SigmaStudio for SHARC (ADSP-SC5xx/ADSP-215xx)

Product Description

SigmaStudio™ is a development environment from Analog Devices for graphically programming ADI's DSPs. SigmaStudio for SHARC includes an extensive set of algorithms to perform audio processing tasks such as filtering and mixing, as well as basic low-level DSP functions, optimized to run on the SHARC family of processors.

The environment is integrated with CrossCore® Embedded Studio.

This version of the product supports all variants of ADSP-SC59x, ADSP-SC58x, ADSP-SC57x, ADSP-2159x, ADSP-2158x, ADSP-2157x and ADSP-2156x processor series and has been tested on ADSP-SC584, ADSP-SC589, ADSP-21584, ADSP-SC573, ADSP-21573, ADSP-21569, ADSP-SC594 and ADSP-21593 processors.

Product Highlights

SigmaStudio for SHARC (ADSP-SC5xx/ADSP-215xx) has been optimized to run on the Analog Devices' SHARC ADSP-SC59x, ADSP-SC58x, ADSP-SC57x, ADSP-2159x, ADSP-2158x, ADSP-2157x and ADSP-2156x processors. The released product can be directly used with the specified version of SigmaStudio to create and fine tune audio systems designed by user. The module has been rigorously tested with various Schematics and input/output scenarios.

The product has three components:

1. SigmaStudio Host DLLs which supports the SHARC processor: These DLLs work in conjunction with a SigmaStudio Host and communicates to the SHARC Target.
2. The target framework: The target framework provides APIs for system initialization component, communication component for communication with host through SPI, audio framework component and inter processor communication (IPC) component.
3. The Target Library for the SHARC Target: The Target Library provides C-callable API's for creating, initializing and processing the SSn. The target framework uses these APIs for creating and initializing an instance of SSn.

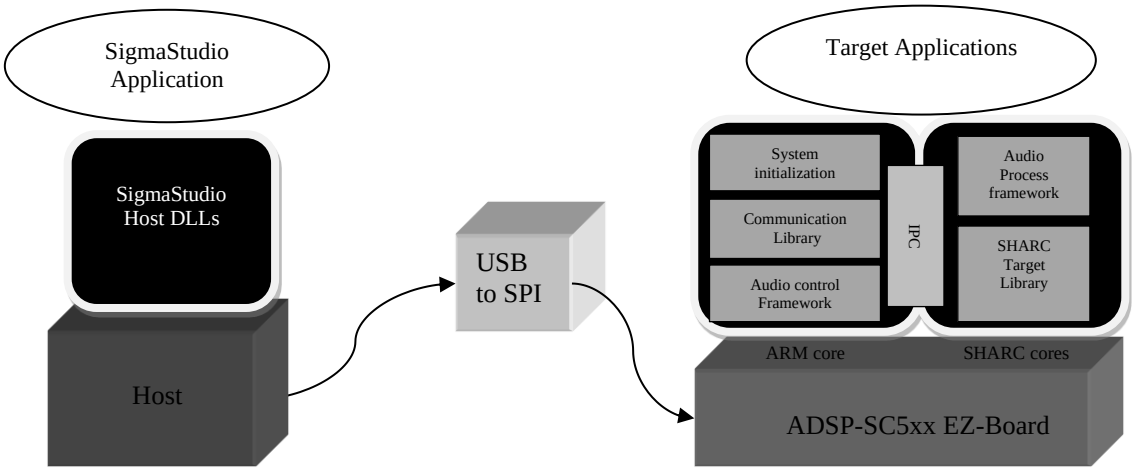

Figure 1: SigmaStudio for SHARC (ADSP-SC5xx)

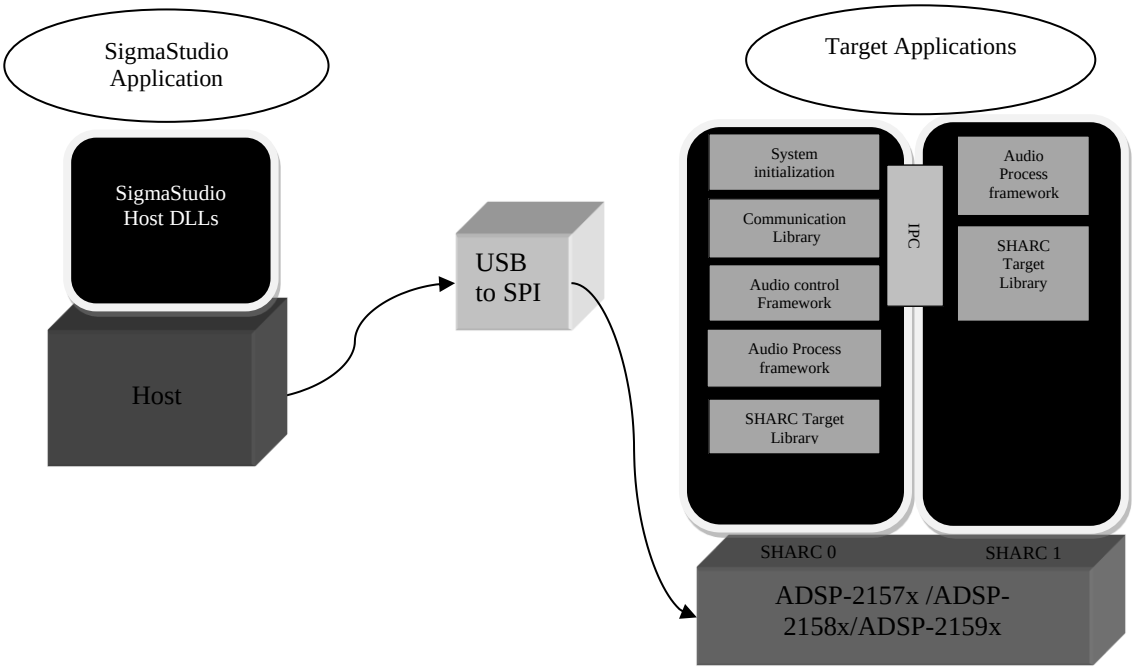

Figure 2: SigmaStudio for SHARC (ADSP-2157x/ADSP-2158x/ADSP-2159x)

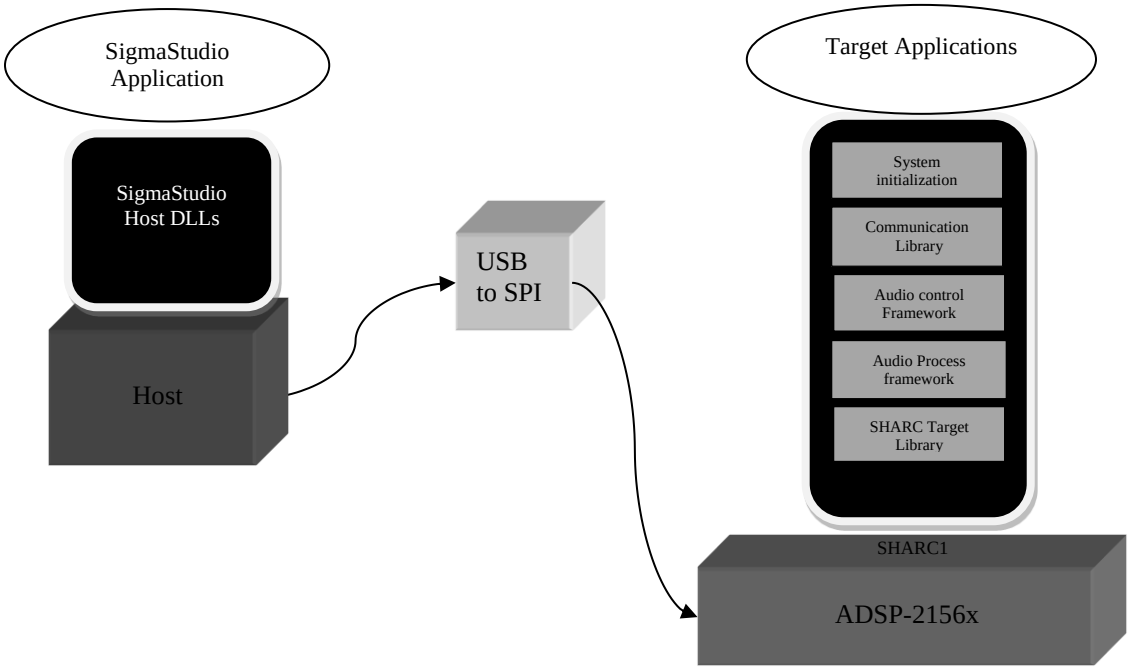

Figure 3: SigmaStudio for SHARC (ADSP-2156x)

An example application has been provided for each of the cores which invoke the target framework APIs for system initialization, host communication, audio processing /rendering and multi core synchronization. The system

initialization, communication and audio control framework components run on the ARM core of ADSP-SC59x, ADSP-SC58x and ADSP-SC57x processors and primary SHARC core of ADSP-2159x, ADSP-2158x and ADSP-2157x processors while the audio process framework component runs on each of the SHARC cores on ADSP-SC59x, ADSP-SC58x, ADSP-SC57x, ADSP-2159x, ADSP-2158x and ADSP-2157x variants. IPC component runs on each of cores for inter core communication and synchronization through shared memory. All the components mentioned above run on the single SHARC core in ADSP-2156x. IPC components is not applicable for ADSP-2156x.

Using this Default Application, users can quickly develop the signal chain and test the features before migrating to custom platform. Users can also create their own application and invoke the target library and communication APIs if they wish not to use the provided framework. An example depicting target and communication library integration is available within the package in the "Examples/LibraryIntegration" folder.

The scope of the SigmaStudio Host DLLs, the target framework and the Target Library is illustrated in Figure 1 and Figure 2.

Executive Summary Table

Tables below captures the memory and performance of the different framework components running on the ARM and SHARC cores of the ADSP-SC5xx, ADSP-2157x/ADSP-2158x/ADSP-2159x and ADSP-2156x.

Table 1: Summary Table for ADSP-SC5xx

	Code (KiB[1])		Data RAM (KiB)		Data ROM (KiB)		Stack Allocated (KiB)	MIPS (Block Processing)
	L1	**L2**	**L1**	**L2**	**L1**	**L2**		
SSn Library	5.497	-	Variable (depends on schematic)	6	0.008	-	4	2.67
SHARC Framework (Audio process framework and IPC)	18.84	-	12.625	0.348	0.41	-		~17
ARM Framework (Audio control framework, Communication, Connection, System and IPC)	-	22.68 (Cached)	-	15.78 (Un-cached) + 10.69 (cached)	-	0.16 (Un-cached)	4	~15

[1] 1 KiB = 1024 Bytes

Table 2: Summary table for ADSP-2157x/ ADSP-2158x/ADSP-2159x

	Code (KiB[2])		Data RAM (KiB)		Data ROM (KiB)		Stack Allocated (KiB)	Mips (Block Processing)
	L1	L2	L1	L2	L1	L2		
SHARC 0 Framework (Audio process framework, Audio control framework, Communication, Connection, System and IPC)	63.934	-	23.55	14.29	1.16	-	4	~15.42
SHARC 1 Framework (Audio process framework and IPC)	17.58	-	12.625	0.395	0.387	-	4	~14.10

Table 3: Summary table for ADSP-2156x

	Code (KiB[3])		Data RAM (KiB)		Data ROM (KiB)		Stack Allocated (KiB)	Mips (Block Processing)
	L1	L2	L1	L2	L1	L2		
SHARC 0 Framework (Audio process framework, Audio control framework, Communication, Connection, System)	52.842	-	1.3223 (uncached) + 19.868(cached)	13.6	1.13	-	6	~13.1

Memory and MIPS measurement is carried out under following conditions:

[2] 1 KiB = 1024 Bytes
[3] 1 KiB = 1024 Bytes

- The setup used for MIPS measurement uses analog audio sampled at 48000 Hz with 8 input channels connected to 8 output channels for each of the cores in the schematic. The SHARC cores are connected in Series Configuration with a single instance of SSn running on both of the SHARC cores. The Process Mode option is set to "Single" and Dual Core option set is to "Single Core" in IC Control Form for both the SHARC cores.
- Block Size used is 64 samples
- The variable part of Data RAM depends on the memory requirements of the worst-case Schematic to be supported.
- The L1 Data RAM requirements of SHARC and ARM frameworks in Table 1 is obtained assuming the following values for the below mentioned framework parameters,
  Number of process blocks = 1
  Block Size = 64 samples,
  Sampling Rate = 48000 Hz,
  Number of input channels = 8,
  Number of output channels = 8
  Internal I/O buffering = triple buffering.
- The stack requirements mentioned in the above table is the stack allocated as part of the applications.
- Table 4 provides the performance figures for two custom Schematics executed on ADSP-SC584 EZ-Board.

Example Schematics	MIPS				Memory (bytes)			
	Average Core 1	Peak Core 1	Average Core 2	Peak Core 2	Code Core 1	Code Core 2	Data Core 1	Data Core 2
ToneGen_Equalizer_Compressor_Block_SC5xx.dsppproj	9.8693	9.8767	3.0128	3.0143	6984	4364	2888	1472
Volume_Mute_Block_SC5xx.dsppproj	8.2950	8.3003	8.0798	8.0813	5694	6192	5996	6440

Table 4: MIPS and memory for ADSP-SC5xx example schematics provided as part of package

- Table 5 provides the performance figures for two custom Schematics executed on ADSP-21569 EZ-Board.

Example Schematics	MIPS		Memory (bytes)	
	Average Core 1	Peak Core 1	Code Core 1	Data Core 1
ToneGen_Equalizer_Compressor_Block_2156x.dsppproj (2 instances running in series on Core1)	12.8644	13.7647	11136	4360
Volume_Mute_Block_2156x.dsppproj	10.196	10.665	6214	6596

Table 5: MIPS and memory for ADSP-2156x example schematics provided as part of package

- Table 6 provides the performance figures for two custom Schematics executed on ADSP-SC589-Mini EZ-Board.

Example Schematics	MIPS				Memory (bytes)			
	Average Core 1	Peak Core 1	Average Core 2	Peak Core 2	Code Core 1	Code Core 2	Data Core 1	Data Core 2
ToneGen_Equalizer_Compressor_Block_SC589_Mini.dsppproj	9.846	9.8475	3.0338	3.0338	6976	4360	2888	1472
Volume_Mute_Block_SC589_Mini.dsppproj	8.3010	8.3085	8.6985	8.6985	5996	6846	5996	6552

Table 6: MIPS and memory for ADSP-SC589-Mini example schematics provided as part of package

Specifications

Table 7 captures the specifications for Target Library, communication library and target framework.

Product Specifications/Features	
Target Processor	ADSP-SC59x, ADSP-SC58x, ADSP-SC57x, ADSP-2159x, ADSP-2158x, ADSP-2157x and ADSP-2156x
Release Format	Object code for SHARC target library with C-callable APIs, object code for ARM communication library for ADSP-SC5xx, object code for SHARC communication library and SigmaStudio Host DLLs. Full source code for framework.
Input	Input samples to the target library should be in 32-bit float format in -1 to +1 range. The conversion from fixed to 32-bit float format is taken care by the provided framework if this is used for integrating with a user application.
Output	Output samples from the target library will be in 32-bit float format in -1 to +1 range with amplitude of 1 representing 0db full scale. The conversion from 32-bit float format to fixed format is taken care by the provided framework if this is used for integrating with a user application.
Input/output Buffer Size	The input / output buffer size depends on the number of channels and number of samples per channel (block size) configured during initialization of the module.
Multi-Instance Support	Up to 3 instances are supported in either serial or parallel configurations.
SigmaStudio Host Support	Schematic capture and Tuning is similar to SigmaStudio for SigmaDSP.
Algorithm Support	Supports more than 75 different Algorithms and their variants.
VISA Support	Supported for ADSP-SC5xx and ADSP-215xx processors.
Algorithm Designer	Supported.
3rd party Module integration	Supported.
Micro-controller Integration support	Supported.
Version Checking	The Target Library does version checking to ensure compatibility between Target and host libraries.
MISRA-C Compliance	The product has undergone MISRA check. The deviations have been documented and are ensured via Code Reviews and Unit Testing.
Error Codes	The Target Library returns 6 unique error codes. The communication library returns 3 unique error codes. The audio framework returns 9 unique error codes.
SHARC Features	Algorithm performance is optimized using the SIMD feature. Core Modules support extended precision float operations.
Testing	The Algorithms have been tested for accuracy and stability.

Table 7: Product Specifications

Memory & Performance

This section presents the MIPS and memory usage of SigmaStudio for SHARC (ADSP-SC5xx/ADSP-215xx). Performance figures are based on several common memory configurations. This information will help the developers to estimate the range of processing requirements and overhead needed for their application.

System Configuration

The MIPS performance is based on worst case calling conditions and is expressed as Average and Peak figures. Developers will need to budget their MIPS requirements according to system and memory layout conditions using the performance figures listed below.

The core and system clocks for the various cores are set as below:

CCLK for SHARCs = 400 MHz for ADSP-SC584 and ADSP-SC573. 450 MHz for ADSP-SC589. 1 GHz for ADSP-SC594. 1 GHz for ADSP-21593. 1 GHz for ADSP-21569.

CCLK for ARM = 400 MHz for ADSP-SC584 and ADSP-SC573. 450 MHz for ADSP-SC589. 450 MHz for ADSP-SC589. 1 GHz for ADSP-SC594.

SCLK = 200 MHz for ADSP-SC584 and ADSP-SC573. 225 MHz for ADSP-SC589. 500 MHz for ADSP-SC594. 500 MHz for ADSP-21593. 500 MHz for ADSP-21569.

MIPS

Table 8 gives the summary of MIPS for selected block processing algorithms measured on ADSP-SC584 EZ-BOARD with the following memory configuration.

- Optimal Memory Layout – All Code and Data of the schematic in L1 memory.

Module	Block Processing	
	MIPS (Average)	Data memory(bytes) (Includes both State and Parameter memory)
Filters → Second Order → Single Precision → 1 Ch General (2nd Order) – growth 8	2.3	288
Filters → Second Order → Single Precision → 1 Ch General (2nd Order) – growth 16	4.398	576
Filters → Second Order → Single Precision → 1 Ch → Type 2 → General (2nd Order) – growth 8	2.13	196
Filters → Second Order → Single Precision → 1 Ch → Type 2 → General (2nd Order) – growth 16	4.008	388
Filters → FIR → Single Precision → Growable → FIR Filter – growth 1, order – 128	3.64	1056
Filters → FIR → Single Precision → Growable → FIR Filter – growth 1, order – 256	6.712	2080
Filters → FIR → Single Precision → Growable → FIR Filter – growth 1, order – 512	12.8581	4128
Filters → FIR → Single Precision → Growable → FIR Filter – growth 16, order – 128	54.273	9096
Filters → FIR → Extended Precision → Growable → FIR Filter – growth 1, order – 128	3.636	1056
Filters → FIR → Extended Precision → Growable → FIR Filter – growth 1, order – 256	6.708	2080

Filters → FIR → Extended Precision → Growable → FIR Filter – growth 1, order – 512	12.853	4128
Filters → FIR → Extended Precision → Growable → FIR Filter – growth 16, order – 128	54.319	9096
Dynamic Processors → RMS → Standard Resolution → Low Range (-90 to +6 dB) → No Post Gain → N-Channel → No Ext Detector Input → N channel – growth 1	1.965	180
Dynamic Processors → RMS → Standard Resolution → Low Range (-90 to +6 dB) → No Post Gain → N-Channel → No Ext Detector Input → N channel – growth 16	28.3185	360
Dynamic Processors → RMS → Standard Resolution → Full Range (-90 to +24 dB) → No Post Gain → Mono → Ext Detector Input → RMS (no gain)	1.9765	204
Dynamic Processors → RMS → Index Selectable → RMS Index Selectable	3.953	212
Dynamic Processors → RMS → Limiter	1.214	20
Dynamic Processors → Peak → Standard Resolution → Full Range (-90 to +24 dB) → No Post Gain → Mono → No Ext Detector Input → Peak Full Range (no gain)	1.7485	196
Filters → Second Order → Look Up → Single Precision Float → 2 Ch →Index Selectable Independent Multiple Band Filter	0.478	64
Filters → Crossover → Extended Precision → 3-Way → Crossover	2.9605	372
Sources → ChimeGenerator → Chime Generator	4.1	184
Filters → Second Order → Look Up → Extended Precision Float → No Slew → General Second Order Index Selectable – 1 Channel	0.6	96

Table 8: MIPS for Selected Block Processing Algorithms

Note that the MIPS figures in Table 8 are for the modules alone and does not include the framework MIPS.

Table 9 gives the summary of the MIPS measured for the FIR and IIR hardware accelerators on the ADSP-21569 EZ-BOARD with the following memory configuration.

- Optimal Memory Layout – All Code and Data of the schematic in L1 memory.

Module	**Block Processing**	
	MIPS (Average)	**Data memory(bytes) (Includes both State and Parameter memory)**
Filters → FIR → Single Precision → FIR Accelerator → FIR Filter – 1 Channel, 1024 Taps (using Hardware Accelerator)	18.358	8780
Filters → Second Order → Single Precision → IIR Accelerator → General Eq (2nd order) - 1 Channel (using Hardware Accelerator)	2.131	1384

Table 9: MIPS for 2156x FIR/IIR Hardware Accelerators

Deliverables

The deliverables for this product include:

- SWC and Byte Addressed Target Library for SHARC cores of ADSP-SC5xx and ADSP-215xx processors with a C-callable API (Application Programming Interface) consistent with other ADI Software Modules.
- Communication libraries for ARM cores of ADSP-SC57x, ADSP-SC58x and ADSP-SC59x processors with a C-callable API.
- Communication libraries for SHARC cores of ADSP-2157x, ADSP-2158x, ADSP-2159x and ADSP-2156x processors with a C-callable API.
- SigmaStudio Host DLLs for SHARC support (SharcPubLib.dll, SharcModules.dll and SharcDesigner.dll).
- C source Default Framework and Application which calls the Target Library and communication library for ADSP-SC573, ADSP-SC584, ADSP-SC589, ADSP-SC594, ADSP-21573, ADSP-21584, ADSP-21593 and ADSP-21569 processors.
- Executable Files (*.DXE) and flashable loader (*.LDR) files for real time demonstration running on ADI evaluation boards.
- Documentation, including Quick Start Guide.

Availability

Please visit www.analog.com/sigmastudio, for information on SigmaStudio and www.analog.com/processors for other product information.

Copyright, Disclaimer & Trademark Statements

Copyright Information

Disclaimer

Analog Devices, Inc. reserves the right to change this product without prior notice. Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use; nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under the patent rights of Analog Devices, Inc.

Trademark and Service Mark Notice

Analog Devices, the Analog Devices logo, SigmaStudio, Blackfin, SHARC, TigerSHARC, CrossCore, VisualDSP, VisualDSP++, EZ-KIT Lite, EZ-Extender and Collaborative are trademarks and/or registered trademarks "®" of Analog Devices, Inc.

All other brand and product names are trademarks or service marks of their respective owners.

Analog Devices' Trademarks and Service Marks may not be used without the express written consent of Analog Devices, such consent only to be provided in a separate written agreement signed by Analog Devices. Subject to the foregoing, such Trademarks and Service Marks must be reproduced according to ADI's Trademark Usage guidelines. Any licensee wishing to reproduce ADI's Trademarks and Service Marks must obtain and follow these guidelines for the specific marks at issue.