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.

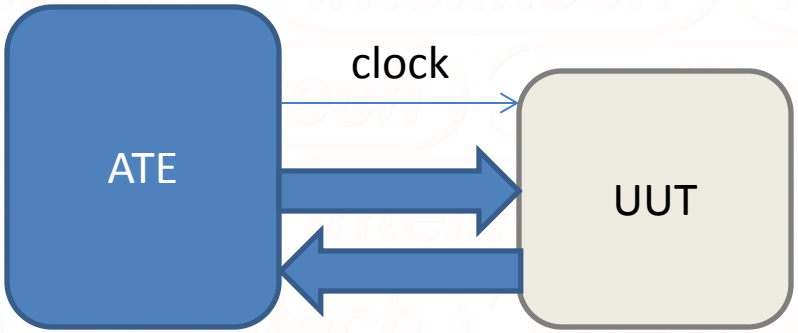
The standard re-uses existing High Speed I/O (HSIO) known in the industry for the High-Speed Test Access Port. The HSIO connects to an on-chip distribution architecture through a common interface. The scope includes the distribution architecture test logic and packet decoder logic. The objective of the 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.

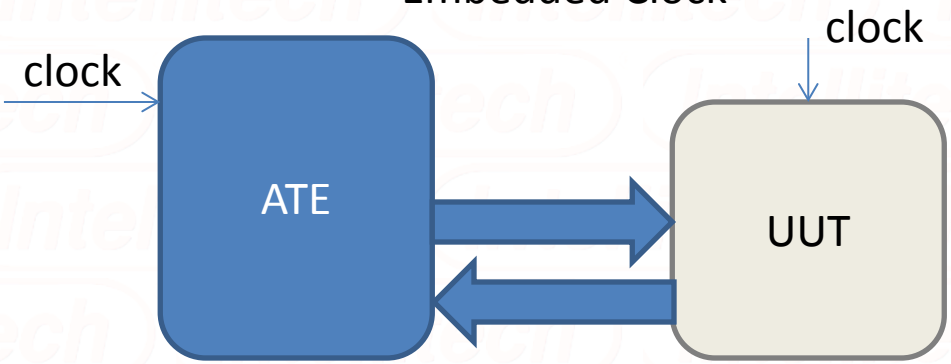
Disclaimer:

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



Embedded Clock



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.

Common Clock - Interface is synchronous to clock driven by ATE

P1149.10

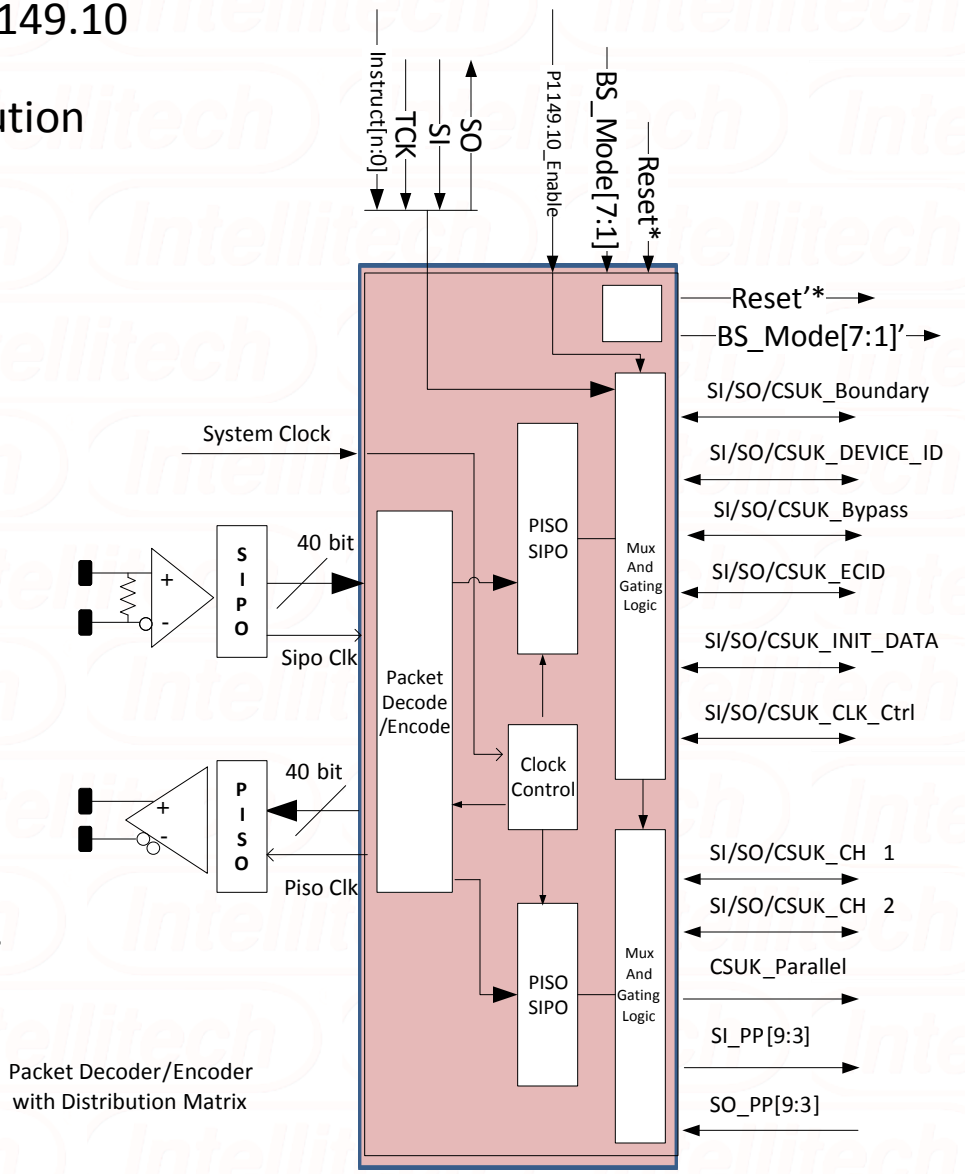
Define packet encoder/decoder & distribution Matrix (shaded area)

Define packets for packet decoder

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

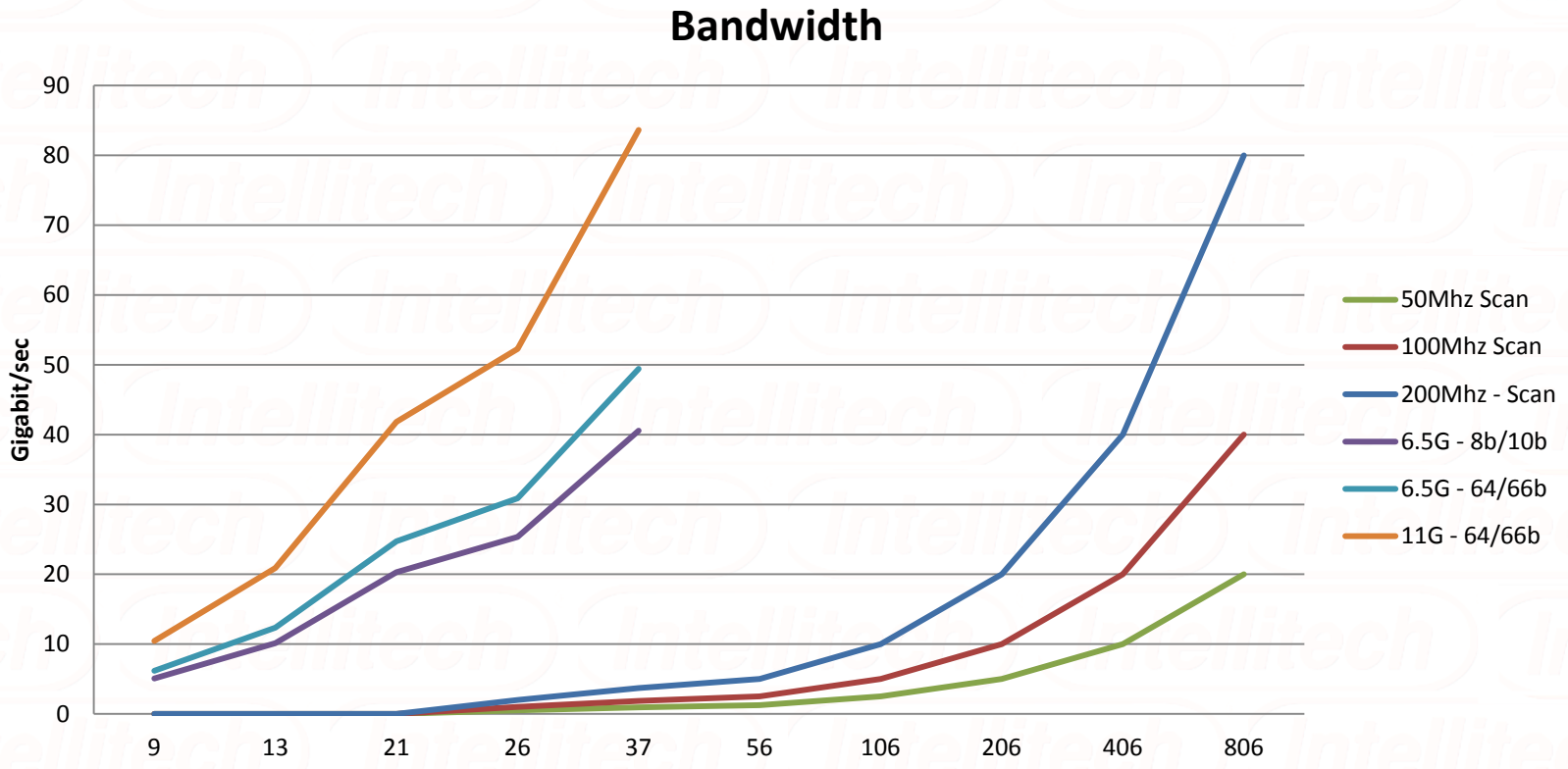
Define vendor documentation for Enabling P1149.10 interface, enabling Loopback and documenting requirements of interface (system clock, diff swing, Encoding (8b/10b, 64/66b, 128/130b etc)

Testing or implementation of SERDES BIST Is not an objective of the standard. Testing P1149.10 interface is not an objective



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

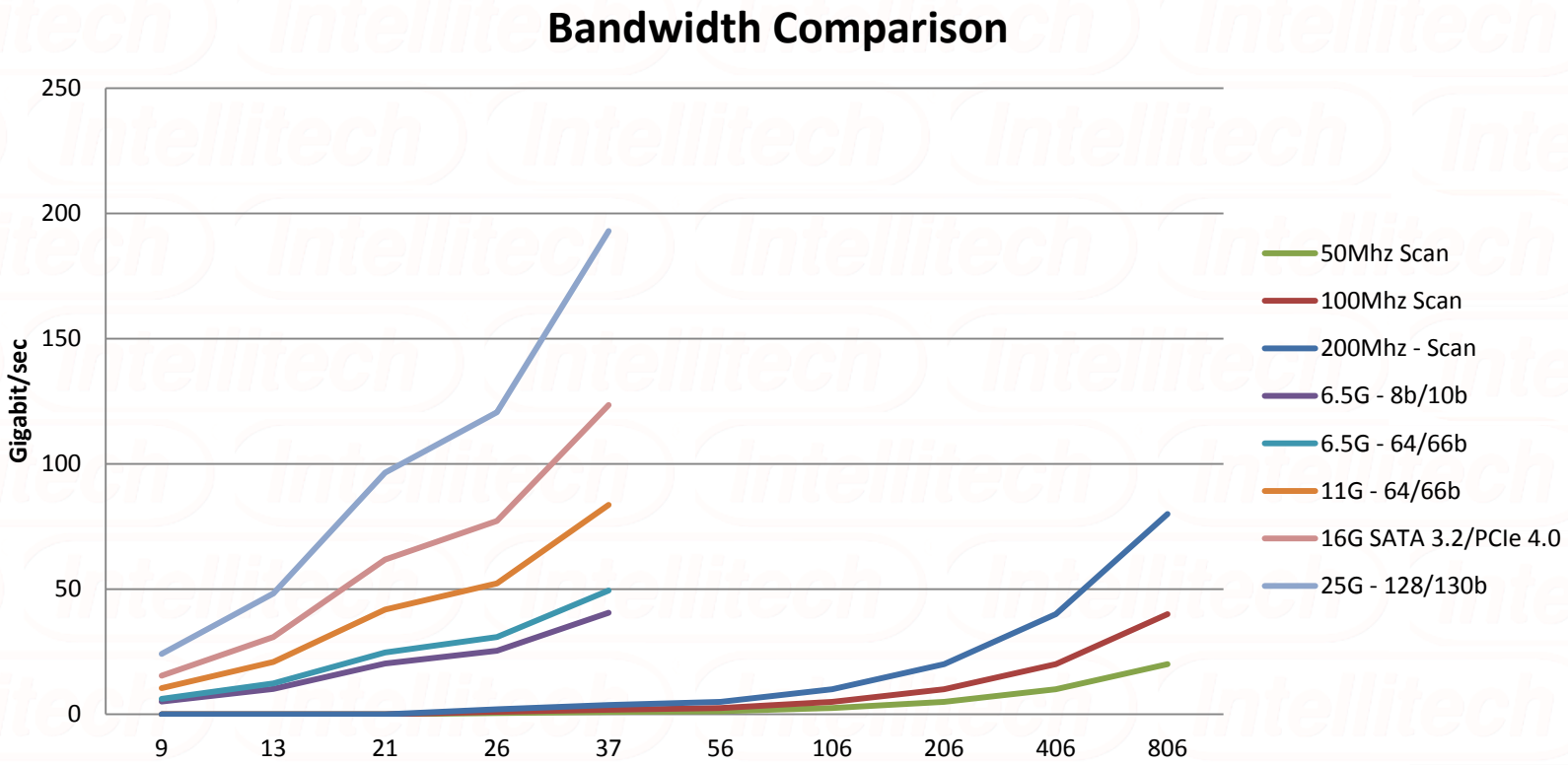
- Common Clock 100 pins = 10G @200mhz
  SERDES 10G bandwidth with 9 pins
  11 sites = 99pins = 110G total bandwidth

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

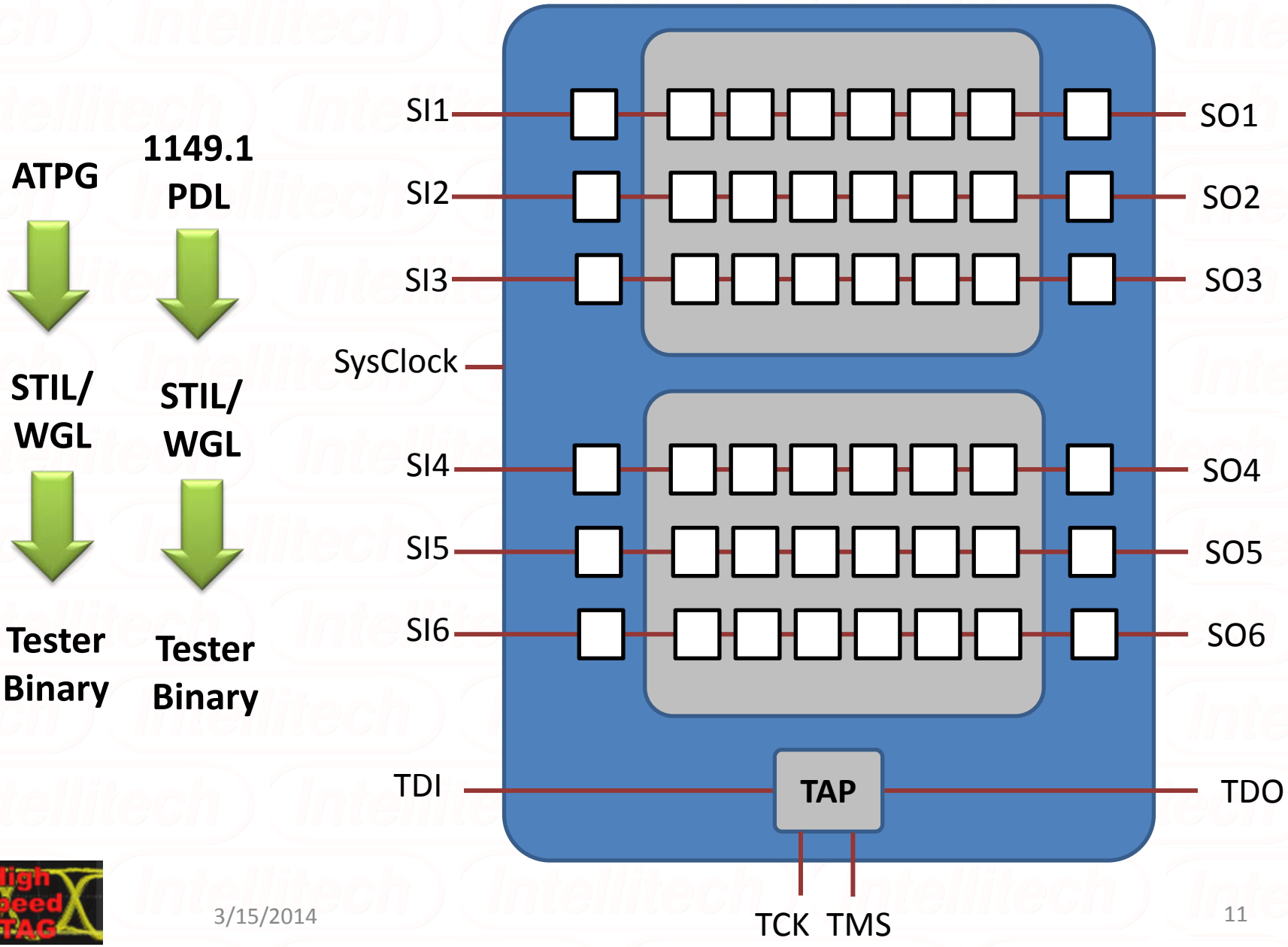
re-using mission mode SERDES so not expecting vendor to implement SERDES just for P1149.10. P1149.10 does not preclude dedicated SERDES either.

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

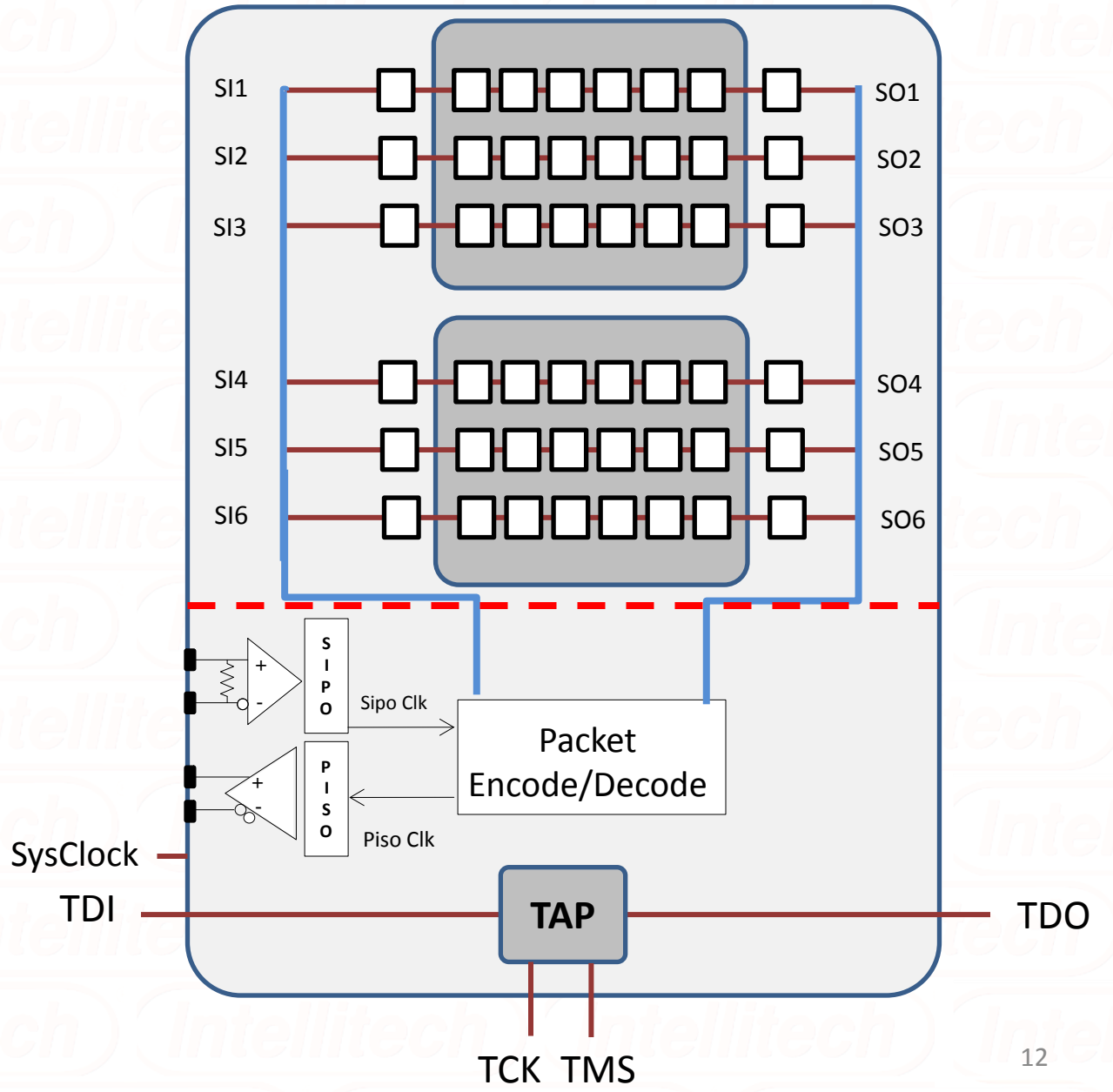
PHY Encoding (8b/10b, etc)

Tester Binary

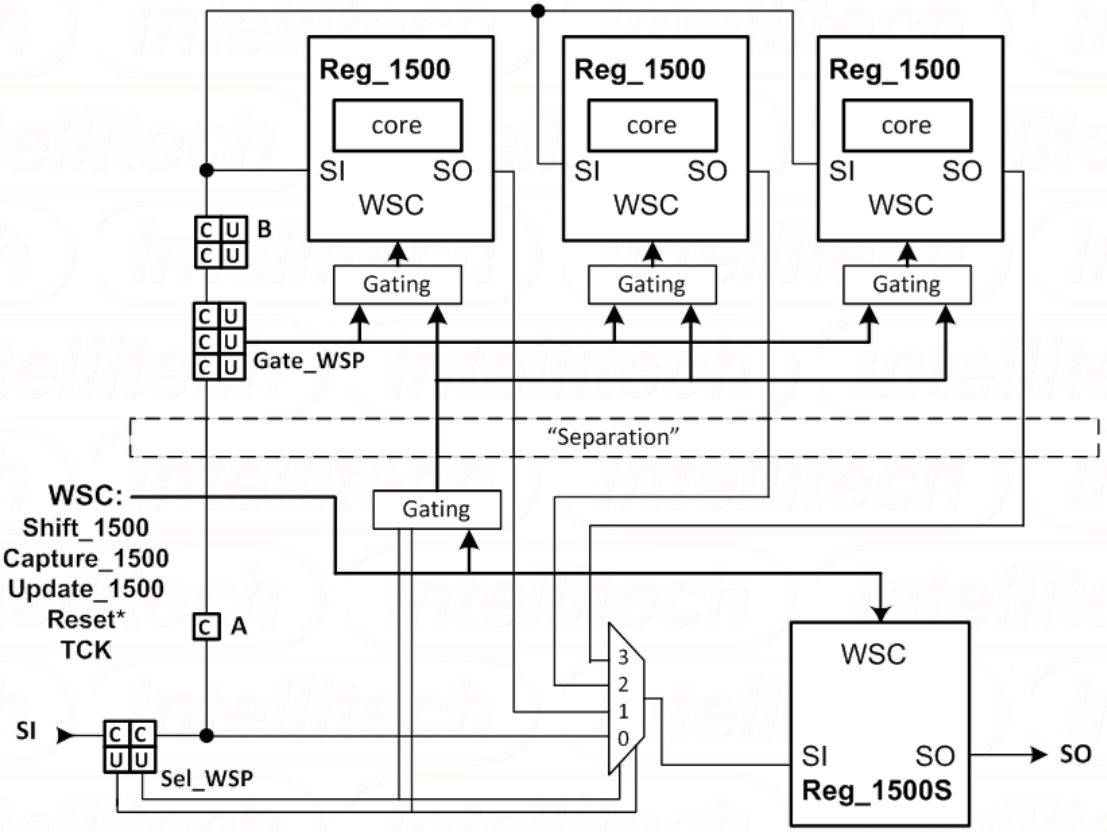
1149.1 PDL

STIL/ WGL

Tester Binary



Multi-core support is via scan architecture (See 1149.1 pg. 411)
Gate_CH selects gating of 3 Target Cores.
Sel_CH selects core to observe response



Note that in P1149.10 multiple SO can be returned to packet encoder

Multi-core support is via scan architecture (See 1149.1 pg. 411)

BSDL Description

```
package REG_1500_ASSM is
   use STD_1149_1_2012.all;
end REG_1500_ASSM ;

package body REG_1500_ASSM is
   use STD_1149_1_2012.all;
   use REG_1500.all;
   use REG_1500S.all;

attribute REGISTER_MNEMONICS of REG_1500_ASSM  : package is
   "CH  ( "&
      "   None   (0B00) <Bypass all CHs>, "&
      "   CH1   (0B01) <Observe CH(1)>, "&
      "   CH2   (0B10) <Observe CH(2)>, "&
      "   CH3   (0B11) <Observe CH(3)>  "&
     " )," &
   "BRDCST ( "&
      "   None      (0B000) <All CH held>, "&
      "   CH1      (0B001) <Scan CH(1) only>, "&
      "   CH2      (0B010) <Scan CH(2) only>, "&
      "   CH3      (0B011) <Scan CH(3) only>, "&
      "   1AND2     (0B110) <Scan just CH(1) and CH(2)>, "&
      "   ALLCH    (0B111) <Scan all CHs >  "&
    "  )";
```

```
Attribute REGISTER_ASSEMBLY of REG_1500_ASSM : package IS
  "Reg_1500_MUX ( " &
    "(Sel_CH[2]  ResetVal(CH(None))    TAPReset  ) ,"&
    "(SELECTMUX " &
      "(WIRE1 is WIRE)," &
      "(ARRAY CH(1 TO 3) IS CH_inst) " &
      "SELECTFIELD (Sel_CH) "&   -- 4:1 selection
      "SELECTVALUES ( "&
  --   Decode logic for connecting a CH to Scan-Out
       "(WIRE1:None) (CH(1):CH1) (CH(2):CH2) (CH(3):CH3) )"&
      "BROADCASTFIELD (Gate_CH) "&  -- Could use CH_common.Gate_CH
      "BROADCASTVALUES ( "&    --  Decode  logic for gating WSC
        "(CH(1),CH(2),CH(3) : ALLCH) "&
        "(CH(1),CH(2)        : 1AND2 ) "&
        "(CH(1)              : CH1) "&
        "(CH(2)              : CH2) "&
        "(CH(3)              : CH3) "&
                    ")"&
    "),"&
    "( CH_1500S is Reg_1500S)" &
-- Reg_1500S comes after MUX
                "),"  &   -- end REG_1500_MUX
  "WIRE ( ( WIRE[0] ) ), "&
  "CH_inst ( "&
      "(CH_common), "&
      "(CH_1500 IS Reg_1500) " &
  "), "&
  "common_seg ( (CH_common IS common) ), "&
  "common ("&
    "(A [1] NOUPD), "&
    "(Gate_CH[3] ResetVal(BRDCST(None)) TAPReset ), "&
    "(B [2] ) "&
  ") " ;


end REG_1500_ASSM;
```

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, Max V , Encoding

(Needs to support multiple pin pairs)
Need to enumerate encodings understood by this standard (8b10b, 64/66b, 128/130b, others?)
Packet interleave size - 1/4/8 bits? **12/2/2013**
Group definitions - define scan chain groups **12/2/2013**
SOF/EOF/IDLE/XON/XOFF/Reset characters

Power descriptions also possible in 1149.1-2013

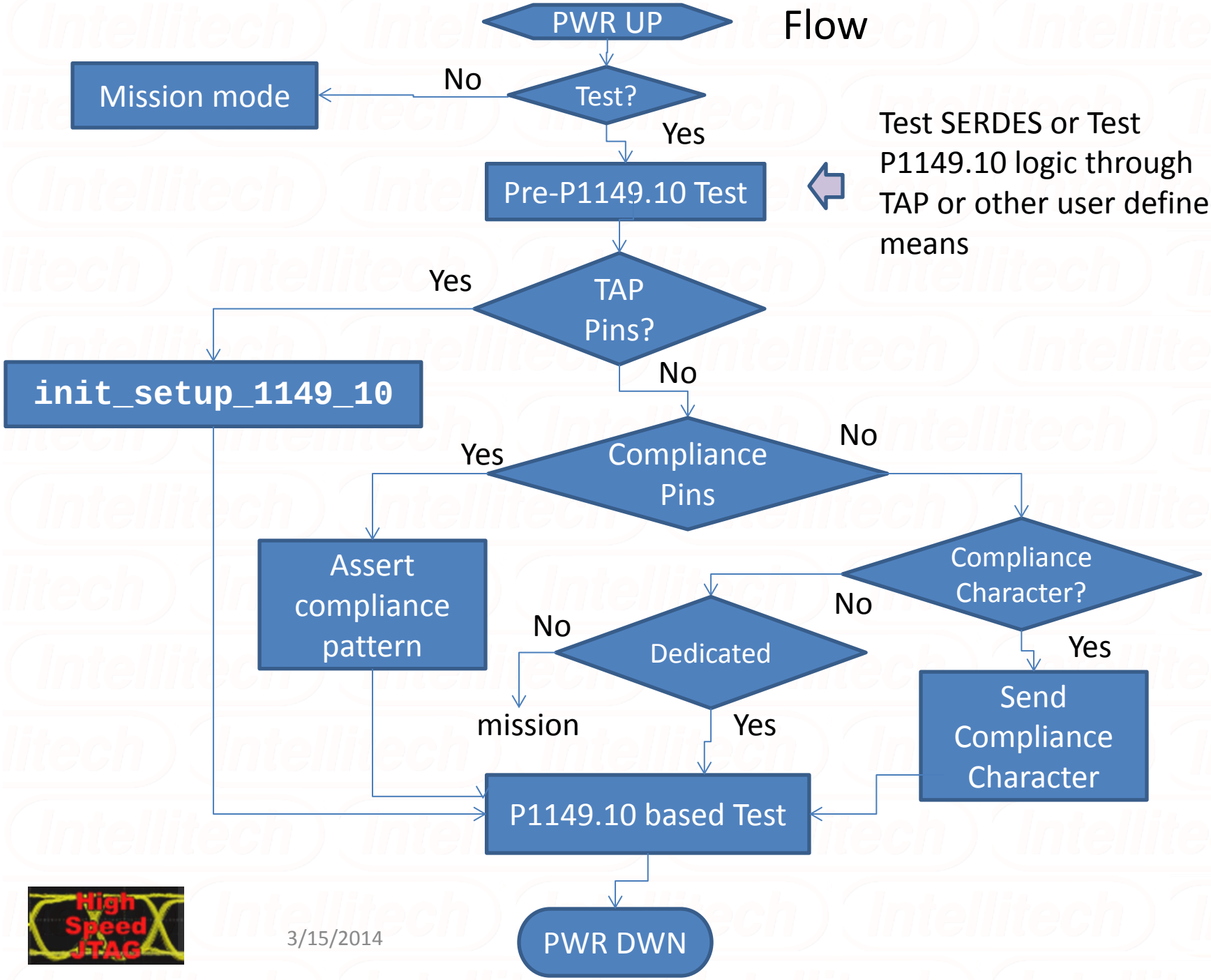
```
# MyCorp_SERDES.pdl
iPDLLevel 0 -version STD_1149_1_2013
iProcGroup MyCorp_SERDES

iProc enable_1149_10 {} {

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

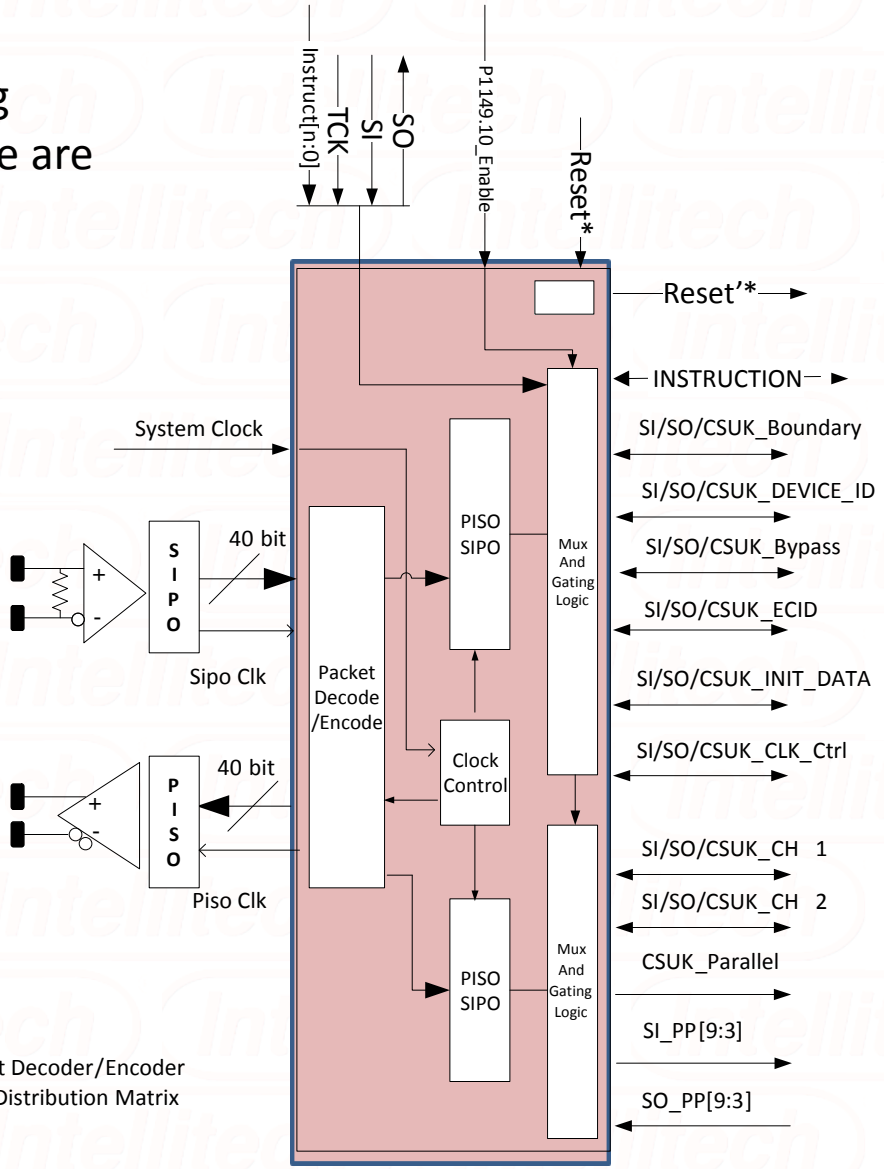
See 1149.1 PDL Tutorial for information of how this gets Translated to ATE patterns

Flow


PWR UP
Test?
No
Mission mode
Yes
Pre-P1149.10 Test
Test SERDES or Test P1149.10 logic through TAP or other user defined means
TAP Pins?
Yes
No
init_setup_1149_10
Compliance Pins
Yes
No
Assert compliance pattern
Compliance Character?
No
Yes
Dedicated
No
mission
Yes
Send Compliance Character
P1149.10 based Test
PWR DWN

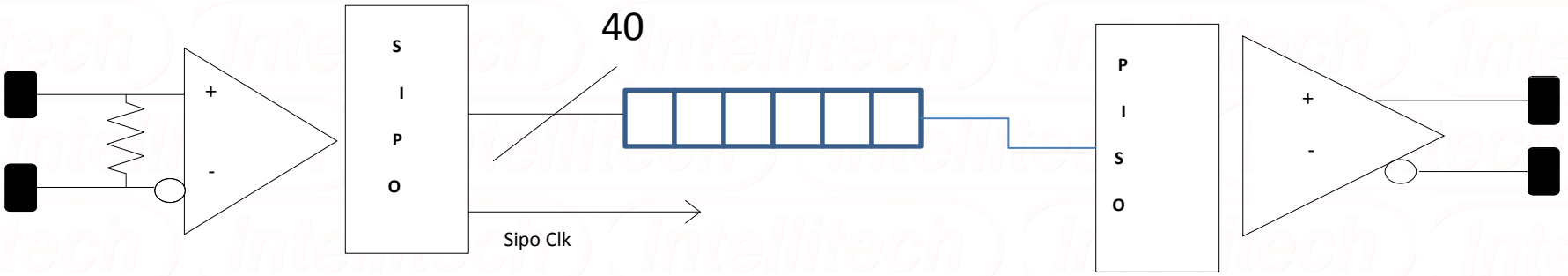
Approach with Instruction Register accessible

8b/10b, 64/66b, 128/129b encoding
Required in order to guarantee there are sufficient 1s/0s in the data stream such that the data and clock can be recovered at receiver.



Packet Decoder/Encoder with Distribution Matrix

Alternatives - Remove packet protocol
- Assume 8b/10b encoding is managed in PHY
- Match scan-chains to width of parallel side of PHY
  - 40 bit wide PHY then 40 scan-chains



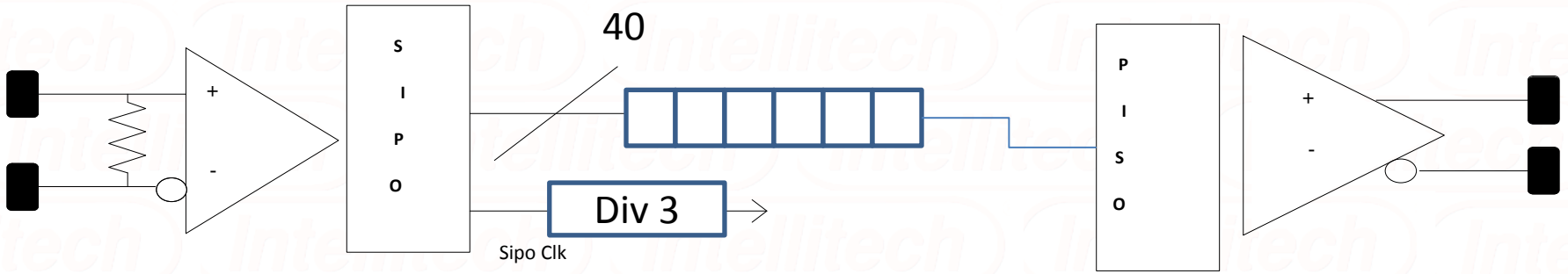
Scan-chains driven by SIPO Clock.

For OIF CEI-25G-LR and 802.3bj 25G/sec rate with 64 bit wide PHY yields 390mhz SIPO scan clock.

Instruction register, device_id and all chains required to run at same clock rate
For efficiency, scan-chains count must be multiple of PHY width. DFT dependent on mission mode PHY characteristics.
Loss of board/3D/MCM use of P1149.10 Tx to P1149.10 Rx daisy chain

Alternatives - Remove packet protocol

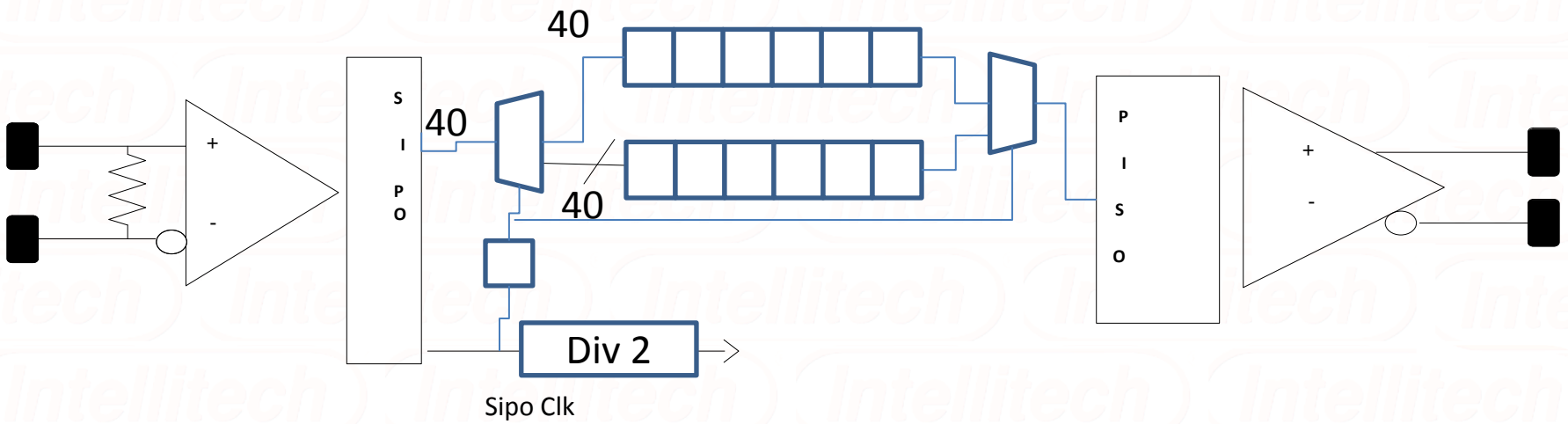
- Could throw away two words of three to get lower clock rate



Cost of probe card etc for 11Gbit high-speed interface but yielding
Only 3.6G/sec bandwidth.

Alternatives - Remove packet protocol

- Could multiplex 40 bit scan chains



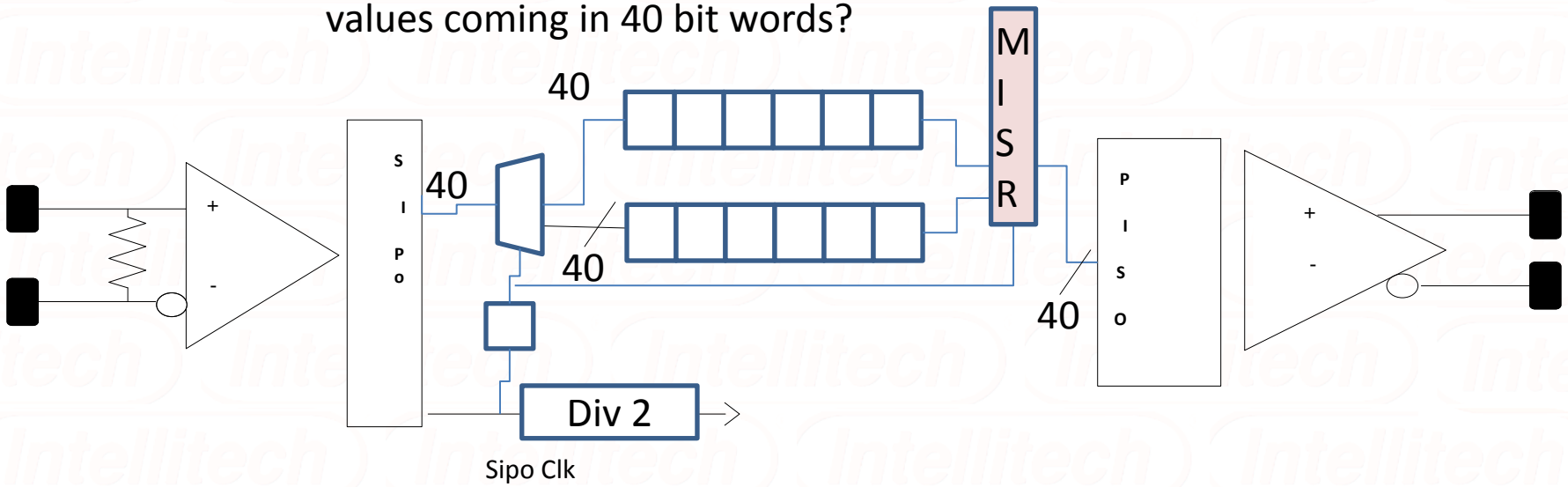
DFT of scan-chains relies on PHY width. Must be multiples of PHY width
Or loss of bandwidth.

41 scan-chains would cut bandwidth to IC as 41/80. 11G interface would
Have a bandwidth of 5.6G/sec

Assumes a VLSI test with large numbers of concurrent active scan-chains
Instrument networks, init_data, boundary registers may have less need for
large interleaved data

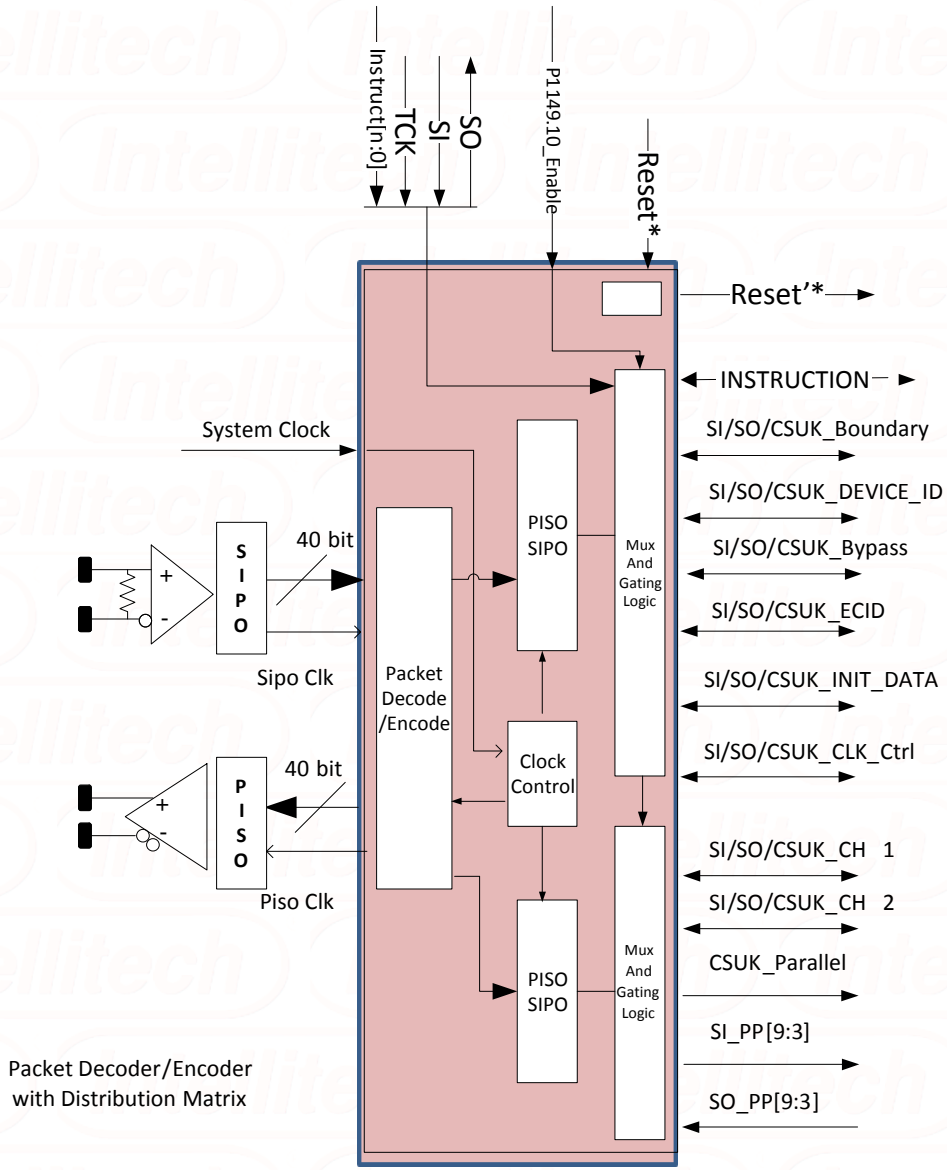
Alternatives - Remove packet protocol

Chains could be going to MISR - how to communicate the bits are MISR values coming in 40 bit words?



Assumes scan-chain/channels only for Wafer based test. Sets as a priority Over other needs such as in-the-field test/configuration.

Can this approach work with minimal logic?



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

SCAN - Interleaved IR/DR Scan Packet

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


SOP
CMD
PAYLOAD
CRC32
EOP

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.

SOP = Start of Frame
EOP = 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

SOP	TARGET	TARGET_ID	CRC32	EOP
	0x02	0x01		

SOP = Start of Frame
EOP = 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

SOP	RESET	TYPE	CRC32	EOP
	0x03			

SOP = Start of Frame
EOP = End of Frame

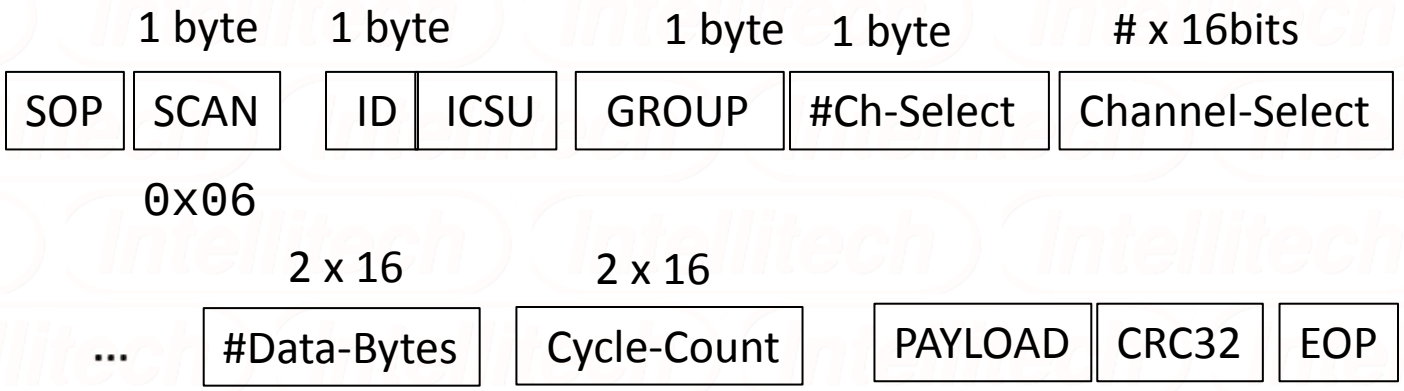
RAW - Put 1149.10 interface in raw data mode

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

SOP = Start of Frame
EOP = End of Frame

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

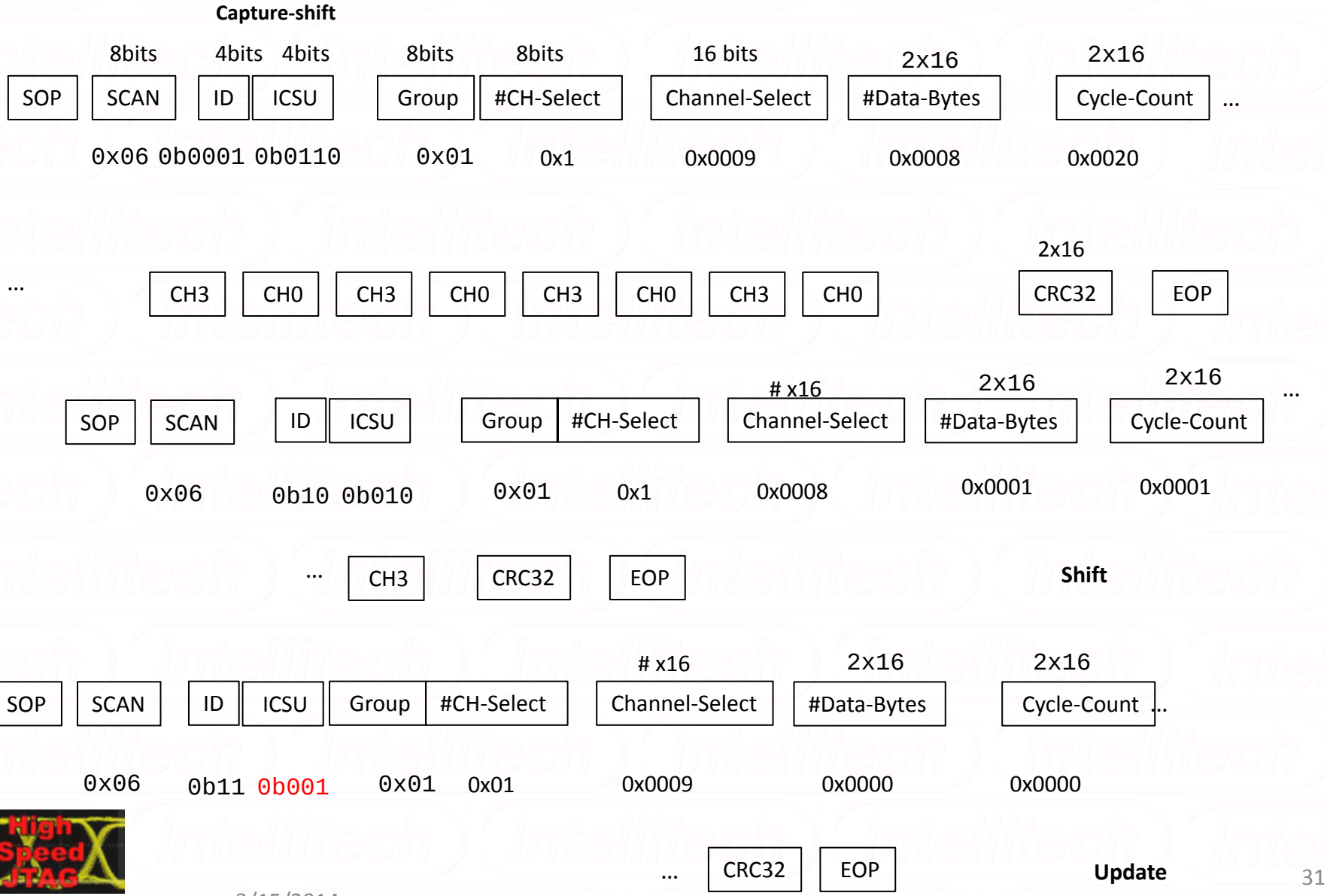
ICSU - IR Scan, Capture, Shift, Update

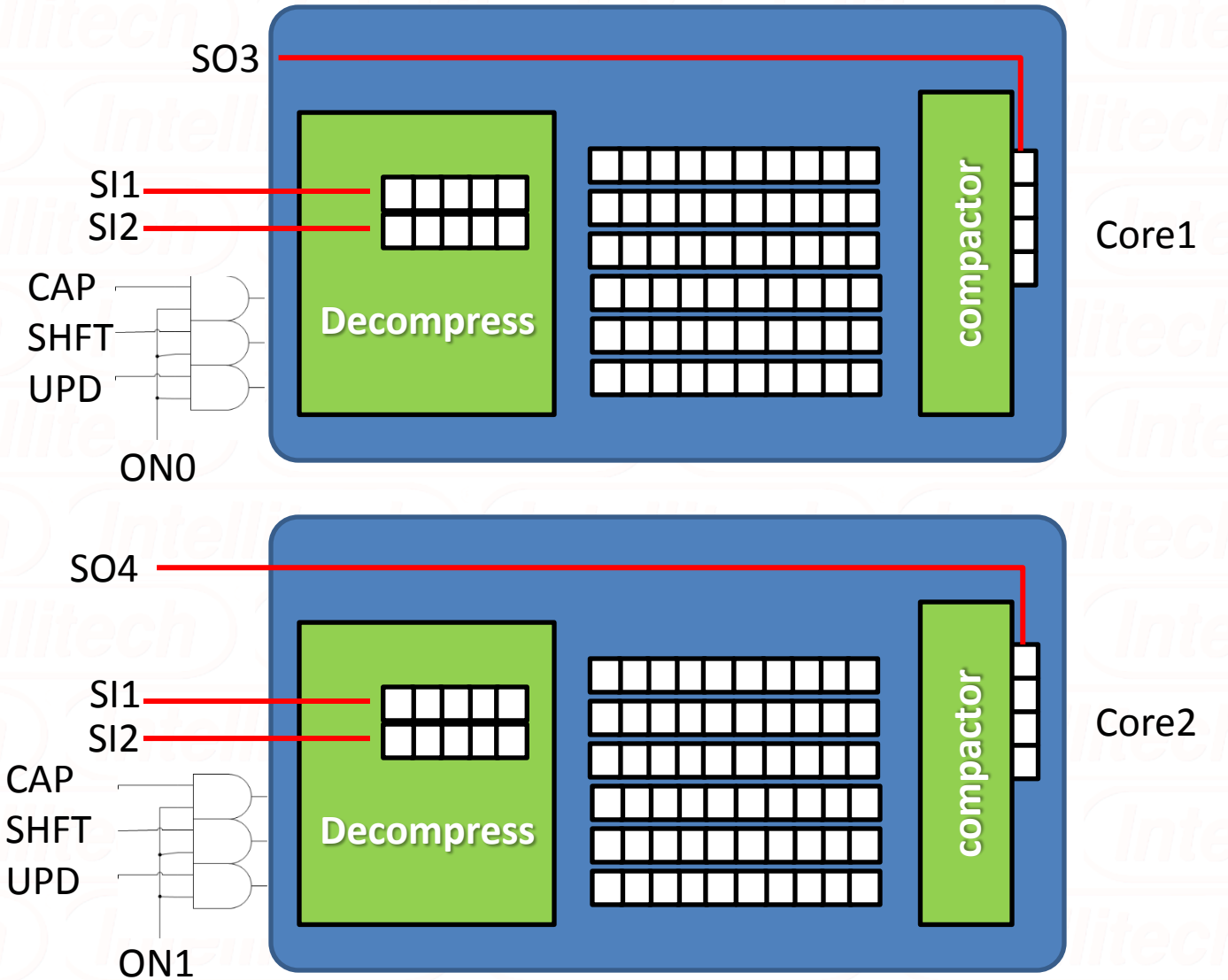


Interleaved SCAN Packet Format - dealing with CHs of different lengths

CH0 = 32bits CH3 = 33bits ICSU = Instruction, Capture, Shift, Update

(Interleave does not need to be transmitted but it is 8 bits in this case)

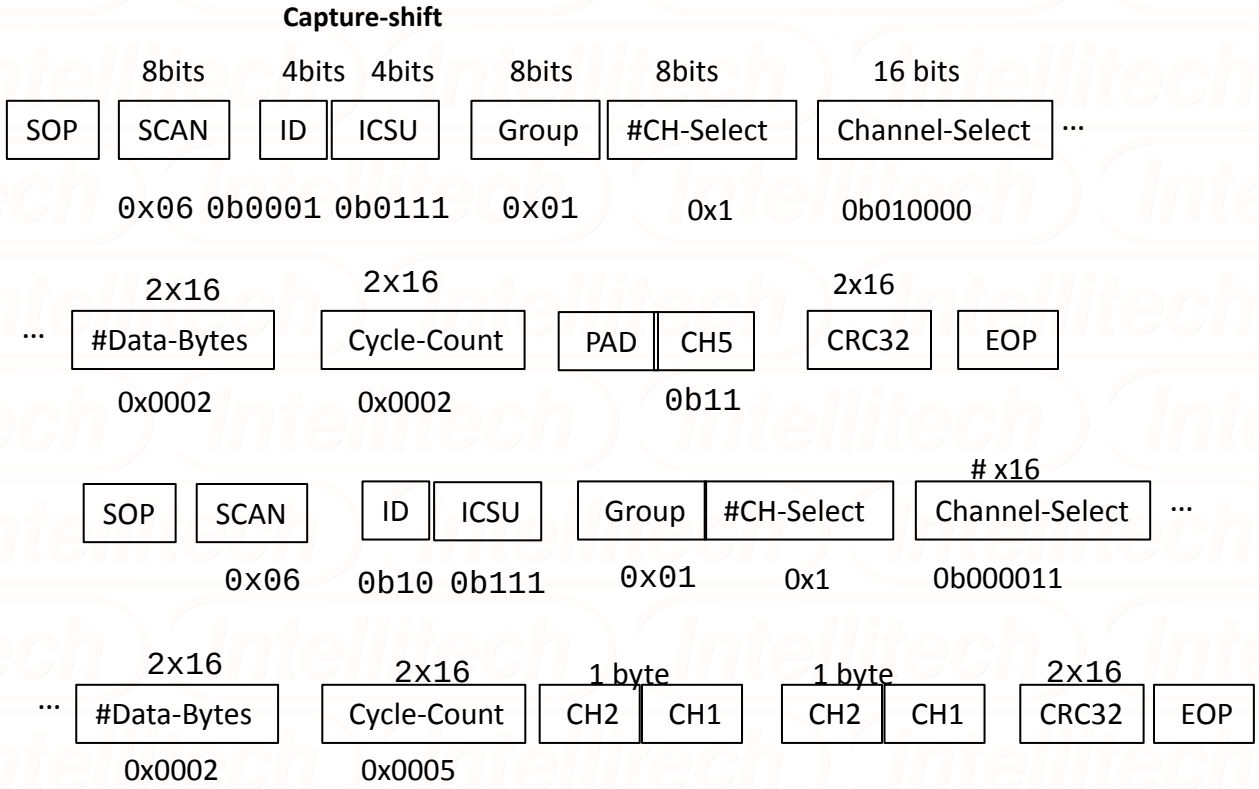




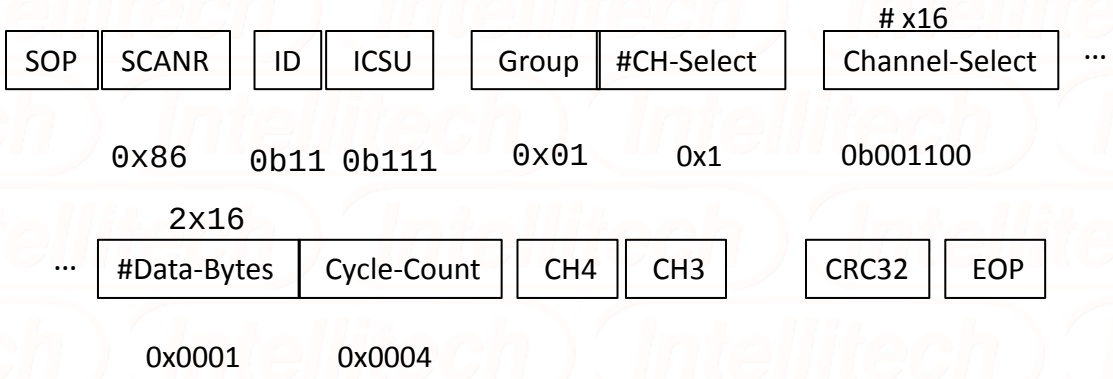
11 = broadcast, 01= Core1, 10 = Core2

SO4 has to equal SO3

Interleaved SCAN Packet Format - CH1 = 5 bits CH2 = 5 bits CH3 = 4 bits CH4 = 4 bits, CH5 = 2 bits



Response


SOP
SCANR
ID
ICSU
Group
#CH-Select
x16
Channel-Select
…
0x86
0b11 0b111
0x01
0x1
0b001100
2x16
…
#Data-Bytes
Cycle-Count
CH4
CH3
CRC32
EOP
0x0001
0x0004

Potential Out-bound Packets

TARGETR - Target Packet Response

CONFIGR - CONFIG Packet Response

RAWR - RAW Packet response

RESETR - Reset Packet response

SCANR - Interleaved Scan Packet Response

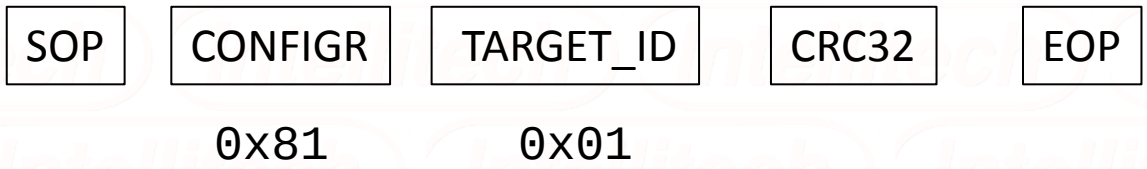
IDLE <n> - Tell ATE to insert N IDLE packets (*is this needed?*)

All inbound packets following a TARGET for an alternative device

CONFIGR

CONFIGR - Response to CONFIG command

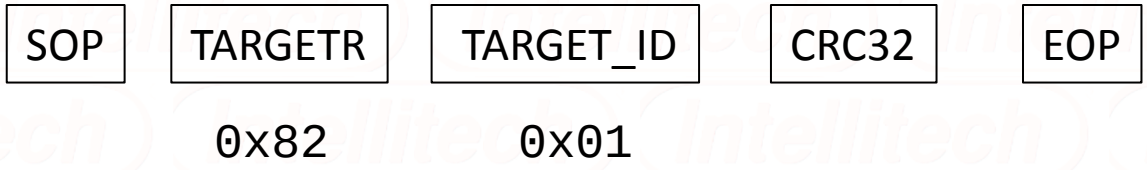
Device powers up with TARGET_ID of 0000. All packets received when ID Is 0000 are processed. Once TARGET_ID is set, only packets following a TARGET packet with TARGET_ID of same will be processed.



SOP = Start of Frame
EOP = End of Frame

TARGETR

TARGETR - Response packet



SOP = Start of Frame
EOP = End of Frame

RESETR - Issue reset* Response

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

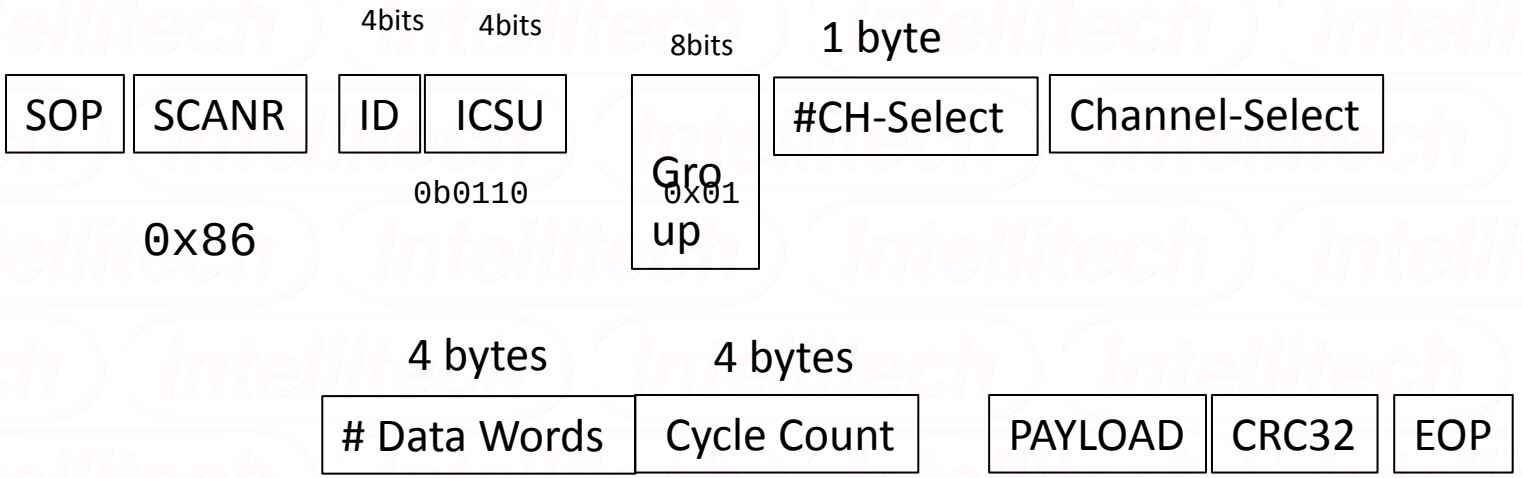
SOP	RESETR	TYPE	CRC32	EOP
	0x83	0x00		

TYPE = TMS 0x00 or TRST* 0x01

SOP = Start of Frame
EOP = End of Frame

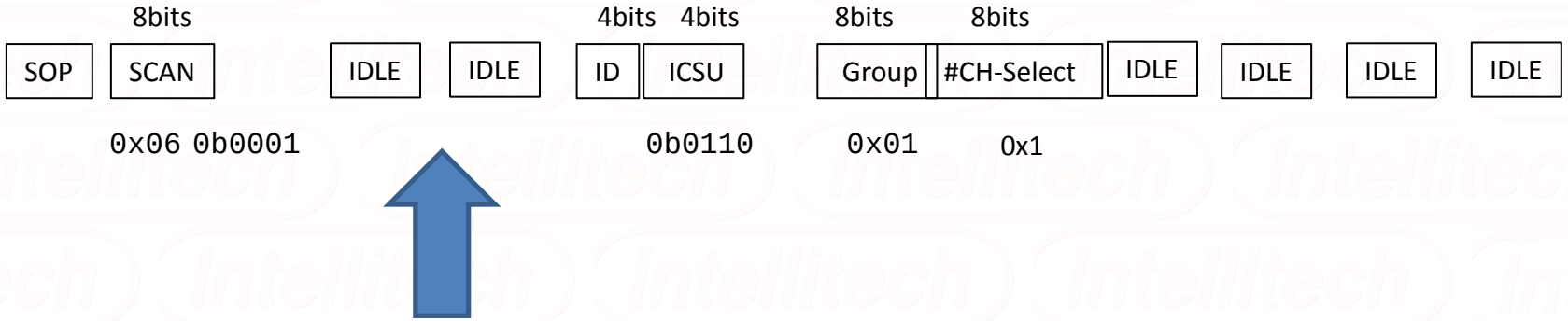
SCANR Packet Respond to IR or DR scan chain(s) as needed.

Can this be condensed?



Special Characters

K28.5 IDLE character



IDLE characters can appear anywhere in the transmitted data stream. Receivers filter out IDLE characters.

Special Characters in 8b/10b

Control symbols

Input				RD = −1	RD = +1
	DEC	HEX	HGF EDCBA	abcdei fghj	abcdei fghj
K.28.0	28	1C	000 11100	001111 0100	110000 1011
K.28.1 †	60	3C	001 11100	001111 1001	110000 0110
K.28.2	92	5C	010 11100	001111 0101	110000 1010
K.28.3	124	7C	011 11100	001111 0011	110000 1100
K.28.4	156	9C	100 11100	001111 0010	110000 1101
K.28.5 †	188	BC	101 11100	001111 1010	110000 0101
K.28.6	220	DC	110 11100	001111 0110	110000 1001
K.28.7 ‡	252	FC	111 11100	001111 1000	110000 0111
K.23.7	247	F7	111 10111	111010 1000	000101 0111
K.27.7	251	FB	111 11011	110110 1000	001001 0111
K.29.7	253	FD	111 11101	101110 1000	010001 0111
K.30.7	254	FE	111 11110	011110 1000	100001 0111

Special Characters in XAUI 64/66b

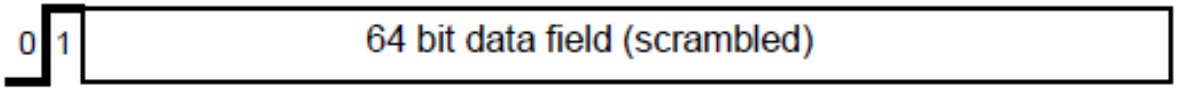
- XAUI defines the following tools (characters):

data (d)	Dxx.y
align (A)	K28.3
sync (K)	K28.5
skip (R)	K28.0
start (S)	K27.7
terminate (T)	K29.7
error (E)	K30.7
ordered set (O)*	K28.2

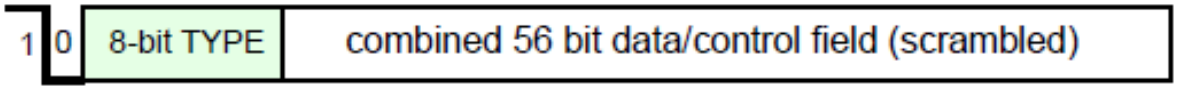
* Proposed to support FC Ordered Sets

Code Overview

Data Codewords have “01” sync preamble



Mixed Data/Control frames are identified with a “10” sync preamble. Both the coded 56-bit payload and TYPE field are scrambled



00,11 preambles are considered code errors and cause the packet to be invalidated by forcing an error (E) symbol on the HARI output

Code Summary

Hari Pattern	Sync		Bit fields 0-63							
DDDD/DDDD	0	1	D0	D1	D2	D3	D4	D5	D6	D7
ZZZZ/ZZZZ	1	0	0x1e	Z0	Z1	Z2	Z3	Z4	Z5	Z6	Z6
ZZZZ/SDDD	1	0	0x33	Z0	Z1	Z2	Z3		D5	D6	D7
SDDD/DDDD	1	0	0x78	D1	D2	D3	D4	D5	D6	D7
TZZZ/ZZZZ	1	0	0x87		Z1	Z2	Z3	Z4	Z5	Z6	Z7
DTZZ/ZZZZ	1	0	0x99	D0		Z2	Z3	Z4	Z5	Z6	Z7
DDTZ/ZZZZ	1	0	0xaa	D0	D1		Z3	Z4	Z5	Z6	Z7
DDDT/ZZZZ	1	0	0xb4	D0	D1	D2		Z4	Z5	Z6	Z7
DDDD/TZZZ	1	0	0xcc	D0	D1	D2	D3		Z5	Z6	Z7
DDDD/DTZZ	1	0	0xd2	D0	D1	D2	D3	D4		Z6	Z7
DDDD/DDTZ	1	0	0xe1	D0	D1	D2	D3	D4	D5		Z7
DDDD/DDDT	1	0	0xff	D0	D1	D2	D3	D4	D5	D6	

Control Characters for P1149.10 Unencoded/UnScrambled
128/130 encoding should be similar to 64/66
64/67 (Interlaken) same as 64/66 with 3 bit encoding (bit 66 indicating invert)

	8b/10b			64/66					
S (Start of Frame)	1	0xFB		0b10	0xFB	0xXX	0xXX	0xXX	
T (Terminate/EOF)	1	0xFD		0b10	0XFD	0xXX	0xXX	0xXX	
K (IDLE)	1	0xBC		0b10	0xBC	0xXX	0xXX	0xXX	
E (ERROR)	1	0xFE		0b10	0xFE	0xXX	0xXX	0xXX	
A(Stop)	1	0x7C		0b10	0x7C	0xXX	0xXX	0xXX	
R (Resume)	1	0x1C		0b10	0x1C	0xXX	0xXX	0xXX	
O (Reset)	1	0x5C		0b10	0x5C	0xXX	0xXX	0xXX	

Reset can be used to reset IC receive channel when errors or other occur.
It can also be used to get out of 'raw' mode.

Attribute PHY_1149_10 of MyChip : entity IS
"(SATA_TXP, SATA_RXP, 3.125E9, 500E-3, 800E-3, 8B_10B) ";

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

<phy description> := attribute PHY_1149_10 of <entity> IS <phy_string> <semicolon>

<entity > := <component name> - defined in 1149.1 already

<phy_string> := <left paren><phy_list><right paren>
{<comma> <left paren><phy_list><right paren> }
<phy_list> := <tx> <comma> <rx> <comma><rate><comma><min swing><comma>
<max swing><comma><encoding>

<tx> := <representative port> | <portID>
<rx> := <representative port> | <portID>
<rate>:= <real> (bits/sec)
<min swing> := <real> (in Volts)
<max swing> := <real>
<encoding> := NONE | 8B_10B | 64B_66B | 64B_67B | 128B_130B | <mnemonic_identifier>

```
Attribute Packet_Map of MyChip : entity IS
"(Interleave_Size:  8 ), " &
"(Scan_Group 1 : 1 to 16 )," &
"(Scan_Group 2 : 17 to 32, 33 )";
```

<Packet_Map_Description> := attribute PACKET_MAP of <target> : entity IS
<Packet_Map_String> <semicolon>

<Packet_Map_String> := <Interleave_string> { <comma> <scan_group_string> }

<interleave_string> := <left paren> INTERLEAVE_SIZE <colon> <integer><right paren>

<scan_group_string> := <left paren> SCAN_GROUP <integer>
<colon><chain_list><right paren>
<chain_list> := <chain_field> { <comma> <chain_field> }
<chain_field> := <range> | <integer> - these elements defined in 1149.1

Communicates the interleave size of the packet and the mapping of the groups chosen by the designer of the decoder/encoder in the IC.

```
Attribute CONTROL_CHARS  of MyChip : entity IS
"(SOP :    0xFB ), " &
"(EOP  :   0xFD )," &
"(IDLE :   0xBC),"&
"(ERROR : 0xFE ),"&
"(XOFF :    0x7C ),"&
"(XON:      0x1C),"&
"(RESET :   0x5C)";
```

<control_character_description> := attribute CONTROL_CHARS of <target>
<colon> entity IS <control_string> <semicolon>
<control_string> := <left paren><control_char> <colon> <hex pattern> <right paren>
{<comma> <left paren><control_char> <colon> <hex pattern> <right paren> }

<control char> := SOP | EOP | IDLE | ERROR | XOFF | XON | RESET

A semantic rule in the draft would require all <control char> to be defined
All <control char> should be passed on via the TX pins when the P1149.10 Interface does not match the current TARGET_ID.

-- example using 64_66B

Attribute CONTROL_CHARS of MyChip : entity IS

```
"(SOP :     0x2_FB_XX_XX_XX_XX_XX_XX_XX ), " &
"(EOP  :    0x2_FD_XX_XX_XX_XX_XX_XX_XX )," &
"(IDLE :    0x2_BC_XX_XX_XX_XX_XX_XX_XX )"&
"(ERROR : 0x2_FE_XX_XX_XX_XX_XX_XX_XX  )"&
"(XOFF :    0x2_7C _XX_XX_XX_XX_XX_XX_XX )"&
"(XON:      0x2_1C_XX_XX_XX_XX_XX_XX_XX )"&
"(RESET :  0x2_5C_XX_XX_XX_XX_XX_XX_XX )";
```

CH1: AEI
CH2: BFJ
CH3: CGK
CH4: DHL

Channel-Select = 0b1111

DATA_SIZE = 8
INTERLEAVE_SIZE = 1
Cycle Count = 3