**LAB3\_pre\_1**

**Consider a 4-bit synchronous binary up counter (q3q2q1q0). 1.1 Draw the logic diagram**

**1.2 Construct Verilog RTL representation for the logics with verification.**

specification:

output:

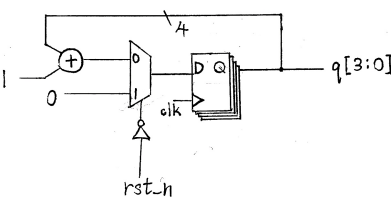
q[3:0]

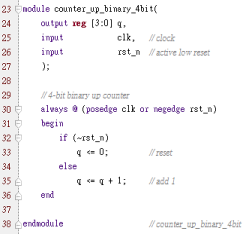
input:

clk // clock

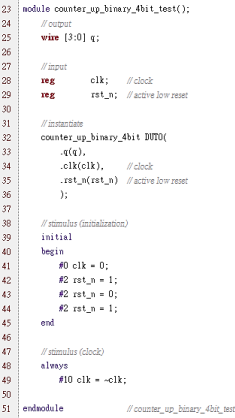
rst\_n // active low reset

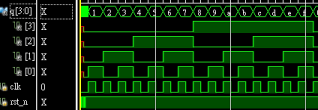
block-diagram/FSM:

RTL representation:



verification:





discussion:

這個 prelab 要做的是一個 4-bit 的二進位上數計數器，設計思路如下：用四個 FF 儲存當下數到多少，並將輸出 Q 連接到一個加法器。加一之後連到 FF 的 D 前 面，等過一個時脈 clk 再存入 FF 更新數值。另，加法器與 FF 之間連接一個 MUX， 用來 reset。

**LAB3\_pre\_2**

**Cascade eight DFFs together as a shift register. Connect the output of the last DFF to the input of the first DFF as a ringer counter. Let the initial value of DFF output after reset be 01010101. Construct the Verilog RTL representation for the logics with verification.**

specification:

output:

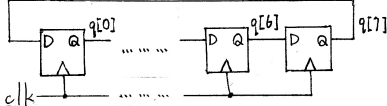
q[7:0]

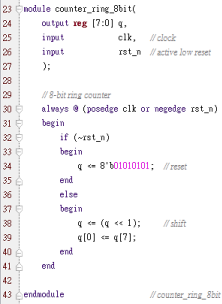
input:

clk // clock

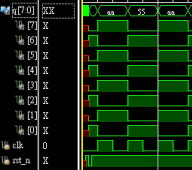
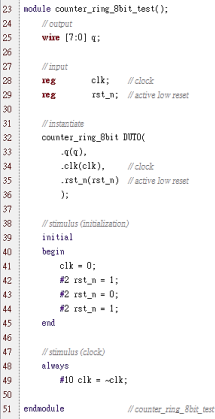
rst\_n // active low reset

block-diagram/FSM:

RTL representation:



verification:

discussion:

這個 prelab 要做的是 8-bit 的環形計數器，設計思路如下：拿八個 FF 來儲存要 shift 的內容，將這八個 FF 頭尾相連（上一個 FF 的 Q 連接到下一個 FF 的 D)， 最後一個 FF 的 Q 則拉回第一個 FF 的 D 前面。如此一來，每一個時脈 clk 都會 使這八個 FF 儲存的內容環繞移動一個單位。

**LAB3\_1**

**Frequency Divider: Construct a 27-bit synchronous binary counter. Use the MSB of the counter, we can get a frequency divider which provides a 1/227 frequency output (fout) of the original clock (fcrystal, 100MHz). Construct a frequency divider of this kind.**

**1.1 Write the specification of the frequency divider.**

**1.2 Draw the block diagram of the frequency divider.**

**1.3 Implement the frequency divider with the following parameters.**

| **I/O** | **fcrystal fout** |  |
| --- | --- | --- |
| **Site** | **W5** | **U16** |

specification:

output:

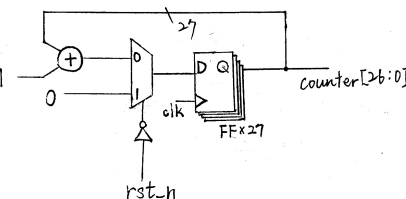
fout // around 1Hz but slightly slower

input:

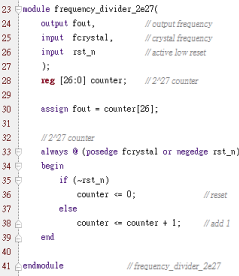
fcrystal // 100M Hz global clock

rst\_n // active low reset

block-diagram/FSM:



Verilog code:



implement:



discussion:

這個 LAB 要做的是一個除頻器，要將 FPGA 的十萬赫茲時脈的頻率除以二的二 十七次方。設計思路如下：因為二進位計數器天生就是除頻器，所以直接參考 prelab1 的做法，將上數計數器中 FF 的數量增加到 27 個，再把最高位的 bit 當 作輸出即可。

**LAB3\_2**

**Frequency Divider: Use a count-for-50M counter and some glue logics to construct a 1 Hz clock frequency. Construct a frequency divider of this kind. 2.1 Write the specification of the frequency divider.**

**2.2 Draw the block diagram of the frequency divider.**

**2.3 Implement the frequency divider with the following parameters.**

| **I/O** | **fcrystal fout** |  |
| --- | --- | --- |
| **Site** | **W5** | **U16** |

specification:

output:

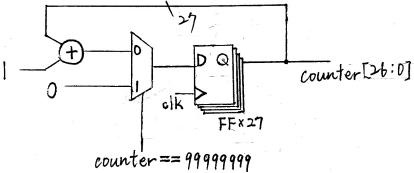
fout // 1Hz 50% duty

input:

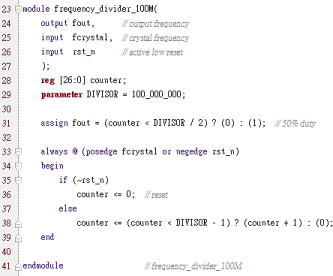
fcrystal // 100M Hz clock

rst\_n // active low reset

block-diagram/FSM:



Verilog code:



implement:



discussion:

這題跟上一題一樣要做除頻器，但這次要將 FPGA 的十萬赫茲時脈處理成剛剛 好一赫茲。我沿用上一題的概念，並稍加修改：由於二的二十七次方比十萬稍 大，所以這題須要另外多拿一個比較器來用。如果那 27 個 FF 的數值數到十萬

減一，就讓 MUX 輸出零，使 FF 的數值在下一個時脈歸零。然而，這麼做雖然 成功地製造出了一赫茲的輸出，但卻不是完美的一半一半（50% duty）。所以， 我最後再多拿一個比較器來用，如果那 27 個 FF 的數值小於十萬的一半，就輸 出零，如果大於則輸出一，如在 Verilog code 的 line 31 處所示。

**LAB3\_3**

**Implement pre-lab1 with clock frequency of 1 Hz. Use the following I/O to demonstrate the counter results.**

| **I/O** | **fcrystal** | **q3** | **q2** | **q1** | **q0** |
| --- | --- | --- | --- | --- | --- |
| **Site** | **W5** |  | **V19 U19 E19 U16** |  |  |

specification:

output:

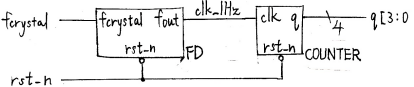
q[3:0] // prelab1 output

input:

fcrystal // 100M Hz global clock

rst\_n // active low reset

block-diagram/FSM:

**]**

TOP:



implement:



discussion:

我在上一題寫的除頻器性質不夠好，所以不能直接拿我上一題的除頻器作為 prelab1 的時脈。將上一題的除頻器部分修改重寫後的 code 如下圖： 我認為上一題的除頻器的問題在於把輸出變成 50% duty 的方法不妥。這一題我 改變方法，直接拿一個 TFF（line 39~47），將 FF 的最高位與 T 相接。並修改比 較器的 code（line 36），讓 FF 的數值數到十萬的一半時就重設成一。透過這樣 的方式所產生的 1Hz 時脈，與上一題一樣都是完美的 50% duty，而且可以直接 連接到 prelab1 當作四位元計數器的時脈。

**LAB3\_4**

**Implement pre-lab2 with clock frequency of 1 Hz. The I/O pins can be assigned by yourself.**

specification:

output:

q[7:0] // output from prelab2

input:

fcrystal // 100M Hz global clock

rst\_n // active low reset

block-diagram/FSM:



TOP:



implement:



discussion:

這一題跟上一題差不多，只差在這題是把一赫茲的時脈連接到 prelab2 的環形計 數器。但由於題目訂定環形計數器的八個 bits 儲存的內容是 01010101，所以實 作之後無法從 FPGA 板的 LED 上看出儲存的 bits 在繞圈（環狀移動），只感覺 它們在交替明滅。必須如下一題一般，設置不重複的 bits 才能看出循環效果。

**LAB3\_5**

**Use the idea from pre-lab2. We can do something on the seven-segment display. Assume we have the pattern of E, H, N, T, U for seven-segment display as shown below. Try to implement the scrolling pre-stored pattern NTHUEE with the four seven-segment displays.**

****

specification:

output:

digit\_active[3:0] // common anode activation

ssd[7:0] // 7 segment display

input:

fcrystal, // 100M Hz global clock

rst\_n // active low reset

block-diagram/FSM:





implement:



discussion:

這題要做的是一個能顯示「NTHUEE」的七段顯示器。然而，FPGA 上的七段 顯示器只有四個，塞不下六個英文字母，因此需要用計時捲動的方式輪流顯 示。本題的設計思路如下：沿用第一題的二十七位二進位上數計數器，取第十 九與二十位的頻率轉成 1-hot 並接到七段顯示器的 common anode。利用人眼的 視覺暫留使七段顯示器看起來同時顯示四個不同的字母。另拿一個四十八位元 的環形計數器，用二進位計數器的第二十六位來當作時脈，一次位移八個位 元。如此一來，該環形計數器就能將預先儲存好的字母樣式定時捲動，再透過 MUX 連接到七段顯示器上。