

Design and Analysis of 6-Input CMOS AND/OR Gate Using Multi-Stage NAND/NOR Implementations

Ayan Garg (B23484) *, Achal Shah (T25119) †, Krishna Gorai (T25111) ‡

* b23484@students.iitmandi.ac.in

† t25119@students.iitmandi.ac.in

‡ t25111@students.iitmandi.ac.in

I. 6-INPUT AND/OR GATE IMPLEMENTATIONS

A. AND Implementation-A (6-input NAND + Inverter)

A single 6-input NAND gate is used:

$$X = \overline{ABCDEF}.$$

The output is then inverted:

$$out = \overline{X} = \overline{\overline{ABCDEF}} = ABCDEF.$$

Thus, a 6-input NAND followed by an inverter realizes the 6-input AND function.

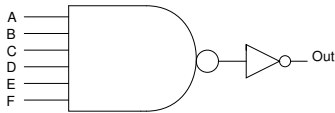


Fig. 1: AND Implementation-A: 6-input AND using 6-input NAND followed by an inverter.

B. AND Implementation-B (3-input NAND + 2-input NAND)

Inputs are grouped into two 3-input NAND gates:

$$X_1 = \overline{ABC}, \quad X_2 = \overline{DEF}.$$

The final stage is a 2-input NAND:

$$out = \overline{X_1 X_2} = \overline{(\overline{ABC})(\overline{DEF})} = ABCDEF.$$

Thus, two 3-input NAND gates followed by one 2-input NAND realize the 6-input AND function.

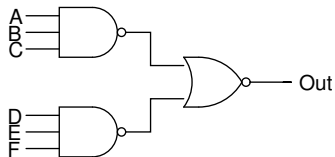


Fig. 2: AND Implementation-B: 6-input AND using 3-input NAND and 2-input NAND decomposition.

C. AND Implementation-C (2-input NAND + 3-input NAND)

Inputs are first paired using three 2-input NAND gates:

$$X_1 = \overline{AB}, \quad X_2 = \overline{CD}, \quad X_3 = \overline{EF}.$$

The final stage is a 3-input NAND:

$$out = \overline{X_1 X_2 X_3} = \overline{(\overline{AB})(\overline{CD})(\overline{EF})} = ABCDEF.$$

Thus, three 2-input NAND gates followed by one 3-input NAND realize the 6-input AND function.

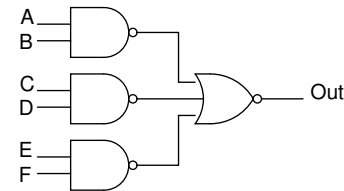


Fig. 3: AND Implementation-C: 6-input AND using 2-input NAND and 3-input NAND decomposition.

D. OR Implementation-A (6-input NOR + Inverter)

A single 6-input NOR gate is used:

$$X = \overline{A + B + C + D + E + F}.$$

The output is then inverted:

$$\begin{aligned} out &= \overline{X} = \overline{\overline{A + B + C + D + E + F}} \\ &= A + B + C + D + E + F. \end{aligned}$$

Thus, a 6-input NOR followed by an inverter realizes the 6-input OR function.

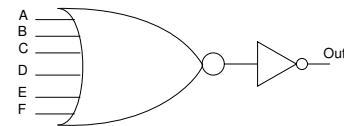


Fig. 4: OR Implementation-A: 6-input OR using 6-input NOR followed by an inverter.

E. OR Implementation-B (3-input NOR + 2-input NOR)

Inputs are grouped into two 3-input NOR gates:

$$X_1 = \overline{A + B + C}, \quad X_2 = \overline{D + E + F}.$$

The final stage is a 2-input NOR:

$$\begin{aligned} out &= \overline{X_1 + X_2} = \overline{(\overline{A + B + C}) + (\overline{D + E + F})} \\ &= A + B + C + D + E + F. \end{aligned}$$

Thus, two 3-input NOR gates followed by one 2-input NOR realize the 6-input OR function.

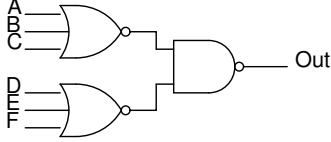


Fig. 5: OR Implementation-B: 6-input OR using 3-input NOR and 2-input NOR decomposition.

F. OR Implementation-C (2-input NOR + 3-input NOR)

Inputs are first paired using three 2-input NOR gates:

$$X_1 = \overline{A + B}, \quad X_2 = \overline{C + D}, \quad X_3 = \overline{E + F}.$$

The final stage is a 3-input NOR:

$$\begin{aligned} out &= \overline{X_1 + X_2 + X_3} = \overline{(\overline{A + B}) + (\overline{C + D}) + (\overline{E + F})} \\ &= A + B + C + D + E + F. \end{aligned}$$

Thus, three 2-input NOR gates followed by one 3-input NOR realize the 6-input OR function.

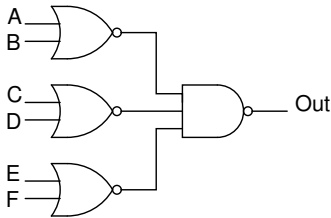


Fig. 6: OR Implementation-C: 6-input OR using 2-input NOR and 3-input NOR decomposition.

II. AREA, DELAY AND POWER COMPARISON

Table I compares the area (transistor count), average delay, and power for all six implementations of the 6-input AND/OR gate.

TABLE I: Delay and Power Comparison for All 6 Implementations

Metric	Avg Delay (ps)	Power (μ W)
AND-A	88.64	2.154
AND-B	84.51	2.625
AND-C	92.10	3.747
OR-A	85.19	3.159
OR-B	83.10	2.871
OR-C	82.80	3.312

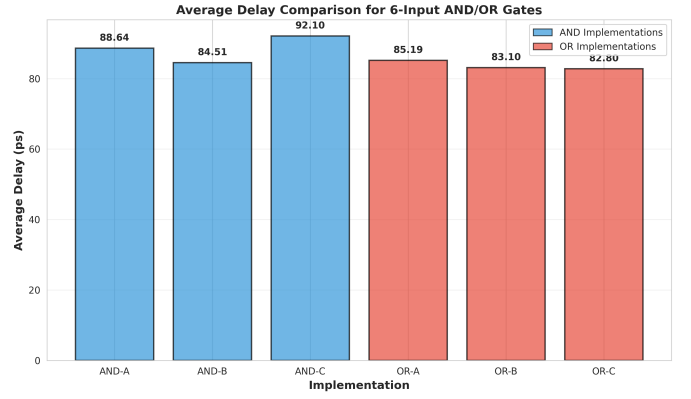


Fig. 7: Graphical Representation of Delay for each configuration

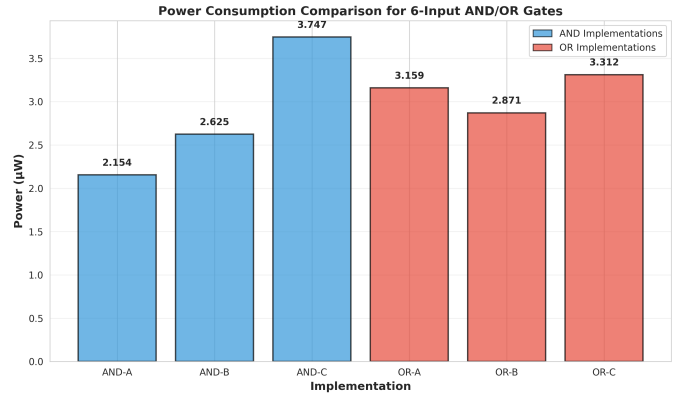


Fig. 8: Graphical Representation of Power for each configuration

Discussion: The transistor count is used as an approximate measure of area, since CMOS gate area is directly proportional to the number of transistors. Implementation-A uses fewer gates, therefore it requires fewer transistors compared to multi-stage decompositions.

The average delay is mainly influenced by the trade-off between *transistor stacking* and *logic depth*. Although Implementation-A has fewer stages, it contains a larger series stack, which increases the effective resistance and hence the propagation delay. In Implementation-B and Implementation-C, the stack height is reduced by using smaller NAND/NOR gates, but the delay may increase due to additional intermediate stages.

Power consumption depends on switching activity and the presence of internal switching nodes. Multi-stage implementations generally show higher power due to extra internal node transitions, which increase dynamic power even though the output load capacitance is fixed at 1 fF. Therefore, the reported results demonstrate the practical tradeoff between area, delay, and power in different CMOS logic decompositions.

III. CHANGING OF SWITCHING ACTIVITY

In this experiment, the switching activity is kept 0.5 and 0.125 for one input each while the other inputs have an switching activity between the two. The input transitions for

each input can be seen in Table II. The resulting average delay and dynamic power for all six implementations are summarized in Table III.

TABLE II: Input Transition Table

Input	T0	T1	T2	T3	T4	T5	T6	T7
A	0	0	0	0	1	1	1	1
B	0	1	0	1	0	1	0	1
C	0	1	0	1	0	1	1	1
D	0	1	0	1	0	1	1	1
E	0	1	0	1	0	1	1	1
F	0	1	0	1	0	1	1	1

TABLE III: Delay, Power and PDP Comparison for All 6 Implementations

Metric	Avg Delay (ps)	Power (μ W)	PDP (fJ)
AND-A	159	1.971	313.389
AND-B	140.4	2.521	353.948
AND-C	164.3	3.527	579.486
OR-A	84.35	3.069	258.867
OR-B	84.05	2.77	232.819
OR-C	85.5	3.084	263.682

Discussion: One input is assigned a high switching activity ($\alpha = 0.5$), another is assigned a low switching activity ($\alpha = 0.125$), while the remaining inputs have intermediate transition rates. Since dynamic power in CMOS circuits is given by

$$P_{dyn} = \alpha C_L V_{DD}^2 f,$$

power consumption is directly proportional to the switching activity factor α .

From Table III, it is observed that delay values increase compared to the previous case. This is because non-uniform input transitions create different effective input arrival patterns, which affect the internal charging and discharging paths.

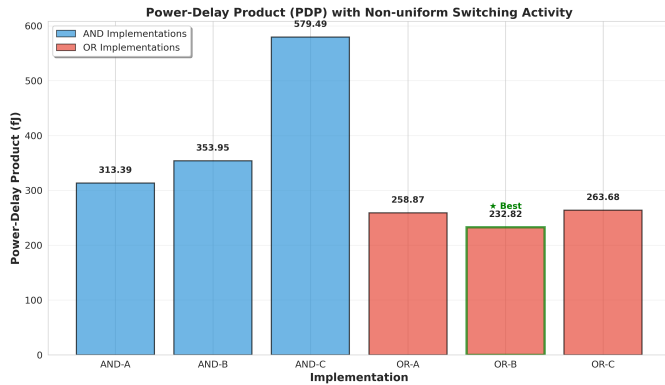


Fig. 9: Power-Delay Product(PDP) with Non-uniform Switching Activity

IV. EFFECT OF INPUT RISE/FALL TIME VARIATION

In this experiment, the rise/fall time of one selected input is increased to 500 ps while keeping all other inputs identical. The resulting average delay and dynamic power for all six implementations are summarized in Table IV.

TABLE IV: Effect of Input Rise/Fall Time (Selected Input = 500 ps)

Metric	Avg Delay (ps)	Power (μ W)
AND-A	200.86	2.615
AND-B	212.55	3.071
AND-C	232.94	4.282
OR-A	213.93	4.473
OR-B	224.06	3.344
OR-C	227.06	3.541

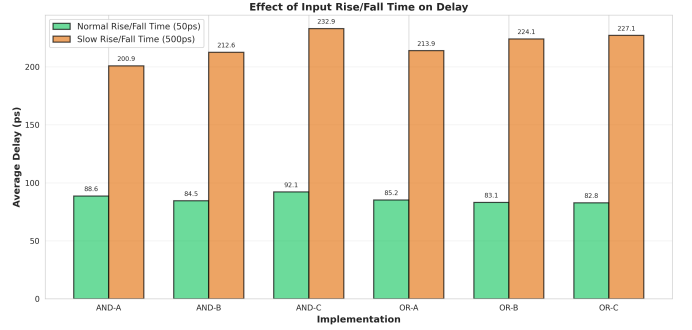


Fig. 10: Effect of Input Rise/Fall Time on Delay

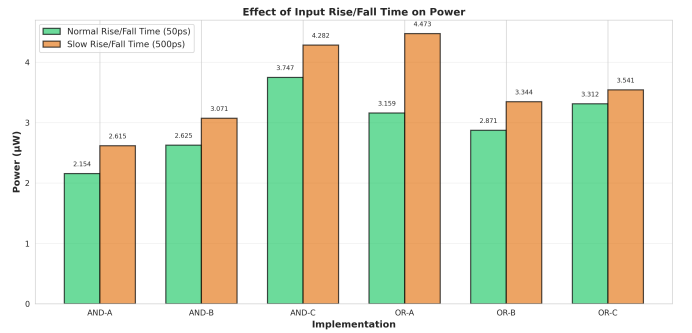


Fig. 11: Rise/Fall Time Effect on Power

Discussion: From Table IV, it is observed that increasing the rise/fall time of one input to 500 ps increases the average propagation delay for all implementations. This occurs because slower input transitions reduce the effective drive strength during switching, causing the output node to charge/discharge more slowly through the pull-up/pull-down networks.

Among the AND implementations, AND-A shows the lowest delay (200.86 ps), while AND-C shows the highest delay (232.94 ps). This trend is mainly due to the increased logic depth and additional internal stages in the decomposed structures, which contribute extra propagation delay when the input transition becomes slow. A similar trend is observed for the OR implementations, where OR-A provides lower delay compared to OR-B and OR-C.

The power results show that implementations with higher transistor count and more internal switching nodes consume higher dynamic power. In particular, AND-C and OR-A show the highest power consumption (4.282 μ W and 4.473 μ W respectively), which is attributed to increased internal node transitions and longer short-circuit current duration caused by slow input edges. Overall, the results demonstrate that

slower rise/fall times degrade delay performance and may also increase power due to increased short-circuit and internal switching effects.

V. LEAKAGE POWER ESTIMATION (IMPLEMENTATION-A)

Leakage power is estimated for Implementation-A (6-input NAND/NOR followed by an inverter) for the specified input vectors. The measured leakage values are reported in Table V.

TABLE V: Leakage Power for Implementation-A Under Different Input Vectors

Input Vector	AND-A Leakage (pW)	OR-A Leakage (pW)
000000	20.12	73.12
000001	20.23	91.23
000011	20.52	38.32
000111	21.46	27.64
001111	24.19	23.45
011111	42.34	21.27
111111	94.38	19.89

Discussion: From Table V, it is observed that leakage power strongly depends on the applied input vector due to the *stacking effect* in series-connected MOS transistors. For AND-A (6-input NAND + inverter), the leakage increases significantly as the input vector moves from 000000 to 111111. This occurs because more NMOS devices in the pull-down stack become ON, reducing the number of OFF stacked transistors and hence reducing the stack-induced leakage suppression. Therefore, the worst-case leakage for AND-A occurs near the all-ones condition (111111).

For OR-A (6-input NOR + inverter), the opposite trend is observed. Leakage is maximum for 000000 and decreases as the input vector approaches 111111. This is because in a NOR gate the PMOS pull-up network is series-connected, and the leakage is higher when more PMOS devices are ON (or fewer OFF devices are stacked). As more inputs become 1, more PMOS devices turn OFF, increasing the stack effect and reducing leakage. Hence, the minimum leakage for OR-A occurs at 111111.

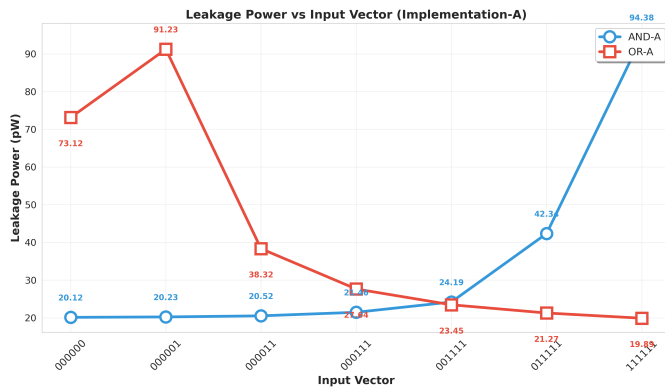


Fig. 12: Comparison plot of leakage power with respect to input vector patterns for implementation-A

VI. EFFECT OF HIGH-ACTIVITY INPUT POSITION ON POWER (IMPLEMENTATION-A)

In this experiment, the input having the highest switching activity is moved from the top of the transistor stack to the bottom in Implementation-A (6-input NAND/NOR followed by an inverter). The measured average power for each position is reported in Table VI.

TABLE VI: Effect of High-Activity Input Position on Power (Implementation-A)

Position of High-Activity Input	OR-A (μ W)	AND-A (nW)
Top (Near Output)	1.945	563.0
2nd	2.534	97.47
3rd	3.490	227.9
4th	4.389	378.7
5th	5.261	557.6
Bottom	5.427	780.8

Discussion: From Table VI, it is observed that the average power changes significantly when the highest-activity input is moved along the transistor stack. This behavior occurs because the input position modifies the effective internal node switching and the short-circuit current contribution during transitions. When the high-activity transistor is placed closer to the bottom of the stack, more internal nodes experience charging/discharging and the effective stack shielding is reduced, resulting in higher dynamic power. Conversely, placing the high-activity input closer to the top reduces unnecessary internal transitions and hence lowers the overall power. Therefore, input ordering in large-stack CMOS gates is an important low-power optimization technique.

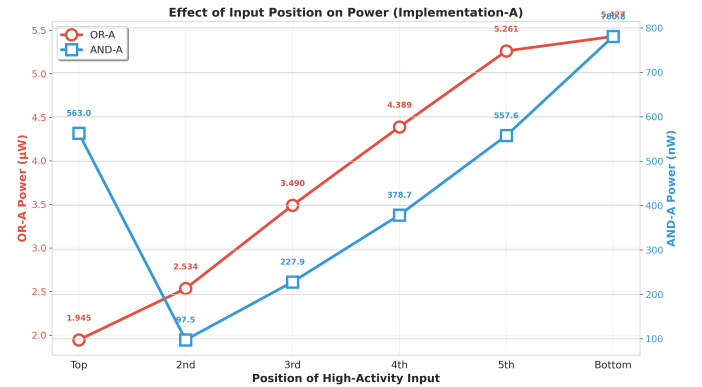


Fig. 13: Comparison plot of power consumption with respect to different input position with highest activity