PHYS 234 Lab 2

Ethan Yang

Output 1 if and only if at least two out of three inputs are 1

To implement this functionality, we set up the FPGA circuit board to have 3 input switches and 1 output LED. Whenever two switches are flipped to true, the LED would light up.

Verilog Code:

```
`timescale 1ns / 1ps
module eelab2(
    input a,
    input b,
    input c,
    output e
    );

assign e = a&((b&~c)|(~b&c))|b&c;

endmodule
```

The results can be demonstrated using the truth table below. The 0s under A, B, and C represent when the switches are flipped to off, and 1s represent when the switches are on. In the output column, a 0 means the LED is off and 1 when the LED is on.

A	B	C	Output	Minterm	Maxterm
0	0	0	0	$\overline{A}\,\overline{B}\,\overline{C}$	$A + B + C$
1	0	0	0	$A\,\overline{B}\,\overline{C}$	$\overline{A} + B + C$
0	1	0	0	$\overline{A}\,B\,\overline{C}$	$A + \overline{B} + C$
0	0	1	0	$\overline{A}\,\overline{B}\,C$	$A + B + \overline{C}$
1	1	0	1	$A\,B\,\overline{C}$	$\overline{A} + \overline{B} + C$
1	0	1	1	$A\,\overline{B}\,C$	$\overline{A} + B + \overline{C}$
0	1	1	1	$\overline{A}\,B\,C$	$A + \overline{B} + \overline{C}$
1	1	1	1	$A\,B\,C$	$\overline{A} + \overline{B} + \overline{C}$

Using the sum of products (when minterms' output = 1), we can determine our circuit:

$$\begin{aligned} y &= AB\overline{C} + A\overline{B}C + \overline{A}BC + ABC \\ &= AB\overline{C} + A\overline{B}C + A\overline{A}(BC) \\ &= A(B\overline{C} + \overline{B}C) + BC \end{aligned}$$

The above statement written in verilog code form will be "a&((b&~c)|(~b&c))|b&c", and we assign this to our output variable e.

Additionally, we could also use the product of sums (when maxterms' output = 0) to determine the circuit, which will look like this:

$$y = (A + B + C)(\overline{A} + B + C)(A + \overline{B} + C)(A + B + \overline{C})$$

Both circuits would generate the same outcomes.

Output 1 if and only if a four-bit input represents a prime number

To implement this functionality, we set up the FPGA circuit board to have 4 input switches where each switch represents a digit in the 4-bit number. The value of the digits is 1 when the switches are flipped on and 0 when flipped off. The LED would light up whenever the 4-bit number is a prime number.

Verilog Code:

```
`timescale 1ns / 1ps
module eelab2(
    input a,
    input b,
    input c,
    input d,
    output e
    );

assign e =
(~a&b&~c&~d)|(a&b&~c&~d)|(a&~b&c&~d)|(a&b&c&~d)|(a&b&~c&d)|(a&~b&c&d)
;

endmodule
```

The results are demonstrated by the truth table below. It shows all combinations of the 4-digit input, its corresponding decimal number, and whether or not it is a prime number. We only record the minterms for the prime numbers as only they are needed to define our circuit.

D	C	B	A	Decimal	Prime?	Minterm
0	0	0	0	0	0	
1	0	0	0	8	0	
0	1	0	0	4	0	
0	0	1	0	2	1	$\overline{A}\,B\,\overline{C}\,\overline{D}$
0	0	0	1	1	0	
1	1	0	0	12	0	
0	1	1	0	6	0	
0	0	1	1	3	1	$A\,B\,\overline{C}\,\overline{D}$
1	0	1	0	10	0	
1	0	0	1	9	0	
0	1	0	1	5	1	$A\,\overline{B}\,C\,\overline{D}$
1	1	1	0	14	0	
0	1	1	1	7	1	$A\,B\,C\,\overline{D}$
1	0	1	1	11	1	$A\,B\,\overline{C}\,D$
1	1	0	1	13	1	$A\,\overline{B}\,C\,D$
1	1	1	1	15	0	

Using the sum of products (when minterms' output = 1) again, we can determine our circuit:

$$y = \overline{A}\,B\,\overline{C}\,\overline{D} + A\,B\,\overline{C}\,\overline{D} + A\,\overline{B}\,C\,\overline{D} + A\,B\,C\,\overline{D} + A\,B\,\overline{C}\,D + A\,\overline{B}\,C\,D$$

In verilog code, this will be
"(~a&b&~c&~d)|(a&b&~c&~d)|(a&~b&c&~d)|(a&b&c&~d)|(a&b&~c&d)|(a&~b&c&d)", and we assign this to our output e again.

In this scenario, we would prefer to use the sum of products method rather than the product of sums method, because it is easier to sum 6 products with an output of 1 than to multiply 10 sums with an output of 0.