Date – July 13th, 2010

Attendees: CJ Clark, Bill Tuthill, Adam Cron, Carl Barnhart, Wim Driessen, Adam Ley, Carol Pyron, Dave Dubberke, Ted Eaton,

Missing with pre-excuse:

Missing: Bill Eklow, Roland Latvala, Heiko Ehrenberg, Ken Parker, Francisco Russi,

Agenda:
Continue the discussion on the INIT attributes.

Minutes:
Meeting called to order at 11:04 am EST

Review of Fridays’ INIT meeting
  When describing register fields we have the ability to describe the capability of
  the field – update, capture, or both.
  Either Mnemonic or value connected to register.
Using keywords
  CAP – capture
  UPD – Update
  BOTH – Capture and Update

  Use : to separate keyword from register.

  Would like ability to specify “X” but 0X means Hexadecimal.

Attribute “register_field_associations” added. Allows ports to be broken out to show
pins.
Attribute “register_field” used to define break out of INIT_DATA register

Register Field Association – Not part of OO compliance enables or part of Boundary
Registers. Allow for diagnostics.
Very specific to INIT_DATA.

Carl - Any capture in INIT_DATA has to have a 1 to 1 relationship with an IO pin.
Digital Pins only. Do not attempt to capture analog or power.

UPD – tells us that this is a control type register. This is data that you would serially shift
in.

Carol – if update keyword means that there is an actual update stage flop than don’t like
that. Could be point of confusion.
Carl – update only means that we are shifting data in.
Ted – maybe use same terminology as in 1687?

CJ – UPD/CAP might be confusion. Maybe a write/read would be better (wr/rd)

Carl – register field name has only a length defined.
  Feels strongly that it should be a range not a length. If it was a random scattering
  of bits there is no way to address individual bits of a field later.

CJ – length tells us how wide the register is. It is a check.

Carl – with only a length we don’t have a way to correctly address the register.
Ted – “to” and “downto” are done in Register_Fields_Association.
Carl – if you infer it how do you deal with a scattering of bits that are not in order?
Ted – bits are always in order.
Carl – it is not clear where bit 0 is? Is it MSB or LSB. VHDL allows “to” and “downto”
  structure of register to define where the MSB and LSB are.
CJ – it will be FIXED (one way) in standard

Carol – do we need to allow people to go into a register field at all? or going into
INIT_DATA sufficient.
Carol – are the mnemonics legal PDL –?
CJ – Yes.

CJ – TDO is on LSB and TDI on MSB. That is in the standard.
  Least significant bits are bit 0
CJ – would like to keep that consistent.

Carl – always a “downto” and zero based?

Carl – given the way the boundary register is defined – after the field name there would
be a ( , separated and then )
  Does this create parsing problems?

Ted – may be right.. Example needs another set of Parenthesis. To be parsed

Carol – need to restrict that you can’t span across registers.
Carl – agrees. Would not allow.
CJ – agrees as well. Makes it easier.

Carl – should continue using “downto” and “to” for ranges. Be consistent with BSDL

Adam L – “to” and “downto” are only used in the context of Port statement and no where
else in BSDL.
Carol – should be consistent with the format that is already in place. Use “downto” and
  “to” in port statement and should continue in the rest of BSDL to be consistent.
Ted – should we allow the : in the Port statement?

Carl – critical to continue the “downto” and “to” in the Port Statement.
In the attribute it is not all that important to use “downto” and “to”. Could use : for range. This is just a string and VHDL doesn’t care.

CJ – attributes are outside of VHDL. Tool has to parse this regardless.
Carl – maybe take this offline and check to make sure nothing is broken using the :

Wim – how long can a line be. Be good to have all of the attribute on one line?
CJ – can use “&” to span multiple lines.

Carol – more comfortable with : for ranges but believes in consistency. Would like Ken Parker’s opinion on using : to describe a range.

WG Action Item.
*Need to decide on : vs Downto/to
*Need to decide on keywords for Update/Capture/Both.

Carol – Will Provide full example to group before Friday.

Ted – do we want to stick with construct that describes how synthesis is done?
CJ – right.. Does this matter? Adam C?
Carl – Doesn’t care if you specify if there is an update flop. Just want to see specification of read/write/both.

Adam C – Doesn’t have a preference. Sounds like read and write is more generic. Not sure we need to know about update/capture. Haven’t given it much through for ATPG.

Carol – Happy with read/write. Wanted to know if tools need to know if write is only a shift in write or update state. Is that relevant to the tools?

CJ – is in favor of write/read. Is there a concern about going through update state to affect change?
Ted – don’t think we need it.
CJ – make sure this works for people and doesn’t break if you don’t have update states for stages for init.
Ted – stating specifically that we can do polling.
CJ- would require us to write the rules to require update.
Adam C – sounds more like a Design issue.
CJ – agrees. However as a WG we are responsible for best practice even if outside our domain. Helps the standard if we are using best practice and works for designers.
Ted – state in rules that register may be polled randomly, than implies that it needs an Update register.
Carl – INIT DATA is designed to scan ONCE. Attribute to say if it is persistent across a TLR. Makes no difference if INIT_DATA has update register. For INIT_DATA it is a non issue.

Generic TDR may be going farther than what we want to do now.

CJ – confused on Ted’s polling. It’s on status
Ted- yes correct. Withdraws previous comment.

CJ – headed towards RD/WR/Both.
Carol – read / write is fine. Will remove proposal. Just don’t want to be same as port names.
Adam L – agrees with Carol’s point, avoid adding new keywords where we can.
Adam L – comfortable with UPD/CAP
CJ – but you are an expert.

Adam L- main reason to opt for keywords as compromise, have situation where both update and capture will use same mnemonic. Instead of duplicating we opted for BOTH keyword.

Ted – PDL doesn’t have concept of shifting in and shifting out. Has concept of read and write.

Carol – use iread/iwrite.
CJ – seems like the more elegant choice.
Carl – most likely won’t conflict
Adam L – can live with the compromise.
CJ – understand update and capture but could be confusing for people that don’t do 1149.1 every day..

Group is converging on
IREAD and IWRITE and BOTH
For keywords in Register_Field attribute

Meeting adjourned: 12:05 EST.

Next Meeting: 7/20/2010  11:00am EST

Action Item by Carl to elaborate on concerns that he has with OO s on power pins and any rules that would need to be added to the standard to address those concerns.

Current Issues listed and who will champion that issue.
1 Observe only. – Ken and Carl
1. Directionality linkage. - CJ
2. Power Pins. - Heiko
3. Pairing power pins with functional I/O - CJ
4. Sample / Capture. – Carol (Freescale) & Roland
5. TRST included in PCB level diagram. – Adam L.
6. Slow to Fall/Rise signaling issue – CJ
7. “No Connect” – Ken and Francisco.
8. Device ID – Still needs work
9. Low-Voltage self observe shorts coverage problem – JJ & Intel
10. Init – Carol & Carl

**Action Items:**

- CJ will post 1149.1 draft on website with line numbers to make it easier to refer to items in discussion
- Comment #10 CJ will take action to look at possibilities to add to the 1149.1WG website a document which shows which standards are based on 1149.1
- Comment #8 CJ will make changes to draft for observe only
- Comment #7 CJ will get in touch with Doug to get input regarding Comments
- Comment #5 CJ will Add a figure and little text to address TRST use with interconnection of components
- Comment #4 Adam L to add comment about TRST. Update figure 6.8
- Comment #3 Adam L will update language for any proposed change for this section.