FPGA系統設計\_第六章\_基本暫存器

**10867024 廖育賢**

# **objective- THE PROBLEM AND PURPOSE**

將除頻電路與查表處理相互配合，來設計一個0.5Hz,16種變化的走馬燈控制器。

# **procedure – DESIGN methods**

## 外部電路配置:

使用基本測試模組， 選擇12組LED燈輸出訊號以XA(3~0)和XB(7~0) 定義。

## 內部電路設計:

20MHz自由計數器之頻率組合表:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Q0 | 10MHz |  | Q10 | 9.7KHz |  | Q20 | 9.5Hz |
| Q1 | 5MHz |  | Q11 | 4.8KHz |  | Q21 | 4.7Hz |
| Q2 | 2.5MHz |  | Q12 | 2.4KHz |  | Q22 | 2.3Hz |
| Q3 | 1.25MHz |  | Q13 | 1.2KHz |  | Q23 | 1.19Hz |
| Q4 | 0.625MHz |  | Q14 | 610Hz |  | Q24 | 0.5Hz |
| Q5 | 0.3125MHz |  | Q15 | 310Hz |  |  |  |
| Q6 | 0.15625MHz |  | Q16 | 152Hz |  |  |  |
| Q7 | 78KHz |  | Q17 | 76Hz |  |  |  |
| Q8 | 39KHz |  | Q18 | 38Hz |  |  |  |
| Q9 | 19KHz |  | Q19 | 19Hz |  |  |  |

1.產生1Hz的控制週期。

2.定義16種XB (7~0)顯示走馬燈動態圖案。

3.取自由計數器最高的4位元處理。

# **simulation results**

## program codes

### **ch06.vhd**

library IEEE;

use IEEE.std\_logic\_1164.all;

use IEEE.std\_logic\_arith.all;

use IEEE.std\_logic\_unsigned.all;

entity CH06 is

port

( CLK : in STD\_LOGIC;

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

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

);

end CH06;

architecture CH06\_ARCH of CH06 is

signal Q : STD\_LOGIC\_VECTOR (24 downto 0);

signal CNT : STD\_LOGIC\_VECTOR (3 downto 0);

begin

process (CLK)

begin

if CLK'event and CLK='1' then

Q <= Q + 1;

end if;

end process;

CNT <= Q(24 downto 21);

XA <= not CNT;

XB <= "11111110" when CNT="0000" else

"11111100" when CNT="0001" else

"11111000" when CNT="0010" else

"11110000" when CNT="0011" else

"11100000" when CNT="0100" else

"11000000" when CNT="0101" else

"10000000" when CNT="0110" else

"00000000" when CNT="0111" else

"00000001" when CNT="1000" else

"00000011" when CNT="1001" else

"00000111" when CNT="1010" else

"00001111" when CNT="1011" else

"00011111" when CNT="1100" else

"00111111" when CNT="1101" else

"01111111" when CNT="1110" else

"11111111";

end CH06\_ARCH;

### **ch06.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;

## observations

