LABORATÓRIO 3 - PORTAS LÓGICAS NAND E NOR

Prof. Fernando Gehm Moraes – Revisão: 12/maio/2025

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*.

Fazer download dos arquivos necessários ao laboratório

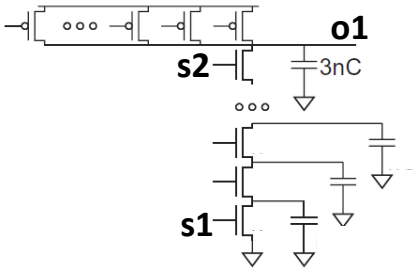
wget https://fgmoraes.github.io//microel/lab3/nand6.sp
wget https://fgmoraes.github.io//microel/lab3/nor6.sp
wget https://fgmoraes.github.io//microel/lab3/st65.scs

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 - 30/março/2025
- Linha **11**: processo 65nm
- Linha **14**: `.param wp=0.4 mob=2.45 cload=4fF`, onde:
  - `wp`: dimensão W_P em μm
  - `mob`: relação de mobilidade ($\mu n/\mu p$) para esta tecnologia
  - `cload`: carga de saída de cada porta NAND
- **Linhas 23-30:** exemplo de descrição *spice* da porta lógica NAND3. Transistores P em paralelo, e os transistores N em série.

```
.SUBCKT nand3 o1 s1 s2 s3 vcc
M1   o1 s1 vcc vcc psvtgp  w=wp          l=0.06
M2   o1 s2 vcc vcc psvtgp  w=wp          l=0.06
M3   o1 s3 vcc vcc psvtgp  w=wp          l=0.06
M10  0  s1 4   0   nsvtgp  w='wp*2/mob' l=0.06
M11  4  s3 2   0   nsvtgp  w='wp*2/mob' l=0.06
M12  2  s2 o1  0   nsvtgp  w='wp*2/mob' l=0.06
.ENDS nand3
```



  - **s2**: entrada que está mudando de estado próxima da **saída**
  - **s1**: entrada que está mudando de estado próximo de ***gnd***
  - as demais entradas (i3 a i6) ficam em 1, para os transistores N em série conduzirem

- Linha **86**: tensão de alimentação (`vcc vcc 0 dc 1.0`)
- linhas **87** e **88**: comandos PWL (geração dos estímulos para as entradas i1 e i2)
- linhas **89** a **92**: tensão fixa em '1.0' 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

  Os transistores P, na porta NAND2, por estarem em paralelo, possuem o mesmo dimensionamento. Neste exemplo: **wp=0.4** μm. Os transistores N, em **série**, devem ter a relação de mobilidade (***mob***) respeitada, devendo ter o dimensionamento de um inversor equivalente, seguindo o princípio do método *logic effort* (inverso da soma dos inversos).

Para a NAND2:

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

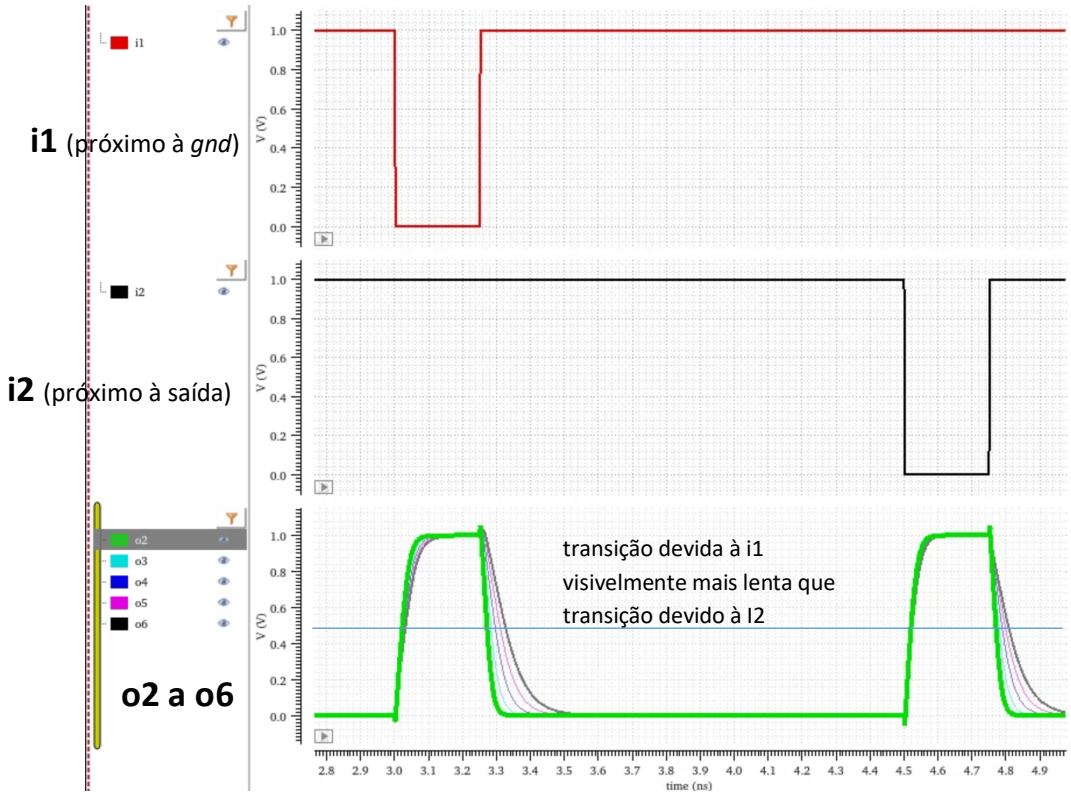
$$\frac{1}{{}^{1}/_{W_N} + {}^{1}/_{W_N}} = \frac{W_P}{mob}$$

$$W_N = \frac{W_P*2}{mob} = \frac{0{,}4*2}{2{,}45} = 0{,}326 \text{ (conforme acima: } \texttt{'wp*2/mob'}\text{)}$$

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 1.0   3n    1.0    3.003n 0    3.25n 0    3.253n 1.0)
v2 i2 0 pwl(0 1.0   4.5n  1.0    4.503n 0    4.75n 0    4.753n 1.0)
```



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_Ro*	*subida_gnd (ps)* *tx_Rs*
2	**19,0076**	**20,4095**	**19,4594**	**21,5395**
3				
4				
5				
6				

Observar: *t2_fo* ≈ *t2_ro* → coerente com o método *logic effort*

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

descida_gnd: t2_Fs (fall gnd - transição i1próximo 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, e no eixo Y o atraso em pico-segundos para as 4 curvas. 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]
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] Porque 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 sejam praticamente iguais. Preencha a tabela abaixo.

N# Entradas	*descida_out (ps)* *tx_Fo*	*descida_gnd (ps)* *tx_Fs*	*subida_out (ps)* *tx_Ro*	*subida_gnd (ps)* *tx_Rs*
2	**19,007**	**20,4099**	**19,4611**	**21,5457**
3				
4				
5				
6				

A diferença em ***descida_out*** deve ser ≈1.54 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)

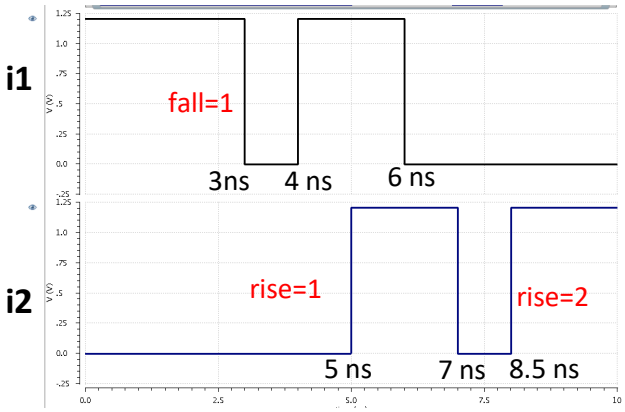
Simulação da porta NOR [4 pt]

- **Completar o *netlist* para simular portas NOR, de 2 a 6 entradas**. Utilizar o arquivo disponibilizado como modelo.

```
* circuitos nor
simulator lang=spectre insensitive=no
include "st65.scs"
simulator lang=spice

.param Cload=4fF mob=2.45  wn=0.3  wp='2*wn*mob'

.SUBCKT     nor2  o1    s1    s2    vcc
M1    10    s1    vcc   vcc   psvtgp      w=wp  l=0.06
M2    o1    s2    10    vcc   psvtgp      w=wp  l=0.06
M10   0     s1    o1    0     nsvtgp      w=wn  l=0.06
M11   0     s2    o1    0     nsvtgp      w=wn  l=0.06
.ENDS nor2

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

```
... 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 1.0
v1 i1 0 pwl(... completar ...)
v2 i2 0 pwl(... completar ...)
v3 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
```

entradas não utilizadas devem estar em 0 (zero)



lembrar que o slew (rampa) das entradas *i1* e *i2* deve ser 3ps.

```
.measure tran n2_subida_vdd  trig v(i1) val=0.5  td=2n fall = 1 targ v(o2) val=0.5  rise = 1
.measure tran n2_descida_vdd trig v(i1) val=0.5  td=2n rise = 1 targ v(o2) val=0.5  fall = 1
.measure tran n2_subida_out trig v(i2)  val=0.5  td=2n fall = 1 targ v(o2) val=0.5  rise = 2
.measure tran n2_descida_out trig v(i2) val=0.5  td=2n rise = 2 targ v(o2) val=0.5  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. [1 pt] Apresentar as formas de onda com as entradas (i1 e i2) e as saídas das **nor** 2 a 6 entradas (questão semelhante à 1 da simulação das portas *nand*)

2. [1 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	15,5434	20,7494	13,5092	16,1076
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] Para a NOR2, aumentar o tamanho do transistor **N** em **25%**, que corresponde ao chaveamento da entrada no plano P próximo ao VDD.

M10 0 s1 o1 0 nsvtgp w=**'wn*1.25'** l=0.06

N# Entradas NOR	*descida_out (ps)* *tx_fo*	*descida_vdd (ps)* *tx_fs*	*subida_out (ps)* *tx_ro*	*subida_vdd (ps)* *tx_rs*
2				

Compare com a questão 2, destacando o tempo que (mais) mudou. Explicar o resultado.

MATERIAL PARA CONSULTA:

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

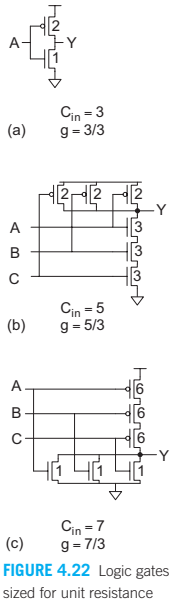
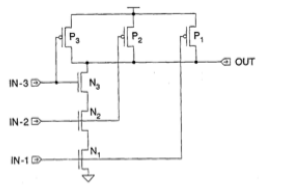
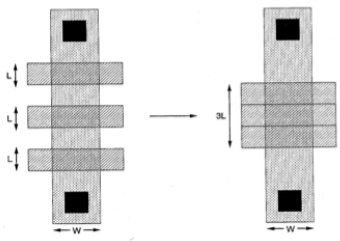


FIGURE 4.22 Logic gates sized for unit resistance





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

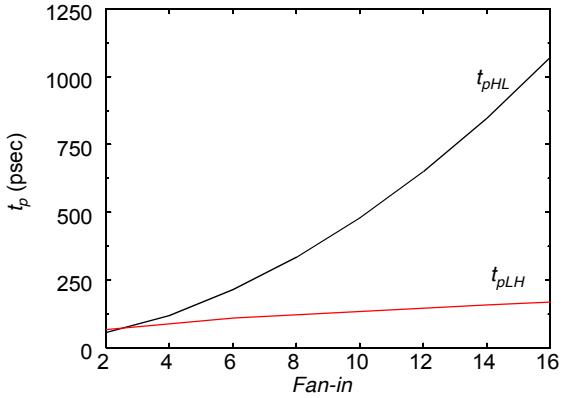


Figure 6.13 Propagation delay of CMOS NAND gate as a function of fan-in. A fan-out of one inverter is assumed, and all pull-down transistors are minimal size.

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

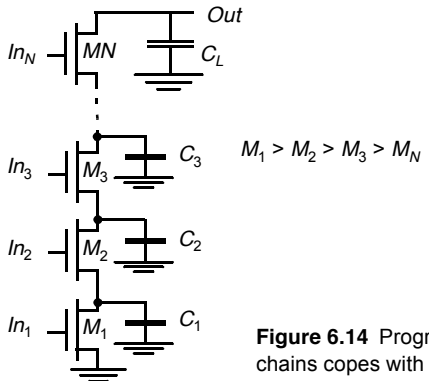


Figure 6.14 Progressive sizing of transistors in large transistor chains copes with the extra load of internal capacitances.

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!