Introduction to Logic Design
EEF205E

Homework 3

Rüzgar Erik
040240783

Istanbul Technical University
Faculty of Electrical and Electronics Engineering

Part 1

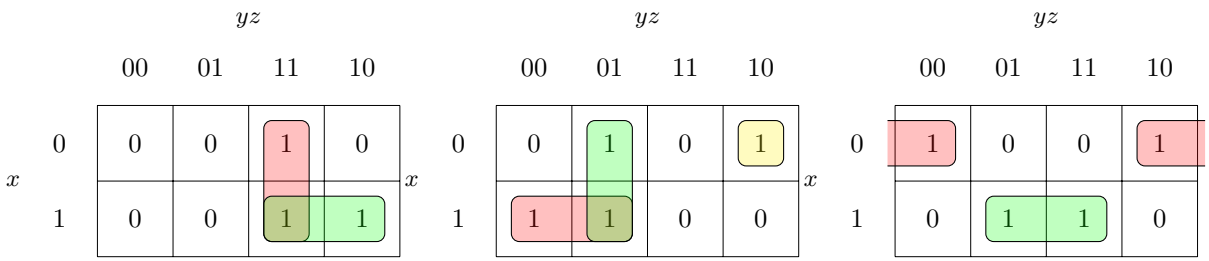
1. Design a combinational circuit with three inputs, x, y and z, and three outputs, A, B, C. When the binary input is 0, 1, 2, or 3, the binary output is one greater than the input. When the binary input is 4, 5, 6, or 7, the binary output is two less than the input.

   Solution: The truth table for the given problem is shown below.

Table 1: Truth Table for the given problem

x	y	z	INPUT	A	B	C	OUTPUT
0	0	0	0	0	0	1	1
0	0	1	1	0	1	0	2
0	1	0	2	0	1	1	3
0	1	1	3	1	0	0	4
1	0	0	4	0	1	0	2
1	0	1	5	0	1	1	3
1	1	0	6	1	0	0	4
1	1	1	7	1	0	1	5

Generating Kmaps For A, B, C outputs:



(a) Kmap for A output (b) Kmap for B output (c) Kmap for C output

Figure 1: Karnaugh maps for outputs A, B, and C

Calculating the minimized expressions for A, B, C outputs:

$$A = yz + xy \tag{1}$$
$$B = xy' + y'z + x'yz' \tag{2}$$
$$C = z' + xz \tag{3}$$

2. A majority circuit is a combinational circuit whose output is equal to 1 if the input variables have more 1's than 0's. The output is 0 otherwise. Design a 3-input majority circuit by finding the circuit's truth table, Boolean equation, and a logic diagram.

   Solution: The truth table for the given problem is shown below.

Table 2: Truth Table for the given problem

Index	x	y	z	Out
0	0	0	0	0
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

The kmap for the given problem is shown below:

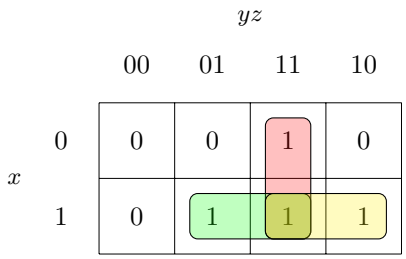


Figure 2: Kmap for the majority circuit

The minimized expression for the majority circuit is:

$$Out = yz + xz + xy \tag{4}$$

The logic diagram for the majority circuit is shown below:

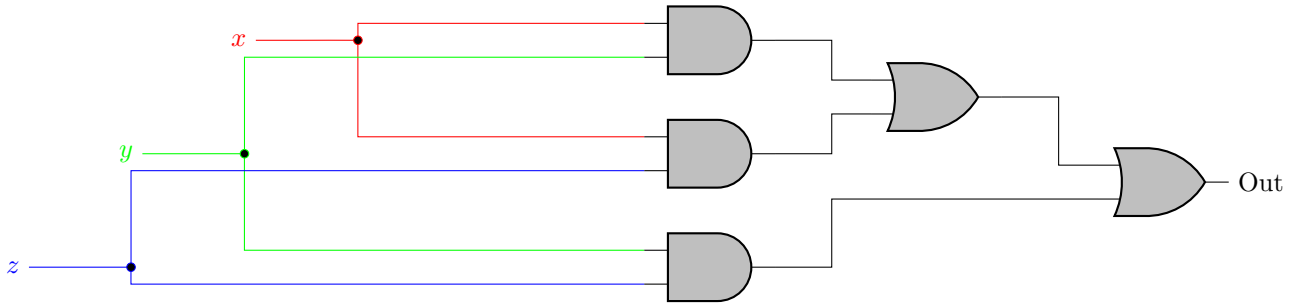


Figure 3: Logic diagram for the majority circuit

Part 2

1. We will design a circuit called half adder (HA) which adds two 1-bit numbers, a, b and produces 2-bit output, c.

   (a) Draw the truth table of the circuit.

   Solution: The truth table for the given problem is shown below.

   Table 3: Truth Table for the given problem (Part 2a)

a	b	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

   (b) Find the Boolean functions of each bit of the output.

   Solution: The Boolean functions for the given problem are shown below.

$$\text{Carry} = ab \tag{5}$$
$$\text{Sum} = a \oplus b \tag{6}$$

   It can be seen from the truth table that the carry bit is the AND of the inputs, and the sum bit is the XOR of the inputs.

   To obtain an XOR gate we can draw the Kmap for the sum bit:

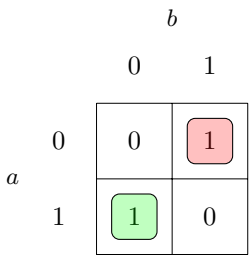


   Figure 4: Kmap for the sum bit

   The minimized expression for the sum bit is:

$$\text{Sum} = a'b + ab' \tag{7}$$

   Or we can simply use the XOR gate.

   (c) Optimize the Boolean functions of each bit of the output.

   Solution: The optimized Boolean functions for the given problem are shown below.

$$\text{Carry} = ab \tag{8}$$
$$\text{Sum} = a \oplus b \tag{9}$$

   The functions are already optimized.

(d) Draw the logic diagram of the circuit.

Solution: The logic diagram for the given problem is shown below.

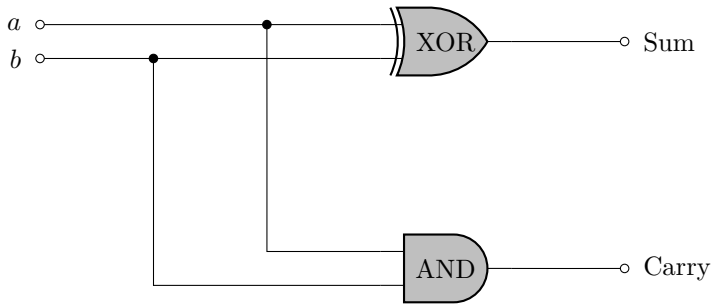


Figure 5: Logic diagram for the half-adder circuit

(e) Write the VHDL code of the logic diagrams by using "Dataflow modeling" method.

Solution: The VHDL code for the given problem is shown below.

Listing 1: half_adder.vhd

```
1  library IEEE;
2
3  use IEEE.STD_LOGIC_1164.ALL;
4
5  entity half_adder is
6      Port ( a : in  STD_LOGIC;
7             b : in  STD_LOGIC;
8             s : out  STD_LOGIC;
9             c : out  STD_LOGIC);
10 end half_adder;
11
12 architecture Behavioral of half_adder is
13 begin
14     s <= a xor b;
15     c <= a and b;
16 end Behavioral;
```

(f) Simulate the circuit that you have designed in 1.e. Prepare a simulation waveform for you report

Solution: The simulation waveform for the given problem is shown below.

TODO: Add the simulation waveform here.

(g) Produce the RTL schematic for the circuit that you have designed in 1.e.

Solution: The RTL schematic for the given problem is shown below.

TODO: Add the RTL schematic here.

2. We will design a circuit called full adder (FA) which adds three 1-bit numbers, a, b, c and produces 2-bit output, d.

   (a) Draw the truth table of the circuit.

Table 4: Truth Table for the given problem (Part 2.2.a)

a	b	Carry_In	Carry_Out	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

   (b) Find the Boolean functions of each bit of the output.

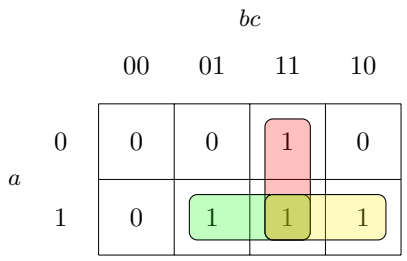
   Solution: The Boolean functions for the given problem are shown below.

$$\text{Carry\_Out} = a'bc + ab'c + abc' + abc \tag{10}$$
$$\text{Sum} = a'b'c + a'bc' + ab'c' + abc \tag{11}$$

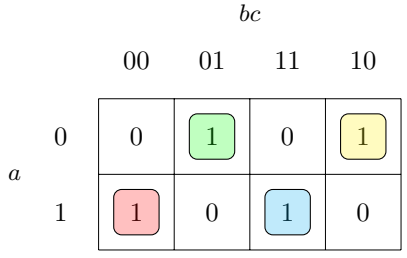
   (c) Optimize the Boolean functions of each bit of the output.

   Solution: The optimized Boolean functions for the given problem are shown below.

   Using Kmaps for the Carry_Out and Sum bits:



(a) Kmap for the Carry_Out bit



(b) Kmap for the Sum bit

Figure 6: Karnaugh maps for Carry_Out and Sum bits

   The optimized expressions for the Carry_Out and Sum bits are:

$$\text{Carry\_Out} = ac + bc + ab \tag{12}$$
$$\text{Sum} = ab'c' + a'b'c + abc + a'bc' \tag{13}$$

Also we can use the XOR and AND gates to implement the circuit.

$$\text{Carry\_Out} = ac + bc + ab \tag{14}$$
$$\text{Sum} = a \oplus b \oplus c \tag{15}$$

(d) Draw the logic diagram of the circuit.

Solution: The logic diagram for the given problem is shown below.

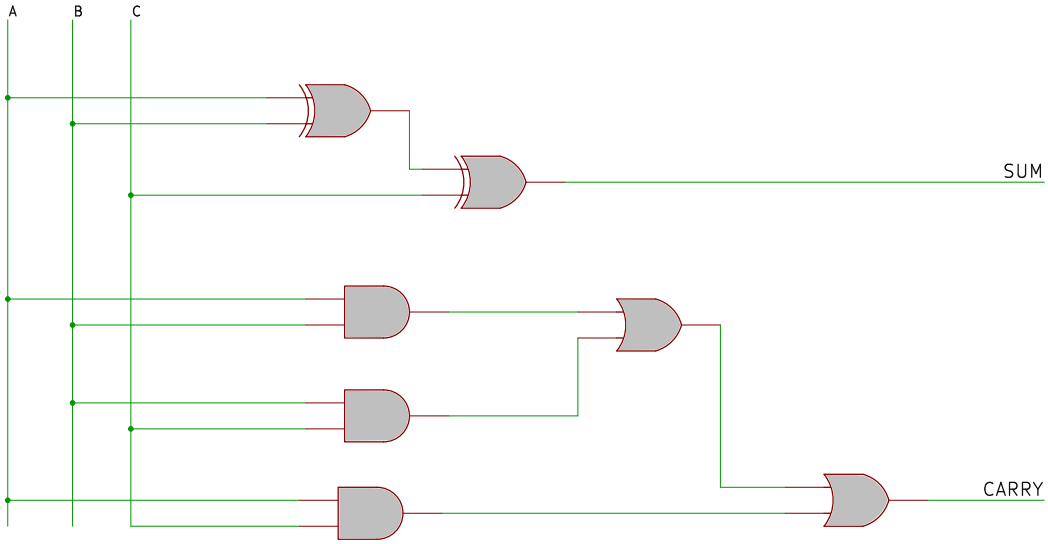


Figure 7: Logic diagram for the full-adder circuit

(e) e. Write the VHDL code of the logic diagrams by using "Dataflow modeling".

Solution: The VHDL code for the given problem is shown below.

Listing 2: full_adder.vhd

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity FullAdder is
    Port (
        a : in STD_LOGIC;
        b : in STD_LOGIC;
        c : in STD_LOGIC;
        Sum : out STD_LOGIC;
        Carry_Out : out STD_LOGIC
    );
end FullAdder;

architecture Behavioral of FullAdder is
begin
    Sum <= a XOR b XOR c;
    Carry_Out <= (a AND b) OR (b AND c) OR (a AND c);
end Behavioral;
```

(f) Simulate the circuit that you have designed in 2.e. Prepare a simulation waveform for your report.

Solution: The simulation waveform for the given problem is shown below.

[todo]

(g) Produce the RTL schematic for the circuit that you have designed in 2.e.

Solution: The RTL schematic for the given problem is shown below.

[todo]

3. We will design a circuit called ripple carry adder (RCA) which adds two 4-bit positive integers, A, B and produces 5-bit output, C.

   (a) Draw the logic diagram of the circuit by using one HA and 4 FAs.

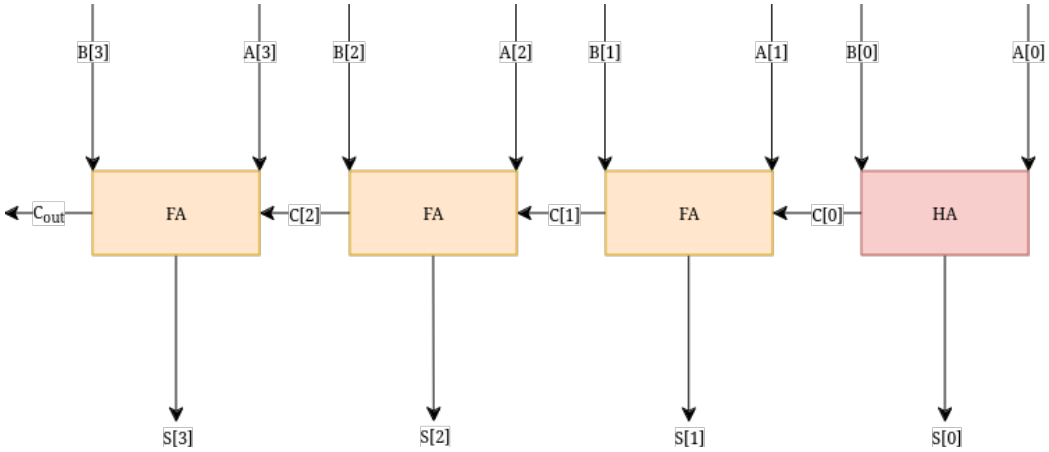


Figure 8: Logic diagram for the ripple carry adder circuit

   (b) Write the VHDL code of the logic diagrams by using “Structural modeling”

   Solution: The VHDL code for the given problem is shown below.

Listing 3: ripple_carry_adder.vhd

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity RippleCarryAdder is
5      Port (
6          A : in STD_LOGIC_VECTOR(3 downto 0);
7          B : in STD_LOGIC_VECTOR(3 downto 0);
8          Sum : out STD_LOGIC_VECTOR(3 downto 0);
9          Cout : out STD_LOGIC
10     );
11 end RippleCarryAdder;
12
13 architecture Structural of RippleCarryAdder is
14     component half_adder
15         Port (
16             a : in STD_LOGIC;
17             b : in STD_LOGIC;
18             s : out STD_LOGIC;
19             c : out STD_LOGIC
20         );
21     end component;
22
23     component FullAdder
24         Port (
25             a : in STD_LOGIC;
26             b : in STD_LOGIC;
27             c : in STD_LOGIC;
28             Sum : out STD_LOGIC;
29             Carry_Out : out STD_LOGIC
30         );
31     end component;
32
33     signal carry_internal : STD_LOGIC_VECTOR(3 downto 0);
```

```
34
35 begin
36     HA0: half_adder port map (
37         a => A(0),
38         b => B(0),
39         s => Sum(0),
40         c => carry_internal(0)
41     );
42
43     FA1: FullAdder port map (
44         a => A(1),
45         b => B(1),
46         c => carry_internal(0),
47         Sum => Sum(1),
48         Carry_Out => carry_internal(1)
49     );
50
51     FA2: FullAdder port map (
52         a => A(2),
53         b => B(2),
54         c => carry_internal(1),
55         Sum => Sum(2),
56         Carry_Out => carry_internal(2)
57     );
58
59     FA3: FullAdder port map (
60         a => A(3),
61         b => B(3),
62         c => carry_internal(2),
63         Sum => Sum(3),
64         Carry_Out => carry_internal(3)
65     );
66
67     Cout <= carry_internal(3);
68
69 end Structural;
```

(c) Produce the RTL schematic for the circuit that you have designed in 3.b.

(d) Simulate the circuit that you have designed in 3.b.