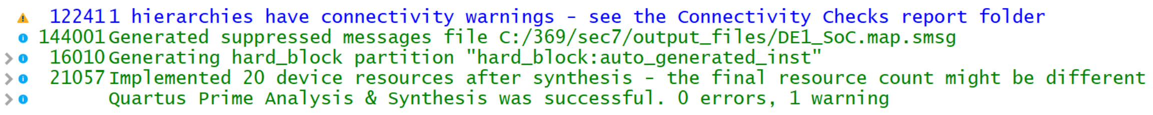
Exercise 1 – Interpreting Messages

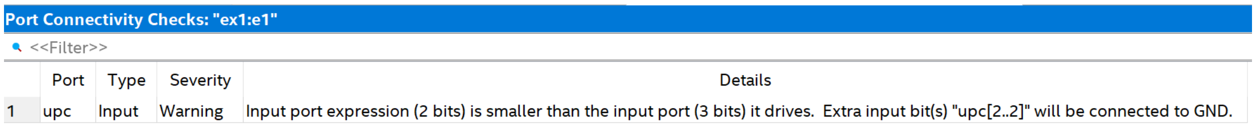
Given the following modules and error messages, identify and fix the bug:

```
module ex1 (output logic dout, input logic [2:0] upc);
  assign dout = upc[1] & upc[0] ^ upc[2]};
endmodule  // ex1
```

```
module DE1_SoC (input logic [9:0] SW, output logic [9:0] LEDR);
  ex1 e1 (.dout(LEDR[0]), .upc(SW[1:0]));
endmodule  // DE1_SoC
```



```
 122411 hierarchies have connectivity warnings - see the Connectivity Checks report folder
 144001 Generated suppressed messages file C:/369/sec7/output_files/DE1_SoC.map.smsg
 16010 Generating hard_block partition "hard_block:auto_generated_inst"
 21057 Implemented 20 device resources after synthesis - the final resource count might be different
        Quartus Prime Analysis & Synthesis was successful. 0 errors, 1 warning
```



Port Connectivity Checks: "ex1:e1"

<<Filter>>

	Port	Type	Severity	Details
1	upc	Input	Warning	Input port expression (2 bits) is smaller than the input port (3 bits) it drives. Extra input bit(s) "upc[2..2]" will be connected to GND.

Exercise 2 – Port Connection Analysis

Given the following modules, analyze the ports instantiations in `ex2` *independently*.

```
module ports (input  logic a,
              input  logic [1:0] b,
              output logic c);
endmodule  // ports
```

```
module ex2 (input  logic a, b, d,
            input  logic [1:0] e,
            output logic c);

  ports option1 (.a, .c);
  ports option2 (.a, .b(e), .c);
  ports option3 (.*);
  ports option4 (.a, .b(e), .c, .d);
  ports option5 (.a, .b(e), .c(d));

endmodule  // ex2
```

- Is there an issue?
- If so, what is it? Do you think it will produce a warning or an error?

Exercise 3 – Psychic Tester Modifications

In Section 6, we worked on a design of the `psychic tester`, where the user needs to correctly guess 8 consecutive 4-bit patterns to be declared a psychic.

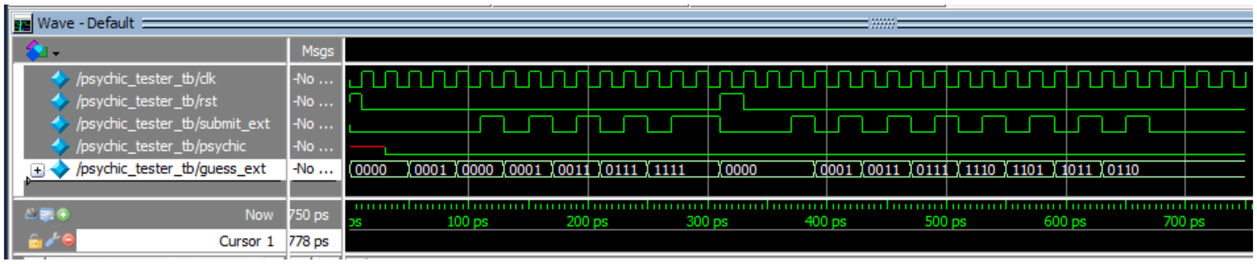
```
module psychic_tester (clk, rst, guess_ext, submit_ext, psychic);
  input  logic       clk, rst, submit_ext;
  input  logic [3:0] guess_ext;
  output logic       psychic;

  logic [3:0] pattern, guess;
  logic correct, next, submit;

  genPatt    pat (.clk, .rst, .pattern, .next);
  userIn     inp (.clk, .rst,
                  .guess_ext, .submit_ext, .guess, .submit);
  checkGuess chk (.pattern, .guess, .correct);
  countRight cnt (.clk, .rst, .correct, .submit, .next, .psychic);
endmodule  // psychic_tester
```

Say we want to modify our design so the next signal comes directly from `user_input`.

One way someone might modify `psychic_tester` could be by removing the **`next`** signal from **`countRight`**, and then assigning **`next`** to **`submit_ext`**. This then gives us:

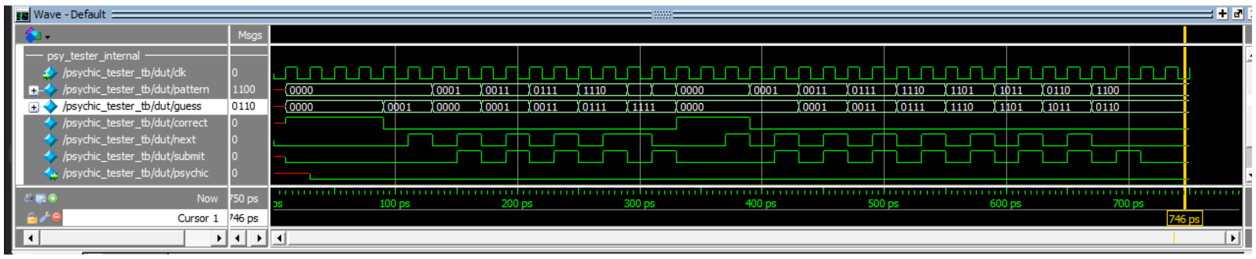
```
module psychic_tester (clk, rst, guess_ext, submit_ext, psychic);
  input  logic       clk, rst, submit_ext;
  input  logic [3:0] guess_ext;
  output logic       psychic;

  logic [3:0] pattern, guess;
  logic correct, next, submit;
  assign next = submit_ext;
  genPatt    pat (.clk, .rst, .pattern, .next);
  userIn     inp (.clk, .rst,
                  .guess_ext, .submit_ext, .guess, .submit);
  checkGuess chk (.pattern, .guess, .correct);
  countRight cnt (.clk, .rst, .correct, .submit, .psychic);
endmodule  // psychic_tester
```

Because we didn't modify how our input and outputs work and how the design should behave, we should be able to use the same testbench! Let's take a look at ModelSim and see if things are behaving the way we expect...


Wave - Default
Msgs
/psychic_tester_tb/clk -No ...
/psychic_tester_tb/rst -No ...
/psychic_tester_tb/submit_ext -No ...
/psychic_tester_tb/psychic -No ...
/psychic_tester_tb/guess_ext -No ...
0000 0001 0000 0001 0011 0111 1111 0000 0001 0011 0111 1110 1101 1011 0110
Now 750 ps
Cursor 1 778 ps
ps 100 ps 200 ps 300 ps 400 ps 500 ps 600 ps 700 ps

We don't get a high signal for psychic anymore! What could be causing this issue? What would you investigate?

Let's analyze our internal signals.


Wave - Default
Msgs
psy_tester_internal
/psychic_tester_tb/dut/clk 0
/psychic_tester_tb/dut/pattern 1100
/psychic_tester_tb/dut/guess 0110
/psychic_tester_tb/dut/correct 0
/psychic_tester_tb/dut/next 0
/psychic_tester_tb/dut/submit 0
/psychic_tester_tb/dut/psychic 0
0000 0001 0011 0111 1110 0000 0001 0011 0111 1110 1101 1011 0110 1100
0000 0001 0000 0001 0011 0111 1111 0000 0001 0011 0111 1110 1101 1011 0110
Now 750 ps
Cursor 1 746 ps
ps 100 ps 200 ps 300 ps 400 ps 500 ps 600 ps 700 ps
746 ps

What incorrect behavior do you notice? Can you go back to the code and see if you can spot the bug?