Summary of Cliff's votes:

SVDB 907 Yes
SVDB 1134 Yes
SVDB 1294 Yes
SVDB 1348 Yes
SVDB 1464 No
SVDB 1468 Minor No
SVDB 1588 No
SVDB 1619 No
SVDB 1792 Yes
SVDB 1940 Yes
SVDB 2024 Yes
SVDB 2056 Yes

SVDB 907 Yes
SVDB 1134 Yes
SVDB 1294 Yes
SVDB 1348 Yes

SVDB 1464 No

As long as we are making corrections to the example, the "assumed to be instantiated" module is also wrong because the matching internal bus declarations are also required.

I would vote yes if the proposed change is modified as follows:

TO
The following example demonstrates the usage of extern module declarations.

```
extern module m (a,b,c,d);
extern module a #(parameter size= 8, parameter type TP = logic [7:0])
(input [size:0] a, output TP b);

module top ();
  wire  [8:0] a;
  logic [7:0] b;
  wire        c, d;

  m mm (.*);
  a aa (.*);
endmodule
```

Modules m and a are then assumed to be instantiated as follows:

```
module top ();
  wire  [8:0] a;
  logic [7:0] b;

  m mm (a,b,c,d);
  a aa (a,b);
endmodule
```

SVDB 1468 Minor No
I would vote yes with the friendly amendment of replacing the personal pronoun "they" with "tools":

"... , tools may check and warn if the behavior ..."

SVDB 1588 No
I believe these directives are preceded and followed by double-underscores and that that makes them unique are SystemVerilog compiler directives. I do not object to the syntax, but since there is no visible break in the double-underscore, I think it must be noted at least once in the description of the compiler directives (even if other languages do or do not use this same syntax - I am assuming this comes from C-programming although that detail is not important). Then I would vote yes.

SVDB 1619 No
I am not sure if this is really a good idea. I understand the desire but there are some real potential problems with this proposed enhancement. I would want to hear Synopsys representatives verbally commit to support this capability in synthesis, which has generally not been the case (initial assignments in declarations). If Synopsys supports this in synthesis, then other tools could be shamed into supporting this feature. There are other synthesis tools, like Synplicity, that ignore initial blocks and hence initial assignments.

Here is the potential problem. If you declare the rst_n input of a module to have a default value of 1'b1 (de-assert reset), and if synthesis tools ignore the declaration, we now a potentially very difficult bug to find. During pre-synthesis simulations, reset will be de-asserted. During post-synthesis simulations, the reset input will float and lots of other logic will propagate X's in the gate-level simulation. It might not be obvious that you have a dangling reset input. Not only that, but we have now introduced another potential problem that might not be discovered until very late (after synthesis, not during simulation or compilation). I see some real dangers in this enhancement.

If the above issues are not considered to be significant (because synthesis would honor the default-inputs), there is still a problem with the alu_accum3 (.name instantiation of the xtend module), because the .rst port error has nothing to do with declarations or defaults. The bug is that there is no rst port on the xtend module, not anything to do with the default declaration. The example should be modified.

SVDB 1792 Yes
SVDB 1940 Yes

SVDB 2024 Yes
SVDB 2056 Yes