

Date – 01/20/2014

Attendees: CJ Clark, Adam Ley, Bill Tuthill, Brian Turmelle, Bob Gottlieb, Craig Stephan, Dharma Konda, Dwayne Burek, Frans de Jong, Gobinathan Athimolom, Gurgun Harutyunyan, Ismed Hartanto, Jon Colburn, Josh Ferry, Marc Hutner, Steve Sunter, Tapan J Chakraborty, Zahi Abuhanmdeh,

Absent with Excuse: Teresa McLaurin,

Not Present for $\frac{3}{4}$ of meeting:

Missing: Bill Huott, Carol Pyron, Jim Wilson, Kent Ng, Kevin Gorman, Saman Adham, Tom Waayers, Dave_Armstrong, Heiko Ehrenburg, Mike Ricchetti, Philippe Lebourg,

Agenda:

- 1) Patent Slides
- 2) Philippe's proposal for error correction
- 3) More on BSDL Attributes
 - a. Proposed grammars in 1149.1 format
- 4) New Business

Meeting Called to order at 11:05 am EST

Minutes:

Review Patent Slide – Slide Presented to the Group.

Solicited input from anybody who is aware of patents that might read on our standard.

No Response

BSDL Attributes.

Another encoding scheme - 64/67 Encoding scheme (Interlaken) same as 64/66 with 3 bit encoding for better d/c bias.

Concerned we use a character in the coding and in the PHY it will generate an error. Problem not evident at this time, but concerned it could be in the future. This is the reason we want to communicate the encoding for the special characters in the BSDL. Should be the same all the time, but may need to be tweaked for a specific PHY.

Also added RESET character using "O" (0x5C) to indicate a reset without powering down PHY

Frans – can you imagine control characters not being defined?

CJ – later in the slides shows a grammar/syntax that needs to define all the characters.

Not sure what the gain would be to not define one of the characters.

IEEE 1149.10 High Speed JTAG Working Group Minutes

Slide 42 p1149_10_high_speed_jtag_basicsv10.pptx

describes 1149.10 bsdL syntax.

Describes encoding key words

8b_10b

64_66B

64_67B

128_130B

<mnemonic_idenfier> (would need some communication between IC supplier and Test manufacturer)

John – need to clean up 8b_10b to match other encoder scheme mnemonics.

CJ – will add b to other encoding mnemonics to make all mnemonics consistent.

<Scan_Group_String> Number is scan chain count starting at 1 (not Zero). Needs to match chip.

Marc – if you want to ignore a chain you might be able to transfer the data for the chains you care about and not the ones you don't care about?

CJ – this is not communicating the scan architecture inside the chip but defining the encoding of the packets to match the packet decoder inside the chip. Not grouping by scan clock

Marc – just saying what is the mapping?

CJ – correct. Just represents how I am sending the data.

When you do the grouping the grouping would only deal with the scan chains in a test at one time.

Dwayne – do we need an input group and an output group?

CJ – maybe. Was thinking of this in terms of packet format. But maybe it creates some sort of restriction. Has to match what your decoder is doing so you have groups.

Would you create groups differently?

Dwayne – would still like to describe the Mentor architecture. Needed to be symmetric at first, but once compression came along you could broadcast inputs and monitor outputs.

CJ – scan chain count is confusing because it is symmetrical. So counting needs to be non-symmetrical?

Dwayne – Treat it like inputs and outputs.

CJ – will update for next time.

Marc – would you enable different subsystems? Would you turn on the outputs of different cores that you are testing?

Dwayne – some way we might want this description but may want to ignore data from a specific core (a mask). Maybe that is handled outside on the tester?

Marc – depends on how much data you are talking about and how it grows.

CJ – This is just communicating the groups of the decoder

Marc- it depends on where on the stack you are applying something.

CJ – that is in the packet decoder and the architecture of the packet decoder applying it to the scan chains.

John – would want the tester to do the compare.

Dwayne – the masking is in the data. Data coming back from the chip can be don't care in a lot of cases. Maybe you want to do it at this level so you don't have to store the masked data.

John – having this description is helpful so the tester can do the masking.

CJ – this is not the data but the format of the packet. The chip is sending the response packet back, in some interleaved format. Would be giving you a header in the group and a numbering of what is coming back to you. The packet format is the data coming in and data going back as opposed to the expected data. We are not defining a language of scan data unless we find that is necessary. The tester is storing the expected data and knows where the X states are based on the ATPG scan data. The packet format is the format of the data coming back. Thinking we are just describing the packet information going back and forth and do what you need on the tester regarding the mask.

Dwayne – will these commands be used to configure the hardware itself?

CJ – picturing the modes are set through scan data.

Dwayne – is there a consistency check to make sure the hardware matches the expected mode.

CJ – you can make your own register. Not defining the DFT but designing the channel that moves the data back and forth.

CJ – do you envision different groupings based on the mode?

Dwayne - yes. I think you do.

CJ – so you think we need to expand this to illustrate multiple groupings?

Dwayne – yes. There will be multiple groupings. And implied the chip is in the right state when you start the test

CJ – what is your definition of Grouping?

Dwayne – enabling a certain access mechanism that brings the scan data or compressed data to “chip ins”

CJ – still think we are not at 100% alignment of what the definition is.

These are the scan chains that works together or part of a certain test and need to be grouped together. Just communicating the number in the group part of the packet means. And how it associates with the number of one-hot encoded data that follows. Just defining what the bits mean for the one-hot encoded bits.

Dwayne – changes thinking some, but thinking this is a meta command to explain what mode I'm in. but if not than I might need something else to explain what mode I'm in.

Do have a pin map which tells me the channel to pin mapping. That is a tester thing.

CJ – really communicating what the designer of the decoder did with this group information. How does he route the bits coming in when the group is a certain value. Expressing the decoders design. As opposed to what is possible in Scan.

Control Characters

Should be consistent from one chip to the next. Might be helpful if we picked a control character that isn't supported by a PHY.

Uses of special K Characters

SOF 0xFB

EOF 0xFD

IDLE 0xBC

ERROR 0xFE

IEEE 1149.10 High Speed JTAG Working Group Minutes

XOFF 0x7c
XON 0x1c
RESET 0x5c

These would be defined different if you are using a different encoding.
Allows the tester to know how to send the information.
If it is in the BSDL it has some flexibility for the future.

No one objected to above use of control characters.

New Business.

Bob – would be good if some of us go through some case studies.
Will try to make one for next meeting

Motion to Adjourn : Dwayne
Seconded : Frans

Meeting adjourned: 12:01 EST

Next Meeting:

January 27th, 2014 11:00am

Motion Summary

0 motions made

Action Items

~~Bill Tuthill – 10-21-2013 – Add minutes and Attendance spreadsheet to the website.~~
~~CJ – 11-11-2013 – Reach out to ATE industry and Probe Industry to get update on future of ATE equipment to see which data speeds and protocols they are heading towards.~~

Philippe – Look into alternative method to create control information (pause, start, terminate, etc) rather than using K characters in packet.

Bob – create a case study to show use of Attributes

NOTES:

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

Conferencing software - Live Meeting

Click here to [Join the Meeting](#)

Audio Information

Computer Audio

To use computer audio, you need speakers and microphone, or a headset.

Telephone conferencing

IEEE 1149.10 High Speed JTAG Working Group Minutes

Use the information below to connect:

Toll: +1 (218) 862-1526

Participant code: 114910

First Time Users:

To save time before the meeting, [check your system](#) to make sure it is ready to use Microsoft Office Live Meeting.

Notes

Troubleshooting

Unable to join the meeting? Follow these steps:

1. Copy this address and paste it into your web browser:
<https://www.livemeeting.com/cc/intellitech/join>
2. Copy and paste the required information:
Meeting ID: DOT10
Location: <https://www.livemeeting.com/cc/intellitech>

If you still cannot enter the meeting, [contact support](#)

Notice

Microsoft Office Live Meeting can be used to record meetings. By participating in this meeting, you agree that your communications may be monitored or recorded at any time during the meeting.