Technical notes on using Analog Devices products and development tools
Visit our Web resources http://www.analog.com/ee-notes and http://www.analog.com/processors or e-mail processor.support@analog.com or processor.tools.support@analog.com for technical support.

Using ADSP-SC84x/2184x High Performance FIR/IIR Accelerators

Contributed by Sanket Nayak *Rev 1 – July 30, 2026*

Introduction

The ADSP-SC84x processors incorporate high-performance hardware accelerators dedicated to performing two widely used signal processing operations: FIR (Finite Impulse Response) and IIR (Infinite Impulse response), hereby referred to as FIRA and IIRA, respectively.

Figure 1 shows that an ADSP-SC84x processor has a SHARC-FX core equipped with two FIR and four IIR accelerators. The SHARC-FX core and accelerators together provide an overall performance of up to *24 Giga 32-bit Floating Point MAC (Multiply and Accumulate) Operations Per Second* when they run at 1.2 GHz.

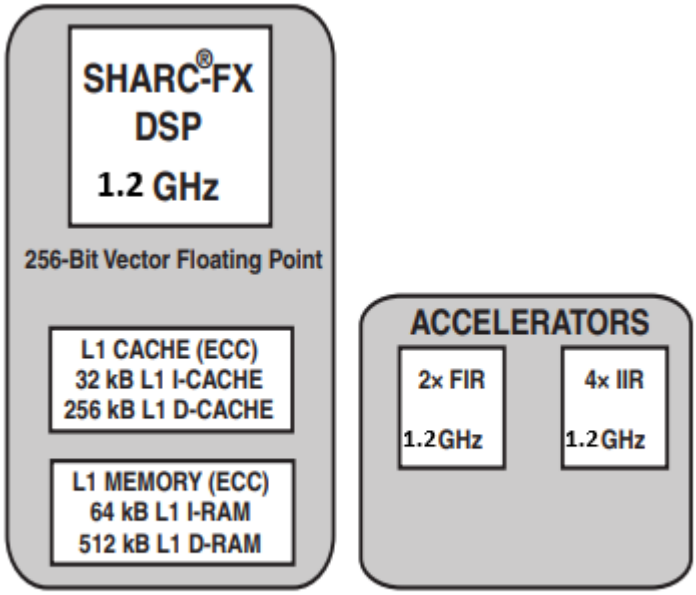


Figure 1: SHARC-FX Core and Accelerators on ADSP-SC84x Processors

Figure 2 shows that these accelerators can run in tandem with the core to perform dedicated fixed-function computations (FIR and IIR), thereby increasing the overall processing capability of the processor.

This EE Note provides a brief overview of the functional and performance improvements achieved with the ADSP-SC84x FIR and IIR accelerators. For this EE Note, FIRs and IIRs are referred to as *accelerators*, when compared with their predecessors.

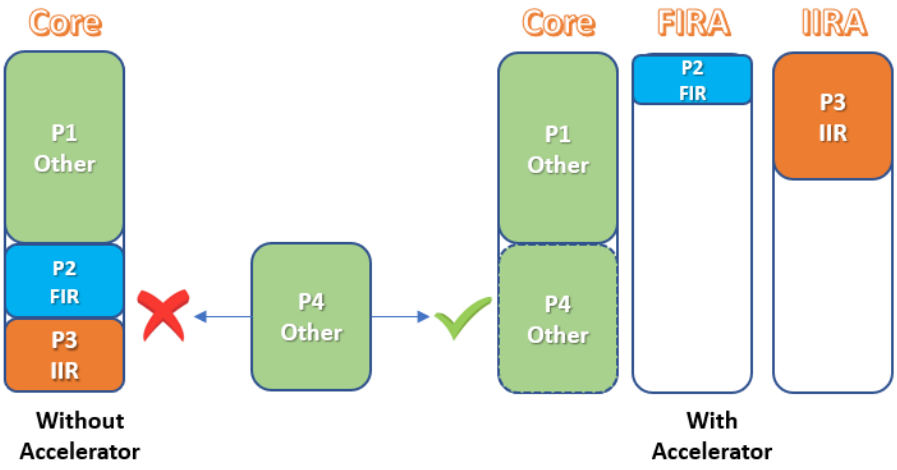


Figure 2: Processing Capability with and without FIR/IIR Accelerators

The document discusses the following topics:

- Accelerator performance under various configurations and contributing factors
- Accelerator drivers from CrossCore® Embedded Studio (CCES), including usage, examples, and benchmark details
- Accelerator usage models for optimum performance in various application scenarios

For more details about the accelerator architecture and programming model, refer to the ADSP-SC84x SHARC-FX Processor Hardware Reference [1].

Most of the concepts in the document are the same as in *Using ADSP-SC59x/2159x High Performance FIR/IIR Accelerators (EE-436)*[5]. In addition to the material in EE-436, this EE note includes feature enhancements, figures, tables, and data specific to the ADSP-SC84x family of processors.

ADSP-SC84x Accelerator Enhancements

Accelerator enhancements include performance and functional improvements.

Performance Improvements

- There are no specific performance enhancements between the accelerator module in the ADSP-SC84x processors and the ADSP-SC59x processors, however since the bus fabric and the SHARC-FX core architecture are different, the overall performance varies due to the different fabric latencies.

Functional Improvements

The functional improvement that the ADSP-SC84x accelerator features is the introduction of a dedicated hardware engine to slew the IIR coefficients in parallel with IIR filtering. This offloads that task from the core and speeds up the slewing process. The slewing engine is placed inside the IIR filter to avoid DMA overheads while loading the slewed coefficients. Configuring slewing mode operation is done in the IIR_CTL1 register.

Refer to the *ADSP-SC84x SHARC-FX Processor Hardware Reference* [1] for more information on coefficient slewing.

Accelerator Performance

The accelerator's processing cycles consist of two components, DMA cycles and compute (MAC) cycles.

For additional details on DMA and MAC cycles, refer to EE-436.

FIR/IIR Accelerators Performance

This section illustrates FIR/IIR accelerator performance with the following different filter parameters and processors:

- ADSP-SC84x FIRA/IIRA at maximum ACLK (=CCLK) = 1.2 GHz
- ADSP-SC59x FIRA/IIRA at maximum ACLK (=CCLK) = 1 GHz
- ADSP-SC84x(SHARC-FX) core at maximum CCLK = 1.2 GHz
- ADSP-2156x (SHARC+) core at maximum CCLK = 1 GHz

Refer to EE-436 for additional applicable comparisons.

During measurements the buffers are placed in L1 memory and core FIR/IIR cycles are measured with the CCES library functions.

Additional measurement and analysis details shown illustrate how various factors affect the accelerator MAC Utilization Efficiency (MUE).

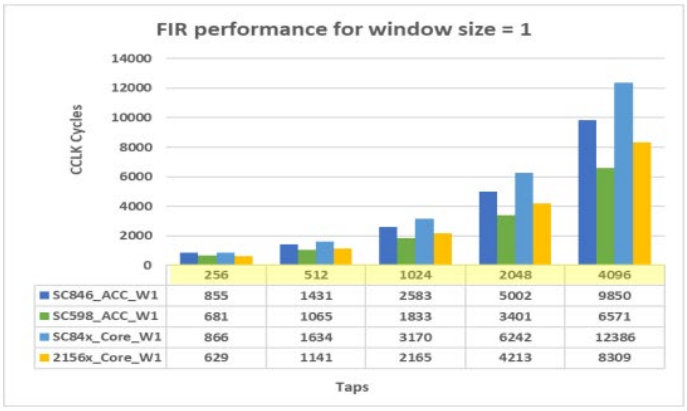
FIR Performance

This section contains the following information:

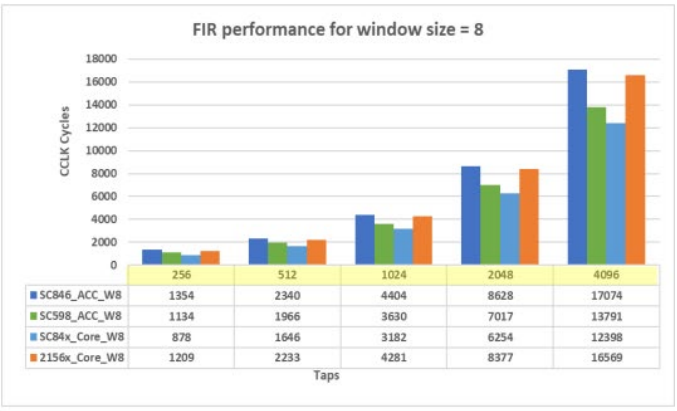
- Measured performance of several cases compared with the ADSP-SC84x FIRA using different filter parameters
- Factors that impact measured FIRA Compute Efficiency (CE)
- Effect of buffer placement (L1/L2/L3) on FIRA performance

ADSP-SC59x Accelerators/SHARC-FX (ADSP-SC84x) versus the SHARC+ (ADSP-2156x) Core

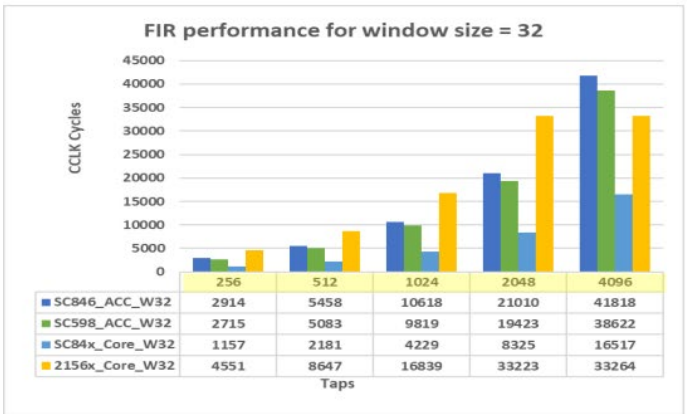
Figure 3 and Figure 4 show the measured FIR performance for five cases, in core cycles and time units (µs). Figure 5 shows the performance improvement factor of one ADSP-SC84x FIR accelerator instance, at different window sizes with a tap length of 512, in comparison with the other cases. The numbers in Figure 3 compare the performance of a single instance of the accelerator on different processors.



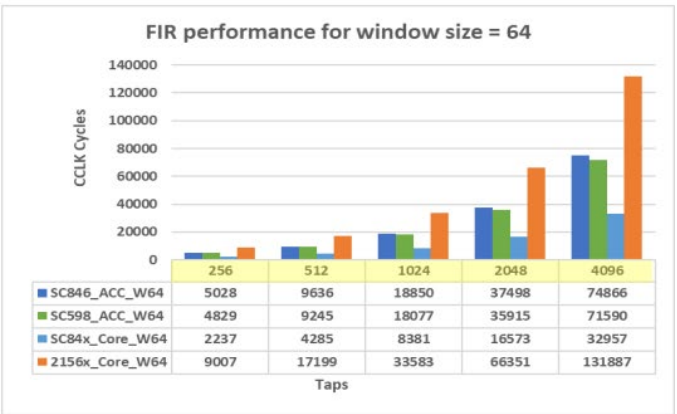
	256	512	1024	2048	4096
SC846_ACC_W1	855	1431	2583	5002	9850
SC598_ACC_W1	681	1065	1833	3401	6571
SC84x_Core_W1	866	1634	3170	6242	12386
2156x_Core_W1	629	1141	2165	4213	8309



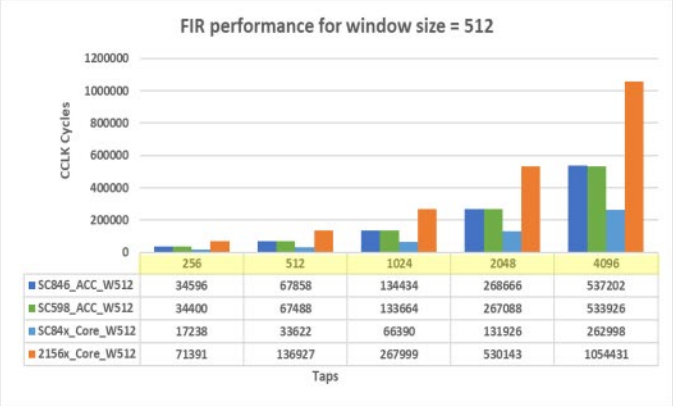
	256	512	1024	2048	4096
SC846_ACC_W8	1354	2340	4404	8628	17074
SC598_ACC_W8	1134	1966	3630	7017	13791
SC84x_Core_W8	878	1646	3182	6254	12398
2156x_Core_W8	1209	2233	4281	8377	16569



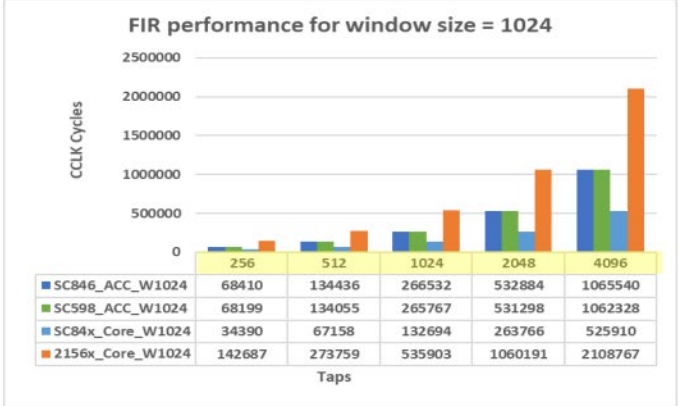
	256	512	1024	2048	4096
SC846_ACC_W32	2914	5458	10618	21010	41818
SC598_ACC_W32	2715	5083	9819	19423	38622
SC84x_Core_W32	1157	2181	4229	8325	16517
2156x_Core_W32	4551	8647	16839	33223	33264



	256	512	1024	2048	4096
SC846_ACC_W64	5028	9636	18850	37498	74866
SC598_ACC_W64	4829	9245	18077	35915	71590
SC84x_Core_W64	2237	4285	8381	16573	32957
2156x_Core_W64	9007	17199	33583	66351	131887

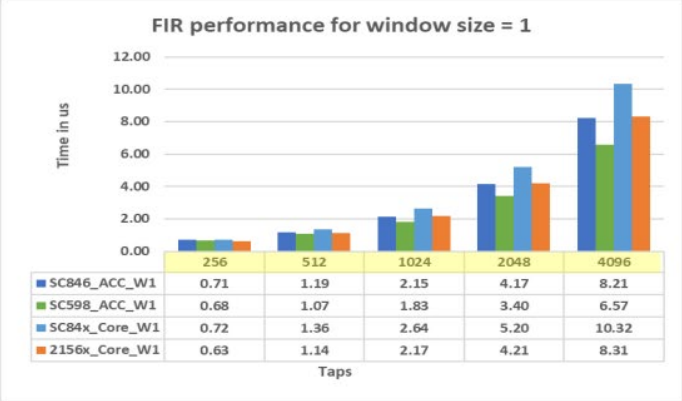


	256	512	1024	2048	4096
SC846_ACC_W512	34596	67858	134434	268666	537202
SC598_ACC_W512	34400	67488	133664	267088	533926
SC84x_Core_W512	17238	33622	66390	131926	262998
2156x_Core_W512	71391	136927	267999	530143	1054431

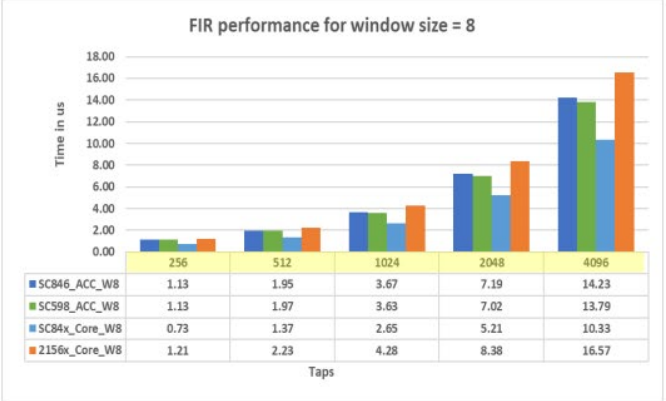


	256	512	1024	2048	4096
SC846_ACC_W1024	68410	134436	266532	532884	1065540
SC598_ACC_W1024	68199	134055	265767	531298	1062328
SC84x_Core_W1024	34390	67158	132694	263766	525910
2156x_Core_W1024	142687	273759	535903	1060191	2108767

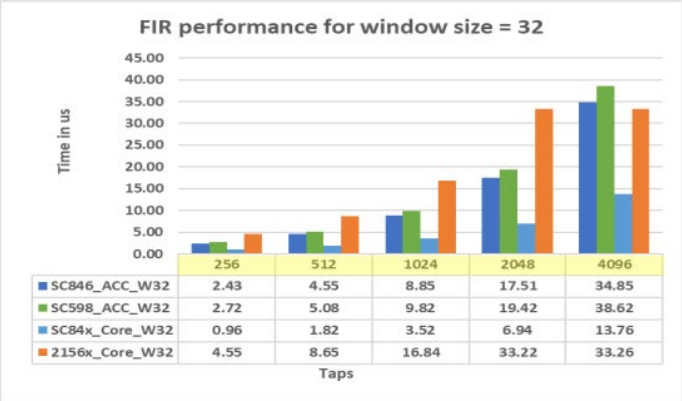
Figure 3: FIR Performance in Core Cycles Across Window Size and Tap Length



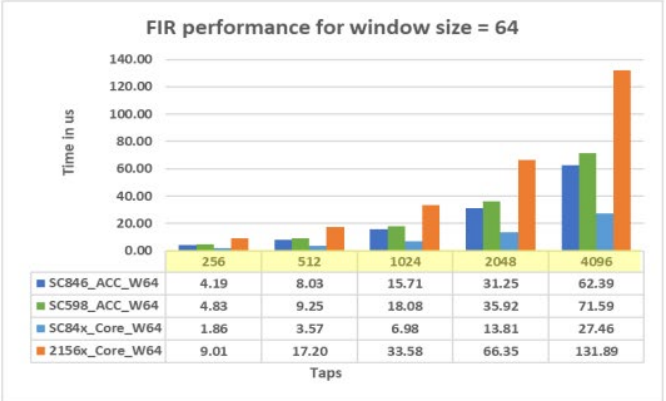
	256	512	1024	2048	4096
SC846_ACC_W1	0.71	1.19	2.15	4.17	8.21
SC598_ACC_W1	0.68	1.07	1.83	3.40	6.57
SC84x_Core_W1	0.72	1.36	2.64	5.20	10.32
2156x_Core_W1	0.63	1.14	2.17	4.21	8.31



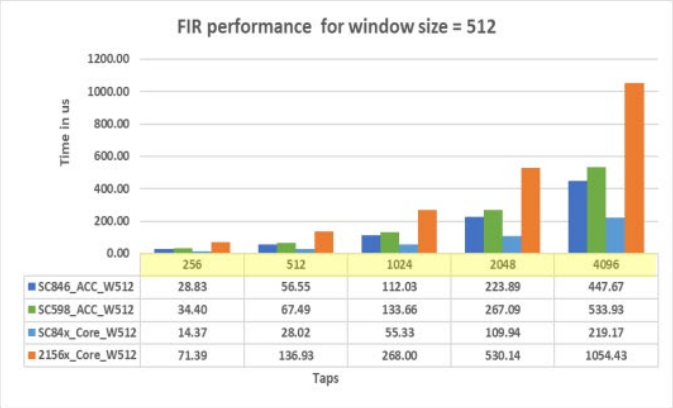
	256	512	1024	2048	4096
SC846_ACC_W8	1.13	1.95	3.67	7.19	14.23
SC598_ACC_W8	1.13	1.97	3.63	7.02	13.79
SC84x_Core_W8	0.73	1.37	2.65	5.21	10.33
2156x_Core_W8	1.21	2.23	4.28	8.38	16.57



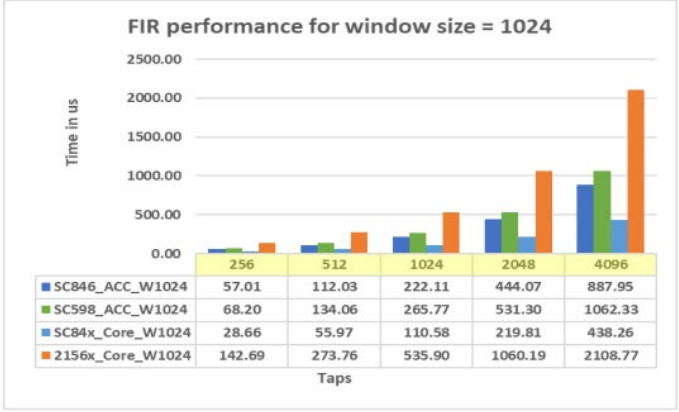
	256	512	1024	2048	4096
SC846_ACC_W32	2.43	4.55	8.85	17.51	34.85
SC598_ACC_W32	2.72	5.08	9.82	19.42	38.62
SC84x_Core_W32	0.96	1.82	3.52	6.94	13.76
2156x_Core_W32	4.55	8.65	16.84	33.22	33.26



	256	512	1024	2048	4096
SC846_ACC_W64	4.19	8.03	15.71	31.25	62.39
SC598_ACC_W64	4.83	9.25	18.08	35.92	71.59
SC84x_Core_W64	1.86	3.57	6.98	13.81	27.46
2156x_Core_W64	9.01	17.20	33.58	66.35	131.89

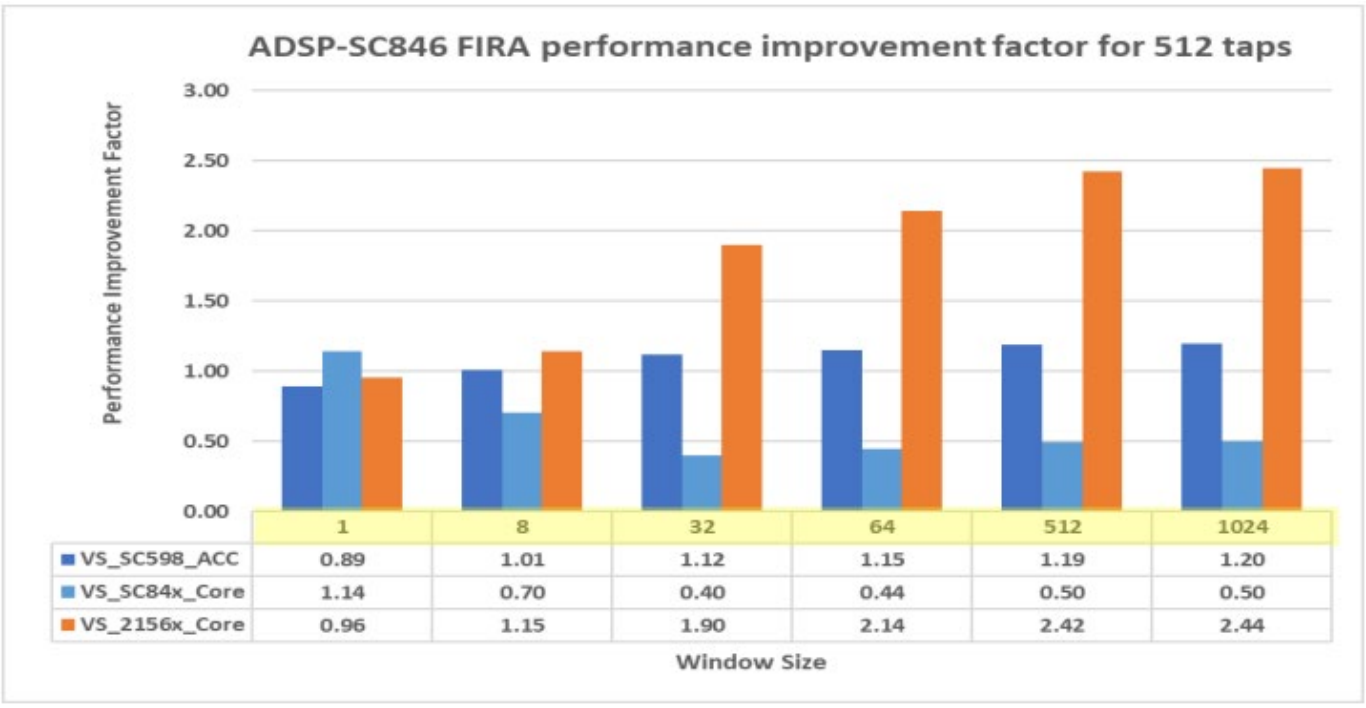


	256	512	1024	2048	4096
SC846_ACC_W512	28.83	56.55	112.03	223.89	447.67
SC598_ACC_W512	34.40	67.49	133.66	267.09	533.93
SC84x_Core_W512	14.37	28.02	55.33	109.94	219.17
2156x_Core_W512	71.39	136.93	268.00	530.14	1054.43



	256	512	1024	2048	4096
SC846_ACC_W1024	57.01	112.03	222.11	444.07	887.95
SC598_ACC_W1024	68.20	134.06	265.77	531.30	1062.33
SC84x_Core_W1024	28.66	55.97	110.58	219.81	438.26
2156x_Core_W1024	142.69	273.76	535.90	1060.19	2108.77

Figure 4: FIR Performance in Time (µs) Across Window Size and Tap Length



	1	8	32	64	512	1024
VS_SC598_ACC	0.89	1.01	1.12	1.15	1.19	1.20
VS_SC84x_Core	1.14	0.70	0.40	0.44	0.50	0.50
VS_2156x_Core	0.96	1.15	1.90	2.14	2.42	2.44

Figure 5: ADSP-SC84x FIR Performance Improvement Factor for 512 Taps Considering Time

The following observations are derived from Figure 5:

- **The ADSP-SC84x FIRA outperforms the ADSP-SC59x/2159x FIRA for large window sizes.** The improvement is roughly 1.2x. This is because of the difference in operating frequencies between the processors, with the ADSP-SC84x FIRA operating at 1.2 GHz and the ADSP-SC59x/2159x FIRA operating at 1 GHz.
- **The ADSP-SC84x SHARC-FX core outperforms the ADSP-SC84x FIRA for large window sizes**. The improvement is roughly 2x. This is because the SHARC-FX core has 8 MAC units, while the ADSP-SC84x FIRA only has 4. Note that for lower block sizes this is not the case

because implementation specific software overhead in the SHARC-FX core dominates the MAC computation cycles.

- **The ADSP-SC84x FIRA outperforms the ADSP-2156x SHARC+ core by roughly 2.4x**. There are two reasons for this improvement.
    - The ADSP-SC84x FIRA has 4 MAC units, while the ADSP-2156x SHARC+ core only has 2.
    - The ADSP-2156x SHARC+ core operates at 1 GHz and the ADSP-SC84x FIRA operates at 1.2 GHz.

  Note that this result does not apply to smaller block sizes where DMA overhead dominates the compute cycles.

Compute Efficiency (CE)

For information on CE, refer to EE-436.

Buffer Placement

For information on buffer placement, refer to EE-436.

IIR Performance

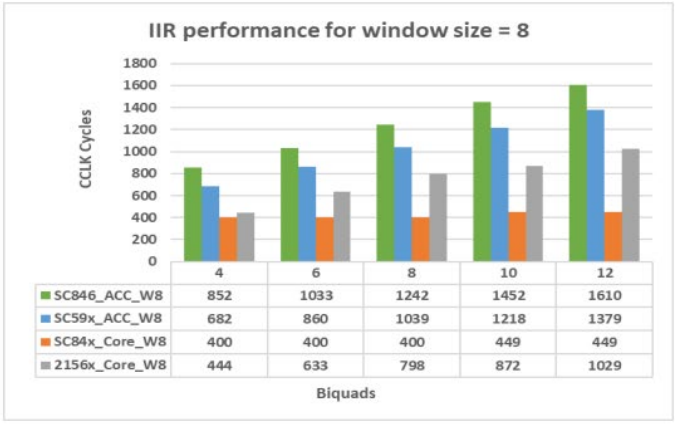
This section contains the following topics:

- Measured performance of several cases compared with the ADSP-SC84x IIRA using different filter
- Factors that impact measured IIRA Compute Efficiency (CE)
- Effect of buffer placement (L1/L2/L3) on IIRA performance

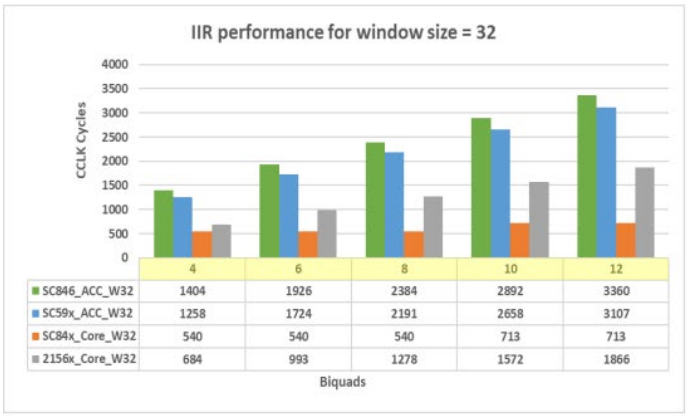
ADSP-SC59x Accelerators/SHARC-FX (measured on ADSP-SC84x) versus the and SHARC+ (ADSP-2156x)

Figure 6 and Figure 7 show the measured IIR performance, for all cases, for one IIRA instance in core cycles and time units (µs). Figure 8 shows the performance factor for the ADSP-SC84x IIRA compared with several cases, both for different window sizes and with 6 biquad filters. The numbers in Figure 6 compare the performance of a single instance of the accelerator on different processors.

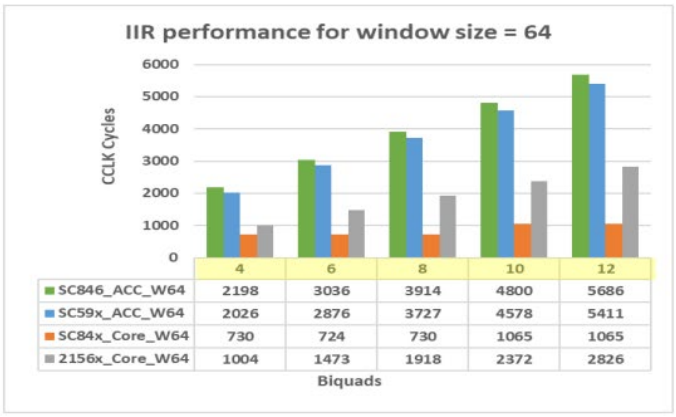
	4	6	8	10	12
SC846_ACC_W1	690	781	919	1014	1142
SC59x_ACC_W1	522	604	703	785	884
SC84x_Core_W1	354	354	354	350	350
2156_Core_W1	189	199	211	223	235



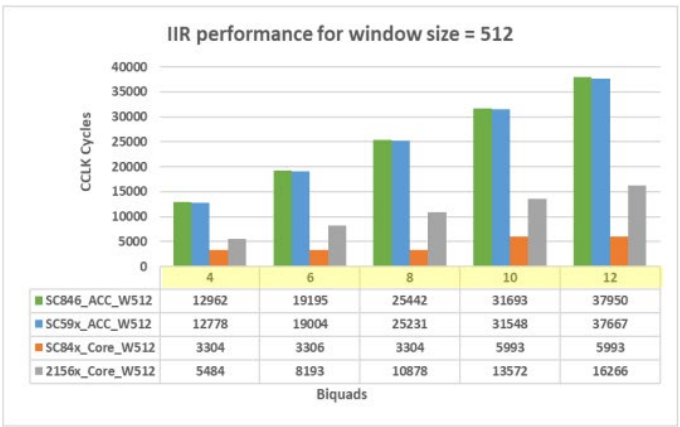
	4	6	8	10	12
SC846_ACC_W8	852	1033	1242	1452	1610
SC59x_ACC_W8	682	860	1039	1218	1379
SC84x_Core_W8	400	400	400	449	449
2156x_Core_W8	444	633	798	872	1029



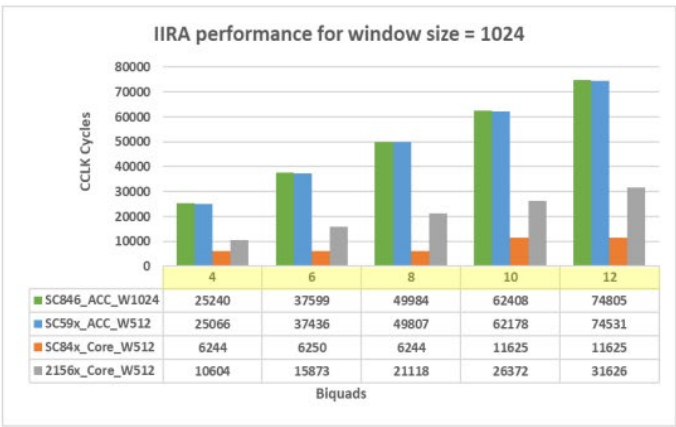
	4	6	8	10	12
SC846_ACC_W32	1404	1926	2384	2892	3360
SC59x_ACC_W32	1258	1724	2191	2658	3107
SC84x_Core_W32	540	540	540	713	713
2156x_Core_W32	684	993	1278	1572	1866



	4	6	8	10	12
SC846_ACC_W64	2198	3036	3914	4800	5686
SC59x_ACC_W64	2026	2876	3727	4578	5411
SC84x_Core_W64	730	724	730	1065	1065
2156x_Core_W64	1004	1473	1918	2372	2826

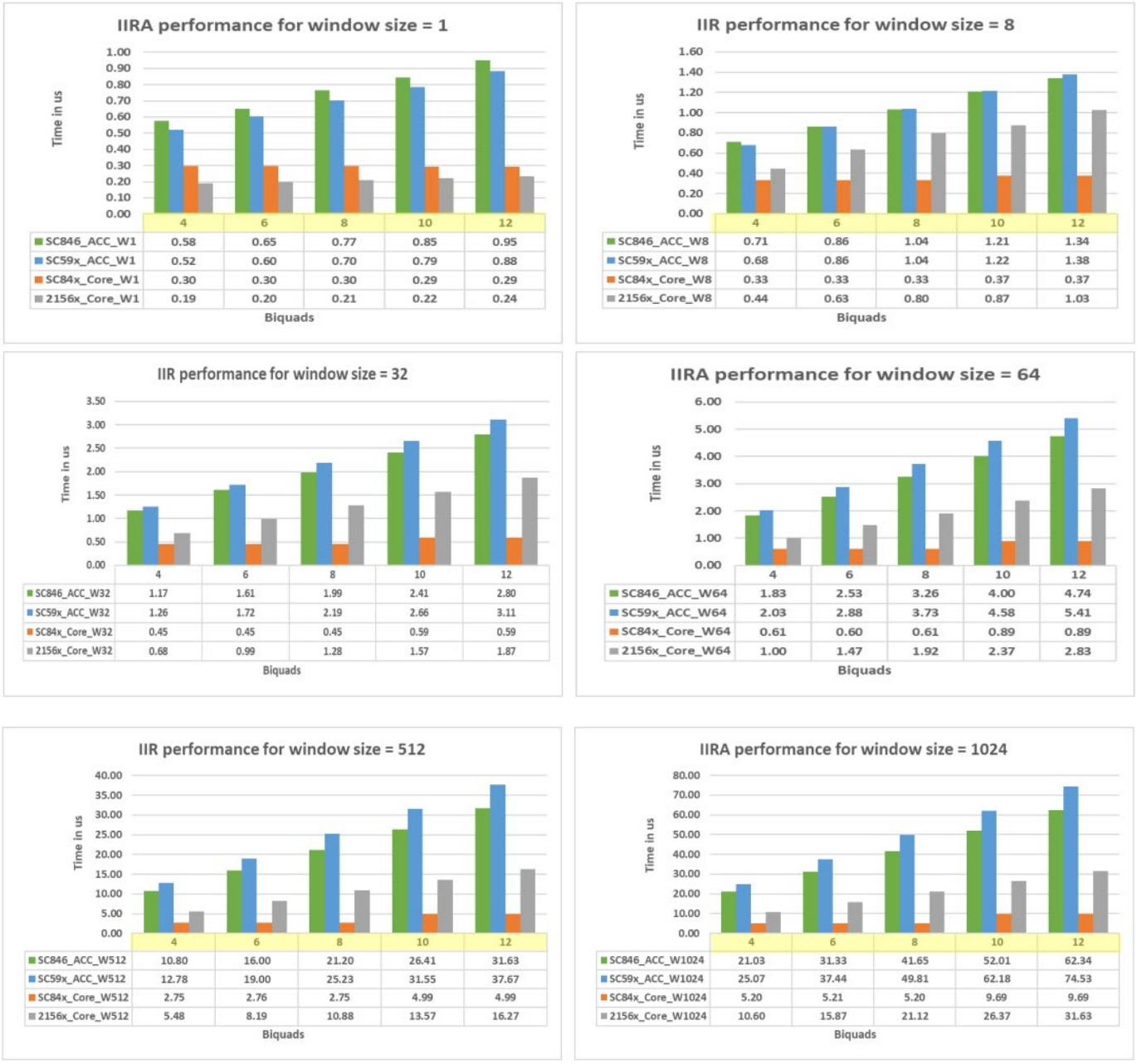


	4	6	8	10	12
SC846_ACC_W512	12962	19195	25442	31693	37950
SC59x_ACC_W512	12778	19004	25231	31548	37667
SC84x_Core_W512	3304	3306	3304	5993	5993
2156x_Core_W512	5484	8193	10878	13572	16266



	4	6	8	10	12
SC846_ACC_W1024	25240	37599	49984	62408	74805
SC59x_ACC_W512	25066	37436	49807	62178	74531
SC84x_Core_W512	6244	6250	6244	11625	11625
2156x_Core_W512	10604	15873	21118	26372	31626

Figure 6: IIR Performance in Core Cycles Across Window Size and Biquad Values



	4	6	8	10	12
SC846_ACC_W1	0.58	0.65	0.77	0.85	0.95
SC59x_ACC_W1	0.52	0.60	0.70	0.79	0.88
SC84x_Core_W1	0.30	0.30	0.30	0.29	0.29
2156x_Core_W1	0.19	0.20	0.21	0.22	0.24

	4	6	8	10	12
SC846_ACC_W8	0.71	0.86	1.04	1.21	1.34
SC59x_ACC_W8	0.68	0.86	1.04	1.22	1.38
SC84x_Core_W8	0.33	0.33	0.33	0.37	0.37
2156x_Core_W8	0.44	0.63	0.80	0.87	1.03

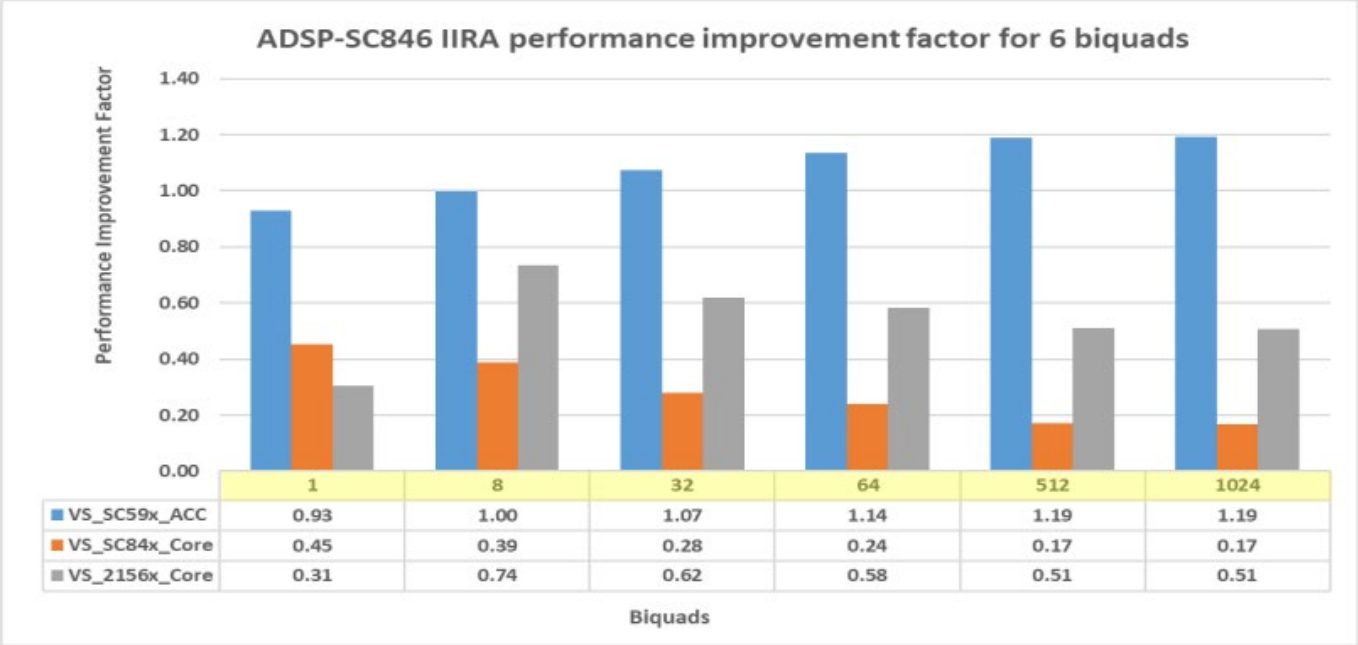
	4	6	8	10	12
SC846_ACC_W32	1.17	1.61	1.99	2.41	2.80
SC59x_ACC_W32	1.26	1.72	2.19	2.66	3.11
SC84x_Core_W32	0.45	0.45	0.45	0.59	0.59
2156x_Core_W32	0.68	0.99	1.28	1.57	1.87

	4	6	8	10	12
SC846_ACC_W64	1.83	2.53	3.26	4.00	4.74
SC59x_ACC_W64	2.03	2.88	3.73	4.58	5.41
SC84x_Core_W64	0.61	0.60	0.61	0.89	0.89
2156x_Core_W64	1.00	1.47	1.92	2.37	2.83

	4	6	8	10	12
SC846_ACC_W512	10.80	16.00	21.20	26.41	31.63
SC59x_ACC_W512	12.78	19.00	25.23	31.55	37.67
SC84x_Core_W512	2.75	2.76	2.75	4.99	4.99
2156x_Core_W512	5.48	8.19	10.88	13.57	16.27

	4	6	8	10	12
SC846_ACC_W1024	21.03	31.33	41.65	52.01	62.34
SC59x_ACC_W1024	25.07	37.44	49.81	62.18	74.53
SC84x_Core_W1024	5.20	5.21	5.20	9.69	9.69
2156x_Core_W1024	10.60	15.87	21.12	26.37	31.63

Figure 7: IIR Performance in Time (µs) Across Window Size and Biquad Values



	1	8	32	64	512	1024
VS_SC59x_ACC	0.93	1.00	1.07	1.14	1.19	1.19
VS_SC84x_Core	0.45	0.39	0.28	0.24	0.17	0.17
VS_2156x_Core	0.31	0.74	0.62	0.58	0.51	0.51

Figure 8: ADSP-SC84x IIR Performance Improvement Factor for 6 Biquads Considering Time

The following observations are derived from Figure 8:

- **The ADSP-SC84x IIRA outperforms theADSP-SC59x IIRA for large window sizes.** The improvement is roughly 1.2x. This is because the ADSP-SC59x IIRA operates at 1 GHz and the ADSP-SC84x IIRA operates at 1.2 GHz.
- **The ADSP-SC84x IIRA outperforms the ADSP-SC84x SHARC-FX execution of IIR code in the core by 0.17x.**
- **The ADSP-SC84x IIRA performance is around 0.42x that of the ADSP-2156x SHARC+ execution of IIR code in the core**. There are three reasons:
  - Each ADSP-SC84x IIRA instance has 1 MAC unit, while each SHARC+ core has 2.
  - The IIRA compute unit takes 6 cycles per biquad filter, and in comparison, the SHARC+ core takes 2.5 cycles.
  - The ADSP-SC84x IIRA operates at 1.2 GHz and the ADSP-2156x SHARC+ core operates at 1 GHz.

Compute Efficiency (CE)

For information on CE, refer to EE-436.

Buffer Placement

For information on buffer placement, refer to EE-436.

Programming the Accelerators

The ADSP-SC84x accelerators are programmable using device drivers provided with the CCES installation package.

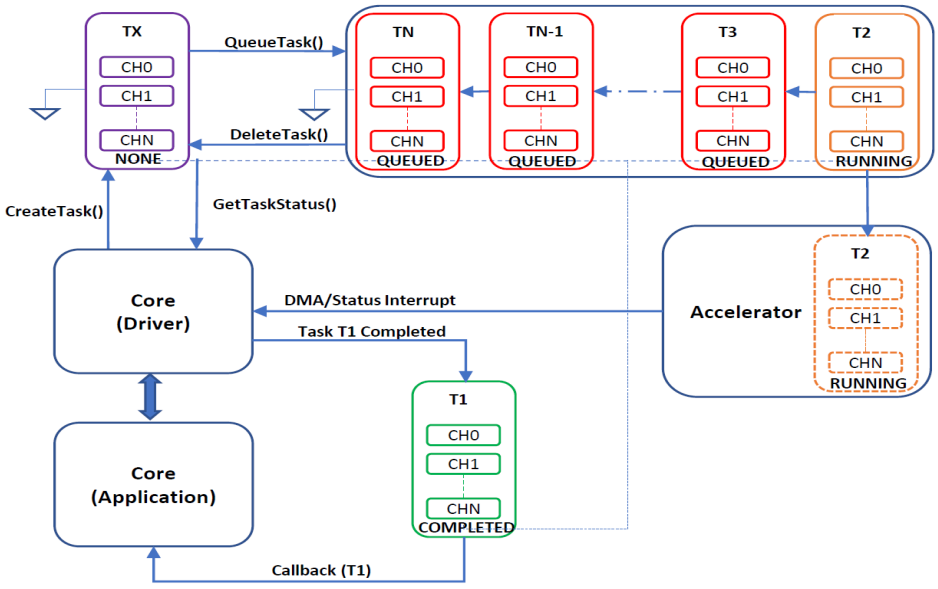


Figure 9: ADSP-SC84x Accelerator Device Driver Structure

The programming model is the same for both the FIR and IIR accelerators. Figure 9 shows the accelerator drivers adhere to a queueing programming model. The module uses abstractions of tasks and queues. A *task* contains a set of *channels* that are processed sequentially when given to the accelerator,. A *queue* is a set of tasks the accelerator processes sequentially and maintains on a first-in, first-processed basis. Applications can create tasks for the accelerator using the `adi_fir(/iir)_CreateTask()` function call and queue those tasks using the `adi_fir(/iir)_QueueTask()` function call.

The driver maintains a processing queue for the tasks given or queued by the application for the accelerator and sequentially schedules the tasks to the accelerator. The applications can register a user-defined callback with an `adi_fir(/iir)_RegisterCallback()` function call. The driver uses the callback to notify the application when the tasks and channels complete their work

The driver maintains the status of all the tasks the application has queued. The application can query the status of the task, when necessary, using the `adi_fir(/iir)_GetFir(/Iir)TaskStatus()` function call. Once the accelerator completes the task in the queue, it is removed from the processing queue. The driver maintains the information for the tasks that are completed or created. The application can later reuse the created tasks by queuing them using the `adi_fir(/iir)_QueueTask()` function call.

The accelerator can be configured using a static configuration to operate in legacy mode or Auto Configuration Mode (ACM). In legacy mode, the driver maintains a software queue (linked list) of the tasks queued by the application and sequentially schedules the tasks to the accelerator. In ACM, the driver uses the hardware halt feature to implement a hardware queuing mechanism Additional channel customized features exist in the ACM, such as interrupts and triggers. For more feature details, refer to the `ADI_FIR(/IIR)_CHANNEL_INFO` topic in the *ADSP-SC84x SSDD API Reference Manual for SHARC-FX*

Core[2] manual provided as part of the CCES help. The driver programming model is the same for both legacy and ACM operation.

The `FIR_Multi_Channel_Processing` and `IIR_Multi_Channel_Processing` code examples provided alongside this document[3] (also available with the ADSP-SC84x EZ-Kit *Board Support Package*), can be used as a reference for the ADSP-SC84x accelerator device drivers.

Table 1 summarizes the accelerator driver benchmarks captured using the ADSP-SC84x EZ-Kit evaluation board. The source code examples used to obtain these performance measurements, `ADSP_SC84x_FIRA_Driver_Benchmark` and `ADSP_SC84x_IIRA_Driver_Benchmark`, can be found in the ZIP file associated with EE-436. Note that these numbers were measured with all buffers in L1 memory.

The driver provides a NOQUEUE mode where only a single task is assigned to the accelerator for processing. It can be run for multiple iterations (Windows) without needing to reload the coefficients into accelerator memory for every iteration. To enable this mode, set the ADI_IIR_CFG_NOQUEUE_MODE macro to 1 in static configuration.

Applications call the `adi_iir_StartTask()` API to assign the task to the accelerator and start processing. The API can be called multiple times to process multiple iterations (Windows) without the need to reload the coefficients for every iteration. Applications call the `adi_iir_UnassignTask()` API to unassign the task from the accelerator.

To enable the slewing feature, set the ADI_IIR_CFG_SLEW_EN macro to 1 in static configuration. To use slewing feature, the NOQUEUE mode in the driver must be enabled.

When creating the task, slewing parameters are configured using the `adi_fir_CreateTask()` API. To update the slewing parameters in an existing task, use the `adi_iir_UpdateTask()` API.

The following is an example of how the application can use the slewing feature:

When an application does not initially require slewing, it can create a task with the ‘nNn’ parameter by setting the ADI_IIR_CHANNEL_INFO macro to 1 to disable slewing. It can then use the `adi_iir_StartTask()` API to process input. To enable slewing during processing, use the `adi_iir_UpdateTask()` API to update the task with the slewing parameters. Then when the application calls the `adi_iir_StartTask()` API, the coefficients are updated using the slewing parameters. It can continue calling this API even after the coefficient slewing is complete. The save state must be disabled during coefficient slewing.

Operation	**Description**	**Measured Core Cycles**			
		FIRA		**IIRA**	
		Legacy	**ACM**	**Legacy**	**ACM**
CreateTask()	Constant Overhead	124	140	158	141
	Per Channel Overhead	198	221	206	225
QueueTask()	Pushing a task to the empty queue	231	512	217	497
	Pushing a task to the non-empty queue	116	339	102	325

Operation	Description	Measured Core Cycles			
		FIRA		IIRA	
		Legacy	ACM	Legacy	ACM
Interrupt Overhead[1]	For a single task in queue	748	749	741	307
	For more than one task in queue	597	616	816	840

Table 1: ADSP-SC84x Accelerator Driver Benchmarks

Additional overhead for cache flushing/invalidation can occur when the buffers are placed in L2 and L3 memories. Analog Devices recommends placing the buffers in L1 memory to avoid cache overhead.

Accelerator Usage Models

For more information on accelerator usage models refer to EE-436.

Real-Time Implementation of an Example FIR/IIR Use Case

Figure 10 shows an example of FIR/IIR real-time processing of 12-channel audio data.

Note the following details in Figure 10:

- Each channel passes through a 1024-tap FIR filter followed by a five-band IIR equalizer.
- For simplification, the FIR filter is an all-pass filter (the b0 coefficient is one, and all other coefficients are zero).
- Each IIR band is an 8th order IIR filter implemented with four cascaded biquad stages. The IIR band details are as shown in Table 2.

  The output of each band is multiplied with the respective gain (from 0.0 min to 1.0 max) and added together to generate the final output. The *IIRGains* flag is set (bypass) or cleared (enable processing) dynamically using the CCES memory browser.
- The block size is 256 samples per channel.
- All buffers are placed in internal (L1) memory.

[1] Round trip cycles include the SEC interrupt dispatch latency.

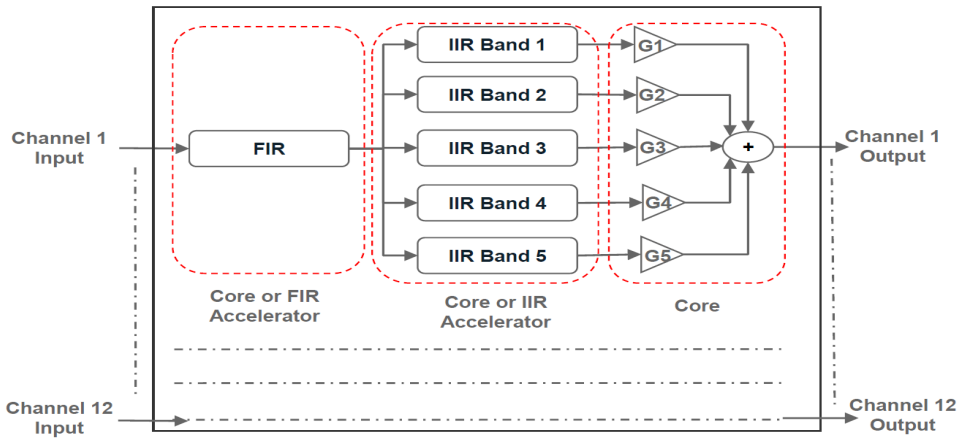


Figure 10: FIR/IIR Real-Time Use Case

Band No.	Lower Cutoff in Hz	Higher Cutoff in Hz	Filter Type
1	-	250	Low Pass
2	250	500	Band Pass
3	500	2000	Band Pass
4	2000	4000	Band Pass
5	4000	-	High Pass

Table 2: IIR Bands Details

The `Direct_Replacement`, `Pipelined`, and `Split_Task example code` supplied with this document [3] implements the RIF/IIR real time use case on the EV-SC846-SOM evaluation board for the different usage models.

`Table` 3 shows the MIPS of the core and accelerator for different usage models along with the core MIPS savings.

From a performance perspective, the *Data Pipelining* model is the best, resulting in saving approximately 94 core MIPS, followed by the *Split Task* model (saving 40 core MIPS) and then the *Direct Replacement* model (saving -5 core MIPS). These results align with the accelerator usage model expectations outlined in EE-436.

Usage Model	MIPS Usage			Core MIPS Saving
	Core	FIRA	IIRA	
Core Only	95	0.00	0.00	-
Direct Replacement	100	78	20	-5
Split Task	55	39	14	40

Data Pipelining	1	78	20	94

Table 3: MIPS Summary for Different Accelerator Usage Models

Conclusion

This application note discusses the performance numbers of the FIR/IIR accelerators on the ADSP-SC84x in comparison with other processors. It also discusses the device drivers provided with CCES installation and the different usage models. A use case is implemented using different usage models and the MIPS numbers are summarized.

References

[1] *ADSP-2184x/ADSP-SC84x SHARC-FX Processor Hardware Reference*, Rev 0.2, January 2026. Analog Devices, Inc.

[2] *ADSP-SC84x SSDD API Reference Manual for SHARC-FX Core*, Version 1.0, CrossCore® Embedded Studio Help.

[3] *Associated ZIP file for EE-482: Using ADSP-SC84x High Performance FIR/IIR Accelerators,* May 2026. Analog Devices, Inc.

[4] *DSP-21591_21593_21594_ADSP-SC591_SC592_SC594 Anomaly List for Revision(s) 0.0 (Rev E, 07-07-2022)*

[5] *EE-436: Using ADSP-SC59x/2159x High Performance FIR/IIR Accelerators,* September 2022. Analog Devices, Inc

Document History

Revision	Description
Rev 1 – July 2026 *by Sanket Nayak*	Initial Release