# LABORATÓRIO 3 – 28nm - PORTAS LÓGICAS NAND E NOR

**Prof. Fernando Gehm Moraes – Revisão: 08/setembro/2026**

## Objetivo deste laboratório

Analisar a influência do número de **transistores em série** no atraso das portas lógicas, **a posição do chaveamento** (qual entrada está mudando de estado), e o uso do método *logic effort*.

**Baixar os arquivos do laboratório e configurar o ambiente para a simulação no VDI:**

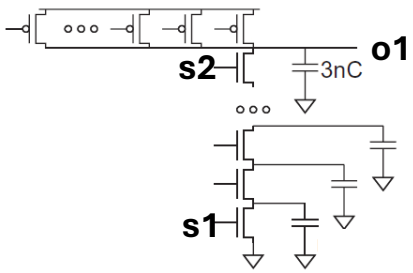
```
cd ~/onedrive
wget https://fgmoraes.github.io/microel/lab3.zip
unzip lab3.zip
cd lab3
source ~/onedrive/micro.sh
```

## Arquivo que descreve as portas NAND, de 2 a 6 entradas

**Abrir** o *netlist* nand6.sp e observar neste arquivo:

- Linha **2**: data de atualização - 08/setembro/2026
- Linha **11**: processo 28nm
- Linha **14**: `.param wp=0.2u mob=1.3 Cload=3fF`, onde:
  - `wp`: dimensão $W_P$ em $\mu m$
  - `mob`: relação de mobilidade ($\mu n/\mu p$) para esta tecnologia (1,3)
  - `cload`: carga de saída de cada porta lógica
- **Linhas 16-21** exemplo de descrição *spice* da porta lógica NAND2. Transistores P em paralelo, e os transistores N em série.

```
.SUBCKT nand2 o1 s1 s2 vcc
M1   o1 s1 vcc vcc pch_mac w=wp          l=30n
M2   o1 s2 vcc vcc pch_mac w=wp          l=30n
M3   0  s1 2   0   nch_mac w='wp*2/mob'  l=30n
M4   2  s2 o1  0   nch_mac w='wp*2/mob'  l=30n
.ENDS nand2
```




  - **s2**: entrada que está mudando de estado próxima da **saída**
  - **s1**: entrada que está mudando de estado próxima de ***gnd***
  - para NAND3 a NAND6 as demais entradas (s3 a s6) ficam em 0.9 V, para os transistores N em série conduzirem

Para equilibrar os tempos de propagação de subida e descida, temos para um inversor:

$$W_P = W_N * mob \rightarrow W_N = \frac{W_P}{mob}$$

Os transistores P, na porta NAND2, por estarem em paralelo, possuem o mesmo dimensionamento ($W_P$). Os transistores N, em **série**, devem ter a relação de mobilidade ($mob$) respeitada, através do inverso da soma dos inversos:

$$\frac{1}{{}^1/_{W_N} + {}^1/_{W_N}} = \frac{W_P}{mob} \rightarrow W_N = \frac{W_P * 2}{mob}$$

conforme netlist: `'wp*2/mob'`

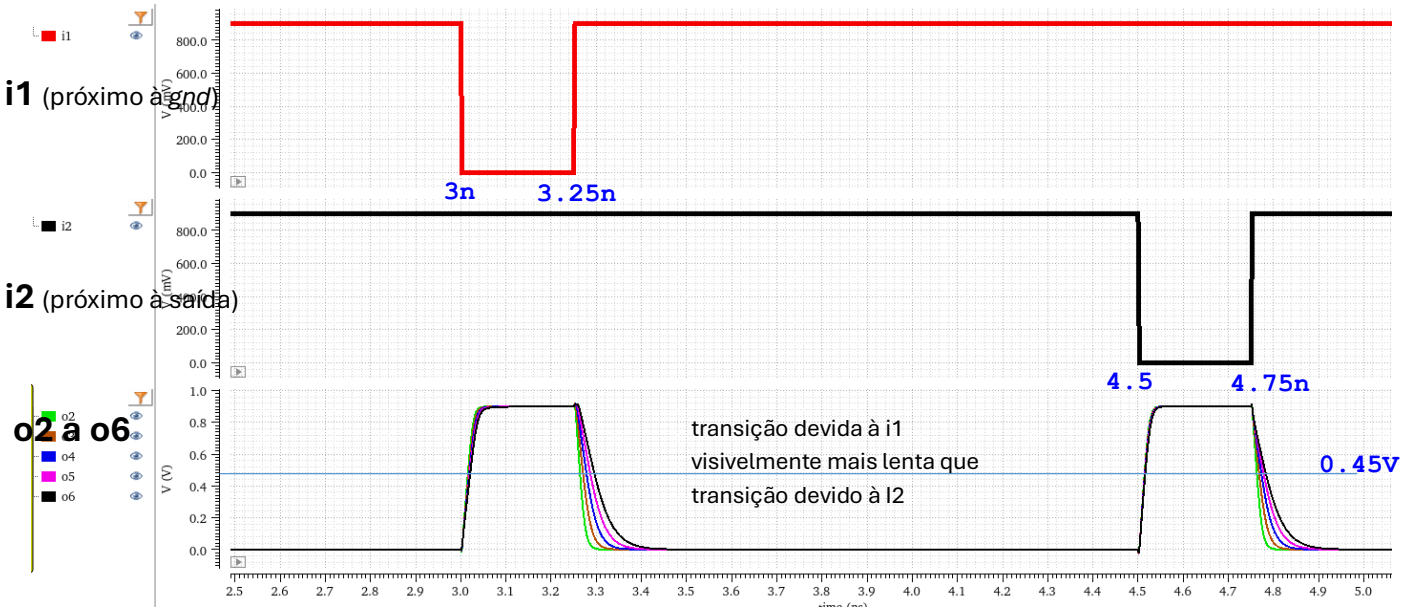
- Linha **86**: tensão de alimentação (`vcc vcc 0 dc 0.9`)
- linhas **87** e **88**: comandos PWL (geração dos estímulos para as entradas i1 e i2)
- linhas **89** a **92**: tensão fixa em '0.9' nas entradas intermediárias da pilha série (i3, i4, i5, i6)

- **Dimensionamento dos transistores**. Para estudarmos o efeito dos transistores em série, **manteremos** o dimensionamento das portas NAND (2 a 6 entradas) igual ao dimensionamento da porta NAND2, e depois faremos a alteração necessária.

## Simulação da porta NAND [6 pt]

1. [1 pt] Apresentar as formas de onda com as entradas (i1 e Ii2) e as saídas das portas lógicas *nand* de 2 a 6 entradas, como abaixo. **Compreender** os comandos *pwl* (pares <tempo> <tensão>):

```
v1 i1 0 pwl(0 0.9   3n    0.9   3.003n 0    3.25n 0    3.253n 0.9)
v2 i2 0 pwl(0 0.9   4.5n  0.9   4.503n 0    4.75n 0    4.753n 0.9)
```




**i1** (próximo à *gnd*)

**i2** (próximo à saída)

**o2 a o6**

Preencher a tabela abaixo para as portas NAND, a partir do arquivo *nand6.measure*.
**O NETLIST SPICE PRECISA SER COMPLETADO COM AS MEDIDAS DAS NANDS 3 a 6.**

| *N# Entradas* | *descida_out (ps)* *tx_fo* | *descida_gnd (ps)* *tx_fs* | *subida_out (ps)* *tx_to* | *subida_gnd (ps )* *tx_rs* |
|---|---|---|---|---|
| **2** | **12,3141** | **13,2997** | **12,1759** | **13,4725** |
| **3** | | | | |
| **4** | | | | |
| **5** | | | | |
| **6** | | | | |

**Observar: *t2_fo* ≈ *t2_ro* → coerente com o método *logic effort* (diferença de 0,14ps)**

**Os tempos são gerados no arquivo *measure*, para a nand2**

| | | |
|---|---|---|
| **descida_gnd:** | t2_fs | (fall supply - transição i1 próximo à gnd) |
| **descida_out:** | t2_fo | (fall out – transição i2 próximo à saída) |
| **subida_gnd:** | t2_rs | (rise, transição em i1) |
| **subida_out:** | t2_ro | (rise, transição em i2) |

2. [1 pt] Plotar **um** gráfico com **4 curvas**, uma para cada coluna da tabela acima. No eixo X teremos o número de entradas (2 a 6), e no eixo Y o atraso em *ps* para as 4 curvas. Para cada curva colocar o respectivo rótulo: **fall(i1)** [corresponde a t2_fs], **fall(i2)** [corresponde a t2f_o], **rise(i1)** [corresponde a t2_rs], **rise(i2)** [corresponde a t2_ro]

3. [1 pt] Explicar o impacto do número de transistores em série no plano N na porta NAND no **tempo de propagação de descida**.

4. [0,5 pt] O **tempo de propagação de descida** é mais afetado quando a entrada que varia está próxima de *gnd* ou da *saída*? Explicar a razão.

5. [0,5 pt] Por que o **tempo de propagação de subida** aumenta, apesar de os transistores P estarem com o mesmo dimensionamento e em paralelo?

6. [1,0 pt] Utilizando o método *logic effort* alterar o dimensionamento dos transistores N de tal forma que o tempo de propagação de **descida próximo à saída** (t2_2_fo, ..., t2_6_fo) para as 5 portas NAND seja praticamente iguais. Preencher a tabela abaixo. Copiar o arquivo nand6.sp para nand6_sized.sp.

| N# Entradas | *descida_out (ps)* ***tx_fo*** | *descida_gnd (ps)* ***tx_fs*** | *subida_out (ps)* ***tx_to*** | *subida_gnd (ps )* ***tx_rs*** |
|---|---|---|---|---|
| **2** | 12,3127 | 13,2998 | 12,176 | 13,4732 |
| **3** | | | | |
| **4** | | | | |
| **5** | | | | |
| **6** | | | | |

A diferença em ***descida_out*** deve ser ≈1,1065 ps. Conferir. O pior tempo foi para a nand4 e o melhor tempo para a nand6 (diferença= t4_fo – t6_fo)

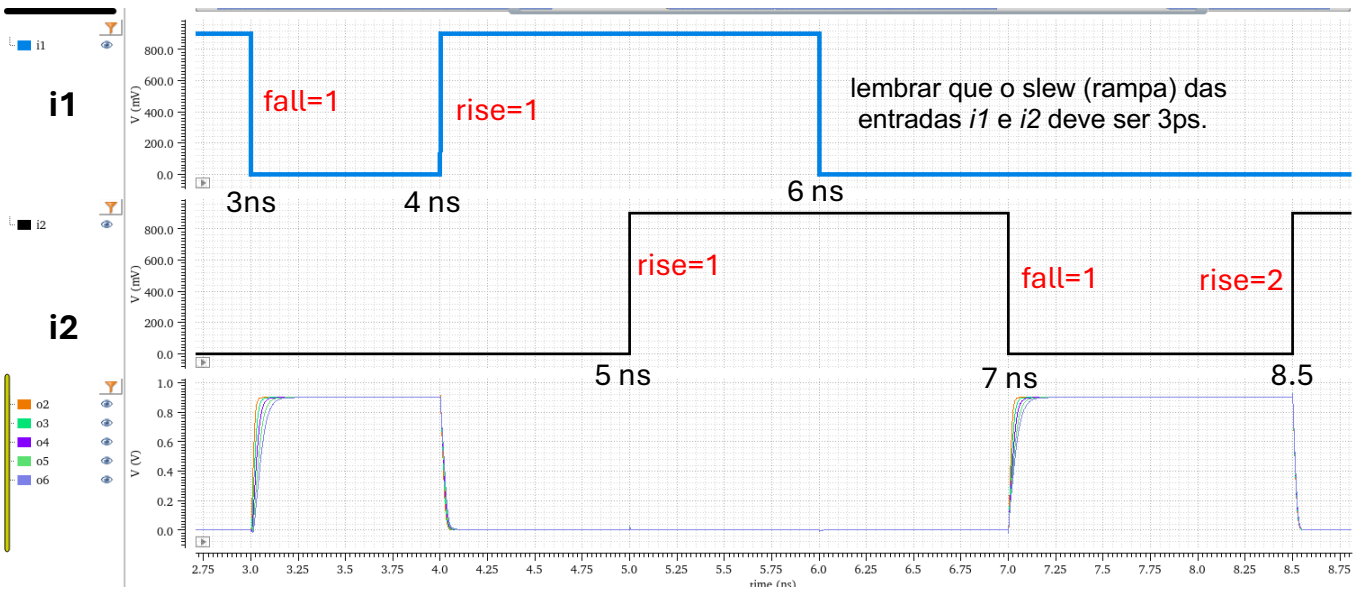
7. [1 pt] Plotar um gráfico com 4 curvas (como na questão 2), usando os tempos obtidos na questão 6. Para cada curva colocar o respectivo rótulo: **fall(i1)** [corresponde a t2_Fs], **fall(i2)** [corresponde a t2_Fo] , **rise(i1)** [corresponde a t2_Rs], **rise(i2)** [corresponde a t2_Ro]

   **Qual dos 4 tempos foi o mais penalizado**? **Por quê?** Dica: o Wn está aumentando à medida que o número de transistores em série aumenta, mas mantém-se o Wp fixo.

   **O que se conclui da aplicação direta do método *logic effort*?** (sugestão de leitura: *progressive transistor sizing* – ao final deste documento)

## Simulação da porta NOR [4 pt]

Saída esperada:



- **Completar o *netlist* para simular portas NOR, de 2 a 6 entradas**. O arquivo abaixo encontra-se no material do laboratório:

```
* circuitos nor
* FERNANDO MORAES   -  PUCRS
* revisão em 08/setembro/2026

simulator lang=spectre insensitive=no
include "/soft64/spectre28/toplevel.scs"  section=top_tt
simulator lang=spice

.param Cload=3fF mob=1.3  wn=0.15u  wp='2*wn*mob'

.SUBCKT nor2    o1  s1  s2  vcc
M1  10  s1  vcc vcc pch_mac w=wp    l=30n
M2  o1  s2  10  vcc pch_mac w=wp    l=30n
M10 0   s1  o1  0   nch_mac w=wn    l=30n
M11 0   s2  o1  0   nch_mac w=wn    l=30n
.ENDS   nor2

* Lembrar: entrada s1 próxima à vcc e entrada s2 próxima à saída
.SUBCKT nor3    o1  s1  s2  s3  vcc
... completar ...
.ENDS   nor3

.SUBCKT nor4    o1  s1  s2  s3  s4  vcc
... completar ...
.ENDS   nor4

.SUBCKT nor5    o1  s1  s2  s3  s4  s5  vcc
... completar ...
.ENDS   nor5

.SUBCKT nor6    o1  s1  s2  s3  s4  s5  s6  vcc
... completar ...
.ENDS   nor6

** circuito propriamente dito
X1 o2 i1 i2 vcc nor2
X2 o3 i1 i2 i3 vcc nor3
X3 o4 i1 i2 i3 i4 vcc nor4
X4 o5 i1 i2 i3 i4 i5 vcc nor5
X5 o6 i1 i2 i3 i4 i5 i6 vcc nor6

** alimentações
vcc  vcc 0  dc 0.9
v1 i1 0 pwl(... completar ...)
v2 i2 0 pwl(... completar ...)
***  entradas não utilizadas em 0v3 i3  0  dc 0
v4 i4  0  dc 0
v5 i5  0  dc 0
v6 i6  0  dc 0

.tran 0.001N 10N

Cl1 o2 0 Cload
Cl2 o3 0 Cload
Cl3 o4 0 Cload
Cl4 o5 0 Cload
Cl5 o6 0 Cload

***  SEMANTICA DAS MEDIDAS <porta><subida/descida><entrada proxima da alim ou proxima da saida>

.measure tran n2_subida_vdd  trig v(i1)  val=0.45  td=2n fall = 1      targ v(o2)  val=0.45        rise = 1
.measure tran n2_descida_vdd trig v(i1)  val=0.45  td=2n rise = 1      targ v(o2)  val=0.45        fall = 1
.measure tran n2_subida_out  trig v(i2)  val=0.45  td=2n fall = 1      targ v(o2)  val=0.45        rise = 2
.measure tran n2_descida_out trig v(i2)  val=0.45  td=2n rise = 2      targ v(o2)  val=0.45        fall = 2

.measure tran t2_Fs   param = '1e12*n2_descida_vdd'
.measure tran t2_Fo   param = '1e12*n2_descida_out'
.measure tran t2_Rs   param = '1e12*n2_subida_vdd'
.measure tran t2_Ro   param = '1e12*n2_subida_out'


...completar as medidas para outras NOR...


.END
```

## RESPONDER:

1. [0,5 pt] Apresentar as formas de onda com as entradas (i1 e i2) e as saídas das **nor** 2 a 6 entradas, conforme o esperado

2. [1,5 pt] Para as portas NOR fazer uma **tabela** equivalente à NAND e plotar um gráfico com **4 curvas**, uma para cada coluna da tabela. No eixo X teremos o número de entradas, e no eixo Y os tempos de propagação.

| N# Entradas NOR | *descida_out (ps)* *tx_fo* | *descida_vdd (ps)* *tx_fs* | *subida_out (ps)* *tx_ro* | *subida_vdd (ps )* *tx_rs* |
|---|---|---|---|---|
| **2** | 13,3118 | 15,1143 | 11,2578 | 12,3465 |
| **3** | | | | |
| **4** | | | | |
| **5** | | | | |
| **6** | | | | |

3. [0,5 pt] Explicar o impacto do número de transistores em série no plano P na porta NOR no **tempo de propagação de subida**.

4. [0,5 pt] O **tempo de propagação de subida** é mais afetado quando a entrada que varia está próxima de vcc ou da saída? Explicar a razão.

5. [1 pt] Na NOR2, o maior tempo de propagação ocorre na descida da saída provocada pela comutação de **i1** (t2_fs, entrada próxima a vcc). Justificar por que a descida por **i1** é mais lenta que a descida por **i2**, considerando o estado do nó interno da pilha P em cada caso. Em seguida, aumentar em 20% a largura do transistor N controlado por i1 (sinal s1 no subcircuito)

```
M10 0   s1  o1  0   nch_mac w='wn*1.2'  l=30n
```

Preencher a tabela abaixo, comparar com a questão 2, identificar o tempo que mais variou e explicar o resultado.

| N# Entradas NOR | *descida_out (ps)* *t2_fo* | *descida_vdd (ps)* *t2_fs* | *subida_out (ps)* *t2_ro* | *subida_vdd (ps )* *t2_rs* |
|---|---|---|---|---|
| **2** | | | | |

**MATERIAL PARA CONSULTA:**

**Livro: RABAEY, 2ª ED, capítulo 6, Seções 6.1 e 6.2.**
**WESTE: 4.4.1 Logical Effort:**

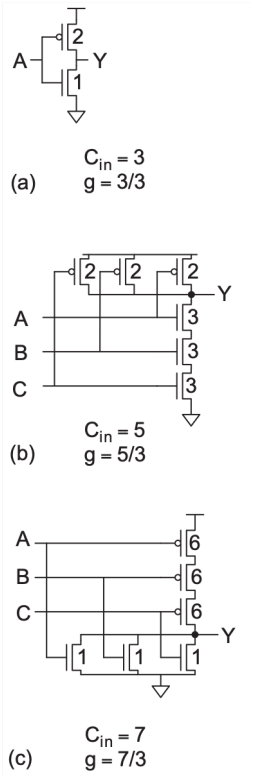
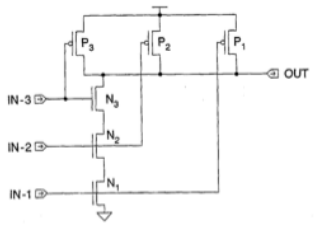
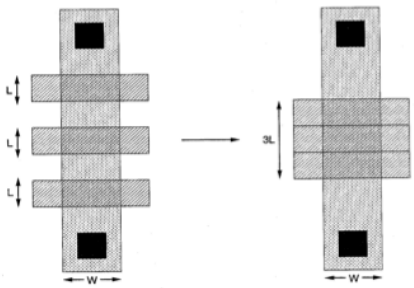



**FIGURE 4.22** Logic gates sized for unit resistance

## Transistors in Series: CMOS NAND




- Several devices in series each with effective channel length $L_{eff}$ can be viewed as a single device of channel length equal to the combined channel lengths of the separate series devices
  - e.g. 3 input NAND: a single device of channel length equal to $3L_{eff}$ could be used to model the behavior of three series devices each with $L_{eff}$ channel length, assuming there is no skew in the increasing gate voltage of the three N pull-down devices.
  - The source/drain junctions between the three devices essentially are assumed as simple zero resistance connections
  - During saturation transient, the bottom two devices will be in their linear region and only the top device will be pinched off.

R. W. Knepper
SC571, page 4-26

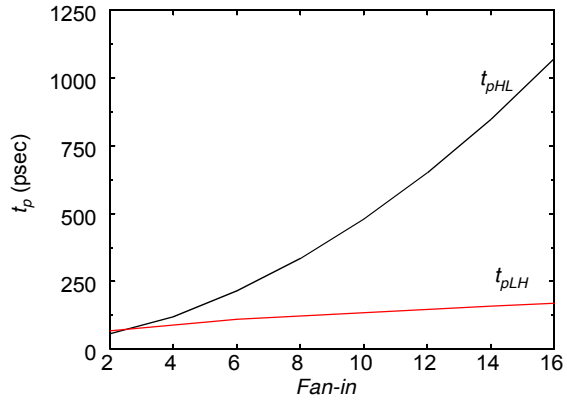



**Figure 6.13** Propagation delay of CMOS NAND gate as a function of fan-in. A fan-out of one inverter is assumed, and all pull-down transistors are minimal size.

**1. Transistor Sizing**

The most obvious solution is to increase the overall transistor size. This lowers the resistance of devices in series and lowers the time constant. However, increasing the transistor size, results in larger parasitic capacitors, which do not only affect the *propagation delay* of the gate in question, but also present a larger load to the preceding gate. This technique should, therefore, be used with caution. If the load capacitance is dominated by the intrinsic capacitance of the gate, widening the device only creates a "self-loading" effect, and the *propagation delay* is unaffected. A more comprehensive approach towards sizing transistors in complex CMOS gates is discussed in the next section.

**2. Progressive Transistor Sizing**

An alternate approach to uniform sizing (in which each transistor is scaled up uniformly), is to use progressive transistor sizing (Figure 6.14). Referring back to Eq. (6.3), we see that the resistance of $M_1$ ($R_1$) appears $N$ times in the delay equation, the resistance of $M_2$ ($R_2$) appears $N$-1 times, etc. From the equation, it is clear that $R_1$ should be made the smallest, $R_2$ the next smallest, etc. Consequently, a progressive scaling of the transistors is beneficial: $M_1 > M_2 > M_3 > M_N$. Basically, in this approach, the important resistance is reduced while reducing capacitance. For an excellent treatment on the optimal sizing of transistors in a complex network, we refer the interested reader to [Shoji88, pp. 131–143]. The reader should be aware of

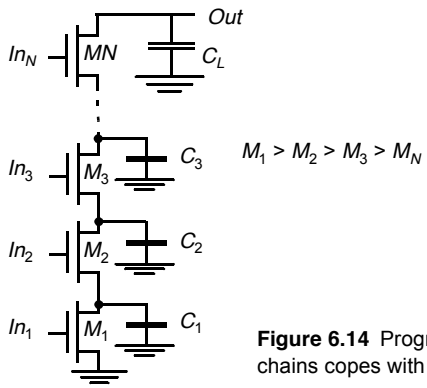



**Figure 6.14** Progressive sizing of transistors in large transistor chains copes with the extra load of internal capacitances.

one important pitfall of this approach. While progressive resizing of transistors is relatively easy in a schematic diagram, it is not as simple in a real layout. Very often, design-rule considerations force the designer to push the transistors apart, which causes the internal capacitance to grow. This may offset all the gains of the resizing!