Parameter passing to IBIS-AMS models

IBIS Advanced Technology Modeling Teleconference

November 28, 2006

Arpad Muranyi
Intel Corporation
Signal Integrity Engineering

According to IBIS v4.2

```
|               Parameters:
|
|               Lists names of parameters that can be passed into an external
|               model file.  Each Parameters assignment must match a name or
|               keyword in the external file or language.  The list of
|               Parameters may span several lines by using the word Parameters
|               at the start of each line.  The Parameters subparameter is
|               optional, and the external model must operate with default
|               settings without any Parameters assignments.
|
|               Parameter passing is not supported in SPICE.  VHDL-AMS and
|               VHDL-A(MS) parameters are supported using "generic" names, and
|               Verilog-AMS and Verilog-A(MS) parameters are supported using
|               "parameter" names.
```

What “assignment” is the IBIS specification talking about?

- none of the examples in the specification make any numerical assignments
- there is no more description detail on how that would be done if allowed
- the word “assignment” may be used improperly in these sentences
- we may need to write a BIRD just to clarify this, if nothing else…

An example from the IBIS spec

```
[External Model]
Language VHDL-AMS
|
| Corner corner_name file_name       circuit_name entity(architecture)
Corner    Typ          buffer_typ.vhd  buffer(buffer_io_typ)
Corner    Min          buffer_min.vhd  buffer(buffer_io_min)
Corner    Max          buffer_max.vhd  buffer(buffer_io_max)
|
| Parameters List of parameters
Parameters delay rate
Parameters preemphasis
|
| Ports List of port names (in same order as in VHDL-AMS)
Ports A_signal A_puref A_pdref A_pcref A_gcref
Ports D_drive D_enable D_receive
|
[End External Model]
```

Note that there in only a list of parameter names in the examples. None of the examples illustrate any assignments.

The intention may have been…

… to provide information to the simulation tools about what the parameters are inside the *-AMS models, so that the tools can make value assignments using GUI and/or other means if needed.

- the IBIS specification spells out that the *-AMS model must work without these assignments, so this seems to be intended to be an overriding mechanism for “what if” type analysis, but not the “usual way” to instantiate and operate *-AMS models.

```
|               Parameters:
|
|               Lists names of parameters that can be passed into an external
|               model file.  Each Parameters assignment must match a name or
|               keyword in the external file or language.  The list of
|               Parameters may span several lines by using the word Parameters
|               at the start of each line.  The Parameters subparameter is
|               optional, and the external model must operate with default
|               settings without any Parameters assignments.
```

The problem statement

- **Imagine an *-AMS model with a significant complexity**
  - multi-tap SERDES driver
  - receiver with DFE
- **Assume that the algorithms of this model can be used for multiple instances within an IBIS file**
- **The only thing that may be different between the instances of this *-AMS model is the actual values of its parameter(s)**
  - according to the current IBIS specification, we would have to duplicate the entire *-AMS model for each of its instances containing the corresponding parameter values, or
  - the simulator would have to pass down the instance specific parameters into the *-AMS model somehow
- **Where would the tool obtain the parameter values from?**
  - based on the above scheme, the IBIS file cannot contain this information
  - should the data come from yet another parameter data file?

A potential solution

- **Using the previous example,**
  - the IBIS file could make multiple calls to the same *-AMS model (as if it was a parameterized template)
  - the IBIS file could contain the data for all (or some) of its parameters
  - the data sets may be different for each instance
- **This scheme would be more efficient**
  - the algorithm in the *-AMS model wouldn't have to be duplicated for each instance
- **This scheme would also be more compact**
  - the IBIS file, which is traditionally a "data container" would retain its functionality as such
  - there would be no need for yet another parameter data file

Complications

- **The *-AMS languages have numerous variable types**
  - integers, reals, strings, scalars, vectors, user defined types…
- **How should we deal with all these in the IBIS file?**
  - should the parameter passing syntax spell out the type and value, as if it was a declaration statement, or
  - is it enough to write: `parameter_name = value` regardless of what the type of `value` is (observing proper spelling rules, such as `"a string"` for strings, `1` for integer, `1.0` for real, etc…)?
- **Are there any other concerns, suggestions?**
  - allow mixed usage of assignments and no assignments
  - allow assignment of typ/min/max values since currently there is one parameter list per possibly multiple (typ/min/max) instantiation
  - allow the definition of parameter ranges, instead of just single values
- **Questions/comments? Time to start our discussion…**