

Date – 06/12/2012

Attendees: CJ Clark, Adam Ley, Adam Cron , Bill Bruce, Bill Eklow, Bill Tuthill, Carl Barnhart, Carol Pyron, Craig Stephan, Heiko Ehrenberg, Hugh Wallace ,Jeff Halnon, John Seibold, Josh Ferry, Ken Parker, Roland Latvala, Dharma Konda, Francisco Russi,

Missing with pre-excuse: Dave Dubberke,

Missing: Lee Whetsel, Matthias Kamm , Mike Richetti, Neil Jacobson, Ted Cleggett, Brian Erickson, Scott Wilkinson, Jason Chodora , Brian Turmelle, John Braden, , Kent NG, Peter Elias, Rich Cornejo, Roger Sowada, Sankaran Menon, Ted Eaton, Wim Driessen,

Agenda:

- 1) Patent Slides and Rules of Etiquette
- 2) Use LiveMeeting “Raised Hand” to be recognized and take the floor
- 3) Motion to allow SEGMENT naming of SEGMUX instances. Currently a SEGMENT can only be used with the SEGSTART, SEGSEL DOMCTRL.

9176 i) A <domain assignment> shall only appear with instances of the SEGSEL, SEGSTART, and DOMCTRL
9177 fields.
9178 NOTE—SEGSEL, SEGSTART, SEGMUX, and DOMCTRL fields are defined in the Standard BSDL Package file - see Clause
9179 B.9.

```
attribute REGISTER_ASSEMBLY of PwrDomStruc : entity IS
  "userreg ( "&
    "(myTDR[31]), "&
    "(SegSel1      IS SegSel  Segment(SEGAB) CHReset), "&  --64 bits
    "(segA[32]), "&
    "(SegMux1      IS SegMux  Segment(SEGAB) ), "&
    ")";
```

Naming both ends of the segment may improve readability (SegMux/SegStart could be very far away from SEGSEL)

- 4) Motion to change rule b) to limit this restriction to standard TDRs. This may mean possibly adding a rule to indicate that an excludable segment must be wholly within an excludable segment (no overlap of SEGMUX from different SEGMENTS. Problem: IP and DIE both can be supplied with SEGSELS/SEGMUXs. During IC integration, if a IP in a TDR has a SEGSEL/SEGMUX in it, then this preclude the designer from also having a SEGSEL/SEGMUX on the TDR. This currently allows TDR segments with package file to be compliant but when assembled with R_A, the design could be non-compliant. Nesting was thought to be too software intensive, however, limited nesting with a excludable segment wholly within an excludable segment seems necessary to support today’s ICs/

3832 **9.4.1 Specifications**

3833 **Rules**

3834 a) Any excludable test data register segment taking advantage of Permission 9.2.1 k) shall be immediately

3835 preceded (closest to TDI) by a segment-selector cell or a segment-start cell and immediately followed by

3836 the segment switching circuit **of the same SEGMENT**

3837 NOTE--See B.8.19 for a full discussion.

3838 b) **No additional excludable segment shall appear within an excludable segment in a standard defined TDR.**

No change needed:

B.8.21.1 Specifications

Rules

...

9299 e) When a SEGMUX segment is encountered, all <register assembly segments> ordered before the SEGMUX

9300 segment and after the closest preceding SEGSEL or SEGSTART segment shall be excludable as a unit.

Bill B. suggests this rule says one cannot nest, however I don't see how it is violated. The SEGMENT is still excludable as a unit

```
attribute REGISTER_ASSEMBLY of MyUserReg: entity IS
  "userreg ( "&
    "(myTDR[1]), "&
    "(SegSel1      IS SegSel  Segment(SEG1) CHReset), "&
    "(xyz_il      IS xyzip ), "&
this instance has a SEGSEL/SEGMUX too.
    "(SegMux1     IS SegMux Segment(SEG1) ), "&
  " )";

attribute REGISTER_ASSEMBLY of PwrDomStruc : entity IS
  "userreg ( "&
    "(myTDR[31]), "&
    "(SegSel1     IS SegSel  Segment(SEGAB) CHReset), "& --64 bits
    "(segA[32]), "&
    "(SegSel2     IS SegSel  Segment(SEGB) CHReset), "& -- 32
bits total
    "(segB[31]), "&
    "(SegMux2     IS SegMux Segment(SEGB) ), "& -- SegMux
SEGB must appear before SEGMUX SEGAB.
    "(SegMux1     IS SegMux Segment(SEGAB) ), "&
  " )";
```

- 5) Motion to make CLAMP_RELEASE a mission mode function. Right now both CLAMP_HOLD and CLAMP_RELEASE are test mode instructions. One cannot check or load the Bypass-Escape register while a system (medical, airborne etc) is functioning. As a safety there is no way to set the Bypass-Escape mechanism such that it is enabled, one must rely on the power-up state and (hopefully) the last B-E state being remembered. CLAMP_RELEASE would make sense to release the TMP and the b-s register control of the I/O. What if you don't want to go back to mission mode? Then don't load CLAMP_RELEASE.
- 6) Discussion of Field_Value_Assignments. Current parsing problem when draft reversed DEFAULT.<object> to <object>.DEFAULT. <object> COLON DEFAULT?
- 7) <field assignments> ::= { <field value assignment> } | [<field reset assignment>] |
- 8) 9469 { <field domain assignment> }
- 9) 9470 <field value assignment> ::= [<field ident> <period>] <value assignment>
- 10) 9471 <field reset assignment> ::= [<field ident> <period>] <reset assignment>
- 11) 9472 <field domain assignment> ::= [<field ident> <period>] <domain assignment>
- 12) 9473 <field ident> ::= { <instance name> <period> } <extended field name>
- 13) 9474 <instance name> ::= <segment ident> | <array instances>
- 14) 9475 <array instances> ::= <array segment ident> <bit list>

Meeting Called to order at 10:35am 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

Review of Working Group Meeting Guidelines

No Objections

Hugh had sent CJ a list of patents. CJ is discussing appropriate action with the IEEE

Discussion on Segment Select

Asking for the ability to name the seg mux. Syntax doesn't allow segment name on segment mux

Bill B - feels that this information is redundant.

CJ – first find the SegSelect . Then follow the tdi.tdo and find the segment mux. Make it explicitly by calling it out

Carl – has no objection. Makes it easier to figure out what the structure is.

CJ – nicer if you use SegStart and having your SegSel be something else.

Carol – makes sense to name them. Adds clarity and ease of use

Ken – SegCell how do you know which bit is controlling the segment.

CJ – from the segment name.

Bill B- if you add names that will be way to check that it matches ordering. Would be redundant with info that is already there.

Carl –would be a double check

CJ- agreed. For documentation purposes though we should put the segment name there.

Hugh – The ordering is done with a syntax or a semantic that enforces order.

CJ – it is a semantics rule

Carl makes motion - [Motion to allow SEGMENT naming of SEGMUX instances.](#)

Bill B seconds

Vote called

Yes

Bill B Carl B Francisco R John S

Bill T Carol P Hugh W Josh F

Brian T Craig S Jeff H Ken P

No

Abstain

Adam L John B Roland L

Motion passes 12 – 0 - 3

Item #4

[limit this restriction to standard TDRs](#)

CJ – afraid that the rules that we have will keep 1149.1 out of other standards.

Have a problem if we don't do anything. Both parties can be compliant but when the parts are added together the design will be non compliant.

Ken – don't see where the TDI side is in picture. Where is it coming from.

Carl – SegSel/SegMux isn't shown correct.

Ken – the left hand core is part of something that is being included in the USER TDR.

CJ – key part is the muxes having a SegSelect we are trying to branch around the two cores.

Ken – just wanted clarification on the picture.

Adam C – If the rules stay the way they are, then the integrator screwed up

CJ – How?

Adam C – by causing this nested selection scheme

CJ – he plugged in IP and didn't look into the IP.

Adam C – in 1838 we could add permissions to allow this

Standard defined TDR – could we change that to TDR defined by this standard.

CJ – we can change that

Carl – Today any private TDR can do what it wants. WE are talking about design specific TDRs that you want to have public and documented. It is worth doing

Entire argument also appears to apply to init data. Do you have the same problem with init data. Initialization segments that are down in the IP. Might be more likely to have a collision in init_data.

CJ – possible. Just trying to contain this as we have had some people object to the nesting. Agrees that you are adding an extra design constraint on the init_data

Carl – instead of all standard TDR maybe it is the boundary register that we are concerned about. Maybe init_data should be allowed to have nested segments.

Carol – do we explicitly exclude segments from ID code?

Carl - yes

Ken – Init_Data has become complex from the beginning. So this looks like another level of complexity.

CJ – just for user defined TDRs. Should have something that is reasonable

Hugh – agrees with CJ. Wants to be able to pick and choose what he is using.

CJ – didn't dive into this. Want a well defined

Carl – provider would have to include the SegSelect bit.. And would probably meet the rules today.

Might be appropriate to keep all the seg cells at the top level and the switching at the lower level

CJ – just trying to limit the restrictions to rule b

Bill B – Recommends taking a shot at what the permission would be. Something like an additional excludable segment can exist with an excludable segment.

CJ – still want to modify rule b)

Bill B – Rule e might need to be modified

CJ – don't see anything right now.

Carl – possibly and will reword it

Adam C – do like 1 bit per one mux and should keep that.

CJ – additional rules – powers up in the short segment and powers up at 0

Roland - if we are going to name the segments and allow them on TDR. What was the objection of allowing them in the boundary register or any other register?

CJ – objection was to allow nesting in the boundary register.

Roland – going to learn how to manage nesting in the TDR. What is the objection of doing it in boundary register.

CJ – they are doing EXTEST based testing only.. so it is an extra hurdle for them to do testing with nested segments in the boundary register.

Carl – boundary register needs to be defined at the top level so it doesn't game anything from additional level of nesting

Ken – Could end up with a cyclic situation if we are not careful where the segment selects are . Need to avoid by strict rules.

Probably should ask for segment selects at high level.

Adam L – seems that the boundary register falls in the same scheme otherwise we need to describe the implications how the IO IP is presented, they can't have the segment select in the IP or it has to be possible to substitute the top level SegSelect that is in the IP.

Doesn't know why the problem can't be solved generally.

CJ – seemed to be most palatable to remove the standard TDRs. Can have SegSelect in the boundary register and ip that have SegSelect in the boundary register, but you can't nest. Not SOL completely on the boundary register. Don't have a reusable boundary selection.

Bill B- is it ok for the power domain to be different for each of these excludable segments.

Carl- yes that is ok. No issues.

Carl – Move to allow strictly wholly nested segments with design specific TDRs.

Bill B Seconds

Roland does that exclude the init-data register.

CJ – does not include but not restricting another motion from being made,

Vote called

Yes

| | | | |
|---------|---------|----------|----------|
| Bill B | Carl B | John S | John B |
| Bill T | Carol P | Hugh W | Josh F |
| Brian T | Craig S | Dharma K | Roland L |

No

Adam L

Abstain

| | | | |
|-------------|--------|---------|--------|
| Francisco R | Jeff H | Heiko E | Ken P. |
|-------------|--------|---------|--------|

Motion passes

12/1/4

CJ opened the discussion about the proposal to make the Clamp Release instruction a mission mode function.

Adam Cron asked about the timing of the setting of clamp mode. CJ said that it would be like any other IR access.

Carl stated that the motion would make the CLAMP_RELEASE instruction select the TMP-Control register, not the boundary register.

Need CLAMP_RELEASE instruction in mission mode to ensure that test mode persistence is cleared.

Ken stated that we turn off the tmp control bit and then the next routine determines if the chip stays in or leaves test mode.

Ken proposed renaming the clamp_release instruction as tmp_release. CJ was ok with this.

In response to Ken P's concern that lower-level routines could set and release the test mode persistence, CJ mentioned that the user should use clamp hold and clamp release in higher level routine, not in lower subroutine.

Carl suggested that an IP provider could provide a routine that includes a clamp_release instruction. CJ made comment that this is not possible because the chip that the IP is integrated into may not support the instruction. Carl agreed and changed his opinion.

The user needs to guarantee that the tmp state is guaranteed to be off. Proposal is to allow the tmp_release instruction to let mission mode set the clamp mode off. The in-field tests could leave the chip in clamp mode. The change would allow for the tmp control register to set the bypass escape bit.

Clamp release did not clear the test mode, only tmp controller .

Carl stated that the tmp controller is currently clocked by the rising edge of tck and will change at least 1 cycle after Update-IR whenever the clamp release or clamp hold instructions are selected. Refer to figure 6-10 of the standard draft.

Adam Lee stated the need for the TMP state to change at a specific point in time. He also stated that this discussion is intertwined with the discussion of the mode tables. Carl thinks that the 2 issues can be solved independently.

Ken P – Output may change timing to change on rising edge. The model could be different than the existing method. Carl said that he could add a half-clock delay to make the output transition with the falling edge. Ken questioned if the different timing is a concern.

Bill Bruce also recommends that the output should continue to clocked with the falling edge of TCK.

CJ questioned the delay for the transition of the TMP output. Carl said it is due to the decodes of the instruction. We could add a pre-decode to make the output transition in Update-IR.

Adam L does care about when the I/Os change state. Falling edge of TCK on Update-IR is essential. We want a method to sample the persistence state or a way to release the tmp controller while remaining in test mode.

Consensus was that we need a way to read status of test mode persistence status. Another instruction was brought up.

Carl to look at the spec and see what solution may work best.

Meeting adjourned: 12:00 pm EST.

Summary of Motions Voted on

2 Motions voted on

- 1) **Motion to allow SEGMENT naming of SEGMUX instances.**
 - a. **12 yes/ 0 no/ 3 abstain**
- 2) **Motion to change rule b) to limit this restriction and add a permission for user defined TDRs to have nested segments.**
 - a. **12 yes/ 1 no/ 4 abstain**

Next Meeting: 6/19/2012 10:30 AM EST

NOTES:

1149.1 working group website - <http://grouper.ieee.org/groups/1149/1/>

To Join the meeting

<https://www.livemeeting.com/cc/intellitech/join?id=2CQ2PQ&role=attend&pw=n%26d%5DNqX%284>

Meeting time: Tuesdays 10:30 AM (EST) (Recurring)

AUDIO INFORMATION

-Computer Audio(Recommended)

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

-Telephone conferencing

Use the information below to connect:

Toll: +1 (218) 862-1526

Participant code: 11491

FIRST-TIME USERS

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

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: F9R6S6

Entry Code: k/d6<@M6j

IEEE 1149.1- 2012 JTAG Working Group Minutes

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.