Optimizing Power Consumption of AT91SAM9261-based Systems

Scope
Power consumption, in a CMOS device, has a dynamic power component while it is operating (switching), and a static power component while it is not operating but still powered (non-switching state). The static power (leakage power) dissipation also occurs when the device is operating, but it is not significant compared to the dynamic power. However, static power becomes more significant when the transistor size decreases.

This document discusses low-power system design and battery backup management for systems based on the AT91SAM9261 ARM® Thumb® microcontroller. It describes how to optimize the dynamic component of power consumption.

The first section introduces the system controller, the backup part of the device and the clock management for the ARM core and the peripherals.

The second part describes the ARM core power-saving modes and gives an example of power consumption versus ARM core performance.

This document has been written for the AT91SAM9261 microcontroller but it is applicable to most devices belonging to the SAM9 family.
1. AT91SAM9261 Power Management

Power management in the AT91SAM9261 is built around the System Controller that manages all vital blocks of the microcontroller (interrupts, clocks, power, time, debug and reset). The two main blocks involved in power management are:

- SHDWC (Shutdown Controller) which controls the main power supply. To do so, it is supplied with VDDBU and manages the wake-up input pin and the shutdown output pin, SHDN.
- PMC (Power Management Controller) which controls all clock domains associated with each individual peripheral.

Figure 1-1. System Controller Block Diagram
1.1 Shutdown Controller

In a typical system the SHDN pin is connected to the shutdown input of the DC/DC Converter providing the main power supplies, in particular VDDCORE and/or VDDIO.

The wake-up input can be connected to a pushbuttons or any signals that can potentially wake up the system. The software is able to control the pin SHDN and to select the wake up event.

In this mode, only the backed-up part of the chip remains powered, maintaining operation of the Real-time Timer (RTT) and the content of General-purpose Backup Registers (GPBR). RTT and GPBR can be used to emulate a Real-time Clock. The backup part of the device may be powered by the battery cell only when main supply is off, thus optimizing battery life. The schematic below gives an example of battery management implementation.

**Figure 1-2.** Battery Management Implementation

![Battery Management Implementation](image)

1.2 Power Management Controller

The AT91SAM9261 clocks can be issued from 2 PLLs, a Main Oscillator, or a 32,768 Hz low-power Oscillator which provides a permanent clock to the system.

The Power Management Controller (PMC) optimizes power consumption by controlling all system and user peripheral clocks. The PMC enables/disables the clock inputs to most of the peripherals and the ARM Processor.

The Power Management Controller provides the following clocks:

- MCK, the Master Clock, programmable from a few hundred Hz to the maximum operating frequency of the device.
- Processor Clock (PCK), switched off when entering processor in idle mode.
- Peripheral Clocks, typically MCK, provided to the embedded peripherals (USART, SSC, SPI, TWI, TC, MCI, etc.) and independently controllable.
- UHP Clock (UHPCK), required by USB Host Port operations.
- HClocks (MCK), provided to the AHB high speed peripherals and independently controllable.
  - HCLK0 for the USB Host
  - HCLK1 for the LCD controller

An appropriate clock management allows a significant reduction in dynamic power. If a given peripheral is not being used, the clock to the peripheral can be stopped.
Furthermore, USB device and USB host are fed by the PLLB that can be stopped if USB is not being used.

The dynamic power consumption is roughly proportional to operating frequency. The clock frequency of the processor can be tuned to the lowest value that meets the required processing performance. Precautions should be taken to maintain required real-time constraints.

2. Definition of Operating Modes

The AT91SAM9261 microcontroller offers different power saving modes. Power consumption in low power modes and system recovery time is the trade-off for the system designer.

2.1 Backup Mode

The device enters Backup Mode after the Shutdown Controller disables the main power by controlling the enable pin of a DC/DC converter. The major part of the device (ARM, memories and peripherals) is therefore no longer powered. Only the backup part of the device stays alive, thus maintaining the Real-time Timer, Shutdown Controller activity and the contents of the four General-purpose Backup Registers. Exiting this mode requires the DC/DC converter to be re-enabled by the wake-up logic to power-on the system. Typical power consumption in this mode is around 3 µW for the backup part.

Example of a shutdown/wake-up sequence to enable level change detection on the Wake-up button connected to the wake-up pin:

```c
/* Configure the Wake-up mode in the shutdown mode register*/
AT91C_BASE_SHDWC->SHDWC_SHMR = AT91C_SHDWC_WKMODE_ANYLEVEL;
/* Activate the shutdown pin in the shutdown control register*/
AT91C_BASE_SHDWC->SHDWC_SHCR = (AT91C_SHDWC_KEY | AT91C_SHDWC_SHDW);
```

2.2 Slow Clock Mode

In general, the more power saved, the longer the latency to return to full performance. The slow clock mode is useful when quick reactivation is required.

In Slow Clock Mode, the system clock is the external 32.768 kHz oscillator. PCK and MCK can be set in the range of 512 Hz to 32.768 kHz. The PLL and main oscillator can be disabled if power consumption must be further reduced.

When switching to slow clock, the developer has to manage all the system timings and re-calculate the refresh rate of the SDRAM.

Example code for PCK = MCK = 512 Hz:

```c
/* Switch Master clock to (Slow clock /64)*/
AT91C_BASE_PMC->PMC_MCKR = AT91C_PMC_CSS_SLOW_CLK | AT91C_PMC_PRES_CLK_64;
/* Wait until the master clock is established */
while (!(*AT91C_PMC_SR & AT91C_PMC_MCKRDY));
/* De-activate PLLB for less power consuming */
AT91C_BASE_CKGR->CKGR_PLLBR = AT91C_CKGR_PLLBCOUNT;
/* PCK = MCK = 512Hz */
```
2.3 Idle Mode

The AT91SAM9261 processor supports an idle mode (ARM926™ “wait for interrupt”) in which the clock to the ARM core stops, reducing the power used when the processor is not busy. Almost all kernel operating systems handle power management and thus put the processor in the idle state when the OS determines there are no active tasks. As most systems have an OS timer interrupt running, the processor may go into the idle state thousands of times per second. The processor leaves the idle state when a programmed interrupt (AIC) occurs.

In Idle Mode, only the processor clock is stopped and the rest of the device is clocked by the master clock. Clocks of unused peripherals should be deactivated to participate in power saving.

The following example, based on the Atmel libV3, shows how to enter idle mode and how to leave idle state when the Debug Unit (DBGU) receives a character.

```c
/* Configure DBGU interrupt */
AT91F_AIC_ConfigureIt(AT91C_BASE_AIC, AT91C_ID_SYS,
                      AT91C_AIC_PRIOR_HIGHEST, AT91C_AIC_SRCTYPE_INT_LEVEL_SENSITIVE,
                      AT91F_ASM_DBGU_Handler);
/* Enable on AIC the DBGU interrupt */
AT91F_AIC_EnableIt(AT91C_BASE_AIC, AT91C_ID_SYS);
/* Enable DBGU Receive interrupt */
AT91F_US_EnableIt((AT91PS_USART) AT91C_BASE_DBGU, AT91C_US_RXRDY);
/* Stop the ARM clock in the PMC. This will take effect as soon as the wait-
for-interrupt mode is entered */
AT91C_BASE_PMC->PMC_SCDR = AT91C_PMC_PCK;
AT91F_ARM_WaitForInterrupt();
/* Here the ARM is in idle mode and the system is waiting for a character on
the DBGU to exit the idle state*/

AT91F_ARM_WaitForInterrupt() is the following access to coprocessor CP15 register:

__inline void AT91F_ARM_WaitForInterrupt(){
    register unsigned int sbz = 0;
    __asm("MCR p15, 0, sbz, c7, c0, 4");
}

The function AT91F_ASM_DBGU_Handler is the C routine called by the interrupt ASM handler:

```
2.4 Suspend Mode

In suspend mode, the system context and the SDRAM content are preserved and the system resume sequence is initiated by an internal or an external interrupt. The application executed by the microcontroller restarts in the same state as before suspend mode was entered.

This mode combines idle mode (processor clock stopped) and slow clock mode for the rest of the device. The lowest power consumption is achieved when the system clock is 512 Hz but this is to the detriment of wake-up time.

The following steps occur when suspend mode is initiated:

1. The OS makes calls to each device driver to put the peripheral into a low power state.
2. All the data and processor registers that are not preserved during suspend state are saved to SDRAM.
3. The SDRAM is put into self refresh mode.
4. The system switches to the lowest slow clock which matches the wake-up time requirement.
5. The processor enters idle mode.

2.5 Active Mode

The active mode allows the processor to reach the maximum performance. It is achieved by programming the PLL and the derived Master Clock to the maximum processor clock frequency given in the Electrical Characteristics section of the product datasheet.

The following sequence gives an example of Power Management Controller (PMC) and Clock Generator (CKGR) programming sequence to reach 198 MHz on the processor clock (PCK) and 99 MHz for Master Clock (MCK). The PMC and CKGR modules are detailed in the corresponding section of the product datasheet.

This example, based on the Atmel libV3, is applicable to the AT91SAM9261-EK board which features an 18.432 MHz main oscillator.

```c
/* Set PLLA to : 18.432 x 97 / 9 = 198.5 MHz ==> MULA =97-1; DIVA=9*/
pCkgr->CKGR_PLLAR = 0x2000BF00 | (96 << 15) | 9;
/* Wait the PLLA lock flag raise*/
while (!(*AT91C_PMC_SR & AT91C_PMC_LOCKA));
/* Switch Master Clock */
pPmc->PMC_MCKR = AT91C_PMC_CSS_PLLA_CLK | AT91C_PMC_PRES_CLK |
AT91C_PMC_MDIV_2;
while (!(*AT91C_PMC_SR & AT91C_PMC_MCKRDY));
```
3. Hardware Considerations

Several specific points concerning hardware must be taken into account when building an AT91SAM9261-based system.

3.1 Voltage Monitor and Reset Controller

The AT91SAM9261 features power-on reset and reset logic (Reset Controller) to reduce the external components required to monitor the voltage and reset the system. Thus the bill of materials is reduced and the power to supply an external reset controller and voltage monitor saved.

3.2 LCD Panel and Backlight

The LCD panel consumes the most energy in a system. A TFT screen with backlight is the one most commonly used in PDA-like devices. The backlight can be CCFL (Cold Cathode Fluorescent Lamp) or LEDs. LED technology for backlight uses much less power and, unlike CCFL, does not require an inverter (source of noise) to generate high voltage.

The AT91SAM9261 LCD controller includes a pulse width modulation controller that dims the backlight by software and, if necessary, disables the display.

3.3 Low Power SDRAM

The AT91SAM9261 external bus interface (EBI) can use low-power SDRAM (Mobile SDRAM) which runs at 1.8 volts. The maximum speed on mobile SDRAM is around 90 MHz with 30 pF load capacitance on the buses and 10 pF on the SDRAM clock (refer to the Electrical Characteristics of the product datasheet).

Table 3-1 compares power consumption of two 32 MB standard SDRAM (3.3V) used on AT91SAM9261-EK board (Micron MT48LC16M16A2-75) and two 32 MB 1.8V mobile SDRAM (Micron MT48H16M16LF-75).

Table 3-1. Power Consumption Comparison for SDRAM and Mobile SDRAM

<table>
<thead>
<tr>
<th>SDRAM Type</th>
<th>Operating Current (mA)</th>
<th>Standby Current (mA)</th>
<th>Operating Power (mW)</th>
<th>Standby Power (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT48LC16M16A2-75</td>
<td>125 (*2)</td>
<td>2 (*2)</td>
<td>825</td>
<td>13.2</td>
</tr>
<tr>
<td>MT48H16M16LF-75</td>
<td>80 (*2)</td>
<td>0.3 (*2)</td>
<td>288</td>
<td>1.08</td>
</tr>
</tbody>
</table>

The operating power ratio (SDRAM vs. mobile SDRAM) is around 3 and rises up to 12 for standby power. The mobile SDRAM also supports a deep power-down mode in which the power falls to 18 µW (typical).
4. Power Consumption Measurement Conditions

SAM9 devices rely on the following power supplies:

- VDDBU, which powers the backup part
- VDDCORE, which powers the ARM core and internal memories
- VDDIOM, which powers the External Memories connected on the EBI
- VDDIOP, which powers the Peripheral I/O lines and the USB transceivers

The measurements below concern VDDBU and VDDCORE. The consumption on VDDIOM and VDDIOP is dependent on the hardware design.

4.1 VDDBU in Backup Mode

VDDBU consumption is always present and is slightly influenced by the ambient temperature. Table 4-1 gives values for VDDBU = 1.2V.

Table 4-1. Backup Consumption vs. Ambient Temperature

<table>
<thead>
<tr>
<th>T (°C)</th>
<th>-40</th>
<th>0</th>
<th>25</th>
<th>85</th>
</tr>
</thead>
<tbody>
<tr>
<td>IVDDBU (µA)</td>
<td>2.1</td>
<td>2.3</td>
<td>2.5</td>
<td>6.2</td>
</tr>
</tbody>
</table>

4.2 VDDCORE in Other Modes

Following conditions are defined for reliable measurement:

1. The external environment is considered to be “typical”: ambient temperature is 25°C, VDDBU is 1.2V and VDDCORE is 1.2V.
2. As the embedded peripherals contribute to the power consumption with their digital part, their clocks are disabled.

/* Disable all the peripheral clocks*/
for(i=2;i<32;i++)
    AT91F_PMC_DisablePeriphClock(AT91C_BASE_PMC,((unsigned int) 1 << i));
// Disable HClocks
AT91F_PMC_DisableHCK(AT91C_BASE_PMC,0);
// Disable UDP and UHP Port Clocks (PMC_SCDR)
AT91C_BASE_PMC->PMC_SCDR = AT91C_PMC_UDP | AT91C_PMC_UHP;
3. All I/O lines should have a pull-up or down, depending on the board design.

4.2.1 Slow Clock Mode

In this mode, current consumption onto VDDCORE is about 400 µA with very few variations from 512 Hz to 32.768 kHz.

In this mode (512 Hz), the leakage current is the most significant part of the current consumption.
4.2.2 Idle Mode

The relationship between power, frequency and voltage is the following:

\[ P = C \times f \times V^2 \]

where

- \( C \): equivalent CMOS capacitance of the device
- \( f \): frequency
- \( V \): voltage

For a voltage level fixed at 1.2V, the power is linearly frequency dependent.

Table 4-2 gives the current on VDDCORE versus the master clock (MCK).

Table 4-2. Consumption vs. Bus Clock in Idle Mode

<table>
<thead>
<tr>
<th>MCK (MHz)</th>
<th>12</th>
<th>24</th>
<th>48</th>
<th>99</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ivddcore (mA)</td>
<td>1.4</td>
<td>2.8</td>
<td>5.5</td>
<td>11.2</td>
</tr>
</tbody>
</table>

4.2.3 Active Mode

This part of the application describes the consumption and performance level reached at different speed, memory and processor configurations. In this mode, power consumption is dependent on the frequency. The Dhrystone rating has been used to demonstrate these influences.

Performance level is measured by counting the number of Dhrystone iterations per second. This synthetic code is mainly based on iterations of string copy. It does not use the CPU in the same way as a large, complex piece of embedded software, such as an operating system. In addition, the Dhrystone rating is dependent on the compiler and the optimization level, making a comparison with other ARM9™ devices extremely difficult.

Nevertheless, these few lines of code give results that can then be used to compare different configurations of the same device.

All the voltages and the figures given are for an AT91SAM9261 device mounted on an AT91SAM9261-EK board. All the code examples are based on AT91SAM9261 libv3.

5. Dhrystone Rating vs. Consumption

5.1 Cache and Tightly Coupled Memory

Caches and Tightly Coupled Memory (TCM) are two different approaches with the aim of enhancing system performance when the external memory is slow compared to the core. These are small fast memories local to the core, running out at the core speed.

A cache is a compromise that takes the advantage of the fact that the main part of the subsequent accesses are also from the cache. A whole line of memory locations is cached when a miss occurs. Performance gain relies on the fact that most accesses occur within a small address distance from the current instruction.

TCM is of benefit only if code and data are located in the TCM. The user must enable the TCM and then copy code and data sections into it. Once enabled, TCM forms part of the system memory map and it adopts deterministic behavior.
5.2 Dhrystone

5.2.1 Test Description

The Dhrystone test is compiled with armcc and “o2” time optimization. The generated code size is 10 Kbytes. This fits in Cache and TCM.

The Dhrystone runs under the following conditions:

- In internal SRAM, with or without Instruction Cache (Icache).
- In TCM, without any Cache influence.
- In external PC100 SDRAM. In this case, MMU and Cache can be used.

5.2.2 Raw Results

The following measurements were made on an AT91SAM9261. This device includes:

- 160 Kbytes internal SRAM
- 16 Kbytes Instruction and Data Cache
- A maximum of 64 Kbytes Instruction and Data TCM enabled by 16 Kbyte blocks

Enabling 64 Kbytes ITCM and 64 Kbytes DTCM implies \(160 - (2 \times 64) = 32\) Kbytes left for the internal SRAM.

In Table 5-1, PCK is the processor clock issued by the PLL and MCK is the master clock (bus and peripheral clock).

<table>
<thead>
<tr>
<th>Operating Condition</th>
<th>Code Location</th>
<th>Cache Enabled</th>
<th>PCK (MHz)</th>
<th>MCK (MHz)</th>
<th>Dhrystone (*1000/s)</th>
<th>Cons. (mA)</th>
<th>Dhrystone / s / mA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SRAM</td>
<td>none</td>
<td>48</td>
<td>48</td>
<td>31</td>
<td>17.7</td>
<td>1762</td>
</tr>
<tr>
<td>2</td>
<td>SRAM</td>
<td>none</td>
<td>96</td>
<td>48</td>
<td>39</td>
<td>27.0</td>
<td>1459</td>
</tr>
<tr>
<td>3</td>
<td>SRAM</td>
<td>none</td>
<td>96</td>
<td>96</td>
<td>62</td>
<td>34.1</td>
<td>1826</td>
</tr>
<tr>
<td>4</td>
<td>SRAM</td>
<td>none</td>
<td>198</td>
<td>99</td>
<td>81</td>
<td>53.4</td>
<td>1522</td>
</tr>
<tr>
<td>5</td>
<td>SRAM</td>
<td>I</td>
<td>48</td>
<td>48</td>
<td>39</td>
<td>18.4</td>
<td>2097</td>
</tr>
<tr>
<td>6</td>
<td>SRAM</td>
<td>I</td>
<td>96</td>
<td>48</td>
<td>55</td>
<td>27.9</td>
<td>1989</td>
</tr>
<tr>
<td>7</td>
<td>SRAM</td>
<td>I</td>
<td>96</td>
<td>96</td>
<td>77</td>
<td>35.5</td>
<td>2180</td>
</tr>
<tr>
<td>8</td>
<td>SRAM</td>
<td>I</td>
<td>198</td>
<td>99</td>
<td>115</td>
<td>55.0</td>
<td>2083</td>
</tr>
<tr>
<td>9</td>
<td>SRAM</td>
<td>MMU+I</td>
<td>198</td>
<td>99</td>
<td>155</td>
<td>60.2</td>
<td>2578</td>
</tr>
<tr>
<td>10</td>
<td>SRAM</td>
<td>MMU+I+D</td>
<td>198</td>
<td>99</td>
<td>324</td>
<td>87.1</td>
<td>3716</td>
</tr>
<tr>
<td>11</td>
<td>TCM</td>
<td>N/A</td>
<td>48</td>
<td>48</td>
<td>76</td>
<td>22.5</td>
<td>3368</td>
</tr>
<tr>
<td>12</td>
<td>TCM</td>
<td>N/A</td>
<td>96</td>
<td>48</td>
<td>150</td>
<td>39.0</td>
<td>3841</td>
</tr>
<tr>
<td>13</td>
<td>TCM</td>
<td>N/A</td>
<td>96</td>
<td>96</td>
<td>151</td>
<td>43.5</td>
<td>3473</td>
</tr>
<tr>
<td>14</td>
<td>TCM</td>
<td>N/A</td>
<td>198</td>
<td>99</td>
<td>310</td>
<td>76.7</td>
<td>4036</td>
</tr>
<tr>
<td>15</td>
<td>SDRAM</td>
<td>none</td>
<td>198</td>
<td>99</td>
<td>39</td>
<td>42.6</td>
<td>887</td>
</tr>
<tr>
<td>16</td>
<td>SDRAM</td>
<td>I</td>
<td>198</td>
<td>99</td>
<td>82</td>
<td>48.2</td>
<td>1709</td>
</tr>
<tr>
<td>17</td>
<td>SDRAM</td>
<td>MMU+I</td>
<td>198</td>
<td>99</td>
<td>100</td>
<td>50.2</td>
<td>1984</td>
</tr>
<tr>
<td>18</td>
<td>SDRAM</td>
<td>MMU+I+D</td>
<td>198</td>
<td>99</td>
<td>320</td>
<td>83.2</td>
<td>3848</td>
</tr>
</tbody>
</table>
5.2.3 Performance and Consumption Results and Analysis

Figure 5-1. Consumption and Dhrystone Results versus Operating Conditions

The best Dhrystone rates are obtained for PCK = 198 MHz and MCK = 99 MHz for the following configurations:

- Running out of SRAM with MMU, Icache and Dcache enabled.
- Running out of SDRAM with MMU, Icache and Dcache enabled.

Note: The Dhrystone code and data fit the Icache and Dcache, limiting the cache misses to a minimum. This configuration explains why the results given above are similar.

- Running out of TCM

Note: TCM memories are mapped into the processor’s address space and are more efficient than caches since the hardware logic required to control the cache is not required. Moreover the layout of memory objects during execution of the application is fixed at compile time and so the memory accesses are predictable. Using TCM in a real-time embedded system or for fast data computing is advantageous. But ideal performance cannot be duplicated. This is because the Dhrystone code contains some literal constants within the code section, thus the Data interface must have access to Instruction TCM. However, penalty cycles occur for these data accesses, thus decreasing the performances by about 5%.

- Running without cache

Note: When the cache is disabled, a cached processor does not perform as well as one with no cache.
The best ratio for number of Dhrystone per mA are obtained when using TCM in all cases. Working with MMU and enabling Instruction and Data Cache also result in good results with SRAM and SDRAM for high-speed configurations.

### 5.3 Conclusion

- For optimal performance running out of SRAM (or SDRAM) with MMU, Icache and Dcache should be enabled at high speed. TCM performance is very close and should be the best for code without literal constants.
- Using TCM leads to the best ratio performance/consumption in all cases. Its usage is highly recommended and optimizes power consumption without penalizing performance.