CoreAI and SmartGen Implementation in Fusion

Table of Contents

Introduction

The Microsemi Fusion® device is the world's first mixed-signal field programmable gate array (FPGA). As the ultimate Programmable System Chip (PSC), Fusion integrates a configurable Analog Block (AB), Flash memory, and FPGA fabric in a monolithic PSC. Coupling Fusion with a discrete or soft microcontroller, such as Microsemi CoreMP7 (an optimized soft-core ARM7™ implementation), as well as CoreAI (Analog Interface) or SmartGen macros opens new doors in the embedded sector by combining the benefits of programmable logic, analog circuitry, and embedded processing in a single chip.

The interface between the Fusion AB and the embedded soft-core processor is user-configurable. The embedded designer has the option of implementing the processor AB interface through either CoreAI or the combination of the processor's I/O port(s) and SmartGen IP macros. The scope of this application note is to identify the benefits and limitations of each so as to provide the embedded designer with enough information to determine which implementation is ideal for a system. Block diagrams are included to assist the designer in the interface implementation.

CoreAI

CoreAI allows simple control of the analog peripherals within the Fusion family of Microsemi devices. Control can be implemented with an internal or external microprocessor or microcontroller (such as CoreMP7), or with user-created custom logic within the FPGA fabric. The industry-standard Advanced Microcontroller Bus Architecture (AMBA) Advanced Peripheral Bus (APB) slave interface is used as the primary control mechanism within CoreAI.

CoreAI instantiates the Fusion AB macro, as shown in Figure 1 on page 2, which includes the Analog Configuration Multiplexer (ACM) interface, Analog Quads, and Real-Time Counter (RTC). The ACM interface, within the AB macro, is used to control configuration of the Analog Quads and RTC in the Fusion device. CoreAI generates the control signals used by the ACM, including its clock signal, which is generated by an internal clock divider. The ACM clock divider is used to ensure that the ACM interface is clocked at a frequency less than or equal to 10 MHz. For more details on the silicon features of the AB macro, such as the Analog Quads, RTC, or ACM, refer to the *Fusion Family of Mixed-Signal Flash FPGAs datasheet*.

Control of the analog to digital converter (ADC) within the AB macro in CoreAI is accomplished by APB reads and writes. After a power-up reset condition, the ADC will come out of its reset state and commence with its internal calibration sequence. When this calibration sequence has finished, the ADC will be ready to use for conversions.

For a detailed description of the CoreAI configuration parameters, refer to the *CoreAI handbook*.

The designer must configure the Analog Quads appropriately via the ACM interface to match the required design-specific voltage, current, and temperature ranges prior to performing ADC conversions; failure to do so may result in damage to the Fusion device. The SmartGen or CoreConsole software can be used to configure the Analog Quads and RTC.

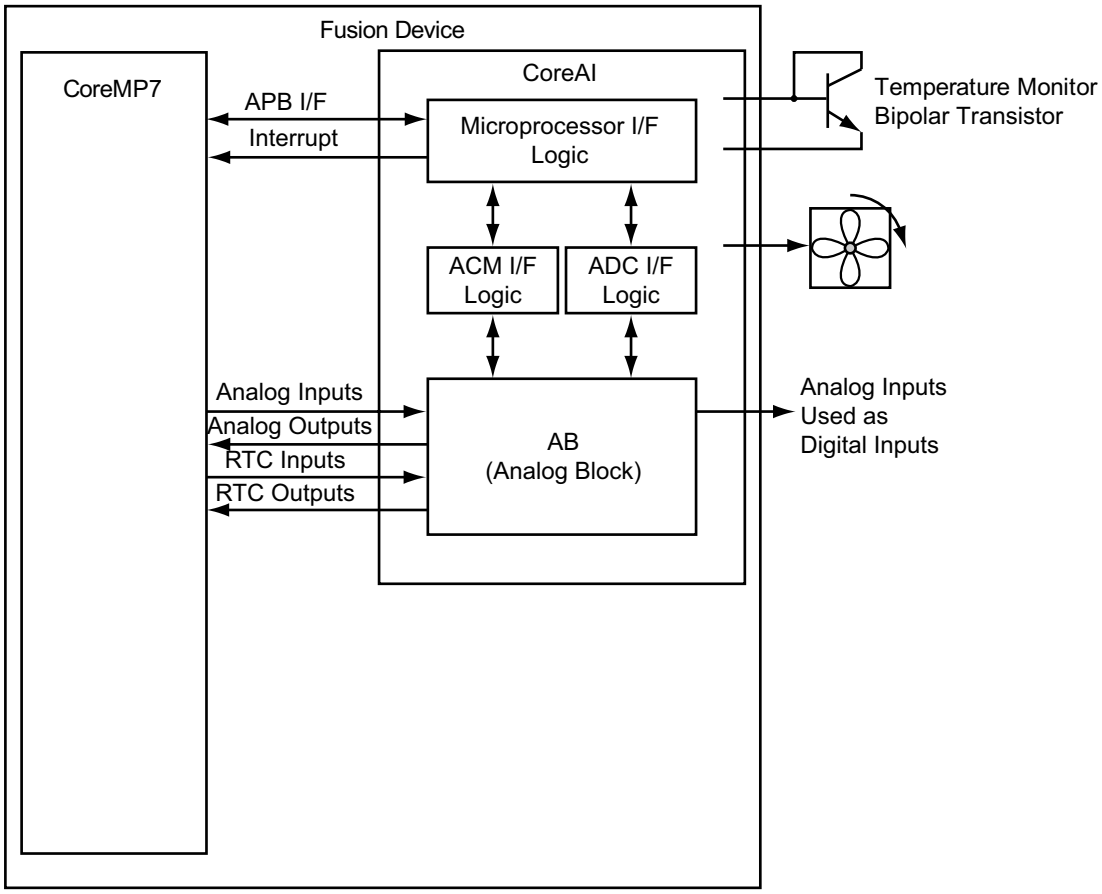

***Figure 1* • CoreAI Block Diagram**

Implementation: CoreAI Using CoreConsole

CoreAI is fully supported by CoreConsole v1.1 and Microsemi Libero® System on Chip (SoC) software Integrated Design Environment (IDE) v7.1 and later software releases. The designer has the option of configuring CoreAI through the CoreConsole GUI, as shown in Figure 2 on page 3. Alternatively, CoreConsole has an **Import** button that allows the designer to import settings from a SmartGen-configured macro. Note that a SmartGen AB configuration macro and a CoreConsole CoreAI GUI configuration cannot be used simultaneously in a design.

If you choose to import SmartGen settings into CoreConsole, Microsemi recommends that you invoke SmartGen outside of Libero IDE and use a separate SmartGen workspace to generate the CCF configuration file used for importation.

For more information on the CoreConsole IP Deployment Platform, refer to the *CoreConsole User's Guide*.

The implementation of CoreAI alongside a microprocessor allows the Fusion AB to be located within the microprocessor memory map. The Fusion AB can then be directly accessed and controlled with the processor's standard LOAD and STORE instructions to and from CoreAI. The CoreAI interface provides a traditional interface to an AB that is common among discrete microcontrollers and embedded system

programmers. Further, CoreAI implements a register file for complete control of the Fusion AB (note that some registers are already contained in the AB macro; CoreAI implements only the missing registers for use with a microprocessor).

CoreAI also implements a configurable-depth FIFO for storing (or pipelining) up to 256 ADC results. When instantiated, the FIFO utilizes the dedicated hardware found on ProASIC®3/E and Fusion devices to keep the number of VersaTiles to a minimum. The FIFO supports continuous sampling through the analog interface without requiring the result to be read until the FIFO has been completely filled.

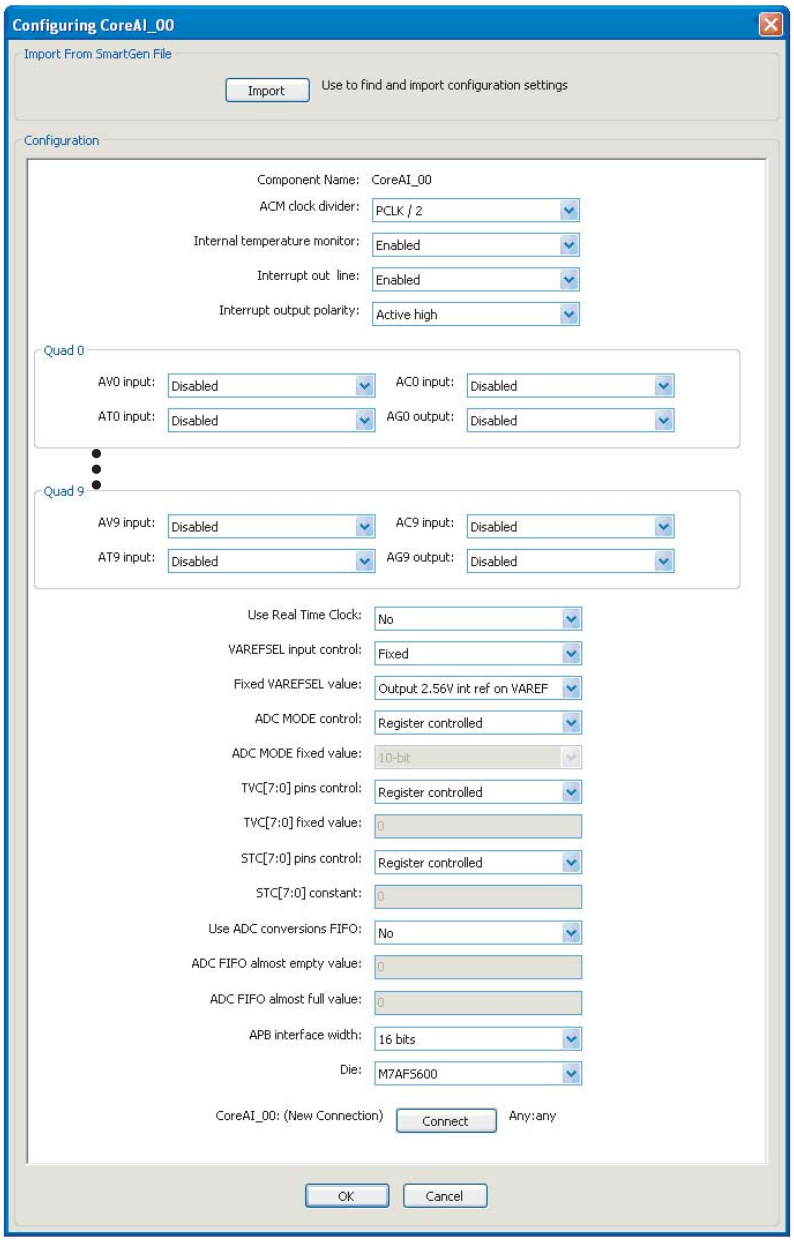

Figure 2 • **CoreConsole CoreAI Configuration Dialog**

CoreAI also provides the necessary signals to implement an interrupt-driven system; it is capable of triggering an interrupt from up to fourteen unique events. However, when an event occurs, the designer must read the CoreAI interrupt status register to determine which event triggered the interrupt.

A limitation of CoreAI is that the microprocessor is responsible for managing the analog conversions. The microprocessor must sequence, read/store results, average results (if desired), etc. The required management of the AB may use a larger amount of processing time, depending upon the level of management required. This is not the case with the SmartGen approach, which is discussed in the "SmartGen".

The implementation size of CoreAI ranges from 150 VersaTiles for the minimum configuration to 460 for the maximum configuration. The maximum configuration also consumes one of the dedicated hardware FIFOs.

SmartGen

The SmartGen Analog System Builder enables you to configure and instantiate the Fusion AB. It is similar to the CoreConsole CoreAI Configuration dialog. Within the Analog System Builder (Figure 3), you can do the following:

- Choose the number of analog input channels to monitor
- Choose the type of each input channel
- Choose the number of analog output channels
- Specify the placement of each channel
- Set channel-specific options
- Sequence the channels in the desired sampling order
- Specify the RTC settings

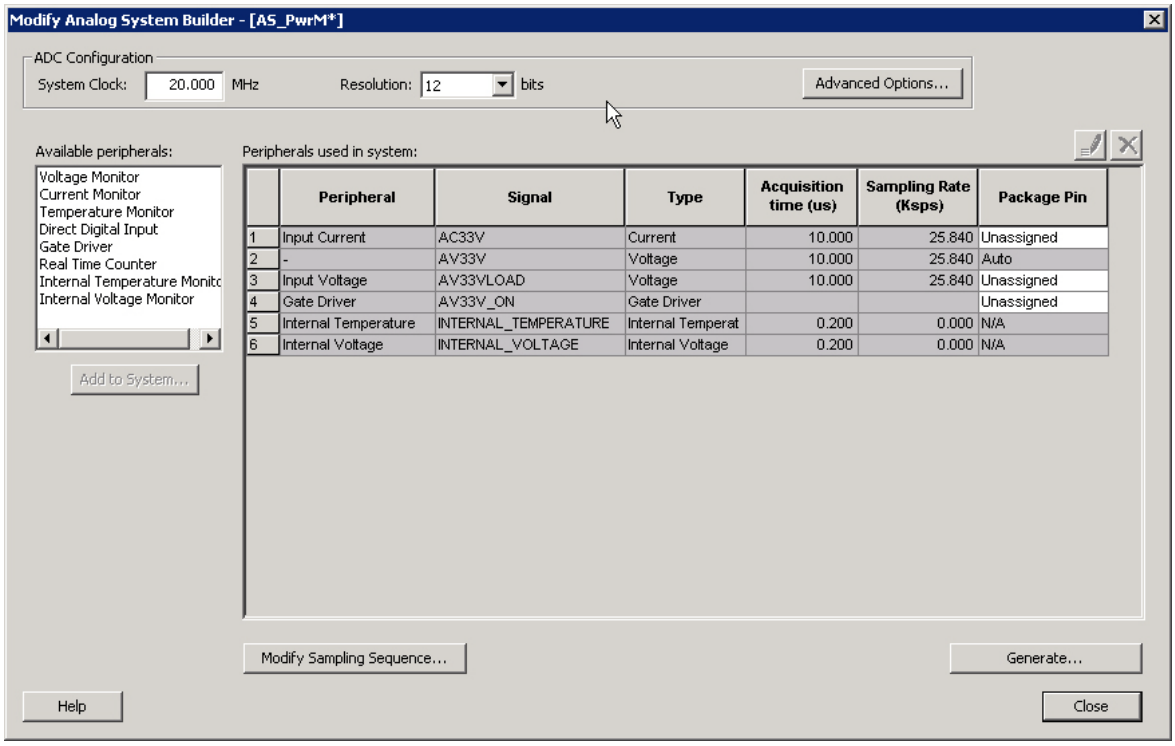

***Figure 3* • SmartGen Analog System Builder Dialog**

With these options, the Analog System Builder can be used to create, configure, and place the following analog peripherals: internal/external voltage, current, and temperature monitors; an RTC; gate drivers; and unused analog input pins as digital inputs.

The typical SmartGen Soft IP block implements the Analog Sample Sequence Controller (ASSC), System Monitor Evaluation (SMEV), and System Monitor Transition (SMTR) using VersaTiles and RAM.

The SMEV is responsible for evaluating the converted analog data, while the SMTR processes the converted data and generates flags conditionally. The SmartGen Soft IP block is typically configured using the INIT/CFG client. The client has a maximum speed of 10 MHz, but once configured, the system can operate at a much higher frequency. The No-Glitch MUX (NGMUX) is utilized to configure the system using a slow clock and switch to a high-speed clock once the configuration has completed through the CALIBRATION signal generated by the Soft IP block.

For more information on SmartGen and the Analog System Builder, refer to the SmartGen, FlashROM, ASB, and Flash Memory System Builder online help.

Unlike the CoreAI solution, SmartGen implements all management of the Fusion AB with FPGA VersaTiles. This implementation offloads the burden of processor management and frees up valuable CPU cycles for other processing requirements. Differing levels of processor management can be achieved with the Analog System Builder by using of the processor's General Purpose I/O (GPIO) ports; this is discussed in the "Implementation: Microprocessor Monitoring of SmartGen Subsystem" section, the "Implementation: Partial Microprocessor Control of SmartGen Subsystem" section on page 7, and the "Implementation: Microprocessor Access to ACM with SmartGen Subsystem" section on page 8.

The SmartGen Soft IP solution is significantly larger, in terms of VersaTile utilization, than a CoreAI implementation. The SmartGen VersaTile count is directly proportional to the implementation of the ASSC and SMTR. A larger number of scheduled sequences and a greater number of flags (signals) triggered by the SMTR significantly increase the VersaTile count.

Implementation: Microprocessor Monitoring of SmartGen Subsystem

This implementation is the simplest form of the inter operation of the two systems. The SmartGen Fusion subsystem runs independently of the microprocessor subsystem. The microprocessor subsystem simply monitors (or only reads) the ADC results and, if desired, the threshold flags. The SmartGen subsystem is configured through the embedded Flash memory using the standard INIT/CFG interface, as a standalone SmartGen system would be (Figure 4 on page 6).

The two systems may be clocked separately, as the microprocessor accesses the ADC results asynchronously.

In this scenario, the microprocessor monitors the BUSY signal generated by the SmartGen Soft IP subsystem. When the BUSY signal is not active, the microprocessor is free to access the processed ADC results via the SMTR. Synchronization logic can be implemented between the processor's GPIO interface and the SmartGen Soft IP block, if desired, to alleviate any timing issues.

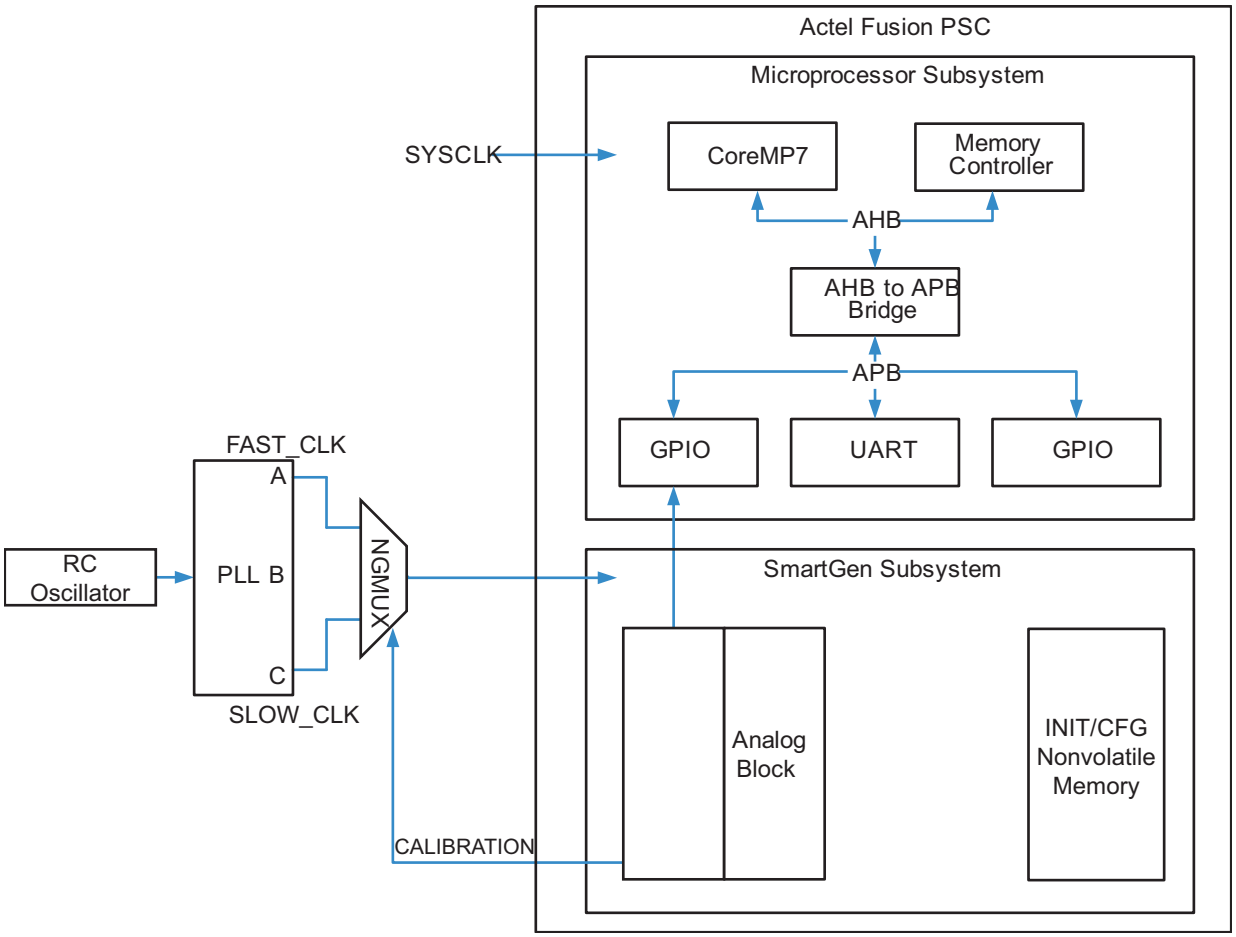

***Figure 4* • Microprocessor Monitoring of SmartGen Subsystem**

Implementation: Partial Microprocessor Control of SmartGen Subsystem

This implementation is slightly more complex than the previous two systems; however, it is similar to a CoreAI system. The SMEV and SMTR have been removed from the Fusion Analog System and are implemented in the microprocessor application. The ASSC remains for sequencing.

In this implementation, the processor's GPIO block is connected to the ASSC interface. The GPIO block simply triggers the ADC to begin a conversion. Once a conversion has been completed, the processor then reads the results. This interface is bidirectional. Due to the nature of the interaction between the two systems, Microsemi recommends that this type of implementation be configured synchronously, with the processor system clock clocking the SmartGen subsystem (Figure 5).

The two systems must be synchronous because the processor is partially controlling the SmartGen Subsystem, which is driven by the processor's clock output (HCLK in CoreMP7).

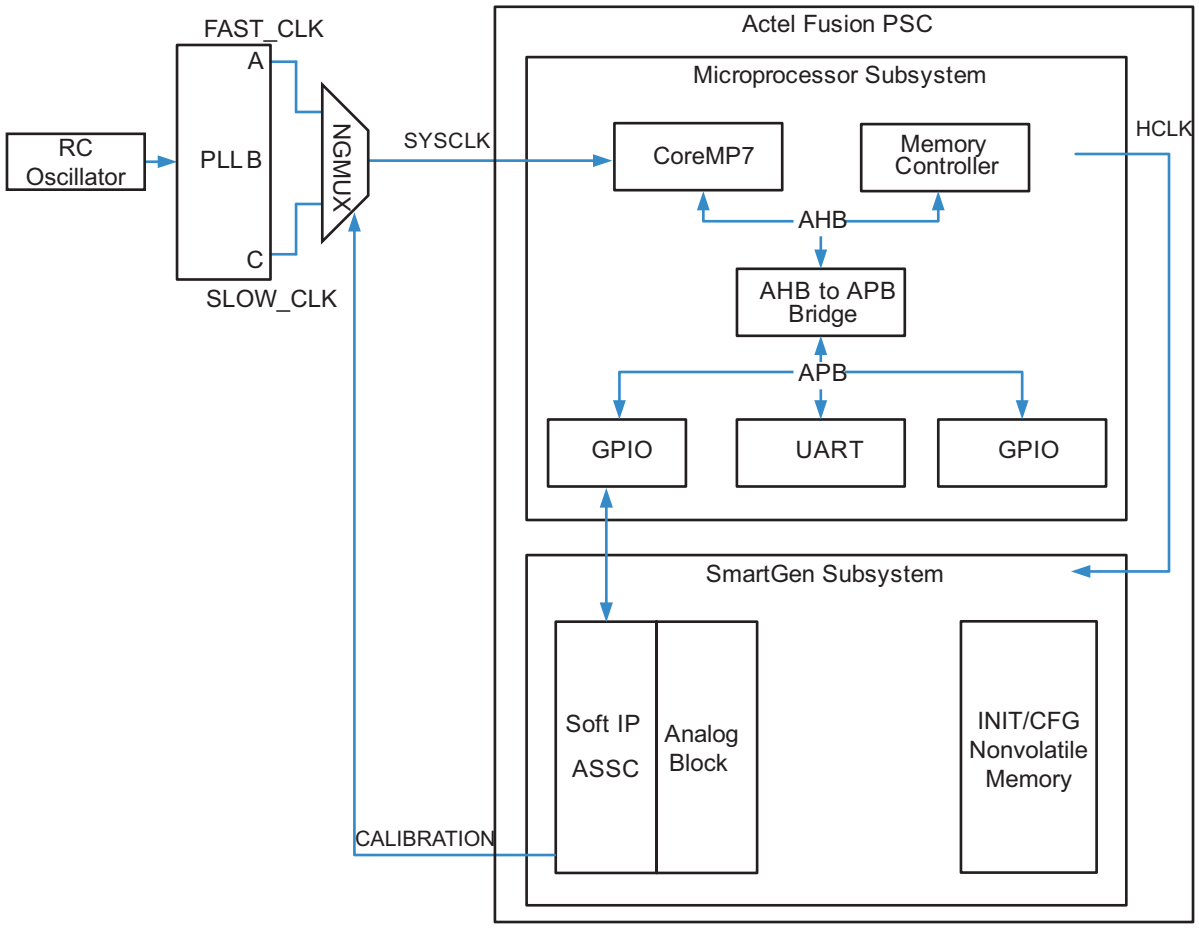

***Figure 5* • Partial Microprocessor Control of SmartGen Subsystem**

Implementation: Microprocessor Access to ACM with SmartGen Subsystem

The final SmartGen implementation is similar to the first configuration; however, the INIT/CFG interface has been removed and has become the responsibility of the embedded microprocessor. The SMEV and SMTR can be implemented in VersaTiles or handled in the processor, as in the "Implementation: Microprocessor Monitoring of SmartGen Subsystem" section. Likewise, the "Implementation: Microprocessor Monitoring of SmartGen Subsystem" section can easily be used in conjunction with this implementation.

In this implementation, the processor's GPIO block is directly connected to the ACM. The processor is responsible for configuring the ACM and generating the CALIBRATION signal once the configuration has completed.

The PLL is used to generate a dedicated clock signal of less than 10 MHz for the ACM, as the MUX cannot be clocked faster than 10 MHz (Figure 6). Typically, this is handled automatically (as it is built into the SmartGen hardware) for other implementations; however, since the processor is configuring the ACM, it is a design concern.

The two systems must be synchronous, as the processor is potentially controlling the SmartGen Subsystem, which is driven by the processor's clock output (HCLK in CoreMP7). A synchronization block is also needed to synchronize the processor with the ACM, as the ACM is most likely asynchronous to the rest of the system.

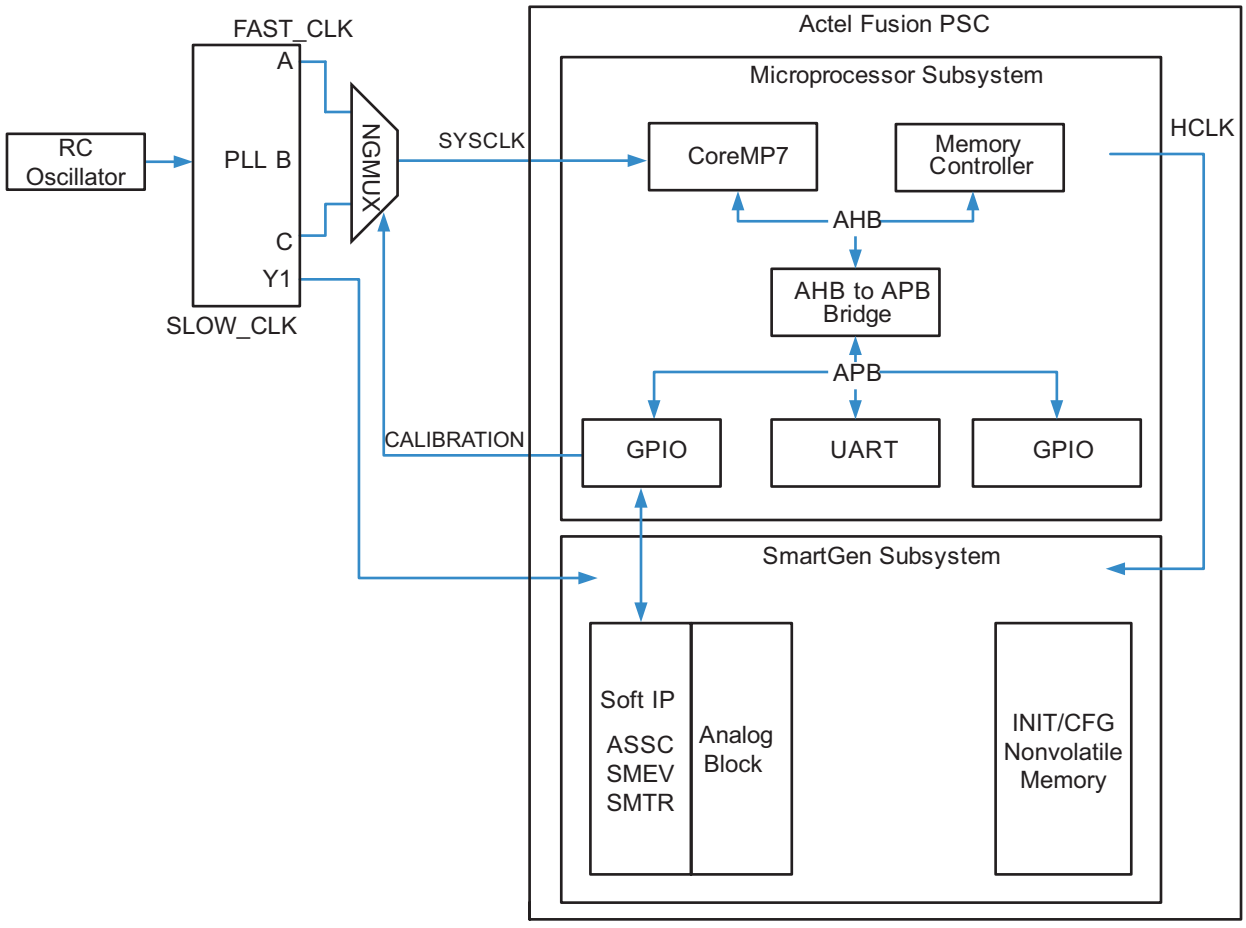

***Figure 6* • Microprocessor Access to ACM with SmartGen Subsystem**

Conclusion

Both the CoreAI and SmartGen implementations have their advantages (and disadvantages). If your design in a Fusion PSC is running low on available VersaTiles, the CoreAI solution has a smaller footprint, but at the price of increased processor overhead. If your processor is performing critical calculations and you have the space, the SmartGen implementation may be the best choice. If you are familiar with programming traditional microcontrollers, the CoreAI implementation creates a simple interface that treats the AB as a processor peripheral, which is common among discrete microcontrollers. If you have an existing design based on the Fusion PSC, you are already familiar with the SmartGen Analog System Builder and may be more comfortable with the SmartGen approach. Each system has its benefits, and it is up to the designer to determine which implementation is correct for a system.

Related Documents

Datasheets

Fusion Family of Mixed-Signal Flash FPGAs

http://www.microsemi.com/documents/Fusion_DS.pdf

CoreAI

http://www.microsemi.com/ipdocs/CoreAI_HB.pdf

User's Guides

CoreConsole User's Guide

http://www.microsemi.com/documents/CoreConsole_UG.pdf

SmartGen, FlashROM, ASB, and Flash Memory System Builder User's Guide or online help

http://www.microsemi.com/documents/genguide_ug.pdf

List of Changes

The following table lists critical changes that were made in each revision of the document.

Revision	Changes	Page
Revision 1 (September 2015)	Non-Technical Updates.	N/A
Revision 0 (June 2006)	Initial Release.	N/A

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

E-mail: sales.support@microsemi.com

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; security technologies and scalable Ethernet Solutions;anti-tamper products; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 3,600 employees globally. Learn more at **www.microsemi.com**.

51900142-1/09.15