Date – 06/24/2011

Minutes of the IEEE-1149.1 Working Group Friday meeting

Attendees:
Adam Ley,
Brian Turmelle,
Carol Pyron,
Craig Stephan,
Roland Latvala,
Carl Barnhart,
CJ Clark,
Francisco Russi,
John Braden,
Dave Dubberke,
Peter Elias,
Josh Ferry,
Heiko Ehrenberg,
Ted Eaton

Excused:
Ken Parker, Wim Driessen

Meeting called to order at 8:30 am MST

New Draft: P1149 1 Draft 20110617.pdf (_clean.pdf)

Agenda/Overview:
Continued discussion of REGISTER_FIELDS

Minutes:
Today's meeting focused the register descriptions. Should the description be additive (PI, PO, UPD) assuming a minimum 'shift only' TDR, or should it be subtractive (NO_PI, NO_PO, NO_UPD) and assume a full TDR with both capture, and update stages?

CJ asked Adam to open the discussion with his 'additive' position and discussion proceeded from there:

- Adam responded that he doesn't see any issues with this approach. If a tool doesn't see the descriptors it will throw errors and the vendor ecosystem of tools will do the right thing.
- CJ commented that some groups would not like to add all the descriptors.
- Carol mentioned that most TDRs don't have update stages so she wouldn't want to have to document this all the time.
- CJ commented that FPGA vendors also don't like to document everything.

- CJ compared this to CLAMP_ HOLD which makes any exceptions subtractive to force designers to consciously think about this, to exclude any special case pins for local user defined tests.
- CJ said that subtractive is the best way he can think of to describe that you don't have a full register, and related this to FPGA vendors.
- Adam said he wasn't sure about the last statement, and asked why if chip vendors don't use UPD stages, and don't want to describe their internal TDR cells as a rule, why is this incorrect?
- CJ commented that one could still write PDL with no harm, but adding the cell descriptions will make things better for PDL writers. They can optimize PDL code if they know the structure of each cell. This can be used to avoid race conditions for example. (See email to Carl this past week for an example).
- Adam still favored the additive strongly typed approach.
- Carol asked how are we supposed to write the BSDL? This presents new issues for chip suppliers. Is this for REGISTER_FIELDS only, or can it be applied to the entire register and exceptiosn be made?
- Carl responded that it has to apply only to REGISTER_FIELDS, we cannot assign to an entire register and then overwrite and mange the sequential tracking of things if the definition changes later on in a REGISTER_FIELD segment of a larger register.
- Carol responded to add that just because you don't use the exta UPD stage, doen't mean you have ripple. Other blocking techniques are common practice.
- Carl mentioned that is non-standard approach.
- CJ pointed out his example wasn't to address the ripple issue, but to optimize PDL for the number of scans to avoid race conditions.
- Josh introduced himself as an FPGA person, who writes BSDL, and TDRs, and Instruments and questioned whether it is best to be overly explicit or waste time in debug later. He doesn't have control over what the tool vendor is doing, but only wants to do this a few times in his life, so wants the instruments to plug and play and be reusable. Bottom line is how much time are we talking about to either add or subtract bits from the description?
- Josh also stated that he has standard TDR design and uses standard instruments in an effect to lock down the design. It is extremely important to understand his design.
- CJ concurred that TDRs must be predictable.
- Carol showed the 6 figures Adam presented this past week.
- Josh responded his is a full TDR and it is up to synthesis to determine whether or not all the features are used.
- CJ mentioned synthesis tools could flag when UPD stages are optimized out.
- Ted asked which synthesis tool writes BSDL? You can't assume what synthesis results look like.
- Carol added a final PD tool writes BSDL for her based on cells and placement.
- John added that if synthesis determines the UPD stage is not used and optimizes it away, why do we care. It won't affect anything. If you define a full function TDR and it is optimized by synthesis it shouldn't matter.

- CJ added that for used GUI applications, end users need to know if an cell is missing or not as this affects their debug thought process. They can get lead astray chasing down bits that don't have any significance to begin with.
- Carol and CJ polled the group and the consensus started leaning towards having a default of a 'full' TDR with pi capture, shift, upd, and po, and to then optionally allow the use of the NO_PI, NO_PO, NO_UPD descriptors.
- Ted asked about these 3 keywords, and thought Carl's prior definitions make more sense from a functional point of view. This was debated for awhile. Francisco didn't like the use of words like no_read, no_write, since this refers more to memory test.
- The WG still came back and kept the NO_PI, NO_PO, NO_UPD definitions but with the underscores removed (NOPI, NOPO, NOUPD).
- Francisco asked about resets for the TDRs, and commented that the 6 figures didn't show the reset options.
- CJ and Carl responded that the resets are handled in the fields descriptions although not currently shown in the latest drawings.
- No effort was going to be made to rework existing drawings for this.

Meeting adjourned: 10:00am MST

Action Items:

- CJ/Carl to continue REGISTER_FIELDS and ASSEMBLY definition.
- Carl will continue with REGISTER_ASSEMBLY and ASSOCIATIONS after that.
- New draft is on the IEEE Site for WG members to review (June17 draft)
- Next new draft will be a couple weeks away per the editor.
-

Next Friday Meeting:

- Next week Friday July 1, 2011