Date – 07/22/2011

Minutes of the IEEE-1149.1 Working Group Friday meeting

Attendees:
Brian Turmelle,
Carol Pyron,
Roland Latvala,
Carl Barnhart,
CJ Clark,
Dave Dubberke,
Ken Parker
Wim Driessen
Ted Eaton
Heiko Ehrenberg
John Braden

Excused:
Adam Ley

Meeting called to order at 8:30 am MST

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

Agenda/Overview:
REGISTER_FIELDS and REGISTER_ASSEMBLY (continued discussions)

Minutes:

Today Carl reviewed additional examples of register fields and assembly. Carl also proposed a new register assembly format to simplify BSDL syntax and coding.

Example: Heir Init_data example.rtf

Ken asked if in register or segment names in package file could cause conflicts.
Carl said based on hierarchical references, the answer was NO.

Discussion of the use of the asterisk *

- Carl - It says that something must be defined before you run your test.
- Ken – That would get filled in using the side file?
- CJ – Yes, needs to be filled in before they proceed.
- Carol – For port associations ports can be assigned to these individual bits
- Ken – Verification is important
- CJ – The '*' forces the test engineer to not proceed until something gets defined

Carl’s example review continued:

- Looking at the next field ‘configuration’ it is a mix of fixed defaults, and other bits which gets defined with a SAFE value
- Only REGISTER_FIELDS defines the length of init_data, init_status, reset_select registers
- Register Assembly attribute:
- Reg or Seg name tokens allowed
- No length allowed
- TDI to TDO order mandatory left to right in ‘to and downto’ and also top to bottom within register_assembly
- Carol – Could add some reserved ‘bread crumb’ bits with key1 key0 to make it more descriptive
- CJ – When we did the 0-17 it looked odd.
- Carl – Should leave to mix it up a little
- Ken – Good to show both to and downto
- Carl - Bit and length checking in register_fields only. Register_assembly simply puts the fields and segments together, without any lengths defined

Carol asked how to define a hierarchical reference into serdes channel protocol value
CJ – Gave a verbal response
Carl – Showed same based on the mnemonics
CJ – I see an error, you might want to have more than one mnemonic group name
We had protocol.safe and tx_swing.safe
We now have SAFE.protocol and SAFE.tx_swing, to tell parser to expect an identifier
CJ – Maybe an example with a 2nd channel called out may be a good idea to help the reader.
Carl – I’ll add to the list
Carol – OK to deviate from my original example
Ken – The question about array of 18 serdes channels. Can I go 0-8, skip 9, the 10-17?
Carl – Yes
CJ – Yes but adds confusion
Carl – I’ll add a semantic check to not allow skipping an indices
CJ – I’ll work with the fields concept to avoid calling out the field multiple times.

Proposal: To define field segments within a register assembly to simplify coding:

Carl showed his new proposal of a modified REGISTER_ASSEMBLY format, that allows a mix of flat and hierarchical notations which can simplify syntax of register fields and make register assembly easier to code for a ‘power user’. Discussion point of this new register assembly proposal:

- Ken – Question - Can reserved1 be reused again?
- Carl – No in register_fields this is a unique field segment and named uniquely.
- Ken – Can we still determine the length of init-data?
- Carl – We are moving away from bit descriptions of registers to mnemonic descriptions.

- CJ- If register length is important it can be run through a tool to find the exact length. For example a register is 16,538 bits. Who really needs to know this?
- Carl – Another example of a 800 bit register and hex characters of the bits cared about took a few days of work. With mnemonics we don't have to worry about hex values again and low level bit positions. Going forward I see register assembly being dominant.
- CJ – Carl this may confuse folks.
- John – When chains have extra or missing bits the length is important. In the lab we still need to count bits.
- CJ – If you have bread crumb bits sprinkled around it will help debug. The question is really, do we want to mix flat descriptions with hierarchical descriptions? Carl's allows power users to simplify things, the original is better for the new user.
- John – OK with power user
- Heiko – Is there anything that keeps us from allowing both?
- Carl – No you could do both.
- Roland – Agreed this could confuse new users. To allow both formats also means more work for the tools. For debug, I like to know register lengths up front too.
- CJ – Yes more work for tools, but tools can report the TDR lengths.
- CJ - Looks like most members like the new version since it's a super set.
- Carl – I'll publish this new register assembly definition into the next version of the doc.

Meeting adjourned: 10:05am MST

Action Items:

- Carl will publish new Std draft with updates highlighted.

Next Friday Meeting:

- Next week Friday July 29, 2011