SANBlaZe
an ellisys company

White Paper

Verification of SRIS/SRNS Clocking

PCIe® Gen5 Update, SBExpress-RM5

SRIS, SRNS and All Other Clocking Modes

SBExpress-RM5 V3 Update

Verification for SSD Drives

Authors

Vince Asbridge, Founder & President, SANBlaze
Bill Andrews, Field Application Engineer, SANBlaze
Haiyan Lin, Sr. Software Engineer, SANBlaze

Table of Contents

Document Revision 2
Introduction 3
Version 3 Gen5 Hardware 3
Understanding the PCIe/NVMe Clocking Modes 3
SSC Defined 3
PCIe Clocking Architectures 3
Common Reference Clock 4
Separate Reference Clocks with No SSC (SRNS) 4
Separate Reference Clocks with SSC (SRIS) 4
PCIe Clocking Modes SANBlaze Supports 5
SBExpress-RM5 Dip Switch Settings 6
Setting Configuration Switches from GUI 6
Setting Configuration Switches from CLI 7
SBExpress-RM4 Dip Switch Settings 8
Link Error Measurement with AER Monitoring 8
Common Clocking Mode Without SSC 9
Common Clocking Mode With SSC 9
Separate Reference Clock with No SSC (SRNS) 9
Separate Reference Clock Independent SSC (Static SRIS) 9
RefClk Without SSC for PEX Only (No RefClk to The Target Drive) 9
RefClk With SSC for PEX Only (No RefClk to the Target Drive, Dynamic SRIS) 10
Example Measurement Results 10
Conclusion 11

Document Revision

V2 - Initial release for SBExpress-RM4 Gen4 NVMe Test System

V3 - Update for SBExpress-RM5 Gen5 NVMe Test System

Introduction

SANBlaze announces the availability of the first NVMe Gen5 test platform with the capability of testing the various modes of clocking required by PCIe NVMe devices. This paper describes in detail the clocking modes required for modern NVMe devices and how to verify that your NVMe device can run correctly with each of the possible clocking schemes.

Version 3 Gen5 Hardware

SANBlaze has introduced a new hardware version of the SBExpress-RM5 Gen5 test system which enables SRIS/SRNS testing at PCIe Gen5 speeds. The new hardware, introduced on May 25th, 2022, can be identified by part number 600-950003 for the mother board and system part number 650-950003 which is found on the label on the back of the chassis.

All combinations of clock configuration as described in V2 of this paper are supported on hardware revision V3. The 8 position DIP switch for system configuration, which was assessable only by operating the physical switch on the Gen5 V3 system is now operated by software so that physical intervention is no longer necessary. Operating the switch from the system SBExpress Web Page and at the CLI prompt are described in this update to the original white paper.

Understanding the PCIe/NVMe Clocking Modes

PCI Express (PCIe) has evolved from PCI/PCI-X (Peripheral Component Interconnect), which first came to market as a parallel interconnect bus on Personal Computers. PCIe is a serial bus enabled by SER/DES (Serializers/Deserializers) built into the endpoint devices (Root Complex or Peripherals). As PCIe has evolved, the speed of the clock, and therefore the data rate and bandwidth of the bus, has increased sixteen-fold from Gen1 to Gen5. PCIe is designed as a point-to-point topology where each endpoint is connected by a number of serial lanes.

SSC Defined

As the PCIe bus expanded outside the host computer (in the case for example of a rack of NVMe drives), and the clock frequencies have increased, the need to decrease the Electromagnetic Interference (EMI) from the interconnect became a priority. PCIe addresses the EMI problem by modulating the reference clock with a "spread spectrum" modulation. This technique is known as Spread Spectrum Clocking, or SSC. SSC clocking reduces the EMI level by spreading the radiated energy over a range of frequencies thereby reducing the peak emissions at the PCIe clock center frequency.

While SSC clocking reduces EMI interference, it does so at the cost of introducing clock jitter into the PCIe subsystem. When SSC is enabled, it is typical to use a common reference clock at both ends of the PCIe connection, for example at the root complex and peripheral device.

PCIe Clocking Architectures

PCIe supports various clocking architectures as described below. A vendor of NVMe devices may need to test one, all, or a combination of these clocking schemes, introducing a significant problem for a test engineer. A host system typically implements one clocking scheme, but not multiple schemes, and may or may not support SSC alone or in combination with independent or common clock architectures.

The SANBlaze SBExpress-RM4 and RM5 V3 support all combinations of PCIe clocking schemes, and therefore provide an ideal testbed to verify correct implementation and stability of your NVMe device.

The following modes of clocking operation are supported.

Common Reference Clock

A common reference clock architecture refers to a configuration where a common clock is supplied to the upstream device. In the case of the SBExpress-RM5 this is a Gen5 PCIe bridge, and the endpoint peripheral in this case is an NVMe drive.

Common clock without SSC is the most basic clocking scheme. It has the highest performance, lowest latency, and is the least likely to generate errors. A single stable clock, with low jitter, is shared by both endpoints.

Separate Reference Clocks with No SSC (SRNS)

The second most common clocking scheme is called SRNS (Separate Reference Clocks with No SSC) and is one where an independent clock is supplied to each end of the PCIe link. For example, Clock 'A' is supplied to the upstream bridge and Clock 'B' is supplied to the endpoint device, the NVMe drive.

The performance and stability of SRNS should be identical to that of a common clock architecture because even though the clocks are independent, they are the same frequency. Buffering on each end of the PCIe link will compensate for jitter, up to 600ppm, between the two independent clocks (+/- 300ppm per clock).

Separate Reference Clocks with SSC (SRIS)

Combining the technique of Spread Spectrum Clocking (SSC) with the use of independent clocking introduces the greatest challenge for design and test engineers. With an allowed jitter of 5000ppm for SSC and the 600ppm for independent clocking, a total jitter of 5600ppm must be accounted for between the endpoints.[1]

The PCIe protocol compensates for the difference in frequency of the two endpoint clocks by the use of SKP Ordered Sets (OS). Each device must implement buffers to compensate for the difference in endpoint clock frequencies while the protocol sends SKP OS Transaction Layer Protocol (TLPs) to synchronize the endpoints, at a rate proportional to the difference in end point frequencies.

Because of the need to transmit SKP TLPs, designers must provide sufficient elasticity in the buffer design of each endpoint, and also expect increased latency.

SANBlaze provides a means to enable each of the PCIe clocking scenarios while providing data integrity, exception testing, and performance monitoring to ensure NVMe devices have correctly implemented supported clocking schemes.

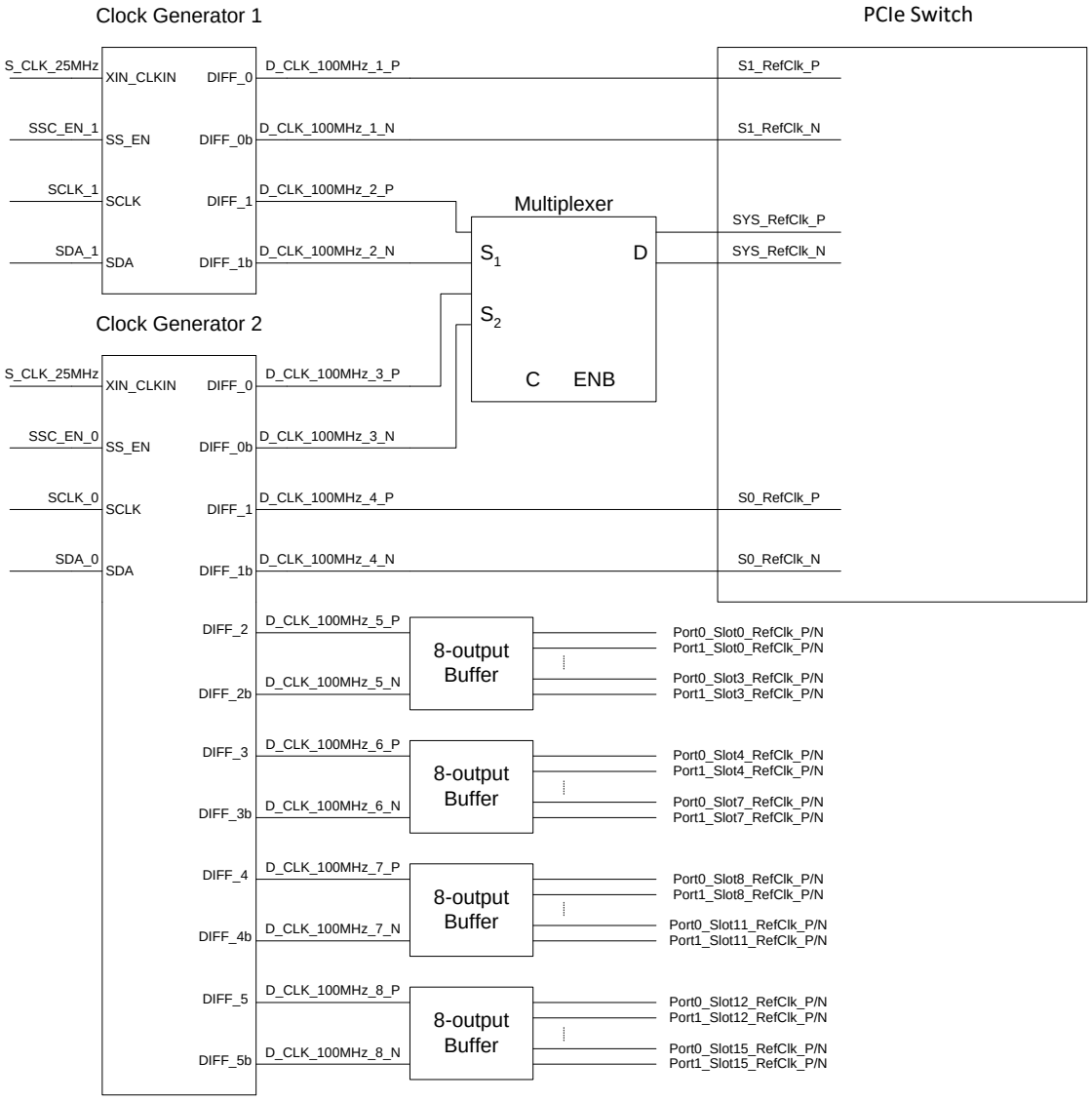
The details of enabling and testing each clocking configuration is described below.

PCIe Clocking Modes SANBlaze Supports

SSD manufacturers need to verify if their SSDs are working as expected in the following different PCIe clocking modes:

- Common clocking mode without SSC
- Common clocking with SSC
- Separate reference clock independent SSC (Static SRIS)
- Separate reference clock with no SSC (SRNS)
- RefClk without SSC for PEX only (no RefClk to the target drive)
- RefClk with SSC for PEX only (no RefClk with SSC to the target drive, Dynamic SRIS)

The SANBlaze PCIe Gen5 system (SBExpress-RM5 V3) has 2 clock generators, and it is designed to support all clocking modes listed above. Following is the clocking scheme implemented in our PCIe Gen5 system.



SBExpress-RM5 Dip Switch Settings

The configuration switch is now controlled by software. The positions of the switch will remain set over reboot and power off / on. The switches can be set by using the SBExpress page on the SANBlaze GUI, or at the CLI using the commands below.

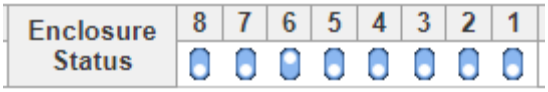
Note: SRIS mode testing requires that the DIP switches be set at power on and before starting each test. You must power cycle the system after setting the switches to enable the next test mode.

Note: Switch 6 is ON by default; all others are OFF by default.

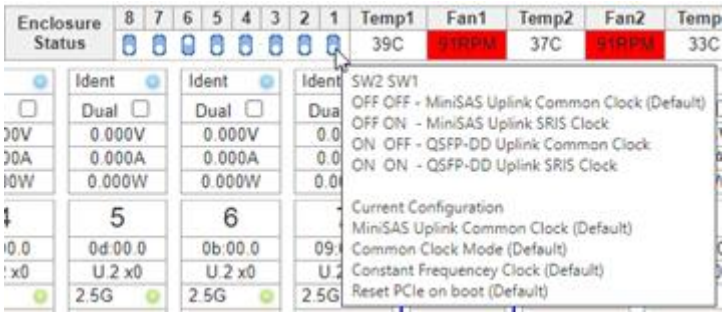
Setting Configuration Switches from GUI

To access the switches from the GUI, select the SBExpress Manager page in the left hand menu.

Selecting any of the switches will change the position of the switch.



Holding the cursor over a switch will bring up help and show current system configuration.



Setting Configuration Switches from CLI

To see the DIP switches in the GUI, you need version 10.5 of the software.

To set the configuration switches from the system CLI issue the following commands:

Switches 1 & 2

2=OFF,1=OFF: **(DEFAULT)** Single host Non-SRIS configuration

iriser -d -3 set SW1EESel0
iriser -d -3 set SW2EESel1

2=OFF,1=ON: Single host SRIS configuration

iriser -d -3 clear SW1EESel0
iriser -d -3 set SW2EESel1

Switch 6

OFF: SRIS/SRNS depending on switch 7

iriser -d -3 set SW6SRIS

ON: (DEFAULT) Common Clock Mode

iriser -d -3 clear SW6SRIS

Switch 7

OFF: (DEFAULT) Spread Spectrum Clocking is turned off

iriser -d -3 set SW7SSCL

ON: Spread Spectrum Clocking is turned on

iriser -d -3 clear SW7SSCL

Switch 8

OFF: (DEFAULT) Reset PCIe on Boot

iriser -d -3 set SW8DisRst

ON: Disable PCIe Reset on Boot

iriser -d -3 clear SW8DisRst

Note: The switch syntax set(1) is switch OFF, and clear(0) is switch ON.
You will see the position of the switch update in the GUI when the switch is changed at the CLI

Note: You can get the names of all GPIOs including the switches using the command:
iriser -d -3 show gpio

Direction: I/O (UC) user writable signal, i/o (LC) signal locked

GPIO	31								23								15								7								
	Bit3 N/C	PwrOnRst	DisUSbL	DisWtchDg	TP3	TP2	TP1	DrvRstL	SW8DisRst	SW7SSCL	SW6SSRIS	SW5FrcPwr	SW4Chain	SW3UslL	SW2EESEL	SW1EESel0	Dua1115L	Dua1114L	Dua1113L	Dua1112L	Dua1111L	Dua1110L	Dua1109L	Dua1108L	Dua1107L	Dua1106L	Dua1105L	Dua1104L	Dua1103L	Dua1102L	Dua1101L	Dua1100L	
Dir:[0x214]0xffffffff	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	
Val:[0x210]0xefdfffff	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

SBExpress-RM4 Dip Switch Settings

The RefClk source and SSC on/off are controlled by back panel switches in the SANBlaze SBExpress-RM4 Rev2 system.

NOTE: SRIS mode testing requires that the DIP switch switches be set at power on and before starting each test. Switches 6 and 8 are on by default.

Switches 1 & 2

2=OFF,1=OFF: **(DEFAULT)** Single host Non-SRIS configuration
2=OFF,1=ON: Single host SRIS configuration

Switch 6

OFF: SRIS/SRNS depending on switch 7
ON: (DEFAULT) Common Clock Mode

Switch 7

OFF: (DEFAULT) Spread Spectrum Clocking is turned off
ON: Spread Spectrum Clocking is turned on

Switch 8

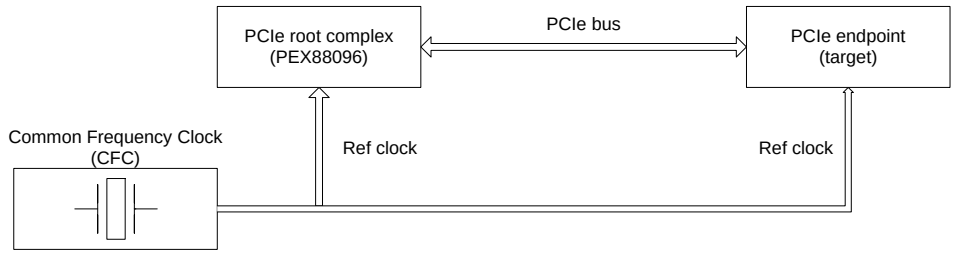
OFF:12V Power to slots is forced on.
ON: **(DEFAULT)** 12V Power to slots controlled using Serial Hot Plug logic

Link Error Measurement with AER Monitoring

The SANBlaze SBExpress-RM5 PCIe Gen5 system supports link error measurements such as Receiver Error, TLP Error, DLLP Error and AER (Advanced Error Reporting) flags monitoring when running I/O on SSDs with all different clocking modes and default De-emphasis (auto tuning for PCIe Gen3 and Gen5 link speed), De-emphasis = -6dB, De-emphasis = -3.5dB and so on. SSD Manufacturers verify the following clocking modes, and SANBlaze supports all of them as follows.

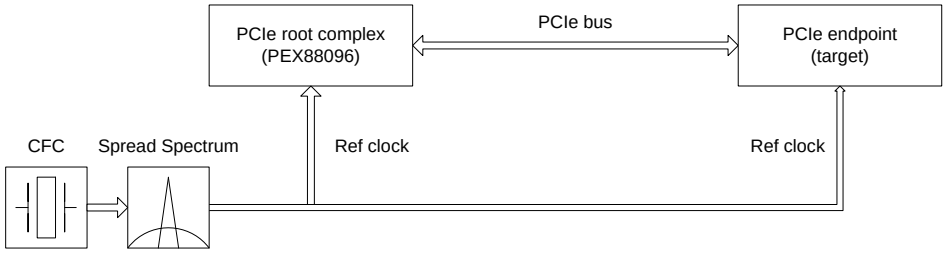
Common Clocking Mode Without SSC

The common clocking mode without SSC is shown in the following chart:



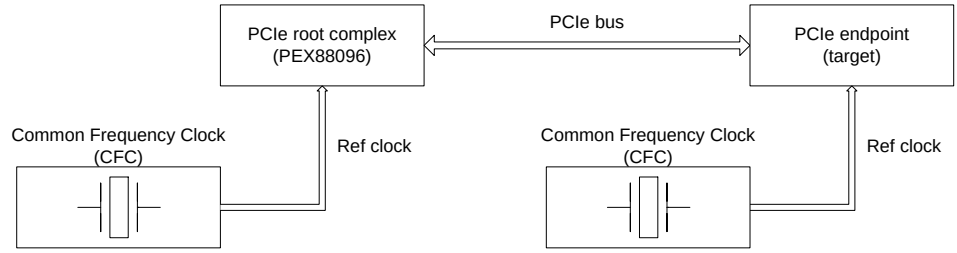
Common Clocking Mode With SSC

The common clocking mode with SSC is shown in the following chart:



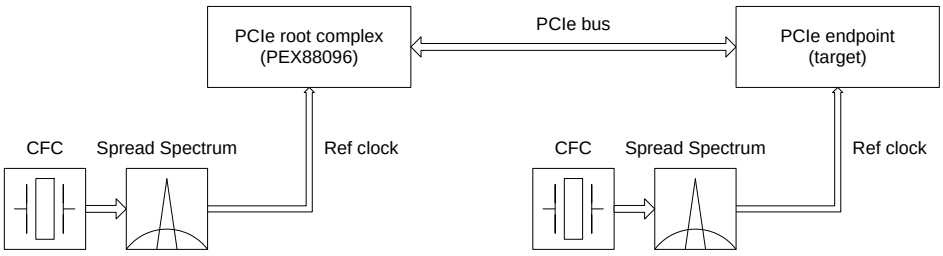
Separate Reference Clock with No SSC (SRNS)

The separate reference clock with no SSC (SRNS) is shown in the following chart:



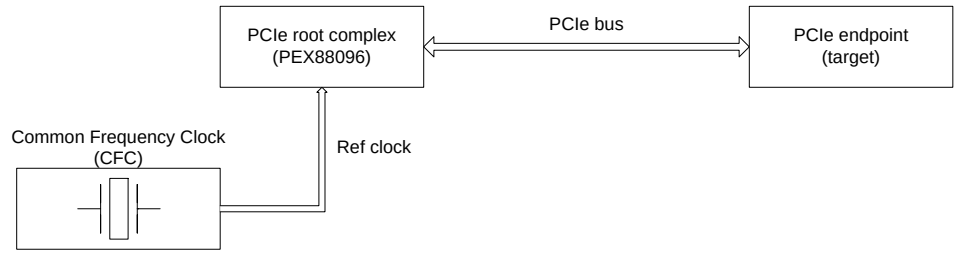
Separate Reference Clock Independent SSC (Static SRIS)

The separate reference clock independent SSC (SRIS) is shown in the following chart:



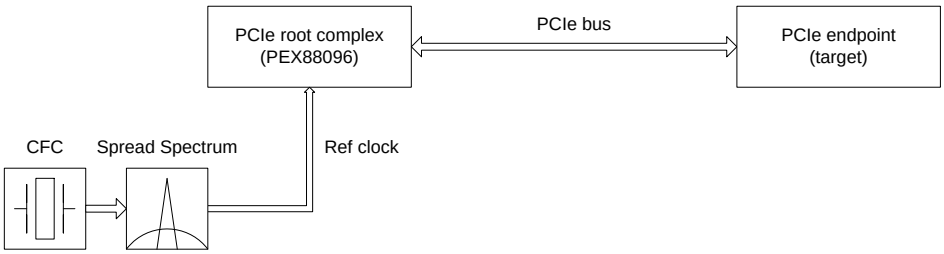
RefClk Without SSC for PEX Only (No RefClk to The Target Drive)

The RefClk without SSC for PEX only (no RefClk to the target drive) is shown in the following chart:



RefClk With SSC for PEX Only (No RefClk to the Target Drive, Dynamic SRIS)

The RefClk with SSC for PEX only (no RefClk to the target drive) is shown in the following chart:



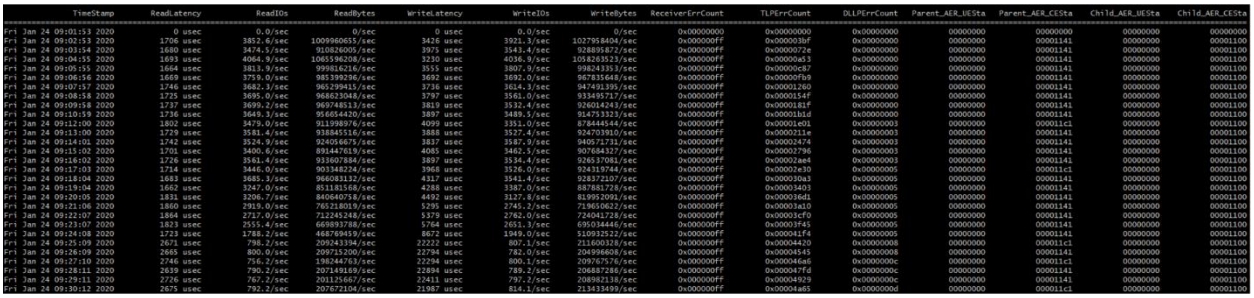
Example Measurement Results

The example output below show two runs with a device that supports dynamic SRIS. First the test was run in the Dynamic SRIS mode with auto tuning de-emphasis. The results are a clean run with no errors. The next case we ran in the Static SRIS mode with the same auto tuning de-emphasis. As expected you can see that the Receiver errors saturate immediately. Many TLP errors and some DLLP errors happened during this I/O testing. The Receiver errors, TLP errors, and DLLP errors reported on PEX88096 match the parent PCIe AER register “CE_Sta” which means the measurement is correct. The link errors are related to the fact that the SSD firmware is running auto-detection mode. This test was to prove that the errors seen correlate to both upstream and downstream devices.

The example output below shows dynamic SRIS running with auto tuning de-emphasis. As you can see, no Receiver errors, TLP errors, or DLLP errors happened during this I/O testing. The reason is that the SSD did not detect any external RefClk so it used its own internal RefClk with SSC on, which will send extra Skip ordered-sets.

TimeStamp	ReadLatency	ReadIOs	ReadBytes	WriteLatency	WriteIOs	WriteBytes	ReceiverErrCount	TLPErrCount	DLLPErrCount	Parent_AER_UESta	Parent_AER_CESta	ChiId_AER_UESta	ChiId_AER_CESta
Thu Feb 06 13:44:58 2020	0 usec	0.0/sec	0/sec	0 usec	0.0/sec	0/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 13:45:59 2020	1272 usec	3812.0/sec	999293927/sec	3930 usec	3826.0/sec	1002963946/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 13:46:59 2020	1441 usec	2888.0/sec	757971872/sec	5410 usec	2917.0/sec	764674048/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 13:48:00 2020	1431 usec	2832.0/sec	742391808/sec	5790 usec	2745.0/sec	719585280/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 13:49:01 2020	1447 usec	2952.9/sec	774109643/sec	5351 usec	2917.9/sec	764934662/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 13:50:01 2020	1451 usec	2841.0/sec	745013248/sec	5464 usec	2894.0/sec	758644736/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 13:51:02 2020	1296 usec	5365.0/sec	1406402560/sec	2440 usec	5289.0/sec	1386479616/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 13:52:03 2020	1153 usec	4413.0/sec	1157365760/sec	3470 usec	4262.0/sec	1117257728/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 13:53:03 2020	1177 usec	4075.0/sec	1068217868/sec	3672 usec	4116.0/sec	1078985782/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 13:54:04 2020	1163 usec	3846.0/sec	1008206832/sec	3863 usec	3993.0/sec	1046742038/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 13:55:04 2020	1192 usec	3926.0/sec	1029178373/sec	3873 usec	3930.0/sec	1030216950/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 13:56:05 2020	1173 usec	3892.0/sec	1020265468/sec	3911 usec	3921.0/sec	1027867651/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 13:57:06 2020	1209 usec	3969.0/sec	1040450576/sec	3928 usec	3838.0/sec	1006109678/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 13:58:06 2020	1197 usec	3888.9/sec	1018215872/sec	3967 usec	3843.0/sec	1007413392/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 13:59:07 2020	1188 usec	3978.0/sec	1042808832/sec	3859 usec	3932.0/sec	1030750208/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 14:00:08 2020	1233 usec	4119.0/sec	1079772215/sec	3744 usec	3959.0/sec	1037829133/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 14:01:08 2020	1184 usec	4002.0/sec	1049100288/sec	3787 usec	4007.0/sec	1050411008/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 14:02:09 2020	1159 usec	3783.0/sec	991690752/sec	3892 usec	3990.0/sec	1045954560/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 14:03:09 2020	1194 usec	4057.0/sec	1063518208/sec	3744 usec	4014.0/sec	1052246016/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 14:04:10 2020	1175 usec	4070.0/sec	1066926080/sec	3743 usec	4046.0/sec	1059061760/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 14:05:11 2020	1160 usec	4030.0/sec	1056441376/sec	3754 usec	4050.0/sec	1061684261/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 14:06:11 2020	1190 usec	3899.0/sec	1022099456/sec	3882 usec	3934.0/sec	1031274496/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 14:07:12 2020	1189 usec	3885.9/sec	1018690565/sec	3944 usec	3873.9/sec	1015144840/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 14:08:13 2020	1198 usec	3806.9/sec	997981210/sec	3957 usec	3884.9/sec	1018428421/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 14:09:13 2020	1188 usec	3857.0/sec	1011090419/sec	3919 usec	3913.0/sec	1025770497/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 14:10:14 2020	1179 usec	3958.0/sec	1037565952/sec	3840 usec	3974.0/sec	1041760256/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 14:11:14 2020	1197 usec	3830.0/sec	1004011520/sec	3923 usec	3903.0/sec	1023148032/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 14:12:15 2020	1228 usec	3707.0/sec	971768779/sec	4001 usec	3831.0/sec	1004274468/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 14:13:16 2020	1268 usec	3836.0/sec	1005584384/sec	3809 usec	3948.0/sec	1034944512/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 14:14:16 2020	1283 usec	3987.0/sec	1045168128/sec	3768 usec	3921.0/sec	1027866624/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Thu Feb 06 14:15:17 2020	1257 usec	4086.0/sec	1071120384/sec	3727 usec	3958.0/sec	1037565952/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000

The example output below shows Static SRIS running with auto tuning de-emphasis. This case did receive the errors expected and they are correlated between upstream and downstream devices.



TimeStamp	ReadLatency	ReadIOs	ReadBytes	WriteLatency	WriteIOs	WriteBytes	ReceiverErrCount	TLPErrCount	DLLPErrCount	Parent_AER_UESta	Parent_AER_CESta	Child_AER_UESta	Child_AER_CESta
Fri Jan 24 09:01:53 2020	0 usec	0.0/sec	0/sec	0 usec	0.0/sec	0/sec	0x00000000	0x00000000	0x00000000	00000000	00000000	00000000	00000000
Fri Jan 24 09:02:53 2020	1706 usec	3852.6/sec	1009960655/sec	3426 usec	3921.3/sec	1027958404/sec	0x000000FF	0x000003bf	0x00000000	00000000	00001141	00000000	00001100
Fri Jan 24 09:03:54 2020	1680 usec	3474.5/sec	910826005/sec	3975 usec	3543.8/sec	928891872/sec	0x000000FF	0x0000072e	0x00000000	00000000	00001141	00000000	00001100
Fri Jan 24 09:04:55 2020	1693 usec	4064.9/sec	1065596208/sec	3235 usec	4036.9/sec	1058263523/sec	0x000000FF	0x00000a83	0x00000000	00000000	00001141	00000000	00001100
Fri Jan 24 09:05:55 2020	1664 usec	3813.9/sec	999816216/sec	3555 usec	3807.9/sec	998243353/sec	0x000000FF	0x00000c87	0x00000000	00000000	00001141	00000000	00001100
Fri Jan 24 09:06:56 2020	1669 usec	3759.0/sec	985399296/sec	3692 usec	3692.0/sec	967815648/sec	0x000000FF	0x00005fb9	0x00000000	00000000	00001141	00000000	00001100
Fri Jan 24 09:07:57 2020	1746 usec	3682.3/sec	965299415/sec	3736 usec	3614.3/sec	947491395/sec	0x000000FF	0x00001260	0x00000000	00000000	00001141	00000000	00001100
Fri Jan 24 09:08:58 2020	1725 usec	3695.0/sec	968623048/sec	3797 usec	3561.0/sec	933495717/sec	0x000000FF	0x0000154f	0x00000000	00000000	00001141	00000000	00001100
Fri Jan 24 09:09:58 2020	1737 usec	3699.2/sec	969748513/sec	3819 usec	3532.4/sec	926014243/sec	0x000000FF	0x00001819	0x00000000	00000000	00001141	00000000	00001100
Fri Jan 24 09:10:59 2020	1736 usec	3649.3/sec	956654420/sec	3897 usec	3489.5/sec	914753323/sec	0x000000FF	0x00001b1d	0x00000000	00000000	00001141	00000000	00001100
Fri Jan 24 09:12:00 2020	1802 usec	3479.0/sec	911998976/sec	4099 usec	3351.0/sec	878444544/sec	0x000000FF	0x00001e01	0x00000003	00000000	000011c1	00000000	00001100
Fri Jan 24 09:13:00 2020	1729 usec	3581.4/sec	938845516/sec	3888 usec	3527.4/sec	924703910/sec	0x000000FF	0x0000211e	0x00000003	00000000	00001141	00000000	00001100
Fri Jan 24 09:14:01 2020	1742 usec	3524.9/sec	924056675/sec	3837 usec	3587.9/sec	940571731/sec	0x000000FF	0x00002474	0x00000003	00000000	00001141	00000000	00001100
Fri Jan 24 09:15:02 2020	1701 usec	3400.6/sec	891447619/sec	4085 usec	3462.5/sec	907684327/sec	0x000000FF	0x00002796	0x00000003	00000000	00001141	00000000	00001100
Fri Jan 24 09:16:02 2020	1726 usec	3561.4/sec	933607884/sec	3897 usec	3534.4/sec	926517081/sec	0x000000FF	0x00002ae4	0x00000003	00000000	00001141	00000000	00001100
Fri Jan 24 09:17:03 2020	1714 usec	3446.0/sec	903348224/sec	3968 usec	3526.0/sec	924319744/sec	0x000000FF	0x00002e30	0x00000005	00000000	000011c1	00000000	00001100
Fri Jan 24 09:18:04 2020	1683 usec	3685.3/sec	966083132/sec	4317 usec	3541.4/sec	928372107/sec	0x000000FF	0x000030a3	0x00000005	00000000	00001141	00000000	00001100
Fri Jan 24 09:19:04 2020	1662 usec	3247.0/sec	851181568/sec	4288 usec	3387.9/sec	887881728/sec	0x000000FF	0x00003403	0x00000005	00000000	00001141	00000000	00001100
Fri Jan 24 09:20:05 2020	1831 usec	3206.7/sec	840640758/sec	4492 usec	3127.8/sec	819952091/sec	0x000000FF	0x000036d1	0x00000005	00000000	00001141	00000000	00001100
Fri Jan 24 09:21:06 2020	1860 usec	2919.0/sec	765218019/sec	5295 usec	2745.2/sec	719650622/sec	0x000000FF	0x00003a10	0x00000005	00000000	00001141	00000000	00001100
Fri Jan 24 09:22:07 2020	1864 usec	2717.0/sec	712245248/sec	5379 usec	2762.0/sec	724041728/sec	0x000000FF	0x00003cf0	0x00000005	00000000	00001141	00000000	00001100
Fri Jan 24 09:23:07 2020	1823 usec	2555.4/sec	669893788/sec	5764 usec	2651.3/sec	695034446/sec	0x000000FF	0x00003f45	0x00000005	00000000	00001141	00000000	00001100
Fri Jan 24 09:24:08 2020	1723 usec	1788.2/sec	468769459/sec	8672 usec	1949.0/sec	510932522/sec	0x000000FF	0x000041f4	0x00000005	00000000	00001141	00000000	00001100
Fri Jan 24 09:25:09 2020	2671 usec	798.2/sec	209243194/sec	22222 usec	807.1/sec	211600328/sec	0x000000FF	0x00004420	0x00000008	00000000	000011c1	00000000	00001100
Fri Jan 24 09:26:09 2020	2665 usec	800.0/sec	209715200/sec	22794 usec	782.0/sec	204996608/sec	0x000000FF	0x00004545	0x00000008	00000000	00001141	00000000	00001100
Fri Jan 24 09:27:10 2020	2746 usec	756.2/sec	198244763/sec	22294 usec	800.1/sec	209767576/sec	0x000000FF	0x000046a6	0x0000000c	00000000	000011c1	00000000	00001100
Fri Jan 24 09:28:11 2020	2639 usec	780.2/sec	207148169/sec	22894 usec	789.2/sec	206887286/sec	0x000000FF	0x000047f7	0x0000000c	00000000	00001141	00000000	00001100
Fri Jan 24 09:29:11 2020	2726 usec	767.2/sec	201125667/sec	22411 usec	797.2/sec	208982138/sec	0x000000FF	0x00004929	0x0000000c	00000000	00001141	00000000	00001100
Fri Jan 24 09:30:12 2020	2675 usec	792.2/sec	207672104/sec	21987 usec	814.1/sec	213433499/sec	0x000000FF	0x00004a65	0x0000000d	00000000	000011c1	00000000	00001100

Conclusion

With multiple ways to implement PCIe clocking on an NVMe device, a test engineer is faced with difficulty in their test bed in attempting to test all possible scenarios. With SANBlaze as their test tool however, this problem simply "goes away." SANBlaze provides testing for every clock mode that an SSD manufacturer could possibly implement, thereby providing seamless and thorough end-to-end testing for NVMe devices.

Jitter Reference[1]

https://blogs.synopsys.com/vip-central/2015/12/15/pcie-spread-spectrum-clocking-ssc-for-digital-verification-engineers/

- **F_{REFCLK}**: Refclk frequency can have +/-300 PPM variation. For separately clocked architecture, the worst case jitter of 600 PPM will have to be tolerated by the receiver.
- **F_{SSC}**: This is frequency of the modulating wave. This is typically triangular.
- **$T_{SS\text{-}FREQ\text{-}DEVIATION}$**: This indicates PCIe uses Down-spread SSC. This spread is applied to reduce the carried frequency up -0.5%. This means an additional 5000 PPM of jitter. So total jitter with the separate clocking having spread spectrum enabled would be 5600 PPM.

