INTRODUCTION TO HDL

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES

Ayala Blvd. cor San Marcelino St. Ermita, Manila

CPET7L Hands On Activity 2

CPET – 2A

Tuesday 4:00PM to 7:00 PM

Submitted By:

Pacis, Lian Gil
Recaña, Jordan
Gutierrez, Geo Kenzter
Arenas, Joseph
Estrada, Adriene Cyruz

Submitted To:

Engr. AIMEE G. ACOBA

CPE Faculty

INTRODUCTION TO HDL

Questions:

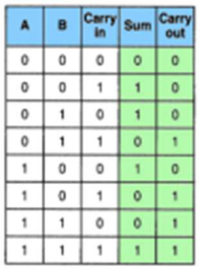
Task Assessment:

Part 1: Programming Exercises: Create a Dataflow Modeling Verilog HDL program for the following statement:

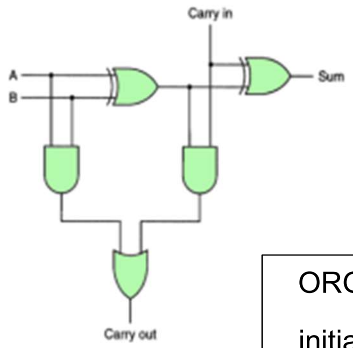
1. Listed down the Basic codes in Verilog program
   a. 2 input - OR gate
   b. 2 input - NAND gate
   c. 3 input - NOR date
   d. 3 input - XOR gate
   e. 3 input - XNOR gate

Part 2: Full Adder

1. Full Adder is the adder which adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as Carry in. The output carry is designated as Carry out and the normal output is designated as S which is SUM.
2. $S = A \oplus B \oplus Cin$ $\quad C = AB + Cin\,(A \oplus B)$



A	B	Carry in	Sum	Carry out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Answers:

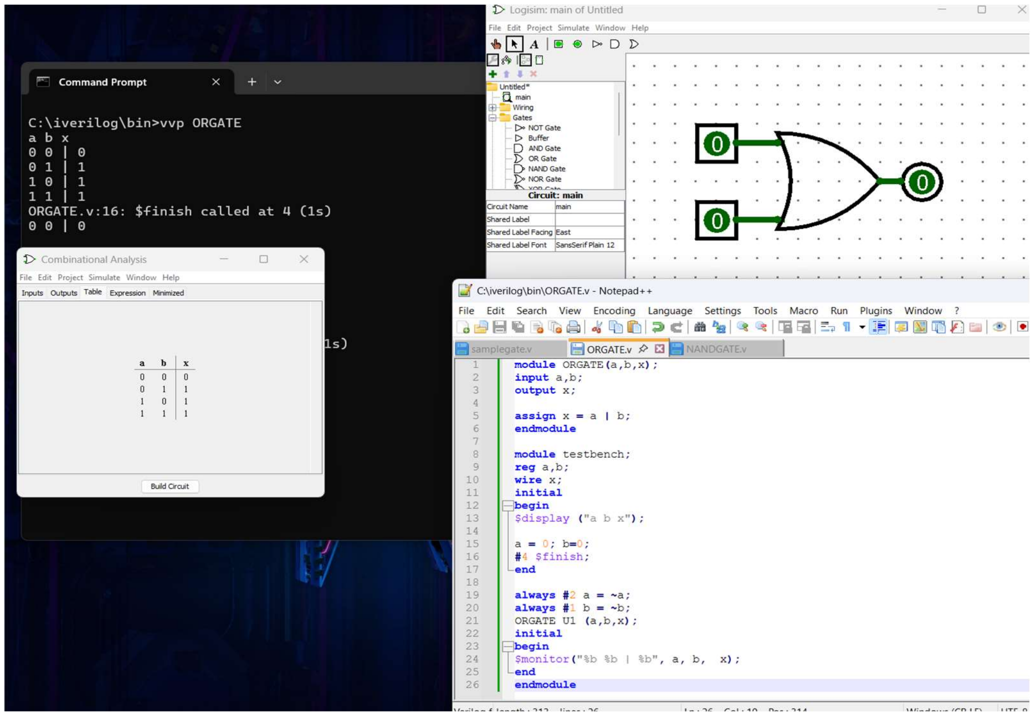
Part 1 A:

```
module ORGATE(a, b, x);
  input a, b;
  output x;
  assign x = a | b;
endmodule
module testbench;
  reg a, b;
  wire x;
  // Instantiate the ORGATE module
```

```
  ORGATE U1 (a, b, x);
  initial begin
    $display ("a b | x");
    a = 0;
    b = 0;
    #4 $finish;
  end
  always #2 a = ~a;
  always #1 b = ~b;
  initial begin
    $monitor("%b %b | %b", a, b, x);
  end
endmodule
```

INTRODUCTION TO HDL

OUTPUT PART 1A:


Logisim: main of Untitled
File Edit Project Simulate Window Help
Untitled*
main
Wiring
Gates
NOT Gate
Buffer
AND Gate
OR Gate
NAND Gate
NOR Gate
Command Prompt
Combinational Analysis
Inputs Outputs Table Expression Minimized
Build Circuit
C:\iverilog\bin\ORGATE.v - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
samplegate.v
ORGATE.v
NANDGATE.v

```
C:\iverilog\bin>vvp ORGATE
a b x
0 0 | 0
0 1 | 1
1 0 | 1
1 1 | 1
ORGATE.v:16: $finish called at 4 (1s)
0 0 | 0
```

a	b	x
0	0	0
0	1	1
1	0	1
1	1	1

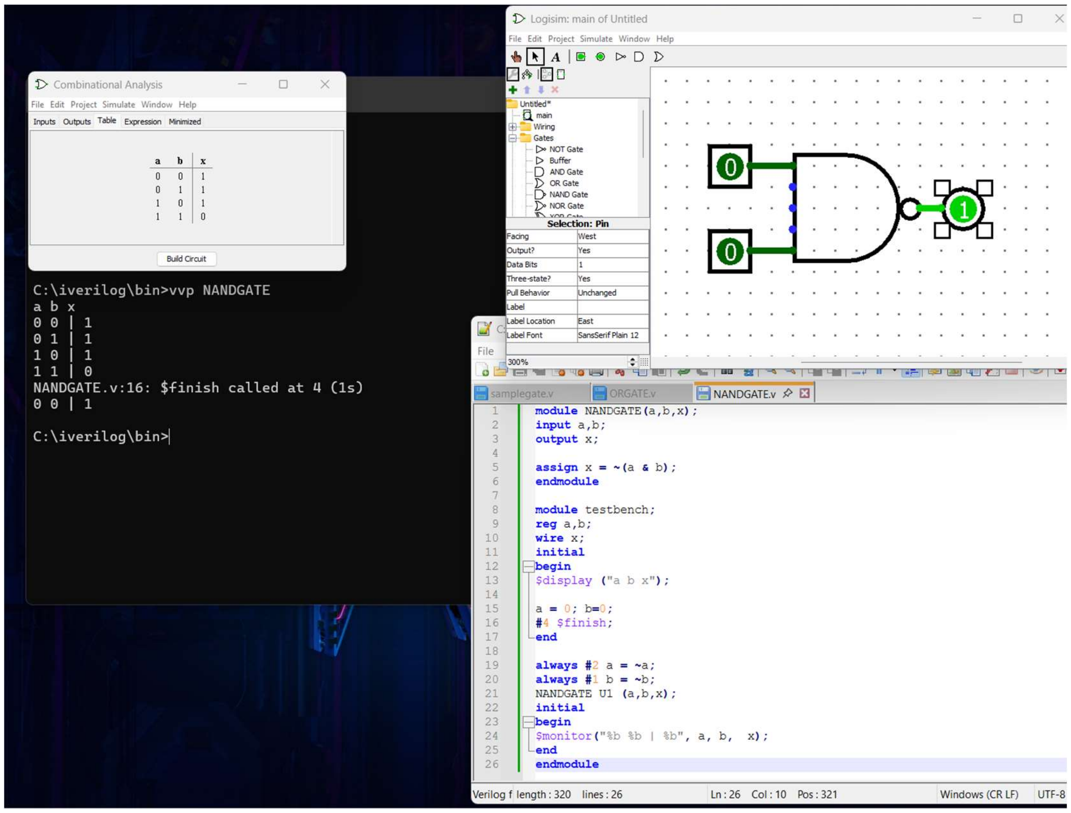
Circuit: main	
Circuit Name	main
Shared Label	
Shared Label Facing	East
Shared Label Font	SansSerif Plain 12

```verilog
module ORGATE(a,b,x);
input a,b;
output x;

assign x = a | b;
endmodule

module testbench;
reg a,b;
wire x;
initial
begin
$display ("a b x");

a = 0; b=0;
#4 $finish;
end

always #2 a = ~a;
always #1 b = ~b;
ORGATE U1 (a,b,x);
initial
begin
$monitor("%b %b | %b", a, b,  x);
end
endmodule
```

Part 1 B:

```
module NANDGATE(a,b,x);
input a,b;
output x;
assign x = ~(a & b);
endmodule
module testbench;
reg a,b;
wire x;
initial
begin
  $display ("a b x");
   a = 0;
  b = 0;
  #4 $finish;
end
always #2 a = ~a;
always #1 b = ~b;
NANDGATE U1 (a,b,x);
initial
begin
  $monitor("%b %b | %b", a, b, x);
end
endmodule
```

OUTPUT PART 1B:


Combinational Analysis
File Edit Project Simulate Window Help
Inputs Outputs Table Expression Minimized
Build Circuit
Logisim: main of Untitled
Selection: Pin
300%
samplegate.v
ORGATE.v
NANDGATE.v

a	b	x
0	0	1
0	1	1
1	0	1
1	1	0

```
C:\iverilog\bin>vvp NANDGATE
a b x
0 0 | 1
0 1 | 1
1 0 | 1
1 1 | 0
NANDGATE.v:16: $finish called at 4 (1s)
0 0 | 1

C:\iverilog\bin>
```

Facing	West
Output?	Yes
Data Bits	1
Three-state?	Yes
Pull Behavior	Unchanged
Label	
Label Location	East
Label Font	SansSerif Plain 12

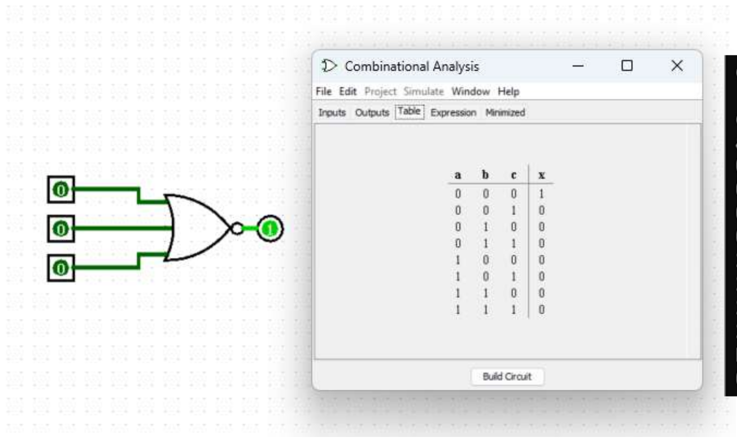
```
module NANDGATE(a,b,x);
input a,b;
output x;

assign x = ~(a & b);
endmodule

module testbench;
reg a,b;
wire x;
initial
begin
$display ("a b x");

a = 0; b=0;
#4 $finish;
end

always #2 a = ~a;
always #1 b = ~b;
NANDGATE U1 (a,b,x);
initial
begin
$monitor("%b %b | %b", a, b,  x);
end
endmodule
```

Verilog f length : 320 lines : 26 Ln : 26 Col : 10 Pos : 321 Windows (CR LF) UTF-8

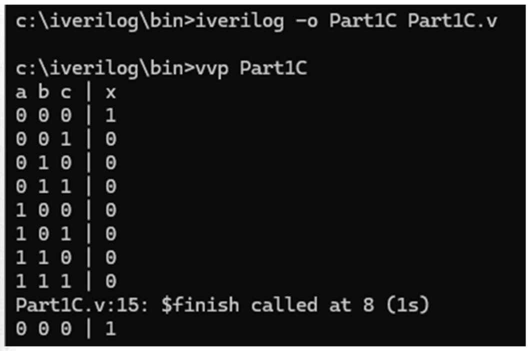
PART 1 C:

```
module norgate_3in(a, b, c, x);
input a, b, c;
output x;
assign x = ~(a | b | c);
endmodule
module test_bench;
reg a, b, c;
wire x;
initial
begin
  $display ("a b c | x");
  a = 0;
  b = 0;
  c = 0;
  #8 $finish;
end
always #4 a = ~a;
always #2 b = ~b;
always #1 c = ~c;
norgate_3in U1 (a, b, c, x);
initial
  $monitor ("%b %b %b | %b", a, b, c, x);
endmodule
```

OUTPUT PART 1C:


Combinational Analysis
File Edit Project Simulate Window Help
Inputs Outputs Table Expression Minimized
Build Circuit

a	b	c	x
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0



```
c:\iverilog\bin>iverilog -o Part1C Part1C.v

c:\iverilog\bin>vvp Part1C
a b c | x
0 0 0 | 1
0 0 1 | 0
0 1 0 | 0
0 1 1 | 0
1 0 0 | 0
1 0 1 | 0
1 1 0 | 0
1 1 1 | 0
Part1C.v:15: $finish called at 8 (1s)
0 0 0 | 1
```

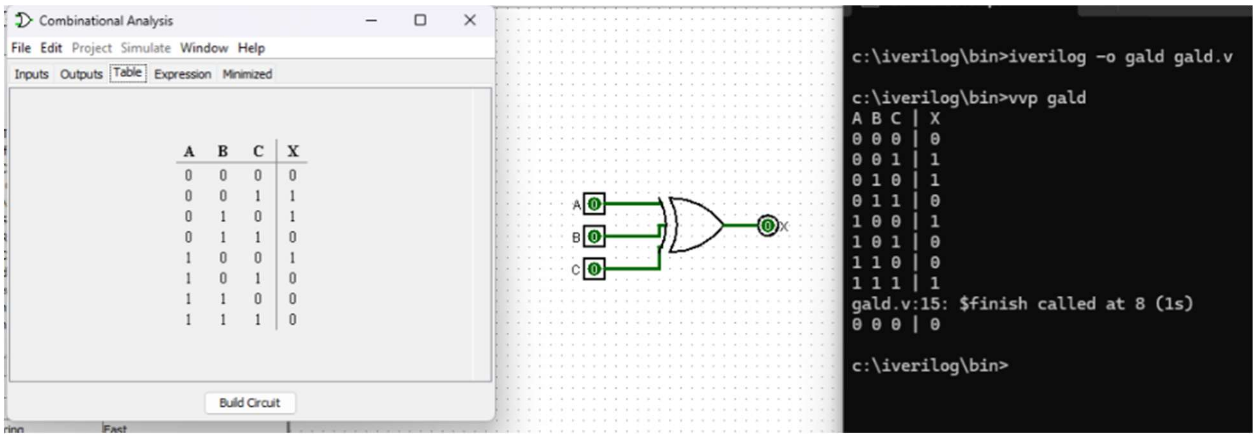
PART 1 D:

```
module gald (a, b, c, x);
input a, b, c;
output x;
assign x = (a ^ b ^ c);
endmodule
module test_bench;
reg a, b, c;
wire x;
initial
begin
  $display ("A B C | X");
  a = 0;
  b = 0;
  c = 0;
  #8 $finish;
end
always #4 a = ~a;
always #2 b = ~b;
```

```
always #1 c = ~c;
gald U1 (a, b, c, x);
initial
  $monitor ("%b %b %b | %b", a, b, c, x);
endmodule
```

INTRODUCTION TO HDL

OUTPUT PART 1D:


Combinational Analysis
File Edit Project Simulate Window Help
Inputs Outputs Table Expression Minimized
Build Circuit
A
B
C
X

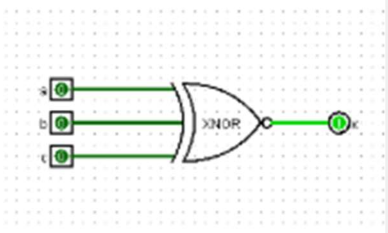
A	B	C	X
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

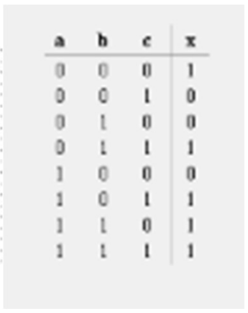
```
c:\iverilog\bin>iverilog -o gald gald.v

c:\iverilog\bin>vvp gald
A B C | X
0 0 0 | 0
0 0 1 | 1
0 1 0 | 1
0 1 1 | 0
1 0 0 | 1
1 0 1 | 0
1 1 0 | 0
1 1 1 | 1
gald.v:15: $finish called at 8 (1s)
0 0 0 | 0

c:\iverilog\bin>
```

PART 1 E:

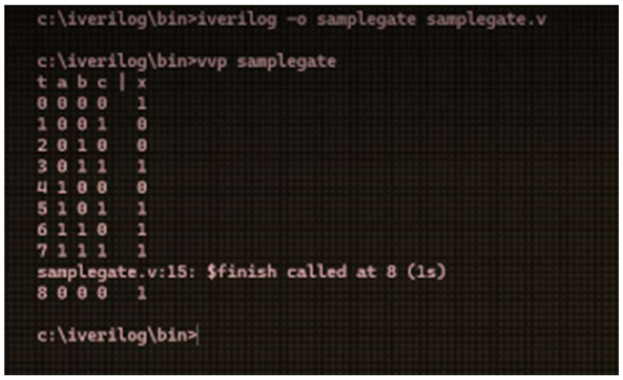
```
module samplegate (a, b, c, x);
input a, b, c;
output x;
assign x = (~a & ~b & ~c) | (b & c) | (a & c) | (a & b);
endmodule
module test_bench;
reg a, b, c;
wire x;
initial
begin
  $display ("t a b c | x");
  a = 0;
  b = 0;
  c = 0;
  #8 $finish;
end
always #4 a = ~a;
always #2 b = ~b;
always #1 c = ~c;
samplegate U1 (a, b, c, x);
initial
  $monitor ("%g %b %b %b   %b", $time, a, b, c, x);
endmodule
```

OUTPUT PART 1E:





a	b	c	x
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



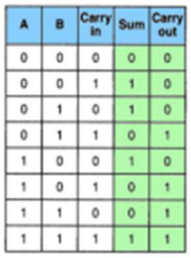
```
c:\iverilog\bin>iverilog -o samplegate samplegate.v

c:\iverilog\bin>vvp samplegate
t a b c | x
0 0 0 0   1
1 0 0 1   0
2 0 1 0   0
3 0 1 1   1
4 1 0 0   0
5 1 0 1   1
6 1 1 0   1
7 1 1 1   1
samplegate.v:15: $finish called at 8 (1s)
8 0 0 0   1

c:\iverilog\bin>
```

PART 2:

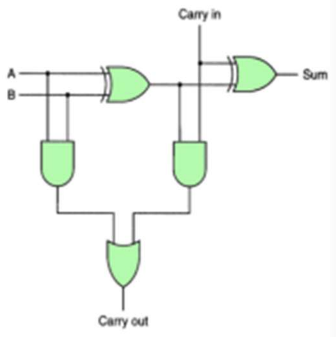
```
module task2 (a,b,cin,sum,cout);

input a,b,cin; output sum,cout;

assign sum = a ^ b ^cin; assign cout = (a & b) |
cin & (a ^ b);

endmodule

module TestBench;

reg a,b,cin; wire sum,cout;

initial

begin

        $display("A B Cin Sum Cout");

        a = 1'b0;

        b = 1'b0;

        cin = 1'b0;

        #8 $finish;

end

always #4 a = ~a;

always #2 b = ~b;

always #1 cin = ~cin;

task2 U1(a,b,cin,sum,cout);

initial

$monitor("%b %b  %b   %b   %b
",a,b,cin,sum,cout);

endmodule
```

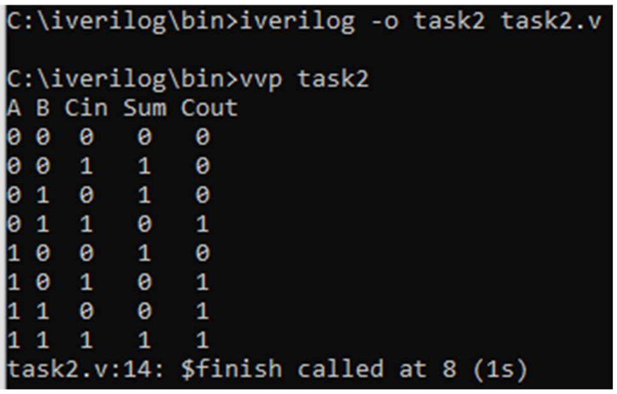
OUTPUT:

2. $S = A \oplus B \oplus Cin$ $C = AB + Cin\,(A \oplus B)$



A	B	Carry in	Sum	Carry out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1





```
C:\iverilog\bin>iverilog -o task2 task2.v

C:\iverilog\bin>vvp task2
A B Cin Sum Cout
0 0  0   0   0
0 0  1   1   0
0 1  0   1   0
0 1  1   0   1
1 0  0   1   0
1 0  1   0   1
1 1  0   0   1
1 1  1   1   1
task2.v:14: $finish called at 8 (1s)
```