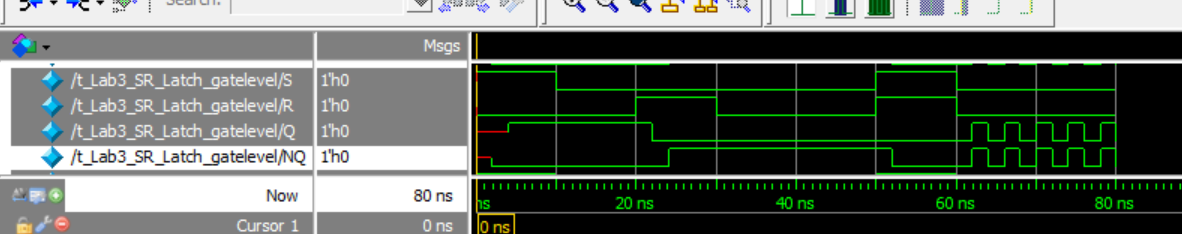
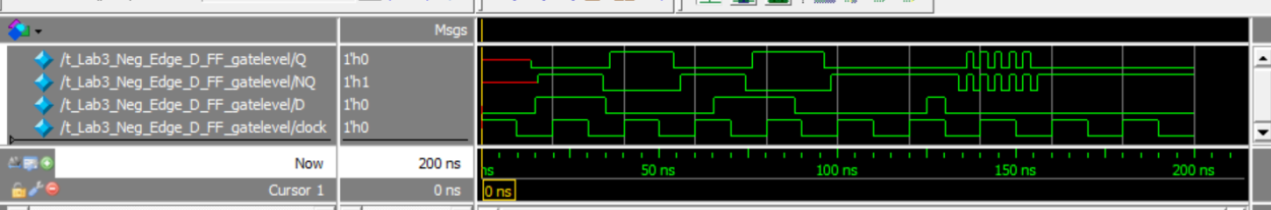
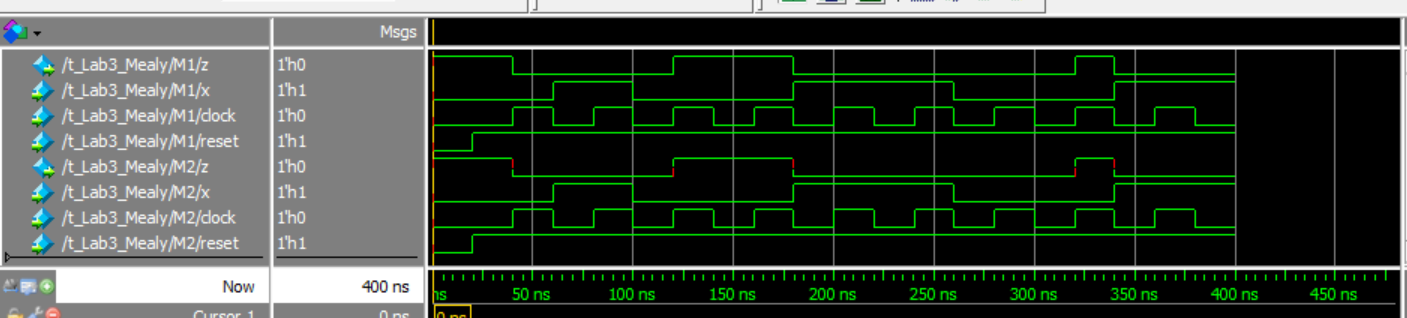
1.

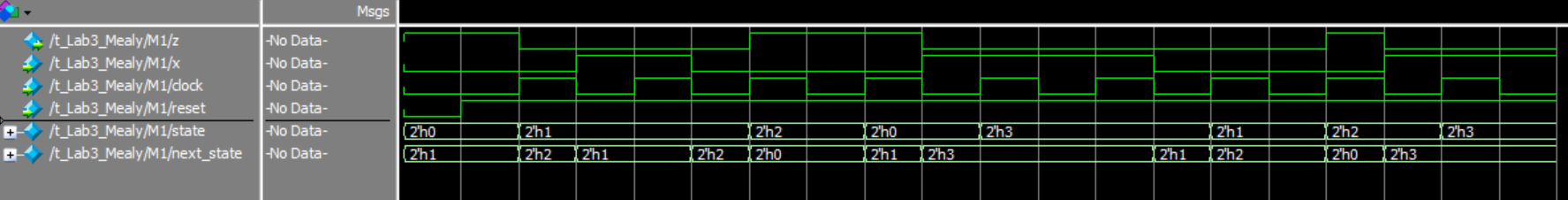
一開始因為S是1，所以會強制NQ變成0，所以延遲只有一個nor的2ns，而Q因為R一開始是零，所以要等到NQ的值過來才能確定，所以延遲會有兩個nor的4ns，而後面S,R是1,0時，Q,NQ會被set成1,0，SR在00時Q,NQ不變，SR在01時Q,NQ會變成0,1，最後面SR是11時，會讓Q,NQ都變成0，但Q,NQ會為補數，不可能相同，導致發生錯誤如圖。

2.

Q的延遲總共14ns，NQ是16ns，一開始的10ns是因為clock為1，讓SR都變0，而因為沒有前一個狀態，所以為不確定值，在10ns時，clock變為0，SR經前面的nor後變為01，這裡有一個nor的延遲，再經後面的SR latch後，Q經過一次nor，延遲共為(10+2+2)ns，NQ又經過兩個，延遲共為(10+2+4)ns。

之後可以發現，Q和NQ都是在clock由1->0時會有變化，在後面的亂流是因為SR同時被設成1的緣故。

3.



由上面第二張圖可以看到，狀態由00->01->01->10->00->11->11->01->10->11，八種狀態都走過，由state和x來判斷z也沒問題。

在設計時，先決定好要走的路徑，再根據該state的x要如何決定才能走到想要的下一個state。而因為clock是正緣觸發，所以在要觸發前的next state才是真的下一個state。

4.這次的作業前兩題感覺還好，不用花太多時間就可以完成，可是第三題卻讓我弄了好久才完成，光是決定要在什麼時間用什麼狀態和輸入，就畫圖畫了好久，也想了很久，才找到適合的數字。