

Microeletrônica: Introdução ao Projeto Físico de Portas Lógicas

Rafael Schivittz, Roberto Almeida, Giane Ulloa, Fábio Silva
Cristina Meinhardt, Paulo F. Butzen

- Aula 1 – Álgebra booleana → Tabela Verdade
→ Rede de chaves.
 - Introdução ao NGSPICE → Fontes DC e PWL
- Aula 2 – Transistores → Lógica Complementar
 - » (Pull-up/Pull-down)
- Aula 3 – Construção das portas lógicas complementares
 - » INV, NAND2, NOR2, AOI
- Aulas 4 – Atrasos
- Aulas 5 e 6 – Dimensionamento, Sub-circuitos.

- Aula 1 – Álgebra booleana → Tabela Verdade
→ Rede de chaves.
 - Introdução ao NGSPICE → Fontes DC e PWL
- Aula 2 – Transistores → Lógica Complementar
 - » (Pull-up/Pull-down)
- Aula 3 – Construção das portas lógicas complementares
 - » INV, NAND2, NOR2, AOI
- Aulas 4 – Atrasos
- Aulas 5 e 6 – Dimensionamento, Sub-circuitos.

- Construção de portas lógicas complementares
- Construção das ondas com PWL
- Arranjo de transistores
- Validação lógica
- Atrasos

- Construção de portas lógicas complementares
- Construção das ondas com PWL
- Arranjo de transistores
- Validação lógica
- Atrasos

Dúvidas??

NAND de 2 entradas – Chaves Complementares



Símbolo



Esquemático de Transistores

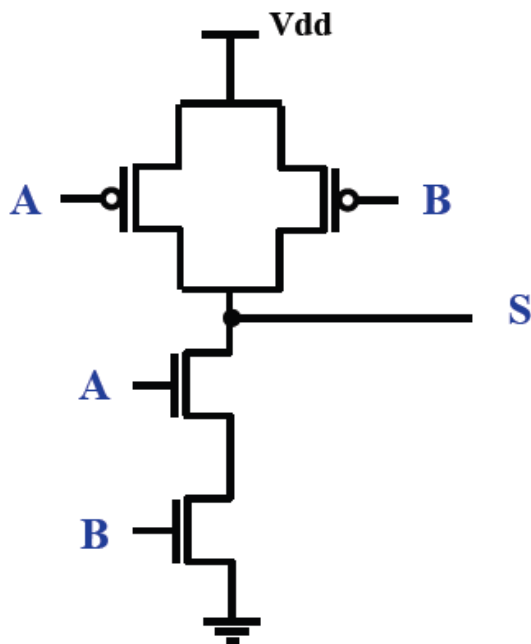
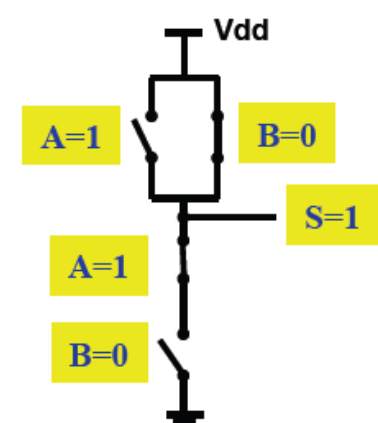
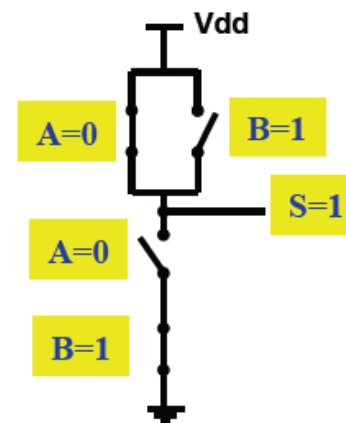
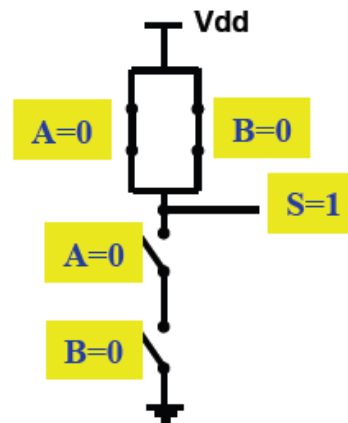
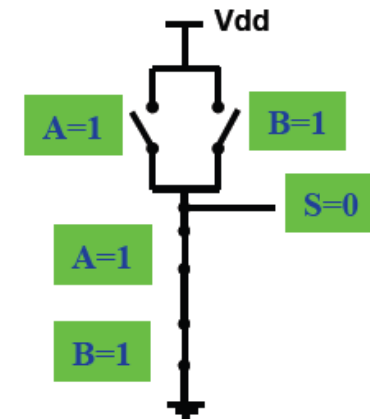


Tabela-Verdade

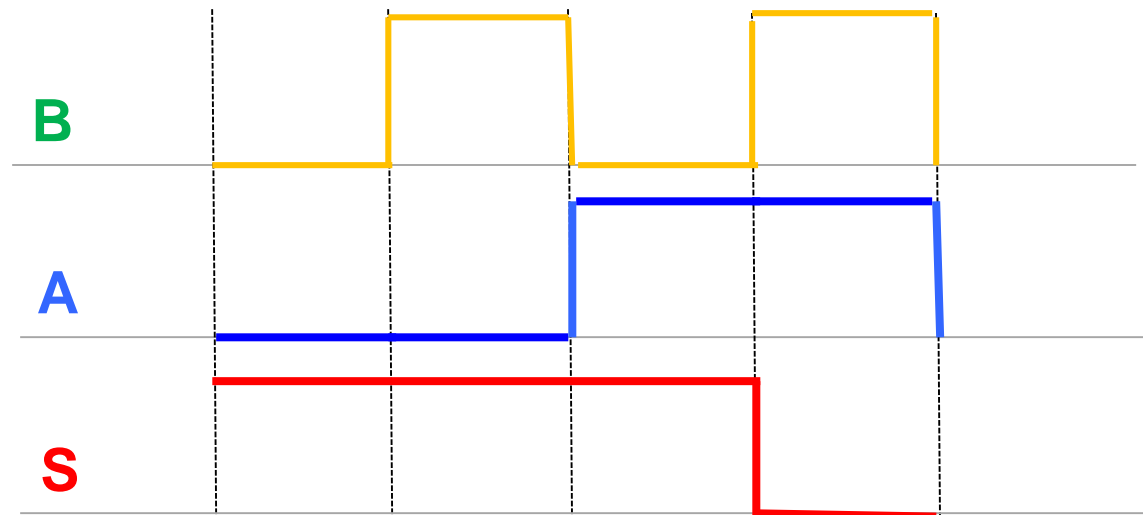
A	B	S
0	0	1
0	1	1
1	0	1
1	1	0



- Definir os sinais de entrada
 - 2 entradas, A e B
 - Testar todas as combinações possíveis
 - Sugestão: Definir sinais de entrada correspondentes a tabela verdade

Tabela-Verdade

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0



Validação Lógica da NAND2

```
.include 32nm_HP.pm
```

```
* Declaração das fontes
```

```
Vvdd vdd gnd 0.9
```

```
Vb
```

```
Va
```

```
*Declaração do Netlist
```

```
.....
```

```
*Tipo de Simulação
```

```
.tran 0.01n TEMPO_TOTAL
```

```
.end
```


Definir 2 fontes Pulse (B e A)



Vb b gnd pulse (.....)

2ns

B

Va a gnd pulse (.....)

A

Tr, Tf = 0.1ns

Definir 2 fontes Pulse (B e A)



Vb b gnd pulse (0 0.9 0 0.1n 0.1n 2n 4.2n)

2ns

B

Va a gnd pulse (0 0.9 0 0.1n 0.1n 4n 8.2n)

A

Tr, Tf = 0.1ns

Esquemático de Transistores

* Rede Pull up

Mpa s a vdd vdd PMOS W=100n L=32n

MPb s b vdd vdd PMOS W=100n L=32n

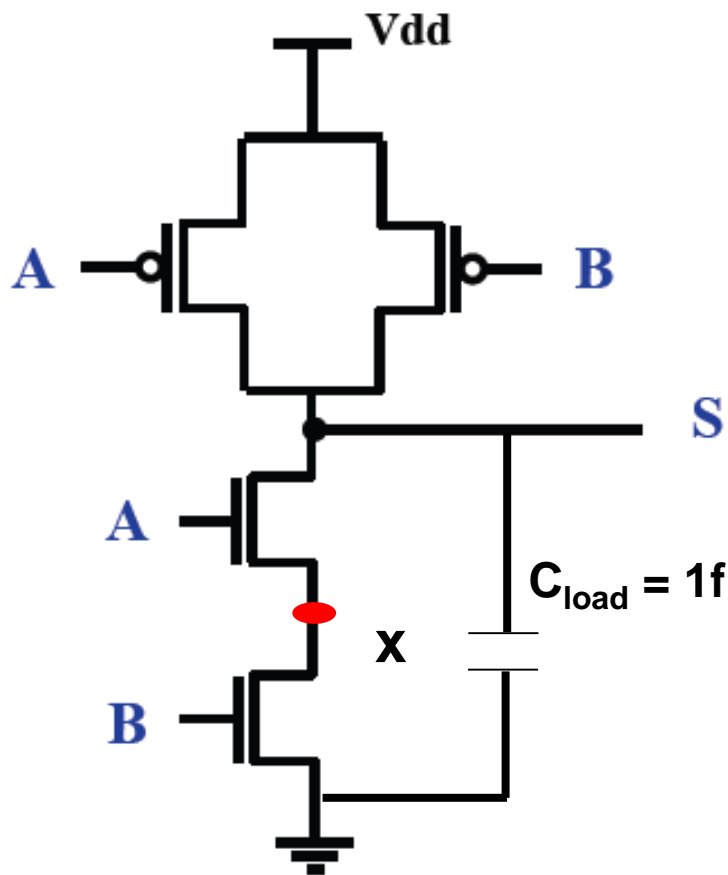
* Rede Pull down

Mna s a x gnd NMOS W=100n L=32n

Mnb x b gnd gnd NMOS W=100n L=32n

* Capacitância da saída

Cload s gnd 1f



- Verifique se seu circuito descrito em SPICE corresponde a uma NAND2

- Definir os atrasos da NAND2
- Medir a ENERGIA total consumida na simulação
 - Determinar a potência

- B deve estar estável
- A saída deve mudar
- Observar a tabela verdade

Tabela-Verdade

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0



**A não mudou
B mudou
Saída não mudou**

- B deve estar estável
- A saída deve mudar
- Observar a tabela verdade

Tabela-Verdade

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0



**A mudou
B mudou
Saída não mudou**

- B deve estar estável
- A saída deve mudar
- Observar a tabela verdade

Tabela-Verdade

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0



**A não mudou
B mudou
Saída mudou**

- B deve estar estável
- A saída deve mudar
- Observar a tabela verdade

Tabela-Verdade

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0




~~A mudou
B mudou
Saída mudou~~

- B deve estar estável
- A saída deve mudar
- Observar a tabela verdade

AB(S) -> AB(S)
01(1) -> 11(0) T_{pHL_a} , T_{fall_a}

Tabela-Verdade

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0



**A mudou
B não mudou
Saída mudou**



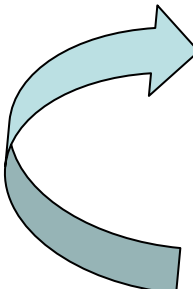
- B deve estar estável
- A saída deve mudar
- Observar a tabela verdade

AB(S) -> AB(S)

01(1) -> 11(0) TpHL_a, Tfall_a

11(0) -> 01(0) TpLH_a, Trise_a

Tabela-Verdade



A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

**A mudou
B não mudou
Saída mudou**



- A deve estar estável
- A saída deve mudar
- Observar a tabela verdade

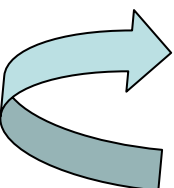
AB(S) -> AB(S)

01(1) -> 11(0) TpHL_a, Tfall_a

11(0) -> 01(0) TpLH_a, Trise_a

11(0) -> 10(0) TpLH_b, Trise_b

Tabela-Verdade



A	B	S
0	0	1
0	1	1
1	0	1
1	1	0


**A não mudou
B mudou
Saída mudou**



- A deve estar estável
- A saída deve mudar
- Observar a tabela verdade

Tabela-Verdade

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0



AB(S) -> AB(S)

01(1) -> 11(0) TpHL_a, Tfall_a

11(0) -> 01(1) TpLH_a, Trise_a

11(0) -> 10(1) TpLH_b, Trise_b

10(1) -> 11(0) TpHL_b, Tfall_b

**A não mudou
B mudou
Saída mudou**



- Temos:

- TpHL_a
- TpHL_b
- TpLH_a
- TpLH_b
- Trise_a
- Trise_b
- Tfall_a
- Tfall_b

- Como medi-los?

- Definir novos formatos para as ondas de entrada, possibilitando que todos os atrasos sejam medidos na simulação

AB(S) -> AB(S)

01(1) -> 11(0) TpHL_a, Tfall_a

11(0) -> 01(1) TpLH_a, Trise_a

11(0) -> 10(1) TpLH_b, Trise_b

10(1) -> 11(0) TpHL_b, Tfall_b

- Identificar os pontos onde acontecem cada um dos atrasos
- Definir as novas formas de onda usando fontes PWL
- Definir os comandos de medida para os 8 atrasos, usando o comando measure

Atrasos e corrente da NAND2

```
.include 32nm_HP.pm
```

```
* Declaração das fontes
```

```
Vvdd vdd gnd 0.9
```

```
Vb
```

```
Va
```

```
*Declaração do Netlist
```

```
.....
```

```
*Tipo de Simulação
```

```
.tran 0.01n TEMPO_TOTAL
```

```
* Measures
```

```
.end
```


AB(S) -> AB(S)

01(1) -> 11(0) TpHL_a, Tfall_a

11(0) -> 01(1) TpLH_a, Trise_a

11(0) -> 10(1) TpLH_b, Trise_b

10(1) -> 11(0) TpHL_b, Tfall_b

PWL ???

- Fonte PWL – Piece-Wise Linear
 - Fonte moldada conforme for declarada
 - V<nome> <n1> <n2> PWL (<t0> <v0> <t1> <v1>
<t2> <v2> <t3> <v3> ... <tn> <vn>)

- Definir as formas de onda que permitam medir os 8 atrasos da porta NAND2
- Definir os comandos para medir os 4 tempos de propagação da porta NAND2
- Simular e anotar os resultados

	Tp_hl_a	Tp_lh_a	Tp_hl_b	Tp_lh_b	Máx
NAND2	25ps	11.9ps	26ps	13.7ps	Tp_hl_b = 26ps

$$\text{ps} = 10^{-12}$$

$$E = 3.34 \cdot 10^{-15}$$

- Potência = Energia * V / tempo
- Energia = Integral da Corrente ao longo do tempo

$$E = \int_{t_0}^{t_f} I(fonte) dt$$

$$P = \frac{E * V}{\Delta t}$$

- .measure tran Name INTEG i(fonte) from= t₀ to=t_f

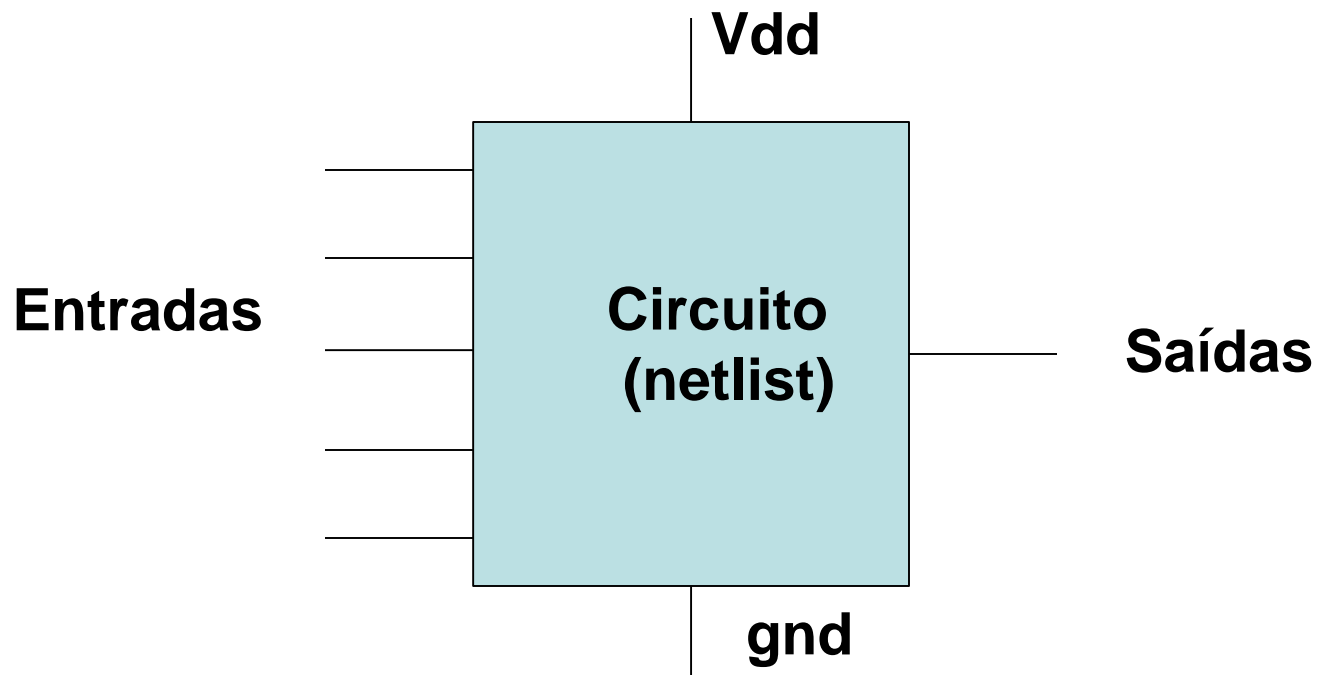
Exemplo:

- .measure tran Iint INTEG i(vvdd) from=0n to=12n

- Para equalizar o tempo de resposta das redes pull- up e pull-down de um circuito, é necessário dimensionar os transistores
- Transistores PMOS são mais “lentos”
- Dimensionamento envolve aumentar a largura (Width) dos transistores para aumentar a sua velocidade

NAND2_WP_Wn	Tp_hl_a	Tp_lh_a	Tp_hl_b	Tp_lh_b	Máx	Energia
NAND2_100_100						
NAND2_200_100						
NAND2_140_70						
NAND2_100_200						

- Tornam a descrição mais clara
- Permitem a reutilização de componentes (instanciação)



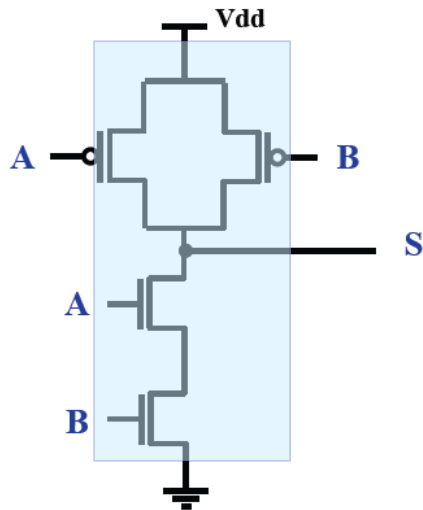
- Declaração

```
.SUBCKT subnam N1 <N2 N3 . . . >  
    netlist do subcircuito  
.ENDS <SUBNAM>
```

- Chamada (instanciação)

```
X1 N1 N2 N3 subnam  
X2 A B C subnam
```


Exemplo – NAND2



```
. Subckt NAND2 In1 In2 V G Out
```

```
  Mpa out in1 v v PMOS W=100n L=32n
```

```
  MPb out in2 v v PMOS W=100n L=32n
```

```
  Mna out in1 x g NMOS W=100n L=32n
```

```
  Mnb  x in2 g g NMOS W=100n L=32n
```

```
.Ends NAND2
```

Instanciação:

```
X1 A B Vdd Gnd S NAND2
```

```
Cload S gnd 1f
```

- Substituir no arquivo de validação lógica o netlist da NAND2 pela utilização de subcircuito e instanciação

Validação Lógica da NAND2

```
.include 32nm_HP.pm
```

```
*Declaração Subcircuitos
```

```
.subckt NAND2 in1 in2 v g s
```

```
*****
```

```
.ends NAND2
```

```
* Declaração das fontes
```

```
*Declaração do Netlist
```

```
X1 a b vdd gnd s NAND2
```

```
Cload s gnd 1f
```

```
*Tipo de Simulação
```

```
.tran 0.01n TEMPO_TOTAL
```

```
.end
```