Combinational Logic and basics of Verilog HDL

Submitted by

Caleb Burke

658780838

Department of Computer Science

Faculty of Science and Technology

Vancouver Island University

CSCI 355 Digital Logic and Computer Organization

Submitted to

Prof. Ajay Shrestha

September 24, 2024

Table of Contents

Contents

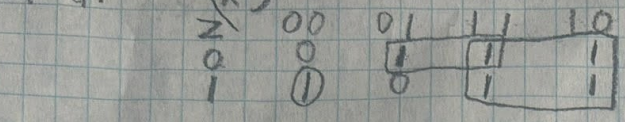
CSCi 355: Lab 4

1. Objectives

1. Familiarization with Karnaugh Maps
2. Designing a minimum sum-of-products function with Karnaugh map
3. Designing a minimum product-of-sums function with Karnaugh map
4. Familiarization with the basics of Verilog HDL
5. Understanding waveforms generated with input/output signals

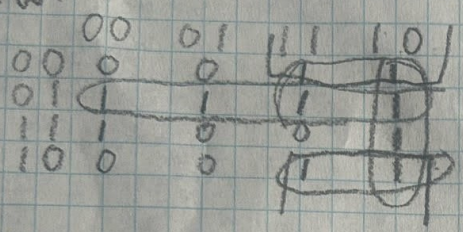
4. Pre-Lab

4.1 a. $F = x'yz' + xz' + xy + xz + y'z$



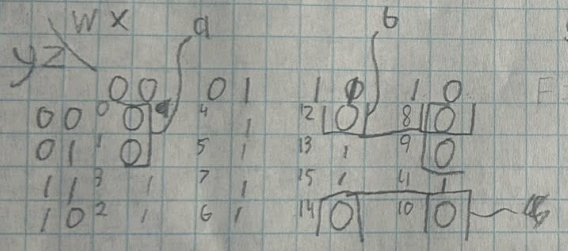
$F_{min} = x + yz' + y'z$

b. $F = xy' + wy' + xz' + wz' + y'z + w'x$



$F = xw' + z'w + xy'$

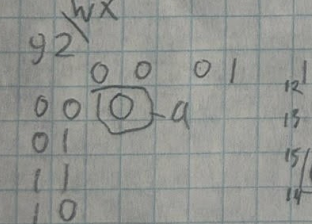
4.2 a) $F = \prod w,x,y,z\,(0,1,8,9,10,12,14)$



group
a: $w=0,1$ $x=0$ $y=0$ $z=0,1$
$= x,y$
b: $w=1$ $x=1,0$ $y=1,0$ $z=0$
$= w'z$

$F = (x+y)(w'+z)$

b) $F = \prod w,x,y,z\,(0,15)$

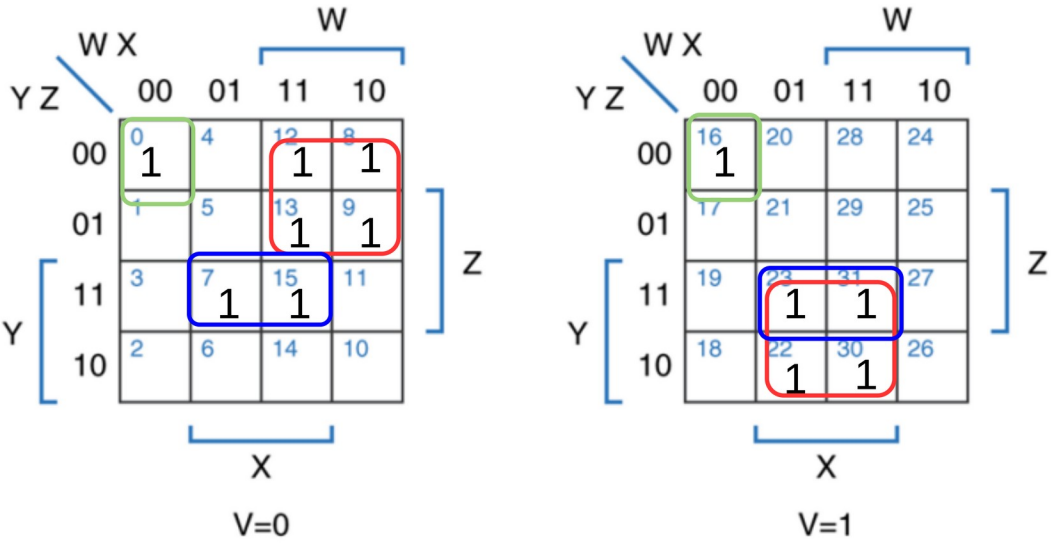


group
a: $w,x,y,z = 0$
b: $w,x,y,z = 1$

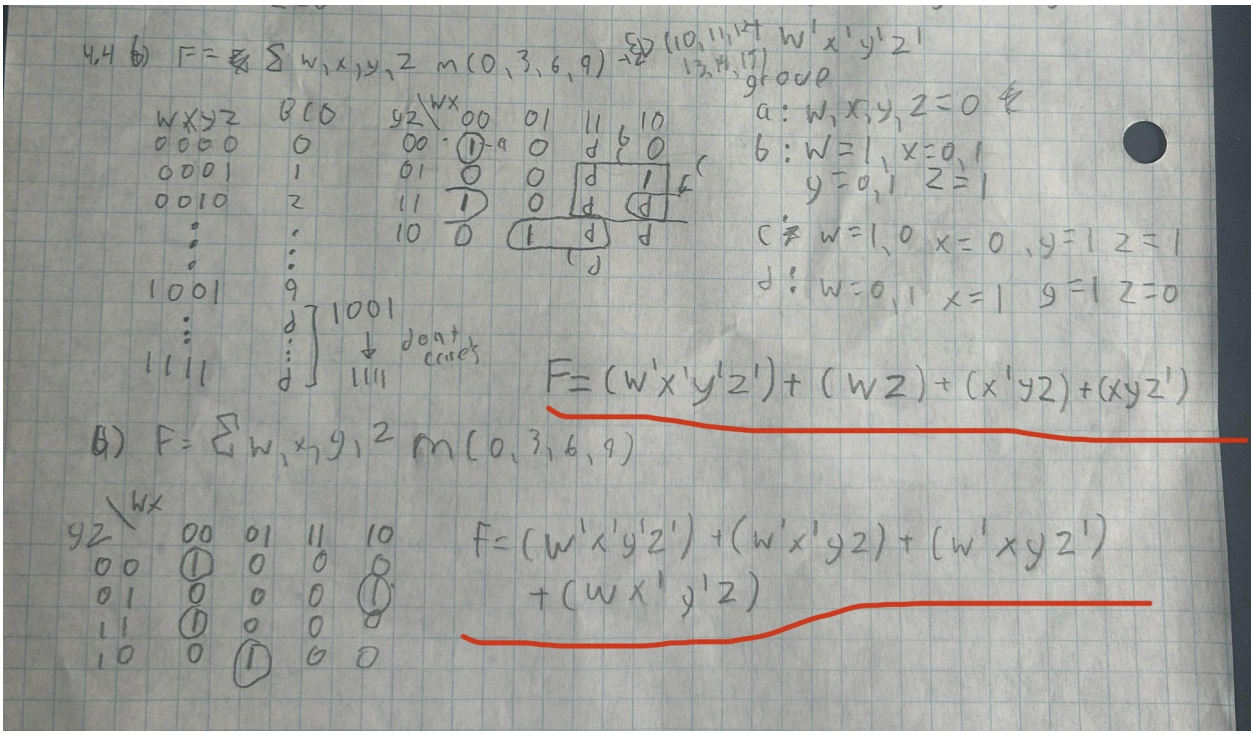
$F = (w'+x'+y'+z')(w+x+y+z)$

4.3

$$F = \sum_{V,W,X,Y,Z} (0,7,8,9,12,13,15,16,22,23,30,31)$$



F = vxy + 'vw'y + 'w'x'y'z + xyz



6. Lab Procedure with Deliverables

6.1 Three-Way Light Control using structural Verilog

Main code:

```
//Structural Specification
module threeWayLight (
   x1,x2,x3,f
);
   input wire x1,x2,x3;
   output wire f;
   wire nx1, nx2, nx3, a1, a2, a3, a4;

   not(nx1, x1); // ~x1
   not(nx2, x2); // ~x2
   not(nx3, x3); // ~x3

   and(a1, nx1, nx2, x3);
   and(a2, nx1, x2, nx3);
   and(a3, x1, x2, x3);
   and(a4, x1, nx2, nx3);
   or(f, a1, a2, a3, a4);

   assign o = (~x1 & ~x2 & x3) | (~x1 & x2 & ~x3) | (x1 & x2 & x3) | (x1 & ~x2 & ~x3);

endmodule
```

Test Bench:

```
`timescale 1ns/1ns
`include "threeWayLight.v"
module threeWayLight_tb (
);
   reg x1, x2, x3;
   wire f;

   threeWayLight ut(x1, x2, x3, f);

   initial begin
      $dumpfile("threeWayLight_tb.vcd");
      $dumpvars(0, threeWayLight_tb);

      x1 = 0;
      x2 = 0;
```

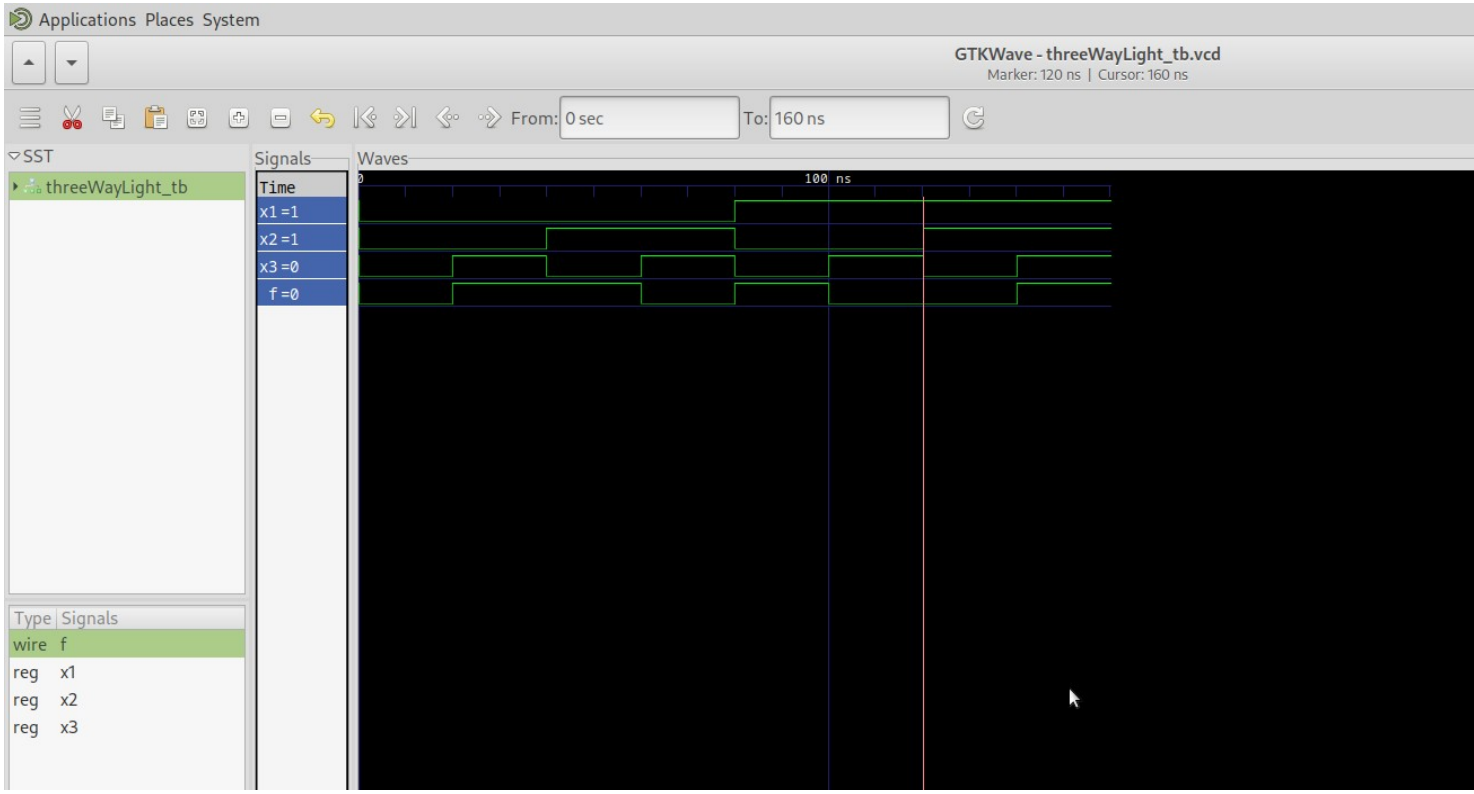
```
x3 = 0;
      #20;
x1 = 0;
x2 = 0;
x3 = 1;
      #20;
x1 = 0;
x2 = 1;
x3 = 0;
      #20;
x1 = 0;
x2 = 1;
x3 = 1;
      #20;
x1 = 1;
x2 = 0;
x3 = 0;
      #20;
x1 = 1;
x2 = 0;
x3 = 1;
      #20;
x1 = 1;
x2 = 1;
x3 = 0;
      #20;
x1 = 1;
x2 = 1;
x3 = 1;
      #20;

$display("Test complete");
```

```
    end

endmodule
```

Timing diagram



Truth Table

x1	x2	x3	f
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

Comparing the truth table and timing digram, the code works as expected.

6.2 Three-Way Light Control using Behavioural Verilog

Main code:

```
//Structural Specification
module threeWayLightFunc (
   x1,x2,x3,f
);
   input wire x1,x2,x3;
   output wire f;
   assign f = (~x1 & ~x2 & x3) | (~x1 & x2 & ~x3) | (x1 & x2 & x3) | (x1 & ~x2 & ~x3);
endmodule
```

Testbench:

```
`timescale 1ns/1ns
`include "threeWayLightFunc.v"
module threeWayLightFunc_tb ();
   reg x1, x2, x3;
   wire f;

   threeWayLightFunc ut(x1, x2, x3, f);

   initial begin
      $dumpfile("threeWayLightFunc_tb.vcd");
      $dumpvars(0, threeWayLightFunc_tb);

      x1 = 0;
      x2 = 0;
      x3 = 0;
            #20;
      x1 = 0;
      x2 = 0;
      x3 = 1;
            #20;
```

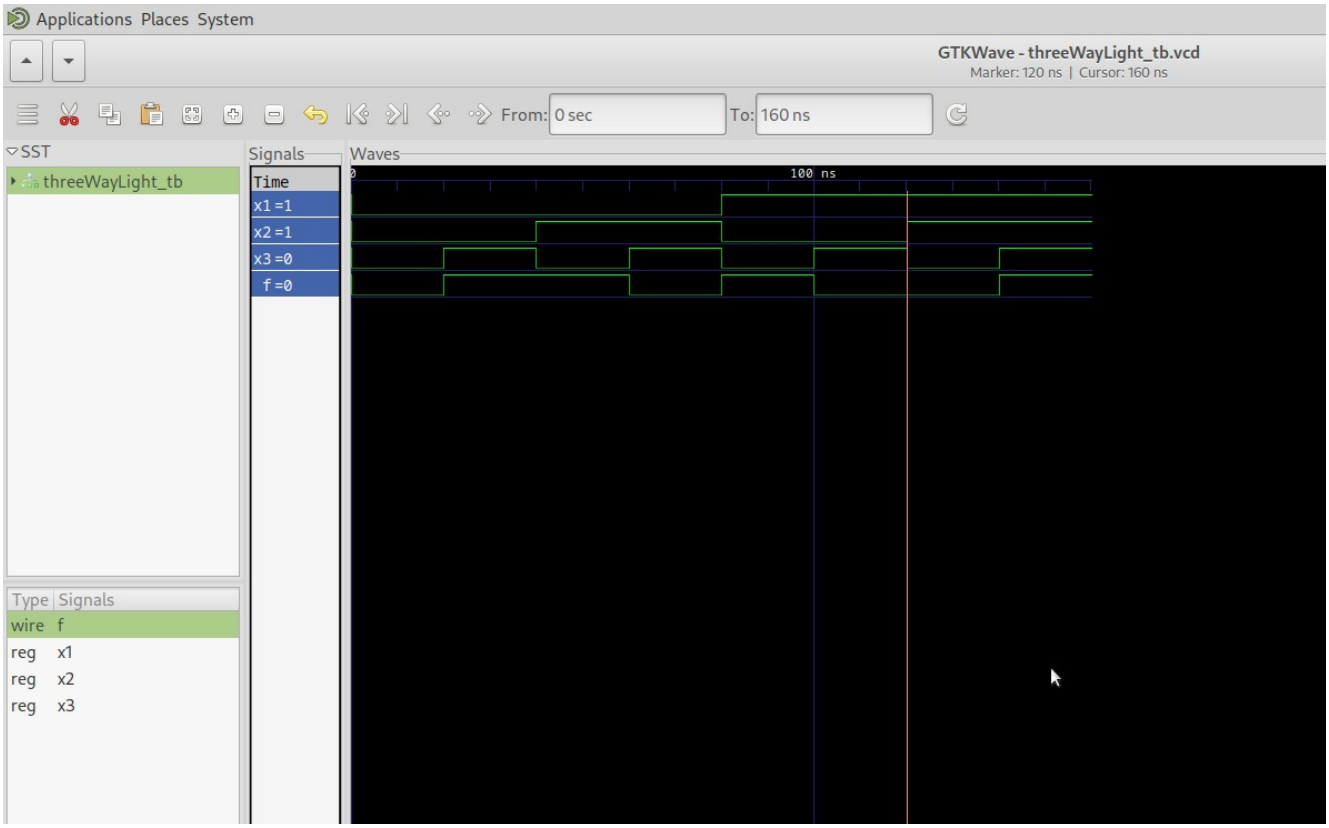
```
        x1 = 0;
        x2 = 1;
        x3 = 0;
            #20;
        x1 = 0;
        x2 = 1;
        x3 = 1;
            #20;
        x1 = 1;
        x2 = 0;
        x3 = 0;
            #20;
        x1 = 1;
        x2 = 0;
        x3 = 1;
            #20;
        x1 = 1;
        x2 = 1;
        x3 = 0;
            #20;
        x1 = 1;
        x2 = 1;
        x3 = 1;
            #20;

        $display("Test complete");
    end

endmodule
```

Timing Diagram

CSCi 355: Lab 4



Truth table

x1	x2	x3	f
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

Comparing the truth table and timing digram, the code works as expected.

Appendix

Signature: Caleb Burke