Lab1 Tutorial
亂數點名器

Table of Contents

1. 實驗目的

按下按鈕後，點名器會產生不斷跳動的亂數，數字跳動的頻率由高頻逐漸下降，最終停在某一個號碼上。

2. 實驗器材

DE2-115 FPGA 板

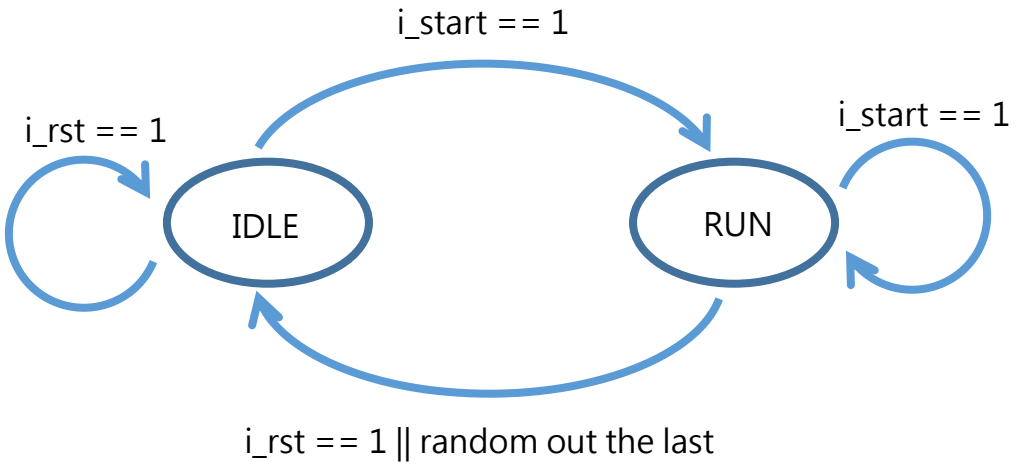
電源線

USB 傳輸線

3. 安裝Quartus

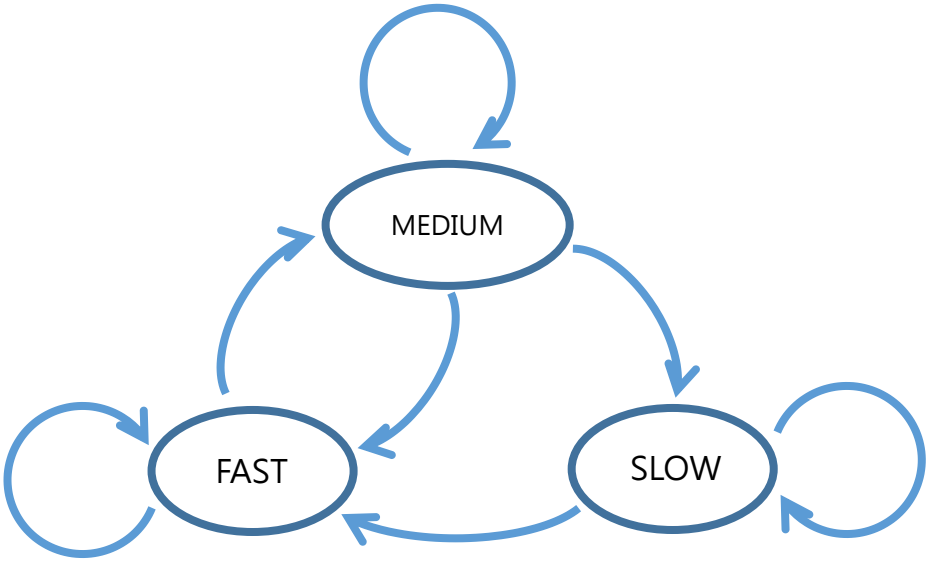
建議從NAS下載，也可以從官方網站直接下載。安裝的時候請選擇最新版本。

4. Finite State Machine 設計 (posedge reset 為例)



用 IDLE state 表示按下 start 之前的閒置狀態，RUN state 是按下 start 之後開始產生亂數，不管在哪個 state 只要 reset 啟動就會回到 IDLE state，並且把所有參數設為初始值。而在 RUN state 的時候按下 start 會重新產生亂數。

接下來介紹在 RUN state 裡的細節，用以達到從高頻到低頻的跳動頻率有許多不同的寫法，這裡提供其中一種。



RUN state 裡設計了三個 state，分別為 FAST、MEDIUM、SLOW，用來產生從快到慢 random out 數字的效果。可以藉由設置兩個 counter 來達到在三個 state 中停留的時間不同長短，以及在三個 state 中產生出不同個數的亂數。例如，在 FAST 停留 0.1 秒並輸出 2 個亂數，在 MEDIUM 停留 0.3 秒並輸出 3 個亂數，在 SLOW 停留 1 秒並輸出 5 個亂數。初始值為 FAST state，在任意 state 進行 reset 的時候就會把參數歸零跳回 FAST state，在 SLOW state 已經產生足夠的亂數也會回到 FAST state。

在 RUN 中的 state 數目是由程式編寫者自行設計，只要實現一個看起來頻率有由快到慢的亂數產生器即可。

5. Random Number Generator

產生亂數的方法亦有許多不同寫法，在此提供一種想法：利用 Linear congruential generator，產生 pseudo-random number，其公式如下:

$$X_{n+1} = aX_n + b \pmod{M},$$

其中常數 a、b 的選擇可以參考維基頁面。

另外，為了讓每次 reset 後產生的亂數都不同，可以將結果和使用者按下

的時間有關。最直接的方法即是每個 clock 都產生一個新的亂數，而在使用者按下的時間(假設在同個時間下按下的機率很小)取得的亂數變會不同。

另外需要注意的是，X 的 LSB 循環比較快，也就是說，比較容易取到相同的數字。因此從 X 擷取亂數的時候，可以擷取較高位數的 bit。例如，M 設定為 16bit，而亂數只需要 4bit，則可以取 X[15:12]而非 X[3:0]來得到較佳的亂數效果。