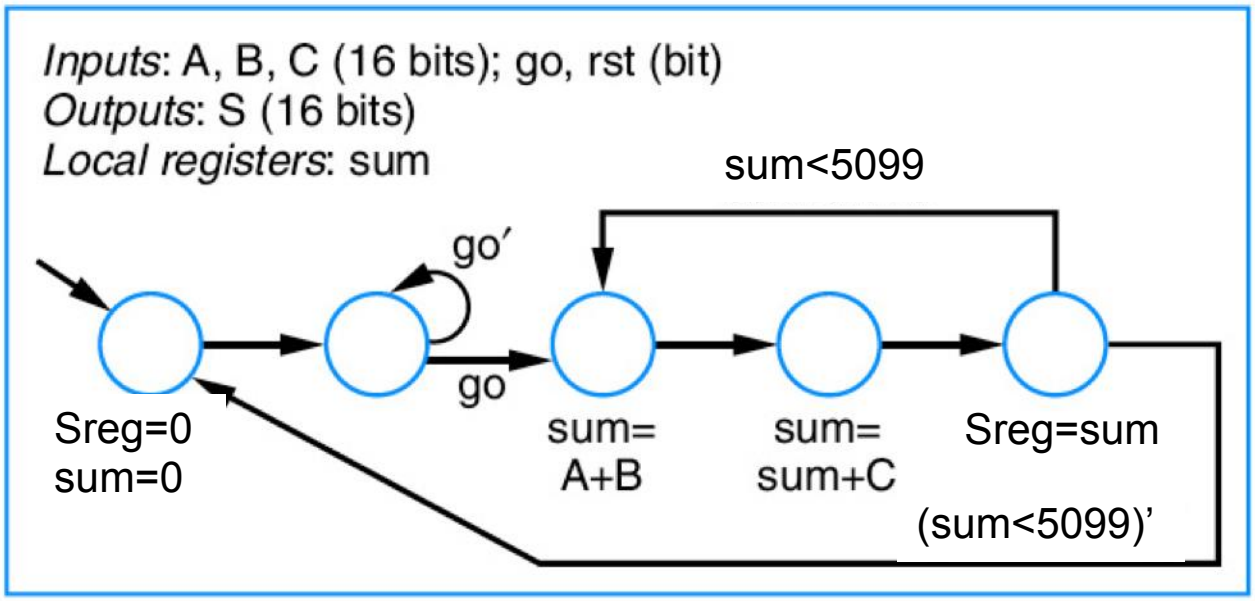
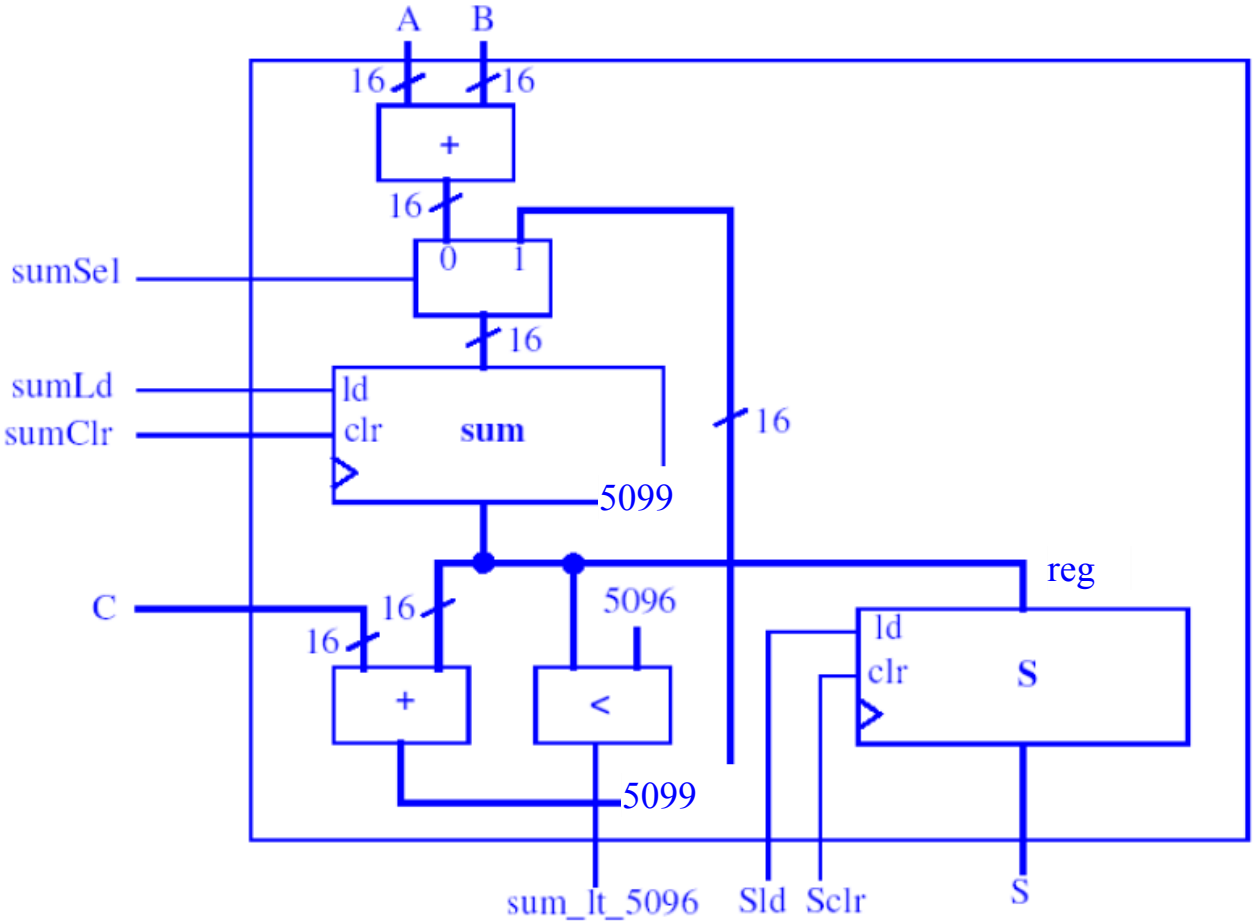
Ve270 Introduction to Logic Design

Homework 9

1. Problem 5.7. (15 points)

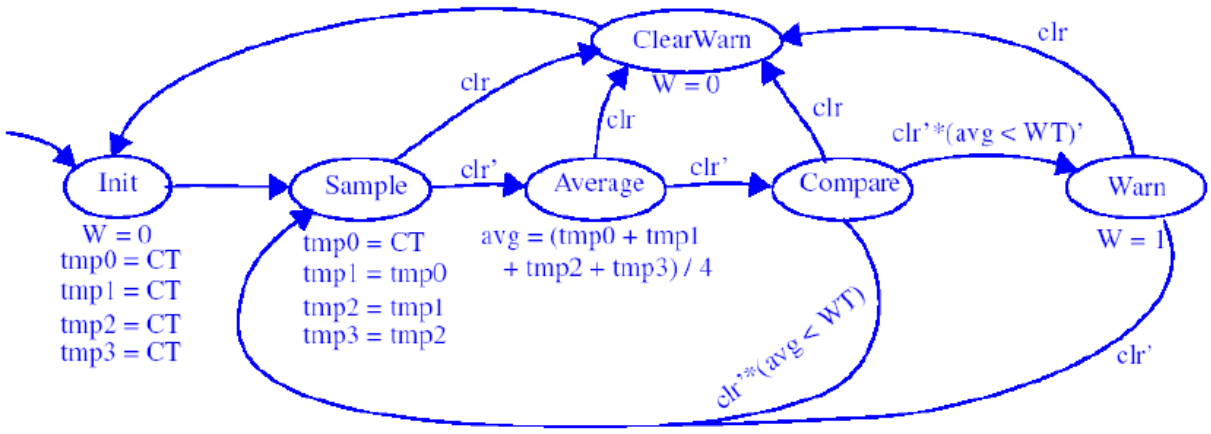




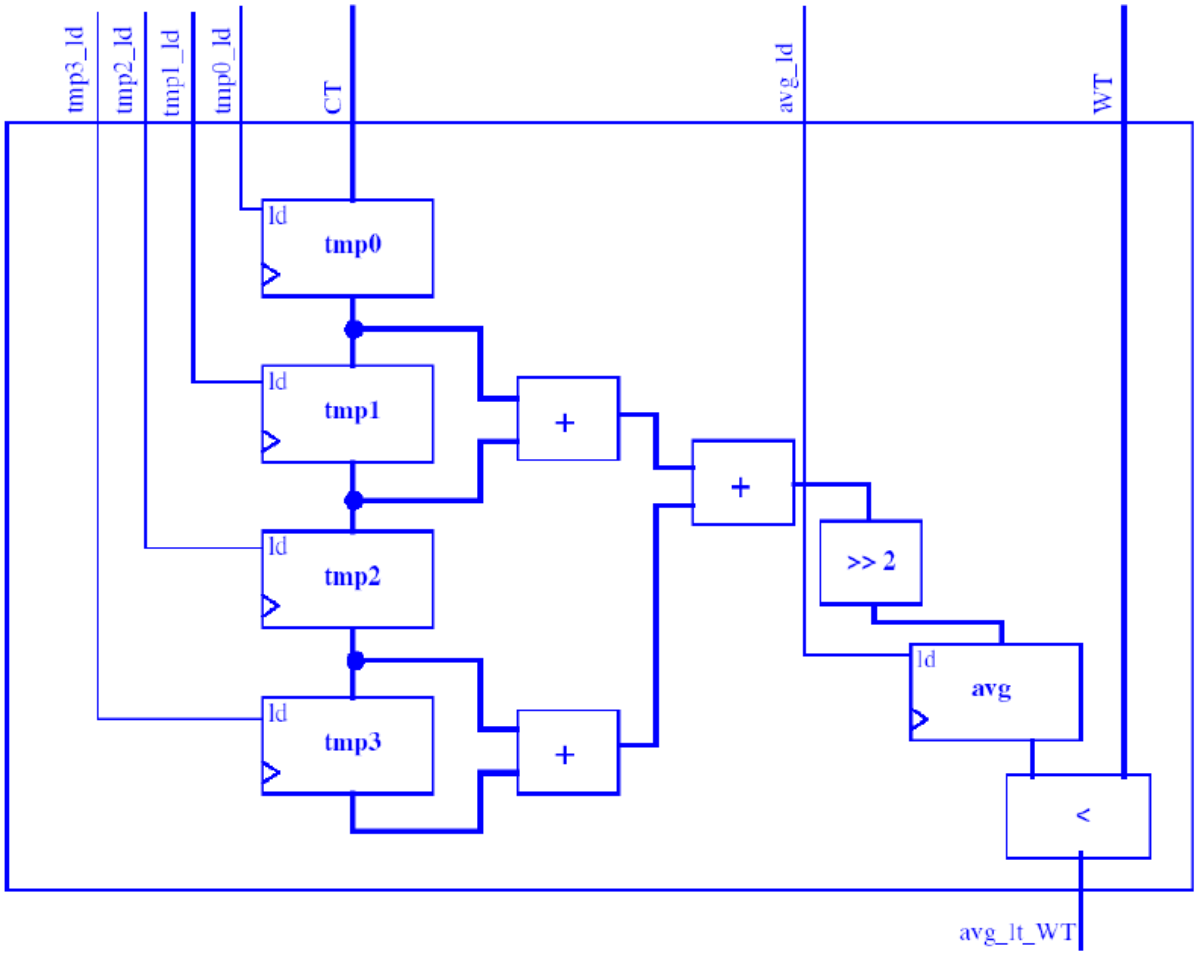
2. Problem 5.14. (15 points)

Step 1 - Capture a high-level state machine

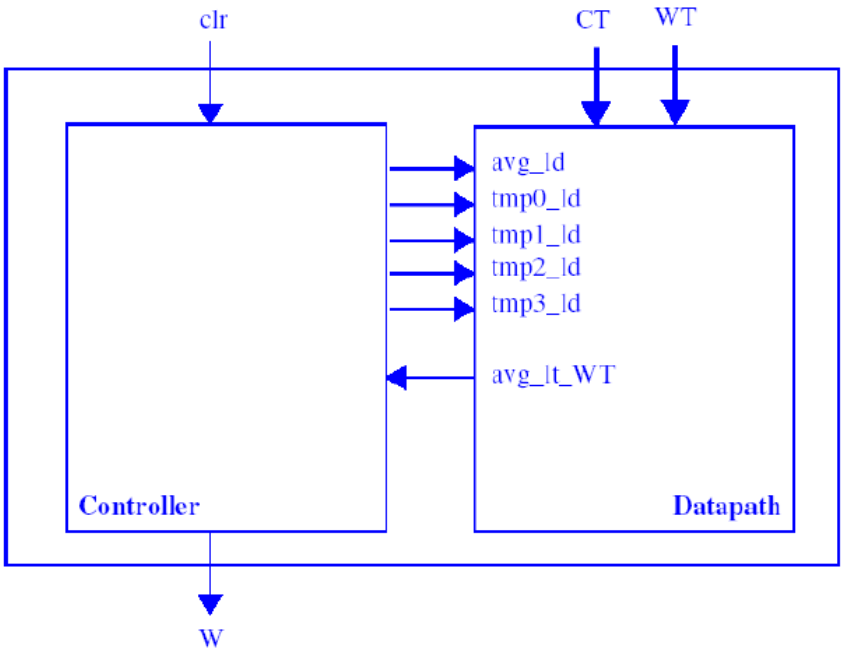
Inputs: CT, WT (32 bits); clr (bit)
Outputs: W (bit)
Local Registers: tmp0, tmp1, tmp2, tmp3, avg (32 bits)



Step 2 - Create a datapath

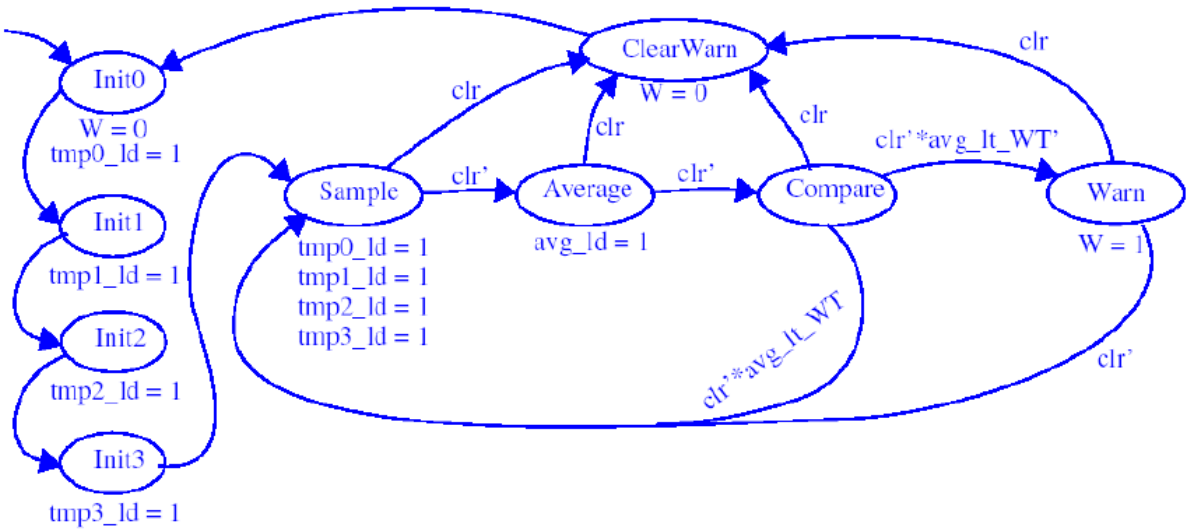


Step 3 - Connect the datapath to a controller

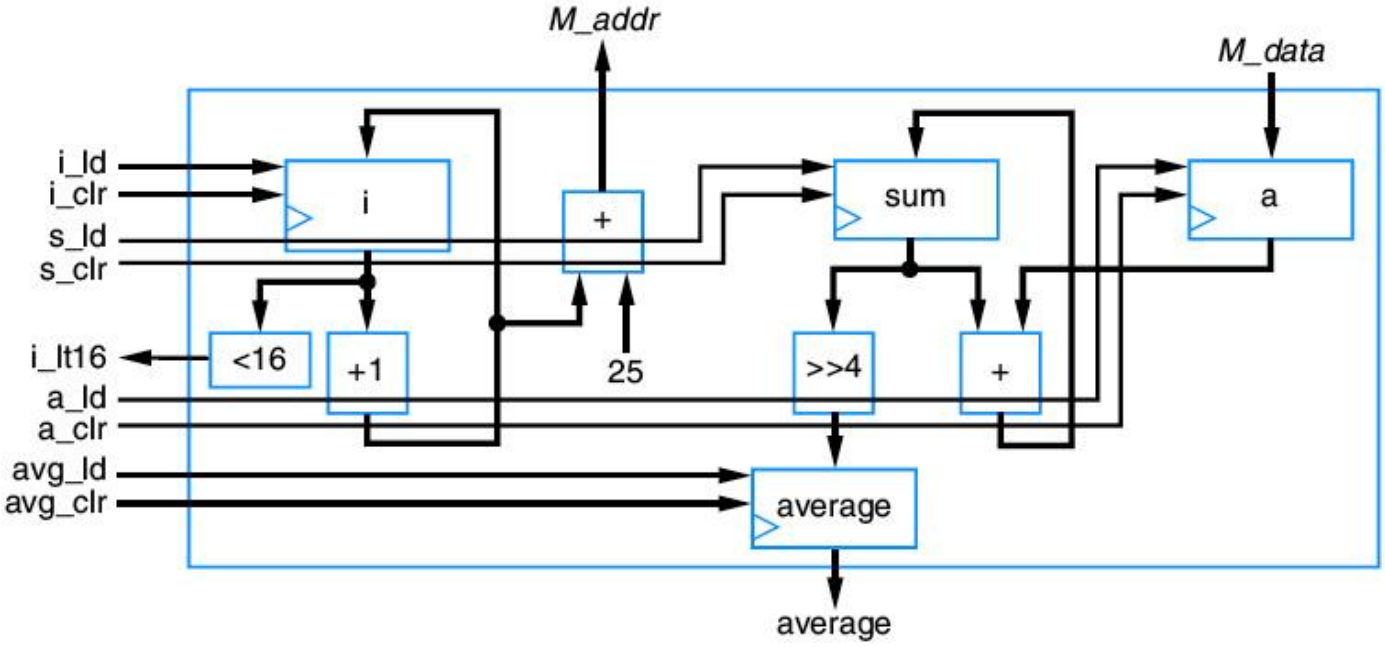


Step 4 - Derive the controller's FSM

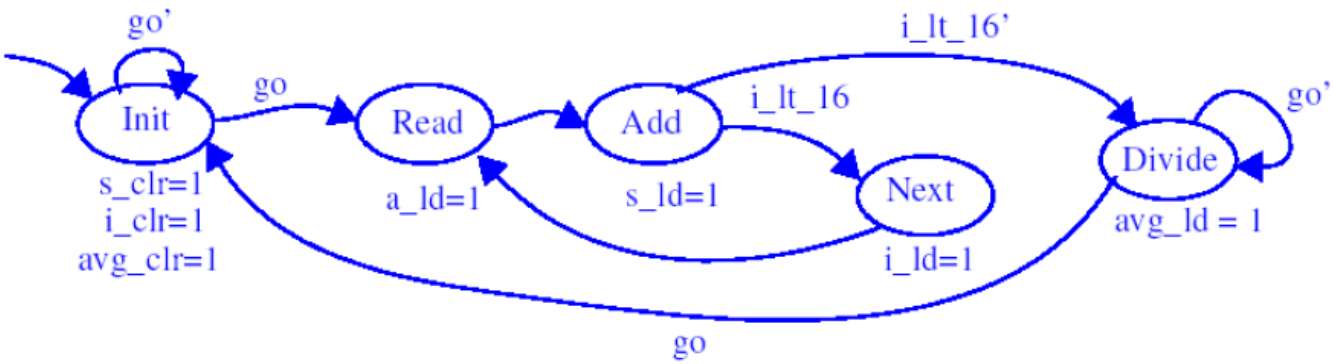
Inputs: CT, WT (32 bits); clr (bit)
Outputs: W (bit)
Local Registers: tmp0, tmp1, tmp2, tmp3, avg (32 bits)



3. Problem 5.16 (35 points)

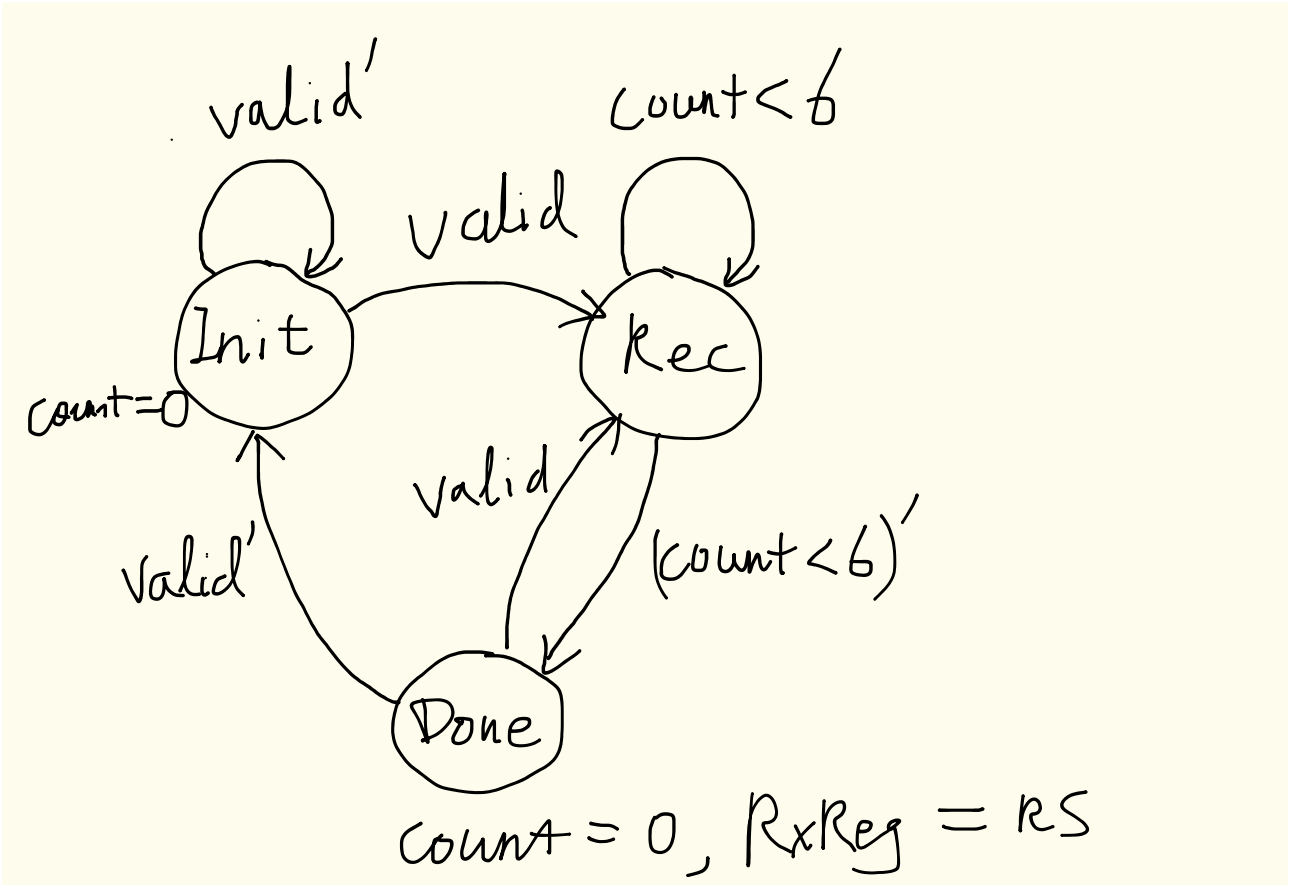


Inputs: go, i_lt_16 (bit)
Outputs: s_clr, i_clr, avg_clr, s_ld, i_ld, a_ld (bit)

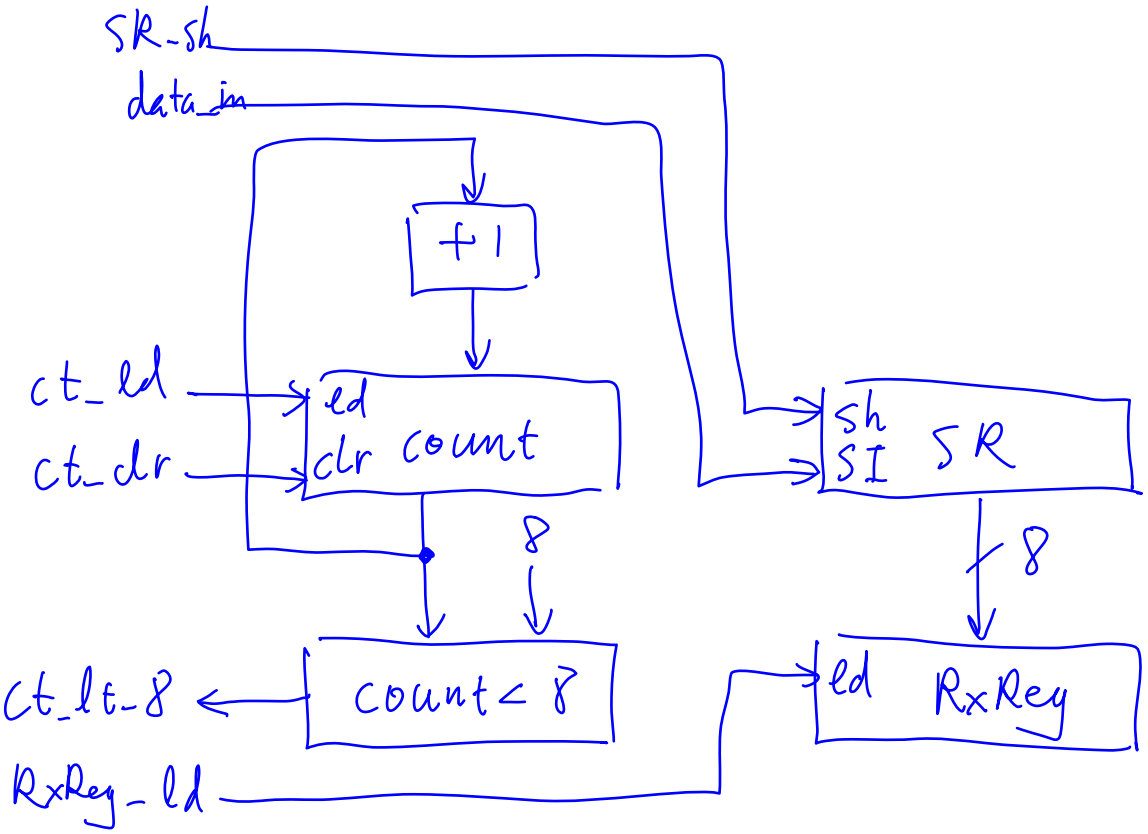


4. Design a circuit called ***Receiver*** that receives two single bit signals, **Valid** and **Data_in**, from another device called ***Transmitter***. The **Valid** signal sent from the ***Transmitter*** will be a 1-clock cycle pulse. After the ***Receiver*** receives the **Valid** pulse, it will start receiving 8 bits through port **Data_in**, bit by bit. After the 8 bit information is received, it should be copied into an 8-bit register called **RxReg**.

Step 1:

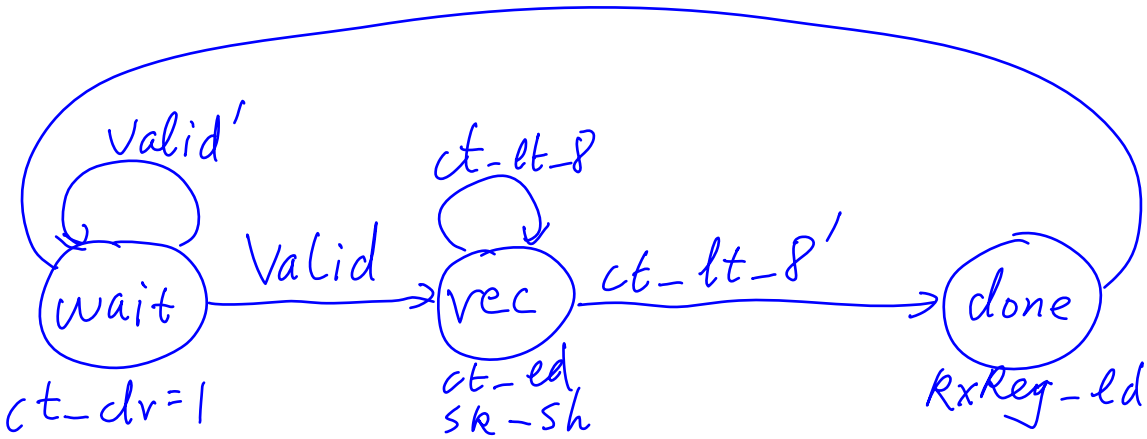


step2:


SR_sh
data_in
+1
ct_ld
ct_clr
ld
clr count
sh
SI SR
8
8
ct_lt_8
count < 8
ld RxReg
RxReg_ld

st 3:

ep


valid'
ct_lt_8
valid
ct_lt_8'
wait
rec
done
ct_clr=1
ct_ed
sk_sh
RxReg_ld