SMART ARM-based Microcontrollers

AT06549: Ultra Low Power Techniques

APPLICATION NOTE

Introduction

This application note will present techniques for SAM D and SAM L series on how to optimize for low power applications. It covers techniques such as sleep modes and SleepWalking in combination with using the Event System and Direct Memory Access Controller. The techniques presented in this application note will cover both system hardware features as well as peripheral features.

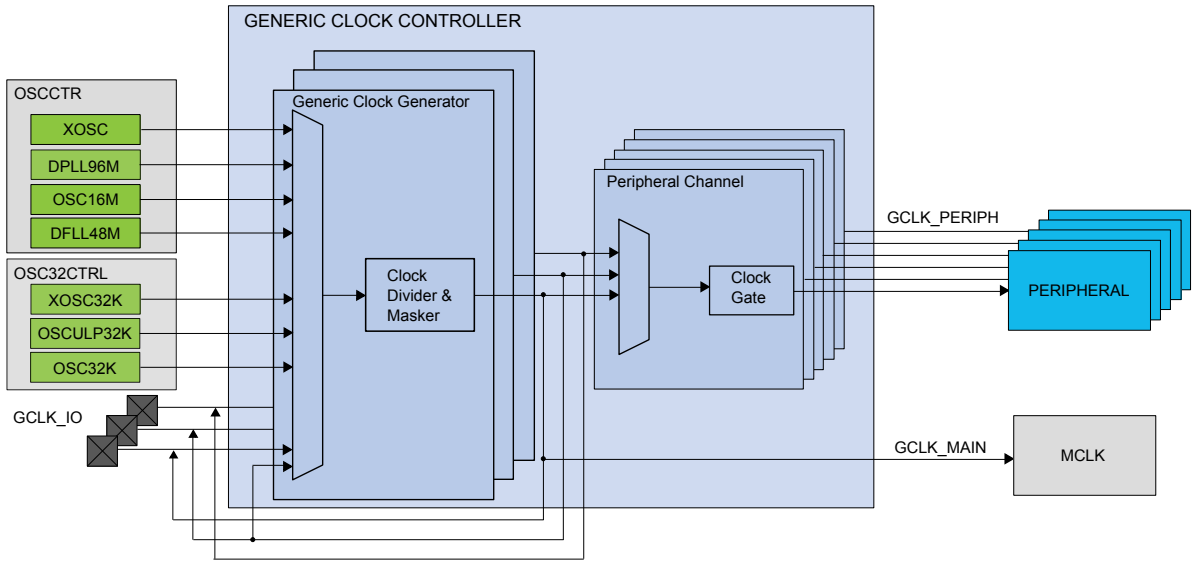
Table of Contents

1. Clock Sources and Clock Generators

In order to reduce the power consumption for an application, all unused clock sources should be disabled. Peripherals often require to run on different frequencies in an application, which often results in different clock sources being used for the different peripherals. As the number of enabled clock sources increases, as will the power consumption.

Devices from the SAM D and SAM L series have a generic clock controller (GCLK) which contain several generic clock generators, refer to Figure 1-1 Device Clocking Diagram on page 3 for GCLK implementation in SAM L series. One clock source can source several generic clock generators, and each generator can divide the clock source separately in order to get different frequencies. Choosing this solution, one can achieve several frequencies without the penalty of several clock sources being enabled. In addition, the clock system on devices from the SAM D and SAM L series allows for the clock source to be automatically turned off when not requested by a peripheral. Achieving this functionality is done by writing a one to the On Demand Control (ONDEMAND) bit for the corresponding clock source.

Figure 1-1 Device Clocking Diagram



2. Sleep Modes

In many applications, the processor does not run continuously and peripherals may be idle much of the time. For instance the application might be waiting for some external trigger before it starts using the processor and peripherals. By taking advantage of the various sleep modes available for devices from the SAM D and SAM L series, one can reduce the power consumption significantly by enabling the device to sleep while waiting.

Devices in the SAM D series has two sleep modes, IDLE and STANDBY, where the Idle mode is divided into three levels; IDLE0, IDLE1, and IDLE2. In the Idle sleep mode the CPU clock is stopped and dependent on the Idle level, additional synchronous clocks are stopped. In STANDBY all clocks are stopped, except for those who are configured to run in standby. For further information about the sleep modes for devices in the SAM D series, refer to their datasheets.

Devices in the SAM L series have four sleep modes; IDLE, STANDBY, BACKUP, and OFF. In IDLE only the CPU clock is stopped. In STANDBY the CPU clock as well as the main clock (MCLK) is stopped, and both regulators and SRAM are in low power mode. The peripheral clocks (GCLK) are only run if requested by the peripheral. In BACKUP all clocks are stopped and SRAM will not be able to retain any data. In OFF the entire device is powered off, and it can only be woken up by an external or power on reset. For further information about the sleep modes in for devices in the SAM L series, refer to their datasheets.

Sleep is entered by issuing the Wait For Interrupt (WFI) instruction. Before entering sleep, the sleep mode must be chosen. For devices in the SAM D series the sleep mode is configured by writing the Sleep Deep bit in the ARM® Cortex® System Control register (SCR.SLEEPDEEP). If SCR.SLEEPDEEP is configured to Idle sleep mode, the Idle Mode Configuration bit group in the Sleep Mode register (SLEEP.IDLE) must also be configured.

For devices in the SAM L series the sleep mode is configured by writing the Sleep Mode bit group in the Sleep Configuration register (SLEEPCFG.SLEEPMODE).

If an application is fully interrupt driven, the device can return to sleep even faster if the Sleep On Exit bit in the ARM Cortex System Control register (SCR.SLEEPONEXIT) is set.

For this and some of the following subsections a temperature monitoring application will be used as an example in order to make it easier to understand some of the ultra low power techniques presented in this application note.

Consider a simple application where the temperature in a room is monitored using a temperature sensor with the analog-to-digital converter (ADC). In order to reduce the power consumption the CPU would be put to sleep and wake up periodically on interrupts from a real time counter (RTC). The measured sensor data is checked against a predefined threshold to decide on further action. If the data does not exceed the threshold, the CPU will be put back to sleep waiting for the next RTC interrupt.

3. Event System

Returning to the temperature monitoring application introduced in Sleep Modes on page 4, the CPU will be woken up each time the ADC performs a measurement. As the CPU is put directly back to sleep if the ADC measurement is below the predefined threshold, the solution above will wake up the CPU without reason. As a result the CPU spends an unnecessary amount of time in Active mode and consumes more power than necessary. By using the event system the CPU's time in Active mode can be reduced even further.

Many of the peripherals on devices in the SAM D and SAM L series can be configured to generate events and the Event System can be used to route these events to a different peripheral. How to act upon an incoming event must be configured in the receiving peripheral. For the temperature monitoring application the RTC must be set to generate an overflow event, which is routed to the ADC by configuring the Event System. The ADC must be configured to start a conversion when it receives an event. By using the Event System, an RTC overflow can trigger an ADC conversion without waking up the CPU. Further the ADC can be configured to generate an interrupt if the threshold is exceeded, and the interrupt will wake up the CPU.

4. Direct Memory Access

SAM D11, D21, L21, L22, and C21 have Direct Memory Access Controllers (DMAC), which can be useful for saving power when the CPU is sleeping. Consider that the CPU has to process the preceding temperature measurements after it has been woken up by the temperature threshold being exceeded. This means that the result of each ADC conversion needs to be saved in the memory. Traditionally the CPU would have been woken up for each ADC conversion, and writing the temperature value to the memory before going back to sleep. DMA is a mechanism for transferring data between memories and peripherals without CPU intervention. A DMA channel has to be set up with a source and destination, as well as a transfer trigger. For the temperature monitoring application the ADC result register would be set up as the source, an SRAM address would be set up as the destination, and the ADC Result Ready Event would be set up as the transfer trigger. When the ADC result is ready, it will trigger a DMA transfer transferring the result into SRAM. For further information about the DMAC, refer to the datasheet.

5. SleepWalking

Taking the temperature monitoring application one step further on the road to low power consumption, the SleepWalking feature available in devices from the SAM D and SAM L series can be used. SleepWalking is when a device can provide its peripherals with a clock only when needed, without waking the CPU. This feature is enabled by writing a one to the On Demand bit in the corresponding clock source register (XOSC/XOSC32K/OSC32K/OSC8M.ONDEMAND).

For the temperature monitoring application this means that the ADC's peripheral clock (GCLK_ADC) will only be running when the ADC is converting. When the ADC receives the overflow event from the RTC, it will request its generic clock from the Generic Clock Controller and GCLK_ADC will start running. GCLK_ADC will stop as soon as the ADC conversion is completed.
Note: When using STANDBY as sleep mode the peripheral must be configured to run in standby. Writing a one to the Run in Standby bit in the peripheral's control register, Control A for ADC (CTRLA.RUNSTDBY), will enable the peripheral to run in standby.

Devices in the SAM L21 series have five power domains where unused power domains can be set to a low power state. Setting power domains to a low power state can be configured to happen dynamically during SleepWalking. For the temperature monitoring application, the device can be configured to set the power domain where the ADC is located to a low power state when the ADC is idle. For further information about power domains, refer to Performance Levels and Power Domains on page 8.
Note: Using dynamically power domains during SleepWalking will result in a responsiveness penalty, as turning on power domains is relatively time consuming.

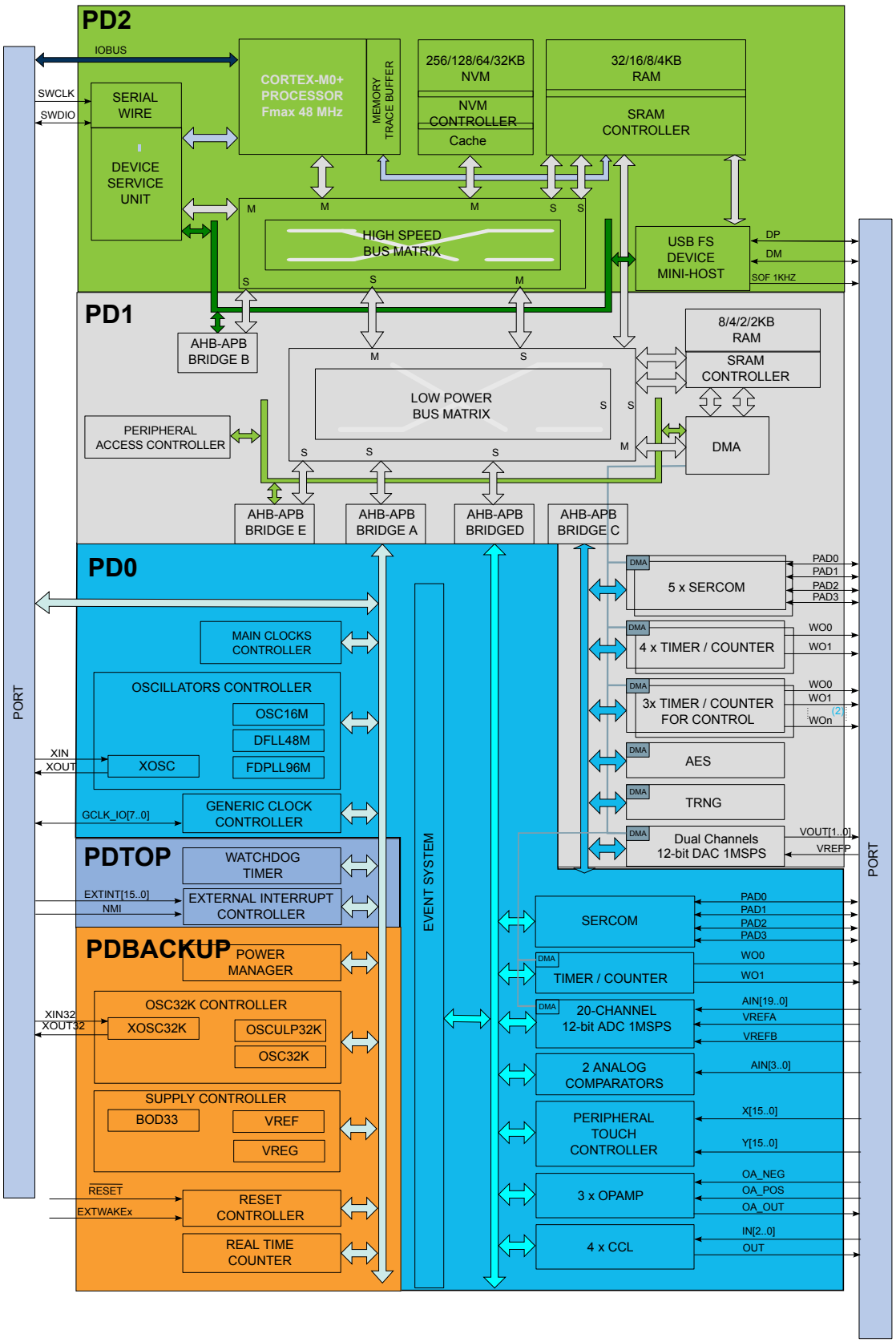
6. Performance Levels and Power Domains

Devices from the SAM L series running from the main voltage regulator (Main VREG) can operate at different performance levels. A change in performance level will change the operational voltage level, which will influence the power consumption. A low performance level will consume less power than a high performance level, but the trade-off is a lower operational frequency. Refer to AT04296: Understanding Performance Levels and Power Domains for more information about the various performance levels.

Devices from the SAM L21 series has five digital power domains, where system modules and peripherals are split amongst these power domains. Refer to Figure 6-1 SAM L21 Power Domains on page 9. The power domains can be either in an active state, retention state, or off state. The retention state is a low power state where the state of registers and SRAM is retained. Taking advantage of the retention and off state for these power domains will reduce the overall power consumption of an application.

Devices from the SAM L22 series does not have the feature to manage the state of power domains.

Figure 6-1 SAM L21 Power Domains



7. Unused Peripherals

Peripherals not required by the application should be disabled (CTRLx.ENABLE is zero). For devices in the SAM D series, the clock signals will propagate to the generic clocks even when the peripheral is not requesting a clock, resulting in an unnecessary increase of the power consumption. Connecting the generic clocks for unused peripherals to a disabled clock generator will prevent this unnecessary power consumption.

8. I/O Pin Configurations

For unused I/O pins, the default configuration should be used, as this disconnects the I/O pads from the I/O pins, reducing the leakage current. With the default configuration for an I/O pin the bit corresponding to the pin in the Data Direction (DIR) register, the Input Enable bit in the Pin Configuration y register (PINCFGy.INEN), and the Pull Enable bit in the Pin Configuration y register (PINCFGy.PULLEN) are all written to zero.

9. Frequency Considerations

In theory, digital CMOS logic consumes power only when the logical signals or the clock signals are toggling, hence keeping the clock frequencies as low as possible is critical for reducing the power consumption. However, lowering the system frequency will give a penalty when it comes to responsiveness. In some cases, where the amount of time spent in Active mode is small, it may be better to run at a high frequency as it will enable for the device to be put back to sleep faster.

For some of the peripherals on devices from the SAM D and SAM L series the generic clock can be prescaled after reaching the peripheral. This feature enables for more clock flexibility but it also requires more power. When the frequency is prescaled in the clock generator instead of in the peripheral, the generic clock propagated to the peripheral is of a lower frequency. A lower frequency means less toggling between the clock generator and the peripheral, which will lower the power consumption.

10. Revision History

Doc. Rev.	Date	Comments
42411B	08/2015	SAM L22 is added
42411A	03/2015	Initial document release

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**