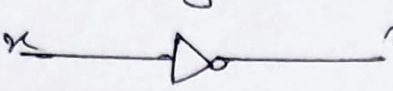
NOR Implementation

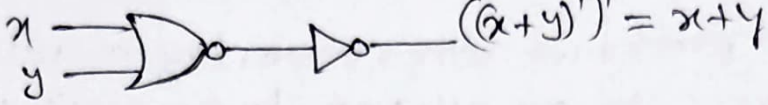
→ The NOR operation is the dual of the NAND operation.

→ Therefore, all procedures and rules for NOR logic are the duals of the corresponding procedures and rules developed for NAND logic.

→ The NOR gate is another universal gate that can be used to implement any Boolean function.

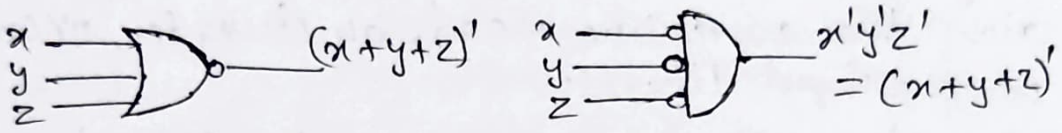
→ The implementation of the complement, OR, and AND operations with NOR gates is as follows:

Inverter 

OR 

AND 

→ The two symbols designate the same NOR operation and are logically identical because of DeMorgan's theorem are as follows:



OR-invert Invert-AND

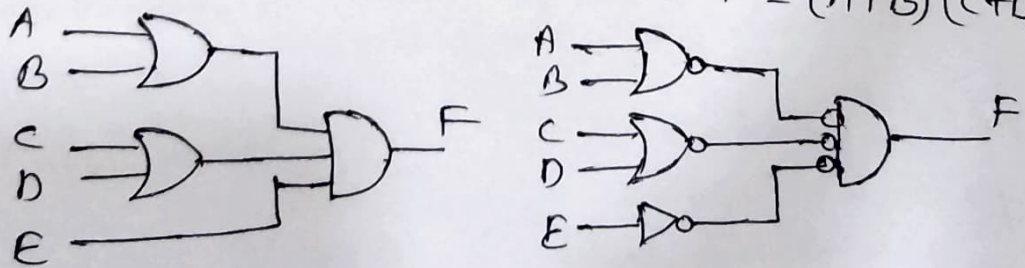
→ A two-level implementation with NOR gates requires that the function be simplified into product-of-sums form.

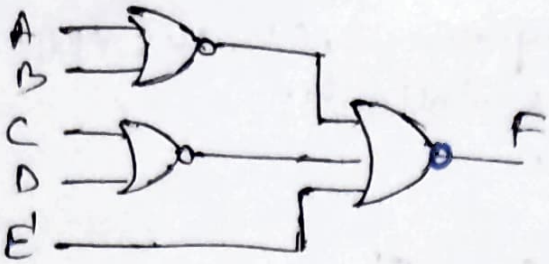
→ A product-of-sums expression is implemented with a first level of OR gates that produce the sum terms followed by a second-level AND gate to produce the product.

→ The transformation from the OR-AND diagram to a NOR diagram is achieved by changing the OR gates to NOR gates with OR-invert graphic symbols and the AND gates to a NOR gate with an invert-AND graphic symbol.

→ A single literal term going into the second-level gate must be complemented.

→ Consider a Boolean function $F = (A+B)(C+D)E$





→ The procedure for converting a multilevel AND-OR diagram to an all-NOR diagram is similar to that for NAND gates

→ For the NOR case, we must convert each OR gate to an OR-invert symbol and each AND gate to an invert-AND symbol.

→ Any bubble that is not compensated by another bubble along the same line needs an inverter, or complementation of the input literal.

Q→ Consider a Boolean function $F = (AB' + A'B)(C + D')$