Verilog HDL

Assignment 5 Extended

By: Magdy Ahmed Abbas Abdelhamid

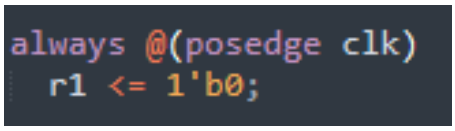
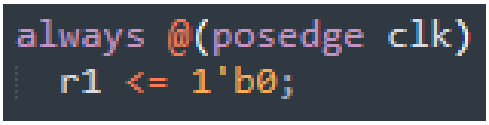
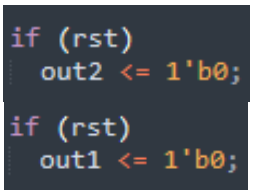
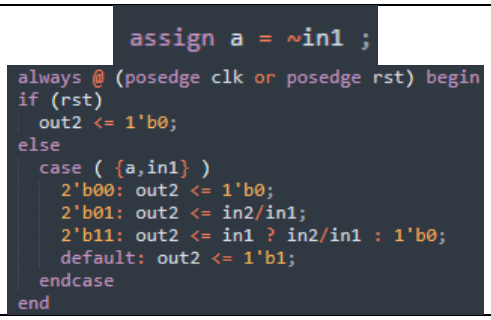
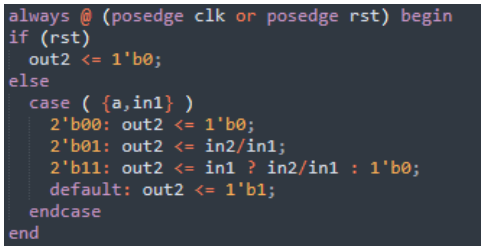
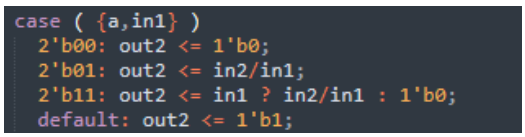
Design Checks

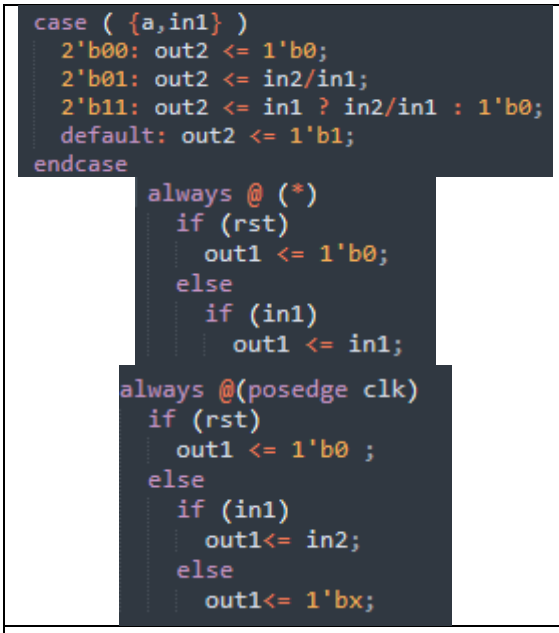
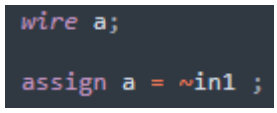
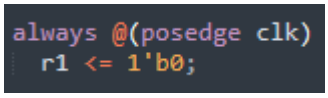
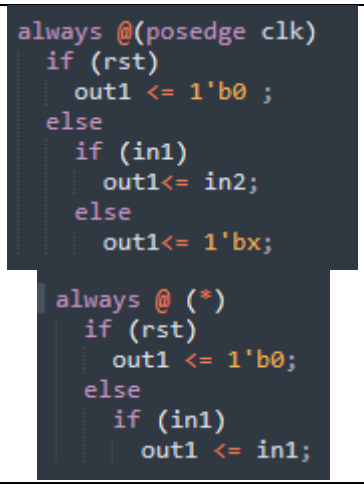
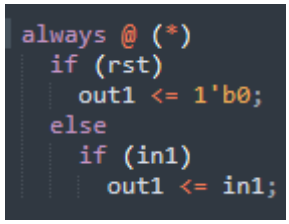
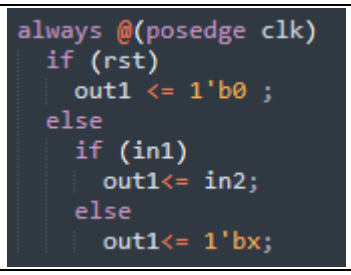
Linting List

Design Verilog Code

```
1   module top(clk, rst, in1, in2, out1, out2);
2   input clk, rst, in1, in2;
3   output reg out1, out2;
4   reg [1:0] cs, ns;
5   reg r1;
6   wire a;
7
8   assign a = ~in1 ;
9
10  always @(posedge clk or posedge rst)
11  if (rst)
12    cs <= 0;
13  else
14    cs <= ns;
15
16  always @ (*) begin
17    case (cs)
18      0   : ns <= 1;
19      1   : ns <= 1;
20      2   : ns <= in1;
21      default : ns <= 0;
22    endcase
23  end
24
25  always @ (posedge clk or posedge rst) begin
26  if (rst)
27    out2 <= 1'b0;
28  else
29    case ( {a,in1} )
30      2'b00: out2 <= 1'b0;
31      2'b01: out2 <= in2/in1;
32      2'b11: out2 <= in1 ? in2/in1 : 1'b0;
33      default: out2 <= 1'b1;
34    endcase
35  end
36
37  always @ (*)
38    if (rst)
39      out1 <= 1'b0;
40    else
41      if (in1)
42        out1 <= in1;
43
44
45  always @(posedge clk)
46    r1 <= 1'b0;
47
48
49  always @(posedge clk)
50    if (rst)
51      out1 <= 1'b0 ;
52    else
53      if (in1)
54        out1<= in2;
55      else
56        out1<= 1'bx;
57
58  endmodule
```

Synthesis Design issues

Code Snippet	Description
	**Incomplete Sensitivity List:** The always block with @(posedge clk) for signal "r1" should also include the rst signal in its sensitivity list. Missing "rst" in the sensitivity list could lead to incorrect behavior during reset.
	**Inferred Latch:** The always block for signal "r1" is missing a reset condition. As a result, an inferred latch may be created, which is not desirable for synthesis. The synthesis tool can flag this issue and suggest adding a reset condition to avoid latches.
	**Non-standard Reset Polarity:** The reset condition in some always blocks is active-high (i.e., "rst" is asserted high for reset). If the standard for the design is active-low reset, synthesis tools can identify this inconsistency.
	**Combining Blocking and Non-Blocking Assignments:** The mixing of blocking and non-blocking assignments in some always blocks can lead to race conditions and undesired behavior during synthesis like in "a" variable. Synthesis tools can catch these occurrences.
	**Non-synthesizable Case Statement:** The case statement in the always block for "out2" has complex conditions involving arithmetic and conditional expressions. Some synthesis tools may not support such complex case statements and might flag them as non-synthesizable.
	**Potential Divide-by-Zero:** The case statement for "out2" has a case where "in1" is 0 includes a division operation ("in2/in1"). Division by zero can lead to undefined behavior, and some synthesis tools may flag this as a potential issue.

	**Multiple driven Outputs:** The outputs “out1” and “out2” are not assigned values together in all scenarios. Synthesis tools can flag these cases where the outputs might be undriven and Latches occur.
	**Unused Wire:** The wire “a” is declared but not used anywhere in the module. Some synthesis tools may provide warnings for unused signals.
	**Unused Signal:** The signal “r1” is declared but not used anywhere in the module. Synthesis tools can identify this unused signal.
	**Multiple Driven:** the output “out1” is assigned multiple times with conflicting conditions in always Blocks in the same time. Synthesis tools may not handle this inconsistency well.
	**Sensitivity to All Signals (*):** Sensitivity to all signals using @(*) in an always block can lead to inefficient simulations and potential issues in larger designs. The linting tool can suggest specific sensitivity lists to improve simulation efficiency.
	**Comparison with 'x':** In the if statement for “out1”, there is a comparison with 1'bx which represents an unknown value. Linting tools can catch such comparisons that may lead to undesired behavior.

Assignment 5 - Extra

Design Issues:

1. ns is stuck at value 1 in line 19.
2. There is a potential of division by zero in line 30 if in1 = 0.
3. Case item line 31 is unreachable is a and in1 are always the same value.
4. Latch inferred line 42.
5. r1 line 46 has no reset and is always stuck at 0.
6. r1 is logic unused in the design.
7. Multiple drivers at line 42 with lines 54, 56.
8. x assignment in line 56.