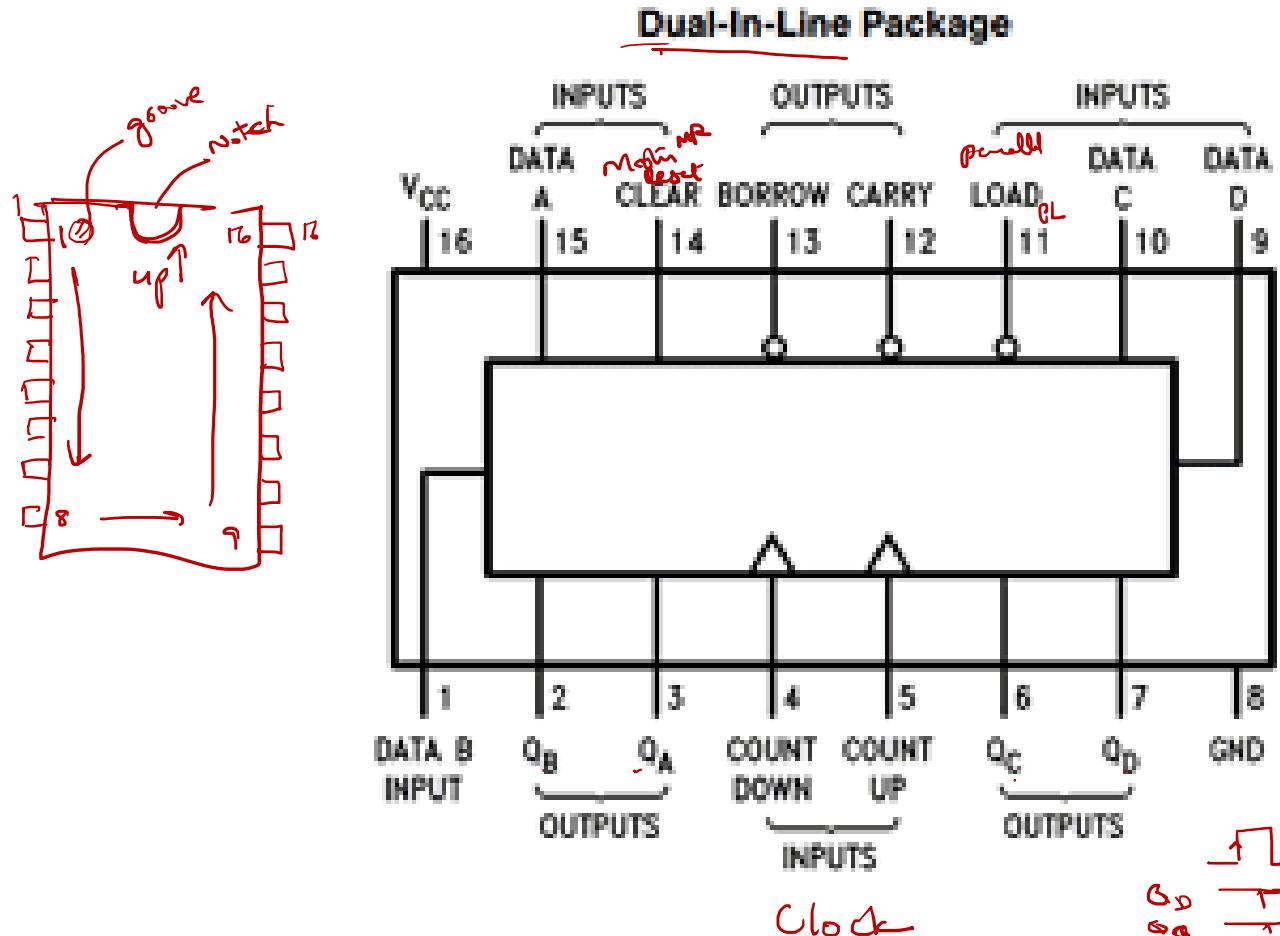


Synchronous counters

- 74193 synchronous counters
- Pre-settable Asynchronous counters and Design
- Pre-settable Synchronous counter and Design

74193 IC: 4-bit up/down synchronous counter

MSB Q_D Q_C Q_B Q_A LSB

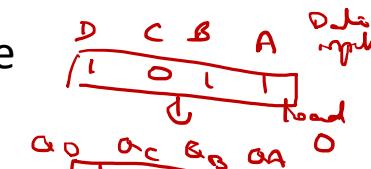


Q_D is the MSB and Q_A is the LSB

Count up/CPu: count up clock input

Count down/CPD: count down clock input

LOAD: Asynchronous parallel load (active low) → use to load your value

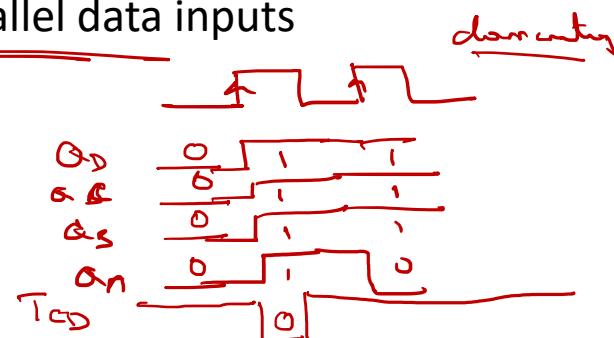
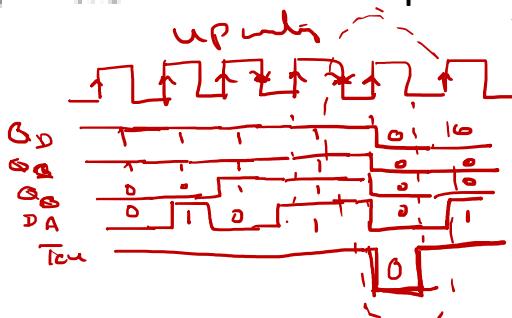


CLEAR: Asynchronous master reset input
HIGH → clears all outputs

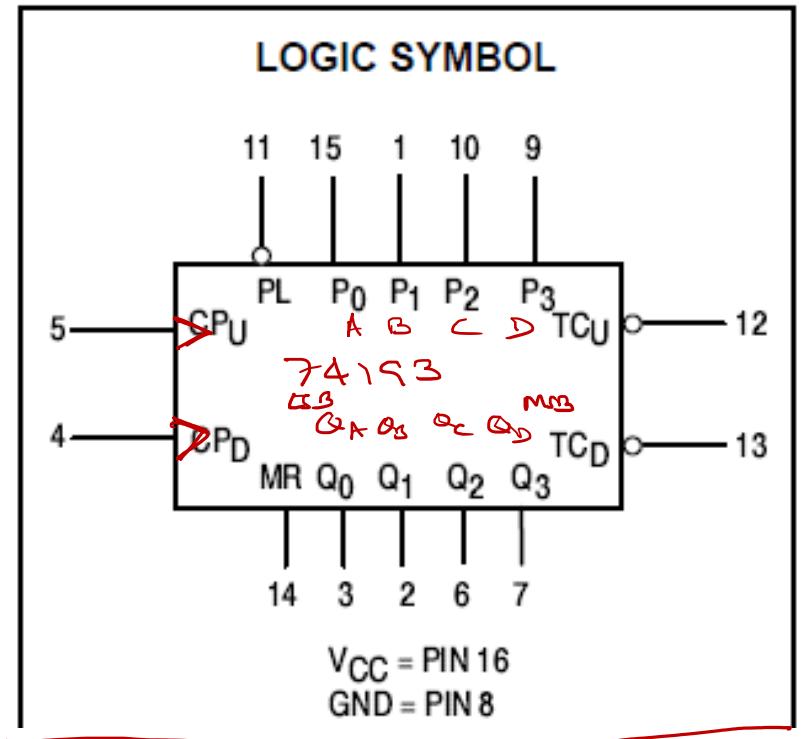
CARRY: Terminal count up output

Borrow: Terminal count down

Data inputs: Parallel data inputs



74193 IC: 4-bit up/down synchronous counter



Normal working
74193

L = LOW Voltage Level
H = HIGH Voltage Level
X = Don't Care
J = LOW-to-HIGH Clock Transition

MODE SELECT TABLE

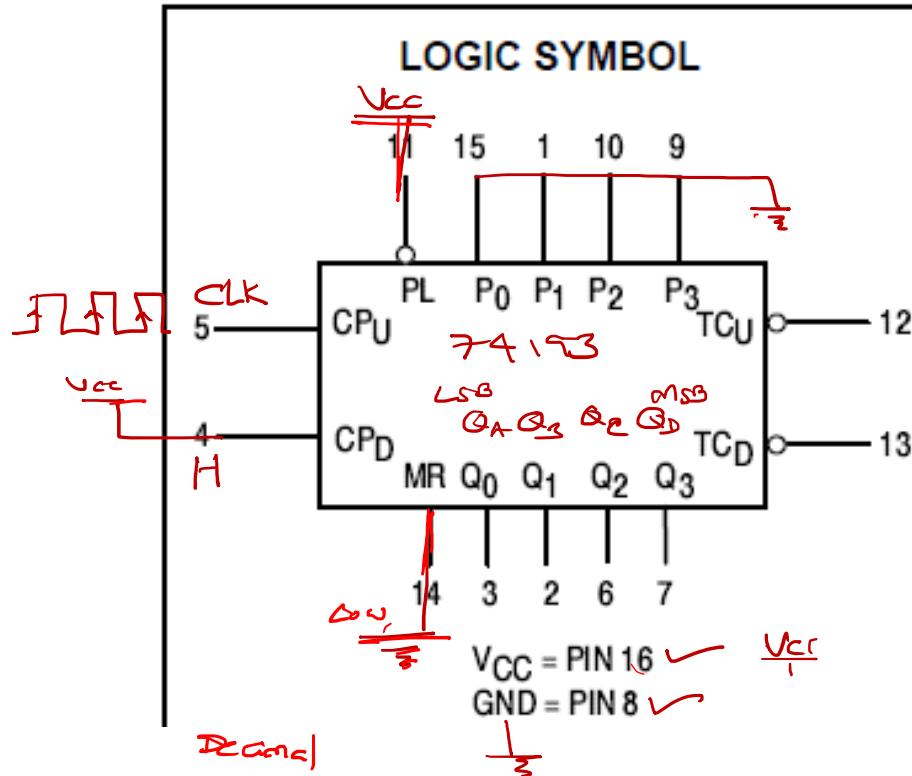
MR	PL	CPU	CPD	MODE
H	X	X	X	Reset (Asyn.)
L	L	X	X	Preset (Asyn.) → preset values on D PBA
L	H	H	H	No Change
L	H	J	H level	Count Up
L	H	J	J	Count Down

→ preset values on D
PBA
Q₀ Q₁ Q₂ Q₃

Ex1:

Design a MOD 16 binary UP counter using 74193 IC

0000
...
1111
~~0000~~



~~2~~

Mod

0000 0000 → 0

to

1111 1111 → 16x16 =
16 × 16

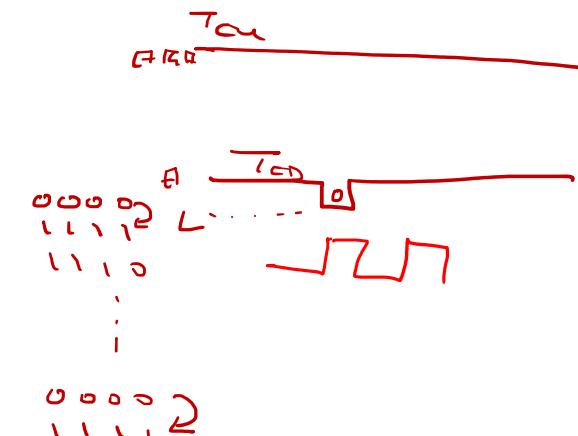
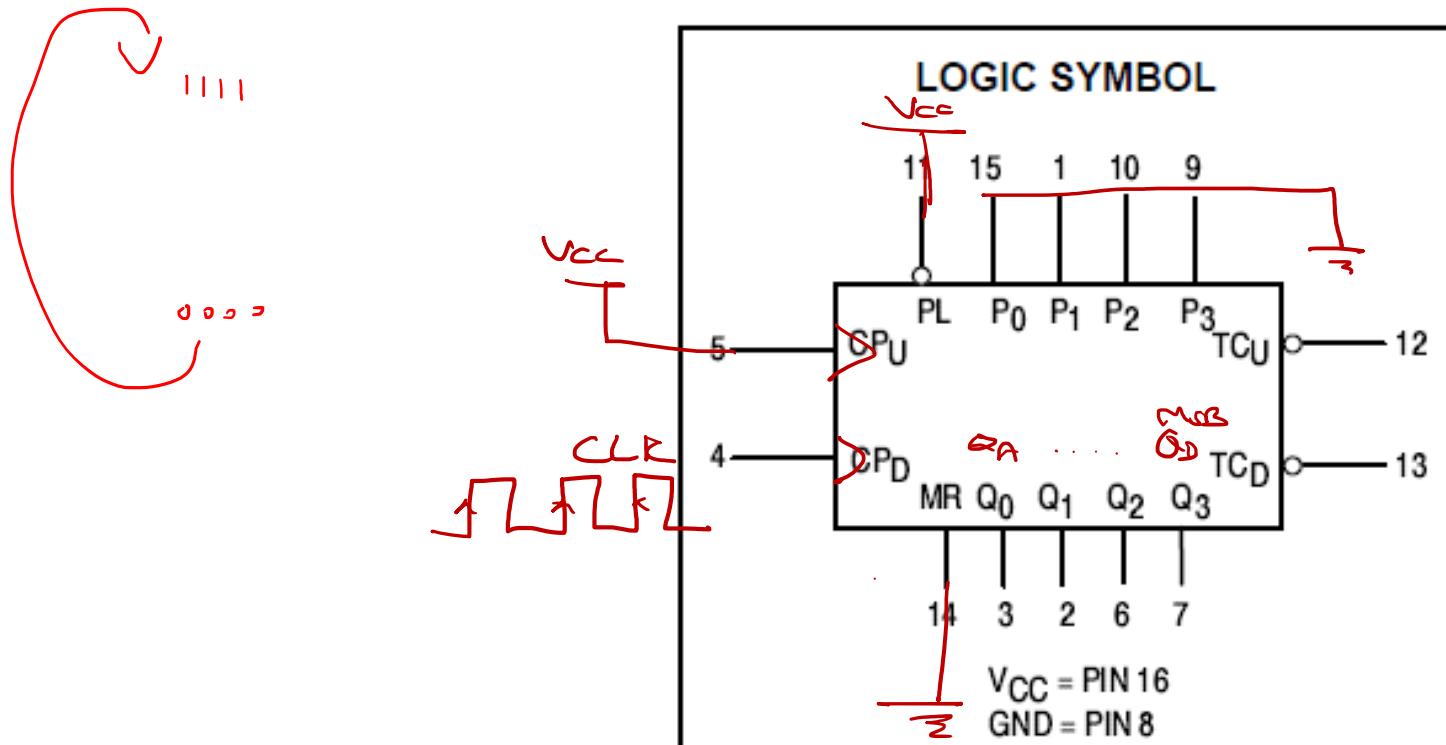
0000

...] ↗ T_{in}

— ↗ T_{out}

Ex 2: Design a MOD 16 binary DOWN counter using 74193 IC

for normal operation

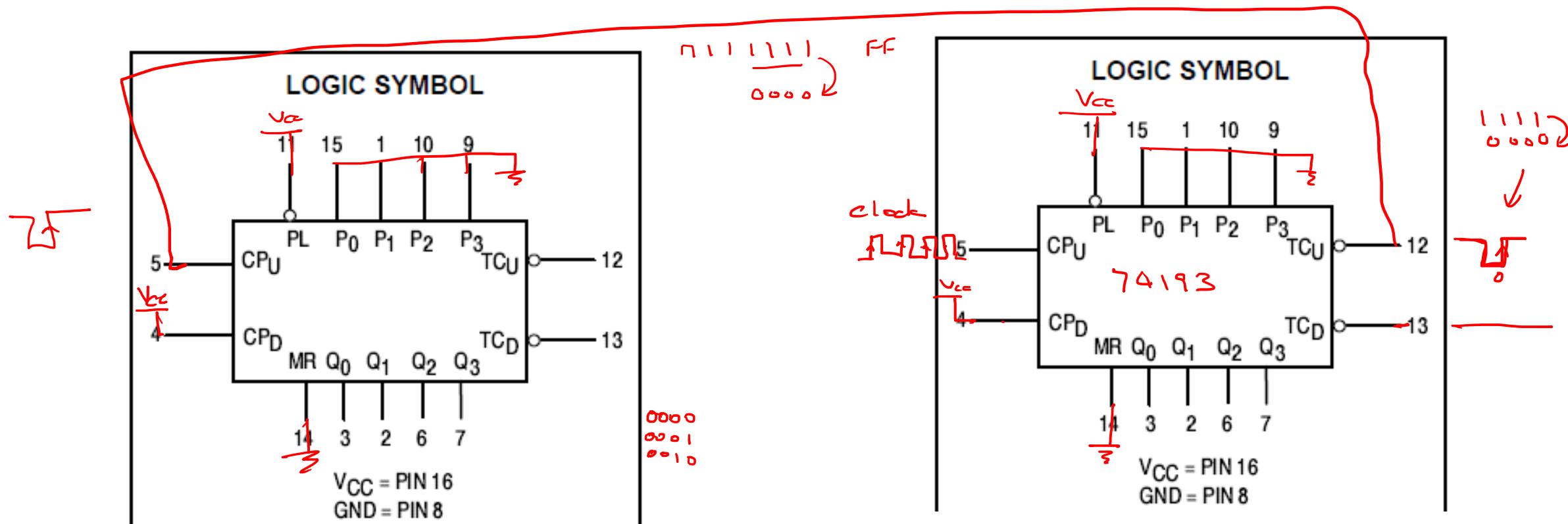


Design a 2-digit hexadecimal UP counter (00-FF H) using 74193 IC

0000 0000 ^{MSD LSD}
00 H

MSD

LSD

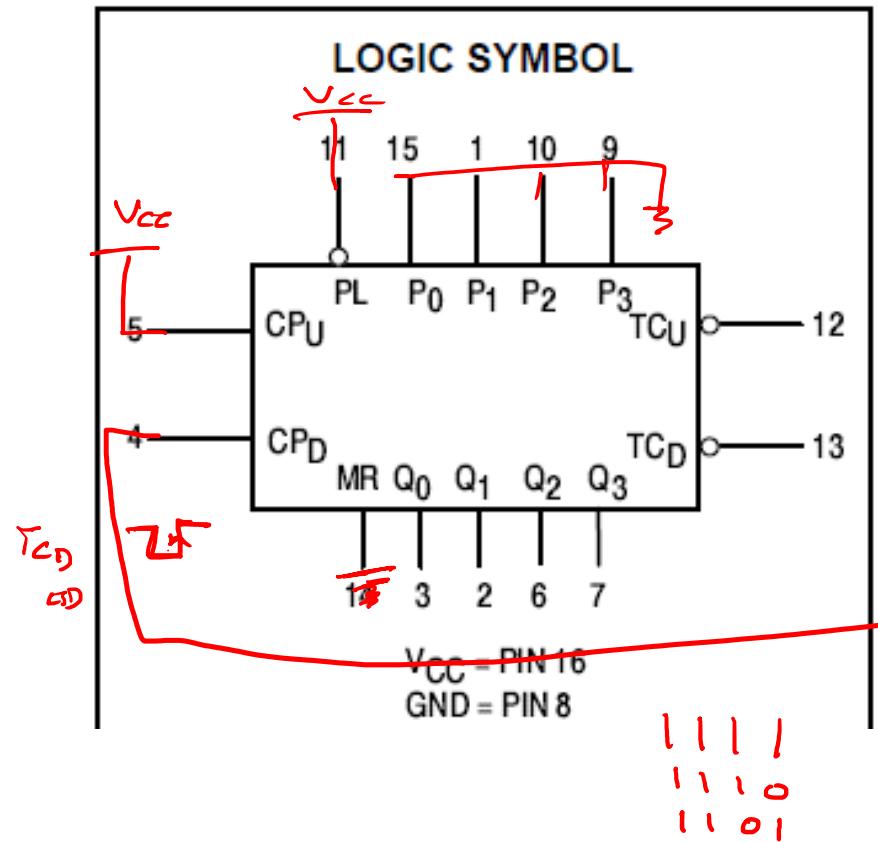


4-bit up counting
mod-16 counter

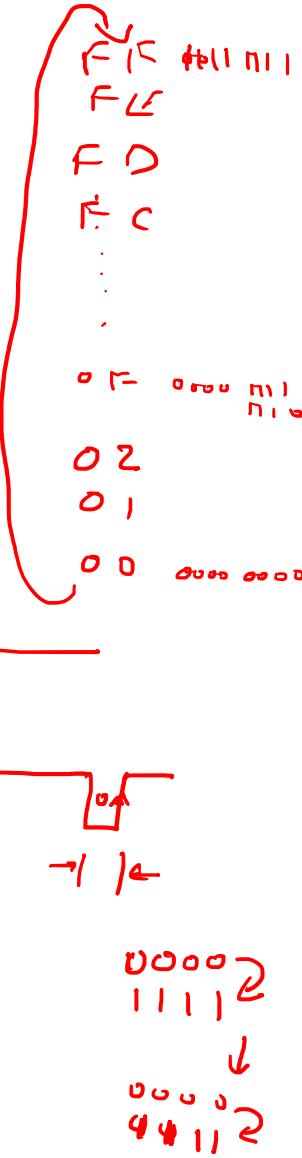
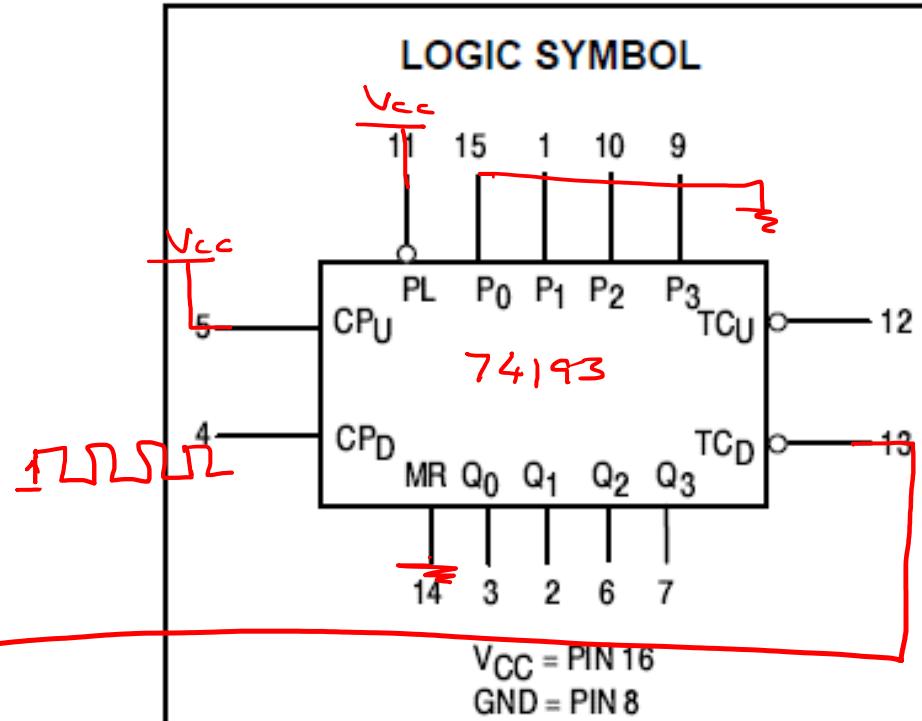
Design a 2-digit hexadecimal ~~UP~~ counter (00-FF H) using 74193 IC

down FF-00H

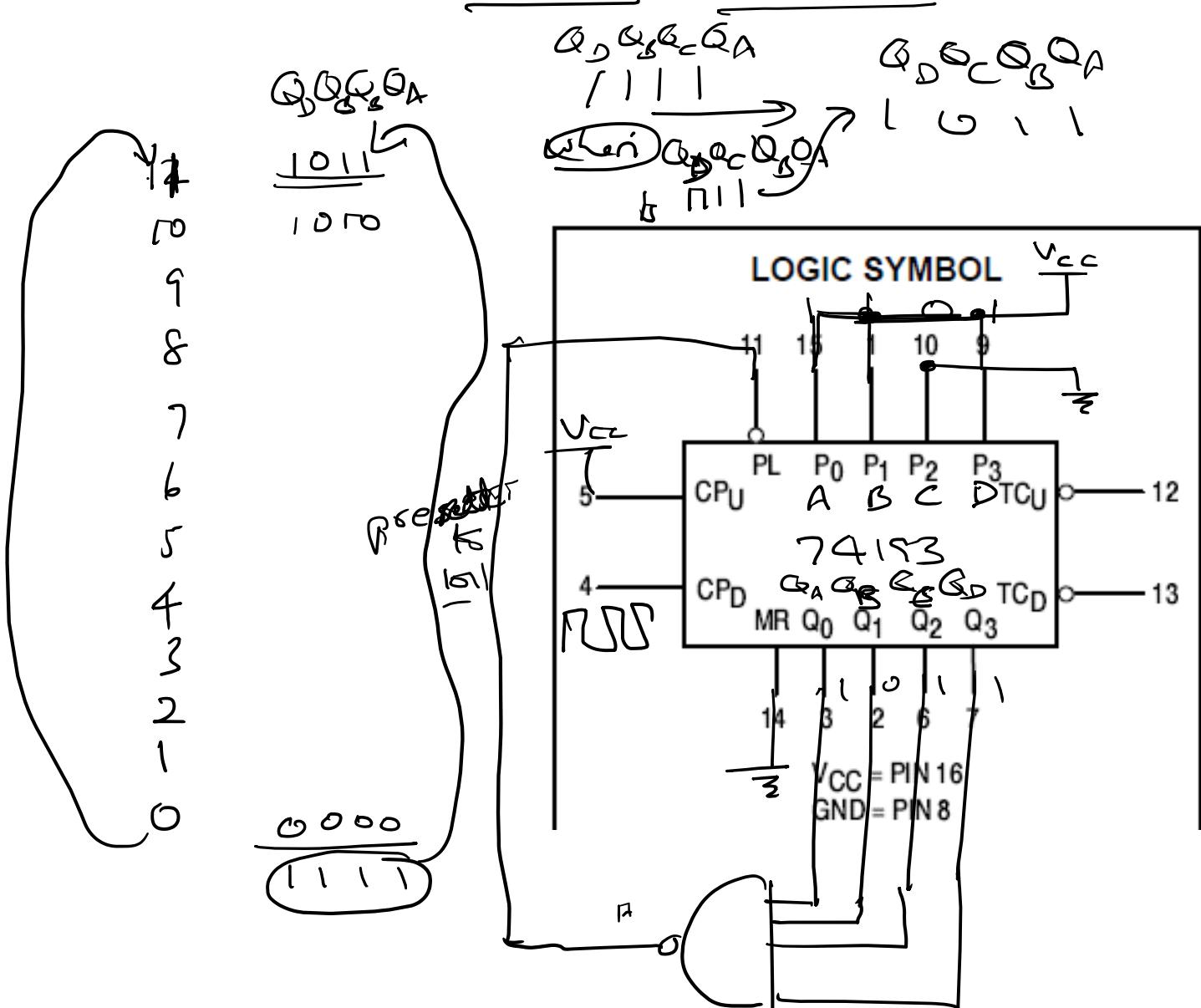
MSD



LSD



Design a MOD 12 binary DOWN counter using 74193 IC



initially PL is high
so that 74193 is in
normal op mode

please note:

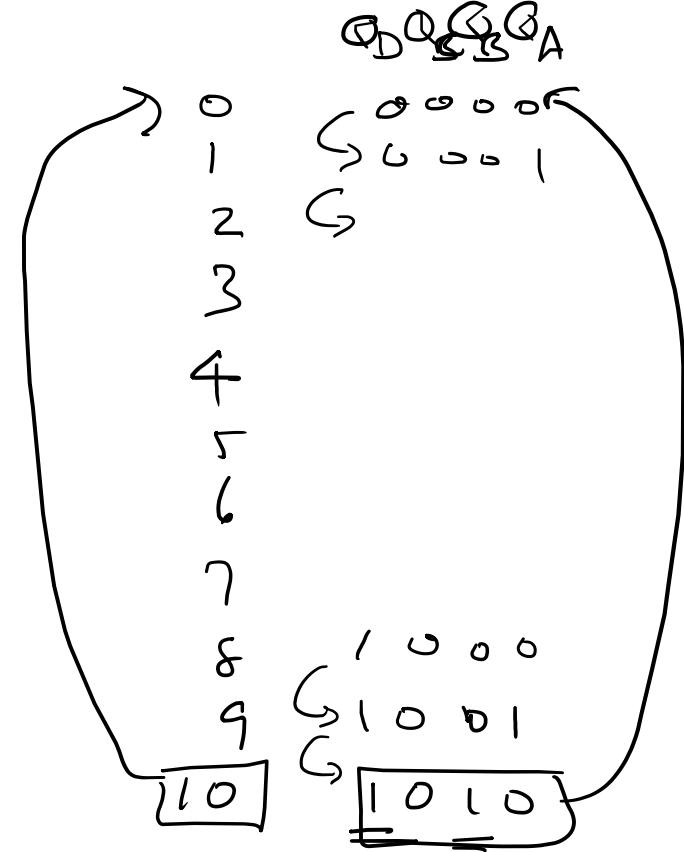
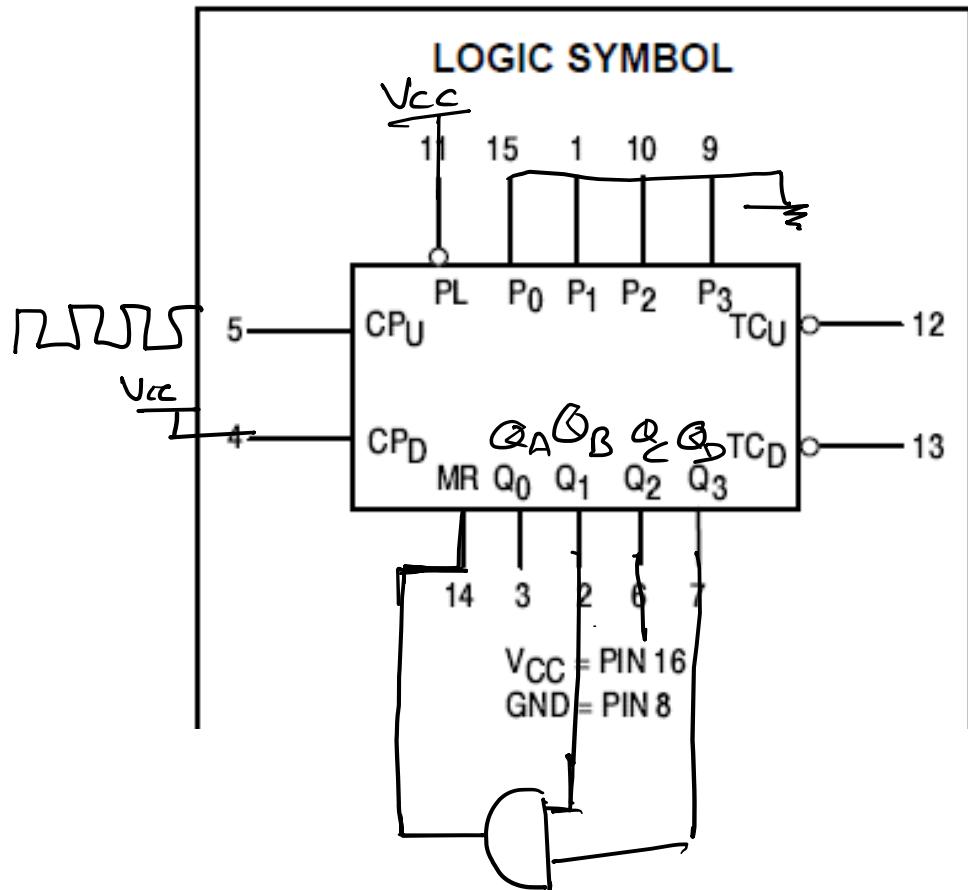
Reset & Status to initial condition
pre-existing

achieved using PL pin

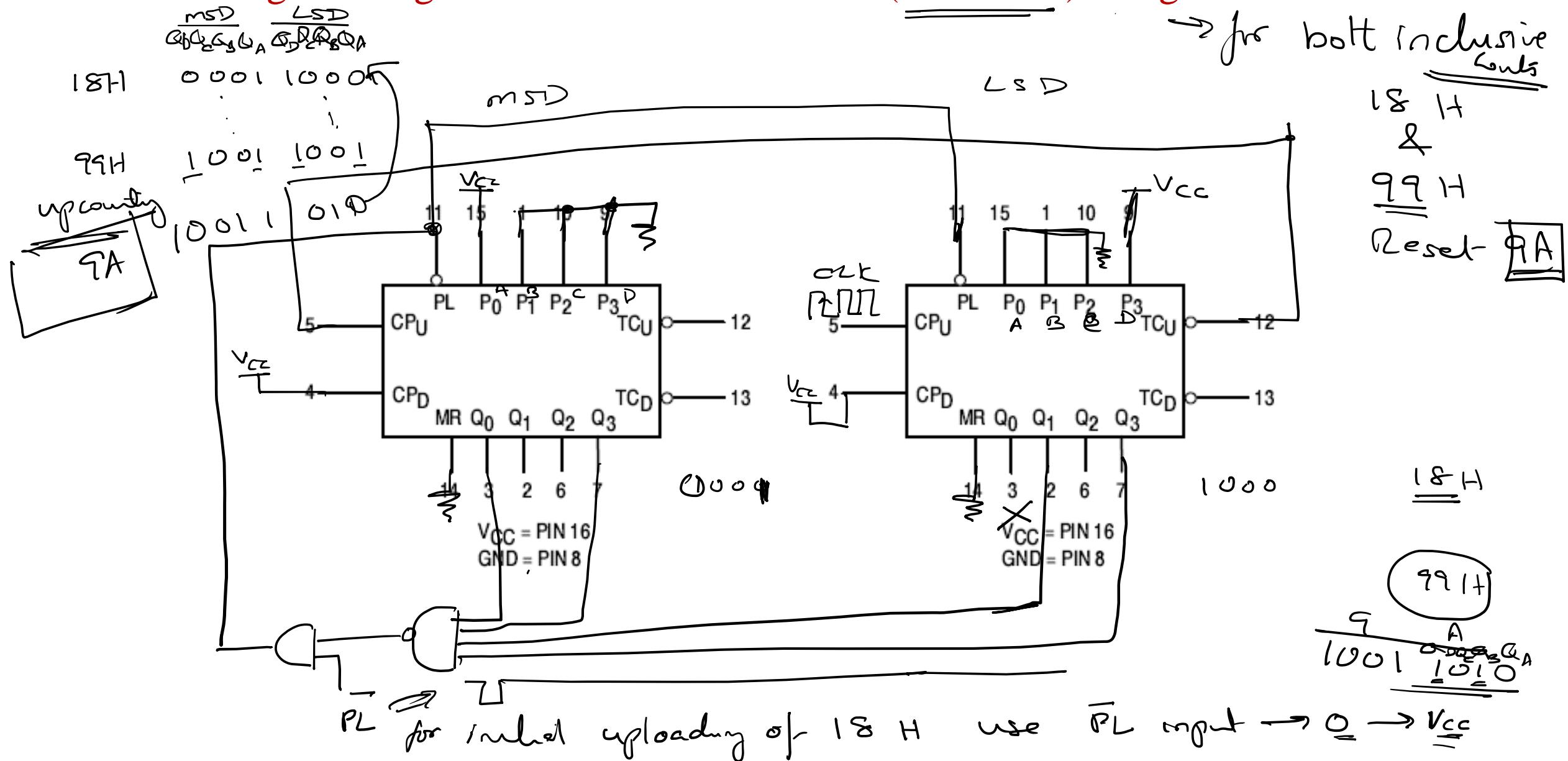
An active low is generated
using NAND logic from the state
where we need to Reset.

Design a Decimal up counter using 74193 IC

mod 10 up counter



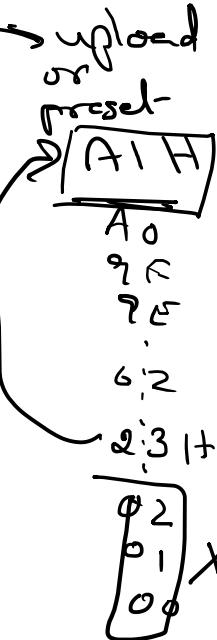
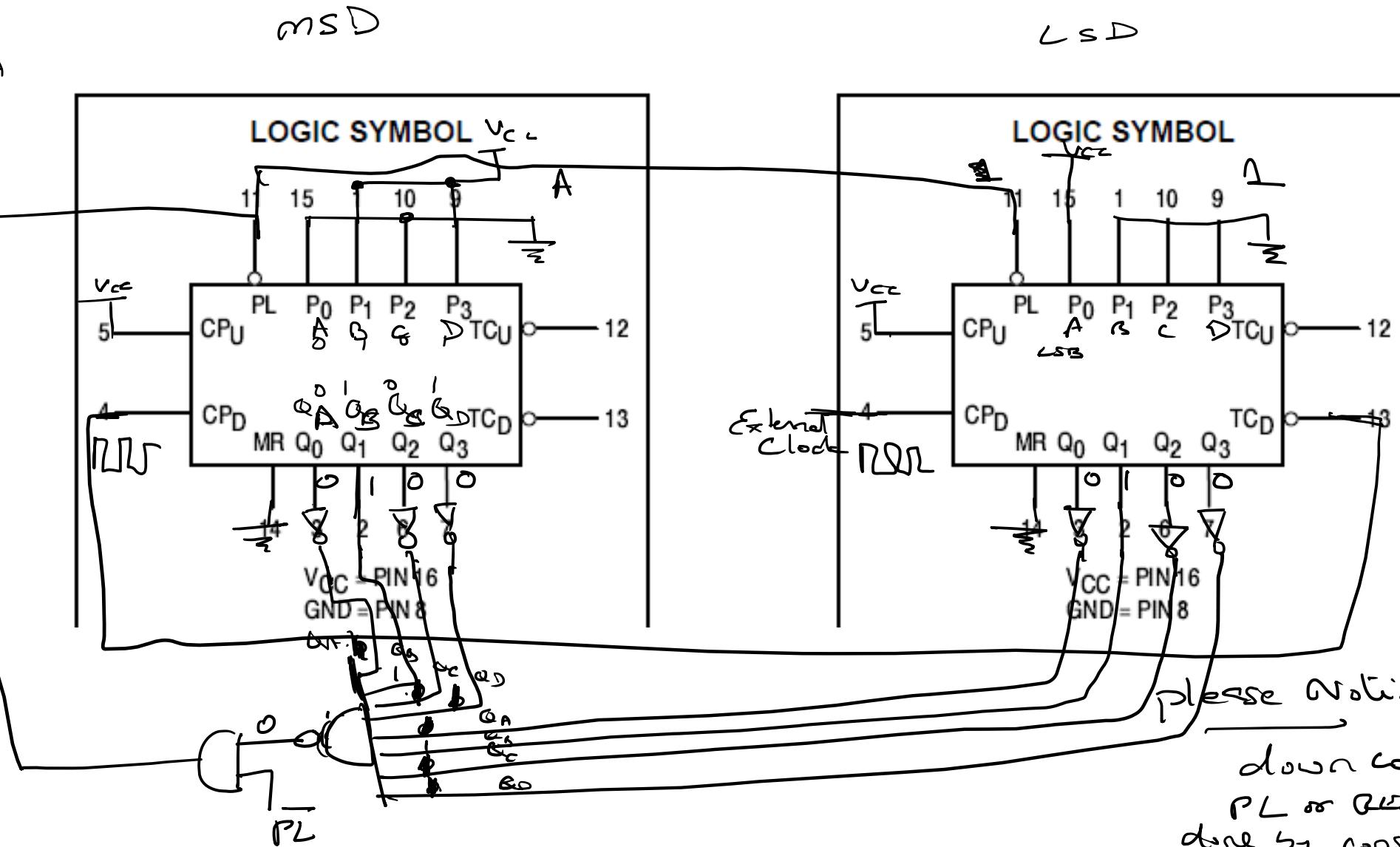
Design a 2-digit hexadecimal UP counter (18H-99 H) using 74193 IC



Design a 2-digit hexadecimal DOWN counter (A1H-23 H) using 74193 IC

from & till 23H 22H → upload or preset

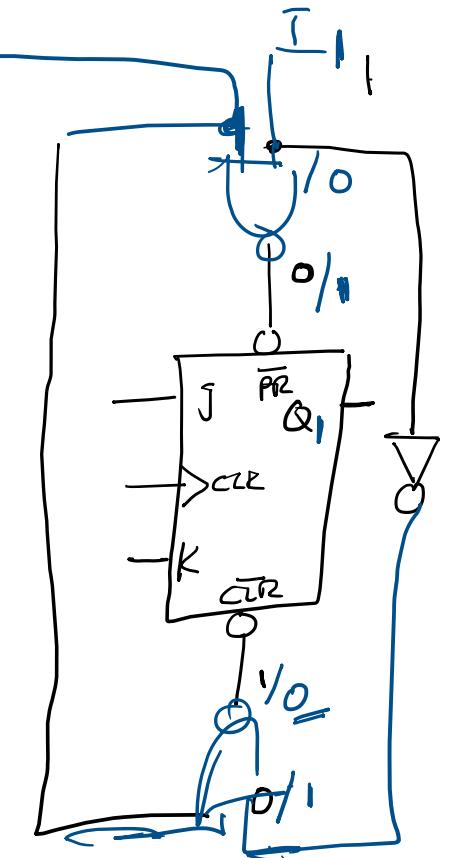
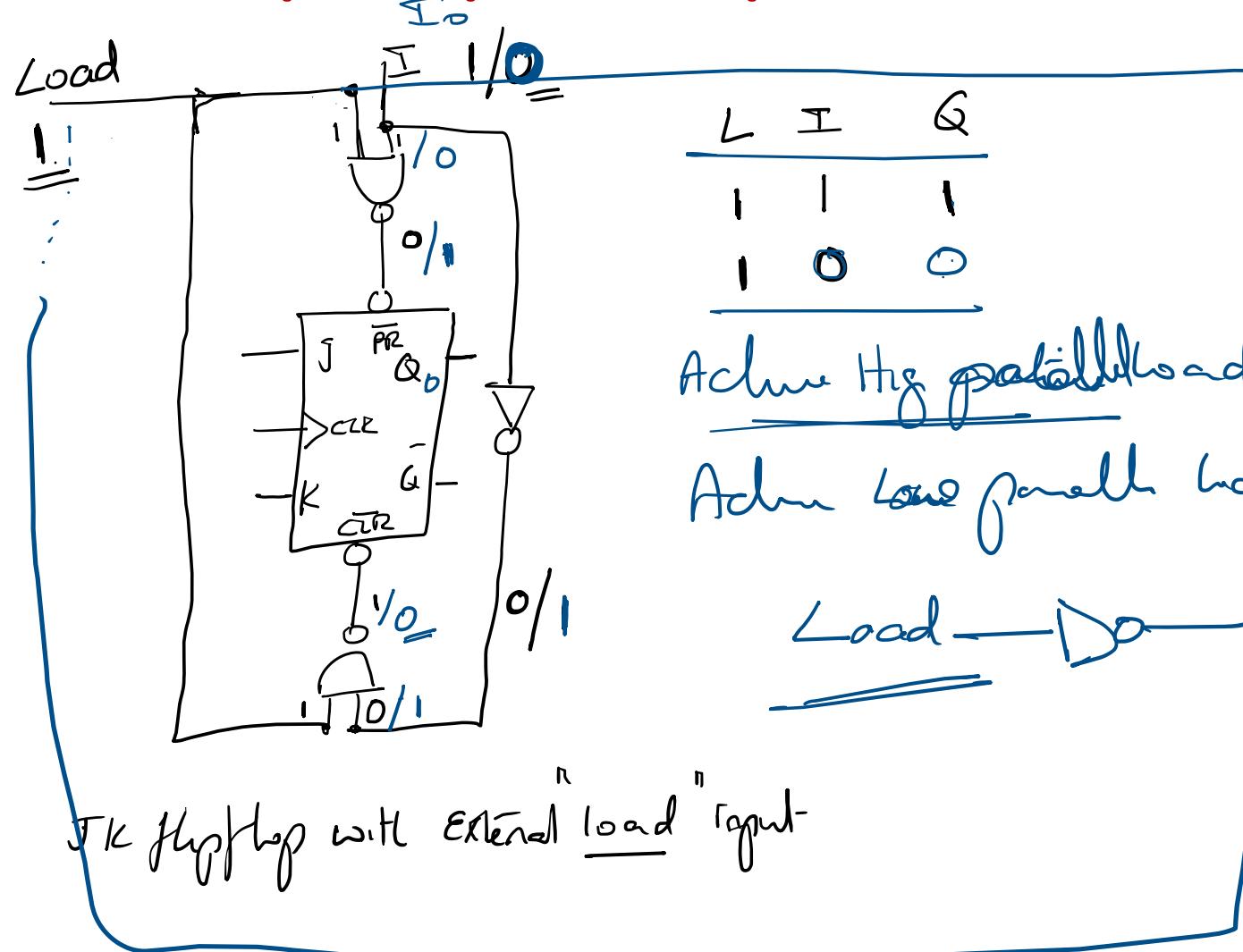
1010 0001 H
 down counting
 initial count A1 H
 A' FF
 final count 23 H
 22 H
 0010 0010
 0110 0010
 6 2
 resetting or
 presetting
 should be
 done JNC1 for
 msd lsd
 0010 0010



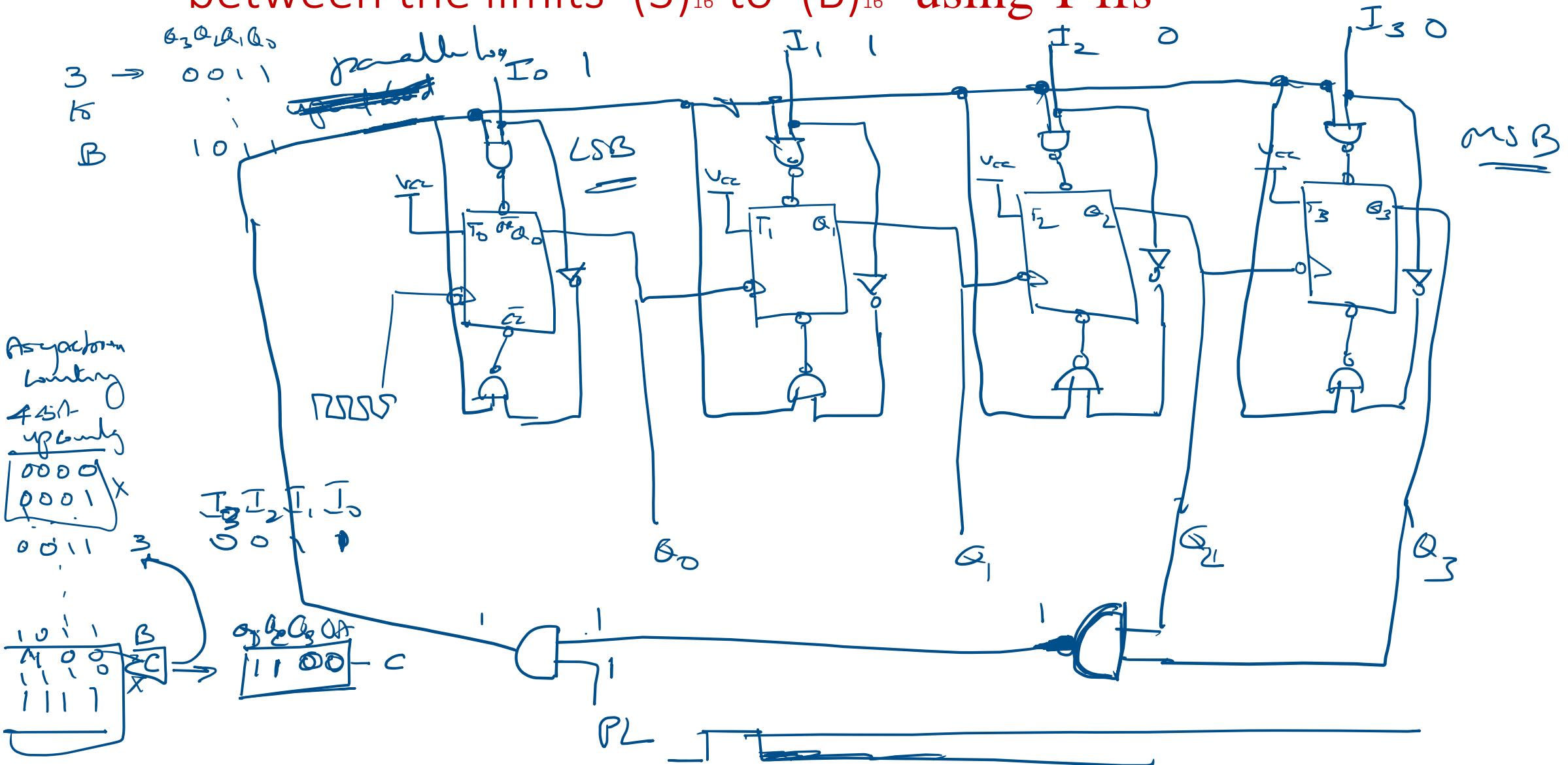
Counters with parallel load(Presettable)

- Flip flops will have an additional asynchronous input referred as ‘load’, which may be active high or low.
- For active low ‘load’ input, if load =0, flip flop should be loaded with external input bit P else output should change according to other synchronous/asynchronous inputs.
- This feature enables the counter to have parallel load capability for transferring an external input /data /count into the counter.

Flip flop with parallel load: design



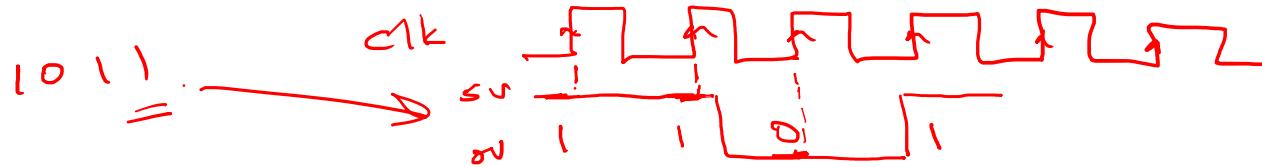
Design an presettable asynchronous counter to count between the limits $(3)_{16}$ to $(B)_{16}$ using T-ffs



A blurred background image of a digital speedometer. The top part shows the number '5' in white. Below it, a white bar has the text '1000 rpm' and 'rpm' partially visible. To the right, there are red digital segments forming the numbers '7' and '1'.

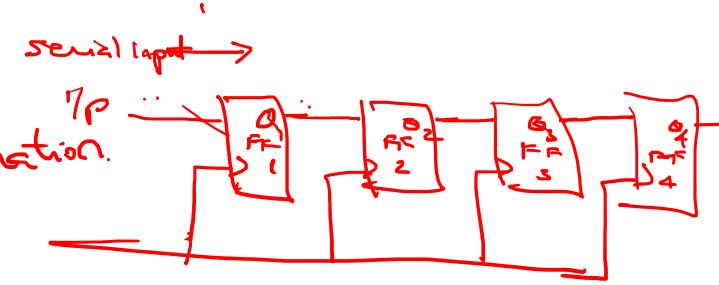
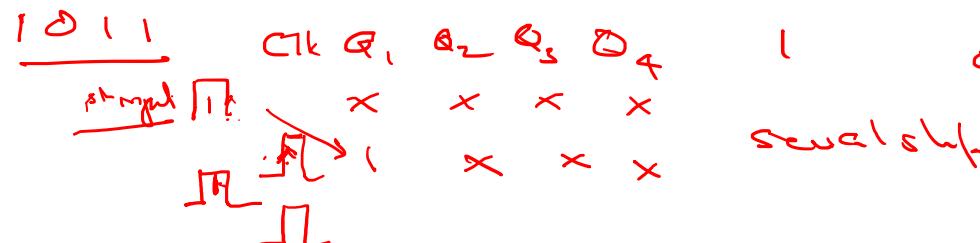
Shift registers and shift register
counters

Registers



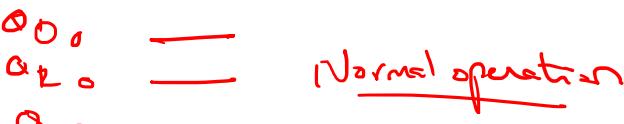
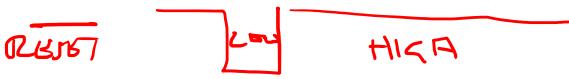
- Register is a group of flip flops (D, or SR, or JK)
- Each flip flop can store one-bit data. n-bit register can hold n-bit data.
- There are two ways to shift the data into the register and two ways to shift the data out of the register.
- Accordingly: 4 categories of shift register are
 - Serial in Serial out (SISO)
 - Serial In parallel out (SIPO)
 - Parallel in Serial out (PISO)
 - Parallel in parallel out (PIPO)

Example: 4-bit shifting operation.



Simple 4-bit shift Register

SISO and SIPO Register



- Simple 4-bit shift register is shown below (Reset is nothing but clear input)
- Register is cleared using reset/clear input.
- It can be used as Serial-in serial out (at Q_D) and Serial-in parallel out shift register.

LITTLE endian concept.
LSB is first

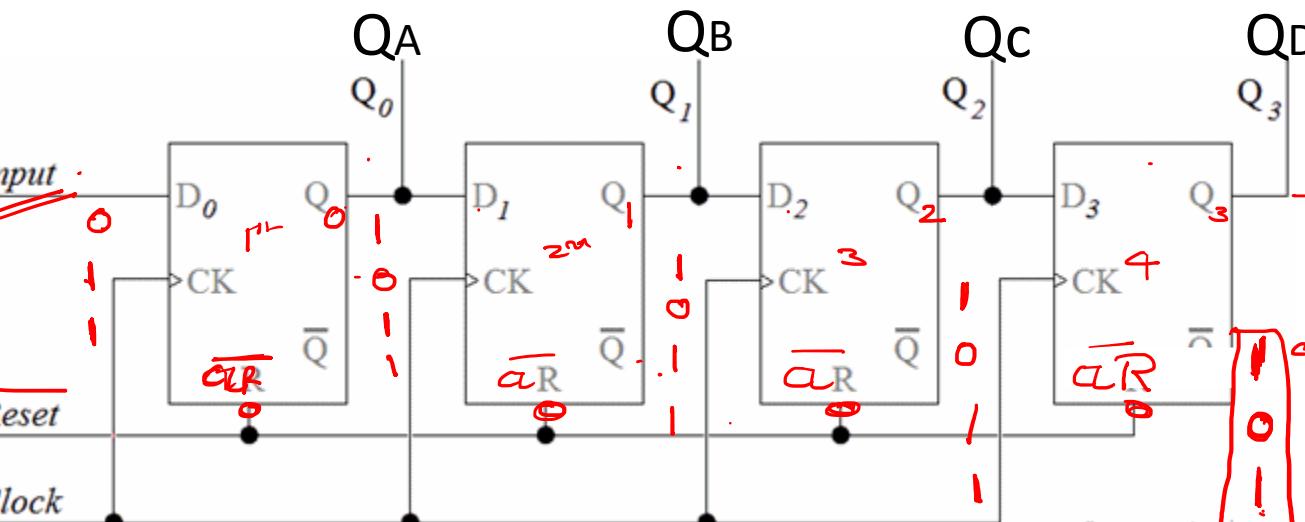
n-bit MSB ... LSB

msb
101

Ex. Lst

In a n-bit shift register
the requires n-clock pulses
to store the n-bit data
serially

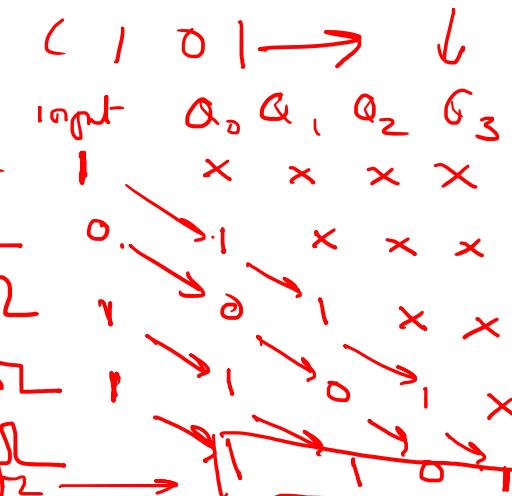
at 1st clock pulse
n-bit (LSB) is available
parallel edge
of flip-flops
serially



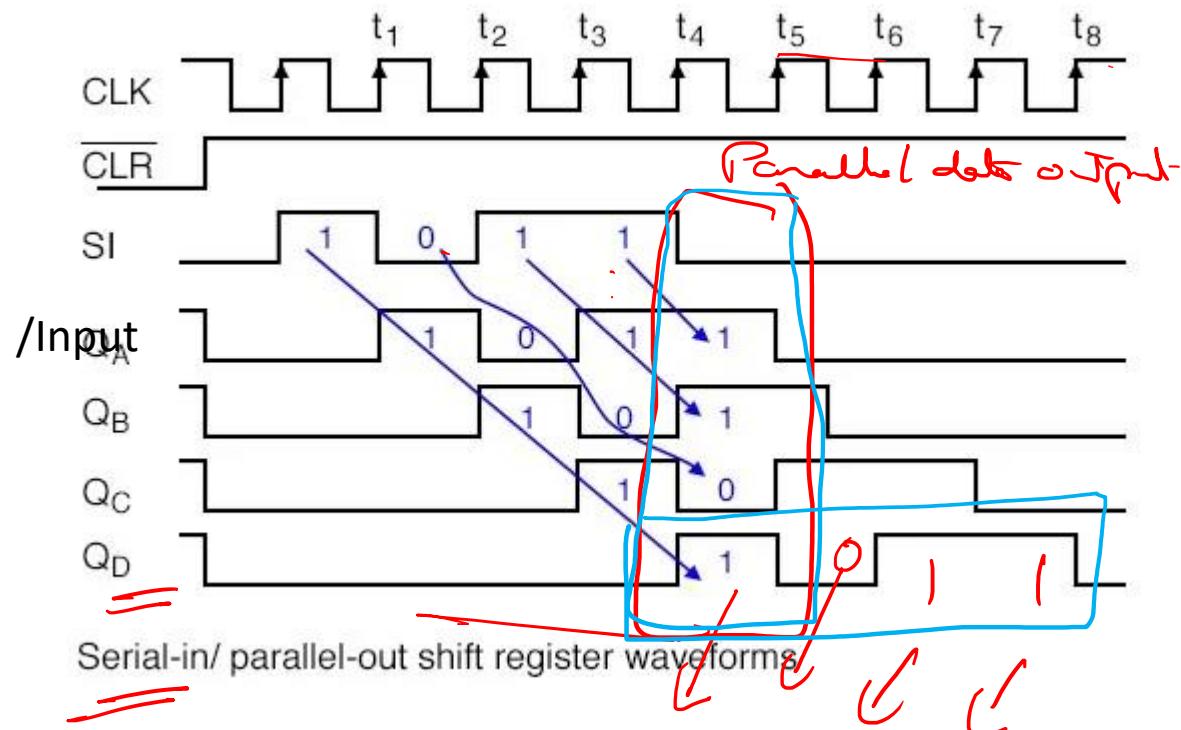
These requires $(2^n - 1)$

clock pulses
for completely shifting out
data serially at
optimum time

© www.petervis.com



4-bit shift register (Serial-in)



With n-bit shift right
→ parallel
2n-clock
→ n-bit serial
Serial output