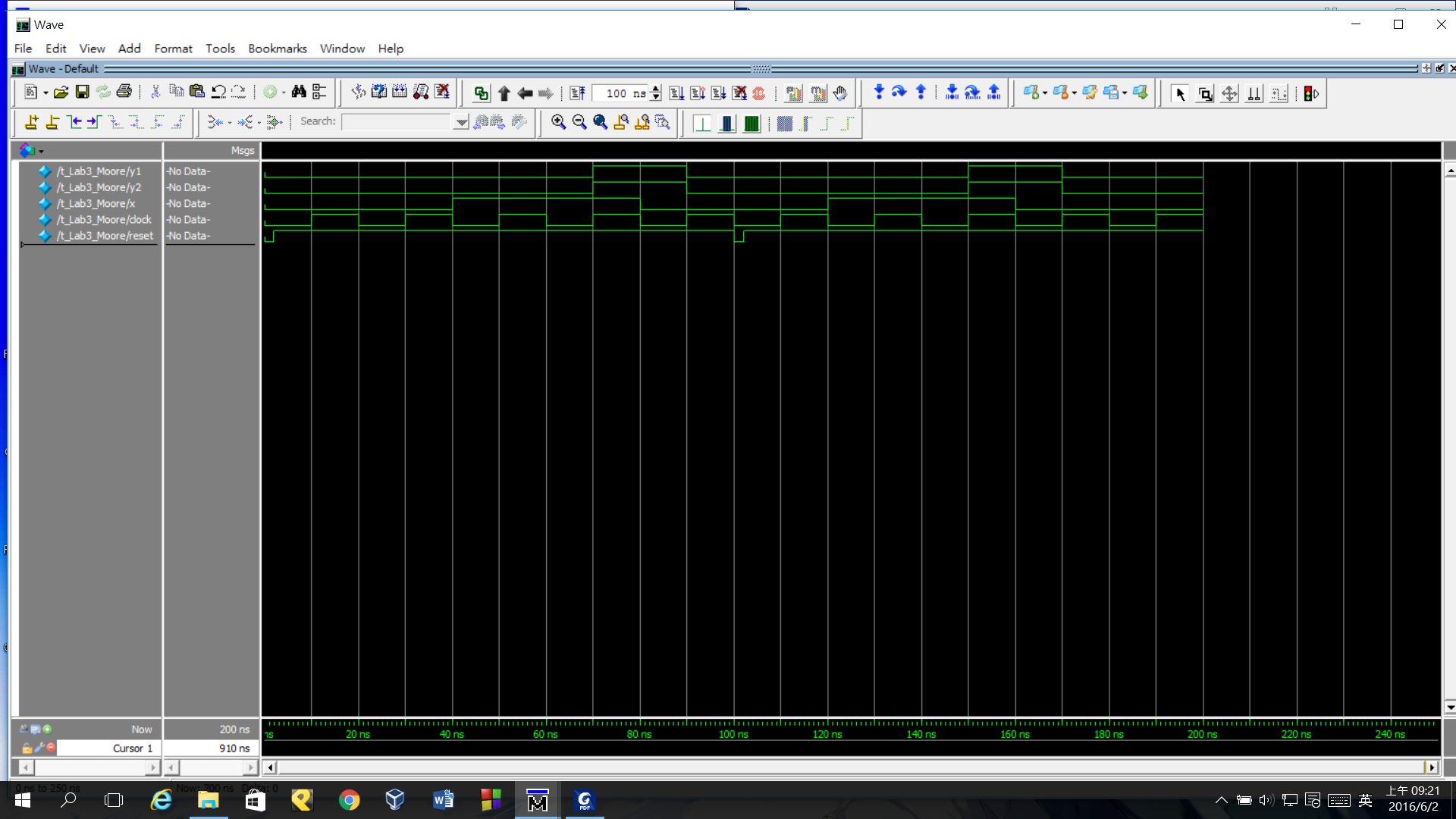
(1)



Testbench 之設計：

先設定reset = 0 ( state=S0 )，clock = 0(電路不能通過)，和x = 0。然後兩秒後reset = 1，每個十秒clock轉換一次，每隔40秒x轉換一次。在第100秒時reset = 0( state = S0)，兩秒後reset = 1。

預測每次clock=1時的輸出變化

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| State | S0 | S1 | S1 | S2 | S3 | S0 | S1 | S2 | S3 | S1 |
| Next state | S1 | S1 | S2 | S3 | S0(reset) | S1 | S2 | S3 | S1 | S1 |
| y | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

與實際的輸出值相符，故波形圖正確。(2A和2B波形圖相同，因是在寫同一個電路，故正確)

(2)

因為之前沒有碰過組合電路的verilog，所以在剛開始寫的時候會有一點在狀況外，到後來才漸漸上手。寫testbench的時候也一樣，因為之前沒又用過那麼多不一樣的語言所以也花了些時間搞懂他們。最後再看波形圖時一開始也是看不太懂(沒有認真上課的後果…….)，到開始寫報告逐一分析時才知道波形圖到底在表示甚麼，也算是報告讓我完全了解了這份作業吧!這是這學期最後一份lab了，下學期開始就要與更複雜的dlab奮戰了 ，希望這學期學到的東西我能一直記著然後到下學期好好的使用它們。