Tektronix®

Characterizing a PCIe® Device's Performance Using the Link Training Status State Machine Monitor

APPLICATION NOTE

Background

Twenty years ago, the Peripheral Component Interconnect Express (PCIe) standard was first defined by the PCI-SIG organization. PCIe was introduced to enable high-speed serial communication between the Central Processing Unit (CPU) and its peripheral components. Since 2003, the PCIe standard has iteratively improved to accommodate the latest bandwidth needs of modern computers. Although PCIe was introduced as a serial interface to replace the parallel bus used in many motherboard architectures, a unique feature of the PCIe is the ability to increase the number of lanes from 1 up to 32. Using this parallel bus feature, a PCIe-compliant device can establish a link with other PCIe-compliant devices with link widths of 1, 2, 4, 8, 16, and up to 32 lanes, as required according to data transmission requirements.

As data rates have evolved in the specification, so has the complexity of the physical layer protocols required to ensure efficient data transmission while safeguarding essential principles of PCIe specifications, which include lane width flexibility and downward compatibility with "Legacy" devices- i.e., developed with different PCIe Generations. For example, a device developed with PCIe Gen4 specification has to be downward compatible — interoperable — with devices designed in earlier PCIe releases, Gen1 and Gen2.

About this Application Note

This Application Note describes the use of the advanced Link Training Status State Machine (LTSSM) Monitoring information provided by the Tektronix TMT4 Margin Tester. This advanced feature, together with the generation of hardware trigger-in/trigger-out capabilities, allow the user to identify anomalies in the Physical Layer interaction with the Data Link Layer according to PCIe State Machine descriptions.

The LTSSM monitor provides information about the various states achieved by the TMT4 Margin Tester because of its interaction with the Device Under Test (DUT) without regard to its role as a Root Complex (RC) or End Point (EP). One practical example of a debugging workflow may involve one or more of the following steps:

A. Verifying the DUT can transition through the available states in a sequence consistent with the PCIe specification, or "allowed transitions".

B. If the DUT does not transition from State-to-State in the expected sequence, determine which is the last known State the DUT was still operating as expected.

C. As the "suspect" State or transition is isolated, the user may choose to capture data on an oscilloscope upon entering a particular state by issuing a hardware trigger out.

PCI Express Architecture and the Need for Link Training

Even though PCIe is defined as a point-to-point protocol, there is a well-defined hierarchy when it comes to interaction between the sources and destinations of data. The structure of the PCIe system consists of many point-to-point interfaces, with multiple peripherals and modules connected through an infrastructure, or fabric. The main CPU (or processor sub-system) sits at the top and is connected to a 'root complex' (RC) using whatever appropriate user interface. This root complex is the top level PCIe interconnect component and is typically connected to Main Memory, which is accessed by the CPU via the Root Complex. The PCIe interfaces appear at the Root Complex, either by direct connection or by connection through a switch (**Figure 1**).

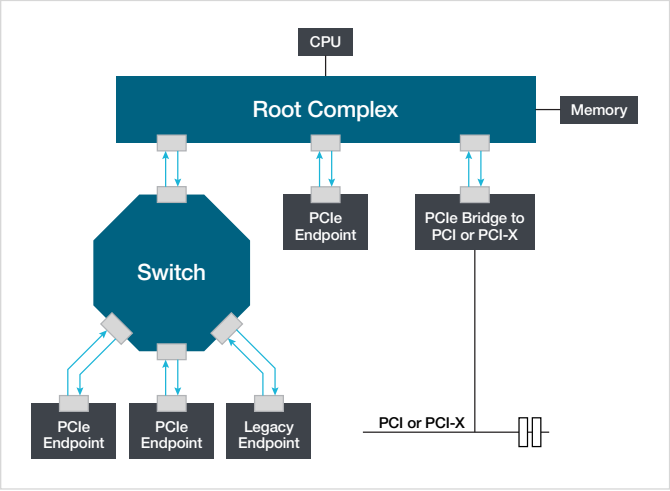


Figure 1. PCIe Hierarchy[1].

Each of the interconnects has a ***downstream link*** (coming from a component upstream such as the Root Complex or RC) and an ***upstream link*** (coming from a component downstream such as an End Point or EP) . Most typical Root Complex (RC) devices include motherboards in personal computers or controller boards in embedded systems. For an End Point, devices such as graphics cards or network interface cards may be connected to the Root Complex via switches, which help expand the number of addressable devices within the specification. It is also possible for EP devices to be communicating directly with the RC, as shown in **Figure 1** above. Often, this communication is not 100% direct, and a signal conditioning device (a retimer, or a redriver, **Figure 2**) is inserted between the RC and the EP device so as to ensure signal quality and compensate for the loss of signal quality over the traces, particularly at high transfer speeds.

In order to determine whether a particular PCIe device has a link partner that it can pair with, the PCIe specification establishes the Link Training process as a means of determining whether a given lane is suitable for transmission of data at the various speeds supported by the interface, and how many of these lanes are available, among various other considerations being made in the Physical Layer.

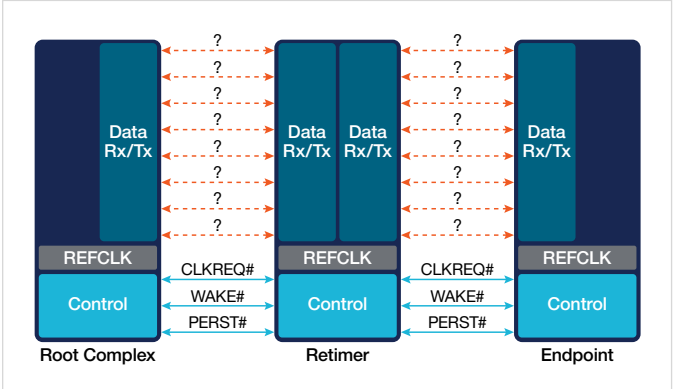


Figure 2. PCIe Physical Layer Connections Showing a Retimer Device[2].

How Is a Link Established in PCIe? How Is It Monitored?

When all the devices (at least one RC and one or more EPs) are powered and a reference clock provided, a PCIe device starts the so-called link training process. The link training process consists of receiver detection (Rx Detect), Polling, Configuration, and Recovery. During this process, the status of the link can be determined by observing the "LTSSM States". As described in the PCIe Specification, there are a total of eleven top-level States, and within each State there are sub-States with further details.

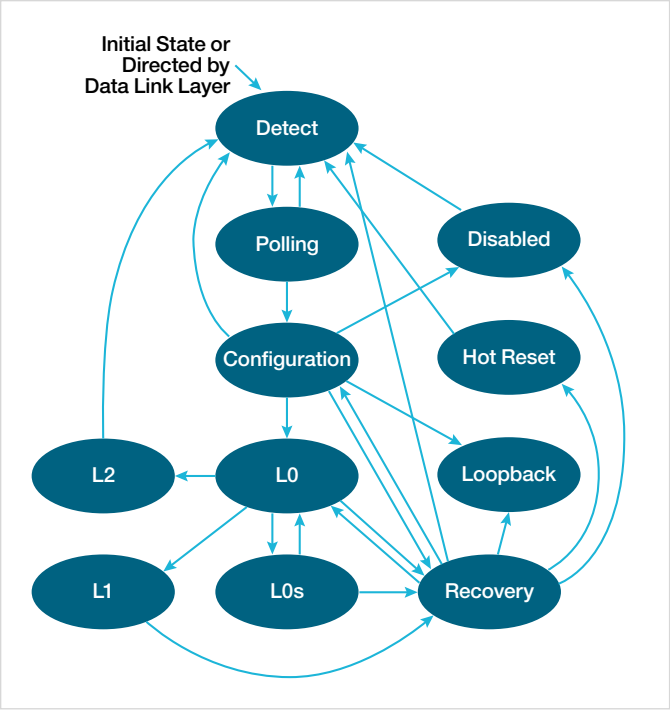


Figure 3. Link Training State Machine[3].

In summary:

- The state of the PCIe link is defined by a Link Training and Status State Machine (LTSSM). From an initial state, the state machine progresses through various major states (Detect, Polling, Configuration, Recovery) to train and configure the link before being fully in a link-up state (L0).
- Other states include power management states "Lx", a 'loopback' mode for test and debug, or a 'hot reset' state to send the link back to its initial state. The disabled state is for configured links where communications are suspended.
- The initial state is typically determined by the Data Link Layer, but in general the default initial state for a PCIe link is "Detect."

The Tektronix TMT4 Margin Tester User Interface displays a graphical representation of the State Machine diagram, as seen above, and when choosing to initiate a LTSSM Monitor test, the live progression recording of the states that the Margin Tester is seen as entering.

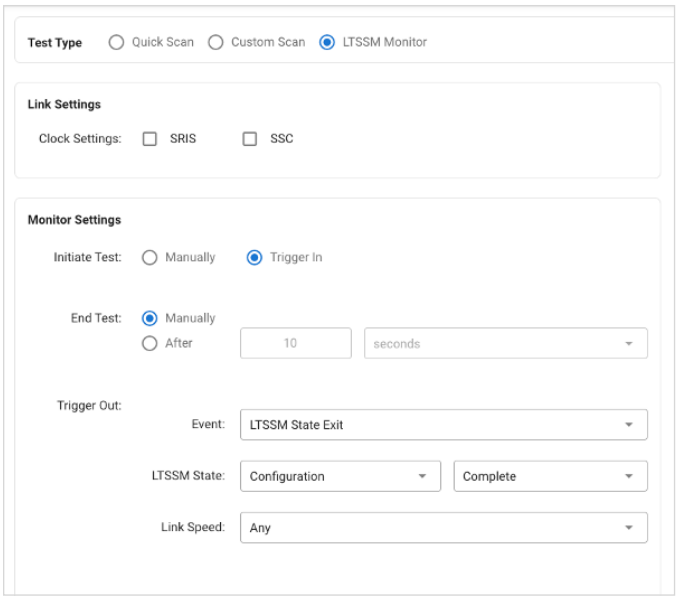


Figure 4. LTSSM Test Configuration.

LTSSM Monitoring can be initiated manually or in the presence of a valid hardware **Trigger-in** signal. Additionally, a **Trigger-out** signal can be generated upon the occurrence of a selected event associated with a LTSSM state (**Figure 4**). When the test is complete, the test results archive (*.zip) contains a log of the observed LTSSM states.

The TMT4 as a Link Partner

The TMT4 Margin tester is a flexible, PCIe-compliant instrument that can be configured as either a Root Complex device, or an End Point device supporting PCIe Gen3 and Gen4 speeds. The key to making the connection with the DUT implies choosing the correct adapter configuration.

For example, when an Add In Card (AIC) is the Device Under Test (DUT), the TMT4 Margin tester is behaving as the Root Complex (**Figure 5**).

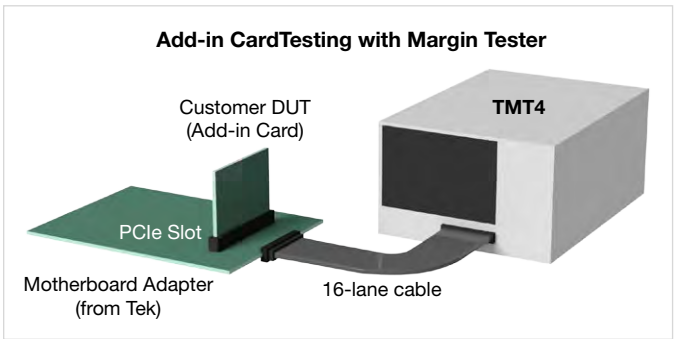


Figure 5. TMT4 as Root Complex.

On the other hand, when performing an evaluation of a typical Root Complex device – such as a computer motherboard, the TMT4 will interact with the DUT as an End Point device (**Figure 6**).

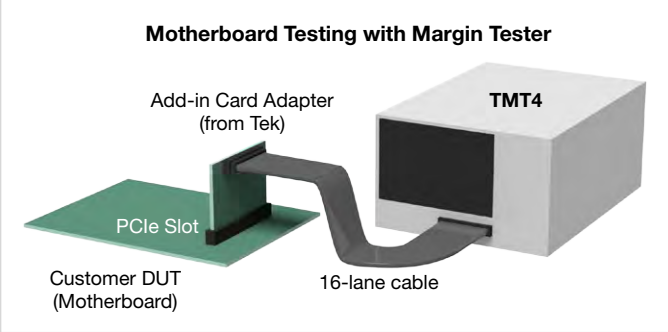


Figure 6. TMT4 as End Point.

A Closer Look inside the Link Training Process

As PCIe devices enter the link training process, there is a two-way data exchange taking place between both ends of the link. As test data and patterns (TS= Training Sets) are sent from one end of the link to the other, the receiving side responds to the transmitting side and switch roles in transmitting and receiving data as progress is made going from one state to the next (**Figure 7**).

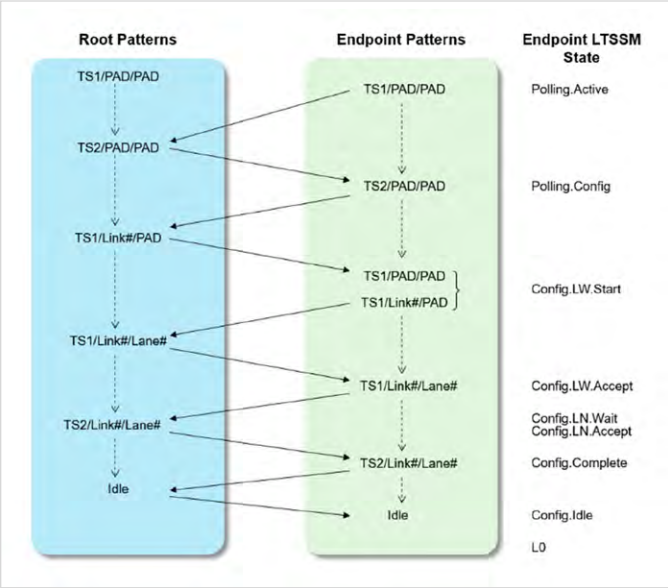


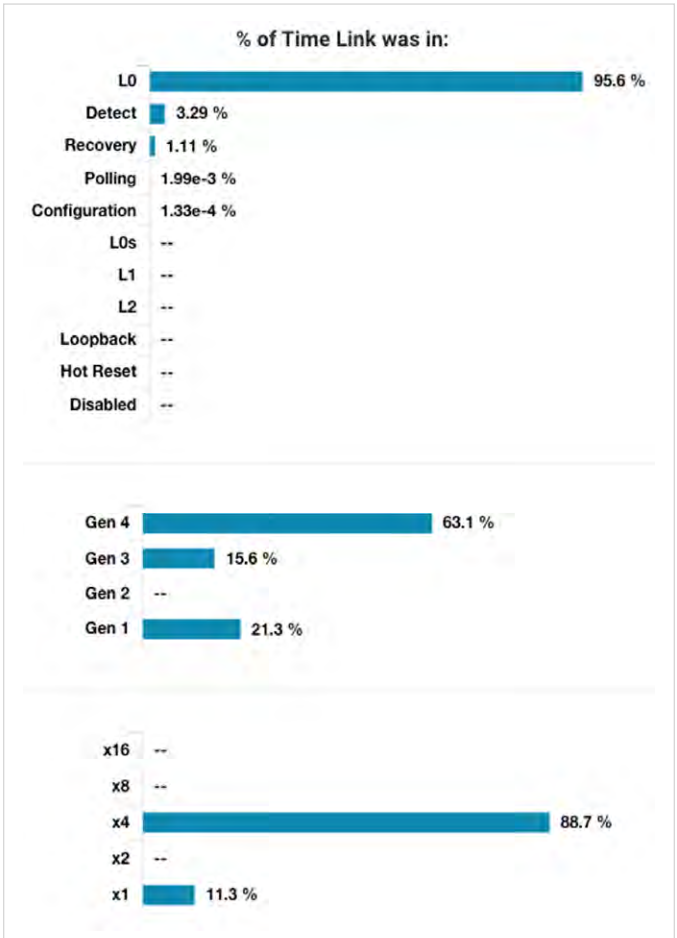
Figure 7. Link training sequence between RC and EP[3].

Whenever performing LTSSM monitoring tests, the recorded logs will always indicate the states achieved from the TMT4 Margin Tester point of view.

Monitoring the Link Training Status State Machine Using the TMT4

During the link training process, the interaction between the DUT and the TMT4 will consist of generation of test patterns that evolve the DUT from the initial **Detect** state through **Polling** and **Configuration**, and then eventually achieving **L0** state indicating that the root complex and the endpoint can successfully communicate with each other.

- For a normal/functioning link as displayed in the LTSSM Test Results below, the TMT4 Is operating as the Root Complex. Under nominal operating conditions, the link training process sees both the DUT and the Link Partner in a **L0** state, only transitioning to a different state upon the link configuration changing any of its most significant Physical Layer attributes: Lane Width, Link Speed or lane inversion, among others.



When the .csv formatted LTSSM log data file is displayed, the user can view the information on the tested DUT as a sequence of the LTSSM states and determine if there is a substantial discrepancy with the expected flow.

Index	Running Time (s)	Time in State	Generation	Link Speed (GT/s)	Link Width	LTSSM State
0	0.000511043	179. us	Gen1	2.5	x1	DETECT_ACT
1	0.042319935	14.6 ms	Gen1	2.5	x1	DETECT_WAIT
2	0.055917735	4.76 ms	Gen1	2.5	x1	POLL_ACTIVE
3	0.056002363	29.6 us	Gen1	2.5	x4	POLL_CONFIG
4	0.056003403	364. ns	Gen1	2.5	x4	CFG_LINKWD_START
5	0.056004522	392. ns	Gen1	2.5	x4	CFG_LINKWD_ACCEPT
6	0.056004651	45.0 ns	Gen1	2.5	x4	CFG_LANENUM_WAIT
7	0.056005611	336. ns	Gen1	2.5	x4	CFG_LANENUM_ACCEPT
8	0.05600574	45.0 ns	Gen1	2.5	x4	CFG_COMPLETE
9	0.056007757	706. ns	Gen1	2.5	x4	CFG_IDLE
10	0.056007979	78.0 ns	Gen1	2.5	x4	L0
11	0.056007997	6.00 ns	Gen1	2.5	x4	RCVRY_LOCK
12	0.056009628	571. ns	Gen1	2.5	x4	RCVRY_RCVRCFG
13	0.056011716	731. ns	Gen1	2.5	x4	RCVRY_SPEED
14	0.056074028	21.8 us	Gen3	8	x1	RCVRY_LOCK
15	0.056074042	5.00 ns	Gen3	8	x1	RCVRY_EQ1
16	0.066559283	3.67 ms	Gen3	8	x4	RCVRY_EQ2
17	0.075163833	3.01 ms	Gen3	8	x4	RCVRY_EQ3
18	0.077706572	890. us	Gen3	8	x4	RCVRY_LOCK
19	0.07770704	164. ns	Gen3	8	x4	RCVRY_RCVRCFG
20	0.077707326	100. ns	Gen3	8	x4	RCVRY_IDLE
21	0.077707609	99.0 ns	Gen3	8	x4	L0
22	0.077707614	2.00 ns	Gen3	8	x4	RCVRY_LOCK
23	0.077708049	152. ns	Gen3	8	x4	RCVRY_RCVRCFG
24	0.077710186	748. ns	Gen3	8	x4	RCVRY_SPEED
25	0.077757192	16.5 us	Gen4	16	x1	RCVRY_LOCK
26	0.077757201	3.00 ns	Gen4	16	x1	RCVRY_EQ1
27	0.088242433	3.67 ms	Gen4	16	x4	RCVRY_EQ2
28	0.098840675	3.01 ms	Gen4	16	x4	RCVRY_EQ3
29	0.102476779	1.97 ms	Gen4	16	x4	RCVRY_LOCK
30	0.102477016	83.0 ns	Gen4	16	x4	RCVRY_RCVRCFG
31	0.102477159	50.0 ns	Gen4	16	x4	RCVRY_IDLE
32	0.102477313	54.0 ns	Gen4	16	x4	L0

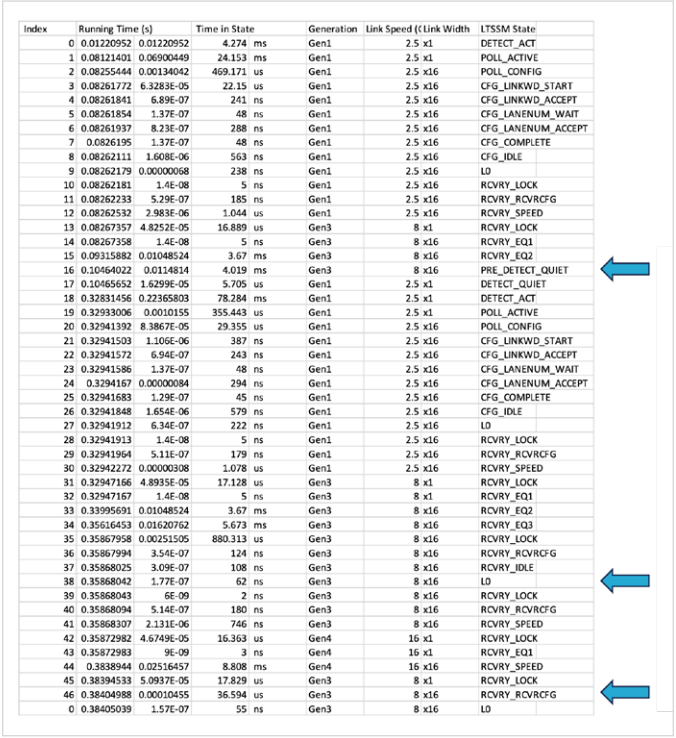
In this sample log, there are three instances where the DUT achieves L0 State, **as seen from the TMT4:**

- First, establishing the basic PCIe Gen1 speed and lane width "by-4" (×4) width moving from **Detect** to **Polling** and finally the **Configuration** state.
- Second, transitioning from PCIe Gen1 (2.5GT/s) to PCIe Gen3 (8GT/s) by first moving into the **Recovery** state and then performing **link equalization.**
- Third, transitioning from PCIe Gen3 (8GT/s) to PCIe Gen4 (16GT/s), also by first moving into the **Recovery** state and then also performing **link equalization.**

Looking at the above three instances, all connected PCIe devices go through an Initial Link Training process to establish basic functionality (the first step), and subsequently they may go through additional link equalization to establish stable and reliable connection among the devices (the second and third steps). Link equalization happens when all devices in the PCIe link can support data rates of PCIe Gen 3 or higher. Link equalization may happen multiple times since PCIe connection has to optimize (link up) the connection at every generation of PCIe above Gen 3, starting with the minimum lane width (×1) and, if successful, expanding to the maximum number of available lanes.

There may be other instances, of course, when one or more of the state transitions observed do not suggest proper operation of the DUT.

For instance, the following log was obtained from a motherboard DUT unable to achieve Gen4 data rate with 16 lanes, so a down-link process was initiated to finally settle at 16 lanes, albeit at Gen3 (8GT/s) speeds.



Index	Running Time (s)	Time in State	Generation	Link Speed ((	Link Width	LTSSM State
0	0.01220952	0.01220952	4.274 ms	Gen1	2.5 x1	DETECT_ACT
1	0.08121401	0.06900449	24.153 ms	Gen1	2.5 x1	POLL_ACTIVE
2	0.08255444	0.00134042	469.171 us	Gen1	2.5 x16	POLL_CONFIG
3	0.08261772	6.3283E-05	22.15 us	Gen1	2.5 x16	CFG_LINKWD_START
4	0.08261841	6.89E-07	241 ns	Gen1	2.5 x16	CFG_LINKWD_ACCEPT
5	0.08261854	1.37E-07	48 ns	Gen1	2.5 x16	CFG_LANENUM_WAIT
6	0.08261937	8.23E-07	288 ns	Gen1	2.5 x16	CFG_LANENUM_ACCEPT
7	0.0826195	1.37E-07	48 ns	Gen1	2.5 x16	CFG_COMPLETE
8	0.08262111	1.608E-06	563 ns	Gen1	2.5 x16	CFG_IDLE
9	0.08262179	0.00000068	238 ns	Gen1	2.5 x16	L0
10	0.08262181	1.4E-08	5 ns	Gen1	2.5 x16	RCVRY_LOCK
11	0.08262233	5.29E-07	185 ns	Gen1	2.5 x16	RCVRY_RCVRCFG
12	0.08262532	2.983E-06	1.044 us	Gen1	2.5 x16	RCVRY_SPEED
13	0.08267357	4.8252E-05	16.889 us	Gen3	8 x1	RCVRY_LOCK
14	0.08267358	1.4E-08	5 ns	Gen3	8 x16	RCVRY_EQ1
15	0.09315882	0.01048524	3.67 ms	Gen3	8 x16	RCVRY_EQ2
16	0.10464022	0.0114814	4.019 ms	Gen3	8 x16	PRE_DETECT_QUIET
17	0.10465652	1.6299E-05	5.705 us	Gen1	2.5 x1	DETECT_QUIET
18	0.32831456	0.22365803	78.284 ms	Gen1	2.5 x1	DETECT_ACT
19	0.32933006	0.0010155	355.443 us	Gen1	2.5 x1	POLL_ACTIVE
20	0.32941392	8.3867E-05	29.355 us	Gen1	2.5 x16	POLL_CONFIG
21	0.32941503	1.106E-06	387 ns	Gen1	2.5 x16	CFG_LINKWD_START
22	0.32941572	6.94E-07	243 ns	Gen1	2.5 x16	CFG_LINKWD_ACCEPT
23	0.32941586	1.37E-07	48 ns	Gen1	2.5 x16	CFG_LANENUM_WAIT
24	0.3294167	0.00000084	294 ns	Gen1	2.5 x16	CFG_LANENUM_ACCEPT
25	0.32941683	1.29E-07	45 ns	Gen1	2.5 x16	CFG_COMPLETE
26	0.32941848	1.654E-06	579 ns	Gen1	2.5 x16	CFG_IDLE
27	0.32941912	6.34E-07	222 ns	Gen1	2.5 x16	L0
28	0.32941913	1.4E-08	5 ns	Gen1	2.5 x16	RCVRY_LOCK
29	0.32941964	5.11E-07	179 ns	Gen1	2.5 x16	RCVRY_RCVRCFG
30	0.32942272	0.00000308	1.078 us	Gen1	2.5 x16	RCVRY_SPEED
31	0.32947166	4.8935E-05	17.128 us	Gen3	8 x1	RCVRY_LOCK
32	0.32947167	1.4E-08	5 ns	Gen3	8 x1	RCVRY_EQ1
33	0.33995691	0.01048524	3.67 ms	Gen3	8 x16	RCVRY_EQ2
34	0.35616453	0.01620762	5.673 ms	Gen3	8 x16	RCVRY_EQ3
35	0.35867958	0.00251505	880.313 us	Gen3	8 x16	RCVRY_LOCK
36	0.35867994	3.54E-07	124 ns	Gen3	8 x16	RCVRY_RCVRCFG
37	0.35868025	3.09E-07	108 ns	Gen3	8 x16	RCVRY_IDLE
38	0.35868042	1.77E-07	62 ns	Gen3	8 x16	L0
39	0.35868043	6E-09	2 ns	Gen3	8 x16	RCVRY_LOCK
40	0.35868094	5.14E-07	180 ns	Gen3	8 x16	RCVRY_RCVRCFG
41	0.35868307	2.131E-06	746 ns	Gen3	8 x16	RCVRY_SPEED
42	0.35872982	4.6749E-05	16.363 us	Gen4	16 x1	RCVRY_LOCK
43	0.35872983	9E-09	3 ns	Gen4	16 x1	RCVRY_EQ1
44	0.3838944	0.02516457	8.808 ms	Gen4	16 x16	RCVRY_SPEED
45	0.38394533	5.0937E-05	17.829 us	Gen3	8 x1	RCVRY_LOCK
46	0.38404988	0.00010455	36.594 us	Gen3	8 x16	RCVRY_RCVRCFG
0	0.38405039	1.57E-07	55 ns	Gen3	8 x16	L0

- At the first blue arrow, the DUT exits the equalization process right back into a "Pre-Detect Quiet" configuration (possibly directed by the Data Link Layer, however this is not an expected PCIe behavior).
- At the second blue arrow, the DUT links up to Gen3 speeds on all 16 lanes using the normal link equalization process, achieving Recovery Idle before transitioning to L0
- At the third blue arrow, however, the DUT appears unable to continue the equalization process on one or more of its lanes and jumps right into L0 from Recovery ReceiverConfig, suggesting a possible mismatch with the Rx settings (and not directed by the Data Link Layer back to Detect).

In order to gain a better understanding of the interaction between the Tx and the Rx settings, the LTSSM monitor can help in this instance to provide possible causes for the unexpected behavior. Additional LTSSM functionality includes the use of hardware triggers to start and stop analog data collection with a real time oscilloscope.

Conclusion

The use of the Tektronix TMT4 Margin Tester as a reliable Link Partner helps provides accurate information about the state of the link with the observation of the LTSSM states during the link training and equalization processes, and the use of hardware trigger signals further supports the acquisition of waveforms to investigate possible areas where link inconsistencies may be resolved.

References:

1. PCI-SIG Specifications
2. Texas Instruments "PCIe Link Training Overview", SNLA415 – AUGUST 2022
3. Simon Southwell "PCI Express Primer Overview – Physical Layer"

Contact Information:

Australia 1 800 709 465
Austria* 00800 2255 4835
Balkans, Israel, South Africa and other ISE Countries +41 52 675 3777
Belgium* 00800 2255 4835
Brazil +55 (11) 3530-8901
Canada 1 800 833 9200
Central East Europe / Baltics +41 52 675 3777
Central Europe / Greece +41 52 675 3777
Denmark +45 80 88 1401
Finland +41 52 675 3777
France* 00800 2255 4835
Germany* 00800 2255 4835
Hong Kong 400 820 5835
India 000 800 650 1835
Indonesia 007 803 601 5249
Italy 00800 2255 4835
Japan 81 (3) 6714 3086
Luxembourg +41 52 675 3777
Malaysia 1 800 22 55835
Mexico, Central/South America and Caribbean 52 (55) 88 69 35 25
Middle East, Asia, and North Africa +41 52 675 3777
The Netherlands* 00800 2255 4835
New Zealand 0800 800 238
Norway 800 16098
People's Republic of China 400 820 5835
Philippines 1 800 1601 0077
Poland +41 52 675 3777
Portugal 80 08 12370
Republic of Korea +82 2 565 1455
Russia / CIS +7 (495) 6647564
Singapore 800 6011 473
South Africa +41 52 675 3777
Spain* 00800 2255 4835
Sweden* 00800 2255 4835
Switzerland* 00800 2255 4835
Taiwan 886 (2) 2656 6688
Thailand 1 800 011 931
United Kingdom / Ireland* 00800 2255 4835
USA 1 800 833 9200
Vietnam 12060128

*** European toll-free number. If not accessible, call:** +41 52 675 3777

Rev. 02.2022

Find more valuable resources at TEK.COM

081123 SBG 64W-74023-0