Hi, All -

Stu informs me that at Thursday's meeting, a new set of deadlines were set and that I am required to make the .* & .name proposal by tomorrow (Monday - three days notice). I sent out the original proposal on December 13th, apparently ignored. At a recent conference call, Anders asked for a friendly amendment to include the .name syntax.

I cannot be on the call this Monday (I fly out of town again tonight). I will try to rearrange my schedule next week to be on that call. I hope my proposals receive reasonable consideration in my absence or that final voting can be done when I am present to defend my proposals. Scheduling meetings on very short notice and issuing proposal deadline ultimatums for topics I have previously spent many engineer-hours developing and had previously sent (in December - both FSMs and interfaces), leaves me less than pleased.

I have recently spent days putting together another FSM example with accompanying FSM coding styles, thinking this was a high priority (the examples were sent out weeks ago). Apparently, I should have spent the time first on the implicit ports proposals.

Below are some of my notes and proposed rules for implicit ports. I am also attaching a re-send (with some updates, including Verilog-2001 modified BNF) showing some of the changes I propose. I will make further wording proposals and add .name examples if my work is not shot down on Monday; otherwise, what's the point?

Regards - Cliff

Per Anders' recommendation on a previous conference call, I suggest two styles of implicit port connections:

(1) .* implicit port connections
(2) .name port connections

The second, and what Anders was suggesting, is instead of repeating names twice (D flip-flop instantiation example):

dff d1 (.q(q), .d(d), .clk(clk), .rst_n(rst_n)); // Verilog-1995/2001

dff d1 (.q, .d, .clk, .rst_n); // Anders' idea

If a signal name mismatch occurs:

dff d1 (.q(out1), .d, .clk, .rst_n); // q-output attached to out1

To me, this is the cleanest way to show implicit connections without having to type the name twice.

My notes, including proposed restrictions, from a previous email (modified to include the .name syntax).

Upon approval of this concept, I will take the extra time to propose wording and examples to be added to section 13.

Pending approval, SystemVerilog will permit an enhanced and abbreviated method to instantiate modules using the implicit port connection token `.*`

In SystemVerilog, if the net connected to a port of an instantiated model matches the name of the port and the size of the port on the instantiated model, an implicit port connection can be made.

Many design teams use the proven methodology of using the same name for a top-level net and all of the ports that connect to that net. These same design teams generally use the safe method of making named port connections when instantiating the top-level blocks.

Engineers who have coded large top-level ASIC designs, have experienced the pain of connecting hundreds and thousands of named ports to tens and hundreds of top-level modules.

SystemVerilog introduces the capability to instantiate modules with highly abbreviated and efficient implicit port connections. Implicit port connections are intended to facilitate the process of instantiating large sub-blocks into upper-level modules without having to type multiple lines of named port connections where the sub-blocks are instantiated.

Implicit port connections reduce the verbose nature of most higher-level modules by limiting the number of named ports that actually have to be listed when a module is instantiated. At the same time, since only those nets or busses that do not match must be listed in the module instantiation, they are emphasized and not hidden in a sea of named port connections.

With a careful naming convention, instantiating large logic blocks into a higher level module can now be greatly facilitated by using SystemVerilog implicit port connections.

When using the `.*` implicit port connection token, any sub-block port that does not match in size or name to the module net or bus connected to the port, must be connected using a named-port connection. Mixing implicit port connections (`.*` connections) with positional port connections is not permitted.

Note: For the purposes of this discussion, the term “implicit port declarations” may or may not also include some named port connections.

PROPOSAL:

`.*` and `.name` implicit port connections must follow these rules:

- `.*` implicit ports shall not be used in an instantiated sub-block with positional ports.
- `.name` implicit ports shall not be used in an instantiated sub-block with positional ports.
- `.*` implicit ports may be used in an instantiated sub-block with named ports.
- `.name` implicit ports may be used in an instantiated sub-block with named ports.
- Modules instantiated with `.name` implicit ports shall not be instantiated in the same module with modules instantiated with `.*` implicit ports. This restriction insures that a design team can enforce a policy that all modules instantiated within the same scope must use the `.name` implicit syntax. Since an IP vendor may choose a different implicit port instantiation style, this restriction does not extend to all scopes in the same design. One scope may choose to use the `.name` implicit style while a sub-scoped module may choose to use the `.*` implicit style.
- It is permitted to have sub-block instantiations with positional ports and sub-block instantiations with named ports and sub-block instantiations with `.*` implicit ports all instantiated in the same upper-level module.
- It is permitted to have sub-block instantiations with positional ports and sub-block instantiations with named ports and sub-block instantiations with `.name` implicit ports all instantiated in the same upper-level module.

- If `.*` implicit port connections are used to instantiate a sub-block, the `.*` token must be placed first in the instantiated port list, before other named ports, if any, are listed.
- For a `.*` or a `.name` implicit connection to be made, the port name on an instantiated sub-block must match the net or bus name of the connecting module.
- For a `.*` or a `.name` implicit connection to be made, the port size on an instantiated sub-block must match the net or bus size of the connecting module.
- Any individual port in an implicitly instantiated module that does not match both size and name of the net or bus of the upper-level module, must be instantiated by name.
- If a port on a `.*` or a `.name` implicitly instantiated sub-block is unconnected in the upper-level module, the port shall be explicitly listed as a named port with empty parentheses, showing there is no connection to the port.
- All nets or busses in the upper-level module that connect to `.*` or `.name` implicit ports must either be explicitly declared as a scalar-net, vector-net, or as a port on the upper-level module. (NOTE: at HDLCON, one user did not want to declare the upper-level nets as required by this bullet).

Arguments in favor of the .* style:

Debugging the .* notation should not be a problem:
The Verilog compiler already had to make the connections so the port names should exist in the data base.

ASIC designers hate to do top-level designs because it is generally just a bunch of verbose connections.

The .* notation removes all of the unnecessary verbosity and only lists ports where there are naming exceptions, size mismatches, and unconnected ports. It is much easier to scan a top-level module with 10's to 100's of instantiated modules and 1,000's of ports, knowing that everything is implicitly connected and just the exceptions are obvious and visible.

If you are doing a simple block-level testbench, you are going to wire up the connections with the same names. The .* notation makes it easy to quick construct a block-level testbench.

Intel's IHDL has used this style for years and Intel wants this capability (i.e. - there is a history of success using this connection style). We asked Intel a bunch of questions concerning debugging problems that they had had with this style. They claim the problems are almost non-existent. I believe it!

Arguments in favor of the .name connections:

For engineers like Anders who want to see all of the signals in the port list, this notation satisfies his requirement.

The notation is a natural abbreviation style of existing Verilog coding (using the .name notation without having to repeat the signal name inside of parentheses)

One important rule needs to be added: you cannot mix .* and .name ports in the same instantiation.

You can mix .* and .name(signal) -OR- .name and .name(signal) connections in the same instantiation. The .name(signal) connections are required for size-mismatch, name-mismatch and unconnected ports.

There is way too much emphasis and importance placed on declarations and port-name visibility. For most designers of very large ASICs, the top-level model is a painful and necessary evil to satisfy the requirements of a simulator. Please keep the .* notation!

Regards - Cliff