LAB REPORT
CSE4010 Computer Architecture
Instructor: Lawrence Orijuela

Name: Daniel Delgado Acosta

SCORE: ___/30

Student ID: 006896598

DUE: 2/28.2023

LAB: Lab2

Report

o What are logic gates? What are the universal gates and why are they important?

A logic gate is a logical function that takes inputs and produces an output depending on the operation, they are also building blocks of digital circuits. NAND gates and NOR gates are called universal gates because they can be used to create any other logic gate.

o Written code on how to make all 7 gates with NAND.

NOR gate

Part A code

AND gate

```
wire C, D;
    NANDgate u1(A, B, C);
    NANDgate u2(C, C, D);
```

OR gate

```
wire C, D, E;
    NANDgate u1(A, A, C);
    NANDgate u2(B, B, D);
    NANDgate u3(C, D, E);
```

NOT gate

```
wire C;
    NANDgate u1(A, A, C);
```

BUFFER gate

```
wire C, D;
    NANDgate u1(A, A, C);
    NANDgate u2(C, C, D);
```

Exclusive-OR gate

```
wire C, D, E, F;
    NANDgate u1(A, B, C);
    NANDgate u2(A, C, D);
    NANDgate u3(B, C, E);
    NANDgate u4(D, E, F);
```

Exclusive-NOR gate

```
wire C, D, E, F, G;
    NANDgate u1(A, B, C);
    NANDgate u2(A, A, D);
    NANDgate u3(B, B, E);
    NANDgate u4(D, E, F);
    NANDgate u4(C, F, G);
```

- Written code on how to make all 7 gates with NOR.

NAND gate

Part B code

AND gate

```
wire C, D, E;
    NORgate u1(A, A, C);
    NORgate u2(B, B, D);
    NORgate u3(C, D, E);
```

OR gate

```
wire C, D;
    NORgate u1(A, B, C);
    NORgate u2(C, C, D);
```

NOT gate

```
wire C;
    NORgate u1(A, A, C);
```

BUFFER gate

```
wire C, D;
    NORgate u1(A, A, C);
    NORgate u2(C, C, D);
```

Exclusive-OR gate

```
wire C, D, E, F, G;
    NORgate u1(A, B, C);
    NORgate u2(A, A, D);
    NORgate u3(B, B, E);
    NORgate u4(D, E, F);
    NORgate u4(C, F, G);
```

Exclusive-NOR gate

```
wire C, D, E, F;
    NORgate u1(A, B, C);
    NORgate u2(A, C, D);
    NORgate u3(B, C, E);
    NORgate u4(D, E, F)
```

Source Code for Parts A and B

Part A

NORusingNAND.v

```
// NAND gate function
module NANDgate (A, B, Q);
    // input variables for verilog
    input A, B;
     // output variable for verilog
    output Q;
    // assigns variable Q equal to NOT(A AND B)
    assign Q = !(A&B);
endmodule

// NOR gate function
module NORusingNAND (A, B, Q);
    // indicates inputs and outputs to verilog
    input A, B;
    output Q;
    // intitilzed wires
    wire C, D, E, F;
    // NOR gate from NAND gates
    NANDgate u1(A, A, C);
    NANDgate u2(B, B, D);
    NANDgate u3(C, D, E);
    NANDgate u4(E, E, F);
    // assigns Q equal to F
    assign Q = F;
endmodule
```

NORusingNAND_tb.v

```verilog
// time limit and include .v file
`timescale 1ns/1ns
`include "NORusingNAND.v"

// test bench
module NORusingNAND_tb;
//denotes wires A, B, and Q
reg A; // input
reg B; // input
wire Q; // output
// instance of NORusingNAND
NORusingNAND uut(A, B, Q);

initial begin
    // creates files for outputs
    $dumpfile("NORusingNAND_tb.vcd");
    $dumpvars(0, NORusingNAND_tb);

    A = 0; B = 0; #20 // sets A and B to 0, wait 20 nanoseconds
    A = 0; B = 1; #20 // sets A to 0 and B to 1, wait 20 nanoseconds
    A = 1; B = 0; #20 // sets A to 1 and B to 0, wait 20 nanoseconds
    A = 1; B = 1; #20 // sets A to 1 and B to 1, wait 20 nanoseconds

    // displays message to terminal
    $display("Complete!");
end
endmodule
```

Part B

NANDusingNOR.v

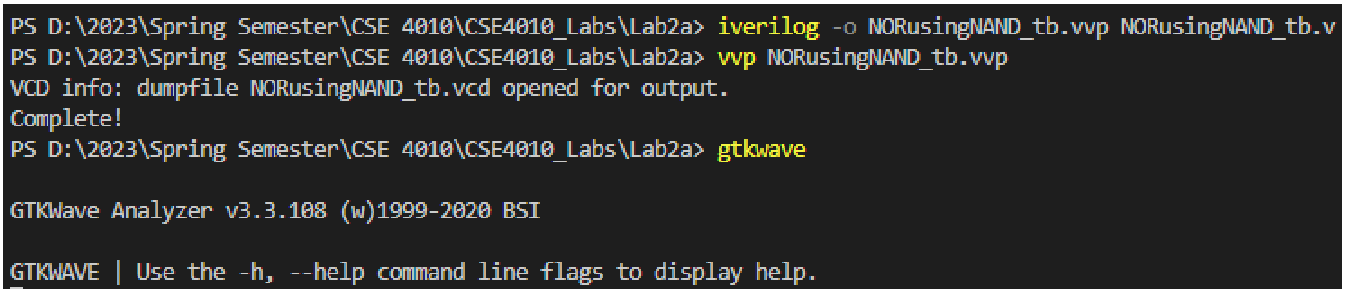
```
// NOR gate function
module NORgate (A, B, Q);
    // input variables for verilog
    input A, B;
     // output variable for verilog
    output Q;
    // assigns variable Q equal to NOT(A OR B)
    assign Q = !(A|B);
endmodule

// NAND gate function
module NANDusingNOR (A, B, Q);
    // indicates inputs and outputs to verilog
    input A, B;
    output Q;
    // intitilzed wires
    wire C, D, E, F;

    // NAND gate from NOR gates
    NORgate u1(A, A, C);
    NORgate u2(B, B, D);
    NORgate u3(C, D, E);
    NORgate u4(E, E, F);
    // assigns Q equal to F
    assign Q = F;
endmodule
```

NANDusingNOR_tb.v

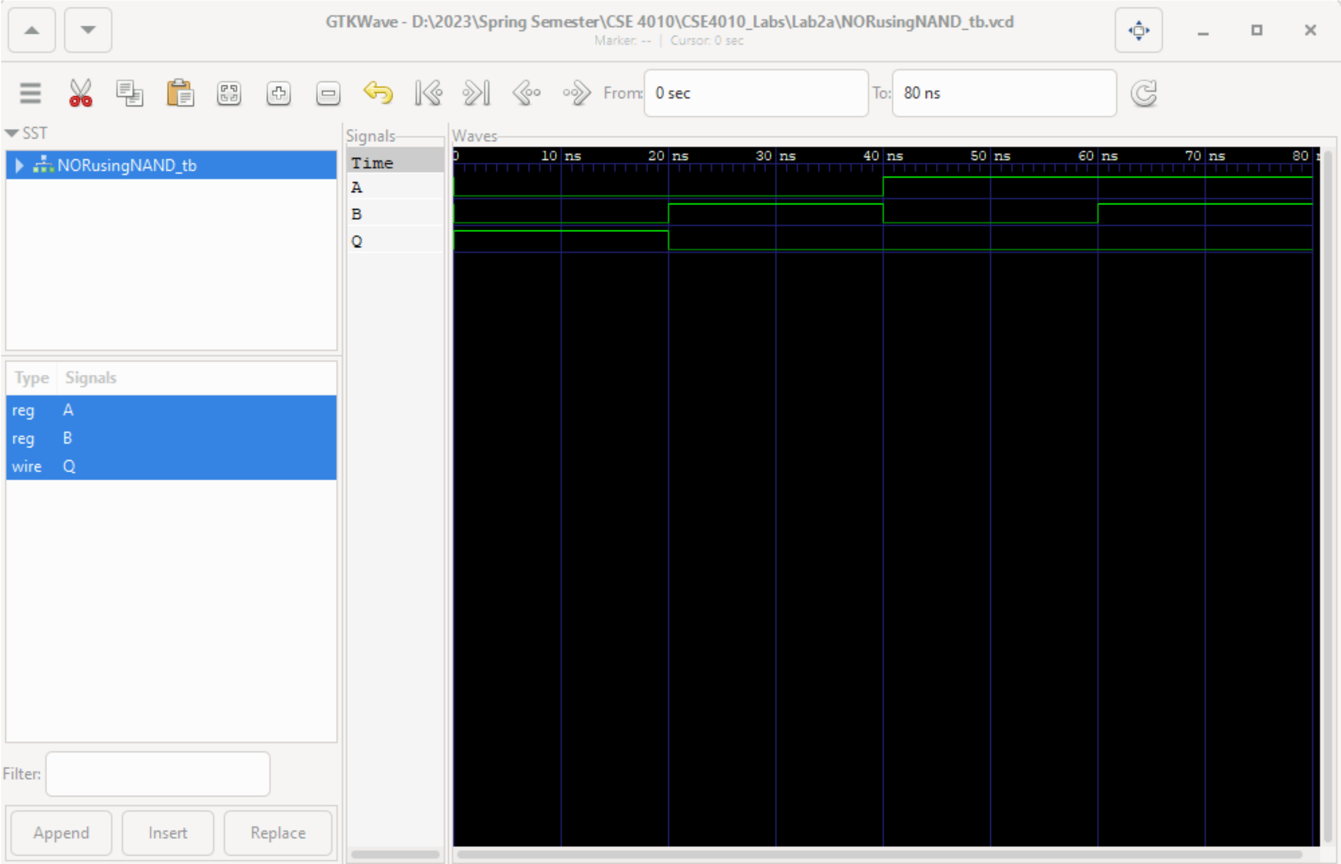
```
// time limit and include .v file
`timescale 1ns/1ns
`include "NANDusingNOR.v"

// test bench
module NANDusingNOR_tb;

//denotes wires A, B, and Q
reg A; // input
reg B; // input
wire Q; // output

// instance of NANDusingNOR
NANDusingNOR uut(A, B, Q);

initial begin
    // creates files for outputs
    $dumpfile("NANDusingNOR_tb.vcd");
    $dumpvars(0, NANDusingNOR_tb);

    A = 0; B = 0; #20 // sets A and B to 0, wait 20 nanoseconds
    A = 0; B = 1; #20 // sets A to 0 and B to 1, wait 20 nanoseconds
    A = 1; B = 0; #20 // sets A to 1 and B to 0, wait 20 nanoseconds
    A = 1; B = 1; #20 // sets A to 1 and B to 1, wait 20 nanoseconds

    //displays message to terminal
    $display("Complete!");
end
endmodule
```

Screenshots for Parts A and B

Part A



```
PS D:\2023\Spring Semester\CSE 4010\CSE4010_Labs\Lab2a> iverilog -o NORusingNAND_tb.vvp NORusingNAND_tb.v
PS D:\2023\Spring Semester\CSE 4010\CSE4010_Labs\Lab2a> vvp NORusingNAND_tb.vvp
VCD info: dumpfile NORusingNAND_tb.vcd opened for output.
Complete!
PS D:\2023\Spring Semester\CSE 4010\CSE4010_Labs\Lab2a> gtkwave

GTKWave Analyzer v3.3.108 (w)1999-2020 BSI

GTKWAVE | Use the -h, --help command line flags to display help.
```



GTKWave - D:\2023\Spring Semester\CSE 4010\CSE4010_Labs\Lab2a\NORusingNAND_tb.vcd

Marker: -- | Cursor: 0 sec

From: 0 sec To: 80 ns

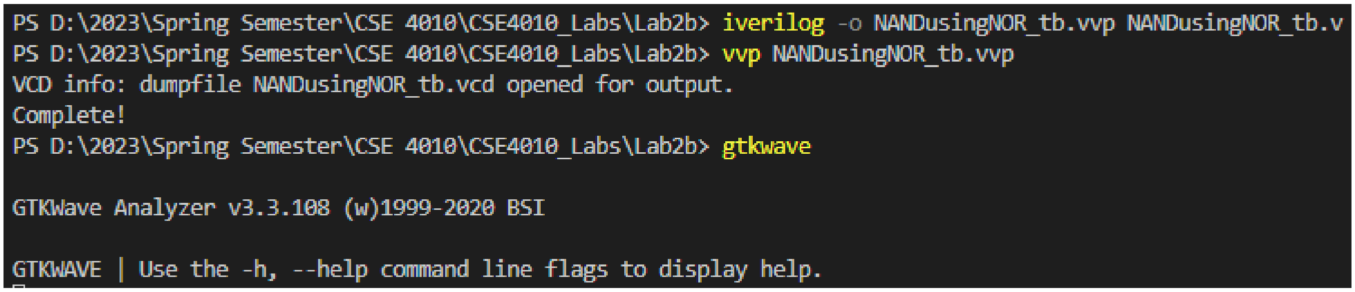
SST

NORusingNAND_tb

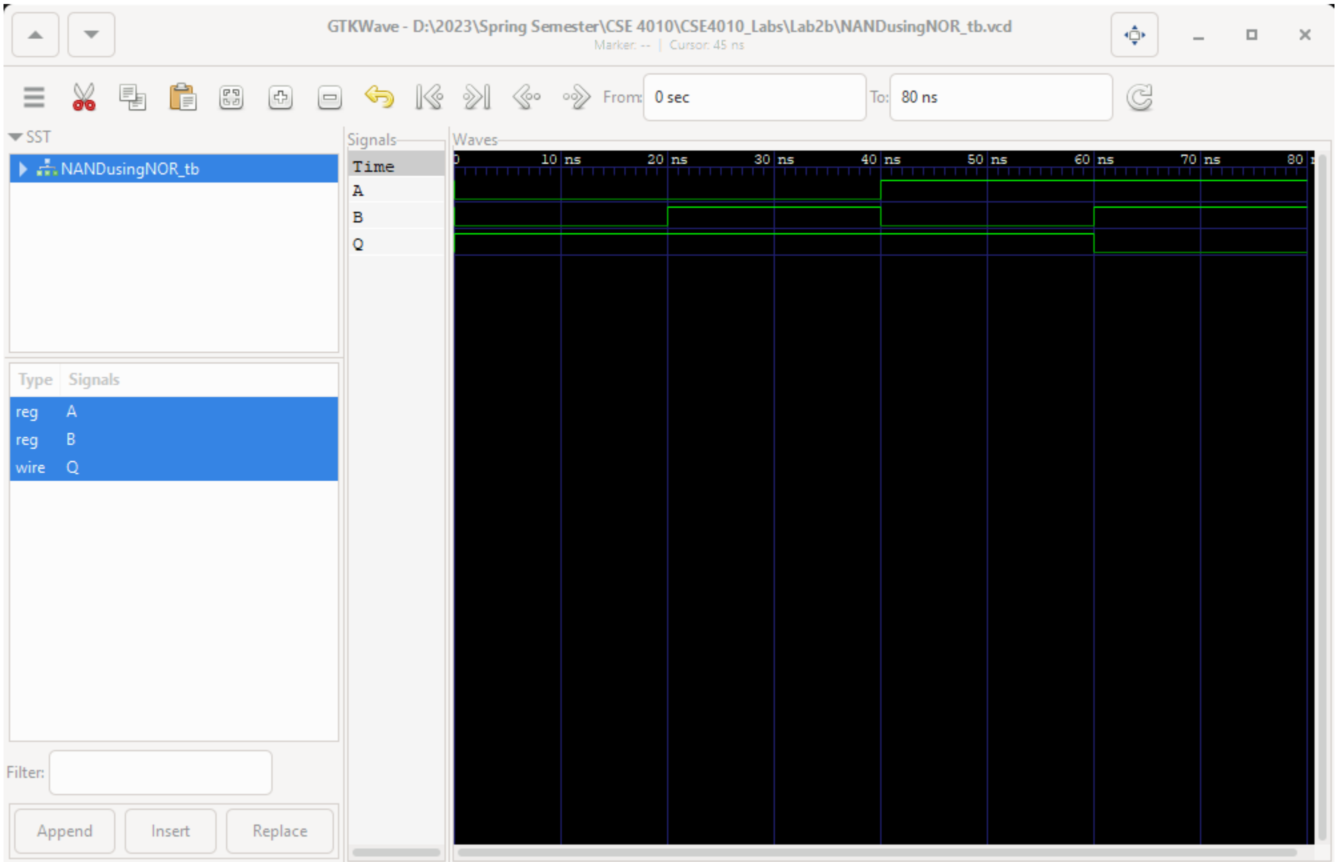
Type	Signals
reg	A
reg	B
wire	Q

Yes, it behaves like a NOR gate.

Part B



```
PS D:\2023\Spring Semester\CSE 4010\CSE4010_Labs\Lab2b> iverilog -o NANDusingNOR_tb.vvp NANDusingNOR_tb.v
PS D:\2023\Spring Semester\CSE 4010\CSE4010_Labs\Lab2b> vvp NANDusingNOR_tb.vvp
VCD info: dumpfile NANDusingNOR_tb.vcd opened for output.
Complete!
PS D:\2023\Spring Semester\CSE 4010\CSE4010_Labs\Lab2b> gtkwave

GTKWave Analyzer v3.3.108 (w)1999-2020 BSI

GTKWAVE | Use the -h, --help command line flags to display help.
```



GTKWave - D:\2023\Spring Semester\CSE 4010\CSE4010_Labs\Lab2b\NANDusingNOR_tb.vcd

Marker: -- | Cursor: 45 ns

From: 0 sec To: 80 ns

SST

NANDusingNOR_tb

Type	Signals
reg	A
reg	B
wire	Q

Yes, it behaves like a NAND gate.