Dylan Herron

Keith Robertson

Prelab 4

The Cortex M0 has three primary sleep modes: normal sleep, deep sleep, and Deep Power-Down Mode.

* Normal Sleep
  + The core clock of the Cortext M0 is stopped until either a reset or an interrupt occurs
  + Peripheral devices are NOT shut down and may interrupt the processor to wake it up
  + The processor state and registers, peripheral registers, and internal SRAM values are maintained
* Deep Sleep
  + The core clock of the Cortext M0 is stopped until either a reset or an interrupt occurs
  + Analog blocks are powered down
  + Flash memory is powered down
  + The Brown Out Detection (BOD) and watchdog timers are NOT disabled
  + The processor state and registers, peripheral registers, and internal SRAM values are maintained
* Deep Power-Down Mode
  + The WIC can reduce power even further than deep sleep mode by stopping all clock signals to the processor and powering down almost all of the processor (putting it in a retention state)
  + The processor state and registers, peripheral registers, and internal SRAM values are NOT maintained except for a small amount of data stored in 5 32 bit registers in the power management unit.
  + When the WIC receives an interrupt, it sends a signal the power management unit to restore power and the clock to the processor. The processor then services the interrupt request.
  + All functional pins are disabled (tri-stated) except for the wakeup pin.
  + Uses State Retention Power Gating (SRPG) to reduce power by powering off most parts of the logic and leaving only a small memory element in each flip-flop to retain the state of the machine.

There are three ways to enter sleep mode. All three methods will take the processor into the sleep mode that has been configured (ie if the Cortex M0 is configured to enter deep sleep, using any of the following three methods will result in the Cortex M0 going into deep sleep).

* WFI
  + Wait for interrupt
  + The wait for interrupt instruction can be woken up either by interrupts of a higher priority than the current interrupt or by debug requests.
* WFE
  + Wait for event
  + The wait for event instruction can be woken up by interrupts or events including external events and debug requests.
* The Sleep on Exit feature
  + When enabled, the sleep on exit feature has the effect of calling WFI after every interrupt request exits.
  + Saves additional power because every time the processor exits the interrupt handler, it does not have to restore the registers because it would just have to put the registers back on the stack when the next interrupt request occurred (the processor is only servicing interrupts, not executing other code).