SmartFusion Programmable Analog

User's Guide

Microsemi

Table of Contents

1 Analog Compute Engine (ACE) 5
ACE Architecture and Interfaces 5
Brief Description of SSE and PPE Architectures 11
PPE Architecture and Functions 13
Dedicated FIFO for PDMA Data Transfers 15
Interrupts 17

2 SmartFusion Analog Front-End (AFE) Overview 21
SmartFusion Analog Features 21
SmartFusion Analog Front-End General Description 21
Analog Front-End Configuration 24
Related Information 26

3 Analog-to-Digital Converter (ADC) 27
Key Features 27
Block Diagram 27
ADC Theory of Operation 28
ADC Terminology 30
SmartFusion ADC Operation 35
ADC Start-Up Sequence 41
ADC Configuration Example 42
ACE Firmware Driver 42
ADC Register Map 42

4 Sigma-Delta Digital-to-Analog Converter (DAC) 49
Sigma-Delta DAC (SDD) Features 49
Sigma-Delta DAC General Description 49
The Sigma-Delta Principle 50
Sigma-Delta DAC Detailed Description 52
Time-Domain and Frequency-Domain Interpretation 56
Sigma-Delta DAC Configuration 60
Related Information 66

5 Active Bipolar Prescaler (ABPS) 67
ABPS Features 67
ABPS General Description 67
ABPS Detailed Description 68
ABPS Configuration 69
Related Information 70

6 Current Monitor 71
Current Monitor Features 71
Current Monitor General Description 71
Current Monitor Detailed Description 72
Current Monitor Configuration 76

Related Information . . . 79

7 Temperature Monitor . . . 81
Temperature Monitor Features . . . 81
Temperature Monitor General Description . . . 81
Temperature Monitor Detailed Description . . . 82
Temperature Monitor Configuration . . . 87
Related Information . . . 89

8 High-Speed Comparators . . . 91
Comparator Features . . . 91
Comparator General Description . . . 91
Comparator Detailed Description . . . 92
Comparator Configuration . . . 94
Comparator Outputs and Interrupts . . . 97
Comparator Application – Using One SDD to Set Multiple Comparator Reference Voltages . . . 99
Related Information . . . 101

9 Direct ADC Inputs . . . 103
Direct ADC Input Features . . . 103
Direct ADC Input General Description . . . 103
Direct ADC Input Detailed Description . . . 104
Direct ADC Input Configuration . . . 107
Related Information . . . 110

10 ACE Interrupts . . . 111
ACE Interrupts Routed to the Cortex-M3 Processor . . . 111
ACE Interrupts/Flags Routed to FPGA Fabric . . . 122

A List of Changes . . . 125

B Product Support . . . 129
Customer Service . . . 129
Customer Technical Support Center . . . 129
Technical Support . . . 129
Website . . . 129
Contacting the Customer Technical Support Center . . . 129
ITAR Technical Support . . . 130

1 – Analog Compute Engine (ACE)

ACE Architecture and Interfaces

The analog compute engine (ACE) is made of two major functional blocks: the sample sequencing engine (SSE) and the post-processing engine (PPE), which are small microcontrollers A detailed block diagram in shown in Figure 1-1.

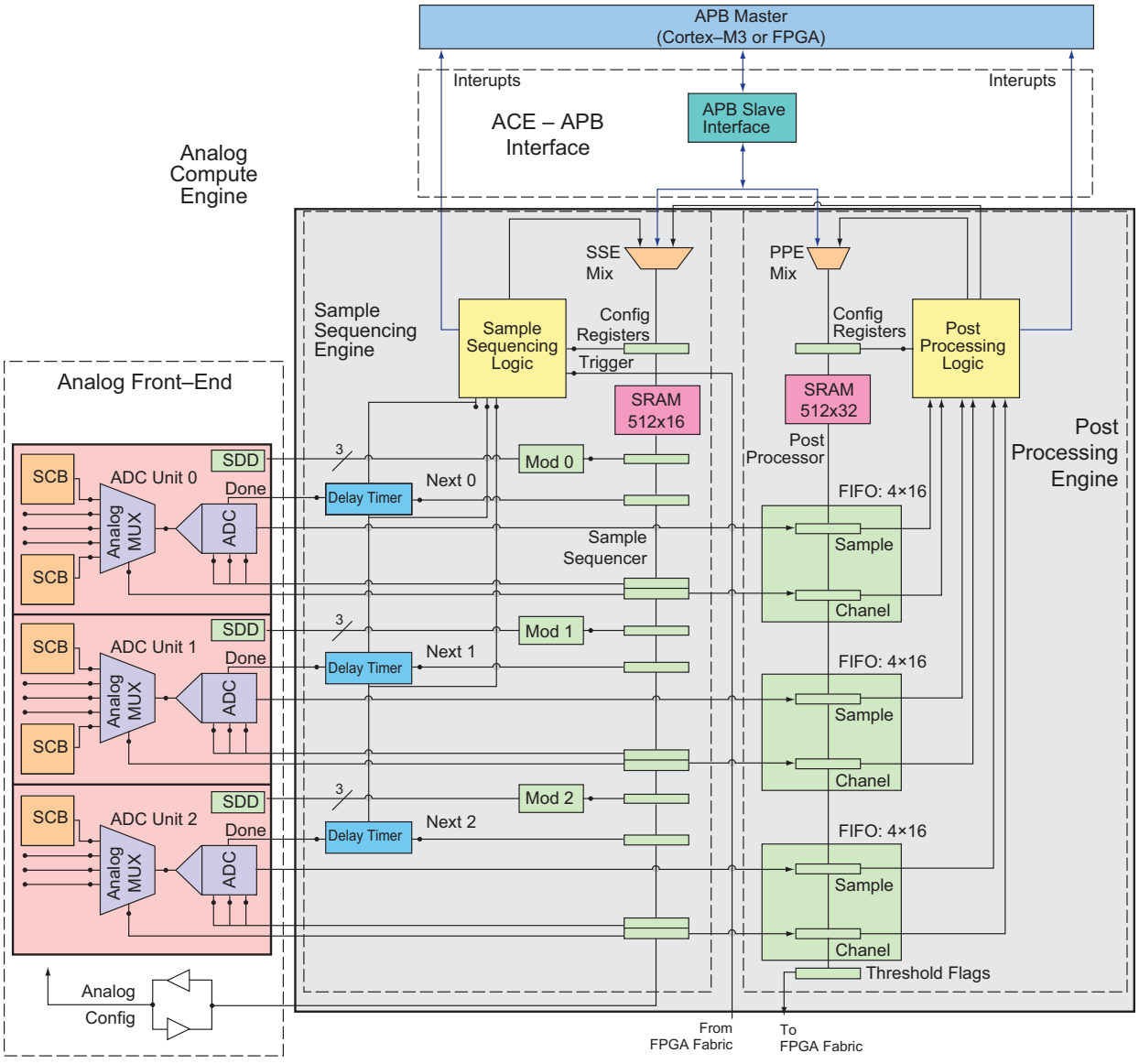


***Figure 1-1* • Analog Compute Engine (A2F200 shown)**

The ACE interfaces with the microcontroller subsystem (MSS) through the APB bus, the FPGA core, and the analog resources, as depicted in the simplified block diagram shown in Figure 1-2. Notice that the ACE is a slave of the APB bus, and can have the ARM® Cortex™-M3 processor, logic in the FPGA fabric, or the PDMA engine as master.

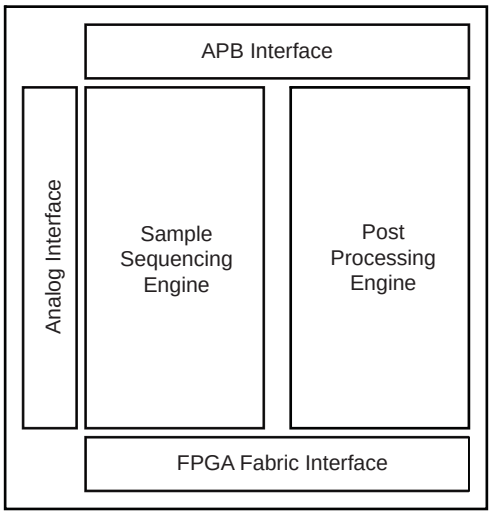


***Figure 1-2* • Overview of ACE Architecture and Interfaces**

This chapter includes a brief introduction to SSE architecture and functions. The SSE is the part of the ACE that allows flexible configuration of the analog front-end (AFE) resources (analog inputs, monitors, comparators, ADCs, and DACs) as well as a variety of simple and sophisticated sampling sequencing. The PPE, also introduced in this chapter, is a self sufficient block allowing data processing such as linear transformation, filtering, and thresholds comparisons.

ACE Interface to the Microcontroller Subsystem

The ACE is a slave on the APB bus and as such it is a simple peripheral to the master of the APB (Cortex-M3, FPGA fabric, or PDMA) with a specific static address. The base address for the ACE peripheral is 0x40020000”.

The APB bus accesses the ACE through two multiplexers: one dedicated to the SSE and one to the PPE, called the SSE MUX and PPE MUX (Figure 1-3). The SSE and PPE MUXes provide access to all the analog block registers from the APB masters as well as the SSE and PPE. Additionally, they handle the arbitration between these sources (three in the case of the SSE MUX and two in the case of the PPE MUX)..

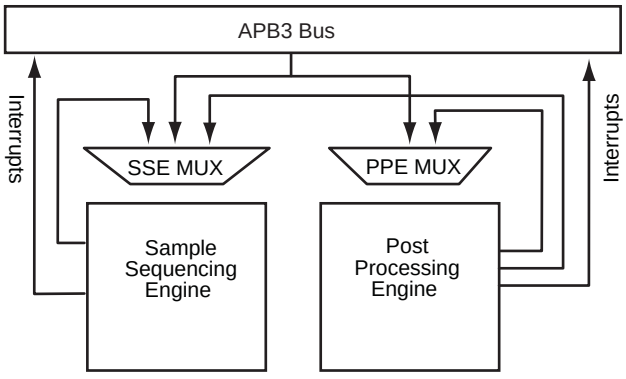


***Figure 1-3* • Sketch of the ACE (SSE and PPE) interface to the MSS**

The priority of access through the multiplexers is given to the ACE (SSE and PPE). This guarantees precise sample timing; however, priority can be given to APB bus master. This can be done through firmware by controlling appropriate bits in SSE_TS_CTRL and PPE_CTRL registers. In this case, the APB master will need to take care to reinitialize any sequences that may have been running prior to direct APB master control using ACE drivers such as SSE_stop, SSE_load, and SSE_restart, defined in the *SmartFusion ACE Driver User's Guide.*

***Table 1-1* • SSE and PPE Control Registers**

Register Name	Address	R/W	Reset Value	Description
SSE_TS_CTRL	0x40020004	R/W	0	SSE time slot control
PPE_CTRL	0x40021404	R/W	0x2	PPE control

SSE_TS_CTRL Register Bitwise Details

***Table 1-2* • SSE_TS_CTRL**

Bit Number	Name	R/W	Reset Value	Description
[7:2]	Reserved			Reserved
1	SSE_SRAM_ENABLE	R/W	0	1– Enable SSE SRAM 0 –Disable SSE SRAM
0	TS_ENABLE	R/W	0	1 – Enable all four time slots. This must be set for normal operation, which allows equal round-robin access every eighth PCLK cycle for the APB master, ADC0, ADC1, and ADC2. 0 – Enables complete access to APB3 master only—this fixes the PREADY signal High to allow zero wait state access to the APB master. This is useful during initialization time after a reset/power-up condition when the APB3 master needs to configure the analog quads, the internal SSE registers, and the internal SSE RAM sequences.

PPE_CTRL Register Bitwise Details

***Table 1-3* • PPE_CTRL**

Name	R/W	Reset Value	Description
Reserved	R/W	0	Reserved
RRDIS2	R/W	0	1 – PPE will empty out the entire result FIFO associated with ADC2 (if it is not empty) before moving back to process FIFO associated with ADC0 in round-robin fashion. 0 – PPE will read one word from FIFO associated with ADC2 (if it is not empty), perform any required post-processing activities associated with the result value linked to the analog channel, then move back to process the FIFO associated with ADC0 in round-robin fashion.
RRDIS1	R/W	0	1 – PPE will empty out the entire result FIFO associated with ADC1 (if it is not empty) before moving back to process FIFO associated with ADC2 in round-robin fashion. 0 – PPE will read one word from FIFO associated with ADC1 (if it is not empty), perform any required post-processing activities associated with the result value linked to the analog channel, then move back to process the FIFO associated with ADC2 in round-robin fashion.
RRDIS0	R/W	0	1 – PPE will empty out the entire result FIFO associated with ADC0 (if it is not empty) before moving back to process FIFO associated with ADC1 in round-robin fashion. 0 – PPE will read one word from FIFO associated with ADC0 (if it is not empty), perform any required post-processing activities associated with the result value linked to the analog channel, then move back to process the FIFO associated with ADC1 in round-robin fashion.
CURRCHAN	R/W	0	Current ADC channel being processed (read-only bits): 000000 – ADC 0, channel 0 … 001111 – ADC 0, channel 15 010000 – ADC 1, channel 0 … 011111 – ADC 1, channel 15 100000 – ADC 2, channel 0 … 101111 – ADC 2, channel 15 110000 – Reserved … 111111 – Reserved
Ci	R/W	0	1 – Set carry in bit to 32-bit adder in PPE ALU to 1. 0 – Clear carry in bit to 32-bit adder in PPE ALU to 0.
Reserved	R/W	0	Reserved
C2d	R/W	0	C register to multiplier d input (write-only): This bit, when a 1 is written, is used to transfer the upper 16 bits of the PPE ALU C[31:0] register to the d[15:0] input to the PPE ALU signed multiplier. When a 0 is written (the normal state of this bit), the D[15:0] register is connected to the d[15:0] input of the multiplier. This write-only bit clears itself after 1 PCLK cycle.

***Table 1-3* • PPE_CTRL (continued)**

Name	R/W	Reset Value	Description
s2B	R/W	0	Adder sum output s to B register (write-only). This bit, when a 1 is written, is used to transfer the PPE ALU adder s[31:0] sum output to the B[31:0] register. When a 0 is written (the normal state of this bit), the PPE ALU multiplier p[31:0] signed output is transferred to the PPE ALU B[31:0] register when updates to the multiplier occur. This write-only bit clears itself after 1 PCLK cycle.
C2a	R/W	0	C register to Adder a input: This bit, when a 1 is written, is used to transfer the PPE ALU C[31:0] register to the a[31:0] input of the PPE ALU adder. When a 0 is written to this bit, either the A[31:0] register or the complement of the A[31:0] register is connected to the a[31:0] input of the PPE ALU adder, depending on the state of the NegA bit in this register.
NegA	R/W	0	Negate A register: This bit, when a 1 is written, selects the negation (one's complement) of the PPE ALU A[31:0] register to connect to the a[31:0] input of the PPE ALU adder, depending on the state of the C2a bit in this register. When a 0 is written to this bit, the A[31:0] register is connected to the a[31:0] input of the PPE ALU adder, depending on the state of the C2a bit in this register.
PPE_IDLE	R/W	1	PPE Idle status (read-only). This read-only bit, if 1, denotes that the PPE is not busy processing. If this bit is 0, the PPE is busy processing.
PPE_EN	R/W	0	PPE Enable (writable only from APB master): 1 – Enable PPE processing 0 – Disable PPE processing

These multiplexers and their use as a data routing gate allow the resolution of conflicting accesses to the ACE registers and RAMs.

Both the SSE and the PPE generate interrupts that are routed to the Cortex-M3 processor or the FPGA logic. The firmware running on the Cortex-M3 or the logic must consider these, mask them temporarily in a preemptive mechanism, or completely ignore them. The PPE generates threshold related interrupts, whereas the SSE generates interrupts related to general purpose SSE events, ADC conversion and calibration events, and comparator events. For details on the interrupts, refer to the "ACE Interrupts" section on page 111.

ACE Interface to the Analog Front-End

The main interface of the ACE to the analog front-end occurs at the SSE level. Configuration and control data for the individual analog functions is stored in registers local to the functions. Access from the sequencer to these registers is accomplished over the analog configuration bus (ACB).

ACE Interfaces to the FPGA Fabric

The ACE has three groups of interfaces to the FPGA fabric: the interface to the sigma-delta DAC (SDD), the PPE threshold event flags, and the FPGA trigger.

Table 1-4 provides a list of fabric interface signals.

***Table 1-4* • Signals Interfacing Between ACE/AFE and FPGA Fabric**

Name	Input To/From FPGA Fabric	ACE/AFE Destination	Function
FABACETRIG	Output	SSE	Trigger input from FPGA fabric that can start SSE sequencing
FABSDD0D	Output	SDD0	Data out from FPGA fabric driving data input to SDD0
FABSDD0CLK	Output	SDD0	Clock out from FPGA fabric driving clock input to SDD0
FABSDD1D	Output	SDD1	Data out from FPGA fabric driving data input to SDD1
FABSDD1CLK	Output	SDD1	Clock out from FPGA fabric driving clock input to SDD1
FABSDD2D	Output	SDD2	Data out from FPGA fabric driving data input to SDD2
FABSDD2CLK	Output	SDD1	Clock out from FPGA fabric driving clock input to SDD2

The FPGA trigger, FABACETRIG, is an input from the FPGA fabric to the sample sequencing logic (part of the SSE) that launches a sequence of analog data sampling.

The interface signals to the SDD are the basic data inputs and clocks to the DAC from logic in the FPGA fabric. Refer to the "Sigma-Delta Digital-to-Analog Converter (DAC)" section on page 49 for further details.

The threshold event flags are generated by the PPE logic and considered by the FPGA fabric logic. These 32 flags are associated with the digital comparison of the ADC results and 10 of them can optionally allow the FPGA fabric logic to monitor internal ACE signals. For details on these flags, refer to the "ACE Interrupts" section on page 111.

Brief Description of SSE and PPE Architectures

This section provides a quick overview of the architectures and the main functions of the SSE and PPE blocks of the ACE. Intimate knowledge of the details of the PPE and SSE blocks are unnecessary due to the GUI-based configuration provided by the SmartDesign MSS configurator and the software drivers that are provided.

SSE Architecture and Functions

Overview of the SSE Architecture

The SSE presents itself as an APB slave on the APB bus, thus allowing an APB master such as the MSS Cortex-M3 processor or an FPGA fabric master to access and control it. The SSE directly interfaces with the various ADCs available, the analog blocks on one side, and the PPE on the other side. The SSE implements a time division multiplexing (TDM)-based mechanism to sequence through ADCs. The SSE uses separate program counters (PC) for sequencing and controlling each ADC independently. To avoid starving the APB master from access to the SSE and program counter, the architecture allocates one in four clock cycles to the APB master.

Figure 1-4 provides an overview of the SSE internal architecture and its interfaces to the PPE, APB3, ADCs, DACs, and ACB.

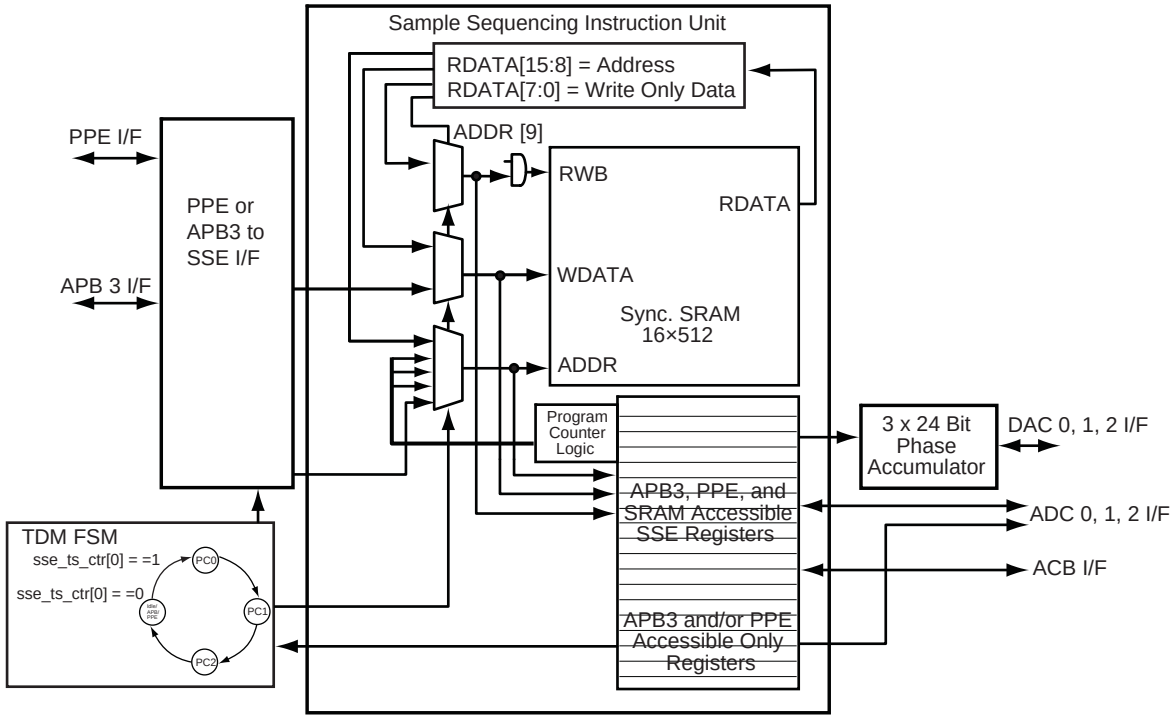


***Figure 1-4* • General Overview of SSE Architecture and Interfaces**

Very briefly, the overall architecture shows four major blocks of the SSE:

1. Sample sequencing instruction unit, which provides multiplexing access to/from a program memory (SRAM) and analog configuration registers.
2. Time division multiplexing FSM, which creates time slots for each of the three ADCs, as well as a shared time slot for APB3 or PPE accesses. ADC time slots are allocated during the configuration of the analog monitors configurator in the SmartDesign MSS configurator.
3. APB/PPE interface, which arbitrates sample sequencing engine access between the APB and PPE.

4. 24-bit phase accumulators, which are the front-end accumulators for the three SDDs.

Note that when the TDM time slot counter is enabled for normal operation of the SSE block (refer to the SSE_TS_CTRL register and Figure 1-5), there are always four time slots, regardless of whether there are one, two, or three ADC instances in a specific device.

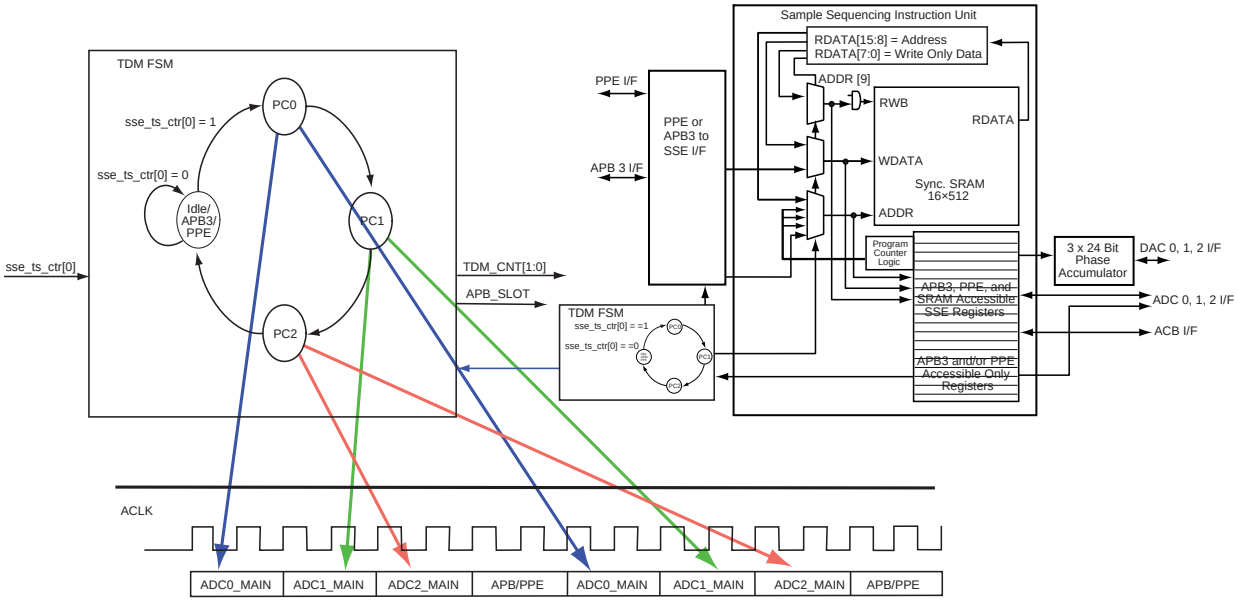


***Figure 1-5* • SSE Micro Architecture**

The TDM FSM can be controlled by the FABACETRIG signal, coming from the FPGA fabric or SSE_TS_CTRL register, which can be written by the APB Master.

- When bit 0 of SSE_TS_CTRL or FABACETRIG is set to 1, all 4 time slots are enabled.
- When bit 0 of SSE_TS_CTRL or FABACETRIG is set to 0, only access to the APB master is enabled—this fixes the PREADY signal High to allow zero wait state access to the APB master.

The SmartFusion® customizable system-on-chip (cSoC) also has the capability of simultaneous sampling on different ADCs. The simultaneous synchronized ADC conversion control register (ADC_SYNC_CONV) allows for a single program to issue synchronized start instructions for all ADCs.

Refer to the appropriate application notes, such as *SmartFusion: ACE Sequencing Control Using Fabric and MSS,* for the specifics of implementation using FABACETRIG and simultaneous sampling.

SSE Main Functions

The SSE micro engine operates with custom micro codes to implement the following functions:

- Flexible ordering of channels to be sampled
- Ability to adjust the ADC resolution, sample time, and clock settings per sample
- Ability to simultaneously start ADC conversions on 1, 2, or 3 of the ADCs
- Ability to simultaneously update DAC data on 1, 2, or 3 DACs
- Periodic sequencing and repeatable, separate sequencing for each of the three ADCs
- Presenting the raw ADC data to the PDMA engine
- Trigger and accept PDMA input data (example: DAC input data)
- Allowing for external trigger from either external master or fabric master to control ADC sequencing
- Nested loop capability within a sequence
- Per channel maskable interrupts

- End of sequence, maskable interrupts

Access to all these functions is available when using the ACE drivers described in *SmartFusion MSS ACE Driver User's Guide*.

PPE Architecture and Functions

Overview of the PPE Architecture

The PPE block presents itself as an APB slave on the APB bus, allowing APB masters such as the hardwired Cortex-M3 processor of the MSS or an FPGA fabric master to access and control it. The PPE interfaces with the SSE and processes raw ADC data to provide filtered and processed data. The PPE generates threshold flags and flag interrupts for servicing by APB masters.

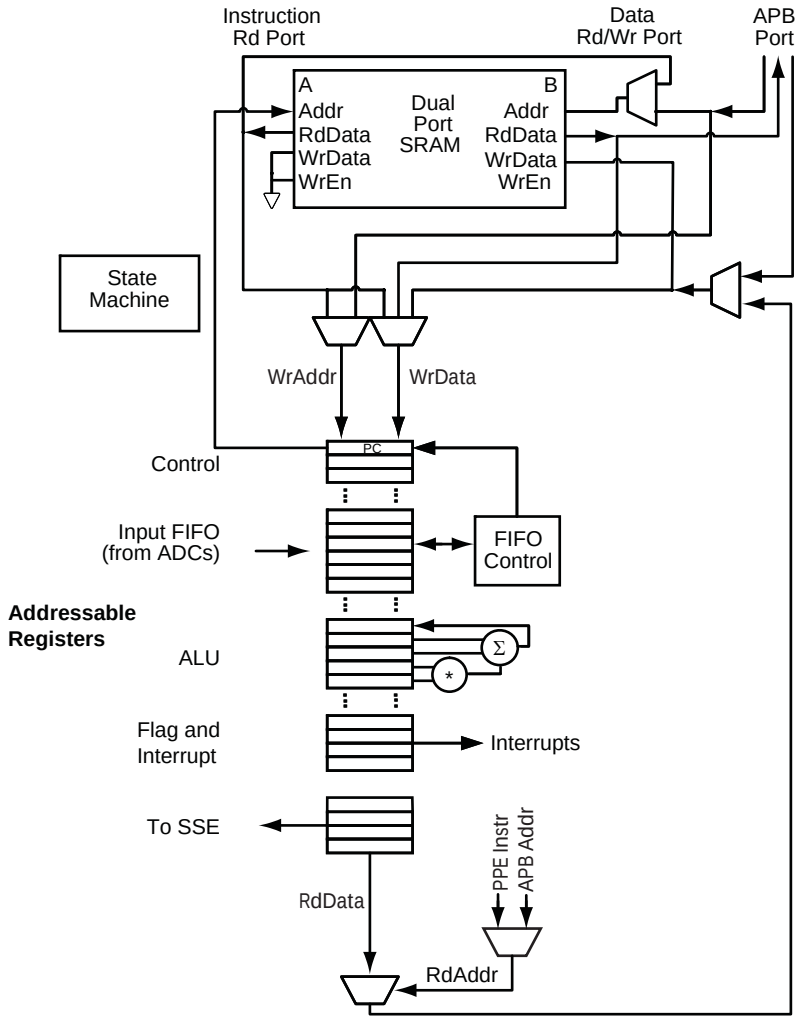


***Figure 1-6* • Overview of the PPE Architecture**

Figure 1-6 provides an overview of the PPE architecture. The main blocks included in the PPE are a set of FIFO blocks with their embedded FIFO controller, a state machine to service these FIFOs, an ALU, and a dedicated dual-port SRAM. The FIFOs are dedicated to each of the ADCs and are used by the PPE to process the raw data coming from the SSE.

The state machine operates in a round robin scheme to service these FIFOs, as depicted in Figure 1-7.

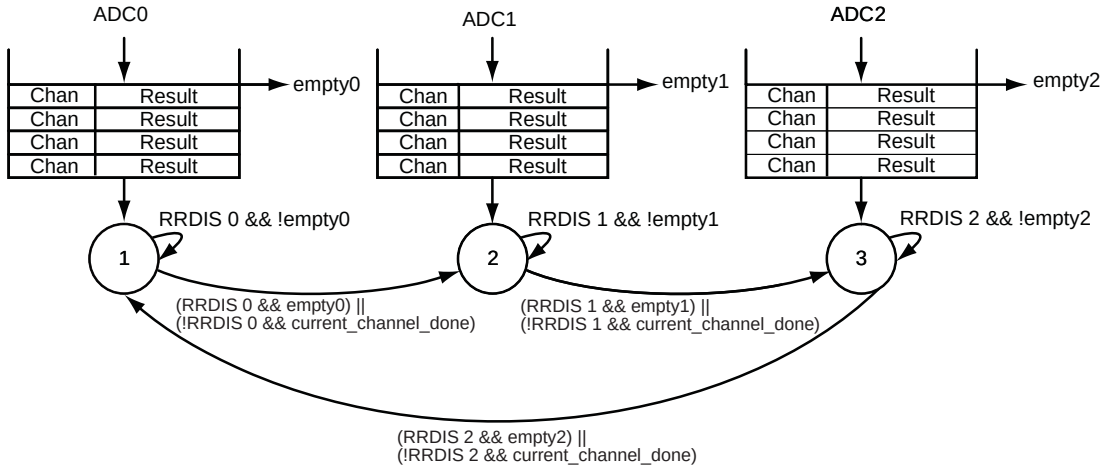


***Figure 1-7* • Round Robin Processing of the PPE FIFO Results**

The other important block in the PPE is the ALU, a fairly flexible block allowing users to perform various computations on the sampled captured data to implement calibration, threshold comparison, or other linear transformations. The main addressable registers of the ALU consist of three 32-bit registers—A, B, and C—and two 16-bit registers D and E (Table 1-5).

***Table 1-5* • Five ALU Working Registers**

Register Name	Address	R/W	Width	Reset Value	Description
A	0x4002120	R/W	32	0	PPE ALU Working Register A
B	0x4002128	R/W	32	0	PPE ALU Working Register B
C	0x4002130	R/W	32	0	PPE ALU Working Register C
D	0x4002138	R/W	16	0	PPE ALU Working Register D
E	0x4002140	R/W	16	0	PPE ALU Working Register E

Data movements to and from these five registers are accomplished via instructions that are stored in the PPE SRAM and are interpreted and acted upon by the internal PPE logic. Figure 1-8 on page 15 provides a sketch of the ALU architecture.

The ALU can be configured as a MAC (multiply accumulator) to implement low pass filtering (LPF) as well.

This built-in ALU of the PPE can be thought of as registers that are address mapped and can be accessed by the APB master for direct control and manipulation.

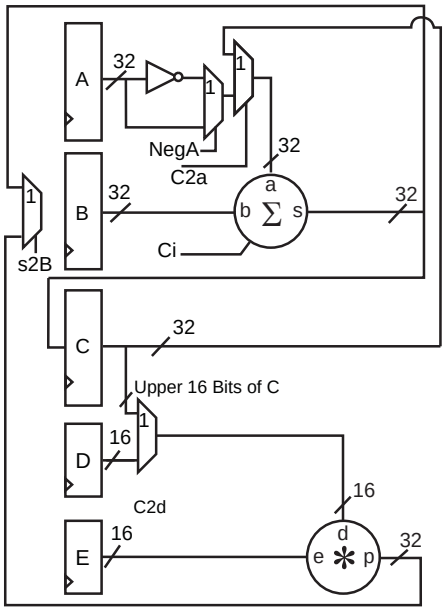


***Figure 1-8* • PPE ALU Architecture Overview**

PPE Main Functions

The PPE operates with custom micro codes that enable implementation of the following functions on the processed and filtered ADC data:

- Linear transformation, y = mx + b (including calibration)
- Digital low pass filtering threshold comparison
- State filtering
- Threshold hysteresis
- Presenting filtered and processed ADC data to PDMA engine

For details on these functions, refer to the *SmartDesign MSS ACE Configuration User's Guide*.

Dedicated FIFO for PDMA Data Transfers

The ACE has a built-in 32×32 FIFO that interfaces with the PDMA engine. Raw ADC data or PPE filtered data or PPE processed data is made available for the PDMA to transfer to other locations in the MSS or the FPGA fabric using this FIFO. This can be configured in SmartDesign MSS configurator. It is possible to PDMA one, two, or all three types of data. When more than one of the data types is used, the user logic must have intelligence to distinguish between the three types of data. There is no encoding available in the data to indicate the data type.

Table 1-6 explains decoding of the 32 bits of PDMA data.

***Table 1-6* • PDMA Data Decoding**

Bit Number	Name	Access	Description
31:30	Reserved	R/W	Reserved
29:28	ADC	R/W	ADC Number: 00 – ADC0 01 –ADC1 10 – ADC2 11 – Reserved
27:24	AMUXSEL	R/W	Binary encoded selection of 1 of 16 inputs connected to the analog multiplexer for ADC selected. This is also known as the logical channel number.
23:0	DATA	R/W	Analog data; can be raw ADC data, PPE filtered data, or PPE processed data. While maximum ADC resolution is 12 bits, the reason there are more bits provided is to account for averaging and transformation effects that require more magnitude or precision. These bits are the upper 24 bits of the 32-bit representation from the ALU in PPE. For typical scenarios, the 12 bits of ADC data will be in bits [19:8]. For instance, if the user directly passed raw unprocessed data using the DMA feature, the 12-bit ADC value would be in bits [19:8]. However, if performing a PPE computation such as averaging or a linear transformation, it is possible the data requires the extended bits. In short, this is application dependent.

Notes:

1. *There is no bit decode available to determine whether the data present is raw/filtered/processed data. The user logic must distinguish data among the three possible alternatives The order of data is always raw ADC data, PPE filtered data and PPE processing data. When more than one data type is chosen using the channel number as key, the user can rely on the order of data for post processing.*
2. *The PDMA status register PDMA_STATUS (at address 0x40021304) in the ACE contains the status of previously mentioned FIFO, such as Full, Input/output Data Ready. This is used to let the PDMA engine determine when to write/read PDMA data.*
3. *The PPE_PDMA_DATAOUT (at address 0x40021308) in the ACE contains the address decode of the PDMA FIFO. PDMA drivers have knowledge of this and the user should write application code using the supplied decode. Refer to the SmartFusion MSS PDMA Driver User's Guide for further details.*

Interrupts

To calculate how ACE interrupts are mapped to the Cortex-M3 processor, add 64 to the ACE interrupt number to determine the corresponding NVIC interrupt in the Cortex-M3. For example, ACE Interrupt 0 + 64 = Cortex-M3 interrupt 64 = INTISR[64].

The ACE has access to 86 interrupt sources to communicate with the Cortex-M3. Each interrupt source is selectable within the ACE configurator. The ACE driver provides the functions necessary to handle interrupts. Summary details of these interrupts are provided in Table 1-7 • ACE Interrupts. Refer to the "ACE Interrupts" section on page 111 for complete details on the interrupts.

***Table 1-7* • ACE Interrupts**

Name	ACE Interrupt	ACE Register / Address	ACE Register Address	Cortex-M3 Interrupt	Function
ACE / General Purpose SSE Events					
ACE_PC0_FLAG0_IRQ	0	SSE_IRQ 0x40021204	0x40021204	INTISR[64]	Program Counter 0 flags: These four general-purpose flags can be used by the SSE to create interrupt events.
ACE_PC0_FLAG1_IRQ	1			INTISR[65]	
ACE_PC0_FLAG2_IRQ	2			INTISR[66]	
ACE_PC0_FLAG3_IRQ	3			INTISR[67]	
ACE_PC1_FLAG0_IRQ	4			INTISR[68]	Program Counter 1 flags: These four general-purpose flags can be used by the SSE to create interrupt events.
ACE_PC1_FLAG1_IRQ	5			INTISR[69]	
ACE_PC1_FLAG2_IRQ	6			INTISR[70]	
ACE_PC1_FLAG3_IRQ	7			INTISR[71]	
ACE_PC2_FLAG0_IRQ*	8			INTISR[72]	Program Counter 2 flags: These four general-purpose flags can be used by the SSE to create interrupt events.
ACE_PC2_FLAG1_IRQ*	9			INTISR[73]	
ACE_PC2_FLAG2_IRQ*	10			INTISR[74]	
ACE_PC2_FLAG3_IRQ*	11			INTISR[75]	
ADC Conversion Done Events					
ACE_ADC0_DATAVALID_IRQ	12			INTISR[76]	These IRQs indicate an end of ADC conversion event for ADC0, ADC1 or ADC2
ACE_ADC1_DATAVALID_IRQ	13			INTISR[77]	
ACE_ADC2_DATAVALID_IRQ*	14			INTISR[78]	

Table 1-7 **• ACE Interrupts**

Name	ACE Interrupt	ACE Register / Address	ACE Register Address	Cortex-M3 Interrupt	Function
ADC Calibration Done Events					
ACE_ADC0_CALDONE_IRQ	15			INTISR[79]	These IRQs indicate an end of calibration event for ADC0, ADC1 or ADC2
ACE_ADC1_CALDONE_IRQ	16			INTISR[80]	
ACE_ADC2_CALDONE_IRQ*	17			INTISR[81]	
ADC Calibration Start Events					
ACE_ADC0_CALSTART_IRQ	18			INTISR[82]	This IRQs indicate a start of calibration event for ADC0, ADC1 or ADC2
ACE_ADC1_CALSTART_IRQ	19			INTISR[83]	
ACE_ADC2_CALSTART_IRQ*	20			INTISR[84]	
Comparator Falling Edge Interrupts					
ACE_COMP0_FALL_IRQ	21	COMP_IRQ	0x40021210	INTISR[85]	These IRQs indicate falling edge events on any or all of the COMPARATOR[11:0] signals from the SCBs
ACE_COMP1_FALL_IRQ	22			INTISR[86]	
ACE_COMP2_FALL_IRQ	23			INTISR[87]	
ACE_COMP3_FALL_IRQ	24			INTISR[88]	
ACE_COMP4_FALL_IRQ	25			INTISR[89]	
ACE_COMP5_FALL_IRQ	26			INTISR[90]	
ACE_COMP6_FALL_IRQ	27			INTISR[91]	
ACE_COMP7_FALL_IRQ	28			INTISR[92]	
ACE_COMP8_FALL_IRQ*	29			INTISR[93]	
ACE_COMP9_FALL_IRQ*	30			INTISR[94]	
Reserved	31			INTISR[95]	
Reserved	32			INTISR[96]	

***Table 1-7* • ACE Interrupts**

Name	ACE Interrupt	ACE Register / Address	ACE Register Address	Cortex-M3 Interrupt	Function
Comparator Rising Edge Interrupts					
ACE_COMP0_RISE_IRQ	33			INTISR[97]	These IRQs indicate rising edge events on any or all of the COMPARATOR[11:0] signals from the SCBs
ACE_COMP1_RISE_IRQ	34			INTISR[98]	
ACE_COMP2_RISE_IRQ	35			INTISR[99]	
ACE_COMP3_RISE_IRQ	36			INTISR[100]	
ACE_COMP4_RISE_IRQ	37			INTISR[101]	
ACE_COMP5_RISE_IRQ	38			INTISR[102]	
ACE_COMP6_RISE_IRQ	39			INTISR[103]	
ACE_COMP7_RISE_IRQ	40			INTISR[104]	
ACE_COMP8_RISE_IRQ*	41			INTISR[105]	
ACE_COMP9_RISE_IRQ*	42			INTISR[106]	
Reserved	43			INTISR[107]	
Reserved	44			INTISR[108]	
FIFO Full / Almost Full / Not Empty Events					
ACE_ADC0_FIFOFULL_IRQ	45	PPE_FIFO_IRQ	0x4002121C	INTISR[109]	Indicates that the ADC0 result FIFO has just become full
ACE_ADC0_FIFOAFULL_IRQ	46			INTISR[110]	Indicates that the ADC0 result FIFO has just become almost full
ACE_ADC0_FIFOEMPTY_IRQ	47			INTISR[111]	Indicates that the ADC0 result FIFO has just become non-empty
ACE_ADC1_FIFOFULL_IRQ	48			INTISR[112]	Indicates that the ADC1 result FIFO has just become full
ACE_ADC1_FIFOAFULL_IRQ	49			INTISR[113]	Indicates that the ADC1 result FIFO has just become almost full
ACE_ADC1_FIFOEMPTY_IRQ	50			INTISR[114]	Indicates that the ADC1 result FIFO has just become non-empty
ACE_ADC2_FIFOFULL_IRQ*	51			INTISR[115]	Indicates that the ADC2 result FIFO has just become full
ACE_ADC2_FIFOAFULL_IRQ*	52			INTISR[116]	Indicates that the ADC2 result FIFO has just become almost full
ACE_ADC2_FIFOEMPTY_IRQ*	53			INTISR[117]	Indicates that the ADC2 result FIFO has just become non-empty

***Table 1-7* • ACE Interrupts**

Name	ACE Interrupt	ACE Register / Address	ACE Register Address	Cortex-M3 Interrupt	Function
PPE Threshold Events					
ACE_PPE_FLAG0_IRQ	54	PPE_FLAGS0_IRQ | PPE_FLAGS1_IRQ | PPE_FLAGS2_IRQ | PPE_FLAGS3_IRQ | PPE_SFFLAGS_IRQ	0x40021228, 0x40021234, 0x40021240, 0x4002124C, 0x40021258	INTISR[118]	Logical OR of PPE_FLAGS0_IRQ, PPE_FLAGS1_IRQ, PPE_FLAGS2_IRQ, PPE_FLAGS3_IRQ and PPESFFLAGS_IRQ
ACE_PPE_FLAG1_IRQ	55			INTISR[119]	
ACE_PPE_FLAG2_IRQ	56			INTISR[120]	
ACE_PPE_FLAG3_IRQ	57			INTISR[121]	
ACE_PPE_FLAG4_IRQ	58			INTISR[122]	
ACE_PPE_FLAG5_IRQ	59			INTISR[123]	
ACE_PPE_FLAG6_IRQ	60			INTISR[124]	
ACE_PPE_FLAG7_IRQ	61			INTISR[125]	
ACE_PPE_FLAG8_IRQ	62			INTISR[126]	
ACE_PPE_FLAG9_IRQ	63			INTISR[127]	
ACE_PPE_FLAG10_IRQ	64			INTISR[128]	
ACE_PPE_FLAG11_IRQ	65			INTISR[129]	
ACE_PPE_FLAG12_IRQ	66			INTISR[130]	
ACE_PPE_FLAG13_IRQ	67			INTISR[131]	
ACE_PPE_FLAG14_IRQ	68			INTISR[132]	
ACE_PPE_FLAG15_IRQ	69			INTISR[133]	
ACE_PPE_FLAG16_IRQ	70			INTISR[134]	
ACE_PPE_FLAG17_IRQ	71			INTISR[135]	
ACE_PPE_FLAG18_IRQ	72			INTISR[136]	
ACE_PPE_FLAG19_IRQ	73			INTISR[137]	
ACE_PPE_FLAG20_IRQ	74			INTISR[138]	
ACE_PPE_FLAG21_IRQ	75			INTISR[139]	
ACE_PPE_FLAG22_IRQ	76			INTISR[140]	
ACE_PPE_FLAG23_IRQ	77			INTISR[141]	
ACE_PPE_FLAG24_IRQ	78			INTISR[142]	
ACE_PPE_FLAG25_IRQ	79			INTISR[143]	
ACE_PPE_FLAG26_IRQ	80			INTISR[144]	
ACE_PPE_FLAG27_IRQ	81			INTISR[145]	
ACE_PPE_FLAG28_IRQ	82			INTISR[146]	
ACE_PPE_FLAG29_IRQ	83			INTISR[147]	
ACE_PPE_FLAG30_IRQ	84			INTISR[148]	
ACE_PPE_FLAG31_IRQ	85			INTISR[149]	

2 – SmartFusion Analog Front-End (AFE) Overview

SmartFusion Analog Features

- One to three ADCs per device
- One to three sigma-delta DACs per device
- Nyquist ADCs are successive approximation type
- ADCs support 8-bit, 10-bit, and 12-bit modes
- Up to 600 KHz sample rate, per ADC
- Each ADC incorporates a sample-and-hold (S&H)
- Simultaneous sampling with multiple ADCs
- Up to twenty-two 0 V–2.56 V unipolar single-ended inputs
- Up to ten bipolar single-ended prescalers
- Active bipolar prescalers have four selectable input ranges: ±2.5 V, ±5.12 V, ±10.24 V, and ±15.36 V, with inputs from –11.5 V to +14.4 V
- Up to five differential-input amplifiers with a gain of 50, optimized for use as current monitors
- Up to five temperature monitors
- Up to ten high-speed analog comparators with selectable hysteresis
- Up to seven unique analog outputs (from DACs) by use of demultiplexing and S&H techniques
- DAC sampling rates of over 600 KHz possible
- Built-in precision voltage reference
- External voltage reference input for ratiometric operation of ADC and DAC
- Pins shared between functions for low pin count

SmartFusion Analog Front-End General Description

SmartFusion devices have a versatile analog front-end (AFE) that provides a very useful complement to the ARM Cortex-M3 microcontroller and general-purpose FPGA fabric. The SmartFusion customizable system-on-chip (cSoC) has a sophisticated controller for the analog front-end called the analog compute engine (ACE), which configures and sequences all the analog functions and post-processes the results, all independently of the Cortex-M3, thus offloading the Cortex-M3 to perform other processing activities. This frees the processor from waiting idle for data to be sampled or transforming the sampled data. Refer to the "Analog Compute Engine (ACE)" section on page 5 for further information.

SmartFusion devices have multiple analog-to-digital converters (ADCs) and digital-to-analog converters (DACs). The A2F060 has one of each, the A2F200 has two of each, and the A2F500 has three of each.

The SmartFusion ADCs are of the capacitor-ratio successive approximation type. Each self-calibrating ADC has a built-in sample-and-hold (S&H) function for predictable results once conversion has started. Each 12-bit ADC can be operated in 8-bit or 10-bit mode by prematurely terminating the conversion, allowing for more conversions to be performed per second. Multiple ADCs can simultaneously sample their respective inputs at the same instant, and multiple SmartFusion devices can even be synchronized. Refer to the "Analog-to-Digital Converter (ADC)" section on page 27 for further information.

The SmartFusion DACs are of the one-bit sigma-delta type. The ACE has a built-in first-order sigma-delta modulator which feeds each one-bit DAC. Alternately, each DAC's input (and clock) can come from the FPGA fabric, where a custom modulator can be instantiated. Each DAC includes an on-chip, three-pole, continuous-time low-pass output filter. A fourth pole of potentially much lower frequency can be added by attaching an external capacitor to the DAC output pin. The DAC can be operated in either

voltage or current output mode. In voltage output mode, a 10 kΩ (nominal) resistor is internally connected to ground. Refer to the "Analog Sigma-Delta Digital to Analog Converter (DAC)" section of the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet for further information.

Many of the analog input functions are organized into signal conditioning blocks (SCBs). Each SCB contains the following:

- Two active bipolar prescalers (ABPS). These single-ended amplifiers have four selectable input ranges, and can operate with high input voltages. Refer to the "Active Bipolar Prescaler (ABPS)" section on page 67 for further information.
- One high-gain current monitor. These auto-zeroing differential-input amplifiers have a fixed gain of 50, and are optimized for sensing the differential voltage across a current sense resistor, even in the presence of high common mode voltages. Refer to the "Current Monitor" section on page 71 for further information.
- One temperature monitor, designed to sense temperature using an external junction diode or diode-connected bipolar transistor. Refer to the "Temperature Monitor" section on page 81 for further information.
- Two high-speed analog comparators with selectable hysteresis levels. The reference voltage for one of each pair can optionally come from any of the SmartFusion DACs, and this signal can be demultiplexed using internal switches and external (board-level) capacitors as part of a sample-and-hold circuit. Refer to the "High-Speed Comparators" section on page 91 for further information.

Each ADC has four additional direct ADC Inputs. These four inputs can also be configured as LVTTL digital inputs if they are not needed for their analog functionality, and are available at the fabric interface for user logic. Refer to the "Fabric Interface and IOMUX" section in the *SmartFusion Microcontroller Subsystem User's Guide* for further information.

Many of the pins of the analog front-end share multiple functions, in order to use as few pins as possible and thus leave as many as possible for the use of the FPGA and the Cortex-M3 dedicated digital peripherals. Nearly all the analog blocks contain an enable configuration bit which can be used to save power on any unused functions. There are also other configuration options, such as the input range selection of the active bipolar prescalers, and the hysteresis selection of the comparators. The current monitor and the temperature monitor can be disabled and bypassed so their input pins go directly to the ADC. Refer to the "Direct ADC Inputs" section on page 103 for information on how to configure each function.

Figure 2-1 on page 23 shows the first ADC and DAC, and the first two SCBs common to both the A2F200 and A2F500. The A2F200 has exactly twice what is shown in Figure 2-1 on page 23. That is, it has one more ADC, one more DAC, and two more SCBs, exactly as shown—but doubled. All the pin and reference designator numbers increase linearly from the highest value shown in Figure 2-1 on page 23. For example, ABPS[0:3] are shown. The remaining prescalers in the A2F200 are ABPS[4:7], numbered in the same order as ABPS[0:3] are in Figure 2-1 on page 23. The CM[0:1] pads continue as CM[2:3]. The A2F500 has a third ADC and DAC, with an additional SCB, versus the normal two SCBs per ADC found in the A2F200. The third ADC still has four additional direct inputs, though two of these do not have a comparator in parallel or a demultiplexer on the DACs, as is the case in all other similar situations.

Figure 2-1 shows a partial block diagram.

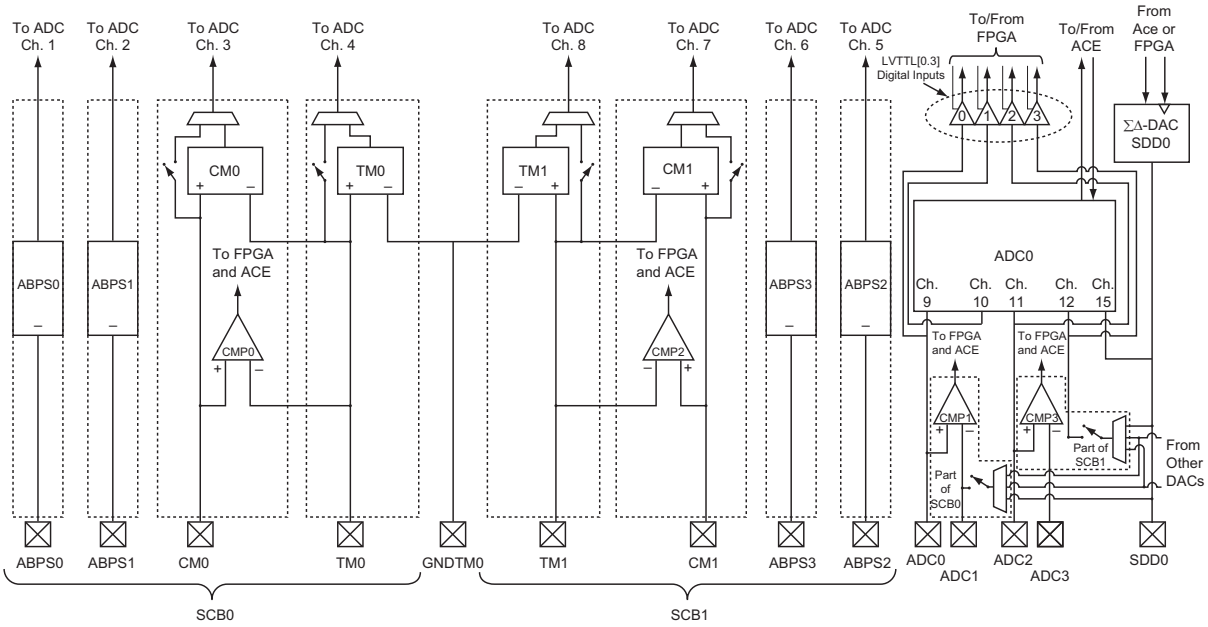


***Figure 2-1* • Partial Block Diagram of SmartFusion Programming Analog Showing One ADC, One DAC, and Two SCBs**

Table 2-1 relates each of the analog front-end functions with the signal conditioning block (SCB) in which it is located:

***Table 2-1* • Analog Front-End Input Pad Designations and Shared Functions**

ADC/ SDD	Signal Conditioning Block	Pad Name	Shared Functions on Pad			
ADC0 SDD0	SCB0	CM0	ADC0_CH3	CMP0_P	CM0_H	
		TM0	ADC0_CH4	CMP0_N	CM0_L	TM0_IO
		ABPS0	ADC0_CH1			
		ABPS1	ADC0_CH2			
		ADC0	ADC0_CH9	CMP1_P	LVTTL0_IN	
		ADC1	ADC0_CH10	CMP1_N	LVTTL1_IN	SDDM0_OUT
	SCB1	CM1	ADC0_CH7	CMP2_P	CM1_H	
		TM1	ADC0_CH8	CMP2_N	CM1_L	TM1_IO
		ABPS2	ADC0_CH5			
		ABPS3	ADC0_CH6			
		ADC2	ADC0_CH11	CMP3_P	LVTTL2_IN	
		ADC3	ADC0_CH12	CMP3_N	LVTTL3_IN	SDDM1_OUT
	(N/A)	SDD0	ADC0_CH15	SDD0_OUT		

*Note: *Components indicated are available only in the A2F500.*

***Table 2-1* • Analog Front-End Input Pad Designations and Shared Functions (continued)**

ADC/ SDD	Signal Conditioning Block	Pad Name	Shared Functions on Pad			
ADC1 SDD1	SCB2	CM2	ADC1_CH3	CMP4_P	CM2_H	
		TM2	ADC1_CH4	CMP4_N	CM2_L	TM2_IO
		ABPS4	ADC1_CH1			
		ABPS5	ADC1_CH2			
		ADC4	ADC1_CH9	CMP5_P	LVTTL4_IN	
		ADC5	ADC1_CH10	CMP5_N	LVTTL5_IN	SDDM2_OUT
	SCB3	CM3	ADC1_CH7	CMP6_P	CM3_H	
		TM3	ADC1_CH8	CMP6_N	CM3_L	TM3_IO
		ABPS6	ADC1_CH5			
		ABPS7	ADC1_CH6			
		ADC6	ADC1_CH11	CMP7_P	LVTTL6_IN	
		ADC7	ADC1_CH12	CMP7_N	LVTTL7_IN	SDDM3_OUT
	(N/A)	SDD1	ADC1_CH15	SDD1_OUT		
ADC2* SDD2*	SCB4*	CM4*	ADC2_CH3*	CMP8_P*	CM4_H*	
		TM4*	ADC2_CH4*	CMP8_N*	CM4_L*	TM4_IO*
		ABPS8*	ADC2_CH1*			
		ABPS9*	ADC2_CH2*			
		ADC8*	ADC2_CH9*	CMP9_P*	LVTTL8_IN*	
		ADC9*	ADC2_CH10*	CMP9_N*	LVTTL9_IN*	SDDM4_OUT*
	(N/A)	SDD2*	ADC2_CH15*	SDD2_OUT*		

*Note: *Components indicated are available only in the A2F500.*

Analog Front-End Configuration

The address range of the analog compute engine (ACE) in the Cortex-M3 address space is 0x40020000 to 0x4002FFFF. Bits 12:0 are used as the address for the advanced peripheral bus (APB) that connects the ACE to the Cortex-M3 via the advanced high-speed bus matrix (ABM).

The offset for the start of the SCB registers is 0x0204, and each SCB (0 through 3) uses twelve bytes (B0 through B11). Only the least significant byte of each 32-bit Cortex-M3 word is utilized, so the 32-bit Cortex-M3 addresses advance by four bytes for each sequential byte actually used; the intervening three bytes available in the Cortex-M3 address space are not used. The Cortex-M3 uses little-endian byte order for its physical memory addressing; the least significant byte is at the lowest address. All addresses and numbers shown in the Table 2-2 on page 25 use big-endian notation; the most significant bit (or hexadecimal nibble) is shown on the left, regardless of how many bits or bytes are represented.

The same bytes can be addressed by the ACE's sample sequencing engine (SSE) or post processing engine (PPE) using 8-bit addresses, starting at 0x81, with the sample sequencer and post processor address advancing by one for each byte used.

***Table 2-2* • Analog Compute Engine Configuration Registers for the Analog Front-End**

Register Name	ABM Address	ACE Address	R/W	Reset Value	Description
SCB0_REG0	0x40020204	0x81	R/W	0x00	SCB0 configuration registers
SCB0_REG1	0x40020208	0x82			
SCB0_REG2	0x4002020C	0x83			
SCB0_REG3	0x40020210	0x84			
SCB0_REG4	0x40020214	0x85			
SCB0_REG5	0x40020218	0x86			
SCB0_REG6	0x4002021C	0x87			
SCB0_REG7	0x40020220	0x88			
SCB0_REG8	0x40020224	0x89			
SCB0_REG9	0x40020228	0x8A			
SCB0_REG10	0x4002022C	0x8B			
SCB0_REG11	0x40020230	0x8C			
SCB1_REG0	0x40020234	0x8D	R/W	0x00	SCB1 configuration registers
SCB1_REG1	0x40020238	0x8E			
⋮	⋮	⋮			
SCB1_REG11	0x40020260	0x98			
SCB2_REG0	0x40020264	0x99	R/W	0x00	SCB2 configuration registers
SCB2_REG1	0x40020268	0x9A			
⋮	⋮	⋮			
SCB2_REG11	0x40020290	0xA4			
SCB3_REG0	0x40020294	0xA5	R/W	0x00	SCB3 configuration registers
SCB3_REG1	0x40020298	0xA6			
⋮	⋮	⋮			
SCB3_REG11	0x400202C0	0xB0			
SCB4_REG0*	0x400202C4*	0xB1*	R/W*	0x00*	SCB4 configuration registers
SCB4_REG1*	0x400202C8*	0xB2*			
⋮*	⋮*	⋮*			
SCB4_REG11*	0x400202F0*	0xBC*			

*Note: *Components indicated are only available in the A2F500.*

Many of the analog functions are calibrated by Microsemi during test of the SmartFusion devices. These calibrations are stored in dedicated nonvolatile memory in each part.

SmartDesign software automatically configures many features of the analog front-end, according to the selections made by the end user using the graphical front-end it provides. It configures static bits, such as enabling used sections and disabling unused sections to save power. It also creates microcode programs for the sample sequencing engine (SSE) and post processing engine (PPE) in the ACE which

will cause the ADC to take samples and to apply the factory-stored compensation values to the readings. In some cases, user-supplied calibration values are merged with the factory-supplied values. This automatically generated program is usually all that is necessary to use the SmartFusion analog front-end. However, it is possible to dynamically change most every setting, including dynamically enabling and disabling functions, or sequentially using an input pin for more than one function by writing custom microcode. Refer to the "Analog Compute Engine (ACE)" section on page 5 for more information.

Related Information

SmartFusion Customizable System-on-Chip (cSoC) datasheet

http://www.microsemi.com/soc/documents/SmartFusion_DS.pdf

3 – Analog-to-Digital Converter (ADC)

The SmartFusion ADC is a 12-bit successive approximation register (SAR) ADC. Figure 3-1 shows a simplified block diagram of the SmartFusion ADC. The A2F060 contains one ADC, the A2F200 contains two ADCs, and the A2F500 contains three ADCs. Multiple ADCs can be triggered to sample signals simultaneously.

Key Features

- Configurable resolution: 8-bit, 10-bit, and 12-bit mode
- Differential nonlinearity (DNL) Error: 0.6 LSB for 10-bit mode
- Integral nonlinearity (INL) Error: 0.8 LSB for 10-bit mode
- Internal VAREF = 2.56 V
- Maximum sample rate = 600 Ksps
- Power-up calibration and dynamic calibration after every sample to compensate for temperature drift over time

Block Diagram

Figure 3-1 shows a simplified block diagram for the ADC.

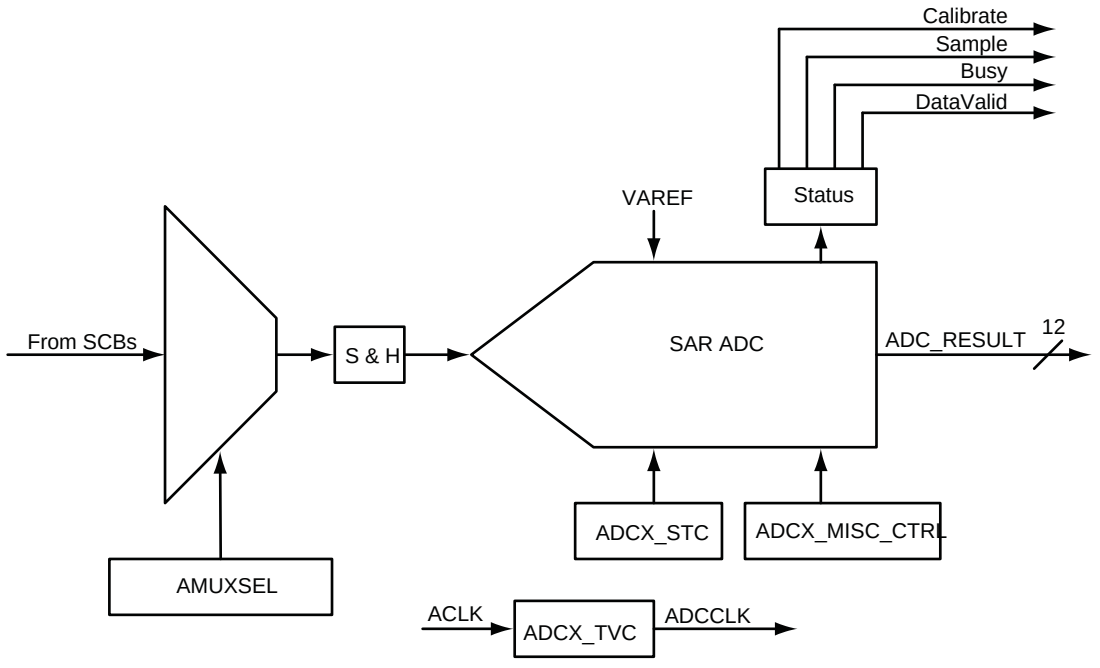


***Figure 3-1* • ADC Simplified Block Diagram**

ADC Theory of Operation

An analog-to-digital converter is used to capture discrete samples of a continuous analog voltage and provide a discrete binary representation of the signal. Analog-to-digital converters are generally characterized in three ways:

- Input voltage range
- Resolution
- Bandwidth or conversion rate

The input voltage range of an ADC is determined by its reference voltage (VREF). SmartFusion devices include an internal 2.56 V reference, or the user can supply an external reference of up to 3.3 V. The following examples use the internal 2.56 V reference, so the full-scale input range of the ADC is 0 to 2.56 V.

The resolution (LSB) of the ADC is a function of the number of binary bits in the converter. The ADC approximates the value of the input voltage using 2^n steps, where *n* is the number of bits in the converter. Each step therefore represents $V_{REF} / 2^n$ volts. In the case of the SmartFusion ADC configured for 12-bit operation, the LSB is 2.56 V / 4096 = 0.625 mV.

Finally, bandwidth is an indication of the maximum number of conversions the ADC can perform each second. The bandwidth of an ADC is constrained by its architecture and several key performance characteristics.

There are several popular ADC architectures, each with advantages and limitations. The analog-to-digital converter in SmartFusion devices is a switched-capacitor successive approximation register (SAR) ADC. It supports 8-, 10-, and 12-bit modes of operation with a cumulative sample rate up to 600,000 samples per second (Ksps). A built-in bandgap circuitry offers 1% internal voltage reference accuracy or an external reference voltage can be used.

As shown in Figure 3-2, a SAR ADC contains N capacitors with binary-weighted values.

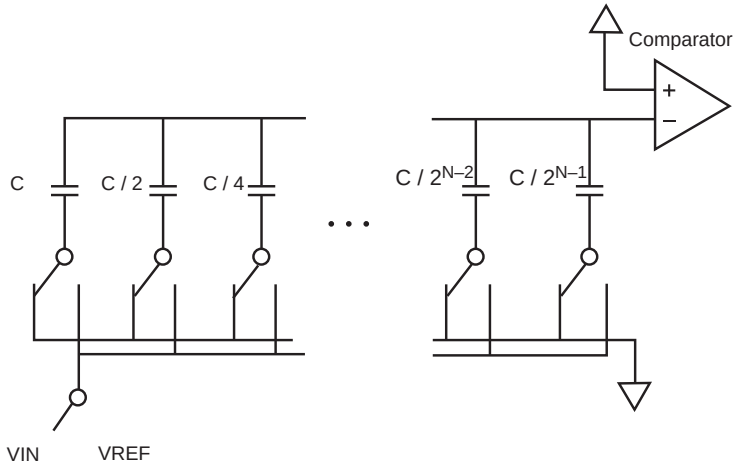


***Figure 3-2* • Example SAR ADC Architecture**

To begin a conversion, all of the capacitors are quickly discharged. Then V_{IN} is applied to all the capacitors for a period of time (acquisition time) during which the capacitors are charged to a value very close to VIN. Then all of the capacitors are switched to ground, and thus –VIN is applied across the comparator. Now the conversion process begins. First, C is switched to VREF. Because of the binary weighting of the capacitors, the voltage at the input of the comparator is then shown by EQ 3-1.

$$\text{Voltage at input of comparator} = -\text{VIN} + \text{VREF} / 2$$

EQ 3-1

If VIN is greater than VREF / 2, the output of the comparator is 1; otherwise, the comparator output is 0. A register is clocked to retain this value as the MSB of the result. Next, if the MSB is 0, C is switched back to ground; otherwise, it remains connected to VREF, and C / 2 is connected to VREF. The result at

the comparator input is now either –VIN + VREF / 4 or –VIN + 3 VREF / 4 (depending on the state of the MSB), and the comparator output now indicates the value of the next most significant bit. This bit is likewise registered, and the process continues for each subsequent bit until a conversion is completed. The conversion process requires some acquisition time plus N + 1 ADC clock cycles to complete.

This process results in a binary approximation of VIN. Generally, there is a fixed interval T, the sampling period, between the samples. The inverse of the sampling period is often referred to as the sampling frequency f_S = 1 / T. The combined effect is illustrated in Figure 3-3.

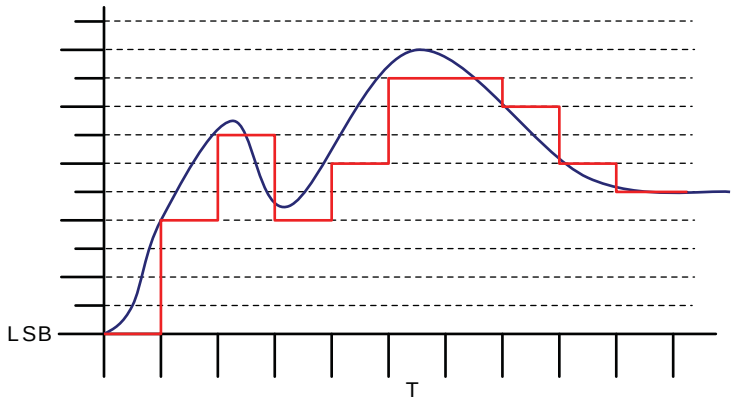


***Figure 3-3* • Conversion Example**

Figure 3-3 demonstrates that if the signal changes faster than the sampling rate can accommodate, or if the actual value of V_{IN} falls between counts in the result, this information is lost during the conversion. There are several techniques that can be used to address these issues.

First, the sampling rate must be chosen to provide enough samples to adequately represent the input signal. Based on the Nyquist-Shannon Sampling Theorem, the minimum sampling rate must be at least twice the frequency of the highest frequency component in the target signal (Nyquist Frequency). For example, to recreate the frequency content of an audio signal with up to 22 KHz bandwidth, the user must sample it at a minimum of 44 ksps. However, as shown in Figure 3-3, significant post-processing of the data is required to interpolate the value of the waveform during the time between each sample.

Similarly, to re-create the amplitude variation of a signal, the signal must be sampled with adequate resolution. Continuing with the audio example, the dynamic range of the human ear (the ratio of the amplitude of the threshold of hearing to the threshold of pain) is generally accepted to be 135 dB, and the dynamic range of a typical symphony orchestra performance is around 85 dB. Most commercial recording media provide about 96 dB of dynamic range using 16-bit sample resolution. But 16-bit fidelity does not necessarily mean that you need a 16-bit ADC. As long as the input is sampled at or above the Nyquist Frequency, post-processing techniques can be used to interpolate intermediate values and reconstruct the original input signal to within desired tolerances.

If sophisticated digital signal processing (DSP) capabilities are available, the best results are obtained by implementing a reconstruction filter, which is used to interpolate many intermediate values with higher resolution than the original data. Interpolating many intermediate values increases the effective number of samples, and higher resolution increases the effective number of bits in the sample. In many cases, however, it is not cost-effective or necessary to implement such a sophisticated reconstruction algorithm. For applications that do not require extremely fine reproduction of the input signal, alternative methods can enhance digital sampling results with relatively simple post-processing. The details of such techniques are out of the scope of this chapter; refer to the *Improving ADC Results through Oversampling and Post-Processing of Data* white paper for more information.

ADC Terminology

Conversion Time

Conversion time is the interval between the release of the hold state (imposed by the input circuitry of a track-and-hold) and the instant at which the voltage on the sampling capacitor settles to within one LSB of a new input value.

DNL – Differential Non-Linearity

For an ideal ADC, the analog-input levels that trigger any two successive output codes should differ by one LSB (DNL = 0). Any deviation from one LSB in defined as DNL (Figure 3-4).

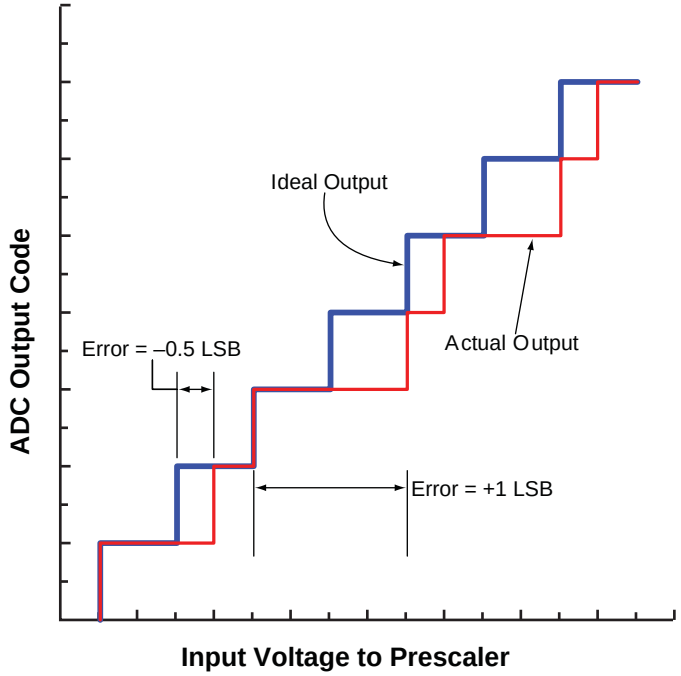


***Figure 3-4* • Differential Non-Linearity (DNL)**

ENOB – Effective Number of Bits

ENOB specifies the dynamic performance of an ADC at a specific input frequency and sampling rate. An ideal ADC's error consists only of quantization of noise. As the input frequency increases, the overall noise (particularly in the distortion components) also increases, thereby reducing the ENOB and SINAD (also see "SINAD – Signal-to-Noise and Distortion" on page 33). ENOB for a full-scale, sinusoidal input waveform is computed using EQ 3-2.

$$ENOB = \frac{SINAD - 1.76}{6.02}$$

EQ 3-2

FS Error – Full-Scale Error

Full-scale error is the difference between the actual value that triggers that transition to full-scale and the ideal analog full-scale transition value. Full-scale error equals offset error plus gain error.

Gain Error

The gain error of an ADC indicates how well the slope of an actual transfer function matches the slope of the ideal transfer function. Gain error is usually expressed in LSB or as a percent of full-scale (%FSR). Gain error is the full-scale error minus the offset error (Figure 3-5).

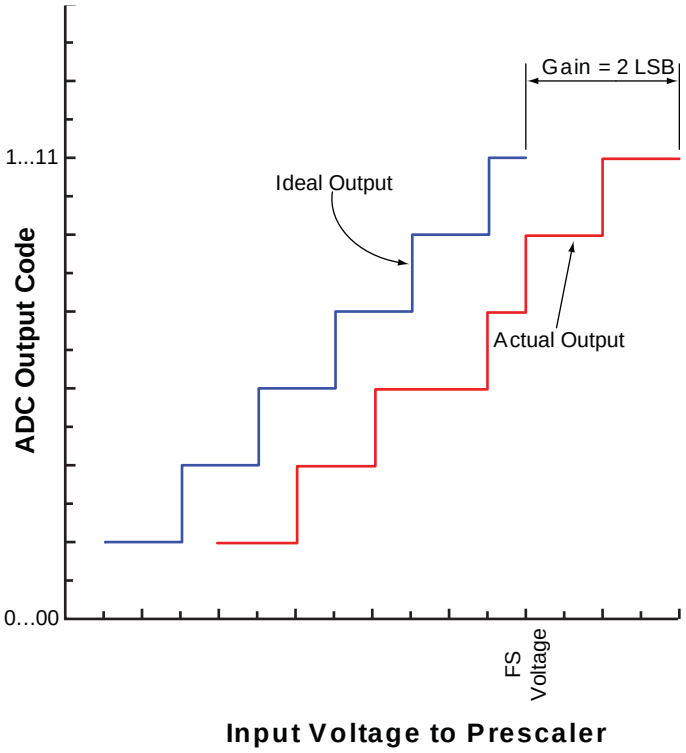


***Figure 3-5* • Gain Error**

Gain Error Drift

Gain-error drift is the variation in gain error due to a change in ambient temperature, typically expressed in ppm/°C.

INL – Integral Non-Linearity

INL is the deviation of an actual transfer function from a straight line. After nullifying offset and gain errors, the straight line is either a best-fit straight line or a line drawn between the end points of the transfer function (Figure 3-6).

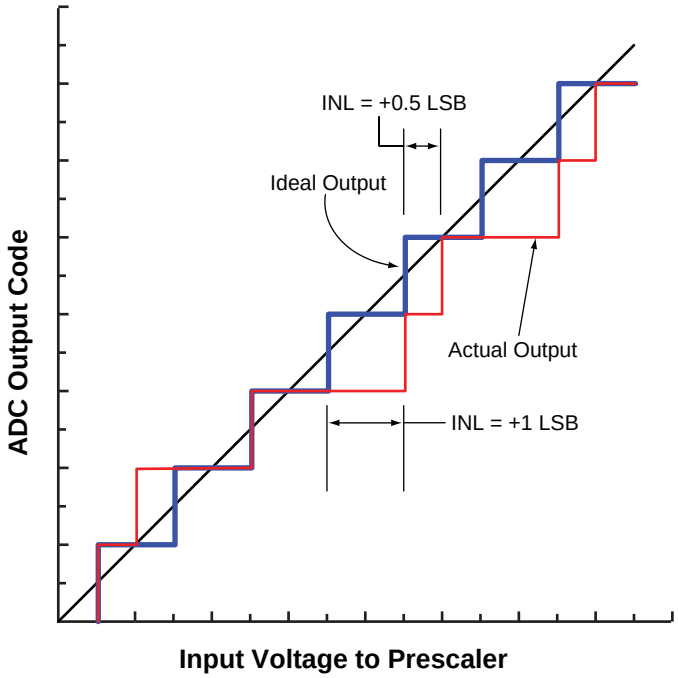


***Figure 3-6* • Integral Non-Linearity (INL)**

LSB – Least Significant Bit

In a binary number, the LSB is the least weighted bit in the group. Typically, the LSB is the furthest right bit. For an ADC, the weight of an LSB equals the full-scale voltage range of the converter divided by 2^N, where N is the converter's resolution.

EQ 3-3 shows the calculation for a 10-bit ADC with a unipolar full-scale voltage of 2.56 V:

$$1\ \text{LSB} = (2.56\ \text{V} / 2^{10}) = 2.5\ \text{mV}$$

EQ 3-3

No Missing Codes

An ADC has no missing codes if it produces all possible digital codes in response to a ramp signal applied to the analog input.

Offset Error

Offset error indicates how well the actual transfer function matches the ideal transfer function at a single point. For an ideal ADC, the first transition occurs at 0.5 LSB above zero. The offset voltage is measured by applying an analog input such that the ADC outputs all zeroes and increases until the first transition occurs (Figure 3-7 on page 33).

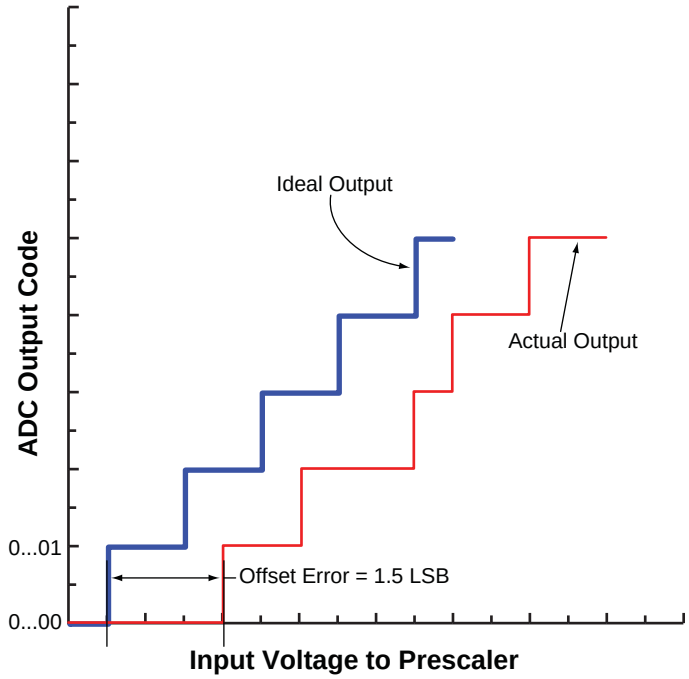


***Figure 3-7* • Offset Error**

Resolution

ADC resolution is the number of bits used to represent an analog input signal. To more accurately replicate the analog signal, resolution needs to be increased.

Sampling Rate

Sampling rate or sample frequency, specified in samples per second (sps), is the rate at which an ADC acquires (samples) the analog input.

SNR – Signal-to-Noise Ratio

SNR is the ratio of the amplitude of the desired signal to the amplitude of the noise signals at a given point in time. For a waveform perfectly reconstructed from digital samples, the theoretical maximum SNR (EQ 3-4) is the ratio of the full-scale analog input (RMS value) to the RMS quantization error (residual error). The ideal, theoretical minimum ADC noise is caused by quantization error only and results directly from the ADC's resolution (N bits):

$$SNR_{dB[MAX]} = 6.02_{dB} \times N + 1.76_{dB}$$

EQ 3-4

SINAD – Signal-to-Noise and Distortion

SINAD is the ratio of the RMS amplitude to the mean value of the root-sum-square of all the other spectral components, including harmonics, but excluding DC. SINAD is a good indication of the overall dynamic performance of an ADC because it includes all components which make up noise and distortion.

Total Harmonic Distortion

THD measures the distortion content of a signal, and is specified in decibels relative to the carrier (dBc). THD is the ratio of the RMS sum of the selected harmonics of the input signal to the fundamental itself. Only harmonics within the Nyquist limit are included in the measurement.

TUE – Total Unadjusted Error

TUE is a comprehensive specification that includes linearity errors, gain error, and offset error. It is the worst-case deviation from the ideal device performance. TUE is a static specification (Figure 3-8).

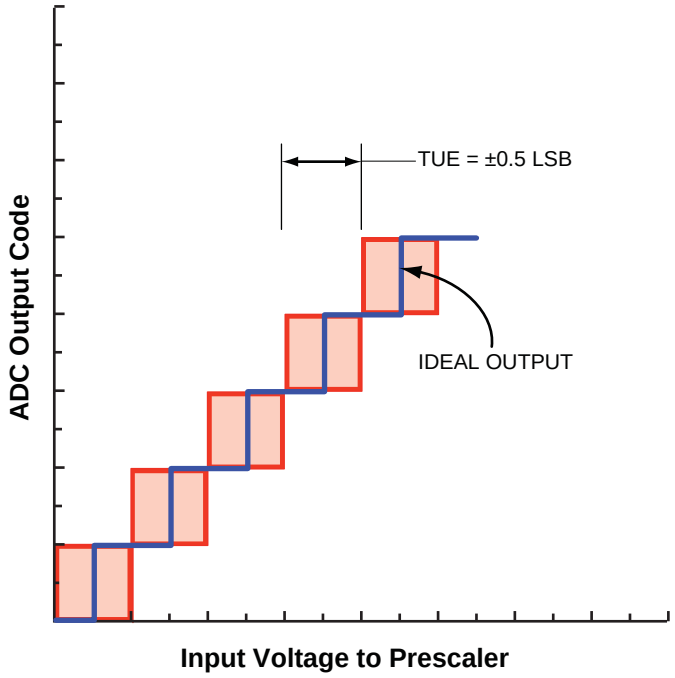


***Figure 3-8* • Total Unadjusted Error (TUE)**

SmartFusion ADC Operation

Analog Multiplexer

The SmartFusion ADC has a built-in analog multiplexer as shown in Figure 3-1 on page 27. Input channels are selected for conversion by setting the AMUXSEL field in the ADCx_CONV_CTRL register to the channel desired. Table 3-1 lists the channels available for conversion by the respective ADCs. Each ADC multiplexer is individually configurable by the respective ADCx_CONV_CTRL register.

***Table 3-1* • ADC0 and ADC1 Channel Assignments**

ADC0		ADC1		ADC2*	
AMUXSEL	**Pad Name**	**AMUXSEL**	**Pad Name**	**AMUXSEL***	**Pad Name***
0	1.5 V	0	1.5 V	0*	1.5V*
1	ABPS0	1	ABPS4	1*	ABPS8*
2	ABPS1	2	ABPS5	2*	ABPS9*
3	CM0	3	CM2	3*	CM4*
4	TM0	4	TM2	4*	TM4*
5	ABPS2	5	ABPS6	5*	NC
6	ABPS3	6	ABPS7	6*	NC
7	CM1	7	CM3	7*	NC
8	TM1	8	TM3	8*	NC
9	ADC0	9	ADC4	9*	ADC8*
10	ADC1	10	ADC5	10*	ADC9*
11	ADC2	11	ADC6	11*	ADC10*
12	ADC3	12	ADC7	12*	ADC11*
13	NC	13	NC	13*	NC
14	NC	14	NC	14*	NC
15	SDD0	15	SDD1	15*	SDD2*

*Note: *Components indicated are available only on the A2F500.*

Acquisition Time or Sample Time Control

Acquisition time (t_{SAMPLE}) specifies how long an analog input signal has to charge the internal capacitor array. Figure 3-9 shows a simplified internal input sampling mechanism of a SAR ADC.

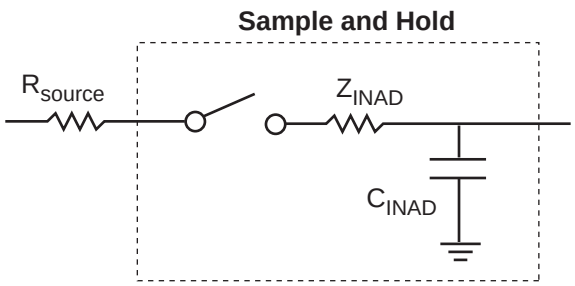


***Figure 3-9* • Simplified Sample and Hold Circuitry**

The internal impedance (Z_{INAD}), external source resistance (R_{SOURCE}), and sample capacitor (C_{INAD}) form a simple RC network. As a result, the accuracy of the ADC can be affected if the ADC is given insufficient time to charge the capacitor. To resolve this problem, the user can either reduce the source resistance or increase the sampling time by changing the acquisition time in the ADCx_STC register.

When the SmartFusion ADC is driven by one of the direct inputs, Microsemi recommends driving the analog input pin with low source impedance (R_{SOURCE}) for fast acquisition time. High source impedance (R_{SOURCE}) is acceptable, but the acquisition time will need to be increased.

EQ 3-5 through EQ 3-7 can be used to calculate the acquisition time required for a given input. The ADCx_STC register contains the number of sample periods in ADCCLK for the acquisition time of the desired signal. If the actual acquisition time is higher than the ADCx_STC register value, the settling time error can affect the accuracy of the ADC, because the sampling capacitor is only partially charged within the given sampling cycle. Example acquisition times are given in Table 3-2 and Table 3-3 on page 37. When controlling the sample time for the ADC along with the use of the active bipolar prescaler, current monitor, or temperature monitor, the minimum sample time(s) for each must be obeyed. EQ 3-8 can be used to determine the appropriate value of the ADCx_STC register.

Users can calculate the minimum actual acquisition time by using EQ 3-5:

$$V_{OUT} = V_{IN}(1 - e^{-t/RC})$$

EQ 3-5

For 0.5 LSB gain error, V_{OUT} should be replaced with (V_{IN} –(0.5 × LSB Value)):

$$(V_{IN} - 0.5 \times \text{LSB Value}) = V_{IN}(1 - e^{-t/RC})$$

EQ 3-6

where V_{IN} is the ADC reference voltage (V_{REF})

Solving EQ 3-6:

$$t = RC \times \ln (V_{IN} / (0.5 \times \text{LSB Value}))$$

EQ 3-7

where $R = Z_{INAD} + R_{SOURCE}$ and $C = C_{INAD}$.

Calculate the value of the ADCx_STC register by using EQ 3-8.

$$t_{SAMPLE} = (2 + \text{ADCx\_STC}) \times (1 / \text{ADCCLK}) \text{ or } t_{SAMPLE} = (2 + \text{ADCx\_STC}) \times (\text{ADC Clock Period})$$

EQ 3-8

where ADCCLK = ADC clock frequency in MHz.

t_{SAMPLE} = 0.449 µs from bit resolution in Table 3-2.

ADC Clock frequency = 10 MHz or a 100 ns period.

ADCx_STC = (t_{SAMPLE} / (1 / 10 MHz)) – 2 = 4.49 – 2 = 2.49.

You must round up to 3 to accommodate the minimum sample time.

***Table 3-2* • Acquisition Time Example with VAREF = 2.56 V**

VIN = 2.56V, R = 4K (R_{SOURCE} ~ 0), C = 18 pF		
Resolution	**LSB Value (mV)**	**Min. Sample/Hold Time for 0.5 LSB (µs)**
8	10	0.449
10	2.5	0.549
12	0.625	0.649

***Table 3-3* • Acquisition Time Example with VAREF = 3.3 V**

VIN = 3.3V, R = 4K (R_{SOURCE} ~ 0), C = 18 pF		
Resolution	**LSB Value (mV)**	**Min. Sample/Hold time for 0.5 LSB (µs)**
8	12.891	0.449
10	3.223	0.549
12	0.806	0.649

ADC Clock

When the SmartFusion ADC is used, the ADC clock (ADCCLK) determines the sampling throughput. ADCCLK is not available to users. As can be seen in the simplified block diagram of the ADC in Figure 3-1 on page 27, the ADCCLK is derived from ACLK. ACLK is the APB_2 clock, which is the MSS main clock FCLK divided by 1, 2, or 4. For details on the relationship between ACLK and the MSS main clock FCLK, refer to the "PLLs, Clock Conditioning Circuitry, and On-Chip Crystal Oscillators" section of the *SmartFusion Microcontroller Subsystem User's Guide*. To generate the ADC clock, ACLK is further divided by a multiple of four. The exact multiple of four used is determined by contents of the ADCx_TVC register and is defined by EQ 3-9.

$$\text{ADC clock frequency (MHz)} = \text{ACLK (MHz)} / (4 \times (\text{ADCx\_TVC} + 1))$$

EQ 3-9

where ADCx_TVC is the contents of the ADCx_TVC register value, from 0 to 255 for ADC x.

The ADCx_TVC register setting is used to ensure that the ADC clock frequency does not exceed 10 MHz nor fall below 0.5 MHz. Note that the 10 MHz maximum frequency for the ADC clock implies that a higher system clock frequency does not always result in a higher ADC clock frequency. For example, a 40 MHz ACLK clock frequency enables a maximum ADC clock frequency of 10 MHz (ADCx_TVC register value of 0), whereas a 50 MHz ACLK results in a slower maximum ADC clock frequency, 6.25 MHz, because a ADCx_TVC register value of 0 would give an ADC clock frequency of 12.5 MHz—above the 10 MHz limit. Setting the ADCx_TVC register value to 1 in this case gives a maximum ADC clock frequency of 6.25 MHz. In general, the performance of the ADC is the rate at which the ADC can acquire or sample the analog input and convert it into a digital value. The inverse of the conversion time is the sampling rate for the channel (EQ 3-10).

$$\text{Sample Rate} = 1 / \text{Conversion Time}$$

EQ 3-10

ADC Resolution

The SmartFusion ADCs can be configured to operate in 8-, 10-, or 12-bit resolution modes by configuring the RESOLUTION field of the ADCx_CONV_CTRL register to the resolution desired, as indicated in Table 3-4. The converted result occupies the most significant bits of the ADC output with the least significant bits set to 0 for 8- and 10-bit modes.

***Table 3-4* • RESOLUTION Field Bit Definitions**

Bit 1	Bit 0	Function
0	0	ADC in 10-bit mode
0	1	ADC in 12-bit mode
1	0	ADC in 8-bit mode
1	1	Reserved

ADC Conversion

Setting the ADCSTART bit in the ADCx_CONV_CTRL register will start the conversion process on the channel that is currently selected by the AMUXSEL field of the ADCx_CONV_CTRL register. Alternatively, setting the ADCSTARTx bit in the ADC_SYNC_CONV register will also start the conversion process (where x is 0, 1, or 2 indicating the desired ADC). Since all three start bits are present in the one ADC_SYNC_CONV register, multiple ADCs may be commanded to sample simultaneously.

A conversion is performed in three phases: the sample phase, the distribution phase, and finally the post-calibration phase. In the sampling phase, the analog input voltage is sampled on the input capacitor. The minimum sample time requirement is defined by EQ 3-7 on page 36.

During distribution phase, the ADC computes the equivalent digital value from the value stored in the input capacitor. The distribution time depends strictly on the number of bits. If the ADC is configured as a 10-bit ADC, then 10 ADCCLK cycles are needed. EQ 3-11 describes the distribution time.

$$t_{DISTRIB} = N \times t_{ADCCLK}$$

EQ 3-11

where N = Number of bits

The last phase is the post-calibration phase. This is an optional phase. The post-calibration phase takes two ADCCLK cycles. Microsemi recommends enabling post-calibration to compensate for drift and temperature-dependent effects. This ensures that the ADC remains consistent over time and with temperature. The post-calibration phase is enabled by clearing the CALIBRATE field of the ADCx_MISC_CTRL register to a 0. EQ 3-12 describes the post-calibration time.

$$t_{POST\_CAL} = \text{CALIBRATE} \times (2 \times t_{ADCCLK})$$

EQ 3-12

The calculation for the conversion time for the ADC is then summarized in EQ 3-13.

$$t_{CONV} = t_{SYNC\_READ} + t_{SAMPLE} + t_{DISTRIB} + t_{POST\_CAL} + t_{SYNC\_WRITE}$$

EQ 3-13

Where:

t_{CONV}: conversion time

t_{SYNC_READ}: maximum time for a signal to synchronize with ACLK. For calculation purposes, the worst case is a period of ACLK, t_{ACLK}.

t_{SAMPLE}: sample time

$t_{DISTRIB}$: distribution time

t_{POST_CAL}: post-calibration time

t_{SYNC_WRITE}: maximum time for a signal to synchronize with ACLK. For calculation purposes, the worst case is a period of ACLK, t_{ACLK}.

Normal Conversion

Figure 3-10 shows normal conversion.

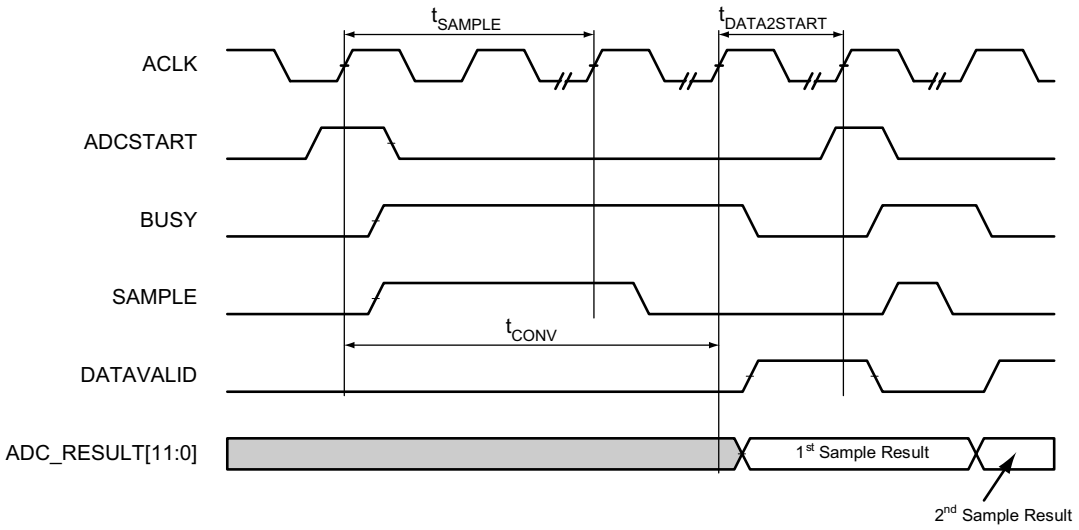


***Figure 3-10* • Normal Conversion**

Intra-Conversion

Performing a conversion during power-up calibration is possible but should be avoided, since the performance is not guaranteed. This is described as intra-conversion and is shown in Figure 3-11.

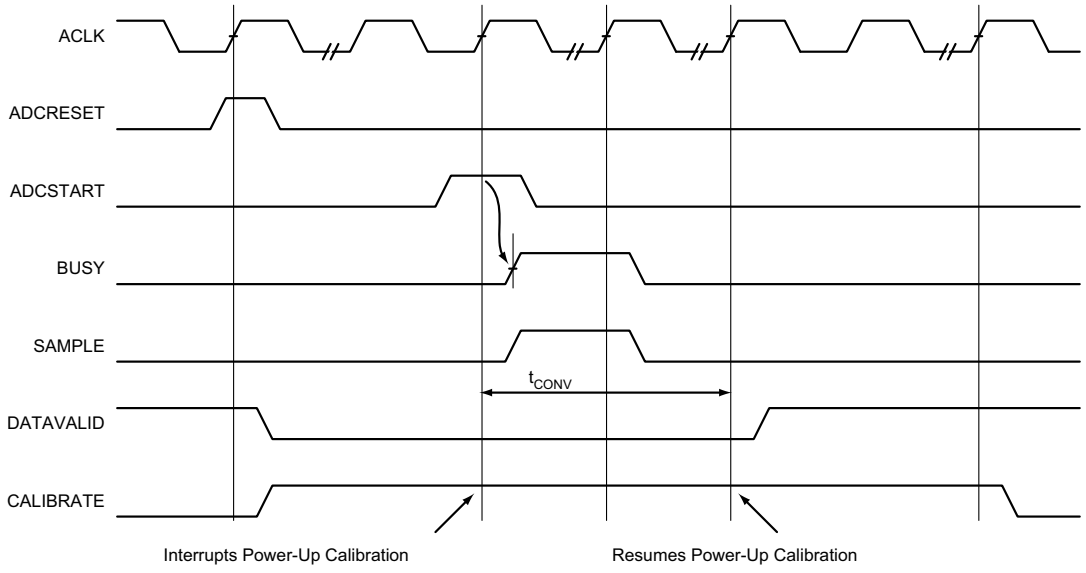


***Figure 3-11* • Intra-Conversion**

Injected Conversion

A conversion can be interrupted by another conversion. Before the current conversion is finished, a second conversion can be started by setting ADCSTART to a 1. When a second conversion is issued before the current conversion is completed, the current conversion is dropped and the ADC starts the second conversion on the rising edge of the ACLK. This is known as an injected conversion. Since the ADC is synchronous, the minimum time to issue a second conversion is two clock cycles of ACLK after the previous one. Figure 3-12 shows injected conversion.

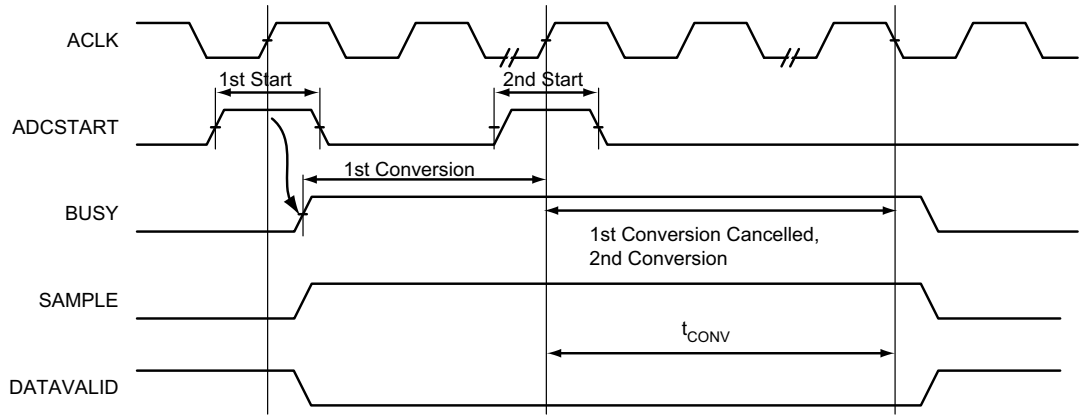


***Figure 3-12* • Injected Conversion**

ADC Conversion Results

The results of the ADC conversion(s) can be read from the ADC_RESULT field in the ADCx_STATUS register.

ADC Power Control

Each ADC can be powered down independently by setting the PWRDWN bit in the ADCx_MISC_CTRL register for an individual ADC or by setting the ADCSPWRDWN bit in the ANA_COMM_CTRL register (which will turn off power for all ADCs in a SmartFusion device). Only the comparators in the ADCs are powered down.

ADC Calibration

Once the ADC has powered up and been released from reset, ADCRESET = 0 in the ADCx_MISC_CTRL register and ADCSRESET = 0 in the ANA_COMM_CTRL register, the ADC will initiate a calibration routine designed to provide optimal ADC performance. The SmartFusion ADC offers a robust calibration scheme to reduce integrated offset and linearity errors. The offset and linearity errors of the main capacitor array are compensated for with an 8-bit calibration capacitor array. The offset/linearity error calibration is carried out in two ways. First, a power-up calibration is carried out when the ADC comes out of reset. The calibration cycle takes 3,840 ADCCLK cycles, as shown in Figure 3-11 on page 39. In this mode, the linearity and offset errors of the capacitors are calibrated.

To further compensate for drift and temperature-dependent effects, every conversion is followed by an optional post-calibration of either the offset or a single bit of the main capacitor array. The post-calibration ensures that, over time and with temperature, the ADC remains consistent. Enabling post conversion calibration is accomplished by clearing the CALIBRATE bit to 0 in the ADCx_MISC_CTRL register.

ADC Interrupts

Table 3-5 lists the interrupts associated with the ADC converters. These interrupts must be enabled in the NVIC of the Cortex-M3 by setting the appropriate bit to a 1. Interrupts are available to indicate whether the ADC is in a calibration cycle and when a converted result is available.

Table 3-5 • **ADC-Related Interrupts**

Name	ACE Interrupt	Cortex-M3 Interrupt	NVIC Address	NVIC Bit at Address	Function
ADC_0_DV_R	12	76	0xE000E108	12	ADC 0 End of Conversion
ADC_1_DV_R	13	77	0xE000E108	13	ADC 1 End of Conversion
ADC_2_DV_F*	14*	78*	0xE000E108	14*	ADC 2 End of Conversion*
ADC_0_CAL_F	15	79	0xE000E108	15	ADC 0 End of Calibration Event
ADC_1_CAL_F	16	80	0xE000E108	16	ADC 1 End of Calibration Event
ADC_2_CAL_F*	17*	81*	0xE000E108	17*	ADC 2 End of Calibration Event*
ADC_0_CAL_R	18	82	0xE000E108	18	ADC 0 Start of Calibration Event
ADC_1_CAL_R	19	83	0xE000E108	19	ADC 1 Start of Calibration Event
ADC_2_CAL_R*	20*	84*	0xE000E108	20*	ADC 2 Start of Calibration Event*

*Note: *The indicated signals are available only in the A2F500.*

Integrated Voltage Reference

The SmartFusion device has an integrated on-chip 2.56 V reference voltage for the ADC. If desired, an external reference voltage of up to 3.3 V can be connected between the VAREFx (where x is = 0, 1, or 2) and GNDVAREF pins. The VAREFSEL bit in the ANA_COMM_CTRL register is used to choose between internal and external references. To use the internal voltage reference, users must connect the VAREFOUT pin to the appropriate ADC V_{REF} input, either any or all of the VAREF0, VAREF1, or VAREF2 pins, on their PCB.

ADC Start-Up Sequence

1. Verify that the 3.3 V supply is valid. Verify that BROWNOUT3_3VN in the DEVICE_SR register is a 1. DEVICE_SR is located at address 0xE0042034 in the memory map.
2. Turn on the analog block by setting ABPWRON in the ANA_COMM_CTRL register to a 1. Wait 1 ms for the internal charge pump to settle (delay time can be consumed by step 3 below).
3. Wait for the VAREF (ADC reference) voltage to stabilize. Reference the VAREF Stabilization Time table in the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet for appropriate delay times.
4. Release the ADC reset by verifying that ADCSRESET in the ANA_COMM_CTRL register is cleared to a 0 and that the individual ADCRESET bit in the ADCx_MISC_CTRL registers are cleared to 0. After releasing the ADC reset, the ADC takes 3,840 ADC clock cycles before calibration is complete. Since the ADC clock needs to be between 0.5 MHz and 10 MHz, this will take from 384 µs to 7.68 mS. Users can optionally poll the CALIBRATE bit in the ADCx_STATUS register for end of completion (CALIBRATE = 0).
5. The ADC is ready for a conversion.

ADC Configuration Example

This example shows how to choose the correct settings to achieve the fastest sample time in 10-bit mode for a system that runs at 66 MHz. We assume the acquisition times defined in Table 3-2 on page 36 for 10-bit mode, which lists 0.549 µs as a minimum hold time.

The period of ACLK: t_{ACLK} = 1/66 MHz = 0.015 µs

For this example, choosing an ADCx_TVC value between 1 and 32 will meet the maximum and minimum period for the ADCCLK requirement. A higher TVC leads to a higher ADCCLK period. The minimum ADCx_TVC value is chosen so that $t_{DISTRIB}$ and t_{POST_CAL} are shorter. The period of ADCCLK with an ADCx_TVC value of 1 can be computed by EQ 3-14.

$$t_{ADCCLK} = 4 \times (1 + ADCx\_TVC) \times t_{ACLK} = 4 \times (1 + 1) \times 0.015\ \mu s = 0.12\ \mu s$$

EQ 3-14

Calculate the value of the ADCx_STC register by using EQ 3-15.

$$ADCx\_STC = (t_{SAMPLE} / t_{ADCCLK}) - 2 = (0.549\ \mu s / 0.12\ \mu s) - 2 = 4.575 - 2 = 2.575.$$

EQ 3-15

You must round up to 3 to accommodate the minimum sample time requirement. So ADCx_STC = 3 for this example. The actual t_{SAMPLE} time is defined by EQ 3-16.

$$t_{SAMPLE} = (2 + ADCx\_STC) \times t_{ADCCLK} = (2 + 3) \times t_{ADCCLK} = 5 \times t_{ADCCLK} = 5 \times 0.12\ \mu s = 0.6\ \mu s$$

EQ 3-16

Since Microsemi recommends post-calibration for temperature drift over time, post-calibration is enabled and the post-calibration time, t_{POST_CAL}, can be computed by EQ 3-17. The post-calibration time is 0.24 µs.

$$t_{POST\_CAL} = CALIBRATE \times (2 \times t_{ADCCLK}) = 1 \times 2 \times 0.12\ \mu s = 0.24\ \mu s$$

EQ 3-17

The distribution time, $t_{DISTRIB}$, is equal to 1.2 µs and can be computed using EQ 3-18.

$$t_{DISTRIB} = N \times t_{ADCCLK} = 10 \times 0.12\ \mu s = 1.2\ \mu s$$

EQ 3-18

The conversion time can now be computed by using EQ 3-19.

$$t_{CONV} = t_{SYNC\_READ} + t_{SAMPLE} + t_{DISTRIB} + t_{POST\_CAL} + t_{SYNC\_WRITE}$$

EQ 3-19

$$= 0.015\ \mu s + 0.60\ \mu s + 1.2\ \mu s + 0.24\ \mu s + 0.015\ \mu s = 2.07\ \mu s = 483\ ksps$$

ACE Firmware Driver

Normal operation of the ADC for most users will occur via interaction with the analog compute engine (ACE) firmware driver. The ACE firmware driver programmatically controls the operation of the sample sequence engine (SSE) and post processing engine (PPE). The SSE automatically controls the setting and clearing of control bits within the ADC register map below to effect the desired operation on the ADC. If users want to control the ADC manually, the SSE and the PPE must be disabled. Disable the SSE by clearing bits 1 (SSE_SRAM_ENABLE) and 0 (TS_ENABLE) of the SSE_TS_CTRL register. Disable the PPE by clearing bit 0 (PPE_EN) of the PPE_CTRL register. Refer to the "Analog Compute Engine (ACE)" section on page 5 for an overview on ACE functionality.

ADC Register Map

Table 3-6 gives the ADC register map.

Table 3-6 • **ADC Memory Map**

Register Name	Address	R/W	Reset Value	Description
SSE_TS_CTRL	0x40020004	R/W	0	Sample sequence engine time slot control
ADC_SYNC_CONV	0x40020008	R/W	0	Synchronized ADC control
ANA_COMM_CTRL	0x4002000C	R/W	1	Common analog block control
ADCx_CONV_CTRL (x = 0)	0x40020050	R/W	0	ADC 0 conversion control
ADCx_STC (x = 0)	0x40020054	R/W	0	ADC 0 sample time control
ADCx_TVC (x = 0)	0x40020058	R/W	0	ADC 0 time division control
ADCx_MISC_CTRL (x = 0)	0x4002005C	R/W	0	ADC 0 control register
ADCx_CONV_CTRL	0x40020090	R/W	0	ADC 1 conversion control
ADC1_STC	0x40020094	R/W	0	ADC 1 sample time control
ADC1_TVC	0x40020098	R/W	0	ADC 1 time division control
ADC1_MISC_CTRL	0x4002009C	R/W	0	ADC 1 control register
ADC2_CONV_CTRL	0x400200D0	R/W	0	ADC 2 conversion control
ADC2_STC	0x400200D4	R/W	0	ADC 2 sample time control
ADC2_TVC	0x400200D8	R/W	0	ADC 2 time division control
ADC2_MISC_CTRL	0x400200DC	R/W	0	ADC 2 control register
ADCx_STATUS (x = 0)	0x40021000	R	0	Status of ADC 0
ADC1_STATUS	0x40021004	R	0	Status of ADC 1
ADC2_STATUS	0x40021008	R	0	Status of ADC 2
PPE_CTRL	0x40021404	R/W	0	Post processing engine control

ADC_SYNC_CONV

Table 3-7 • **ADC_SYNC_CONV**

Bit Number	Name	R/W	Reset Value	Function
7	Reserved	R/W	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
6	CONVWAIT2	R/W	0	1 – When ADCSTART2 is set, the program counter that initiated this instruction in the Sample Sequence Engine increments once, then waits for the conversion to complete. 0 – When ADCSTART2 is set the program counter that set this bit in the sample sequence engine does not wait for the conversion to complete before incrementing.
5	CONVWAIT1	R/W	0	1 – When ADCSTART1 is set, the program counter that initiated this instruction in the sample sequence engine increments once, then waits for the conversion to complete. 0 – When ADCSTART1 is set the program counter that set this bit in the Sample Sequence Engine does not wait for the conversion to complete before incrementing.
4	CONVWAIT0	R/W	0	1 – When ADCSTART0 is set, the program counter that initiated this instruction in the sample sequence engine increments once, then waits for the conversion to complete. 0 – When ADCSTART0 is set the program counter that set this bit in the Sample Sequence Engine does not wait for the conversion to complete before incrementing.
3	Reserved	R/W	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
2	ADCSTART2	R/W	0	1 – Start the conversion on ADC2 0 – No effect. This bit self clears after one ACLK cycle.
1	ADCSTART1	R/W	0	1 – Start the conversion on ADC1, 0 - no effect. This bit self clears after one ACLK cycle.
0	ADCSTART0	R/W	0	1 – Start the conversion on ADC0, 0 - no effect. This bit self clears after one ACLK cycle.

ANA_COMM_CTRL

***Table 3-8* • ANA_COMM_CTRL**

Bit Number	Name	R/W	Reset Value	Function
7:5	Reserved	R/W	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
4	ACB_RESETN	W	0	1 – Reset analog control block 0 – No effect. This bit self clears after one ACLK cycle.
3	ABPOWERON	R/W	0	1 – Power on the entire analog block 0 – Power-down the entire analog block
2	ADCSPWRDWN	R/W	0	1 – Power-down ADC0, ADC1, and ADC2, 0 – Power-up ADC0, ADC1, and ADC2. This bit is logically ORed with the individual ADC PWRDWN bits found in the ADCx_MISC_CTRL registers.
1	ADCSRESET	R/W	0	1 – Reset ADC0, ADC1, and ADC2 0 – No effect. This bit self clears after one ACLK cycle. Note that this bit is logically ORed with bit 4 (ADCRESET) of the ADCx_MISC_CTRL registers to allow common control or individual reset control of the ADC.
0	VAREFSEL	R/W	1	1 – Select external reference voltage for ADC0, ADC1, and ADC2 0 – Select internal reference voltage for ADC0, ADC1, and ADC2

ADCx_CONV_CTRL

***Table 3-9* • ADCx_CONV_CTRL**

Bit Number	Name	R/W	Reset Value	Function
7	ADCSTART	R/W	0	1 – Start ADCx conversion 0 – No effect. This bit self clears after one ACLK cycle
6	CONVWAIT	W	0	1 – When ADCSTART is set the program counter (PCx) increments once, then waits for the conversion to complete. 0 – When ADCSTART is set the program counter (PCx) does not wait for the conversion to complete before incrementing.
5:3	Reserved	R/W	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
3:0	AMUXSEL	R/W	0	Binary encoded selection of 1 of 16 inputs connected to the analog multiplexer for ADCx

ADCx_STC

***Table 3-10* • ADCx_STC**

Bit Number	Name	R/W	Reset Value	Function
7:0	ADCx_STC	R/W	0	ADCx sample time control. t_{sample} = (2 + STC) x t_{ADCCLK} Defines the number of ADCCLK periods that are used for the sample time.

ADCx_TVC

***Table 3-11* • ADCx_TVC**

Bit Number	Name	R/W	Reset Value	Function
7:0	ADCx_TVC	R/W	0	ADCx ACLK divider control. t_{ADCCLK} = 4 x (1 + TVC) x t_{ACLK}

ADCx_MISC_CTRL

***Table 3-12* • ADCx_MISC_CTRL**

Bit Number	Name	R/W	Reset Value	Function
7:6	Reserved	R/W	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
5	PWRDWN_x	R/W	0	1 – Power-down ADC_x 0 – ADC_x is powered on. Note that this bit is logically ORed with bit 2 of the ANA_COMM_CTRL register to generate the power down input signal to the ADC.
4	ADCRESET	R/W	0	1 – Reset ADC_x 0 = No effect. This bit self clears back to 0 after 1 ACLK cycle.
3	CALIBRATE	R/W	0	0 – Perform internal calibration after each conversion. Two ADCCLK cycles are used after the calibration. 1– No calibration after conversion.
2	PWRDWN	R/W	0	0 – Power-down ADC after conversion complete. 1 – No power-down of ADC after conversion complete.
1:0	RESOLUTION	R/W	0	00 – 10-Bit 01 – 12-Bit 10 – 8-Bit 11 – Unused

ADCx_STATUS

Table 3-13 • **ADCx_STATUS**

Bit Number	Name	R/W	Reset Value	Function
31:16	Reserved	R	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
15	CALIBRATE	R	0	1 – ADCx calibrating 0 – ADCx is not calibrating
14	SAMPLE	R	0	1 – ADCx sampling analog input 0 – ADCx is not sampling analog input
13	BUSY	R	0	1 – ADCx is busy converting 0 – ADCx is not busy converting
12	DATAVALID	R	0	1 – ADCx finished conversion 0 – ADCx still converting
11:0	ADC_RESULT	R	0	ADCx converted result

4 – Sigma-Delta Digital-to-Analog Converter (DAC)

Sigma-Delta DAC (SDD) Features

- Voltage or current output modes
- One-bit sigma-delta (ΣΔ) architecture
- First-order ΣΔ modulator included in the analog compute engine (ACE) for each SDD
- The ACE ΣΔ modulators accept 8-bit, 16-bit, and 24-bit unsigned inputs
- Optional 1-bit input from FPGA fabric from custom ΣΔ or pulse-width modulators
- Includes continuous-time third-order low-pass output filter
- Optional fourth output pole with one external capacitor
- Quantization noise versus bandwidth tradeoff
- Nominal 10 KΩ output impedance in voltage mode
- Each SCB can demultiplex any SDD onto an output pin with an optional external hold capacitor
- SDDs can provide reference voltage for one comparator in each SCB
- SDD gain is factory trimmed
- Nearly perfect differential non-linearity
- Low output offset voltage (or current)
- Good gain accuracy and linearity
- Low operating power
- Even lower power sleep mode
- Output loops back to an ADC input

Sigma-Delta DAC General Description

SmartFusion devices contain a number of sigma-delta (SD) digital-to-analog converters (DACs). There are equal numbers of sigma-delta DACs (SDDs) and ADCs in each device: A2F060 devices have one SDD, A2F200 devices have two SDDs, and A2F500 devices have three SDDs. Each SDD has a dedicated output pin, and is looped back internally to channel 15 of the associated ADC input multiplexer. It is also internally routed to some analog switches, routing to one comparator input pin in each signal conditioning block (SCB), where it can be used as an analog output or to generate a reference threshold voltage for the comparator.

The SDD can put out either a voltage or a current, depending upon how it is configured. In current output mode, the nominal output range is from zero to 256 µA, and the SDD has a high output impedance. In voltage output mode, a nominal 10 KΩ internal resistor load is applied, making the nominal output range zero to 2.56 V, with a 10 KΩ output impedance.

When an SDD is disabled, the output is in a high impedance state.

These DACs are based upon the sigma-delta technique. The 8-bit, 16-bit, or 24-bit unsigned binary digital input word to be converted is fed to an all-digital first-order sigma-delta modulator in the analog compute engine (ACE). The binary value is held at the input of the SD modulator until it is overwritten by a new word. The modulator converts the value from a parallel binary number to a high-speed 1-bit serial sequence of one and zero symbols having a duty factor representing the value of the input word.

This one-bit digital sequence goes to a one-bit digital-to-analog converter that puts out an analog current proportional to the VAREF voltage for a one input symbol, and outputs zero current for a zero input symbol. The continuous-time analog output current is held constant for one full clock period for each digital input symbol.

Finally, the current is smoothed by an on-chip continuous-time analog third-order low-pass filter. The output of the analog filter represents the exponentially-weighted time average of the analog current, having an instantaneous value very nearly proportional to the original binary input value, but slightly delayed by the filter. This filtered current can be output directly, or optionally applied to a 10 KΩ resistor (nominal), which converts the current to a voltage. Since in voltage mode the output impedance is roughly 10 KΩ, it is easy to add another low-pass filter pole by attaching a capacitor to ground at the output pin. Depending upon the application requirements, this pole can potentially be at a much lower frequency than the on-chip third-order filter, since the board-level capacitor can be much larger than it could reasonably be if it were integrated on-chip. This flexibility gives the system designer a tradeoff between the effective resolution and output noise of the DAC versus its signal bandwidth.

The DAC reference current is trimmed by Microsemi using factory-determined flash bits for both voltage and current output modes, so the resulting overall SDD gain is nominally correct for whichever mode is selected. The 10 KΩ internal load resistor used in voltage mode is not trimmed; therefore the SDD output impedance in voltage mode will vary with the chip fabrication process.

Sigma-Delta DAC Symbol

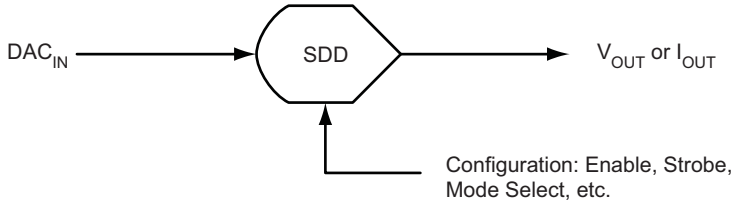


Figure 4-1 **• Sigma-Delta DAC (SDD) Symbol**

The Sigma-Delta Principle

Sigma-delta modulation is a subset of the class of duty factor modulation methods, as is pulse-width modulation (PWM). In all of these, the output duty factor approximates the input to the modulator over some time period. Typically, the input signal changes slowly, and the output of the modulator is clocked at a higher sample rate, but with a lower output resolution. In the limiting (and commonly used) case, the output only has one-bit of resolution.

Unlike pulse-width modulation, in sigma-delta modulation the number of transitions from zero to one or vice versa is not especially well minimized. Pulse-width modulators are often used to switch large currents, such as in power supplies and motor controls, and each transition equates to a significant loss of energy efficiency; therefore it is important to minimize the number of transitions per unit time. In a sigma-delta modulator, the goal instead is to shape the quantization noise frequency spectrum in an ideal way. If most of the quantization noise can be placed in a different part of the output spectrum from the signal of interest, then the signal and the noise can be separated by frequency-selective filters. In the context of a DAC, this means that the filtered output of a one-bit DAC using sigma-delta modulation can perform similarly, in many respects, to a much higher-resolution conventional DAC whose output quantization noise is determined in large part by the number of bits it uses. In the conventional DAC, the quantization noise is usually spread more-or-less evenly across the entire spectrum, and thus cannot be easily separated from the desired signal.

To implement a sigma-delta modulator, a quantizer is placed inside a feedback loop. The block diagrams in Figure 4-2 illustrates this.

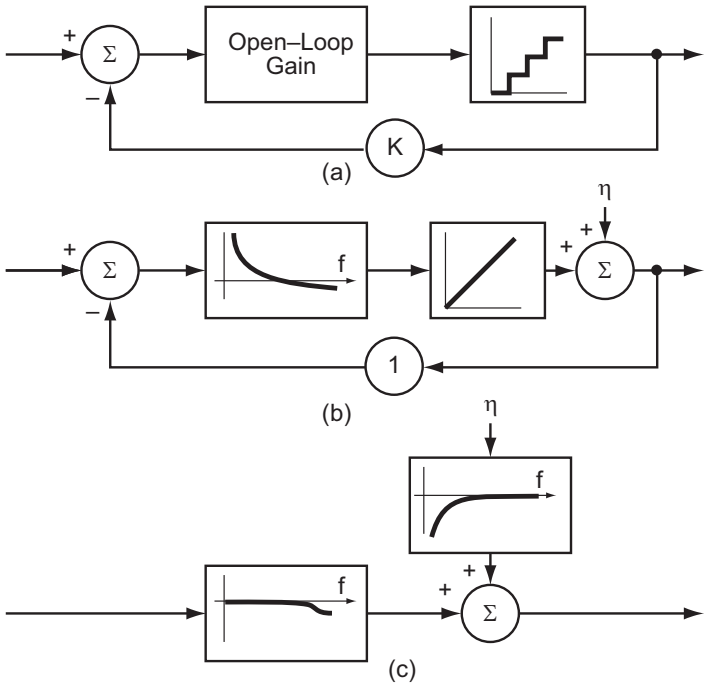


***Figure 4-2* • Generic Sigma-Delta Modulator Block Diagrams**

Figure 4-2 (a) shows the general architecture of a generic sigma-delta modulator. The input is brought to a summing junction, where the approximated output is subtracted, resulting in a signal representing the error between the input and the output. This error is shaped by an open-loop transfer function, and then quantized, to create the output signal.

In Figure 4-2 (b), a typical open-loop gain as a function of frequency is depicted. It has high gain at low frequencies (i.e., the frequencies of interest), and low gain elsewhere. The other elements of the design are not frequency dependent. Note that the frequency response of a continuous time integrator or a sampled-time digital accumulator has this characteristic: an infinite gain at DC which rolls off at higher frequencies. Often several integrators or accumulators are combined in the open-loop gain function to make high-order sigma-delta modulators; or several such lower-order loops are combined. The gain and phase characteristic of the open loop transfer function are chosen so that the overall feedback loop it is contained within remains stable when closed. This can be a substantial challenge for higher-order loops, especially those with highly non-linear one-bit quantizers.

In Figure 4-2 (b), the quantizer has been modeled by an ideal linear response, plus a noise source (η) to represent the quantization noise. Under some assumptions about the input to the quantizer, the frequency spectrum of this noise can be assumed to be nearly flat, or white. It is important to note that the points in the loop where the input signal and the noise are applied are quite different.

In Figure 4-2 (c), the closed loop frequency domain magnitude response of the input-to-output and the noise-to-output transfer functions have been depicted. For the input signal, the high open loop gain (as shown in Figure 4-2 (b)) is in the feed-forward part of the loop; that is, it is in the path between the input summer and the output. Recall that in any feedback system, when the open loop gain is high, the closed loop transfer function is determined mainly by the (reciprocal of the) feedback path gain. The feedback gain is unity for this input and output combination. One thus concludes that the closed loop gain of the system from input to output is therefore very near unity at low frequencies where the open loop gain is high. The input signal is passed with little modification in this part of the frequency range. At higher frequencies the gain may fall slightly below unity because there is not much open loop gain; but that falls outside of the frequency range of interest.

For the quantization noise, the story is quite different. Considering the noise as the input to the loop, the high gain element is in the feedback path; that is, it is in the path between the output and the summer where the noise is added. The feed-forward gain path is unity. As before, at frequencies where there is a high open loop gain, the transfer function is mainly determined by the reciprocal of the feedback gain. In this case the feedback gain is very high, therefore the closed-loop transfer function from the noise input to the output is very low at low frequencies. The noise spectrum has been shaped so that most of the noise energy is in the higher frequency portions of the spectrum.

In summary, at low frequencies the input signal is represented virtually unchanged, and there is not much quantization noise. At higher frequencies the input signal may have some gain and phase error, and a large amount of quantization noise. Therefore, by filtering away the high frequencies (that step is not shown in Figure 4-2 on page 51), we are left with a high-fidelity representation of the low frequency portions of the input signal.

The trick in sigma-delta modulation is that the quantization noise, which can be very substantial for a quantizer with only one or a few bits, is forced into a different part of the frequency spectrum from the desired signal. This alternate representation of the signal (plus noise) might only require one or a few bits, but the useful frequency range for the signal is usually only a small fraction (10% or even much less) of the sampling frequency. However, because the signal and noise are spectrally separated, a high fidelity copy of the signal can be obtained from the sigma-delta modulated version by filtering away the frequency bands where the quantization noise is present, leaving only the frequency band where the signal is, with the signal faithfully reconstructed minus the noise.

The sigma-delta representation is sometimes useful for purposes such as data transmission, analog-to-digital conversion, and digital-to-analog conversion, since it uses fewer bits than the original representation. In this case, the SmartFusion device can use a one-bit DAC and achieve similar performance to a conventional multi-bit DAC at a lower cost. In some measures, such as differential non-linearity, the one-bit architecture is far superior to the conventional approach.

Sigma-Delta DAC Detailed Description

The block diagram in Figure 4-3 shows the main sections of the SmartFusion sigma-delta DAC (SDD). The sigma-delta DAC is comprised of three main blocks: a sigma-delta modulator, a one-bit DAC, and an analog low-pass filter.

Unlike many predominantly analog circuits, a significant and critically important portion of the SDD, namely the sigma-delta modulator, is implemented entirely in digital logic. The sigma-delta modulator approximates the input binary words with a stream of shorter words using a sigma-delta style algorithm. In the case of the SmartFusion sigma-delta modulators, the input values are 8-bit, 16-bit, or 24-bit unsigned binary words, and the output is a stream of one-bit binary symbols; each output symbol is either a zero or a one {0, 1}. The SmartFusion built-in modulators have first-order dynamics. These modulators are implemented as part of the analog compute engine logic. There is one modulator dedicated to each SDD. Sigma-delta modulators belong to the class of duty factor modulators. The average number of one output symbols (relative to the total number of symbols in a given time period) is directly proportional to the input value to the modulator.

The one-bit digital symbol is converted to a continuous time analog waveform by the internal one-bit DAC. The input data and clock for the one-bit DAC is usually the sigma-delta modulator in the ACE, but it is also possible to bit-bang the one-bit input data by writing a one or a zero to a certain register in the ACE (refer to the "Sigma-Delta DAC Configuration" section on page 60). Also, it is possible to select a data bit and a clock routed from the FPGA fabric. This allows for the possibility of configuring higher-order sigma-delta modulators in the FPGA fabric; or some fundamentally different style of one-bit modulator can be designed to use the one-bit DAC, such as a pulse-width modulator.

In approximating a higher-precision input word with a one-bit signal, a rather substantial amount of quantization noise is added to the desired signal. After all, a one bit signal is a poor approximation to a 16-bit signal. Much of this quantization noise can be filtered away by the third main block in the sigma-delta DAC: an analog low-pass filter. The output of the low-pass filter is an exponentially-weighted average of the DAC output waveform. Since the number of ones going into the DAC is proportional to the binary word input to the SDD (at least over the medium or long term), the output of the analog filter is an analog signal which is very nearly proportional to the input binary word; especially for lower frequency inputs.

Since the quantization noise is not white but is shaped by the sigma-delta modulator, the analog filter can do a much better job of removing it. Furthermore, as the bandwidth of the filter is decreased (such as can be done by adding a board-level capacitor to the output), the quantization noise is rapidly reduced due to its spectral shaping.

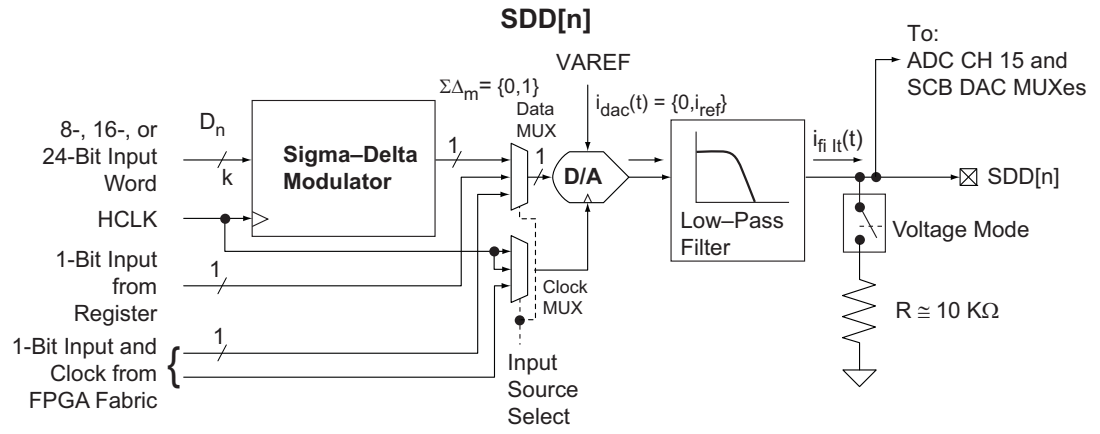


***Figure 4-3* • SmartFusion Sigma-Delta DAC Block Diagram**

The Smart Fusion Sigma-Delta Modulators

Figure 4-4 shows conceptually how the built-in sigma-delta modulators in SmartFusion devices are implemented. There is one modulator in the ACE for each sigma-delta DAC. They work with an N-bit unsigned binary input, where N is 8, 16, or 24.

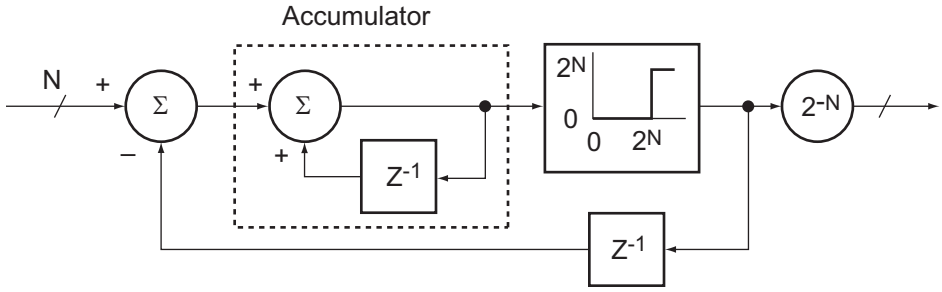


***Figure 4-4* • SmartFusion Sigma-Delta Modulator Block Diagram**

The frequency-dependent open loop function used in a SmartFusion cSoC is a first-order digital accumulator. It has infinite gain at DC, and rolls off approximately as 1/f until it gets near the Nyquist frequency, where the gain flattens out to unity. Since the accumulator has infinite gain at DC, the quantization noise is completely suppressed at DC. Any long-term error between the input and output is accumulated with infinite gain, causing the feedback system to ultimately correct itself.

The quantizer is a simple one-bit digital comparator. If the accumulated error is equal to or exceeds 2^N, then 2^N is fed back and subtracted from the next input to the accumulator. In that case, a one symbol is output from the modulator. If the error is less than 2^N, then nothing is fed back and a zero symbol is output. The delay element in the feedback is required to make the feedback loop realizable (i.e., non-algebraic).

A simple example may serve to illustrate how the sigma-delta modulator computes the correct duty factor for the output bit stream. Suppose that N = 8, so 2^N = 256. Assume that the input is a constant value of 100. On the first clock cycle, the accumulator will go from a stored state value of zero, to 100. This is less than 256, so a zero symbol is output from the modulator. On the next cycle the accumulator increases to 200, and another zero is output. On the third cycle, the accumulator goes to 300. This is greater than or

equal to 256, so a one symbol is output from the modulator. Also, 256 is subtracted from the accumulator on the next cycle, so instead of going to 400, the accumulator only goes to 144, and since this is less than 256 a zero is output for that cycle. This sequence continues, with the accumulator state sequence being (0), 100, 200, **300**, 144, 244, **344**, 188, **288**, 132, 232, **332**, 176, **276,** 120, 220, etc., and the corresponding output bits from the modulator being (0), 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, etc. You can observe that there are somewhat fewer than 50% ones, as would be expected for an ideal long-term duty factor of 100/256. Since 100 is added to the accumulator on each clock cycle, and it takes a value of 256 or more to trigger a one output symbol, it takes, on average, 2.56 clock cycles to accumulate 256 and output a one.

Another way of viewing this is that the output of the modulator is the carry-out bit of the accumulator. Whenever the total in the accumulator exceeds 255, the carry-out bit sets the output of the sigma-delta modulator to a one, and then the carry-out is discarded, effectively dropping the accumulator value by 256.

The sigma-delta modulator performs another important signal processing function: It interpolates the input samples it is provided by holding the most recent one until it is replaced by a new sample. The built-in sigma-delta modulators are clocked by HCLK, which is typically in the range of 80 MHz to 100 MHz. Input samples are usually provided at a much lower rate; from DC to a few hundred KHz, in most cases. The input register of the sigma-delta modulators will hold the most recent binary input sample. For example, if the end application is generating samples for the SDD at a 10 KHz rate and the HCLK frequency is 100 MHz, then each input sample is used for 10,000 HCLK samples of the sigma-delta modulator. During this time, the output duty factor will have an average value close to the input number divided by the full-scale value chosen (i.e., 2^8 = 256, 2^{16} = 65,536, or 2^{24} = 16,777,216).

The One-Bit Digital-to-Analog Converter

The one-bit DAC converts the digital {0, 1} symbols coming from the sigma-delta modulator (or from one of the alternate sources) to a continuous-time analog signal current. A zero input symbol is converted to nominally zero Amps and a one input symbol is converted to nominally 256 µA. Each of these output levels is held from one rising edge of HCLK to the next rising edge, resulting in a rectangular waveform whose exact shape is set by the input sequence.

The reference current for this conversion process is derived from the VAREF voltage using the SmartFusion internal bandgap voltage reference. Note that the internal VAREF is used even if an external reference is used for the corresponding ADC. This voltage (and thus the DAC reference current) should be relatively immune to variations in the SmartFusion power supply voltages, and to variations in temperature. The VAREF voltage, and the DAC current and voltage gains, are trimmed using digital flash bits at the Microsemi factory to remove most process-related variations.

The action of holding the sample for one HCLK cycle has a subtle effect on the frequency response of the DAC. This function is often referred to in the electrical engineering literature as a zero-order hold. The effect is a slight linear phase shift (i.e., a constant group delay) equal to one-half the clock period, and a magnitude response in the shape of a sin(x)/x function with the peak at DC and the nulls at all multiples of the sampling frequency. Since the sampling frequency of the one-bit DAC is so high relative to the frequencies of interest, this effect can usually be ignored. Note that conventional DACs also implement a zero-order hold function, though the effect is generally much more pronounced in them since the sampling rate is usually much closer to the frequency of the signals being processed.

Since the rectangular steps in the output of the one-bit DAC are full scale in amplitude, any clock jitter has a relatively significant effect on the output noise. Therefore, it is important to use a clock source with low jitter if low output noise is desired from the SDD.

The Analog Low-Pass Filter

The analog filter at the output of the internal one-bit DAC is an important part of the overall sigma-delta DAC architecture. The sigma-delta modulator adds a significant amount of quantization noise to the input signal since it represents the input signal using only one bit. However, rather than the quantization noise being spread evenly over the frequency spectrum, it is forced away from the frequencies of interest at the lower end of the spectrum and is pushed into the higher frequency region. This allows much of the quantization noise to be filtered away by the analog output filter.

Each SDD in a SmartFusion cSoC has a dedicated on-chip third-order continuous-time low-pass filter. This is formed using three real poles, all located near 300 KHz, giving the on-chip filter a 60 dB/decade noise roll-off slope above the corner frequency. Since the first-order Sigma-Delta noise shaping gives the quantization noise spectrum an upward slope that is roughly proportional to frequency (i.e., +20 dB/decade) at low frequencies, and the analog filter is providing a downward slope roughly inversely proportional to frequency cubed, it is apparent that the filter quickly overtakes and rapidly reduces the noise spectrum above its corner frequency. See Figure 4-5 graph (a).

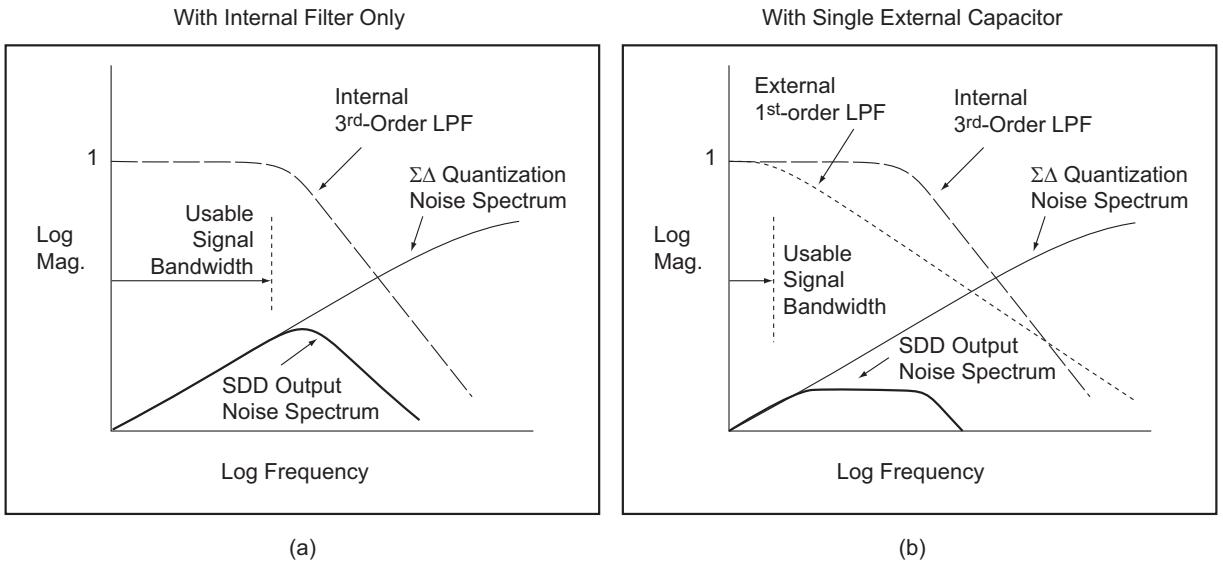


***Figure 4-5* • Effect of External Capacitor on Sigma-Delta Quantization Noise and Signal Bandwidth**

It is possible to add additional off-chip filtering. The usual reason for doing this would be to reduce the filter cut-off frequency in order to filter away more of the quantization noise (at the expense of signal bandwidth). In many cases, the signal frequencies of interest are well below 300 KHz; sometimes only signals at or near DC are of interest. Audio signals, for example, may only need a bandwidth of 8 KHz (for voice or most audible tones), or 20 KHz at most, (for high fidelity music). Another reason for a higher-order filter would be if a custom higher-order sigma-delta modulator is implemented in the FPGA fabric. In this case, a forth or higher order filter might have some advantages in rolling off the quantization noise spectrum. While external passive or active multi-pole filters are possible, the simplest approach is just to add a single passive board-level capacitor from the SDD output pin to ground, with the SDD in voltage mode. This adds a fourth pole to the analog filter that can easily be customized by the selection of the capacitor. See Figure 4-5, graph (b).

In voltage mode, the SDD has an output impedance of roughly 10 KΩ. A dominant single-pole low frequency corner can be obtained with a single physically small external capacitor. For example, a 0.1 µF, 6.3 V, ±10% tolerance capacitor, with X7R dielectric (±15% over the temperature range of -55°C to +125°C) is available in a small 0402 (1005 metric) surface-mount package. This gives a nominal corner frequency of 159 Hz, corresponding to a 1 ms time-constant. Exponential settling to a step input for a single dominant pole to 0.1% of the final value takes roughly 7 time-constants, or about 7 ms.

Alternatively, for a fourth pole at approximately 300 KHz, an external 50 pF capacitor may be used. Each of the four cascaded 300 KHz poles has a time-constant of roughly 0.5 µs; none is dominant so all must be taken into consideration. Settling to 0.1% for a step input takes roughly 7 µs. Many options are possible, with tradeoffs for precision, size, temperature range, temperature stability, settling time, bandwidth, and so on. Since the on-chip filter components, and the internal 10 KW load resistor are not trimmed, if a very precise corner frequency is desired, then a precision off-chip load resistor should be used as well as a precision capacitor (for example, using NPO dielectric), with the SDD operated in current output mode. The roll-off of a first-order low-frequency low-pass filter set by a single external capacitor will just cancel the rising noise spectrum due to the first-order sigma-delta noise shaping, resulting in a rising (proportional to frequency) noise power spectrum from DC to the filter corner frequency; a flat spectrum from the filter corner frequency up to the 300 KHz on-chip filter cut-off; and a

quickly falling noise spectrum above 300 KHz (inversely proportional to frequency cubed). See Figure 4-5 on page 55, graph (b). The total integrated noise will be dominated by the central flat plateau, the level of which is proportional to the low single-pole filter cut-off frequency. For cut-off frequencies well below 300 KHz, where the noise bandwidth is not changing appreciably, the total integrated noise falls nearly linearly as the low-pass filter bandwidth is decreased. For example, if the filter cut-off is reduced from 10 KHz to 1 KHz, the RMS noise contributed from quantization effects will drop by roughly a factor of ten. This is equivalent to adding more than three bits to the effective resolution of the SDD. This is in contrast to a conventional DAC, where the quantization noise is flat in the signal passband, and the total integrated noise only falls roughly as the square-root of the bandwidth (or only about three-to-one for the example given, adding a little more than one bit of improved noise performance). If a second-order external low-pass filter is used, the effect of bandwidth on quantization noise is even greater because both the power spectral density and the bandwidth are reduced when the filter corner is lowered.

Time-Domain and Frequency-Domain Interpretation

This section gives a qualitative description of the signal processing that occurs in the various stages of the sigma-delta DAC. The description is not mathematically rigorous, but it may increase understanding of the principal of operation of the sigma-delta DAC, especially if there are plans to customize it in some way. In many applications, especially where the signals being converted are at DC or relatively low frequencies and are sufficiently over-sampled, it is not required to have a detailed understanding of the following material in order to use the SDD effectively.

Refer to Figure 4-3 on page 53 for the location of the signals in the following plots. Figure 4-6 shows sketches of key signal time-domain waveforms and their corresponding frequency-domain magnitude spectrums. The sketches are qualitative only, and are not drawn to scale.

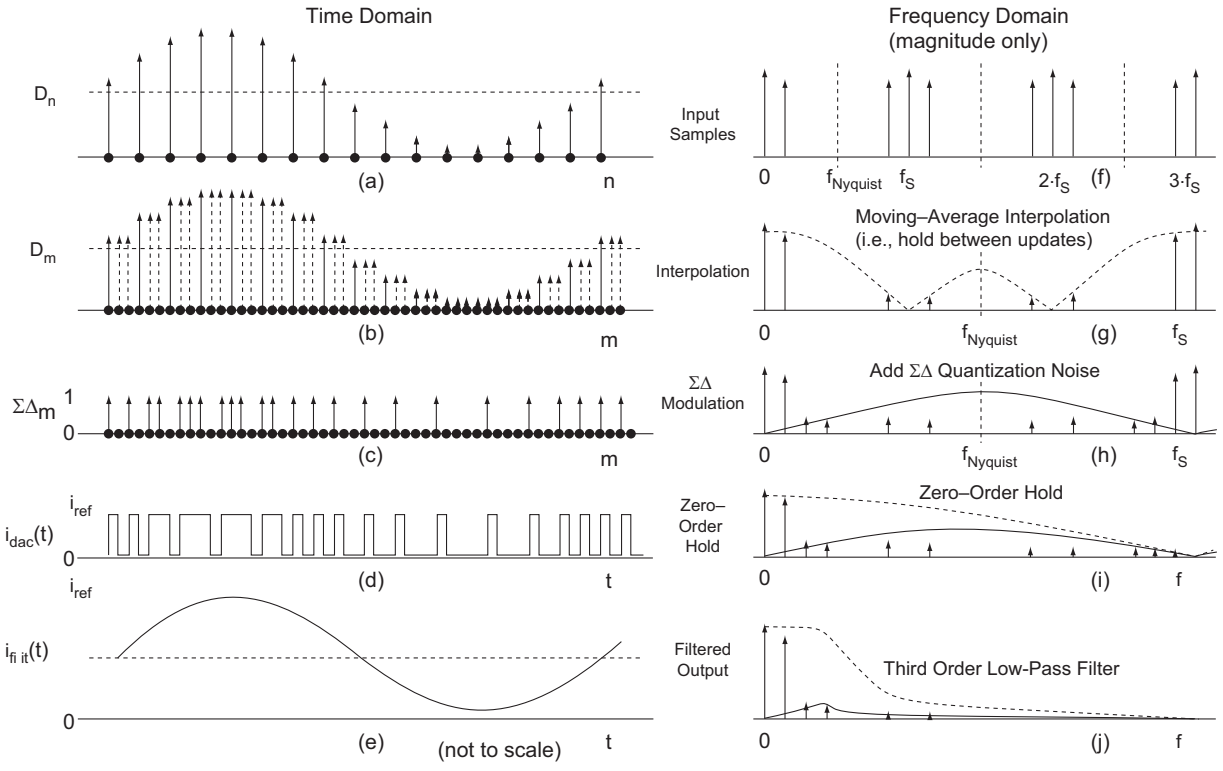


***Figure 4-6* • Sketches of Key Sigma-Delta DAC Signals in the Time-Domain and the Frequency-Domain**

Input Samples

Figure 4-6, graph (a) shows a representative sampled offset sine-wave input signal comprised of Dirac delta function impulses at the sampling instants. The signal is represented as an indexed sequence of binary unsigned numbers; D_n mid-scale is indicated with a dotted line for reference purposes only.

Figure 4-6, graph (f) shows the frequency spectrum of the signal in graph (a). Since the signal is periodically sampled, its representation in the frequency domain is also periodic, as described by the well-known Sampling Theorem. Since the input signal is real, its frequency domain representation is symmetric about DC. The positive and negative frequency phasors forming the sine wave repeat about every multiple of the sampling frequency, f_S, in what are called "images" of the lowest frequency frequency-domain impulse. The half-scale offset creates a large DC term. When these factors are considered, it is apparent that the spectrum from DC to the Nyquist frequency (½ f_S) is mirrored from the Nyquist frequency to the sampling frequency, and again from the sampling frequency to 1.5 times the sampling frequency, ad infinitum. The DC component is periodic in the frequency domain at the sampling frequency.

Interpolation and Reconstruction

In Figure 4-6 on page 56, graph (b), the effect of resampling the input signal on the usually much higher frequency HCLK is shown, and the index for the resulting sequence D_m is shown as m. In the figure, the input is resampled at only a three times higher frequency, so the sketch is easier to understand. Some more typical values might be where the input sampling frequency is 100 KHz and the resampling frequency is 100 MHz, for a 1,000 to 1 ratio. Not only is the signal resampled, but the fact that each sample is held (replicated) until the next input sample effectively provides a simple reconstruction filter. Holding has the same effect as passing the samples through an N-tap equally-weighted (first-order) moving average finite impulse response (FIR) filter, where N is the resampling ratio. In a typical example, each input is held for 1,000 clocks, until the next input sample arrives. (In the figure, for clarity, the original samples are replicated twice more, for a total of three identical samples in a row.)

Figure 4-6 on page 56, graph (g) is a sketch of the corresponding frequency-domain representation of the interpolated and reconstructed signal. The first-order moving-average filter has a well-known sin(x)/x (also called a "sinc" function) magnitude response. This has zero crossings at all multiples of the original sampling frequency except at DC, where it has a relatively flat passband, as shown with the dotted line. While the resampling ratio for the sketch was only 3:1, recall that much larger values are more typical. N – 1 of the images of the original sine-wave fall near to these notches and are partially suppressed. Still being a real (versus a complex) sampled signal, it is periodic and symmetric about DC in the frequency domain, but the sampling frequency is now much higher, so the main images (every Nth one of the original images) are now further apart. How well the original intermediate images are suppressed depends upon the various frequencies involved. If the sine wave was fairly highly over-sampled originally, then its images are proportionally closer to the sampling frequency and its multiples, thus deeper in the notches of the sinc filter, and more heavily filtered. Conversely, if the original sine-wave signal was not very heavily over-sampled, then the effect of the reconstruction filter on the passband will be more significant—the main sine wave frequency-domain impulse will not be in the flattest part of the response—and the image impulses will be more prominent in the result. The original images of the DC component are right at the zero crossings of the sinc function, and are completely suppressed, except for those corresponding to multiples of the new, higher, sampling frequency.

Sigma-Delta Modulation

The next step is for the digital sigma-delta modulator to convert the signal to a one-bit duty factor modulated stream of symbols. This is shown in Figure 4-6 on page 56, graph (c). Where the input value is high, such as at the peaks of the sine wave, the sigma-delta modulator outputs a higher percentage of one symbols; where low, the zero symbols predominate. The sequence $\Sigma\Delta_m$ can be thought of as a very noisy one-bit version of the sequence D_m that is the input to the modulator.

In the frequency domain, as shown in Figure 4-6 on page 56, graph (h), all the signals from graph (g) are still present, to some degree or another. The low frequency signals, such as at DC and the main images of the sine wave, are relatively faithfully reproduced. That is, as described above, due to the high internal loop gain of the sigma-delta modulator at these frequencies which drives the loop error signal to zero over the long term. However, a substantial amount of energy has been transferred to the quantization

noise. The exact characteristics of the modulator output sequence depends upon the details of the input sequence and the modulator design, but generally the quantization noise is low at low frequencies and gets larger closer to the Nyquist frequency, as shown in Figure 4-6 on page 56. Due to the highly non-linear nature of the quantizer, some harmonics of the input frequencies may appear. For higher-order sigma-delta modulators, the noise shaping spectrum will be more predictable and pseudo-random in nature with less energy in the signal harmonics; for lower order modulators the noise may be more periodic and dependent upon the specific input values and frequencies. The important feature of a sigma-delta modulated sequence is that, while the total quantization noise is large, most of it is concentrated in a different part of the frequency spectrum than the signal of interest.

Digital-to-Analog Conversion

As shown in the Figure 4-6 on page 56, graph (d), the sequence of sampled digital symbols output from the sigma-delta modulator is fed through the one-bit DAC and converted to a continuous-time analog waveform. The DAC relies upon the internal 2.56 V voltage reference to generate a (nominally) 256 µA output current for a one symbol, and zero output current for a zero symbol that is independent of power supply voltages or temperature. The output of the DAC is held for one HCLK period, at which time the next input symbol arrives. This is called a zero-order hold function; for each input impulse (sample), it outputs a short continuous-time rectangular pulse whose amplitude is proportional to the input. In the case of a one-bit DAC, there are only two possible output levels.

Figure 4-6 on page 56, graph (i) shows the effect in the frequency domain. The zero-order hold function effectively acts like a low-pass filter, with a sin(x)/x shape, corresponding to the Fourier Transform of the rectangular pulse-shaped impulse response it provides. Since the signal is now continuous time, the sampling theorem no longer applies. The images in the DAC's sampled input stream are filtered by the low-pass filter action of the zero-order hold. The signals of interest, such as at DC and other low frequencies (up to around 300 KHz) are passed relatively unchanged. The images of a DC input that were exactly at multiples of the sampling frequency (100 MHz, for example) are completely suppressed, since they are precisely at the points where the sin(x)/x magnitude function goes through zero; the images of low frequency (highly over-sampled) sine waves are substantially reduced by the zero-order hold as they are near to these notches in the frequency response occurring at the multiples of the sampling frequency. Much of the quantization noise is also passed by the zero-order hold, with some reduction in amplitude for the images of the noise that were above the Nyquist frequency.

Low-Pass Filtering and Output

In the last figures in the sequence, Figures 6(e) and (j), the result of the analog low-pass filter is depicted. The analog filter removes the vast majority of the energy attributable to the quantization noise. As stated several times before, this noise is largely at higher frequencies (a broad range centered about the DAC's Nyquist frequency, for example 50 MHz), while the signals of interest are at relatively low frequencies (for example, DC to 300 KHz). Not only is this typically less than 1% of the former sampling frequency, the quantization noise spectral power density is substantially lower within this frequency range than elsewhere; therefore, most of the noise is removed by the analog filter. This leaves (predominantly) the signals of interest, such as the continuous-time DC-offset sine wave as depicted in the time and frequency domains in the figures, along with some small residual noise and distortion due to harmonics and other frequency spurs such as from the sampling-induced images or from intermodulation effects.

The output of the sigma-delta DAC can be a positive output current ranging from zero to 256 µA, or alternatively, a nominal 10 KΩ resistor can internally be switched in as a load, converting the output current to a voltage with a range of zero to 2.56 V.

Shared Pins

Each sigma-delta DAC has a dedicated output pin. The voltage on this pin is also fed back to the associated ADC's input multiplexer on channel 15.

Furthermore, each sigma-delta DAC output is routed to a multiplexer in each signal conditioning block (SCB) which can be used to select the output from any of the DACs present on the chip. Each of these multiplexers can be connected to another SmartFusion pad via an analog switch.

This topology allows the any of the DAC outputs to be used as a reference input for one comparator in each SCB. One DAC can potentially service all such comparator input pins—up to five in the A2F500—while the remaining DACs are used for other purposes. See the "Comparator Configuration" section on page 94 for information on the sample-and-hold mode of operation.

Table 4-1 relates each of the sigma-delta DACs with their output pin and the pins in each SCB that the DAC can be programmed to drive. Though the sigma-delta DAC multiplexer functions (SDDM[n]_OUT) are associated with a particular ADC and SCB, as shown in Table 4-1, the multiplexer inputs come from every available sigma-delta DAC on the chip, regardless of which SCB the multiplexer resides in. Therefore, any of the odd-numbered ADC direct input pads can alternatively be used as an output pad for any of the available on-chip DACs.

Table 4-1 • **SDD Input/Output Pad Designations**

ADC	Signal Conditioning Block	DAC Name	Pad Name	Function Name	Shared Functions on Pad		
ADC0	(N/A)	SDD0	SDD0	SDD0_OUT	ADC0_CH15		
	SCB0		ADC1	SDDM0_OUT	ADC0_CH10	LVTTL1_IN	CMP1_N
	SCB1		ADC3	SDDM1_OUT	ADC0_CH12	LVTTL3_IN	CMP3_N
ADC1	(N/A)	SDD1	SDD1	SDD1_OUT	ADC1_CH15		
	SCB2		ADC5	SDDM2_OUT	ADC1_CH10	LVTTL5_IN	CMP5_N
	SCB3		ADC7	SDDM3_OUT	ADC1_CH12	LVTTL7_IN	CMP7_N
ADC2*	(N/A)	SDD2	SDD2	SDD2_OUT	ADC2_CH15*		
	SCB4*		ADC9	SDDM4_OUT	ADC2_CH10	LVTTL9_IN	CMP9_N

*Note: *Components indicated are only available in the A2F500.*

Sigma-Delta DAC Configuration

The sigma-delta DACs can be dynamically configured via registers in the analog compute engine (ACE). These registers can be controlled by the ACE itself, a Cortex-M3 program, by user logic (or soft processor) via the AHB bus master connected to the FPGA fabric, or by the peripheral DMA bus master.

Each DAC has a control register and three one-byte data registers in the ACE as input to the sigma-delta modulator. As an alternative to the three 1-byte data registers, there is also a 16-bit input data register for each modulator. In addition, there are a few bits scattered about in several of the ACE registers associated with configuring the signal conditioning blocks (SCBs) that are dedicated to SDD functions (for example, selecting current vs. voltage output mode). Finally, there is a register which can force simultaneous updates of the input data to several of the sigma-delta modulators at the same instant.

Triggering an update to the built-in sigma-delta modulators can occur in one of three ways, depending upon how the configuration bits are set:

1. If synchronous updates are enabled for the given SDDs, then writing a 1 to the designated update bits for the desired SDDs in the DAC_SYNC_CTRL register will force the input to the modulator(s) to be updated. Since the update bits for all the SDDs are in the same register, all the SDDs can be updated on the same clock cycle, if desired. Naturally, the input data should be staged in the appropriate data input registers before commanding the update.
2. If synchronous updates are not enabled, then writing to the least-significant byte of the three one-byte registers for a given SDD will cause an update—if they are configured as the input data source. The upper bytes should be staged in advance of writing the LSB, in this case.
3. Writing data to the SDD's two-byte input register will cause an update, if it is the selected source (and synchronous updates are not enabled).

For more information on how to use the demultiplexer and analog switch in each of the signal conditioning blocks for selecting and possibly holding (with an optional board-level capacitor) a sigma-delta DAC output on a comparator reference (inverting) input pin, see the "High-Speed Comparators" section on page 91.

Table 4-2 on page 61 gives a list of all the relevant registers for controlling the sigma-delta DACs. Table 4-3 on page 62 and Table 4-4 on page 64 show the bit fields, and describe their functions. Only those functions that are used for the sigma-delta DAC functions are shown; other bits in the same registers used for other functions are not listed here.

***Table 4-2* • ACE Registers Used in Configuring or Using the Sigma-Delta DACs**

Register Name	Address	R/W	Reset Value	Description
DAC_SYNC_CTRL	0x40020010	R/W	0	Common SDD control
DAC0_CTRL	0x40020060	R/W	0	SDD0 control
DAC0_BYTE0	0x40020064	R/W	0	SDD0 byte 0
DAC0_BYTE1	0x40020068	R/W	0	SDD0 byte 1
DAC0_BYTE2	0x4002006C	R/W	0	SDD0 byte 2
DAC1_CTRL	0x400200A0	R/W	0	SDD1 control
DAC1_BYTE0	0x400200A4	R/W	0	SDD1 byte 0
DAC1_BYTE1	0x400200A8	R/W	0	SDD1 byte 1
DAC1_BYTE2	0x400200AC	R/W	0	SDD1 byte 2
DAC2_CTRL *	0x400200E0	R/W	0	SDD2 control
DAC2_BYTE0 *	0x400200E4	R/W	0	SDD2 byte 0
DAC2_BYTE1 *	0x400200E8	R/W	0	SDD2 byte 1
DAC2_BYTE2 *	0x400200EC	R/W	0	SDD2 byte 2
SCB0_B6	0x4002021C	R/W	0	SCB0 byte 6
SCB0_B10	0x4002022C	R/W	0	SCB0 byte 10
SCB0_B11	0x40020230	R/W	0	SCB0 byte 11
SCB1_B10	0x4002025C	R/W	0	SCB1 byte 10
SCB1_B11	0x40020260	R/W	0	SCB1 byte 11
SCB2_B6	0x4002027C	R/W	0	SCB2 byte 6
SCB2_B10	0x4002028C	R/W	0	SCB2 byte 10
SCB2_B11	0x40020290	R/W	0	SCB2 byte 11
SCB3_B10	0x400202BC	R/W	0	SCB3 byte 10
SCB3_B11	0x400202C0	R/W	0	SCB3 byte 11
SCB4_B6 *	0x400202DC	R/W	0	SCB4 byte 6
SCB4_B10 *	0x400202EC	R/W	0	SCB4 byte 10
SCB4_B11 *	0x400202F0	R/W	0	SCB4 byte 11
SSE_DAC0_BYTES01	0x40020500	R/W	0	SSE SDD0 byte 0 and byte 1
SSE_DAC1_BYTES01	0x40020504	R/W	0	SSE SDD1 byte 0 and byte 1
SSE_DAC2_BYTES01 *	0x40020508	R/W	0	SSE SDD2 byte 0 and byte 1 *

*Note: * Components or functions indicated are only available in the A2F500*

Table 4-3 and Table 4-4 on page 64 show the bits and describe their functions for the registers controlling sigma-delta DACs.

***Table 4-3* • Relative/Absolute Addresses for Sigma-Delta DAC Related Configuration Control Bits**

SCB Byte and Bit Address (relative to each SCB) or ACE Register Name	ABM Address (absolute 32-bit address) [1]	ACE/SSE/PPE Address (absolute 8-bit address)	Bit Name	Description
Common DAC Control Register (0x40020010/0x04)				
DAC_SYNC_CTRL[0]	0x40020010[0]	0x04[0]	DAC_SYNC_UPDATE	SDD0 synchronized update
DAC_SYNC_CTRL[1]	0x40020010[1]	0x04[1]	DAC_SYNC_UPDATE	SDD1 synchronized update
DAC_SYNC_CTRL[2][2]	0x40020010[2	0x04[2]	DAC_SYNC_UPDATE	SDD2 synchronized update
DAC_SYNC_CTRL[4]	0x40020010[4]	0x04[4]	DAC_SYNC_EN	SDD0 synchronized update enable
DAC_SYNC_CTRL[5]	0x40020010[5]	0x04[5]	DAC_SYNC_EN	SDD1 synchronized update enable
DAC_SYNC_CTRL[6] [2]	0x40020010[6]	0x04[6]	DAC_SYNC_EN	SDD2 synchronized update enable
SDD0 Control and One-Byte Data Registers (0x40020060/0x18 through 0x4002006C/0x1B)				
DAC0_CTRL[1:0]	0x40020060 [1:0]	0x18 [1:0]	DAC_SEL[1:0]	SDD0 input bit source selection
DAC0_CTRL[3:2]	0x40020060 [3:2]	0x18 [3:2]	DAC_RES[1:0]	SDD0 modulator input resolution
DAC0_CTRL[4]	0x40020060 [4]	0x18 [4]	SW_OBD	SDD0 input bit
DAC0_CTRL[5]	0x40020060 [5]	0x18 [5]	EN	SDD0 enable
DAC0_CTRL[6]	0x40020060 [6]	0x18 [6]	REG_SEL	SDD0 modulator input selection
DAC0_BYTE0[7:0]	0x40020064 [7:0]	0x19[7:0]	DAC_BYTE0[7:0]	SDD0 input data byte 0
DAC0_BYTE1[7:0]	0x40020068 [7:0]	0x1A[7:0]	DAC_BYTE1[7:0]	SDD0 input data byte 1
DAC0_BYTE2[7:0]	0x4002006C [7:0]	0x1B[7:0]	DAC_BYTE2[7:0]	SDD0 input data byte 2
SDD1 Control and One-Byte Data Registers (0x400200A0/0x28 through 0x400200AC/0x2B)				
DAC1_CTRL[1:0]	0x400200A0 [1:0]	0x28 [1:0]	DAC_SEL[1:0]	SDD1 input bit source selection
DAC1_CTRL[3:2]	0x400200A0 [3:2]	0x28 [3:2]	DAC_RES[1:0]	SDD1 modulator input resolution
DAC1_CTRL[4]	0x400200A0 [4]	0x28 [4]	SW_OBD	SDD1 input bit
DAC1_CTRL[5]	0x400200A0 [5]	0x28 [5]	EN	SDD1 enable
DAC1_CTRL[6]	0x400200A0 [6]	0x28 [6]	REG_SEL	SDD1 modulator input selection
DAC1_BYTE0[7:0]	0x400200A4 [7:0]	0x29[7:0]	DAC_BYTE0[7:0]	SDD1 input data byte 0
DAC1_BYTE1[7:0]	0x400200A8 [7:0]	0x2A[7:0]	DAC_BYTE1[7:0]	SDD1 input data byte 1
DAC1_BYTE2[7:0]	0x400200AC [7:0]	0x2B[7:0]	DAC_BYTE2[7:0]	SDD1 input data byte 2

Notes:

1. *Only the least-significant byte of each 32-bit data word is utilized*
2. *Components or functions indicated are only available in the A2F500.*
3. *ABM = AHB bus matrix; SCB = signal conditioning block; ACE = analog compute engine; SSE = sample sequencing engine; PPE = post processing engine.*
4. *Return-to-Zero (RTZ) mode is deprecated. The NO_CHOPPING bit should always be set to 1 for NRTZ mode.*

***Table 4-3* • Relative/Absolute Addresses for Sigma-Delta DAC Related Configuration Control Bits (continued)**

SCB Byte and Bit Address (relative to each SCB) or ACE Register Name	ABM Address (absolute 32-bit address) [1]	ACE/SSE/PPE Address (absolute 8-bit address)	Bit Name	Description
SDD2 Control and One-Byte Data Registers (0x400200E0/0x38 through 0x400200EC/0x3B) 2				
DAC2_CTRL[1:0] [2]	0x400200E0 [1:0]	0x38 [1:0]	DAC_SEL[1:0]	SDD2 input bit source selection
DAC2_CTRL[3:2] [2]	0x400200E0 [3:2]	0x38 [3:2	DAC_RES[1:0]	SDD2 modulator input resolution
DAC2_CTRL[4] [2]	0x400200E0 [4]	0x38 [4]	SW_OBD	SDD2 input bit
DAC2_CTRL[5] [2]	0x400200E0 [5]	0x38 [5]	EN	SDD2 enable
DAC2_CTRL[6] [2]	0x400200E0 [6]	0x38 [6]	REG_SEL	SDD2 modulator input selection
DAC2_BYTE0[7:0] [2]	0x400200E4 [7:0]	0x39[7:0]	DAC_BYTE0[7:0]	SDD2 input data byte 0
DAC2_BYTE1[7:0] [2]	0x400200E8 [7:0]	0x3A[7:0]	DAC_BYTE1[7:0]	SDD2 input data byte 1
DAC2_BYTE2[7:0] [2]	0x400200EC [7:0]	0x3B[7:0]	DAC_BYTE2[7:0]	SDD2 input data byte 2
SCB_0 (0x40020204/0x81 through 0x40020230/0x8C) – Including SDD0 Bits				
B6[0]	0x4002021C [0]	0x87 [0]	CUR_VOLB	SDD0 current mode select
B6[1]	0x4002021C [1]	0x87 [1]	NO_CHOPPING	SDD0 NRTZ mode[4]
B10[5]	0x4002022C [5]	0x8B [5]	COMP_VREF_SW	SCB0 SDD analog switch (to CMP1)
B11[1:0]	0x40020230 [1:0]	0x8C [1:0]	DAC_MUXSEL[1:0]	SDDM0 analog MUX (to CMP1)
SCB_1 (0x40020234/0x8D through 0x40020260/0x98)				
B10[5]	0x4002025C [5]	0x97 [5]	COMP_VREF_SW	SCB1 SDD analog switch (to CMP3)
B11[1:0]	0x40020260 [1:0]	0x98 [1:0]	DAC_MUXSEL[1:0]	SDDM1 analog MUX (to CMP3)
SCB_2 (0x40020264/0x99 through 0x40020290/0xA4) – Including SDD1 Bits				
B6[0]	0x4002027C [0]	0x9F [0]	CUR_VOLB	SDD1 current mode select
B6[1]	0x4002027C [1]	0x9F [1]	NO_CHOPPING	SDD1 NRTZ mode[4]
B10[5]	0x4002028C [5]	0xA3 [5]	COMP_VREF_SW	SCB2 SDD analog switch (to CMP5)
B11[1:0]	0x40020290 [1:0]	0xA4 [1:0]	DAC_MUXSEL[1:0]	SDDM2 analog MUX (to CMP5)
SCB_3 (0x40020294/0xA5 through 0x400202C0/0xB0)				
B10[5]	0x400202BC [5]	0xAF [5]	COMP_VREF_SW	SCB3 SDD analog switch (to CMP7)
B11[1:0]	0x400202C0 [1:0]	0xB0 [1:0]	DAC_MUXSEL[1:0]	SDDM3 analog MUX (to CMP7)

Notes:

1. *Only the least-significant byte of each 32-bit data word is utilized*
2. *Components or functions indicated are only available in the A2F500.*
3. *ABM = AHB bus matrix; SCB = signal conditioning block; ACE = analog compute engine; SSE = sample sequencing engine; PPE = post processing engine.*
4. *Return-to-Zero (RTZ) mode is deprecated. The NO_CHOPPING bit should always be set to 1 for NRTZ mode.*

***Table 4-3* • Relative/Absolute Addresses for Sigma-Delta DAC Related Configuration Control Bits (continued)**

SCB Byte and Bit Address (relative to each SCB) or ACE Register Name	ABM Address (absolute 32-bit address) [1]	ACE/SSE/PPE Address (absolute 8-bit address)	Bit Name	Description
SCB_4[2] (0x400202C4/0xB1 through 0x400202F0/0xBC) – Including SDD2 Bits				
B6[0] [2]	0x400202DC [0]	0xB7 [0]	CUR_VOLB	SDD2 current mode select
B6[1] [2]	0x400202DC [1]	0xB7 [1]	NO_CHOPPING	SDD2 NRTZ mode[4]
B10[5] [2]	0x400202EC [5]	0xBB [5]	COMP_VREF_SW	SCB4 SDD analog switch (to CMP9)
B11[1:0] [2]	0x400202F0 [1:0]	0xBC [1:0]	DAC_MUXSEL[1:0]	SDDM4 analog MUX (to CMP9)
SDD Two-Byte Data Registers (0x40020500/0x28 through 0x40020508AC/0x142)				
SSE_DAC0_BYTES01 [15:0]	0x40020500[15:0]	0x140[15:0]	DAC_BYTES01[15:0]	SDD0 input data byte 1 and byte 0
SSE_DAC1_BYTES01 [15:0]	0x40020504[15:0]	0x141[15:0]	DAC_BYTES01[15:0]	SDD1 input data byte 1 and byte 0
SSE_DAC2_BYTES01 [15:0] [2]	0x40020508[15:0]	0x142[15:0]	DAC_BYTES01[15:0]	SDD2 input data byte 1 and byte 0

Notes:

1. *Only the least-significant byte of each 32-bit data word is utilized*
2. *Components or functions indicated are only available in the A2F500.*
3. *ABM = AHB bus matrix; SCB = signal conditioning block; ACE = analog compute engine; SSE = sample sequencing engine; PPE = post processing engine.*
4. *Return-to-Zero (RTZ) mode is deprecated. The NO_CHOPPING bit should always be set to 1 for NRTZ mode.*

***Table 4-4* • Functional Description of Sigma-Delta DAC Related Configuration Control Bits**

Bit Name	Description	R/W	Reset Value	Function
DAC_SYNC_UPDATE	SDD synchronized update	R/W	0b0	Writing a 1 will cause up to 24 bits input to the corresponding sigma-delta modulator to be updated, depending upon the resolution selected, if the corresponding DAC_SYNC_EN bit is set. The source of the input data is controlled by the corresponding REG_SEL bit. Since the update bits for all three SDDs are in the same register, they can be written simultaneously. These bits self-clear after one PCLK cycle.
DAC_SYNC_EN	SDD sync. update enable	R/W	0b0	1 = Enables synchronous updates to the corresponding sigma-delta modulator using the DAC_SYNC_UPDATE bit. 0 = Updates occur with writes to the DACn_BYTE0 register or writes to the SSE_DACn_BYTES01 register (whichever is selected by the REG_SEL bit in the corresponding DACn_CTRL register).

*Note: *Return-to-Zero (RTZ) mode is deprecated. The NO_CHOPPING bit should always be set to 1 for NRTZ mode.*

***Table 4-4* • Functional Description of Sigma-Delta DAC Related Configuration Control Bits (continued)**

Bit Name	Description	R/W	Reset Value	Function
DAC_SEL[1:0]	DAC input bit and clock source selection	R/W	0b00	00 = Select the ACE 1-bit sigma-delta modulator output (for data) and HCLK (for clock). 01 = Select signal F2ASDDn (Data n) and F2ASDDCLKn (clock n) from the FPGA fabric to drive the one-bit analog DAC. 10 = Select the SW_OBD bit (register data bit DACn_CTRL[4]) and HCLK (clock). 11 = Reserved
DAC_RES[1:0]	SDD modulator input resolution	R/W	0b00	00 = 8 bits 01 = 16 bits 10 = 24 bits 11 = Reserved
SW_OBD	DAC input bit	R/W	0b0	Register data bit used to drive the analog one-bit DAC if DAC_SEL = 10 (i.e., the software "bit-bang" mode is selected)
EN	SDD enable	R/W	0b0	Enables the sigma-delta DAC, including the digital modulator logic and the analog DAC and filter sections.
REG_SEL	SDD modulator input selection	R/W	0b0	Sigma-delta modulator input: selects the data source for the lower two bytes. 0 = Select DAC_BYTE1[7:0] concatenated with DAC_BYTE0[7:0] 1 = Select DAC_BYTES01[15:0] (upper byte is always DAC_BYTE2[7:0])
DAC_BYTE0[7:0]	SDD input data byte 0	R/W	0x00	LSB (bits [7:0]) of data input to the sigma-delta modulator, if REG_SEL is 0. Writing this register will cause up to 24 bits of the modulator to update if REG_SEL is 0 and DAC_SYNC_EN is 0, depending upon the resolution selected.
DAC_BYTE1[7:0]	SDD input data byte 1	R/W	0x00	Bits [15:8] of data input to the sigma-delta modulator, if REG_SEL is 0 and 16-bit or 24-bit resolution is selected
DAC_BYTE2[7:0]	SDD input data byte 2	R/W	0x00	Bits [23:16] of data input to the sigma-delta modulator if 24-bit resolution is selected
CUR_VOLB	SDD current mode select	R/W	0b0	Selects current (vs. voltage) output mode for the SDD, if set.
NO_CHOPPING	SDD NRTZ mode	R/W	0b0	Selects non-return-to-zero (NRTZ, vs. RTZ, also known as chopping) mode for the one-bit DAC, if set.*
COMP_VREF_SW	SDD analog switch	R/W	0b0	Analog switch connecting the SDD to comparator reference (inverting) input pin: 0 = SDD Vref switch off (high impedance) 1 = SDD Vref switch on (conducting)

*Note: *Return-to-Zero (RTZ) mode is deprecated. The NO_CHOPPING bit should always be set to 1 for NRTZ mode.*

***Table 4-4* • Functional Description of Sigma-Delta DAC Related Configuration Control Bits (continued)**

Bit Name	Description	R/W	Reset Value	Function
DAC_MUXSEL[1:0]	SDD analog multiplexer	R/W	0b00	Analog multiplexer in SCB selecting which SDD drives the comparator reference input pin: 00 = Select SDD0_OUT 01 = Select SDD1_OUT 10 = Select SDD2_OUT * 11 = Reserved
DAC_BYTE01[15:0]	SDD input data byte 0 and byte 1	R/W	0x0000	Bits [15:0] of the data input to the sigma-delta modulator, if REG_SEL is 1. Writing this register will cause up to 24 bits of the modulator to update if REG_SEL is 1 and DAC_SYNC_EN is 0, depending upon the resolution selected.

*Note: *Return-to-Zero (RTZ) mode is deprecated. The NO_CHOPPING bit should always be set to 1 for NRTZ mode.*

Related Information

SmartFusion Customizable System-on-Chip (cSoC) datasheet
http://www.microsemi.com/soc/documents/SmartFusion_DS.pdf

5 – Active Bipolar Prescaler (ABPS)

ABPS Features

- Selectable bipolar input ranges
- High-voltage input ranges
- High input impedance
- Factory trimmed and calibrated
- Low offset voltage
- Good gain accuracy
- Wide bandwidth/fast settling time
- Low noise
- Low operating power
- Even lower power sleep mode
- Output drives ADC input multiplexer

ABPS General Description

SmartFusion devices contain a number of active bipolar voltage-to-voltage prescalers (ABPS). There are two identical prescalers in each SmartFusion signal conditioning block (SCB). A2F060 devices have one SCB, for a total of two prescalers in each device. A2F200 devices have four SCBs, for a total of eight prescalers in each device. The A2F500 has five SCBs, for a total of ten prescalers in each device. The output of each prescaler goes to the analog multiplexer of one of the analog-to-digital converters (ADCs), which can select it for sampling and conversion.

The prescalers are constructed from continuous time op-amps in an inverting configuration. This allows the prescalers to have a large input voltage range and relatively high input impedance. Each prescaler has an individual dynamically configurable enable that can be used to save power in case a particular prescaler is currently not being used. The gain of each prescaler is dynamically configurable, resulting in the following four nominal full-scale input ranges: ±2.56 V, ±5.12 V, ±10.24 V, and ±15.36 V. Note that the maximum operating input voltage is –11.5 V to +14 V for reliability considerations, so the full implied input range of the ±15.36 V setting cannot be used. Also note that for the range selected the entire peak to peak voltage is seen by the ADC. If your signal only occupies 1/2 of the peak to peak input signal range, you end up using only 1/2 of the available ADC dynamic range.

The prescalers are coarsely trimmed for gain and offset during factory test, so their output range is just slightly less than the 0 V to Varef (nominal 2.56 V) input range of the ADC, to guarantee the input range limits are within range of the ADC. By default, the post processing engine (PPE) in the analog compute engine (ACE) is used to digitally compensate most remaining gain and offset errors using values determined during Microsemi's factory test and stored in eNVM (flash memory) in each device. In this step the gain is digitally re-inverted to account for the analog inversion in the analog op-amp stage. Even if you choose not to use the stored factory calibration values, the default operation is for the PPE to apply a gain of –1 to re-invert the signal, though the default PPE microcode can always be overwritten.

ABPS Symbol

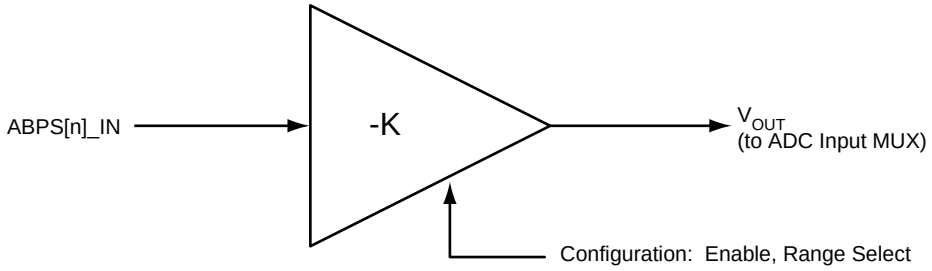


***Figure 5-1* • Active Bipolar Prescaler (ABPS) Symbol**

ABPS Detailed Description

The active bipolar prescaler (ABPS) senses the voltage on an input pad and scales it to fit the range of the ADC (nominally 0 V to 2.56 V, if the internal voltage reference, Varef, is used). It is designed using a conventional operational amplifier with input and feedback resistances arranged in an inverting configuration. A configurable current source provides a DC offset, so the input range of the prescaler is shifted to be centered about ground (Figure 5-1). The feedback resistance and the current source are dynamically configurable to provide one of four nominal gain and offset ranges: ±2.56 V, ±5.12 V, ±10.24 V, or ±15.36 V. The feedback resistance and the level-shift current are adjusted simultaneously to implement the four ranges with the correct offset current so that the input voltage range is always centered about zero volts.

During room-temperature factory test, Microsemi selects values for the gain trim, implemented as a small discrete change in the nominal 1M ohm input resistance; and the offset trim, implemented as a small discrete change in the nominal zero volts applied to the non-inverting pin of the op-amp.

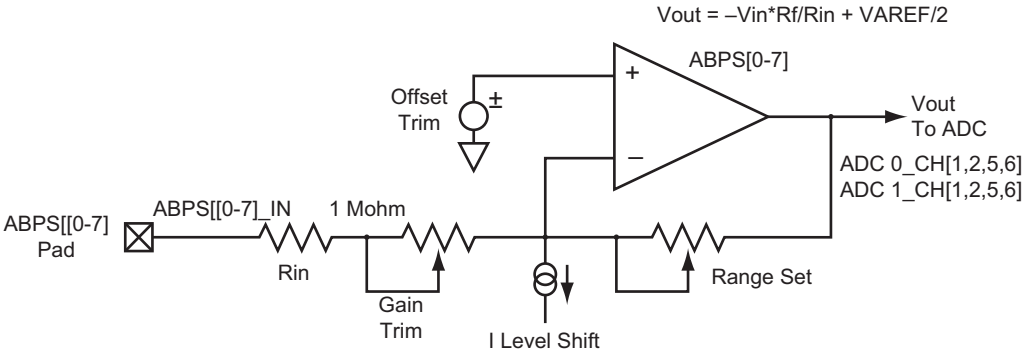


***Table 5-1* • Functional Block Diagram of Active Bipolar Prescaler**

Table 5-2 relates each of the eight or ten ABPS blocks with the Signal Conditioning Block (SCB) in which it is located:

***Table 5-2* • Active Bipolar Prescaler Input Pad Designations**

ADC	Signal Conditioning Block	ABPS Name	Pad Name	Function Name	Shared Functions on Pad	ADC Channel Used for ABPS Output
ADC0	SCB0	ABPS0	ABPS0	ABPS0_IN	None	ADC0_CH1
		ABPS1	ABPS1	ABPS1_IN		ADC0_CH2
	SCB1	ABPS2	ABPS2	ABPS2_IN		ADC0_CH5
		ABPS3	ABPS3	ABPS3_IN		ADC0_CH6
ADC1	SCB2	ABPS4	ABPS4	ABPS4_IN		ADC1_CH1
		ABPS5	ABPS5	ABPS5_IN		ADC1_CH2
	SCB3	ABPS6	ABPS6	ABPS6_IN		ADC1_CH5
		ABPS7	ABPS7	ABPS7_IN		ADC1_CH6
ADC2*	SCB4*	ABPS8*	ABPS8	ABPS8_IN		ADC2_CH1
		ABPS9*	ABPS9	ABPS9_IN		ADC2_CH2

Note: Components indicated are available only in the A2F500.

ABPS Configuration

The active bipolar prescalers (ABPS) can be dynamically configured via registers in the analog compute engine (ACE). These registers can be controlled by the ACE itself, or any of the following three AHB bus masters: a Cortex-M3 program, by user logic (or soft processor) via the AHB bus master connected to the FPGA fabric, or by the peripheral DMA bus master.

Each ABPS can individually be placed in a power save mode (disabled), and each ABPS can be configured for the desired gain/range selection.

The ABPS configuration options can be set statically using the SmartDesign MSS configurator. They can also be changed at any time dynamically using any of the three ABM bus masters listed above, or by the ACE's sample sequencer engine (SSE) or post-processing engine (PPE).

Table 5-3 summarizes the configuration settings related to the active bipolar prescalers (ABPS). Only the ABPS-related bits are described here; bits in the same or adjacent registers that are used for other SCB configuration functions are not documented in this section.

***Table 5-3* • Relative and Absolute Addresses for ABPS-Related Configuration Control Bits**

SCB Byte and Bit Address (relative to each SCB)	Advanced Bus Matrix (ABM) Address (absolute 32-bit address)[1]	Analog Compute Engine SSE and PPE Address (absolute 8-bit address)	Bit Name	Description
SCB_0 (0x40020204/0x81 – 0x40020230/0x8C)				
B8 [0]	0x40020224 [0]	0x89 [0]	ABPS_EN	ABPS0 enable
B8 [2:1]	0x40020224 [2:1]	0x89 [2:1]	GDEC [1:0]	ABPS0 range select
B8 [4]	0x40020224 [4]	0x89 [4]	ABPS_EN	ABPS1 enable

Notes:

1. *Only the least-significant byte of each 32-bit data word is utilized.*
2. *Components indicated by an asterisk (*) are available only in the A2F500.*

***Table 5-3* • Relative and Absolute Addresses for ABPS-Related Configuration Control Bits (continued)**

B8 [6:5]	0x40020224 [6:5]	0x89 [6:5]	GDEC [1:0]	ABPS1 range select
SCB_1 (0x40020234/0x8D – 0x40020260/0x98)				
B8 [0]	0x40020254 [0]	0x95 [0]	ABPS_EN	ABPS2 enable
B8 [2:1]	0x40020254 [2:1]	0x95 [2:1]	GDEC [1:0]	ABPS2 range select
B8 [4]	0x40020254 [4]	0x95 [4]	ABPS_EN	ABPS3 enable
B8 [6:5]	0x40020254 [6:5]	0x95 [6:5]	GDEC [1:0]	ABPS3 range select
SCB_2 (0x40020264/0x99 – 0x40020290/0xA4)				
B8 [0]	0x40020284 [0]	0xA1 [0]	ABPS_EN	ABPS4 enable
B8 [2:1]	0x40020284 [2:1]	0xA1 [2:1]	GDEC [1:0]	ABPS4 range select
B8 [4]	0x40020284 [4]	0xA1 [4]	ABPS_EN	ABPS5 enable
B8 [6:5]	0x40020284 [6:5]	0xA1 [6:5]	GDEC [1:0]	ABPS5 range select
SCB_3 (0x40020294/0xA5 – 0x400202C0/0xB0)				
B8 [0]	0x400202B4 [0]	0xAD [0]	ABPS_EN	ABPS6 enable
B8 [2:1]	0x400202B4 [2:1]	0xAD [2:1]	GDEC [1:0]	ABPS6 range select
B8 [4]	0x400202B4 [4]	0xAD [4]	ABPS_EN	ABPS7 enable
B8 [6:5]	0x400202B4 [6:5]	0xAD [6:5]	GDEC [1:0]	ABPS7 range select
SCB_4* (0x400202C4/0xB1 – 0x400202F0/0xBC)				
B8 [0]C*	0x400202E4 [0]*	0xB9 [0]*	ABPS_EN	ABPS8 enable*
B8 [2:1]*	0x400202E4 [2:1]*	0xB9 [2:1]*	GDEC [1:0]	ABPS8 range select*
B8 [4]*	0x400202E4 [4]*	0xB9 [4]*	ABPS_EN	ABPS9 enable*
B8 [6:5]*	0x400202E4 [6:5]*	0xB9 [6:5]*	GDEC [1:0]	ABPS9 range select*

Notes:

1. *Only the least-significant byte of each 32-bit data word is utilized.*
2. *Components indicated by an asterisk (*) are available only in the A2F500.*

Table 5-4 describes comparator-related configuration control bits.

***Table 5-4* • Functional Description of Comparator-Related Configuration Control Bits**

Bit Name	Description	R/W	Reset Value	Function
ABPS_EN	ABPS enable	R/W	0b0	0 = Power save mode; 1 = Operational mode
GDEC [1:0]	ABPS range select	R/W	0b00	00 = ± 15.36 V range (nominal gain = –0.08333 V/V) 01 = ± 10.24 V range (nominal gain = –0.125 V/V) 10 = ± 5.12 V range (nominal gain = –0.25 V/V) 11 = ± 2.56 V range (nominal gain = –0.5 V/V)

Related Information

SmartFusion Customizable System-on-Chip (cSoC) datasheet
http://www.microsemi.com/soc/documents/SmartFusion_DS.pdf

6 – Current Monitor

Current Monitor Features

- High differential gain
- Tolerance to high common mode voltages
- Common mode voltage rejection
- Factory calibrated
- High input impedance
- Low offset voltage
- Good gain accuracy and linearity
- Low noise
- Low operating power
- Even lower power sleep mode
- Supports simultaneous sampling
- Output drives ADC

Current Monitor General Description

SmartFusion devices contain a number of current monitors. There is one current monitor in each SmartFusion signal conditioning block (SCB). A2F060 devices have one SCB, for a total of one current monitor in each device. A2F200 devices have four SCBs, for a total of four current monitors in each device. A2F500 devices have five SCBs, for a total of five current monitors in each device. The output of each current monitor goes to the analog multiplexer of one of the analog-to-digital converters (ADCs), which can select it for conversion.

The current monitor amplifies a small differential voltage across its two input pins, while rejecting the common mode voltage. The primary use intended for the current monitor is to measure the voltage drop across a small external sense resistor placed in the path of a current. A small differential voltage proportional to the current is developed across the resistor in accordance with Ohm's Law (V = I × R). For example, the current monitor and sense resistor may be used to measure the current a power supply is providing to its load, or the current through a motor winding. It can also be used as a general purpose differential pre-amplifier for inputs to the ADC.

Current Monitor Symbol

Figure 6-1 shows the current monitor symbol.

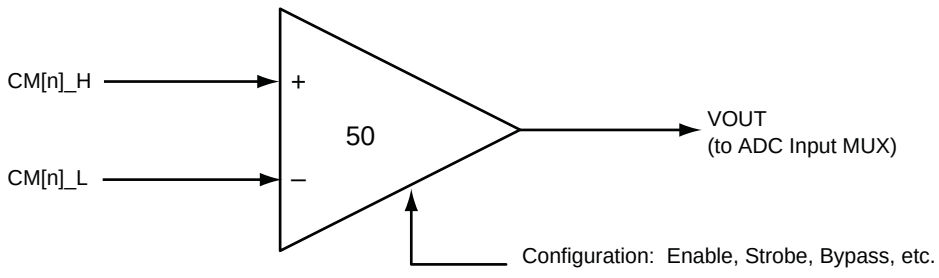


***Figure 6-1* • Current Monitor (CM) Symbol**

Current Monitor Detailed Description

The current monitors are constructed from switched-capacitor circuits. This allows the current monitors to have a high common mode input voltage range, a large common mode voltage rejection ratio, and a relatively high input impedance.

The current monitor High input signal (CM[n]_H) is attached to the CM[n] pad, and the current monitor Low input signal (CM[n]_L) is attached to the TM[n] pad (Figure 6-2). The current monitor senses only positive differential voltages. That is, the CM[n]_H signal voltage must be greater than the CM[n]_L signal voltage. Also, the common mode voltage is required to be always positive.

Because the current monitor is designed to sense only the differential voltage on its two input pins, and effectively reject the common mode voltage, the sense resistor can be placed on either side of the load—the ground side of the load, where the common mode voltage is low, or the high side, where the common mode voltage is high—with good results in measuring the differential voltage in either case. Placing the sense resistor on the grounded side of the load has the undesirable effect of raising the ground voltage the load sees; and also of increasing the ground resistance, which might affect the noise or stability of the load circuits. So it is often preferable to place the sense resistor on the high side of the load instead. The current monitor has been designed to tolerate a relatively large common mode voltage of 14.4 V. This means that the sense resistor can be placed on the high side of the load when monitoring the current of a nominal 12 V power supply, allowing for a 20% power supply tolerance.

In either placement, on the high side or low side, the voltage seen by the load will be reduced slightly due to the voltage drop across the sense resistor. The current monitor is designed to work with the small full-scale differential voltage of just 51.2 mV, to minimize this effect. In most cases this small voltage drop will have a negligible effect on the overall system performance, especially if there is a closed loop system to compensate for it; for example, a power supply having a voltage sense feedback control loop.

The current monitor has a nominal differential gain of 50 V/V, so a zero-to-51.2 mV input is scaled up to drive the SmartFusion analog-to-digital converter (ADC) to which it is connected with the optimal full-scale signal of zero-to-2.56 V. The high gain of the current monitor allows smaller sense resistors to be used. Besides resulting in a lower voltage drop, the power dissipated in the sense resistor is much less than if a higher resistance value was required, according to the square-law relationship shown in EQ 6-1:

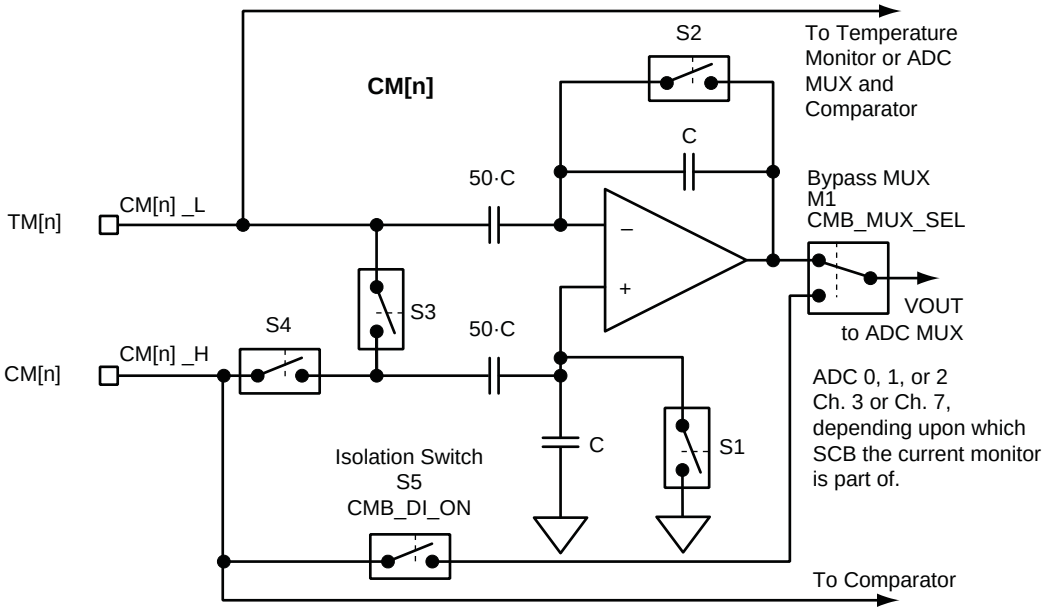
$$\text{Power (Watts)} = \text{Current (Amps)}^2 \times \text{Resistance (Ohms)}$$

EQ 6-1

Each current monitor has an individual dynamically configurable enable that can be used to save power in case that particular current monitor is currently not being used.

By default, factory calibration values are used to digitally compensate gain and offset errors using the post processing engine (PPE) in the analog compute engine (ACE).

Figure 6-2 shows a simplified schematic of the current monitor.



Note: See also Table 6-4 on page 76.

***Figure 6-2* • Functional Block Diagram of Current Monitor**

If you are using Libero® System-on-Chip (SoC) SmartDesign flow SmartDesign MSS configurator, many of the details of the current monitor operation will be transparent to you. SmartDesign MSS configurator automatically generates sample sequencer engine instructions for the analog compute engine to control the current monitor and ADC strobes that are required to use the current monitor. These are suitable for most situations.

Table 6-1 on page 74 shows the sequencing of the switches in the current monitor. This sequencing is fully automated by the current monitor logic, and once started cannot be controlled except by aborting the cycle. The sequence is initiated (advanced from the idle state to the other states) by applying one of the current monitor strobes. Removing the strobe returns the current monitor to the idle state.

1. In the first phase, the current monitor is standing by, or idling. In this phase most of the offset of the circuit is sensed and removed, acting as an auto-zero circuit. The circuit is effectively armed and ready for a measurement.
2. In the second phase, the common mode input, as present on the low side input pad (TM[n]), is stored on the two large input capacitors.
3. The third is a short phase needed just to ensure proper sequencing of the switches so the pads are not shorted (break-before-make).

In the fourth and final phase, the input voltage on the large input capacitor on the op-amp's non-inverting input is switched from the low side pad (TM[n]), to which it was connected in phase 2, to the high side pad (CM[n]) voltage. This change in voltage, equal to the differential input voltage, forces a current to flow through the input capacitor, which in turn creates a 50-times-larger voltage change on the smaller series capacitor.

***Table 6-1* • Current Monitor Switch Sequencing**

Phase	Operation	S1	S2	S3	S4
1	Idle (auto-zero)	Closed	Closed	Closed	Open
2	Sample low side input	Open	Open	Closed	Open
3	Transition phase (break-before-make)	Open	Open	Open	Open
4	Amplify and Track the differential-mode input	Open	Open	Open	Closed
1	Idle (auto-zero)	Closed	Closed	Closed	Open

***Table 6-2* • Switch and MUX to Bypass Current Monitor when Going Directly to the ADC**

Reference	Controlled by Configuration Bit	Function
M1	CMB_MUX_SEL	Select current monitor, or bypass for direct ADC input.
S5	CMB_DI_ON	Isolate downstream circuitry from potentially high current monitor voltages, or allow direct ADC input.

Any changes to the differential voltage are tracked while in the fourth phase, with a nominal gain of 50 V/V. Common mode noise is nominally rejected during the tracking mode (phase 4).

The ADC should sample the op-amp output while the current monitor is still in the fourth phase. The current monitor will stay in the fourth phase until its strobe is removed, at which time it will return to the idle state (phase 1). Some time should be allowed for the current monitor op-amp to transition from zero volts to its final value and settle. Therefore, the ADC_START signal should not be asserted until 1 µs after the current monitor strobe has been raised. It should be noted that when the ADC starts its own sampling phase, it connects a capacitor to its input (which is the output of the current monitor, if selected), and this can also cause a transient response from the current monitor op-amp. However, this smaller transient will settle during the time the ADC allows for its input sampling phase. The current monitor strobe should not be taken low until after the ADC_SAMPLE output from the ADC indicates that the ADC has transitioned from sample mode to hold mode, because that would zero the current monitor output before the ADC goes to hold mode, resulting in the wrong voltage being captured by the ADC. Figure 6-3 shows the relative timing relationships of the current monitor strobe, the current monitor op-amp output voltage, and the ADC sampling status output.

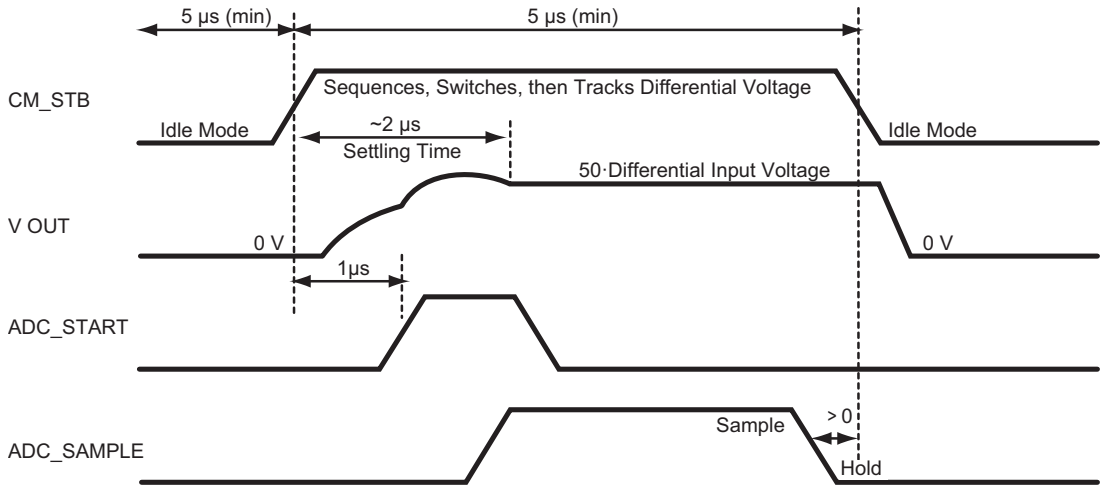


***Figure 6-3* • Relative Timing of Current Monitor Strobe and ADC Control Signals**

Table 6-3 lists the timing requirements for current monitor strobe operations.

Table 6-3 • **Timing Requirements for Current Monitor Strobe Operation**

Parameter	Minimum	Maximum
CM_STB(LOW)	5 µs	
CM_STB(HIGH)	5 µs	200 µs
Time between Strobe start and ADC_START	1 µs	
Time after ADC_SAMPLE(LOW) before CM_STB(LOW)	0 µS	

To facilitate simultaneous sampling of multiple current monitors, all the current monitor strobes are duplicated in one ACE register. These are referred to as the global current monitor strobes. Being collected into one register allows several current monitor strobes to be set at the same instant. Their function is exactly the same as the regular current monitor strobes. The global and regular strobes for each current monitor are logically ORed together. Therefore, it is important that both are returned to the low state after finished with their use. Starting with both the regular and global strobe low (the current monitor is in the idle state), then setting either the global or regular strobe high will start the automated current monitor switch sequence, ending with the current monitor in track mode (phase 4) until such time as the strobe is driven low once again, forcing the current monitor back to idle mode (phase 1).

It is common (though not required) to use some sort of digital filtering or averaging with the current monitor in order to improve the accuracy and resolution of the result by reducing the effect of random noise. The current monitor noise is relatively white, so the noise is reduced by roughly a factor of $\sqrt{N}$ for an N-point equally-weighted average. Note that the ACE post processing engine has a built-in first-order infinite impulse response (IIR) filter that can be used. It weights inputs exponentially, with the most recent samples receiving the highest weight. The noise is reduced by roughly the square root of the filter passband bandwidth.

The current monitor input pads are shared with other functions. Both current monitor input pads can be used as direct ADC inputs. When used this way, the signal on the CM[n] input pin (the positive input to the current monitor) displaces the output of the current monitor into the ADC multiplexer. In other words, the CM[n] pin can be the input to the ADC, or the output of the current monitor op-amp can be the input to the ADC, but not both at the same time since they share the same ADC multiplexer channel. Note that direct ADC inputs have a much lower voltage range and voltage tolerance than the current monitors when the bypass option is selected, so care must be taken to restrict the input voltage accordingly. When using the current monitor, there is an isolation switch which must be open to protect the downstream circuitry from the potentially high voltages used with it. For information about direct ADC inputs, see the "Direct ADC Inputs" section on page 103.

The signal on the TM[n] input pad can also be used as a direct ADC input. Similarly to the CM[n] bypass, if used in this way the TM[n] direct input signal displaces the temperature monitor output at the input of the ADC multiplexer.

The TM[n] input pad can also be used as the input/output pad for the temperature monitor. When used with the temperature monitor, the current monitor cannot be used, though the CM[n] pin could still be used as a direct ADC input. The temperature monitor is designed to withstand the same input voltages as the current monitor, but it operates over a much smaller voltage range when used for temperature sensing. This pad also has a high-voltage isolation switch which is open when using the current monitor but closed when using the TM[n] pad as a direct input to the ADC. For more information on the temperature monitor, see the "Temperature Monitor" section on page 81.

Finally, there is also a comparator that shares the two input pads with the current monitor. The comparator positive (non-inverting) input is on the CM[n] pad, and the negative (inverting) input is on the TM[n] pad. The comparator is designed to withstand the same input voltages as the current monitor (up to 14.4 V) when it is disabled, but a slightly smaller voltage range (up to 9 V) when enabled, and it is only functional over an even narrower operating voltage range (up to 2.56 V). For more about the comparators, see the "High-Speed Comparators" section on page 91.

To understand how to set the ADC input multiplexer to a selected channel, or how to configure and operate the ADC itself, see the "Analog-to-Digital Converter (ADC)" section on page 27.

Table 6-4 relates each of the current monitor blocks with the signal conditioning block (SCB) in which it is located, along with the shared functions for the current monitor input pads:

***Table 6-4* • Current Monitor Input Pad Designations**

ADC	Signal Conditioning Block	Current Monitor Name	Pad Name	Function Name	Shared Functions on Pad		
ADC0	SCB0	CM0	CM0	CM0_H	AD0_CH3	COMP0_P	
			TM0	CM0_L	ADC0_CH4	CMP0_N	TM0_IO
	SCB1	CM1	CM1	CM1_H	ADC0_CH7	CMP2_P	
			TM1	CM1_L	ADC0_CH8	CMP2_N	TM1_IO
ADC1	SCB2	CM2	CM2	CM2_H	ADC1_CH3	CMP4_P	
			TM2	CM2_L	ADC1_CH4	CMP4_N	TM2_IO
	SCB3	CM3	CM3	CM3_H	ADC1_CH7	CMP6_P	
			TM3	CM3_L	ADC1_CH8	CMP6_N	TM3_IO
ADC2*	SCB4*	CM4*	CM4*	CM4_H*	ADC4_CH3*	CMP8_P*	
			TM4*	CM4_L*	ADC4_CH4*	CMP8_N*	TM4_IO*

*Note: * Components indicated are available only in the A2F500.*

Current Monitor Configuration

The current monitors can be dynamically configured via registers in the analog compute engine (ACE). These registers can be controlled by any of the following:

1. The ACE itself,
2. An ARM Cortex-M3 program
3. User logic (or soft processor) via the AHB bus master connected to the FPGA fabric
4. The peripheral DMA bus master

Each current monitor can individually be placed in a power save mode (disabled).

The strobes used to individually or collectively start and stop the current monitor automatic switch sequence are also located in ACE registers, and can be accessed in all the ways listed above.

The CM[n] and TM[n] pads used by the current monitor can be used as direct inputs to the ADC. To facilitate this, there is a multiplexer in the current monitor block which selects between the current monitor function and the direct input function for the CM[n] pad. There is also, in each case, an isolation switch which must be turned on in order to pass the direct input signal to the ADC. When the current monitor is used, these two switches must be off (non-conducting) in order to protect the downstream circuitry from the potentially high current monitor input voltages. Though the multiplexer and isolation switches have separate configuration control bits (for factory test purposes), they are generally set or cleared together in normal operation.

The current monitor configuration options (the enable) can be set statically using the SmartDesign MSS configurator in Libero SoC. They can also be changed at any time dynamically using any of the three ABM masters listed above, or by the ACE's sample sequencer engine (SSE) or post-processor engine (PPE). Besides the static configuration options, the SmartDesign tool also automatically creates the SSE microcode required to control the current monitor strobes and ADC control registers.

Table 6-5 on page 77 summarizes the configuration settings related to the current monitors and current monitor input pad shared functions. Only the current-monitor-related bits are described here; bits in the same or adjacent registers that are used for other SCB configuration functions are not documented in

this section. For information on how to select a particular ADC input channel or how to operate the ADC, see the "Analog-to-Digital Converter (ADC)" section on page 27.

***Table 6-5* • Addresses for Current-Monitor-Related Configuration Control Bits**

SCB Byte and Bit Address (relative to each SCB)	Advanced Bus Matrix (ABM) Address (absolute 32-bit address)1	ACE SSE and PPE Address (absolute 8-bit address)	Bit Name	Description
SCB_0 (0x40020204/0x81 – 0x40020230/0x8C)				
B7 [0]	0x40020220 [0]	0x88 [0]	CM_STB_G[0]	Global CM0 strobe
B7 [1]	0x40020220 [1]	0x88 [1]	CM_STB_G[1]	Global CM1 strobe
B7 [2]	0x40020220 [2]	0x88 [2]	CM_STB_G[2]	Global CM2 strobe
B7 [3]	0x40020220 [3]	0x88 [3]	CM_STB_G[3]	Global CM3 strobe
B7 [4]*	0x40020220 [4] *	0x88 [4]*	CM_STB_G*	Global CM4 strobe*
B9 [0]	0x40020228 [0]	0x8A [0]	CMB_MUX_SEL	CM0 bypass MUX
B9 [1]	0x40020228 [1]	0x8A [1]	CMB_DI_ON	CM0 Isolation switch
B9 [2]	0x40020228 [2]	0x8A [2]	CM_EN	CM0 enable
B9 [3]	0x40020228 [3]	0x8A [3]	CM_STB	CM0 strobe
B9 [4]	0x40020228 [4]	0x8A [4]	COMP_EN	CMP0 comparator enable
B10 [0]	0x4002022C [0]	0x8B [0]	TMB_MUX_SEL	TM0 bypass MUX
B10 [1]	0x4002022C [1]	0x8B [1]	TMB_DI_ON	TM0 isolation switch
SCB_1 (0x40020234/0x8D – 0x40020260/0x98)				
B9 [0]	0x40020258 [0]	0x96 [0]	CMB_MUX_SEL	CM1 bypass MUX
B9 [1]	0x40020258 [1]	0x96 [1]	CMB_DI_ON	CM1 isolation switch
B9 [2]	0x40020258 [2]	0x96 [2]	CM_EN	CM1 enable
B9 [3]	0x40020258 [3]	0x96 [3]	CM_STB	CM1 strobe
B9 [4]	0x40020258 [4]	0x96 [4]	COMP_EN	CMP2 comparator enable
B10 [0]	0x4002025C [0]	0x97 [0]	TMB_MUX_SEL	TM1 bypass MUX
B10 [1]	0x4002025C [1]	0x97 [1]	TMB_DI_ON	TM1 isolation switch
SCB_2 (0x40020264/0x99 – 0x40020290/0xA4)				
B9	0x40020288 [0]	0xA2 [0]	[0]CMB_MUX_SEL	CM2 bypass MUX
B9 [1]	0x40020288 [1]	0xA2 [1]	CMB_DI_ON	CM2 isolation switch
B9 [2]	0x40020288 [2]	0xA2 [2]	CM_EN	CM2 enable
B9 [3]	0x40020288 [3]	0xA2 [3]	CM_STB	CM2 strobe
B9 [4]	0x40020288 [4]	0xA2 [4]	COMP_EN	CMP4 comparator enable
B10 [0]	0x4002028C [0]	0xA3 [0]	TMB_MUX_SEL	TM2 bypass MUX
B10 [1]	0x4002028C [1]	0xA3 [1]	TMB_DI_ON	TM2 isolation switch

Notes:

1. *Only the least-significant byte of each 32-bit data word is utilized.*
2. *Components indicated with an asterisk (*) are available only in the A2F500.*

***Table 6-5* • Addresses for Current-Monitor-Related Configuration Control Bits (continued)**

SCB_3 (0x40020294/0xA5 – 0x400202C0/0xB0)				
B9 [0]	0x400202B8 [0]	0xAE [0]	CMB_MUX_SEL	CM3 bypass MUX
B9 [1]	0x400202B8 [1]	0xAE [1]	CMB_DI_ON	CM3 isolation switch
B9 [2]	0x400202B8 [2]	0xAE [2]	CM_EN	CM3 enable
B9 [3]	0x400202B8 [3]	0xAE [3]	CM_STB	CM3 strobe
B9 [4]	0x400202B8 [4]	0xAE [4]	COMP_EN	CMP6 comparator enable
B10 [0]	0x400202BC [0]	0xAF [0]	TMB_MUX_SEL	TM3 bypass MUX
B10 [1]	0x400202BC [1]	0xAF [1]	TMB_DI_ON	TM3 isolation switch
SCB_4* (0x400202C4/0xB1 – 0x400202F0/0xBC)				
B9 [0]*	0x400202E8 [0]*	0xBA [0]*	CMB_MUX_SEL*	CM4 bypass MUX*
B9 [1]*	0x400202E8 [1]*	0xBA [1]*	CMB_DI_ON*	CM4 isolation switch*
B9 [2]*	0x400202E8 [2] *	0xBA [2]*	CM_EN*	CM4 enable*
B9 [3]*	0x400202E8 [3] *	0xBA [3]*	CM_STB*	CM4 strobe*
B9 [4]*	0x400202E8 [4*	0xBA [4]*	COMP_EN*	CMP8 comparator enable*
B10 [0]*	0x400202EC [0]*	0xBB [0]*	TMB_MUX_SEL*	TM4 bypass MUX*
B10 [1]*	0x400202EC [1]*	0xBB [1]*	TMB_DI_ON*	TM4 isolation switch*

Notes:

1. *Only the least-significant byte of each 32-bit data word is utilized.*
2. *Components indicated with an asterisk (*) are available only in the A2F500.*

Table 6-6 gives functional descriptions for comparator-related configuration control bits.

***Table 6-6* • Functional Description of Comparator-Related Configuration Control Bits**

Bit Name	Description	R/W	Reset Value	Function
CMB_MUX_SEL	CM bypass MUX	R/W	0b0	0 = ADC input MUX driven from CM op-amp 1 = ADC input MUX driven directly from CM[n] pad
CMB_DI_ON	CM isolation switch	R/W	0b0	0 = Isolation switch open (isolating high voltages) 1 = Isolation switch closed (direct ADC input)
CM_EN	CM enable	R/W	0b0	0 = Current monitor power save mode 1= Current monitor operational mode
CM_STB	CM strobe	R/W	0b0	0 = Idle mode (if appropriate CM_STB_G) is also 0) 1 = Initiate sequence and leave in track mode
CM_STB_G	Global CM strobe	R/W	0b0	0 = Idle mode (if appropriate CM_STB is also 0) 1 = Initiate sequence and leave in track mode
COMP_EN	CM enable	R/W	0b0	0 = Comparator power save mode 1 = Comparator operational mode
TMB_MUX_SEL	TM bypass MUX	R/W	0b0	0 = ADC input MUX driven from temperature monitor 1 = ADC input MUX driven directly from TM[n] pad
TMB_DI_ON	TM isolation switch	R/W	0b0	0 = Isolation switch open (isolating high voltages) 1 = Isolation switch closed (direct ADC input)

In general, when using the current monitor, the other shared functions of the pads are not used. If the current monitor is used with high input voltages, care must be taken to turn off the comparator and isolate the ADC inputs. Therefore, the most common configuration when using the current monitor function is as shown in Table 6-7.

Table 6-7 • **Common Configuration for Current Monitor**

Parameter	Function
MB_MUX_SEL = 0	Selects current monitor over direct input option.
CMB_DI_ON = 0	Isolates the ADC from potentially high input voltages on CM[n].
CM_EN = 1	Turns on the current monitor.
CM_STB = {0, 1}	Start in idle mode (0) and strobe (1) to start sequence.
CM_STB_G = 0	ORed with CM_STB. Not needed unless simultaneous operation of multiple current monitors is required.
COMP_EN = 0	Turns off comparator (so it will tolerate highest voltages).
TMB_MUX_SEL = 0	Does not really matter, so long as the isolation switch is used.
TMB_DI_ON = 0	Isolates the ADC from potentially high input voltages on TM[n].

Related Information

Datasheets

SmartFusion Customizable System-on-Chip datasheet

http://www.microsemi.com/soc/documents/SmartFusion_DS.pdf

7 – Temperature Monitor

Temperature Monitor Features

- Converts temperature to voltage
- Only one external component
- Low pin count
- Factory calibrated
- Good temperature accuracy
- Low noise
- Low operating power
- Even lower power sleep mode
- Output drives ADC

Temperature Monitor General Description

SmartFusion devices contain a number of temperature monitors. There is one temperature monitor in each SmartFusion signal conditioning block (SCB). A2F060 devices have one SCB, for a total of one temperature monitor in each device. A2F200 devices have four SCBs, for a total of four temperature monitors in each device. A2F500 devices have five SCBs, for a total of five temperature monitors in each device. The output of each temperature monitor goes to the analog multiplexer of one of the analog-to-digital converters (ADCs) input multiplexers, which can select it for conversion.

The temperature monitor converts temperature to voltage using an external PN-junction, such as a diode-connected bipolar transistor or a PN-junction diode, as a temperature sensor. The diode has its anode connected to the temperature monitor's input-output (I/O) pad. The cathode is connected to the temperature monitor ground pad. The temperature monitor ground pad is shared between the temperature monitors in two adjacent SCBs.

The temperature monitor generates two small output currents which it applies to its input-output pad sequentially when strobed. The difference in the two voltages developed across the PN-junction is amplified and then digitized by the ADC. This voltage is proportional to absolute temperature using the well-known temperature dependence of the current-versus-voltage characteristic of PN-junction diodes. The post processing engine (PPE) in the analog compute engine (ACE) applies the factory-determined offset compensation value.

Temperature Monitor Symbol

Figure 7-1 shows the temperature monitor symbol.

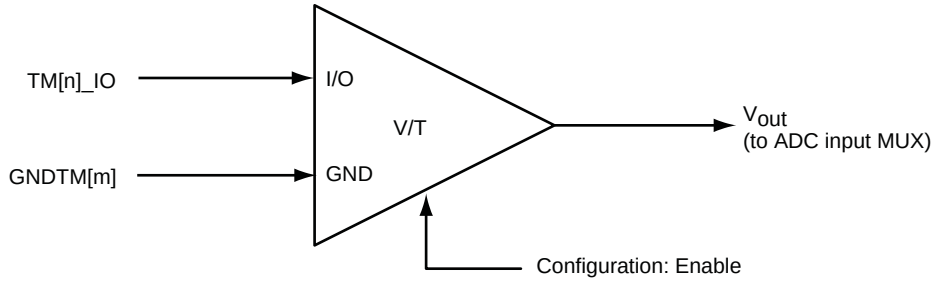


***Figure 7-1* • Temperature Monitor Symbol**

Each SCB has one temperature monitor. Temperature monitors in every pair of SCBs share a temperature monitor ground. The temperature monitor outputs the stimulus currents on its I/O pin, and inputs the resulting voltages (with respect to ground) on the same pin.

In the A2F200 there are four temperature monitors, labeled TM0 through TM3, with input/output pads also labeled TM0 through TM3. These pads are connected to signals/functions named TM0_IO through TM3_IO. There are two ground pads and associated signals named GNDTM0 and GNDTM1. The voltages on pads TM0 and TM1 are referenced to GNDTM0, and the voltages on pads TM2 and TM3 are referenced to GNDTM1.

In the A2F060 there is one temperature monitor, TM0, with input/output pad also labeled TM0. In the A2F500 there are five temperature monitors, labeled TM0 through TM4, with input/output pads also labeled TM0 through TM4. These pads are connected to signals/functions named TM0_IO through TM4_IO. There are three ground pads and associated signals named GNDTM0, GNDTM1, and GNDTM2. The voltages on pads TM0 and TM1 are referenced to GNDTM0, the voltages on pads TM2 and TM3 are referenced to GNDTM1, and the voltage on pad TM4 is referenced to GNDTM2.

Temperature Monitor Detailed Description

The temperature monitors are constructed from switched-capacitor circuits. This allows the temperature monitors to have a high sensitivity and a low offset.

The temperature monitor uses one input-output pad to output two sense currents, one after the other, and to measure the resulting voltage change. It is designed for use with a PN-junction connected from its I/O pad to the special low-noise ground pins. One good choice for the temperature-sensing PN-junction is a 2N3904 bipolar transistor (such as can be obtained from National Semiconductor or Infineon), with the collector and base connected together to configure it as a diode.

The temperature characteristic of the current-voltage relationship of a PN-junction allows a junction diode or diode-connected bipolar transistor to be used as a temperature sensor. The change in voltage across the diode or the base-to-emitter voltage of a bipolar transistor for two sense currents is approximated by EQ 7-1:

$$V_{BE1} - V_{BE2} = (KT_K/q) \times (\ln (I1/I2)) \qquad \textit{EQ 7-1}$$

where:

I1/I2 is the ratio of the two sense currents, and is designed to be 10:1 in the SmartFusion temperature monitors.

ln is the logarithm, base ***e*** (Euler's number, 2.71828…).

K is Boltzman's constant, approximately 1.3807×10^{-23} Joules/Kelvin.

q is the charge of an electron, approximately 1.6022×10^{-19} Coulombs.

T_K is the temperature in Kelvin. The Kelvin scale is zero at the temperature absolute zero, and 273.15° at zero degrees Celsius. It increases by one for every one degree Celsius increase in temperature.

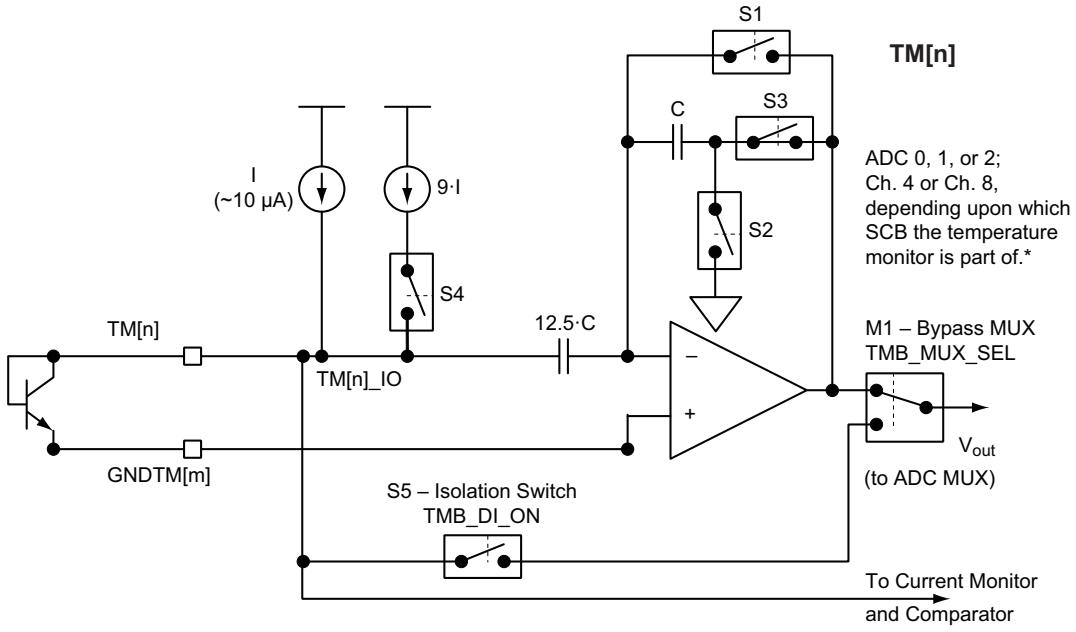
This equates to a delta V_{BE} of approximately –200 µV/degree when the current is dropped by a ten-to-one ratio. Since the gain of the circuit is nominally –12.5, the voltage seen by the ADC is about 2.5 mV per degree. In 12-bit mode, that is nominally 4 LSB-counts (or 1 count in 10-bit mode).

Table 7-1 gives the nominal scale factor for each mode.

***Table 7-1* • Nominal Scale Factor for ADC Resolution Selected**

ADC Resolution Selected	Nominal Scale Factor
8-bit mode	4.0 Kbit
10-bit mode	1.0 Kbit
12-bit mode	0.25 Kbit

Figure 7-2 shows a simplified schematic of the temperature monitor.



*Note: *See Table 7-4 on page 87.*

***Figure 7-2* • Functional Block Diagram of Temperature Monitor**

If you are using the SmartDesign MSS configurator, many of the details of the temperature monitor operation will be transparent to you. Smart Design automatically generates sample sequencer engine microcode to control the temperature monitor and ADC strobes that are required to use the temperature monitor that is suitable for most situations. However, it is possible to write customized microcode.

Table 7-2 shows the sequencing of the switches in the temperature monitor.

***Table 7-2* • Temperature Monitor Switch Sequencing**

Phase	Operation	S1	S2	S3	S4
1	Idle (auto-zero)	Closed	Closed	Open	Closed
2	Disconnect unity feedback	Open	Closed	Open	Closed
3	Disconnect auto-zero ground	Open	Open	Open	Closed

Table 7-2 • **Temperature Monitor Switch Sequencing**

4	Connect auto-zero capacitor in feedback	Open	Open	Closed	Closed
5	Reduce driven current 10:1, inverting and amplifying the change in input voltage	Open	Open	Closed	Open
1	Idle (auto-zero)	Closed	Closed	Open	Closed

This sequencing is fully automated by the temperature monitor logic, and once started cannot be controlled except by aborting the cycle. The sequence is initiated (advanced from the idle state to the other states) by applying the temperature monitor strobes, as described in the steps below. Removing the strobe returns the temperature monitor to the idle state.

1. In the first phase, the temperature monitor is standing by, or idling. In this phase the offset of the op-amp is sensed, acting as an auto-zero circuit. The circuit is effectively armed and ready for a measurement. The offset of the op-amp is stored on the small capacitor. The voltage at the input-output pad is stored on the large capacitor (less the op-amp offset).
2. In the second short transition phase, the unity feedback of the op-amp is removed by opening switch S1. The temperature monitor automatically sequences from phase 1 to 5 when the input strobe is applied.
3. The third phase is another short transition phase to disconnect the auto-zero capacitor (C) from ground so it is floating, by opening switch S2.
4. In the fourth phase the auto-zero capacitor is connected in the feedback of the op-amp. The op-amp output should go to zero volts. The effect of the op-amp offset is cancelled. The input voltage due to the current being driven through the external PN-junction diode is also blocked. The op-amp is now configured with an inverting gain of 12.5, due to the ratio of the feedback and input capacitors.
5. In the fifth and final phase, the current being driven through the external diode is dropped by a ratio of 10:1. This causes a corresponding temperature-related drop in the input voltage, which is inverted and amplified by the op-amp. The temperature monitor remains in this state until the strobe is removed.
6. After the op-amp output is digitized, the input strobe can be removed, putting the temperature monitor back in its idle state.

The ADC should sample the op-amp output while the temperature monitor is still in the fifth phase. The temperature monitor will stay in the fifth phase until its strobe is removed, at which time it will return to the idle state (phase 1). Some time should be allowed for the temperature monitor op-amp to transition from zero volts to its final value and settle. Therefore, the ADC_START signal should not be asserted until 5 µs after the temperature monitor strobe has been raised. Note that when the ADC starts its own sampling phase, it connects a capacitor to its input (which is the output of the temperature monitor, if selected), and this can also cause a transient response from the temperature monitor op-amp. However, this smaller transient will settle during the time the ADC allows for its input sampling phase. The temperature monitor strobe should not be taken low until after the ADC_SAMPLE output from the ADC indicates that the ADC has gone from sample mode to hold mode. Otherwise the temperature monitor output would go to zero before the ADC goes to hold mode, resulting in the wrong voltage being captured by the ADC. Figure 7-3 and Table 7-3 on page 86 show the relative timing relationships of the

temperature monitor strobe, the temperature monitor op-amp output voltage, and the ADC sampling status output.

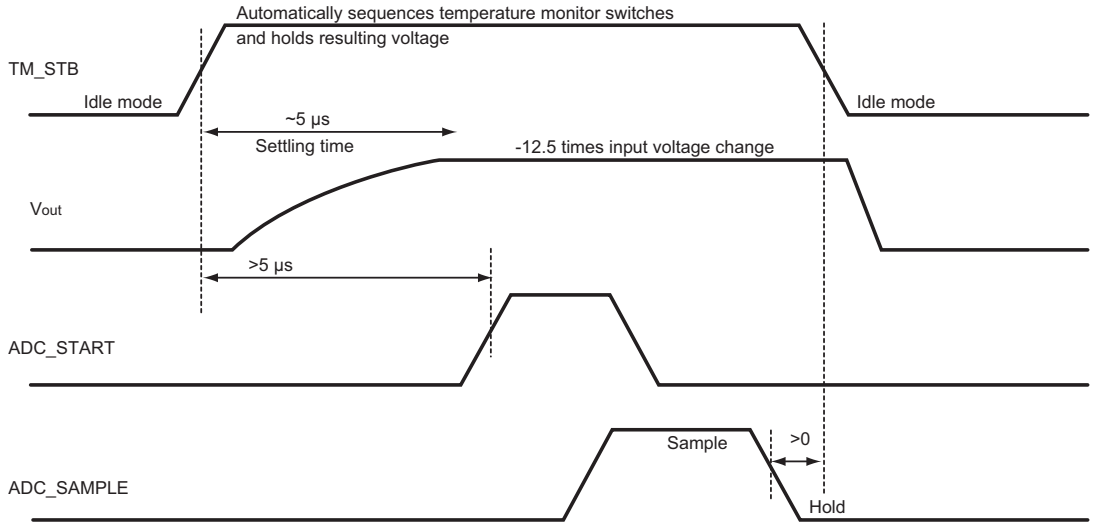


***Figure 7-3* • Relative Timing of Temperature Monitor Strobe and ADC Control Signals**

Table 7-3 • **Timing Requirements for Temperature Monitor Strobe Operation**

Parameter	Minimum	Maximum
TM_STB (High)	10 µS	105 µS
TM_STB (Low)	5 µs	
Time between strobe start and ADC_START	5 µS	
Time after ADC_SAMPLE (Low) before CM_STB (Low)	0 µs	

It is common to use some sort of digital filtering or averaging with the temperature monitor in order to improve the accuracy and resolution of the result by reducing the effect of random noise. The temperature monitor noise is relatively white, so the noise is reduced by roughly a factor of $\sqrt{N}$ for an N-point equally-weighted average. For example, if 64 samples are averaged, the random noise will be reduced by roughly a factor of 8. So, if the RMS noise for individual samples was 2 degrees, the noise for the results after averaging might be around 1/4 degree RMS.

The ACE post processing engine has a built-in first order infinite impulse response (IIR) filter that can be used. It weights inputs exponentially, with the most recent samples receiving the highest weight. The noise is reduced by roughly the square-root of the filter passband bandwidth.

Each temperature monitor has an individual dynamically configurable enable that can be used to save power in case that particular temperature monitor is currently not being used.

By default, factory calibration values are used to digitally compensate offset errors using the post processing engine (PPE) in the analog compute engine (ACE). The offset of each temperature monitor is calibrated near room temperature at the Microsemi factory using an external 2N3904 transistor from National Semiconductor or Infineon as the temperature sensor, and the value stored in nonvolatile memory (eNVM) in the device. Since the measurements are only made at one temperature, there is no temperature slope correction; however, the factory offset correction will nominally zero the combined gain and offset errors at room temperature (rather than at absolute zero).

The temperature monitor input pads are shared with other functions. The temperature monitor input-output pad can be used as a direct ADC input. When used this way, the signal on the TM[n] input-output pad displaces the output of the temperature monitor into the ADC multiplexer. In other words, the TM[n] pin can be the input to the ADC, or the output of the temperature monitor can be the input to the ADC, but not both at the same time since they share the same ADC multiplexer channel.

The TM[n] input pad can also be used as the low (inverting) pad for the current monitor. When used with the current monitor, the temperature monitor cannot be used. The temperature monitor is designed to withstand the same input voltages as the current monitor, but it operates over a much smaller voltage range when used for temperature sensing. For more information on the current monitor, see the "Current Monitor" section on page 71.

Note that direct ADC inputs have a much lower voltage range and voltage tolerance than the current monitor and temperature monitor, so care must be taken to restrict the input voltage to the TM[n] pad when the direct ADC bypass is selected and the isolation switch is conducting. For information about direct ADC inputs, see the "Direct ADC Inputs" section on page 103.

Finally, there is also a comparator that shares the one input pad with the temperature monitor. The comparator negative (inverting) input is on the TM[n] pad. The comparator is designed to withstand the same input voltages as the current monitor (up to 14.4 V) when it is disabled, but a slightly smaller voltage range (up to 9 V) when enabled. It is functional over an even narrower operating voltage range (up to 2.56 V). For more about the comparators, see the "High-Speed Comparators" section on page 91.

To understand how to set the ADC input multiplexer to a selected channel, or how to configure and operate the ADC itself, see the "Analog-to-Digital Converter (ADC)" section on page 27.

Table 7-4 relates each of the temperature monitor blocks to the signal conditioning block (SCB) in which it is located, along with the shared functions for the temperature monitor input pad.

***Table 7-4* • Temperature Monitor Input Pad Designations**

ADC	SCB	Temp. Monitor Name	Pad Name	Function Name	Shared Functions on Pad		
ADC0	SCB0	TM0	TM0	TM0_IO	ADC0_CH4	CMP0_N	CM0_L
			GNDTM0	TM0_GND TM1_GND			
	SCB1	TM1					
			TM1	TM1_IO	ADC0_CH8	CMP2_N	CM1_L
ADC1	SCB2	TM2	TM2	TM2_IO	ADC1_CH4	CMP4_N	CM2_L
			GNDTM1	TM2_GND TM3_GND			
	SCB3	TM3					
			TM3	TM3_IO	ADC1_CH8	CMP6_N	CM3_L
ADC2*	SCB4*	TM4*	TM4*	TM4_IO*	ADC2_CH4*	CMP8_N*	CM4_L*
			GNDTM2*	TM4_GND*			

*Note: *Components indicated are available only in the A2F500.*

Temperature Monitor Configuration

The temperature monitors can be dynamically configured via registers in the analog compute engine (ACE). These registers can be controlled by any of the following:

1. The ACE itself
2. An ARM Cortex-M3 program
3. User logic (or soft processor) via the AHB bus master connected to the FPGA fabric
4. The peripheral DMA bus master

Each temperature monitor can individually be placed in a power save mode (disabled). The strobes used to individually or collectively start and stop the temperature monitor automatic switch sequence are also located in ACE registers, and can be accessed in all the ways listed above.

The temperature monitor configuration options (the enable, for example) can be set statically using the SmartDesign MSS configurator. They can also be changed at any time dynamically using any of the three ABM bus masters listed above, or by the ACE's sample sequencer engine (SSE) or post-processor engine (PPE). Besides the static configuration options, the SmartDesign tool also automatically creates the SSE microcode required to control the temperature monitor strobes and ADC control registers.

Table 7-5 on page 88 summarizes the configuration settings related to the temperature monitors and temperature monitor input pad shared functions. Only the temperature monitor related bits are described here; bits in the same or adjacent registers that are used for other SCB configuration functions are not documented in this section. For information on how to select a particular ADC input channel or how to operate the ADC, refer to the "Analog-to-Digital Converter (ADC)" section on page 27.

Table 7-5 and Table 7-6 on page 89 summarize configuration settings related to the temperature monitor.

***Table 7-5* • Addresses for Temperature Monitor-Related Configuration Control Bits**

SCB Byte and Bit Address (relative to each SCB)	Advanced Bus Matrix (ABM) Address (absolute 32-bit address)[1]	Analog Compute Engine SSE and PPE Address (absolute 8-bit address)	Bit Name	Description
SCB_0 (0x40020204/0x81 – 0x40020230/0x8C)				
B9 [2]	0x40020228 [2]	0x40020228 [2]	CM_EN	CM0 enable
B9 [4]	0x40020228 [4]	0x8A [4]	COMP_EN	CMP0 comparator enable
B10 [0]	0x4002022C [0]	0x8B [0]	TMB_MUX_SEL	TM0 bypass MUX
B10 [1]	0x4002022C [1]	0x8B [1]	TMB_DI_ON	TM0 isolation switch
B10 [2]	0x4002022C [2]	0x8B [2]	TM_EN	TM0 enable
B10 [3]	0x4002022C [3]	0x8B [3]	TM_STB	TM0 strobe
SCB_1 (0x40020234/0x8D – 0x40020260/0x98)				
B9 [2]	0x40020258 [2]	0x96 [2]	CM_EN	CM1 enable
B9 [4]	0x40020258 [4]	0x96 [4]	COMP_EN	CMP2 comparator enable
B10 [0]	0x4002025C [0]	0x97 [0]	TMB_MUX_SEL	TM1 bypass MUX
B10 [1]	0x4002025C [1]	0x97 [1]	TMB_DI_ON	TM1 isolation switch
B10 [2]	0x4002025C [2]	0x97 [2]	TM_EN	TM1 enable
B10 [3]	0x4002025C [3]	0x97 [3]	TM_STB	TM1 strobe
SCB_2 (0x40020264/0x99 – 0x40020290/0xA4)				
B9 [2]	0x40020288 [2]	0xA2 [2]	CM_EN	CM2 enable
B9 [4]	0x40020288 [4]	0xA2 [4]	COMP_EN	CMP4 comparator enable
B10 [0]	0x4002028C [0]	0xA3 [0]	TMB_DI_ON	TM2 bypass MUX
B10 [1]	0x4002028C [1]	0xA3 [1]	TMB_DI_ON	TM2 isolation switch
B10 [2]	0x4002028C [2]	0xA3 [2]	TM_EN	TM2 enable
B10 [3]	0x4002028C [3]	0xA3 [3]	TM_STB	TM2 strobe
SCB_3 (0x40020294/0xA5 – 0x400202C0/0xB0)				
B9 [2]	0x400202B8 [2]	0xAE [2]	CM_EN	CM3 enable
B9 [4]	0x400202B8 [4]	0xAE [4]	COMP_EN	CMP6 comparator enable
B10 [0]	0x400202BC [0]	0xAF [0]	TMB_DI_ON	TM3 bypass MUX
B10 [1]	0x400202BC [1]	0xAF [1]	TMB_DI_ON	TM3 isolation switch
B10 [2]	0x400202BC [2]	0x97 [2]	TM_EN	TM3 enable
B10 [3]	0x400202BC [3]	0x97 [3]	TM_STB	TM3 strobe

Notes:

1. *Only the least-significant byte of each 32-bit data word is utilized.*
2. *Components indicated with an asterisk (*) are available only in the A2F500.*

***Table 7-5* • Addresses for Temperature Monitor-Related Configuration Control Bits (continued)**

SCB Byte and Bit Address (relative to each SCB)	Advanced Bus Matrix (ABM) Address (absolute 32-bit address)[1]	Analog Compute Engine SSE and PPE Address (absolute 8-bit address)	Bit Name	Description
SCB_4* (0x400202C4/0xB1 – 0x400202F0/0xBC)				
B9 [2]	0x400202E8 [2] *	0xBA [2] *	CM_EN*	CM4 enable*
B9 [4] *	0x400202E8 [4] *	0xBA [4] *	COMP_EN*	CMP8 comparator enable*
B10 [0] *	0x400202EC [0] *	0xBB [0] *	TMB_DI_ON*	TM4 bypass MUX *
B10 [1] *	0x400202EC [1] *	0xBB [1] *	TMB_DI_ON*	TM4 isolation switch*
B10 [2] *	0x400202EC [2] *	0xBB [2] *	TM_EN*	TM4 enable*
B10 [3] *	0x400202EC [3] *	0xBB [3] *	TM_STB*	TM4 strobe*

Notes:

1. *Only the least-significant byte of each 32-bit data word is utilized.*
2. *Components indicated with an asterisk (*) are available only in the A2F500.*

***Table 7-6* • Functional Description of Comparator-Related Configuration Control Bits**

Bit Name	Description	R/W	Reset Value	Function
TM_EN	TM enable	R/W	0b0	0 = Power save mode 1 = Operational mode
TM_STB	TM strobe	R/W	0b0	0 = Idle mode 1 = Initiate sequence and enter hold mode
CM_EN	CM enable	R/W	0b0	0 = Current monitor power save mode 1 = Current monitor operational mode
COMP_EN	CMP enable	R/W	0b0	0 = Comparator power save mode 1 = Comparator operational mode
TMB_MUX_SEL	TM bypass MUX	R/W	0b0	0 = ADC input MUX driven from temperature monitor 1 = ADC input MUX driven directly from TM[n] pad
TMB_DI_ON	TM isolation switch	R/W	0b0	0 = Isolation switch open (isolating high voltages) 1 = Isolation switch closed (direct ADC input)

When using the temperature monitor, a common configuration would be the one shown in Table 7-7:

***Table 7-7* • Common Configuration for Temperature Monitor**

Parameter	Function
TM_EN = 1	Temperature monitor operational mode
TM_STB = {0, 1}	Start in idle mode (0) and strobe (1) to start sequence.
CM_EN = 0	Disable current monitor.
COMP_EN = 0	Disable comparator.
TMB_MUX = 0	Selects temperature monitor over direct input option.
TMB_DI_ON = 0	High-voltage isolation switch open

Related Information

SmartFusion Customizable System-on-Chip (cSoC) datasheet
http://www.microsemi.com/soc/documents/SmartFusion_DS.pdf

8 – High-Speed Comparators

Comparator Features

- Fast response time
- Continuous-time operation
- Differential inputs
- Programmable hysteresis
- Large common mode input range
- Large over-voltage tolerance
- DAC can supply threshold voltage
- Low input bias and offset currents
- Low offset voltage
- Low operating power
- Even lower power sleep mode
- Output drives FPGA fabric directly
- Output can be read by ARM Cortex-M3
- Cortex-M3 edge-triggered interrupts

Comparator General Description

SmartFusion devices contain a number of differential, analog input, high speed, continuous-time comparators with programmable hysteresis. There are two identical comparators in each SmartFusion signal conditioning block (SCB), though their input pads are connected to different shared circuits. A2F200 devices have four SCBs, for a total of eight comparators in each device. A2F500 devices have five SCBs, for a total of ten comparators in each device. The digital output of each comparator goes to the FPGA fabric, where it can be used as an input to the configurable logic. Output also goes to the analog compute engine (ACE) where it can be read or used to generate edge-triggered interrupts to the Cortex-M3.

Symbol

Figure 8-1 shows the Comparator symbol.

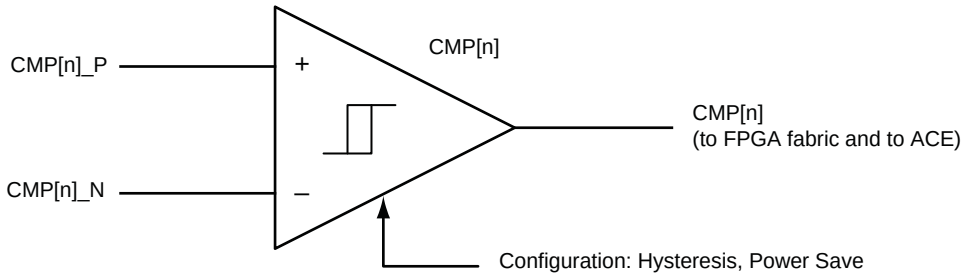
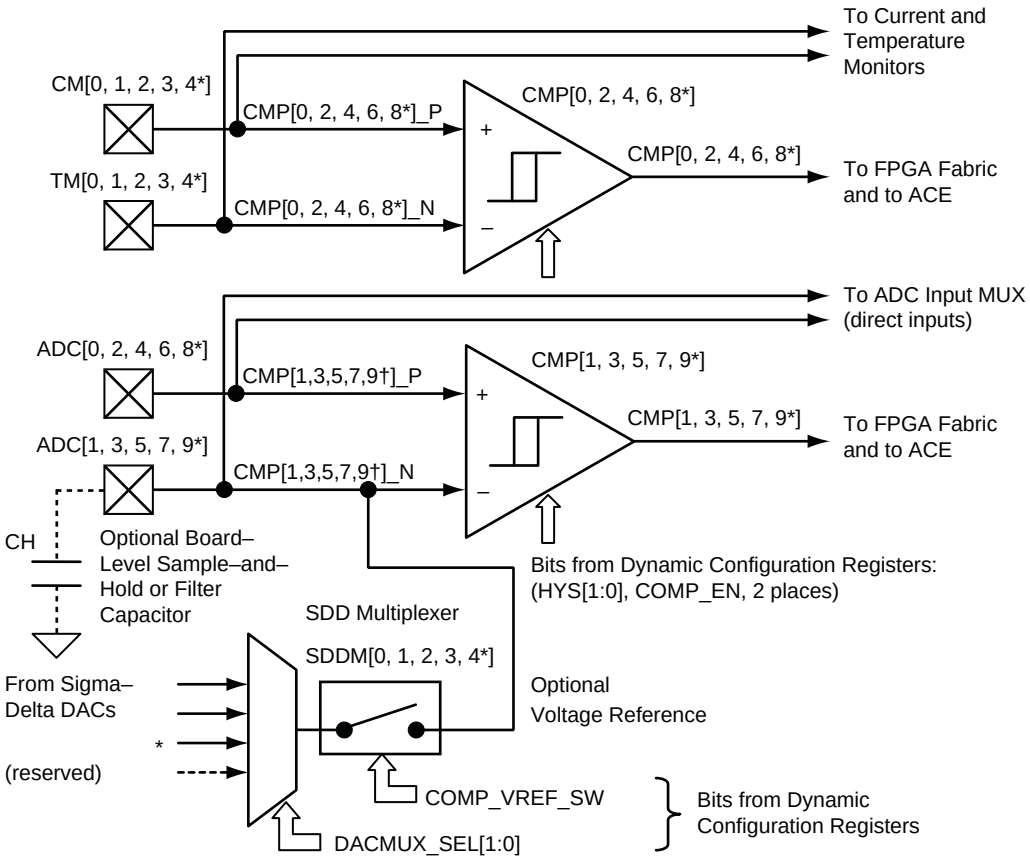


***Figure 8-1* • Comparator Symbol**

Comparator Detailed Description

The comparator generates a digital output signal based upon the instantaneous voltage difference measured at its two input pins. If the positive input, CMP[n]_P, voltage is greater than the negative, or inverting input, CMP[n]_N, (also sometimes called the reference input in DC applications), the output, CMP[n], is a logic 1. The character [n] is replaced by the comparator instance number, 0 through 1 in the A2F060, 0 through 7 in the A2F200 or 0 through 9 in the A2F500. If the negative input has the more positive voltage, the comparator output is a logic zero. The comparators are designed to respond almost exclusively to the differential input voltage, and to be insensitive to the common mode voltage on the inputs. The design of the comparator also minimizes the effect of power supply variations and temperature on performance.

One comparator in each SCB (designated CMP0, CMP2, CMP4, CMP6, or CMP8; with an even instance number) gets its two inputs from pins which it shares with the current monitor and temperature monitor. The other comparator (designated CMP1, CMP3, CMP5, CMP7, or CMP9; with an odd instance number) gets its two inputs from pins shared with SmartFusion direct ADC and LVTTL inputs. See Figure 8-2. Since the comparators share their input pins with these other functions, it may not be feasible in many cases to use a comparator if the alternate pin function is being used. The comparators are designed to tolerate the large common mode or differential over-voltages that may be normal for the alternate function, such as the current monitor, which is designed to measure the current on power supplies of +12 V or more if the comparator is disabled. If enabled, the comparator voltage tolerance is reduced somewhat, and in any case the comparator only functions with input voltages up to 2.56 V. See the "Comparator" section the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet.



*Note: *Components are available only in the A2F500.*

***Figure 8-2* • Connections for the Two Comparators in SmartFusion SCBs**

For a detailed description of the current monitor function, which shares pads CM[n] and TM[n] with a comparator, see the "Current Monitor" section on page 71.

For a detailed description of the temperature monitor function, which shares the pad TM[n] with a comparator, see the "Temperature Monitor" section on page 81.

To understand the alternate functions of the ADC direct input pads, including their LVTTL function, see the "Direct ADC Inputs" section on page 103.

To understand how to set the ADC input multiplexer to a selected channel, or how to configure and operate the ADC itself, see the "Analog-to-Digital Converter (ADC)" section on page 27.

Table 8-1 relates each of the current monitor blocks to the signal conditioning block (SCB) in which it is located along with the shared functions for the current monitor input pads:

***Table 8-1* • Comparator Input Pad Designations and Shared Functions**

ADC/ SDD	Signal Conditioning Block	Comparator Name	Pad Name	Function Name	Shared Functions on Pad		
ADC0 SDD0	SCB0	CMP0	CM0	CMP0_P	ADC0_CH3	CM0_H	
			TM0	CMP0_N	ADC0_CH4	CM0_L	TM0_IO
		CMP1	ADC0	CMP1_P	ADC0_CH9	LVTTL0_IN	
			ADC1	CMP1_N	ADC0_CH10	LVTTL1_IN	SDDM0_OUT
	SCB1	CMP2	CM1	CMP2_P	ADC0_CH7	CM1_H	
			TM1	CMP2_N	ADC0_CH8	CM1_L	TM1_IO
		CMP3	ADC2	CMP3_P	ADC0_CH11	LVTTL2_IN	
			ADC3	CMP3_N	ADC0_CH12	LVTTL3_IN	SDDM1_OUT
ADC1 SDD1	SCB2	CMP4	CM2	CMP4_P	ADC1_CH3	CM2_H	
			TM2	CMP4_N	ADC1_CH4	CM2_L	TM2_IO
		CMP5	ADC4	CMP5_P	ADC1_CH9	LVTTL4_IN	
			ADC5	CMP5_N	ADC1_CH10	LVTTL5_IN	SDDM2_OUT
	SCB3	CMP6	CM3	CMP6_P	ADC1_CH7	CM3_H	
			TM3	CMP6_N	ADC1_CH8	CM3_L	TM3_IO
		CMP7	ADC6	CMP7_P	ADC1_CH11	LVTTL6_IN	
			ADC7	CMP7_N	ADC1_CH12	LVTTL7_IN	SDDM3_OUT
ADC2* SDD2*	SCB4*	CMP8*	CM4*	CMP8_P*	ADC2_CH3*	CM4_H*	
			TM4*	CMP8_N*	ADC2_CH4*	CM4_L*	TM4_IO*
		CMP9*	ADC8*	CMP9_P*	ADC2_CH9*	LVTTL8_IN*	
			ADC9*	CMP9_N*	ADC2_CH10*	LVTTL9_IN*	SDDM4_OUT*

*Note: *Components indicated are only available in the A2F500.*

The negative input terminal of the second (odd-numbered) comparator can be fed a reference signal selected by an analog multiplexer from any of the SmartFusion on-chip sigma-delta DACs (SDDs). The selected DAC output signal is connected to the comparator and ADC direct input pin via an analog switch that can be controlled by the ACE. Thus, any of the SmartFusion SDDs can be used to provide the reference (inverting) input to the odd-numbered comparator in each signal conditioning block. By placing a sample-and-hold capacitor on this pin at the board level (shown as C_H in Figure 8-2 on page 92), it is possible to timeshare the SDD output. The sample-and-hold capacitor must be recharged periodically when the voltage level stored on the capacitor droops due to current leakage. Since the hold (droop) time

can be designed to be longer than the sample time, one SDD can service multiple sample-and-hold capacitors, being de-multiplexed using the analog switch in each SCB.

Alternatively, the SDD can be dedicated to continuously supply a reference voltage to one or possibly more of the odd-numbered comparators in different SCBs in parallel, in which case a board-level capacitor is optional, since the SDD includes a third-order on-chip low-pass filter already. Using the capacitor in this case may help further reduce the noise coming from the SDD or from other sources by forming a fourth low-pass filter pole, potentially with a much lower cutoff frequency than the on-chip filter poles, due to the availability of much larger capacitors at the board level.

The hysteresis of each comparator is individually configurable. There are four possible settings: no hysteresis (0 mV), ±10 mV, ±30 mV, and ±100 mV nominal hysteresis. The hysteresis acts upon the threshold for both rising and falling signals. For example, if the hysteresis were set to ±10 mV, the comparator would only respond to a rising signal once the voltage on the positive input terminal exceeded that on the negative terminal by approximately 10 mV. For a falling signal, the comparator output would switch state only once the voltage on the positive input terminal was 10 mV less than that on the negative terminal. Therefore, a total of 20 mV exists between the positive-going and negative-going threshold voltages when using the ±10 mV setting. These are nominal hysteresis values which will only be approximated in an actual device. The comparator input offset voltage will shift both threshold voltages slightly up or down, as well.

Comparator Configuration

The comparator can be dynamically configured via registers in the analog compute engine (ACE). These registers can be controlled by the ACE itself, a Cortex-M3 program, by user logic (or soft processor) via the AHB bus master connected to the FPGA fabric, or by the peripheral DMA bus master.

The comparators can individually be placed in a power save mode (disabled), and each comparator can be configured for the desired hysteresis selection.

Configuration settings closely related to the comparator include the SDD analog multiplexer, which selects the DAC that will potentially drive the comparator reference pin in each signal conditioning block, and the analog switch, which actually connects or disconnects the selected digital-to-analog converter to the odd-numbered comparator negative input pin. The analog switch also can be used to provide sample-and-hold operation if a board-level sample-and-hold capacitor is connected to the ADC[1, 3, 5, 7, 9] pad (the CMP[1, 3, 5, 7, 9]_N function): when the switch is closed, the SDD charges the capacitor (sample mode). When the switch is open, the capacitor holds the voltage to which it has been charged (hold mode).

The comparator and comparator-related configuration options can be set statically using the SmartDesign MSS configurator. They can also be changed at any time dynamically using any of the three ABM bus masters listed above, or by the ACE's sample sequencer engine (SSE) or post-processing engine (PPE).

Table 8-2 gives a comparison of relative and absolute addresses for configuration control bits.

***Table 8-2* • Relative and Absolute Addresses for Comparator-Related Configuration Control Bits**

SCB Byte and Bit Address (relative to each SCB)	Advanced Bus Matrix (ABM) Address (absolute 32-bit address)[1]	Analog Compute Engine SSE and PPE Address (absolute 8-bit address)	Bit Name	Description
SCB_0 (0x40020204/0x81 – 0x40020230/0x8C)				
B9[4]	0x40020228 [4]	0x8A [4]	COMP_EN	CMP0 Enable
B9[7:6]	0x40020228 [7:6]	0x8A [7:6]	HYS[1:0]	CMP0 Hysteresis

Notes:

1. *Only the least-significant byte of each 32-bit data word is utilized.*
2. *Components indicated with an asterisk (*) are available only in the A2F500.*

Table 8-2 • **Relative and Absolute Addresses for Comparator-Related Configuration Control Bits (continued)**

SCB Byte and Bit Address (relative to each SCB)	Advanced Bus Matrix (ABM) Address (absolute 32-bit address)[1]	Analog Compute Engine SSE and PPE Address (absolute 8-bit address)	Bit Name	Description
B10[4]	0x4002022C [4]	0x8B [4]	COMP_EN	CMP1 Enable
B10[5]	0x4002022C [5]	0x8B [5]	COMP_VREF_SW	SCB0 SDD Analog Switch (to CMP1)
B10[7:6]	0x4002022C [7:6]	0x8B [7:6]	HYS[1:0]	CMP1 Hysteresis
B11[1:0]	0x40020230 [1:0]	0x8C [1:0]	DAC_MUXSEL[1:0]	SDDM0 Analog MUX (to CMP1)
SCB_1 (0x40020234/0x8D – 0x40020260/0x98)				
B9[4]	0x40020258 [4]	0x96 [4]	COMP_EN	CMP2 Enable
B9[7:6]	0x40020258 [7:6]	0x96 [7:6]	HYS[1:0]	CMP2 Hysteresis
B10[4]	0x4002025C [4]	0x97 [4]	COMP_EN	CMP3 Enable
B10[5]	0x4002025C [5]	0x97 [5]	COMP_VREF_SW	SCB1 SDD Analog Switch (to CMP3)
B10[7:6]	0x4002025C [7:6]	0x97 [7:6]	HYS[1:0]	CMP3 Hysteresis
B11[1:0]	0x40020260 [1:0]	0x98 [1:0]	DAC_MUXSEL[1:0]	SDDM1 Analog MUX (to CMP3)
SCB_2 (0x40020264/0x99 – 0x40020290/0xA4)				
B9[4]	0x40020288 [4]	0xA2 [4]	COMP_EN	CMP4 Enable
B9[7:6]	0x40020288 [7:6]	0xA2 [7:6]	HYS[1:0]	CMP4 Hysteresis
B10[4]	0x4002028C [4]	0xA3 [4]	COMP_EN	CMP5 Enable
B10[5]	0x4002028C [5]	0xA3 [5]	COMP_VREF_SW	SCB2 SDD Analog Switch (to CMP5)
B10[7:6]	0x4002028C [7:6]	0xA3 [7:6]	HYS[1:0]	CMP5 Hysteresis
B11[1:0]	0x40020290 [1:0]	0xA4 [1:0]	DAC_MUXSEL[1:0]	SDDM2 Analog MUX (to CMP5)
SCB_3 (0x40020294/0xA5 – 0x400202C0/0xB0)				
B9[4]	0x400202B8 [4]	0xAE [4]	COMP_EN	CMP6 Enable
B9[7:6]	0x400202B8 [7:6]	0xAE [7:6]	HYS[1:0]	CMP6 Hysteresis
B10[4]	0x400202BC [4]	0xAF [4]	COMP_EN	CMP7 Enable
B10[5]	0x400202BC [5]	0xAF [5]	COMP_VREF_SW	SCB3 SDD Analog Switch (to CMP7)
B10[7:6]	0x400202BC [7:6]	0xAF [7:6]	HYS[1:0]	CMP7 Hysteresis
B11[1:0]	0x400202C0 [1:0]	0xB0 [1:0]	DAC_MUXSEL[1:0]	SDDM3 Analog MUX (to CMP7)
SCB_4* (0x400202C4/0xB1 – 0x400202F0/0xBC)				
B9[4]*	0x400202E8 [4]*	0xBA [4]*	COMP_EN*	CMP8 Enable*
B9[7:6]*	0x400202E8 [7:6]*	0xBA [7:6]*	HYS[1:0]*	CMP8 Hysteresis*
B10[4]*	0x400202EC [4]*	0xBB [4]*	COMP_EN*	CMP9 Enable*

Notes:

1. *Only the least-significant byte of each 32-bit data word is utilized.*
2. *Components indicated with an asterisk (*) are available only in the A2F500.*

***Table 8-2* • Relative and Absolute Addresses for Comparator-Related Configuration Control Bits (continued)**

SCB Byte and Bit Address (relative to each SCB)	Advanced Bus Matrix (ABM) Address (absolute 32-bit address)[1]	Analog Compute Engine SSE and PPE Address (absolute 8-bit address)	Bit Name	Description
B10[5]*	0x400202EC [5]*	0xBB [5]*	COMP_VREF_SW*	SCB4 SDD Analog Switch (to CMP9)*
B10[7:6]*	0x400202EC [7:6]*	0xBB [7:6]*	HYS[1:0]*	CMP9 Hysteresis*
B11[1:0]*	0x400202F0 [1:0]*	0xBC [1:0]*	DAC_MUXSEL[1:0]*	SDDM4 Analog MUX (to CMP9)*

Notes:

1. *Only the least-significant byte of each 32-bit data word is utilized.*
2. *Components indicated with an asterisk (*) are available only in the A2F500.*

Table 8-3 gives the functions of the Comparator configuration control bits.

***Table 8-3* • Functional Description of Comparator-Related Configuration Control Bits**

Bit Name	Description	R/W	Reset Value	Function
COMP_EN	Comparator Enable	R/W	0b0	0 = Power Save (Comparator Output = TBD) 1 = Operational Mode
HYS[1:0]	Comparator Hysteresis	R/W	0b00	00 = No Hysteresis 01 = ± 10 mV nominal 10 = ± 30 mV nominal 11 = ± 100 mV nominal
COMP_VREF_SW	SDD Analog Switch	R/W	0b0	0 = SDD V_{REF} switch off (high impedance) 1 = SDD V_{REF} switch on (conducting)
DAC_MUXSEL[1:0]	SDD Analog Multiplexer	R/W	0b00	00 = Select SDD0_OUT 01 = Select SDD1_OUT 10 = Select SDD2_OUT* 11 = Reserved

*Note: * Components indicated are only available in the A2F500.*

Comparator Outputs and Interrupts

The comparator outputs go directly to the FPGA fabric, where they can be used to drive configurable logic, and also to the analog compute engine (ACE), where they are mapped into the memory space of the Cortex-M3 via the advanced high-speed bus matrix (ABM). There they can be read by three ABM bus masters: the Cortex-M3, the FPGA fabric, and the peripheral DMA bus masters. In addition, the ACE can use any of the comparator outputs to generate an edge-triggered interrupt to the Cortex-M3.

To use the high-speed comparator outputs in user logic in the FPGA fabric, the relevant comparators must be instantiated in the SmartDesign MSS configurator for the analog compute engine, and then the ACE output ports that appear can be stitched to other blocks in the design. Table 8-4 gives the register map for the comparator.

Table 8-4 • **Comparator Register Map**

Register Name	ABM Address	R/W	Reset Value	Description
COMPARATOR_STATUS	0x4002100C	R	0x000	Comparator Output Levels
COMP_IRQ_EN	0x4002120C	R/W	0x000000	Comparator interrupt enables
COMP_IRQ	0x40021210	R	0x000000	Comparator edges detected
COMP_IRQ_CLR	0x40021214	W	0x000000	Clears COMP_IRQ bits

The COMPARATOR_STATUS register (Table 8-5) is used to read the output status of the comparators by any of the three bus masters with access to the ACE slave. If the comparator INP pin is more positive than the INN pin at the time of the read command (accounting for propagation delays), then the corresponding bit in the comparator output status register will be read as a logical 1.

Table 8-5 • **Comparator Status Output Register Bit Fields**

Bit Number	Name	R/W	Reset Value	Description / Location
31:10	Reserved	R		Reserved
9*	COMPARATOR	R	0b0	CMP9 Output*
8*	COMPARATOR	R	0b0	CMP8 Output*
7	COMPARATOR	R	0b0	CMP7 Output
6	COMPARATOR	R	0b0	CMP6 Output
5	COMPARATOR	R	0b0	CMP5 Output
4	COMPARATOR	R	0b0	CMP4 Output
3	COMPARATOR	R	0b0	CMP3 Output
2	COMPARATOR	R	0b0	CMP2 Output
1	COMPARATOR	R	0b0	CMP1 Output
0	COMPARATOR	R	0b0	CMP0 Output

*Note: * Bits indicated are only available in the A2F500; otherwise they are reserved.*

The registers shown in Table 8-6 on page 98 record an edge-triggered event on any of the comparator digital output signals. Each of the eight or ten comparator outputs drives two bits: one for detecting a rising edge, and one for detecting a falling edge. These events can be used to trigger an interrupt in the Cortex-M3 priority interrupt logic.

Events can be masked and cleared by other registers, as shown in Table 8-6.

***Table 8-6* • Bit Mapping Between Comparators, Interrupt Control Registers, and Cortex-M3 Interrupts**

COMP_IRQ_EN	COMP_IRQ	COMP_IRQ_CLR	Register Name	
0x4002120C	0x40021210	0x40021214	Register Address in ABM Address Space	
R/W	R	W	R/W	
0x000000	0x000000	0x000000	Reset Value	
Bit Number			Cortex-M3 Interrupt No.	Relevant Comparator and Edge (Rising or Falling)
31:24	31:24	31:24		Reserved
Comparator Rising Edge Interrupts				
23:22	23:22	23:22	108-107	Reserved
21*	21*	21*	106*	CMP9 Output Rising*
20*	20*	20*	105*	CMP8 Output Rising*
19	19	19	104	CMP7 Output Rising
18	18	18	103	CMP6 Output Rising
17	17	17	102	CMP5 Output Rising
16	16	16	101	CMP4 Output Rising
15	15	15	100	CMP3 Output Rising
14	14	14	99	CMP2 Output Rising
13	13	13	98	CMP1 Output Rising
12	12	12	97	CMP0 Output Rising
Comparator Falling Edge Interrupts				
11:10	11:10	11:10	96:95	Reserved
9*	9*	9*	94*	CMP9 Output Falling*
8*	8*	8*	93*	CMP8 Output Falling*
7	7	7	92	CMP7 Output Falling
6	6	6	91	CMP6 Output Falling
5	5	5	90	CMP5 Output Falling
4	4	4	89	CMP4 Output Falling
3	3	3	88	CMP3 Output Falling
2	2	2	87	CMP2 Output Falling
1	1	1	86	CMP1 Output Falling
0	0	0	85	CMP0 Output Falling

*Note: * Bits indicated are only available in the A2F500; otherwise they are reserved.*

COMP_IRQ_EN bits are logically ANDed with the bits in the COMP_IRQ register to create interrupts to the Cortex-M3. If any of the bits in this register are logic 1, the corresponding bit in the COMP_IRQ register is enabled; otherwise the corresponding bit in the COMP_IRQ register is disabled/masked in its contribution to the interrupt outputs.

Set bits in the COMP_IRQ register indicate rising or falling edge events on any or all digital outputs of the eight comparators in the SmartFusion FPGA Signal Conditioning Blocks (SCBs). These bits are bitwise logically ANDed with bits in COMP_IRQ_EN to create Cortex-M3 interrupts.

The write-only bits in the COMP_IRQ_CLR register are used to clear corresponding bits in the COMP_IRQ register. Writing a 1 to any of the bits will clear the corresponding bits in the COMP_IRQ register, while writing a 0 to any of the bits will have no effect. In the event that writing a 1 to these clear bits coincides with a set event in the COMP_IRQ register, the set event shall have higher priority.

Comparator Application – Using One SDD to Set Multiple Comparator Reference Voltages

Each sigma-delta digital-to-analog converter (SDD) goes to an analog multiplexer in each of the signal conditioning blocks (SCBs). The output of the multiplexer goes to an analog switch, which can be used as input to a board-level hold capacitor (see Figure 8-2 on page 92.) One SDD can drive several of these hold capacitors—up to one per SCB—by setting the multiplexer in each SCB to select the same SDD. Then, by controlling the analog switches, the selected DAC can alternately be connected to each of the several hold capacitors while simultaneously changing the digital value at the input of the SDD in a time-division fashion. The analog switches act together to de-multiplex the piecewise analog signal, with one piece (potentially unique voltage) being applied to each of the hold capacitors.

The approximate time-constant and cutoff (–3 dB) frequency of the new pole generated by adding a board-level capacitor (C_H) can be calculated using the nominal output impedance of the sigma-delta DAC, which is R_{DAC} = 10 KΩ:

$$\tau_c = R_{DAC} \times C_H \text{ seconds}$$

EQ 8-1

$$f_c = 1 / (2\pi R_{DAC} \times CH) \text{ Hz}$$

EQ 8-2

For example, the time-constant and cutoff frequency formed by setting C_H = 100 nF are 1 ms and 159 Hz, respectively.

To achieve settling to a 0.1 percent accuracy (2.56 mV) requires approximately 7 time-constants (note $10^{-3} \cong e^{-6.9}$), or for this example, about 7 milliseconds. This would be the sample time required for the SDD to impose a large change of voltage on C_H.

Bias and leakage currents will cause the hold capacitor to droop while it is in hold mode. The droop rate is proportional to the bias current and inversely proportional to the size of the hold capacitor. For example, if C_H = 100 nF and the bias current is I_B = 0.5 µA (maximum), the droop rate is 5 µV / µs, maximum. The shortest time in which a one percent error could develop would be 5.1 ms.

The DC bias current creates a DC offset voltage. This is computed as the bias current times the DAC output impedance. If the bias current is I_B = 0.5 µA and the DAC output resistance is R_{DAC} = 10 KΩ, then the offset voltage generated would be V_{OFFSET} = 5 mV.

When refreshing a nominally constant voltage there will be some ripple due to the droop that is caused by bias currents during the hold mode, followed by the exponential charging in the sample mode (in

combination with the continuing bias currents). In the steady state, these two voltages are equal in amplitude (Figure 8-3.

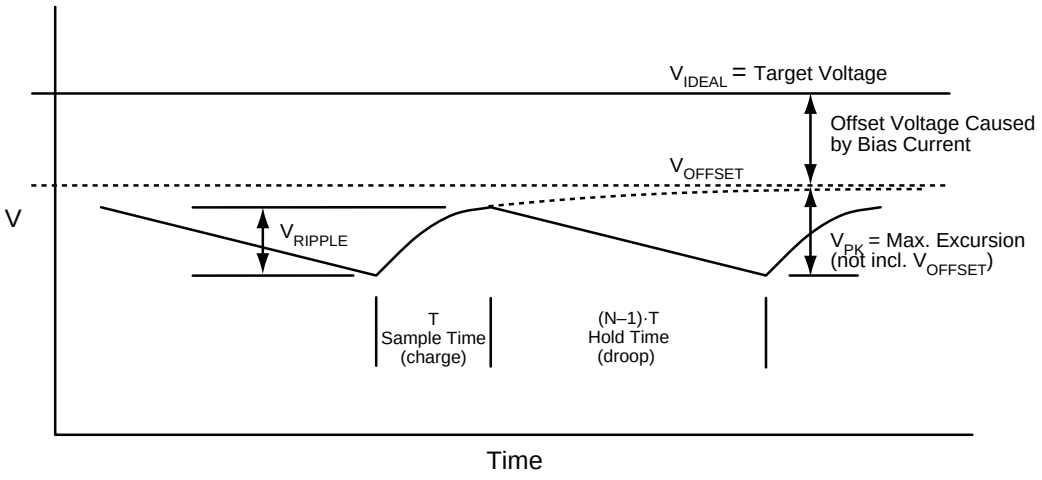


***Figure 8-3* • Steady-State Waveform with Ripple Due to Drooping and Recharging**

The length of the sample is time T and the total time to cycle through N capacitors is then N times T, with N being 2, 3, or 4, since the A2F200 has four SCBs. Each capacitor is refreshed for a time, T, and then droops for a time, (N-1) × T, while the remaining capacitors are serviced.

EQ 8-3 shows V_{RIPPLE} computed two ways; one based upon the droop during the hold period, and the other based upon the exponential charging that occurs during the sample period less the droop during that phase:

$$V_{RIPPLE} = \frac{I_B}{C_H} \cdot (N-1) \cdot T = e^{-\left(\frac{T}{R_{DAC} \cdot C_H}\right)} \cdot V_{PK} - \left(\frac{I_B}{C_H} \cdot T\right)$$

EQ 8-3

where *e* is Euler's number (2.71828...)

Solving for V_{PK} gives EQ 8-4.

$$V_{PK} = \frac{I_B \cdot N \cdot T \cdot e^{\left(\frac{T}{C_H \cdot R_{DAC}}\right)}}{C_H}$$

EQ 8-4

For example, if:

I_B = 0.5 µA (maximum bias current)

C_H = 100 nF (hold capacitor capacitance)

R_{DAC} = 10 KΩ (SDD typical output impedance)

T = 100 µs (sample time)

N = 4 (total number of sample periods)

Then:

V_{OFFSET} = 5 mV (maximum DC offset due to bias current)

V_{RIPPLE} = 1.5 mV (peak-to-peak AC ripple due to droop and refresh cycle)

V_{PK} = 2.21 mV (maximum excursion due to refresh cycle; not including V_{OFFSET})

τ_C = 1 ms (exponential charging time-constant)

f_C = 159 Hz (–3 dB cutoff frequency)

V_{RIPPLE} and V_{PK} can be reduced by selecting a larger hold capacitor, in which case the charging time-constant will also increase (so the time required for a large change in voltage will be longer) and the cutoff frequency will decrease. V_{RIPPLE} and V_{PK} can also be reduced by decreasing the sample time, though if it gets too short then secondary effects such as charge injection from the analog switch could start to become a factor.

Related Information

Datasheets

SmartFusion Customizable System-on-Chip (cSoC) datasheet

http://www.microsemi.com/soc/documents/SmartFusion_DS.pdf

User's Guides

SmartFusion Microcontroller Subsystem User's Guide

http://www.microsemi.com/soc/documents/SmartFusion_MSS_UG.pdf

9 – Direct ADC Inputs

Direct ADC Input Features

- Input drives ADC input MUX directly
- No active amplifiers in signal chain
- Full ADC accuracy
- Wide bandwidth
- Each channel factory calibrated
- Low noise
- No additional power required

Direct ADC Input General Description

SmartFusion devices contain a number of direct ADC input paths. These signal paths go directly to the ADC main multiplexer without going through any active gain stages; they go only through some analog switches. If that channel is selected by the ADC multiplexer, the input pad is effectively connected directly to the sampling capacitor of the ADC. The operational input voltage range of direct inputs is from zero to VAREF (nominally 2.56 V, unless VAREF is externally supplied).

The direct ADC inputs are arrayed as follows:

- For each signal conditioning block (SCB) there are two direct ADC inputs, both sharing their input pads with the current monitor (and other functions).
- There are four additional direct ADC Inputs associated with every ADC, almost all of which share their input pads with comparators.
- There is one direct ADC input associated with each DAC, sharing its input pad with the DAC output.

In the A2F200 there are four SCBs, which have eight possible direct ADC inputs, plus there are two ADCs with another eight direct ADC inputs, and two DACs, each with a direct ADC input, for a total of up to 18 direct ADC inputs.

In the A2F500 there are five SCBs having ten possible direct ADC inputs, plus there are three ADCs having another twelve direct ADC inputs, and three DACs, each with a direct ADC input, for a total of up to 25 direct ADC inputs. Because the third ADC in the A2F500 has only one SCB associated with it instead of two, two of the direct ADC inputs associated with the ADC do not have a comparator connected in parallel, as all the others do.

The post Processing Engine (PPE) in the Analog Compute Engine (ACE) applies the factory-determined offset and gain compensation values for each Direct ADC Input channel.

Direct ADC Input Symbol

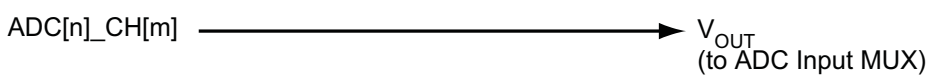


***Figure 9-1* • Direct ADC Input Symbol**

In the signal/function name shown, the ADC number “[n]” can be one of [0, 1] for the A2F200 and [0, 1, 2] for the A2F300. The channel number “m” can be one of [3, 4, 7, 8, 9, 10, 11, 12, 15].

Channels [3, 4, 7, 8] are those that are associated with current and temperature monitors. (Those monitors can be bypassed for a Direct ADC Input path). Channels [9, 10, 11, 12] are those which share inputs with comparators, and channel 15 shares its input with the SDD output. This is shown more explicitly in Figure 9-2 on page 105.

Direct ADC Input Detailed Description

There are three main types of direct ADC inputs:

1. Those that share pads with the current monitor (in addition to other functions) which have potentially high input voltages at their input pads
2. Those that share pads with comparators only
3. Those that share pads with the Sigma-Delta DAC output

The inputs that are shared with the current monitor have additional multiplexing in the signal path in order to select the direct inputs, bypassing the current monitor and/or temperature monitor outputs with which they share ADC input channels. They also have isolation switches to protect the ADC and the ADC main input multiplexer from the potentially high voltages that could be present on these pads when the current monitor is in use.

The inputs that are shared only with the comparators, which operate at voltages compatible with the ADC, go directly to the ADC main input multiplexer, as do the inputs that are shared with the DAC outputs.

Because the only circuit elements in the signal path are relatively low impedance analog switches, the performance of the Direct ADC Inputs is dominated by the ADC itself. Because there is no intentional filtering and there are no active gain stages in these paths, they have the highest bandwidth of any SmartFusion ADC inputs. Signals very close to the Nyquist frequency may be sampled by the ADC via these input channels.

Figure 9-1and Figure 9-2 show simplified schematics of the Direct ADC Inputs:

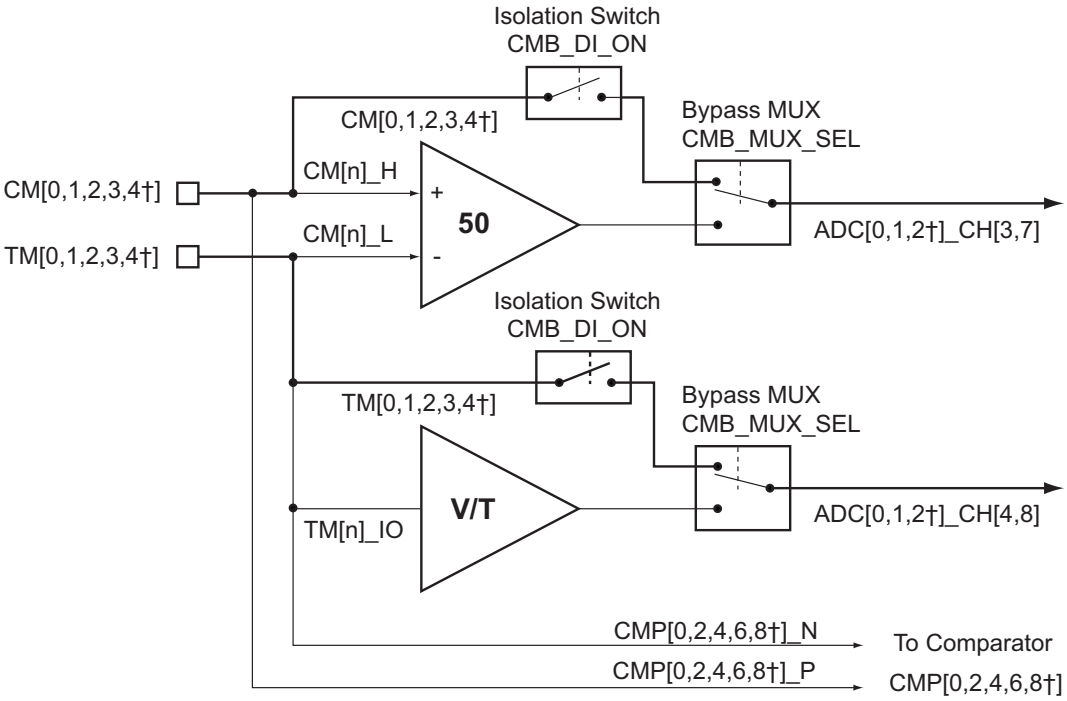


***Table 9-1* • Direct ADC Inputs Sharing Inputs with the Current and Temperature Monitors (direct paths highlighted)**

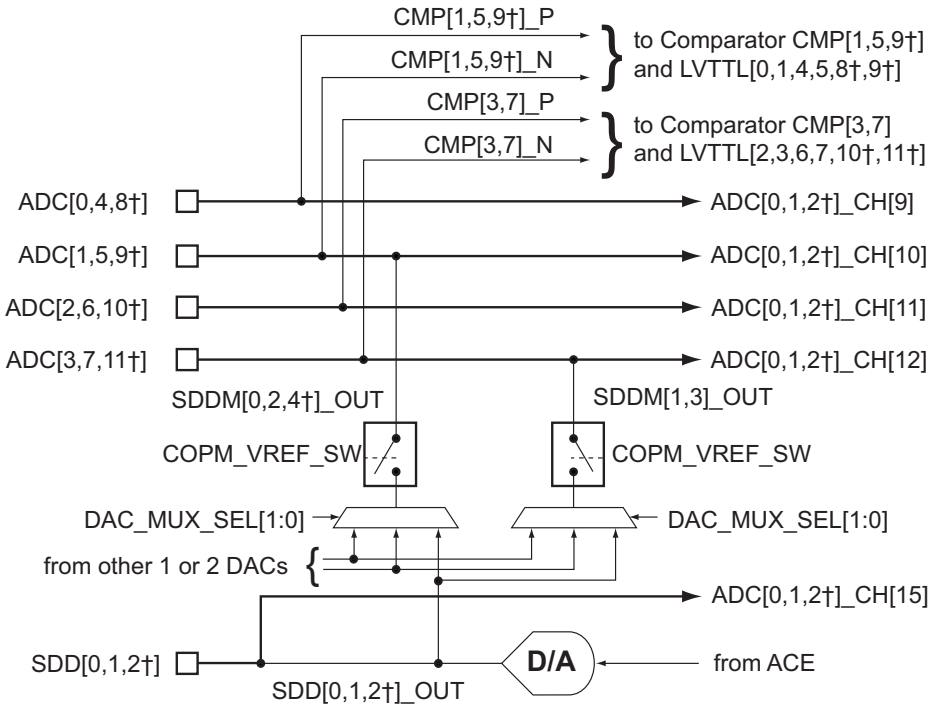


***Figure 9-2* • Direct ADC Inputs Sharing Inputs with Comparators, and with the Sigma-Delta DAC (direct paths highlighted)**

If you are using the Libero SmartDesign flow, many of the details of the Direct ADC Inputs will be transparent to you. When you select a Direct ADC Input, Smart Design automatically configures the various isolation and bypass switches, and creates the Sample Sequencer Engine microcode to control the ADC strobes that are required to use the Direct ADC Inputs that is suitable for most situations. However, it is possible to write customized microcode.

By default, factory calibration values are used to digitally compensate offset and gain errors using the Post Processing Engine (PPE) in the Analog Compute Engine (ACE).

The Direct ADC Inputs pads are shared with other functions. The temperature monitor input-output pad (TM[n]) can be used in conjunction with an external temperature sensing diode to sense temperature. When used as a Direct ADC Input, the signal on the TM[n] pad displaces the output of the temperature monitor into the ADC multiplexer. In other words, the TM[n] pad can be the input to the ADC, or the output of the temperature monitor can be the input to the ADC, but not both at the same time since they share the same ADC multiplexer channel. For more information on the temperature monitor, see the Temperature Monitor section of the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet.

The CM[n] and TM[n] input pads can alternatively be used for the current monitor. When used with the current monitor, the current monitor output uses the ADC input channel which could have been used for the CM[n] pad as a Direct ADC Input. Similar to the case with the temperature monitor, the ADC channel can be used for either the current monitor output, or the CM[n] pad Direct ADC Input, but not both.

If you are using the current monitor, it may be possible to also use the TM[n] input as a Direct ADC Input to measure the voltage on the low side of the current monitor sense resistor, but great care must be taken to observe the maximum voltage ratings of the TM[n] inputs in this case. Also be aware that the ADC input range might not include ground due to small offset voltages. Taking into consideration these limitations, this technique might be useful for measuring both the current and the voltage of a power supply that is within the Direct ADC Input range, for example a 1.5V or 1.8V supply, using a high-side current sense resistor and measuring the low side of the resistor using the direct input path from the TM[n] pad.

Note that direct ADC inputs have a much lower voltage range and voltage tolerance than the current monitor and temperature monitor, so care must be taken to restrict the input voltages to the CM[n] and

TM[n] pads when the direct ADC bypass(es) are selected and the high voltage isolation switch(es) are conducting. For more information on the current monitor, see the "Current Monitor" section on page 71.

Finally, there is also a comparator that shares the same two pads (TM[n] and CM[n]). Since the Direct ADC Input and the comparator have generally compatible voltage ranges, it may be feasible to use both on the same signals. For more about the comparators, see the "High-Speed Comparators" section on page 91 and the comparator section of the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet.

For every ADC there are four input pads designated as Direct ADC Inputs, (ADC[n]). These pads are also shared with comparators (which use two pads each). The reference (i.e. negative) input of the comparators can be driven by the Sigma-Delta DAC; generally this is done in a time-shared fashion, if used. See the "Direct ADC Inputs" section on page 103 where demultiplexing the DAC output to generate reference voltages for one or more comparators is described in more detail.

These four input pads also go to LVTTL digital input buffers. So, if the pads are not needed for Direct ADC Inputs or for use with the comparators, they may be used as standard LVTTL digital inputs. These inputs go directly to the FPGA fabric. Enables for each of the LVTTL input buffers come directly from the FPGA fabric.

To understand how to set the ADC input multiplexer to a selected channel, or how to configure and operate the ADC itself, see the "Analog-to-Digital Converter (ADC)" section on page 27.

Configuration of direct ADC pads as LVTTL inputs is done via flash cells. As a result, the configuration is not affected by MSS reset initiated system boot code rerun.

Table 9-2 relates each of the direct ADC inputs with the shared functions on the same input pads.

***Table 9-2* • Direct ADC Input Pad Designations**

ADC/ SDD	Signal Conditioning Block	Pad Name	Function Name	Shared Functions on Pad		
ADC0 SDD0	SCB0	CM0	ADC0_CH3	CMP0_P	CM0_H	
		TM0	ADC0_CH4	CMP0_N	CM0_L	TM0_IO
		ADC0	ADC0_CH9	CMP1_P	LVTTL0_IN	
		ADC1	ADC0_CH10	CMP1_N	LVTTL1_IN	SDDM0_OUT
	SCB1	CM1	ADC0_CH7	CMP2_P	CM1_H	
		TM1	ADC0_CH8	CMP2_N	CM1_L	TM1_IO
		ADC2	ADC0_CH11	CMP3_P	LVTTL2_IN	
		ADC3	ADC0_CH12	CMP3_N	LVTTL3_IN	SDDM1_OUT
	(N/A)	SDD0	ADC0_CH15	SDD0_OUT		
ADC1 SDD1	SCB2	CM2	ADC1_CH3	CMP4_P	CM2_H	
		TM2	ADC1_CH4	CMP4_N	CM2_L	TM2_IO
		ADC4	ADC1_CH9	CMP5_P	LVTTL4_IN	
		ADC5	ADC1_CH10	CMP5_N	LVTTL5_IN	SDDM2_OUT
	SCB3	CM3	ADC1_CH7	CMP6_P	CM3_H	
		TM3	ADC1_CH8	CMP6_N	CM3_L	TM3_IO
		ADC6	ADC1_CH11	CMP7_P	LVTTL6_IN	
		ADC7	ADC1_CH12	CMP7_N	LVTTL7_IN	SDDM3_OUT
	(N/A)	SDD1	ADC1_CH15	SDD1_OUT		

*Note: *Components indicated are only available in the A2F500.*

***Table 9-2* • Direct ADC Input Pad Designations**

ADC/ SDD	Signal Conditioning Block	Pad Name	Function Name	Shared Functions on Pad		
ADC2* SDD2*	SCB4	CM4	ADC2_CH3	CMP8_P	CM4_H	
		TM4	ADC2_CH4	CMP8_N	CM4_L	TM4_IO
		ADC8	ADC2_CH9	CMP9_P	LVTTL8_IN	
		ADC9	ADC2_CH10	CMP9_N	LVTTL9_IN	SDDM4_OUT
	(N/A)	SDD2	ADC2_CH15	SDD2_OUT		

*Note: *Components indicated are only available in the A2F500.*

Direct ADC Input Configuration

The Direct ADC Inputs can be dynamically configured via registers in the Analog Compute Engine (ACE). These registers can be controlled by the ACE itself, a Cortex-M3 program, by user logic (or soft processor) via the AHB bus master connected to the FPGA Fabric, or by the Peripheral DMA bus master.

The Direct ADC Input configuration options can be set statically using the Libero SmartDesign tool inside the provided Analog Compute Engine graphical configurator. They can also be changed at any time dynamically using any of the three advanced bus matrix (ABM) bus masters listed above, or by the ACE's Sample Sequencer Engine (SSE) or Post-Processor Engine (PPE). Besides the static configuration options, the Smart Design tool also automatically creates the SSE microcode required to control the ADC and the ADC's main multiplexer control registers and strobes.

To use the Direct ADC Inputs associated with the CM[n] and TM[n] pads, the appropriate isolation switch(es) must be conducting, and the bypass multiplexer put into the bypass mode. Special care must be taken when the isolation switch is conducting to observe the reduced maximum ratings of the inputs.

To use the Direct ADC Inputs associated with pads ADC[1, 3, 5, 7, 9, or 11] the COMP_VREF_SW should be in the off, or non-conducting state.

To use the Direct ADC Inputs associated with the SDD[n] output pads, the DAC should be in the current output mode. In the voltage output mode there will be an additional 10 Khoum load to ground. Also, the DAC duty factor should be zero, or else it will be driving a current out on this pin.

In most cases, it will also be desirable to put any unused alternate functions attached to the pads used for Direct ADC Inputs into their respective power-down states.

Table 9-3 summarizes the configuration settings related to the Direct ADC Inputs and functions sharing the same input pads which may need controlling in order to use the direct paths. Only the Direct ADC Input related bits are described here; bits in the same or adjacent registers that are used for other SCB configuration functions are not documented in this section. For information on how to select a particular ADC input channel or how to operate the ADC see the ADC section of this datasheet.

***Table 9-3* • Relative and Absolute Addresses for Direct ADC Input Related Configuration Control Bits**

SCB Byte and Bit Address (relative)	ABM Address (absolute 32-bit address)[1]	ACE, SSE, and PPE Address (absolute 8-bit Address)	Bit Name	Description
SCB_0 (0x40020204/0x81 through 0x40020230/0x8C)				
B6[0]	0x4002021C [0]	0x87 [0]	CUR_VOLB	SDD0 current/voltage select
B9[0]	0x40020228 [0]	0x8A [0]	CMB_MUX_SEL	CM0 bypass MUX
B9[1]	0x40020228 [1]	0x8A [1]	CMB_DI_ON	CM0 isolation switch
B9[2]	0x40020228 [2]	0x8A [2]	CM_EN	CM0 enable
B9 [4]	0x40020228 [4]	0x8A [4]	COMP_EN	CMP0 comparator enable
B10 [0]	0x4002022C [0]	0x8B [0]	TMB_MUX_SEL	TM0 bypass MUX
B10 [1]	0x4002022C [1]	0x8B [1]	TMB_DI_ON	TM0 isolation switch
B10 [2]	0x4002022C [2]	0x8B [2]	TM_EN	TM0 enable
B10[4]	0x4002022C [4]	0x8B [4]	COMP_EN	CMP1 comparator enable
B10[5]	0x4002022C [5]	0x8B [5]	COMP_VREF_SW	SCB0 SDD analog switch (to ADC1)
SCB_1 (0x40020234/0x8D through 0x40020260/0x98)				
B9[0]	0x40020258 [0]	0x96 [0]	CMB_MUX_SEL	CM1 bypass MUX
B9[1]	0x40020258 [1]	0x96 [1]	CMB_DI_ON	CM1 isolation switch
B9[2]	0x40020258 [2]	0x96 [2]	CM_EN	CM1 enable
B9 [4]	0x40020258 [4]	0x96 [4]	COMP_EN	CMP2 comparator enable
B10 [0]	0x4002025C [0]	0x97 [0]	TMB_MUX_SEL	TM1 bypass MUX
B10 [1]	0x4002025C [1]	0x97 [1]	TMB_DI_ON	TM1 isolation switch
B10 [2]	0x4002025C [2]	0x97 [2]	TM_EN	TM1 enable
B10[4]	0x4002025C [4]	0x97 [4]	COMP_EN	CMP3 comparator enable
B10[5]	0x4002025C [5]	0x97 [5]	COMP_VREF_SW	SCB1 SDD analog switch (to ADC3)

Notes:

1. *Only the least-significant byte of each 32-bit data word is utilized.*
2. *Components indicated are only available in the A2F500.*

***Table 9-3* • Relative and Absolute Addresses for Direct ADC Input Related Configuration Control Bits**

SCB Byte and Bit Address (relative)	ABM Address (absolute 32-bit address)[1]	ACE, SSE, and PPE Address (absolute 8-bit Address)	Bit Name	Description
SCB_2 (0x40020264/0x99 through 0x40020290/0xA4)				
B6[0]	0x4002027C [0]	0x9F [0]	CUR_VOLB	SDD1 current/voltage Sel.
B9[0]	0x40020288 [0]	0x A2 [0]	CMB_MUX_SEL	CM2 bypass MUX
B9[1]	0x40020288 [1]	0x A2 [1]	CMB_DI_ON	CM2 isolation switch
B9[2]	0x40020288 [2]	0x A2 [2]	CM_EN	CM2 enable
B9 [4]	0x40020288 [4]	0xA2 [4]	COMP_EN	CMP4 comparator enable
B10 [0]	0x4002028C [0]	0xA3 [0]	TMB_DI_ON	TM2 bypass MUX
B10 [1]	0x4002028C [1]	0xA3 [1]	TMB_DI_ON	TM2 isolation switch
B10 [2]	0x4002028C [2]	0xA3 [2]	TM_EN	TM2 enable
B10[4]	0x4002028C [4]	0xA3 [4]	COMP_EN	CMP5 comparator enable
B10[5]	0x4002028C [5]	0xA3 [5]	COMP_VREF_SW	SCB2 SDD analog switch (to ADC5)
SCB_3 (0x40020294/0xA5 through 0x400202C0/0xB0)				
B9[0]	0x400202B8 [0]	0x AE [0]	CMB_MUX_SEL	CM3 bypass MUX
B9[1]	0x400202B8 [1]	0x AE [1]	CMB_DI_ON	CM3 isolation switch
B9[2]	0x400202B8 [2]	0x AE [2]	CM_EN	CM3 enable
B9 [4]	0x400202B8 [4]	0xAE [4]	COMP_EN	CMP6 comparator enable
B10 [0]	0x400202BC [0]	0xAF [0]	TMB_DI_ON	TM3 bypass MUX
B10 [1]	0x400202BC [1]	0xAF [1]	TMB_DI_ON	TM3 isolation switch
B10 [2]	0x400202BC [2]	0xAF [2]	TM_EN	TM3 enable
B10[4]	0x400202BC [4]	0xAF [4]	COMP_EN	CMP7 comparator enable
B10[5]	0x400202BC [5]	0xAF [5]	COMP_VREF_SW	SCB3 SDD analog switch (to ADC7)
SCB_4 (0x400202C4/0xB1 through 0x400202F0/0xBC)[2]				
B6[0]	0x400202DC [0]	0xB7 [0]	CUR_VOLB	SDD2 current/voltage select
B9[0]	0x400202E8 [0]	0x BA [0]	CMB_MUX_SEL	CM4 bypass MUX
B9[1]	0x400202E8 [1]	0x BA [1]	CMB_DI_ON	CM4 isolation switch
B9[2]	0x400202E8 [2]	0x BA [2]	CM_EN	CM4 enable
B9 [4]	0x400202E8 [4]	0xBA [4]	COMP_EN	CMP8 comparator enable
B10 [0]	0x400202EC [0]	0xBB [0]	TMB_DI_O	TM4 bypass MUX
B10 [1]	0x400202EC [1]	0xBB [1]	TMB_DI_ON	TM4 isolation switch
B10 [2]	0x400202EC [2]	0xBB [2]	TM_EN	TM4 enable
B10[4]	0x400202EC [4]	0xBB [4]	COMP_EN	CMP9 comparator enable
B10[5]	0x400202EC [5]	0xBB [5]	COMP_VREF_SW	SCB4 SDD analog switch (to ADC9)

Notes:

1. *Only the least-significant byte of each 32-bit data word is utilized.*
2. *Components indicated are only available in the A2F500.*

***Table 9-4* • Functional Description of Direct ADC Input Related Configuration Control Bits**

Bit Name	Description	R/W	Reset Value	Function
CUR_VOLB	SDD Current Mode	R/W	0b0	0 = Voltage mode 1 = Current mode
COMP_EN	CMP Enable	R/W	0b0	0 = Comparator Power Save mode 1 = Comparator Operational mode
CM_EN	CM Enable	R/W	0b0	0 = Power Save mode 1 = Operational mode
CMB_MUX_SEL	CM Bypass MUX	R/W	0b0	0 = ADC input MUX driven from current monitor 1 = ADC input MUX driven directly from CM[n] pad
CMB_DI_ON	CM Isolation Switch	R/W	0b0	0 = Isolation switch open (isolating high voltages) 1= Isolation switch closed (direct ADC input)
TM_EN	TM Enable	R/W	0b0	0 = Power Save mode 1 = Operational mode
TMB_MUX_SEL	TM Bypass MUX	R/W	0b0	0 = ADC input MUX driven from temperature monitor 1 = ADC input MUX driven directly from TM[n] pad
TMB_DI_ON	TM Isolation Switch	R/W	0b0	0 =Isolation switch open (isolating high voltages) 1 = Isolation switch closed (direct ADC input)
COMP_VREF_SW	SDD Analog Switch	R/W	0b0	0 = SDD Vref Switch off (high impedance) 1 = SDD Vref Switch on (conducting)

Related Information

SmartFusion Customizable System-on-Chip (cSoC) datasheet
http://www.microsemi.com/soc/documents/SmartFusion_DS.pdf

10 – ACE Interrupts

A total of eighty-six ACE interrupts are routed to the Cortex-M3 processor via an interrupt logic control/ aggregation block. These interrupts map to the top eighty-six, [149:64], of the total implemented Cortex-M3 interrupts in the SmartFusion device. In addition, thirty-two of these interrupts are routed to the FPGA. The lower ten, [9:0], are multiplexed between PPE status interrupts and aggregated PPE flag interrupts, while the other twenty-two, [31:10], are the aggregated PPE flag interrupts

These interrupts provide visibility into different stages of signal processing in the ACE block. ACE interrupt handler functions that are part of the Microsemi firmware driver APIs can be used in the applications running on the Cortex-M3 processor for real-time data access and decision making. Refer to the *SmartFusion MSS ACE Driver User's Guide* for more details on the APIs.

The subset of thirty-two interrupts routed to the FPGA allows applications running on the APB Master in FPGA Fabric access to real time ACE block information for decision making. The following sections describe these interrupts in detail along with the associated registers.

ACE Interrupts Routed to the Cortex-M3 Processor

The ACE interrupts routed to the Cortex-M3 processor are summarized in Figure 10-1 and detailed in the sections below.

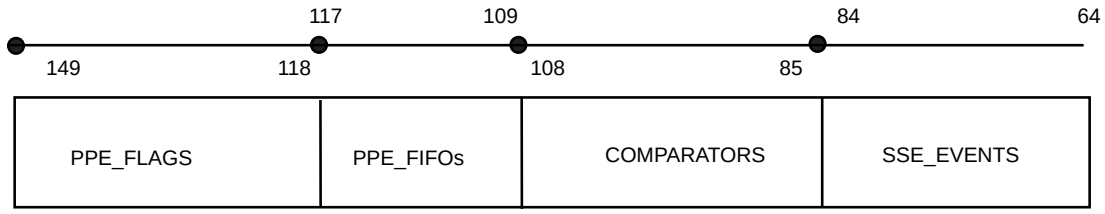


***Figure 10-1* • Breakdown of ACE Interrupts to Cortex-M3 Processor**

Comparator Interrupt Events

The comparator-related events generate 24 flags that are routed to the Cortex-M3 processor. These correspond to the 12 rising and 12 falling events on the comparators in the analog quads. Given that a maximum of 10 comparators exist in the SmartFusion devices, only the lowest 10 bits of the possible 12 rising and lowest 10 bits of the possible falling event flags are used. The 3 registers corresponding to these interrupt events and their bit fields are described in Table 10-1.

***Table 10-1* • Comparator Interrupt Event Registers**

Register Name	Address	Access	Description
ACE_COMP_IRQ_EN	0x4002120C	R/W	Enable/mask register for ACE_COMP_IRQ bit interrupt contributions
ACE_COMP_IRQ	0x40021210	R	Analog comparator related interrupts
ACE_COMP_IRQ_CLR	0x40021214	W	Control register to clear ACE_COMP_IRQ bits

Comparator IRQ Enable Register (ACE_COMP_IRQ_EN)

Table 10-2 • Comparator IRQ Enable Register Description

Bit Number	Name	Access	Description
[31:24]	Reserved	N/A	Reserved
[23:0]	IRQ_EN	R/W	These bits are logically ANDed with the bits in the COMP_IRQ register to create interrupts that are brought out as Cortex-M3 interrupts [108:85]. If any of the bits in this register is 1, the corresponding bit in the COMP_IRQ register is enabled; otherwise it is disabled/masked.

Comparator IRQ Register (ACE_COMP_IRQ)

Table 10-3 • Comparator IRQ Register Description

Bit Number	Name	Access	Description
[31:22]	Reserved		Reserved
[21:12]	COMP_RISE	R	Comparator rising edge events: these IRQs indicate rising edge events on any or all of the comparator signals from the analog quads. These bits are ANDed with ACE_COMP_IRQ_EN [21:12] to create the Cortex-M3 interrupts [94:85]. The top two bits are reserved as the maximum comparator count is 10.
[11:10]	Reserved	R	Reserved
[9:0]	COMP_FALL	R	Comparator falling edge events: these IRQs indicate falling edge events on any or all of the comparator signals from the analog quads. These bits are ANDed with ACE_COMP_IRQ_EN[9:0] to create the Cortex-M3 interrupts [106:97]. The top two bits are reserved as the maximum comparator count is 10.

Comparator IRQ Clear Register (ACE_COMP_IRQ_CLR)

Table 10-4 • Comparator IRQ Clear Register

Bit Number	Name	Access	Description
[31:24]	Reserved		Reserved
[23:0]	IRQ_CLR	W	These bits are used to clear corresponding bits in the ACE_COMP_IRQ register. Writing 1 to any of the bits will clear the corresponding bits in the ACE_COMP_IRQ register. Writing 0 to any of the bits will have no effect. In the event that writing a 1 to these clear bits coincides with a set event in the COMP_IRQ register, the set event has higher priority.

SSE Related Interrupt Events

The SSE related events generate 21 flags that are routed to the Cortex-M3 processor. These correspond to the three ADC blocks and the corresponding program counters. These include 3 ADC conversion done events, 3 ADC calibration start events, 3 ADC calibration done events, and 12 general purpose SSE events. These 12 general purpose SSE events can be used to create user-defined interrupt events based on ADC program counter status.

The 3 registers corresponding to these interrupt events and their bit fields are described in Table 10-5.

***Table 10-5* • SSE Related Interrupt Event Registers**

Register Name	Address	R/W	Reset Value	Description
ACE_SSE_IRQ_EN	0x40021200	R/W	0	Enable/mask register for ACE_SSE_IRQ bit interrupt contributions
ACE_SSE_IRQ	0x40021204	R	0	ACE SSE related interrupts
ACE_SSE_IRQ_CLR	0x40021208	W	0	Control register to clear ACE_SSE_IRQ bits

SSE IRQ Enable Register (ACE_SSE_IRQ_EN)

***Table 10-6* • SSE IRQ Enable Register Description**

Bit Number	Name	Access Mode	Reset Value	Description
[31:21]	Reserved			Reserved
[20:0]	IRQ_EN	R/W	0	These bits are logically ANDed with the bits in the ACE_SSE_IRQ register to create interrupts that are brought out as Cortex-M3 interrupts [84:64]. If any of the bits in this register is 1, the corresponding bit in the ACE_SSE_IRQ register is enabled; otherwise it is disabled/masked.

SSE IRQ Register (ACE_SSE_IRQ)

***Table 10-7* • SSE IRQ Register**

Bit Number	Name	Description
[31:21]	Reserved	Reserved
20	ADC2_CAL_RISE	ADC2_CALIBRATE signal rising edge; this IRQ indicates a start of calibration event for ADC2. This bit is ANDed with ACE_SSE_IRQ_EN[20] to create Cortex-M3 Interrupt [84].
19	ADC1_CAL_RISE	ADC1_CALIBRATE signal rising edge; this IRQ indicates a start of calibration event for ADC1. This bit is ANDed with ACE_SSE_IRQ_EN[19] to create Cortex-M3 interrupt [83].
18	ADC0_CAL_RISE	ADC0_CALIBRATE signal rising edge; this IRQ indicates a start of calibration event for ADC0. This bit is ANDed with ACE_SSE_IRQ_EN[18] to create Cortex-M3 interrupt [82].
17	ADC2_CAL_FALL	ADC2_CALIBRATE signal falling edge; this IRQ indicates an end of calibration event for ADC2. This bit is ANDed with ACE_SSE_IRQ_EN[17] to create Cortex-M3 interrupt [81].
16	ADC1_CAL_FALL	ADC1_CALIBRATE signal falling edge; this IRQ indicates an end of calibration event for ADC1. This bit is ANDed with ACE_SSE_IRQ_EN[16] to create Cortex-M3 interrupt [80].

***Table 10-7* • SSE IRQ Register (continued)**

Bit Number	Name	Description
15	ADC0_CAL_FALL	ADC0_CALIBRATE signal falling edge; this IRQ indicates an end of calibration event for ADC0. This bit is ANDed with ACE_SSE_IRQ_EN[15] to create Cortex-M3 interrupt [79].
14	ADC2_DV_RISE	ADC2_DATAVALID signal rising edge; this IRQ indicates an end of ADC2 conversion event. This bit is ANDed with ACE_SSE_IRQ_EN [14] to create Cortex-M3 interrupt [78].
13	ADC1_DV_RISE	ADC1_DATAVALID signal rising edge; this IRQ indicates an end of ADC1 conversion event. This bit is ANDed with ACE_SSE_IRQ_EN [13] to create Cortex-M3 interrupt [77].
12	ADC0_DV_RISE	ADC0_DATAVALID signal rising edge; this IRQ indicates an end of ADC0 conversion event. This bit is ANDed with ACE_SSE_IRQ_EN [12] to create Cortex-M3 interrupt [76].
11:8	PC2_FLAGS	Program Counter2 flags; these four general purpose flags can be used by the SSE to create interrupt events based on application requirement. These are not directly associated with any of ADC2 inherent events listed in bits [20:12]. They are ANDed with ACE_SSE_IRQ_EN[11:8] to create Cortex-M3 interrupts [75:72].
7:4	PC1_FLAGS	Program Counter1 flags; these four general purpose flags can be used by the SSE to create interrupt events SSE to create interrupt events based on application requirement. These are not directly associated with any of ADC1 inherent events listed in bits [20:12]. They are ANDed with ACE_SSE_IRQ_EN[7:4] to create Cortex-M3 Interrupts [71:68].
3:0	PC0_FLAGS	Program Counter0 flags; these 4 general purpose flags can be used by the SSE to create interrupt events based on application requirement. These are not directly associated with any of ADC0 inherent events listed in bits [20:12]. They are ANDed with ACE_SSE_IRQ_EN[3:0] to create Cortex-M3 interrupts [67:64].

SSE IRQ Clear Register (ACE_SSE_IRQ_CLR)

***Table 10-8* • SSE IRQ Clear Register**

Bit Number	Name	R/W	Reset Value	Description
[31:21]	Reserved			Reserved
[20:0]	IRQ_CLR	W	0	These write-only bits are used to clear corresponding bits in the ACE_SSE_IRQ register. Writing a 1 to any of the bits clears the corresponding bits in the ACE_SSE_IRQ register, while writing a 0 to any of the bits has no effect. In the event that writing a 1 to these clear bits coincides with a set event in the ACE_SSE_IRQ register, the set event has higher priority.

Post Processing Engine (PPE) Interrupts

ADC FIFO Status Interrupt Events

The PPE FIFO related events generate nine flags that are routed to the Cortex-M3 processor. Six among these nine correspond to ADC result FIFO almost full/full events. The other three interrupts correspond to ADC result FIFO empty flags. The three registers corresponding to these interrupt events and their bit fields are described in Table 10-9.

***Table 10-9* • ADC Result FIFO Empty Flag Interrupt Event Registers**

Address	Register Name	Access Mode	Description
0x40021218	ACE_PPE_FIFO_IRQ_EN	R/W	Enable/mask register for ACE_PPE_FIFO_IRQ bit interrupt contributions
0x4002121C	ACE_PPE_FIFO_IRQ	R	PPE ADC FIFO related events
0x40021220	ACE_PPE_FIFO_IRQ_CLR	W	Control register to clear ACE_PPE_FIFO_IRQ

PPE FIFO IRQ Enable Register (ACE_PPE_FIFO_IRQ_EN)

***Table 10-10* • PPE FIFO IRQ Enable Register Description**

Bit Number	Name	Access Mode	Description
[31:9]	Reserved		Reserved
[8:0]	IRQ_EN	R/W	These bits are logically ANDed with the bits in the ACE_PPE_FIFO_IRQ register to create interrupts that are brought out as Cortex-M3 interrupts [117:109]. If any of the bits in this register is 1, the corresponding bit in the ACE_PPE_FIFO_IRQ register is enabled; otherwise it is disabled/masked.

PPE FIFO IRQ Register (ACE_ PPE_FIFO _IRQ)

***Table 10-11* • PPE FIFO IRQ Register Description**

Bit Number	Name	R/W	Description
[31:9]	Reserved		Reserved
8	EMPTY2_FALL	R	ADC2_FIFO_EMPTY signal falling edge; this IRQ indicates that the ADC2 result FIFO has just become non-empty. This bit is ANDed with ACE_PPE_FIFO_IRQ_EN[8] to create Cortex-M3 interrupt [117].
7	AFULL2_RISE	R	ADC2_FIFO_AFULL signal rising edge; this IRQ indicates that the ADC2 result FIFO has just become almost full; i.e., there are now three words of result data the PPE can process. This bit is logically ANDed with ACE_PPE_FIFO_IRQ_EN[7] to create Cortex-M3 interrupt [116].
6	FULL2_RISE	R	ADC2_FIFO_FULL signal rising edge; this IRQ indicates that the ADC2 result FIFO has just become full; i.e., there are now four words of result data the PPE can process. This bit is ANDed with ACE_PPE_FIFO_IRQ_EN[6] to create Cortex-M3 interrupt [115].

***Table 10-11* • PPE FIFO IRQ Register Description**

Bit Number	Name	R/W	Description
5	EMPTY1_FALL	R	ADC1_FIFO_EMPTY signal falling edge: this IRQ indicates that the ADC1 result FIFO has just become non-empty. This bit is ANDed with ACE_PPE_FIFO_IRQ_EN[5] to create Cortex-M3 interrupt [114].
4	AFULL1_RISE	R	ADC1_FIFO_AFULL signal rising edge; this IRQ indicates that the ADC1 result FIFO has just become almost full; i.e., there are now three words of result data the PPE can process. This bit is ANDed with ACE_PPE_FIFO_IRQ_EN[4] to create Cortex-M3 Interrupt [113].
3	FULL1_RISE	R	ADC1_FIFO_FULL signal rising edge; this IRQ indicates that the ADC1 result FIFO has just become full; i.e., there are now four words of result data the PPE can process. This bit is ANDed with ACE_PPE_FIFO_IRQ_EN[3] to create Cortex-M3 interrupt [112].
2	EMPTY0_FALL	R	ADC0_FIFO_EMPTY signal falling edge; this IRQ indicates that the ADC0 result FIFO has just become non-empty. This bit is ANDed with ACE_PPE_FIFO_IRQ_EN[2] to create Cortex-M3 interrupt [111].
1	AFULL0_RISE	R	ADC0_FIFO_AFULL signal rising edge; this IRQ indicates that the ADC0 result FIFO has just become almost full; i.e., there are now three words of result data the PPE can process. This bit is ANDed with ACE_PPE_FIFO_IRQ_EN[1] to create Cortex-M3 interrupt [110].
0	FULL0_RISE	R	ADC0_FIFO_FULL signal rising edge; this IRQ indicates that the ADC0 result FIFO has just become full; i.e., there are now four words of result data the PPE can process. This bit is ANDed with ACE_PPE_FIFO_IRQ_EN[0] to create Cortex-M3 interrupt [109].

PPE FIFO IRQ Clear Register (ACE_ PPE_FIFO _IRQ_CLR)

***Table 10-12* • PPE FIFO IRQ Register Description**

Bit Number	Name	Access	Description
[31:9]	Reserved		Reserved
[8:0]	IRQ_CLR	W	These write-only bits are used to clear corresponding bits in the ACE_PPE_FIFO_IRQ register. Writing a 1 to any of the bits clears the corresponding bits in the ACE_PPE_FIFO_IRQ register, while writing a 0 to any of the bits has no effect. In the event that writing a 1 to these clear bits coincides with a set event in the ACE_PPE_FIFO_IRQ register, the set event has higher priority.

PPE Threshold and State Filter Flag Interrupt Events

There are four banks of PPE hysteresis and threshold flags that are arranged in 32-bit registers. There is one bank of state filter flags arranged into a 32-bit register. These 128 threshold flags and 32 state filter flags are aggregated into 32 resultant interrupts. Effectively these are logically ORed, as shown in Figure 10-13.

These four hysteresis and threshold flag registers are populated with application-specific flags generated by the user during ACE configuration in SmartDesign MSS configurator. These registers are filled from LSB to MSB starting with PPE_FLAGS0 register to PPE_FLAGS3. Similarly, the state filter flag register gets filled from LSB to MSB.

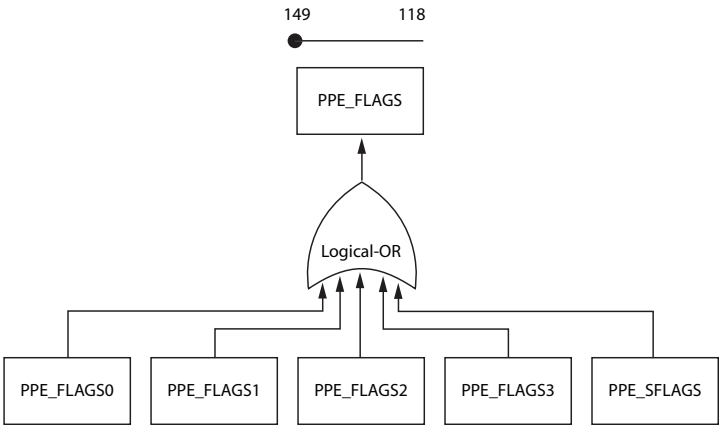


***Table 10-13* • PPE Flag Aggregation Overview**

These aggregated interrupts are routed to the Cortex-M3 processor as ACE interrupts [149:118] and to fabric as ACEFLAGS[31:0].

PPE Threshold Flag Bank Zero (PPE_FLAG0)

These flags are generated by PPE based on user configuration in MSS configurator. The three registers corresponding to these interrupt events and their bit fields are described in Table 10-14.

***Table 10-14* • PPE Threshold Flag Bank Zero Description**

Register Name	Address	R/W	Description
ACE_PPE_FLAGS0_IRQ_EN	0x40021224	R/W	Enable/mask register for ACE_PPE_FLAGS0_IRQ bit interrupt contributions
ACE_PPE_FLAGS0_IRQ	0x40021228	R	ACE PPE FLAGS0 threshold related events
ACE_PPE_FLAGS0_IRQ_CLR	0x4002122C	W	Control register to clear ACE_PPE_FLAGS0_IRQ bits

PPE FLAGS0 IRQ Enable Register (ACE_PPE_ FLAGS0 _IRQ_EN)

***Table 10-15* • PPE FLAGS0 IRQ Enable Register Description**

Bit Number	Name	Access	Description
[31:0]	IRQ_EN	R/W	These bits are logically ANDed with the bits in the ACE_PPE_FLAGS0_IRQ register to create interrupts that are aggregated with the other PPE flags and brought out as Cortex-M3 interrupts [149:118]. If any of the bits in this register is 1, the corresponding bit in the ACE_PPE_FLAGS0_IRQ register is enabled; otherwise the corresponding bit in the ACE_PPE_FLAGS0_IRQ register is disabled/masked.

PPE FLAGS0 IRQ Register (ACE_ PPE_ FLAGS0 _IRQ)

***Table 10-16* • PPE FLAGS0 IRQ Register Description**

Bit Number	Name	R/W	Reset Value	Description
[31:0]	FLAGS0_RISE	R	0	PPE FLAGS0 rising edges; these IRQ signals indicate that one or more of the PPE_FLAGS0 threshold flags has transitioned from 0 to 1, indicating that a signal is either greater than or less than the threshold for that particular signal. These bits are ANDed with ACE_PPE_FLAGS0_IRQ_EN[31:0] and aggregated with other PPE flags to create ACE interrupts [149:118].

PPE FLAGS0 IRQ Clear Register (ACE_ PPE_ FLAGS0 _IRQ_CLR)

***Table 10-17* • PPE FLAGS0 IRQ Clear Register**

Bit Number	Name	R/W	Description
[31:0]	IRQ_CLR	W	These write-only bits are used to clear corresponding bits in the ACE_PPE_FLAGS0_IRQ register. Writing a 1 to any of the bits clears the corresponding bits in the ACE_PPE_FLAGS0_IRQ register, while writing a 0 to any of the bits has no effect. Writing a 1 to these clear bits coincides with a set event in the ACE_PPE_FLAGS0_IRQ register, the set event has higher priority.

PPE Threshold Flag Bank One (PPE_FLAG1)

These flags are generated by PPE based on user configuration in MSS configurator. The three registers corresponding to these interrupt events and their bit fields are described in Table 10-18.

***Table 10-18* • PPE Threshold Flag Bank One Description**

Register Name	Address	Access	Description
ACE_PPE_FLAGS1_IRQ_EN	0x40021230	R/W	Enable/mask register for ACE_PPE FLAGS1_IRQ bit interrupt contributions
ACE_PPE_FLAGS1_IRQ	0x40021234	R	ACE PPE FLAGS1 threshold related events
ACE_PPE_FLAGS1_IRQ_CLR	0x40021238	W	Control register to clear the bits of ACE_PPE_FLAGS1_IRQ

PPE FLAGS1 IRQ Enable Register (ACE_PPE_ FLAGS1 _IRQ_EN)

***Table 10-19* • PPE FLAGS1 IRQ Enable Register Description**

Bit Number	Name	R/W	Description
[31:0]	IRQ_EN	R/W	These bits are ANDed with the bits in the ACE_PPE_FLAGS1_IRQ register to create interrupts that are aggregated with the other PPE flags and brought out as ACE interrupts [149:118]. If any of the bits in this register is logic 1, the corresponding bit in the ACE_PPE_FLAGS1_IRQ register is enabled; otherwise the corresponding bit in the ACE_PPE_FLAGS1_IRQ register is disabled or masked.

PPE FLAGS1 IRQ Register (ACE_ PPE_ FLAGS1 _IRQ)

***Table 10-20* • PPE FLAGS1 IRQ Register Description**

Bit Number	Name	Description
[31:0]	FLAGS1_RISE	PPE FLAGS1 rising edges; these IRQ signals indicate that one or more of the PPE_FLAGS1 threshold flags have transitioned from logic 0 to logic 1, indicating that a signal is either greater than or less than the desired threshold for that particular signal. These bits are logically ANDed with ACE_PPE_FLAGS1_IRQ_EN[31:0] and aggregated with other PPE flags to create ACE interrupts [149:118].

PPE FLAGS1 IRQ Clear Register (ACE_ PPE_ FLAGS1 _IRQ_CLR)

***Table 10-21* • PPE FLAGS1 IRQ Clear Register Description**

Bit Number	Name	R/W	Description
[31:0]	IRQ_CLR	W	These write-only bits are used to clear corresponding bits in the ACE_PPE_FLAGS1_IRQ register. Writing a 1 to any of the bits clears the corresponding bits in the ACE_PPE_FLAGS1_IRQ register. Writing 0 to any of the bits has no effect. If writing a 1 to these clear bits coincides with a set event in the ACE_PPE_FLAGS1_IRQ register, the set event has higher priority.

PPE Threshold Flag Bank Two (PPE_FLAG2)

These flags are generated by PPE based on user configuration in MSS configurator. The three registers corresponding to these interrupt events and their bit fields are described in Table 10-22.

***Table 10-22* • PPE Threshold Bank Two Description**

Register Name	Address	R/W	Description
ACE_PPE_FLAGS2_IRQ_EN	0x4002123C	R/W	Enable/mask for ACE_PPE_FLAGS2_IRQ bit interrupt contributions
ACE_PPE_FLAGS2_IRQ	0x40021240	R	ACE PPE FLAGS2 threshold related events
ACE_PPE_FLAGS2_IRQ_CLR	0x40021244	W	Control register to clear ACE_PPE_FLAGS2_IRQ bits

PPE FLAGS2 IRQ Enable Register (ACE_PPE_ FLAGS2 _IRQ_EN)

***Table 10-23* • PPE FLAGS2 IRQ Enable Register**

Bit Number	Name	R/W	Description
[31:0]	IRQ_EN	R/W	These bits are ANDed with the bits in the ACE_PPE_FLAGS2_IRQ register to create interrupts that are aggregated with the other PPE flags and eventually brought out as Cortex-M3 interrupts [149:118]. If any of the bits in this register is logic 1, the corresponding bit in the ACE_PPE_FLAGS2_IRQ register is enabled, otherwise the corresponding bit in the ACE_PPE_FLAGS2_IRQ register is disabled or masked.

PPE FLAGS2 IRQ Register (ACE_ PPE_ FLAGS2 _IRQ)

***Table 10-24* • PPE FLAGS2 IRQ Register Description**

Bit Number	Name	R/W	Description
[31:0]	FLAGS2_RISE	R	PPE FLAGS2 rising edges; these IRQ signals indicate that one or more of the PPE_FLAGS2 threshold flags have transitioned from logic 0 to logic 1, indicating that a signal is either greater than or less than the threshold for that particular signal. These bits are ANDed with ACE_PPE_FLAGS2_IRQ_EN[31:0] and aggregated with other PPE flags to create Cortex-M3 interrupts [149:118].

PPE FLAGS2 IRQ Clear Register (ACE_ PPE_ FLAGS2 _IRQ_CLR)

***Table 10-25* • PPE FLAGS2 IRQ Clear Register Description**

Bit Number	Name	R/W	Description
[31:0]	IRQ_CLR	W	These write-only bits are used to clear corresponding bits in the ACE_PPE_FLAGS2_IRQ register. Writing 1 to any of the bits clears the corresponding bits in the ACE_PPE_FLAGS2_IRQ register. Writing 0 to any of the bits has no effect. If 1 is written to these clear bits and coincides with a set event in the ACE_PPE_FLAGS2_IRQ register, the set event has higher priority.

PPE Threshold Flag Bank Three (PPE_FLAG3)

These flags are generated by PPE based on user configuration in MSS configurator. The three registers corresponding to these interrupt events and their bit fields are described in Table 10-26. PPE flag aggregation details are shown in Figure 10-2 on page 121.

***Table 10-26* • PPE Threshold Flag Bank Three Description**

Register Name	Address	R/W	Description
ACE_PPE_FLAGS3_IRQ_EN	0x40021248	R/W	Enable/mask register for ACE_PPE_FLAGS3_IRQ bit interrupt contributions
ACE_PPE_FLAGS3_IRQ	0x4002124C	R	ACE PPE FLAGS3 threshold related events
ACE_PPE_FLAGS3_IRQ_CLR	0x40021250	W	Control register to clear ACE_PPE_FLAGS2_IRQ bits

PPE FLAGS3 IRQ Enable Register (ACE_PPE_ FLAGS3 _IRQ_EN)

***Table 10-27* • PPE FLAGS3 IRQ Enable Register**

Bit Number	Name	R/W	Description
[31:0]	IRQ_EN	R/W	These bits are ANDed with the bits in the ACE_PPE_FLAGS3_IRQ register to create interrupts that are aggregated with the other PPE flags and eventually brought out as Cortex-M3 interrupts [149:118]. If any of the bits in this register is logic 1, the corresponding bit in the ACE_PPE_FLAGS3_IRQ register is enabled; otherwise the bit in the ACE_PPE_FLAGS3_IRQ register is disabled/masked.

PPE FLAGS3 IRQ Register (ACE_ PPE_ FLAGS3 _IRQ)

***Table 10-28* • PPE FLAGS3 IRQ Register**

Bit Number	Name	R/W	Description
[31:0]	FLAGS3_RISE	R	PPE FLAGS3 rising edges; these IRQ signals indicate that one or more of the PPE_FLAGS3 threshold flags have transitioned from logic 0 to logic 1, indicating that a signal is either greater than or less than the desired threshold for that particular signal. These bits are logically ANDed with ACE_PPE_FLAGS3_IRQ_EN[31:0] and aggregated with other PPE flags to create Cortex-M3interrupts [149:118].

PPE FLAGS3 IRQ Clear Register (ACE_ PPE_ FLAGS3 _IRQ_CLR)

***Table 10-29* • PPE FLAGS3 IRQ Clear Register**

Bit Number	Name	R/W	Reset Value	Description
[31:0]	IRQ_CLR	W	0	These write-only bits are used to clear corresponding bits in the ACE_PPE_FLAGS3_IRQ register. Writing a 1 to any of the bits clears the corresponding bits in the ACE_PPE_FLAGS3_IRQ register, while writing a 0 to any of the bits has no effect. In the event that writing a 1 to these clear bits coincides with a set event in the ACE_PPE_FLAGS3_IRQ register, the set event has higher priority.

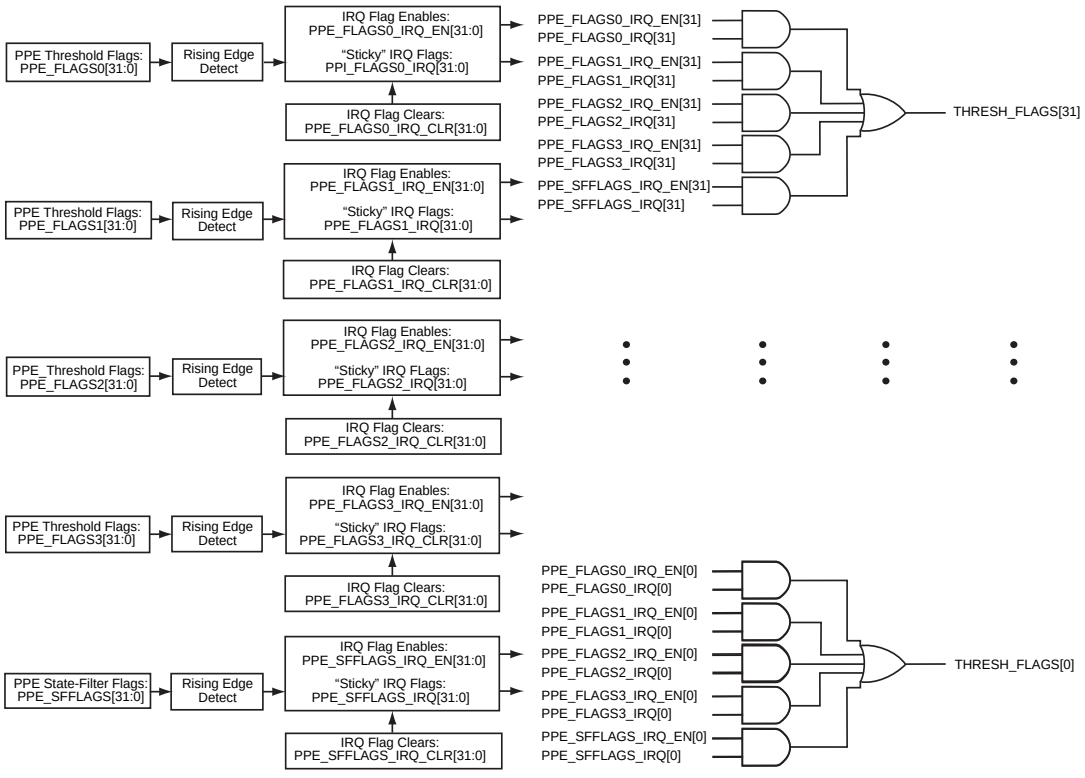


***Figure 10-2* • PPE Flag Aggregation Details**

ACE Interrupts/Flags Routed to FPGA Fabric

32 FPGA flags are available to the FPGA master. The top 22 bits [31:10] are the aggregated PPE threshold flags, while the lower 10 bits [9:0] are multiplexed between threshold flags PPE status and FIFO flags, as shown in Figure 10-3.

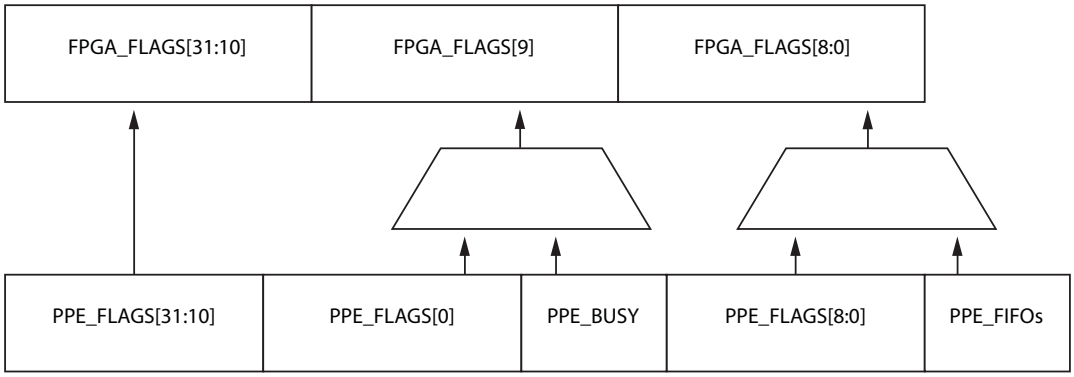


***Figure 10-3* • ACE Interrupts to FPGA**

The 32 bit register ACE_FPGA_FLAGS_SEL register enables selection of the lower 10 bits between these sources. This is selectable in the MSS Configurator. The register is described in Table 10-30.

***Table 10-30* • ACE_FPGA_FLAGS_SEL Register Description**

Register Name	Address	R/W	Description
ACE_FPGA_FLAGS_SEL	0x40021260	R/W	Control register to select alternative internal ACE signals to monitor via the FPGA_FLAGS[9:0] to the FPGA fabric

ACE FPGA FLAG Selection Register (ACE_FPGA_FLAGS_SEL)

***Table 10-31* • ACE FPGA FLAG Selection Register Description**

Bit Number	Name	R/W	Description
[31:10]	Reserved		Reserved
9	PPE_BUSY_SEL	R	1 – Select PPE_BUSY internal signal to be connected to the FPGA_FLAGS[9] output. 0 – Select THRESH_FLAGS[9] internal signal to be connected to the FPGA_FLAGS[9] output.
8	FIFO_FULL2_SEL	R	1 – Select ADC2_FIFO_FULL internal signal to be connected to the FPGA_FLAGS[8] output. 0 – Select THRESH_FLAGS[8] internal signal to be connected to the FPGA_FLAGS[8] output.
7	FIFO_FULL1_SEL	R	1 – Select ADC1_FIFO_FULL internal signal to be connected to the FPGA_FLAGS[7] output. 0 – Select THRESH_FLAGS[7] internal signal to be connected to the FPGA_FLAGS[7] output.
6	FIFO_FULL0_SEL	R	1 – Select ADC0_FIFO_FULL internal signal to be connected to the FPGA_FLAGS[6] output. 0 – Select THRESH_FLAGS[6] internal signal to be connected to the FPGA_FLAGS[6] output.

***Table 10-31* • ACE FPGA FLAG Selection Register Description (continued)**

Bit Number	Name	R/W	Description
5	FIFO_AFULL2_SEL	R	1 – Select ADC2_FIFO_AFULL internal signal to be connected to the FPGA_FLAGS[5] output. 0 – Select THRESH_FLAGS[5] internal signal to be connected to the FPGA_FLAGS[5] output.
4	FIFO_AFULL1_SEL	R	1 – Select ADC1_FIFO_AFULL internal signal to be connected to the FPGA_FLAGS[4] output. 0 – Select THRESH_FLAGS[4] internal signal to be connected to the FPGA_FLAGS[4] output.
3	FIFO_AFULL0_SEL	R	1 – Select ADC0_FIFO_AFULL internal signal to be connected to the FPGA_FLAGS[3] output. 0 – Select THRESH_FLAGS[3] internal signal to be connected to the FPGA_FLAGS[3] output.
2	FIFO_EMPTY2_SEL	R	1 – Select ADC2_FIFO_EMPTY internal signal to be connected to the FPGA_FLAGS[2] output. 0 – Select THRESH_FLAGS[2] internal signal to be connected to the FPGA_FLAGS[2] output.
1	FIFO_EMPTY1_SEL	R	1 – Select ADC1_FIFO_EMPTY internal signal to be connected to the FPGA_FLAGS[1] output. 0 – Select THRESH_FLAGS[1] internal signal to be connected to the FPGA_FLAGS[1] output that is routed from the ACE to the FPGA fabric (default).
0	FIFO_EMPTY0_SEL	R	1 – Select ADC0_FIFO_EMPTY internal signal to be connected to the FPGA_FLAGS[0] output that is routed from the ACE to the FPGA fabric. 0 – Select THRESH_FLAGS[0] internal signal to be connected to the FPGA_FLAGS[0] output.

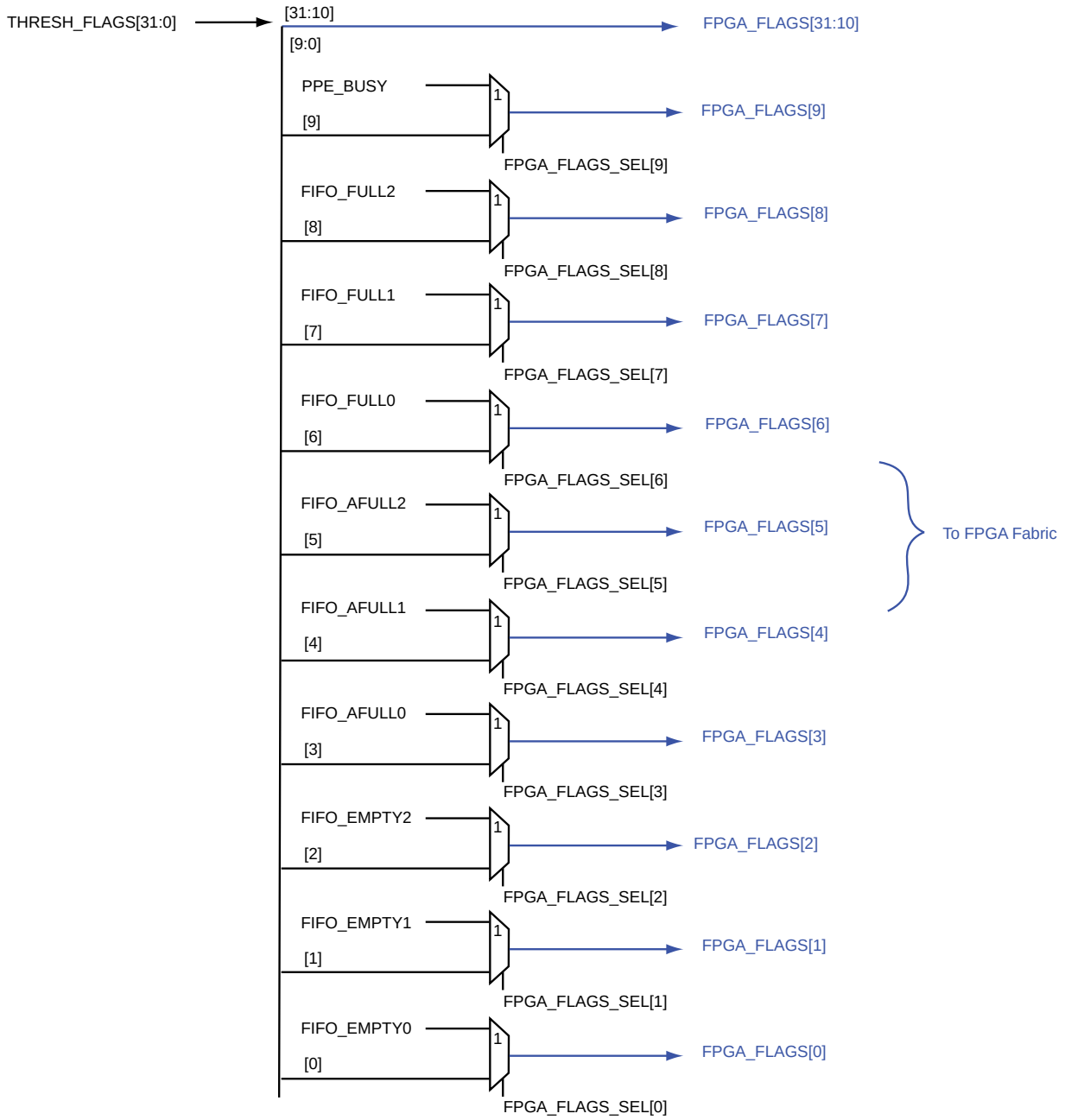


Figure 10-4 • **ACE FPGA Flag Details**

A – List of Changes

The following table lists critical changes that were made in the current version of the SmartFusion Programmable Analog User's Guide.

Revision	Changes	Page
Revision 1 (December 2011)	The majority of the "Analog Compute Engine (ACE)" chapter is new content (SAR 35395).	5
	The "Interrupts" section was updated (SAR 24909). Table 1-7 • ACE Interrupts is new (SAR 25087).	17
	The "SmartFusion Analog Features" section was revised, changing twenty-five to twenty-two in the bullet, "Up to twenty-two 0 V–2.56 V unipolar single-ended inputs" (SAR 24776).	21
	A2F060 was added to text throughout where appropriate. For example, the "SmartFusion Analog Features" section was revised for A2F060 so that it now states there are one to three rather than two or three ADCs and SDDs per device (SAR 35472).	21, 27, 49, 67, 71, 81
	Information about the analog compute engine configuration registers was added to the "Analog Front-End Configuration" section.	24
	Figure 2-1 • Partial Block Diagram of SmartFusion Programming Analog Showing One ADC, One DAC, and Two SCBs and Figure 7-2 • Functional Block Diagram of Temperature Monitor were revised to remove the ground from the GNDTMx pins. GNDTMx pins should not be connected to ground (SAR 28787). In Figure 2-1 • Partial Block Diagram of SmartFusion Programming Analog Showing One ADC, One DAC, and Two SCBs, the OBD reference voltage multiplexer is now assigned only to comparators CMP1 and CMP3 and is not ADC direct input for Ch10 and Ch12 (SAR 25997).	23, 83
	The units for VAREF Capacitor Value in Table 3-6 • VAREF Stabilization Time were corrected from F to µF (SAR 31682). The table was moved to the *SmartFusion Customizable System-on-Chip (cSoC)* datasheet because it is data extracted from characterization (SAR 24298).	N/A
	The following sentence was added to the "Sigma-Delta DAC General Description" section: When an SDD is disabled, the output is in a high impedance state (SAR 27381).	49
	The "Shared Pins" section was revised to remove reference to the use of DAC output as ADC input (SAR 33110).	59
	Table 4-2 • ACE Registers Used in Configuring or Using the Sigma-Delta DACs was revised to change the address for register SCB0_B11 to 0x40020230 (SAR 29306).	61
	The maximum operating input voltage was corrected from ±15 V to –11.5 V to +14 V in the "ABPS General Description" section (SAR 35109).	
	In the "Current Monitor Detailed Description" section, the following sentence was corrected by revising 1 ms to 1 µs: "Therefore, the ADC_START signal should not be asserted until 1 µs after the current monitor strobe has been raised."	72
	The "ACE Interrupts" chapter is new (SARs 25760, 35395).	111

Revision	Changes	Page
Revision 0 (March 2010)	The "Direct ADC Input Detailed Description" section was revised to include a statement that configuration of direct ADC pads as LVTTL inputs is done via flash cells and not affected by MSS reset initiated system boot code rerun (SAR 31131).	104
	Table 2-1 • Analog Front-End Input Pad Designations and Shared Functions was replaced.	23
	The integral nonlinearity (INL) error for 10-bit mode was revised to 0.8 in the "Key Features" section.	27
	Figure 3-2 • Example SAR ADC Architecture was revised to indicate there is one $/2^{N-1}$ capacitor, not two.	28
	The "Acquisition Time or Sample Time Control" section was revised to change the units of sample time to microseconds (µs) rather than seconds.	35
	Figure 3-11 • Intra-Conversion and Figure 3-12 • Injected Conversion were interchanged; they had been mislabeled in the previous version of the document.	39, 40
	EQ 3-17 was revised to indicate that the value of CALIBRATE is 1.	42
	VREFSEL was changed to VAREFSEL in Table 3-12 • ADCx_MISC_CTRL.	46
	Bit 5 was changed from PWRDWN to PWRDWN_x in Table 3-8 • ANA_COMM_CTRL.	45
	Figure 6-2 • Functional Block Diagram of Current Monitor was corrected by changing SBC to SCB. CBM MUX_SEL was changed to CMB_MUX_SEL.	73
	The first column heading in Table 6-5 • Addresses for Current-Monitor-Related Configuration Control Bits was changed to clarify that the addresses are relative to each SCB. The CM_STB_G entries were revised to CM_STB_G[0] through CM_STB_G[3].	77
	The first column heading in Table 8-2 • Relative and Absolute Addresses for Comparator-Related Configuration Control Bits was changed to clarify that the addresses are relative to each SCB.	94
	Table 8-6 • Bit Mapping Between Comparators, Interrupt Control Registers, and Cortex-M3 Interrupts was revised to change the relative comparator and edge from CMP7 and CMP6 output falling to CMP9 and CMP8 for bits 9 and 8.	98
Draft B (December 2009)	Figure 1-1 • Analog Compute Engine (A2F500 shown) was revised to show that ADC Unit 2 has one signal conditioning block (SCB), not two.	5
	The "SmartFusion Analog Features" section was revised.	21
	The "Key Features" section was revised.	27
	Table 3-6 • ADC Memory Map was revised to change all register addresses from an address of the form 0x4001XXXX to an address of the form 0x4002XXXX.	43
	The "ABPS General Description" section was updated to revise the paragraph beginning with the sentence, "The prescalers are constructed from continuous time op-amps in an inverting configuration."	67
	The "ABPS Configuration" section was revised to clarify that the ACE registers can be controlled by AHB bus masters.	69
	The first column heading in Table 5-3 • Relative and Absolute Addresses for ABPS-Related Configuration Control Bits was changed to clarify that the addresses are relative to each SCB.	69
	The "Current Monitor Configuration" section was revised.	76

Revision	Changes	Page
Draft B (December 2009)	The "Temperature Monitor Detailed Description" section was revised to suggest a choice for a temperature-sensing PN-junction.	82
	Table 7-1 • Nominal Scale Factor for ADC Resolution Selected is new.	83
	The first column heading in Table 7-5 • Addresses for Temperature Monitor-Related Configuration Control Bits was changed to clarify that the addresses are relative to each SCB.	88

B – Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call 800.262.1060
From the rest of the world, call 650.318.4460
Fax, from anywhere in the world, 650.318.8044

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues, and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

Visit the Customer Support website (www.microsemi.com/soc/support/search/default.aspx) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the website.

Website

You can browse a variety of technical and non-technical information on the SoC home page, at www.microsemi.com/soc.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech_itar@microsemi.com. Alternatively, within My Cases, select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.

Index

A

ABPS
  block diagram 68
  configuration 69
  control bits addresses 69
  detailed description 68
  features 67
  general description 67
  input pad designations 69
ACE 5
  architecture 5
  dedicated FIFO for PDMA 15
  FPGA flag details 124
  interface to analog front-end 10
  interface to FPGA fabric 10
  interface to MSS 6
  interfaces 6
  interrupts 9
  interrupts routed to Cortex-M3 processor 111
  interrupts to FPGA 122
  register map 61
ACE interrupts 111
ACE_COMP_IRQ 112
ACE_COMP_IRQ_CLR 112
ACE_COMP_IRQ_EN 112
ACE_FPGA_FLAGS_SEL 122
ACE_PPE_FIFO_IRQ 115
ACE_PPE_FIFO_IRQ_CLR 116
ACE_PPE_FIFO_IRQ_EN 115
ACE_PPE_FLAGS0_IRQ_CLR 118
ACE_PPE_FLAGS0_IRQ_EN 117
ACE_PPE_FLAGS1_IRQ 119
ACE_PPE_FLAGS1_IRQ_CLR 119
ACE_PPE_FLAGS1_IRQ_EN 118
ACE_PPE_FLAGS2_IRQ 120
ACE_PPE_FLAGS2_IRQ_CLR 120
ACE_PPE_FLAGS2_IRQ_EN 119
ACE_PPE_FLAGS3_IRQ 121
ACE_PPE_FLAGS3_IRQ_CLR 121
ACE_PPE_FLAGS3_IRQ_EN 120
ACE_SSE_IRQ 113
ACE_SSE_IRQ_CLR 114
ACE_SSE_IRQ_EN 113
acquisition time control 35
active bipolar prescaler (ABPS) 67
ADC
  block diagram 27
  calibration 40
  clock 37
  configuration example 42
  conversion 38
  example architecture 28
  interrupts 41
  key features 27
  power control 40
  register map 42
  resolution 38
  startup sequence 41
  terminology 30
  theory of operation 28
ADC_SYNC_CONV 44
ADCx_CONV_CTRL 45
ADCx_MISC_CTRL 46
ADCx_STATUS 47
ADCx_STC 46
ADCx_TVC 46
AFE
  configuration 24
  functions and location in SCB 23
  general description 21
  overview 21
ANA_COMM_CTRL 45
analog
  block diagram 23
  low-pass filter 54
analog multiplexer (MUX) 35
analog-to-digital converter (ADC) 27

C

comparator
  application 99
  bit mapping 98
  configuration 94
  connections 92
  control bits addresses 94
  detailed description 92
  features 91
  input pad designations 93
  interrupt event registers 111
  interrupts 97
  IRQ enable register 112
  IRQ register 112
  register map 97
  shared functions 93
  status output register bit fields 97
contacting Microsemi SoC Products Group
  customer service 129
  email 129
  web-based technical support 129
conversion example 29
current monitor
  common configuration 79
  control bits addresses 77

detailed description 72
features 71
general description 71
input pad designations 76
switch sequence 74
timing 74
current monitor configuration 76
customer service 129

D

DAC
one-bit converter 54
differential non-linearity (DNL) 30
direct ADC input
configuration 107
control bits addresses 107
detailed description 104
features 103
input pad designations 106
sharing inputs 104, 105
DNL 30

E

effective number of bits (ENOB) 30
ENOB 30

F

FS error 31
full-scale error (FS error) 31

G

gain error 31
drift 31

I

INL 32
integral non-linearity (INL) 32
interface
fabric interface signals to ACE/AFE 10
interrupts
ACE 17, 111
ADC 41
comparator 98

L

least significant bit (LSB) 32
LSB 32

M

Microsemi SoC Products Group
email 129
web-based technical support 129
website 129
modulator
block diagram 53

O

offset error 32

P

PPE
ALU architecture 15
architecture 13
flag aggregation details 121
flag aggregation overview 117
interface to MSS 6
interrupts 9, 115
main functions 15
round robin processing 14
threshold and state filter flag interrupt events 117
PPE_CTRL 8
PPE_FLAG0 117
PPE_FLAG1 118
PPE_FLAG2 119
PPE_FLAG3 120
product support
customer service 129
email 129
My Cases 130
outside the U.S. 130
technical support 129
website 129

R

register map
ACE 61
ADC 43
comparator 97
resolution 33

S

sample time control 35
sampling rate 33
SCB
features contained in each 22
SD DAC
configuration 60
control bits and addresses 62
description 49
detailed description 52
effect of external capacitor 55
frequency domain 56
generic block diagram 51
input examples 57
input/output pad designations 59
interpolation and reconstruction 57
low-pass filtering 58
modulation 57
principle 50
shared pins 59
SmartFusion block diagram 53
time-domain 56
sigma-delta digital-to-analog converter (SD DAC) 49

signal-to-noise and distortion (SINAD) 33
signal-to-noise ratio (SNR) 33
SINAD 33
SNR 33
SSE
  architecture 11
  interface to MSS 6
  interrupt events 112
  interrupts 9
  main functions 12
  micro architecture 12
SSE_TS_CTRL 7

T

tech support
  ITAR 130
  My Cases 130
  outside the U.S. 130
technical support 129
temperature monitor
  block diagram 83
  common configuration 89
  control bit addresses 88
  description 81
  detailed description 82
  features 81
  input pad designations 87
  nominal scale factor 83
  switch sequencing 83
  timing 85
timing
  current monitor strobe operation 75
  temperature monitor 85
total harmonic distortion 34
total unadjusted error 34
TUE 34

V

voltage reference, integrated 41

W

web-based technical support 129

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

Microsemi Corporation (NASDAQ: MSCC) offers a comprehensive portfolio of semiconductor solutions for: aerospace, defense and security; enterprise and communications; and industrial and alternative energy markets. Products include high-performance, high-reliability analog and RF devices, mixed signal and RF integrated circuits, customizable SoCs, FPGAs, and complete subsystems. Microsemi is headquartered in Aliso Viejo, Calif. Learn more at **www.microsemi.com**.

50200251-1/12.11