PSoC Creator 101: PSoC 4200 Low Power Modes

Hello, I'm Alan Hawse; Executive Vice President of Software at Cypress Semiconductor.

PSoC 4 was designed for your low power application. It is the perfect chip to use in IOT (Internet Of Things) and wearable applications; whether it's adjusting clock frequencies, moving in and out of low power modes or lowering the CPU load by pushing critical functions into the hardware. PSoC gives you the ability to create a low power design that fits your requirements. This lesson will cover the PSoC 4 power modes that are used to lower the power consumption.

For more information on these topics and example projects, please see the application note AN86233 entitled PSoC 4 Low-Power Modes and Power Reduction Techniques.

PSoC 4 features five power modes; in order of power consumption and functionality they are: Active, Sleep, Deep-sleep, Hibernate and Stop. Cypress followed the ARM standard names. PSoC Creator provides API's for entering each of the low power modes. Methods for returning to active mode vary based on the resources available in the current low power mode.

Active Mode provides full device functionality; each of the subsequent modes provides a subset of CPU and peripherals, wake and up and reset sources, and power mode transition behavior. Active mode power consumption is dependent on the CPU clock frequency and the number and frequency of the active peripherals.

In Sleep mode the CPU is disabled while all of the hardware peripherals are kept active. This mode is an excellent option if there's a hardware task that the CPU must wait for; such as a CapSense scan or a communication transfer. Entering Sleep mode can be done by calling the CySysPmSleep API. The part will return to Active mode when any interrupt is triggered. All interrupt sources are available in the Sleep mode. Issuing a reset will also bring the part back to Active mode. The Sleep mode current is also dependent on the IMO frequency. At 12MHz with the I2C wakeup low power comparators and the watchdog enabled, the part will typically draw about 1.7 mA.

The Deep-sleep mode puts the processor to sleep and then shuts down the UDB's, analog peripherals, such as the ADC and the CTBMs and the high speed clocks. Low-speed peripherals such as the low power comparators, I2C block and the watchdog timer are still enabled and can be used as wakeup sources. Deep-sleep should be used when the part is not currently executing a task. For instance, between system ticks, but a periodic wakeup is still required. Deep-sleep is entered through CySysPmDeepSleep API. The part will return to Active mode when an interrupt is triggered. In Deep-sleep the interrupt sources available are the PICU, the low power comparator, the SCB I2C address match and the watchdog timer. Issuing a reset will also bring the part back to Active mode. In Deep-sleep mode the part will typically draw about 15 uA, three orders of magnitude less that Sleep mode.

In Hibernate mode all of the clocks are turned off but the part retains its logical states. Such as the pin drive modes, the SRAM, and some UDB registers. Hibernate should be used for long periods of inactivity where the device is not responsible for any critical tasks and a periodic wakeup is not required. You can put the part into Hibernate mode using the CySysPmHibernate API. Exiting Hibernate can be done through a pin interrupt or a low power comparator interrupt. Any exit out of Hibernate will issue a device reset. However, as the SRAM state is retained during Hibernate mode the source of the reset can be detected using the CySysPmGetResetReason API. If you want the pins to retain their state upon wakeup from Hibernate; they should be frozen before entering the Hibernate state using the CySysPmFreezeIo API, and then unfrozen upon wakeup from Hibernate using the CySysPmUnfreezeIo API. In Hibernate mode the part will typically draw about 150 nA; that's not very much current.

Stop mode offers the lowest power consumption without actually removing power from the PSoC pins. All the peripherals are disabled and the register states including the SRAM are not retained. The device pins may be frozen to

retain their drive modes and their pin states while in stop mode. Stop mode should be used when the absolute lowest power is required and the system states do not need to be retained. Entering Stop mode is done through the CySysPmStop API. This API automatically freezes the GPIO states. Wakeup from Stop mode occurs when the dedicated stop wakeup pin; port 0, pin 7 is triggered. The pin can be triggered on either a rising or a falling edge. When the part comes out of Stop mode it issues a reset. The pins in the chip will remain frozen until the API CySysPmUnfreezeIo is called. The stop reset event can also be detected at startup using the CySysPmGetResetReason API. In Stop mode the PSoC 4 will typically draw about 20 nAs.

In summary, the power modes of the PSoC 4 are Active, Sleep, Deep-sleep, Hibernate and Stop. Each one uses a decreasing amount of power. In the next lesson I will describe techniques for lowering your power. As always, if you have questions about PSoC Creator or PSoC in general; you're welcome to email me, alan_hawse@cypress.com and I will make sure that your questions are answered.