7 節顯示器用掃描模式練習

seg7_scan.vhd

```
library ieee ;
use ieee.std_logic_1164.all ;
use ieee.std_logic_unsigned.all ;
use ieee.std_logic_arith.all ;
--*****************************
entity seg7_scan is
  port ( clk2 : in std_logic;--掃描頻率，4 個 7 節顯示器輪流閃亮時間
            a,b,c,d : in integer range 0 to 9;-- 4 個 7 節顯示器之輸入值
            scan : out std_logic_vector(3 downto 0);-- 4 個 7 節顯示器掃描之訊號
            seg : out std_logic_vector(0 to 6)) ;-- 7 節顯示器之輸出訊號
end seg7_scan ;
--******************************
architecture A_with_select_when of seg7_scan is
signal bcd : integer range 0 to 9;
begin
Scan_a:
process(Clk2)
  variable Scan1 : std_logic_vector(1 downto 0):="00";--將 scan1 變數初始值設定為”00”
   begin
   --if Clk2='1' and Clk2'event then
      wait until clk2='1' ;
      if (Scan1="00") then
            Scan <= "0111" ;--千位亮，值為 d
            bcd <= d ;
      elsif (Scan1="01") then
            Scan <= "1011" ;--百位亮，值為 c
            bcd <= c;
      elsif (Scan1="10") then
            Scan <= "1101";   --十位亮，值為 b
            bcd <= b;
      else Scan <= "1110" ; --個位亮，值為 a
            bcd <= a;
      end if ;
      if Scan1 >= "11" then Scan1 := "00" ;
      else Scan1 := Scan1 + 1 ;--scan1 加 1，輪流閃亮
      end if ;
end process scan_a ;
--*********************************
with bcd select      --7 節顯示器顯示電路
  seg <= "1111110"    when 0 ,
```

```
            "0110000"    when 1 ,
            "1101101"    when 2 ,
            "1111001"    when 3 ,
            "0110011"    when 4 ,
            "1011011"    when 5 ,
            "0011111"    when 6 ,
            "1110000"    when 7 ,
            "1111111"    when 8 ,
            "1110011"    when 9 ,
            "1001111"    when others ;
   end A_with_select_when    ;
```

SEG7_SCAN

** DEVICE SUMMARY **

Chip/ POF	Device	Input Pins	Output Pins	Bidir Pins	LCs	Shareable Expanders	% Utilized
seg7_scan							
	EPM3064ALC44-10	17	11	0	17	0	26 %
User Pins:		17	11	0			

Project Information d:\ex\seg7\seg7_scan.rpt

** AUTO GLOBAL SIGNALS **

INFO: Signal 'clk2' chosen for auto global Clock

Project Information d:\ex\seg7\seg7_scan.rpt

** PIN/LOCATION/CHIP ASSIGNMENTS **

User Assignments	Actual Assignments (if different)	Node Name	User Assignments	Actual Assignments (if different)	Node Name
seg7_scan@34		a0	seg7_scan@39		scan2
seg7_scan@33		a1	seg7_scan@37		scan3
seg7_scan@31		a2	seg7_scan@4		seg0
seg7_scan@29		a3	seg7_scan@5		seg1
seg7_scan@28		b0	seg7_scan@6		seg2
seg7_scan@27		b1	seg7_scan@8		seg3
seg7_scan@26		b2	seg7_scan@9		seg4
seg7_scan@25		b3	seg7_scan@11		seg5
seg7_scan@2		clk2	seg7_scan@12		seg6
seg7_scan@21		c0			
seg7_scan@20		c1			
seg7_scan@19		c2			
seg7_scan@18		c3			
seg7_scan@41		scan0			
seg7_scan@40		scan1			

4 位元計數器 VHDL(1)--由 0~9999 計數

(7 節顯示器用掃描模式)

seg7_4

```
library ieee ;
use ieee.std_logic_1164.all ;
use ieee.std_logic_unsigned.all ;
use ieee.std_logic_arith.all ;
--******************************
entity seg7_4 is
  port ( clk1,clk2: in std_logic;
            scan: out std_logic_vector(3 downto 0);
            seg : out std_logic_vector(0 to 6)) ;
end seg7_4 ;
--******************************
architecture A_with_select_when of seg7_4 is
signal count    :      std_logic_vector(15 downto 0);
signal bcd              :      std_logic_vector(3 downto 0);
begin
inc_1:
process(clk1)----加 1 電路部份
      begin
            wait until clk1='1' ;
            if count(3 downto 0)<=8 then
               count(3 downto 0)<=count(3 downto 0)+1;
               elsif count(7 downto 4)<=8 then
                        count(3 downto 0)<="0000";
                        count(7 downto 4)<=count(7 downto 4)+1;
               elsif count(11 downto 8)<=8 then
                        count(7 downto 0)<="00000000";
                        count(11 downto 8)<=count(11 downto 8)+1;
               elsif count(15 downto 12)<=8 then
                        count(11 downto 0)<="000000000000";
                        count(15 downto 12)<=count(15 downto 12)+1;
            else count(15 downto 0)<="0000000000000000";
            end if;
end process inc_1;
--*****************************************************
Scan_a:
process(Clk2)
  variable Scan1 : std_logic_vector(1 downto 0):="00";
    begin
```

十位數加 1
個位數歸 0

百位數加 1
十位個位數歸 0

千位數加 1
百位十位個位數歸 0

```
    --if Clk2='1' and Clk2'event then
        wait until clk2='1' ;
        if (Scan1="00") then
                Scan <= "0111" ;
                bcd <= Count(15 downto 12) ;
        elsif (Scan1="01") then
                Scan <= "1011" ;
                bcd <= Count(11 downto 8);
        elsif (Scan1="10") then
                Scan <= "1101" ;
                bcd <= Count(7 downto 4);
        else Scan <= "1110" ;
                bcd <= Count(3 downto 0);
        end if ;
        if Scan1 >= "11" then Scan1 := "00" ;
        else Scan1 := Scan1 + 1 ;
        end if ;
end process scan_a ;
--*******************************************************
with bcd select
  seg <= "1111110"    when "0000" ,
             "0110000"    when "0001" ,
             "1101101"    when "0010" ,
             "1111001"    when "0011" ,
             "0110011"    when "0100" ,
             "1011011"    when "0101" ,
             "0011111"    when "0110" ,
             "1110000"    when "0111" ,
             "1111111"    when "1000" ,
             "1110011"    when "1001" ,
             "1001111"    when others ;
    end A_with_select_when    ;
```

seg7_4.rpt 內容節錄

***** Project compilation was successful

SEG7_4

** DEVICE SUMMARY **

Chip/ POF	Device	Input Pins	Output Pins	Bidir Pins	LCs	Shareable Expanders	% Utilized
seg7_4	EPM3064ALC44-10	2	11	0	64	29	100%
User Pins:		2	11	0			

Project Information d:\ex\seg7\seg7_4.rpt

** AUTO GLOBAL SIGNALS **

INFO: Signal 'clk2' chosen for auto global Clock

INFO: Signal 'clk1' chosen for auto global Clock

Project Information d:\ex\seg7\seg7_4.rpt

** PIN/LOCATION/CHIP ASSIGNMENTS **

User Assignments	Actual Assignments (if different)	Node Name
seg7_4@43		clk1
seg7_4@2		clk2
seg7_4@41		scan0
seg7_4@40		scan1
seg7_4@39		scan2
seg7_4@37		scan3
seg7_4@4		seg0
seg7_4@5		seg1
seg7_4@6		seg2
seg7_4@8		seg3
seg7_4@9		seg4
seg7_4@11		seg5
seg7_4@12		seg6

4 位元計數器 VHDL(2)

seg7_4a.vhd

```
library ieee ;
use ieee.std_logic_1164.all ;
use ieee.std_logic_unsigned.all ;
use ieee.std_logic_arith.all ;
--******************************
entity seg7_4a is
  port ( clk1,clk2: in std_logic;
            scan: out std_logic_vector(3 downto 0);
            seg : out std_logic_vector(0 to 6)) ;
end seg7_4a ;
--******************************
architecture A_with_select_when of seg7_4a is
signal c1,c2,c3,c4,bcd    :      integer range 0 to 9;
begin
inc_1:
process(clk1)
      begin
        wait until clk1='1' ;
        if c1<=8 then c1<=c1+1;
            elsif c2<=8 then
                  c1<=0;
                  c2<=c2+1;
            elsif c3<=8 then
                  c1<=0;
                  c2<=0;
                  c3<=c3+1;
            elsif c4<=8 then
                  c1<=0;
                  c2<=0;
                  c3<=0;
                  c4<=c4+1;
            else c1<=0;
                   c2<=0;
                   c3<=0;
                   c4<=0;
        end if;
end process inc_1;
--*************************************
Scan_a:
process(Clk2)
  variable Scan1 : std_logic_vector(1 downto
0):="00";
    begin
    --if Clk2='1' and Clk2'event then
       wait until clk2='1' ;
       if (Scan1="00") then
              Scan <= "0111" ;
              bcd <= c4 ;
       elsif (Scan1="01") then
              Scan <= "1011" ;
              bcd <= c3;
       elsif (Scan1="10") then
              Scan <= "1101" ;
              bcd <= c2;
       else Scan <= "1110" ;
              bcd <= c1;
       end if ;
       if Scan1 >= "11" then Scan1 := "00" ;
       else Scan1 := Scan1 + 1 ;
       end if ;
end process scan_a ;
--*************************************
with bcd select
  seg <= "1111110"    when 0 ,
            "0110000"    when 1 ,
            "1101101"    when 2 ,
            "1111001"    when 3 ,
            "0110011"    when 4 ,
            "1011011"    when 5 ,
            "0011111"    when 6 ,
            "1110000"    when 7 ,
            "1111111"    when 8 ,
            "1110011"    when 9 ,
            "1001111"    when others ;
    end A_with_select_when    ;
```

SEG7_4A.rpt 節錄

***** Project compilation was successful

SEG7_4A

** DEVICE SUMMARY **

Chip/ POF	Device	Input Pins	Output Pins	Bidir Pins	LCs	Shareable Expanders	% Utilized
seg7_4a	EPM3064ALC44-10	6	11	0	59	26	92 %
seg7_4a1	EPM3032ALC44-4	1	4	0	5	1	15 %
TOTAL:		7	15	0	64	27	66 %
User Pins:		2	11	0			

Project Information d:\ex\seg7\seg7_4a.rpt

** PROJECT COMPILATION MESSAGES **

Info: Trying to find new partition/fit after discarding assignments as requested with the Partitioner/Fitter Status dialog box

Project Information d:\ex\seg7\seg7_4a.rpt

** AUTO GLOBAL SIGNALS **

INFO: Signal 'clk2' chosen for auto global Clock

INFO: Signal 'clk1' chosen for auto global Clock

Project Information d:\ex\seg7\seg7_4a.rpt

** MULTIPLE PIN CONNECTIONS **

For node name 'c13'

Connect: {seg7_4a1@38, seg7_4a@18}

For node name 'clk1'

Connect: {seg7_4a1@43, seg7_4a@43}

For node name 'c12'

Connect: {seg7_4a1@39, seg7_4a@20}

For node name 'c11'

Connect: {seg7_4a1@40, seg7_4a@21}

For node name 'c10'

Connect: {seg7_4a1@41, seg7_4a@19}

Project Information d:\ex\seg7\seg7_4a.rpt

** PIN/LOCATION/CHIP ASSIGNMENTS **

User Assignments	Actual Assignments (if different)	Node Name		
seg7_4a@43		clk1	seg7_4a@5	seg1
seg7_4a@2		clk2	seg7_4a@6	seg2
seg7_4a@41		scan0	seg7_4a@8	seg3
seg7_4a@40		scan1	seg7_4a@9	seg4
seg7_4a@39		scan2	seg7_4a@11	seg5
seg7_4a@37		scan3	seg7_4a@12	seg6
seg7_4a@4		seg0		

4 位元計數器 VHDL(3)

seg7_4c.vhd

```
library ieee ;
use ieee.std_logic_1164.all ;
use ieee.std_logic_unsigned.all ;
use ieee.std_logic_arith.all ;
--******************************
entity seg7_4c is
  port ( clk1,clk2: in std_logic;
            scan: out std_logic_vector(3 downto 0);
            seg : out std_logic_vector(0 to 6)) ;
end seg7_4c ;
--******************************
architecture A_with_select_when of seg7_4c is
signal c1,c2,c3,c4,bcd    :      integer range 0 to 9;
begin
inc_1:
process(clk1)
      begin
        wait until clk1='1' ;
        if c1<9 then c1<=c1+1;
            elsif c2<9 then
                  c1<=0;
                  c2<=c2+1;
            elsif c3<9 then
                  c1<=0;
                  c2<=0;
                  c3<=c3+1;
            elsif c4<9 then
                  c1<=0;
                  c2<=0;
                  c3<=0;
                  c4<=c4+1;
            else c1<=0;
                   c2<=0;
                   c3<=0;
                   c4<=0;
        end if;
end process inc_1;
--*************************************
Scan_a:
process(Clk2)
  variable Scan1 : std_logic_vector(1 downto
0):="00";
    begin
    --if Clk2='1' and Clk2'event then
       wait until clk2='1' ;
       if (Scan1="00") then
              Scan <= "0111" ;
              bcd <= c4 ;
       elsif (Scan1="01") then
              Scan <= "1011" ;
              bcd <= c3;
       elsif (Scan1="10") then
              Scan <= "1101" ;
              bcd <= c2;
       else Scan <= "1110" ;
              bcd <= c1;
       end if ;
       if Scan1 >= "11" then Scan1 := "00" ;
       else Scan1 := Scan1 + 1 ;
       end if ;
end process scan_a ;
--*************************************
with bcd select
  seg <= "1111110"    when 0 ,
            "0110000"    when 1 ,
            "1101101"    when 2 ,
            "1111001"    when 3 ,
            "0110011"    when 4 ,
            "1011011"    when 5 ,
            "0011111"    when 6 ,
            "1110000"    when 7 ,
            "1111111"    when 8 ,
            "1110011"    when 9 ,
            "1001111"    when others ;
    end A_with_select_when    ;
```

SEG7_4C.rpt 節錄

SEG7_4C

** DEVICE SUMMARY **

Chip/ POF	Device	Input Pins	Output Pins	Bidir Pins	Shareable LCs	Expanders	% Utilized
seg7_4c	EPM3064ALC44-10	2	11	0	57	28	89 %
User Pins:		2	11	0			

Project Information d:\ex\seg7\seg7_4c.rpt

** PROJECT COMPILATION MESSAGES **

Warning: Project has user pin or logic cell assignments, but has never been compiled before. For best fitting results, let the Compiler choose the first set of assignments instead.

Project Information d:\ex\seg7\seg7_4c.rpt

** AUTO GLOBAL SIGNALS **

INFO: Signal 'clk2' chosen for auto global Clock

INFO: Signal 'clk1' chosen for auto global Clock

Project Information d:\ex\seg7\seg7_4c.rpt

** PIN/LOCATION/CHIP ASSIGNMENTS **

User Assignments	Actual Assignments (if different)	Node Name
seg7_4c@43		clk1
seg7_4c@2		clk2
seg7_4c@41		scan0
seg7_4c@40		scan1
seg7_4c@39		scan2
seg7_4c@37		scan3
seg7_4c@4		seg0
seg7_4c@5		seg1
seg7_4c@6		seg2
seg7_4c@8		seg3
seg7_4c@9		seg4
seg7_4c@11		seg5
seg7_4c@12		seg6

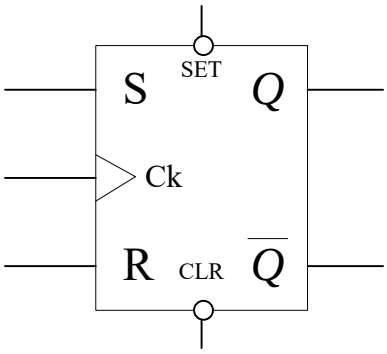
4 位元計數器 VHDL(4)

seg7_4d.vhd

```
library ieee ;
use ieee.std_logic_1164.all ;
use ieee.std_logic_unsigned.all ;
use ieee.std_logic_arith.all ;
--******************************
entity seg7_4d is
  port ( clk1,clk2: in std_logic;
            sw1: in std_logic:='1';
            scan: out std_logic_vector(3 downto 0);
            seg : out std_logic_vector(0 to 6)) ;
end seg7_4d ;
--******************************
architecture A_with_select_when of seg7_4d is
signal c1,c2,c3,c4,bcd    :      integer range 0 to 9;
begin
inc_1:
process(clk1)
      begin
        wait until clk1='1' ;
      if sw1='1' then ----sw1 歸零
            c1<=0;
            c2<=0;
            c3<=0;
            c4<=0;
            elsif c1<9 then c1<=c1+1;
            elsif c2<9 then
                  c1<=0;
                  c2<=c2+1;
            elsif c3<9 then
                  c1<=0;
                  c2<=0;
                  c3<=c3+1;
            elsif c4<9 then
                  c1<=0;
                  c2<=0;
                  c3<=0;
                  c4<=c4+1;
            else c1<=0;
                  c2<=0;
                  c3<=0;
                  c4<=0;
        end if;
end process inc_1;
--**********************************
Scan_a:
process(Clk2)
  variable Scan1 : integer range 0 to 3;
    begin
    --if Clk2='1' and Clk2'event then
      wait until clk2='1' ;
      if (Scan1=0) then
            Scan <= "0111" ;
            bcd <= c4 ;
      elsif (Scan1=1) then
            Scan <= "1011" ;
            bcd <= c3;
      elsif (Scan1=2) then
            Scan <= "1101" ;
            bcd <= c2;
      else Scan <= "1110" ;
            bcd <= c1;
      end if ;
      if Scan1 = 3 then Scan1 := 0 ;
        else Scan1 := Scan1 + 1 ;
      end if ;
end process scan_a ;
--*******************************
with bcd select
  seg <= "1111110"    when 0 ,
            "0110000"    when 1 ,
            "1101101"    when 2 ,
            "1111001"    when 3 ,
            "0110011"    when 4 ,
            "1011011"    when 5 ,
            "0011111"    when 6 ,
            "1110000"    when 7 ,
            "1111111"    when 8 ,
            "1110011"    when 9 ,
            "1001111"    when others ;
    end A_with_select_when    ;
```

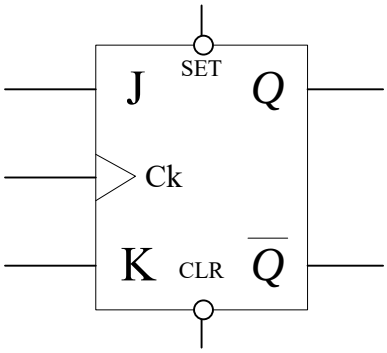
順序邏輯

正緣觸發型 SR 正反器



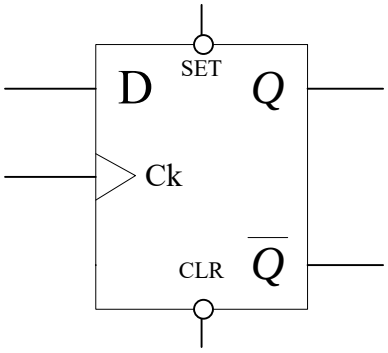
輸入					輸出
SET	CLR	CK	S	R	Q_{n+1}
0	0	x	x	x	*
0	1	x	x	x	1
1	0	x	x	x	0
1	1	↑	0	0	Q_n
1	1	↑	0	1	0
1	1	↑	0	0	1
1	1	↑	0	1	*

JK 正反器



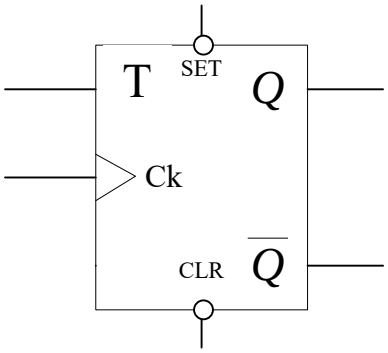
輸入					輸出
SET	CLR	CK	S	R	Q_{n+1}
0	0	x	x	x	*
0	1	x	x	x	1
1	0	x	x	x	0
1	1	↑	0	0	Q_n
1	1	↑	0	1	0
1	1	↑	0	0	1
1	1	↑	0	1	Q_n’反相

D 型正反器—資料暫存器



輸入				輸出
SET	CLR	CK	D	Q_{N+1}
0	0	X	X	*
0	1	X	X	1
1	0	X	X	0
1	1	↑	0	0
1	1	↑	1	1

T 型正反器—反轉型正反器，除 2 功能



輸入		輸出
T	CK	Q_{n+1}
0	X	Q_n
1	1	Q_n
1	0	Q_n
1	↑	Q_n’反相
1	1	Q_n

正緣觸發型 SR 正反器 VHDL

```
--*****************************
entity RS_FFV is
  port ( PR,CLR,S,R,CK : in    std_logic ;
             Q                    : out std_logic ) ;
end RS_FFV ;
--*****************************
architecture A_table of RS_FFV is
signal Q_temp : std_logic ;
begin
process(PR,CLR,CK,R,S)
   begin
      if CLR='0' then
          Q_temp <= '0' ;
      elsif PR='0' then
          Q_temp <= '1' ;
      elsif CK'event and CK='1' then –正緣觸發
          if S='0' and R='0' then
              Q_temp <= Q_temp ;
          elsif S='0' and R='1' then
                  Q_temp <= '0' ;
          elsif S='1' and R='0' then
                  Q_temp <= '1' ;
          elsif S='1' and R='1' then
                  Q_temp <= not Q_temp ;
          end if ;
      end if;
      Q <= Q_temp ;
end process ;
end A_table;
```

CLR 有優先權

CK'event and CK='1' 表示正緣觸發型正反器

CK'event and CK='0' 表示負緣觸發型正反器

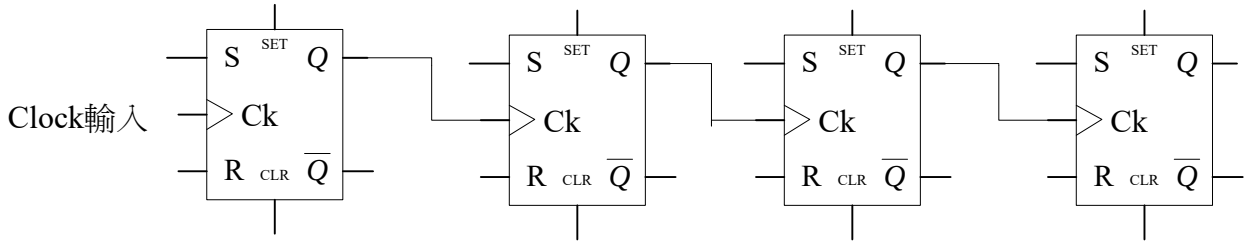
負緣觸發型 JK 正反器 VHDL

```
--*****************************
entity JK_FFV is
  port (PR,CLR,J,K,CK : in    std_logic ;
             Q                    : out std_logic ) ;
end JK_FFV ;
--*****************************
architecture A_table of JK_FFV is
signal Q_temp : std_logic ;
begin
process(PR,CLR,CK)
begin
   if PR='0' then
       Q_temp <= '1' ;
   elsif CLR='0' then
       Q_temp <= '0' ;
   elsif CK'event and CK='0' then    --負緣觸發
       if J='0' and K='0' then
           Q_temp <= Q_temp ;
       elsif J='0' and K='1' then
           Q_temp <= '0' ;
       elsif J='1' and K='0' then
           Q_temp <= '1' ;
       elsif J='1' and K='1' then
           Q_temp <= not Q_temp ;
       end if ;
   end if;
end process ;
Q <= Q_temp ;
end A_table ;
```

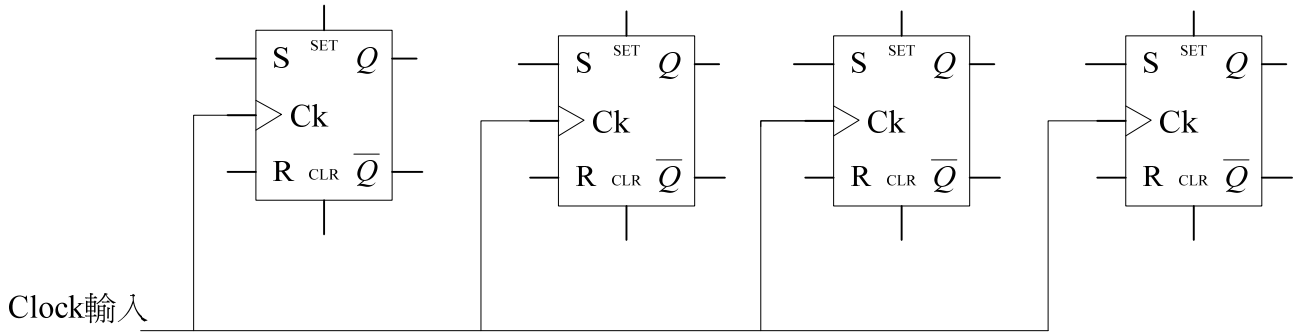
PR 有優先權

計數器

1. 非同步計數器—又稱漣波計數器或異步計數器



2. 同步計數器



延遲時間：

EPM3064ALC44-4

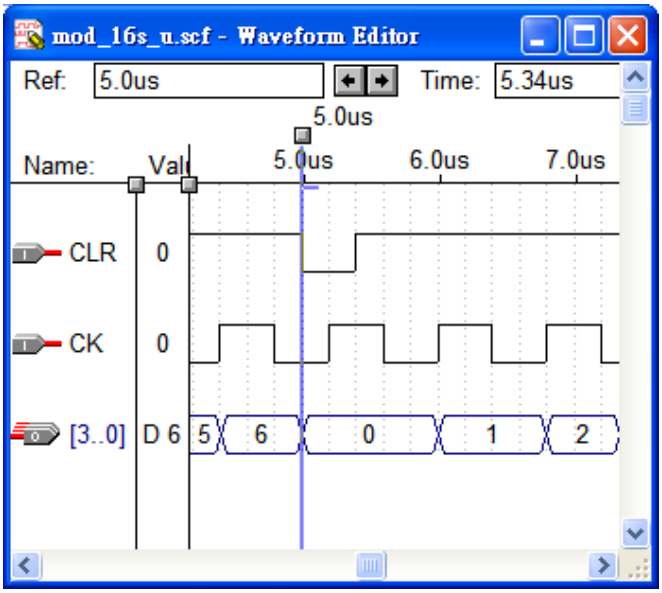
延遲時間為 4nS，工作頻率(速度)為 1/4n HZ=250MHZ

EPM3064ALC44-10

延遲時間為 10nS，工作頻率(速度)為 1/10n HZ=100MHZ

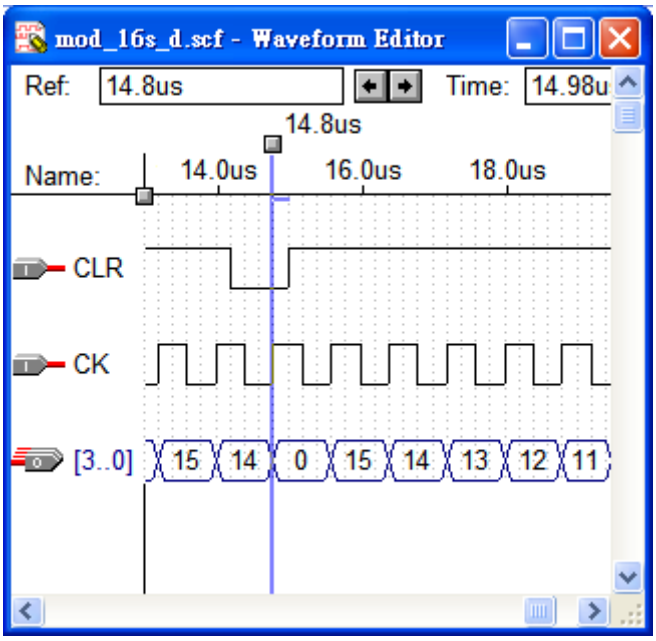
16 模同步清除上數計數器 VHDL

```
--*******************************
entity MOD_16S_U is
  port ( CLR,CK : in    std_logic ;
             Q    : out std_logic_vector(3 downto 0)) ;
end MOD_16S_U ;
--*******************************
architecture A_clear_a of MOD_16S_U is
signal Q_temp : std_logic_vector(3 downto 0);
begin
    process(CLR,CK)
    begin
      if CLR='0' then
           Q_temp <= "0000" ;
      elsif CK'event and CK='1' then
           Q_temp <= Q_temp + 1 ;
      end if;
    end process ;
    Q <= Q_temp ;
end A_clear_a ;
```



16 模同步清除下數計數器 VHDL

```
--*******************************
entity MOD_16S_d is
  port ( CLR,CK : in    std_logic ;
             Q       : out integer range 15 downto 0 ) ;
end MOD_16S_d ;
--*******************************
architecture A_clear_S of MOD_16S_d is
begin
    process(CLR,CK)
    variable Q_temp : integer range 15 downto 0 ;
    begin
      if CK'event and CK='1' then
          if CLR='0' then
               Q_temp := 0 ;
          else
               Q_temp := Q_temp - 1 ;
          end if;
      end if ;
      Q <= Q_temp ;
    end process ;
end A_clear_S ;
```



1000 模同步清除、載入的上數計數器 VHDL

```
--******************************
entity MOD_1000_S_u is
  port ( CLR,LOAD,CK : in    std_logic ;
             Di                  : in    integer range 0 to 999    ;
             Q                    : out integer range 0 to 999 ) ;
end MOD_1000_S_u ;
--******************************
architecture A_clr_load of MOD_1000_S_u is
begin
     process(CLR,CK)
     variable Q_temp : integer range 0 to 999 ;
     begin
        if CK'event and CK='1' then
             if CLR='0' then Q_temp := 0 ;
             elsif LOAD='0' then Q_temp := Di ;
             elsif Q_temp = 999 then Q_temp := 0 ;
             else    Q_temp := Q_temp + 1 ;
             end if;
        end if ;
        Q <= Q_temp ;
     end process ;
end A_clr_load ;
```

載入 Di 值

N (60) 模同步清除、載入上數計數器 VHDL

mod_60_s_u.vhd

```
--******************************
entity MOD_60_S_u is
  generic ( N : integer := 59 ) ;
  port ( CLR,LOAD,CK : in    std_logic ;
             Di    : in    integer range N downto 0      ;
             Q      : out integer range N downto 0 ) ;
end MOD_60_S_u ;
--******************************
architecture A_generic of MOD_60_S_u is
begin
     process(CLR,CK)
     variable Q_temp : integer range 0 to N ;
     begin
        wait until CK='1'    ;
             if CLR='0' then Q_temp := 0 ;
             elsif LOAD='0' then Q_temp := Di ;
             elsif Q_temp = N then Q_temp := 0 ;
             else    Q_temp := Q_temp + 1 ;
             end if;
        Q <= Q_temp ;
     end process ;
end A_generic ;
```

generic 設定元件參數

語法：

generic(名稱　：　資料型態　：=預設值)

正緣觸發

10 模同步清除上、下數計數器 VHDL

```
--******************************
entity MOD_10_UD is
  port ( CLR,DIR,CK : in    std_logic ;
              Q : out std_logic_vector(3 downto 0)) ;
end MOD_10_UD ;
--******************************
architecture A_up_down of MOD_10_UD is
signal Q_temp : std_logic_vector(3 downto 0);
begin
   process(CK)
   begin
     if CK'event and CK='1' then
       if CLR='0' then Q_temp <= "0000" ;
       elsif DIR='1' then ----上數
         if Q_temp="1001" then Q_temp <= "0000" ;
                else Q_temp <= Q_temp + 1 ;
                end if ;
       elsif DIR='0' then ----下數
         if Q_temp="0000" then Q_temp <= "1001" ;
                else Q_temp <= Q_temp - 1 ;
                end if ;
           end if;
       end if;
   end process ;
   Q <= Q_temp ;
end A_up_down ;
```

具預設載入功能兩位數的 BCD 碼計數器

```
--*******************************
entity MOD_10_double is
  port ( CLR,LOAD,EN,CK : in    std_logic ;
             D1,D0    : in    std_logic_vector(3 downto 0) ;
             Q1,Q0    : out std_logic_vector(3 downto 0));
end MOD_10_double ;
--*******************************
architecture A_multi_if of MOD_10_double is
begin
     process(CK)
     variable Q1_temp,Q0_temp : std_logic_vector(3 downto 0) ;
     begin
        if CK'event and CK='1' then
            if CLR='0' then
                Q1_temp := "0000" ;
                Q0_temp := "0000" ;
            elsif LOAD='0' then
                Q1_temp := D1 ;
                Q0_temp := D0 ;
            elsif EN='0' then
               if Q0_temp = "1001" then Q0_temp := "0000" ;
                   if Q1_temp = "1001" then Q1_temp := "0000" ;
                   else    Q1_temp := Q1_temp + 1 ;
                   end if ;
               else    Q0_temp := Q0_temp + 1 ;
               end if;
            end if;
end if;
       Q1 <= Q1_temp ;
       Q0 <= Q0_temp ;
       end process ;
end A_multi_if ;
```

清除優先

Q1_temp 十位數
Q0_temp 個位數

具預設載入功能兩位數的 BCD 碼計數器

上題改由 7 節顯示器輸出

```
library ieee ;
use ieee.std_logic_1164.all ;
use ieee.std_logic_unsigned.all ;
use ieee.std_logic_arith.all ;

--******************************
entity MOD_10_double is
  port ( CLR,LOAD,EN,CK,clk2 : in    std_logic ;
             D1,D0    : in    std_logic_vector(3 downto 0) ;
             scan: out std_logic_vector(3 downto 0);
             seg : out std_logic_vector(0 to 6)) ;
end MOD_10_double ;
--******************************
architecture A_multi_if of MOD_10_double is
signal Q1_temp,Q0_temp,bcd : std_logic_vector(3 downto 0) ;
begin
     process(CK,load)
     begin
             if LOAD='1' then
                  Q1_temp <= D1 ;
                  Q0_temp <= D0 ;
             elsif CK'event and CK='1' then
             if CLR='0' then
                  Q1_temp <= "0000" ;
                  Q0_temp <= "0000" ;
             elsif EN='0' then
                if Q0_temp = "1001" then Q0_temp <= "0000" ;
                     if Q1_temp = "1001" then Q1_temp <= "0000" ;
                     else    Q1_temp <= Q1_temp + 1 ;
                     end if ;
                else    Q0_temp <= Q0_temp + 1 ;
                end if;
            end if;
       end if;
end process ;

Scan_a:
process(Clk2)
  variable Scan1 : integer range 0 to 3;
    begin
```

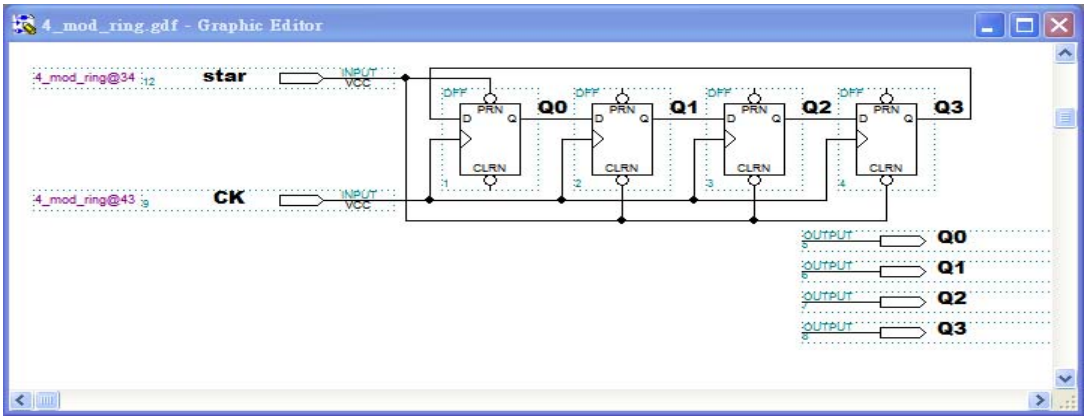
載入優先

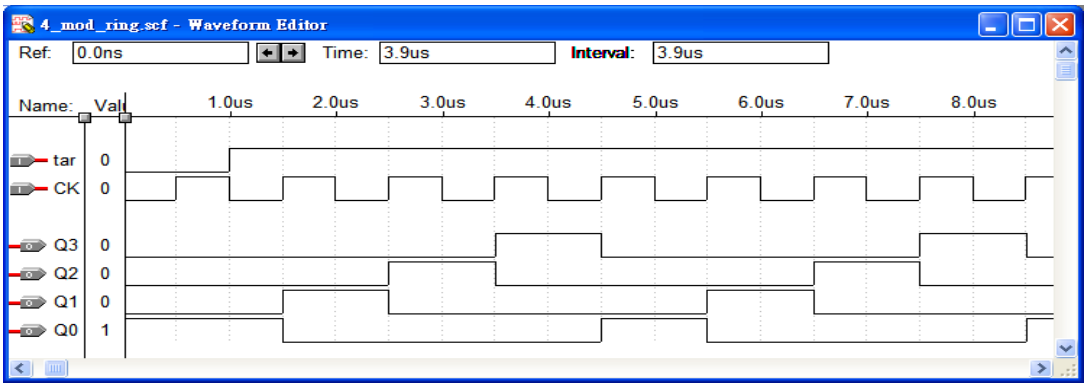
```
   --if Clk2='1' and Clk2'event then
      wait until clk2='1' ;
      if (Scan1=0) then
            Scan <= "1111" ;
            bcd <= "0000" ;
      elsif (Scan1=1) then
            Scan <= "1111" ;
            bcd <= "0000";
      elsif (Scan1=2) then
            Scan <= "1101" ;
            bcd <= Q1_temp;
      else Scan <= "1110" ;
            bcd <= Q0_temp;
      end if ;
      if Scan1 = 3 then Scan1 := 0 ;
        else Scan1 := Scan1 + 1 ;
      end if ;
end process scan_a ;
--*********************************
with bcd select
   seg <= "1111110"   when "0000" ,
          "0110000"   when "0001" ,
          "1101101"   when "0010" ,
          "1111001"   when "0011" ,
          "0110011"   when "0100" ,
          "1011011"   when "0101" ,
          "0011111"   when "0110" ,
          "1110000"   when "0111" ,
          "1111111"   when "1000" ,
          "1110011"   when "1001" ,
          "1001111"   when others ;

end A_multi_if ;
```

只有兩位數，可改為兩位數掃描
Scan 由 “1101” 與 “1110”之間變換

環形計數器：N 模計數器，需 N 個正反器，輸出輪流為 1，所以不用解碼。

4 模環形計數器

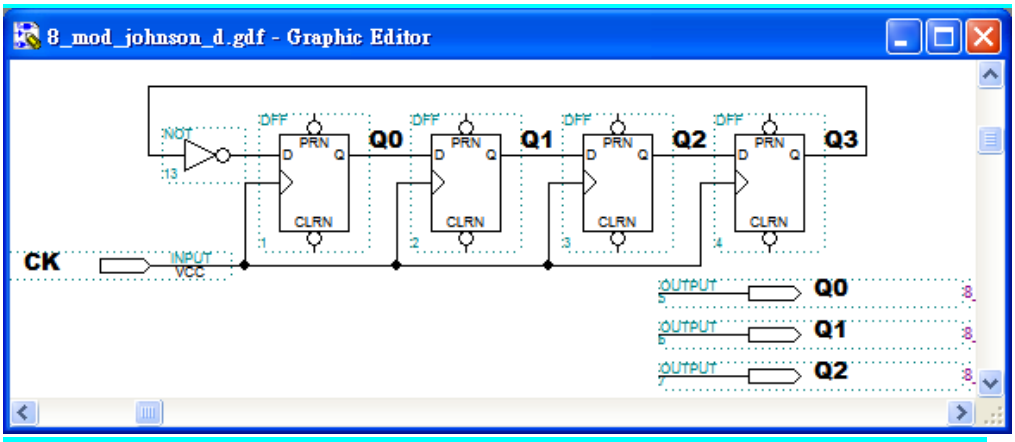


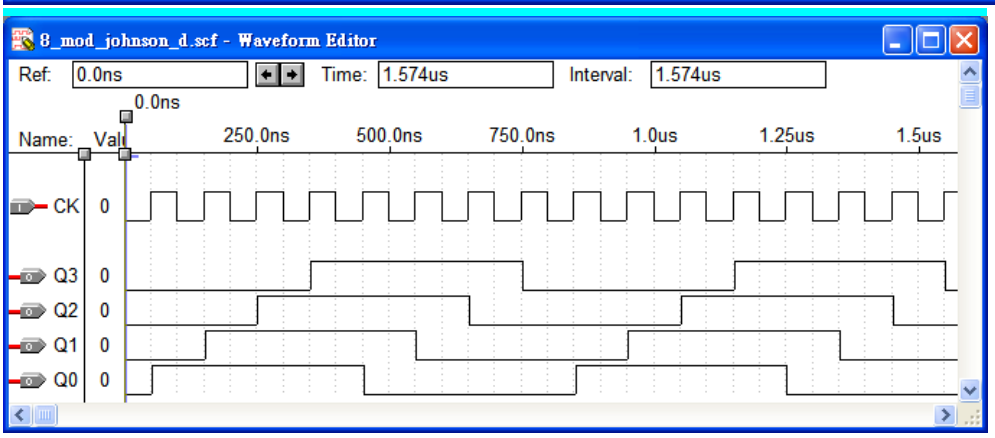


4 模環形計數器 VHDL

```
--*******************************
entity MOD_4_Ring is
  port ( STAR,CK : in    std_logic ;
              Q : out std_logic_vector(0 to 3)) ;
end MOD_4_Ring ;
--*******************************
architecture A_generic of MOD_4_Ring is
     signal Q_temp : std_logic_vector(0 to 3) ;
begin
     process(STAR,CK)
     begin
        if STAR='0' then    Q_temp <= "1000" ;--預置 Q_temp0=1
        elsif CK'event and CK='1' then
             for i in 1 to 3 loop
                  Q_temp(i) <= Q_temp(i-1);
             end loop ;
        Q_temp(0) <= Q_temp(3);
        end if ;
     end process ;
     Q <= Q_temp ;
end A_generic ;
```

強生計數器：將環形計數器最後一級正反器的輸出反相後，回接至第一級正反器輸入端。
又稱扭環計數器。





8 模強生計數器 VHDL

```
--*****************************
entity MOD_8_Johnson is
  generic ( N : integer := 3 ) ;
  port ( CK : in    std_logic ;
             Q : out std_logic_vector(0 to N)) ;
end MOD_8_Johnson ;
--*****************************
architecture A_generic of MOD_8_Johnson is
     signal Q_temp : std_logic_vector(0 to N) ;
begin
     process(CK)
     begin
        if CK'event and CK='1' then
             Q_temp(0) <= not Q_temp(N);
             for i in 1 to N loop
                  Q_temp(i) <= Q_temp(i-1);
             end loop ;
        end if ;
     end process ;
     Q <= Q_temp ;
end A_generic ;
```

除頻

上數除頻(除 20)VHDL

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity up_mdu is
port(
        fin:in std_logic;
        fout:buffer std_logic
      );
end up_mdu;
architecture beh of up_mdu is
begin
   process(fin)
    variable cnt:std_logic_vector(3 downto 0);
   begin
    if fin='1' and fin'event then
       if cnt>=9 then
            fout<=not fout;
            cnt:="0000";
         else
            cnt:=cnt+1;
         end if;
      end if;
   end process;
end beh;
```

fout 使用回授式描述法(fout<=not fout)，故需宣告為 buffer

下數除頻

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity dn_mdu is
port(
        fin:in std_logic;
        fout:buffer std_logic
      );
end dn_mdu;
architecture beh of dn_mdu is
begin
   process(fin)
     variable cnt:integer range 0 to 9;
   begin
     if fin='1' and fin'event then
        if cnt=0 then
             fout<=not fout;
             cnt:=9;
        else
             cnt:=cnt-1;
        end if;
      end if;
   end process;
end beh;
```

使用整數(integer)模式可增加程式的可讀性

cnt=0 電路合成是一個零檢知器，比 cnt>=計數上限值的電路簡單，當計數位元增加時，cnt>=計數上限值所佔用的硬體資源會相當可觀。

輸出頻率 fout=fin/[2(n+1)]

下數除頻

```
library ieee ;
use ieee.std_logic_1164.all ;
use ieee.std_logic_unsigned.all ;
use ieee.std_logic_arith.all ;
--******************************
entity seg7_4e is
  port ( clk1: in std_logic;
              clr: in std_logic:='1';
              scan: out std_logic_vector(3 downto 0);
              seg : out std_logic_vector(0 to 6);
              fout:buffer std_logic) ;
end seg7_4e ;
--******************************
architecture A_with_select_when of seg7_4e is
signal c1,c2,c3,c4,bcd     :       integer range 9 downto 0;
begin
    process(clk1)
      variable cnt:integer range 0 to 4999;
    begin
      if clk1='1' and clk1'event then
         if cnt=0 then
              fout<=not fout;
              cnt:=4999;
         else
              cnt:=cnt-1;
         end if;
       end if;
    end process;

inc_1:
process(fout)
       begin
         wait until fout='1' ;
       if clr='1' then
              c1<=9;
              c2<=9;
              c3<=9;
              c4<=9;
              elsif c1/=0 then c1<=c1-1;
              elsif c2/=0 then
                     c1<=9;
```

除 10KHZ

```
                c2<=c2-1;
            elsif c3/=0 then
                c1<=9;
                c2<=9;
                c3<=c3-1;
            elsif c4/=0 then
                c1<=9;
                c2<=9;
                c3<=9;
                c4<=c4-1;
            else c1<=9;
                  c2<=9;
                  c3<=9;
                  c4<=9;
        end if;
end process inc_1;
--***********************************
Scan_a:
process(clk1)
  variable Scan1 : integer range 0 to 3;
   begin
      wait until clk1='1' ;
      if (Scan1=0) then
              Scan <= "0111" ;
              bcd <= c4 ;
      elsif (Scan1=1) then
              Scan <= "1011" ;
              bcd <= c3;
      elsif (Scan1=2) then
              Scan <= "1101" ;
              bcd <= c2;
      else Scan <= "1110" ;
              bcd <= c1;
      end if ;
      if Scan1 = 3 then Scan1 := 0 ;
        else Scan1 := Scan1 + 1 ;
      end if ;
end process scan_a ;
--*******************************
with bcd select
  seg <= "1111110"    when 0 ,
            "0110000"    when 1 ,
```

```
        "1101101"    when 2 ,
        "1111001"    when 3 ,
        "0110011"    when 4 ,
        "1011011"    when 5 ,
        "0011111"    when 6 ,
        "1110000"    when 7 ,
        "1111111"    when 8 ,
        "1110011"    when 9 ,
        "1001111"    when others ;
end A_with_select_when    ;
```