Hunting PCIE Flow Control Bugs with Gun and Camera

Abstract

Debugging PCIE Gen 3 links is similar to any bug hunt: finding bugs is a lot easier if you have a map.

In the Tektronix Logic Protocol Analyzer (LPA), new visualizations provide high-ground visibility to quickly identify PCIE flow control problems. To debug PCIE flow control, a tool must account for six separate buffers across the Transaction and Data Link layers in the up and down directions on eight separate virtual channels. This white paper describes the LPA's innovative Bird's Eye View, how it provides high-ground visibility to target flow control bugs and its powerful tools to locate and display detailed flow control information.

Finding a Scent

One of the most difficult debugging challenges in PCIE involves flow control. In stark contrast to PCI, in which flow control was handled through sideband signals, PCIE flow control is an in-band, point-to-point mechanism using both DLLPs (UpdateFC packets) and TLPs to update flow control state between the ends of a single Link (not the ends of an entire set of Links). Through a scheme of debits and credits, the components at each end of a Link determine whether to continue transmitting. In addition, ordering rules permit the "passing" of high priority traffic past lower priority traffic. The PCIE specification does not require any specific design for a flow control scheme; rather it specifies two quantities the transmitter gating function must track, Credits Consumed and Credit Limit as well as two quantities the receiver must track, Credits Allocated and Credits Received. Using these four variables, designers are permitted a variety of possible algorithms to tune their flow control gating functions. See Sidebar for an excerpt from the PCIE Specification for an idea of the complexity of tracking these values.

When the link is not being managed correctly due to a flaw in the flow control design or a bug in its implementation, the effects can vary: dropped packets may simply reduce overall quality of service (QoS) or they may result in link degradation and deadlock. In the worst cases, dropped packets result in silent data corruption – for example, a dropped Posted Write could mean bad data is written (or good information is not written) to the disk.

Hunting PCIE Flow Control Bugs with Gun and Camera

Because the flow control mechanism relies on both DLLPs and TLPs, the bug hunter must account for credits and debits across these two layers of the protocol. Complicating matters further, the specification divides flow control packets into three categories: P, NP and CPL types corresponding to Posted, Non-Posted and ComPLetion type requests. Posted Requests include Messages and Memory Writes; Non-Posted requests include All Reads, I/O Writes, Configuration Writes and AtomicOps; Completions are associated with their corresponding NP Requests.

Within each category, flow control information is further sub-divided into HDR (header) and DATA types. Flow control is inherently a bi-directional operation: accounting must occur on both sides of the link. And if that weren't complicated enough, the specification allows for separate flow control operations on each of the eight Virtual Channels (VC).

Even limiting the situation to VC0 (the only required channel in which flow control must be implemented), manually tracking just the 12 possible subtypes is not straightforward. Flow control credits are visible through the HdrFC and DataFC fields (in DLLPs) but debits are calculated for each type of TLP that participates. The problem of accounting for TLP debits is discussed a little further on.

Excerpt from the PCIE Gen 3 Version 1 Specification for Flow Control

CREDITS_CONSUMED (At Transmitter)

- Count of the total number of FC units consumed by TLP Transmissions made since Flow Control initialization, modulo $2^{[Field Size]}$ (where [Field Size] is 8 for PH, NPH, and CPLH and 12 for PD, NPD and CPLD).
- Set to all 0's at interface initialization.
- Updated for each TLP the Transaction Layer allows to pass the Flow Control gate for Transmission.
- Updated as shown:
  CREDITS_CONSUMED :=(CREDITS_CONSUMED + Increment) mod $2^{[Field Size]}$.
  Where Increment is the size in FC credits of the corresponding part of the TLP passed through the gate, and [Field Size] is 8 for PH, NPH, and CPLH and 12 forPD, NPD, and CPLD.

CREDITS_ALLOCATED (At Receiver)
Count of the total number of credits granted to the Transmitter since initialization, modulo $2^{[Field Size]}$ (where [Field Size] is 8 for PH, NPH, and CPLH and 12 for PD, NPD and CPLD).

- Initially set according to the buffer size and allocation policies of the Receiver.
- This value is included in the Init FC and Update FC DLLPs (see Section 3.4).
- Incremented as the Receiver Transaction Layer makes additional receive buffer space available by processing Received TLPs.
- Updated as shown:
  CREDITS_ALLOCATED :=(CREDITS_ALLOCATED + Increment) mod $2^{[Field Size]}$.
  Where Increment corresponds to the credits made available, and [Field Size] is 8 forPH, NPH, and CPLH and 12 for PD, NPD, and CPLD.

CREDITS_RECEIVED (Optional – for optional error check described below)

- Count of the total number of FC units consumed by valid TLPs Received since Flow Control initialization, modulo $2^{[Field Size]}$ (where [Field Size] is 8 for PH, NPH, and CPL Hand 12 for PD, NPD and CPLD).
- Set to all 0's at interface initialization.

The values transmitted in flow control DLLPs do not contain the current available receiver buffering, but rather contain the cumulative total credits that the initialization (using modulo 2^{field_size} arithmetic). An initial credit grant is communicated using InitFCDLLPs and subsequent grants are communicated using UpdateFC DLLPs.

Tracking TLPs is even more troublesome because the values are not transmitted with a packet; they are calculated based on the type of TLP, the amount of data it carries, the Traffic Class (TC) assigned to the TLP, and so forth. For example, per the PCIE specification: I/O and Configuration Write Requests consume 1 NPH (Non-Posted Request Header) and 1 NPD (Non-Posted Data payload) credit each.

To keep track of all of these flow control values, the bug hunter must do the following:

1. Capture the Init FCs as the link is being established.
2. Track all Update FCs as they are issued.
3. If multiple VCs are in use, track the Traffic Class (TC) to Virtual Channel (VC) mapping.
4. Track all TLPs that participate in flow control, calculate the appropriate credits and debit the accounts properly.
5. Track each buffer's (PH, PD, NPH, NPD, ClpH, ClpD) current credit capacity at any given time in the acquisition.

For anything but the most trivial acquisitions, tracking and calculating all of these values by hand is time consuming, prone to error and expensive. But what if an instrument could perform all of these tasks for the bug hunter? Would that reduce the burden on the bug hunter? Perhaps, but simply displaying these values within the stream of packets still requires enormous effort to isolate where credit violations may have occurred. Displaying and calculating the proper values is only one of several problems the instrument has to solve.

In addition to acquiring the InitFCs, all UpdateFCs and all pertinent TLPs, the instrument's acquisition must still be long enough to capture the point of interest where flow control issues appear. For example, it is possible the problem occurs well after the initialization phase, perhaps several seconds or minutes into the circuit's operation. Even the 16GB of acquisition memory available in the LPA, twice the amount available from the closest competitor, and using aggressive hardware filtering to only capture Flow Control DLLPs and TLPs, may not be sufficient to capture the necessary time window.

Creating a Bug Trap, Relatively Speaking

We call the process of capturing every credit and debit starting from InitFCs "absolute flow control accounting." In this scenario, the initial credit grant is known exactly and every change, whether from a TLP or UpdateFC DLLP, is tracked and accounted for. Although this form of flow control accounting is accurate it has two limitations:

- It depends on capturing data during the initialization phase.
- It is limited to the length of acquisition memory.

We ran a couple of experiments to determine how long a time window the LPA could acquire. Through creative storage techniques and filtering in the LPA, capturing only INITFCs, Update FCs and TLPs, we were able to extend the time window on one of the test systems to as long as 24 minutes. While this time window may be more than sufficient to contain flow control problems, it isn't guaranteed to be long enough. Unfortunately, the length of the time window isn't constant: it depends on how frequently a system issues UpdateFCs; different systems issue UpdateFCs at different rates. For example, on a different system we tested, UpdateFCs were issued 10 times more frequently, reducing the overall time window to just 2.5 minutes.

If the problem occurs at some arbitrary time much later in the acquisition, outside of the storage time window, can we still debug flow control without the buffer initialization values? What if the LPA could calculate flow control credits within any given region irrespective of "knowing" the real absolute value of the transmitting buffers?

There are a couple of ways around identifying flow control issues without capturing the initialization:

1. Ask the bug hunter to provide starting values for the credit buffers for the start of the acquisition.
2. Assume a starting value and use the MAX credit allocation value as the buffer size.

The first alternative is problematic in a couple of ways: a) the bug hunter is busy — the system shouldn't require a lot of work to get setup, but more importantly, b) whatever number the bug hunter provides (other than "INFINITE") reflects an expected value for the initial allocated credits buffers. Unfortunately, the allocated credits values are only half of the equation; the bug hunter still doesn't know the current values of the credits consumed. In addition, because the PCIE protocol allows a receiver to add or remove buffers "on the fly," the actual allocated credits buffer may differ from the expected value at the point when the acquisition was taken.

Assuming a starting value for the start of the acquisition, the second work-around, is a perfectly reasonable simplification for most flow control debugging situations. In this simplified model, the LPA assumes a starting value of ½ the maximum buffer size specified in the PCIE Gen 3 specification (128 for Hdr credits, 2048 for Data credits). The Credits Consumed values are presumed to be zero at the beginning of the acquisition just as in the absolute credit tracking case.

A few problems are introduced by this simplification:

1. If the actual Credits Consumed values at the start of the acquisition are not zero, they may go "negative," an event that obviously cannot occur in the absolute tracking case.
2. If the actual Credits Allocated buffers are much larger than MAX/2, the system may provide false indications of buffer overflows.
3. If the actual Credits Allocated buffers are much lower than MAX/2, the system may fail to report errors.

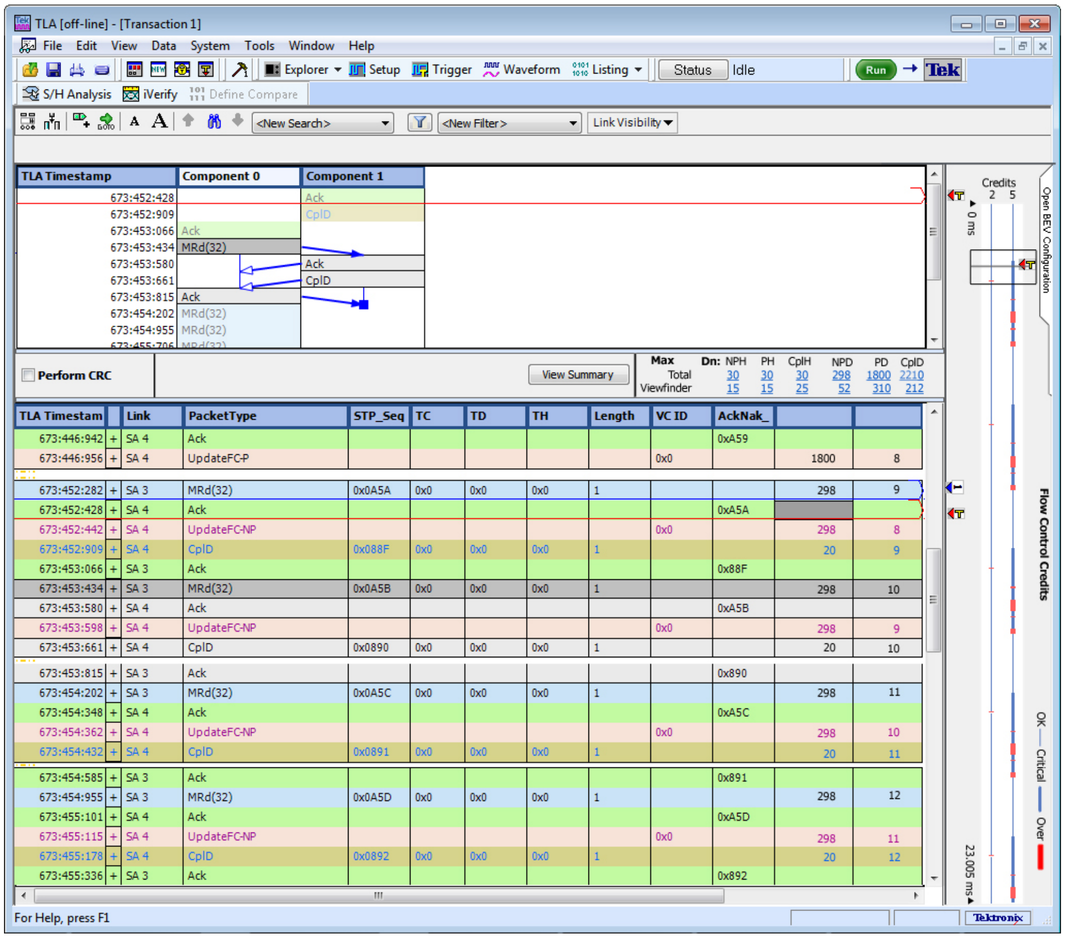
The first problem is relatively trivial: once the bug hunter understands the reason Credits Consumed values go negative, the sign can be ignored.

In the second and third cases, the problems are reduced by allowing the bug hunter to set a threshold for buffer overflows. By establishing a threshold (for example, when Credits Consumed equals 80% of MAX), the bug hunter provides a "high water mark" above which the system can assume there is a problem.

Hunting PCIE Flow Control Bugs with Gun and Camera

Mapping the Terrain: Getting the Bird's Eye View

As part of the Transaction Window, Figure 1, the LPA provides the high ground view of all of the data captured in the entire acquisition – the Bird's Eye View (BEV). The Flow Control Visualization in the Bird's Eye View displays the current relative flow control buffer sizes in a way that clearly indicates if there is a problem.


TLA [off-line] - [Transaction 1]
File Edit View Data System Tools Window Help
Explorer Setup Trigger Waveform Listing Status Idle Run Tek
S/H Analysis iVerify Define Compare
<New Search> <New Filter> Link Visibility
Credits 2 5
0 ms
Open BEV Configuration
Flow Control Credits
OK — Critical — Over
23.005 ms
View Summary
Tektronix

TLA Timestamp	Component 0	Component 1
673:452:428		Ack
673:452:909		CplD
673:453:066	Ack	
673:453:434	MRd(32)	
673:453:580		Ack
673:453:661		CplD
673:453:815	Ack	
673:454:202	MRd(32)	
673:454:955	MRd(32)	

Perform CRC

Max	Dn: NPH	PH	CplH	NPD	PD	CplD
Total	30	30	30	298	1800	2210
Viewfinder	15	15	25	52	310	212

TLA Timestam		Link	PacketType	STP_Seq	TC	TD	TH	Length	VC ID	AckNak_		
673:446:942	+	SA 4	Ack							0xA59		
673:446:956	+	SA 4	UpdateFC-P						0x0		1800	8
673:452:282	+	SA 3	MRd(32)	0x0A5A	0x0	0x0	0x0	1			298	9
673:452:428	+	SA 4	Ack							0xA5A		
673:452:442	+	SA 4	UpdateFC-NP						0x0		298	8
673:452:909	+	SA 4	CplD	0x088F	0x0	0x0	0x0	1			20	9
673:453:066	+	SA 3	Ack							0x88F		
673:453:434	+	SA 3	MRd(32)	0x0A5B	0x0	0x0	0x0	1			298	10
673:453:580	+	SA 4	Ack							0xA5B		
673:453:598	+	SA 4	UpdateFC-NP						0x0		298	9
673:453:661	+	SA 4	CplD	0x0890	0x0	0x0	0x0	1			20	10
673:453:815	+	SA 3	Ack							0x890		
673:454:202	+	SA 3	MRd(32)	0x0A5C	0x0	0x0	0x0	1			298	11
673:454:348	+	SA 4	Ack							0xA5C		
673:454:362	+	SA 4	UpdateFC-NP						0x0		298	10
673:454:432	+	SA 4	CplD	0x0891	0x0	0x0	0x0	1			20	11
673:454:585	+	SA 3	Ack							0x891		
673:454:955	+	SA 3	MRd(32)	0x0A5D	0x0	0x0	0x0	1			298	12
673:455:101	+	SA 4	Ack							0xA5D		
673:455:115	+	SA 4	UpdateFC-NP						0x0		298	11
673:455:178	+	SA 4	CplD	0x0892	0x0	0x0	0x0	1			20	12
673:455:336	+	SA 3	Ack							0x892		

For Help, press F1

Figure 1. A snapshot of the LPA Transaction Window with the BEV to the right.

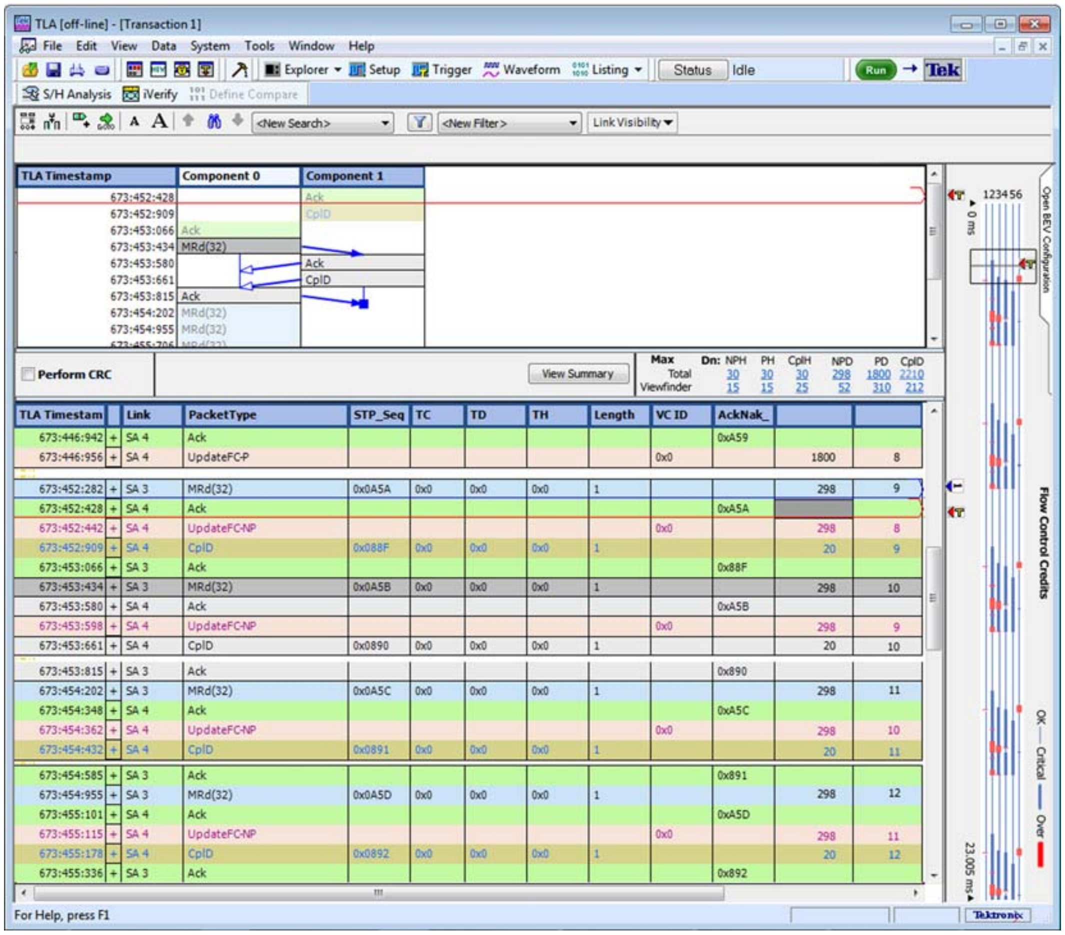
Even in this thumbnail image, the Flow Control Visualization in the BEV clearly shows a problem on both the down and up sides of the link: on one side, the Flow Control Credits Consumed values are increasing over time until they pass a threshold and indicate an overflow; on the other side, the Credits Consumed are abruptly passing the threshold.

When the Credits Consumed values stay around zero (meaning there has been a reasonable balance of credits consumed and credits allocated) the visualization shows a thin blue line. As the values become unbalanced, the line thickens but remains blue. Finally, as the balance exceeds the threshold defined by the bug hunter (by default, 80% of the PCIE Specification for FC limits) the line is thickened and painted red.

By adjusting this threshold the bug hunter can "fine tune" the sensitivity of the visualization.

Hunting PCIE Flow Control Bugs with Gun and Camera

Figure 1 shows only two lines, but the visualization permits as many as six different values. Consider a problem that is isolated to one direction: the values might be just the Header and Data buffers for that direction, as shown in Figure 2, or in the case of payloads, perhaps only the Data values are most interesting to the bug hunter.


TLA [off-line] - [Transaction 1]
File Edit View Data System Tools Window Help
Explorer Setup Trigger Waveform Listing Status Idle Run Tek
S/H Analysis iVerify Define Compare
<New Search> <New Filter> Link Visibility
123456
0 ms
Open BEV Configuration
Flow Control Credits
OK — Critical — Over
23.005 ms
Perform CRC
View Summary
For Help, press F1
Tektronix

TLA Timestamp	Component 0	Component 1
673:452:428		Ack
673:452:909		CplD
673:453:066	Ack	
673:453:434	MRd(32)	
673:453:580		Ack
673:453:661		CplD
673:453:815	Ack	
673:454:202	MRd(32)	
673:454:955	MRd(32)	

Max	Dn: NPH	PH	CplH	NPD	PD	CplD
Total	30	30	30	298	1800	2210
Viewfinder	15	15	25	52	310	212

TLA Timestam		Link	PacketType	STP_Seq	TC	TD	TH	Length	VC ID	AckNak_		
673:446:942	+	SA 4	Ack							0xA59		
673:446:956	+	SA 4	UpdateFC-P						0x0		1800	8
673:452:282	+	SA 3	MRd(32)	0x0A5A	0x0	0x0	0x0	1			298	9
673:452:428	+	SA 4	Ack							0xA5A		
673:452:442	+	SA 4	UpdateFC-NP						0x0		298	8
673:452:909	+	SA 4	CplD	0x088F	0x0	0x0	0x0	1			20	9
673:453:066	+	SA 3	Ack							0x88F		
673:453:434	+	SA 3	MRd(32)	0x0A5B	0x0	0x0	0x0	1			298	10
673:453:580	+	SA 4	Ack							0xA5B		
673:453:598	+	SA 4	UpdateFC-NP						0x0		298	9
673:453:661	+	SA 4	CplD	0x0890	0x0	0x0	0x0	1			20	10
673:453:815	+	SA 3	Ack							0x890		
673:454:202	+	SA 3	MRd(32)	0x0A5C	0x0	0x0	0x0	1			298	11
673:454:348	+	SA 4	Ack							0xA5C		
673:454:362	+	SA 4	UpdateFC-NP						0x0		298	10
673:454:432	+	SA 4	CplD	0x0891	0x0	0x0	0x0	1			20	11
673:454:585	+	SA 3	Ack							0x891		
673:454:955	+	SA 3	MRd(32)	0x0A5D	0x0	0x0	0x0	1			298	12
673:455:101	+	SA 4	Ack							0xA5D		
673:455:115	+	SA 4	UpdateFC-NP						0x0		298	11
673:455:178	+	SA 4	CplD	0x0892	0x0	0x0	0x0	1			20	12
673:455:336	+	SA 3	Ack							0x892		

Figure 2. Flow control Visualization with additional values.

Regardless of the values of interest, the bug hunter adjusts the flow control BEV using a simple configuration panel to quickly reveal patterns in the data.

The BEV displays the state of the flow control credits from the start of the acquisition to the end; if an INITFC was captured, the lines represent absolute credit tracking, if not, the lines represent relative credit tracking. The visualization provides immediate at-a-glance indications of issues; the BEV provides additional tools to the intrepid hunter.

Taking a Closer Look at the Bug

Just clicking on a spot in the BEV updates the adjacent views to show the packets flowing on the link at that time. Clicking on a red region scrolls the adjacent detail information to the first instance of the packet causing the overflow. Using the BEV as a navigational aid, the hunter quickly moves around the acquisition exploring potential problems revealed in the Packet View's patterns of data.

In addition to navigating through the data, the bug hunter uses the BEV to collect statistics about the data occurring in particular areas of the acquisition.

Within the BEV is an adjustable rectangular region called the Viewfinder.

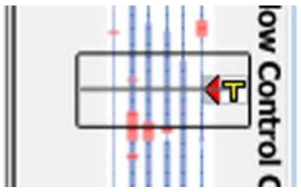
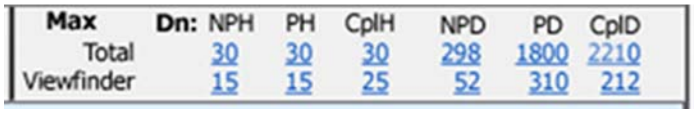


Figure 3. Viewfinder Region in BEV.

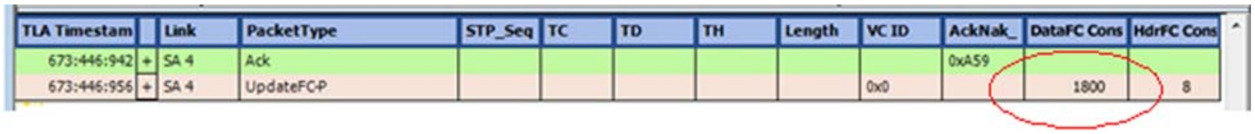
Using the Viewfinder, the bug hunter displays statistics about the flow control credits occurring within the region. Statistics are displayed adjacent to the BEV in the Statistics Panel.



Max	Dn: NPH	PH	CplH	NPD	PD	CplD
Total	30	30	30	298	1800	2210
Viewfinder	15	15	25	52	310	212

Figure 4. Statistics Panel for flow control.

The bug hunter scans two sets of values in the table: the maximum Credits Consumed across the entire acquisition (Total) and the maximum Credits Consumed within the Viewfinder region. A click on one of the blue underlined (hyperlinked) elements scrolls the adjacent Packet View to the location in the trace where that value occurs.



TLA Timestam		Link	PacketType	STP_Seq	TC	TD	TH	Length	VC ID	AckNak_	DataFC Cons	HdrFC Cons
673:446:942	+	SA 4	Ack							0xA59		
673:446:956	+	SA 4	UpdateFC-P						0x0		1800	8

Figure 5. First Instance of value from statistics panel.

Clicking on the hyperlinked value always brings the bug hunter to the first instance of that value. Using the Viewfinder, in combination with displaying the different credit types, the bug hunter quickly isolates the moment in time and the type of credit causing the problem.

Closing the Trap (Quickly)

Historically, the benefit of the Tektronix Logic Analyzer (TLA) has been its ability to correlate across disparate buses. In the LPA this tradition continues through the capability of acquiring information on several links at once. Issues with flow control on one link might be the root cause of problems occurring on a second link. No matter how many links are captured in the LPA, the BEV can display the information.

As an example, assume the hunt is for a problem with flow control, but it is not clear whether the problem is on one link or another. If the system is instrumented with two TLA7SAxx modules, both links will appear by default in the Transaction Window, and both will be analyzed in the BEV. By default, the visualization shows the maximum Credits Consumed values for all credit types on both links. If there are no violations (if the BEV displays thin blue lines), the bug hunter can be confident flow control was working within that acquisition across all of the links.

If on the other hand, there appears to be a problem (red lines or thick lines in the visualization), clicking the line itself (in the BEV) or the associated hyperlink (in the statistics panel) will display the offending packet on the appropriate link. Alternatively, the bug hunter uses the BEV's configuration panel to quickly isolate information to a specific link.

Efficiently stomping bugs requires a speedy and fluid experience. How long does the system take to process and display all this information? For modest acquisitions (< 16M symbols), the LPA posts the initial information to the Transaction Window within 500 ms. Even with the deepest acquisition of 160MSyms, the Transaction Window and the BEV display the first piece of information within a few seconds of triggering.

The BEV is designed to display information as it comes in. There are three main advantages to this approach:

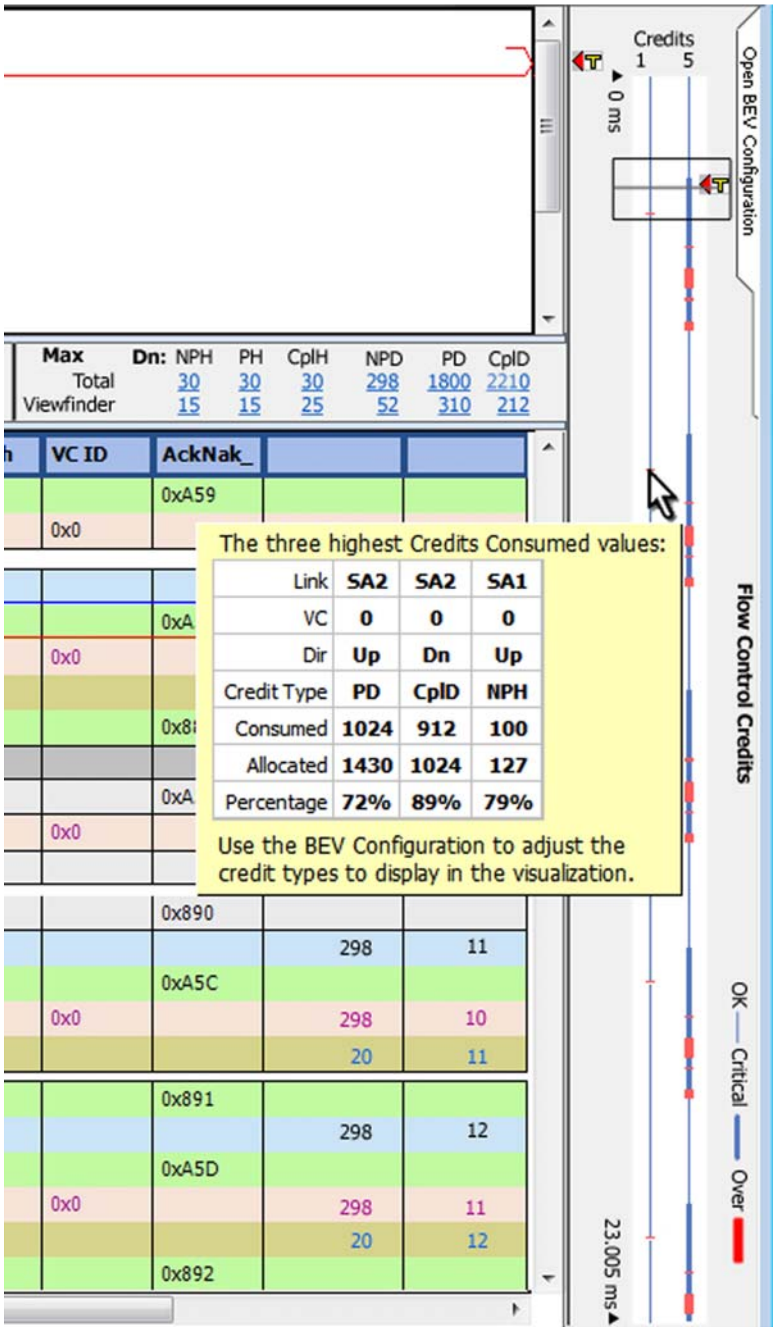
1. There is no time wasted preprocessing the data.
2. The hunt can continue through the acquisition even if the processing is not complete.
3. If the trigger did not work as expected, the bug hunter can interrupt the BEV display processing and take a new acquisition.

Rather than having to wait for several minutes until the final data is displayed, the bug hunter can rework the trigger, take additional acquisitions and repeat the process until the instrument has triggered on the desired information.

Even as the system processes the acquisition to display items of interest in the BEV, the hunt can continue using more traditional methods such as filters, searches and the like. Eventually, once the entire acquisition has been processed, the response time for navigating through the data is consistently less than half a second.

Updating the BEV to display different credit types, isolate links, or adjust the threshold value occurs immediately, allowing the bug hunter to rapidly and fluidly view different aspects of the flow control values.

Brushing the mouse cursor over one of the visualizations in the BEV displays a report of the top three Credits Consumed values for the time region under the cursor.



Max	Dn: NPH	PH	CplH	NPD	PD	CplD
Total	30	30	30	298	1800	2210
Viewfinder	15	15	25	52	310	212

VC ID	AckNak_		
	0xA59		
0x0			
	0xA		
0x0			
	0x8		
	0xA		
0x0			
	0x890		
		298	11
	0xA5C		
0x0		298	10
		20	11
	0x891		
		298	12
	0xA5D		
0x0		298	11
		20	12
	0x892		

The three highest Credits Consumed values:

Link	SA2	SA2	SA1
VC	0	0	0
Dir	Up	Dn	Up
Credit Type	PD	CplD	NPH
Consumed	1024	912	100
Allocated	1430	1024	127
Percentage	72%	89%	79%

Use the BEV Configuration to adjust the credit types to display in the visualization.

Figure 6. Rollover Report of Credits Consumed at current time in BEV.

By establishing a filter in the adjacent Packet View to display only InitFC and UpdateFCs in one direction and TLPs in the other, the bug hunter improves the display of flow control information. The hunter scrolls through the Packet View to see the change in Credits Consumed values in the area around an overflow event.

Conclusion

This white paper has covered a lot of ground. It has described in detail the use of the Bird's Eye View–a completely new visualization to investigate flow control. Not only does the BEV provide a full-acquisition view of the trace, it provides new analysis that is only available in the LPA.

The BEV has been designed to quickly and fluidly display information as the hunter needs it, regardless of the scale of the acquisition – whether on one link or across many.

In addition, the BEV is completely integrated into the Transaction Window with detailed and statistical views into the PCIE protocol data. By seeing a potential problem in the BEV, then using the BEV to navigate to specific data, the bug hunter quickly isolates the underlying packets causing the problem.

The BEV and the LPA provide the intrepid bug hunter with a map of previously uncharted territory, the proverbial "high-ground" from which to see patterns of flow control information otherwise invisible in the data.

About the Author

Leo Frishberg is the Principal Architect of the User Experience for the Logic Analyzer Product Line at Tektronix, Inc.