Verilog Macros & Semicolons

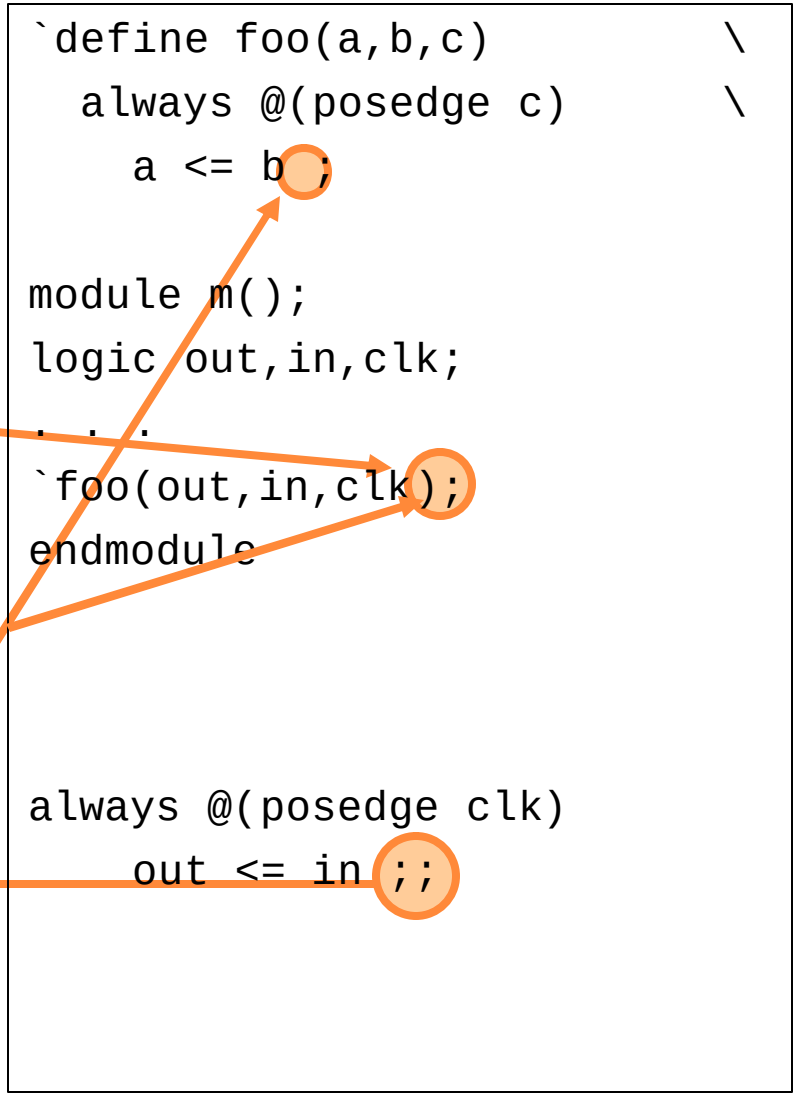
Matt Maidment

Overview

- Description
- Workaround
- BNF
- Questions

Description

- When making extensive use of macros one would like to use them as function calls
  - In Expressions or
  - Statements
- Want natural termination with a semicolon
- Between definition and usage
  - semicolon use excessive
  - null statements introduced



Description

- Verilog further complicates matters with begin-end constructs
- When using this macro **CANNOT** terminate with a semicolon
- Many Verilog Constructs Complicate Macros
  - generate/endgenerate
  - fork/join
  - begin/end
  - module/endmodule
  - function/endfuntion
  - task/endtask

```
`define foo(a,b,c)          \
  always @(posedge c) begin\
    a <= b;                 \
  end
```

Description

- This is not a new problem, well-known in C
  - See ‘info cpp’ *Swallowing the Semicolon*
  - Suggestion is to wrap macro in

```
do {...} while (0);
```

- Is there a well-known workaround in Verilog or are macros too new?

Workaround

- For any Macro that ends with an join/end* terminate macro with

  `initial if(0)`

- It's not pretty but it works for macros intended for module scope

- Similar workaround could be extended to macros used in always/initial blocks
  - Note true branch for safety just in case

```
`define foo(a,b,c)              \
  always @(posedge c) begin     \
    a <= b;                     \
  end initial if(0)

`foo(a,b,clk); //it works!



`define foobar(d)               \
  begin                         \
    d = d + 1;                  \
  end if (1)


always_comb begin
`foo(d)     //OOPS NO ;
a = b;      //THIS COMPILES!
```

BNF

- Verilog does not have a universal notion of statements
  - Modules contain instances, tf definitions and constructs (always/initial)
  - “Statements” live in constructs and tf definitions
- Null statements are defined but only in special cases
- Null module item does not exist

Questions

- Was the lack of null ever considered?
  - Why not allow `end;` or `;;`
- Could null be added?
- What is committee's take on this?