P1149.10

High Speed Test Access Port and Distribution Architecture

CJ Clark, Intellitech

Scope

This standard defines a high speed test access port for delivery of test data, a packet format for describing the test payload and a distribution architecture for converting the test data to/from on-chip test structures.

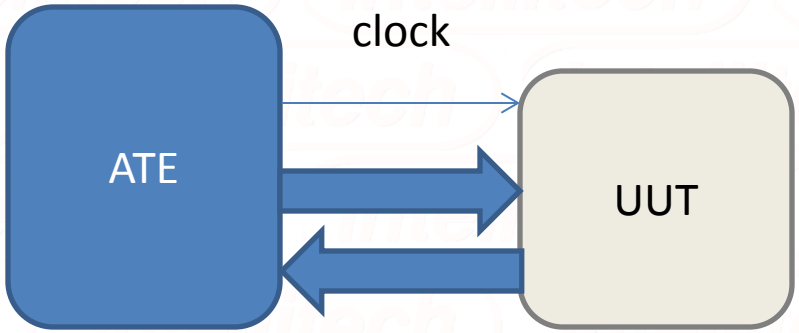
The standard re-uses existing High Speed I/O (HSIO) known in the industry for the High-Speed Test Access Port. The HSIO connects to an on-chip distribution architecture through a common interface. The scope includes the distribution architecture test logic and packet decoder logic. The objective of the distribution architecture and packet decoder is that it can be readily re-used with different Integrated Circuits (ICs) that host different HSIO technology such that the standard addresses as large a part of the industry as possible.

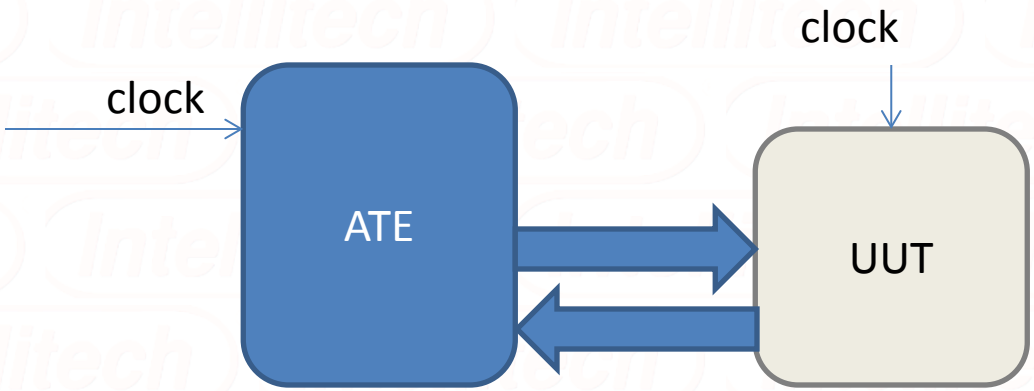
The scope includes IEEE 1149.1 Boundary Scan Description Language (BSDL) and Procedural Description Language (PDL) documentation which can be used for configuring a mission mode HSIO to a test mode compatible with the High Speed Test Access Port (HSTAP). The same BSDL and PDL can then be used to deliver high-speed data to the on-chip test structures.

Disclaimer:

This is an organizing/teaching presentation to communicate at least one technical approach to realizing the standard. Nothing in the presentation has been voted on by the working group. Alternative technical solution presentations are encouraged.

Common Clock - Interface is synchronous to clock driven by ATE





Embedded Clock - Clock is embedded in the interface

- While ATE may drive the system clock needed by UUT, Clock for UUT could come from other source.

P1149.10

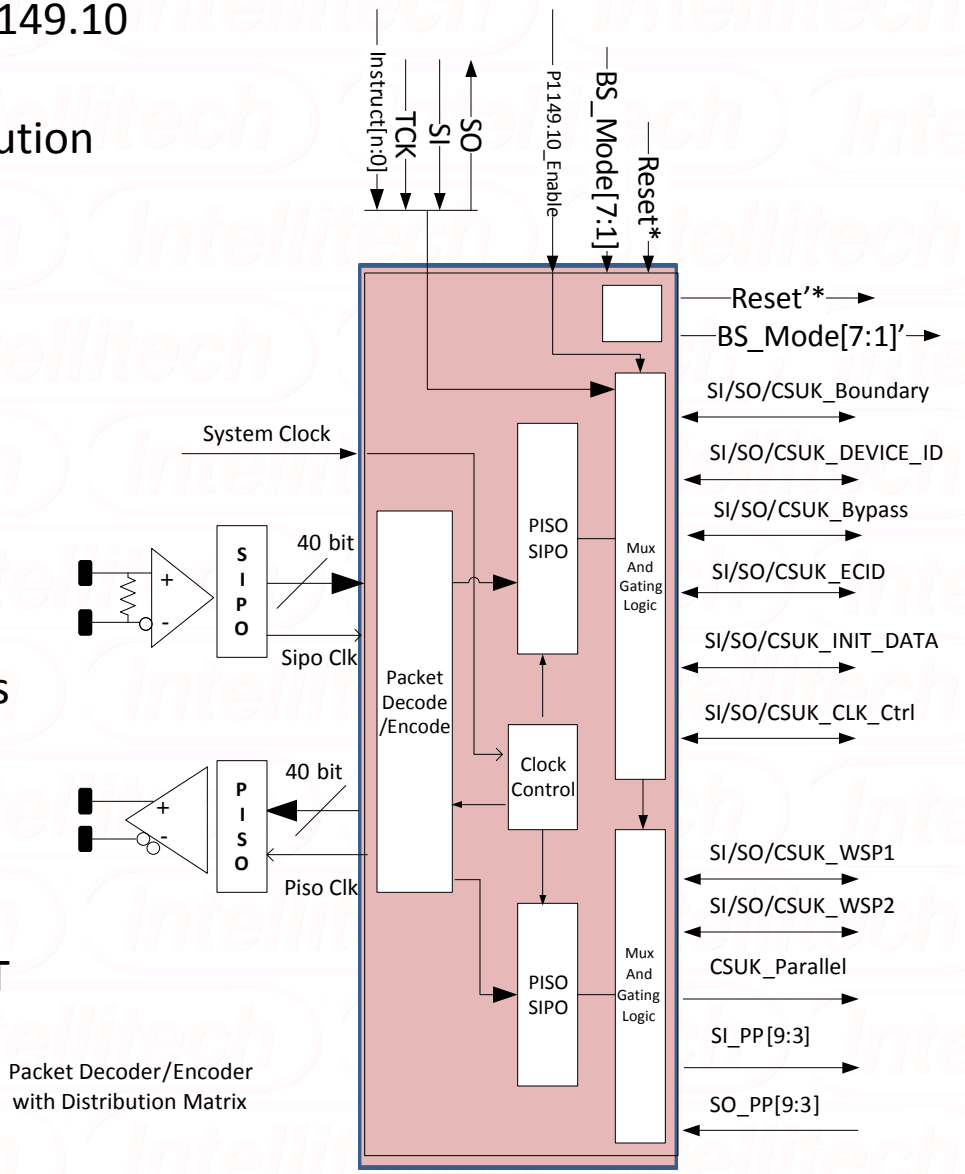
Define packet encoder/decoder & distribution Matrix (shaded area)

Define packets for packet decoder

Enable packet formats to validate link during testing (CRC32, running disparity, etc)

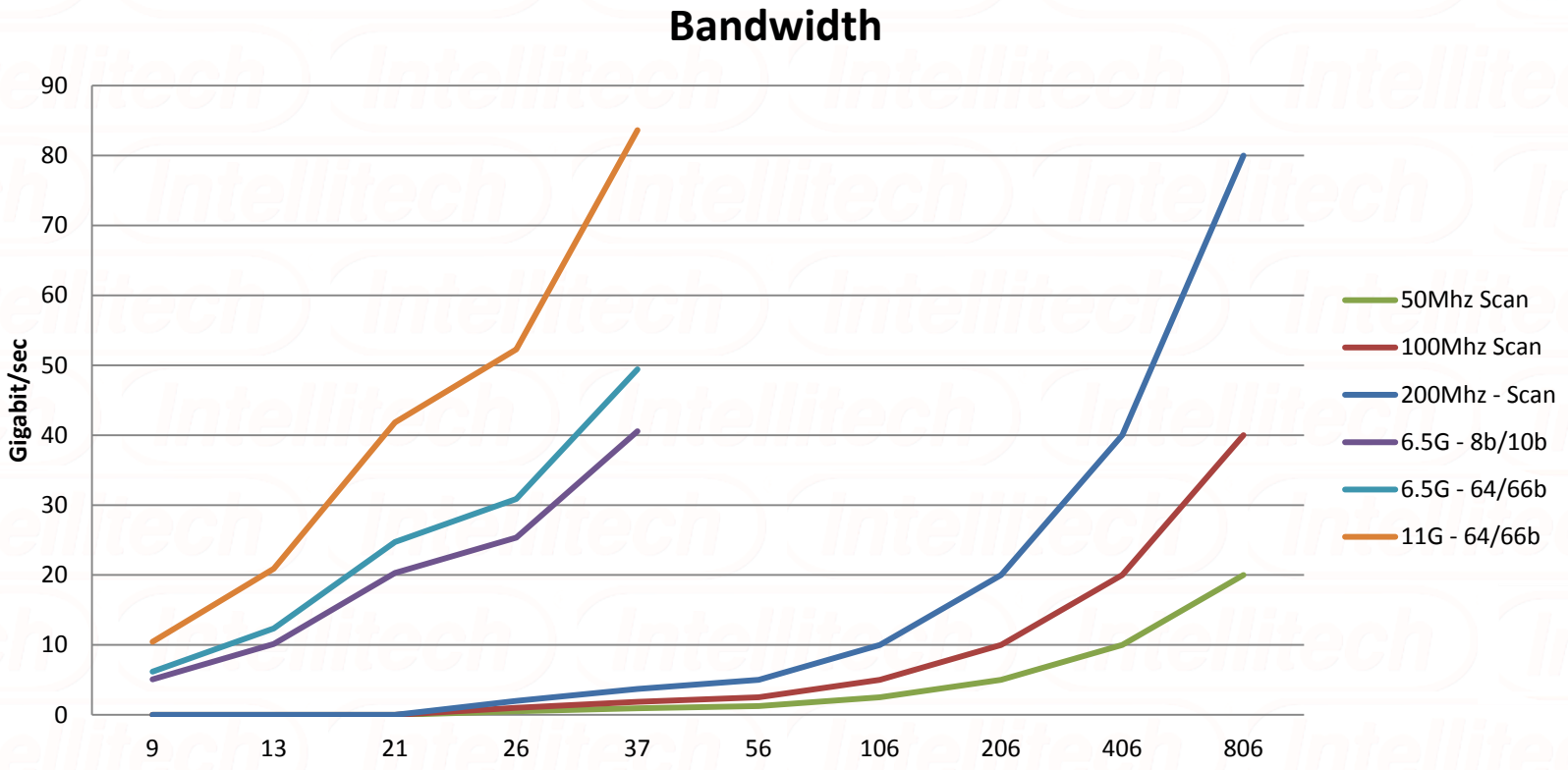
Define vendor documentation for Enabling P1149.10 interface, enabling Loopback and documenting requirements of interface (system clock, diff swing, Encoding (8b/10b, 64/66b, 128/130b etc)

Testing or implementation of SERDES BIST Is not an objective of the standard.



Packet Decoder/Encoder with Distribution Matrix

Common Clock = 80G/sec = 400 chains, 806 pins and 200Mhz operation
Embedded Clock SERDES = 80G/sec = User Defined chains, 37 pins and user defined Mhz



SI/SO+TAP+CLK		9	13	21	20	37	50	100	200	400	800
Chains		X	X	X	10	X	25	50	100	200	400
Tester Chan		9	13	21	26	37	56	106	206	406	806

SERDES = 4 pins + 4 pin TAP + clock. Channel bonding is used to have 2,4 and 8 Serdes lanes.
5 SERDES lanes just included for comparison purposes.
Bandwidth adjusted for encoding bit loss and 2% overhead of packet

Observations

80G bandwidth - 806 pins running 400 scan chains at 200Mhz

P1149.10 - More flexible with number of internal scan-chains and scan-chain architecture. Channel bonding can be used for higher bandwidth or multi-site testing Users can make their own economic choices

Embedded clock SERDES is scalable. More bandwidth coming

Common Clock - reaching ceiling - needs more pins and higher clock rates

Normalizing for ATE pins Embedded Clock is 5 -10x bandwidth than common clock

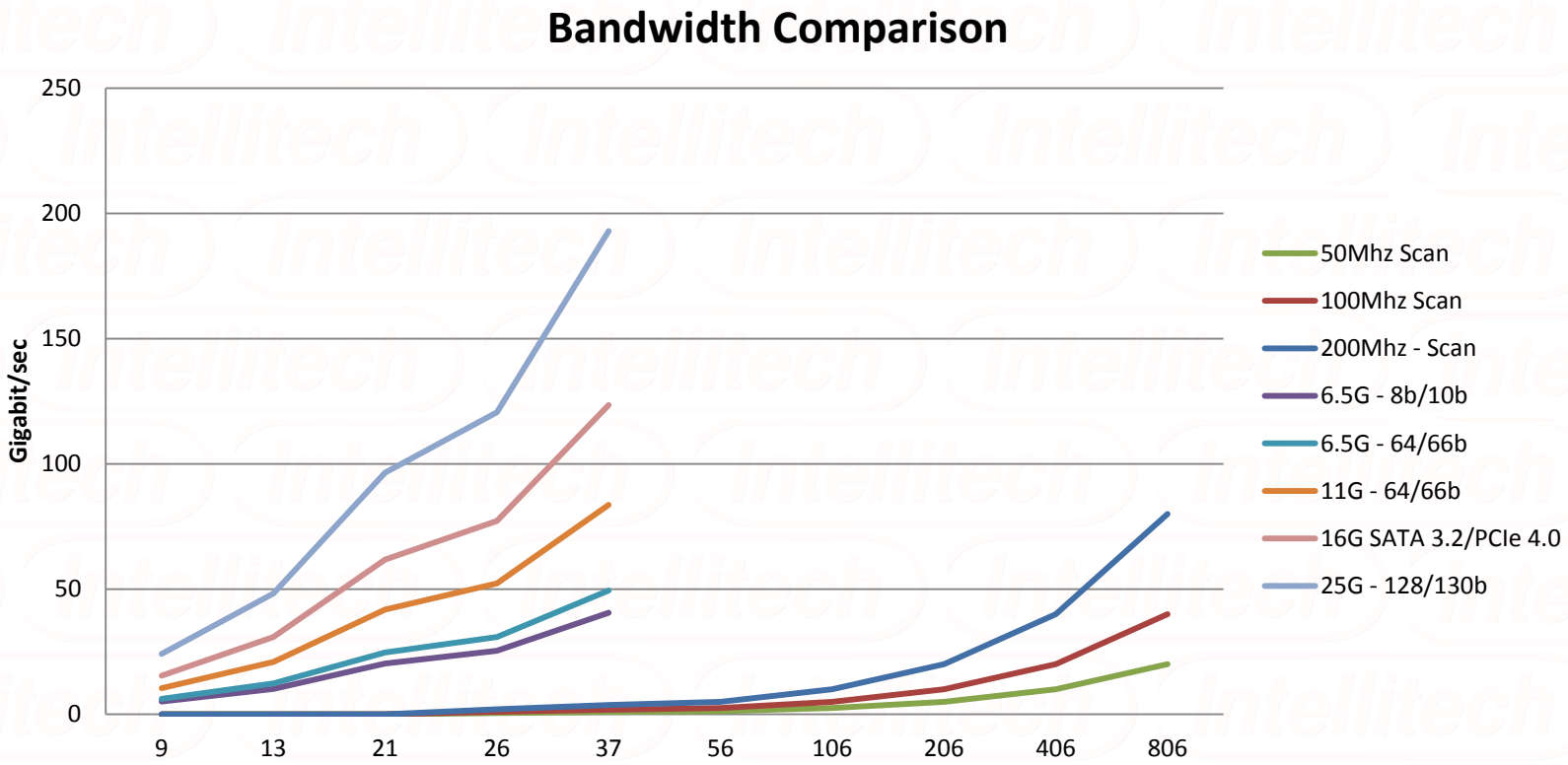
- Common Clock 100 pins = 10G @200mhz
  SERDES 10G bandwidth with 9 pins
  11 sites = 99pins = 110G total bandwidth

CEI-25G 2013 25G/sec 16G/sec SATA 3.2 now - PCIe 4.0 - coming

Credo, Snowbush, Avago all with 25G SERDES IP in 2013 - Others?
Some with 28G- Xilinx Virtex 7 HT (shipping) GTZ run to 28G/sec

2 CEI-25G SERDES will yield 50G bandwidth

Need 240 pins at 200mhz to equal one CEI-25G interface link



Cost Trade-offs

Probe Card:

P1149.10 requires Gigabit probe card ('increased' cost) and handler infrastructure

Compare with common clock requires increased cost probe card (800+ contacts for single location).

Handler

Requires gigabit connections but less of them. Could use (Coax however SATA, CAT-6A, Fiber are also low cost choices).

Silicon

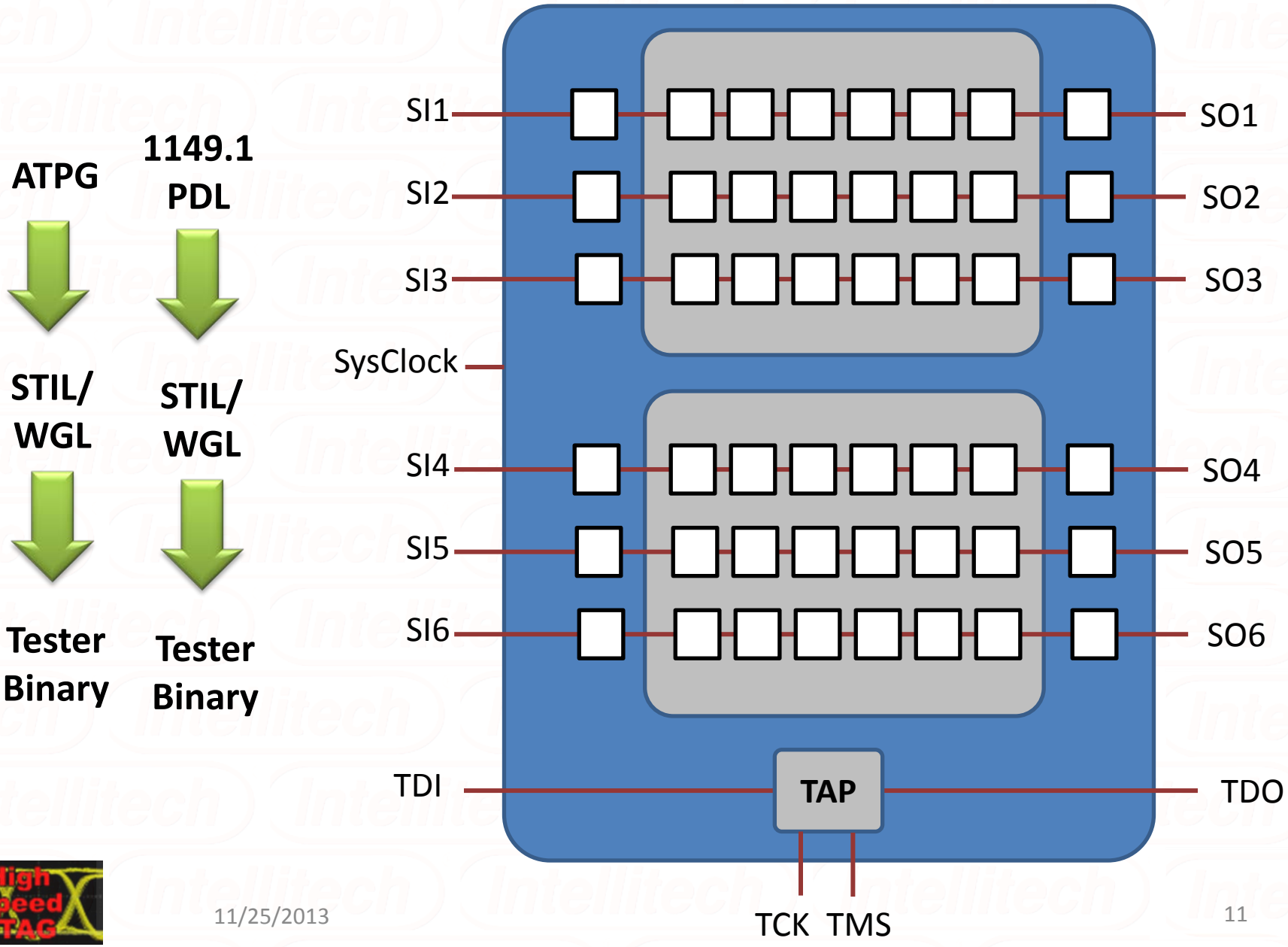
re-using mission mode SERDES so not expecting vendor to implement SERDES just for P1149.10. P1149.10 does not preclude dedicated SERDES either.

No use P1149.10 does not mandate it is used during wafer test. So one may use P1149.10 in later stage processes. Instrument access at benchtop, characterization and FPGA configuration.

The numbers

200	0	0	0	2	3.70	5	10	20	40	80
100	0	0	0	1	1.85	2.5	5	10	20	40
50	0	0	0	0.5	0.93	1.25	2.5	5	10	20
6.5G w/ 8b/10b	5.07	10.14	20.28	25.35	40.56					
6.5G w/ 64/66b	6.18	12.35	24.71	30.88	49.42					
11G w/64/66b	10.45	20.91	41.81	52.27	83.63					
16G SATA 3.2/PCIe 4.0	15.44	30.88	61.76	77.19	123.51					
25G w/ 128/130b	24.12	48.25	96.49	120.62	192.98					
SI/SO+TAP+CLK	9	13	21	20	37	50	100	200	400	800
Chains	X	X	X	10	X	25	50	100	200	400
Tester Chan	9	13	21	26	37	56	106	206	406	806

Common Scan Clock - Simplified block diagram



Embedded Clock - block diagram - Scan virtualized

ATPG

STIL/WGL

P1149.10 Packets

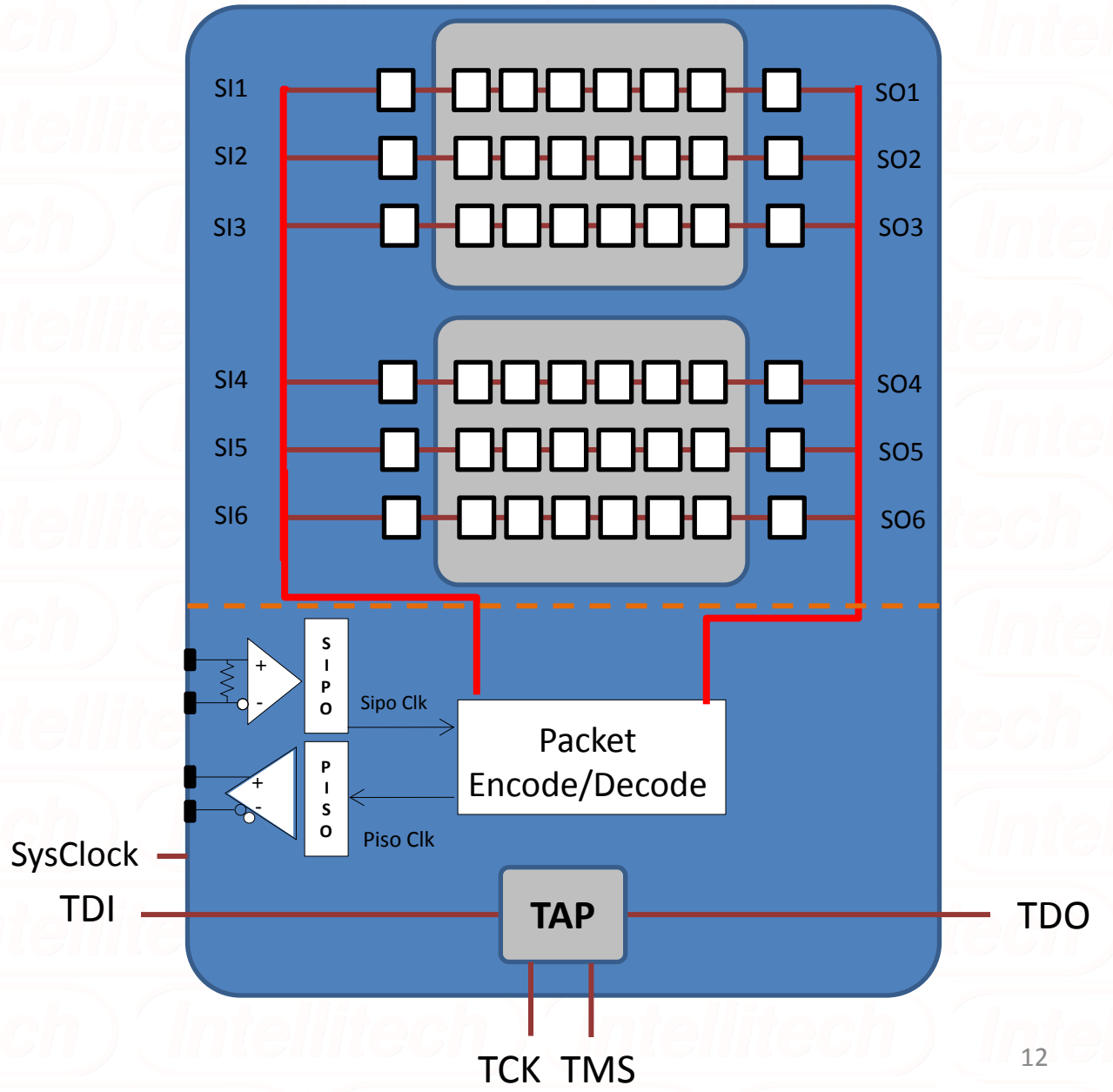
PHY Encoding (8b/10b, etc)

Tester Binary

1149.1 PDL

STIL/ WGL

Tester Binary



BSDL Attributes & PDL

Need to communicate: Sysclk(s) frequency, DiffSwing, encoding
How to get SERDES into P1149.10 Mode
Anything needed shall be communicated

In 1149.1-2013

Attribute SYSCLOCK_REQUIREMENTS of MyChip : entity IS
"(SysClk, 198.5e6, 201.5e6, 1149_10_Enable) ";

Pin, Min F, Max F, Instruction, Instruction

Possible Addition

Attribute PHY_1149_10 of MyChip : entity IS
"(SATA_TXP, SATA_RXP, 500E-3, 800E-3, 8B10B) ";

TX Rep Port, RX Rep Port, Min V, MaxV , Encoding

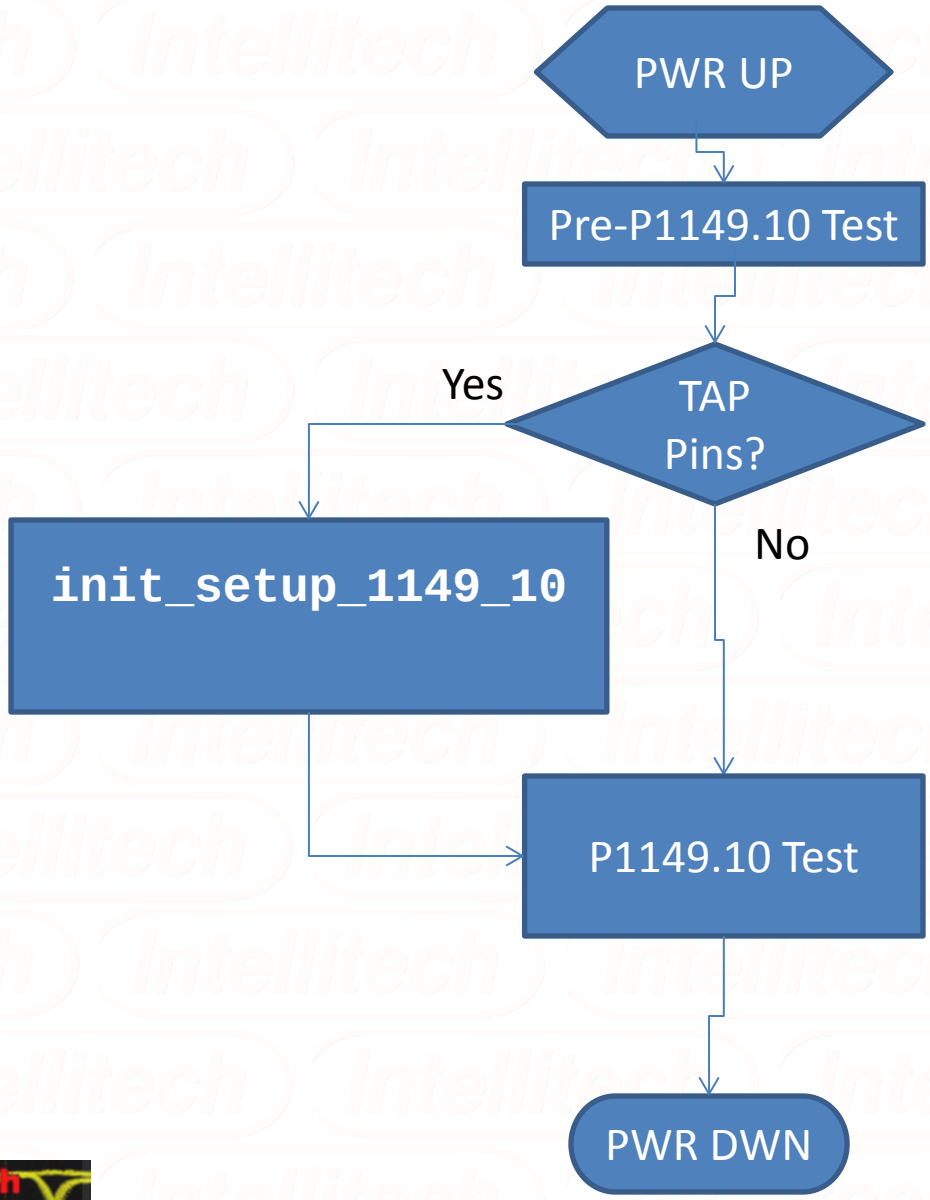
(Needs to support multiple pin pairs)
Need to enumerate encodings understood by this standard
(8b10b, 64/66b, 128/130b, others?)

Power descriptions also possible in 1149.1-2013

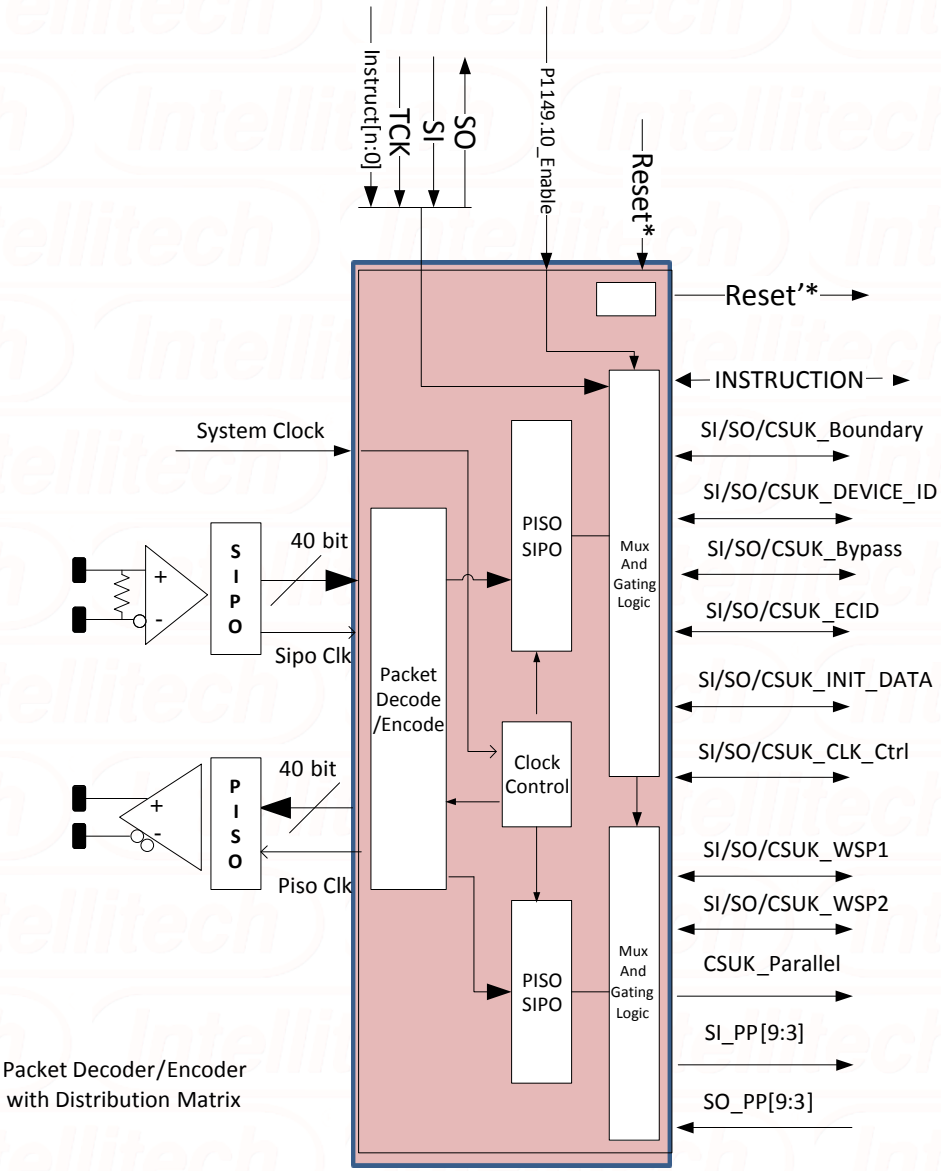
```
# MyCorp_SERDES.pdl
iPDLLevel 0 -version STD_1149_1_2013
iProcGroup MyCorp_SERDES

iProc init_setup_1149_10 {} {

  iWrite PLL     125Mhz  ;# @40bits = 5G
  iWrite mode    1149_10
  iWrite encoding 8b10b
  iWrite TX_Swing 800mv
  iWrite Power    ON;# disable pwrdn
  iApply
}
```

See 1149.1 PDL Tutorial for information of how this gets Translated to ATE patterns

Flow


PWR UP
Pre-P1149.10 Test
TAP Pins?
Yes
No
init_setup_1149_10
P1149.10 Test
PWR DWN

Alternate approach with Instruction Register accessible



Packet Decoder/Encoder with Distribution Matrix

Potential In-bound Packets

Packet descriptions shown without bit level encoding

CONFIG- Enable uninitialized P1149.10 interface to be enumerated to 'n'

TARGET <n> - specify where packets go

RESET - Assert reset* or TRST* (internally different signals)

RAW - Enable Interface in a RAW data mode (suitable for BER testing)
Data is not processed by packet processor subsequently and all RX data is sent to TX

XOFFA - XOFF Acknowledge (SATA uses HOLD/HOLDA)

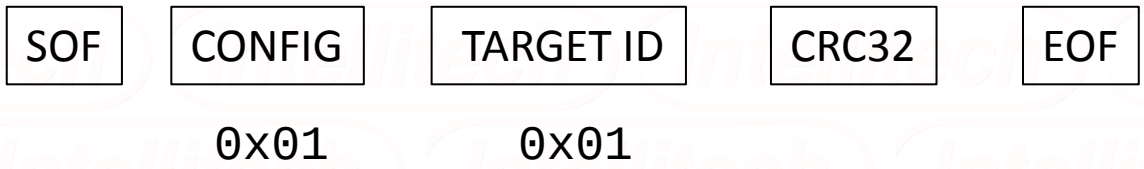
SCAN - Interleaved IR/DR Scan Packet

All "R" response packets are ignored and forwarded to TX

CONFIG

CONFIG - Enable uninitialized P1149.10 interface to be enumerated to 'n'

Device powers up with TARGET ID of 0000. All packets received when ID Is 0000 are processed. Once TARGET ID is set, only packets following a TARGET packet with TARGET ID of same will be processed.



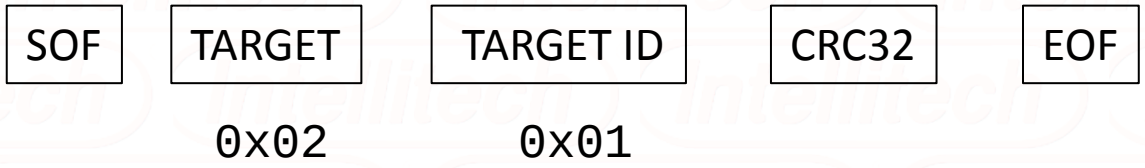
SOF = Start of Frame
EOF = End of Frame

TARGET <n>

TARGET - Sets device to listen for subsequent packets

If TARGET ID matches TARGET ID of device, it
a) sends TARGETR response packet
b) accepts incoming packets and sends "R" responses on TX until next TARGET ID

If TARGET ID does not match TARGET ID of device
a) device forwards all packets received to TX, examining each for TARGET ID packet



SOF = Start of Frame
EOF = End of Frame

RESET - Issue reset*

RESET - Issues test logic reset equivalent to going to TLR in state machine

Assert RESET* - Device determines necessary length of time/TCK cycles needed. ATE should not care.

TMS	0x00	TRST*	0x01

SOF	RESET	TYPE	CRC32	EOF
	0x03	0x00		

SOF = Start of Frame
EOF = End of Frame

RAW - Put 1149.10 interface in raw data mode

RAW - enable BER testing of P1149.10 interface without packet decode
- Requires 1149.1 access to reset or power-cycle

SOF = Start of Frame
EOF = End of Frame

XOFFA - Acknowledge a XOFF from UUT

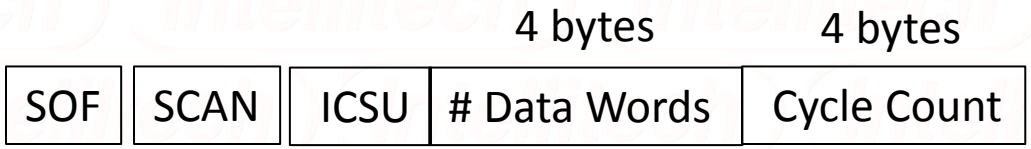
XOFFA - Acknowledge that UUT has asked to stop sending via XOFF

SOF = Start of Frame
EOF = End of Frame

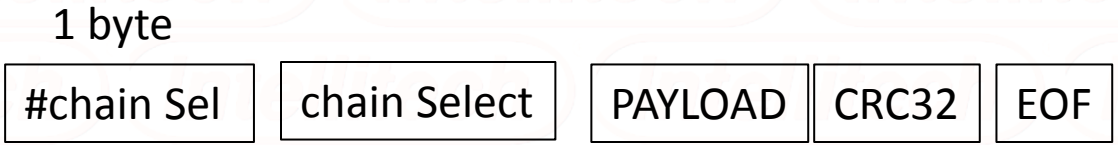
SCAN Packet Send data to IR or DR scan chain(s) as needed.

ICSU - IR Scan, Capture, Shift, Update

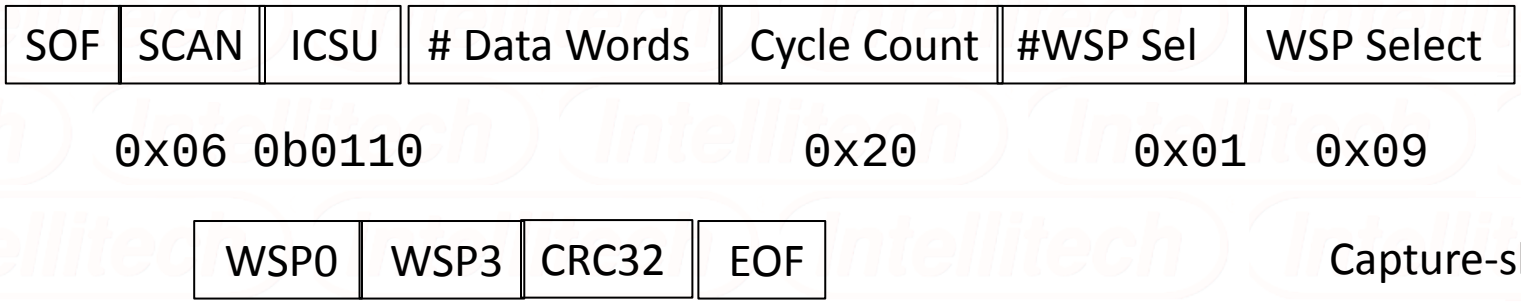
One-hot chain select is large for large # scan chains
Reserve 0 for IR Scan? 400 scan chains = 50 bytes of chain select



0x07



Interleaved SCAN Packet Format
- dealing with WSPs of different lengths
- WSP0 = 32bits WSP3 = 33bits



Potential Out-bound Packets

TARGETR - Target Packet Response

CONFIGR - CONFIG Packet Response

RAWR - RAW Packet response

RESETR - Reset Packet response

XOFF - Tell ATE to hold off sending more packets
XON - Tell ATE to resume

SCANR - Interleaved Scan Packet Response

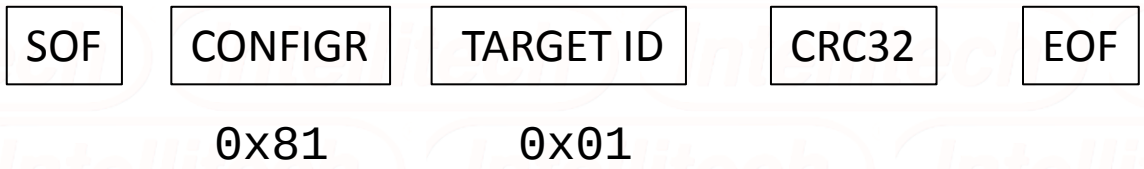
IDLE <n> - Tell ATE to insert N IDLE packets

All inbound packets following a TARGET for an alternative device

CONFIGR

CONFIGR - Response to CONFIG command

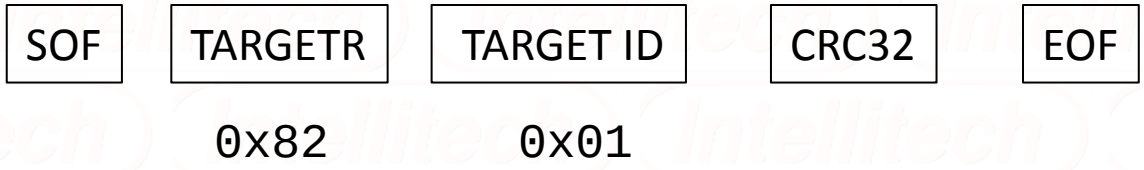
Device powers up with TARGET ID of 0000. All packets received when ID Is 0000 are processed. Once TARGET ID is set, only packets following a TARGET packet with TARGET ID of same will be processed.



SOF = Start of Frame
EOF = End of Frame

TARGETR

TARGETR - Response packet



SOF = Start of Frame
EOF = End of Frame

RESETR - Issue reset* Response

RESETR - Issues test logic reset equivalent to going to TLR in state machine

TMS	0x00	TRST*	0x01

SOF	RESETR	TYPE	CRC32	EOF
	0x83	0x00		

SOF = Start of Frame
EOF = End of Frame

RAWR - Put 1149.10 interface in raw data mode

RAWR - Respond to RAW packet

SOF = Start of Frame
EOF = End of Frame

XON/XOFF- Send to ATE XOFF/XON packet

XON/XOFF - Send to ATE XOFF/XON

SOF = Start of Frame
EOF = End of Frame

SCAN Packet Send data to IR or DR scan chain(s) as needed.

This may need to be condensed? If return scan data is not Same as in-scan data, what difficulty exists for packet encoder to Determine #datawords and #cycle count?