24G SAS: What’s New and How to Test It

March 18, 2020
Today’s Presenters

Mike Micheletti
Senior Product Manager, Teledyne LeCroy

Cameron Brett
President, SCSI Trade Association;
Director of Enterprise Marketing, KIOXIA America, Inc.
Agenda

- Introduction
- What’s New in 24G SAS
  - “Packet-Mode” Structures
  - 24G Link Training
  - Forward Error Correction (FEC)
  - Active Phy Transmitter Adjustment (APTA)
  - Persistent Connections

March 18, 2020
STA At-a-Glance

- Formed over 20 years ago as industry alliance of key ecosystem partners
- STA markets and promotes Small Computer System Interface (SCSI) and Serial Attached SCSI (SAS)
- We welcome all to join STA to promote and contribute to the evolution of SAS

Learn more at www.scsita.org/content
SAS Remains Primary Storage Interface

SAS Infrastructure Enables >62% of Enterprise Storage Drives and >84% of Enterprise Storage Capacity thru 2023

Source: IDC, Dec 2019
24G SAS Highlights

Physical Layer Enhancements

- 2.4 GB/s effective single-lane bandwidth (22.5 Gbaud rate)
- Enhanced 20-bit Forward Error Correction (FEC)
- SAS-4 transmitter training algorithm

- Higher throughput and IOPs performance
- More robust data reliability and connectivity
- Better signal integrity via continuous optimal signal tuning

Protocol & Block Level Enhancements

- Fairness enhancements
- Storage intelligence and persistent connections
- SMP priorities

- Performance consistency across large and mixed protocol topologies
- Improves SSD efficiency, latency, and QoS
- Determines priority for management-class communications
Who is Teledyne LeCroy?

- Measures electrical signals to allow engineers to Visually Observe how electronic circuits are behaving

- Records Data traffic from digital systems to allow System Engineers to debug data-communications at the higher layers

Digital Oscilloscope

Protocol Analyzer
How to Test What’s New in 24G SAS

- “Packet-Mode” Structures
- 24G Link Training
- Forward Error Correction (FEC)
- Active Phy Transmitter Adjustment (APTA)
- Persistent Connections
### “DWORD” mode vs “PACKET” mode

**TEST UNIT READY Command**

#### DWORD mode (≤12G)

<table>
<thead>
<tr>
<th>II</th>
<th>T1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOF</td>
<td></td>
</tr>
<tr>
<td>06F3C259</td>
<td></td>
</tr>
<tr>
<td>00BB13F</td>
<td></td>
</tr>
<tr>
<td>00000000</td>
<td></td>
</tr>
<tr>
<td>00000000</td>
<td></td>
</tr>
<tr>
<td>5103FFFF</td>
<td></td>
</tr>
<tr>
<td>00000000</td>
<td></td>
</tr>
<tr>
<td>00000000</td>
<td></td>
</tr>
<tr>
<td>00000000</td>
<td></td>
</tr>
<tr>
<td>00000000</td>
<td></td>
</tr>
<tr>
<td>00000000</td>
<td></td>
</tr>
<tr>
<td>00000000</td>
<td></td>
</tr>
<tr>
<td>CRC: FC15EDE4</td>
<td></td>
</tr>
</tbody>
</table>

#### PACKET mode (24G)

<table>
<thead>
<tr>
<th>I3</th>
<th>T3</th>
</tr>
</thead>
<tbody>
<tr>
<td>[01b]SOF</td>
<td></td>
</tr>
<tr>
<td>ALIGN 1</td>
<td></td>
</tr>
<tr>
<td>ALIGN 2</td>
<td></td>
</tr>
<tr>
<td>[10b]06D6D87</td>
<td></td>
</tr>
<tr>
<td>00FAA838</td>
<td></td>
</tr>
<tr>
<td>00001000</td>
<td></td>
</tr>
<tr>
<td>00000000</td>
<td></td>
</tr>
</tbody>
</table>

One SPL Segment

- [CRC: 8A6F0732]
- [01b]B_EOF(2)

---

March 18, 2020
SAS-4 SPL “Packet Mode”

- Each block contains:
  - 2-bit SPL Packet Header
  - 128-bit SPL Packet Payload
  - 20-bit Forward Error Correction (FEC) Information
SAS-4 SPL Packet

Table 54 – SPL PACKET HEADER field

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00b</td>
<td>The SPL packet payload contains a scrambled idle segment.</td>
</tr>
<tr>
<td>01b</td>
<td>The SPL packet payload descriptor contains a primitive segment containing the following that are not scrambled: a) primitives; b) binary primitives; c) primitive parameters; or d) extended binary primitive.</td>
</tr>
<tr>
<td>10b</td>
<td>The SPL packet payload contains a scrambled: a) segment of an SSP frame; b) segment of an SMP frame; c) segment of an address frame; d) segment of an STP frame; or e) idle dword segment.</td>
</tr>
<tr>
<td>11b</td>
<td>The SPL packet payload contains a scrambled idle segment.</td>
</tr>
</tbody>
</table>

*a* The selection of 00b or 11b is as described in 6.8.3.
SAS-4 SPL Packet
SAS-4 SPL Packet Payload

8b/10b = 20% Overhead
128b/150b = 17% Overhead
SPL Packet Types

128 bits = 4 DWORDS

- SPL Packet payload
- SPL Packet payload
- SPL Packet payload
- SPL Packet payload
- SPL Packet payload

- Primitive 0
  - 01b
  - 28 bits
  - 30 bits

- Primitive 1
  - 01b
  - 28 bits
  - 30 bits

- Primitive 2
  - 01b
  - 28 bits
  - 30 bits

- Primitive 3
  - 01b
  - 28 bits
  - 30 bits

- Binary primitive 0
  - 01b
  - 28 bits
  - 30 bits

- Binary primitive 1
  - 01b
  - 28 bits
  - 30 bits

- Binary primitive 2
  - 01b
  - 28 bits
  - 30 bits

- Binary primitive 3
  - 01b
  - 28 bits
  - 30 bits

- Primitive parameter
  - 01b
  - 28 bits
  - 30 bits

- Extended binary primitive
  - 01b
  - 28 bits
  - 30 bits

- Scrambled idle segment
  - 00b or 11b
  - 128 bits

- SPL frame segment or idle dword segment
  - 10b
  - 128 bits
SPL Data/Address Frame Segment

Contains 4 scrambled DWORDS

Table 54 – SPL PACKET HEADER field

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00b</td>
<td>The SPL packet payload contains a scrambled idle segment.</td>
</tr>
<tr>
<td>01b</td>
<td>The SPL packet payload descriptor contains a primitive segment containing the following that are not scrambled: a) primitives; b) binary primitives; c) primitive parameters; or d) extended binary primitive.</td>
</tr>
<tr>
<td>10b</td>
<td>The SPL packet payload contains a scrambled: a) segment of an SSP frame; b) segment of an SMF frame; c) segment of an address frame; d) segment of an STP frame; or e) idle dword segment.</td>
</tr>
<tr>
<td>11b</td>
<td>The SPL packet payload contains a scrambled idle segment.</td>
</tr>
</tbody>
</table>

\[Note\]: The selection of 00b or 11b is as described in 6.8.3.
SPL Data/Address Frame Segment

Contains 4 scrambled DWORDS
SPL Packet Type: Primitive

Identified by Packet Header = 01b

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00b</td>
<td>The SPL packet payload contains a scrambled idle segment.</td>
</tr>
</tbody>
</table>
| 01b | The SPL packet payload descriptor contains a primitive segment containing the following that are not scrambled:  
|     | a) primitives;  
|     | b) binary primitives  
|     | c) primitive parameters; or  
|     | d) extended binary primitive. |
| 10b | The SPL packet payload contains a scrambled:  
|     | a) segment of an SSP frame;  
|     | b) segment of an SMF frame;  
|     | c) segment of an address frame;  
|     | d) segment of an STP frame; or  
|     | e) idle dword segment. |
| 11b | The SPL packet payload contains a scrambled idle segment. |

\[\text{a} \quad \text{The selection of 00b or 11b is as described in 6.8.3.}\]
SAS-4 Primitive SPL Segment

Table 57 – Primitive segment primitive data character placement

<table>
<thead>
<tr>
<th>Byte\Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>n+1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>n+2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>n+3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

- Represents a K28.5 if set to 00b or a K28.3 if set 11b.
SPL Packet Type: Primitive

**Primitive Segment**

**PRIMITIVE PARAMETER SELECT**

- “00b” = K28.5
- “11b” = K28.3

### Decoded Primitive Description

<table>
<thead>
<tr>
<th>Index</th>
<th>Description</th>
<th>PH</th>
<th>Payload (128-bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OPENACCEPT - R_RDY NORMAL - R_RDY NORMAL - R_RDY NORMAL</td>
<td>01b</td>
<td>(0b) -24E -24E -24E</td>
</tr>
</tbody>
</table>

**OPEN_ACCEPT = (0b) 24E 24E 24E**

K28.5  D16.7  D16.7  D16.7
Anatomy of an SSP Command Frame

<table>
<thead>
<tr>
<th>No.</th>
<th>Start Time</th>
<th>Port</th>
<th>Speed</th>
<th>Source Addr.</th>
<th>Marker</th>
<th>Destination Addr.</th>
<th>Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>5765</td>
<td>0:00:011_070_332 (min)</td>
<td>11</td>
<td>24 G</td>
<td>5000e5000000100</td>
<td></td>
<td>5000e500059c0235</td>
<td>0x00:00:00:00:00:00:00:00</td>
</tr>
<tr>
<td>5766</td>
<td>0:00:011_070_336 (min)</td>
<td>11</td>
<td>24 G</td>
<td>5000e5000000100</td>
<td></td>
<td>5000e500059c0235</td>
<td>0x00:00:00:00:00:00:00:00</td>
</tr>
<tr>
<td>5767</td>
<td>0:00:011_070_382 (min)</td>
<td>11</td>
<td>24 G</td>
<td>5000e5000000100</td>
<td></td>
<td>5000e500059c0235</td>
<td>0x00:00:00:00:00:00:00:00</td>
</tr>
</tbody>
</table>

**DWORD1**

<table>
<thead>
<tr>
<th>Index</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOF</td>
</tr>
<tr>
<td>2</td>
<td>DATA</td>
</tr>
<tr>
<td>3</td>
<td>DATA</td>
</tr>
<tr>
<td>4</td>
<td>DATA</td>
</tr>
<tr>
<td>5</td>
<td>DATA</td>
</tr>
<tr>
<td>6</td>
<td>B_EOF(2)</td>
</tr>
</tbody>
</table>

**DWORD2**

**DWORD3**

**DWORD4**

**Payload (128-bit)**

```
0x2A:Write (10)
```
Anatomy of an SSP Command Frame

<table>
<thead>
<tr>
<th>Index</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOF</td>
</tr>
<tr>
<td>2</td>
<td>DATA - DATA - DATA - DATA</td>
</tr>
<tr>
<td>3</td>
<td>DATA - DATA - DATA - DATA</td>
</tr>
<tr>
<td>4</td>
<td>DATA - DATA - DATA - DATA</td>
</tr>
<tr>
<td>5</td>
<td>DATA - CRC - Padding</td>
</tr>
<tr>
<td>6</td>
<td>B_EOF(2)</td>
</tr>
</tbody>
</table>

**Payload with FEC**

<table>
<thead>
<tr>
<th></th>
<th>11001</th>
<th>11011</th>
<th>00101</th>
<th>11000</th>
<th>10001</th>
<th>10001</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x2A</td>
<td>0x00</td>
<td>0x10</td>
<td>0x11</td>
<td>0x10</td>
<td>0x11</td>
<td>0x10</td>
</tr>
</tbody>
</table>

0x2A: Write (10)
Anatomy of an SSP Command Frame

<table>
<thead>
<tr>
<th>Index</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOF</td>
</tr>
<tr>
<td>2</td>
<td>DATA - DATA - DATA - DATA</td>
</tr>
<tr>
<td>3</td>
<td>DATA - DATA - DATA - DATA</td>
</tr>
<tr>
<td>4</td>
<td>DATA - DATA - DATA - DATA</td>
</tr>
<tr>
<td>5</td>
<td>DATA - CRC - Padding</td>
</tr>
<tr>
<td>6</td>
<td>B_EOF(2) - --- - --- - --- - ---</td>
</tr>
</tbody>
</table>

Frame Inspector View:

- **Length:** 64 bytes
- **Payload:** 128 bits

Payload with FEC:

- DWORD1: 06BA3499 (06)
- DWORD2: 00BF68EE (11)
- DWORD3: 00000000 (1C)
- DWORD4: 00000000 (0F)

- FEC (5-bit symbol): 0x2A:Write (10)
24G SAS Link Bring-up Sequence
24G SAS Speed Negotiation

SNW-1 through TX_Train / RX_Train

500 µs  109 µs
24G SAS Speed Negotiation

SNW-1 through TX_Train / RX_Train
## SAS PHY Capability Exchange

### Tables

<table>
<thead>
<tr>
<th>No.</th>
<th>Start Time</th>
<th>End Time</th>
<th>Source SAS Address</th>
<th>Target SAS Address</th>
<th>Frame Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>72</td>
<td>0x08,080,944,520 (min)</td>
<td>0x10,080,944,520 (max)</td>
<td>0x08,080,944,520 (min)</td>
<td>0x10,080,944,520 (max)</td>
<td>ALIN 10011</td>
</tr>
<tr>
<td>73</td>
<td>0x08,080,370,007 (min)</td>
<td>0x10,080,370,007 (max)</td>
<td>0x08,080,370,007 (min)</td>
<td>0x10,080,370,007 (max)</td>
<td>ALIN 10011</td>
</tr>
<tr>
<td>74</td>
<td>0x08,080,390,007 (min)</td>
<td>0x10,080,390,007 (max)</td>
<td>0x08,080,390,007 (min)</td>
<td>0x10,080,390,007 (max)</td>
<td>ALIN 10011</td>
</tr>
<tr>
<td>75</td>
<td>0x08,080,390,007 (min)</td>
<td>0x10,080,390,007 (max)</td>
<td>0x08,080,390,007 (min)</td>
<td>0x10,080,390,007 (max)</td>
<td>ALIN 10011</td>
</tr>
<tr>
<td>76</td>
<td>0x08,080,390,007 (min)</td>
<td>0x10,080,390,007 (max)</td>
<td>0x08,080,390,007 (min)</td>
<td>0x10,080,390,007 (max)</td>
<td>ALIN 10011</td>
</tr>
<tr>
<td>77</td>
<td>0x08,080,390,007 (min)</td>
<td>0x10,080,390,007 (max)</td>
<td>0x08,080,390,007 (min)</td>
<td>0x10,080,390,007 (max)</td>
<td>ALIN 10011</td>
</tr>
<tr>
<td>78</td>
<td>0x08,080,390,007 (min)</td>
<td>0x10,080,390,007 (max)</td>
<td>0x08,080,390,007 (min)</td>
<td>0x10,080,390,007 (max)</td>
<td>ALIN 10011</td>
</tr>
<tr>
<td>79</td>
<td>0x08,080,390,007 (min)</td>
<td>0x10,080,390,007 (max)</td>
<td>0x08,080,390,007 (min)</td>
<td>0x10,080,390,007 (max)</td>
<td>ALIN 10011</td>
</tr>
</tbody>
</table>

### Diagram

- **I_G5 Without SSC**
- **I_G5 With SSC**

---

March 18, 2020
TX Training Pattern in “SPL Packet Mode”

1. Pattern marker
2. TTIU
3. 59 SPL packet payloads (scrambled idle)
4. One END_TRAIN primitive
Transmitter Training Information Unit

- No Change to TTIU message format

Table 78 – Control/Status TTIU

<table>
<thead>
<tr>
<th>Byte\Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PATTERN TYPE (000b)</td>
<td>Reserved</td>
<td>COEFFICIENT SETTINGS</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coefficient Request byte</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td>COEFFICIENT 3 REQUEST</td>
<td>COEFFICIENT 2 REQUEST</td>
<td>COEFFICIENT 1 REQUEST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training Status word</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRAIN COMP</td>
<td>TX INIT</td>
<td>BALANCE</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td>COEFFICIENT 3 STATUS</td>
<td>COEFFICIENT 2 STATUS</td>
<td>COEFFICIENT 1 STATUS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 00b – Hold
- 10b – Decrement
- 01b – Increment
- 00b – Ready
- 01b – Update Comp.
- 10b – Min
- 11b - Max
TX Training Pattern in “SPL Packet Mode”

1. Pattern marker
2. TTIU
3. 59 SPL packet payloads (scrambled idle) and
4. one END_TRAIN primitive
Train_TX SNW in “SPL Packet Mode”

- Train_Tx Pattern Repeats
- Both sides assert “TRAIN COMPLETE” when done
Train_Rx-SNW in “SPL Packet Mode”

Receiver trained and SPL packet synchronization acquired

Time (not to scale)

PACKET_SYNC received

Rate change delay time (RCDT)

Training lock time (TLT)

Train_Rx-SNW time

Maximum receiver training time (MRTT)

Maximum Train_Rx-SNW time (MTWT)

Key:

Long time of negotiation idle

SPL packet containing a PACKET_SYNC_LOST extended binary primitive

SPL packet from link layer

SPL packet containing a PACKET_SYNC extended binary primitive

12G Initiator

12G Target

12G

TRAIN(x6)

XXXX(x58)

XXXX(x58)

TRAIN(x6)

XXXX(x58)

24G Initiator

24G Target

24G

PACKET_SYNC_LOST(x63)

PACKET_SYNC(x63)

PACKET_SYNC_LOST(x57)

PACKET_SYNC(x63)

PACKET_SYNC(x63)
Train_Rx-SNW in “SPL Packet Mode”

1. PACKET_SYNC_LOST
2. PACKET_SYNC
3. ALIGN @ 24G

Key:
- SPL packet containing a PACKET_SYNC_LOST
- SPL packet from link layer
- SPL packet containing a PACKET_SYNC
- Long time of negotiation idle
- Rate change delay time (RCDT)
- Training lock time (TLT)
- Train_Rx-SNW time
- Maximum receiver training time (MRTT)
- Maximum Train_Rx-SNW time (MTWT)
FEC, APTA, and Persistent Connections
Forward Error Correction (FEC)

- Corrects up to two Bit-Errors (per SPL Segment)
- Allows 1e-15 BER
- 20-Bit FEC (Reed-Solomon)
- Four 5-bit patterns interleaved

Example:
130 bit SPL Header & Payload

100010111000110100100010111000110100101110001101001011100011010010001011100011011000110101000101010101110011011

001011100011010010111000110100100010111000110110001101001011100011010010001011100011011000110101000101010101110011011
Forward Error Correction (FEC)

- Corrects up to two Bit-Errors (per SPL Segment)
- Enables 30dB loss budget
- 20-Bit FEC (Reed-Solomon)
- Four 5-bit patterns interleaved

Example:
150 bit Segment transmitted over link

100010111000110100100010111000(11001)11010010111000110(10110)00100010
1010001(10000)0111000110100101110001101(10110)00100010
11100011011000110(11000)0101000101010101011100110111

SPL packet header and payload
Two-bit Header 16 payload bytes

realignment

26 five-bit message symbols

Reed Solomon code encoding function

Transmitted codeword polynomial:
$T(x) = P(x) + (x^3 \times M(x))$

M(x)

Four five-bit parity symbols P(x)

reordering

Symbols transmitted on physical link
30 five-bit symbols
FEC Test Considerations

- Testing PHY recovery from FEC errors

Sierra M244 Analyzer / Jammer
FEC Test Considerations

◆ How will Analyzer show Correctable FEC Errors?
  ● Corrected FEC Errors Not Shown (corrected in real-time by analyzer)

◆ How will Analyzer show Uncorrectable FEC Errors?
  ● If parity check fails; send a “Decode Failure” message to the PHY
  ● Discard the SPL segment
  ◇ Analyzer shows “Unknown Packet” (FEC, CRC, and/or Frame Error)
APTA: Active PHY Transmitter Adjustment

- Inline Transmitter Training without Resetting Link
  - What Mechanism will be used to initiate APTA mode?
    - SAS-4 PHYs detect bit-errors during normal operation
    - Requests Management Application Layer start APTA
  - SAS-4 PHYs are *required* to support APTA
    - Exceptions: Optical & Active Cable links
APTA Example

APTA_ADJUST (START) → APTA_ADJUST (READY)
APTA_COEFFICIENT_1 (INCREMENT) → APTA_COEFFICIENT_1 (UPDATED)
APTA_COEFFICIENT_2 (INCREMENT) → APTA_COEFFICIENT_2 (UPDATED)
APTA_COEFFICIENT_3 (INCREMENT) → APTA_COEFFICIENT_3 (UPDATED)
APTA_ADJUST (COMPLETE)

1ms
APTA: When is it used?

- When can SAS-4 PHYs initiate APTA?
  - PHY in “Ready” state
  - No Active Connection

- How does PHY reject an APTA Request?
  - PHY Sets Flag Disabling APTA in Management Application Layer:
    - Active Connection
    - OOB in progress
    - Low PHY Power Condition
APTA: How to Test it?

**Use Exerciser:** to verify Valid *APTA_Start*
- Send APTA _ADJ_(Start) and Verify DUT Reports “Ready” & “Updated”

```
APTA_ADJUST(START);
APTA_ADJUST(REady);
APTA_COEFFICIENT_3(INC);
APTA_COEFFICIENT_3(UPDATED);
```

**Use Jammer:** to verify Invalid *APTA_Start*
- Replace OPEN_ACCEPT with APTA _ADJ_(START)

```
1:Open Address Frame
OPEN_ACCEPT;
APTA_ADJUST(START);
```

March 18, 2020
Persistent Connections

- Designed to reduce Overhead / Latency
- Intended for Direct-Attached (Server) Storage
- Sets SSP Persistent Capable Bit = “1” in IDENTIFY
- Connections stay open
  - While receiving/sending SSP frame

Transmit Extend Connection timer: 100 µs
Persistent Connections

- Designed to reduce Overhead / Latency
- Intended for Direct-Attached (Server) Storage
- Sets SSP Persistent Capable Bit = “1” in IDENTIFY
- Connections stay open
  - or sending EXTEND CONNECTION (NORMAL)

Transmit Extend Connection timer: \(100 \mu s\)
Persistent Connection Timeout timer: \(1 \text{ ms}\)
Persistent Connections: Jammer Test Cases

- “Persistent Connection Timeout Timer” (1 ms) Test

  - Wait for 100 EXT_CONNECTION primitives
  - Drop 10 EXT_CONNECTION primitives
  - Verify Initiator Sends DONE, CLOSE

- “Credit Starved” Condition Test

  - Wait for 10,000 EXT_CONNECTION primitives
  - Wait for READ-10
  - Drop Every other R-RDY primitives
  - Verify Target Sends DONE (CREDIT TIMEOUT)
Sierra T244 & M244 24G SAS Verification Platforms

Sierra T244 Analyzer

- **Unmatched Accuracy**
  - T.A.P.4™ probe: minimal impact on dynamic equalization
- **Flexible Analysis software**
- **Efficient Debug**
  - Hardware based triggering

Sierra M244 Analyzer / Jammer / Exerciser

- **Complete Solution**
  - Jammer & Exerciser options for end-to-end Verification
- **Easy drag-and-drop Test development**
- **Efficient Debug**
  - Hardware based triggering
  - Python-based Automation framework
Ecosystem is on track for SAS-4 production readiness in 2020

- SAS-4 analyzers have been sampling since last year

- Cables and connectors: both existing and new form-factors ready for 24G SAS

- SAS-4 controllers and expanders aligned with upcoming PCIe Gen4 platform launches

- New HDD/SSD capabilities to intersect with 24G SAS ecosystem
  - MultiLink SSDs
  - Multiple Actuator
  - Hybrid SMR
  - HAMR / MAMR
Questions?
Thank You!

For more information, go to:

www.scsita.org/content
https://teledynelecroy.com/

Also follow STA on social media: