FPGA系統設計\_第八章\_計時處理

**10867024 廖育賢**

# **objective- THE PROBLEM AND PURPOSE**

結合計時電路、除頻電路和波形控制，建立由案件控制的可程式頻率產生器。

# **procedure – DESIGN methods**

## 外部電路配置:

使用基本測試模組， 利用4組按鍵XC(3~0)並以8組LED燈輸出訊號XB(7~0) 。

XC(0):計數輸入，XC(1):正反控制，XC(2):相位控制，XC(3):計數歸零

XA(0):A相輸出波形，XA(1):B相輸出波形，XA(2):脈衝輸出波形，XA(3):方波輸出波形

## 內部電路設計:

此電路分成四個Block:

自由計數器 : 18位元，產生16Hz赫茲的抽樣訊號。

上下計數器 : 由按鍵的時間長短控制上數或下數，藉以改變頻率設定值。

頻率控制器 : 藉由設定值自由改變輸出的頻率。

波形產生器 : 分別產生AB相、脈衝和方波式的輸出訊號。

# **simulation results**

## program codes

### **ch08.vhd**

library IEEE;

use IEEE.std\_logic\_1164.all;

use IEEE.std\_logic\_arith.all;

use IEEE.std\_logic\_unsigned.all;

entity CH08 is

port

( CLK : in STD\_LOGIC;

XA : out STD\_LOGIC\_VECTOR (3 downto 0);

XB : out STD\_LOGIC\_VECTOR (7 downto 0);

XC : in STD\_LOGIC\_VECTOR (3 downto 0)

);

end CH08;

architecture CH08\_ARCH of CH08 is

signal SAMPLE : STD\_LOGIC;

signal FOUT,DOUT : STD\_LOGIC;

signal VALUE : STD\_LOGIC\_VECTOR (7 downto 0);

begin

FREE\_COUNTER : block

signal Q : STD\_LOGIC\_VECTOR (19 downto 0);

signal DLY : STD\_LOGIC;

begin

process (CLK)

begin

if CLK'event and CLK='1' then --CLK rising

DLY <= Q(19); --delay

Q <= Q+1; --UP counter

end if;

end process;

SAMPLE <= Q(19) and not DLY; --differential

end block FREE\_COUNTER;

UPDOWN\_COUNTER : block

signal Q : STD\_LOGIC\_VECTOR (7 downto 0);

signal RST,EC,DIR : STD\_LOGIC;

begin

process (CLK,RST)

begin

if RST='1' then --counter reset

Q <= "00000000"; --Q=0

elsif CLK'event and CLK='1' then --CLK rising

if EC='1' then --CLK enable

if DIR='0' then --UP/DOWN direction

Q <= Q+1; --UP counter

else

Q <= Q-1; --DOWN counter

end if;

end if;

end if;

end process;

RST <= not XC(3); --RST signal

DIR <= not XC(1); --DIR signal

EC <= SAMPLE and not XC(0); --EC signal

XB <= not Q; --output signal

VALUE <= Q; --parameter value

end block UPDOWN\_COUNTER;

FREQ\_GENERATION : block

signal Q : STD\_LOGIC\_VECTOR (8 downto 0);

signal DLY,EC : STD\_LOGIC;

begin

process (CLK)

begin

if CLK'event and CLK='1' then --CLK rising

DLY <= Q(8);

if EC='1' then --CLK enable

Q <= Q + ('0' & VALUE); --integration

end if;

end if;

end process;

EC <= SAMPLE;

DOUT <= Q(8) xor DLY; --differential

FOUT <= Q(8);

end block FREQ\_GENERATION;

PHASE\_GENERATION : block

signal EC,DIR,A,B : STD\_LOGIC;

begin

process (CLK)

begin

if CLK'event and CLK='1' then --CLK rising

if EC='1' then --CLK enable

A <= DIR xor B; --A phase generation

B <= DIR xor not A; --B phase generation

end if;

end if;

end process;

EC <= DOUT; --base frequency

DIR <= XC(2); --direction control

XA(0) <= A; --A-phase output

XA(1) <= B; --B-phase output

XA(2) <= not DOUT; --pulse output

XA(3) <= FOUT; --square output

end block PHASE\_GENERATION;

end CH08\_ARCH;

### **CH08.UCF**

NET CLK LOC = B12;

NET XA<0> LOC = N2;

NET XA<1> LOC = T1;

NET XA<2> LOC = T2;

NET XA<3> LOC = U2;

NET XB<0> LOC = M3;

NET XB<1> LOC = M4;

NET XB<2> LOC = W3;

NET XB<3> LOC = W4;

NET XB<4> LOC = Y3;

NET XB<5> LOC = Y4;

NET XB<6> LOC = W6;

NET XB<7> LOC = Y6;

NET XC<0> LOC = V1;

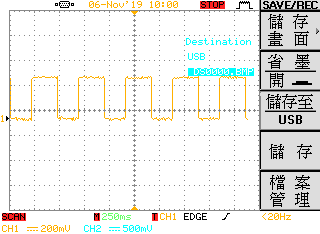
NET XC<1> LOC = V2;

NET XC<2> LOC = W1;

NET XC<3> LOC = W2;

## observations

### 1.脈衝輸出波形:



### 2. 方波輸出波形:

### 3. A相B相輸出波形

