

Date – 06/17/2011

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,
Francisco Russi,
Ken Parker,
Peter Elias,
Josh Ferry

Excused:

Wim Driessen

Meeting called to order at 8:30 am MST

New Draft: [P1149 1 Draft 20110528.pdf \(_clean.pdf\)](#)

Agenda/Overview:

- CJ lead a working session on REGISTER_ASSEMBLY and REGISTER_FIELDS attributes

General items:

- Carl will update current draft of B.8.17 and send out a new copy by early next week based on emails this past week and feedback from Dave Dubberke.
- Carol will proceed with BSDL example focusing on the Serdes init_data register example we've been using to date.

Today's Minutes:

CJ lead continued discussions on REGISTER_FIELDS and started initial discussions of REGISTER_ASSEMBLY.

REGISTER_FIELDS discussion:

- CJ raised some questions about range specifiers, whether they should be in common set of parentheses (5,20,36), or each have its own set as currently defined. (5),(20),(36)

- It was agreed that the range should be a comma separated list contained in a single set of parentheses.
- Ken asked if we still have the array notation for hierarchy? CJ responded that he intends to but he was trying to avoid hierarchy for now.
- Discussion continued about package file segment lengths could be longer than the specified register above. It was agreed this should cause the tool to error out.
- It was discussed that the since the init-data register is a BSDL keyword, then any package file segment by that same name must be associated with the init-data register. However this is not a requirement for User TDRs.
- Carl recommended we wait for REGISTER_ASSEMBLY attribute before worrying too much about package file to REGISTER_FIELD type issues.
- Section B.8.18.2 – CJ raised the point that he doesn't like the hierarchical references of package files to register fields bit definitions. This seems like it is impractical, in that we have predefined package files that test engineers have to go and count bits in to map them to the register fields. Easy for a machine, not practical for humans. Seems like we are redefining the bits here. A tool could flatten this.
- Carl pointed out that when we defined this syntax we had not yet defined the REGISTER_ASSEMBLY attribute so this can be revisited now. We wanted the ability to force strongly typed cross checking to ensure correctness. Carl wants to see how the register assembly handles this before he casts his vote to change anything.
- CJ agreed with the strong type but wants to avoid busy work in practice.

REGISTER_ASSEMBLY discussion:

- CJ lead and introductory discussion on register assembly. This session worked out some of the issues related to managing the scope and long strings in path names:
- CJ asked if we need to add hierarchy all the time or only if something is ambiguous?
 - Ken recommended you could use a predefined prefix to shorten hierarchical references. Something along the lines of iScope, named 'USING which could be defined once before a set of common hierarchy and used in context of Use statements:
 - Examples
 - iScope Memd
 - i2: MBist
 - mytdr
 - "USING Memd" &
 - "(i2: MBist)," &
- Carl pointed out that if 'USING belongs to the mnemonics and the fields that makes sense.
- Carl also raised the question about what if part of the field is hierarchical and part is not. What do we do in that case?
- Carol asked if the 'USING comes after the flat part?
- Ken asked if 'USING was case sensitive?
- CJ pointed out that: USING Memd and USING MEMD were the same.

- CJ asked again if we are going to allow hierarchy in REGISTER_FIELDS?
- The group was polled and consensus was to use whatever is simplest.
- CJ will put together a future example which is more refined.
- CJ raised another question in context of Francisco's STIL examples that register path names get very long and was wondering if 'USING' can be reused in the context of shortening names for better readability?
- The group thought ALIAS or PREFIX were better keywords to use, since USING should be kept in context of Use statement.
- Carl asked in REGISTER_ASSEMBLY can do away with REGISTER_FIELDS?
 - CJ replied no, because we still need to define all the bits in the fields. The assembly here does not deal with individual bits.
 - Carl made note then that REGISTER_ASSEMBLY requires all bits to be contiguous, no swizzling allowed here.
 - Bit swizzling is defined in REGISTER_FIELDS as needed.
 - CJ pointed out to predefine bits in REGISTER_FIELDS, and then assemble in REGISTER_ASSEMBLY.
- Ken asked for an example of PREFIX usage.
 - CJ showed the STIL file with many redundant and long prefix paths up to registers.
- Carl said we'll update the examples
- CJ said we are in pretty good shape, but he will review and continue the work.
 - PREFIX does not go as far as ALIAS does but has worked for years in its context of making strings for readable.
- Carol pointed out the referring back to the prior PREFIX definition could be tedious in a large BSDL.
- CJ acknowledged this and said we should start with PREFIX and see if it needs to be expanded to ALIAS later on.

Meeting adjourned: 10:00am MST

Action Items:

- CJ to continue REGISTER_ASSEMBLY and REGISTER_FIELDS work.
- Carol to update the BSDL example.

Next Friday Meeting:

- Next week Friday June 24, 2011