

數位系統設計作業

HW4

學號: 01257027 | 姓名: 林承羿

第一題

程式碼

單一個 BCD (9~0) :

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_ARITH.ALL;
4  use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6  entity downBCD_4bits is
7      port(
8          casin:in std_logic;           -- 可否工作
9          clk:in std_logic;             -- 時脈
10         reset:in std_logic;            -- 歸零
11         casout:out std_logic;           -- 進位輸出
12         Q:out std_logic_vector(3 downto 0)
13     );
14 end downBCD_4bits;
15
16 architecture downBCD_4bits of downBCD_4bits is
17     signal cnt:std_logic_vector(3 downto 0);
18 begin
19     process(clk, reset)
20     begin
21         if(reset = '0') then
22             cnt <= x"0";                -- 強制歸零
23         elsif rising_edge(clk) then
24             if(casin = '1') then
25                 if(cnt = x"0") then
26                     cnt <= x"9";        -- 可以工作
27                                         -- BCD計數器規則
28                 else
29                     cnt <= cnt - '1';   -- 可以下數
30                 end if;
31             end if;
32         end process;
33     Q <= cnt;
34
35     process(cnt, casin)
36     begin
37         if ((cnt = x"0") and (casin = '1')) then -- 如果是零，應該回到九，BCD計數器規則
38             casout <= '1';
39         else
40             casout <= '0';
41         end if;
42     end process;
43 end downBCD_4bits;
```

從以上可以看出，主要即看看 casin 決定是否下數，且 0 應該回到 9，而非 F。在第二個 process 主要寫組合邏輯判斷是否介位。

三個 BCD 組合：

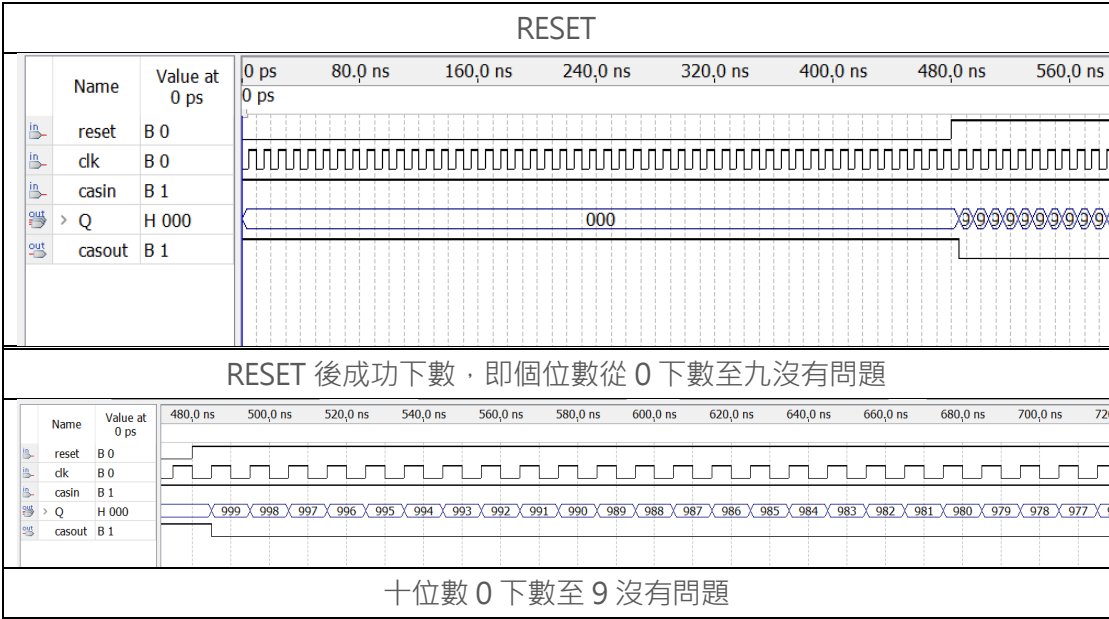
```

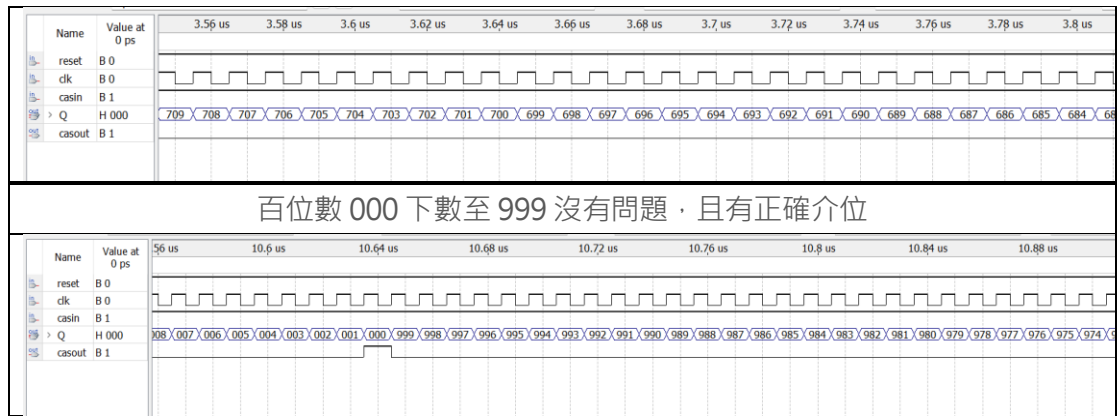
1  You, 3 小时前 | 1 author (You)
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4  use IEEE.STD_LOGIC_ARITH.ALL;
5  use IEEE.STD_LOGIC_UNSIGNED.ALL;
6
7  entity downBCD_12bits is
8      port(
9          casin:in std_logic;
10         clk:in std_logic;
11         reset:in std_logic;
12         casout:out std_logic;
13         Q:out std_logic_vector(11 downto 0)
14     );
15 end entity;
16
17 architecture downBCD_12bits of downBCD_12bits is
18     signal casout1, casout2:std_logic;
19     component downBCD_4bits is
20         port(
21             casin:in std_logic;
22             clk:in std_logic;
23             reset:in std_logic;
24             casout:out std_logic;
25             Q:out std_logic_vector(3 downto 0)
26         );
27     end component;
28
29     begin
30         downBCD1:downBCD_4bits port map(casin, clk, reset, casout1, Q(3 downto 0));
31         downBCD2:downBCD_4bits port map(casout1, clk, reset, casout2, Q(7 downto 4));
32         downBCD3:downBCD_4bits port map(casout2, clk, reset, casout, Q(11 downto 8));
33     end downBCD_12bits;

```

可以看出主要引用單個 entity 串接出三個的組合。而值得注意的是進位的兩個為內部訊號，不須輸出。

波形圖





加分題

程式碼

單個 BCD 上下數：

```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_ARITH.ALL;
4  use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6  entity up_downBCD_4bits is
7      port(
8          casin:in std_logic;
9          up_down:in std_logic;
10         clk:in std_logic;
11         reset:in std_logic;
12         casout:out std_logic;
13         Q:out std_logic_vector(3 downto 0)
14     );
15 end up_downBCD_4bits;

```

比原本 BCD 下數 entity 多個紀錄上下數的 up_down

```

16
17 architecture up_downBCD_4bits of up_downBCD_4bits is
18     signal cnt:std_logic_vector(3 downto 0);
19 begin
20     process(reset, clk)
21     begin
22         if(reset = '0') then
23             if (up_down = '1') then
24                 cnt <= x"0";
25             elsif(up_down = '0') then
26                 cnt <= x"9";
27             end if;
28         elsif(rising_edge(clk)) then
29             if(casin = '1') then
30                 if(up_down = '1') then
31                     if(cnt = x"9") then
32                         cnt <= x"0";
33                     else
34                         cnt <= cnt + '1';
35                     end if;
36                 elsif(up_down = '0') then
37                     if(cnt = x"0") then
38                         cnt <= x"9";
39                     else
40                         cnt <= cnt - '1';
41                     end if;
42                 end if;
43             end if;
44         end if;
45     end process;
46     Q <= cnt;

```

先確認是否 RESET，是否正元觸發，又是否可以上 or 下數，最後上下數區間是否正確

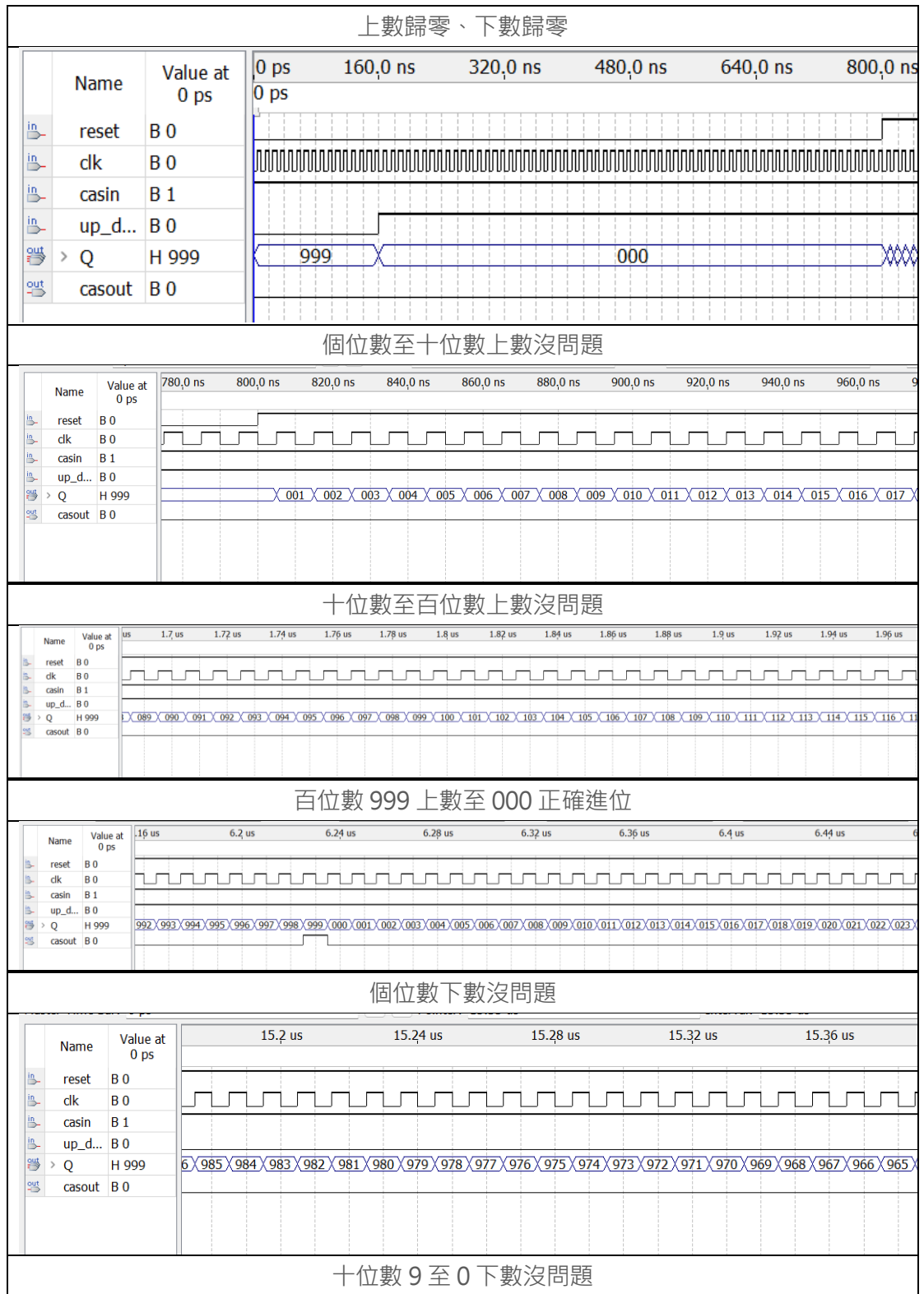
```

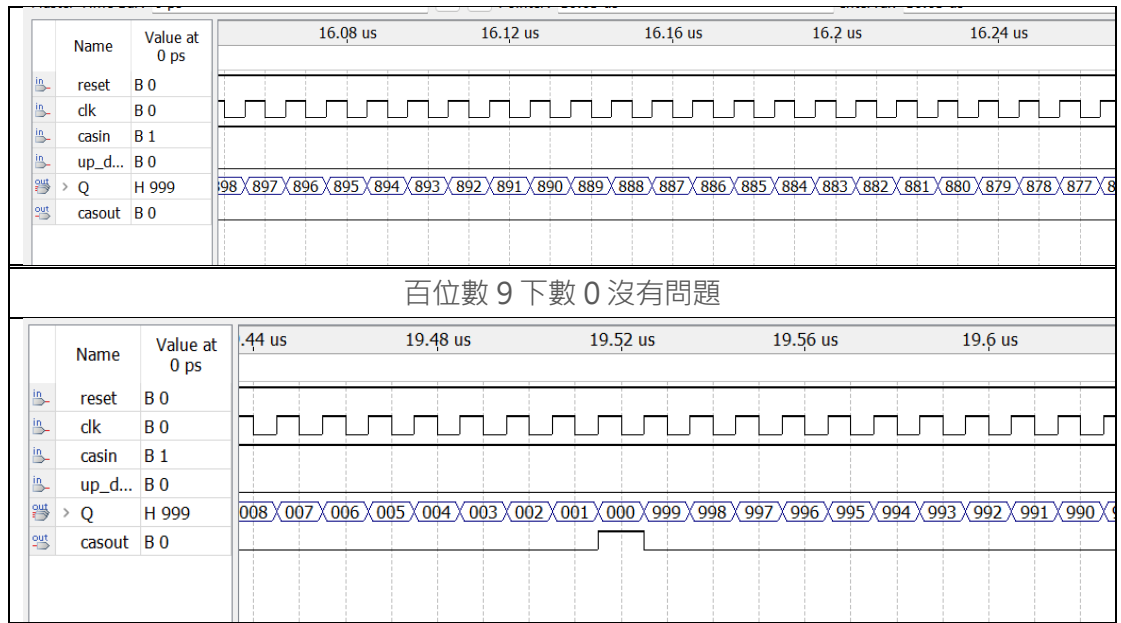
47     process(cnt, casin, up_down)
48     begin
49         if(casin = '1') then
50             if((up_down = '1') and (cnt = x"9"))then
51                 casout <= '1';
52             elsif((up_down = '0') and (cnt = x"0"))then
53                 casout <= '1';
54             else
55                 casout <= '0';
56             end if;
57         else
58             casout <= '0';
59         end if;
60     end process;
61 end up_downBCD_4bits ;

```

確認上下數區間是否進位、借位 (組合邏輯)

波形圖





心得

藉由此次的實驗，我更了解何謂計數器，且了解到組合邏輯、序項邏輯的分類不應該以是否以 process 實作為依據，應該以大方向我該做什麼來判斷這小部分是某與時脈有關為判斷依據。根據加分題的代碼，我更了解到何謂優先序，哪些事情是要在前提成立下才可以實作的，也更讓我了解寫程式更像在與機器溝通，也寫出看得懂、容易維護的代碼。