Low Power Design
Materials

- LPC11xx user manual: Chapter 3, 4, 5
- AN11027: Using the LPC1100 low power modes and wake-up times on the LPCXpresso
The Challenge

• High-performance systems
  – Cooling and packaging cost
    • > 40W: 1W -> $1
    • Air-cooling: reaching limits
  – Electricity bill
  – Reliability
  – Desktop PCs consume around 10% power in US

• Usability of portable systems
  – Battery lifetime

• Restriction factor for high-performance server design
  – Power determines processor density
Power consumption

Source: Intel®

Year

Power Density (W/cm²)

- Hot Plate
- Nuclear Reactor
- Rocket Nozzle
- Sun’s Surface
- Pentium®

Power consumption

Power in a CMOS Gate

$V_{DD}$

$Ground$

$i_{DD}(t)$
More details

\[ P_{\text{total}} (0 \rightarrow 1) = C_L V_{DD}^2 + t_{sc} V_{DD} I_{\text{peak}} + V_{DD} I_{\text{leakage}} \]
Dynamic power

\[ v_i(t) \rightarrow R_{on} \rightarrow V_{DD} \rightarrow i_c(t) \rightarrow v_o(t) \]

\[ R=large \]

\[ C_L \]

Ground
Dynamic power

\[ V_{DD} - V_{Tp} \]

\[ V_{DD} \]

\[ V_{i}(t) \]

\[ V_{o}(t) \]

\[ i_{sc}(t) \]

\[ t_B \]

\[ t_E \]

\[ V_{Tn} \]

\[ I_{scmaxf} \]

\[ GND \]

\[ V_{DD} \]

\[ V_{o}(t) \]

\[ V_{i}(t) \]

\[ V_{DD} \]

\[ V_{Tn} \]

\[ I_{scmaxf} \]

\[ GND \]
Static power

High-k Dielectric reduces leakage substantially

Benefits compared to current process technologies

<table>
<thead>
<tr>
<th></th>
<th>High-k vs. SiO₂</th>
<th>Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacitance</td>
<td>60% greater</td>
<td>Much faster transistors</td>
</tr>
<tr>
<td>Gate dielectric leakage</td>
<td>&gt; 100x reduction</td>
<td>Far cooler</td>
</tr>
</tbody>
</table>
Power optimization

- Clock gating
- Voltage scaling
Power optimization
The design of the Cortex-M0 processor is very small

Cortex-M0 utilizes many low-power design techniques

High code density hence lower power of the flash memory

The Cortex-M0 offers much higher performance than others

Reduces noise to allow better accuracy in analog applications like sensors

Reduces interference in wireless and radio frequency

Allows a simpler and cost-effective power supply design
ARM Cortex M0

- Stopping some or all of the clock signals
- Reducing the clock frequency to some parts of the microcontrollers
- Reducing voltage to various parts of the microcontroller
- Turning off the power supply to some parts of the microcontroller
ARM Cortex MO

- Run mode
- Sleep mode
- Deep sleep mode
- Deep power down mode
Run Mode

- Microprocessor in normal operation
- Certain components, e.g., ADC, PLL, can be power off or clock disabled
In Sleep mode, the system clock to the ARM Cortex-M0 core is stopped, and execution of instructions is suspended until either a reset or an enabled interrupt occurs.

1. Peripheral functions, if selected to be clocked in the SYSAHBCLKCTRL register, continue operation during Sleep mode and may generate interrupts to cause the processor to resume execution. Sleep mode eliminates dynamic power used by the processor itself, memory systems and their related controllers, and internal buses. The processor state and registers, peripheral registers, and internal SRAM values are maintained, and the logic levels of the pins remain static.
Deep Sleep

In Deep-sleep mode, the system clock to the processor is disabled as in Sleep mode. All analog blocks are powered down, except for the BOD circuit and the watchdog oscillator, which must be selected or deselected during Deep-sleep mode in the PDSLEEPCFG register. See section 3.5 for more details.

1. Deep-sleep mode eliminates all power used by the flash, analog peripherals and all dynamic power used by the processor itself, memory systems and their related controllers, and internal buses. The processor state and registers, peripheral registers, and internal SRAM values are maintained, and the logic levels of the pins remain static.
Deep Power Down

In Deep Power-down mode, power and clocks are shut off to the entire chip with the exception of the WAKEUP pin.

1. During Deep power-down mode, the contents of the SRAM and registers are not retained except for a small amount of data which can be stored in five 32-bit general purpose registers of the power management unit block.
ARM Cortex M0

- Stopping some or all of the clock signals
- Reducing the clock frequency to some parts of the microcontrollers
- Reducing voltage to various parts of the microcontroller
- Turning off the power supply to some parts of the microcontroller
Run-time flow
### System Control Reg (SCR)

<table>
<thead>
<tr>
<th>Bits</th>
<th>Field</th>
<th>Type</th>
<th>Reset Value</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:5</td>
<td>Reserved</td>
<td>—</td>
<td>—</td>
<td>Reserved.</td>
</tr>
<tr>
<td>4</td>
<td>SEVONPEND</td>
<td>R/W</td>
<td>0</td>
<td>When set to 1, an event is generated for each new pending of an interrupt. This can be used to wake up the processor if Wait-for-Event (WFE) sleep is used.</td>
</tr>
<tr>
<td>3</td>
<td>Reserved</td>
<td>—</td>
<td>—</td>
<td>Reserved.</td>
</tr>
<tr>
<td>2</td>
<td>SLEEPDEEP</td>
<td>R/W</td>
<td>0</td>
<td>When set to 1, deep sleep mode is selected when sleep mode is entered. When this bit is 0, normal sleep mode is selected when sleep mode is entered.</td>
</tr>
<tr>
<td>1</td>
<td>SLEEPONEXIT</td>
<td>R/W</td>
<td>0</td>
<td>When set to 1, enter sleep mode (Wait-for-Interrupt) automatically when exiting an exception handler and returning to thread level. When set to 0, this feature is disabled.</td>
</tr>
<tr>
<td>0</td>
<td>Reserved</td>
<td>—</td>
<td>—</td>
<td>Reserved.</td>
</tr>
</tbody>
</table>
ARM Cortex M0

- Execution of a Wait-for-Event (WFE) instruction
  - __WFE()

- Execution of a Wait-for-Interrupt (WFI) instruction
  - __WFI()

- Using the Sleep-on-Exit feature
  - SCB->SCR = SCB->SCR | 0x2
How-to

SCB->SCR = SCB->SCR | 0x2; // Enable Sleep-On-Exit feature

while (1)
{
    __WFI(); // Execute WFI and enter sleep
}

{ __WFI(); // Execute WFI and enter sleep
};
The PCON register selects whether one of the ARM Cortex-M0 controlled power-down modes (Sleep mode or Deep-Sleep mode) or the Deep power-down mode is entered. It also provides the flags for Sleep or Deep-sleep modes and Deep power-down modes respectively.
# PCON

<table>
<thead>
<tr>
<th>Bit</th>
<th>Symbol</th>
<th>Value</th>
<th>Description</th>
<th>Reset value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>Reserved. Do not write 1 to this bit.</td>
<td>0x0</td>
</tr>
<tr>
<td>1</td>
<td>DPDEN</td>
<td>Deep power-down mode enable</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>ARM WFI will enter Sleep or Deep-sleep mode (clock to ARM Cortex-M0 core turned off).</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>ARM WFI will enter Deep-power down mode (ARM Cortex-M0 core powered-down).</td>
<td></td>
</tr>
<tr>
<td>7:2</td>
<td>-</td>
<td>-</td>
<td>Reserved. Do not write ones to this bit.</td>
<td>0x0</td>
</tr>
<tr>
<td>8</td>
<td>SLEEPFLAG</td>
<td>Sleep mode flag</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>Read: No power-down mode entered. LPC111x/LPC11C1x is in Active mode. Write: No effect.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Read: Sleep/Deep-sleep or Deep power-down mode entered. Write: Writing a 1 clears the SLEEPFLAG bit to 0.</td>
<td></td>
</tr>
<tr>
<td>10:9</td>
<td>-</td>
<td>-</td>
<td>Reserved. Do not write ones to this bit.</td>
<td>0x0</td>
</tr>
<tr>
<td>11</td>
<td>DPDFLAG</td>
<td>Deep power-down flag</td>
<td>0x0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>Read: Deep power-down mode not entered. Write: No effect.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Read: Deep power-down mode entered. Write: Clear the Deep power-down flag.</td>
<td></td>
</tr>
<tr>
<td>31:12</td>
<td>-</td>
<td>-</td>
<td>Reserved. Do not write ones to this bit.</td>
<td>0x0</td>
</tr>
</tbody>
</table>
The registers

- **SCR**: System control register
  - Controls entry to and exit from low power state

- **AHBCLKCTRL**: System AHB clock control register
  - Enables the clocks to individual system and peripherals

- **PDRUNCFG**: Power-down configuration register
  - Control the power to the various blocks

- **PDWAKECFG**: Wake-up configuration register
  - Determine the state the chip enters when it is waking up from deep-sleep mode

- **PDSLEEPCFG**: Deep-sleep mode configuration register
  - Controls watchdog (WD) oscillator and BOD circuit during deep-sleep mode
Be careful when developing applications with deep sleep mode or deep power down mode. In these two modes you could lose connectivity between the in-circuit debugger and the microcontroller. If you power down the microcontroller soon after it starts running, you could end up being unable to connect the debugger to the microcontroller to carry out debug operations. This also affects flash programming. There are various solutions to this problem:

1. During software development you could add conditional executed code at the initialization stage so that you can switch a pin at reset to disable the deep sleep or power-down operation. This conditional executed code could be removed from the project later on once you are sure that the power management code is working correctly.

2. Depending on the microcontroller product, there can be a special boot mode to disable the execution of the application programmed in the flash memory. In the NXP LPC111x, port 0 bit 1 can be used in such situation. The NXP111x has an in-system programming (ISP) feature to allow the flash to be programmed using the boot loader and the serial port. By pulling bit 1 of port 0 to low at powerup reset, the ISP program in the boot loader will be executed. You can use the ISP feature to update the flash or to connect the in-circuit debugger to the microcontroller and update the flash. For example, if you are using Keil MCB1000 board with the NXP LPC1114 microcontroller and accidentally lock up the board because you have used the powerdown feature, you can disconnect power to the board, press and hold the boot button, and connect the power again. Then you should be able to reprogram the flash.
Brown Out Detection

- $V_{CC}$
- $V_{BOT^-}$
- $V_{BOT^+}$
- RESET
- TIME-OUT
- INTERNAL RESET
- $t_{TOUT}$