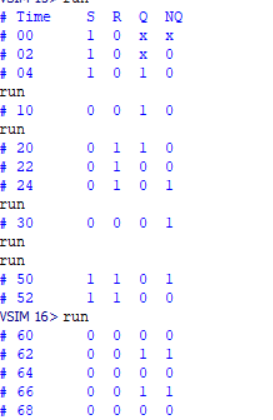
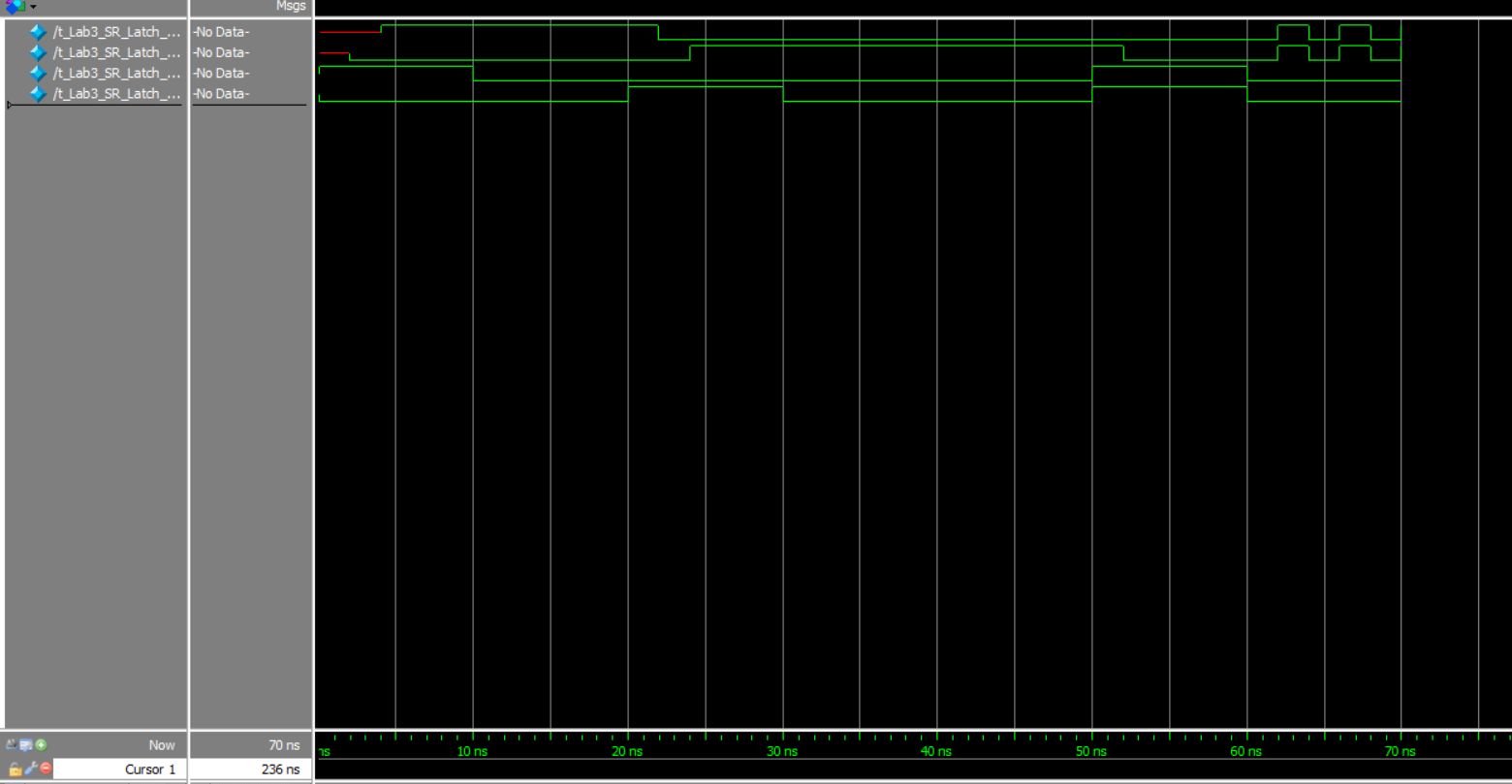
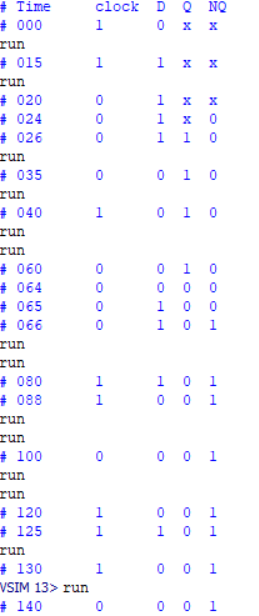
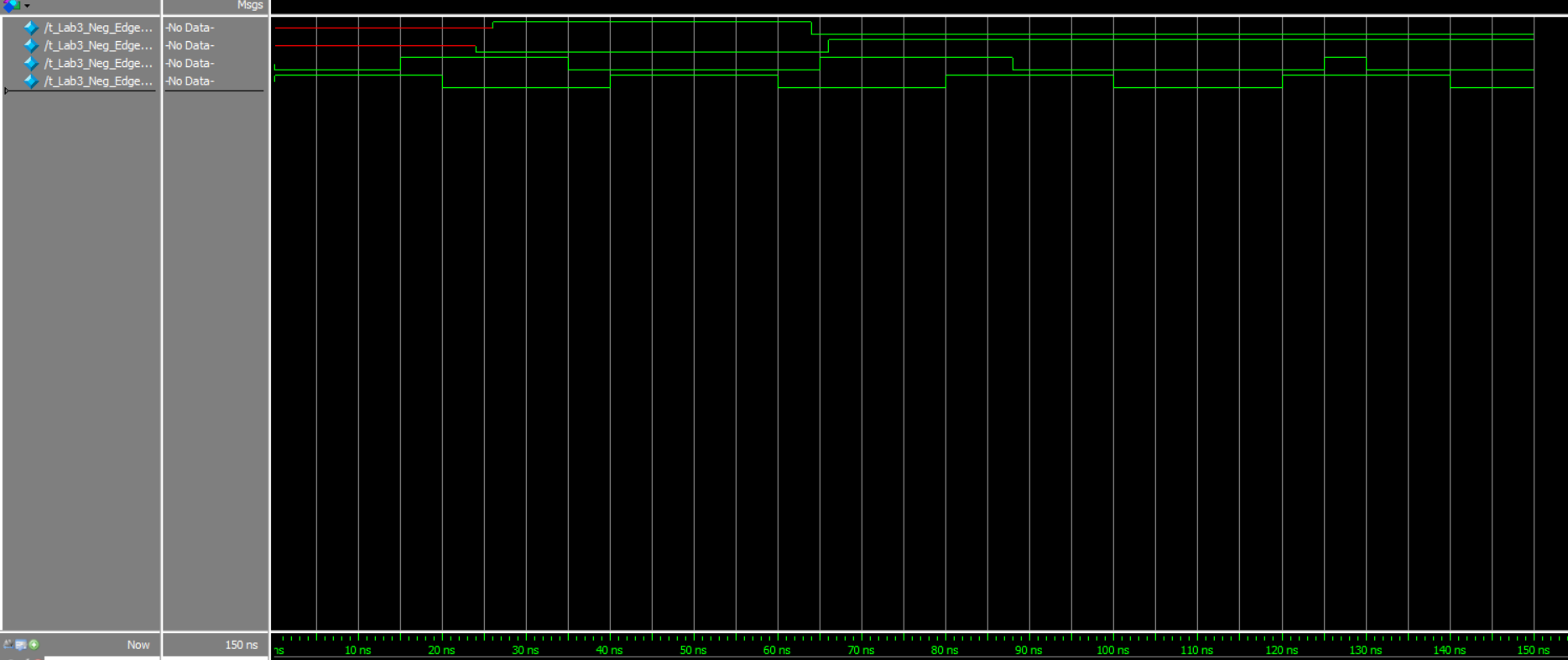
(1)

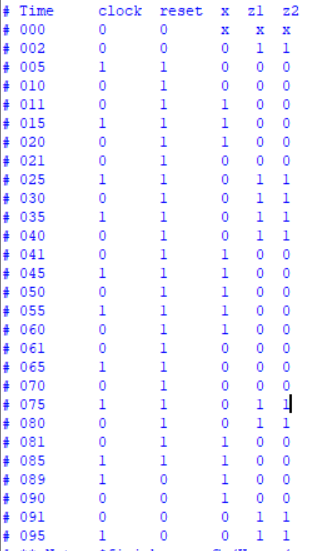
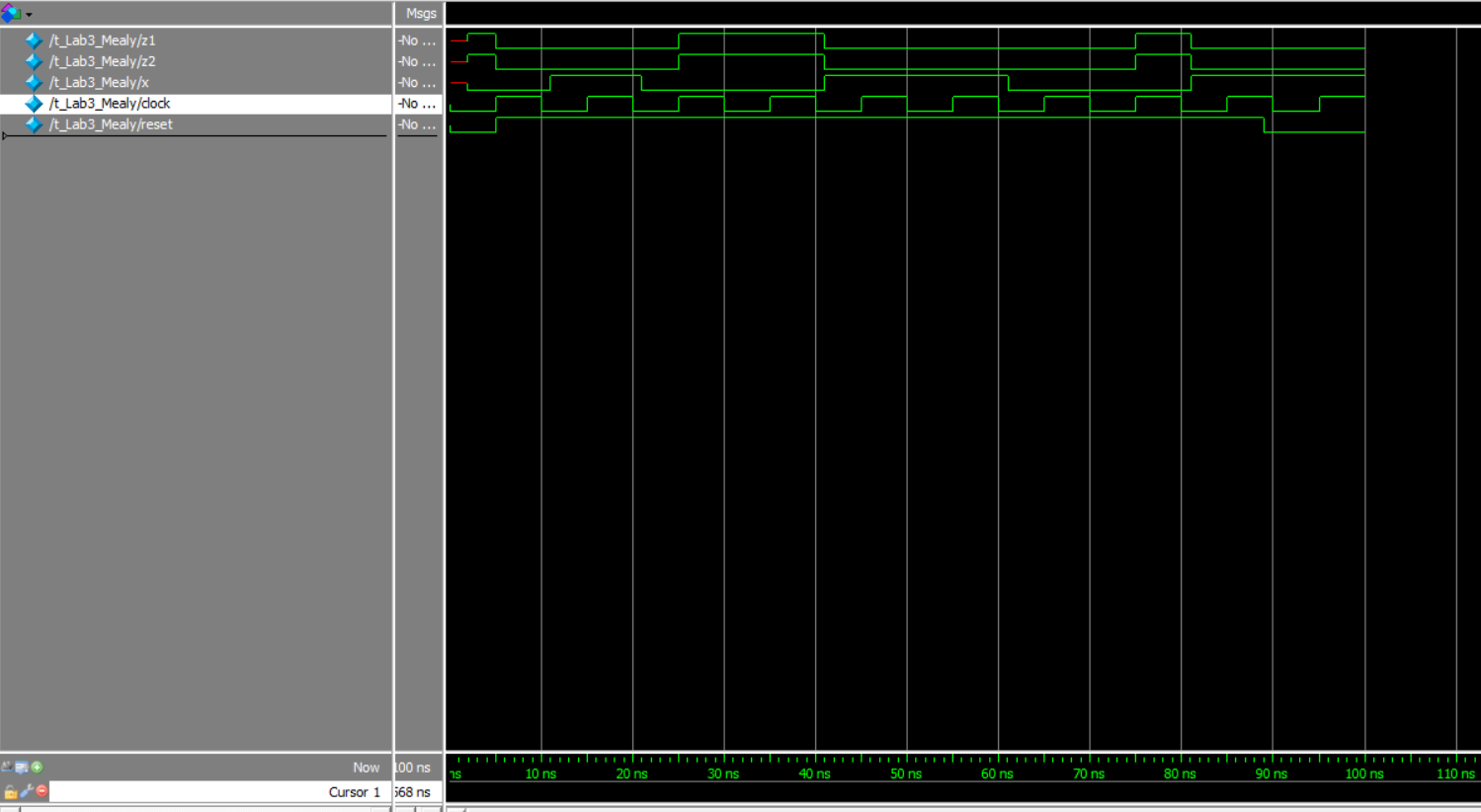


在時間為0的時候，因為S的輸入為1，所以經過2ns的延遲後，NQ的值變為0，Q的值還是維持unknown，此時0的值從NQ送到連接Q的NOR gate後，與原先輸入的R值一起決定了Q的值，所以當得到Q和NQ值的時間為4ns。在10ns的時候，因為S、R輸入值皆為0，所以Q、N的值維持不變，當20ns的時候，R為1，導致Q此時被設為0，這時Q的值會再傳到下方的NOR gate，與S一起決定NQ的值，所以得到穩定輸入值的時候是24ns。在30和40ns的時候R、S的輸入值皆為0，所以Q和NQ皆維持不變。在50ns的時候，因為輸入都為1，所以Q和NQ的值都變回0。在60ns的時候輸入值都是0，這時Q和NQ的值皆應維持不變，但因為Q和NQ此時也是皆為0，所以在經過delay後，Q和NQ都變成1，此時所得到的值再傳回NOR gate中，導致下一個Q和NQ的值又變成0，此結果會一直重複下去，直到輸入值有所改變為止，因為我的程式是在70ns時結束，所以看不到後續持續變動的波形圖，以上的描述與我的波形圖對應發現並無錯誤，所以我的模擬並沒做錯。

(2)

一開始當clock = 1的時候，輸入進SR-latch的值皆為0，所以Q和NQ的直接保持不變，當到20ns的時候，clock變為0，D為1時，此時輸入進latch的S、R值分別為1、0，所以Q在經過NOR gate的delay後，就會變成1、NQ為0。當35ns的時候D改變成0，此時輸入進SR-latch的值皆為0，所以Q和NQ保持不變，當40ns時，clock變成1，所以輸入進SR-latch的值皆為0，所以Q和NQ還是維持不變，在65ns時，clock變為0，D改為1，此時輸入進SR-latch的S、R值會變成0、1，所以最終Q和NQ會變成0、1。當80ns時，clock又變成1，所以輸入進SR-latch的值皆為0，所以Q和NQ維持不變。在100ns時，clock變成0，此時D為0，進入latch的S、R值分別為0、1，所以最終的Q和NQ為0、1。在120ns時，clock變成1，不管D如何變動，Q和NQ的值保持不變。以上的描述與波形圖所是相同，所以波形圖是正確的。

(3)



我testbench的設計是利用state diagram來設計，當reset生效後，state會從00開始，此時如果輸入為0，輸出就會是1，狀態會變成01，我這時的x值變成1，所以輸出是0，狀態會維持01，當x值變為0，狀態會變成10，輸出為0，x繼續維持0，狀態會變成00，輸出為1。之後x改成1，狀態會變成11，輸出為0，x繼續保持1，會維持在原狀態11，輸出變成0，當x變成0後，狀態會到01，輸出為0，因為x還是0，所以狀態會到10，輸出為0，此時因為x持續為0，所以輸出會變成1狀態變成00，此時reset啟動所以狀態回歸到00，因為x為，所以輸出為1。

(4)

這次的lab非常難做，我在C小題卡了很久，最後上網查資料才發現原本我把clock跟x設定成在同時有變化，導致輸出一直錯誤，因為compiler會自動挑選要先處理clock或是x，所以輸出值一直不對，而且還要考慮到clock的變化而產生的相對應結果，真的很辛苦，要在這麼多條件下找出問題所在十分不容易。