**微算機系統**

**小組專案報告**

實驗七

組別： 18

班級、姓名與學號：

資工二 蕭耕宏 110590005

資工二 楊榮鈞 110590034

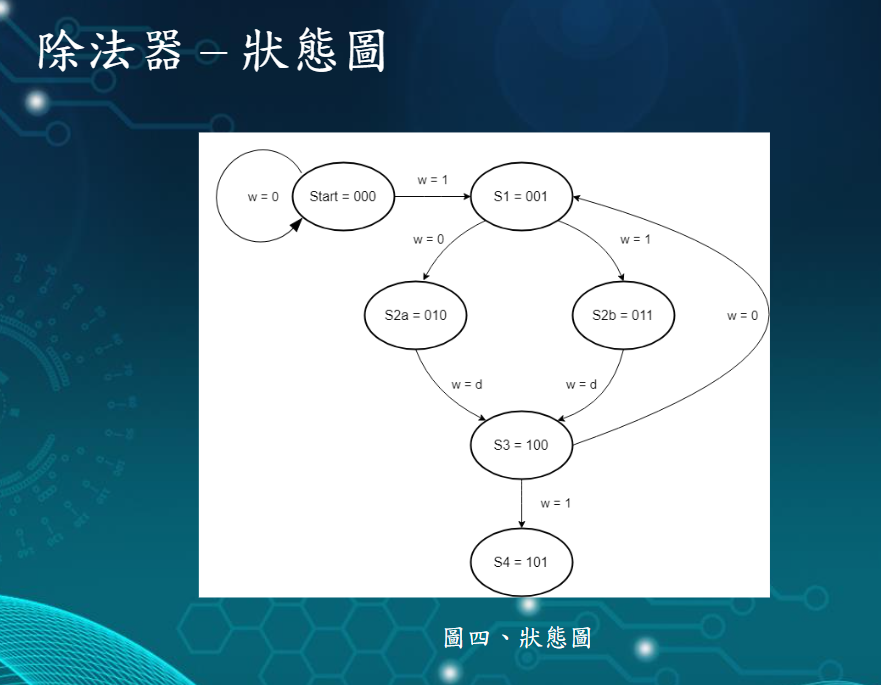
日期： 2022.12.06

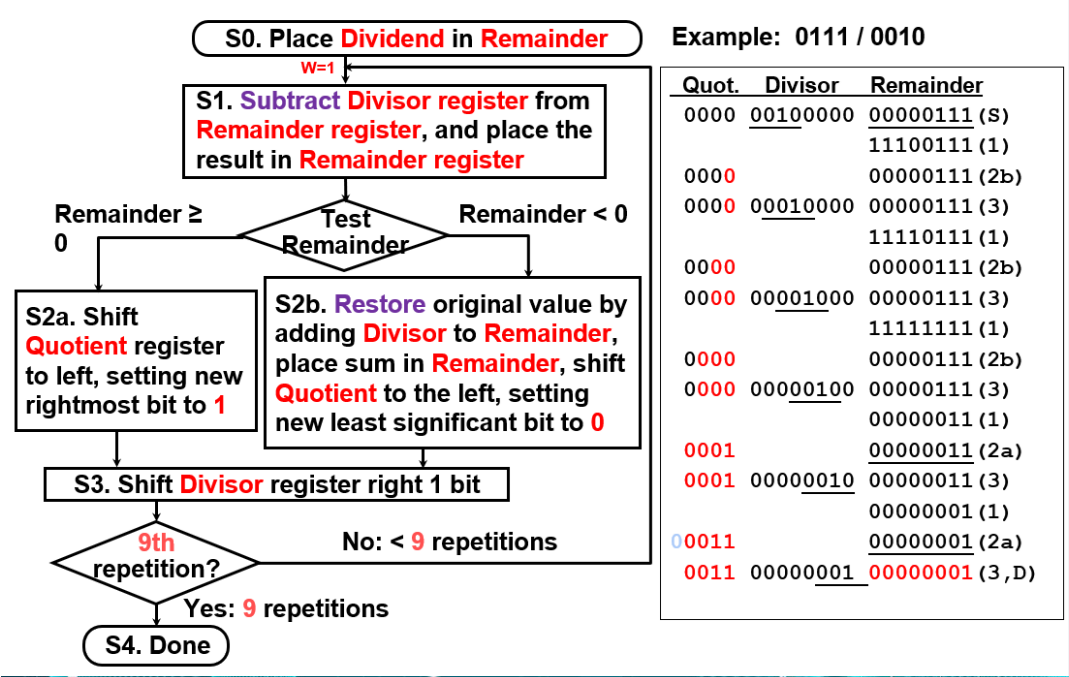
1. 實驗內容：

利用moore機(lab6)和萬用移位暫存器(lab5)，實作8 bits除法器，利用按鈕改變clock觸發事件並以w來切換輸入訊號，除數和被除數的輸入範圍為1-255，顯示商和餘在LED上，並在計算完成時，把商和餘以16進位顯示在七段顯示器上。

1. 實驗過程及結果：

Moore狀態圖





實驗的結果

|  |  |
| --- | --- |
|  |  |
| reset | S0 |
|  |  |
| S1(first round) | S2(first round,S2b) |
|  |  |
| S3(first round) | S4 |

1. 程式碼

|  |
| --- |
| 進階題 |
| Lab7.vhd |
| library ieee;  use ieee.std\_logic\_1164.all;  use ieee.std\_logic\_unsigned.all;  use ieee.numeric\_std.all;  entity Lab7 is  port( clk,reset : in std\_logic; --clear reset  Divisor,Dividend : in std\_logic\_vector(7 downto 0);  Quotient : buffer std\_logic\_vector(7 downto 0);  Remainder : buffer std\_logic\_vector(15 downto 0);  HEX0,HEX1,HEX2,HEX3 : out std\_logic\_vector(0 to 6)  );  end Lab7;  architecture behavior of Lab7 is  signal D\_sor : std\_logic\_vector(15 downto 0);    type state\_type is (S0,S1,S2a,S2b,S3,S4);  signal S\_now : state\_type;    shared variable S\_next : state\_type;  shared variable w : std\_logic;  shared variable count : integer;  shared variable temp : std\_logic\_vector(15 downto 0);    begin  process(clk,reset)  begin  if (reset = '1') then  S\_now <= S0;  Remainder <= (others => '0');  Quotient <= (others => '0');  HEX3 <= (others => '1'); --quotient msb  HEX2 <= (others => '1'); --quotient lsb  HEX1 <= (others => '1'); --remainder msb  HEX0 <= (others => '1'); --remainder lsb  elsif (clk'event and clk = '1') then  case S\_now is    when S0 =>  count := 0;  D\_sor <= Divisor & "00000000";  Remainder <= "00000000" & Dividend;  w := '1';    if w = '0' then  S\_next := S0;  else  S\_next := s1;  end if;    when S1 =>  Remainder <= Remainder - D\_sor;  temp := Remainder - D\_sor;    if temp(15) = '1' then  w := '1';  else  w := '0';  end if;    if w = '0' then  S\_next := S2a;  else  S\_next := S2b;  end if;    when S2a =>  for i in 7 downto 1 loop  Quotient(i) <= Quotient(i-1);  end loop;  Quotient(0) <= '1';    S\_next := S3;    when S2b =>  Remainder <= Remainder + D\_sor;  for i in 7 downto 1 loop  Quotient(i) <= Quotient(i-1);  end loop;  Quotient(0) <= '0';    S\_next := S3;    when S3 =>  for i in 0 to 14 loop  D\_sor(i) <= D\_sor(i+1);  end loop;  D\_sor(15) <= '0';    count := count + 1;  if count = 9 then  w := '1';  else  w := '0';  end if;    if w = '1' then  S\_next := S4;  else  S\_next := S1;  end if;    when S4 =>  S\_next := S4;  when others =>  -------------  end case;  S\_now <= S\_next;    -- a is MSB and hex is abcdefg  case Quotient (7 downto 4) is  when "0000" => HEX3 <= "0000001";  when "0001" => HEX3 <= "1001111";  when "0010" => HEX3 <= "0010010";  when "0011" => HEX3 <= "0000110";  when "0100" => HEX3 <= "1001100";  when "0101" => HEX3 <= "0100100";  when "0110" => HEX3 <= "0100000";  when "0111" => HEX3 <= "0001111";  when "1000" => HEX3 <= "0000000";  when "1001" => HEX3 <= "0001100";  when "1010" => HEX3 <= "0001000";  when "1011" => HEX3 <= "1100000";  when "1100" => HEX3 <= "1110010";  when "1101" => HEX3 <= "1000010";  when "1110" => HEX3 <= "0110000";  when "1111" => HEX3 <= "0111000";  when others => HEX3 <= "1111111";  end case;  case Quotient (3 downto 0) is  when "0000" => HEX2 <= "0000001";  when "0001" => HEX2 <= "1001111";  when "0010" => HEX2 <= "0010010";  when "0011" => HEX2 <= "0000110";  when "0100" => HEX2 <= "1001100";  when "0101" => HEX2 <= "0100100";  when "0110" => HEX2 <= "0100000";  when "0111" => HEX2 <= "0001111";  when "1000" => HEX2 <= "0000000";  when "1001" => HEX2 <= "0001100";  when "1010" => HEX2 <= "0001000";  when "1011" => HEX2 <= "1100000";  when "1100" => HEX2 <= "1110010";  when "1101" => HEX2 <= "1000010";  when "1110" => HEX2 <= "0110000";  when "1111" => HEX2 <= "0111000";  when others => HEX2 <= "1111111";  end case;  case Remainder (7 downto 4) is  when "0000" => HEX1 <= "0000001";  when "0001" => HEX1 <= "1001111";  when "0010" => HEX1 <= "0010010";  when "0011" => HEX1 <= "0000110";  when "0100" => HEX1 <= "1001100";  when "0101" => HEX1 <= "0100100";  when "0110" => HEX1 <= "0100000";  when "0111" => HEX1 <= "0001111";  when "1000" => HEX1 <= "0000000";  when "1001" => HEX1 <= "0001100";  when "1010" => HEX1 <= "0001000";  when "1011" => HEX1 <= "1100000";  when "1100" => HEX1 <= "1110010";  when "1101" => HEX1 <= "1000010";  when "1110" => HEX1 <= "0110000";  when "1111" => HEX1 <= "0111000";  when others => HEX1 <= "1111111";  end case;  case Remainder (3 downto 0) is  when "0000" => HEX0 <= "0000001";  when "0001" => HEX0 <= "1001111";  when "0010" => HEX0 <= "0010010";  when "0011" => HEX0 <= "0000110";  when "0100" => HEX0 <= "1001100";  when "0101" => HEX0 <= "0100100";  when "0110" => HEX0 <= "0100000";  when "0111" => HEX0 <= "0001111";  when "1000" => HEX0 <= "0000000";  when "1001" => HEX0 <= "0001100";  when "1010" => HEX0 <= "0001000";  when "1011" => HEX0 <= "1100000";  when "1100" => HEX0 <= "1110010";  when "1101" => HEX0 <= "1000010";  when "1110" => HEX0 <= "0110000";  when "1111" => HEX0 <= "0111000";  when others => HEX0 <= "1111111";  end case;  end if;  end process;  end behavior; |

1. 本次實驗過程說明與解決方法:

實驗過程:

本次lab會使用到前面moore機(lab6)和萬用移位暫存器(lab5)的code，把除法器的硬體邏輯寫好後與萬用移位暫存器合併。

我們把code燒到板子後發現state跑到一定的次數會卡住，之後發現問題是w、next\_state(S\_next)是signal，而且我們是用remainder - divisor的MSB去給w值，產生非預期的狀況，導致moore壞掉。

解決方式：

經過多次嘗試我們發現signal會在跑完architecture才會改變值，variable可以立刻改變。

於是我們把w、next\_state(S\_next)改用shared variable讓他可以直接在process裡面直接改變，將remainder buffer(remainder) - divisor buffer(D\_sor)用temp(shared variable)去接，使用temp的MSB判斷來給w值，於是就完成了。