Date – 10/21/2013
Attendees: CJ Clark, Adam Ley, Bill Tuthill, Brian Turmelle, Bob Gottlieb, Craig Stephan, Dave Armstrong, Dharma Konda, Dwayne Burek, Frans de Jong, Gobinathan Athimolom, Ismed Hartanto,Josh Ferry, Marc Hutner, Mike Ricchetti, Philippe Lebourg, Steve Sunter, Zahi Abuhanmdeh

Missing: Kent Ng, Tom Waayers, Bill Huott, Tapan J Chakraborty, Saman Adham, Kevin Gorman, Roger J. Sowada, Gurgen Harutyunyan, Teresa McLaurin

Agenda:

Further discussion on P1149.10 WG P & P
USB 3.0 and IEEE 802.3-2012 PDF files in Private area
(Material on 8B/10B encoding/decoding)

To access the private area:

http://grouper.ieee.org/groups/1149/10/private/

Continued discussion on distribution matrix and packet format.

New Business

Meeting Called to order at 11:00 am EDT

Minutes:

Review of Rev 2 of the Policy &Procedures document that CJ has drafted. This had gone out over email
CJ would like to vote up or down for the P&P next week.
Steve would like to have lots of notice for voting on the P&P and does not object to waiting until next week.
No questions or concerned were raised regarding the P&P sent out.
Frans – This there might need to be some solicitation for input since everyone was quite. May need to poll a few people to if they are for the P&P.
CJ – Feels that no one spoke up when asked for objection so we should go forward with a vote next week without too much concern.
Frans – P&P is not taken lightly so just checking.
CJ – points out that many other working groups don't have P&P and they are not bothered by that. Wants to get one in place so we have an objective to follow when things come up.

Private Area
CJ has put PDF files of the USB 3.0 spec and the IEEE 802.3-2012 spec in the private area. Can see some of 802.3's protocols for examples for us to use.

CJ gave the username and password for private area during presentation.
Adam – Verified that the password works
CJ has not prepared any new material for this meeting.
Review of distribution network diagram dated 10-14-2013
In bound packet would contain a PID (packet ID) taken from USB3.0
Output packet would have a tag that would show that it was a return packet.

XOFF packets may tell the ATE that it can add IDLEs and not need an IDLE packet

Frans – if you are in a number of IDLE packets. Would you allow interference from the DOT1 TAP?
CJ – No. This would be up to the working group but doesn't seem practical. It's all or nothing. Either p1149.10 mode or 1149.1 mode. More work than we want to manage to interleave the two.
Steve – Need to have a method to get back to 1149.1 since SERDES interface would be less reliable.
CJ- we are going to use a Test Logic Reset.
Steve – so that would be the only thing that would allow 1149.1 traffic?
CJ – Yes. We could possibly allow an instruction to switch back.
Steve- would want to have enable that has superiority.
Frans – a reset would be disruptive.
CJ – the reset* signal has to control
Steve – if you have multiple SERDES interfaces and one of them screws-up you should have a method to disable only one SERDES and not reset the whole chip.
CJ – This would be accomplished by deasserting the p1149.10 enable.
Frans – The signal would be directly controlled and not through a TAP instruction.
Steve – It is implied by the diagram that the enable would revert to TAP control.
CJ – Would like to think of it as reverting back to mission mode.
Steve – not sure we would want that. If you disable the p1149.10 interface now the TAP is back in control still in test mode. That is less disruptive. If you go back into mission mode you disrupt everything.
Dwayne. - CJ is talking about mission mode on the SERDES device.
CJ – We are all taking about slightly different things.

Haven't determined the method to go in to p1149.10 mode. When the chip powers up, the SERDES should be in its normal mission mode and there will be no data coming into packet encoder/decoder. So the question is "what is the method to enable p1149.10?"

Use INIT to setup SERDES. Will want to setup the diff swing in a valid range and enable the p1149.10 mode.

Philippe – Feels that this should be an extension of 1149.1
Dwayne – technically if we are part of the reset sequence.
CJ – This is not an extension of 1149.1. Would require having instructions and a TAP if it was an extension of 1149.1.
Philippe – Don't you require these things for p1149.10?
CJ – We could want a dedicated SERDES.
Philippe – Is interested how they would use the SERDES before test of the whole circuit

Steve – Didn't we say that we always wanted a TAP because it is reliable? And the SERDES is not.
CJ – Not sure the p1149.10 rules would require a TAP and require to control everything through the TAP.
Philippe – You would need some reliable interface to setup the SERDES.
CJ – it is a Test Access mechanism.
Philippe – if your test relies on this, how can you expect the customer or manufacture to rely on SERDES?
CJ – someone may potentially want a dedicated P1149.10 interface. And decide if the rules can rule that out. Can't think of a reason why you would want to have a TAP present.
Steve – the TAP is far more reliable. The SERDES reliability isn't considered "functionality".
CJ – That is subjective. We can create SERDES IP and have them with a high reliability rate. How does one go about testing the interface? The testing would be systematic like USB3.0 or PCI Express. The test communicates over the USB3.0 interface and looks for packets coming back and makes a decision. And tests more functionality based on the results.
Philippe – This is not true. We don't communicate on high speed on USB interface for test purpose.
Bob – Don't we need 1149.1 TAP to configure the SERDES interface? In our implementations we would use TAP or side band to configure it.
CJ – That is why we are talking about having the TAP.
Bob – we need the 1149.1 there to configure the SERDES so we know what the SERDES interface looks like.
CJ – Would you outlaw it through a rule to not have 1149.1 interface?
Steve – yes.
Bob – not sure.
Steve –The feeling Steve is hearing from the rest of the group is that a Dot1 TAP would be mandatory for lots of reasons.
CJ – Is not as convinced as you are. But as this unfolds we will see.
What we are going to describe is a high speed and distribution network. What will come down is to decide if we mandate a TAP interface.
This is not our mission to decide today however.
If you are an FPGA company who has tested their SERDES and know it works. Are you required to use the TAP to configure your device? The TAP is there for board test but you don't need to configure your SERDES to program your device. Would you have a rule to configure your SERDES through the TAP?
Philippe – are we speaking about test here?
CJ – we are talking about the interface?
Philippe – is it Test
CJ- it is not only about test.
Philippe – so TAP is not "TEST Access Port"
CJ – The Par shows it is for programming FPAGs as well.
Steve – it's called a TEST Access Port.
Dwayne – it is a grey area.

Steve – when it was conceived it was for only test.
CJ – early papers show it was for other things than test.
Steve – Would argue that is only for test.
Zahi – When we initialize a PH,Y having an 1149.1 TAP would make a difference.
CJ – the question is if you have a dedicated interface, one could follow all the rules and implement the 1149.10, would you allow it?
Dwayne – You could mandate it and if it is already configured this could be a NOP during configuration.
Dwayne – Has a question regarding the upper and lower set of registers.

In 1149.1 we have an Instruction register to select instructions. In this scheme, is there an Instruction register that is associated with this packet decoder?
CJ –It is decoded in the packet. Not a register per se, it is data in the packet.
Dwayne – there is not a state that would configure the register? Nothing that sets the clocking rate or signaling used?
CJ – You would have a dedicated IPROC procedure in INIT.

Data is coming through thru the p1149.10 interface in the packet and the decoder is picking out the scan chain from it. Doesn't need the instruction register. There is a MODE bit and that needs to be part of the packet format to tell it if it is in test mode or mission mode.
There could be a need to have the IOs held constant like the CLAMP function. Would like to have that ability like the instruction register can.
Reset would also be part of the packet format.
Dwayne – could have system clock and the shift clock is a ratio of that clock. At the board level the ratio might be different. If that is part of the packet as part of a mode that would be ok. So we need to convey that mode/setup information.
CJ – setup packet would be able to setup this information.
Dwayne – there would be a state of this decoder that would be like an Instruction Register.
CJ – clock control would be another setup register?
Dwayne – there would be a register associated with this decode for setup that isn't shown?
CJ – you could picture the clock control as one of these Data Registers. Could be more cumbersome if it is an Instruction Register when melding the 1149.1 interface with the P1149.10 interface.
Dwayne – as long as there is a register for the mode information that is ok.
CJ – if you wanted to load the boundary register but didn't want to be in test mode than you would set the mode bit through the p1149.10 interface.
Dwayne – do we standardize what the INIT Data register is so each 1149.10 interface has the same setup?
CJ – that packet is right for standardization. Data would be unique to manufacturer. Behavior rules should be standardized. Has Control packet. Up to the group. But a good place to standardize.
Dwayne – at least understand there is some necessity to deal with that. In dot 1 the TCK ratio is always 1 to 1. When we shift the TCK that shifts the Boundary Register. Here in 1149.10 we have a ratio. It is unique and doesn't exist in DOT 1.
Steve – The Dot 10 packet is addressable. And the Mode bits are another Data Register.

CJ – yes. Mode bits are configured through a data register. So you can set the Mode so the IO can be in TEST mode while you blast data in.
Steve – Rather than having broad mode bits, you could have a different mode bit on each register.
CJ – potentially you code do that, but thinking more in terms of 1149.1. Don't think we should expand the definition of MODE into something else.
Steve – If you are controlling the mode of the IO's, you could have a dedicated mode register, you could mix and match and put the IO's into whatever mod you want. And internal logic would be in mode
CJ – there are no mode bits in 1149.1 for these registers. So we don't have Mode bits. 1149.10 would see this just as packet data.
More going after this mode bit that says mission mode or test mode. Other modes would be generic packet data.
Steve – this mode bit in the diagram is a signal bit and has no other definition other than mission mode or test mode.
CJ – shown here as a signal bit.
Steve – the mode you are showing in the diagram is mode for the chip, DOT10 interface, boundary scan?
CJ – It is the traditional mode signal that comes from the TAP that is used for boundary scan.
Steve – only for boundary scan. Why have interface have something special for boundary scan? Why not for another register?
Dwayne – has that DOT1 dependency.
Steve – Understood you have mode bits for boundary scan. Why not have mode bits for anything on the chip? Why do we have to do anything special for mode bit for Boundary Scan.
Dwayne – that is where the 1500 interface is relevant.
CJ – this mode is special. Referring to the mode of the boundary. Other test data registers are packet information. It is data in the packet. What is unique by the mode of the boundary register is that it is set from the instruction register. Need an equivalent method for P1149.10 to set the mode.
Bob – any operation or mode that gets set by the instruction without a data register would fall under this category?
CJ – Correct.
Dwayne – do we want to mimic the instruction register in this interface?
CJ – no we don't because all of the other methods described are ad hoc
Dwayne – I think we will see that people encode that information in instruction register of the TAP. Comfortable making a data register that sets that information.
CJ – only thing that can't be represented as a data register is the Boundary Register.
Specially talking about Boundary Scan MODE on the diagram.
Other modes exist but represented as packet data. Data that is shifted into the scan chain.
Bob – we are saying we don't provide the ability to have an action on the Update IR. We need to specify it as a DR?
CJ – that is correct. That you should be doing anyway.
CJ – in 1149.1-2013 you have that same paradigm.

We are not adding more things into the Instruction Register.
We are doing this with the INIT Setup/Data register.

Steve – Motions to adjourn.
Frans- Seconds.

Meeting adjourned: 12:01EDT.

Action Items
Bill Tuthill – 10-21-2013 -Add minutes and Attendance spreadsheet to the website.

Next Meeting:
October 28th, 2013 11:00am

NOTES:

1149.10 working group website - http://grouper.ieee.org/groups/1149/10/

Conferencing software GoToMeeting
1. Please join my meeting.
https://global.gotomeeting.com/join/135671157
2. Use your microphone and speakers (VoIP) - a headset is recommended. Or, call in using your telephone.

Access Code: **135-671-157**
Audio PIN: Shown after joining the meeting

United States: +1 (213) 493-0618 Australia: +61 2 8355 1036
Austria: +43 (0) 7 2088 1036 Belgium: +32 (0) 28 08 4345
Canada: +1 (647) 497-9372 Denmark: +45 (0) 69 91 89 24
Finland: +358 (0) 942 41 5788 France: +33 (0) 170 950 586
Germany: +49 (0) 811 8899 6928 Ireland: +353 (0) 19 030 053
Italy: +39 0 693 38 75 53 Netherlands: +31 (0) 208 080 212
New Zealand: +64 (0) 4 974 7243 Norway: +47 21 04 29 76
Spain: +34 931 81 6713 Sweden: +46 (0) 852 500 182
Switzerland: +41 (0) 225 3311 20 United Kingdom: +44 20 7151 1801