Using Nested If Statements

THE XILINX HDL ADVISOR

by Roberta Fulton, Technical Marketing Engineer, Alliance Series, roberta.fulton@xilinx.com

Improper use of the "NESTED IF" statement can result in increased area and longer delays. Each IF keyword specifies priority-encoded logic. To avoid long path delays, do not use extremely long NESTED IF constructs as shown in the following VHDL/Verilog examples. These designs are shown implemented in gates in **Figure 1**. Following these examples are VHDL and Verilog designs that use the CASE construct with the NESTED IF to more effectively describe the same function. The CASE construct reduces the delay by approximately 3 ns (using an XC4005E-2 part). The implementation of this design is shown in **Figure 2**.

Inefficient Use of Nested If Statement

VHDL EXAMPLE

```
– NESTED_IF.VHD
– May 1997
Library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_UNSIGNED.all;
use IEEE.STD_LOGIC_ARITH.all;
entity nested_if is
   port (ADDR_A: in std_logic_vector (1 downto 0); – ADDRESS Code
        ADDR_B: in std_logic_vector (1 downto 0); – ADDRESS Code
        ADDR_C: in std_logic_vector (1 downto 0); – ADDRESS Code
        ADDR_D: in std_logic_vector (1 downto 0); – ADDRESS Code
        RESET: in std_logic;
        CLK : in std_logic;
        DEC_Q: out std_logic_vector (5 downto 0)); – Decode OUTPUT
end nested_if;

architecture xilinx of nested_if is
begin
———— NESTED_IF PROCESS ————
NESTED_IF: process (CLK)
begin
   if (CLK'event and CLK = '1') then
if (RESET = '0') then
   if (ADDR_A = "00") then
      DEC_Q(5 downto 4) <= ADDR_D;
      DEC_Q(3 downto 2) <= "01";
      DEC_Q(1 downto 0) <= "00";
        if (ADDR_B = "01") then
           DEC_Q(3 downto 2) <= unsigned(ADDR_A) + '1';
           DEC_Q(1 downto 0) <= unsigned(ADDR_B) + '1';
             if (ADDR_C = "10") then
                DEC_Q(5 downto 4) <= unsigned(ADDR_D) + '1';
                if (ADDR_D = "11") then
                   DEC_Q(5 downto 4) <= "00";
           end if;
             else
                DEC_Q(5 downto 4) <= ADDR_D;
             end if;
           end if;
        else
           DEC_Q(5 downto 4) <= ADDR_D;
           DEC_Q(3 downto 2) <= ADDR_A;
           DEC_Q(1 downto 0) <= unsigned(ADDR_B) + '1';
        end if;
      else
        DEC_Q <= "000000";
      end if;
   end if;
end process;
end xilinx;
```

VERILOG EXAMPLE

```
//////////////////////////////////////
// NESTED_IF.V            //
// Nested If vs. Case Design Example //
// August 1997            //
//////////////////////////////////////
module nested_if (ADDR_A, ADDR_B, ADDR_C, ADDR_D, RESET, CLK,
 DEC_Q);
   input [1:0] ADDR_A ;
   input [1:0] ADDR_B ;
   input [1:0] ADDR_C ;
   input [1:0] ADDR_D ;
   input RESET, CLK ;
   output [5:0] DEC_Q ;
   reg [5:0] DEC_Q ;
// Nested If Process //
always @ (posedge CLK)
begin
   if (RESET == 1'b1)
     begin
       if (ADDR_A == 2'b00)
          begin
            DEC_Q[5:4] <= ADDR_D;
            DEC_Q[3:2] <= 2'b01;
            DEC_Q[1:0] <= 2'b00;
              if (ADDR_B == 2'b01)
                 begin
                   DEC_Q[3:2] <= ADDR_A + 1'b1;
                   DEC_Q[1:0] <= ADDR_B + 1'b1;
                     if (ADDR_C == 2'b10)
                        begin
                          DEC_Q[5:4] <= ADDR_D + 1'b1;
                            if (ADDR_D == 2'b11)
                              DEC_Q[5:4] <= 2'b00;
                        end
                     else
                        DEC_Q[5:4] <= ADDR_D;
                 end
          end
       else
          DEC_Q[5:4] <= ADDR_D;
          DEC_Q[3:2] <= ADDR_A;
          DEC_Q[1:0] <= ADDR_B + 1'b1;
     end
   else
        DEC_Q <= 6'b000000;
end
endmodule
```

Continued on the following page

Continued from the previous page

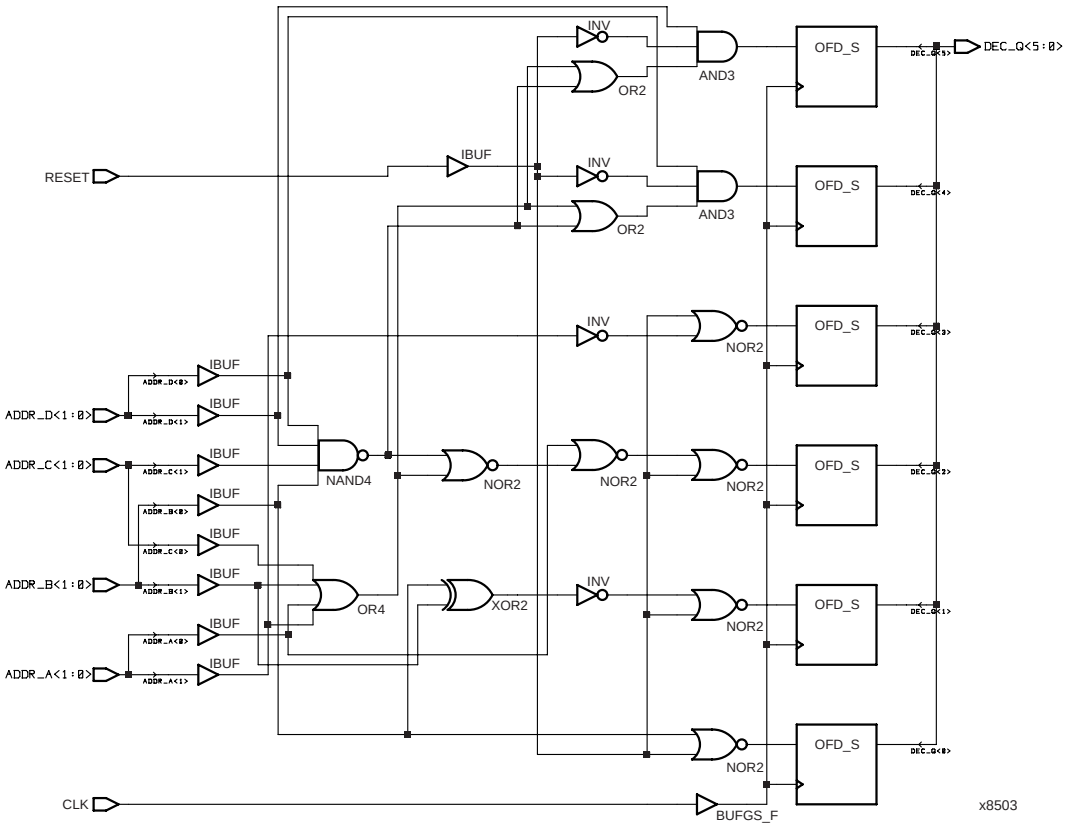

Figure 1: The gate implementation for a design with an inefficient use of a nested if statement.

Nested If Example Modified to Use If-Case

Note: *In the following example, the hyphens ("don't cares") used for bits in the CASE statement may evaluate incorrectly to False for some synthesis tools.*

VHDL EXAMPLE

```
— IF_CASE.VHD
— May 1997
Library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_UNSIGNED.all;
use IEEE.STD_LOGIC_ARITH.all;
entity if_case is
   port (ADDR_A: in std_logic_vector (1 downto 0); — ADDRESS
 Code
       ADDR_B: in std_logic_vector (1 downto 0); — ADDRESS Code
       ADDR_C: in std_logic_vector (1 downto 0); — ADDRESS Code
       ADDR_D: in std_logic_vector (1 downto 0); — ADDRESS Code
       RESET: in std_logic;
       CLK : in std_logic;
       DEC_Q: out std_logic_vector (5 downto 0)); — Decode
 OUTPUT
end if_case;

architecture xilinx of if_case is
signal ADDR_ALL : std_logic_vector (7 downto 0);
begin
——concatenate all address lines ————————-
ADDR_ALL <= (ADDR_A & ADDR_B & ADDR_C & ADDR_D) ;
———Use 'case' instead of 'nested_if' for efficient gate
 netlist——
IF_CASE: process (CLK)
begin
   if (CLK'event and CLK = '1') then
     if (RESET = '0') then
       case ADDR_ALL is
         when "00011011" =>
           DEC_Q(5 downto 4) <= "00";
           DEC_Q(3 downto 2) <= unsigned(ADDR_A) + '1';
           DEC_Q(1 downto 0) <= unsigned(ADDR_B) + '1';
         when "000110—" =>
           DEC_Q(5 downto 4) <= unsigned(ADDR_D) + '1';
           DEC_Q(3 downto 2) <= unsigned(ADDR_A) + '1';
           DEC_Q(1 downto 0) <= unsigned(ADDR_B) + '1';
         when "0001——" =>
           DEC_Q(5 downto 4) <= ADDR_D;
           DEC_Q(3 downto 2) <= unsigned(ADDR_A) + '1';
           DEC_Q(1 downto 0) <= unsigned(ADDR_B) + '1';
         when "00————" =>
           DEC_Q(5 downto 4) <= ADDR_D;
           DEC_Q(3 downto 2) <= "01";
           DEC_Q(1 downto 0) <= "00";
         when others =>
           DEC_Q(5 downto 4) <= ADDR_D;
           DEC_Q(3 downto 2) <= ADDR_A;
           DEC_Q(1 downto 0) <= unsigned(ADDR_B) + '1';
       end case;
     else
       DEC_Q <= "000000";
     end if;
   end if;
end process;
end xilinx;
```

VERILOG EXAMPLE

```
//////////////////////////////////////
// IF_CASE.V              //
// Nested If vs. Case Design Example  //
// August 1997             //
//////////////////////////////////////
module if_case (ADDR_A, ADDR_B, ADDR_C, ADDR_D, RESET, CLK,
 DEC_Q);
   input [1:0] ADDR_A ;
   input [1:0] ADDR_B ;
   input [1:0] ADDR_C ;
   input [1:0] ADDR_D ;
   input RESET, CLK ;
   output [5:0] DEC_Q ;
   wire [7:0] ADDR_ALL ;
   reg [5:0] DEC_Q ;
// Concatenate all address lines //
assign ADDR_ALL = {ADDR_A, ADDR_B, ADDR_C, ADDR_D} ;
// Use 'case' instead of 'nested_if' for efficient gate netlist
 //
always @ (posedge CLK)
begin
   if (RESET == 1'b1)
     begin
       casex (ADDR_ALL)
         8'b00011011: begin
            DEC_Q[5:4] <= 2'b00;
            DEC_Q[3:2] <= ADDR_A + 1;
            DEC_Q[1:0] <= ADDR_B + 1'b1;
          end
         8'b000110xx: begin
            DEC_Q[5:4] <= ADDR_D + 1'b1;
            DEC_Q[3:2] <= ADDR_A + 1'b1;
            DEC_Q[1:0] <= ADDR_B + 1'b1;
          end
         8'b0001xxxx: begin
            DEC_Q[5:4] <= ADDR_D;
            DEC_Q[3:2] <= ADDR_A + 1'b1;
            DEC_Q[1:0] <= ADDR_B + 1'b1;
          end
         8'b00xxxxxx: begin
                   DEC_Q[5:4] <= ADDR_D;
            DEC_Q[3:2] <= 2'b01;
            DEC_Q[1:0] <= 2'b00;
          end
         default: begin
            DEC_Q[5:4] <= ADDR_D;
            DEC_Q[3:2] <= ADDR_A;
            DEC_Q[1:0] <= ADDR_B + 1'b1;
          end
       endcase
     end
   else
     DEC_Q <= 6'b000000;
end
endmodule
```

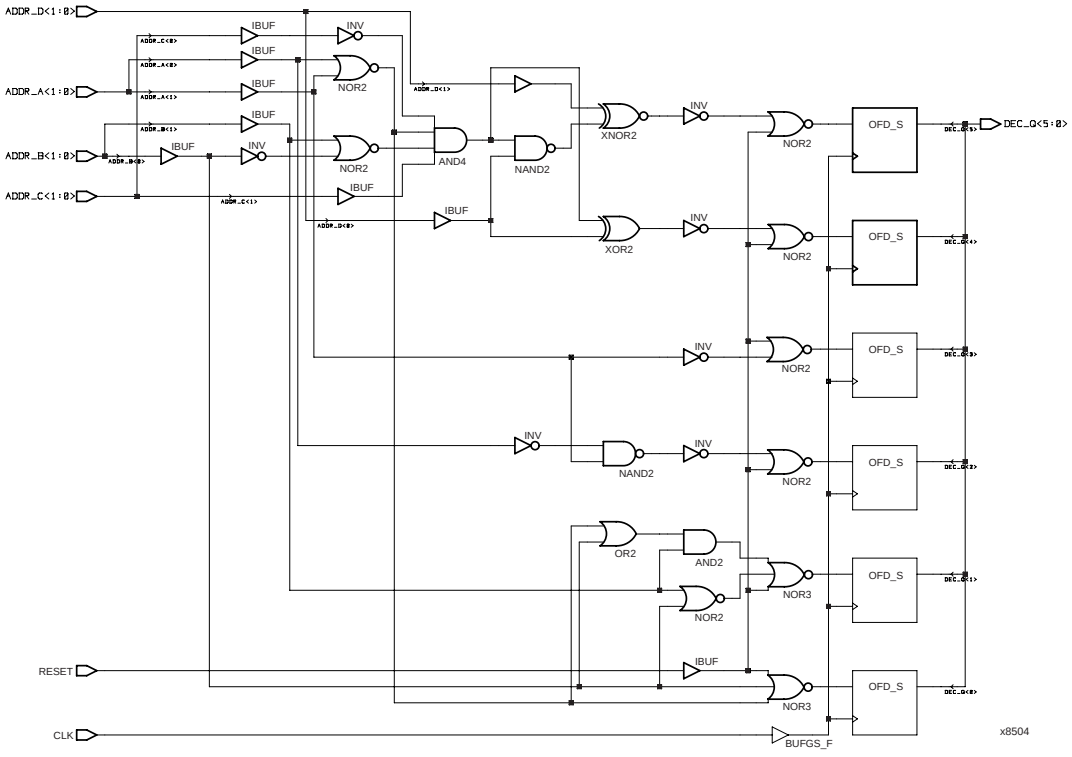

Figure 2: The gate implementation for a design modified to use if-case.

Comparing the If Statement and the Case Statement

The IF statement generally produces priority-encoded logic and the CASE statement generally creates balanced logic. An IF statement can contain a set of different expressions while a CASE statement is evaluated against a common controlling expression. In general, use the CASE statement for complex decoding and use the IF statement for speed critical paths. Most current synthesis tools can determine if the IF-ELSEIF conditions are mutually exclusive, and will not create extra logic to build the priority tree.