

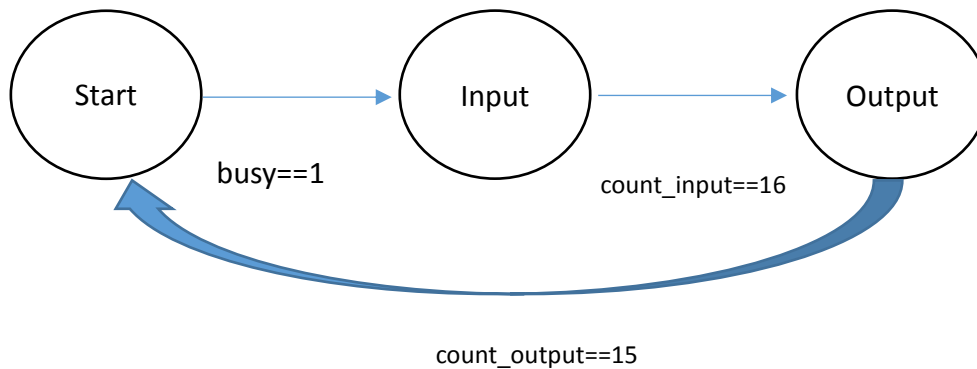
## CVSD\_HW2 Report

➤ R05943092 曾育為

➤ Method:

■ Use Finite State Machine : divide into three state(Start, Input, Output)

■ State Transition

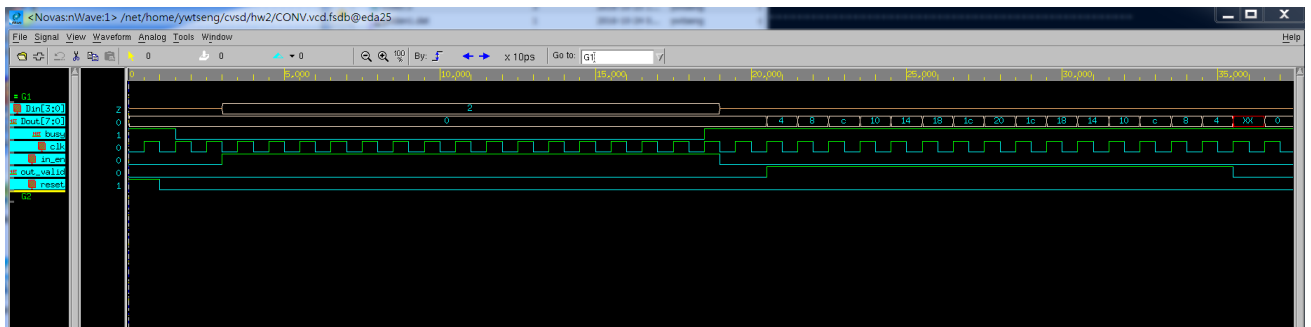


■ State set variable

	Start	Input	Output
busy_temp	1	0/1	1
out_valid_temp	0	0	1/0
count_input_temp	0	0/count+1	0
count_output_temp	0	0	count+1

- ✓ 在 input\_state 時，如果  $\text{count\_input} < 15$ ，則  $\text{busy\_temp} = 0$ ，否則為  $\text{busy\_temp} = 1$
- ✓ 在 input\_state 時，如果  $\text{count\_input} \leq 16$ ，則  $\text{count\_input\_temp} = \text{count\_input} + 1$ ，否則為  $\text{count\_input\_temp} = 0$
- ✓ 在 output\_state 時，如果  $\text{count\_output} < 15$ ，則  $\text{out\_valid\_temp} = 1$ ，否則為  $\text{out\_valid\_temp} = 0$

## ➤ Simulation Result



## ➤ Conclusion

- 所有 Variable\_temp 負責儲存暫存的值，再於 clk 來的時候，傳入 Variable 當中
- Reset signal 將所有變數回到初始值
- in\_en signal and busy signal 控制是否輸入資料
- out\_valid signal 控制是否輸出資料
- count\_input 和 count\_output 類似於計時器，計算目前 input 和 output 到哪個位數，並設定一個臨界值，協助狀態轉換