

## **Characterizing Custom Analog Blocks in Mixed-Signal Microcontrollers**

**Jay Jayakumar**

[jay.jayakumar@luminarymicro.com](mailto:jay.jayakumar@luminarymicro.com)

Luminary Micro

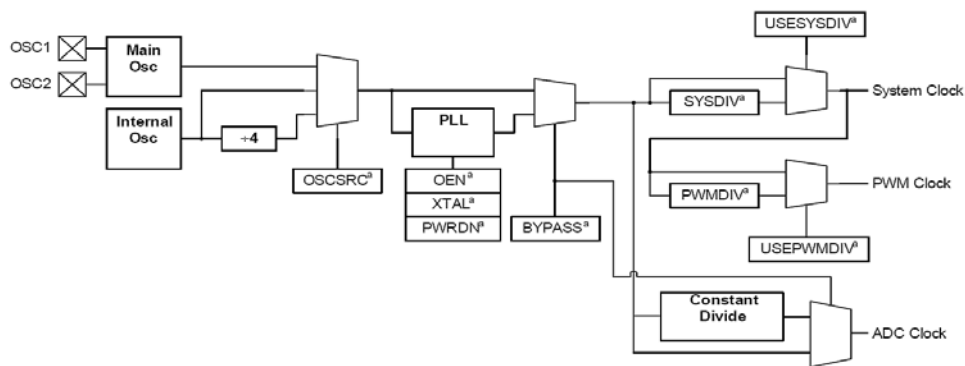
At Luminary Micro, we design ARM® Cortex™-M3-based microcontrollers (MCUs) targeted at embedded consumer applications and industrial control. The Luminary Stellaris® MCUs also contain custom analog blocks such as oscillators, PLLs, and ADCs to support clocking and mixed-signal applications. Designing and verifying these custom analog blocks impacts our time-to-market, and we constantly look for ways to speedup these tasks. This article describes our use of the Analog FastSPICE™ circuit simulator (AFS) from Berkeley Design Automation for verifying the design of a crystal oscillator, transistor-level closed-loop PLL simulation, and characterizing a 10-bit, 8-channel ADC. With our traditional SPICE simulator, we had to wait for two to three days for the longer simulations to complete. AFS provides identical results in overnight runs. Because we require a large number of simulations for verification and characterization, we estimate that AFS has improved our custom block design productivity by as much as 40%.

### **Mixed-Signal Microcontrollers Background**

Luminary Micro's Stellaris microcontrollers incorporate the ARM Cortex-M3 32-bit MCU core, a low drop-out voltage regulator, battery backed low-power hibernation, analog comparators, a 10-bit ADC module, and peripherals designed specifically for intense industrial motor control such as motion control pulse width modulation (PWM) and quadrature encoder inputs. Our controllers support a wide range of applications including motion control, monitoring (remote, fire/security, etc.), building controls, energy monitoring and conversion, network appliances, factory automation, electronic point-of-sale machines, test and measurement equipment, and medical instrumentation.

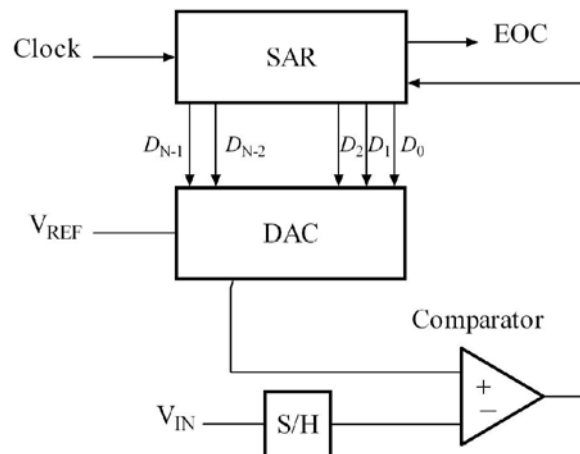
In addition to the ARM core implementation, Luminary microcontrollers have several custom analog blocks, including crystal oscillators, PLLs and ADCs.

The oscillators and PLL generate the various clocks for the MCU. Figure 1 shows the high-level block diagram for the main clock tree. The clock sources are the main external oscillator, the internal oscillator, and the PLL. Both sources and the PLL have the ability to drive the system clock. When the PLL is in use, it outputs a 200-MHz clock signal, and when combined with the system divider (SYSDIV), generates the system clock. The PWM module clock is derived from the system clock. The ADC clock uses the PLL as the clock source and a constant divider to meet the required operating range.



**Figure 1. PLL and Clock Tree**

The ADC module features 10-bit conversion resolution and supports eight input channels plus an internal temperature sensor. Figure 2 shows the high-level block diagram of the ADC, which implements successive approximation architecture (SAR). [1]



**Figure 2. ADC Block Diagram**

The SAR ADC module also includes a programmable sequencer which enables sampling multiple analog input sources without controller intervention. Each sample sequence provides flexible programming with fully configurable input source, trigger events, interrupt generation, and sequence priority. In addition to traditional single-ended sampling, the ADC module supports differential sampling of two analog input channels.

### **Analog Block Verification and Characterization Requirements**

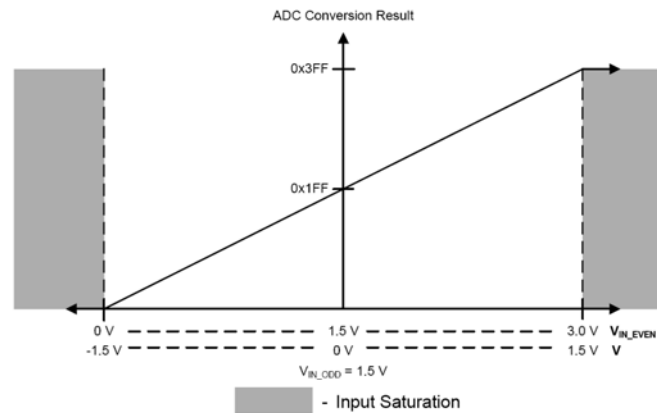
This section describes requirements for verifying the crystal oscillator and PLL and for characterizing the 10-bit, 8-channel ADC. We simulated the custom blocks at the transistor level with parasitically extracted netlists. In many cases, we used corner simulations to verify the impact of variations in process, voltage, and temperature (PVT) on key performance specifications.

The clock tree includes two custom blocks: an on-chip crystal oscillator and an integer-N PLL that drive the system and peripherals. Crystal oscillators are known for requiring very long SPICE simulations to start up and reach steady state. This is especially true for low-power designs. Our on-chip crystal oscillator must work in first silicon, so our verification methodology requires transistor-level SPICE transient simulation.

Our PLL verification includes closed-loop transistor-level transient simulations to verify locking and jitter. The PLL simulation is time-consuming because we simulate lock time and jitter across multiple PVT corners. After reaching lock, we extend the simulations under various power supply noise conditions to ensure that the jitter continues to meet specifications. These supply noise simulations often run for several days.

Our SAR ADC module characterization methodology also requires multiple, lengthy simulations. The ADC generates a 10-bit output value for selected analog input. The converter uses an internal reference that yields sample values ranging from 0x000 at 0V input to 0x3FF at 3V input in single-ended input mode. Figure 3 shows an example of the

differential ADC transfer curve with the negative input centered at 1.5 V. In this configuration the differential range spans from -1.5 V to 1.5 V. [2]



**Figure 3. ADC Differential Sampling Range Example**

Our ADC characterization includes validating the ADC transfer curve. We ramp a DC voltage at the ADC input to generate the transfer curve. We examine the curve for missing codes and perform root-cause analysis for any problems. We also simulate PVT corners on a sufficient sample of input voltage values to ensure proper operation across all conditions.

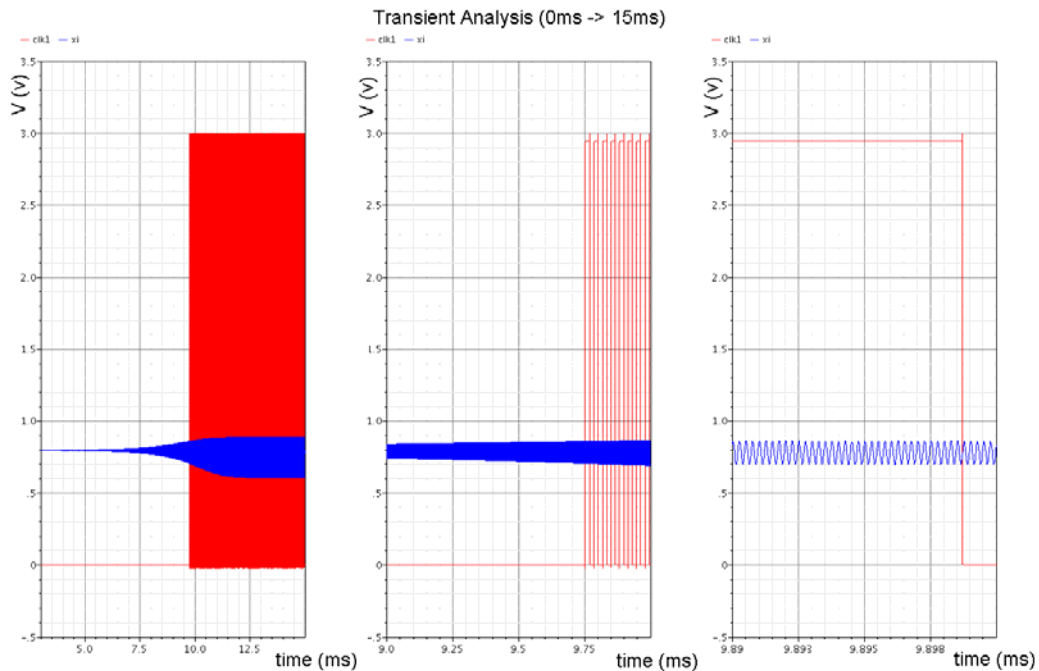
### **Analog FastSPICE Methodology and Results**

This section shows the Analog FastSPICE (AFS) simulation results for our crystal oscillator, PLL, and SAR ADC.

With our traditional SPICE simulator, we had to wait as long as three days for transient results. AFS provides us identical results down to the SPICE noise floor 5x-10x faster. This means we get the results overnight which enables us to work much more efficiently. Table 1 summarizes the AFS runtimes and speedup versus traditional SPICE for the crystal oscillator, PLL, and ADC.

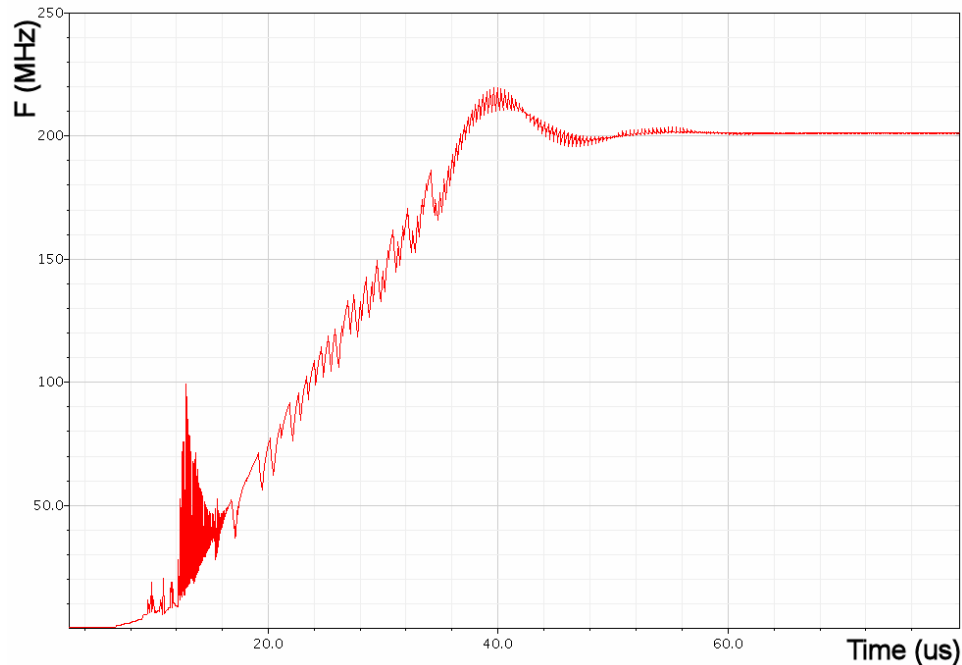
<b>Custom Block</b>	<b>Total # Elements</b>	<b>SPICE Runtime</b>	<b>AFS Runtime</b>	<b>AFS Speedup</b>
<b>Crystal Oscillator</b>	750	44hrs	8.2hrs	5.4X
<b>Integer-N PLL</b>	1,250	72hrs	12hrs	6.0X
<b>10 Bit ADC</b>	10,050	60hrs	9.5hrs	6.3X

Although the oscillator only contains 750 elements including startup circuitry, the simulations with traditional SPICE lasted 44 hours. AFS finished in 8.2 hours: 5.4 times faster. Figure 4 shows our crystal oscillator simulation results. The red curve is the final clock output and the blue curve is the oscillator output. The plot on the left shows the overall simulation results for the oscillator, and the other plots show the details as the oscillator reaches a steady state.



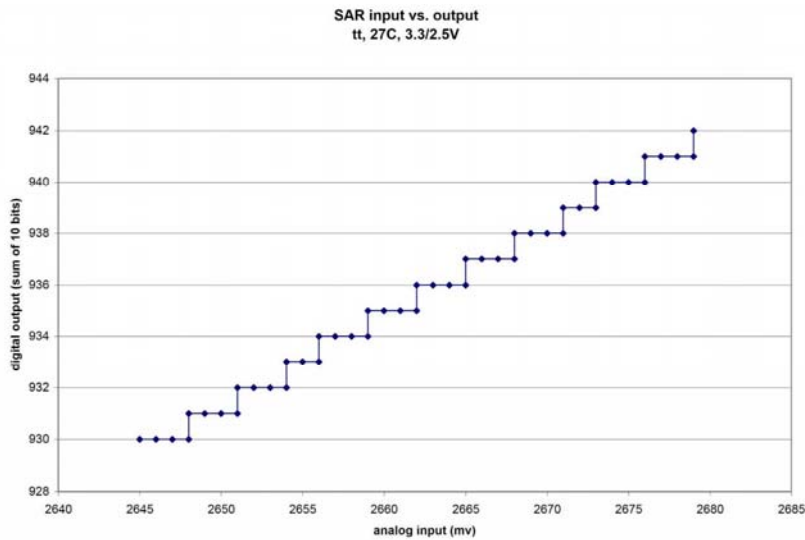
**Figure 4. Crystal Oscillator Start-up Simulation Results**

The PLL locking simulation took three days to run with our traditional SPICE tool. AFS finished in 12 hours: 6 times faster and, more importantly, fast enough to give us results overnight. Figure 5 shows the PLL locking simulation. In addition to the locking simulations, we ran post-lock transient simulations to ensure correct operation under varying power supply levels and switching noise conditions.



**Figure 5. PLL Output Simulation Results**

SAR ADC characterization is another time-consuming step in our design methodology. A single ADC run took 2.5 days with traditional SPICE. AFS finished in 9.5 hours: 6.3 times faster. Again, we got results overnight instead of waiting for several days. Figure 6 shows a representative section of the 10-bit SAR ADC transfer curve. We obtained the transfer curve by ramping a DC voltage at the ADC input in 1mV increments to generate all  $2^{10}$  code conditions. The X-axis shows the voltage increments and the Y-axis shows the sum of the 10-bit digital output. Once we had a transfer curve, we checked for missing or multiple codes. We also ran PVT simulations on a sufficient sample of output codes to validate operation under varying corner conditions.



**Figure 6. ADC Characterization Results**

### Summary and future work

We used the Berkeley Design Automation Analog FastSPICE circuit simulator to verify a crystal oscillator, simulate transistor-level closed-loop PLL performance, and characterize a 10-bit, 8-channel SAR ADC. The AFS results completed in overnight runs and matched the output of our standard SPICE simulator obtained from 2-3 days long simulations. With AFS in our custom block verification flow, we estimate that we improved our design productivity by as much as 40% due to increased efficiency from much faster turnaround times. Our improved productivity also let us perform even more thorough validation of our custom designs before tapeout. Perhaps just as importantly, AFS required negligible effort to learn or integrate into our flow.

For more information on Luminary Micro company and products, visit [www.luminarymicro.com](http://www.luminarymicro.com)

### About The Author

Jay Jayakumar joined Luminary Micro in 2006, where he leads a team responsible for the design of custom blocks. Prior to that, he was a consultant for 12 years, focused on the design of analog and mixed signal circuits. Jay has held engineering and management positions at Advanced Micro Devices. He has a Ph.D in Electrical Engineering from Southern Methodist University, Dallas, Texas.

## References

- [1] R. J. Baker, CMOS Circuit Design, Layout, and Simulation, Revised Second Edition, Wiley-IEEE, 2008. ISBN 978-0-470-22941-5
- [2] LM3S5769 Microcontroller Datasheet, [www.luminarymicro.com](http://www.luminarymicro.com)