ECE 4300
Discrete System Design Using VHDL

Midterm 1

Choi Tim Antony Yung (ID:013499993)

March 10, 2021

1 Combinational Circuit

Code

D Flip-flop

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY DFlipflop IS
    PORT (
        D : IN STD_LOGIC;
        clk : IN STD_LOGIC;
        rst : IN STD_LOGIC;
        Q : OUT STD_LOGIC
    );
END DFlipflop;

ARCHITECTURE Behavioral OF DFlipflop IS
BEGIN

    PROCESS (clk, rst)
    BEGIN
        IF (rst = '1') THEN
                Q <= '0';
        ELSIF (rising_edge(clk)) THEN
                Q <= D;
        END IF;
    END PROCESS;

END Behavioral;
```

Combinational Circuit

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY ECE4304_Midterm1_1 IS
    PORT (
        top_clk : IN STD_LOGIC;
        top_rst : IN STD_LOGIC;
        A : IN STD_LOGIC;
        B : IN STD_LOGIC;
        C : IN STD_LOGIC;
        D : IN STD_LOGIC;
        Z : OUT STD_LOGIC
    );
END ECE4304_Midterm1_1;

ARCHITECTURE Structural OF ECE4304_Midterm1_1 IS

    COMPONENT DFlipflop IS
        PORT (
```

```
            D : IN STD_LOGIC;
            clk : IN STD_LOGIC;
            rst : IN STD_LOGIC;
            Q : OUT STD_LOGIC
        );
    END COMPONENT;

    SIGNAL E : STD_LOGIC;
    SIGNAL E_Q : STD_LOGIC;
    SIGNAL F : STD_LOGIC;
    SIGNAL F_Q : STD_LOGIC;

BEGIN

    E <= (A AND B AND C) OR D;
    F <= A NAND (B NOR C);

    EFF : DFlipflop
    PORT MAP(
        D => E,
        clk => top_clk,
        rst => top_rst,
        Q => E_Q
    );

    FFF : DFlipflop
    PORT MAP(
        D => F,
        clk => top_clk,
        rst => top_rst,
        Q => F_Q
    );

    Z <= E_Q XOR (NOT F_Q);

END Structural;
```

Testbench

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.math_real.ALL;
USE IEEE.numeric_std.ALL;

ENTITY ECE4304_Midterm1_1_tb IS
    --  Port ( );
END ECE4304_Midterm1_1_tb;

ARCHITECTURE Behavioral OF ECE4304_Midterm1_1_tb IS

    CONSTANT clock_period : TIME := 10 ns;
```

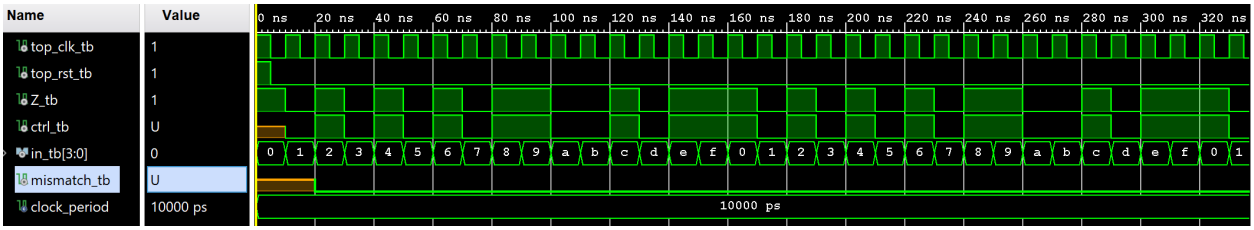
```
    SIGNAL top_clk_tb : STD_LOGIC;
    SIGNAL top_rst_tb : STD_LOGIC;
    SIGNAL Z_tb : STD_LOGIC;
    SIGNAL ctrl_tb : STD_LOGIC;

    SIGNAL in_tb : STD_LOGIC_VECTOR(3 DOWNTO 0);
    SIGNAL mismatch_tb : STD_LOGIC;

    COMPONENT ECE4304_Midterm1_1 IS
        PORT (
            top_clk : IN STD_LOGIC;
            top_rst : IN STD_LOGIC;
            A : IN STD_LOGIC;
            B : IN STD_LOGIC;
            C : IN STD_LOGIC;
            D : IN STD_LOGIC;
            Z : OUT STD_LOGIC
        );
    END COMPONENT;

BEGIN

    UUT : ECE4304_Midterm1_1
    PORT MAP(
        top_clk => top_clk_tb,
        top_rst => top_rst_tb,
        A => in_tb(3),
        B => in_tb(2),
        C => in_tb(1),
        D => in_tb(0),
        Z => Z_tb
    );

    CLK_GEN : PROCESS
    BEGIN
        top_clk_tb <= '1';
        WAIT FOR 0.5 * clock_period;
        top_clk_tb <= '0';
        WAIT FOR 0.5 * clock_period;
    END PROCESS;

    INIT_RST : PROCESS
    BEGIN
        top_rst_tb <= '1';
        WAIT FOR 0.5 * clock_period;
        top_rst_tb <= '0';
        WAIT;
    END PROCESS;

    TEST_MAIN : PROCESS
    BEGIN
        FOR i IN 0 TO 2 ** 4 - 1 LOOP -- all possible combination of 4 input
            in_tb <= STD_LOGIC_VECTOR(to_unsigned(i, 4));
            ctrl_tb <= ((in_tb(3) AND in_tb(2) AND in_tb(1)) OR in_tb(0)) XOR (in_tb(3) AN
```

```
            mismatch_tb <= Z_tb XOR ctrl_tb; -- can be found using "Next Transition" buttor
            WAIT FOR clock_period;
        END LOOP;
    END PROCESS;

END Behavioral;
```

Simulation Result

As can be seen in figure 1, all possible input combination was simulated and the circuit behave as expected with no mismatch found.

Figure 1: Simulation Waveform of the Circuit

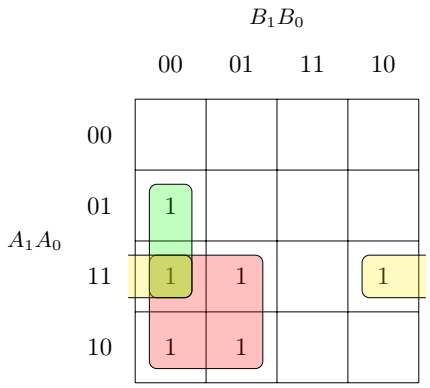


2 2-bit Comparator

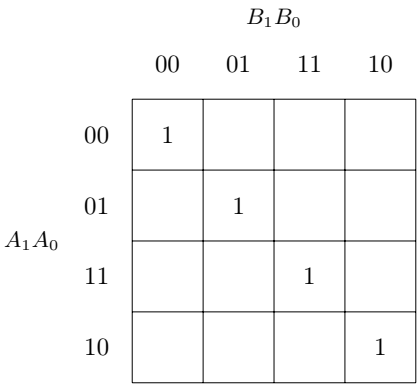
Truth Table

A_1	A_0	B_1	B_0	$A > B$	$A = B$	$A < B$
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

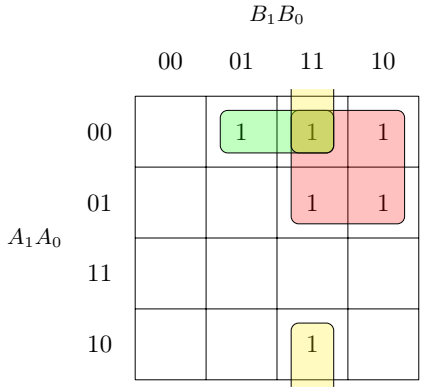
Karnaugh Maps



$$(A > B) = A_1\overline{B_1} + A_0\overline{B_1}\,\overline{B_0} + A_1A_0\overline{B_0}$$



$$\begin{aligned}(A = B) &= \overline{A_1}\,\overline{A_0}\,\overline{B_1}\,\overline{B_0} + \overline{A_1}A_0\overline{B_1}B_0 + A_1\overline{A_0}B_1\overline{B_0} + A_1A_0B_1B_0 \\ &= (A_1 \oplus B_1)(A_0 \oplus B_0)\end{aligned}$$



$$\begin{aligned}(A < B) &= \overline{A_1}B_0 + \overline{A_1}\,\overline{A_0}B_0 + \overline{A_0}B_1B_0 \\ &= \overline{(A > B) + (A = B)}\end{aligned}$$

Code

Comparator Circuit

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY ECE4304_Midterm1_2 IS
    PORT (
        A : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
        B : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
        AGB : OUT STD_LOGIC; -- A>B
        AEB : OUT STD_LOGIC; -- A=B
        ALB : OUT STD_LOGIC -- A<B
    );
END ECE4304_Midterm1_2;

ARCHITECTURE Structural OF ECE4304_Midterm1_2 IS

    SIGNAL AGB_temp : STD_LOGIC;
    SIGNAL AEB_temp : STD_LOGIC;

BEGIN

    AGB_temp <= (A(1) AND (NOT B(1))) OR
        (A(0) AND (NOT B(1)) AND (NOT B(0))) OR
        (A(1) AND A(0) AND (NOT B(0)));
    AEB_temp <= (A(1) XNOR B(1)) AND (A(0) XNOR B(0));
    AGB <= AGB_temp;
    AEB <= AEB_temp;
    ALB <= AGB_temp NOR AEB_temp;

END Structural;
```

Testbench

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.math_real.ALL;
USE IEEE.numeric_std.ALL;

ENTITY ECE4304_Midterm1_2_tb IS
    --  Port ( );
END ECE4304_Midterm1_2_tb;

ARCHITECTURE Behavioral OF ECE4304_Midterm1_2_tb IS

    CONSTANT clock_period : TIME := 10 ns;
    SIGNAL top_clk_tb : STD_LOGIC;
    SIGNAL top_rst_tb : STD_LOGIC;
    SIGNAL AGB_tb : STD_LOGIC;
```

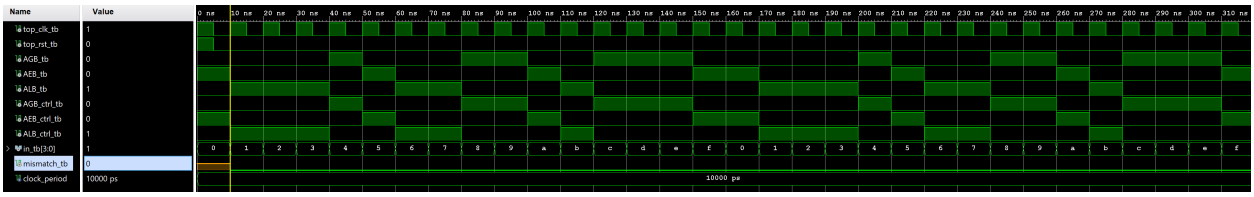
```
    SIGNAL AEB_tb : STD_LOGIC;
    SIGNAL ALB_tb : STD_LOGIC;
    SIGNAL AGB_ctrl_tb : STD_LOGIC;
    SIGNAL AEB_ctrl_tb : STD_LOGIC;
    SIGNAL ALB_ctrl_tb : STD_LOGIC;

    SIGNAL in_tb : STD_LOGIC_VECTOR(3 DOWNTO 0);
    SIGNAL mismatch_tb : STD_LOGIC;

    COMPONENT ECE4304_Midterm1_2 IS
        PORT (
            A : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
            B : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
            AGB : OUT STD_LOGIC; -- A>B
            AEB : OUT STD_LOGIC; -- A=B
            ALB : OUT STD_LOGIC -- A<B
        );
    END COMPONENT;

BEGIN

    UUT : ECE4304_Midterm1_2
    PORT MAP(
        A => in_tb(3 DOWNTO 2),
        B => in_tb(1 DOWNTO 0),
        AGB => AGB_tb,
        AEB => AEB_tb,
        ALB => ALB_tb
    );

    CLK_GEN : PROCESS
    BEGIN
        top_clk_tb <= '1';
        WAIT FOR 0.5 * clock_period;
        top_clk_tb <= '0';
        WAIT FOR 0.5 * clock_period;
    END PROCESS;

    INIT_RST : PROCESS
    BEGIN
        top_rst_tb <= '1';
        WAIT FOR 0.5 * clock_period;
        top_rst_tb <= '0';
        WAIT;
    END PROCESS;

    TEST_MAIN : PROCESS
        VARIABLE A_int : INTEGER;
        VARIABLE B_int : INTEGER;
    BEGIN
        FOR i IN 0 TO 2 ** 4 - 1 LOOP -- all possible combination of 4 input
            in_tb <= STD_LOGIC_VECTOR(to_unsigned(i, 4));
            A_int := i/4;
            B_int := i MOD 4;
```

```
            IF (A_int > B_int) THEN
                AGB_ctrl_tb <= '1';
            ELSE
                AGB_ctrl_tb <= '0';
            END IF;
            IF (A_int = B_int) THEN
                AEB_ctrl_tb <= '1';
            ELSE
                AEB_ctrl_tb <= '0';
            END IF;
            IF (A_int < B_int) THEN
                ALB_ctrl_tb <= '1';
            ELSE
                ALB_ctrl_tb <= '0';
            END IF;
            mismatch_tb <= (AGB_ctrl_tb XOR AGB_tb) OR
                (AEB_ctrl_tb XOR AEB_tb) OR
                (ALB_ctrl_tb XOR ALB_tb);
            -- can be found using "Next Transition" button in sim
            WAIT FOR clock_period;
        END LOOP;
    END PROCESS;

END Behavioral;
```

Simulation Result

As can be seen in figure 2, all possible input combination was simulated and the circuit behave as expected with no mismatch found.

Figure 2: Simulation Waveform of the Circuit



3 Clock Divider

Code

D Flip-flop

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY DFlipflop IS
    PORT (
        D : IN STD_LOGIC;
        clk : IN STD_LOGIC;
```

```
        rst : IN STD_LOGIC;
        Q : OUT STD_LOGIC
    );
END DFlipflop;

ARCHITECTURE Behavioral OF DFlipflop IS
BEGIN

    PROCESS (clk, rst)
    BEGIN
        IF (rst = '1') THEN
                Q <= '0';
        ELSIF (rising_edge(clk)) THEN
                Q <= D;
        END IF;
    END PROCESS;

END Behavioral;
```

Clock Divider Circuit

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY ECE4304_Midterm1_3 IS
    PORT (
        clk200MHZ : IN STD_LOGIC;
        top_rst : IN STD_LOGIC;
        clk1M563HZ : OUT STD_LOGIC
    );
END ECE4304_Midterm1_3;

ARCHITECTURE Structural OF ECE4304_Midterm1_3 IS

    -- upcounter as clock divider circuit
    -- each flip-flop divide clock by 2
    -- 2^n * 1.563 = 200
    -- with n being the number of flip-flop needed
    -- log(200/1.563)/log(2) = log(128)/log(2) = 7
    -- 7 flip-flops needed

    COMPONENT DFlipflop IS
        PORT (
            D : IN STD_LOGIC;
            clk : IN STD_LOGIC;
            rst : IN STD_LOGIC;
            Q : OUT STD_LOGIC
        );
    END COMPONENT;

    CONSTANT N : INTEGER := 7;
    SIGNAL top_Q : STD_LOGIC_VECTOR(N - 1 DOWNTO 0);
    SIGNAL top_notQ : STD_LOGIC_VECTOR(N - 1 DOWNTO 0);
```

```
    SIGNAL top_clk_FF : STD_LOGIC_VECTOR(N - 1 DOWNTO 0);

BEGIN

    top_notQ <= NOT top_Q;
    top_clk_FF <= top_notQ(N - 2 DOWNTO 0) & clk200MHZ; -- carry when falling edge
    clk1M563HZ <= top_Q(N - 1);
    DFF_GEN : FOR i IN 0 TO N - 1 GENERATE
    BEGIN
        DFF : DFlipflop
        PORT MAP(
            D => top_notQ(i), -- toggle each clock
            clk => top_clk_FF(i),
            rst => top_rst,
            Q => top_Q(i)
        );
    END GENERATE;

END Structural;
```

Testbench

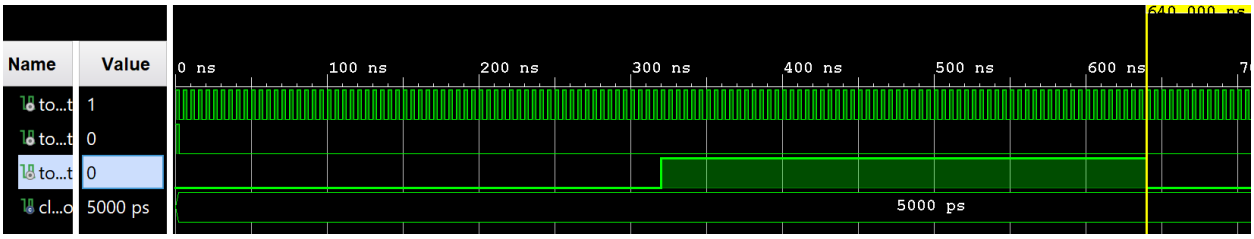
```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.math_real.ALL;
USE IEEE.numeric_std.ALL;

ENTITY ECE4304_Midterm1_3_tb IS
    --  Port ( );
END ECE4304_Midterm1_3_tb;

ARCHITECTURE Behavioral OF ECE4304_Midterm1_3_tb IS

    CONSTANT clock_period : TIME := 5 ns; -- 1/200MHZ is 5 ns
    SIGNAL top_clk_tb : STD_LOGIC;
    SIGNAL top_rst_tb : STD_LOGIC;
    SIGNAL top_clk_out_tb : STD_LOGIC;

    COMPONENT ECE4304_Midterm1_3 IS
        PORT (
            clk200MHZ : IN STD_LOGIC;
            top_rst : IN STD_LOGIC;
            clk1M563HZ : OUT STD_LOGIC
        );
    END COMPONENT;

BEGIN

    UUT : ECE4304_Midterm1_3
    PORT MAP(
        clk200MHZ => top_clk_tb,
```

```
        top_rst => top_rst_tb,
        clk1M563HZ => top_clk_out_tb
    );

    CLK_GEN : PROCESS
    BEGIN
        top_clk_tb <= '1';
        WAIT FOR 0.5 * clock_period;
        top_clk_tb <= '0';
        WAIT FOR 0.5 * clock_period;
    END PROCESS;

    INIT_RST : PROCESS
    BEGIN
        top_rst_tb <= '1';
        WAIT FOR 0.5 * clock_period;
        top_rst_tb <= '0';
        WAIT;
    END PROCESS;

END Behavioral;
```

Simulation Result

As can be seen in figure 3, the generated clock have a frequency of $f = \frac{1}{640\,\text{ns}} = 1.5625\,\text{MHz} \approx 1.563\,\text{MHz}$

Figure 3: Simulation Waveform of the Circuit

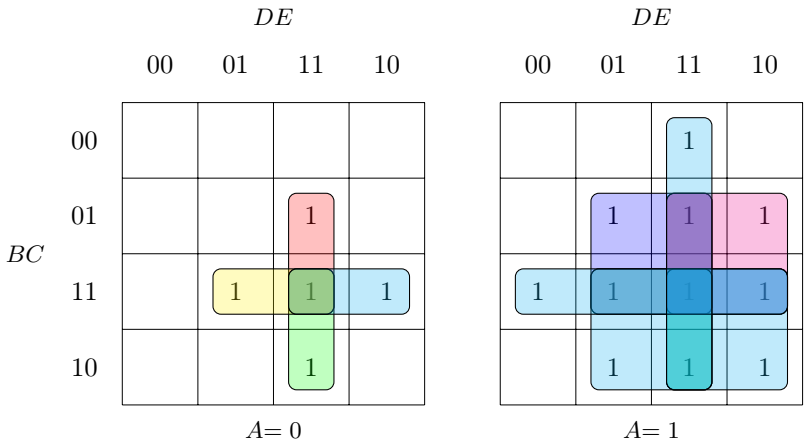


4 Majority

Truth Table

A	B	C	D	E	maj
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	0	1	1
1	0	1	1	0	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

Karnaugh Maps



function

$$maj(A,B,C,D,E) = CDE + BDE + BCD + BCE + ACE + ACD + ABE + ABD + ABC + ADE$$

Code

Majority Circuit

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY ECE4304_Midterm1_4 IS
    PORT (
        A : IN STD_LOGIC;
        B : IN STD_LOGIC;
        C : IN STD_LOGIC;
        D : IN STD_LOGIC;
        E : IN STD_LOGIC;
        maj : OUT STD_LOGIC
    );
END ECE4304_Midterm1_4;

ARCHITECTURE Structural OF ECE4304_Midterm1_4 IS

    -- 5 input majority circuit output 1 when
    -- there are >= 3 input on high
    -- we can implement this as sum of product of
    -- all possible 3 input combinations since
    -- if there is no majority none of the products
    -- will output high, whereas if there is a majority
    -- at least three of the inputs will be high and will
    -- trigger at least one of the products

BEGIN

    maj <= (A AND B AND C) OR
        (A AND B AND D) OR
        (A AND B AND E) OR
        (A AND C AND D) OR
        (A AND C AND E) OR
        (A AND D AND E) OR
        (B AND C AND D) OR
        (B AND C AND E) OR
        (B AND D AND E) OR
        (C AND D AND E);

END Structural;
```

Testbench

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.math_real.ALL;
USE IEEE.numeric_std.ALL;
```

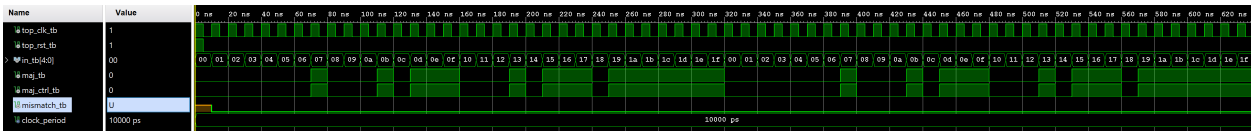
```vhdl
ENTITY ECE4304_Midterm1_4_tb IS
    --  Port ( );
END ECE4304_Midterm1_4_tb;

ARCHITECTURE Behavioral OF ECE4304_Midterm1_4_tb IS

    CONSTANT clock_period : TIME := 10 ns;
    SIGNAL top_clk_tb : STD_LOGIC;
    SIGNAL top_rst_tb : STD_LOGIC;

    SIGNAL in_tb : STD_LOGIC_VECTOR(4 DOWNTO 0);
    signal maj_tb : std_logic;
    signal maj_ctrl_tb : std_logic;
    SIGNAL mismatch_tb : STD_LOGIC;

    COMPONENT ECE4304_Midterm1_4 IS
        PORT (
            A : IN STD_LOGIC;
            B : IN STD_LOGIC;
            C : IN STD_LOGIC;
            D : IN STD_LOGIC;
            E : IN STD_LOGIC;
            maj : OUT STD_LOGIC
        );
    END COMPONENT;

BEGIN

    UUT : ECE4304_Midterm1_4
    PORT MAP(
        A => in_tb(4),
        B => in_tb(3),
        C => in_tb(2),
        D => in_tb(1),
        E => in_tb(0),
        maj => maj_tb
    );

    CLK_GEN : PROCESS
    BEGIN
        top_clk_tb <= '1';
        WAIT FOR 0.5 * clock_period;
        top_clk_tb <= '0';
        WAIT FOR 0.5 * clock_period;
    END PROCESS;

    INIT_RST : PROCESS
    BEGIN
        top_rst_tb <= '1';
        WAIT FOR 0.5 * clock_period;
        top_rst_tb <= '0';
        WAIT;
    END PROCESS;
```

```
    TEST_MAIN : PROCESS
        variable high_num: integer;
    BEGIN
        FOR i IN 0 TO 2 ** 5 - 1 LOOP -- all possible combination of 4 input
            in_tb <= STD_LOGIC_VECTOR(to_unsigned(i, 5));
            high_num := i/2**4 + ((i/2**3) mod 2)+ ((i/2**2) mod 2)+ ((i/2**1) mod 2)+ (i r
            if (high_num > 2) then maj_ctrl_tb <= '1'; else maj_ctrl_tb <= '0'; end if;
            mismatch_tb <= maj_tb xor maj_ctrl_tb;
            -- can be found using "Next Transition" button in sim
            WAIT FOR clock_period;
        END LOOP;
    END PROCESS;

END Behavioral;
```

Simulation Result

As can be seen in figure 4, all possible input combination was simulated and the circuit behave as expected with no mismatch found.

Figure 4: Simulation Waveform of the Circuit

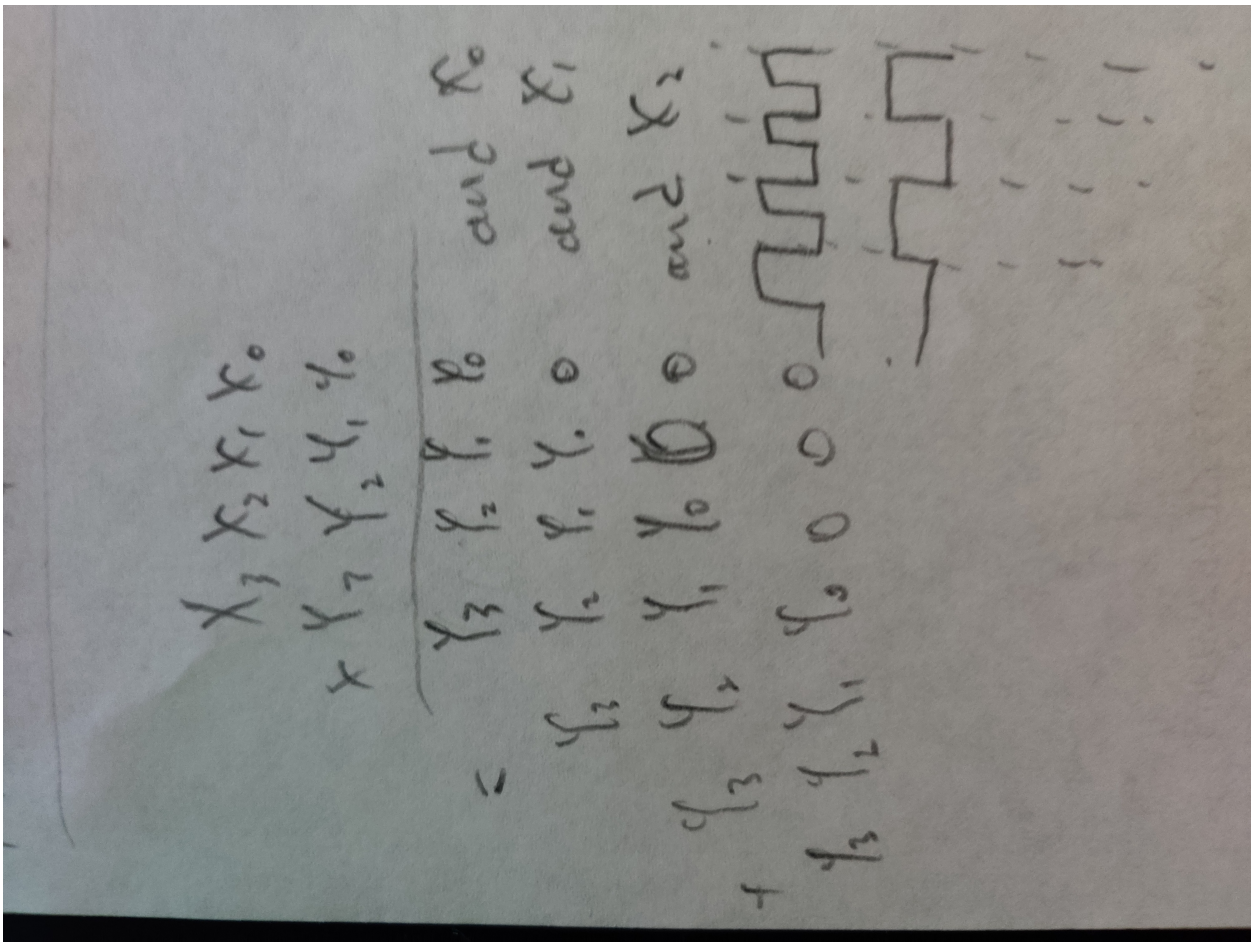


5 N-bit Generic Multiplier

5.a Theory

Full Adder can be used to compose a multiplier as seen in 5

Figure 5: Sketch



Code

N-bit Full Adder

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY NFA IS
    GENERIC (N : INTEGER := 1);
    PORT (
        cin : IN STD_LOGIC;
        X : IN STD_LOGIC_VECTOR(N - 1 DOWNTO 0);
        Y : IN STD_LOGIC_VECTOR(N - 1 DOWNTO 0);
        S : OUT STD_LOGIC_VECTOR(N - 1 DOWNTO 0);
        cout : OUT STD_LOGIC
```

```
    );
END NFA;

ARCHITECTURE Structural OF NFA IS

    SIGNAL C : STD_LOGIC_VECTOR(N DOWNTO 0);

BEGIN

    cout <= C(N);
    C(0) <= cin;
    FA_GEN : FOR i IN 0 TO N - 1 GENERATE
        C(i + 1) <= (X(i) AND Y(i)) OR (X(i) AND C(i)) OR (Y(i) AND C(i));
        S(i) <= X(i) XOR Y(i) XOR C(i);
    END GENERATE;

END Structural;
```

Majority Circuit

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY ECE4304_Midterm1_5 IS
    GENERIC (N : INTEGER := 3);
    PORT (
        X : IN STD_LOGIC_VECTOR(N - 1 DOWNTO 0);
        Y : IN STD_LOGIC_VECTOR(N - 1 DOWNTO 0);
        P : OUT STD_LOGIC_VECTOR((2 * N) - 1 DOWNTO 0)
    );
END ECE4304_Midterm1_5;

ARCHITECTURE Structural OF ECE4304_Midterm1_5 IS

    COMPONENT NFA IS
        GENERIC (N : INTEGER := N);
        PORT (
            cin : IN STD_LOGIC;
            X : IN STD_LOGIC_VECTOR(N - 1 DOWNTO 0);
            Y : IN STD_LOGIC_VECTOR(N - 1 DOWNTO 0);
            S : OUT STD_LOGIC_VECTOR(N - 1 DOWNTO 0);
            cout : OUT STD_LOGIC
        );
    END COMPONENT;

    TYPE vector_array IS ARRAY (N - 1 DOWNTO 0) OF STD_LOGIC_VECTOR((2 * N) - 1 DOWNTO 0);
    SIGNAL sum_vecar : vector_array;
    SIGNAL X_expand_vecar : vector_array; -- expand X(i) to match width for bitwise and
    SIGNAL summand_vecar : vector_array;

BEGIN

    P <= sum_vecar(N - 1);
```

```
    X_expand_vecar(0) <= (OTHERS => X(0)); -- expand X(i) to match width for bitwise and
    SIG_PREP_1 : FOR i IN 0 TO N - 1 GENERATE
    BEGIN
        sum_vecar(0)(i) <= Y(i) AND X_expand_vecar(0)(i);
    END GENERATE;
    sum_vecar(0)(N) <= '0';

    SIG_PREP : FOR i IN 1 TO N - 1 GENERATE
    BEGIN
        X_expand_vecar(i) <= (OTHERS => X(i)); -- expand X(i) to match width for bitwise ar
        summand_vecar(i)(N + i - 1 DOWNTO i) <= X_expand_vecar(i)(N + i - 1 DOWNTO i) AND Y
        summand_vecar(i)(i - 1 DOWNTO 0) <= (OTHERS => '0');
        summand_vecar(i)(N + i) <= '0';
    END GENERATE;

    NFA_GEN : FOR i IN 1 TO N - 1 GENERATE
    BEGIN
        NFAUnit : NFA
        GENERIC MAP(N => N + i)
        PORT MAP(
            cin => '0',
            X => sum_vecar(i - 1)(N - 1 + i DOWNTO 0),
            Y => summand_vecar(i)(N - 1 + i DOWNTO 0),
            S => sum_vecar(i)(N - 1 + i DOWNTO 0),
            cout => sum_vecar(i)(N + i)
        );
    END GENERATE;
END Structural;
```

Testbench

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.math_real.ALL;
USE IEEE.numeric_std.ALL;

ENTITY ECE4304_Midterm1_5_tb IS
    --  Port ( );
END ECE4304_Midterm1_5_tb;

ARCHITECTURE Behavioral OF ECE4304_Midterm1_5_tb IS

    CONSTANT clock_period : TIME := 10 ns;
    CONSTANT N_tb : INTEGER := 3;

    SIGNAL top_clk_tb : STD_LOGIC;
    SIGNAL top_rst_tb : STD_LOGIC;

    SIGNAL X_tb : STD_LOGIC_VECTOR(N_tb - 1 DOWNTO 0);
    SIGNAL Y_tb : STD_LOGIC_VECTOR(N_tb - 1 DOWNTO 0);
```

```
    SIGNAL P_tb : STD_LOGIC_VECTOR((2 * N_tb) - 1 DOWNTO 0);

    SIGNAL mismatch_tb : STD_LOGIC;

    COMPONENT ECE4304_Midterm1_5 IS
        GENERIC (N : INTEGER := N_tb);
        PORT (
            X : IN STD_LOGIC_VECTOR(N - 1 DOWNTO 0);
            Y : IN STD_LOGIC_VECTOR(N - 1 DOWNTO 0);
            P : OUT STD_LOGIC_VECTOR((2 * N) - 1 DOWNTO 0)
        );
    END COMPONENT;

BEGIN

    UUT : ECE4304_Midterm1_5
    GENERIC MAP(N => N_tb)
    PORT MAP(
        X => X_tb,
        Y => Y_tb,
        P => P_tb
    );

    CLK_GEN : PROCESS
    BEGIN
        top_clk_tb <= '0';
        WAIT FOR 0.5 * clock_period;
        top_clk_tb <= '1';
        WAIT FOR 0.5 * clock_period;
    END PROCESS;

    INIT_RST : PROCESS
    BEGIN
        top_rst_tb <= '1';
        WAIT FOR 0.5 * clock_period;
        top_rst_tb <= '0';
        WAIT;
    END PROCESS;

    TEST_MAIN : PROCESS
    BEGIN
        FOR i IN 0 TO 2 ** N_tb - 1 LOOP -- all possible combination of inputs
            FOR j IN 0 TO 2 ** N_tb - 1 LOOP
                X_tb <= STD_LOGIC_VECTOR(to_unsigned(i, N_tb));
                Y_tb <= STD_LOGIC_VECTOR(to_unsigned(j, N_tb));
                WAIT FOR clock_period;
                IF (conv_integer(P_tb) = i * j) THEN
                    mismatch_tb <= '0';
                ELSE
                    mismatch_tb <= '1';
                END IF;
            END LOOP;
        END LOOP;
    END PROCESS;
```

```
END Behavioral;
```

Simulation Result

As can be seen in figure 6 and 7, all possible input combination was simulated and the circuit behave as expected with no mismatch found.

Figure 6: Simulation Waveform of the Circuit

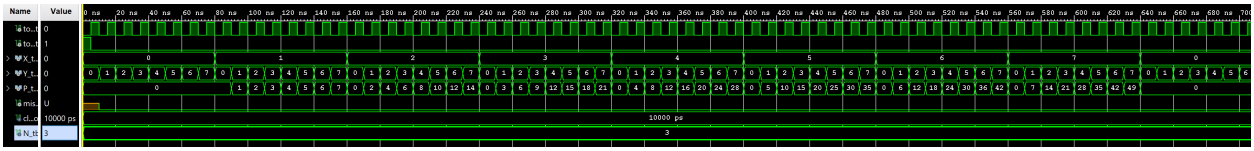


Figure 7: Simulation Waveform of the Circuit