Introduction to Logic Design
EEF205E

Homework 2

Rüzgar Erik
040240783

Istanbul Technical University
Faculty of Electrical and Electronics Engineering

Part 1

1. Demonstrate the validity of the following identities by means of truth tables:

a. $((x+y+z)\,x\;y)' \;=\; x'y'z'+x+y$

Solution:

Table 1: Truth Table for the Function $f(x,y,z) = ((x+y+z)\,x\;y)'$

x	y	z	$x+y+z$	$(x+y+z)\cdot x\cdot y$	$f(x,y,z) = ((x+y+z)\cdot x\cdot y)'$
0	0	0	0	0	1
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	1	0	1
1	0	0	1	0	1
1	0	1	1	0	1
1	1	0	1	1	0
1	1	1	1	1	0

Calculating the right side of the equation as $g(x,y,z) = x'y'z'+x+y$:

Table 2: Truth Table for the Function $g(x,y,z) = x'y'z'+x+y$

x	y	z	x'	y'	z'	$x'y'z'$	$x+y$	$g(x,y,z) = x'y'z'+x+y$
0	0	0	1	1	1	1	0	1
0	0	1	1	1	0	0	0	0
0	1	0	1	0	1	0	1	1
0	1	1	1	0	0	0	1	1
1	0	0	0	1	1	0	1	1
1	0	1	0	1	0	0	1	1
1	1	0	0	0	1	0	1	1
1	1	1	0	0	0	0	1	1

As seen from the truth tables the given equation is not valid. The non-equal rows are colored red in the Table 3

Table 3: Combined Truth Table for $f(x,y,z) = ((x+y+z)\cdot x\cdot y)'$ and $g(x,y,z) = x'y'z'+x+y$

x	y	z	$f(x,y,z) = ((x+y+z)\cdot x\cdot y)'$	$g(x,y,z) = x'y'z'+x+y$
0	0	0	1	1
0	0	1	1	0
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	0	1
1	1	1	0	1

b. $x(x' + y + z) = xy + xz$

Solution:

Table 4: Combined Truth Table for $f(x, y, z)$, $g(x, y, z)$, and $h(x, y, z) = x(x' + y + z)$

x	y	z	x'	$x' + y + z$	$f(x, y, z) = x(x' + y + z)$	xy	xz	$g(x, y, z) = xy + xz$
0	0	0	1	1	0	0	0	0
0	0	1	1	1	0	0	0	0
0	1	0	1	1	0	0	0	0
0	1	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0	0
1	0	1	0	1	1	0	1	1
1	1	0	0	1	1	1	0	1
1	1	1	0	1	1	1	1	1

By comparing the truth tables of the functions $f(x, y, z) = x(x' + y + z)$ and $g(x, y, z) = xy + xz$ in Table 4, it can be seen that the given equation is valid.

2. Simplify the following Boolean expressions to a minimum number of literals:

a. $\mathbf{ABC + A'B + ABC'}$

Solution:

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

b. $\mathbf{(x + y)'\ (x' + y')}$

Solution:

$$\begin{aligned} (x + y)'\ (x' + y') &= x'y' \cdot (x' + y') \\ &= (x'y'x') + (x'y'y') \\ &= (x'y') + (x'y') \\ &= x'y' \end{aligned}$$

c. $\mathbf{xy + x\,(wz + wz')}$

Solution:

$$\begin{aligned} xy + x\,(wz + wz') &= xy + xw(z + z') \\ &= xy + xw \end{aligned}$$

d. $(\mathbf{a}' + \mathbf{c}')\,(\mathbf{a} + \mathbf{b}' + \mathbf{c}')$

Solution:

$$\begin{aligned}(a' + c')(a + b' + c') &= a(a' + c') + b'(a' + c') + c'(a' + c') \\ &= (aa' + ac') + (a'b' + b'c') + (c'a' + c'c') \\ &= (0 + ac') + (a'b' + b'c') + (a'c' + c') \\ &= ac' + a'b' + b'c' + a'c' + c' \qquad\qquad = c'(a + b' + a' + 1) + a'b' \\ &= c' + a'b'\end{aligned}$$

3. Boolean functions for the outputs $c_2(a_1, a_0, b_1, b_0)$, $c_1(a_1, a_0, b_1, b_0)$ and $c_0(a_1, a_0, b_1, b_0)$ shown by the below truth table will be implemented in Part 2.

Table 5: Truth Table for the Outputs c_2, c_1, and c_0

Index	a_1	a_0	b_1	b_0	c_2	c_1	c_0
0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	1
2	0	0	1	0	0	1	0
3	0	0	1	1	0	1	1
4	0	1	0	0	0	0	1
5	0	1	0	1	0	1	0
6	0	1	1	0	0	1	1
7	0	1	1	1	1	0	0
8	1	0	0	0	0	1	0
9	1	0	0	1	0	1	1
10	1	0	1	0	1	0	0
11	1	0	1	1	1	0	1
12	1	1	0	0	0	1	1
13	1	1	0	1	1	0	0
14	1	1	1	0	1	0	1
15	1	1	1	1	1	1	0

a. Find sum of products expressions for c_2, c_1, and c_0.

Solution:

i . SOP for c_2

Firstly the minterms for c_2 are found by looking at the rows where $c_2 = 1$. The minterms are 7, 10, 11, 13, 14, 15 The sum of products expression for c_2 is:

$$c_2 = \Sigma(7, 10, 11, 13, 14, 15) = a_1'a_0b_1b_0 + a_1a_0'b_1b_0' + a_1a_0'b_1b_0 + a_1a_0b_1'b_0 + a_1a_0b_1b_0' + a_1a_0b_1b_0$$

ii . SOP for c_1

The minterms for c_1 are 2, 3, 5, 6, 8, 9, 12, 15. The sum of products expression for c_1 is:

$$\begin{aligned} c_1 &= \Sigma(2,3,5,6,8,9,12,15) \\ &= a_1'a_0'b_1b_0' + a_1'a_0'b_1b_0 + a_1'a_0b_1'b_0 + a_1'a_0b_1b_0' \\ &\quad + a_1a_0'b_1'b_0' + a_1a_0'b_1'b_0 + a_1a_0b_1'b_0' + a_1a_0b_1b_0 \end{aligned}$$

iii . SOP for c_0

The minterms for c_0 are 1, 3, 4, 6, 9, 11, 12, 14. The sum of products expression for c_0 is:

$$\begin{aligned} c_0 &= \Sigma(1,3,4,6,9,11,12,14) \\ &= a_1'a_0'b_1'b_0 + a_1'a_0'b_1b_0 + a_1'a_0b_1'b_0' + a_1'a_0b_1b_0' \\ &\quad + a_1a_0'b_1'b_0 + a_1a_0'b_1b_0 + a_1a_0b_1'b_0' + a_1a_0b_1b_0' \end{aligned}$$

b. Simplify the sum of products expressions for c_2, c_1, and c_0.

Solution:

i . Solving for c_2

Karnaugh map method was chosen to simplify the sum of products expression for all of the questions. The Karnaugh map for c_2 is shown in Figure 1.

a_1a_0 \ b_1b_0	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	1	1	1
10	0	0	1	1

Figure 1: Karnaugh Map for c_2

The simplified sum of products expression for c_2 is: $c_2 = a_0b_1b_0 + a_1b_1 + a_1a_0b_0$

ii . Solving for c_1

The Karnaugh map for c_1 is shown in Figure 2.

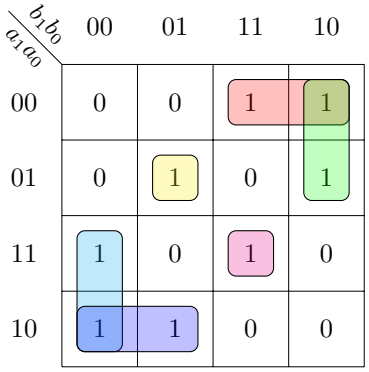

Figure 2: Karnaugh Map for c_1

The simplified sum of products expression for c_1 is: $c_1 = a_1'a_0'b_1 + a_1'a_0b_1'b_0 + a_1'b_1b_0' + a_1a_0'b_1' + a_1b_1'b_0' + a_1a_0b_1b_0$

iii . Solving for c_0

The Karnaugh map for c_0 is shown in Figure 3.

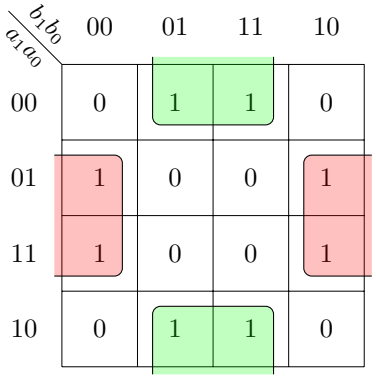

Figure 3: Karnaugh Map for c_0

The simplified sum of products expression for c_0 is: $c_0 = a_0'b_0 + a_0b_0'$

Part 2

Boolean Function Case Statement

Using the truth table a behavioral model for the circuit is implemented in VHDL. The code is shown below:

Listing 1: Boolean_Function_Case_Statement.vhd

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

entity Boolean_Function_Case_Statement is
    Port ( a1, a0, b1, b0 : in STD_LOGIC;
           c2, c1, c0 : out STD_LOGIC);
end Boolean_Function_Case_Statement;


architecture Behavioral of Boolean_Function_Case_Statement is
    signal input_vector : STD_LOGIC_VECTOR(3 downto 0);
begin

    input_vector <= a1 & a0 & b1 & b0;

    process(input_vector)
    begin
        case input_vector is
            when "0000" => c2 <= '0'; c1 <= '0'; c0 <= '0';
            when "0001" => c2 <= '0'; c1 <= '0'; c0 <= '1';
            when "0010" => c2 <= '0'; c1 <= '1'; c0 <= '0';
            when "0011" => c2 <= '0'; c1 <= '1'; c0 <= '1';
            when "0100" => c2 <= '0'; c1 <= '0'; c0 <= '1';
            when "0101" => c2 <= '0'; c1 <= '1'; c0 <= '0';
            when "0110" => c2 <= '0'; c1 <= '1'; c0 <= '1';
            when "0111" => c2 <= '1'; c1 <= '0'; c0 <= '0';
            when "1000" => c2 <= '0'; c1 <= '1'; c0 <= '0';
            when "1001" => c2 <= '0'; c1 <= '1'; c0 <= '1';
            when "1010" => c2 <= '1'; c1 <= '0'; c0 <= '0';
            when "1011" => c2 <= '1'; c1 <= '0'; c0 <= '1';
            when "1100" => c2 <= '0'; c1 <= '1'; c0 <= '1';
            when "1101" => c2 <= '1'; c1 <= '0'; c0 <= '0';
            when "1110" => c2 <= '1'; c1 <= '0'; c0 <= '1';
            when "1111" => c2 <= '1'; c1 <= '1'; c0 <= '0';
            when others => c2 <= '0'; c1 <= '0'; c0 <= '0';
        end case;
    end process;

end Behavioral;
```

Testbench for the Boolean Function Case Statement

The testbench for the behavioral model is implemented in VHDL. The code is shown below:

Listing 2: Boolean_Function_Case_Statement_tb.vhd

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

ENTITY Boolean_Function_Case_Statement_tb IS
```

```
END Boolean_Function_Case_Statement_tb;

ARCHITECTURE behavior OF Boolean_Function_Case_Statement_tb IS
    COMPONENT Boolean_Function_Case_Statement
        PORT(
            a1, a0, b1, b0 : in STD_LOGIC;
            c2, c1, c0 : out STD_LOGIC
        );
    END COMPONENT;

    signal a1_tb, a0_tb, b1_tb, b0_tb: STD_LOGIC := '0';
    signal c2_tb, c1_tb, c0_tb : STD_LOGIC;
    signal input_vector_tb : STD_LOGIC_VECTOR(3 downto 0) := "0000";


BEGIN
    UUT: Boolean_Function_Case_Statement PORT MAP (
        a1 => a1_tb,
        a0 => a0_tb,
        b1 => b1_tb,
        b0 => b0_tb,
        c2 => c2_tb,
        c1 => c1_tb,
        c0 => c0_tb
    );

    stim_proc: process
    begin
        for i in 0 to 15 loop
            input_vector_tb <= std_logic_vector(to_unsigned(i, 4));
            wait for 1 ns;
            a1_tb <= input_vector_tb(3);
            a0_tb <= input_vector_tb(2);
            b1_tb <= input_vector_tb(1);
            b0_tb <= input_vector_tb(0);

            wait for 10 ns;

        end loop;

        wait;
    end process;

END behavior;
```

RTL Schematic of the Boolean Function Case Statement

The RTL schematic of the behavioral model is shown in Figure 4.

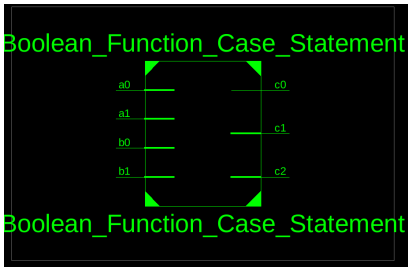

Figure 4: RTL Schematic of the Boolean Function Case Statement

Going one more layer deep, the schematic of the individual components of the behavioral model is shown in Figure 5.

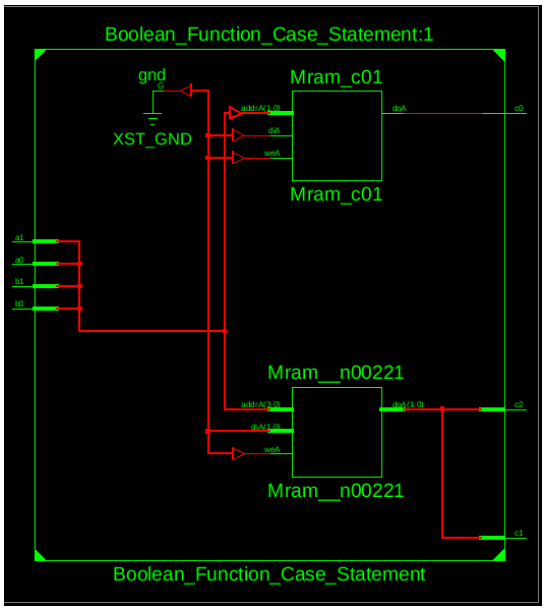

Figure 5: RTL Schematic of the Components of the Boolean Function Case Statement

Simulation Results

The simulation results of the testbench for the behavioral model is shown in Figure 6.

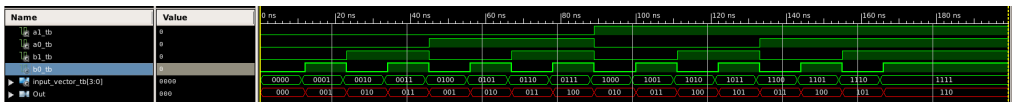

Figure 6: Simulation Results of the Testbench for the Boolean Function Case Statement

It can be seen that from inputs at the bottom of the waveform, the outputs c_2, c_1, and c_0 are as expected from the truth table in Table 5.

Dataflow Model

The dataflow model for the circuit is implemented in VHDL. The code is shown below:

Listing 3: Boolean_Function_Data_Flow.vhd

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

entity Boolean_Function_Dataflow is
    Port ( a1, a0, b1, b0 : in STD_LOGIC;
           c2, c1, c0 : out STD_LOGIC);
end Boolean_Function_Dataflow;

architecture Dataflow of Boolean_Function_Dataflow is

begin
    c2 <= (a1 and a0 and b0) or
          (a0 and b1 and b0) or
          (a1 and b1);

    c1 <= (not a1 and not a0 and b1) or
          (not a1 and a0 and not b1 and b0) or
          (not a1 and b1 and not b0) or
          (a1 and not a0 and not b1) or
          (a1 and not b1 and not b0) or
          (a1 and a0 and b1 and b0);

    c0 <= (not a0 and b0) or
          (a0 and not b0);

end Dataflow;
```

Testbench for the Boolean Function Data Flow

The testbench for the dataflow model is implemented in VHDL. The code is shown below:

Listing 4: Boolean_Function_Data_Flow_tb.vhd

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

ENTITY Boolean_Function_Dataflow_tb IS
END Boolean_Function_Dataflow_tb;

ARCHITECTURE behavior OF Boolean_Function_Dataflow_tb IS
    COMPONENT Boolean_Function_Dataflow
        PORT(
            a1, a0, b1, b0 : in STD_LOGIC;
            c2, c1, c0 : out STD_LOGIC
        );
    END COMPONENT;

    signal a1_tb, a0_tb, b1_tb, b0_tb: STD_LOGIC := '0';
    signal c2_tb, c1_tb, c0_tb : STD_LOGIC;
    signal input_vector_tb : STD_LOGIC_VECTOR(3 downto 0) := "0000";

BEGIN
```

```
    UUT: Boolean_Function_Dataflow PORT MAP (
        a1 => a1_tb,
        a0 => a0_tb,
        b1 => b1_tb,
        b0 => b0_tb,
        c2 => c2_tb,
        c1 => c1_tb,
        c0 => c0_tb
    );

    stim_proc: process
    begin
        for i in 0 to 15 loop
            input_vector_tb <= std_logic_vector(to_unsigned(i, 4));
            wait for 1 ns;
            a1_tb <= input_vector_tb(3);
            a0_tb <= input_vector_tb(2);
            b1_tb <= input_vector_tb(1);
            b0_tb <= input_vector_tb(0);

            wait for 10 ns;
        end loop;
        wait;
    end process;
END behavior;
```

0.0.1 RTL Schematic of the Boolean Function Data Flow

The RTL schematic of the dataflow model is shown in Figure 7.

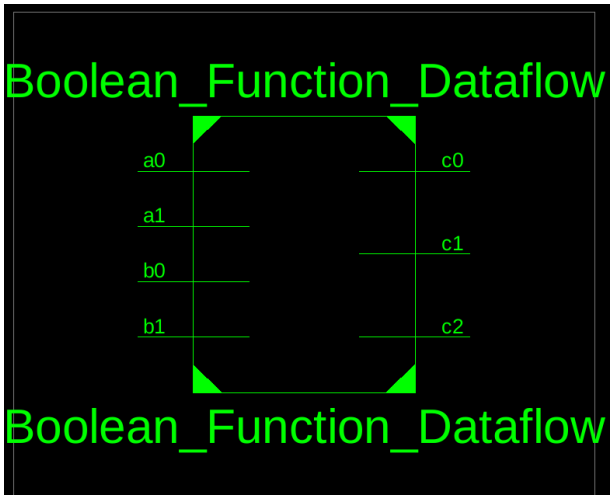

Figure 7: RTL Schematic of the Boolean Function Data Flow

The schematic of the individual components of the dataflow model is shown in Figure 8.

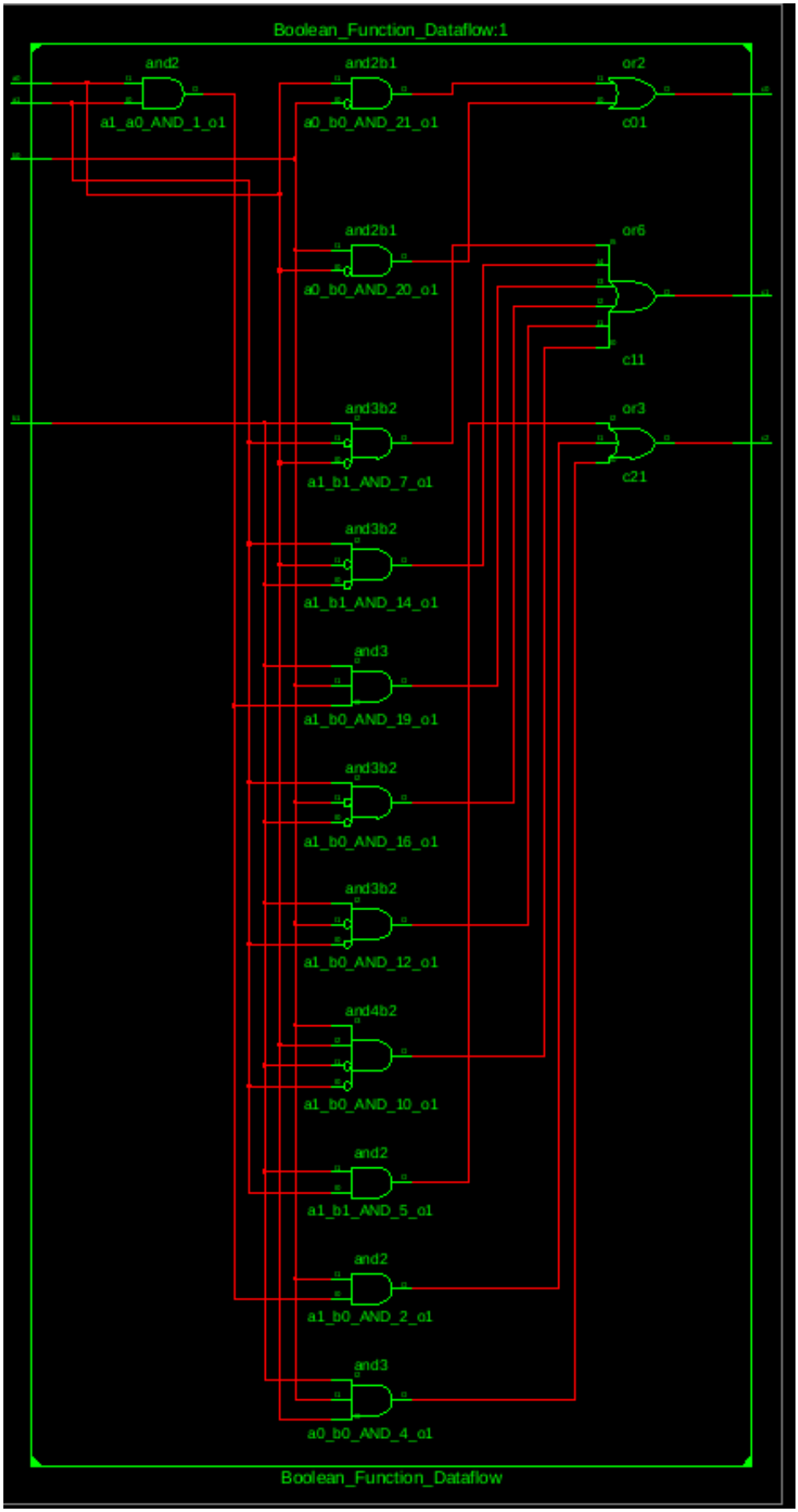

Figure 8: RTL Schematic of the Components of the Boolean Function Data Flow

Simulation Results

The simulation results of the testbench for the dataflow model is shown in Figure 9.

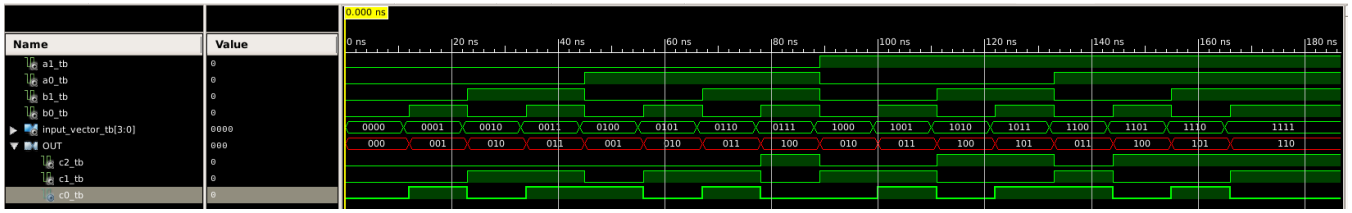

Figure 9: Simulation Results of the Testbench for the Boolean Function Data Flow

It can be seen that from inputs at the bottom of the waveform, the outputs c_2, c_1, and c_0 are as expected from the truth table in Table 5.