

# Computer-Aided Design for VLSI Design

Homework1 ( Student ID: 41047047S | Name: 楊子萱 )

1. Provide a simple explanation of your code.

引用 library

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.NUMERIC_STD.ALL;
```

定義電路的 input/output

```
5  entity My_ckt_1 is
6  Port ( A : in STD_LOGIC_VECTOR (7 downto 0);
7        B : in STD_LOGIC_VECTOR (7 downto 0);
8        S : in STD_LOGIC_VECTOR (1 downto 0);
9        Y : out STD_LOGIC_VECTOR (15 downto 0));
10 end My_ckt_1;
```

Bitwise OR

```
17      when "00" => -- Mode 1: Bitwise OR
18          Y(7 downto 0) <= STD_LOGIC_VECTOR(UNSIGNED(A) OR UNSIGNED(B));
19          Y(15 downto 8) <= (others => '0');
```

乘法

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

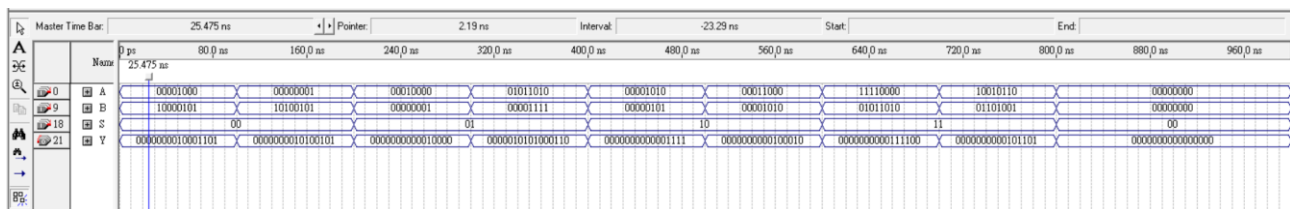
加法

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

取餘數

```
27      when "11" => -- Mode 4: Modulus
28      if UNSIGNED(B) /= 0 then
29          Y(7 downto 0) <= STD_LOGIC_VECTOR(RESIZE(UNSIGNED(A) MOD UNSIGNED(B), 8));
30          Y(15 downto 8) <= (others => '0');
31      else
32          Y <= (others => '0'); -- 避免除以零錯誤
33      end if;
```

2. Waveform diagram here (Simulation Results)



### 3. Reflections and discussions

由於進行乘法運算時，兩個 8 位元的數值相乘可能產生最多 16 位元的結果，因此需要使用 **UNSIGNED** 型別來正確處理位元擴展和溢位問題。此外，在執行取餘數 (**MOD**) 運算時，需特別考慮除數為 0 的情況。根據數學定義，除以 0 是未定義行為，因此在電路設計中，通常將此情況視為錯誤並將輸出設為 0，以避免不確定的狀態或硬體邏輯異常。透過在程式中加入條件判斷 (**if B != 0**) 來處理這個邊界情況，可確保電路在所有可能的輸入條件下都能穩定運行。