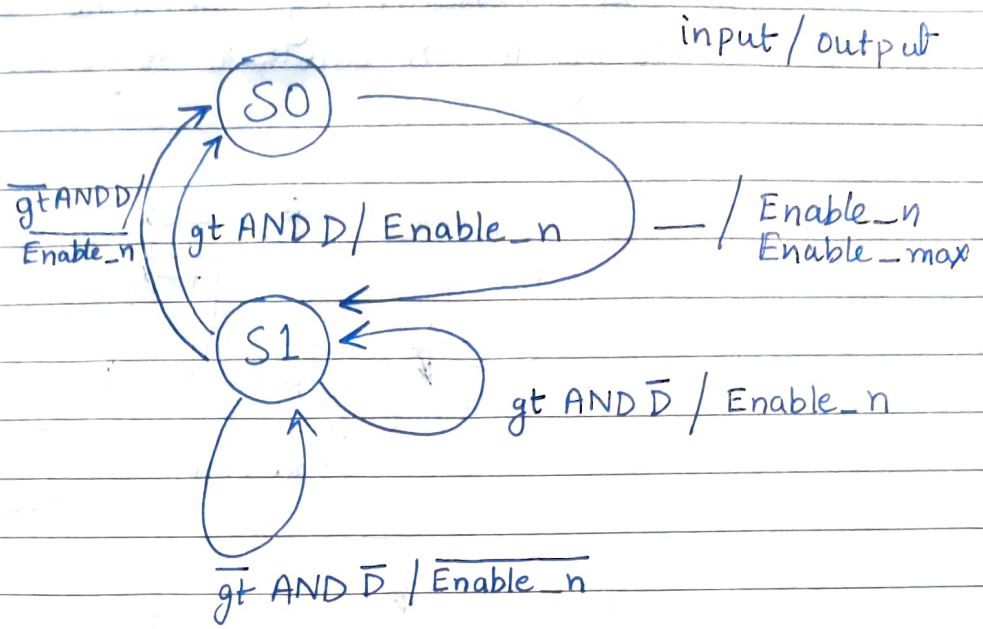
__/__/____

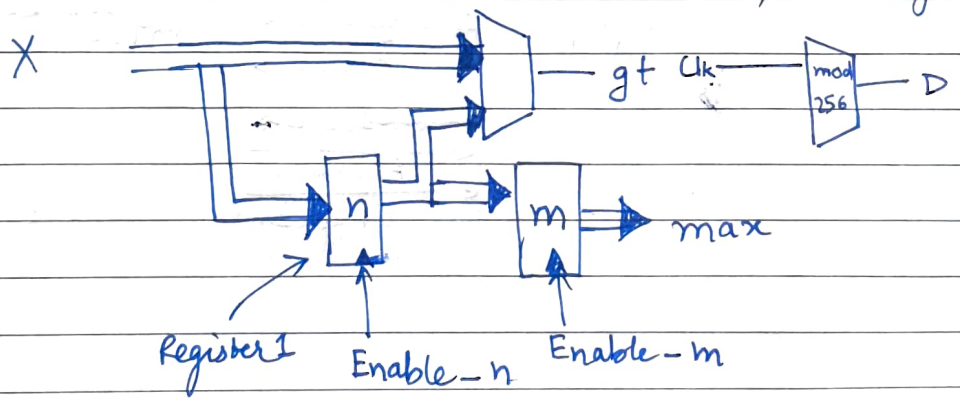
# Tut 7

① (i)



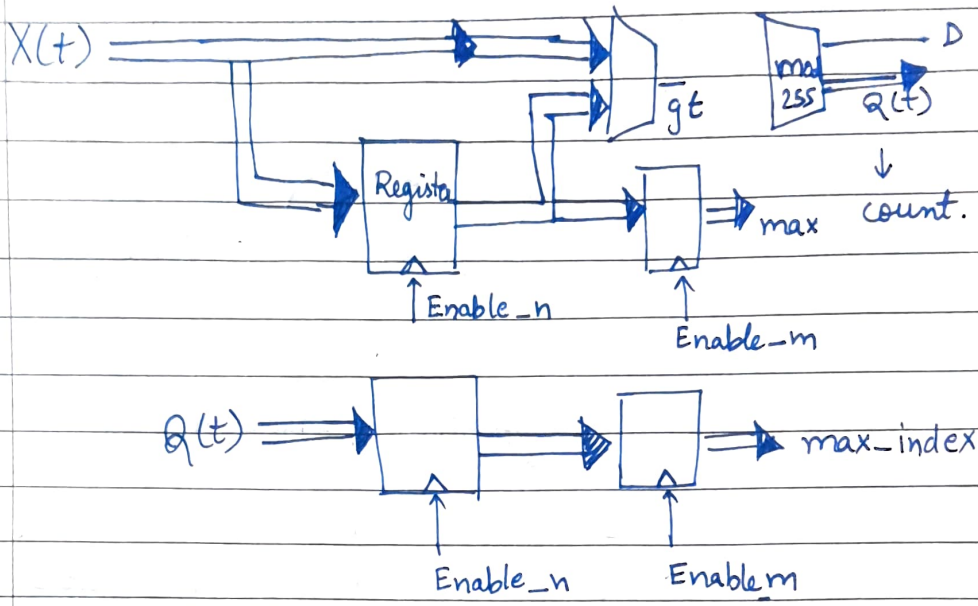

gt → 1 when input > stored value
D → 1 when counter hits 255
Enable_n → if 1 then register 1 stores input.
Enable_max → if 1 then read the value in Register 1 & keep in Register 2

Data Part: We need 2 registers, 1 comparator. We also need a counter, with logic to turn D=1 at 255

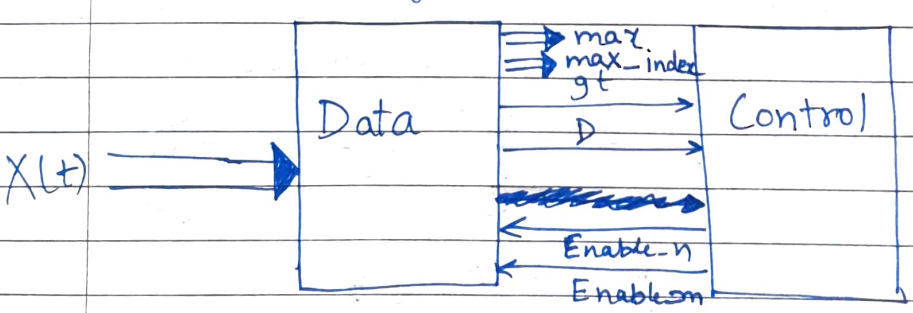



Classifying Status signals & control signals:
D & gt are generated by the datapart, so they are Status signals.
Enable signals are given by control so they are control.

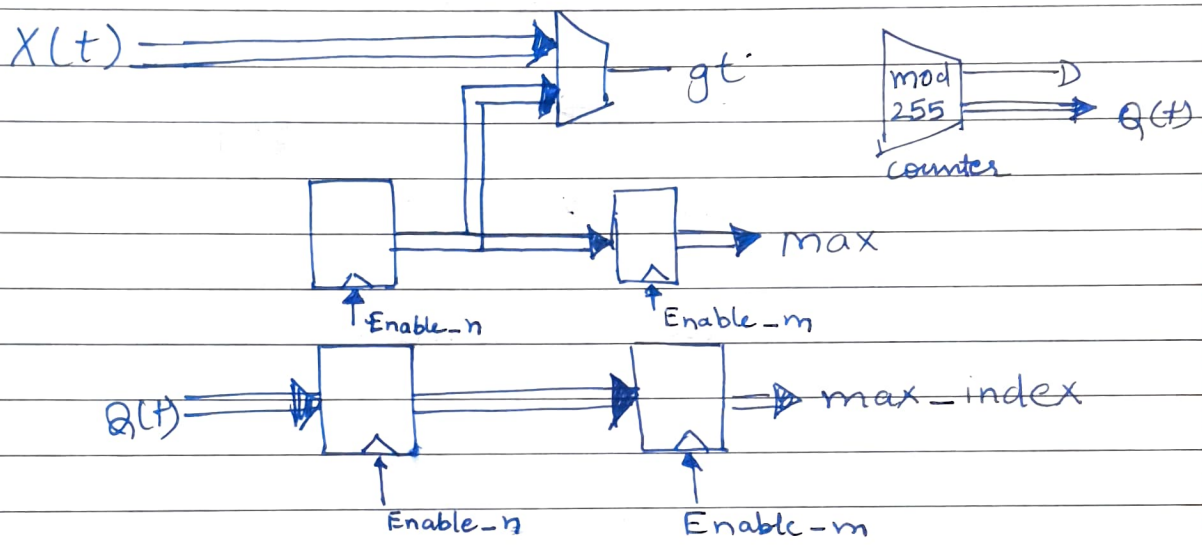
(ii) Can be achieved by using the counter output an storing it in another register.

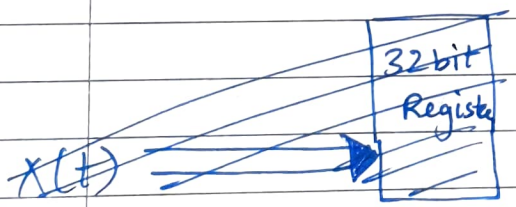


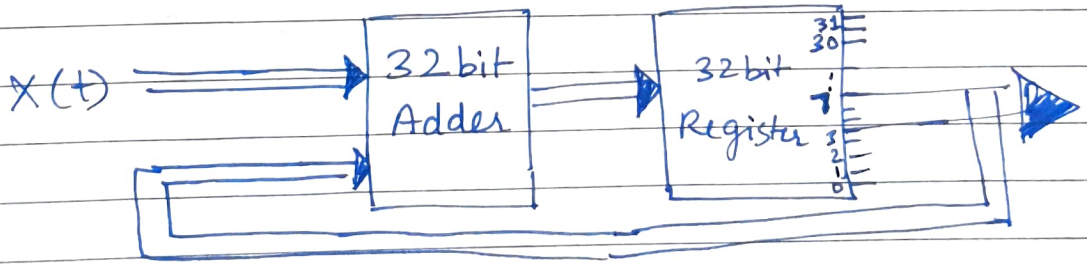

State machine will be same, since only data part is the modified. Also, new signal Q(t) is completely internal.



(iii) Sum of 256 16 bit numbers cannot exceed $2^{16} \times 256 = 2^{24}$.
hence a 32 bit Register & 32 bit adder will suffice.



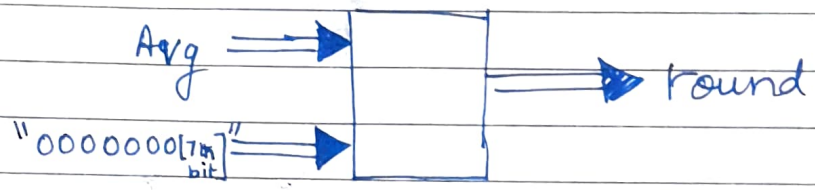




Avg output will ignore ~~first~~ last 8 bits in the 32 bit Register. So Avg output is taken as bits from 8 to 23 (inclusive

(iv) Rounding means if output is...00110.1 then in our avg answer, we add 1, i.e. 00111, otherwise avg is the answer.

So we can add the 7th bit of the output to bits 8 to 23.




rest all will be same.