Gate Level Modeling Part-II

Feb-9-2014

ASIC WORLD Directory
Verilog
Tutorial
Examples
Questions
Tools
Books
Links
FAQ

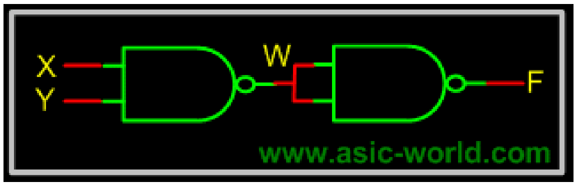
Sponsor
Home
Disclaimer
FAQ

Hot Jobs

Designing Using Primitives

Designing using primitives is used only in library development, where the ASIC vendor provides the ASIC library Verilog description, using Verilog primitives and user defined primitives (UDP).

AND Gate from NAND Gate



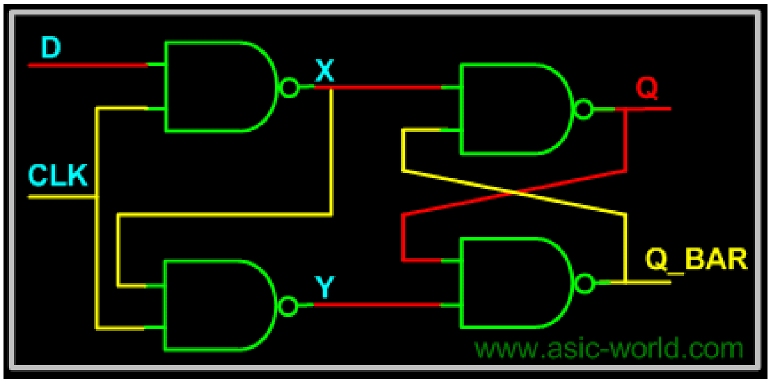
Code

```verilog
// Structural model of AND gate from two NANDS
module and_from_nand();

reg X, Y;
wire F, W;
// Two instantiations of the module NAND
nand U1(W,X, Y);
nand U2(F, W, W);

// Testbench Code
initial begin
  $monitor ("X = %b Y = %b F = %b", X, Y, F);
  X = 0;
  Y = 0;
  #1  X = 1;
  #1  Y = 1;
  #1  X = 0;
  #1  $finish;
end

endmodule
```

You could download file and_from_nand.v here

```
X = 0 Y = 0 F = 0
X = 1 Y = 0 F = 0
X = 1 Y = 1 F = 1
X = 0 Y = 1 F = 0
```

D-Flip flop from NAND Gate



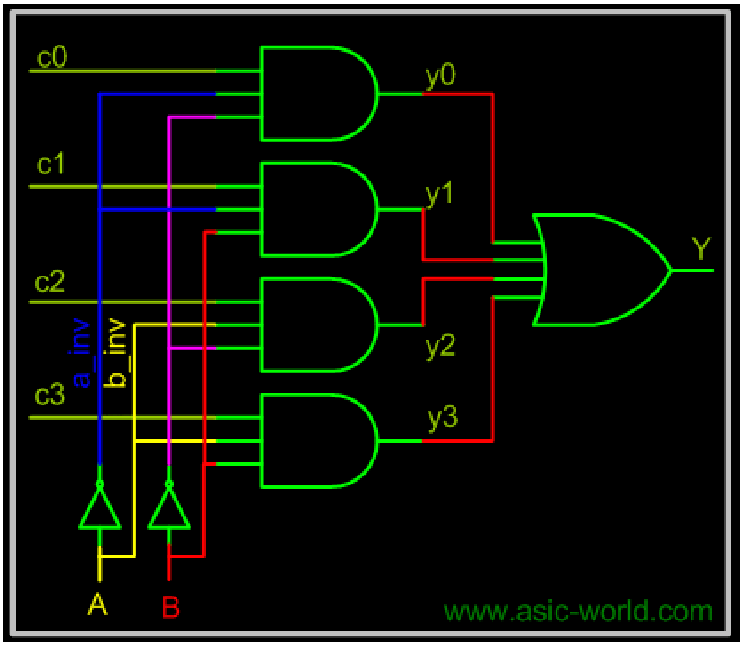
Verilog Code

```
module dff_from_nand();
wire Q,Q_BAR;
reg D,CLK;

nand U1 (X,D,CLK) ;
nand U2 (Y,X,CLK) ;
nand U3 (Q,Q_BAR,X);
nand U4 (Q_BAR,Q,Y);

// Testbench of above code
initial begin
  $monitor("CLK = %b D = %b Q = %b Q_BAR = %b",CLK, D, Q, Q_BAR);
  CLK = 0;
  D = 0;
  #3  D = 1;
  #3  D = 0;
  #3  $finish;
end

always  #2  CLK = ~CLK;

endmodule
```

You could download file dff_from_nand.v here

```
CLK = 0 D = 0 Q = x Q_BAR = x
CLK = 1 D = 0 Q = 0 Q_BAR = 1
CLK = 1 D = 1 Q = 1 Q_BAR = 0
CLK = 0 D = 1 Q = 1 Q_BAR = 0
CLK = 1 D = 0 Q = 0 Q_BAR = 1
CLK = 0 D = 0 Q = 0 Q_BAR = 1
```

Multiplexer from primitives



Verilog Code

```
module mux_from_gates ();
reg c0,c1,c2,c3,A,B;
wire Y;
//Invert the sel signals
not (a_inv, A);
not (b_inv, B);
// 3-input AND gate
and (y0,c0,a_inv,b_inv);
and (y1,c1,a_inv,B);
and (y2,c2,A,b_inv);
and (y3,c3,A,B);
// 4-input OR gate
or (Y, y0,y1,y2,y3);

// Testbench Code goes here
initial begin
  $monitor (
   "c0 = %b c1 = %b c2 = %b c3 = %b A = %b B = %b Y = %b",
   c0, c1, c2, c3, A, B, Y);
  c0 = 0;
  c1 = 0;
  c2 = 0;
  c3 = 0;
  A = 0;
  B = 0;
  #1  A  = 1;
  #2  B  = 1;
  #4  A  = 0;
  #8  $finish;
end

always  #1  c0 = ~c0;
always  #2  c1 = ~c1;
always  #3  c2 = ~c2;
always  #4  c3 = ~c3;

endmodule
```

You could download file mux_from_gates.v here

```
c0 = 0 c1 = 0 c2 = 0 c3 = 0 A = 0 B = 0 Y = 0
c0 = 1 c1 = 0 c2 = 0 c3 = 0 A = 1 B = 0 Y = 0
c0 = 0 c1 = 1 c2 = 0 c3 = 0 A = 1 B = 0 Y = 0
c0 = 1 c1 = 1 c2 = 1 c3 = 0 A = 1 B = 1 Y = 0
c0 = 0 c1 = 0 c2 = 1 c3 = 1 A = 1 B = 1 Y = 1
c0 = 1 c1 = 0 c2 = 1 c3 = 1 A = 1 B = 1 Y = 1
c0 = 0 c1 = 1 c2 = 0 c3 = 1 A = 1 B = 1 Y = 1
c0 = 1 c1 = 1 c2 = 0 c3 = 1 A = 0 B = 1 Y = 1
c0 = 0 c1 = 0 c2 = 0 c3 = 0 A = 0 B = 1 Y = 0
c0 = 1 c1 = 0 c2 = 1 c3 = 0 A = 0 B = 1 Y = 0
c0 = 0 c1 = 1 c2 = 1 c3 = 0 A = 0 B = 1 Y = 1
c0 = 1 c1 = 1 c2 = 1 c3 = 0 A = 0 B = 1 Y = 1
c0 = 0 c1 = 0 c2 = 0 c3 = 1 A = 0 B = 1 Y = 0
c0 = 1 c1 = 0 c2 = 0 c3 = 1 A = 0 B = 1 Y = 0
c0 = 0 c1 = 1 c2 = 0 c3 = 1 A = 0 B = 1 Y = 1
```

BACK HOME NEXT