LAB REPORT
CSE4010 Computer Architecture
Instructor: Lawrence Orijuela

Name:_____Cameron Hartling_____ SCORE: ____/30

Student ID: _____007735059_____ DUE: __10/3/2024__

LAB: 2

Report

Logic gates are the basic building blocks of a digital system. They are an electronic circuit which takes one or more inputs and only one output. The different types of gates represent different relationships between the one or more inputs and the output. A universal gate is a gate that can be combined with itself in a certain way to represent all other gates. NAND and NOR gates are universal gates and are used most often as they are economical and easier to fabricate.

Code For The Seven Gates

```
// Simulates a NOR gate
module NORgate (A, B, Q); // Create module called NORgate
    // Input wires
    input A, B;
    // Output wires
    output Q;

    // Assign the output Q to the opposite of A OR B
    // NOR gate = NOT(A OR B)
    assign Q = !(A|B);

endmodule // End module NORgate

// Simulates a NAND gate
```

```
module NANDgate (A, B, Q); // Create module called NANDgate
    // Input wires
    input A, B;
    // Output wires
    output Q;

    // Assign the output Q to the opposite of A AND B
    // NAND gate = NOT(A AND B)
    assign Q = !(A&B);

endmodule // End module NANDgate

// Simulates all 7 gates with just NAND OR NOR gates
module Gates (A, B, QNOR, RNOR, SNOR, TNOR, UNOR, VNOR, XNOR, QNAND, RNAND,
SNAND, TNAND, UNAND, VNAND, XNAND); // Create module called Gates
    // Input wires
    input A, B;
    // Output wires
    output QNOR, RNOR, SNOR, TNOR, UNOR, VNOR, XNOR;
    output QNAND, RNAND, SNAND, TNAND, UNAND, VNAND, XNAND;

    // Other non input, non output wires
    wire C, D, E, F, G;

    // NOR gate
_____________________________________________________________________
    // Using NOR gates
    NORgate u1(A, B, C); // NOT(A + B)
    assign QNOR = C;
```

```
    // Using NAND gates
    NANDgate u23(A, A, C); // NOT A
    NANDgate u24(B, B, D); // NOT B
    NANDgate u25(C, D, E); // A + B
    NANDgate u26(E, E, F); // NOT(A + B)
    assign QNAND = F;

    // NAND gate
_____________________________________________________________________

    // Using NOR gates
    NORgate u2(A, A, C); // NOT A
    NORgate u3(B, B, D); // NOT B
    NORgate u4(C, D, E); // A * B
    NORgate u5(E, E, F); // NOT(A * B)
    assign RNOR = F;
    // Using NAND gates
    NANDgate u27(A, B, C); // NOT(A * B)
    assign RNAND = C;

    // NOT gate
_____________________________________________________________________

    // Using NOR gates
    NORgate u6(A, A, C); // NOT A
    assign SNOR = C;
    // Using NAND gates
    NANDgate u28(A, A, C); // NOT A
    assign SNAND = C;
```

```
    // AND gate
_____________________________________________________________________________

    // Using NOR gates
    NORgate u7(A, A, C); // NOT A
    NORgate u8(B, B, D); // NOT B
    NORgate u9(C, D, E); // A * B
    assign TNOR = E;
    // Using NAND gates
    NANDgate u29(A, B, C); // NOT(A * B)
    NANDgate u30(C, C, D); // A * B
    assign TNAND = D;

    // OR gate
_____________________________________________________________________________

    // Using NOR gates
    NORgate u10(A, B, C); // NOT (A + B)
    NORgate u11(C, C, D); // A + B
    assign UNOR = D;
    // Using NAND gates
    NANDgate u31(A, A, C); // NOT A
    NANDgate u32(B, B, D); // NOT B
    NANDgate u33(C, D, E); // A + B
    assign UNAND = E;

    // Buffer gate
_____________________________________________________________________

    // Using NOR gates
    NORgate u12(A, A, C); // NOT A
    NORgate u13(C, C, D); // A
```

```
    assign VNOR = D;
    // Using NAND gates
    NANDgate u34(A, A, C); // NOT A
    NANDgate u35(C, C, D); // A
    assign VNAND = D;

    // Exclusive-OR gate
-------------------------------------------------------------
    // Using NOR gates
    NORgate u14(A, A, C); // NOT A
    NORgate u15(B, B, D); // NOT B
    NORgate u16(A, B, E); // NOT (A + B)
    NORgate u17(C, D, F); // A * B
    NORgate u18(E, F, G); // A XOR B
    assign WNOR = G;
    // Using NAND gates
    NANDgate u36(A, B, C); // NOT (A * B)
    NANDgate u37(A, C, D); // (NOT A) + B
    NANDgate u38(B, C, E); // A + (NOT B)
    NANDgate u39(D, E, F); // A XOR B
    assign WNAND = F;

    // Exclusive-NOR gate
-------------------------------------------------------------
    // Using NOR gates
    NORgate u19(A, B, C); // NOT (A + B)
    NORgate u20(A, C, D); // (NOT A) * B
    NORgate u21(B, C, E); // A * (NOT B)
    NORgate u22(D, E, F); // A XNOR B
```

```
    assign XNOR = F;
    // Using NAND gates
    NANDgate u40(A, A, C); // NOT A
    NANDgate u41(B, B, D); // NOT B
    NANDgate u42(A, B, E); // NOT (A * B)
    NANDgate u43(C, D, F); // A + B
    NANDgate u44(E, F, G); // A XNOR B
    assign XNAND = G;

endmodule // End module Gates
```

Source Code for Parts A and B

Part A

NORusingNAND.v

```
// Simulates a NAND gate
module NANDgate (A, B, Q); // Create module called NANDgate
    // Input wires
    input A, B;
    // Output wires
    output Q;

    // Assign the output Q to the opposite of A AND B
    // NAND gate = NOT(A AND B)
    assign Q = !(A&B);

endmodule // End module NANDgate

// Simulates a NOR gate with a collection of NAND gates
module NORusingNAND (A, B, Q); // Create module called NORusingNAND
    // Input wires
    input A, B;
    // Output wires
    output Q;

    // Other non input, non output wires
    wire C, D, E, F;

    // NAND A, A, = C
    NANDgate u1(A, A, C); // NOT A
    // NAND B, B, = D
```

```
    NANDgate u2(B, B, D); // NOT B
    // NAND u1, u2, = E
    NANDgate u3(C, D, E); // A + B
    // NAND u3, u3, = F
    NANDgate u4(E, E, F); // NOT(A + B) = NOR

    // Assign the result of the four NAND gates to the output Q
    assign Q = F;

endmodule // End module NORusingNAND
```

NORusingNAND_tb.v

```
`timescale 1ns/1ns          // Sets time interval of wireTest
`include "NORusingNAND.v"   // Links NORusingNAND.v to this file

module NORusingNAND_tb; // Create module called NORusingNAND_tb

    reg A;  // Input wire called A
    reg B;  // Input wire called B
    wire Q; // Output wire called Q

    NORusingNAND uut(A, B, Q); // Instantiate wireTest as uut

    initial begin // Starts clock

        $dumpfile("NORusingNAND_tb.vcd"); // Sets file for output of the test
        $dumpvars(0, NORusingNAND_tb);
```

```
        A = 0; B = 0; #20 // Set wire A to 0, B to 0, and wait 20ns
        A = 0; B = 1; #20 // Set wire A to 0, B to 1, and wait 20ns
        A = 1; B = 0; #20 // Set wire A to 1, B to 0, and wait 20ns
        A = 1; B = 1; #20 // Set wire A to 1, B to 1, and wait 20ns

        $display("Complete!"); // Print the string

    end // Ends clock

endmodule // End module NORusingNAND_tb
```

NANDusingNOR.v

```
// Simulates a NOR gate
module NORgate (A, B, Q); // Create module called NORgate
    // Input wires
    input A, B;
    // Output wires
    output Q;

    // Assign the output Q to the opposite of A OR B
    // NOR gate = NOT(A OR B)
    assign Q = !(A|B);

endmodule // End module NORgate

// Simulates a NAND gate with a collection of NOR gates
module NANDusingNOR (A, B, Q); // Create module called NANDusingNOR
    // Input wires
```

```
    input A, B;
    // Output wires
    output Q;

    // Other non input, non output wires
    wire C, D, E, F;

    // NOR A, A, = C
    NORgate u1(A, A, C); // NOT A
    // NOR B, B, = D
    NORgate u2(B, B, D); // NOT B
    // NOR u1, u2, = E
    NORgate u3(C, D, E); // A * B
    // NOR u3, u3, = F
    NORgate u4(E, E, F); // NOT(A * B) = NAND

    // Assign the result of the four NAND gates to the output Q
    assign Q = F;

endmodule // End module NANDusingNOR
```

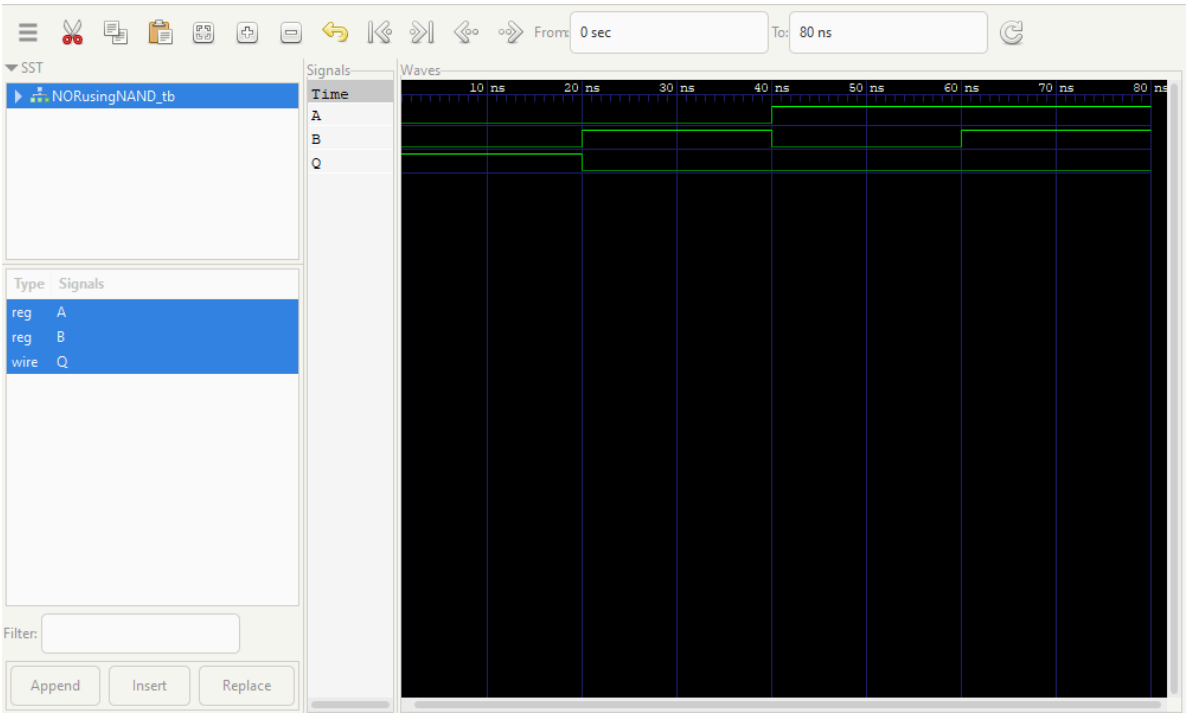
NANDusingNOR_tb.v

```
`timescale 1ns/1ns          // Sets time interval of wireTest
`include "NANDusingNOR.v"   // Links NANDusingNOR.v to this file

module NANDusingNOR_tb; // Create module called NANDusingNOR_tb

    reg A;  // Input wire called A
```

```
    reg B;  // Input wire called B
    wire Q; // Output wire called Q

    NANDusingNOR uut(A, B, Q); // Instantiate wireTest as uut

    initial begin // Starts clock

        $dumpfile("NANDusingNOR_tb.vcd"); // Sets file for output of the test
        $dumpvars(0, NANDusingNOR_tb);

        A = 0; B = 0; #20 // Set wire A to 0, B to 0, and wait 20ns
        A = 0; B = 1; #20 // Set wire A to 0, B to 1, and wait 20ns
        A = 1; B = 0; #20 // Set wire A to 1, B to 0, and wait 20ns
        A = 1; B = 1; #20 // Set wire A to 1, B to 1, and wait 20ns

        $display("Complete!"); // Print the string

    end // Ends clock

endmodule // End module NANDusingNOR_tb
```

Screenshots for Parts A and B

Part A



Part B