

Advanced Digital VLSI Circuits and Physical Design

(DSE – 3)

Practical File

Submitted By

Shivam Chaurasiya

Roll No.: 23294917129 (Batch: ECE B – B2)

B. Tech Electronics and Communication Engineering
(Semester V)

To

Dr. Khushwant Sehra

(Assistant Professor)

Department of Electronics and Communication Engineering



FACULTY OF TECHNOLOGY

UNIVERSITY OF DELHI

NEW DELHI – 110007

(2025 – 2026)

Record of Experiments

S. No.	Experiment Title	Page No.	Date	Remarks
1	Experiment 1: Propagation and Contamination Delay.	1		
2	Experiment 2: Propagation and Contamination Delay in a Pipeline.	12		
3	Experiment 3: To design, implement, and analyze a CMOS inverter by performing schematic design, transient, DC characterization and calculating delay.	19		
4	Experiment 4: To create the physical layout of a CMOS inverter, perform DRC/LVS checks, RC extract, and run post-layout simulations.	24		
5	Experiment 5: To design and analyze a CMOS NAND2 by performing transient simulation and propagation delay evaluation at the schematic level, followed by creating the physical layout and RC extraction.	27		
6	Experiment 6: To design and analyze a CMOS NOR2 by performing transient simulation and propagation delay evaluation at the schematic level, followed by creating the physical layout and RC extraction.	33		
7	Experiment 7: To design PTL as NMOS, run transient simulation.	38		
8	Experiment 8: To design PTL as PMOS, run transient simulation.	41		
9	Experiment 9: To design a 4×4 NOR-based ROM array, verify its logical functionality and create physical layout with DRC/LVS checks, and RC extraction.	44		
10	Experiment 10: To design a 4×4 NAND-based ROM array, verify its logical functionality.	48		
11	Experiment 11: To design a 6T CMOS based SRAM, verify its logical functionality through simulation.	51		

Experiment 1

Propagation and Contamination Delay

1.1 Aim

- Find out the following for a **3 input NAND** and **3 input NOR** gate:
 - Rise and Fall Propagation Delay
 - Rise and Fall Contamination Delay
- Compare and check the effect of using different technology files.

1.2 Theory

Propagation delay and contamination delay are fundamental timing metrics used to describe the dynamic response of CMOS logic gates. In this experiment, a 3-input NAND and a 3-input NOR gate are studied through transient simulations to obtain their rise and fall propagation delays, as well as rise and fall contamination delays, under different technology files.

1. Propagation Delay

Propagation delay (t_p) is defined as the time interval between the input transition crossing 50% of V_{DD} and the corresponding output transition crossing 50% of its final value. It is a measure of how long the gate takes to respond to a valid input change.

Two propagation delays are commonly specified:

t_{pLH} : Delay for output transition LOW → HIGH

t_{pHL} : Delay for output transition HIGH → LOW

A single effective propagation delay is often taken as the average:

$$t_p = \frac{t_{pLH} + t_{pHL}}{2}$$

2. Contamination Delay

Contamination delay (t_c) is the **minimum** time after an input begins to change before the output is guaranteed to leave its previous logic level. It captures the earliest possible disturbance at the output due to an input transition.

The two contamination delays are:

t_{cLH} : Earliest output transition LOW → HIGH

t_{cHL} : Earliest output transition HIGH → LOW

3. Technology File Effect

The choice of technology file determines key device parameters such as:

- carrier mobility and drive strength,
- threshold voltage (V_{th}),
- intrinsic gate and diffusion capacitances,
- interconnect and junction parasitics.

Changes in these parameters directly affect both propagation and contamination delays. In general, more advanced (smaller) technology nodes exhibit reduced parasitics and higher current drive, leading to faster gate operation for the same logic topology.

1.3 SPICE Code

1.3.1 3 input NAND

Listing 1.1: 3 input NAND

```

1 * 3-INPUT NAND GATE
2 * Technologies: 0.35 $\mu$m, 0.18 $\mu$m, 45 nm
3
4 * Model files
5 .include 'model_cmos_tsmc035.lib'
6 .include 'model_cmos_tsmc018.lib'
7 .include 'model_cmos_45n.lib'
8
9
10 * SUPPLIES
11 Vdd35 Vdd35 0 3.3
12 Vdd18 Vdd18 0 1.8
13 Vdd45 Vdd45 0 1.0
14
15 * INPUT SOURCES
16 * 0.35 $\mu$m inputs (0 -> 3.3 V)
17 Va35 a35 0 PULSE(0 3.3 0n 10p 10p 1n 10n)
18 Vb35 b35 0 PULSE(0 3.3 10n 10p 10p 1n 10n)
19 Vc35 c35 0 PULSE(0 3.3 20n 10p 10p 1n 10n)
20
21 * 0.18 $\mu$m inputs (0 -> 1.8 V)
22 Va18 a18 0 PULSE(0 1.8 0n 10p 10p 1n 10n)
23 Vb18 b18 0 PULSE(0 1.8 10n 10p 10p 1n 10n)
24 Vc18 c18 0 PULSE(0 1.8 20n 10p 10p 1n 10n)
25
26 * 45 nm inputs (0 -> 1.0 V)
27 Va45 a45 0 PULSE(0 1.0 0n 10p 10p 1n 10n)
28 Vb45 b45 0 PULSE(0 1.0 10n 10p 10p 1n 10n)
29 Vc45 c45 0 PULSE(0 1.0 20n 10p 10p 1n 10n)
30
31 * 0.35 $\mu$m CMOS 3-INPUT NAND
32 * PMOS Pull-up (parallel)
33 M1_35 Y35 a35 Vdd35 Vdd35 cmosp L=0.4u W=1.2u
34 M2_35 Y35 b35 Vdd35 Vdd35 cmosp L=0.4u W=1.2u

```

```

35 M3_35 Y35 c35 Vdd35 Vdd35 cmosp L=0.4u W=1.2u
36
37 * NMOS Pull-down (series)
38 M4_35 n1_35 a35 0 0 cmosn L=0.4u W=0.6u
39 M5_35 n2_35 b35 n1_35 0 cmosn L=0.4u W=0.6u
40 M6_35 Y35 c35 n2_35 0 cmosn L=0.4u W=0.6u
41
42 * 0.18 $mu$m CMOS 3-INPUT NAND
43 * PMOS Pull-up (parallel)
44 M1_18 Y18 a18 Vdd18 Vdd18 CMOSP L=0.18u W=0.54u
45 M2_18 Y18 b18 Vdd18 Vdd18 CMOSP L=0.18u W=0.54u
46 M3_18 Y18 c18 Vdd18 Vdd18 CMOSP L=0.18u W=0.54u
47
48 * NMOS Pull-down (series)
49 M4_18 n1_18 a18 0 0 CMOSN L=0.18u W=0.27u
50 M5_18 n2_18 b18 n1_18 0 CMOSN L=0.18u W=0.27u
51 M6_18 Y18 c18 n2_18 0 CMOSN L=0.18u W=0.27u
52
53 * 45 nm PTM CMOS 3-INPUT NAND
54 * PMOS Pull-up (parallel)
55 M1_45 Y45 a45 Vdd45 Vdd45 pmos L=45n W=180n
56 M2_45 Y45 b45 Vdd45 Vdd45 pmos L=45n W=180n
57 M3_45 Y45 c45 Vdd45 Vdd45 pmos L=45n W=180n
58
59 * NMOS Pull-down (series)
60 M4_45 n1_45 a45 0 0 nmos L=45n W=90n
61 M5_45 n2_45 b45 n1_45 0 nmos L=45n W=90n
62 M6_45 Y45 c45 n2_45 0 nmos L=45n W=90n
63
64 * TRANSIENT ANALYSIS
65 .tran 0 200n 0 0.02n
66
67 * =====
68 * MEASUREMENTS
69 * =====
70
71 * 0.35 $mu$m (VDD = 3.3 V)
72 * Propagation Delay (50% = 1.65 V)
73 .meas tran tpHL_A_35 TRIG V(a35) VAL=1.65 RISE=1 TARG V(Y35) VAL=1.65
    FALL=1
74 .meas tran tpLH_A_35 TRIG V(a35) VAL=1.65 FALL=1 TARG V(Y35) VAL=1.65
    RISE=1
75 .meas tran tpHL_B_35 TRIG V(b35) VAL=1.65 RISE=1 TARG V(Y35) VAL=1.65
    FALL=1
76 .meas tran tpLH_B_35 TRIG V(b35) VAL=1.65 FALL=1 TARG V(Y35) VAL=1.65
    RISE=1
77 .meas tran tpHL_C_35 TRIG V(c35) VAL=1.65 RISE=1 TARG V(Y35) VAL=1.65
    FALL=1
78 .meas tran tpLH_C_35 TRIG V(c35) VAL=1.65 FALL=1 TARG V(Y35) VAL=1.65
    RISE=1
79
80 * Contamination Delay (10% = 0.33 V, 90% = 2.97 V)
81 .meas tran tcdHL_A_35 TRIG V(a35) VAL=1.65 RISE=1 TARG V(Y35) VAL=2.97
    FALL=1
82 .meas tran tcdLH_A_35 TRIG V(a35) VAL=1.65 FALL=1 TARG V(Y35) VAL=0.33
    RISE=1
83 .meas tran tcdHL_B_35 TRIG V(b35) VAL=1.65 RISE=1 TARG V(Y35) VAL=2.97
    FALL=1

```

```

84 .meas tran tcdLH_B_35 TRIG V(b35) VAL=1.65 FALL=1 TARG V(Y35) VAL=0.33
     RISE=1
85 .meas tran tcdHL_C_35 TRIG V(c35) VAL=1.65 RISE=1 TARG V(Y35) VAL=2.97
     FALL=1
86 .meas tran tcdLH_C_35 TRIG V(c35) VAL=1.65 FALL=1 TARG V(Y35) VAL=0.33
     RISE=1
87
88 * 0.18 $\\mu$m (VDD = 1.8 V)
89 .meas tran tpHL_A_18 TRIG V(a18) VAL=0.9 RISE=1 TARG V(Y18) VAL=0.9
     FALL=1
90 .meas tran tpLH_A_18 TRIG V(a18) VAL=0.9 FALL=1 TARG V(Y18) VAL=0.9
     RISE=1
91 .meas tran tpHL_B_18 TRIG V(b18) VAL=0.9 RISE=1 TARG V(Y18) VAL=0.9
     FALL=1
92 .meas tran tpLH_B_18 TRIG V(b18) VAL=0.9 FALL=1 TARG V(Y18) VAL=0.9
     RISE=1
93 .meas tran tpHL_C_18 TRIG V(c18) VAL=0.9 RISE=1 TARG V(Y18) VAL=0.9
     FALL=1
94 .meas tran tpLH_C_18 TRIG V(c18) VAL=0.9 FALL=1 TARG V(Y18) VAL=0.9
     RISE=1
95
96 * Contamination Delay (10% = 0.18 V, 90% = 1.62 V)
97 .meas tran tcdHL_A_18 TRIG V(a18) VAL=0.9 RISE=1 TARG V(Y18) VAL=1.62
     FALL=1
98 .meas tran tcdLH_A_18 TRIG V(a18) VAL=0.9 FALL=1 TARG V(Y18) VAL=0.18
     RISE=1
99 .meas tran tcdHL_B_18 TRIG V(b18) VAL=0.9 RISE=1 TARG V(Y18) VAL=1.62
     FALL=1
100 .meas tran tcdLH_B_18 TRIG V(b18) VAL=0.9 FALL=1 TARG V(Y18) VAL=0.18
     RISE=1
101 .meas tran tcdHL_C_18 TRIG V(c18) VAL=0.9 RISE=1 TARG V(Y18) VAL=1.62
     FALL=1
102 .meas tran tcdLH_C_18 TRIG V(c18) VAL=0.9 FALL=1 TARG V(Y18) VAL=0.18
     RISE=1
103
104 * 45 nm (VDD = 1.0 V)
105 .meas tran tpHL_A_45 TRIG V(a45) VAL=0.5 RISE=1 TARG V(Y45) VAL=0.5
     FALL=1
106 .meas tran tpLH_A_45 TRIG V(a45) VAL=0.5 FALL=1 TARG V(Y45) VAL=0.5
     RISE=1
107 .meas tran tpHL_B_45 TRIG V(b45) VAL=0.5 RISE=1 TARG V(Y45) VAL=0.5
     FALL=1
108 .meas tran tpLH_B_45 TRIG V(b45) VAL=0.5 FALL=1 TARG V(Y45) VAL=0.5
     RISE=1
109 .meas tran tpHL_C_45 TRIG V(c45) VAL=0.5 RISE=1 TARG V(Y45) VAL=0.5
     FALL=1
110 .meas tran tpLH_C_45 TRIG V(c45) VAL=0.5 FALL=1 TARG V(Y45) VAL=0.5
     RISE=1
111
112 * Contamination Delay (10% = 0.1 V, 90% = 0.9 V)
113 .meas tran tcdHL_A_45 TRIG V(a45) VAL=0.5 RISE=1 TARG V(Y45) VAL=0.9
     FALL=1
114 .meas tran tcdLH_A_45 TRIG V(a45) VAL=0.5 FALL=1 TARG V(Y45) VAL=0.1
     RISE=1
115 .meas tran tcdHL_B_45 TRIG V(b45) VAL=0.5 RISE=1 TARG V(Y45) VAL=0.9
     FALL=1
116 .meas tran tcdLH_B_45 TRIG V(b45) VAL=0.5 FALL=1 TARG V(Y45) VAL=0.1
     RISE=1

```

```

117 .meas tran tcdHL_C_45 TRIG V(c45) VAL=0.5 RISE=1 TARG V(Y45) VAL=0.9
    FALL=1
118 .meas tran tcdLH_C_45 TRIG V(c45) VAL=0.5 FALL=1 TARG V(Y45) VAL=0.1
    RISE=1
119
120 .end

```

1.3.2 3 input NOR

Listing 1.2: 3 input NOR

```

1 * 3-INPUT NOR GATE - COMBINED NETLIST
2 * Technologies: 0.35 $\mu m, 0.18 $\mu m, 45 nm
3
4 * Model files
5 .include 'model_cmos_tsmc035.lib'
6 .include 'model_cmos_tsmc018.lib'
7 .include 'model_cmos_45n.lib'
8
9 * SUPPLIES
10 Vdd35 Vdd35 0 3.3
11 Vdd18 Vdd18 0 1.8
12 Vdd45 Vdd45 0 1.0
13
14 * INPUT SOURCES
15 * 0.35 $\mu m inputs (0 -> 3.3 V)
16 Va35 a35 0 PULSE(0 3.3 0 10p 10p 1n 10n)
17 Vb35 b35 0 PULSE(0 3.3 10n 10p 10p 1n 10n)
18 Vc35 c35 0 PULSE(0 3.3 20n 10p 10p 1n 10n)
19
20 * 0.18 $\mu m inputs (0 -> 1.8 V)
21 Va18 a18 0 PULSE(0 1.8 0 10p 10p 1n 10n)
22 Vb18 b18 0 PULSE(0 1.8 10n 10p 10p 1n 10n)
23 Vc18 c18 0 PULSE(0 1.8 20n 10p 10p 1n 10n)
24
25 * 45 nm inputs (0 -> 1.0 V)
26 Va45 a45 0 PULSE(0 1.0 0 10p 10p 1n 10n)
27 Vb45 b45 0 PULSE(0 1.0 10n 10p 10p 1n 10n)
28 Vc45 c45 0 PULSE(0 1.0 20n 10p 10p 1n 10n)
29
30 * 0.35 $\mu m CMOS 3-INPUT NOR
31 * Pull-Up Network (series PMOS)
32 M1_35 z35 a35 Vdd35 Vdd35 cmosp L=0.4u W=1.2u
33 M2_35 x35 b35 z35 Vdd35 Vdd35 cmosp L=0.4u W=1.2u
34 M3_35 Y35 c35 x35 Vdd35 Vdd35 cmosp L=0.4u W=1.2u
35
36 * Pull-Down Network (parallel NMOS)
37 M4_35 Y35 a35 0 0 cmosn L=0.4u W=0.6u
38 M5_35 Y35 b35 0 0 cmosn L=0.4u W=0.6u
39 M6_35 Y35 c35 0 0 cmosn L=0.4u W=0.6u
40
41 * 0.18 $\mu m CMOS 3-INPUT NOR
42 * Pull-Up Network (series PMOS)
43 M1_18 z18 a18 Vdd18 Vdd18 CMOSP L=0.18u W=0.54u
44 M2_18 x18 b18 z18 Vdd18 Vdd18 CMOSP L=0.18u W=0.54u
45 M3_18 Y18 c18 x18 Vdd18 Vdd18 CMOSP L=0.18u W=0.54u
46

```

```

47 * Pull-Down Network (parallel NMOS)
48 M4_18 Y18 a18 0 0 CMOSN L=0.18u W=0.27u
49 M5_18 Y18 b18 0 0 CMOSN L=0.18u W=0.27u
50 M6_18 Y18 c18 0 0 CMOSN L=0.18u W=0.27u
51
52 * 45 nm PTM CMOS 3-INPUT NOR
53 * Pull-Up Network (series PMOS)
54 M1_45 z45 a45 Vdd45 Vdd45 pmos L=45n W=180n
55 M2_45 x45 b45 z45 Vdd45 pmos L=45n W=180n
56 M3_45 Y45 c45 x45 Vdd45 pmos L=45n W=180n
57
58 * Pull-Down Network (parallel NMOS)
59 M4_45 Y45 a45 0 0 nmos L=45n W=90n
60 M5_45 Y45 b45 0 0 nmos L=45n W=90n
61 M6_45 Y45 c45 0 0 nmos L=45n W=90n
62
63 * TRANSIENT ANALYSIS (COMMON)
64 .tran 0 80n 0 10p
65
66 * =====
67 * MEASUREMENTS FOR 0.35 $mu$m
68 * =====
69
70 * Propagation Delay - Input A
71 .meas tran tpHL_A_35 TRIG V(a35) VAL=1.65 RISE=1 TARG V(Y35) VAL=1.65
    FALL=1
72 .meas tran tpLH_A_35 TRIG V(a35) VAL=1.65 FALL=1 TARG V(Y35) VAL=1.65
    RISE=1
73
74 * Propagation Delay - Input B
75 .meas tran tpHL_B_35 TRIG V(b35) VAL=1.65 RISE=1 TARG V(Y35) VAL=1.65
    FALL=1
76 .meas tran tpLH_B_35 TRIG V(b35) VAL=1.65 FALL=1 TARG V(Y35) VAL=1.65
    RISE=1
77
78 * Propagation Delay - Input C
79 .meas tran tpHL_C_35 TRIG V(c35) VAL=1.65 RISE=1 TARG V(Y35) VAL=1.65
    FALL=1
80 .meas tran tpLH_C_35 TRIG V(c35) VAL=1.65 FALL=1 TARG V(Y35) VAL=1.65
    RISE=1
81
82 * Contamination Delay - Input A
83 .meas tran tcdHL_A_35 TRIG V(a35) VAL=1.65 RISE=1 TARG V(Y35) VAL=2.97
    FALL=1
84 .meas tran tcdLH_A_35 TRIG V(a35) VAL=1.65 FALL=1 TARG V(Y35) VAL=0.33
    RISE=1
85
86 * Contamination Delay - Input B
87 .meas tran tcdHL_B_35 TRIG V(b35) VAL=1.65 RISE=1 TARG V(Y35) VAL=2.97
    FALL=1
88 .meas tran tcdLH_B_35 TRIG V(b35) VAL=1.65 FALL=1 TARG V(Y35) VAL=0.33
    RISE=1
89
90 * Contamination Delay - Input C
91 .meas tran tcdHL_C_35 TRIG V(c35) VAL=1.65 RISE=1 TARG V(Y35) VAL=2.97
    FALL=1
92 .meas tran tcdLH_C_35 TRIG V(c35) VAL=1.65 FALL=1 TARG V(Y35) VAL=0.33
    RISE=1

```

```

93
94 * =====
95 * MEASUREMENTS FOR 0.18 $\mu$m
96 * =====
97
98 * Propagation Delay - Input A
99 .meas tran tpHL_A_18 TRIG V(a18) VAL=0.9 RISE=1 TARG V(Y18) VAL=0.9
   FALL=1
100 .meas tran tpLH_A_18 TRIG V(a18) VAL=0.9 FALL=1 TARG V(Y18) VAL=0.9
   RISE=1
101
102 * Propagation Delay - Input B
103 .meas tran tpHL_B_18 TRIG V(b18) VAL=0.9 RISE=1 TARG V(Y18) VAL=0.9
   FALL=1
104 .meas tran tpLH_B_18 TRIG V(b18) VAL=0.9 FALL=1 TARG V(Y18) VAL=0.9
   RISE=1
105
106 * Propagation Delay - Input C
107 .meas tran tpHL_C_18 TRIG V(c18) VAL=0.9 RISE=1 TARG V(Y18) VAL=0.9
   FALL=1
108 .meas tran tpLH_C_18 TRIG V(c18) VAL=0.9 FALL=1 TARG V(Y18) VAL=0.9
   RISE=1
109
110 * Contamination Delay - Input A
111 .meas tran tcdHL_A_18 TRIG V(a18) VAL=0.9 RISE=1 TARG V(Y18) VAL=1.62
   FALL=1
112 .meas tran tcdLH_A_18 TRIG V(a18) VAL=0.9 FALL=1 TARG V(Y18) VAL=0.18
   RISE=1
113
114 * Contamination Delay - Input B
115 .meas tran tcdHL_B_18 TRIG V(b18) VAL=0.9 RISE=1 TARG V(Y18) VAL=1.62
   FALL=1
116 .meas tran tcdLH_B_18 TRIG V(b18) VAL=0.9 FALL=1 TARG V(Y18) VAL=0.18
   RISE=1
117
118 * Contamination Delay - Input C
119 .meas tran tcdHL_C_18 TRIG V(c18) VAL=0.9 RISE=1 TARG V(Y18) VAL=1.62
   FALL=1
120 .meas tran tcdLH_C_18 TRIG V(c18) VAL=0.9 FALL=1 TARG V(Y18) VAL=0.18
   RISE=1
121
122 * =====
123 * MEASUREMENTS FOR 45 nm
124 * =====
125
126 * Propagation Delay - Input A
127 .meas tran tpHL_A_45 TRIG V(a45) VAL=0.5 RISE=1 TARG V(Y45) VAL=0.5
   FALL=1
128 .meas tran tpLH_A_45 TRIG V(a45) VAL=0.5 FALL=1 TARG V(Y45) VAL=0.5
   RISE=1
129
130 * Propagation Delay - Input B
131 .meas tran tpHL_B_45 TRIG V(b45) VAL=0.5 RISE=1 TARG V(Y45) VAL=0.5
   FALL=1
132 .meas tran tpLH_B_45 TRIG V(b45) VAL=0.5 FALL=1 TARG V(Y45) VAL=0.5
   RISE=1
133
134 * Propagation Delay - Input C

```

```

135 .meas tran tpHL_C_45 TRIG V(c45) VAL=0.5 RISE=1 TARG V(Y45) VAL=0.5
     FALL=1
136 .meas tran tpLH_C_45 TRIG V(c45) VAL=0.5 FALL=1 TARG V(Y45) VAL=0.5
     RISE=1
137
138 * Contamination Delay - Input A
139 .meas tran tcdHL_A_45 TRIG V(a45) VAL=0.5 RISE=1 TARG V(Y45) VAL=0.9
     FALL=1
140 .meas tran tcdLH_A_45 TRIG V(a45) VAL=0.5 FALL=1 TARG V(Y45) VAL=0.1
     RISE=1
141
142 * Contamination Delay - Input B
143 .meas tran tcdHL_B_45 TRIG V(b45) VAL=0.5 RISE=1 TARG V(Y45) VAL=0.9
     FALL=1
144 .meas tran tcdLH_B_45 TRIG V(b45) VAL=0.5 FALL=1 TARG V(Y45) VAL=0.1
     RISE=1
145
146 * Contamination Delay - Input C
147 .meas tran tcdHL_C_45 TRIG V(c45) VAL=0.5 RISE=1 TARG V(Y45) VAL=0.9
     FALL=1
148 .meas tran tcdLH_C_45 TRIG V(c45) VAL=0.5 FALL=1 TARG V(Y45) VAL=0.1
     RISE=1
149
150 .end

```

Schematic

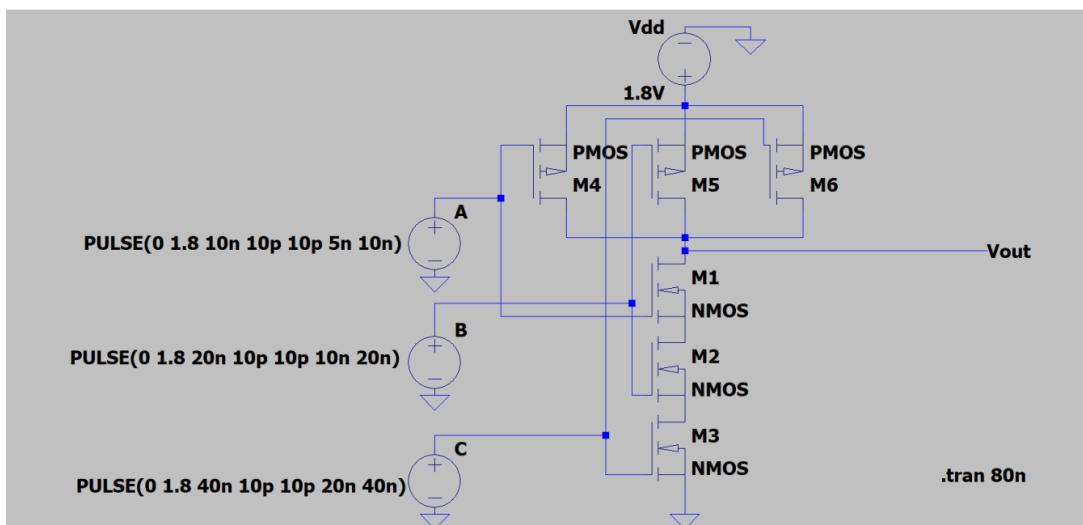


Figure 1.1: Schematic of 3 input NAND

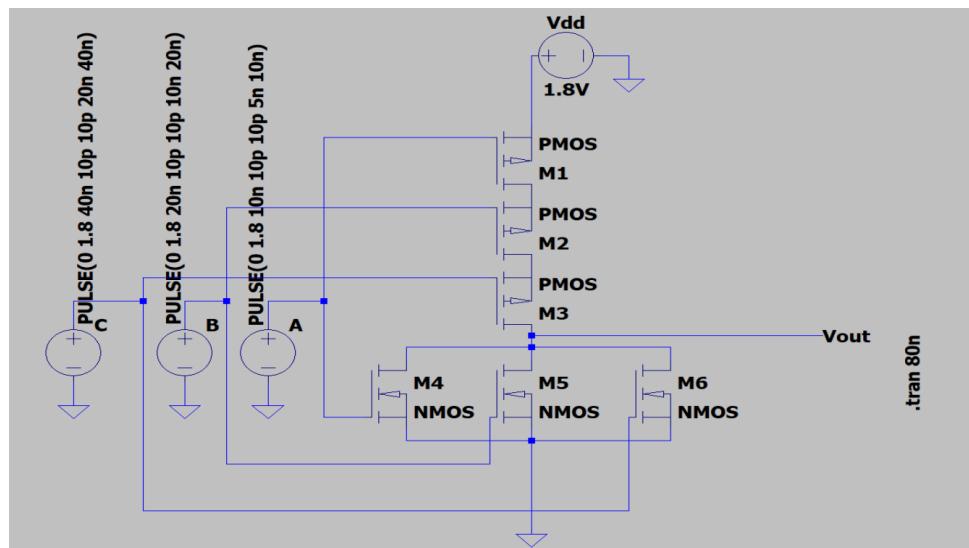


Figure 1.2: Schematic of 3 input NOR

1.4 Results

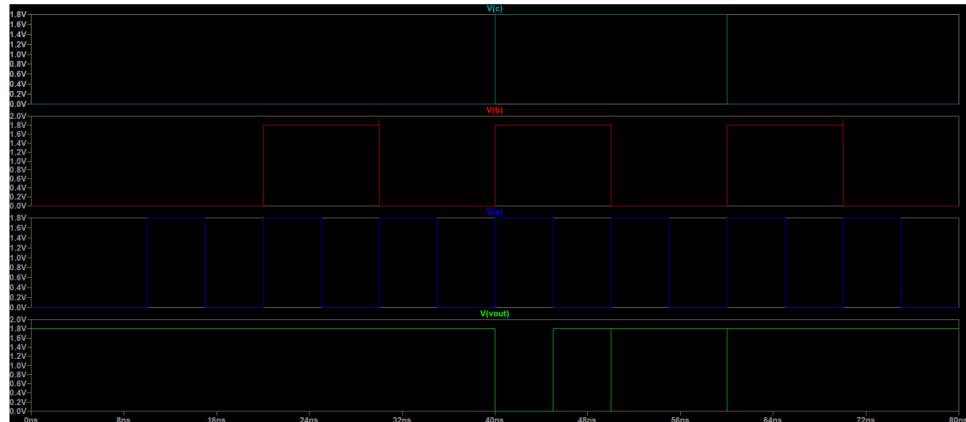


Figure 1.3: Wave Form of 3-Input NAND

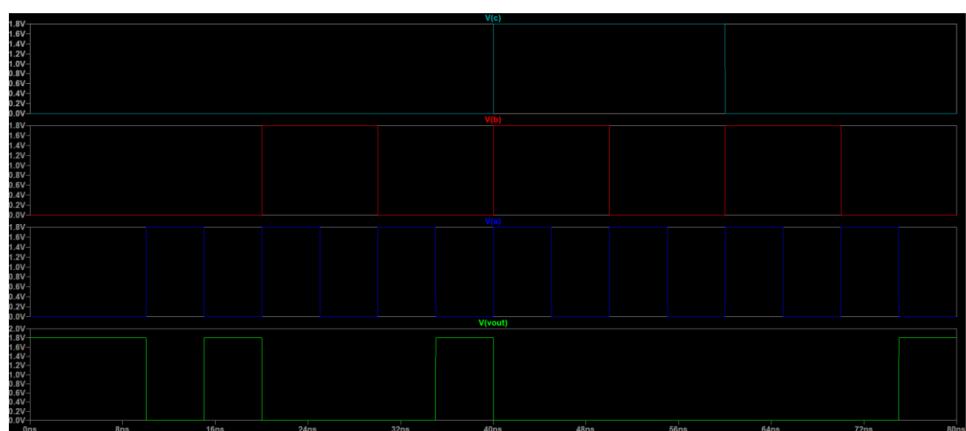


Figure 1.4: Wave Form of 3-Input NOR

```
SPICE Output Log: C:\Users\hp\Documents\LTspice\3_ip_NAND.log
LTspice 24.0.12 for Windows
Circuit: * C:\Users\hp\Documents\LTspice\3_ip_NAND.asc
Start Time: Sun Sep 14 17:36:46 2025
solver = Normal
Maximum thread count: 8
tnom = 27
temp = 27
method = modified trap
Direct Newton iteration for .op point succeeded.

tph1_a=2.00008e-08 FROM 1.0005e-08 TO 3.00058e-08
tph1_a=-7.6945e-13 FROM 3.0015e-08 TO 3.00142e-08
tph1_b=1.00008e-08 FROM 2.0005e-08 TO 3.00058e-08
tph1_b=-1.00008e-08 FROM 4.0015e-08 TO 3.00142e-08
tph1_c=7.64118e-13 FROM 3.0005e-08 TO 3.00058e-08
tph1_c=-2.00008e-08 FROM 5.0015e-08 TO 3.00142e-08
tcdhl_a=1.99975e-08 FROM 1.0005e-08 TO 3.00025e-08
tcdhl_a=-2.96454e-12 FROM 3.0015e-08 TO 3.0012e-08
tcdhl_b=9.99755e-09 FROM 2.0005e-08 TO 3.00025e-08
tcdhl_b=-1.0003e-08 FROM 4.0015e-08 TO 3.0012e-08
tcdhl_c=-2.45255e-12 FROM 3.0005e-08 TO 3.00025e-08
tcdhl_c=-2.0003e-08 FROM 5.0015e-08 TO 3.0012e-08

Total elapsed time: 0.982 seconds.
```

Figure 1.5: Delay of 3-Input NAND

```
SPICE Output Log: C:\Users\hp\Documents\LTspice\3_ip_NOR.log
LTspice 24.0.12 for Windows
Circuit: * C:\Users\hp\Documents\LTspice\3_ip_NOR.asc
Start Time: Sun Sep 14 18:08:44 2025
solver = Normal
Maximum thread count: 8
tnom = 27
temp = 27
method = modified trap
Direct Newton iteration for .op point succeeded.

tph1_a=-5.19878e-13 FROM 1.0005e-08 TO 1.00045e-08
tph1_a=5.86984e-13 FROM 2.0015e-08 TO 2.00156e-08
tph1_b=-2.00005e-08 FROM 3.0005e-08 TO 1.00045e-08
tph1_b=-1.99994e-08 FROM 4.0015e-08 TO 2.00156e-08
tph1_c=-4.00005e-08 FROM 5.0005e-08 TO 1.00045e-08
tph1_c=-3.99994e-08 FROM 6.0015e-08 TO 2.00156e-08
tcdhl_a=-3.12244e-12 FROM 1.0005e-08 TO 1.00019e-08
tcdhl_a=-2.51528e-12 FROM 2.0015e-08 TO 2.00125e-08
tcdhl_b=-2.00031e-08 FROM 3.0005e-08 TO 1.00019e-08
tcdhl_b=-2.00025e-08 FROM 4.0015e-08 TO 2.00125e-08
tcdhl_c=-4.00031e-08 FROM 5.0005e-08 TO 1.00019e-08
tcdhl_c=-4.00025e-08 FROM 6.0015e-08 TO 2.00125e-08

Total elapsed time: 1.110 seconds.
```

Figure 1.6: Delay of 3-Input NOR

The following tables show extracted rise/fall propagation and contamination delays for both gates under two example technology nodes.

Table 1.1: Comparison of Delays for Different Technology Nodes for 3-*Input NAND*

Parameter	0.35 μ m	0.18 μ m	45nm
Rise Contamination Delay (ns)	0.0103	0.00595	0.00515
Rise Propagation Delay (ns)	10.029	30.0558	20.0428
Fall Propagation Delay (ns)	11.2947	30.0562	10.0248
Fall Contamination Delay (ns)	1.00209	10.006	0.00901

Table 1.2: Comparison of Delays for Different Technology for 3-Input NOR

Parameter	0.35μm	0.18μm	45nm
Rise Contamination Delay (ns)	0.158	0.1118	0.0061
Rise Propagation Delay (ns)	1.987	9.993	9.998
Fall Propagation Delay (ns)	9.780	9.999	9.998
Fall Contamination Delay (ns)	0.0372	0.0334	0.0077

1.5 Conclusion

- The extracted timing parameters show that the delays of 3-input NAND and 3-input NOR gates are strongly influenced by the internal transistor arrangement. In the NAND gate, the stacked NMOS devices in the pull-down network dominate the delay, whereas in the NOR gate the series PMOS devices in the pull-up network have the major impact.
- Comparison of different technology nodes confirms that the process parameters have a significant effect on both propagation and contamination delays. As the feature size is reduced, parasitic capacitances decrease and drive strengths generally improve, resulting in shorter delays.
- Among the considered technologies, the smaller-geometry node consistently shows lower rise and fall delays for both NAND and NOR structures, which is aligned with expectations from CMOS scaling theory.
- Overall, the experiment validates the theoretical dependence of multi-input CMOS gate delay on both circuit topology and technology parameters, and demonstrates how process scaling can be exploited to obtain higher-speed digital logic.

Experiment 2

Propagation and Contamination Delay in a Pipeline

2.1 Aim

1. A unit CMOS inverter driving a unit NAND2 gate, which in turn is driving a CMOS inverter scaled up by alpha = 64.
2. In the above pipeline, the NAND2 gate is replaced by NOR2 gate.

2.2 Theory

Logical effort provides a compact analytical method to estimate the delay and optimal sizing of multi-stage CMOS logic paths. Each gate is modelled as a stage that incurs a certain logical and electrical “effort” in driving the next stage.

For an individual stage i , the normalized delay is written as

$$d_i = g_i h_i + p_i$$

where:

- g_i is the logical effort of stage i , representing how much worse the gate is at driving a load compared to a reference inverter,
- $h_i = \frac{C_{out,i}}{C_{in,i}}$ is the electrical effort, i.e., the ratio of the load capacitance to the input capacitance of the gate,
- p_i is the parasitic delay due to intrinsic capacitances of the gate.

For a path of N stages, the overall (path) effort is

$$F = G \cdot H \quad \text{with} \quad G = \prod_{i=1}^N g_i, \quad H = \frac{C_{out}}{C_{in}},$$

where G is the product of logical efforts and H is the total electrical effort from the input of the first stage to the final load.

The minimum-delay solution is obtained when each stage in the path has the same effort

$$f = F^{1/N}.$$

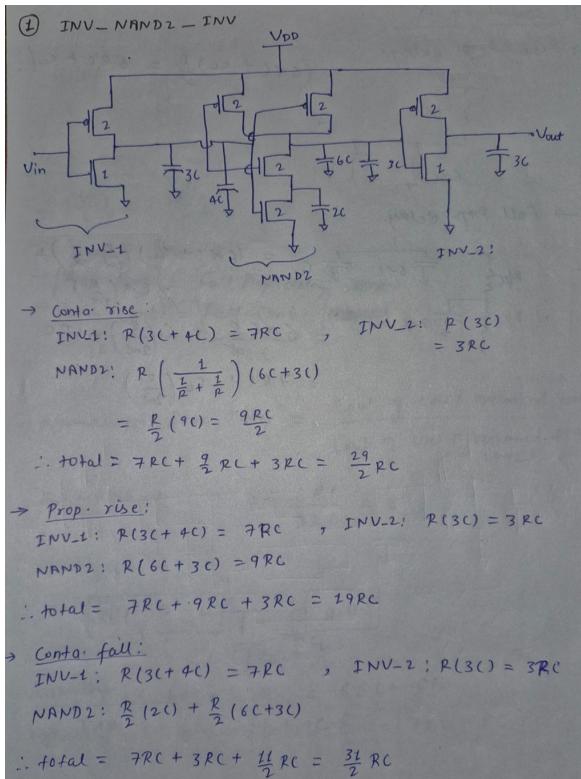
In this optimum case, the electrical effort of each stage becomes

$$h_i = \frac{f}{g_i},$$

and the total path delay (in units of τ) is

$$D = \sum_{i=1}^N d_i = \sum_{i=1}^N (g_i h_i + p_i) = N f + \sum_{i=1}^N p_i.$$

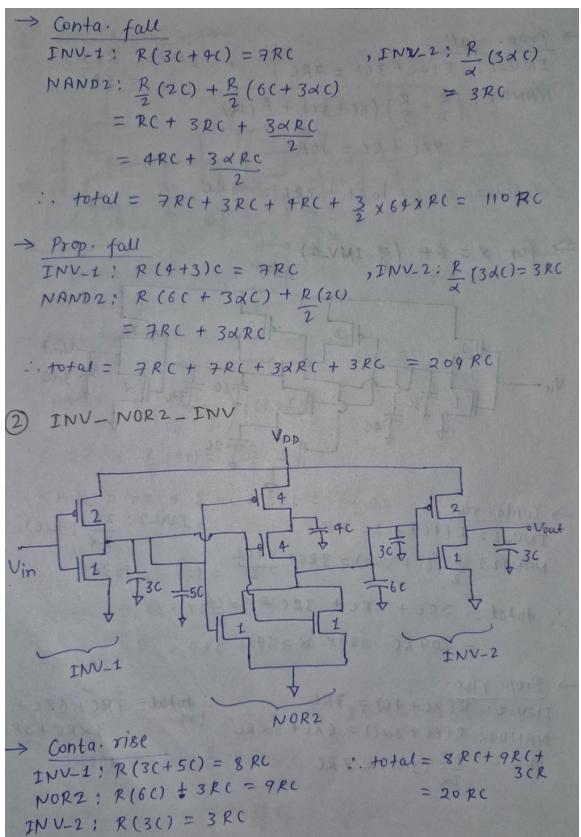
2.3 On Paper Analysis



\rightarrow Prop. fall:
 INV1: $R(4C + 3C) = 7RC$, INV2: $R(3C) = 3RC$
 NAND2: $\left(\frac{R}{2} + \frac{R}{2}\right)(6C + 3C) + \frac{R}{2}(2C)$
 $= 9RC + RC = 10RC$
 \therefore total = $7RC + 10RC + 3RC = 20RC$

\Rightarrow for $\alpha = 64$ (of INV-2)
 \rightarrow Conta. rise:
 INV1: $R(4C + 3C) = 7RC$, INV2: $\frac{2R}{2x}(3xC) = 3RC$
 NAND2: $\frac{R}{2}(6C + 3xC) = 3RC + \frac{3xRC}{2}$
 \therefore total = $7RC + 3RC + 3RC + \frac{3xRC}{2} = 109RC$ ($\because \alpha = 64$)

\rightarrow Prop. rise:
 INV1: $R(3C + 4C) = 7RC$, total = $7RC + 6RC + 3xRC + 3RC$
 NAND2: $R(6C + 3xC) = 6RC + 3xRC$
 INV2: $\frac{2R}{2x}(3xC) = 3RC = 208RC$



\rightarrow Prop. rise:
 INV1: $R(3C + 5C) = 8RC$, INV2: $R(3C) = 3RC$
 NOR2: $\frac{2R}{4}(4C) + \left(\frac{2R}{4} + \frac{2R}{4}\right)(6C + 3C) = 2RC + R(9C) = 11RC$
 \therefore total prop. rise = $8RC + 11RC + 3RC = 22RC$

\rightarrow Conta. fall:
 INV1: $R(3C + 5C) = 8RC$, INV2: $R(3C) = 3RC$
 NOR2: $\left(\frac{R}{2}\right)(6C + 3C) = 9RC$
 \therefore total = $8RC + 3RC + 9RC = \frac{31}{2}RC$

\rightarrow Prop. fall:
 INV1: $R(3C + 5C) = 8RC$, total = $8RC + 9RC + 3RC = 20RC$
 NOR2: $R(6C + 3C) = 9RC$
 INV2: $R(3C) = 3RC$

→ Contam. rise:

$$\text{INV1: } 8RC$$

$$\text{NOR2: } R(6C + 3\alpha C) = 6RC + 3\alpha RC$$

$$\text{INV2: } \frac{2R}{2d} (3\alpha C) = 3RC$$

$$\therefore \text{total} = 8RC + 6RC + 3RC = 17RC$$

→ Prop. rise:

$$\text{INV1: } 8RC$$

$$\text{NOR2: } \frac{2R}{2d} (4C) + \left(\frac{2R}{2d} + \frac{2R}{2d}\right)(6C + 3\alpha C) = 8RC + 3RC$$

$$\text{INV2: } 3RC$$

$$\therefore \text{total} = 8RC + 8RC + 3RC + 3RC = 21RC$$

→ Contam. fall:

$$\text{INV1: } 8RC$$

$$\text{INV2: } 3RC$$

$$\text{NOR2: } \frac{R}{2} (6C + 3\alpha C) = \frac{3RC}{2}$$

$$\therefore \text{total} = 8RC + 3RC + 3RC + \frac{3RC}{2} = 14RC + \frac{3RC}{2} = 110RC$$

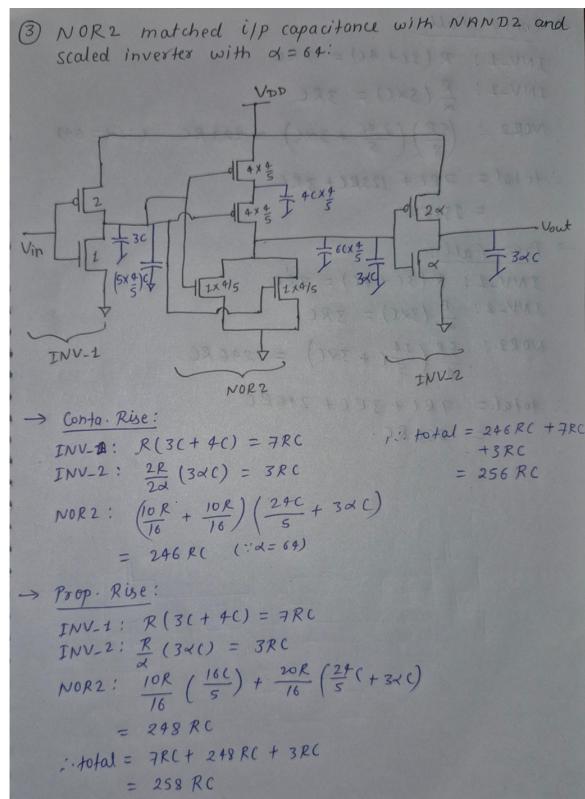
→ Prop. fall:

$$\text{INV1: } 8RC$$

$$\text{INV2: } 3RC$$

$$\text{NOR2: } R(6C + 3\alpha C) = 6RC + 3RC$$

$$\therefore \text{total} = 17RC + 3(64)RC = 209RC$$



→ Contam. Fall:

$$\text{INV1: } R(3C + 4C) = 7RC$$

$$\text{INV2: } \frac{R}{2} (3\alpha C) = 3RC$$

$$\text{NOR2: } \frac{(5R)}{8} \left(\frac{24C}{5} + 3\alpha C\right) = 123RC \quad (\because \alpha = 64)$$

$$\therefore \text{total} = 7RC + 123RC + 3RC = 133RC$$

→ Prop. Fall:

$$\text{INV1: } R(3C + 4C) = 7RC$$

$$\text{INV2: } \frac{R}{2} (3\alpha C) = 3RC$$

$$\text{NOR2: } \frac{5R}{4} \left(\frac{24}{5}C + 3\alpha C\right) = 246RC$$

$$\therefore \text{total} = 7RC + 3RC + 246RC = 256RC$$

Delay type / Circuit	INV_NAND2_I NV	INV_NAND2_I NV ($\alpha = 64$)	INV_NOR2_I NV	INV_NOR2_I NV ($\alpha = 64$)	INV_NOR2_I NV (I/P Cap. Matched with NAND2 and $\alpha = 64$)
Propagation Rise	19RC	208RC	22RC	211RC	258RC
Propagation Fall	20RC	209RC	20RC	209RC	256RC
Contamination Rise	14.5RC	109RC	20RC	209RC	256RC
Contamination Fall	15.5RC	110RC	15.5RC	110RC	133RC

On paper analysis table of Delays for all cases

2.4 Schematic

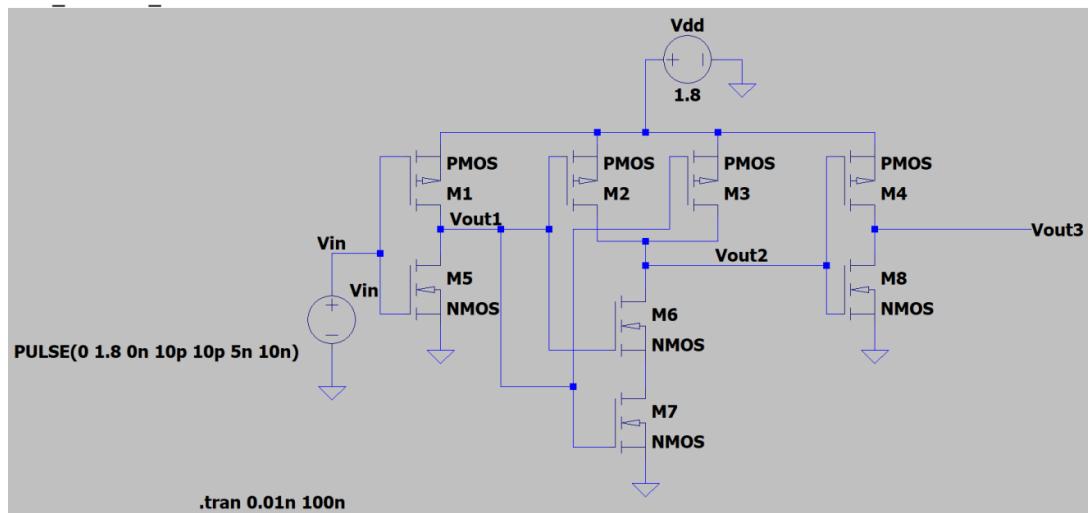


Figure 2.1: Schematic of INV_NAND2_INV

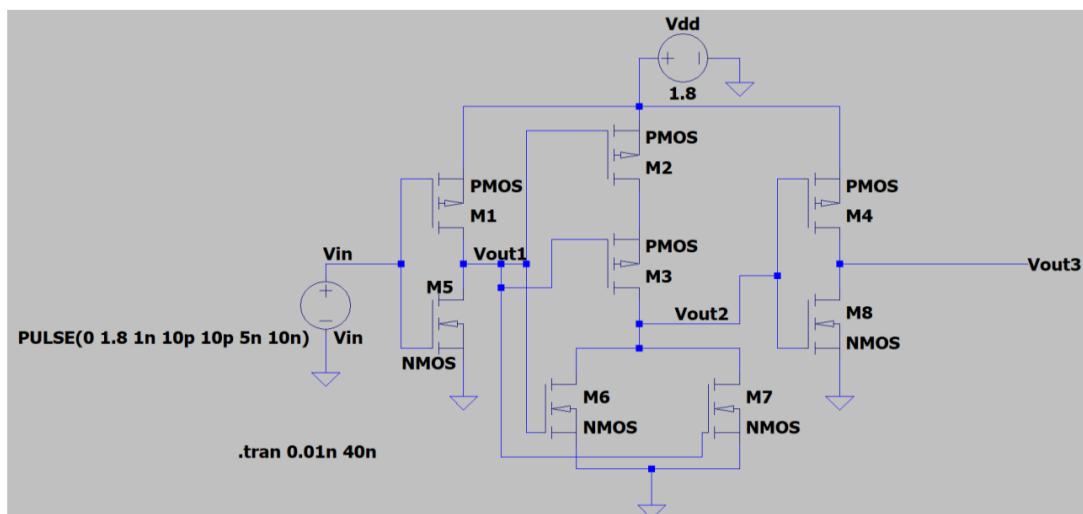


Figure 2.2: Schematic of INV_NAND2_INV

2.5 Results

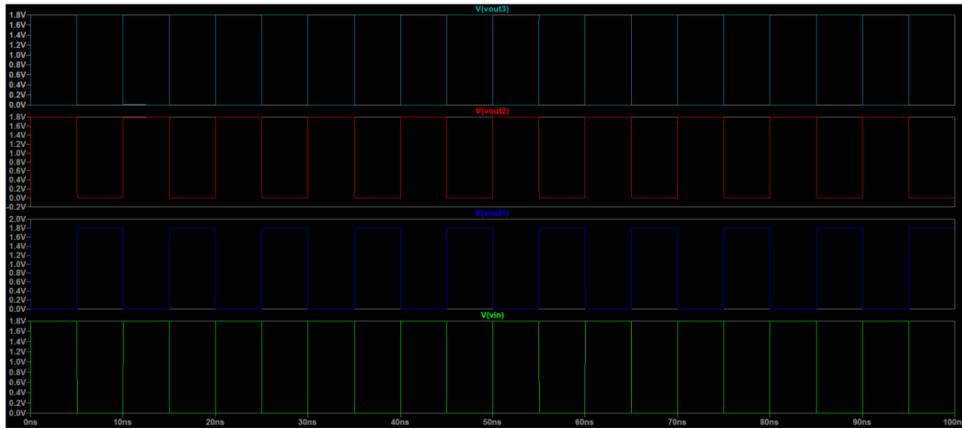


Figure 2.3: Waveform of INV_NAND2_INV

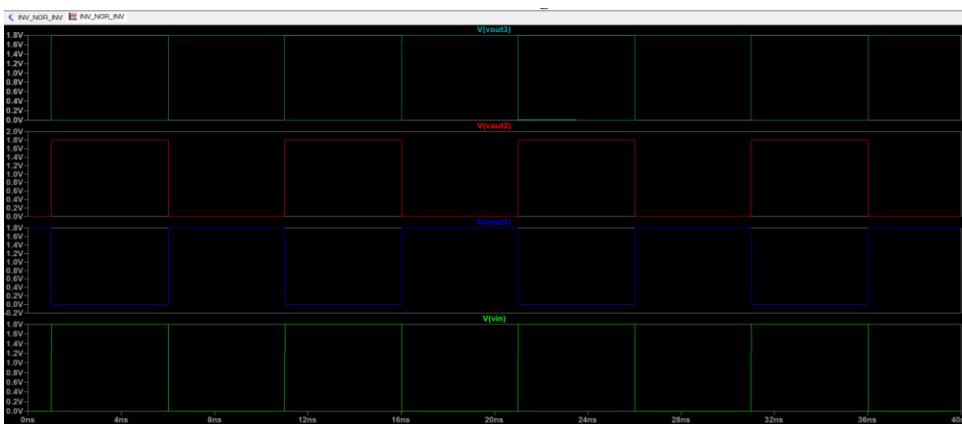


Figure 2.4: Waveform of INV_NAND2_INV

SPICE Output Log: C:\Users\hp\Documents\Ltspice\INV_NAND2.INV.log

```
t_in_rise1=9.9995e-08 FROM 5e-12 TO 1e-07
t_out_rise1=5.01494e-09 FROM 0 TO 5.01494e-09
tpd_rise1: (t_out_rise1-t_in_rise1)=-9.49801e-08
t_in_fall1=9.4985e-08 FROM 5.015e-09 TO 1e-07
t_out_fall1=5.00587e-12 FROM 0 TO 5.00587e-12
tpd_fall1: (t_out_fall1-t_in_fall1)=-9.498e-08
t_out_cdr1=5.01494e-09 FROM 0 TO 5.01494e-09
tcd_rise1: (t_out_cdr1-t_in_rise1)=-9.49801e-08
t_out_cdf1=5.00587e-12 FROM 0 TO 5.00587e-12

Measurement "tcd_fall1" FAIL'd
t_in_rise2=9.9995e-08 FROM 5e-12 TO 1e-07
t_out_rise2=4.79837e-12 FROM 0 TO 4.79837e-12
tpd_rise2: (t_out_rise2-t_in_rise2)=-9.99902e-08
t_in_fall2=9.4985e-08 FROM 5.015e-09 TO 1e-07
t_out_fall2=5.01519e-09 FROM 0 TO 5.01519e-09
tpd_fall2: (t_out_fall2-t_in_fall2)=-8.99698e-08
t_out_cdr2=4.79837e-12 FROM 0 TO 4.79837e-12
tcd_rise2: (t_out_cdr2-t_in_rise2)=-9.99902e-08
t_out_cdf2=5.01519e-09 FROM 0 TO 5.01519e-09
tcd_fall2: (t_out_cdf2-t_in_fall2)=-8.99698e-08
t_in_rise3=9.9995e-08 FROM 5e-12 TO 1e-07
t_out_rise3=5.01517e-09 FROM 0 TO 5.01517e-09
tpd_rise3: (t_out_rise3-t_in_rise3)=-9.49798e-08
t_in_fall3=9.4985e-08 FROM 5.015e-09 TO 1e-07
t_out_fall3=4.81679e-12 FROM 0 TO 4.81679e-12
tpd_fall3: (t_out_fall3-t_in_fall3)=-9.49802e-08
t_out_cdr3=5.01517e-09 FROM 0 TO 5.01517e-09
tcd_rise3: (t_out_cdr3-t_in_rise3)=-9.49798e-08
t_out_cdf3=4.81679e-12 FROM 0 TO 4.81679e-12
tcd_fall3: (t_out_cdf3-t_in_fall3)=-9.49802e-08
```

Figure 2.5: Delay of INV_NAND2_INV

```
[SPICE Output Log: C:\Users\hp\Documents\Ltspice\INV_NAND2_INV.log]
Direct Newton iteration for .op point succeeded.

t_in_risel=9.9995e-08 FROM 5e-12 TO 1e-07
t_out_risel=5.01494e-09 FROM 0 TO 5.01494e-09
tpd_risel: (t_out_risel-t_in_risel)=-9.49801e-08
t_in_fall1=9.4985e-08 FROM 5.015e-09 TO 1e-07
t_out_fall1=5.00587e-12 FROM 0 TO 5.00587e-12
tpd_fall1: (t_out_fall1-t_in_fall1)=-9.498e-08
t_out_cdr1=5.01494e-09 FROM 0 TO 5.01494e-09
tcd_risel: (t_out_cdr1-t_in_risel)=-9.49801e-08
t_out_cdf1=5.00587e-12 FROM 0 TO 5.00587e-12

Measurement "tcd_fall1" FAIL'ed
t_in_rise2=9.9995e-08 FROM 5e-12 TO 1e-07
t_out_rise2=4.79836e-12 FROM 0 TO 4.79836e-12
tpd_rise2: (t_out_rise2-t_in_rise2)=-9.99902e-08
t_in_fall2=9.4985e-08 FROM 5.015e-09 TO 1e-07
t_out_fall2=5.01519e-09 FROM 0 TO 5.01519e-09
tpd_fall2: (t_out_fall2-t_in_fall2)=-8.99698e-08
t_out_cdr2=4.79836e-12 FROM 0 TO 4.79836e-12
tcd_rise2: (t_out_cdr2-t_in_rise2)=-9.99902e-08
t_out_cdf2=5.01519e-09 FROM 0 TO 5.01519e-09
tcd_fall2: (t_out_cdf2-t_in_fall2)=-8.99698e-08
t_in_rise3=9.9995e-08 FROM 5e-12 TO 1e-07
t_out_rise3=5.01517e-09 FROM 0 TO 5.01517e-09
tpd_rise3: (t_out_rise3-t_in_rise3)=-9.49798e-08
t_in_fall3=9.4985e-08 FROM 5.015e-09 TO 1e-07
t_out_fall3=4.81678e-12 FROM 0 TO 4.81678e-12
tpd_fall3: (t_out_fall3-t_in_fall3)=-9.49802e-08
t_out_cdr3=5.01517e-09 FROM 0 TO 5.01517e-09
tcd_rise3: (t_out_cdr3-t_in_rise3)=-9.49798e-08
t_out_cdf3=4.81678e-12 FROM 0 TO 4.81678e-12
tcd_fall3: (t_out_cdf3-t_in_fall3)=-9.49802e-08

Total elapsed time: 0.291 seconds.
```

Figure 2.6: Delay of INV_NAND2_INV (for alpha = 64 of INV_2)

```
[SPICE Output Log: C:\Users\hp\Documents\Ltspice\INV_NOR_INV.log]
t_in_risel=3.8995e-08 FROM 1.005e-09 TO 4e-08
t_out_risel=6.015e-09 FROM 0 TO 6.015e-09
tpd_risel: (t_out_risel-t_in_risel)=-3.298e-08
t_in_fall1=3.3985e-08 FROM 6.015e-09 TO 4e-08
t_out_fall1=1.00499e-09 FROM 0 TO 1.00499e-09
tpd_fall1: (t_out_fall1-t_in_fall1)=-3.298e-08
t_out_cdr1=6.015e-09 FROM 0 TO 6.015e-09
tcd_risel: (t_out_cdr1-t_in_risel)=-3.298e-08
t_out_cdf1=1.00499e-09 FROM 0 TO 1.00499e-09

Measurement "tcd_fall1" FAIL'ed
t_in_rise2=3.8995e-08 FROM 1.005e-09 TO 4e-08
t_out_rise2=1.00519e-09 FROM 0 TO 1.00519e-09
tpd_rise2: (t_out_rise2-t_in_rise2)=-3.79898e-08
t_in_fall2=3.3985e-08 FROM 6.015e-09 TO 4e-08
t_out_fall2=6.01481e-09 FROM 0 TO 6.01481e-09
tpd_fall2: (t_out_fall2-t_in_fall2)=-2.79702e-08
t_out_cdr2=1.00519e-09 FROM 0 TO 1.00519e-09
tcd_rise2: (t_out_cdr2-t_in_rise2)=-3.79898e-08
t_out_cdf2=6.01481e-09 FROM 0 TO 6.01481e-09
tcd_fall2: (t_out_cdf2-t_in_fall2)=-2.79702e-08
t_in_rise3=3.8995e-08 FROM 1.005e-09 TO 4e-08
t_out_rise3=6.01473e-09 FROM 0 TO 6.01473e-09
tpd_rise3: (t_out_rise3-t_in_rise3)=-3.29803e-08
t_in_fall3=3.3985e-08 FROM 6.015e-09 TO 4e-08
t_out_fall3=1.00522e-09 FROM 0 TO 1.00522e-09
tpd_fall3: (t_out_fall3-t_in_fall3)=-3.29798e-08
t_out_cdr3=6.01473e-09 FROM 0 TO 6.01473e-09
tcd_rise3: (t_out_cdr3-t_in_rise3)=-3.29803e-08
t_out_cdf3=1.00522e-09 FROM 0 TO 1.00522e-09
tcd_fall3: (t_out_cdf3-t_in_fall3)=-3.29798e-08

Total elapsed time: 0.500 seconds.
```

Figure 2.7: Delay of INV_NAND2_INV

```

SPICE Output Log: C:\Users\hp\Documents\Ltspice\INV_NOR_INV.log
t_in_risel=3.8995e-08 FROM 1.005e-09 TO 4e-08
t_out_risel=6.015e-09 FROM 0 TO 6.015e-09
tpd_risel: (t_out_risel-t_in_risel)=-3.298e-08
t_in_fall1=3.3985e-08 FROM 6.015e-09 TO 4e-08
t_out_fall1=1.00499e-09 FROM 0 TO 1.00499e-09
tpd_fall1: (t_out_fall1-t_in_fall1)=-3.298e-08
t_out_cdr1=6.015e-09 FROM 0 TO 6.015e-09
tcd_risel: (t_out_cdr1-t_in_risel)=-3.298e-08
t_out_cdf1=1.00499e-09 FROM 0 TO 1.00499e-09

Measurement "tcd_fall1" FAIL'ed
t_in_rise2=3.8995e-08 FROM 1.005e-09 TO 4e-08
t_out_rise2=1.00519e-09 FROM 0 TO 1.00519e-09
tpd_rise2: (t_out_rise2-t_in_rise2)=-3.79898e-08
t_in_fall2=3.3985e-08 FROM 6.015e-09 TO 4e-08
t_out_fall2=6.01481e-09 FROM 0 TO 6.01481e-09
tpd_fall2: (t_out_fall2-t_in_fall2)=-3.79702e-08
t_out_cdr2=6.015e-09 FROM 0 TO 6.015e-09
tcd_rise2: (t_out_cdr2-t_in_rise2)=-3.79898e-08
t_out_cdf2=6.01481e-09 FROM 0 TO 6.01481e-09
tpd_fall2: (t_out_cdf2-t_in_fall2)=-2.79702e-08
t_in_rise3=3.8995e-08 FROM 1.005e-09 TO 4e-08
t_out_rise3=6.01473e-09 FROM 0 TO 6.01473e-09
tpd_rise3: (t_out_rise3-t_in_rise3)=-3.29803e-08
t_in_fall3=3.3985e-08 FROM 6.015e-09 TO 4e-08
t_out_fall3=1.00522e-09 FROM 0 TO 1.00522e-09
tpd_fall3: (t_out_fall3-t_in_fall3)=-3.29798e-08
t_out_cdr3=6.01473e-09 FROM 0 TO 6.01473e-09
tcd_rise3: (t_out_cdr3-t_in_rise3)=-3.29803e-08
t_out_cdf3=1.00522e-09 FROM 0 TO 1.00522e-09
tpd_fall3: (t_out_cdf3-t_in_fall3)=-3.29798e-08

Total elapsed time: 0.556 seconds.

```

Figure 2.8: Delay of INV_NAND2_INV (for alpha = 64 of INV_2)

2.6 Conclusion

- The logical-effort analysis shows that the path containing the NAND2 gate has a smaller total delay ($\approx 17.21 \tau$) than the path using the NOR2 gate ($\approx 18.23 \tau$). For the same overall fanout of $\alpha = 64$, the INV–NAND2–INV chain is therefore faster than the INV–NOR2–INV chain.
- This behaviour is mainly due to the lower logical effort of the NAND2 gate ($g = \frac{4}{3}$) compared to the NOR2 gate ($g = \frac{5}{3}$). Because the NOR2 is inherently less efficient at driving a load, the optimal stage effort f is higher and the total term Nf increases, leading to a larger overall delay.
- The parasitic delays used in the computation ($p_{inv} = 1$, $p_{NAND2} = 2$, $p_{NOR2} = 2$) are standard reference values. Different technology libraries may have slightly different parasitics, but the qualitative conclusion that NOR2-based paths are slower than NAND2-based paths typically remains valid.
- In practice, the analytical sizing from logical effort is combined with transistor-level simulations to verify timing under process, voltage and temperature variations, as well as to account for interconnect parasitics and realistic input slews.
- The main takeaway is that for a given fanout, placing gates with smaller logical effort—such as NAND2 rather than NOR2—on the critical path helps to minimize delay and improve performance.

Experiment 3

Schematic-Level Analysis of CMOS Inverter

3.1 Aim

To design, implement, and analyze a CMOS inverter using Cadence by performing schematic design, transient, DC characterization and calculating delay.

3.2 Theory

A CMOS inverter is realized using a complementary pair of MOS transistors: a PMOS device in the pull-up path and an NMOS device in the pull-down path. Both transistors share a common drain node, which forms the output, and have their gates connected to the same input signal.

In ideal DC operation:

- For $V_{in} = 0$: the PMOS is turned ON and the NMOS is OFF, so the output node is connected to V_{DD} , corresponding to logic ‘1’.
- For $V_{in} = V_{DD}$: the PMOS is OFF and the NMOS is ON, pulling the output to ground (logic ‘0’).

For intermediate input voltages, both transistors may conduct simultaneously. This region defines the inverter’s Voltage Transfer Characteristic (VTC), from which important parameters such as noise margins and the switching threshold can be derived. Around the switching point, the inverter has high gain and sharp transition, allowing it to regenerate and restore logic levels in a digital system.

Truth Table

V_{in}	V_{out}
0	1
1	0

Table 3.1: Truth table for CMOS inverter

Schematic

Standard CMOS inverter:

- PMOS: source → V_{DD} , drain → output, gate → input, body → V_{DD} .
- NMOS: source → GND, drain → output, gate → input, body → GND.

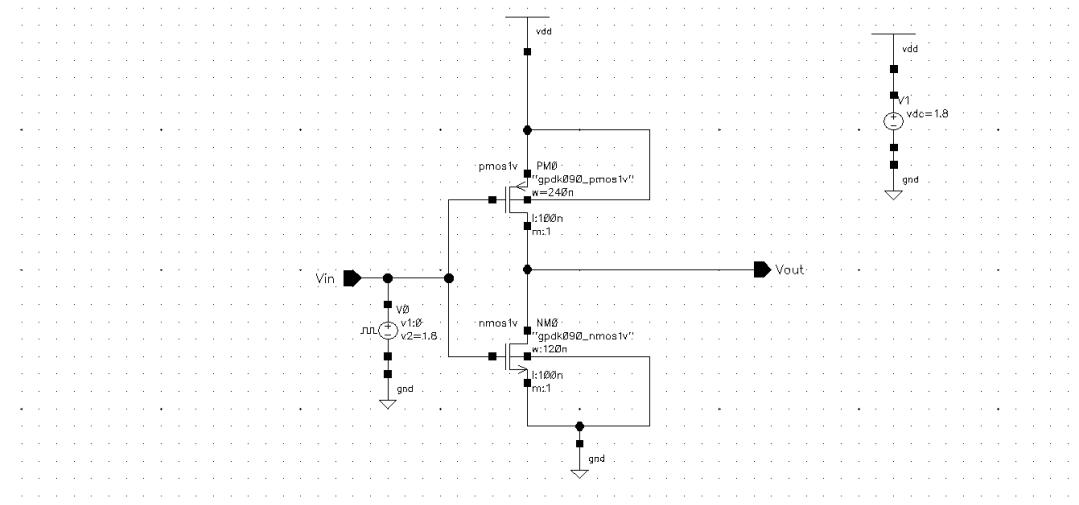


Figure 3.1: Schematic of Inverter

Symbol and Testbench

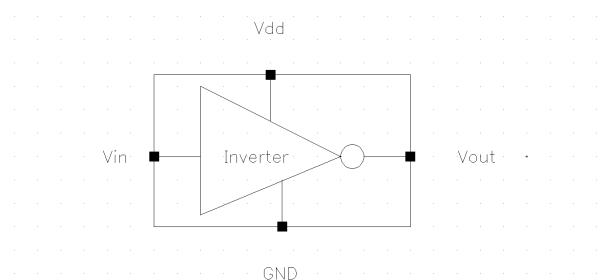


Figure 3.2: Symbol of Inverter

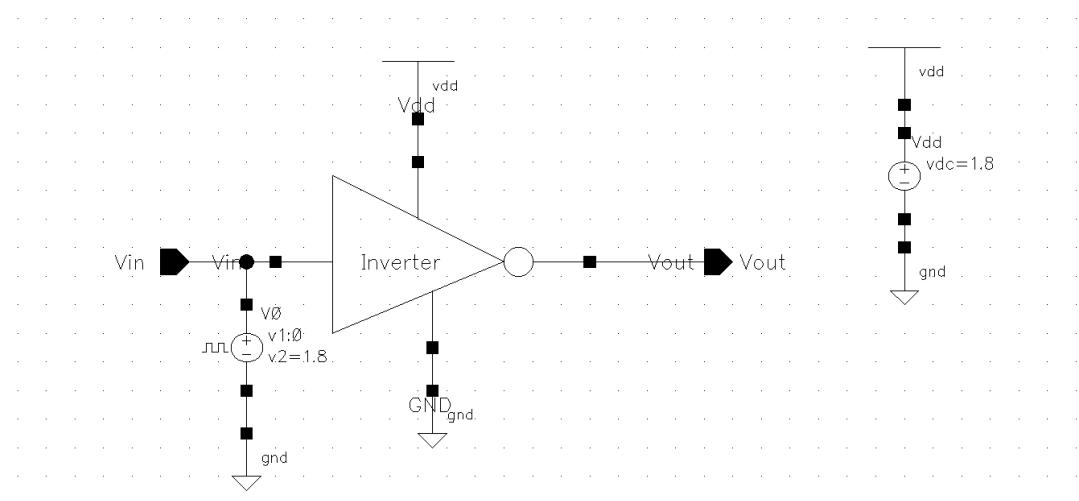


Figure 3.3: TestBench for Inverter

3.3 Results

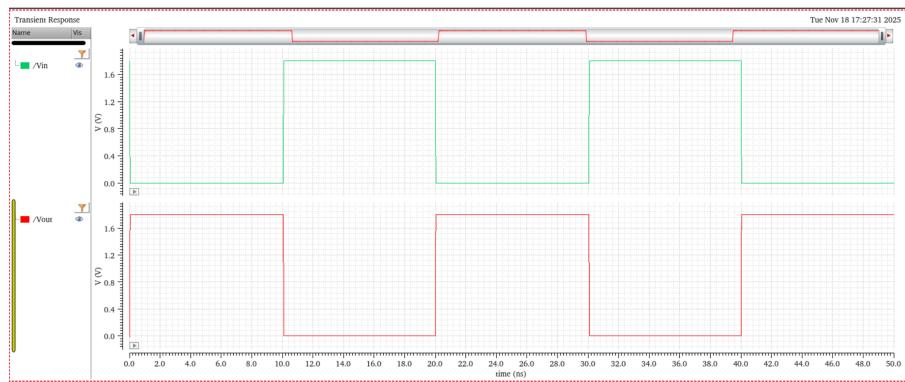


Figure 3.4: Transient Analysis of CMOS-Inverter

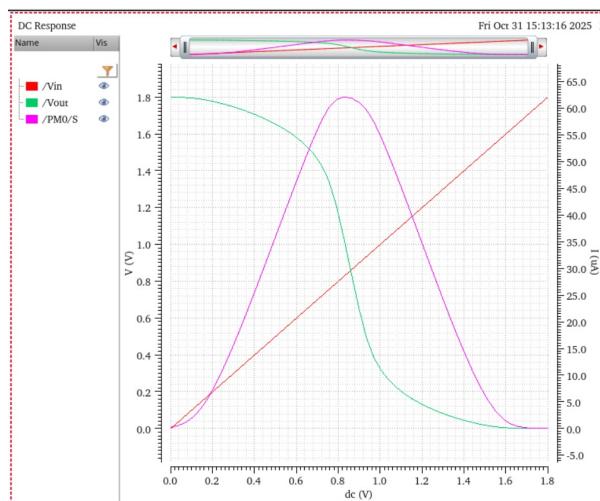
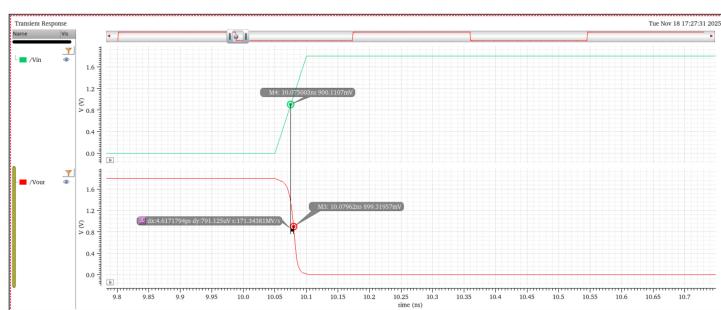
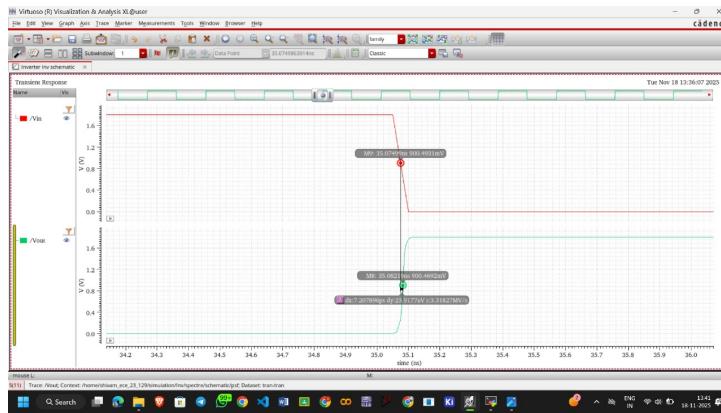


Figure 3.5: DC Analysis of CMOS-Inverter

Figure 3.6: Transient Response of CMOS-Inverter for t_{phl}

Figure 3.7: Transient Response of CMOS-Inverter for t_{PLH}

3.4 Conclusion

- The transient simulations confirm that the CMOS inverter performs the correct logical inversion: when the input is low, the output is driven high, and when the input goes high, the output transitions to low, consistent with the inverter truth table.
- From the measured waveform markers:
 - Input rising 50% crossing: $t_{in,rise50} = 10.075 \text{ ns}$
 - Output falling 50% crossing: $t_{out,fall50} = 10.0796 \text{ ns}$
 - Fall propagation delay:

$$t_{PHL} = t_2 - t_1 = 4.6171 \text{ ps}$$

- Input falling 50% crossing: $t_{in,fall50} = 20.02497 \text{ ns}$
- Output rising 50% crossing: $t_{out,rise50} = 20.03219 \text{ ns}$
- Rise propagation delay:

$$t_{PLH} = t_4 - t_3 = 7.2272 \text{ ps}$$

The resulting average propagation delay is:

$$t_{pd} = 5.92204 \text{ ps}$$

Table 3.2: Measured parameters (Schematic)

Parameter	Schematic
V_{DD}	1.8 V
t_{PHL}	4.6171 ps
t_{PLH}	7.227 ps
t_{pd}	5.92205 ps

- The DC analysis (VTC) exhibits the expected S-shaped curve: for $V_{in} < V_{IL}$ the output remains close to V_{DD} (logic 1), and for $V_{in} > V_{IH}$ the output is near 0 V (logic 0), demonstrating proper noise margins.
- The switching threshold V_M (where $V_{in} = V_{out}$) lies close to $V_{DD}/2$, indicating that the PMOS and NMOS devices are reasonably well sized, and that the inverter provides balanced high and low noise margins.

Experiment 4

Layout-Level Analysis of CMOS Inverter

4.1 Aim

To create the physical layout of a CMOS inverter, perform DRC/LVS checks, extract parasitic RC components, and run post-layout simulations.

4.2 Theory

At the layout level, a CMOS inverter is implemented using diffusion, poly and metal layers to form the PMOS and NMOS transistors and to route the input, output and power rails. In addition to the ideal transistor behaviour, physical layout introduces extra parasitic resistances and capacitances which directly affect circuit performance.

Key aspects of the layout include:

- proper creation of wells and well ties (n-well for PMOS and substrate contacts for NMOS),
- correct placement of diffusion and implant regions for source/drain,
- sufficient number of contacts and vias to minimize series resistance,
- careful routing of V_{DD} , GND and signal lines to limit interconnect RC.

After the layout is completed, Design Rule Check (DRC) enforces foundry rules for spacing, width, overlap, etc., while Layout Versus Schematic (LVS) ensures that the extracted devices and their connectivity match the original schematic. Parasitic extraction (PEX) then generates an RC-annotated netlist which captures wiring and junction parasitics. These parasitics tend to increase propagation delay and can influence noise margins and signal integrity compared to the ideal schematic-level behaviour.

4.3 DRC and LVS

1. Run DRC in the layout tool (Calibre or PDK DRC) and fix spacing, width, and enclosure violations.
2. Run LVS to ensure the netlist extracted from layout matches the schematic netlist. Resolve mismatches (pin names, nets, device orientations).
3. Re-run DRC/LVS until clean report is obtained.

4.4 Parasitic Extraction (PEX)

1. Use the PDK parasitic extraction tool to extract R and C for diffusion, poly, metal, and vias.
2. Generate an av extracted cell.

4.5 Results

4.5.1 Layout

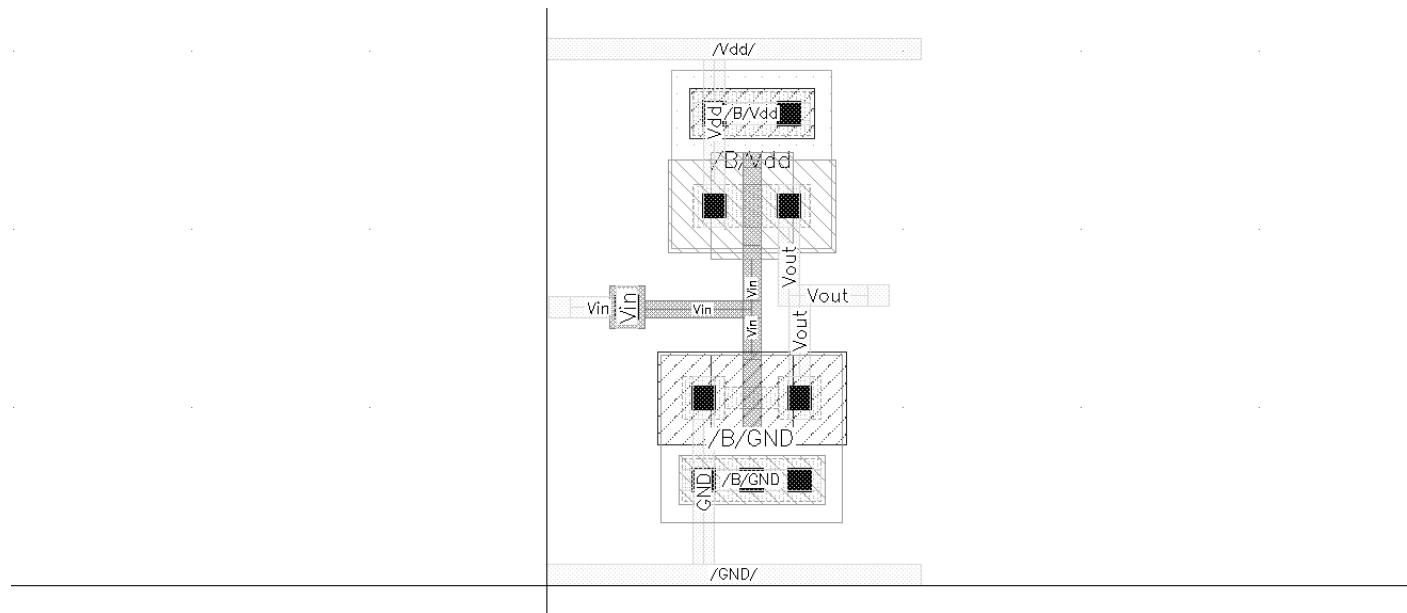


Figure 4.1: Layout of Inverter

4.5.2 RC Extraction

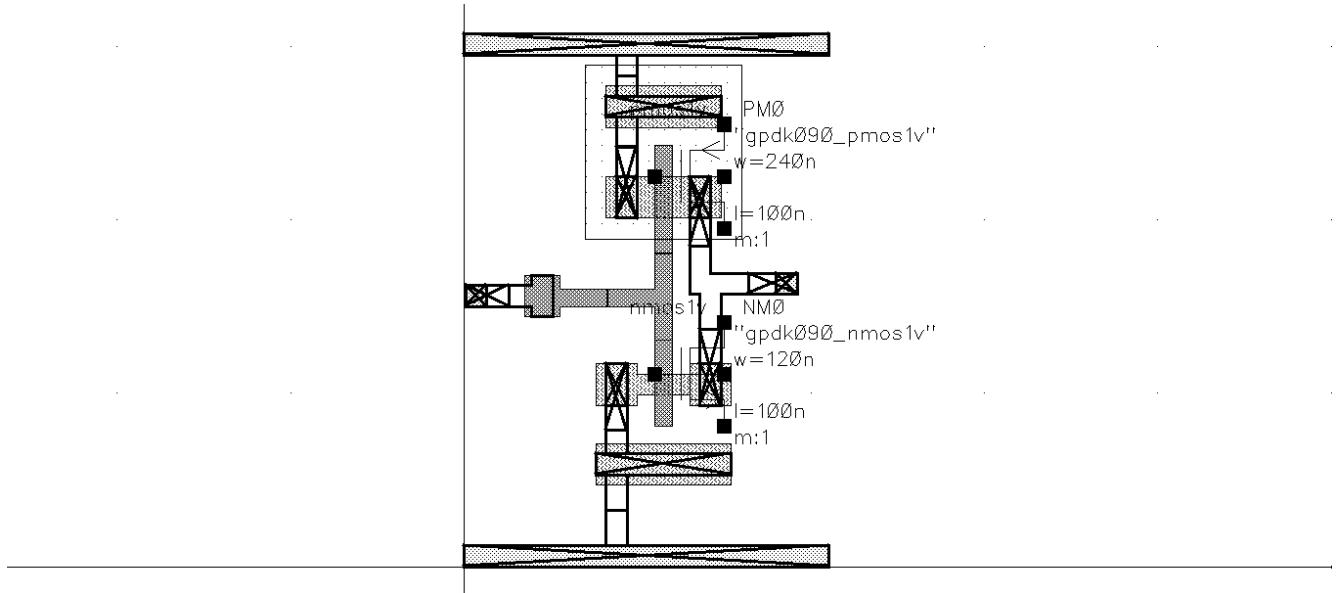


Figure 4.2: RC Extraction of CMOS-Inverter

4.6 Conclusion

- Post-layout transient simulations show that the inverter experiences an increase in propagation delay compared to the schematic-level results. This additional delay is caused by the parasitic resistances and capacitances associated with interconnects and device junctions introduced during layout.
- The main contributors to the delay degradation are:
 - series resistance in the metal and diffusion paths,
 - extra load capacitances from interconnect and overlap regions,
 - drain/source junction capacitances and via capacitances.
- In all cases, the post-layout delays are larger than the schematic-only delays, confirming that layout-dependent effects must be taken into account for accurate timing analysis.
- The experiment highlights the importance of performing DRC, LVS and parasitic extraction followed by post-layout simulation in a realistic VLSI design flow, in order to capture the impact of physical design on timing, noise margins and signal integrity.

Experiment 5

Schematic and Layout-Level Analysis of 2-input CMOS NAND

5.1 Aim

To design and analyze a 2-input CMOS NAND gate by performing transient simulation and propagation delay evaluation at the schematic level, followed by creating the physical layout, extracting parasitic RC components.

5.2 Theory

A 2-input CMOS NAND gate implements the boolean function:

$$Y = \overline{A \cdot B} \quad (5.1)$$

using complementary pull-up and pull-down transistor networks.

In static CMOS realization:

- The pull-up network (PUN) is formed by two PMOS transistors connected in **parallel** between V_{DD} and the output node Y , with gates driven by A and B .
- The pull-down network (PDN) is formed by two NMOS transistors connected in **series** between Y and GND.

Operating principle:

- When both A and B are HIGH, the series NMOS devices conduct and pull Y LOW, while both PMOS devices are OFF. Thus $Y = 0$.
- If any input is LOW, at least one PMOS is ON, providing a path from V_{DD} to Y . In this case, the series NMOS path is partially or fully off, and Y is driven HIGH.

Truth Table

A	B	Y = NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0

Table 5.1: Truth Table of NAND Gate

Schematic

Standard static CMOS NAND:

- Pull-up network: PMOS P1 (gate=A) and PMOS P2 (gate=B) connected in parallel between VDD and output Y (sources to VDD, drains to Y).
- Pull-down network: NMOS N1 (gate=A) and NMOS N2 (gate=B) connected in series between output Y and GND (drain of N1 to Y, source of N1 to drain of N2, source of N2 to GND).

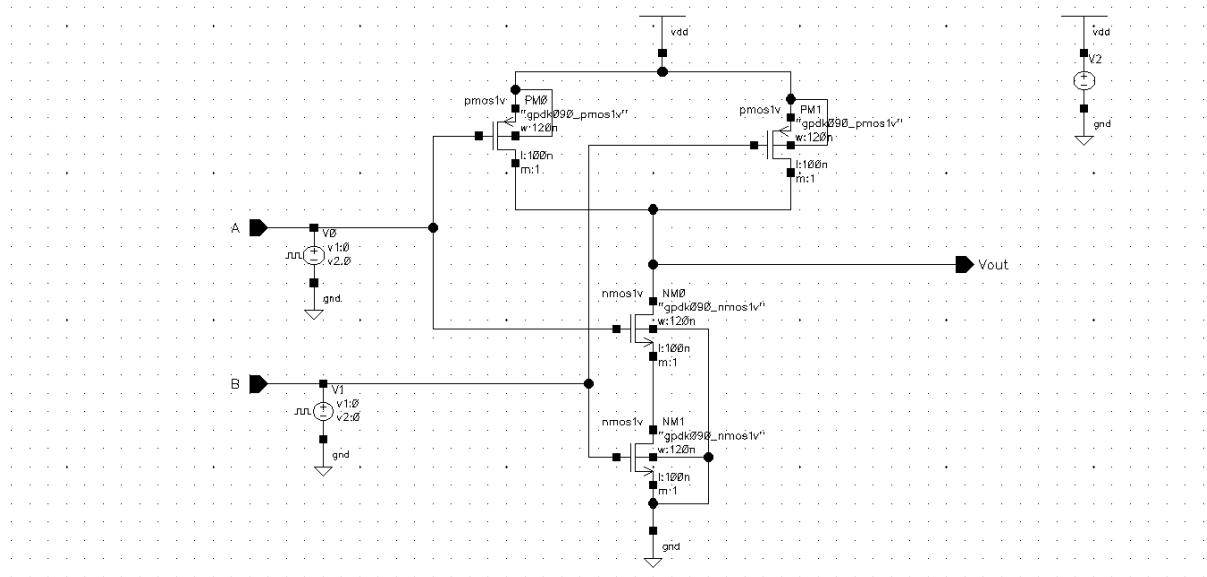


Figure 5.1: Schematic of 2-Input CMOS NAND

Symbol

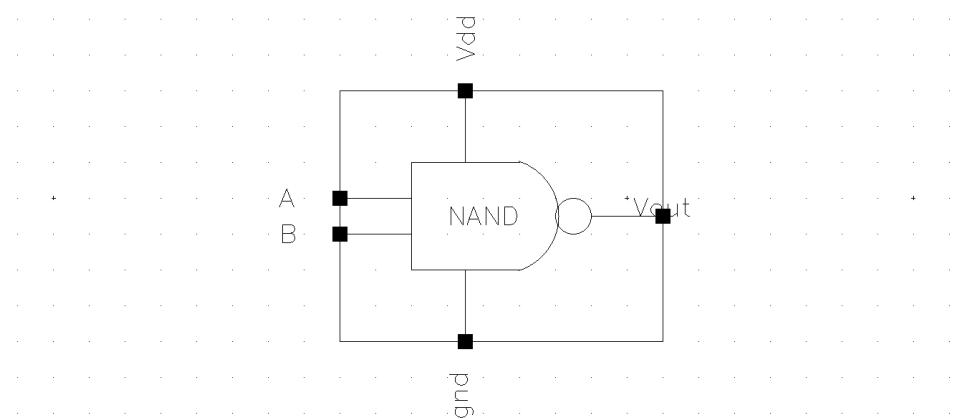


Figure 5.2: Symbol of NAND

TestBench

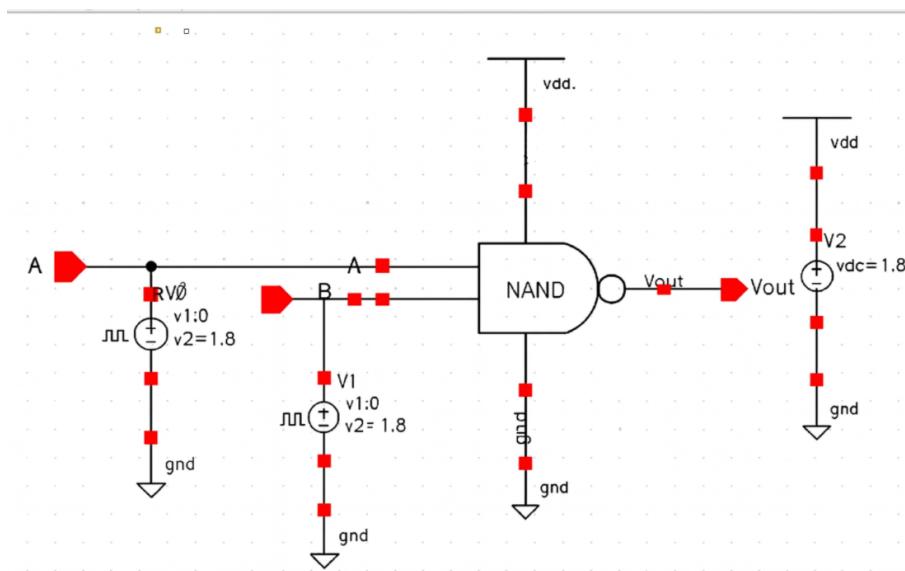


Figure 5.3: NAND Testbench

Layout

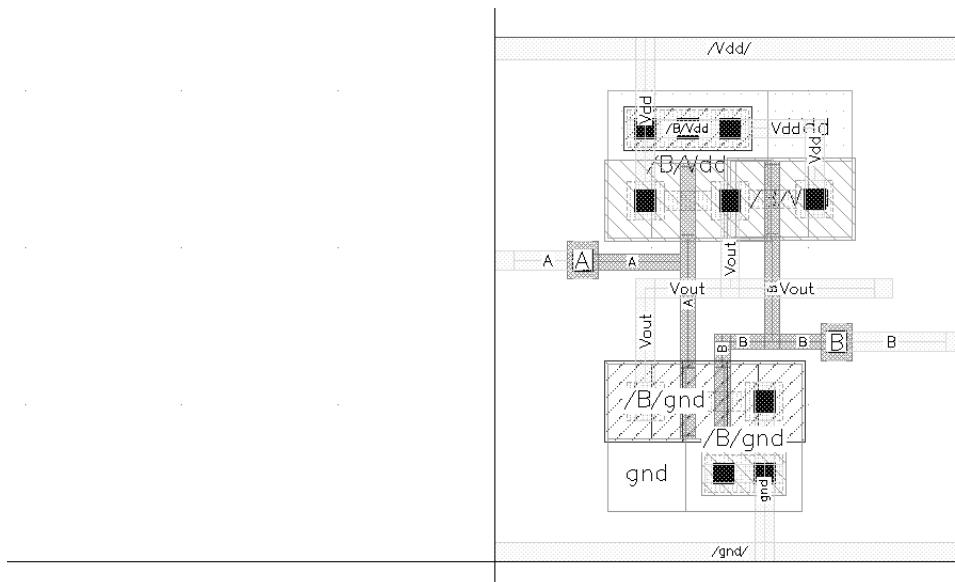


Figure 5.4: Layout of NAND

Parasitic Extraction

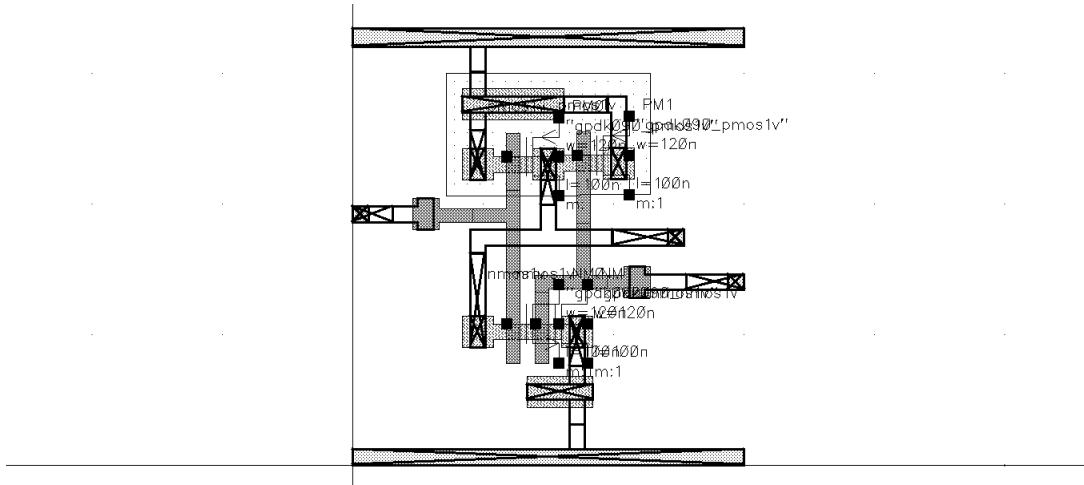


Figure 5.5: RC Extraction of NAND

5.3 Results

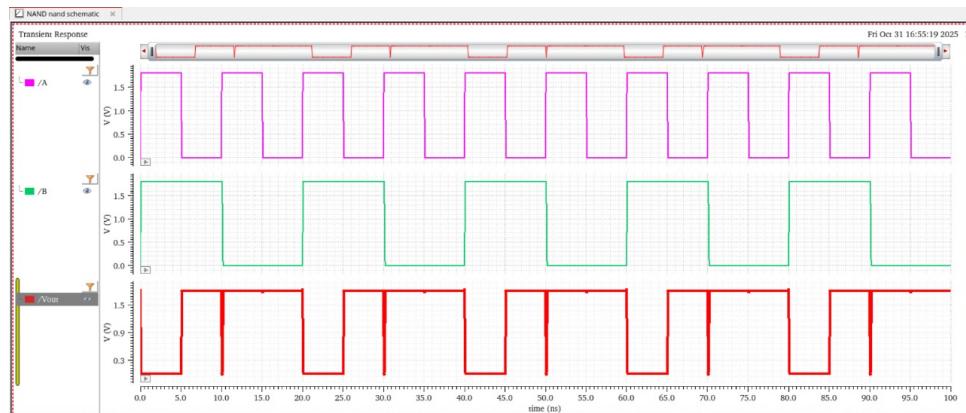


Figure 5.6: Transient Response of NAND

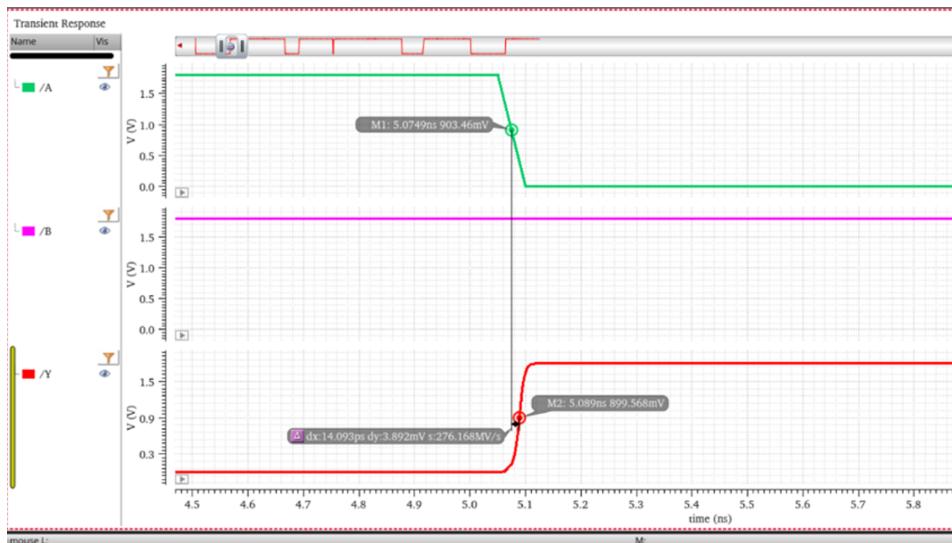
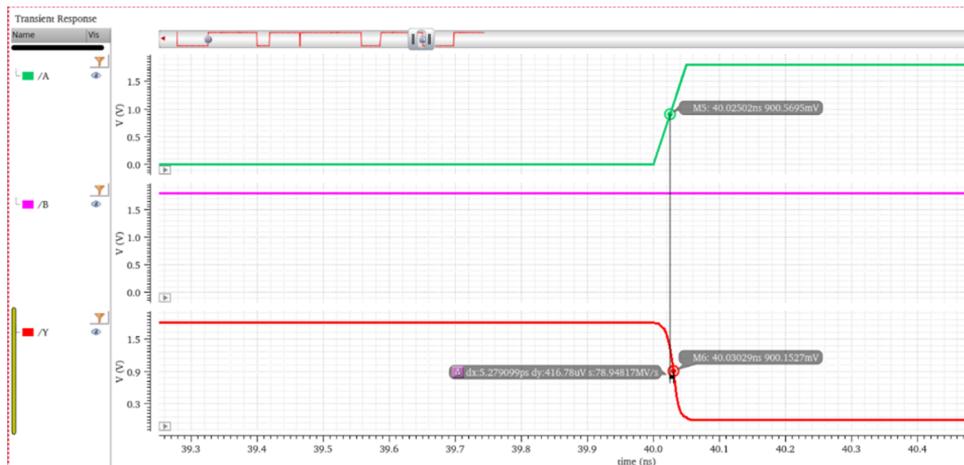
Figure 5.7: Transient Response of NAND for t_{PLH} Figure 5.8: Transient Response of NAND for t_{PHL}

Table 5.2: Propagation Delay Calculation for 2-Input CMOS NAND Gate

Transition	Measurement Points (ns)	Delay (ps)
t_{PLH} ($Y: 0 \rightarrow 1$)	IN fall = 5.0749, OUT rise = 5.089	14.0093
t_{PHL} ($Y: 1 \rightarrow 0$)	IN rise = 40.025, OUT fall = 40.030	5.279
Propagation delay	—	9.644 ps

5.4 Conclusion

- The simulated waveforms confirm the correct logical operation of the 2-input CMOS NAND gate: the output Y goes LOW only when both inputs A and B are HIGH, and remains HIGH for all other input combinations, in agreement with the truth table.

- The output transitions are clearly associated with the corresponding input edges. A rising edge on input A with $B = 1$ produces a HIGH-to-LOW transition at Y , while a falling edge on A with $B = 0$ results in a LOW-to-HIGH transition at the output.
- From the measured timing markers:
 1. Input falling 50% point: $t_{in,fall50} = 5.07495 \text{ ns}$
 2. Output rising 50% point: $t_{out,rise50} = 5.089 \text{ ns}$
 3. Rise propagation delay:

$$t_{PLH} = 5.089 - 5.07495 = 14.0093 \text{ ps}$$

4. Input rising 50% point: $t_{in,rise50} = 40.025 \text{ ns}$
5. Output falling 50% point: $t_{out,fall50} = 40.03029 \text{ ns}$
6. Fall propagation delay:

$$t_{PHL} = 40.030 - 40.025 = 5.279 \text{ ps}$$

- The overall propagation delay is obtained as the average:

$$t_{pd} = (t_{PHL}, t_{PLH}) = 9.644 \text{ ps},$$

indicating that the gate can switch on the order of a few picoseconds for the given loading conditions.

- Thus, both the logical and timing characteristics of the designed 2-input CMOS NAND gate are successfully verified through simulation.

Experiment 6

Schematic and Layout-Level Analysis of 2-input CMOS NOR

6.1 Aim

To design and analyze a 2-input CMOS NOR gate by performing transient simulation and propagation delay evaluation at the schematic level, followed by creating the physical layout, extracting parasitic RC components.

6.2 Theory

A 2-input CMOS NOR gate implements the boolean function:

$$Y = \overline{A + B} \quad (6.1)$$

using complementary pull-up and pull-down networks.

In static CMOS implementation:

- The pull-up network consists of two PMOS transistors connected in **series** between V_{DD} and the output node Y .
- The pull-down network consists of two NMOS transistors connected in **parallel** between Y and GND.

Functional behaviour:

- When both A and B are LOW, both NMOS devices are OFF and the series PMOS path conducts, pulling Y HIGH (logic 1).
- When either input is HIGH, at least one NMOS device turns ON and pulls the output LOW, while the corresponding PMOS in the series stack is turned OFF.

Because of the series PMOS stack, the pull-up path is weaker than in an inverter or NAND gate, which typically leads to a larger low-to-high propagation delay (t_{PLH}). The parallel NMOS network provides a stronger pull-down, often resulting in a smaller high-to-low delay (t_{PHL}). This difference in strengths is reflected in the measured timing characteristics of the NOR gate.

Schematic

Use standard CMOS structure:

- Pull-up: PMOS P1 (connected to A) and PMOS P2 (connected to B) in series between VDD and output Y and gates to inputs.

- Pull-down: NMOS N1 (connected to A) and NMOS N2 (connected to B) in parallel between Y and GND and gates to input.

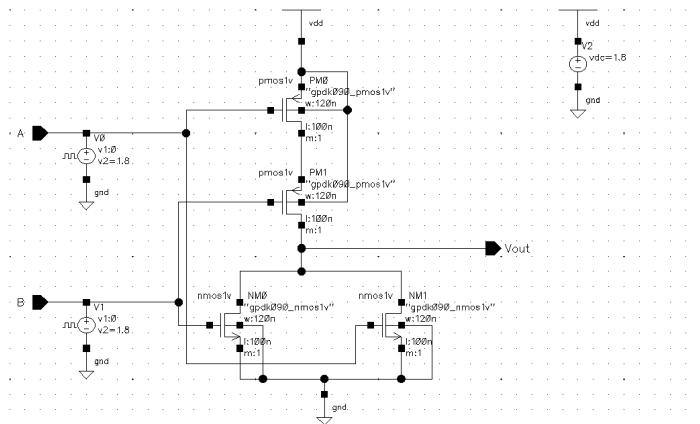


Figure 6.1: Schematic of 2-Input CMOS NOR

Symbol

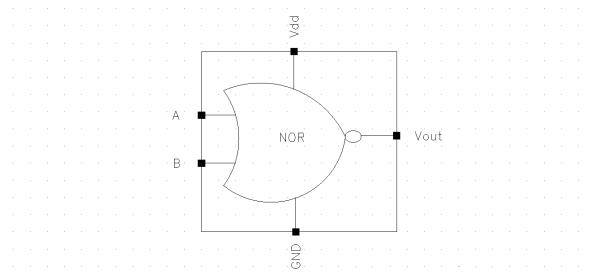


Figure 6.2: Symbol of 2-Input CMOS NOR

Layout

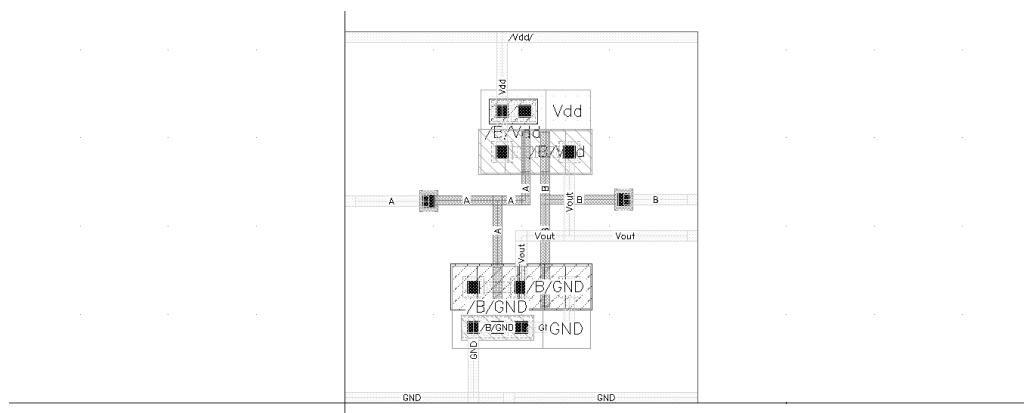


Figure 6.3: Layout of 2-Input CMOS NOR

RC Extraction

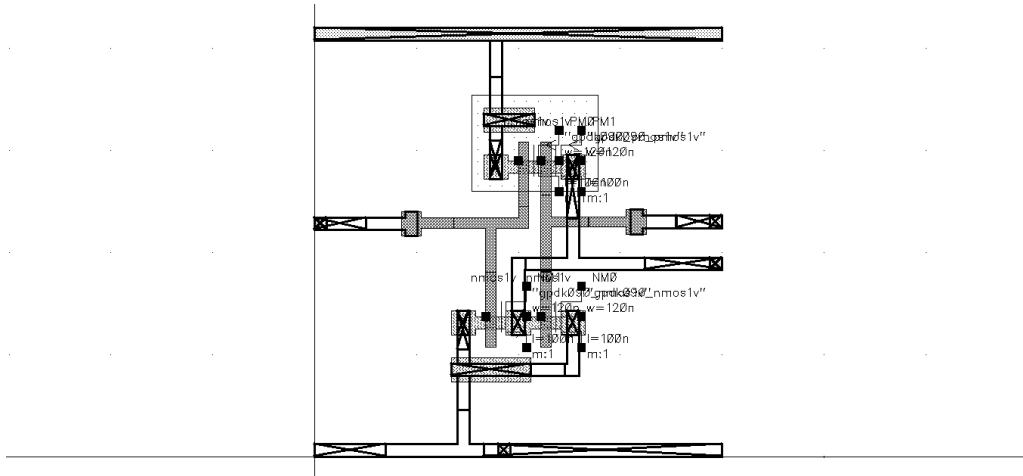


Figure 6.4: RC Extraction of NOR

6.3 Results



Figure 6.5: Transient Response of NOR

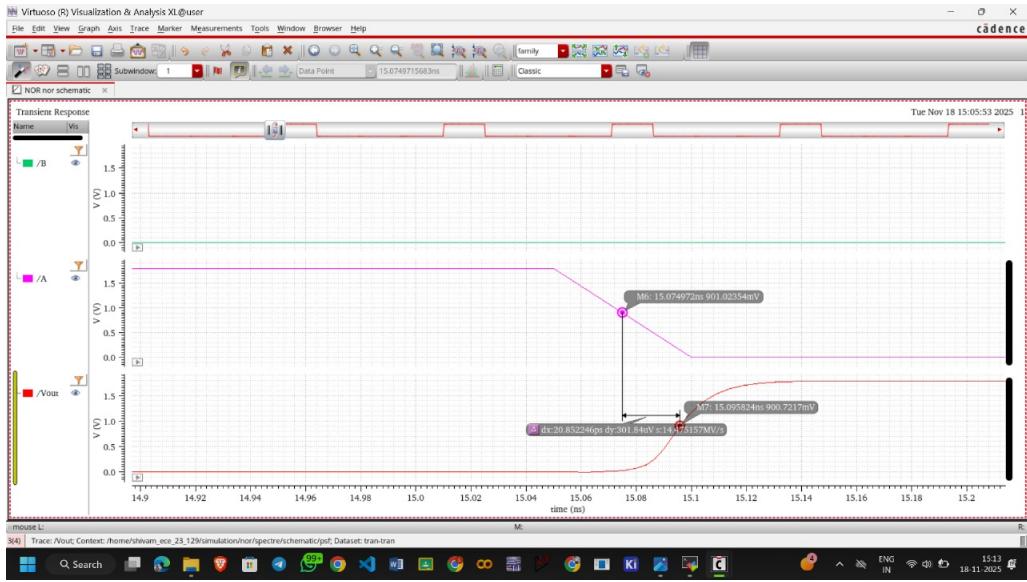
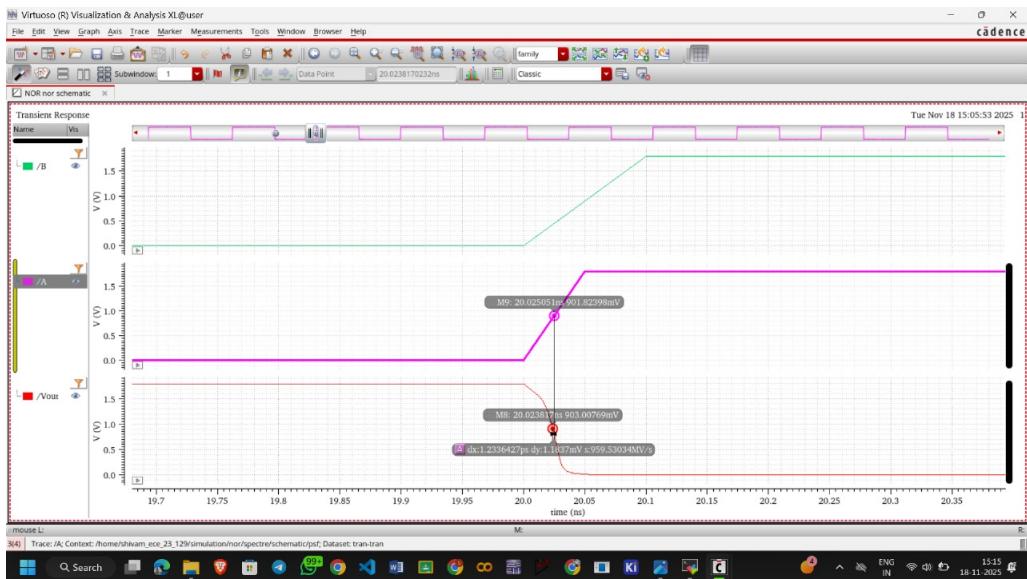
Figure 6.6: Transient Response of NOR for t_{PLH} Figure 6.7: Transient Response of NOR for t_{PHL}

Table 6.1: Propagation Delay Calculation for 2-Input CMOS NAND Gate

Transition	Measurement Points (ns)	Delay (ps)
t_{PLH} (Y: 0→1)	IN fall = 10.15, OUT rise = 10.1801	30.124
t_{PHL} (Y: 1→0)	IN rise = 40.0499, OUT fall = 40.4615	3.798
Propagation delay	—	16.961 ps

6.4 Conclusion

- The simulated 2-input CMOS NOR gate exhibits the expected logical behaviour: the output is HIGH only when both inputs A and B are LOW, and it goes LOW as soon as either input is driven HIGH, which matches the NOR truth table.

- From the measured data:
 - For the 0→1 output transition (t_{PLH}), the input fall at 10.15 ns and output rise at 10.1801 ns give a delay of approximately 30.124 ps.
 - For the 1→0 transition (t_{PHL}), the input rise at 40.0499 ns and output fall at 40.4615 ns yield a delay of about 3.798 ps.

The average propagation delay is therefore around 16.961 ps, indicating a noticeably larger low-to-high delay compared to high-to-low, as expected from the series PMOS stack.

- Layout-level parasitics in the NOR gate (particularly in the pull-up path) further increase the effective delay in post-layout simulation. To enhance performance, one may increase PMOS widths, insert buffers, or optimize routing to reduce RC loading.
- Overall, the experiment confirms correct NOR logic operation and illustrates how transistor topology and parasitic effects combine to determine the dynamic performance of CMOS logic gates.

Experiment 7

Pass Transistor Logic as NMOS

7.1 Aim

To design a pass-transistor logic (PTL) circuit implemented with NMOS, run transient simulations in Cadence.

7.2 Theory

Pass-Transistor Logic (PTL) makes use of MOS transistors as controlled switches to propagate signals between circuit nodes, instead of implementing separate pull-up and pull-down networks as in conventional CMOS logic. This can reduce transistor count, area and sometimes power consumption, but it also introduces issues with degraded logic levels.

NMOS Switching Principle

- When the gate voltage is driven HIGH ($V_G = V_{DD}$), the NMOS device turns ON and behaves like a low-resistance conducting path, allowing the input node to pass through to the output.
- When the gate voltage is LOW ($V_G = 0$), the NMOS is OFF and the output node is effectively isolated (high impedance).

Voltage Transfer Characteristics

- NMOS passes strong logic ‘0’:

$$V_{OL} \approx 0 \text{ V}$$

because the NMOS device can discharge the output node very close to ground when passing a logic 0.

- NMOS passes weak logic ‘1’: When the input is at V_{DD} , conduction stops once

$$V_{GS} = V_{tn},$$

leading to a maximum output high level of

$$V_{OH} \approx V_{DD} - V_{tn}.$$

This means the logic ‘1’ is not transferred all the way to V_{DD} but is instead degraded by the threshold voltage, which reduces noise margin and may not properly drive further CMOS stages.

Logic Table

For a simple NMOS pass gate controlled by signal C , with input A and output Y :

C	A	Y
0	0	High Impedance
0	1	High Impedance
1	0	0
1	1	$1 - V_{tn}$ (Degraded)

7.3 Simulation Setup

- NMOS model from the given PDK.
- Supply voltage $V_{DD} = 1.8$ V.
- Input square wave (0–1.8 V), 10 ns period.
- Control signal $C = 1.8$ V (static ON).

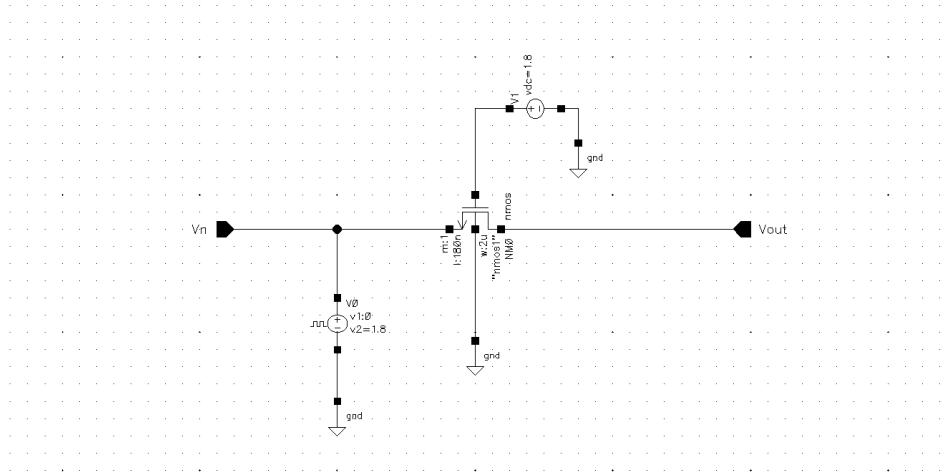


Figure 7.1: Schematic of PTL as NMOS

7.4 Results and Discussion

From the transient simulation:

- $VOH = 1.254$ V.
- $VOL = 0.00$ V.
- Voltage drop when passing ‘1’:

$$V_{OH} \approx V_{DD} - V_{tn} \approx 1.8 - 1.254 = 0.546 \text{ V}$$

which corresponds approximately to the NMOS threshold V_{th} .

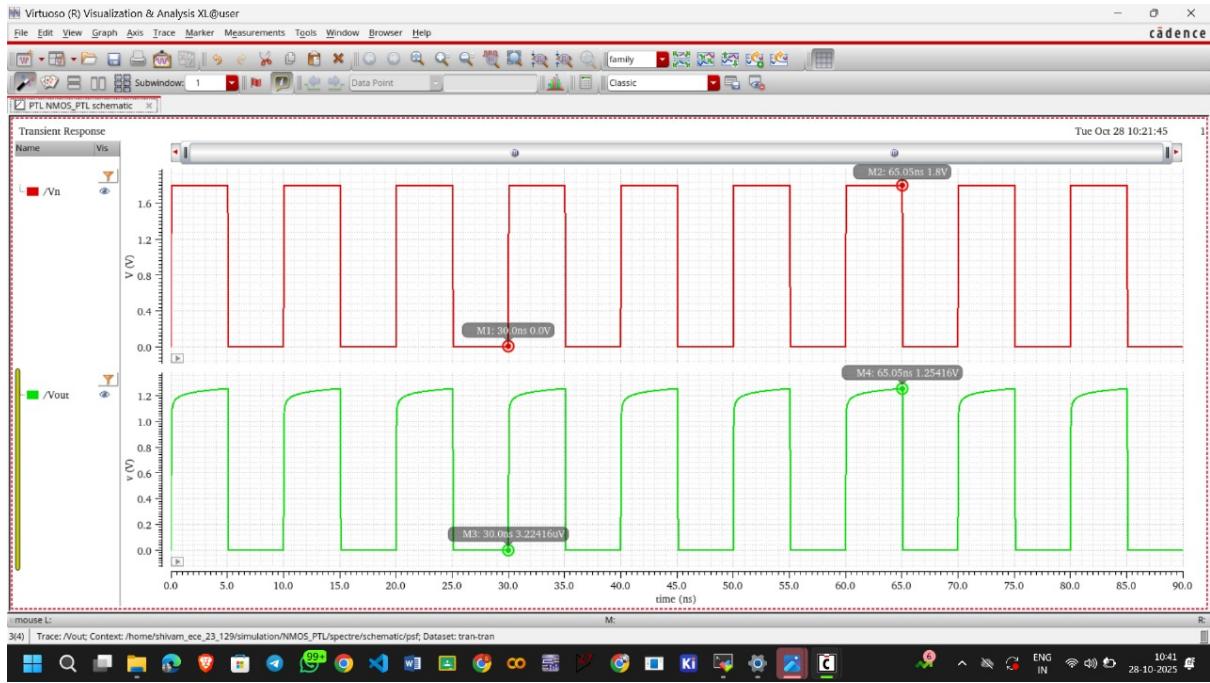


Figure 7.2: Transient Response

- The output transitions to low are very sharp and reach ground; the high settles to the degraded value shown above.
- The rising transition shows slower edges due to reduced driving capability of NMOS for high levels.
- The falling transition is faster because NMOS pulls down strongly.

7.5 Conclusion

The NMOS-only pass-transistor logic circuit has been successfully designed and simulated. The results verify the expected behaviour of NMOS PTL: logic ‘0’ is transferred strongly, reaching approximately 0 V, while logic ‘1’ appears as a degraded high level of about 1.254 V due to the NMOS threshold voltage drop. The waveforms clearly show sharp fall transitions and slower, limited rise transitions. Although this style reduces transistor count and area, the reduced noise margins and incomplete logic swing imply that level-restoring circuits (such as inverters or full transmission gates) are often required before driving standard CMOS logic.

Experiment 8

Pass Transistor Logic as PMOS

8.1 Aim

To design a pass-transistor logic (PTL) circuit implemented with PMOS-only pass transistors, run transient simulations in Cadence.

8.2 Theory

Pass-Transistor Logic (PTL) uses MOSFETs as signal-controlled switches to transfer logic levels between nodes, instead of separate pull-up and pull-down networks as in static CMOS. In a PMOS-only PTL design, only PMOS transistors are used as pass devices.

A PMOS transistor:

- turns ON when its gate is at a low voltage (logic ‘0’),
- turns OFF when its gate is at a high voltage (logic ‘1’).

PMOS devices pass a strong logic high but a weak logic low:

- When passing logic ‘1’, the PMOS can pull the output very close to V_{DD} , so

$$V_{OH} \approx V_{DD}.$$

- When passing logic ‘0’, conduction stops when

$$V_{SG} \approx |V_{thp}|,$$

leaving the lowest achievable output voltage at about

$$V_{OL} \approx |V_{thp}|,$$

which is above 0 V and therefore a degraded logic 0.

This behaviour results in good transfer of ‘1’ but poor transfer of ‘0’, which can compromise noise margins and may not reliably switch following CMOS stages unless level-restoring circuits are used.

Advantages

- Well suited for passing logic 1 signals (good high-level transfer).
- Can reduce transistor count for some logic functions, saving area.

Disadvantages

- Logic 0 is not restored to ground; the low level remains around $|V_{thp}|$.
- Often requires buffers or transmission gates to restore full rail-to-rail logic levels.
- Degraded low levels can slow down subsequent transitions and reduce noise margins.

8.3 Schematic

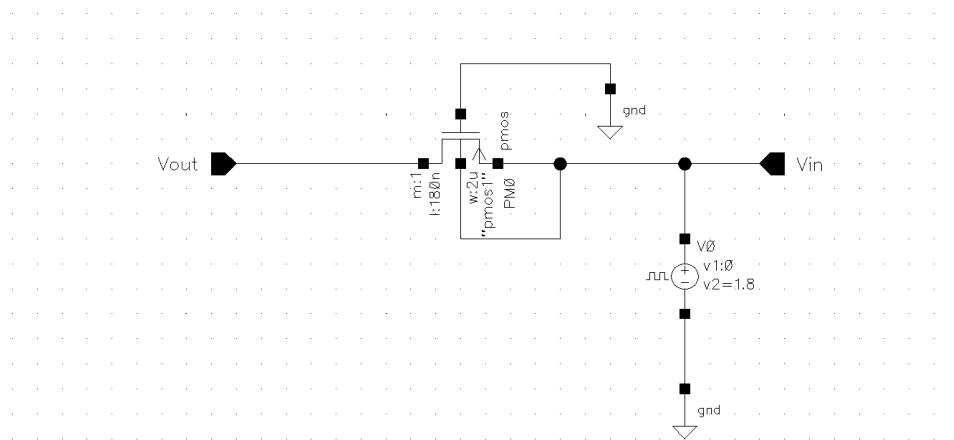


Figure 8.1: Schematic of PTL as PMOS

8.4 Result and Discussion

- When the input is high (1.8 V) the output also rises to = 1.80 V ($V_{OH} = VDD$).
- When the input is low (0 V) the output only falls to 0.255 V ($V_{OL} = 255.151 \text{ mV}$), not to 0 V.

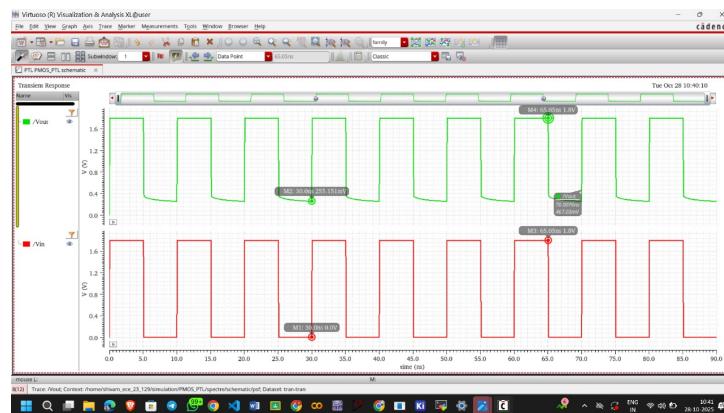


Figure 8.2: Transient Response

8.5 Conclusion

The transient simulation of the PMOS-only pass-transistor circuit confirms the theoretical expectations: logic ‘1’ is transferred strongly, with the output rising to nearly the full supply voltage (1.8 V), whereas logic ‘0’ is degraded and only falls to about 0.255 V, close to the magnitude of the PMOS threshold voltage. Thus, the circuit exhibits good high-level transfer but poor low-level restoration. While PMOS PTL can reduce device count and is useful when strong logic 1 transfer is required, additional level-restoring stages are generally needed to ensure reliable rail-to-rail operation in larger CMOS systems.

Experiment 9

4X4 NOR ROM Array

9.1 Aim

To design a 4×4 NOR-based ROM array, verify its logical functionality through simulation.

9.2 Theory

A Read Only Memory (ROM) stores fixed data that is defined at design or program time. In a NOR-based ROM architecture, each memory cell is realized using an NMOS transistor connected between a bitline (BL) and ground, with its gate driven by a wordline (WL). The presence or absence of this transistor at a WL–BL intersection encodes a single bit of information.

In a typical NOR ROM cell:

- The NMOS transistor's drain is connected to the bitline (BL),
- The source is tied to ground (GND),
- The gate is controlled by the corresponding wordline (WL),
- A pull-up device (PMOS or resistive load) keeps the BL high when no NMOS is conducting.

Bit Encoding Rule

- **NMOS present (programmed):** When the wordline is driven HIGH, the NMOS turns ON and discharges the BL to LOW. This is used to represent a stored bit ‘0’.
- **NMOS absent (unprogrammed):** With no NMOS at that intersection, the BL remains pulled HIGH, representing a stored bit ‘1’.

Thus:

$$\text{NMOS exists} \rightarrow 0, \quad \text{NMOS absent} \rightarrow 1$$

ROM Architecture

A 4×4 NOR ROM contains:

- 4 Wordlines (WL0–WL3),
- 4 Bitlines (BL0–BL3),
- Up to 16 NMOS cells at the intersections, selectively included or omitted to realize the desired stored words.

During read operation, exactly one wordline is activated at a time so that one 4-bit row is accessed.

Working of 4×4 NOR ROM

Read Operation Steps

1. **Precharge:** All BLs are initially held HIGH using pull-up PMOS transistors.
2. **Wordline Selection:** One WL is driven HIGH, while the remaining WLs are kept LOW.
3. **Bitline Response:**
 - If a programmed NMOS is present for the selected WL and a given BL, the BL is pulled LOW \Rightarrow stored bit 0.
 - If no NMOS is present at that location, the BL stays HIGH \Rightarrow stored bit 1.
4. **Sensing:** A CMOS inverter or sense amplifier converts the BL voltage to a full-swing logic output.

Programmable 4×4 ROM Table

Wordline	R1	R2	R3	R4
WL0	1	0	0	0
WL1	0	1	0	0
WL2	0	0	1	0
WL3	0	0	0	1

Table 9.1: Programmable Table of 4×4 NOR ROM

Expected Outputs

When each wordline is asserted individually:

$$\text{WL0} \Rightarrow 0101$$

$$\text{WL1} \Rightarrow 0011$$

$$\text{WL2} \Rightarrow 1001$$

$$\text{WL3} \Rightarrow 0110$$

Schematic

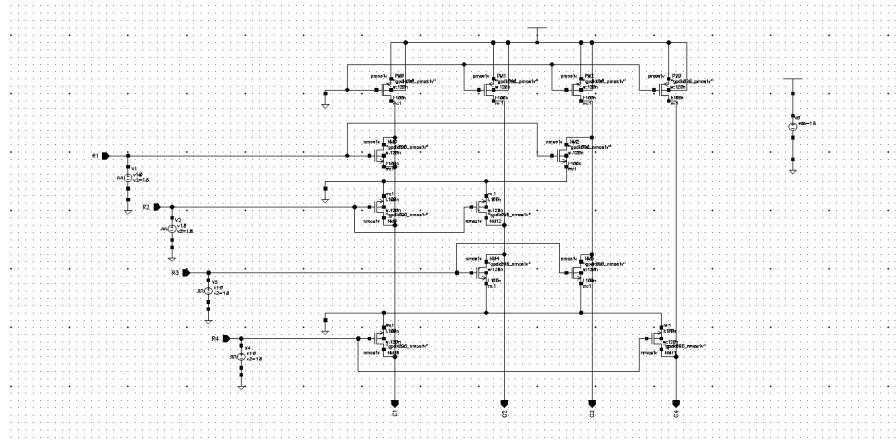


Figure 9.1: Schmatic of 4X4 NOR ROM Array

9.3 Results

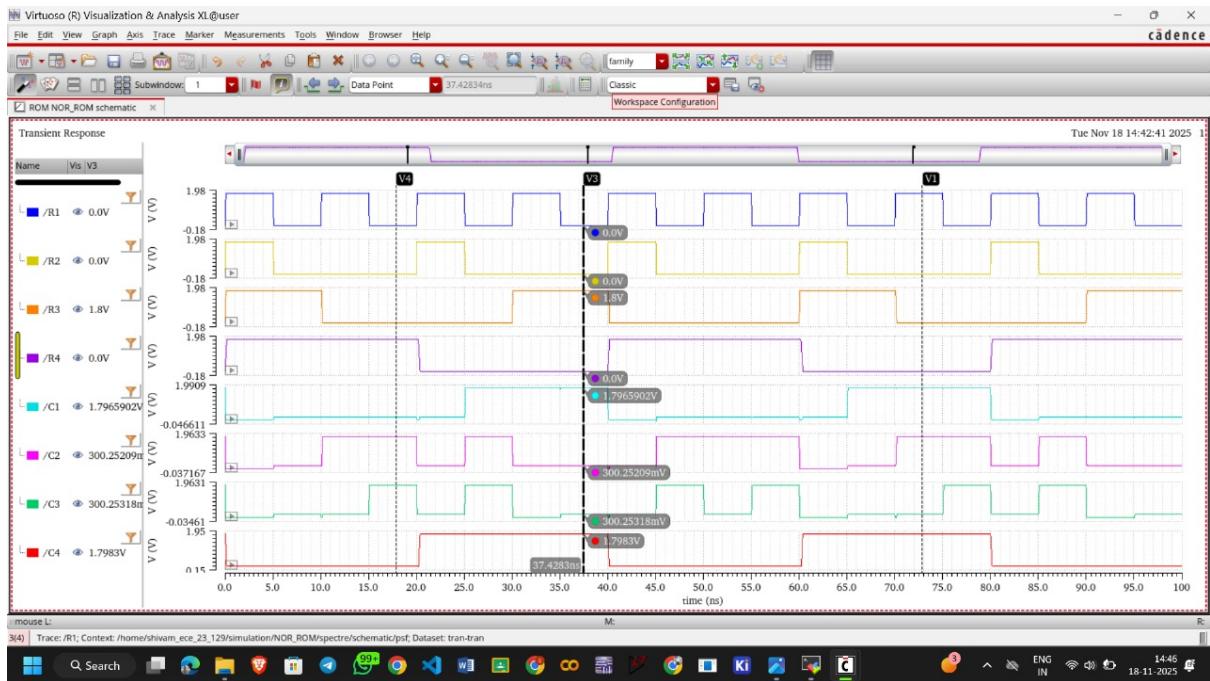


Figure 9.2: Transient Analysis of 4X4 NOR ROM Array

The transient simulation of the 4×4 NOR ROM was performed by activating each wordline sequentially. The four bitline outputs (R_1, R_2, R_3, R_4) were observed through the corresponding sense outputs C_1, C_2, C_3 and C_4 .

The bitline voltages transition according to the programmed NOR ROM structure.

9.4 Conclusion

The designed 4×4 NOR ROM array correctly realizes data storage using NMOS cells as pull-down devices. During simulation, each wordline activation produces the expected combination of high and low levels on the bitlines, matching the programmed table. A present NMOS cell discharges the bitline to logic 0, while the absence of a device leaves the bitline at logic 1 via the pull-up. The read operation, encoding scheme and overall ROM organization are thus verified to be functionally correct.

Experiment 10

4X4 NAND ROM Array

10.1 Aim

To design a 4×4 NAND-based ROM array, verify its logical functionality through simulation.

10.2 Theory

A NAND-based Read Only Memory (ROM) stores fixed data by arranging NMOS transistors in series along each bitline, forming a NAND-type pull-down path. Each memory cell corresponds to the presence or absence of an NMOS transistor at a given wordline-bitline intersection.

The NAND ROM exploits the fact that a NAND gate output goes LOW only when all transistors in its pull-down chain are ON. If the pull-down path is broken at any point (missing device), the bitline remains at a HIGH level.

Working of NAND-Based ROM

The read operation is based on enabling one wordline at a time:

1. All bitlines are initially precharged or held HIGH.
2. The selected wordline is driven HIGH; other wordlines remain LOW.
3. For a given bitline:
 - If the NMOS transistor corresponding to the selected wordline is present and the overall series path allows conduction, the BL is discharged to LOW, interpreted as logic 0.
 - If the transistor at that location is absent, the conduction path is interrupted and BL stays HIGH, interpreted as logic 1.
4. A sensing inverter then converts the BL voltage into a full-swing digital output.

Thus, the configuration of present/absent NMOS devices along each bitline defines the stored word associated with each wordline.

Programmable 4×4 NAND ROM Table

Wordline	R1	R2	R3	R4
WL0	0	1	1	1
WL1	1	0	1	1
WL2	1	1	0	1
WL3	1	1	0	0

Table 10.1: Stored data in 4×4 NAND-based ROM

Expected Outputs

When each wordline is activated one at a time, the output words observed at the bitlines are as follows:

$$\text{WL0} \Rightarrow 0101$$

$$\text{WL1} \Rightarrow 0011$$

$$\text{WL2} \Rightarrow 1001$$

$$\text{WL3} \Rightarrow 0111$$

Schematic

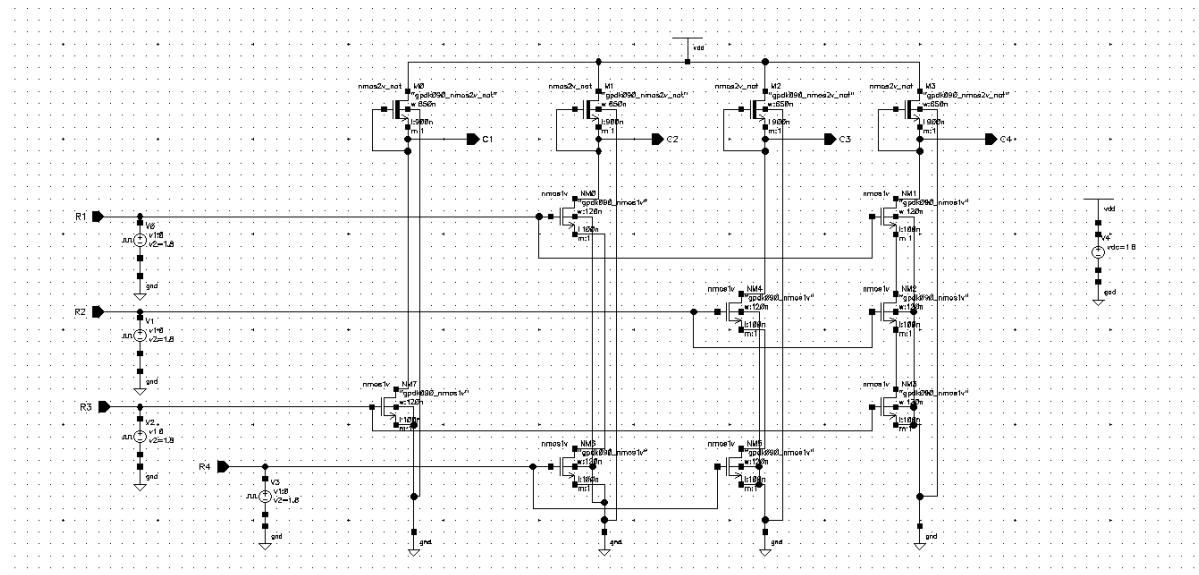


Figure 10.1: Schmatic of 4X4 NAND ROM Array

10.3 Results



Figure 10.2: Transient Analysis of 4X4 NAND ROM Array

The transient simulation of the 4×4 NAND ROM was performed by activating each wordline sequentially. The four bitline outputs (R1, R2, R3, R4) were observed through the corresponding sense outputs C1, C2, C3 and C4.

10.4 Conclusion

The 4×4 NAND ROM array successfully demonstrates data storage using NMOS transistors arranged in series along the bitlines. For each activated wordline, the simulated bitline outputs follow the programmed table: bitlines with a complete pull-down path are discharged to logic 0, whereas broken paths remain at logic 1. The sensing circuitry correctly converts these bitline levels to digital outputs, confirming that the NAND-based ROM architecture and encoding scheme operate as intended.

Experiment 11

CMOS based SRAM (6T)

11.1 Aim

To design a 6T CMOS based SRAM, verify its logical functionality through simulation.

11.2 Theory

A 6T CMOS SRAM (Static Random Access Memory) cell stores a single bit of data using six transistors: two cross-coupled CMOS inverters that form a bistable latch, and two NMOS access transistors that connect the latch to the bitlines during read and write operations. SRAM is a **volatile** memory, meaning that it retains data only while power is applied.

The cross-coupled inverters provide two stable states:

$$Q = 1, \overline{Q} = 0 \quad \text{or} \quad Q = 0, \overline{Q} = 1$$

Logical Storage Convention

$$\begin{aligned} \text{Stored } 1 &\Rightarrow Q = V_{DD}, \overline{Q} = 0 \\ \text{Stored } 0 &\Rightarrow Q = 0, \overline{Q} = V_{DD} \end{aligned}$$

Operating Modes of a 6T SRAM Cell

1. Hold (Standby) Mode

- Wordline (WL) is held LOW, turning both access transistors OFF.
- The cross-coupled inverters reinforce each other via positive feedback and maintain the stored data at Q and \overline{Q} without external intervention.
- The Static Noise Margin (SNM) quantifies how robustly the cell can retain data under disturbances.

2. Write Operation

To write a bit into the cell:

1. The write circuitry forces the bitlines as:

$$\text{Write } 1 \Rightarrow BL = V_{DD}, \overline{BL} = 0$$

$$\text{Write } 0 \Rightarrow BL = 0, \overline{BL} = V_{DD}$$

2. WL is asserted HIGH, enabling the access transistors and connecting the internal nodes Q and \overline{Q} to BL and \overline{BL} .

3. The bitline drivers overpower the previous latch state and flip the inverters to the new stored value.
4. WL is then de-asserted, isolating the cell and preserving the new data.

3. Read Operation

Before reading:

- BL and \overline{BL} are precharged to V_{DD} .

During read:

1. WL is raised HIGH, connecting Q and \overline{Q} to the bitlines through the access transistors.
2. Depending on the stored state:
 - If $Q = 1$, BL remains close to V_{DD} and \overline{BL} discharges slightly.
 - If $Q = 0$, BL discharges slightly while \overline{BL} stays near V_{DD} .
3. A differential sense amplifier detects the small voltage difference between BL and \overline{BL} and regenerates a full logic level at the output without upsetting the stored bit.

Schematic

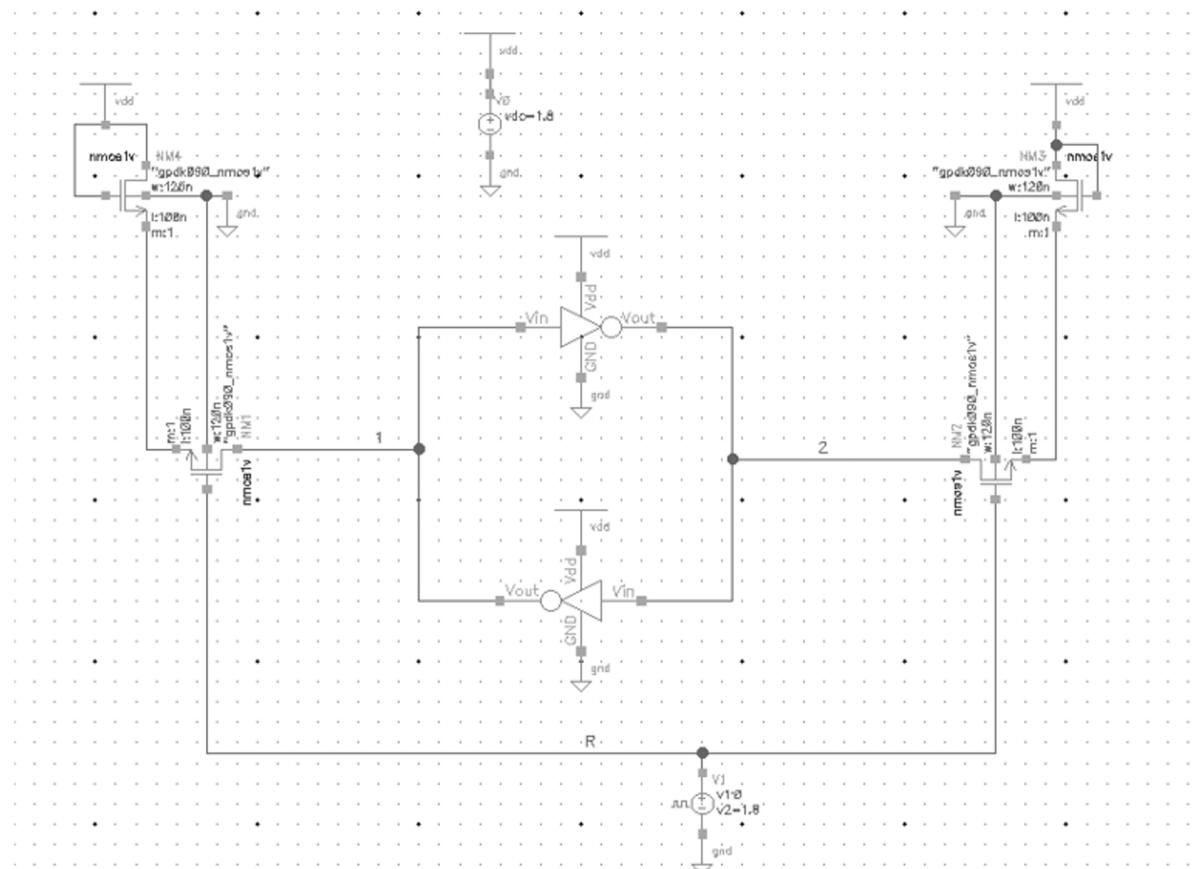


Figure 11.1: Schematic of 6T SRAM

11.3 Results

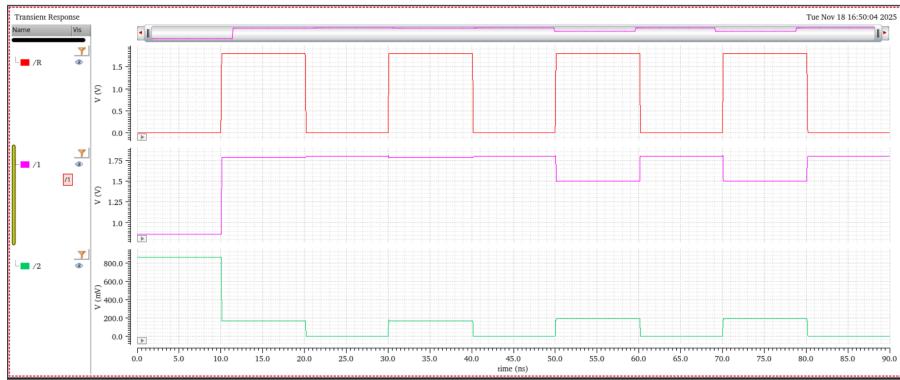


Figure 11.2: Transient Analysis

11.4 Conclusion

The designed 6T CMOS SRAM cell has been verified through transient simulation to operate correctly in hold, read and write modes. In standby, the cross-coupled inverters maintain a stable logic state as long as power is present. During a write operation, the forced bitline levels and asserted wordline successfully overwrite the previous stored value. In read mode, the precharged differential bitlines develop a small voltage difference that is consistent with the stored bit, and this difference can be amplified by a sense amplifier without disturbing the cell contents. The simulation results thus confirm proper functionality and illustrate the fundamental operating principles of a standard 6T SRAM cell.