Main principles and needs for the SCE-MI standard evolution

Joseph Bulone, ST MIcroelectronics

1.0 Need of the ability to compare with a simulated reference

This need is to be able to compare, if necessary, in a cycle base manner i.e. more or less in a systematic way the result of a co-emulation relatively to a simulation or a co-simulation seen as references. This is to verify a priori the good implementation of the DUV onto the emulator. Cycle based co-emulation or time congruency are enough to cover this goal. The user needs this time congruency at least when comparing with a cycle based model, but once he considers he has enough validated his co-emulation platform, he may decide not to use this option in order to have better speed performances: his final goal is to verify an IP or a SoC, not to verify the correct mapping onto the emulator.

2.0 Need of repeatable sessions

To allow the exploitation of several debug sessions, it is necessary to have a transactional mode[1] which ensure repeatable results. That means modes that ensure a deterministic behaviour of the infrastructure. By this mean, if the testbench, the DUV and the transactors have also deterministic behaviours then the global system will have a deterministic behaviour. This is essential to simplify the debug task. Thus, each implementation should provide such a mode, and the user should have the option to use it or not.

3.0 Need of functional congruency

The exact behaviour of two sessions using different emulators does not need to be the same. The determinism of the infrastructure may be emulator dependent. What will happen is that the interoperability will depend onto how the user manages the global synchronization (with the testbench, the transactor and the DUV) and also onto the infrastructure implementation. What is needed between two different emulator is a functional congruency not a time congruency.

1. Some design may not be very well suited for transactional modes, i.e. the transactional interface will not provide huge speed improvement because of too high interactivity at the interface.

4.0 Some comments onto the “agreed” principles[1]

4.1 Variable Length Messages

The justification of VLM is to allow speed improvements or to simplify the software view. The speed improvement is obtained by removing call-backs at the emission or reception side. One way to do that is to perform transactions using several vector transfers. That consists in an extension of the rendez-vous mechanism still with a word basis for the TransmitReady and ReceiveReady signals, but now with a burst basis for the callbacks. A more generic solution with a dynamical message length would provide also the burst functionality but it is necessary to avoid adding a specific header. In fact, this header would complexify the FSM managing the messages when dealing with fixed-length messages: only the payload bits should be put into the messages. In this way transformation of SCE-MI 1.0 transactors to a more efficient burst/variable length mode transactor would be greatly simplified.

As already mentioned in previous meetings, each callback registration could be optional in case of an application which does not need it. It is also essential to unlink the notion of uclock and of VLM.

4.2 Zero Time Operation

The zero time operation is a required feature. The issue is in its implementation in order to have the lowest impact onto the overall system speed. The use of an ‘over-clocking’ mechanism is not a general enough solution as in this case, for many emulators, the fastest clock of the system and its ratio with the DUV clocks will directly determine the overall speed, while with SCE-MI 1.0 the clock ratios and but also the frequency and the duration (in term of the number of cycles of the uclock) of the transactions will determine the overall performances. Depending onto the design, the overall speed of these two implementations of ZTO may differ from several order of magnitude[2]. This is an essential reason to impose to each new version of SCE-MI to provide the current uclock/cclock mechanism. The user would have the option to use it or not if other mechanism are proposed. A new version of SCE-MI is to be an enhancement on all the features it adds or replaces or removes.

1. Another document will comment more precisely Matt’s document onto ”Ënhancements to SCE-MI 1.0, First Principles”
2. More explanations with chronograms will be provided ASAP