Assignment 1

Name: Phạm Quang Vinh

ID: ITITIU21347

* 1. AND GATE:

Verilog Code:

module and\_gate(

output x,

input a,

input b

);

assign x = a & b;

endmodule

module test\_and;

reg t\_a, t\_b;

wire t\_x;

and\_gate DUT\_AND(t\_x, t\_a, t\_b);

initial begin

$monitor("a=%b, b=%b, x=%b", t\_a, t\_b, t\_x);

t\_a = 0; t\_b = 0;

#1 t\_a = 0; t\_b = 1;

#1 t\_a = 1; t\_b = 0;

#1 t\_a = 1; t\_b = 1;

#1 $finish;

A table with numbers and symbols

Description automatically generated end

endmodule

Truth table

A black background with white text

Description automatically generatedSimulation result

* 1. OR GATE:

Verilog Code:

module or\_gate(

output x,

input a,

input b

);

assign x = a | b;

endmodule

module test\_or;

reg t\_a, t\_b;

wire t\_x;

or\_gate DUT\_OR(t\_x, t\_a, t\_b);

initial begin

$monitor("a=%b, b=%b, x=%b", t\_a, t\_b, t\_x);

t\_a = 0; t\_b = 0;

#1 t\_a = 0; t\_b = 1;

#1 t\_a = 1; t\_b = 0;

#1 t\_a = 1; t\_b = 1;

#1 $finish;

end

endmodule

Truth table

A diagram of a number

Description automatically generated with medium confidence

Simulation result

A black screen with white text

Description automatically generated

* 1. NOT GATE:

Verilog Code:

module not\_gate(

input a,

output x

);

assign x = ~a;

endmodule

module test\_NOT;

reg t\_a;

wire t\_x;

not\_gate DUT\_NOT(t\_a, t\_x);

initial begin

$monitor("a=%b, x=%b", t\_a, t\_x);

t\_a = 0;

#1 t\_a = 1;

#1 $finish;

end

endmodule

A black and white cross with numbers

Description automatically generatedTruth table

Simulation result

A black background with white text

Description automatically generated

* 1. BUFFER GATE:

Verilog Code:

module buffer\_gate(

input a,

output x

);

assign x = a;

endmodule

module test\_BUFFER;

reg t\_a;

wire t\_x;

buffer\_gate DUT\_BUFFER(t\_a, t\_x);

initial begin

$monitor("a=%b, x=%b", t\_a, t\_x);

t\_a = 0;

#1 t\_a = 1;

#1 $finish;

end

endmodule

A black and white diagram with black text

Description automatically generated with medium confidenceTruth table

Simulation result

A black background with white text

Description automatically generated

* 1. NAND GATE:

Verilog Code:

module nand\_gate(

output x,

input a,

input b

);

assign x = ~(a & b);

endmodule

module test\_nand;

reg t\_a, t\_b;

wire t\_x;

nand\_gate DUT\_NAND(t\_x, t\_a, t\_b);

initial begin

$monitor("a=%b, b=%b, x=%b", t\_a, t\_b, t\_x);

t\_a = 0; t\_b = 0;

#1 t\_a = 0; t\_b = 1;

#1 t\_a = 1; t\_b = 0;

#1 t\_a = 1; t\_b = 1;

#1 $finish;

end

endmodule

Truth table

A black and white diagram with numbers

Description automatically generated

Simulation result

A black background with white text

Description automatically generated

* 1. NOR GATE:

Verilog Code:

module nor\_gate(

output x,

input a,

input b

);

assign x = ~(a | b);

endmodule

module test\_nor;

reg t\_a, t\_b;

wire t\_x;

nor\_gate DUT\_NOR(t\_x, t\_a, t\_b);

initial begin

$monitor("a=%b, b=%b, x=%b", t\_a, t\_b, t\_x);

t\_a = 0; t\_b = 0;

#1 t\_a = 0; t\_b = 1;

#1 t\_a = 1; t\_b = 0;

#1 t\_a = 1; t\_b = 1;

#1 $finish;

end

endmodule

Truth table

A black and white text

Description automatically generated with medium confidence

Simulation result

A black background with white text

Description automatically generated

* 1. XOR GATE:

Verilog Code:

module xor\_gate(

output x,

input a,

input b

);

assign x = (~a&b)|(a&~b);

endmodule

module test\_xor;

reg t\_a, t\_b;

wire t\_x;

xor\_gate DUT\_XOR(t\_x, t\_a, t\_b);

initial begin

$monitor("a=%b, b=%b, x=%b", t\_a, t\_b, t\_x);

t\_a = 0; t\_b = 0;

#1 t\_a = 0; t\_b = 1;

#1 t\_a = 1; t\_b = 0;

#1 t\_a = 1; t\_b = 1;

#1 $finish;

end

endmodule

Truth table

A black and white image of a cross with numbers

Description automatically generated with medium confidence

Simulation result

A black screen with white text

Description automatically generated

* 1. XNOR GATE:

Verilog Code:

module xnor\_gate(

output x,

input a,

input b

);

assign x = ~((~a&b)|(a&~b));

endmodule

module test\_xnor;

reg t\_a, t\_b;

wire t\_x;

xnor\_gate DUT\_XNOR(t\_x, t\_a, t\_b);

initial begin

$monitor("a=%b, b=%b, x=%b", t\_a, t\_b, t\_x);

t\_a = 0; t\_b = 0;

#1 t\_a = 0; t\_b = 1;

#1 t\_a = 1; t\_b = 0;

#1 t\_a = 1; t\_b = 1;

#1 $finish;

end

endmodule

Truth table

A black and white image of a number

Description automatically generated with medium confidence

Simulation result

A black screen with white text

Description automatically generated

* 1. Solution:

Verilog Code:

module Problem\_3\_1(F, A, B, C);

input A, B, C;

output F;

wire nA, nC;

wire W\_A1, W\_A2, W\_A3;

not G1(nC, C);

not G2(nA, A);

and G3(W\_A1, A, B, C);

and G4(W\_A2, A, B, nC);

and G5(W\_A3, nA, B, C);

or G6(F, W\_A1, W\_A2, W\_A3);

endmodule

module test\_or\_problem\_3\_1;

reg t\_A, t\_B, t\_C;

wire t\_F;

Problem\_3\_1 DUT(t\_F, t\_A, t\_B, t\_C);

initial begin

$monitor ("| A=%b| B=%b | C=%b | F=%b |", t\_A, t\_B, t\_C, t\_F);

t\_A = 0; t\_B = 0; t\_C = 0;

#1 t\_A = 0; t\_B = 0; t\_C = 1;

#1 t\_A = 0; t\_B = 1; t\_C = 0;

#1 t\_A = 0; t\_B = 1; t\_C = 1;

#1 t\_A = 1; t\_B = 0; t\_C = 0;

#1 t\_A = 1; t\_B = 0; t\_C = 1;

#1 t\_A = 1; t\_B = 1; t\_C = 0;

#1 t\_A = 0; t\_B = 1; t\_C = 1;

#1 $finish;

end

endmodule

Simulation result

A screen shot of a computer

Description automatically generated

* 1. Solution:

S

Verilog Code:

module Problem\_3\_2(F, A, B, C);

input A, B, C;

output F;

wire nA, nC;

wire W\_A1, W\_A2, W\_A3, W\_A4;

not G1(nC, C);

not G2(nA, A);

and G3(W\_A1, A, B);

or G4(W\_A2, C, nC);

and G5(W\_A3, W\_A1, W\_A2);

and G6(W\_A4, nA, C);

or G7(F, W\_A3, W\_A4);

endmodule

module test\_or\_problem\_3\_2;

reg t\_A, t\_B, t\_C;

wire t\_F;

Problem\_3\_2 DUT(t\_F, t\_A, t\_B, t\_C);

initial begin

$monitor ("| A=%b| B=%b | C=%b | F=%b |", t\_A, t\_B, t\_C, t\_F);

t\_A = 0; t\_B = 0; t\_C = 0;

#1 t\_A = 0; t\_B = 0; t\_C = 1;

#1 t\_A = 0; t\_B = 1; t\_C = 0;

#1 t\_A = 0; t\_B = 1; t\_C = 1;

#1 t\_A = 1; t\_B = 0; t\_C = 0;

#1 t\_A = 1; t\_B = 0; t\_C = 1;

#1 t\_A = 1; t\_B = 1; t\_C = 0;

#1 t\_A = 0; t\_B = 1; t\_C = 1;

#1 $finish;

end

endmodule

Simulation result

A screen shot of a computer code

Description automatically generated

* 1. Solution:

Verilog Code:

module Problem\_3\_2(F, A, B, C);

input A, B, C;

output F;

wire nA;

wire W\_A1, W\_A2;

not G1(nA, A);

and G2(W\_A1, A, B);

and G3(W\_A2, nA, C);

or G5(F, W\_A1, W\_A2);

endmodule

module test\_or\_problem\_3\_2;

reg t\_A, t\_B, t\_C;

wire t\_F;

Problem\_3\_2 DUT(t\_F, t\_A, t\_B, t\_C);

initial begin

$monitor ("| A=%b| B=%b | C=%b | F=%b |", t\_A, t\_B, t\_C, t\_F);

t\_A = 0; t\_B = 0; t\_C = 0;

#1 t\_A = 0; t\_B = 0; t\_C = 1;

#1 t\_A = 0; t\_B = 1; t\_C = 0;

#1 t\_A = 0; t\_B = 1; t\_C = 1;

#1 t\_A = 1; t\_B = 0; t\_C = 0;

#1 t\_A = 1; t\_B = 0; t\_C = 1;

#1 t\_A = 1; t\_B = 1; t\_C = 0;

#1 t\_A = 0; t\_B = 1; t\_C = 1;

#1 $finish;

end

endmodule

Simulation result

A screen shot of a computer

Description automatically generated