# Minutes of the Accellera C working Group
# November 7th 2000

Attendees:
Dan Gajski UCI
David Springer CynApps
John Sanguinetti CynApps
Andrew Guyler Mentor Graphics
Brian Bailey Mentor Graphics
Kevin Kranen Synopsys – phone

Firstly, an apology to anyone who tried to call in on the published teleconference number. I am still not quite sure what happened here, but we had a bad number. In the future, I will try the number before the meeting starts to ensure that it is correct.
Kevin did manage to contact us and we made a direct connect to him.

The objective of the meeting to was to identify what elements were missing from Dan's current document, to go over any areas of comments (the only ones submitted were from David and Frederic) and to discuss the open issues such that the document could be completed.

## Missing:

Special cases – resets especially asynchronous. Two main ways of accomplishing this were suggested. The first is to use "if reset" statements before each assignment block. While tedious this is the easy method. The second is to use a second parallel process that has a higher pririty on the asignment. However, both of these define syntax. What is required is a general mechanism to allow one assignemnt to have priority over another. This may be useful later in protocol work as well.
** Dan will work on this

Multicycle operations:
Pipelining is the same as the insertion of registers. But function pipelining and multicycle are distinct. With a function pipeline it is possible to know that the inputs are valid in the first cycle, but for multicycle this assumption cannot be made. There are many syntax or language issues here in order to make the definition and design of pipelines easy for the user. Constraints may exist in the pipeline such as input holds etc.

Control registers do not appear until level 5. This creates a problem and adds a delay into the data pipeline. This may however allow the clock time to be reduced. It was not clear if any additional semantics need to be added to fully support this.

Control and Data
When we have a variable that is used in both the control and datapath such as loading a new value into a PC. This action affects both the datapath (loading the register) and calculating the next address in the FSM. This means that the definition of Control and data ports is artificial and should be removed. However, it is generally agreed that a tools life is made a lot easier by knowing this information. For example, how does a tool know where an operator should go? Where do we separate semantics from tool directives from implementation issues etc. Dan admits that there may be some problems here. He is trying to simplify whereas Dave thinks it should be a problem left to the tool.
Agreed that this perhaps should be an attribute on inputs/outputs and thus not mandatory but would make life easier if it can be known.

Remove the restriction on inputs to the controller being Boolean. This was agreed.

There was a long discussion regarding the scope of the document as to the safe zone versus the dirty reality of the design community. The current document discusses only the safe zone but does not preclude the

other cases. It was felt that many of these were design styles that should be discouraged and not explicit support should probably be added for them, but to not close the door on some of the more common ones.

One particular discussion centered around one objective of the original bubble diagrams and involves the locating of dependance of expressions when they cross multiple FSMD's. This was one area that was felt to be necessary to ensure that support was there
** Dan to add alu to alu communications between 2 FSMD without clocks. In a more general case when do things get evaluated when the relationship between clocks of 2 FSMD's is not know or if one of them doesn't even have a clock.

Variables can be of any type, with the one restriction that it can be loaded in a single cycle. However, there is an implication that there is a bit representation for it so that mapping into registers can be performed. A brief discussion was held regarding the adoption of VSI datatypes. Does RTL need all of them or should we restrict to the subset necessary for this level of abstraction. At the lower levels (5 and 6) we really only have bit and bit-vector left.

Agreed that the level numbers should be removed in favor of Dan's new naming scheme. Levels 2,3 and 4 get renamed to mapped RTL. Still require a good name for level 5

Discussion regarding the role of the new IEEE group. Our goal is to get this finished as soon as possible and to get it approved by IEEE. Do not want to allow a competitive effort to divert attention. A meeting will be held this evening to discuss this with Rajesh Gupta. In general the group felt that we should get Rajesh up to speed with what we have been doing and suggest that he could help lead this work through the IEEE

We then came back to one of the thorny discussions about our role in terms of producing a sample implementation. Is the existing Semantics Document something that can be standardized or do we have to add a sample syntax? There is precedence that IEEE has standardized things that are not implementations such as MPEG so we should be able to do this. We are defining a basic set of the classes required.
For interoperability we do not need syntax. We should possibly define what the set of visible classes are, but not the structure of them.

The meeting adjourned at about 12:30.
The date of the next meeting is to be set