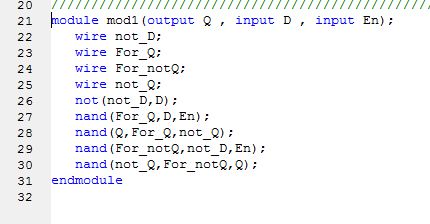
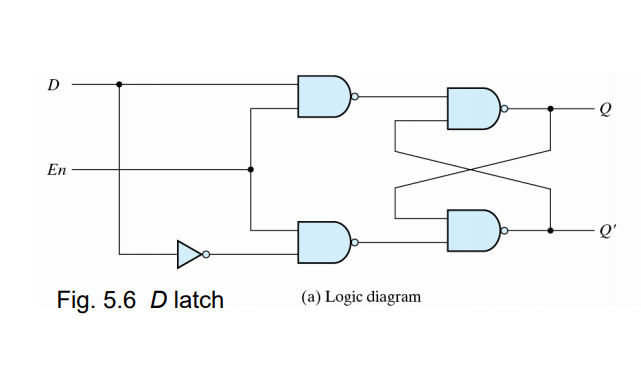
今天的實驗為實作Latch及Flip-Flop

第一個實驗為實作D Latch

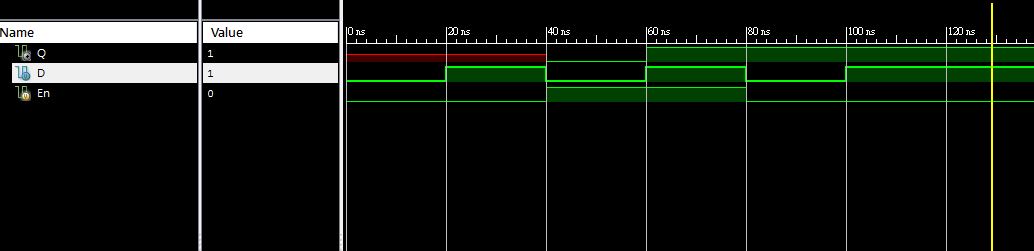


為一個很簡單的實作

比較會擔心的是:

nand何時會讀取回朔訊號(Q’(t)產出Q(t+1)或Q(t)產出Q’(t+1))

Ans:考慮訊號改變的前一刻Q(t)、Q’(t)狀況



可以看到當enable為1時Q會跟D相等 enable為0時保持不變

在40ps之前，因為enable一直沒有開啟，所以也無法成功assign值，Q為unknown

第二個實驗為Master-Slave D Flip-Flop

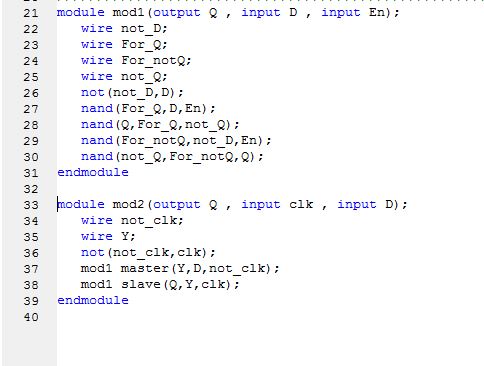
此Flip-Flop為正源觸發(postedge)

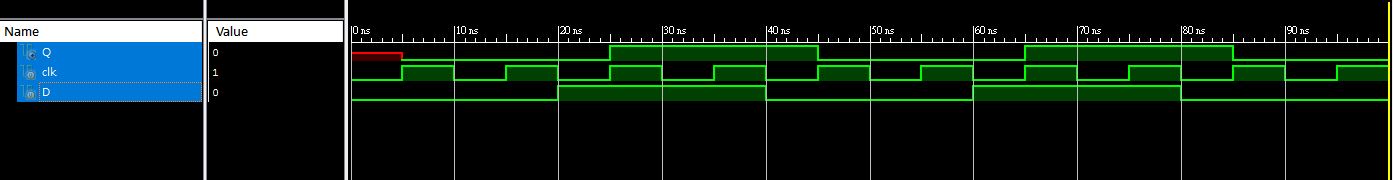
一張含有 螢幕擷取畫面 的圖片

自動產生的描述

主要就直接拿兩個實驗一的D Latch接在一起

要注意的是clk 必須兩個相反，不然就等同於原本的D Latch





可以看到訊號在20ps~25ps，雖然D=1，但clk不是1，所以無法啟動slave

25ps~30ps，clk=1，slave讀取master的Y output，因此Q為1

40ps~45ps，D=0，但clk不是1，目前改到的只有master

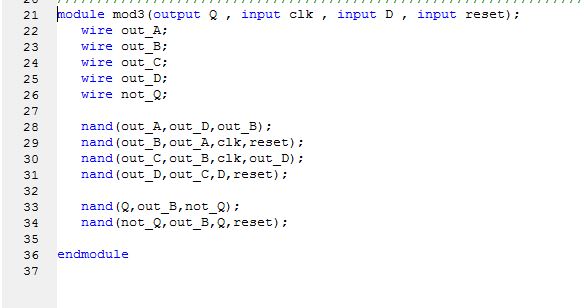
45ps~50ps，clk=1，slave讀取master的Y output，因此Q為0

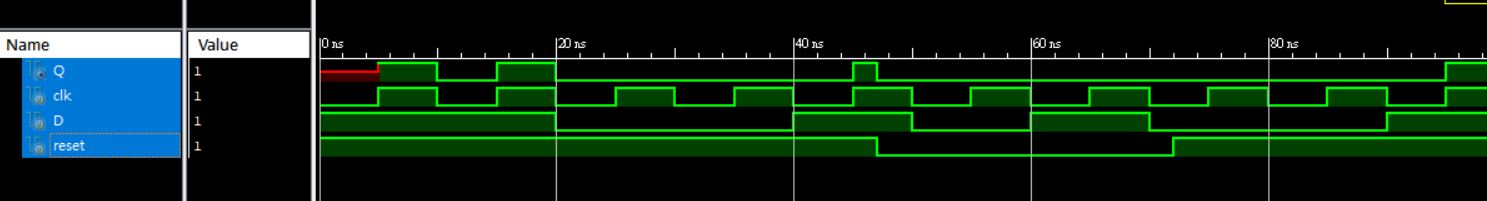
第三個實驗為D flip-flop with asynchronous reset

注意!! Reset 為0時才會進行reset，為1時不影響原本Flip-Flop

一張含有 文字, 地圖 的圖片

自動產生的描述





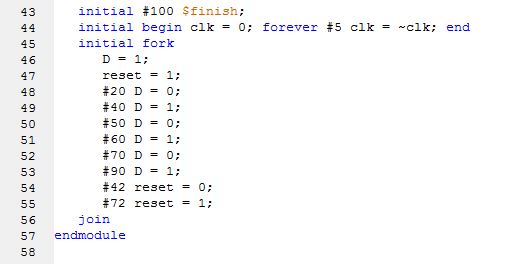
0ps~47ps reset = 1 ，行為如同 D Flip-Flop

47ps reset =0，一下去後直接馬上將Q reset 成0

47ps~72ps reset = 0，Q = 0

72ps~100ps reset = 1，行為如同 D Flip-Flop

Note:



新的testbench寫法

Line43: 表示執行100ps後停止模擬

Line44: 表示clk一開始為0，每5ps將clk反向

Line45~56: fork用法 ，利用 “# number”指定在number時間時執行後面指令，最後用join，完成fork完整語法