Boot Time Estimation for ADSP-SC59x/ADSP-2159x SHARC+ Processors

Contributed by Saikia, Juganta *Rev 1 – August 23, 2021*

Introduction

A faster boot time creates the first impression to an end user about the device. In automotive infotainment application, users are always concerned about *'Time to Audio'* when booting their application. This EE-note will guide the users to estimate the boot time in ADSP-SC59x/ADSP-2159x SHARC+® processors accurately, which is useful information to design an application and define the boot strategy to adopt, bringing down the overall system boot-up time.

Boot time in ADSP-SC59x/ADSP-2159x SHARC+ processors depend on numerous factors which includes boot mode used, size of the loader streams and different types of blocks present inside the loader stream. This EE-note explains the effects of all these factors and provides an estimation of boot time in ADSP-SC59x/ADSP-2159x SHARC+ processors. Boot time comparison with previous products (ADSP-SC57x and ADSP-2156x processor families) has also been documented here, which will help the users to choose the right processor based on their application's boot time requirement.

This guide describes:

- Boot Time Estimation
- Boot Time Non-Linearity
- Factors affecting Total Boot Time
- Secure Boot Time Enhancement
- Comparison of Boot Time of various processors

Terminology

The following is a list of most used terms in this EE-note:

- ECDSA - Elliptical Curve Digital Signature Algorithm
- BLp - Boot Loader plaintext, Plaintext Format
- BLx - Boot Loader without key, Keyless Format
- BLw - Boot Loader wrapped, Wrapped Format
- STR - Single transfer rate
- DTR - Dual transfer rate
- GPIO – General-purpose input/output

Further explanation will be provided in the following sections.

Boot Time

Total boot time in ADSP-SC59x/ADSP-2159x SHARC+ processors is the sum of preboot time of the processor, and the boot time consumed for loading the application in a given boot mode.

- Preboot time is responsible for configuration of all system resources prior to executing the required boot operation.
- Application loading time will be dependent on the nature of loader stream as well the peripheral being used for a particular boot mode.

Several factors affect the boot time in a processor, out of which, size of the loader stream and presence of fill blocks and init block inside the loader stream impact the most. This EE-note explains, how we can estimate the boot time accurately for a given loader stream by considering the impact of these factors.

Boot Time Estimation

Considering the preboot takes a fix amount of time under certain conditions, application loading time (boot time) can be computed separately for any loader stream using Boot ROM API. Boot time has been captured with different sizes of loader streams in each boot mode, based on the which, linear plots are generated for different boot modes supported in ADSP-SC59x/ADSP-2159x SHARC+ processors with maximum system frequency (1GHz CCLK and 500MHz SYSCLK), where the boot time varies linearly with the size of the loader stream. Using these plots as shown in Figure 1 , linear equations have been extrapolated in the form of $y = ax + b$, where x is size of loader stream in kilo bytes and y is boot time in milli seconds. Using these equations, one can estimate the boot time for any loader stream of any size.

The associated .zip file has boot application code ‘Application_Code’, which toggles a GPIO pin at start up to measure boot time. Similarly, example code ‘BootTimeMsmt_ROMAPI’ has been included which calls the ROM API boot.

Hardware Setup Used

EV-SC594-SOM-EZKIT, EV-21593-SOM-EZKIT and EV-SOMCRR-EZKIT were used for all the experiments done at our end.

ADSP-SC59x/ADSP-2159x SHARC+ Processors support normal as well as secure boot. Secure boot support integrity, authentication and confidentiality in three formats for protecting the IP and data:

- Plaintext (BLp) Format
- Wrapped (BLw) Format
- Keyless (BLx) Format

For more information on secure boot, please refer to the Secure Booting Guide for Blackfin+® and SHARC+® Processors (EE-366). Rev 2, October 19,2016[1].

Secure boot with BLw and BLx format in ADSP-SC59x/ADSP-2159x SHARC+ does not support the application mapped to off chip RAM space. '-chipmemsplitter' switch in CCES elf loader utility can be used, which causes additional loader file outputs for on-chip and off-chip memory to be produced along with the usual loader file output. The on-chip loader stream can be booted with BLw/BLx format, and off-chip loader stream can be booted with BLp format in such case. Example is attached with this EE-note. This is incorporated for larger size loader streams in the linear equations provided below, where the application data is mapped to off chip RAM space. Please refer to CrossCore® Embedded Studio 2.9.0 Loader and Utilities Manual[2] for more information on CCES elf loader utility.

Below sections will have the boot time linear equations for different boot modes supported in ADSP-SC59x/ADSP-2159x SHARC+ processors.

SPI Flash Boot

SPI Flash Boot (SYS_BMODE 1) mode supports booting from flash device through SPI peripheral. In ADSP-SC59x/ADSP-2159x SHARC+ processors, SPI2 instance drives the default SPI boot, which supports single, dual and quad STR modes. Table 1 shows different BCODEs supported for maximum frequency as per IS25LP512M flash device, which is present on EV-SC594-SOM-EZKIT/EV-21593-SOM-EZKIT.

Boot Mode	Max SPI Clock	BCODE Value
Single STR	75 MHz	0x2
Dual STR	75 MHz	0x3, 0x5
Quad STR	75 MHz	0x4, 0x9

Table 1: Supported BCODE values for SPI flash boot

Different BCODEs are provided to support different I/O modes of different flash devices depending on number of dummy cycles needed for the flash. Please refer to SPI Master BCODE Configuration Lookup Table in the ADSP-2159x/ADSP-SC59x SHARC+ Processor Hardware Reference Manual[3] for all the details.

In this section, linear boot time equations are provided for all the SPI flash boot modes at 62.5 MHz SPI clock including normal boot and secure boot with ECDSA-256 authentication.

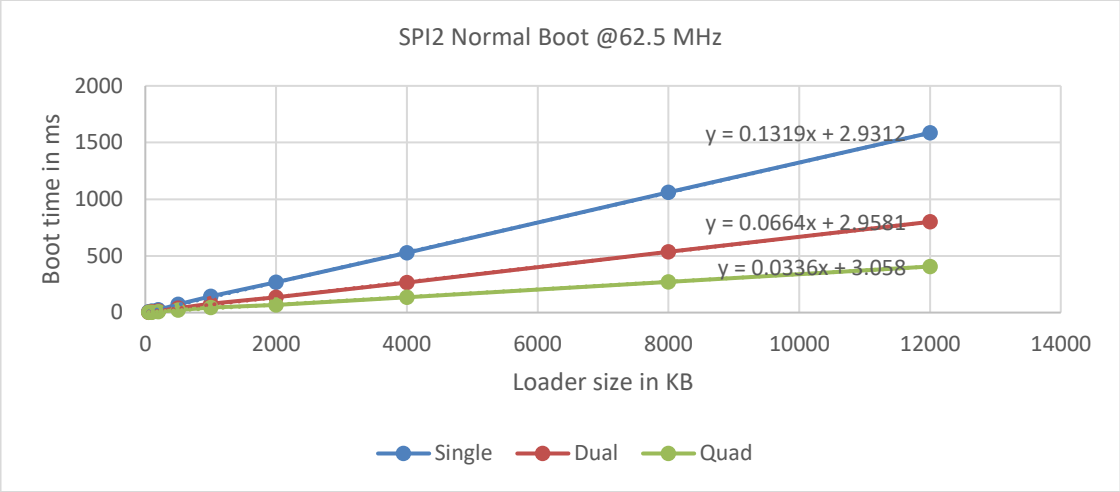

Figure 1: SPI2 Flash Normal boot at 62.5 MHz in ADSP-21593

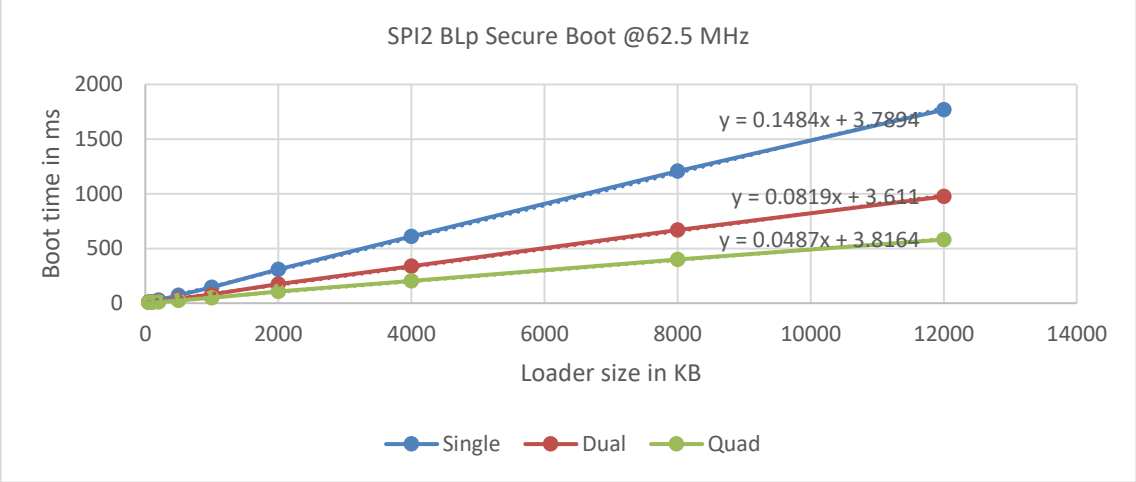

Figure 2: SPI2 Flash BLp boot at 62.5 MHz in ADSP-21593

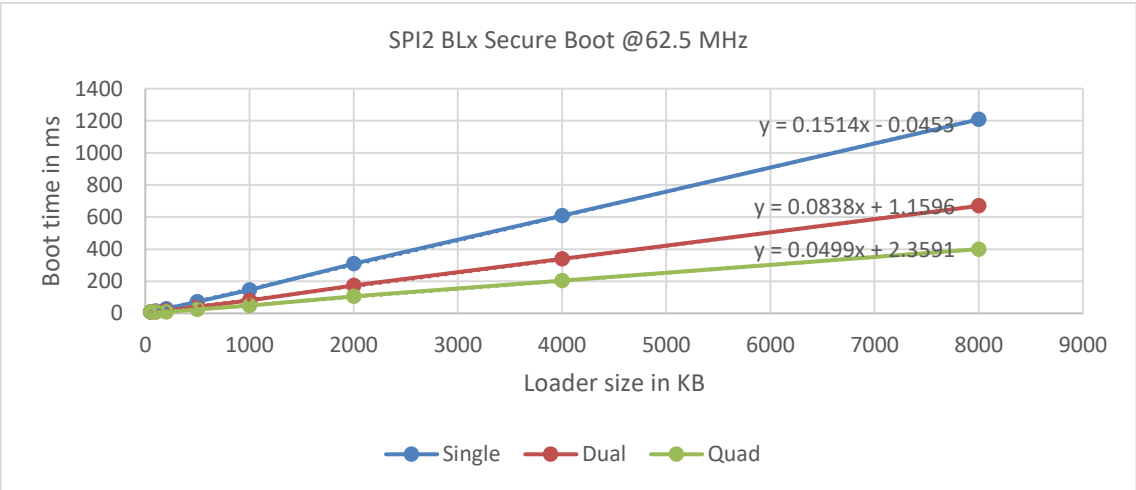

Figure 3: SPI2 Flash BLx boot at 62.5 MHz in ADSP-21593

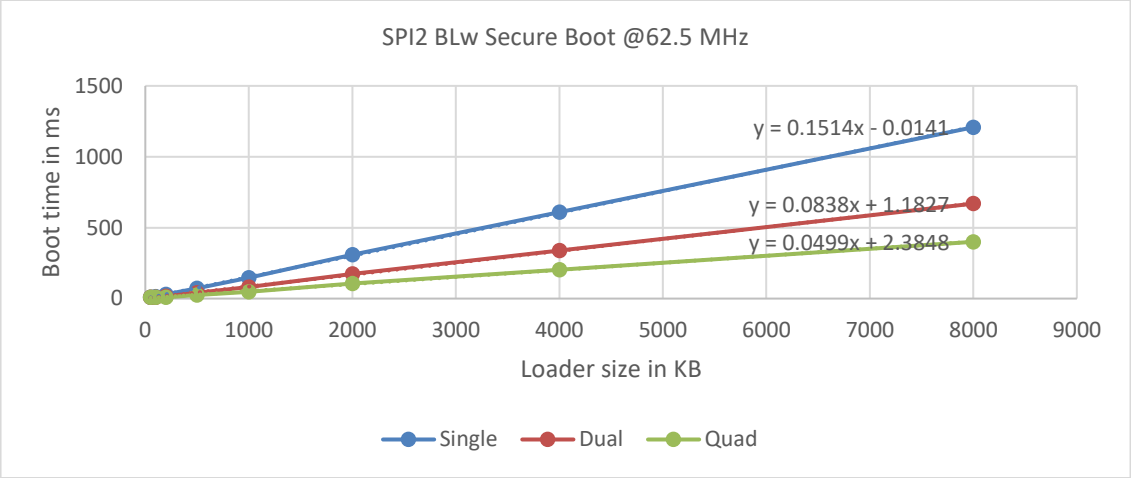

Figure 4: SPI2 Flash BLw boot at 62.5 MHz in ADSP-21593

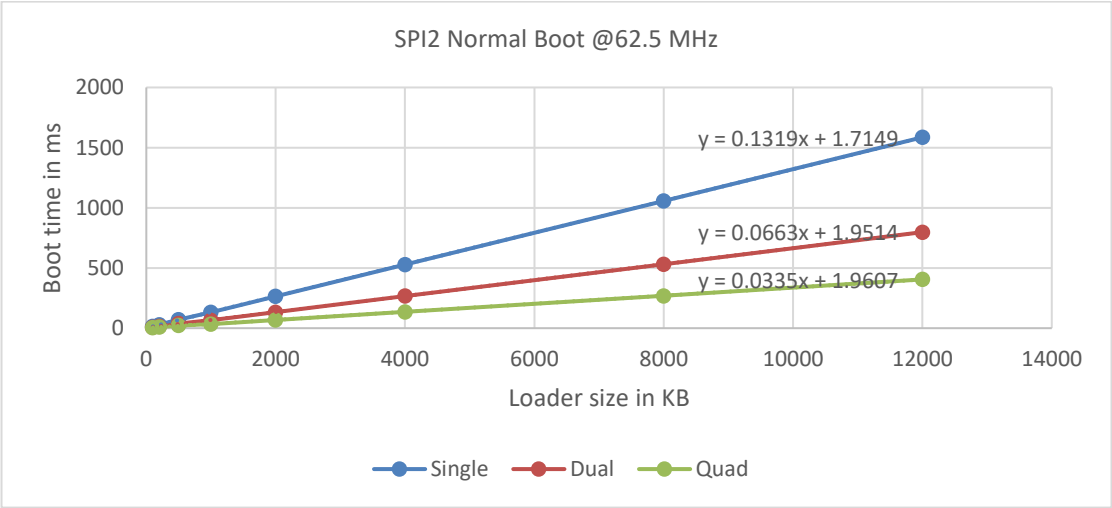

Figure 5: SPI2 Flash Normal boot at 62.5 MHz in ADSP-SC594

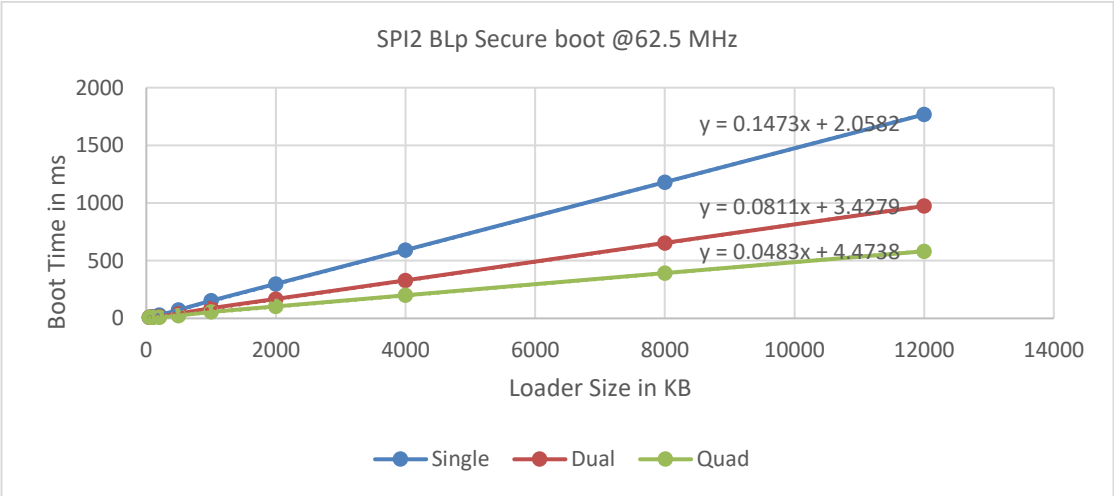

Figure 6: SPI2 Flash BLp boot at 62.5 MHz in ADSP-SC594

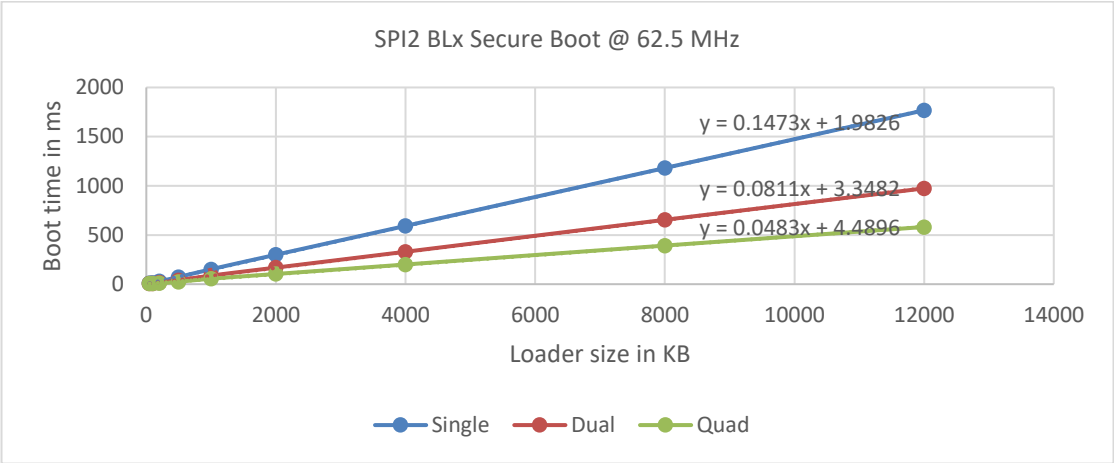

Figure 7: SPI2 Flash BLx boot at 62.5 MHz in ADSP-SC594

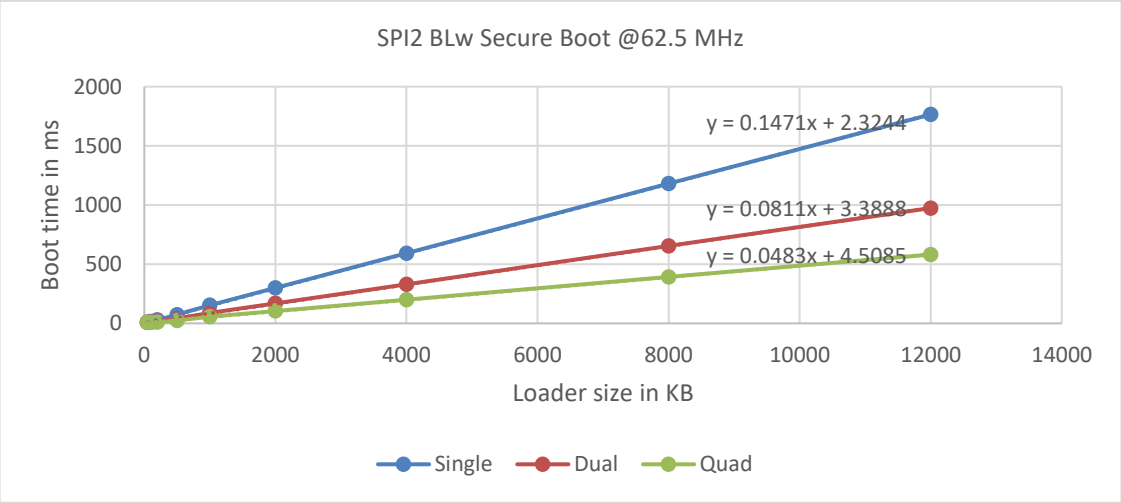

Figure 8: SPI2 Flash BLw boot at 62.5 MHz in ADSP-SC594

Due to the authentication and decryption routines performed by the Crypto Engine in secure boot, difference in normal and secure boot time is observed.

The above numbers can be scaled to max 75 MHz SPI Clock accurately, which will further improve the boot time as per Table 2 below:

Condition with SPI single STR boot	Boot time in ms
Estimated Boot time at 62.5 MHz	68.88
Estimated Boot time after linear scaling at 75MHz	57.4
Measured boot time at 75 MHz	58.58

Table 2: Boot time conversion for Max SPI clock supported

A small margin can be expected, as there will be a tradeoff in CCLK and SYSCLK while attaining 75MHz SPI CLK.

OSPI Flash Boot

OSPI Flash Boot (SYS_BMODE 5) supports booting from SPI flash or Octal flash device through the OSPI peripheral which includes single-STR, dual-STR, quad-STR, single-DTR, dual-DTR and quad-DTR modes. OSPI boot supports maximum of 62.5MHz OSPI Clock. ADSP-2159x processor family supports a single flash device, where SPI and OSPI signals share the same pin multiplexing, whereas ADSP-SC59x processor family supports two flash devices at a time, where SPI and OSPI signals are on different port pins.

Table 3 shows different BCODEs supported for OSPI flash boot with maximum frequency as per IS25LP512M SPI flash device on EV-21593-SOM. OSPI flash boot time data is available only for ADSP 21593 processor.

Boot Mode	Max OSPI Clock	BCODE Value
Single STR	62.5 MHz	0x2
Dual STR	62.5 MHz	0x3
Quad STR	62.5 MHz	0x5
Single DTR	62.5 MHz	0x8
Dual DTR	62.5 MHz	0xA
Quad DTR	62.5 MHz	0xC

Table 3: Supported BCODE values for OSPI flash boot

Different BCODEs are provided to support different I/O modes of different flash devices depending on number of dummy cycles needed for the flash. Please refer to OSPI Master BCODE Configuration Lookup Table in the ADSP-2159x/ADSP-SC59x SHARC+ Processor Hardware Reference Manual[3] for all the details.

In this section, linear boot time equations are provided for all the OSPI flash boot modes at 62.5 MHz SPI clock including normal boot and secure boot with ECDSA-256 authentication.

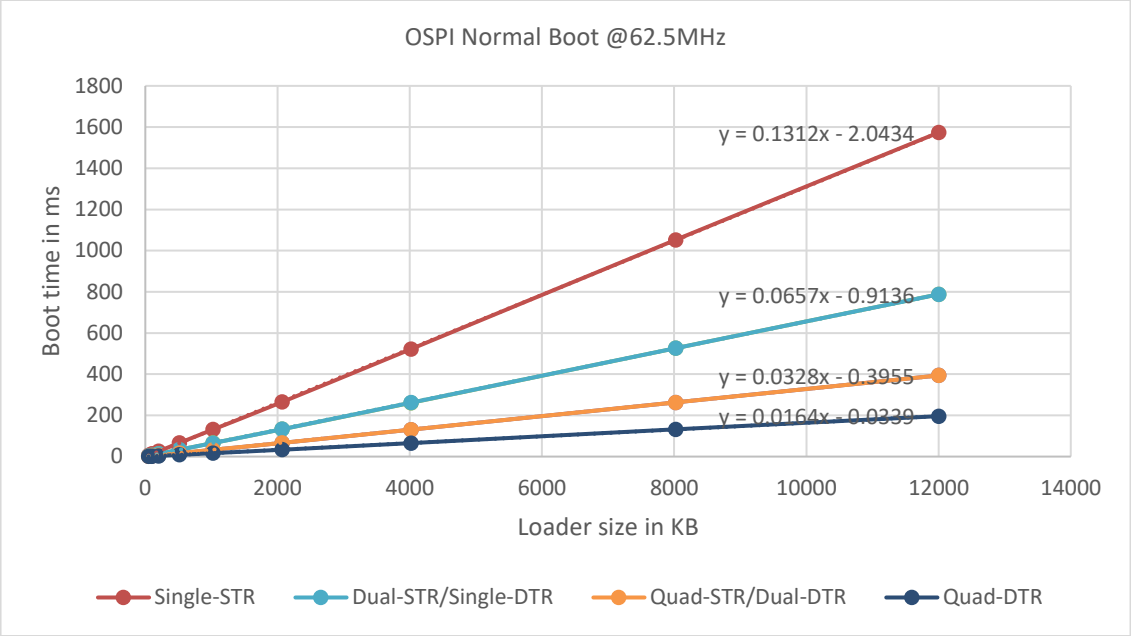

Figure 9: OSPI normal boot at 62.5 MHz

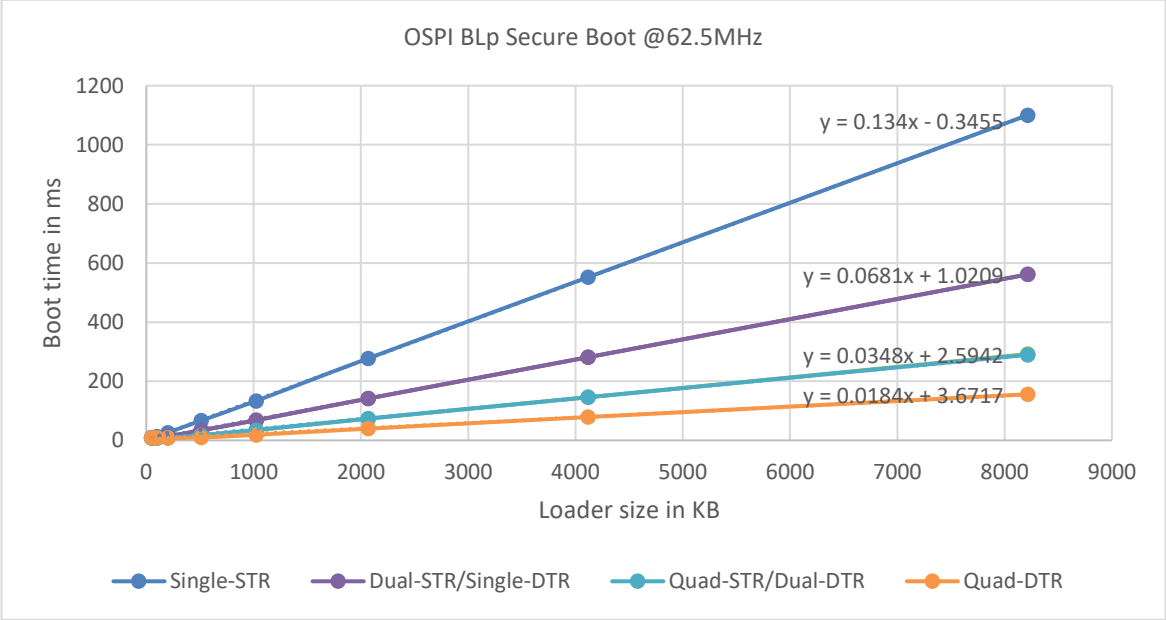

Figure 10: OSPI BLp secure boot at 62.5 MHz

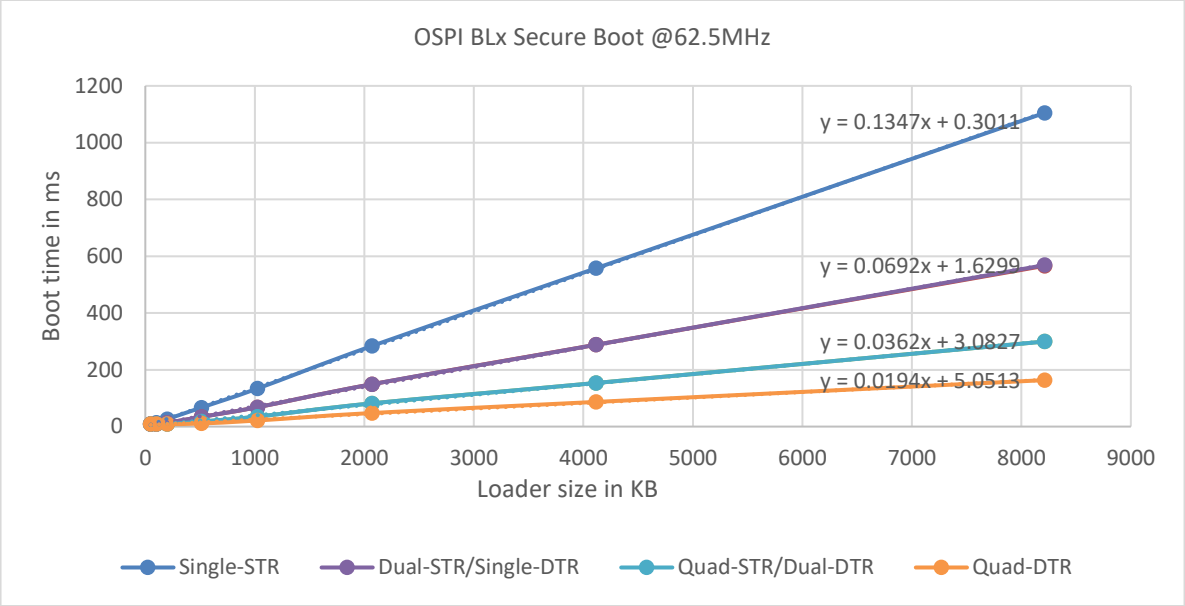

Figure 11: OSPI BLx secure boot at 62.5 MHz

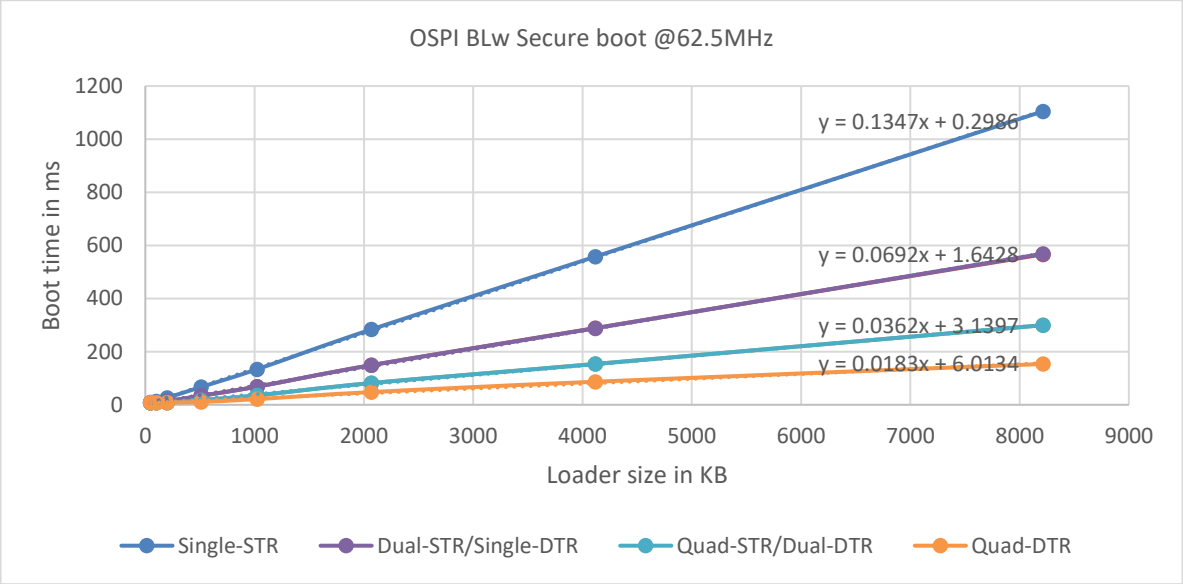

Figure 12: OSPI BLw secure boot at 62.5 MHz

There is an extra overhead present in DMA transfer in SPI secure boot, where MSIZE always remains 2 so that the MDMA channels operates at all the 2 byte aligned page addresses due to the presence of the secure header of 212 bytes. This will add an extra delay in each 32-bit DMA transfer. Due to which, SPI secure boot numbers are more than OSPI secure boot.

OSPI controller in ADSP-SC59x/ADSP-2159x SHARC+ processors do not support Octal boot by default. But it can be supported through a secondary stage boot, which improves the boot time as compared to Quad-

STR or Dual-DTR modes. This can be achieved by using a custom boot implementation or a hook function. Example code (Test Code Name: BootTimeMsmt_ROMAPI) has been attached with this EE-note for octal boot.

Figure 13 shows the Octal-STR boot timings with normal and secure (BLp256 format) boot, which was captured with the MX66LM1G45GMI00 flash device present on EV-SOMCRR-EZKIT.

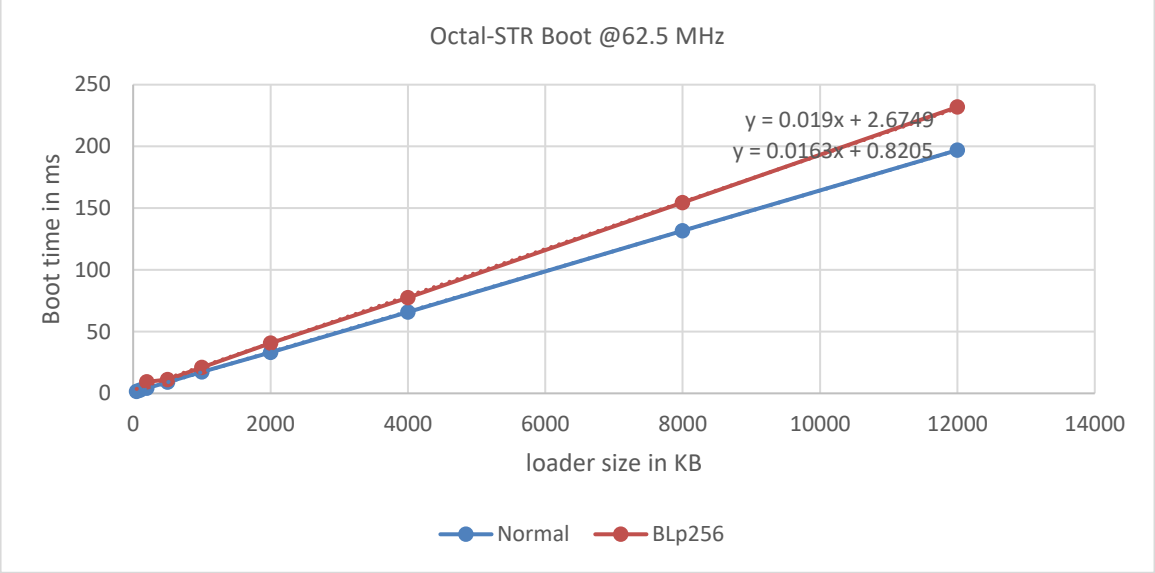

Figure 13: OSPI Octal-STR boot at 62.5 MHz

Linkport External Host Boot

Link port boot (SYS_BMODE 4) is an external host boot mode in which the processor receives boot data from an external link port host through link port 0. The link port is configured as receiver with 8-bit SDR, where host also needs to send the data in same mode. All transfers from the link port to memory are performed under the control of DMA. The maximum supported operating frequency of the link port is 125 MHz for which the host boot source is responsible for driving the clock frequency. This can be achieved by using init code or OTP to configure the CGU for max clock supported (CCLK 1GHz, SYSCLK 500MHz and SCLK0 125MHz). The link port receiver operates at an asynchronous frequency up to the maximum supported operating frequency.

Linear boot time equation for Linkport normal boot at 125 MHz LP CLK is provided below (Figure 14).

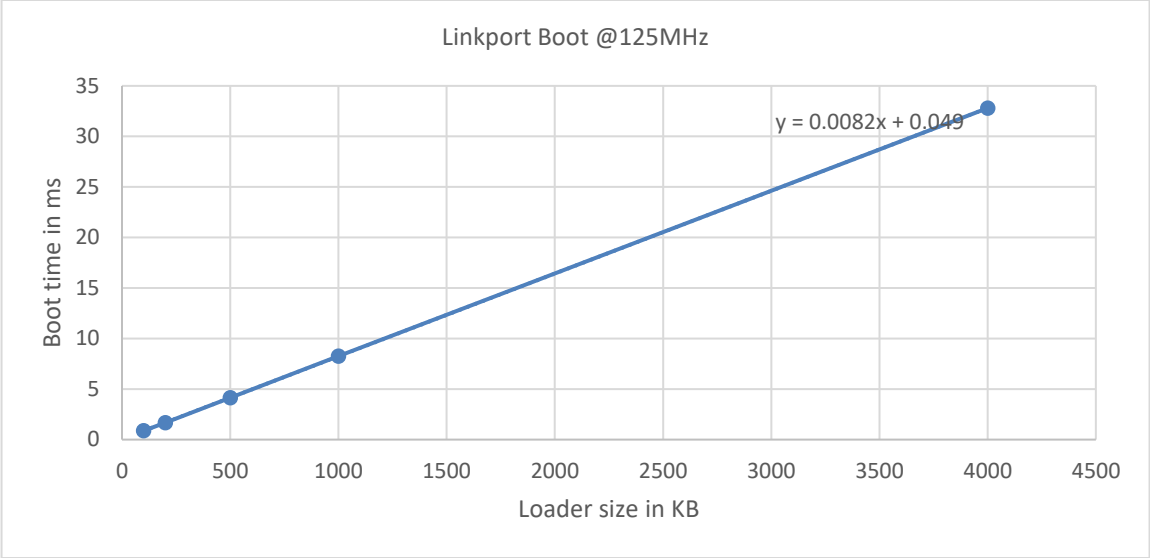

Figure 14: Linkport Normal External Host Boot at 125 MHz in ADSP-SC594/ADSP-21593

UART External Host Boot

UART External Host Boot (SYS_BMODE 3) in ADSP-SC59x/ADSP-2159x SHARC+ processors is an external host boot mode, where the processor receives boot data from a UART host device connected to the UART interface. UART0 is the default booting peripheral. Maximum UART Clock supported for UART boot is 7.8 MHz which can be achieved by using init code or OTP to configure the CGU for max clock supported (CCLK 1GHz, SYSCLK 500MHz and SCLK0 125MHz). As UART is a slow speed peripheral, boot time for normal and secure boot is identical, where boot time is mainly dependent on the loading time of the peripheral.

Linear boot time equation for UART normal boot at 7.8 MHz UART CLK is provided below (Figure 15).

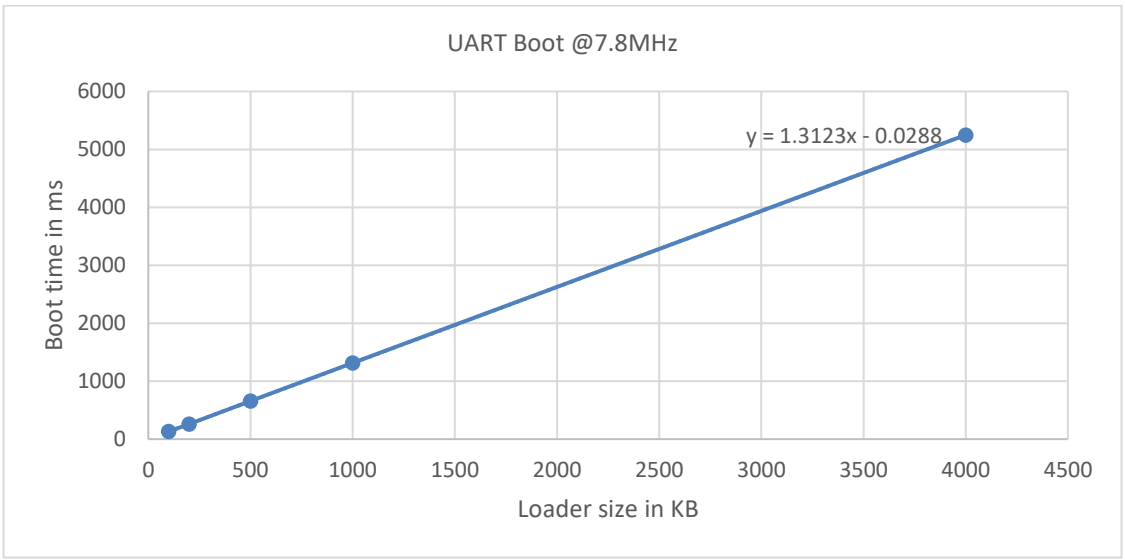

Figure 15: UART External Host Boot at 7.8 MHz in ADSP-SC594/ADSP-21593

Boot Time Non-Linearity

All the plots shown in earlier sections provide linear equations for boot time calculation over different sizes of loader stream. But linearity is not consistent when the loader stream is of lower size for both non secure and secure boot. This section shows how boot time in small loader stream follows nonlinearity in terms of non-secure boot and reach saturation in terms of secure boot.

Nonlinearity in Non-Secure Boot Time

Figure 16 shows that, the boot time for SPI Quad Flash boot (BCODE 9) with smaller size loader stream of size less than 10 KB is nonlinear, where estimating the boot time using the equations given in the earlier sections will not be very accurate. Therefore, it is advisable for users to test the exact boot time for smaller loader streams, which are of less than 10KB.

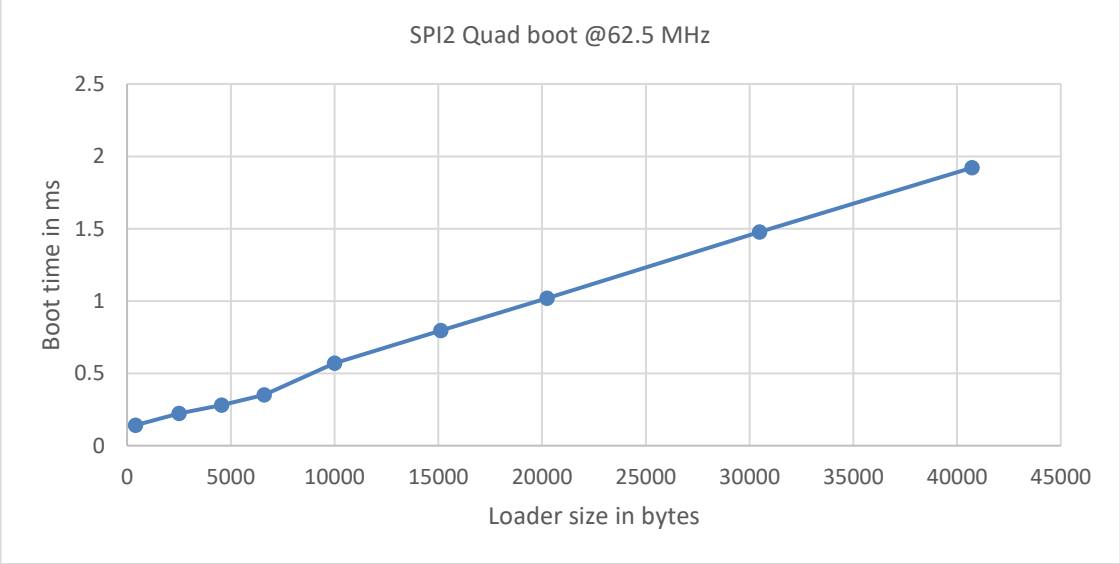

Figure 16: SPI2 Master boot time non-linearity at 62.5 MHz in ADSP-SC594 at CCLK=1GHz and SPI CLK=62.5MHz

Nonlinearity in non-secure boot time is not due to any issue in Boot ROM, but due to the less sample size measured for smaller loader stream.

Non-Linearity in Secure Boot Time

Authentication routine, which consumes a considerable amount of time in the crypto engine, plays an important role in deciding the total secure boot time. In previous products, Boot ROM had to wait for the whole loader stream to be loaded for the crypto engine to generate the signature from the computed SHA hash digest, which is needed for the authentication routine to be completed. Authentication process is a bottleneck for lower boot image, which decides the total secure boot time. Due to the sequential process, total time consumed by the authentication process in earlier products was more, which affected the total secure boot time.

In ADSP-SC59x/ADSP-2159x SHARC+ processors, this issue has been resolved to an extent where pre-calculated SHA hash digest in the secure header is utilized to start authentication process parallelly in the

booting process while loading the full image for processing hash and decryption. Due to which in large boot images, secure boot time gets completely independent of the authentication routine. For smaller boot images, where loading time is less than authentication routine completion time, secure boot time reaches a saturation point, which is completely decided by the authentication routine due to the parallel operation.

Figure 17 shows that the saturation level for SPI BLw secure quad boot (With CCLK 1GHz, SYSCLK 500MHz and SPI CLK 62.5MHz) hits at 200 KB, and the boot time at the saturation point is near 9 ms. In this case, linear equation for computing boot time in BLw secure boot as plotted in the earlier sections will support for loader stream of size larger than 200KB.

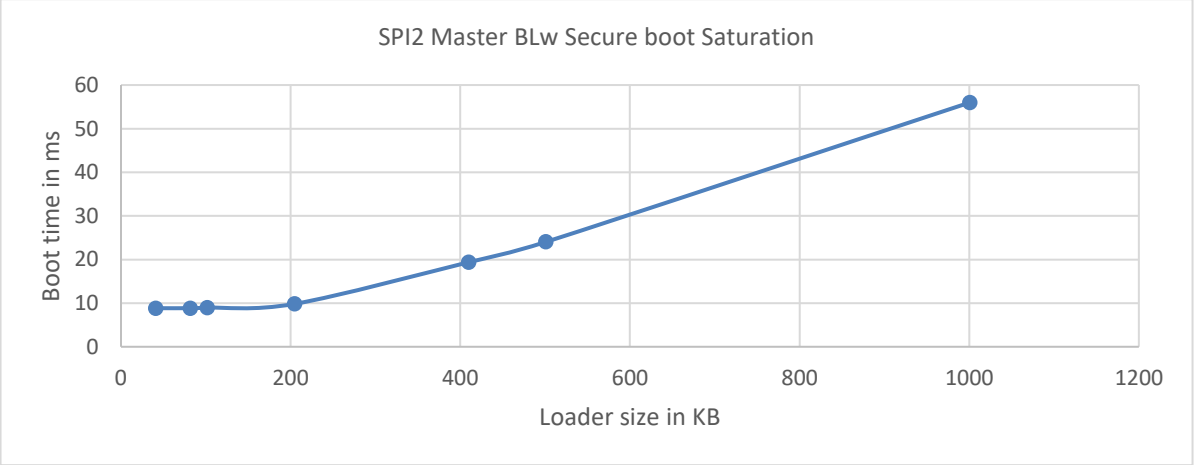

Figure 17: SPI2 secure quad (BCODE 9) boot time saturation at 62.5 MHz in ADSP-SC594 at CCLK=1GHz and SPI CLK=62.5MHz

Table 4 gives the approximated saturation points for secure boot with ECDSA-256 authentication in all the flash boot modes at max frequency (CCLK 1GHz, SYSCLK 500 MHz and SPI/OSPI CLK 62.5 MHz). Boot time at below saturation in all these cases is approximately 9 ms.

Boot Mode	Saturation Point
SPI Master Single Mode	50 KB
SPI Master Dual Mode	100 KB
SPI Master Quad Mode	200 KB
OSPI Master Single-STR	50 KB
OSPI Master Dual-STR	100 KB
OSPI Master Quad-STR	200 KB
OSPI Master Single-DTR	100 KB
OSPI Master Dual-DTR	200 KB
OSPI Master Qual-DTR	400 KB

Table 4: Saturation point for different flash boot modes.

As the crypto engine runs on SYSCLK, the above saturation points will vary as the frequency changes. For example, if SYSCLK changes from 500MHz to any lower value, authentication routine completion time will increase, which will also result in an increase in saturation point.

Factors Affecting Total Boot Time

Boot time in ADSP-SC59x/ADSP-2159x SHARC+ processors is primarily affected by the size of the application. All the linear equations provided in the earlier sections consider only the loader stream size. There are some other factors including preboot time, fill blocks, init block etc. which will also affect the total boot time. These factors are explained in detail in the following sections, which will make the total boot time estimation for a given application more accurate and robust.

Pre-Boot Time

As mentioned in earlier section, pre-boot time counts for configuration of all system resources prior to executing the required boot operation, which includes:

- Core Initialization
- SPU & SMPU configuration
- Secure debug key processing
- CGU configuration
- DMC configuration
- Fault Configuration
- L1 memory initialization etc.

CGU configuration and DMC initialization routines impact the preboot execution time when OTP is programmed. Please refer to ADSP-SC59x SHARC+ Processor Hardware Reference Manual[3] for more details in preboot. Table 5 shows preboot time under different conditions:

Condition	Preboot time in ADSP-SC594 (ms)	Preboot time in ADSP-21593 (ms)
Default	2.415	2.049
CGU configuration enabled (OTP programmed for 1GHz CCLK)	2.344	2.002

Table 5: Preboot time at different conditions

OTP programming for DMC initialization in preboot should not be used as per Anomaly :20000117. It can be performed either through initcode routine or multistage boot.

Effect of Fill Block in Boot Time Estimation

A fill block instructs the boot kernel to perform a 32-bit memory fill of the memory region. Fill blocks help to minimize the size of a boot stream when an application contains large arrays of data that needs to be initialized at startup of the application. The size of a loader stream does not depend on this block, as this block is of only 16 bytes. But having multiple fill blocks in a fixed size loader stream does affect the total boot time which includes the extra time to process each fill block header as well as time required for DMA configuration in each block.

All the boot time equations provided in previous sections were computed considering there were no fill blocks present in the loader stream. Figure 18 provides the linear equation based on testing with SPI quad boot (BCODE 9) at 62.5MHz SPI CLK, which can be used to compute the extra boot time taken due to the presence of fill blocks. Currently the equation is extrapolated for all the master boot modes which uses MDMA channels, assuming fill block processing time is independent of peripheral loading time. More data will be added to make it generalized across all the boot modes in future revision of this EE-note.

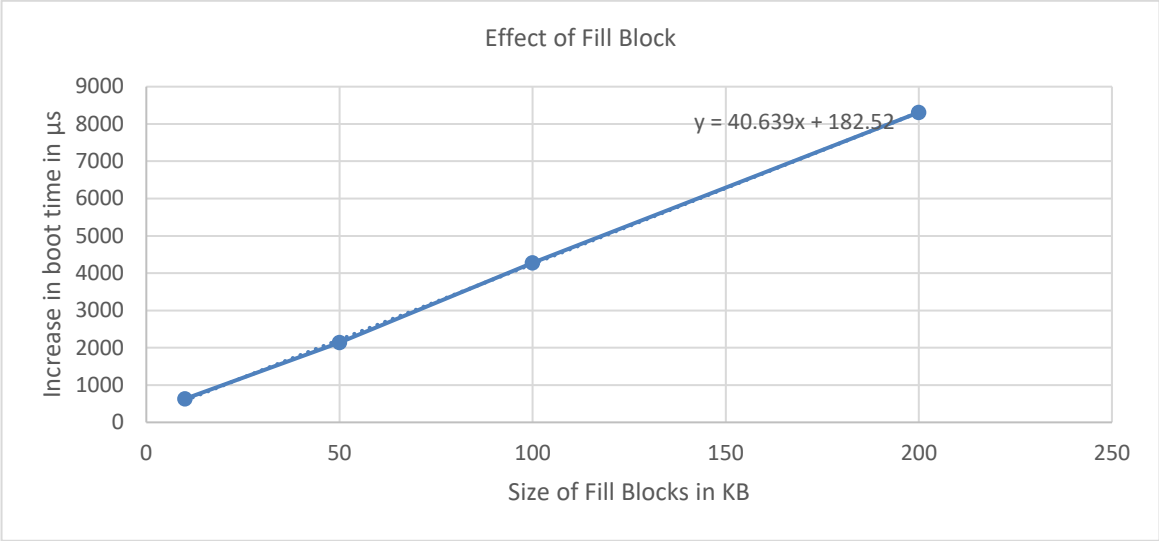

Figure 18: Extra Boot time consumed due to the presence of fill blocks

Above plot gives the measurement for 10 fill blocks present at a time in the loader stream. For any other number of fill blocks, the boot time numbers can be scaled by the same factor.

Accurate Boot Time Estimation in Presence of Fill Blocks

This section provides an example with experimental results, which shows how accurately we can estimate the boot time for a 3 MB loader stream in the presence of fill blocks. Considering the loader stream contains 20 no. of 10KB fill blocks, 15 no. of 50KB fill blocks, 10 no. of 100KB fill blocks and 5 no. of 200KB fill blocks, below equations and Table 6 show the estimated as well as measured boot time with SPI master quad boot at 62.5MHz SPI CLK for the given loader stream.

Estimated boot time without considering fill blocks = $0.0336x + 3.058 = 0.0336 * 3090 + 3.058 = 106.88$ ms (Refer Figure 1)

Extra Boot time consumed in Processing 20 10KB fill blocks = $2 * (40.639x + 182.52) = 2 * (40.639 * 10 + 182.52) = 1177.8\ \mu s$ (Refer Figure 18)

Extra Boot time consumed in Processing 15 50KB fill blocks = $1.5 * (40.639x + 182.52) = 1.5 * (40.639 * 50 + 182.52) = 3321.7\ \mu s$ (Refer Figure 18)

Extra Boot time consumed in Processing 10 100KB fill blocks = $(40.639x + 182.52) = (40.639 * 100 + 182.52) = 4246.4\ \mu s$ (Refer Figure 18)

Extra Boot time consumed in Processing 5 200KB fill blocks = $0.5 * (40.639x + 182.52) = 0.5 * (40.639 * 200 + 182.52) = 4155.1\ \mu s$ (Refer Figure 18)

Condition	Result
Estimated Boot Time without considering Fill blocks	106.88 ms
Estimated Boot Time considering Fill blocks	119.79 ms
Measured Boot Time	118.6 ms
Accuracy without considering Fill blocks	89.03%
Accuracy considering Fill blocks	99%

Table 6: Accurate Boot Time estimation example

This shows that, in presence of multiple fill blocks, estimation of boot time by only considering the loader stream size factor may not be very accurate, whereas taking fill blocks into consideration makes the boot time estimation more accurate. Accuracy has been improved from 89.03 % to 99% as shown in the above example.

Larger loader stream will always add an extra penalty for consuming more space in the external flash, which will also increase the overall system cost. Fill blocks in such cases, help in reducing the loader stream size at the cost of risking the total boot time. There should be a good trade-off between boot time and size of the loader stream.

Effect of Init Block

An initialization or init block instructs the boot kernel to perform a function call to the target address after the entire block has loaded. The function called is referred to as the initialization code (Initcode) routine. Initcode routines can be used to speed up and customize booting mechanisms exposed by the boot kernel. Traditionally, an Initcode routine is used to set up the system PLL, bit rates, wait states, and the external memory controllers. Boot time can be significantly reduced when an init block is executed early in the boot process.

Init Code execution time depends on all the features that are enabled inside the initcode application, which will impact the total boot time. For more information on implementation of init block, please refer to ADSP-2159x/ADSP-SC59x SHARC+ Processor Hardware Reference Manual[3].

Initcode routines are not supported in secure boot scenario.

Secure Boot Time Enhancement

As mentioned in earlier section, secure boot architecture has been updated to speed up the authentication process in ADSP-SC59x/ADSP-2159x processor families. This is done by using the pre-calculated SHA hash digest in the secure header to start authentication process early in the booting process without having to wait for the full image to be loaded. Whereas in previous products, Boot ROM had to wait for the whole loader stream to be loaded for the crypto engine to generate the signature from the computed SHA hash digest, which is needed for the authentication routine to be completed.

This will ultimately save some additional time in case of secure boot, which will be beneficial to users, where they want to have a secondary stage booting and want to minimize the boot time for the primary boot application. As in such cases, primary application is of smaller size, and the authentication routine in crypto engine with pre-calculated hash will primarily decide the boot time (Saturation point as mentioned in earlier section).

Table 7 shows that there is improvement of ~3ms when tested across SPI2 quad boot (BCODE 9) at 62.5MHz SPI CLK with a 200 KB loader stream.

Processor Variant	BLp256 Secure Boot Time in ms
ADSP-21569	12.7
ADSP-SC594	9.755

Table 7: Secure boot time enhancement in ADSP-SC594 vs ADSP-21569

Comparison of Boot Time of various processors

This section will provide total boot time comparison across various processors which includes ADSP-SC57x, ADSP-2156x and ADSP-2159x/ADSP-SC59x families. Best possible boot time is provided for all the variants, where it is SPI quad-STR boot for ADSP-SC57x processor family, OSPI quad-DTR boot for ADSP-2156x processor family and OSPI quad-DTR boot for ADSP-2159x/ADSP-SC59x processor family.

Table 8 to Table 11 give the boot time comparison across all the three processor families in different modes, i.e., Normal, Secure BLp, Secure BLx and Secure BLw.

Normal Boot Time Comparison

Loader Stream Size in KB	SPI Quad-STR Boot in ms	OSPI Quad-DTR Boot in ms	
	ADSP-SC573	ADSP-21569	ADSP-21593
	SPI Clock=75 MHz	OSPI Clock= 62.5 MHz	
500	24.5325	10.0755	10.2661
1000	43.2825	18.5255	18.4661
2000	80.7825	35.4255	34.8661
4000	155.7825	69.2255	67.6661
8000	305.7825	136.8255	133.2661
12000	455.7825	204.4255	198.8661

Table 8: Normal Boot time comparison across various processors.

BLp Secure Boot Time Comparison

Loader Stream Size in KB	SPI Quad-STR Boot in ms	OSPI Quad-DTR Boot in ms	
	ADSP-SC573	ADSP-21569	ADSP-21593
	SPI Clock=75 MHz	OSPI Clock= 62.5 MHz	
500	40.23	17.2902	14.9717
1000	62.68	27.2402	24.1717
2000	107.583	47.1402	42.5717
4000	197.383	86.9402	79.3717
8000	376.9	166.5402	152.9717
12000	556.5	246.1402	226.5717

Table 9: Secure BLp Boot time comparison across various processors.

BLx Secure Boot Time Comparison

Loader Stream Size in KB	SPI Quad-STR Boot in ms	OSPI Quad-DTR Boot in ms	
	ADSP-SC573	ADSP-21569	ADSP-21593
	SPI Clock=75 MHz	OSPI Clock= 62.5 MHz	
500	54.877	19.7049	16.8513
1000	92.627	30.2549	26.5513
2000	168.127	51.3549	45.9513
4000	319.127	93.5549	84.7513
8000	621.127	177.9549	162.3513
12000	923.127	262.3549	239.9513

Table 10: Secure BLx Boot time comparison across various processors.

BLw Secure Boot Time Comparison

Loader Stream Size in KB	SPI Quad-STR Boot in ms	OSPI Quad-DTR Boot in ms	
	ADSP-SC573	ADSP-21569	ADSP-21593
	SPI Clock=75 MHz	OSPI Clock= 62.5 MHz	
500	54.026	19.5518	17.2634
1000	91.726	30.1018	26.4134
2000	167.126	51.2018	44.7134
4000	317.926	93.4018	81.3134
8000	619.526	177.8018	154.5134
12000	921.126	262.2018	227.7134

Table 11: Secure BLw Boot time comparison across various processors

Conclusion

Boot time estimation allows the user to construct an optimized loader stream which will give a faster boot time. Linear equations for boot time estimation in different boot modes based on the boot image size were provided in the EE-note. It also reviewed the effects of different factors affecting the total boot time, as well explained on how to incorporate those to make the boot time estimation more accurate and robust.

This guide can also be useful to a user to design a multistage boot strategy. For example, a user wants to boot an application of loader stream of size 2 MB inside 50ms time using OSPI Quad DTR boot mode. In such case, user will only be able to achieve 23.4375 MHz OSPI Clock (With 750MHz CCLK and 375MHz SYSCLK) at power up, with which 50ms boundary cannot be met (As per the equation in Figure 9). In such situation, updating the CGU and Boot Command for 62.5 MHz OSPI clock with correct RDCR value will solve the problem. So, user will have the following different options to incorporate:

- Normal Boot with OTP, where CGU initialization, Boot Command and RDCR programming can be done through OTP, while DMC initialization can be done through initcode or primary application in multistage boot.
- Normal Boot without OTP, where CGU & DMC initialization, Boot Command and RDCR programming can be initialized through initcode or primary application in multistage boot.
- Secure Boot, where CGU initialization, security keys, Boot Command and RDCR can be programmed in OTP. DMC initialization needs be done through a primary application using multistage boot.

As in such cases where a multistage boot strategy needs to be adapted, one can minimize the size for the primary application using this guide, so that the total boot time requirement is met. This way boot time estimation can be useful for a user to decide a complex multi-phase boot strategy, where they can choose the size of each boot stage to help meet their system requirements as well as reducing the overall boot time of whole system.

References

[1] *EE-366: Secure Booting Guide for Blackfin+® and SHARC+® Processors. Rev 2, October 19,2016.*

[2] *CrossCore® Embedded Studio 2.9.0 Loader and Utilities Manual (Rev. 2.3), May 2019.*

[3] *ADSP-SC59x/ADSP-2159x SHARC+® Processor Hardware Reference. Rev 0.2, May 2021 Analog Devices, Inc.*

Document History

Revision	Description
Rev 1 – August 23, 2021	Initial Release