CMOS Combinational Circuits

Subsystem Design and Layout

- In all designs, a logical and systematic approach is necessary

  Eg: 500 transistor design ⟶ 2 Engr months

  500,000 transistor design ⟶ 170 Engr years

- Subsystems are the basic leaf-cells from which larger systems are composed
- The most basic leaf-cells are the common logic gates

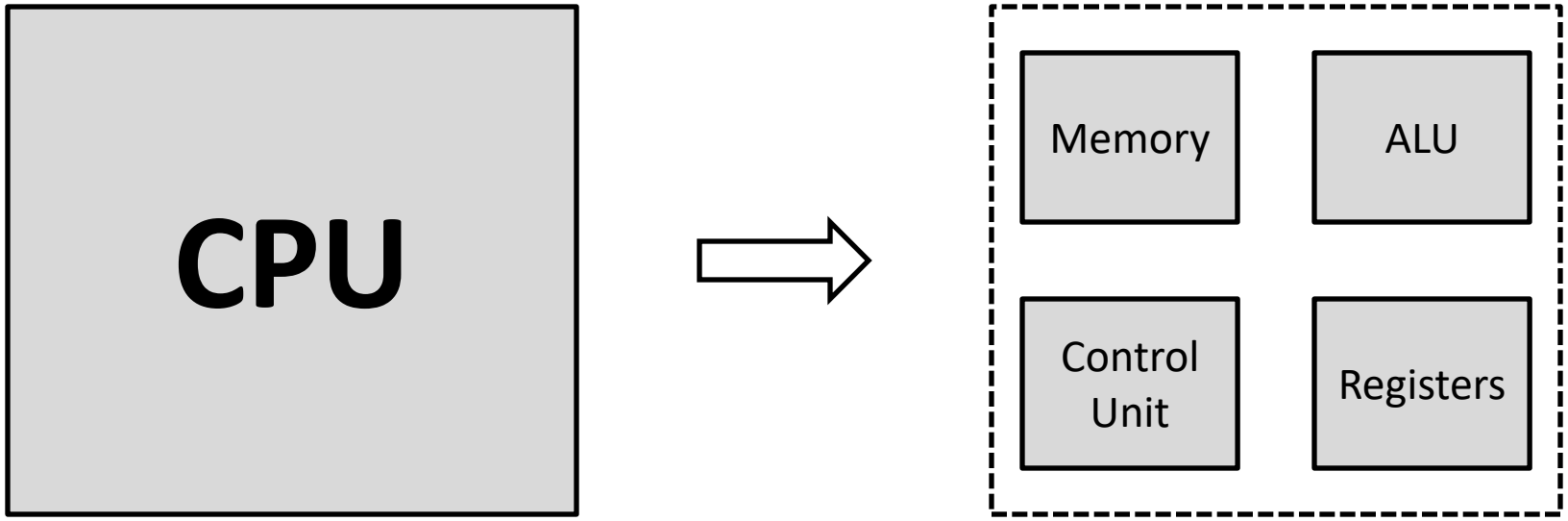
- The concepts of structured design leads to system designs of high ‘regularity’
- This leads to the detailed design of relatively a few leaf cells, which are then replicated many times & interconnected to form the system
- This way, we can cope up with the complexity

Structured CMOS Design Steps / Guidelines

1. Define the requirements (properly and carefully)
2. Partition the overall architecture into appropriate subsystems
3. Consider communication paths carefully in order to develop sensible interrelationships between subsystems
4. Draw a floor plan of how the system is to map onto the silicon (alternate between steps 2, 3 & 4 as necessary)
5. Aim for regular structures so that design is largely a matter of replication
6. Draw suitable (stick or symbolic) diagrams of the leaf-cells of the subsystems

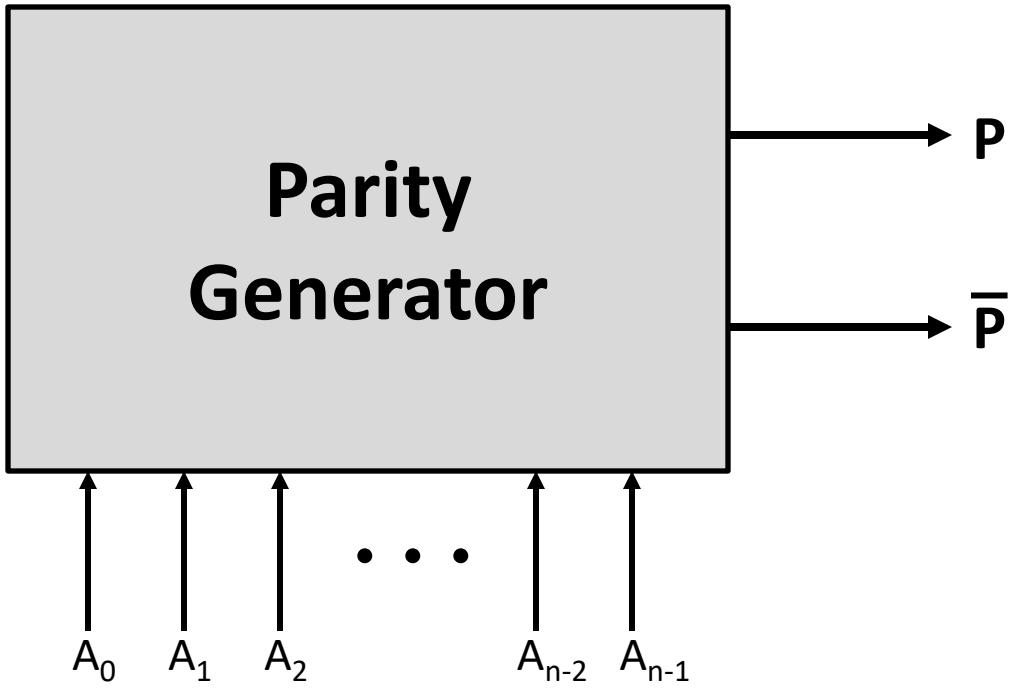
Structured CMOS Design Steps / Guidelines

7. Convert each cell into a layout
8. Carefully and thoroughly carryout a design rule check on each cell
9. Simulate the performance of each cell / subsystem

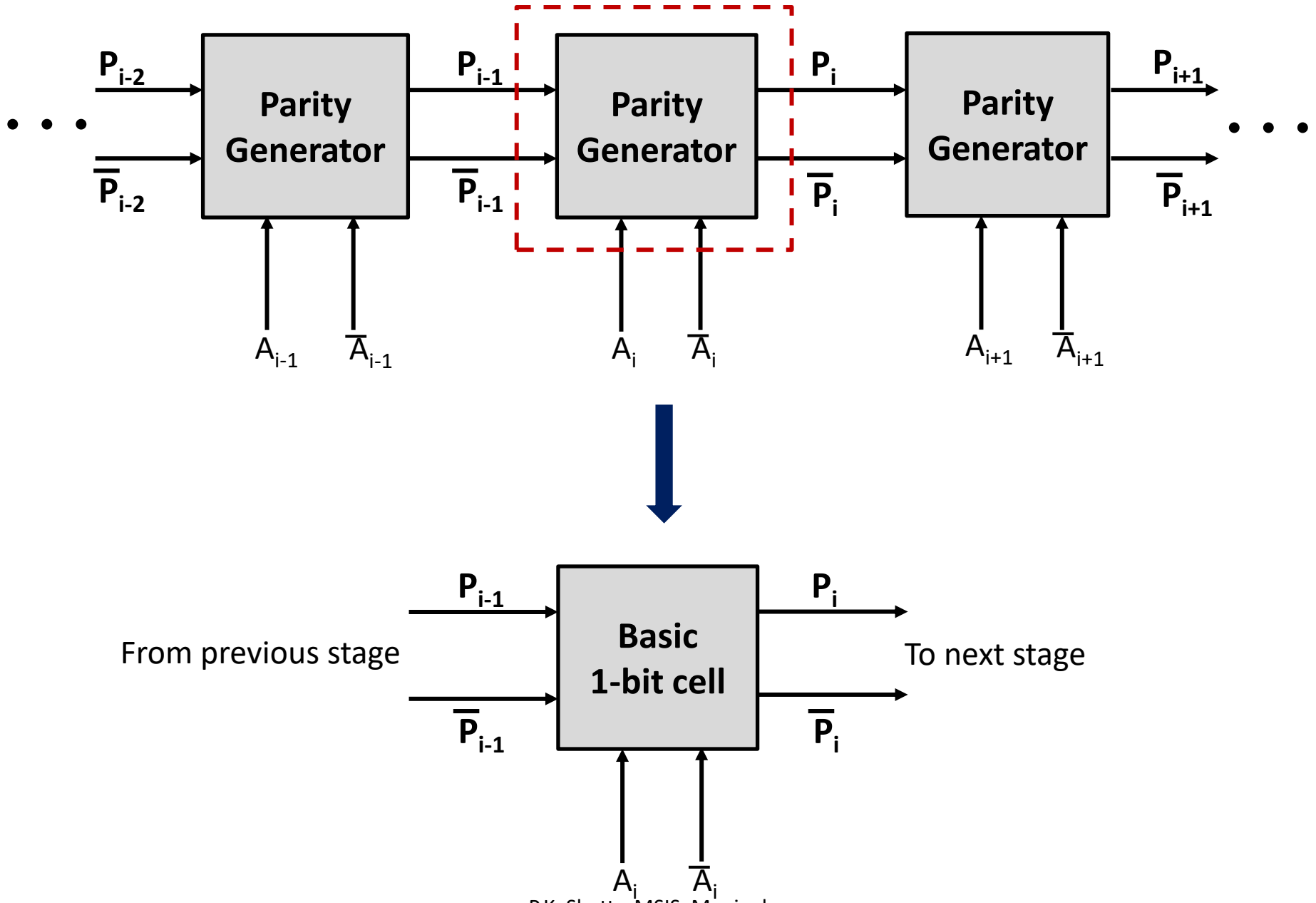


Structured Design - Example

- Design a circuit to indicate the parity of a binary number or word of n-bit wide



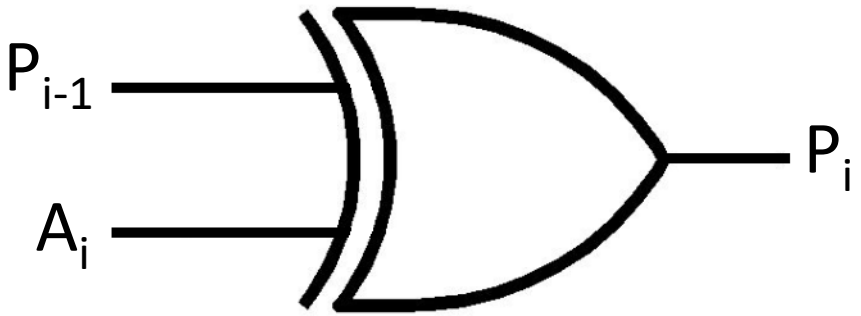
P = 1, Even Parity
P = 0, Odd Parity


P_{i-2}
$\overline{P}_{i-2}$
Parity Generator
P_{i-1}
$\overline{P}_{i-1}$
Parity Generator
P_i
$\overline{P}_i$
Parity Generator
P_{i+1}
$\overline{P}_{i+1}$
A_{i-1}
$\overline{A}_{i-1}$
A_i
$\overline{A}_i$
A_{i+1}
$\overline{A}_{i+1}$
From previous stage
P_{i-1}
$\overline{P}_{i-1}$
Basic 1-bit cell
P_i
$\overline{P}_i$
To next stage
A_i
$\overline{A}_i$

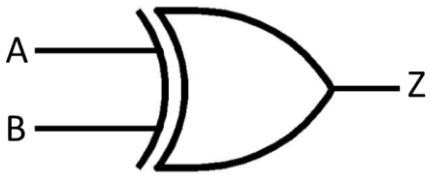
If $A_i = 1$: Parity is changed, $P_i = \overline{P}_{i-1}$

If $A_i = 0$: Parity is unchanged, $P_i = P_{i-1}$

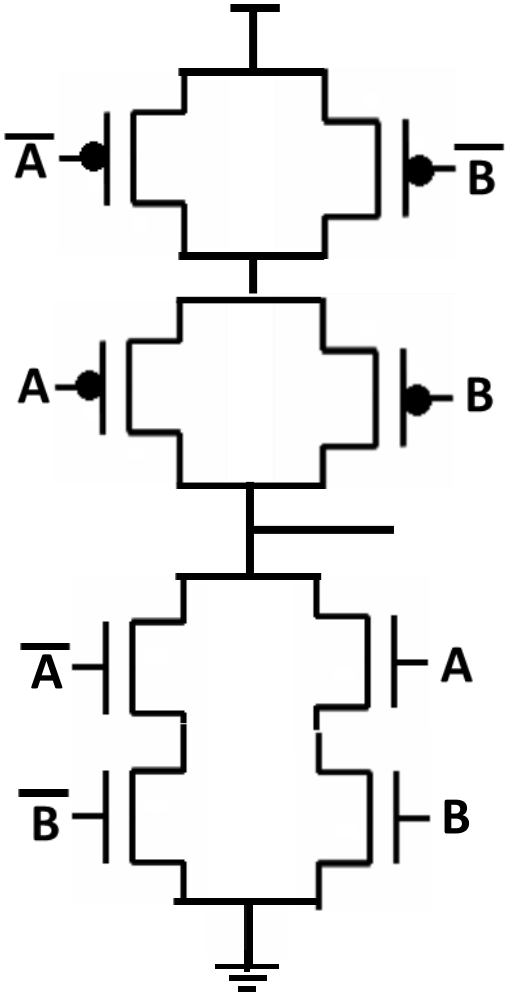
$$\Rightarrow \quad P_i = \overline{P}_{i-1} A_i + P_{i-1} \overline{A}_i$$



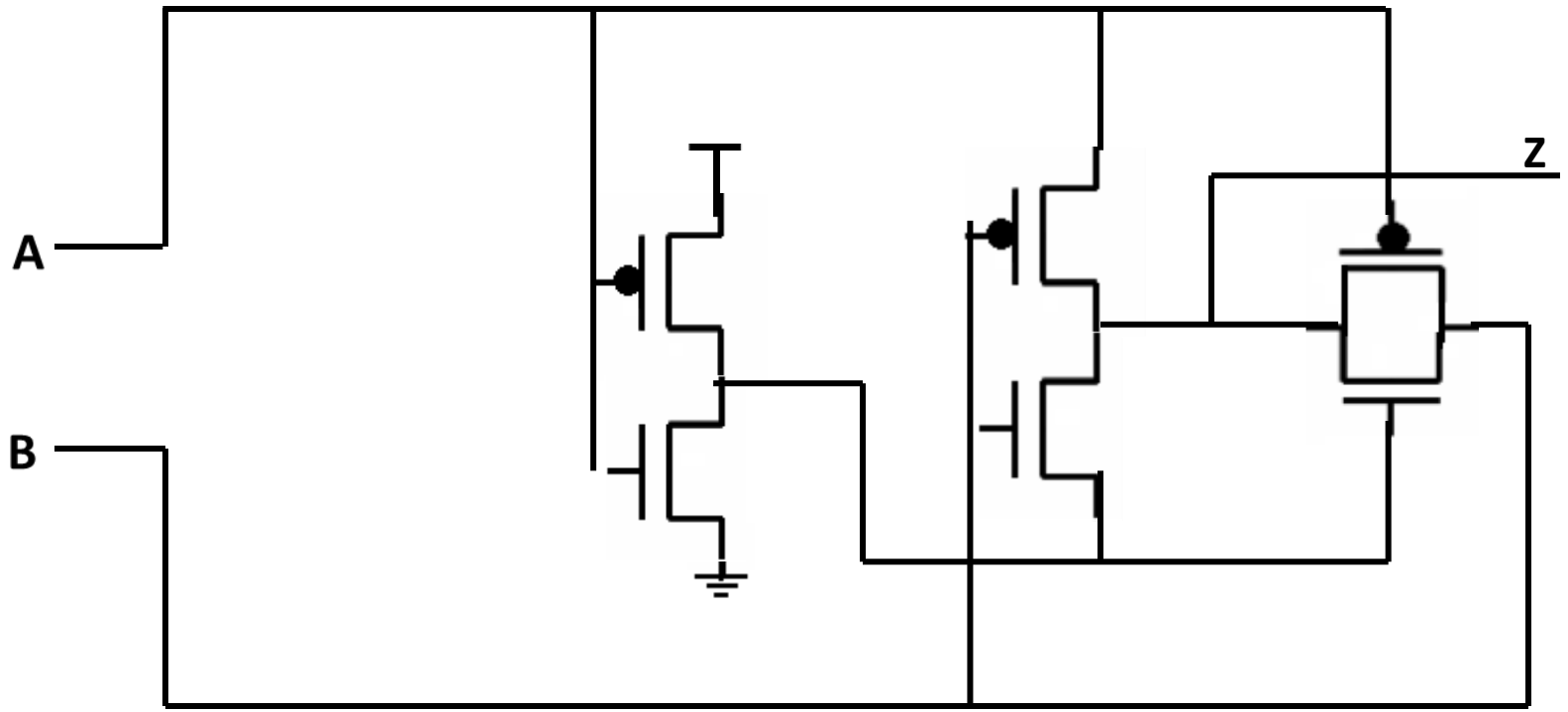
XOR gate



A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0



XOR gate - Another Implementation

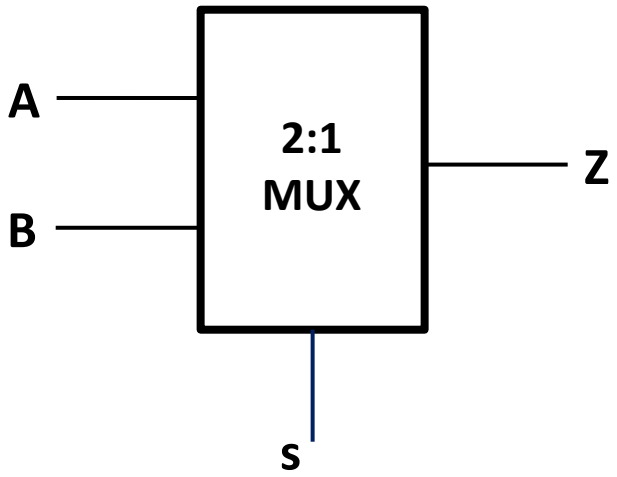


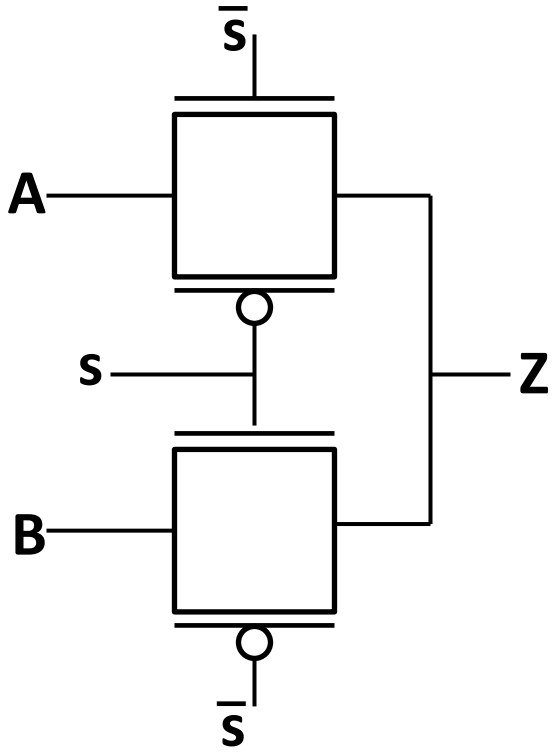
Multiplexers

2:1 Multiplexers

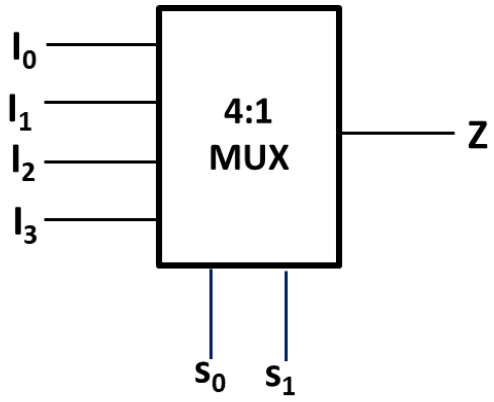
s	Z
0	A
1	B

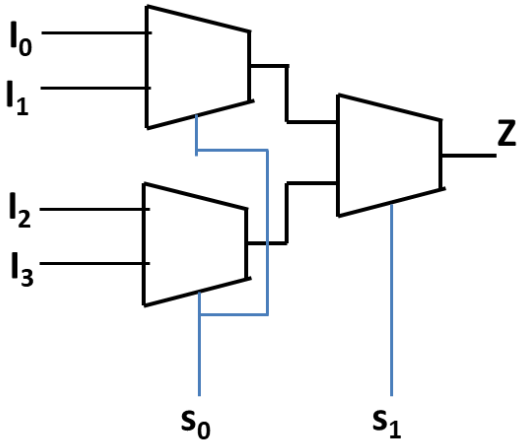
$Z = A\bar{s} + Bs$

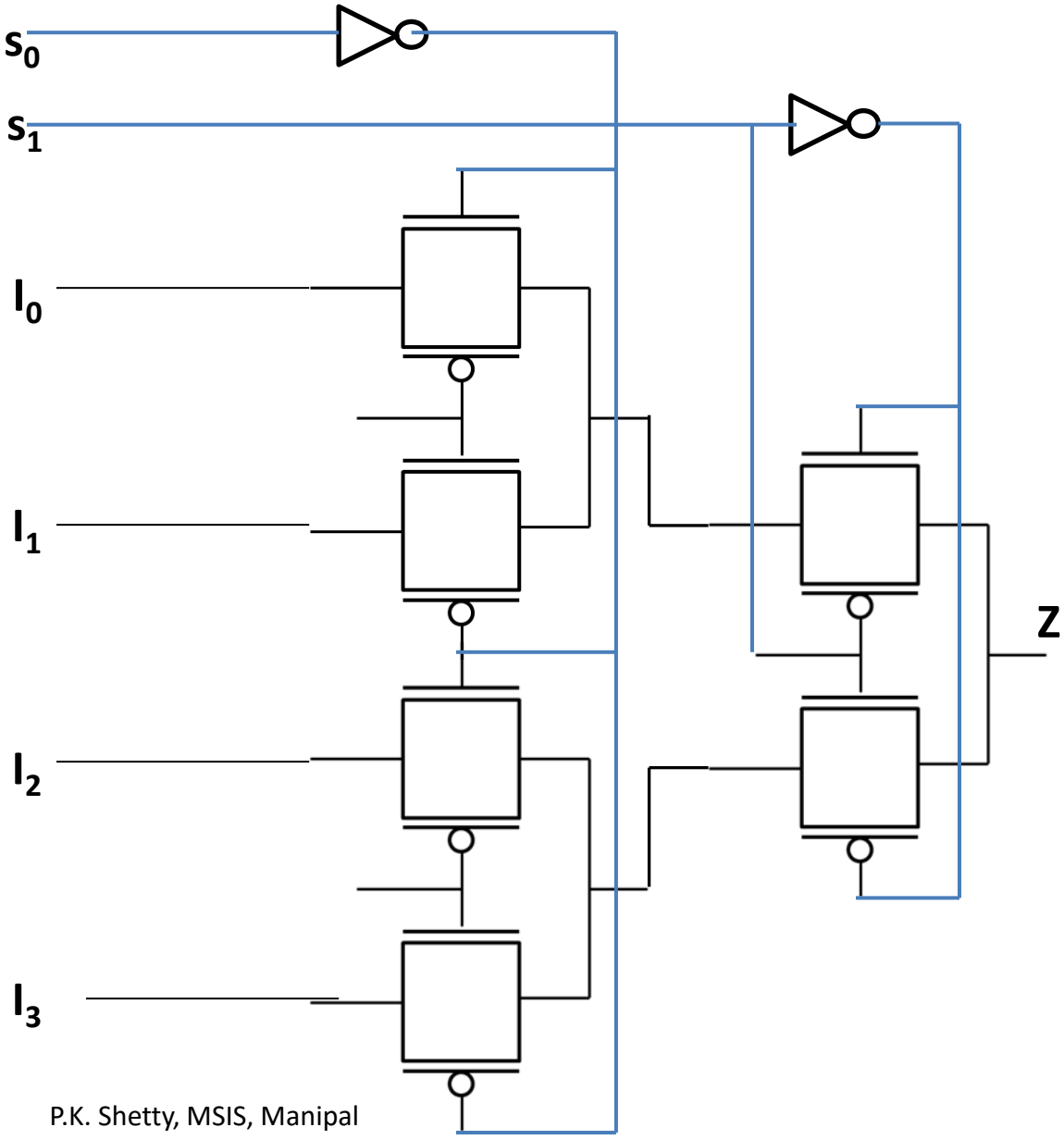




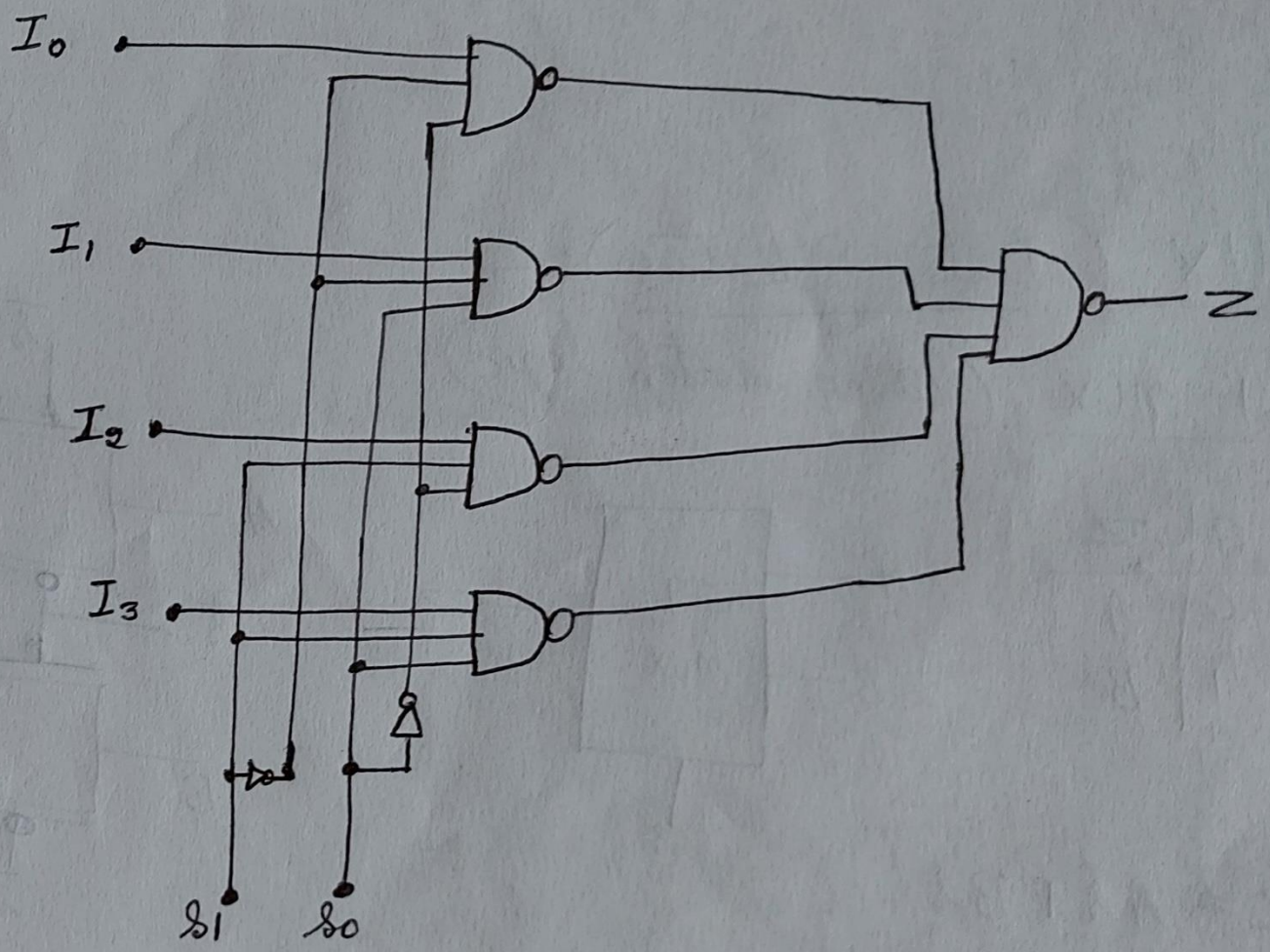
4:1 Multiplexers







4:1 MUX using gate logic:



$$Z = \bar{S}_1 \bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3$$

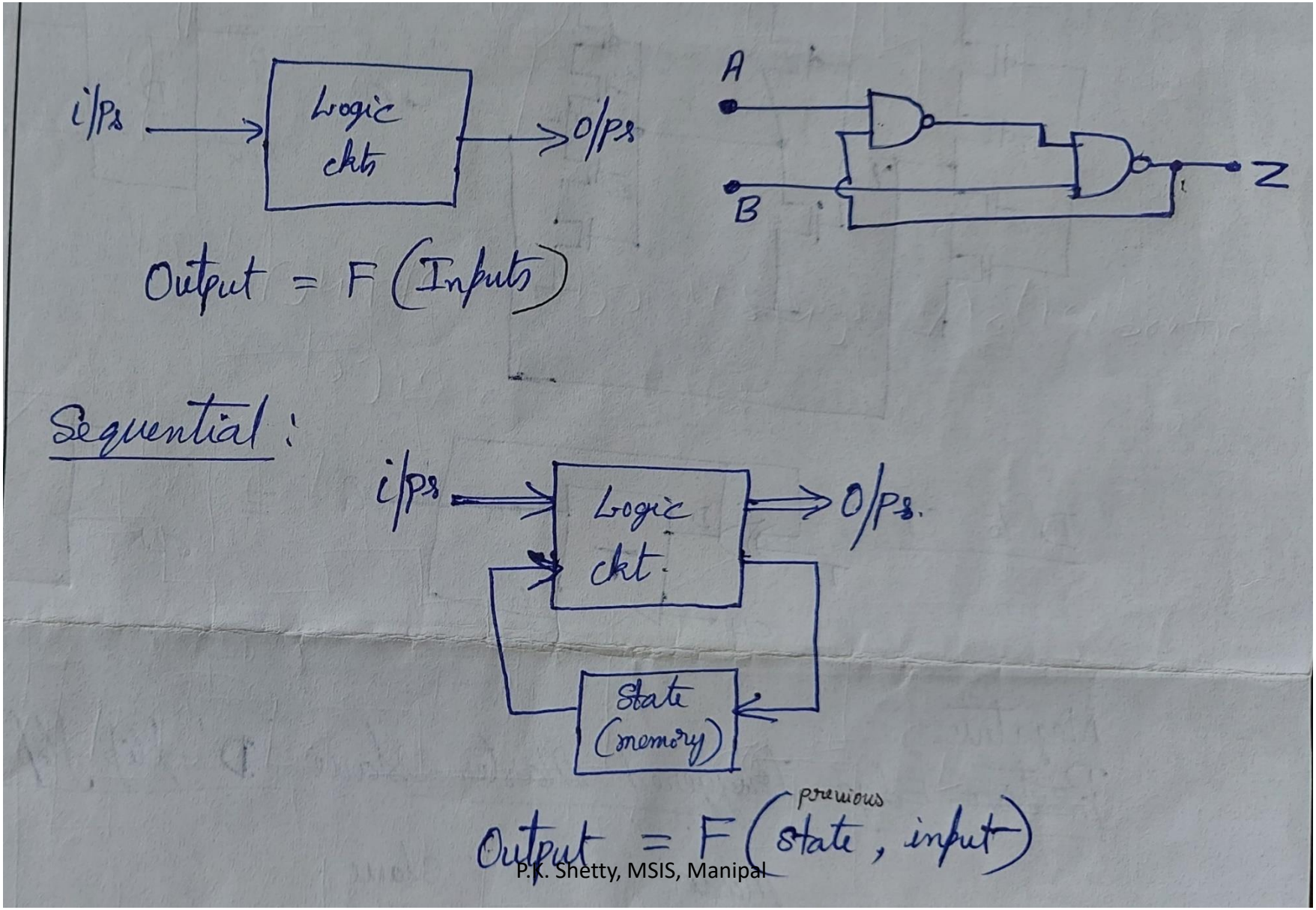
Adders

Subtractor

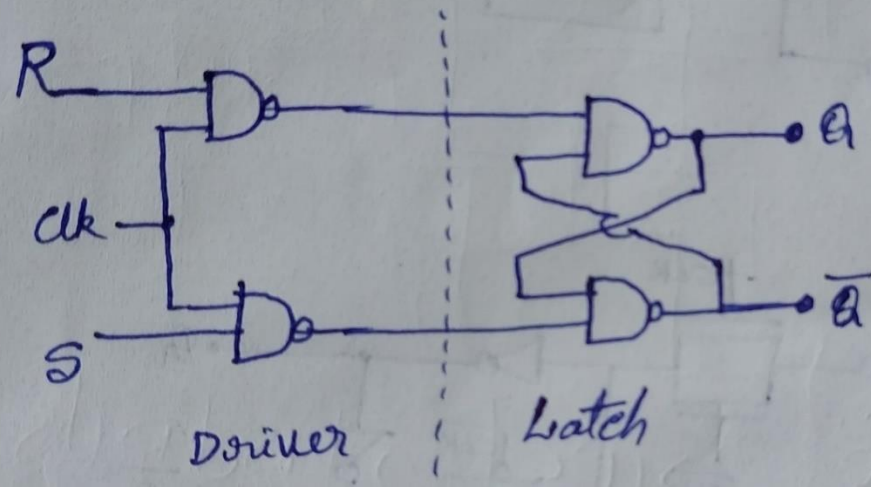
Adder / Subtractor

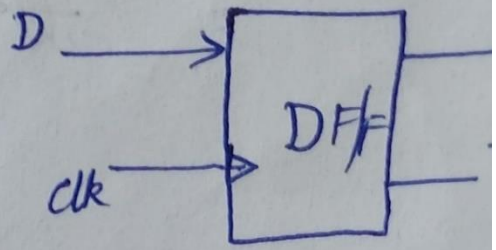
Comparators

Sequential Circuits

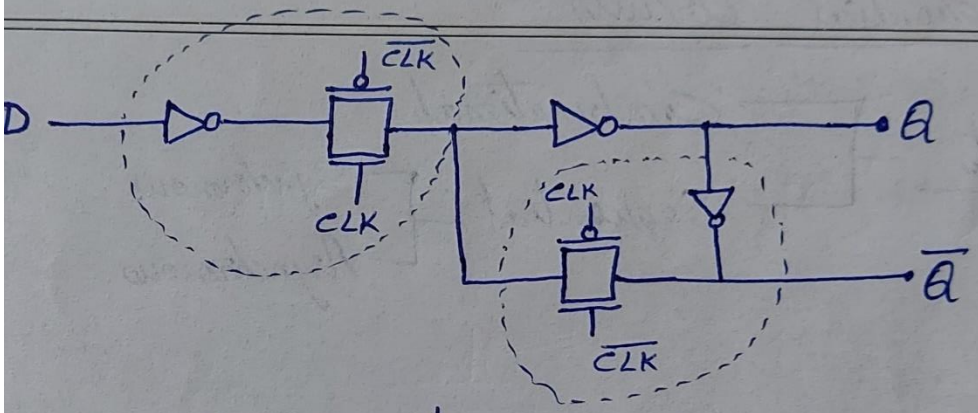


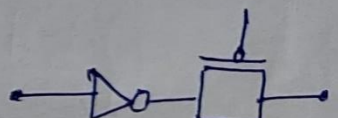
Simple D Flip flop: [+ve level triggered]



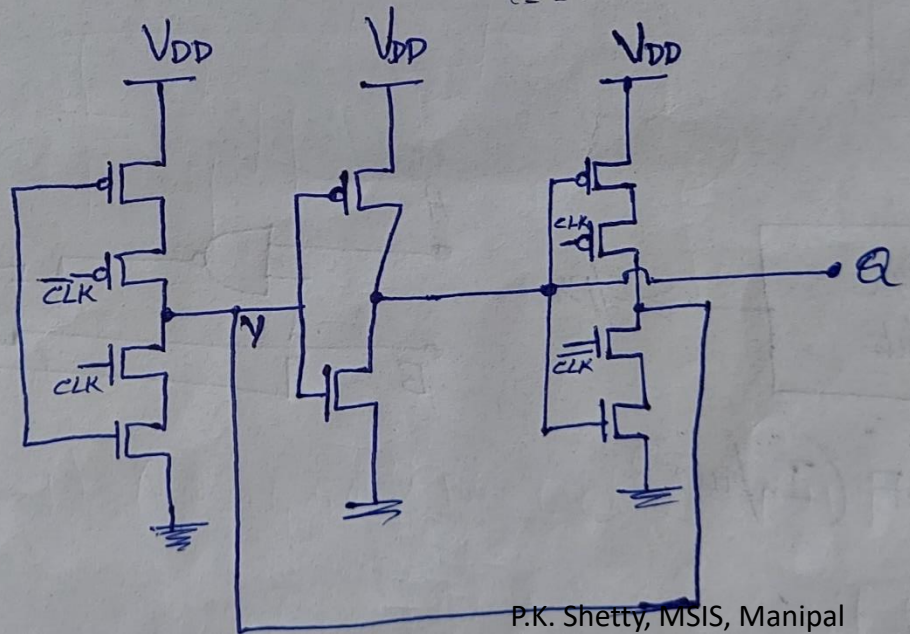


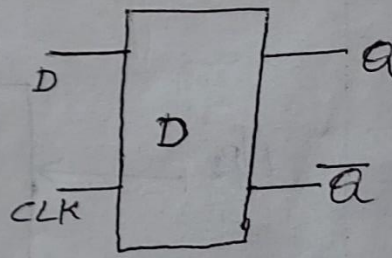
Positive Level triggered 'D' flip flop:-



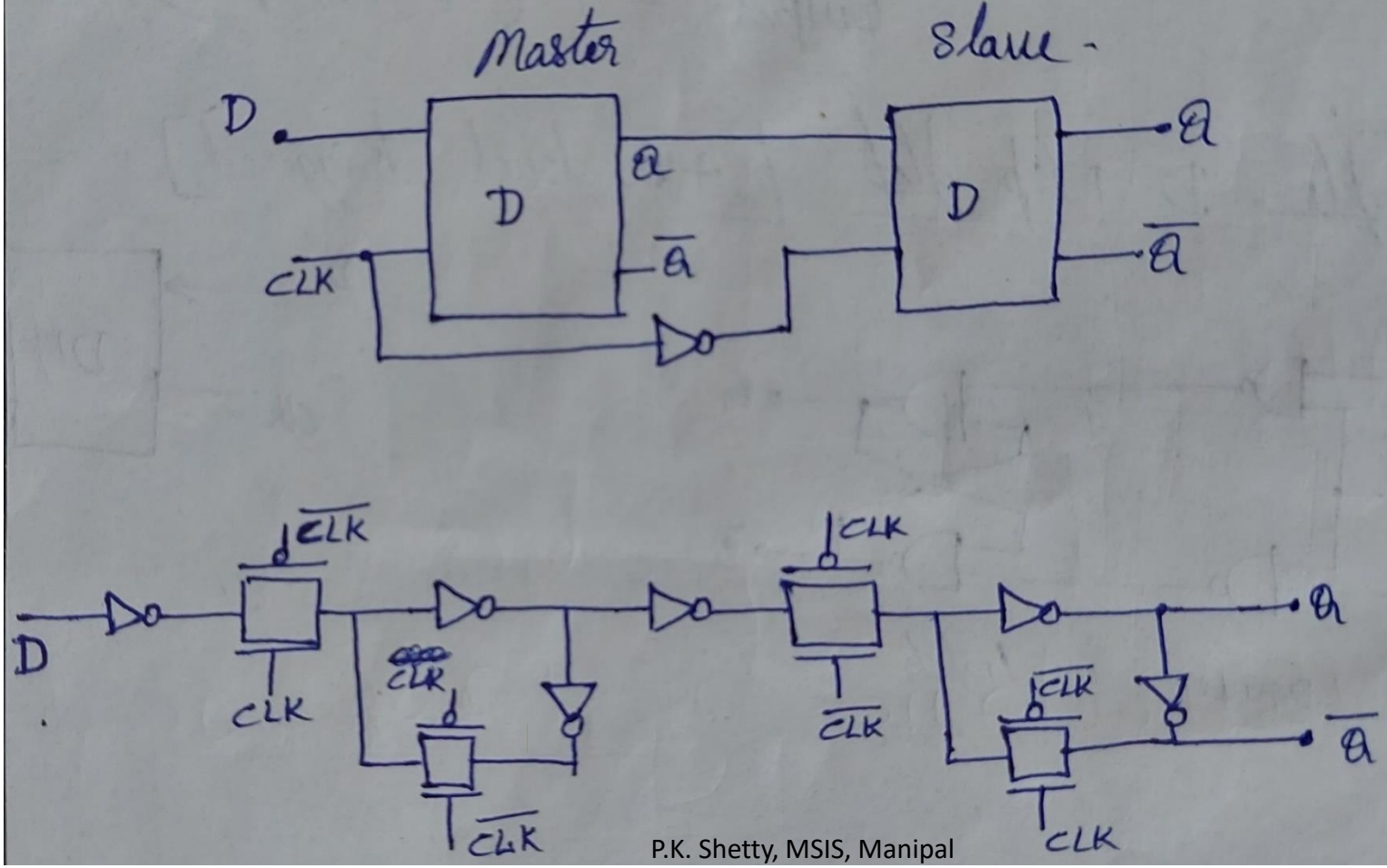


Tristate inverter

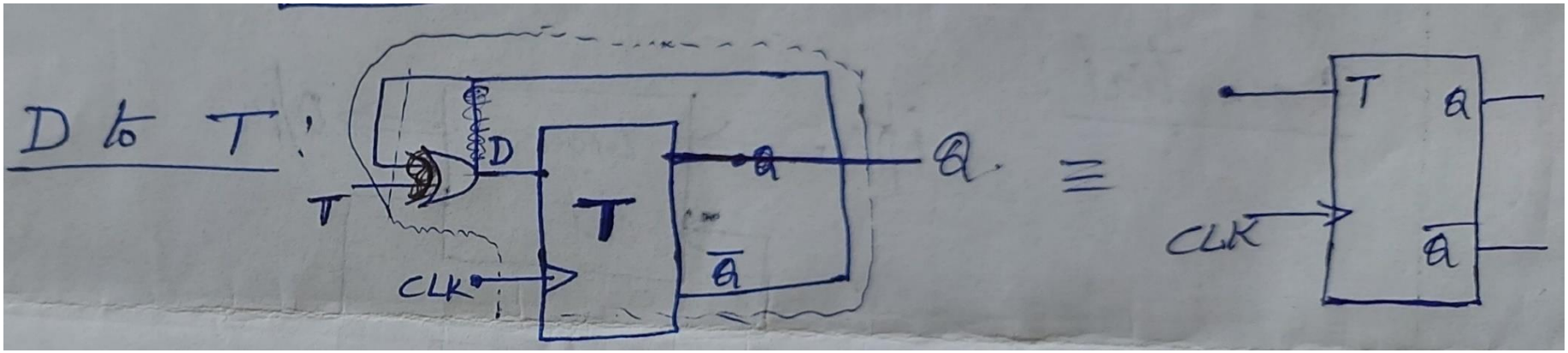




Negative ~~Positive~~ Edge triggered Master-Slave 'D' flip flop :-



D to T:



4-bit Synchronous Counter