Date – 05/29/2012

Attendees: CJ Clark, Adam Ley, Bill Bruce, Bill Tuthill, Brian Turmelle, Carl Barnhart, Carol Pyron, Craig Stephan, Dharma Konda, Dave Dubberke, Francisco Russi, Hugh Wallace, John Braden, John Seibold, Josh Ferry, Ken Parker, Peter Elias, Rich Cornejo, Roland Latvala, Wim Driessen,

Missing with pre-excuse: Adam Cron ,

Missing: Lee Whetsel, Matthias Kamm , Mike Richetti, Neil Jacobson, Ted Cleggett, Brian Erickson,
Ted Eaton, Roger Sowada, Bill Eklow, Heiko Ehrenberg, Jeff Halnon, Kent NG ,
Sankaran Menon,

Agenda:

1) Patent Slides and Rules of Etiquette
2) Use LiveMeeting "Raised Hand" to be recognized and take the floor
3) Motion to use IEEE Standard 1364 Verilog definition of 'expression' for use with register constraints. (Standard was put on website in private area). We may not support any more operators than what has been proposed, but the form of the expression would follow 1364 and we will reference the standard (including as much detail as possible such that a user does not need 1364 to implement a constraint, but perhaps a tool vendor can reference it and our expression form follows that of another standard. The alternative is the editor creates a new expression format which we have only seen a partial definition to date.
4) Motion to use { } in the grammar for REGISTER_CONSTRAINTS to delimit Mnemonic_identifiers from operators. Problem: M_I can include operator characters +3.3V, 125Mhz+-10%. This is useful in GUIs and coding. R_C uses these special characters for operators. Compiling a BSDL with a M_I which has a operator in it forces the user to use a space to separate out the operator from operand. Some don't like that. Discussion on Friday led to use of { } around M_I (this was proposed some time ago as well). Such that in the event you are using a M_I with a operator character in it you would have to use { } around it.

```
attribute REGISTER_CONSTRAINTS of XYZ_SERDES : package is
"init_data (" &
  " ((CMMV == {+1V} ) && (Protocol == SRIO) ) "&
     " WARNING <A CMMV of 1V is not valid with SRIO. "&
     "The driver will do it, but communication may not work.> "&
")";
```

5) Motion to support both INIT_SETUP and INIT_SETUP_TEST. INIT_SETUP would remain a mission mode instruction and INIT_SETUP_TEST is the equivalent. Flexibility is obtained by having both. Another possibility would be to mandate CLAMP_HOLD/CLAMP_RELEASE with INIT_SETUP, which I believe would give a similar behavior to the two choices. PRELOAD, CLAMP_HOLD, INIT_SETUP I believe is the same as PRELOAD, INIT_SETUP_TEST.

6) Ballot. Need to see a final draft based on the three above decisions. Time to review it and then proceed with balloting.

Meeting Called to order at 10:34 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 responses

Review of Working Group Meeting Guidelines
No Objections

Carl: makes a motion that we use a proper subset of IEEE standard 1364 Verilog definitions of 'expressions' for use with register constraints.
Brain T. seconds the motion.
Carl :
CJ: the current grammar in the draft is following what is in Verilog?
Carl: the grammar would not change. Some rules would need to change. We don't treat X the same way and that would change.
Adam L – won't meet the needs the needs in the standards as they were previously understood. Some variances of 1364 that need to be made that would not make a proper subset. It would seem that VHDL would be a more suitable target for the expression evaluation than Verilog. Does support reference to a standard.
CJ – onehot would be different.
Carl – subset I correct in motion but should be "modified subset". Motion is modified
Bill B- unclear what Yes means if you vote for it.
Carl – a yes here means that RegA can stand alone.
Bill B- operators?
CJ – wouldn't expand what is already in the current grammar.
Bill B- yes is status quo.
CJ – yes. But a few addition rules. Like defining True and False. Would leave grammar as is. Some clean up and work to do.
Hugh – key thing here is that we are trying to say that you don't have to put a conditional thing such as ==
If we are going to site the standard we should copy the eBNF and copy out what you don't want.
Also agree with Adam L that we should go with VHDL because the standard is based on it but can go either way
Carl – reason for not going with VHDL is that there is a lot more people out there that are used to expression style that are used in many different languages.

Not many VHDL expression evaluators. And PDL is going to evaluate the expression
CJ – believes that we have the same grammar. OneHot is different. But our grammar is the same
Bill B-wants to point out if you want to say REGA >0 you can still do it.

Motion to use a modified subset of IEEE standard 1364 Verilog definitions of 'expressions' for use with register constraints.
Yes
Bill B.
Bill T.
Brian T.
Craig S.
Dharma K
Francisco R
Hugh Wallace
Josh Ferry
No
Adam L
Abstain
Carl B.
Dave D.
John B.
John S.
Ken P.
Peter E.
Rich C.
Roland L.
Wim D.

8 yes 1 no 9 abstain
Passes

Carl – Motion to use { } in the grammar for register constraints to delimit mnemonic_identifiers
Bill B. seconds

Hugh – this is conventions. A VHDL identifier would work here.
Identifier needs to be used in PDL.
If we adopt this it has 2 consequences.

1) When you use an express, if the expression is complex it is hard to visually parse.
2) This does not lend itself to go into C/C++. Identifier will break most other languages

Bill B- motion is that all identifiers
CJ – was just planning on complex identifier.
Bill B – Situation in that the definition does not need it but the reference needs it.
In this case it is required to use it when a mnemonic id is referenced. That is what will keep the mnemonic id definition out of the way.
CJ – this would make it easier to see what a mnemonic is.
Bill B- definition does not it. Another alternative. Like $ in TCL.
Take the same approach here. You could use some introduction thing, but $ is not appropriate here. Maybe an @ sign.
CJ- does not have a closing though.
Bill B- rules would make a space to terminate it.

Adam L- states that he is against the motion as stated.
Ken – having a preceding character will not work if you have embedded operator signs in the identifier. I do think we need a pair of bracketing character.

Also support Hugh's concern. If this mnemonic is going to get sucked up into other languages, we should give guidance how to go into another language.
CJ – This is a reasonable compromise. That will let us keep the robustness of the identifier.
Bill B – does this carry into PDL.
CJ – only BSDL
Carl – point out that the definition of register mnemonic has been there for a long time and translating to other language is not dependant of this motion
Hugh – when we parse this and convert into PDL the {} disappears?
CJ – correct
Hugh – the problem is once you start writing identifiers the +/% is not in the right place. Unless you keep what the expression identifier is, you would have a hard time getting the language it is sent into to be able to understand this.
Bill B- points out that in the case of PDL for our purposes are in iread/iwrite. This proposal would keep the current definition. Without the {} in PDL you would have to do a process of elimination to know what you have.
Hugh – still have the alias problem.
CJ – PDL does accept the mnemonic identifiers that we have defined currently.
Ken – Hugh keeps bringing up points that are worrisome. Feel like we just don't have it worked out right.
Hugh – point is that this expression has to be evaluated in PDL, doesn't feel this is parable in any language.
Carl – raw expression doesn't get sent to PDL. There is an intermediate step
Hugh – implementation detail that you have in mind. Trying to keep the expression manageable by many languages. After you strip off {} the meaning of the expression is lost.

Question called
Motion to use { } in the grammar for register constraints to delimit mnemonic_identifiers

Yes
Bill B.
Bill T.
Brian T.
Carl B.
Craig S.
Dharma K.
Francisco R.
Josh F.
No
Adam L.
Hugh Wallace
John S.
Ken P.
Peter E.
Wim D.
Abstain
Carol P.
Dave D
John B.
Rich C.
Roland L
8 yes 6 no 5 abstain

Passes

Init_Setup and Init_setup_test

Would have 2 instructions in the draft rather than just 1.

Ken - would use use clamp_hold before or after

CJ – would use it before init_setup

Carl – in order to get test mode if you didn't want test mode you would do it after.

Ken – you would be turning the control of IO before you set it up.

Carl – had discussion of this.

Carl - **motion to support both Init_Setup(mission mode) and Init_setup_test(test mode)**

Carol seconded motion

Carl – in simple situations you don't need test mode. In more complex situations are more indeterminate. The flexibility is needed for the test engineer. Can't decide ahead of time what is right in any situation.

Adam L – having test mode isn't flexibility. Like telling something if it hurts when you walk into the wall why don't you try running into it.

Still working on text.

If we have this flexibility, much prefer to mandate the clamp hold than have the two distinct instructions.

Radical idea – once we have introduced the flexibility of PDL, why do we need the constraints of init_setup,init_data,init_run. Just have procedure called init_setup that can do all.

Carol – we would need in the text if you use the test mode init_setup_test, another chip on the board cannot expect to detect correct signaling levels if you told the output to drive a 1. Because the IO's haven't been fully setup.

Need to state that the capture DR of other parts should expect indeterminate results. The purpose of the test mode is to create a safe state on the board.

Roland- if we have both, does it allow one given device to have an init_setup and init_setup_test. Can you run them in sequence? May have to do away with init_run. Or is it an either or?

CJ – I think you can use both. Problem of which command happens first.

Carol – if you support one do you have to support both?

Carl – would be written as if you provide init_setup you have to provide init_setup and init_setup_test

Motion modified to add "as a pair"

Motion to support both Init_Setup(mission mode) and Init_setup_test(test mode)

Yes

Bill T.
Brian T.
Carl B.
Carol P.
Craig S.
Dave D.
Dharma K.
John B.
John S.
Josh F.
Ken P.
Rich C.
Roland L.

No

Adam L.
Dave D.

Abstain

Bill B.
Francisco R.
Peter E.
Wim D.

12 yes 2 no 4 abstained

Passes

Carl should have a new draft in the coming week.
CJ will try and give a hand with examples.
Adam L will try to get presentation out soon

Meeting adjourned: 12:03 pm EST.

Summary of Motions Voted on
3 Motions voted on

1. **use a modified subset of IEEE standard 1364 Verilog definitions of ‘expressions’ for use with register constraints**
   a. **Passed**
2. **use { } in the grammar for register constraints to delimit mnemonic_identifiers**
   a. **Passed**
3. **support both Init_Setup(mission mode) and Init_setup_test(test mode)**
   a. **Passed**

Next Meeting: 6/5/2012 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
   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.