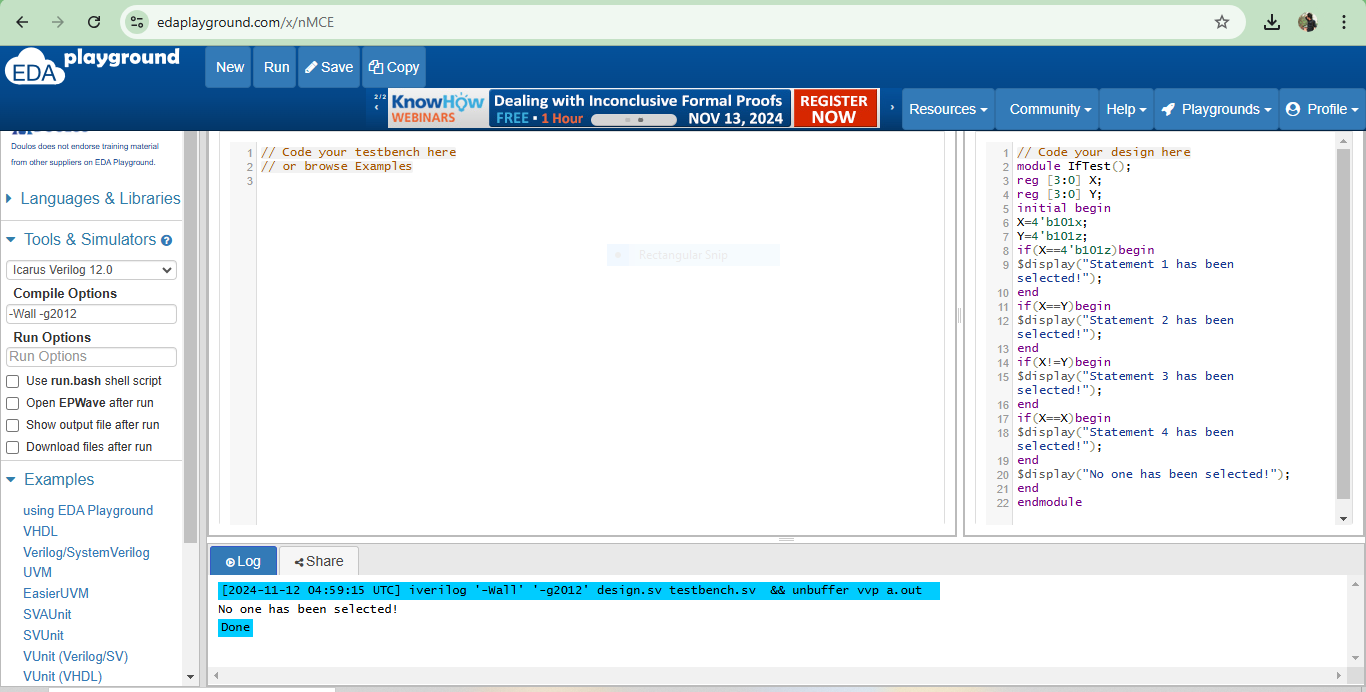
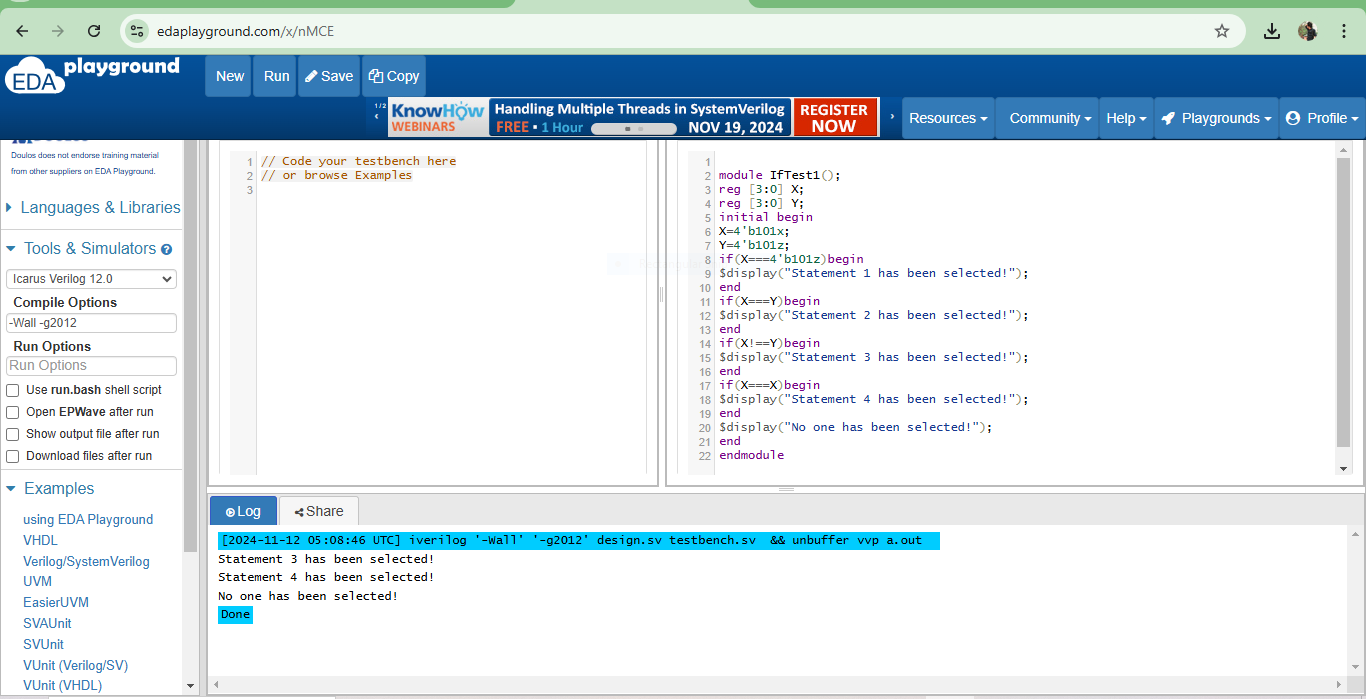
I. "if" Conditional statements



The result is that none of the if conditions pass because of the presence of x and z values in X and Y, leading to only the final message, No one has been selected!, being printed.

- if(a===yyyy): no matter what a’s value is, even x or z , if LHS exactly matches  
RHS, (a===yyyy) = 1.



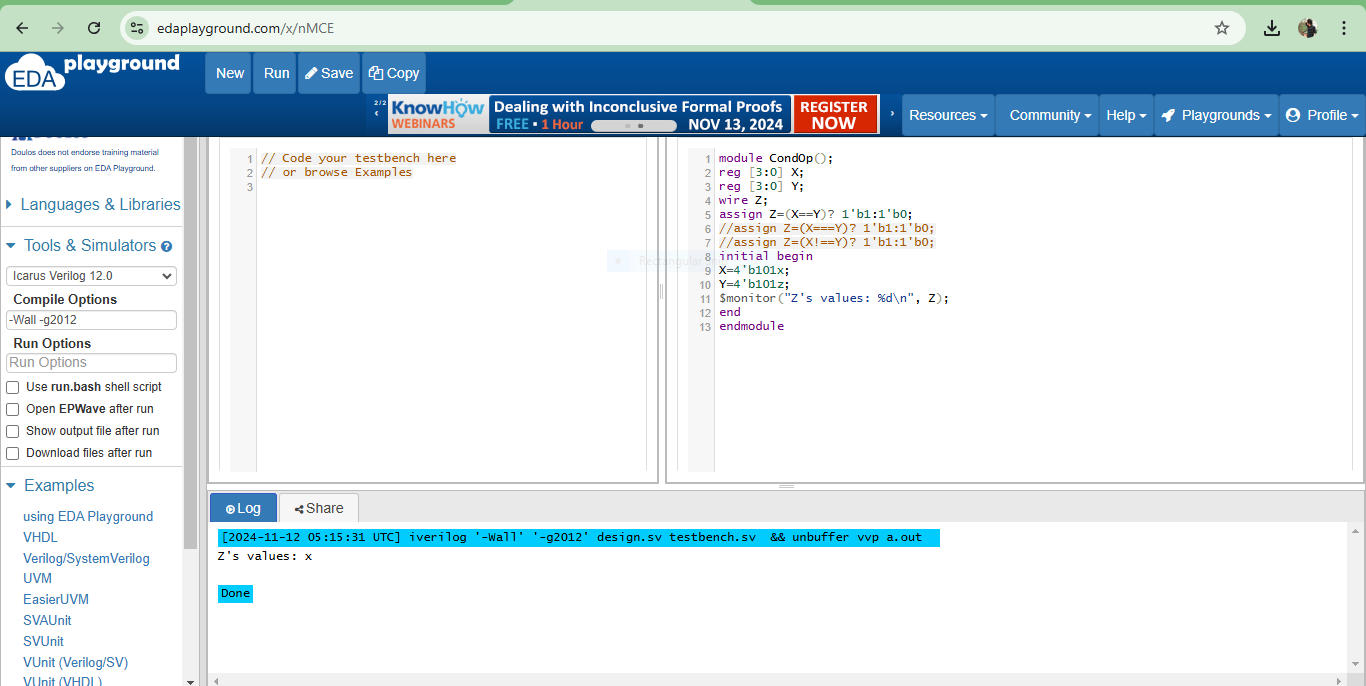
 **Statement 3 has been selected!**: This line is printed because X (4'b101x) is not exactly equal to Y (4'b101z), as the !== operator considers x and z to be different values.

 **Statement 4 has been selected!**: This line is printed because X is exactly equal to itself (X === X), even with the x state, as === always returns true for a variable compared to itself.

 **No one has been selected!**: This line is printed as the final message in the initial block, indicating that all conditions have been evaluated.

II. Continuous assignment with conditional operator

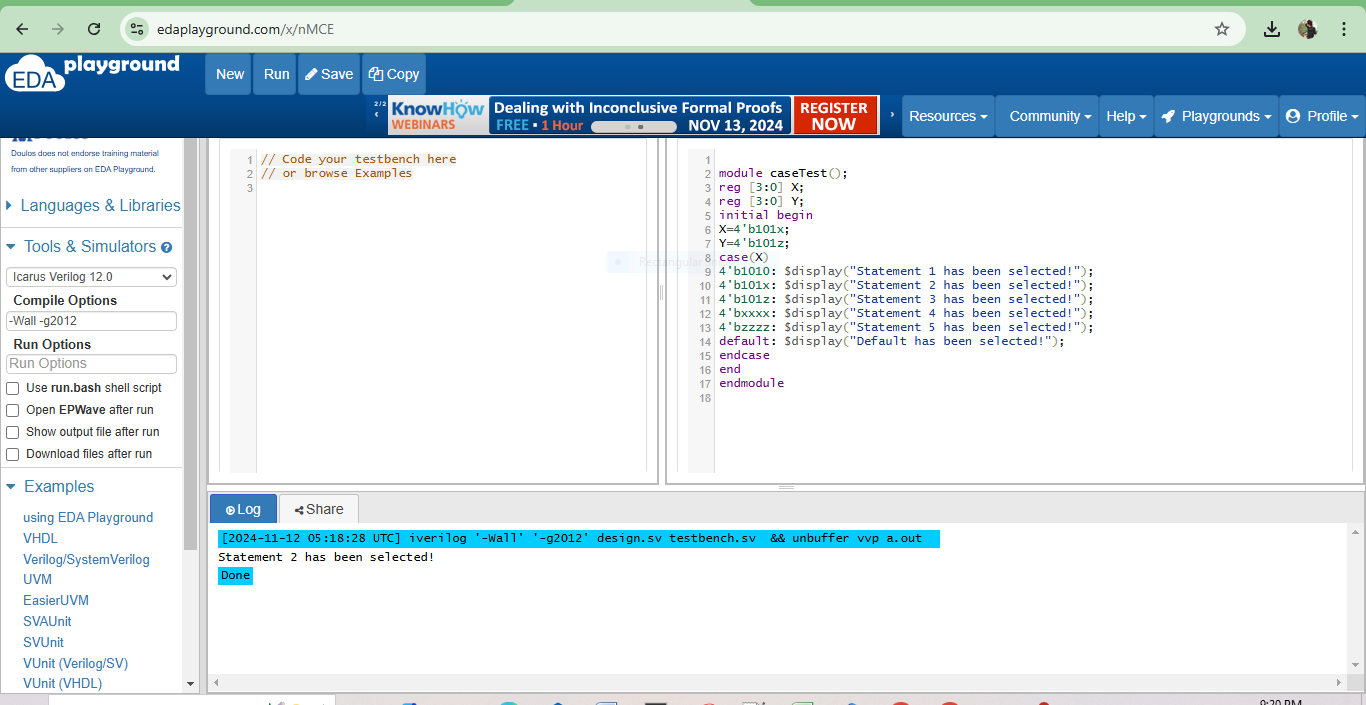
- If a has x or z in conditional operation, then a = x instead of other values in  
assignment



 In Verilog, when a conditional operation involves x or z in a regular equality comparison (==), the result is x rather than 1 or 0.

 **Impact of === and !==**: If you change the comparison to X === Y or X !== Y, Verilog would treat x and z as exact values, and the condition would yield either 1 or 0 based on a strict bitwise comparison. However, with ==, the result is indeterminate, leading to Z taking on the value x.

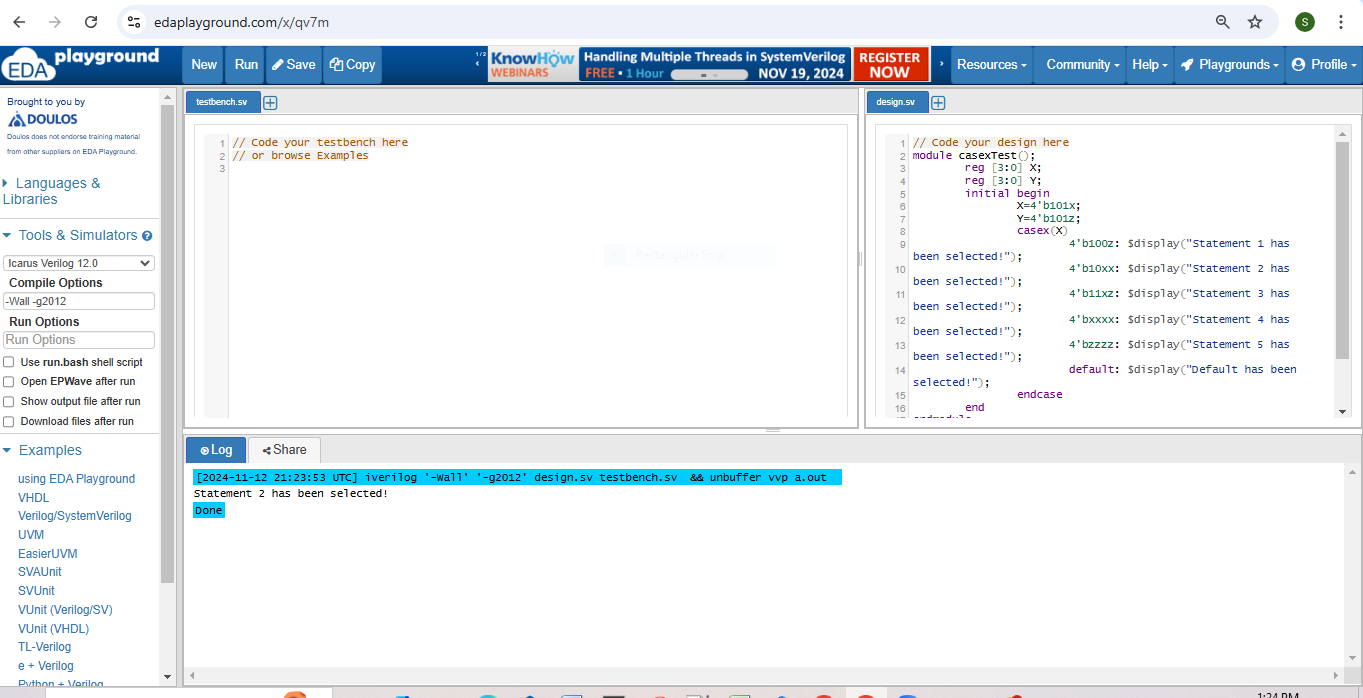
ΙΙΙ. "case" conditional statement  
- "case" is to match exactly for 1, 0 , x and z



This program illustrates that the case statement in Verilog allows for exact matching, treating x and z as specific states rather than indeterminate values. In this example, only Statement 2 is printed because X (4'b101x) matches exactly with 4'b101x in the case options.

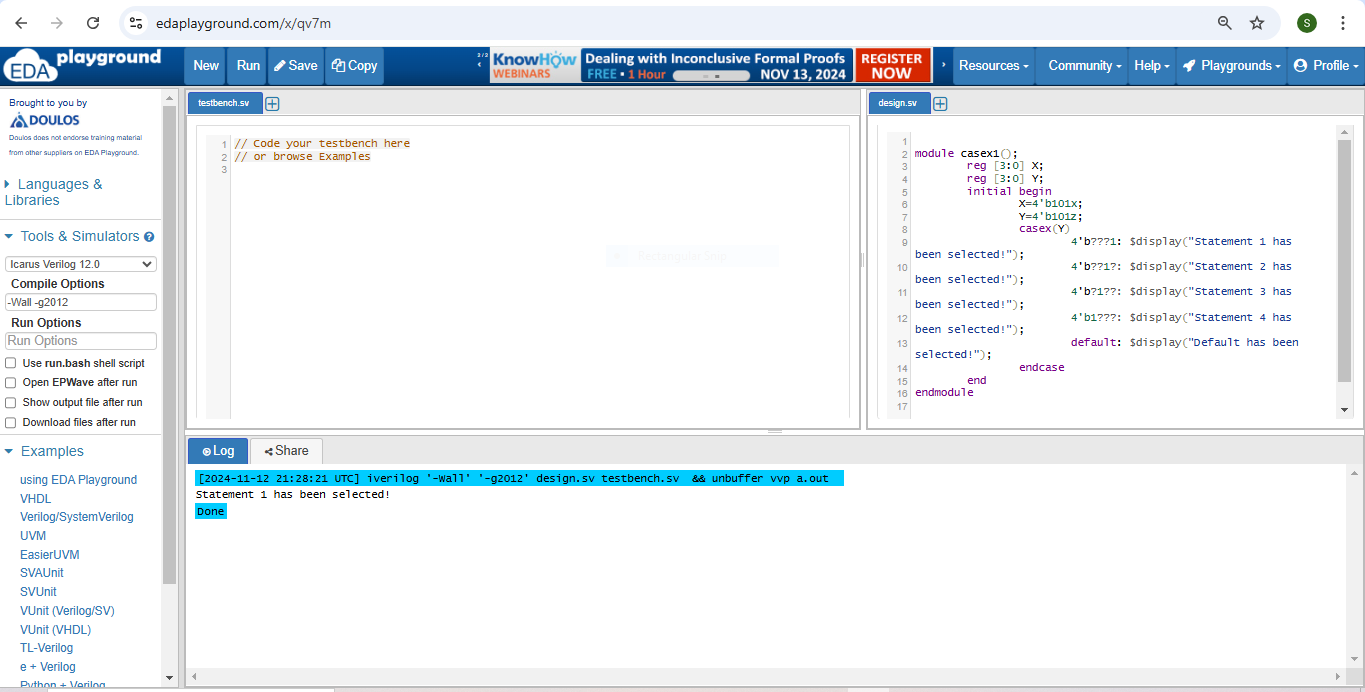
**IV."casex" conditional statement**

- "casex" is to make *x/z* match *1, 0, x* and *z*



Statement 2 is selected because 4'b10xx matches 4'b101x, as the last two bits are treated as "don't care" values. This allows X = 4'b101x to successfully match with 4'b10xx.

**- Note: in this case, *x* looks like wildcard (*?=1/0/x/z*). So, we can write as follows:**

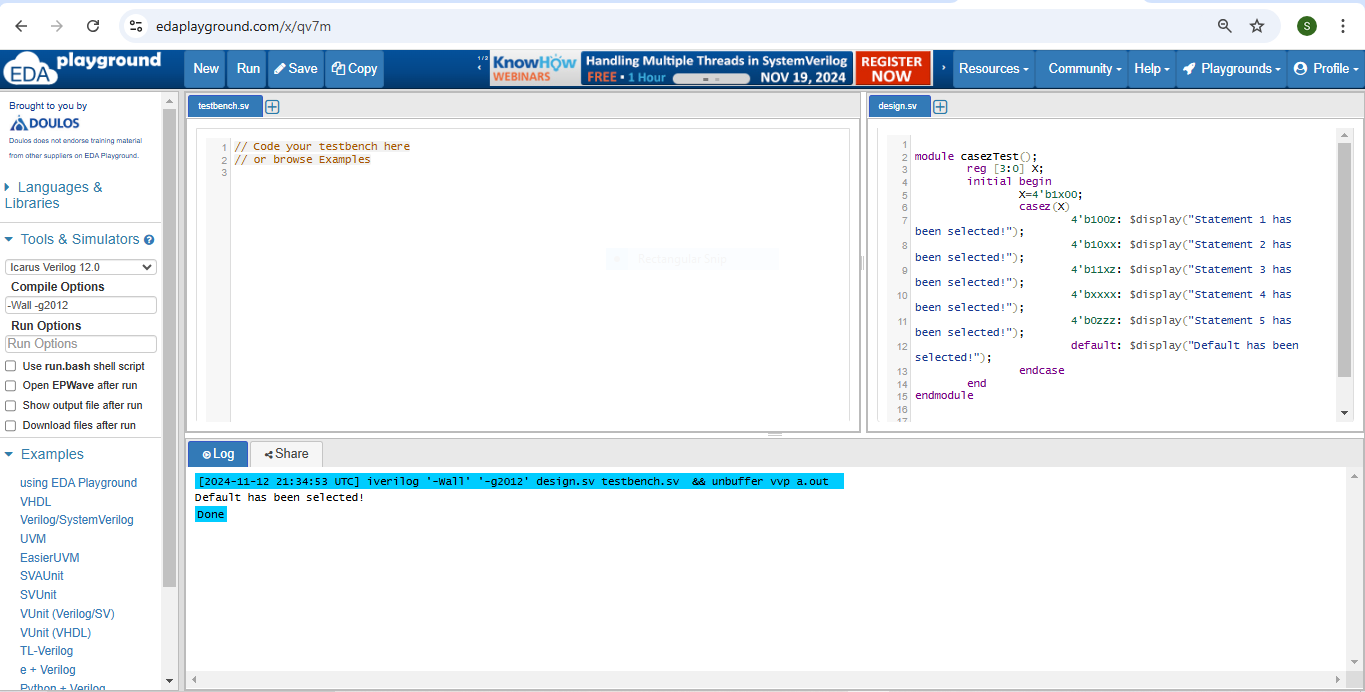


The output is "Statement 1 has been selected!" because:

* The casex wildcard ? treats the z in Y = 4'b101z as matching 1 in the pattern 4'b???1.
* Therefore, the casex statement selects Statement 1 as the first matching condition.

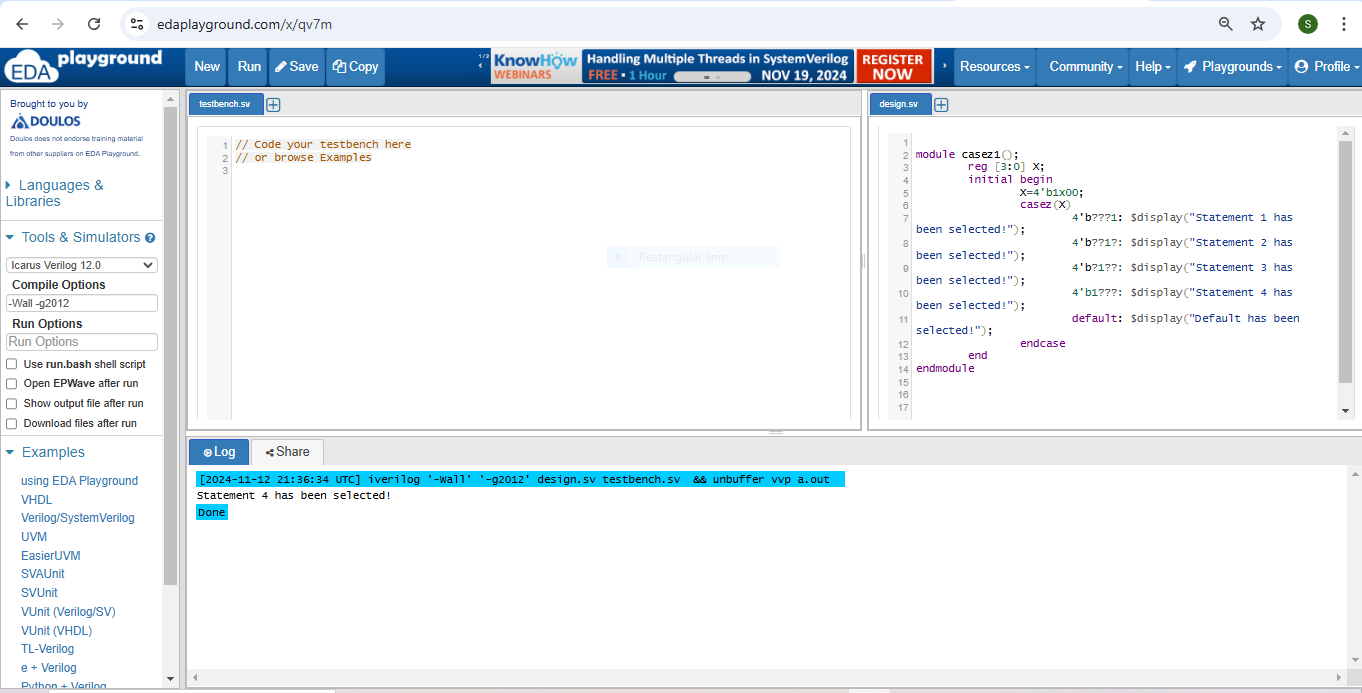
**V."casez" conditional statement**

- "casez" is to make *z* match *1, 0* and *z*, but not *x*.



Since casez does not allow x in the input to be treated as a wildcard, none of the provided patterns match X = 4'b1x00. Therefore, the default case is chosen, resulting in the output "Default has been selected!".

**- In "casez" block, wildcard *? = 1/0/x/z*.**



The casez block treats ? as a wildcard that can match any value. The pattern 4'b1??? matches X = 4'b1x00 because:

* The first bit of X is 1, which aligns with the 1 in the first position of 4'b1???.
* The remaining bits (x, 0, 0) match the ? wildcards, which can represent any value.