APPLICATION NOTE

AT03498: Low Power Modes in SAM3 Family

Atmel AT91 ARM Cortex Microcontrollers

Introduction

The Atmel® SAM3 family of microcontrollers is based on the ARM® Cortex®-M3 CPU architecture. Thanks to Cortex-M3 system sleep mode that can stop the Cortex-M3 and system clocks for greater power reductions, the SAM3 family features various low power modes that can be used to optimize power consumption in applications where low power is a key. This application note is for the Atmel SAM3U/3S/3N series and it is also applicable for other SAM3 series.

The purpose of this document is to introduce the part that is used for power supply and various low power modes of SAM3 family, how to enter and exit each low power mode, how to use these low power modes and give suggestions on how to optimize power consumption. This application note also provides a software example to demonstrate how to enter/exit low power modes, and how to measure the power consumption using the Atmel SAM3S-EK board.

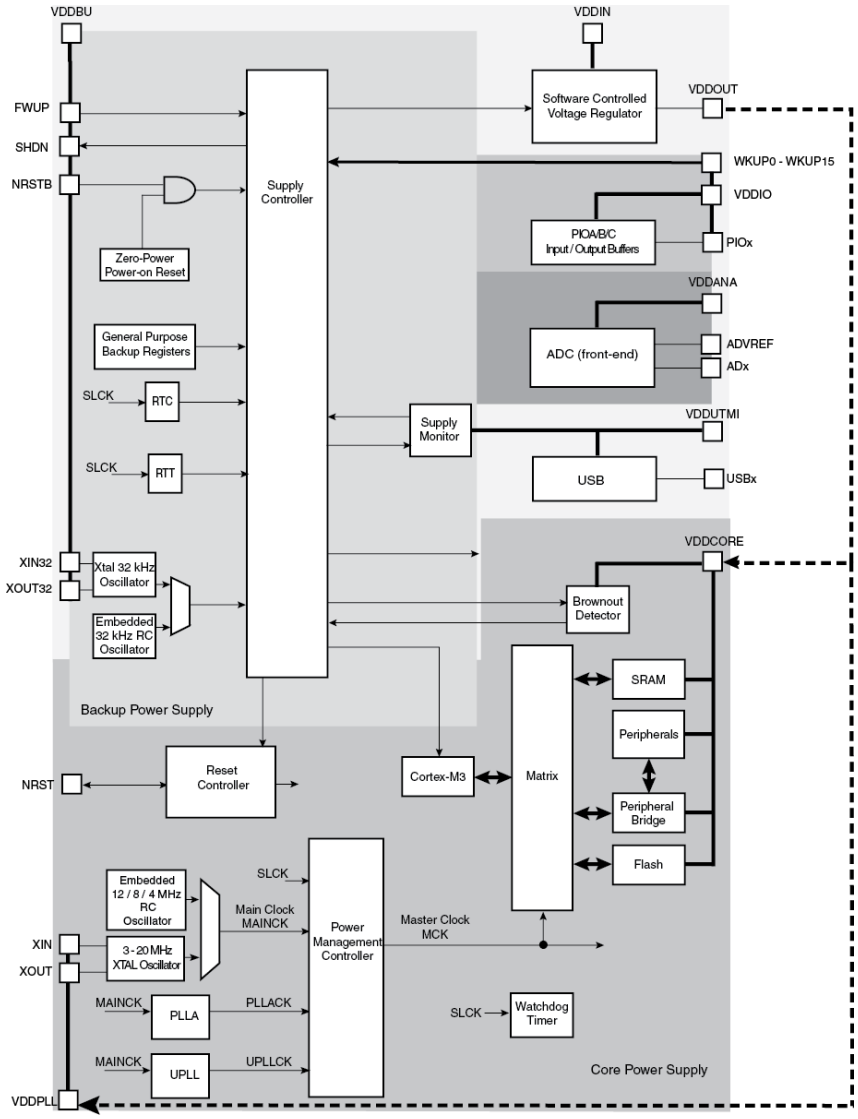
Table of Contents

1. Power Supply

1.1 Power Supply of SAM3U

Figure 1-1 gives an overview of the Supply Controller of the Atmel SAM3U.

Figure 1-1. Supply Controller of SAM3U.



The SAM3U product has several types of power supply pins:

VDDCORE: Powers the core, the embedded memories and the peripherals

VDDIO: Powers the peripherals I/O lines

VDDIN: Powers the voltage regulator

VDDOUT: It is the output of the voltage regulator; Intended to supply the core of the device

VDDBU: Powers the Slow Clock oscillator and a part of Controller

VDDPLL: Powers the PLL A, UPLL and 3MHz to 20MHz oscillator

VDDUTMI: Powers the USB UTMI + interface

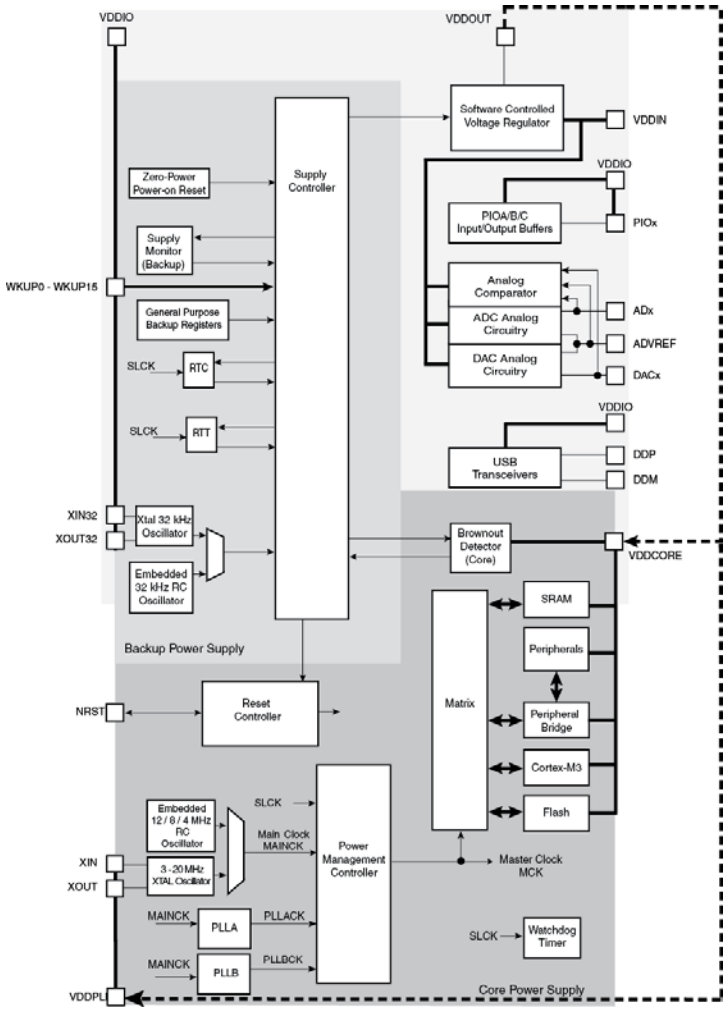
VDDANA: Powers the ADC cells

The device can be divided into two power supply areas: Backup Power Supply and Core Power Supply.

1.2 Power Supply of SAM3S

Figure 1-2 gives an overview of the Supply Controller of the Atmel SAM3S.

Figure 1-2. Supply Controller of SAM3S.



There are several types of power supply pins:

VDDCORE: Powers the core, the embedded memories and the peripherals

VDDIO: Powers the peripherals I/O lines (Input/Output Buffers); USB transceiver; Backup part, 32kHz crystal oscillator and oscillator pads

VDDIN: Voltage Regulator Input, ADC, DAC and Analog Comparator Power Supply

VDDOUT: It is the output of the voltage regulator; Intended to supply the core of the device

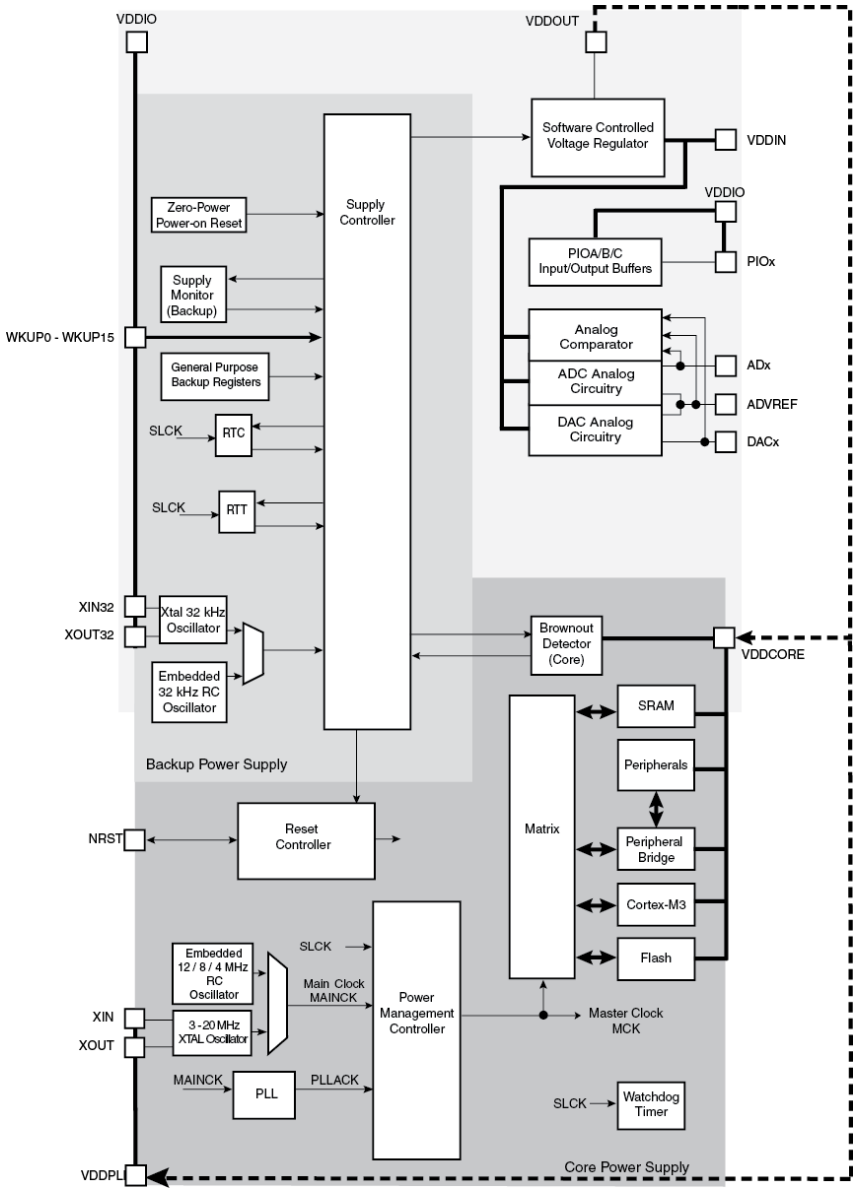
VDDPLL: Powers the PLLA, PLLB, the Fast RC and the 3MHz to 20MHz oscillator

The device can be divided into two power supply areas: Backup Power Supply and Core Power Supply.

1.3 Power Supply of SAM3N

Figure 1-3 gives an overview of the Supply Controller of the Atmel SAM3N.

Figure 1-3. Supply Controller of SAM3N.



There are several types of power supply pins:

VDDCORE: Powers the core, the embedded memories and the peripherals

VDDIO: Powers the peripherals I/O lines (Input/Output Buffers); USB transceiver; Backup part, 32kHz crystal oscillator and oscillator pads

VDDIN: Voltage Regulator Input, ADC, DAC and Analog Comparator Power Supply

VDDOUT: It is the output of the voltage regulator; Intended to supply the core of the device

VDDPLL: Powers the PLL, the Fast RC and the 3MHz to 20MHz oscillator

The device can be divided into two power supply areas: Backup Power Supply and Core Power Supply.

1.4 Supply Controller (SUPC)

The Supply Controller controls the supply voltage of the Core of the system and manages the Backup mode by controlling the Embedded Voltage Regulator.

The Supply Controller starts up the device by enabling the voltage regulator. Then it generates the proper reset signals to the core power supply.

It also enables to set the system in different low power modes and to wake it up from a wide range of events.

1.5 Voltage Regulator

The voltage regulator is managed by the Supply Controller. This internal regulator is intended to supply the core of the device but can be used to supply other parts in the application. It features two different operating modes:

- In Normal mode, the voltage regulator supplies power to core domain, internal adaptive biasing automatically adjusts the regulator quiescent current depending on the required load current. In Wait mode quiescent current is only 7µA
- In Shutdown mode, the voltage regulator is disabled and its output is driven internally to GND, it consumes less than 1µA. When system enters Backup mode, the SUPC disable voltage regulator automatically

In addition, when the user does not use the internal voltage regulator and wants to supply VDDCORE by an external supply, it is possible to disable the voltage regulator.

1.6 Supply Monitor

The supply monitor can be used to prevent the processor from falling into an unpredictable state if the Main power supply drops below a certain level. For the Atmel SAM3U series, it monitors VDDUTMI power supply, for SAM3S/3N series, it monitors VDDIO power supply.
The threshold of the supply monitor is programmable. It can be selected from 1.9V to 3.4V by steps of 100mV. This threshold is programmed in the SMTH field of the Supply Controller Supply Monitor Mode Register (SUPC_SMMR).

To reduce power consumption, Supply Monitor can be disabled by writing the SMSMPL field to 0 in SUPC_SMMR.

1.7 Brown-out Detector (BOD)

Brown-out detector monitors on VDDCORE. By default, it is enabled.

To reduce power consumption, Brown-out detector can be disabled by writing the BODDIS bit to 1 in SUPC_MR.

1.8 General Purpose Backup Registers (GPBR)

The device embeds four 32-bit general-purpose backup registers which can store 16 bytes of user application data. When system enter Backup mode, the voltage regulator is disabled, and VDDCORE is off but the backup registers remain powered by VDDIO (on Atmel SAM3S/3N) or VDDBU (on Atmel SAM3U).They retain the content when the system wakes up from Backup mode.

1.9 Power Management Controller (PMC)

The Power Management Controller (PMC) controls all system and peripheral clocks. It acts important role for optimizing power consumption.

1.10 Typical Powering Schematics

There are three typical powering schematics for the system:

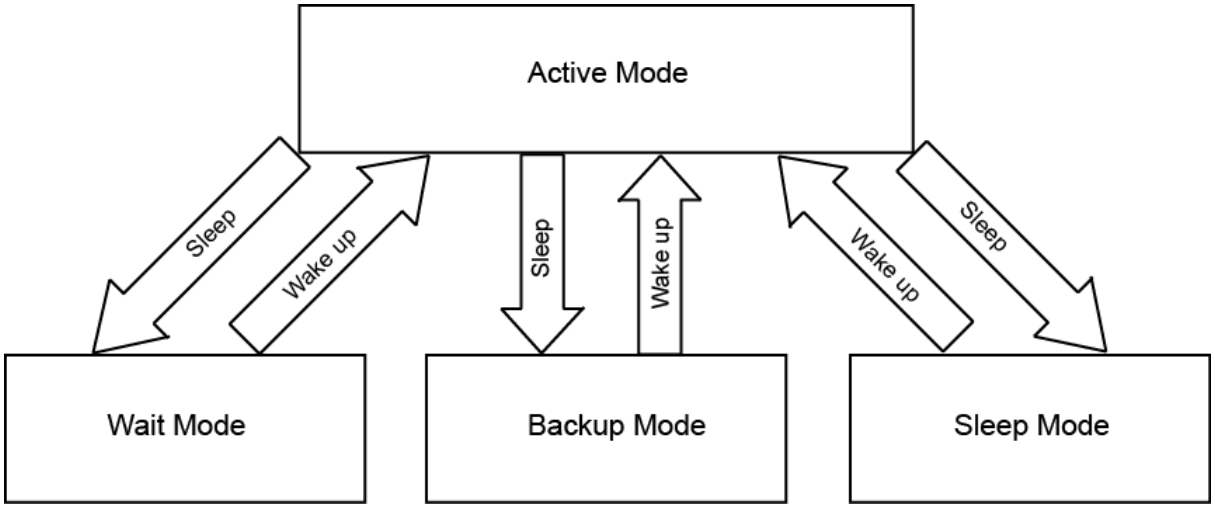
- Single Supply: The system is powered by single main supply. The internal regulator input connected to the main supply and its output feeds VDDCORE and VDDPLL
- Core Externally Supplied: The system is power by main supply while the VDDCORE and VDDPLL are powered by core supply
- Backup Batteries Used: The system is powered by main supply while the backup power supply part is powered by batteries

For further information, you can refer to Chapter “Power Considerations” of relevant device datasheet.

2. Various Power Modes

The Atmel SAM3 family of microcontrollers has various power modes, include Active mode and different low power modes (Sleep, Wait and Backup). Various power modes are shown in Figure 2-1. Each low power mode can be entered from Active mode and wake up by related enabled wake-up event. The various low power modes are described below.

Figure 2-1. SAM3 Various Power Modes.



2.1 Active Mode

Active mode is the normal running mode. By default, the microcontroller is in Active mode after power on. Following means can reduce power consumption in Active mode:

- Slow down the system clock
- Disable the peripheral clocks when they are unused

The power management controller (PMC) can be used to adapt the frequency and to disable the peripheral clocks.

2.2 Sleep Mode

In Sleep mode, the execution of instructions is suspended because the core clock is stopped while the peripheral clocks can be enabled. The purpose of sleep mode is to optimize power consumption of the device versus response time. The core needs lowest wakeup time in this mode.

2.2.1 Enter the Sleep Mode

Sleep mode is entered via Wait for Interrupt (WFI) or Wait for Event (WFE) instructions with LPM = 0 in PMC_FSMR. This mode is achieved by disabling the core clock by PMC. Following code snippet shows how to enter Sleep mode:

```
/**
 * \brief Enter Sleep Mode.
 * Enter condition: (WFE or WFI) + (SLEEPDEEP bit = 0) + (LPM bit = 0)
 *
 * \param type 0 - wait for interrupt, 1 - wait for event.
 */
void EnterSleepMode(uint8_t type)
{
    PMC->PMC_FSMR &= (uint32_t)~PMC_FSMR_LPM;
    SCB->SCR &= (uint32_t)~SCR_SLEEPDEEP;
    if (type == 0) {
        __WFI();
    } else {
        __WFE();
    }
}
```

When Sleep mode is entered, the current instruction is finished before the clock is stopped, but this does not prevent data transfers from other masters of the system bus.

2.2.2 Exit the Sleep Mode

Sleep mode is exited from an interrupt if WFI instruction of the Cortex-M3 is used, or from an event if the WFE instruction is used to enter this mode, or by the reset of the device. The PMC automatically re-enables the processor clock (PCK) and the core executes the next instruction of the program counter.

2.3 Wait Mode

The purpose of the wait mode is to achieve very low power consumption while maintaining the whole device in a powered state for a startup time of less than 10µs. In this mode, the clocks of the core, peripherals and memories are stopped. However, the core, peripherals and memories power supplies are still powered.

2.3.1 Enter the Wait Mode

Wait mode is entered via Wait for Event (WFE) instructions with LPM = 1 (Low Power Mode bit in PMC_FSMR). The Fast RC Oscillator is disabled by the PMC automatically when entering Wait mode. The step of entering Wait mode as below:

- Select the 4/8/12 MHz Fast RC Oscillator as Main Clock
- Set the LPM bit in the PMC Fast Startup Mode Register (PMC_FSMR)
- Execute the Wait-For-Event (WFE) instruction of the processor

Note: Internal Main clock resynchronization cycles are necessary between the writing of MOSCRCEN bit and the effective entry in Wait mode. Depending on the user application, waiting for MOSCRCEN bit to be cleared is recommended to ensure that the core will not execute undesired instructions.

In most application, the external XTAL oscillator is used instead of on-chip RC oscillator for accurate clock. Following code snippet shows how to switch MCK from external XTAL oscillator to Fast RC oscillator:

```
/**
 * \brief Switch MCK to FastRC (Main On-Chip RC Oscillator).
 *
 * \param moscrcf Main On-Chip RC Oscillator Frequency Selection.
 * \param pres    Processor Clock Prescaler.
 */
```

```
void SwitchMck2FastRC(uint32_t moscrcf, uint32_t pres)
{
    /* Enable Fast RC oscillator but DO NOT switch to RC now . Keep MOSCSEL to 1
*/
    PMC->CKGR_MOR = CKGR_MOR_MOSCSEL | (0x37 << 16)
        | CKGR_MOR_MOSCXTEN | CKGR_MOR_MOSCRCEN;
    /* Wait the Fast RC to stabilize */
    while (!(PMC->PMC_SR & PMC_SR_MOSCRCS));

    /* Switch from Main Xtal osc to Fast RC */
    PMC->CKGR_MOR = (0x37 << 16) | CKGR_MOR_MOSCRCEN | CKGR_MOR_MOSCXTEN ;
    /* Wait for Main Oscillator Selection Status bit MOSCSELS */
    while (!(PMC->PMC_SR & PMC_SR_MOSCSELS));

    /* Disable Main XTAL Oscillator */
    PMC->CKGR_MOR = (0x37 << 16) | CKGR_MOR_MOSCRCEN;

    /* Change frequency of Fast RC oscillator */
    PMC->CKGR_MOR = (0x37 << 16) | PMC->CKGR_MOR | moscrcf;
    /* Wait the Fast RC to stabilize */
    while (!(PMC->PMC_SR & PMC_SR_MOSCRCS));

    /* Switch to main clock */
    PMC->PMC_MCKR = (PMC->PMC_MCKR & (uint32_t)~PMC_MCKR_CSS)
        | PMC_MCKR_CSS_MAIN_CLK;
    while (!(PMC->PMC_SR & PMC_SR_MCKRDY));
    PMC->PMC_MCKR = (PMC->PMC_MCKR & (uint32_t)~PMC_MCKR_PRES)
        | pres;
    while (!(PMC->PMC_SR & PMC_SR_MCKRDY));

    /* Stop PLL A */
    /* STMODE must be set at 2 when the PLLA is Off */
    PMC->CKGR_PLLAR = 0x2 << 14;

    /* Stop PLLB */
    /* STMODE must be set at 2 when the PLLB is OFF */
    PMC->CKGR_PLLBR = 0x2 << 14;
}
```

Following code snippet shows how to enter Wait mode:

```
/**
 * \brief Enter Wait Mode.
 * Enter condition: WFE + (SLEEPDEEP bit = 0) + (LPM bit = 1)
 */
void EnterWaitMode(void)
{
    uint32_t i;

    PMC->PMC_FSMR |= PMC_FSMR_LPM;
    SCB->SCR &= (uint32_t)~SCR_SLEEPDEEP;
    __WFE();

    /* Waiting for MOSCRCEN bit is cleared is strongly recommended
     * to ensure that the core will not execute undesired instructions
     */
    for (i = 0; i < 500; i++) {
        __NOP();
    }
    while (!(PMC->CKGR_MOR & CKGR_MOR_MOSCRCEN));
}
```

2.3.2 Exit the Wait Mode

Wait mode can be exited from several asynchronous fast start-up sources that have to be programmed prior to entering this mode, include:

- WKUP0-15 pins
- RTT or RTC alarm
- USB Wake-up

As soon as the fast start-up signal is asserted, the PMC automatically restarts the embedded 4/8/12 MHz fast RC oscillator, switches the master clock on this 4/8/12 MHz clock and re-enables the processor clock and then the core executes the next instruction of the program counter.

2.4 Backup Mode

The purpose of backup mode is to achieve the lowest power consumption in a system but not requiring fast startup time (<0.5ms). Backup mode is based on the Cortex-M3 deep-sleep mode with the voltage regulator disabled (no power supply for VDDCORE and VDDPLL). The core power supply domain is powered off and the SRAM, flash memory, PLL and peripherals are also switched off. The backup power supply domain is still powered so the Supply Controller, zero-power power-on reset, RTT, RTC, Backup registers and 32kHz oscillator are running.

2.4.1 Enter the Backup Mode

Backup mode is entered by using WFE instructions with the SLEEPDEEP bit in the System Control Register of the Cortex-M3 set to 1. The regulator and the core supply are off.

Following code snippet shows how to enter Backup mode:

```
/**
 * \brief Enter Backup Mode.
 * Enter condition: WFE + (SLEEPDEEP bit = 1)
 */
void EnterBackupMode(void)
{
    SCB->SCR |= SCR_SLEEPDEEP;
    __WFE();
}
```

2.4.2 Exit the Backup Mode

Backup mode can be exited from several wake-up sources that must first be programmed prior to entering this mode, includes:

- WKUPEN0-15 pins (level transition, configurable debouncing)
- RTC alarm
- RTT alarm
- Supply Monitor alarm (on Atmel SAM3S series)
- BOD alarm (on Atmel SAM3U series)
- FWUP pin (on SAM3U series)

When waking up from Backup mode, the program execution restarts in the same way as system startup except backup region (RTC, RTT, GPBR and Supply Controller) are not reset.

2.5 Low Power Mode Summary

Table 2-1 shows a summary of the configurations of the low power modes.

Table 2-1. Low Power Mode Configuration Summary.

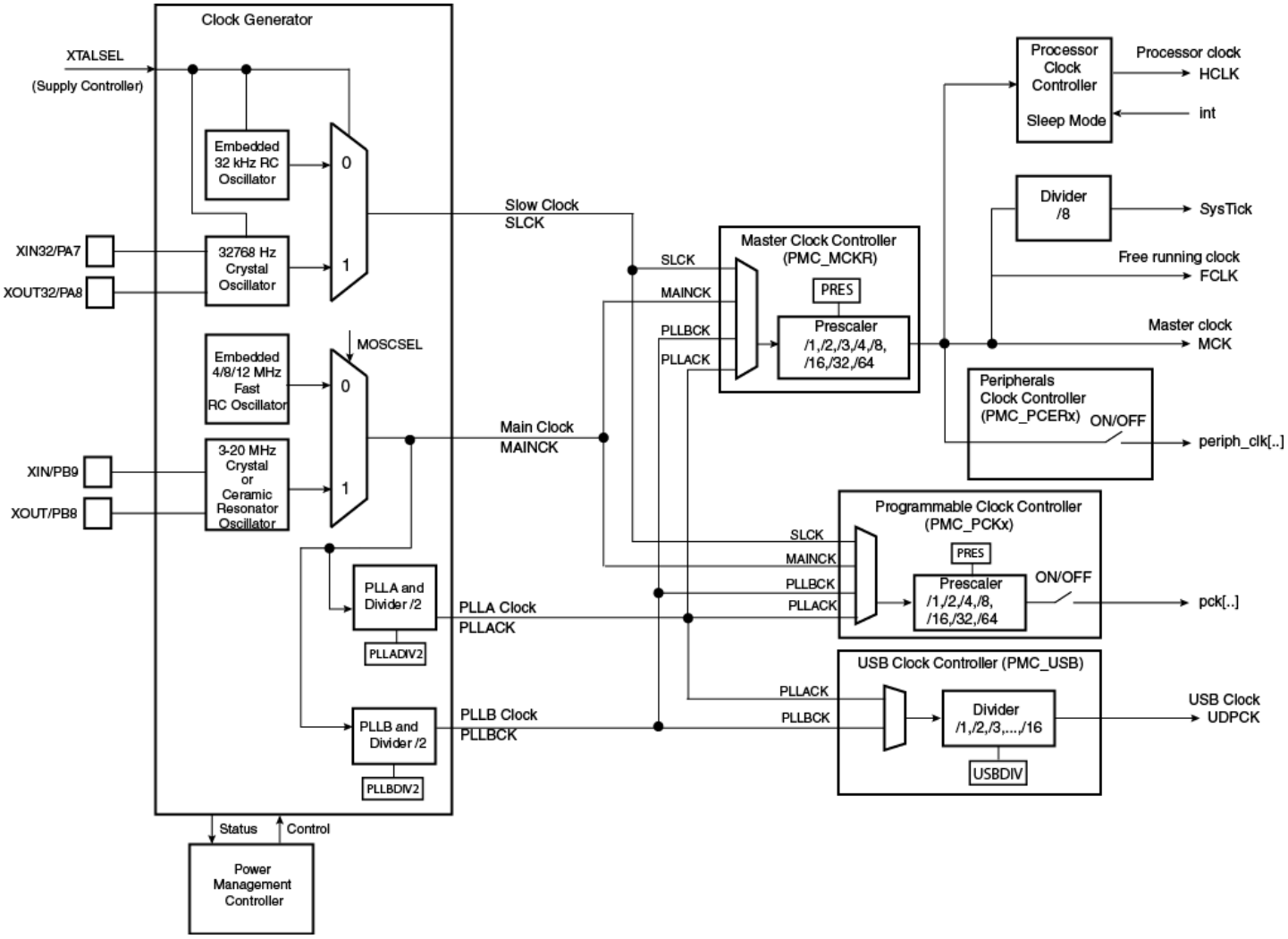
Mode	SUPC, 32kHz oscillator RTC RTT backup registers, POR (backup region)	Voltage regulator	Core memory peripherals	Mode entry	Potential wake up sources	Core at wake up
Backup mode	ON	OFF SHDN =0	OFF (Not powered)	WFE +SLEEPDEEP bit = 1	FWUP pin (on SAM3U) SM alarm (on SAM3S) WUP0-15 pins BOD alarm RTC alarm RTT alarm	Reset
Wait mode	ON	ON SHDN =1	Powered (Not clocked)	WFE +SLEEPDEEP bit = 0 +LPM bit = 1	Any Event from: Fast startup through WUP0-15 pins RTC alarm RTT alarm USB wake-up	Clocked back
Sleep mode	ON	ON SHDN =1	Powered (Not clocked)	WFE or WFI +SLEEPDEEP bit = 0 +LPM bit = 0	Entry mode =WFI Interrupt Only; Entry mode =WFE Any Enabled Interrupt and/or Any Event from: Fast start-up through WUP0-15 pins RTC alarm RTT alarm USB wake-up	Clocked back

3. Using Low Power Modes

3.1 Optimizing Power Consumption

Normally, the system power consumption increases with the clock frequency. There is a trade-off between power consumption and system performance. The Atmel SAM3 family of microcontrollers embeds a Power Management Controller (PMC) for optimizing power consumption by controlling all system and user peripheral clocks.

Figure 3-1. General Clock Block Diagram of SAM3S.



Normally, the system needs not to run at maximum frequency. According to system performance requirement, proper Master Clock (MCK) needs to be found for optimizing power consumption. The user can choose one of the following clocks as master clock source by writing the relevant value in the CSS field of the PMC_MCKR register:

- Slow clock
- Main clock
- PLLA clock
- PLLB clock on Atmel SAM3S / UPLLCK on Atmel SAM3U

And the Master Clock (MCK) can be divided in the range of 1 to 64 by writing the relevant value in the PRES field of the PMC_MCKR register.

The peripheral clocks are automatically disabled after a reset. The user can individually enable and disable the Master Clock on the peripherals by writing into the Peripheral Clock Enable (PMC_PCER) and Peripheral Clock Disable (PMC_PCDR) registers. In order to reduce power consumption, it is recommended to disable the clock of unused peripherals.

If PLL clock is not used, it is recommended to disable it for power saving.

The Brownout Detector monitors VDDCORE. It is enabled by default. It is especially recommended to disable it during low power modes for saving power consumption. It can be deactivated by software through the Supply Controller (SUPC_MR).

When the core is not required for executing instruction, the low power modes can be used to further reduce power consumption. Following sections have more discussion on this.

3.2 Sleep Mode vs. Polling Loops

In most applications, there are some polling loops for waiting for operation finish. For example, you need to send a large amount of data via USART; the pseudo code might like this:

```
SendDataViaUSART()
{
    Begin to send data via USART by PDC;
    Wait for transfer finish;
    Continue to execute next operation;
}
```

There are two ways for waiting for transfer finish:

- Polling loops, like while(transfer not finish) {1}
- Enter sleep mode and set the core be woken up by an interrupt when transfer finish.

By using polling loops, the instructions are still executed thereby increasing power consumption.

By using Sleep mode, the core clock is stopped thereby reducing power consumption while USART peripheral clock is still working, so data transfer continue. When the transfer finish, an interrupt will wake up the core immediately.

So in the case of the core is not running while waiting for peripherals transfer data done (with DMA or PDC), it is recommended to use the Sleep mode instead of polling loops.

3.3 Sleep Mode vs. Wait Mode

Some applications are not running all the time. For example, an alarm application waits for signal from a sensor to wake up the microcontroller to launch a task. In this case, the microcontroller can be suspended during idle time and woken up in time when specific event happens. Both Sleep mode and Wait mode may be used for reducing power consumption. Some main factors should be considered:

- wakeup source
- power consumption
- wakeup time

If the wakeup source is an interrupt in the application, only Sleep mode can be used because Wait mode can only be woken up by programmed events. Refer Table 2-1 for the different wakeup sources for Sleep mode and Wait mode.

For power consumption, in Wait mode, current consumption is typically 15µA when the internal voltage regulator is used. In Sleep mode, current consumption depends on master clock and peripherals in used. The current consumption in Sleep mode is as low as Wait mode only when the master clock is running at 500Hz with all peripherals clock off. The higher master clock frequency the higher power consumption in Sleep mode is.

The microcontroller can be woken up in several microseconds in both Sleep and Wait mode. When waking up from Sleep mode, the core clock is re-enabled by PMC automatically. But In Wait mode, it needs more time. When waking up from Wait mode, the PMC automatically restarts the embedded 4/8/12 MHz fast RC oscillator and then switches the master clock on this clock. Normally, user needs to switch master clock to PLL clock to get higher frequency.

Note: It is necessary to switch master clock to 4/8/12 MHz fast RC oscillator before entering Wait mode.

In a short, the trade-off between wakeup time and power consumption must be taken into account when both Sleep mode and Wait mode can be used in the application. Generally, Wait mode can save more power than Sleep mode but Sleep mode has faster startup time than Wait mode.

3.4 Using Backup Mode

In some applications, very low power consumption is necessary. In this case, the Backup mode can be considered. In Backup mode, timer can keep running and some important data can be stored in GPBR. Make sure that the Backup Supply domain is still powered when entering Backup mode.

4. Low Power Example

A “lowpower” example in the Atmel SAM3S Software Package demonstrates how to enter/exit low power modes and how to measure the power consumption on each power modes using the Atmel SAM3S-EK board. The example code can be downloaded from the Atmel website: http://www.atmel.com/dyn/products/tools_card_mcu.asp?tool_id=4705

4.1 Hardware Setup

This example runs on SAM3S-EK board:

- An RS-232 serial cable must be connected between UART(J7) and PC serial port
- To measure current consumption, an ampere meter has to be plugged instead of the VIN jumper

4.2 Running the Example

Follow the steps to run the example:

- Build the program and download it into internal flash
- Open and configure a terminal application on PC:
  - 115200 baud
  - 8 bits of data
  - No parity
  - 1 stop bit
  - No flow control
- Start the example, a menu will show on the terminal windows:

```
=============================================================
Menu: press a key to change the configuration or select a mode.
=============================================================
Config:
  f : Clock      = PCK=24 MCK=24
Mode:
  A : Active Mode
  S : Sleep Mode
  W : Wait Mode
  B : Backup Mode(Had entered 0 times).
Quit:
  Q : Quit test.
-------------------------------------------------------------
```

This menu is divided into two parts:

- First part allows the user to change the master clock
- Second part allows the user to enter different power modes for measuring power consumption

Select one of the options in the menu to perform the corresponding action.

5. References

[1]. Atmel SAM3U/3S/3N datasheets, available at:
http://www.atmel.com/dyn/products/datasheets_mcu.asp?family_id=605#2127

[2]. Cortex-M3 Technical Reference Manual, available at:
http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0337g/index.html

6. Revision History

Doc. Rev.	Date	Comments
42160A	07/2013	Initial document release

Atmel Corporation
1600 Technology Drive
San Jose, CA 95110
USA
Tel: (+1)(408) 441-0311
Fax: (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan G.K.
16F Shin-Osaki Kangyo Building
1-6-4 Osaki, Shinagawa-ku
Tokyo 141-0032
JAPAN
Tel: (+81)(3) 6417-0300
Fax: (+81)(3) 6417-0370