Computer-Aided Design for VLSI Design

Homework1（Student ID: 41147046S｜Name: 楊子萱）

1. Provide a simple explanation of your code.

   Hw1 的設計，可供後續直接使用

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

entity My_ckt_1 is
    Port ( A : in STD_LOGIC_VECTOR (7 downto 0);
           B : in STD_LOGIC_VECTOR (7 downto 0);
           S : in STD_LOGIC_VECTOR (1 downto 0);
           Y : out STD_LOGIC_VECTOR (15 downto 0));
end My_ckt_1;

architecture Behavioral of My_ckt_1 is
begin
    process (A, B, S)
    begin
        case S is
            when "00" =>  -- Mode 1: Bitwise OR
                Y(7 downto 0) <= STD_LOGIC_VECTOR(UNSIGNED(A) OR UNSIGNED(B));
                Y(15 downto 8) <= (others => '0');

            when "01" =>  -- Mode 2: Multiplication
                Y <= STD_LOGIC_VECTOR(RESIZE(UNSIGNED(A) * UNSIGNED(B), 16));

            when "10" =>  -- Mode 3: Addition
                Y <= STD_LOGIC_VECTOR(RESIZE(UNSIGNED(A) + UNSIGNED(B), 16));

            when "11" =>  -- Mode 4: Modulus
                if UNSIGNED(B) /= 0 then
                    Y(7 downto 0) <= STD_LOGIC_VECTOR(RESIZE(UNSIGNED(A) MOD UNSIGNED(B), 8));
                    Y(15 downto 8) <= (others => '0');
                else
                    Y <= (others => '0'); -- ??K???H?s???~
                end if;

            when others =>
                Y <= (others => '0');
        end case;
    end process;
end Behavioral;
```

   D 型正反器的設計

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

entity D_FF is
    Port (
        clk : in STD_LOGIC;
        D   : in STD_LOGIC_VECTOR;
        Q   : out STD_LOGIC_VECTOR
    );
end D_FF;

architecture Behavioral of D_FF is
begin
    process(clk)
    begin
        if rising_edge(clk) then
            Q <= D;
        end if;
    end process;
end Behavioral;

```

定義模組名稱為 My_ckt_2，輸入為 clk, A, B, S，輸出為 Y。

```vhdl
entity My_ckt_2 is
    Port (
        clk : in STD_LOGIC;
        A   : in STD_LOGIC_VECTOR(7 downto 0);
        B   : in STD_LOGIC_VECTOR(7 downto 0);
        S   : in STD_LOGIC_VECTOR(1 downto 0);
        Y   : out STD_LOGIC_VECTOR(15 downto 0)
    );
end My_ckt_2;
```

訴編譯器我們稍後會在此模組中用到 D_FF 和 My_ckt_1，這兩個是子模組。

```
17      component D_FF
18          Port (
19              clk : in STD_LOGIC;
20              D   : in STD_LOGIC_VECTOR;
21              Q   : out STD_LOGIC_VECTOR
22          );
23      end component;
24
25      component My_ckt_1
26          Port (
27              A : in STD_LOGIC_VECTOR(7 downto 0);
28              B : in STD_LOGIC_VECTOR(7 downto 0);
29              S : in STD_LOGIC_VECTOR(1 downto 0);
30              Y : out STD_LOGIC_VECTOR(15 downto 0)
31          );
32      end component;
```

定義內部的中介訊號，用來連接 flip-flop 與 My_ckt_1，並暫存資料。

```
34      -- 中介線路
35      signal A_reg, B_reg : STD_LOGIC_VECTOR(7 downto 0);
36      signal S_reg        : STD_LOGIC_VECTOR(1 downto 0);
37      signal Y_temp       : STD_LOGIC_VECTOR(15 downto 0);
38      signal Y_reg        : STD_LOGIC_VECTOR(15 downto 0);
```

每個輸入都先經過 D flip-flop 暫存，以實現同步輸入。

```
42      DFF_A : D_FF
43          port map(clk => clk, D => A, Q => A_reg);
44
45      -- 註冊輸入 B
46      DFF_B : D_FF
47          port map(clk => clk, D => B, Q => B_reg);
48
49      -- 註冊輸入 S
50      DFF_S : D_FF
51          port map(clk => clk, D => S, Q => S_reg);
52
53      -- 註冊輸出 Y
54      DFF_Y : D_FF
55          port map(clk => clk, D => Y_temp, Q => Y_reg);
```

呼叫之前寫好的 My_ckt_1 模組，進行加法、乘法等運算，輸出結果給 Y_temp。

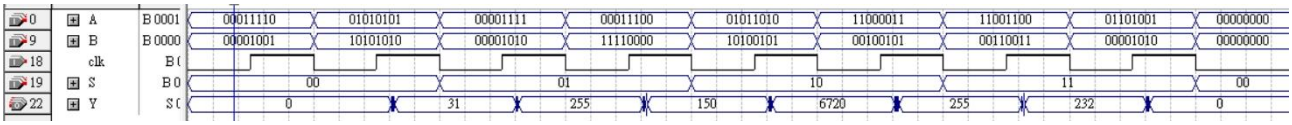
```
58      U1 : My_ckt_1
59          port map(
60              A => A_reg,
61              B => B_reg,
62              S => S_reg,
63              Y => Y_temp
64          );
```

把 D flip-flop 的輸出送到頂層輸出 Y

```
67      Y <= Y_reg;
```

2. Waveform diagram here (Simulation Results)

   波形圖含在.zip 中



3. Reflections and discussions

   在此設計中，考量到乘法運算時兩個 8 位元輸入可能產生 16 位元結果，因此需使用 UNSIGNED 型別搭配 RESIZE，以避免位元溢位問題。針對 MOD 運算，需避免除以 0 的未定義行為，因此透過 if 條件判斷來確保輸出穩定。另一方面，本電路採用 D flip-flop 並以時脈控制輸出，確保輸出僅在上升沿更新，具有時序同步的特性。從模擬結果可觀察到輸出 Y 會隨著 clk 的 rising edge 對應更新輸出值，使整體電路在時序控制與資料鎖存方面更具可靠性與一致性。