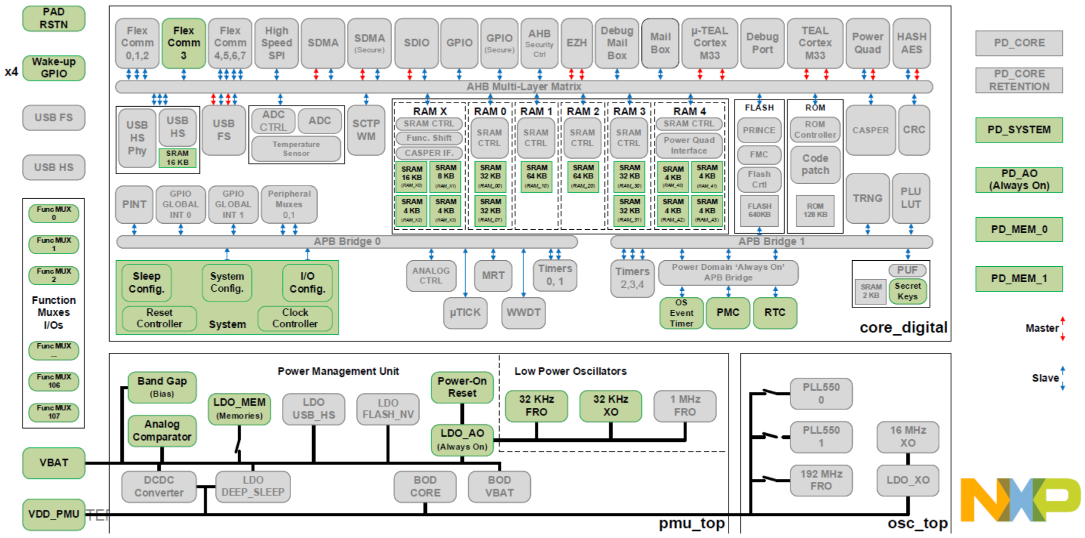
In power-down mode, some peripherals are power off, so the peripherals with power off lose it's register value, it's external pad logic can not keep either. For example, if a GPIO pin is set up as GPIO output in active mode, in power down and deep power down mode, the GPIO output logic can not retain, and become float, the LED will be dim during the power down and deep power down modes. After waking up from power down mode, the peripherals with power off have to be reinitialized, for example, the LED GPIO port has to reinitialized so that the LED can light after power down mode. Because the processor executes next instruction where it enters the power down mode, so the reinitialized code must locate at the next line of api function which enters power down mode.

2. Peripherals power state in power-down mode

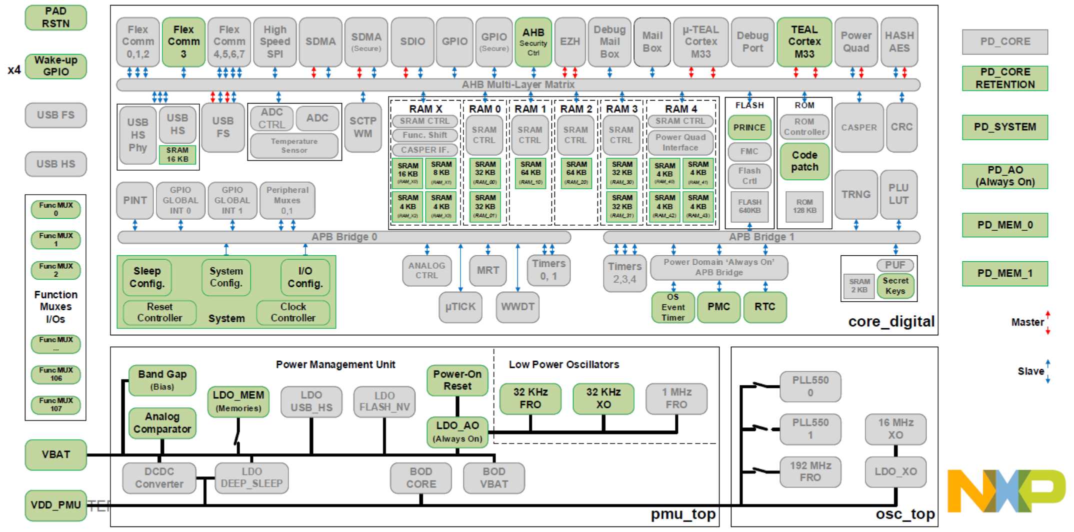
In deep power mode, most of peripherals and flash, SRAM are power off, so the peripherals also lose it's register values, but the processor goes through the entire reset process when a deep-power down wake-up event occur, so it is unnecessary to do any reinitialization.

The following two figures show the peripherals which retain or lose their power, the gray means power off, the green means power on in power down mode.

POWER-DOWN 2-1



POWER-DOWN 2-2 (with CPU State Retention)



From above list, we can see that FlexComm0/1/2/4/5/6/7, SPI,SDMA, SDIO, GPIO,USB-FS, USB-HS, ADC, SCT/PWM, PowerQuade, Hash-AES, CRC, TRNG,Casper, Ctimer, WWDT, MRT,uTick, Flash and ROM are power off. In other words, after waking up from power down modes, all above peripherals must be reinitiatilized.

But the RTC and OSEVENT Timer are power on, the System Config module(SYSCON), I/O Config(IOCON) modules are still powered on, it is unnecessary to reinitialize the RTC, OSEVENT Timer, SYSCON and IOCON modules after waking-up from power down mode.

3. POWER_EnterPowerDown() description

```
void POWER_EnterPowerDown(uint32_t exclude_from_pd,
                          uint32_t sram_retention_ctrl,
                          uint64_t wakeup_interrupts,
                          uint32_t cpu_retention_ctrl);
```

The POWER_EnterPowerDown() is called to enters power down mode, this is a ROM based function.

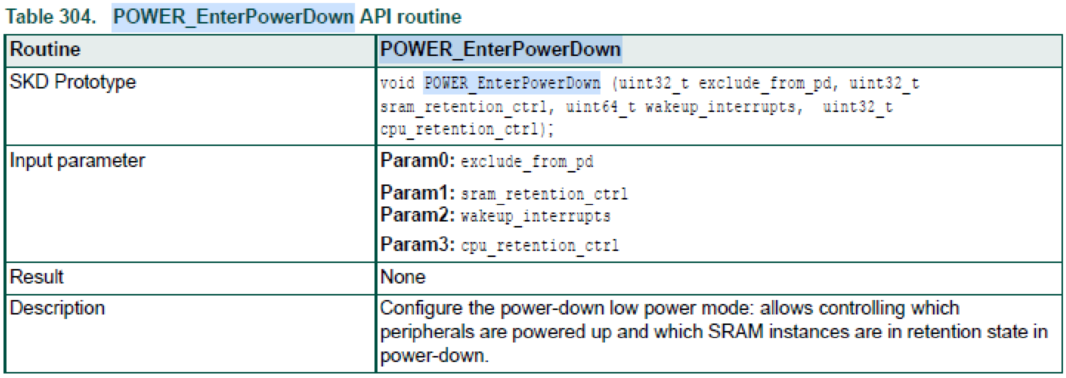
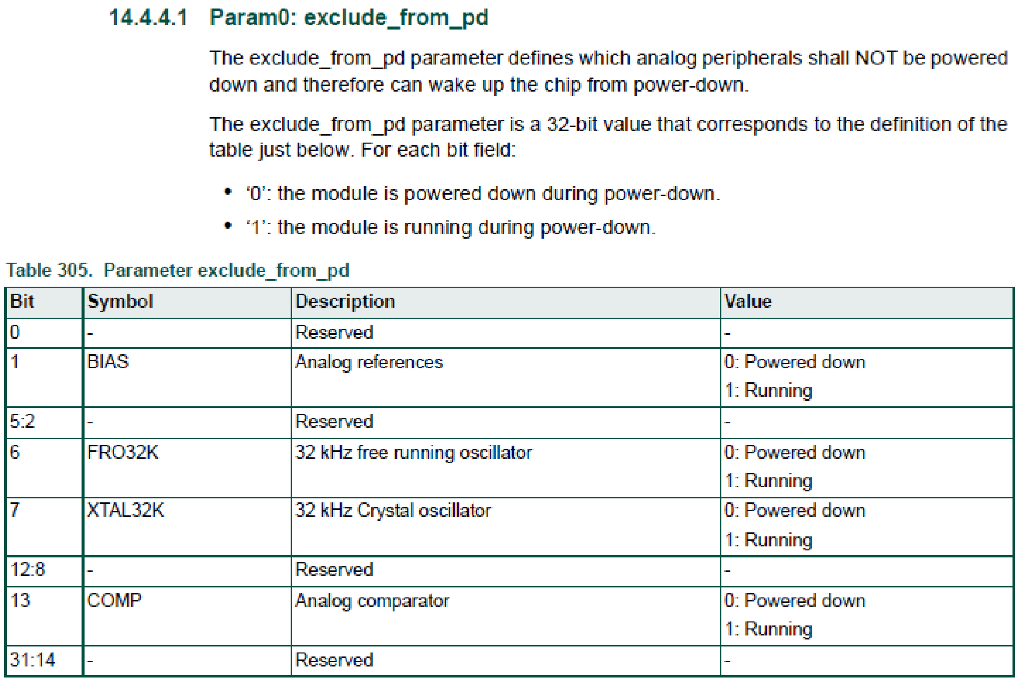


Table 304. POWER_EnterPowerDown API routine

Routine	POWER_EnterPowerDown
SKD Prototype	`void POWER_EnterPowerDown (uint32_t exclude_from_pd, uint32_t sram_retention_ctrl, uint64_t wakeup_interrupts, uint32_t cpu_retention_ctrl);`
Input parameter	**Param0:** `exclude_from_pd` **Param1:** `sram_retention_ctrl` **Param2:** `wakeup_interrupts` **Param3:** `cpu_retention_ctrl`
Result	None
Description	Configure the power-down low power mode: allows controlling which peripherals are powered up and which SRAM instances are in retention state in power-down.

For the 4 parameters, I list them one by one.

3.1 uint32_t exclude_from_pd



14.4.4.1 Param0: exclude_from_pd

The exclude_from_pd parameter defines which analog peripherals shall NOT be powered down and therefore can wake up the chip from power-down.

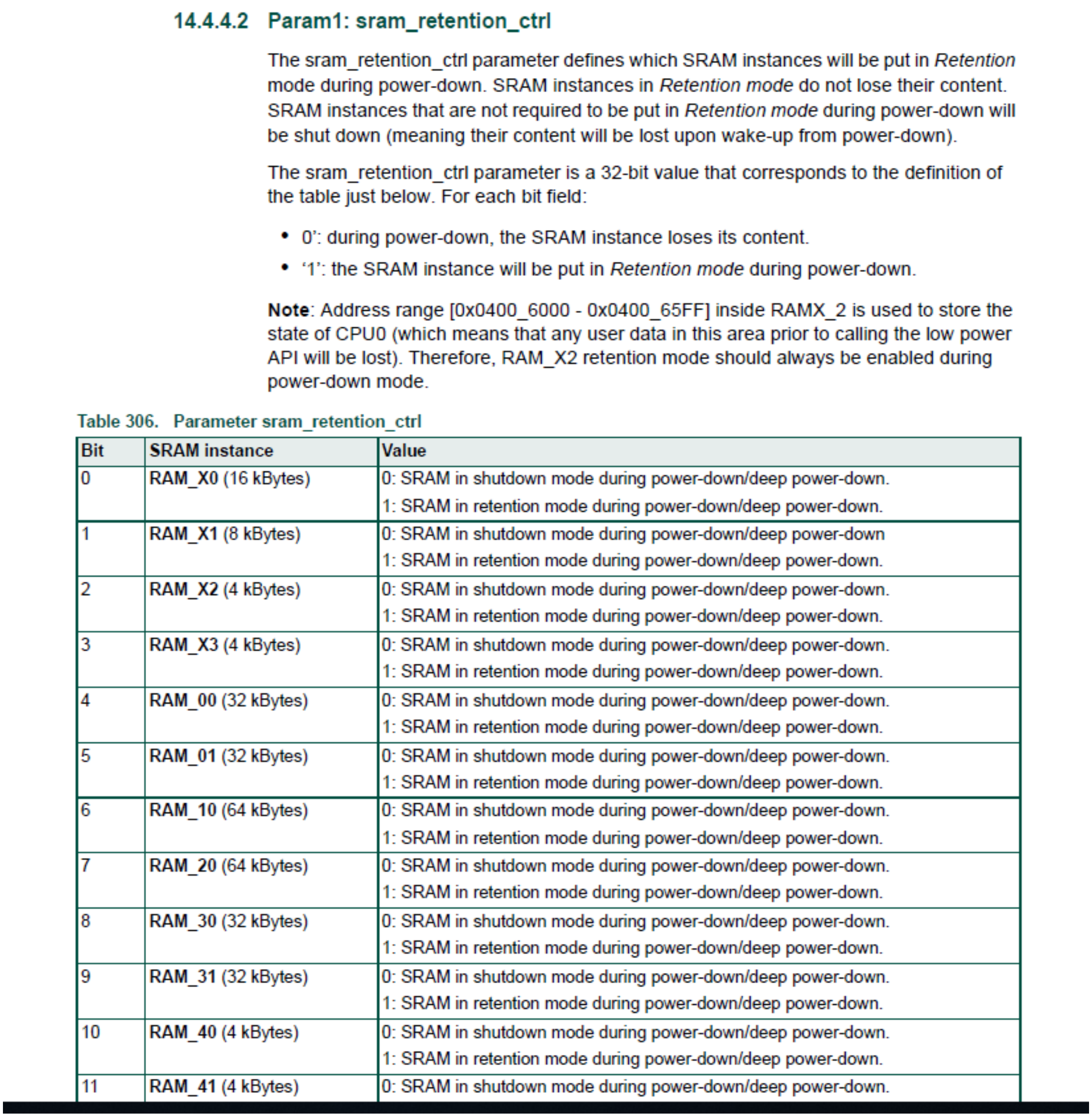
The exclude_from_pd parameter is a 32-bit value that corresponds to the definition of the table just below. For each bit field:

- '0': the module is powered down during power-down.
- '1': the module is running during power-down.

Table 305. Parameter exclude_from_pd

Bit	Symbol	Description	Value
0	-	Reserved	-
1	BIAS	Analog references	0: Powered down 1: Running
5:2	-	Reserved	-
6	FRO32K	32 kHz free running oscillator	0: Powered down 1: Running
7	XTAL32K	32 kHz Crystal oscillator	0: Powered down 1: Running
12:8	-	Reserved	-
13	COMP	Analog comparator	0: Powered down 1: Running
31:14	-	Reserved	-

3.2 uint32_t sram_retention_ctrl



14.4.4.2 Param1: sram_retention_ctrl

The sram_retention_ctrl parameter defines which SRAM instances will be put in *Retention* mode during power-down. SRAM instances in *Retention mode* do not lose their content. SRAM instances that are not required to be put in *Retention mode* during power-down will be shut down (meaning their content will be lost upon wake-up from power-down).

The sram_retention_ctrl parameter is a 32-bit value that corresponds to the definition of the table just below. For each bit field:

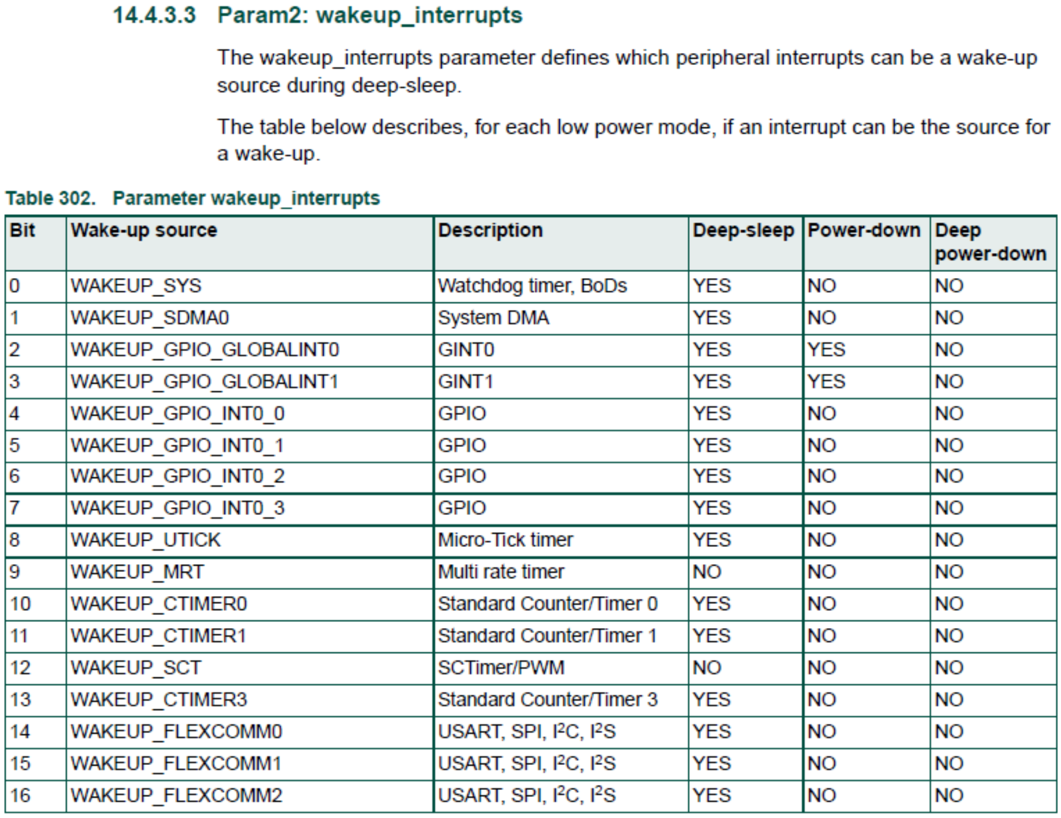
- 0’: during power-down, the SRAM instance loses its content.
- ‘1’: the SRAM instance will be put in *Retention mode* during power-down.

Note: Address range [0x0400_6000 - 0x0400_65FF] inside RAMX_2 is used to store the state of CPU0 (which means that any user data in this area prior to calling the low power API will be lost). Therefore, RAM_X2 retention mode should always be enabled during power-down mode.

Table 306. Parameter sram_retention_ctrl

Bit	SRAM instance	Value
0	**RAM_X0** (16 kBytes)	0: SRAM in shutdown mode during power-down/deep power-down. 1: SRAM in retention mode during power-down/deep power-down.
1	**RAM_X1** (8 kBytes)	0: SRAM in shutdown mode during power-down/deep power-down 1: SRAM in retention mode during power-down/deep power-down.
2	**RAM_X2** (4 kBytes)	0: SRAM in shutdown mode during power-down/deep power-down. 1: SRAM in retention mode during power-down/deep power-down.
3	**RAM_X3** (4 kBytes)	0: SRAM in shutdown mode during power-down/deep power-down. 1: SRAM in retention mode during power-down/deep power-down.
4	**RAM_00** (32 kBytes)	0: SRAM in shutdown mode during power-down/deep power-down. 1: SRAM in retention mode during power-down/deep power-down.
5	**RAM_01** (32 kBytes)	0: SRAM in shutdown mode during power-down/deep power-down. 1: SRAM in retention mode during power-down/deep power-down.
6	**RAM_10** (64 kBytes)	0: SRAM in shutdown mode during power-down/deep power-down. 1: SRAM in retention mode during power-down/deep power-down.
7	**RAM_20** (64 kBytes)	0: SRAM in shutdown mode during power-down/deep power-down. 1: SRAM in retention mode during power-down/deep power-down.
8	**RAM_30** (32 kBytes)	0: SRAM in shutdown mode during power-down/deep power-down. 1: SRAM in retention mode during power-down/deep power-down.
9	**RAM_31** (32 kBytes)	0: SRAM in shutdown mode during power-down/deep power-down. 1: SRAM in retention mode during power-down/deep power-down.
10	**RAM_40** (4 kBytes)	0: SRAM in shutdown mode during power-down/deep power-down. 1: SRAM in retention mode during power-down/deep power-down.
11	**RAM_41** (4 kBytes)	0: SRAM in shutdown mode during power-down/deep power-down.

3.3 uint64_t wakeup_interrupts description



14.4.3.3 Param2: wakeup_interrupts

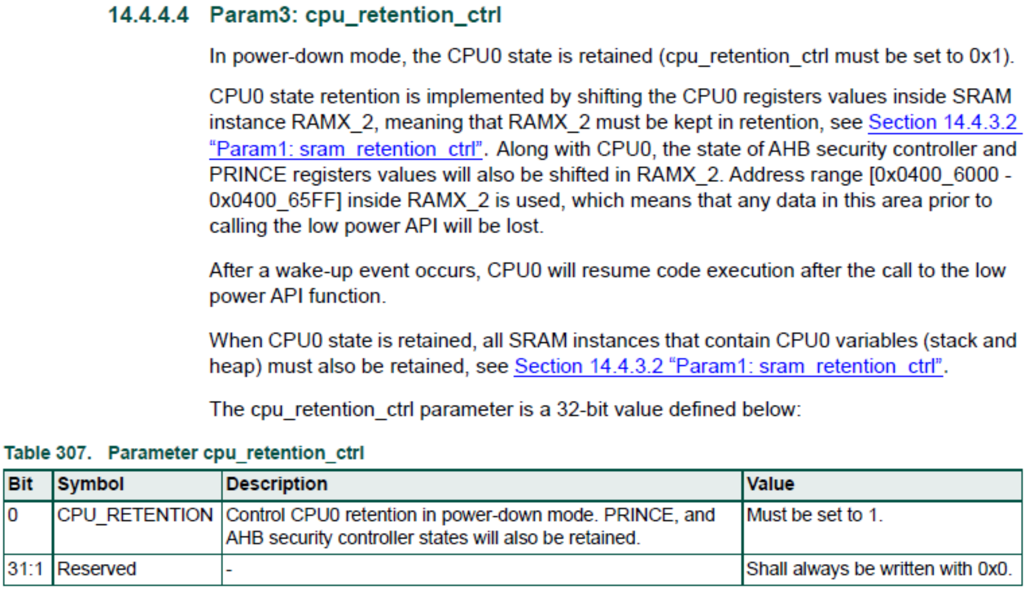
The wakeup_interrupts parameter defines which peripheral interrupts can be a wake-up source during deep-sleep.

The table below describes, for each low power mode, if an interrupt can be the source for a wake-up.

Table 302. Parameter wakeup_interrupts

Bit	Wake-up source	Description	Deep-sleep	Power-down	Deep power-down
0	WAKEUP_SYS	Watchdog timer, BoDs	YES	NO	NO
1	WAKEUP_SDMA0	System DMA	YES	NO	NO
2	WAKEUP_GPIO_GLOBALINT0	GINT0	YES	YES	NO
3	WAKEUP_GPIO_GLOBALINT1	GINT1	YES	YES	NO
4	WAKEUP_GPIO_INT0_0	GPIO	YES	NO	NO
5	WAKEUP_GPIO_INT0_1	GPIO	YES	NO	NO
6	WAKEUP_GPIO_INT0_2	GPIO	YES	NO	NO
7	WAKEUP_GPIO_INT0_3	GPIO	YES	NO	NO
8	WAKEUP_UTICK	Micro-Tick timer	YES	NO	NO
9	WAKEUP_MRT	Multi rate timer	NO	NO	NO
10	WAKEUP_CTIMER0	Standard Counter/Timer 0	YES	NO	NO
11	WAKEUP_CTIMER1	Standard Counter/Timer 1	YES	NO	NO
12	WAKEUP_SCT	SCTimer/PWM	NO	NO	NO
13	WAKEUP_CTIMER3	Standard Counter/Timer 3	YES	NO	NO
14	WAKEUP_FLEXCOMM0	USART, SPI, I^{2}C, I^{2}S	YES	NO	NO
15	WAKEUP_FLEXCOMM1	USART, SPI, I^{2}C, I^{2}S	YES	NO	NO
16	WAKEUP_FLEXCOMM2	USART, SPI, I^{2}C, I^{2}S	YES	NO	NO

3.4 uint32_t cpu_retention_ctrl description



14.4.4.4 Param3: cpu_retention_ctrl

In power-down mode, the CPU0 state is retained (cpu_retention_ctrl must be set to 0x1).

CPU0 state retention is implemented by shifting the CPU0 registers values inside SRAM instance RAMX_2, meaning that RAMX_2 must be kept in retention, see Section 14.4.3.2 "Param1: sram_retention_ctrl". Along with CPU0, the state of AHB security controller and PRINCE registers values will also be shifted in RAMX_2. Address range [0x0400_6000 - 0x0400_65FF] inside RAMX_2 is used, which means that any data in this area prior to calling the low power API will be lost.

After a wake-up event occurs, CPU0 will resume code execution after the call to the low power API function.

When CPU0 state is retained, all SRAM instances that contain CPU0 variables (stack and heap) must also be retained, see Section 14.4.3.2 "Param1: sram_retention_ctrl".

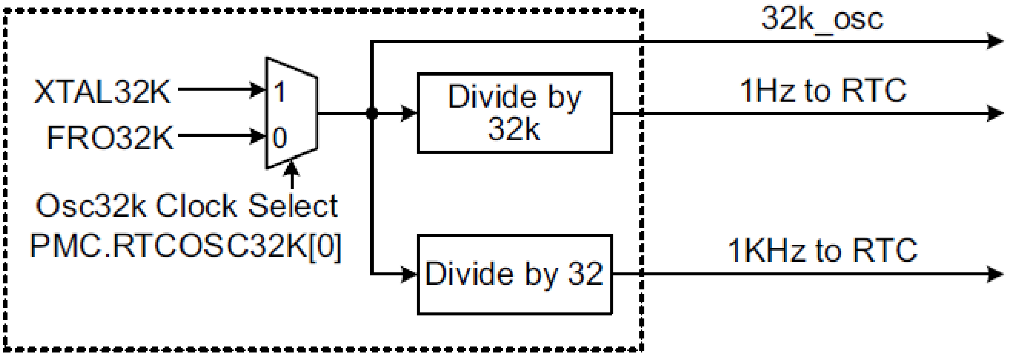
The cpu_retention_ctrl parameter is a 32-bit value defined below:

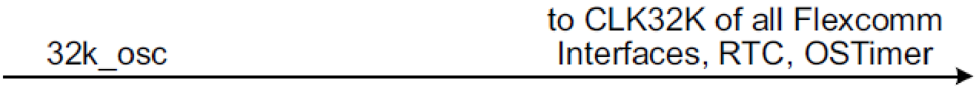
Table 307. Parameter cpu_retention_ctrl

Bit	Symbol	Description	Value
0	CPU_RETENTION	Control CPU0 retention in power-down mode. PRINCE, and AHB security controller states will also be retained.	Must be set to 1.
31:1	Reserved	-	Shall always be written with 0x0.

4. The preparation to enter power-down mode.

In power down mode, the main_clk is from 32K_osc clock node, in other words, the processor runs in 32KHZ. So the internal FRO32K clock source must be powered, and switch to 32K_osc clock node.





1) power on the internal FRO32K clock by clearing the PDEN_FRO32K bit in **PMC->PDRUNCFG0 register.**
2) 32k_osc node selects the FRO32K clock source by clearing the SEL bit in PMC->**RTCOSC32K,**
3) The main_clk node selects the 32k_osc as clock by configuring the SEL bits SYSCON->MAINCLKSELB register, in this way, after waking up, the core will run in 32KHz.

In power down mode, the PMC, RTC modules are powered by PD_AO(Always On) power domain, so they still function. The SYSCON and IOCON modules are powered by PD_SYSTEM power domain, which is switched to PD_AO power domain in power down mode, so they also function.

```
void DEMO_PreLowPower(void)
{
    /*!< Configure RTC OSC */
    POWER_EnablePD(kPDRUNCFG_PD_XTAL32K); /*!< Powered down the XTAL 32 kHz
RTC oscillator */
    POWER_DisablePD(kPDRUNCFG_PD_FRO32K); /*!< Powered the FRO 32 kHz RTC
oscillator */
    CLOCK_AttachClk(kFRO32K_to_OSC32K);   /*!< Switch OSC32K to FRO32K */

    CLOCK_SetFLASHAccessCyclesForFreq(32768U); /*!< Set FLASH wait states
for core */

    /*!< Set up dividers */
    CLOCK_SetClkDiv(kCLOCK_DivAhbClk, 1U, false); /*!< Set AHBCLKDIV
divider to value 1 */

    /*!< Set up clock selectors - Attach clocks to the peripheries */
    CLOCK_AttachClk(kOSC32K_to_MAIN_CLK); /*!< Switch MAIN_CLK to OSC32K */

    /*< Set SystemCoreClock variable. */
    SystemCoreClock = 32768U;
}
```

5. Reinitialization after Waking-up from power down mode

Waking-up sources in power-down mode:
Reset pin, Flexcomm3 module, analog comparator, GPIO group interrupt signals, RTC, OS Event Timer.
In the example of power_manager_lpc in SDK package, the GINT0 module is used to wake-up the processor from power-down mode.

The example uses Group GPIO Input Interrupt GINT0 module to wake-up from power-down mode, only PIO0_5 pin is used to generate GINT0 interrupt, in other words, in the example, only PIO0_5 can wake-up from the power down mode. But you can add the other PIO0 port pin to wake-up from power down mode.
During the power down mode, the GPIO pin driving LED is float due to the GPIO module power off, I hope the LED lights after waking up from power down mode, so I reinitialize the LED GPIO after the POWER_EnterPowerDown() function.
Note that the waking-up goes through interrupt process, in other words, the ISR of waking-up interrupt is executed. So the ISR has to be provided.
The skeleton of code to enter power down mode and waking-up from the low power mode.

```
Void main()
{
……………………………………………..
    APP_InitWakeupPin()
    While(1)
    {
        ……………………………….
        switch (gCurrentPowerMode)
        {
            case kPmu_PowerDown: /* Enter power down mode. */
                DEMO_PreLowPower();
                POWER_EnterPowerDown(APP_EXCLUDE_FROM_POWERDOWN, 0x7FFF,

                    WAKEUP_GPIO_GLOBALINT0| WAKEUP_GPIO_GLOBALINT1, 1);
                DEMO_PowerDownWakeup();
```

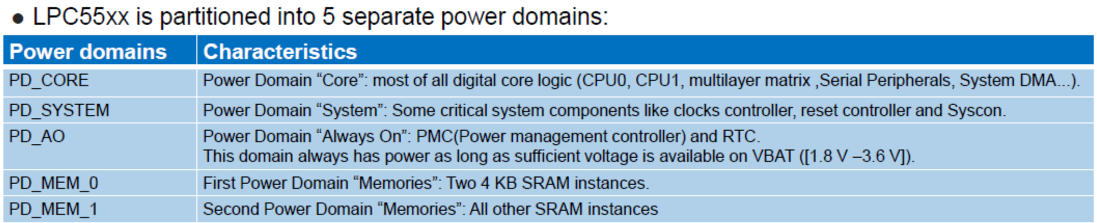
```
                    APP_InitWakeupPin();
                    Init_PIO1_7();
          }
     }
}
```

For example, the LED is connected to

```
static void APP_InitWakeupPin(void)
{

    /* Initialize GINT0*/
    GINT_Init(GINT0);

    /* Setup GINT0 for edge trigger, "OR" mode */
    GINT_SetCtrl(GINT0, kGINT_CombineOr, kGINT_TrigEdge, gint0_callback);

    /* Select pins & polarity for GINT0 */
    GINT_ConfigPins(GINT0, DEMO_GINT0_PORT, DEMO_GINT0_POL_MASK,
DEMO_GINT0_ENA_MASK);

    /* Enable callbacks for GINT0 & GINT1 */
    GINT_EnableCallback(GINT0);
}

//LEDG PIO1_7
void Init_PIO1_7(void)
{
    //enable gated GPIOP1 clock
    SYSCON->AHBCLKCTRL.AHBCLKCTRL0|=1<<15;
    //set the mux

    //set GPIO direction reg
    GPIO->DIR[1]|=1<<7;
    //toggle the PIO1_4
    GPIO->SET[1]|=1<<7;
    __asm("nop");
    GPIO->CLR[1]|=1<<7;  //clearing the bit will light the LED


}
```

6. Power domain diagram

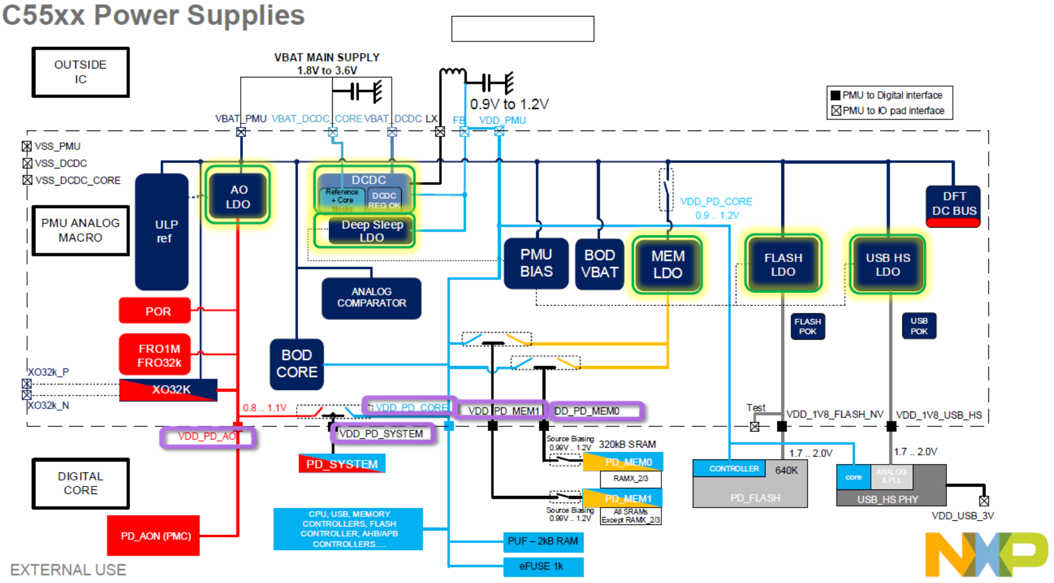
The LPC55xx family has 5 separate power domains as the following Fig.



- LPC55xx is partitioned into 5 separate power domains:

Power domains	Characteristics
PD_CORE	Power Domain “Core”: most of all digital core logic (CPU0, CPU1, multilayer matrix ,Serial Peripherals, System DMA…).
PD_SYSTEM	Power Domain “System”: Some critical system components like clocks controller, reset controller and Syscon.
PD_AO	Power Domain “Always On”: PMC(Power management controller) and RTC. This domain always has power as long as sufficient voltage is available on VBAT ([1.8 V –3.6 V]).
PD_MEM_0	First Power Domain “Memories”: Two 4 KB SRAM instances.
PD_MEM_1	Second Power Domain “Memories”: All other SRAM instances

LPC55xx Power Supplies



5 EXTERNAL USE