



Universidade de Brasília
Departamento da Engenharia Elétrica
Laboratório de Sistemas Digitais

Relatório 01

Circuitos no LogicSim

José Antônio Alcântara da Silva de Andrade Mat: 232013031

Professor:
Eduardo B. R. F. Paiva
Turma 08

Brasília, DF
29 de outubro de 2024

1 Objetivos

1. Familiarização com circuitos elétricos
2. Simulação de circuitos digitais

2 Atividades

Essa sessão do laboratório foi realizada objetivando resolver quatro exercícios de construção de circuitos.

2.1 Exercício 1

O exercício 1 é bem simples em execução, requisitando a implementação da porta lógica OR usando-se apenas AND e NOT, e da porta AND usando-se apenas OR e NOT. Para tal, usa-se o Teorema de DeMorgan:

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

Para a implementação da porta AND, usa-se de uma porta OR recebendo o sinal inverso de A e de B, finalmente invertendo-se o resultado, como visto na figura 1:

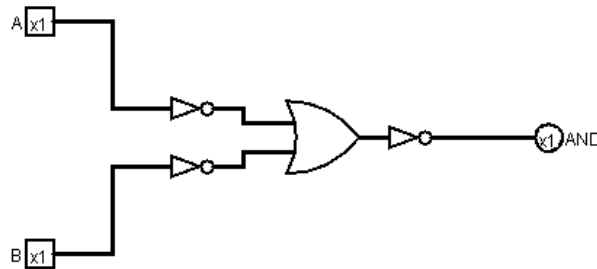


Figura 1: AND implementado com OR e NOT

A inversão da entrada do sinal causa um espelhamento da tabela verdade na vertical, e a inversão da saída altera os sinais finais, como segue:

A	B	\overline{A}	\overline{B}	$A + B$	$\overline{A} + \overline{B}$	$A \cdot B$
0	0	1	1	0	1	0
0	1	1	0	1	1	0
1	0	0	1	1	1	0
1	1	0	0	1	0	1

Como visto na figura 2, o circuito é exatamente o mesmo para a implementação da porta OR, realizando-se apenas a troca da porta OR do circuito anterior pela porta AND (como requisitado do exercício).

Novamente, inicia-se com a inversão dos sinais, e termina-se com a inversão da saída, efetivamente simulando o Teorema de DeMorgan:

A	B	\overline{A}	\overline{B}	$A \cdot B$	$\overline{A} \cdot \overline{B}$	$A + B$
0	0	1	1	0	1	0
0	1	1	0	0	0	1
1	0	0	1	0	0	1
1	1	0	0	1	0	1

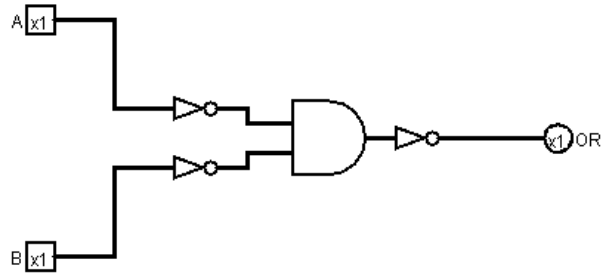


Figura 2: OR implementado com AND e NOT

2.2 Exercício 2

O segundo exercício é dividido em dois itens, ambos com objetivo de simular um circuito lógico usando apenas as portas AND, OR e NOT.

2.2.1 Item A

O item A requer a simulação do sistema lógico $C_{out} = AB + AC_{in} + BC_{in}$. Para tal, realiza-se uma operação AND com cada par de inputs (A , B e C_{in}) e finalmente realiza-se uma operação OR com as saídas destes AND. A tabela verdade resultante desse circuito é equivalente à tabela verdade da equação. O circuito resultante está representado na figura 3, a seguir:

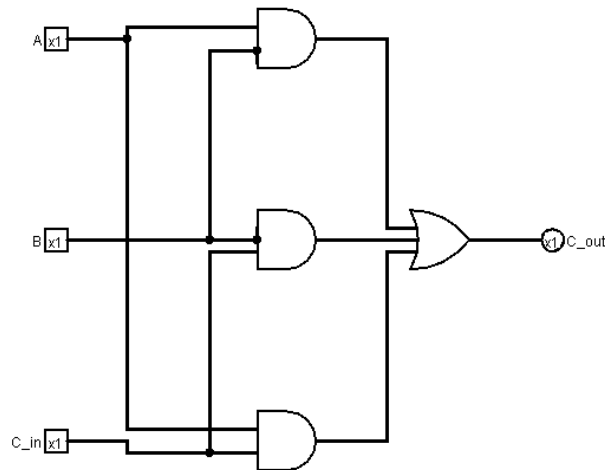


Figura 3: C_{out}

ABC_{in}	C_{out}
000	0
001	0
010	0
011	1
100	0
101	1
110	1
111	1

2.2.2 Item B

O item B é similar em processo, simulando o circuito $S = \overline{A}BC_{in} + \overline{A}B\overline{C_{in}} + A\overline{B}C_{in} + ABC_{in}$. É possível reduzir o circuito ao colocar A e \overline{A} em evidência:

$$S = \overline{A}(\overline{BC_{in}} + \overline{BC_{in}}) + A(\overline{BC_{in}} + BC_{in})$$

Daqui, é possível notar uma relação de similaridade entre os circuitos. Especificamente, o circuito $BC_{in} + \overline{BC_{in}}$ tem como inverso o circuito $\overline{BC_{in}} + BC_{in}$, demonstrado pelo teorema de DeMorgan.

$$\begin{aligned} \overline{BC_{in} + \overline{BC_{in}}} &= \overline{BC_{in}} \cdot \overline{\overline{BC_{in}}} \\ &= (\overline{B} + \overline{C_{in}}) \cdot (B + C_{in}) \\ &= \overline{B}B + \overline{B}C_{in} + \overline{C_{in}}B + \overline{C_{in}}C_{in} \\ &= 0 + \overline{B}C_{in} + \overline{B}C_{in} + 0 \\ &= \overline{B}C_{in} + \overline{B}C_{in} \end{aligned}$$

Então, pode-se construir o circuito $BC_{in} + \overline{BC_{in}}$ primeiramente, e usar sua saída com a entrada A em portas AND para terminar o circuito, como demonstrado na figura 4.

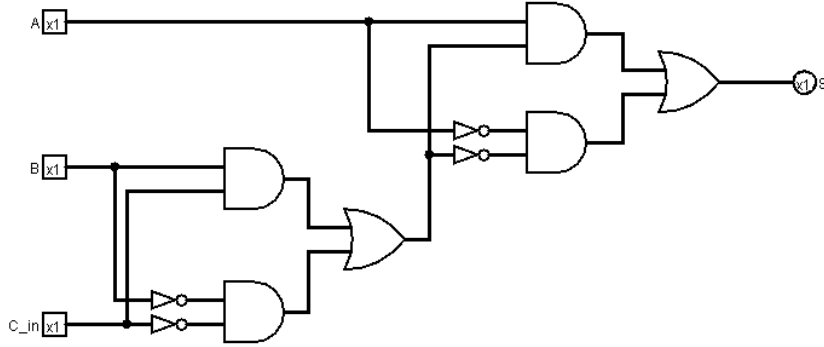


Figura 4: S construído de forma modular

A tabela verdade do circuito também segue. Uma observação sobre o sistema: as entradas B e C estão numa relação XOR, e a entrada A inverte o sinal da saída. Ainda mais, esse sistema pode ser visualizado como um encadeamento de duas portas XOR.

ABC_{in}	S	$A \oplus (B \oplus C_{in})$
000	0	0
001	1	1
010	1	1
011	0	0
100	1	1
101	0	0
110	0	0
111	1	1

2.3 Exercício 3

O exercício 3 consiste também em dois itens, os mesmos do exercício 2. Agora, contudo, o circuito deve ser feito usando apenas portas NAND. Para tal, usa-se algumas adaptações, a fim de obter outras portas lógicas usando apenas NANDs, como visto na figura 5.

2.3.1 Item A

Para o item A, primeiro troca-se as portas AND por NAND com a saída invertida (veja 5a) e depois realiza-se uma operação com a porta OR. Contudo, após a troca dessas portas, nota-se

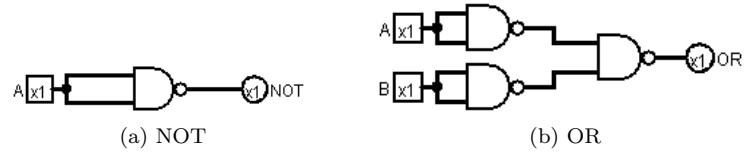


Figura 5: Portas NOT e OR implementadas com NANDs

uma redundância de inversões de sinais, as quais podem ser removidas. Finalmente, chega-se ao circuito da figura 6.

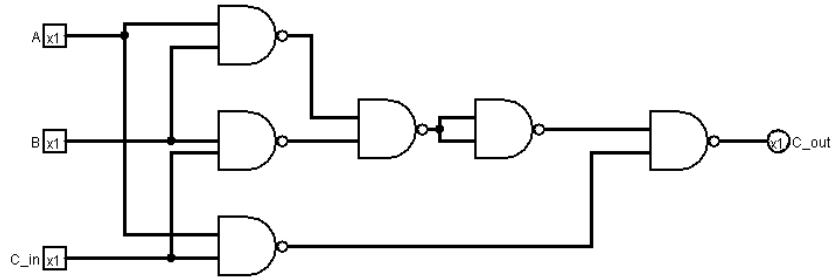


Figura 6: C_{in} construído com NANDs

A tabela verdade do circuito segue:

ABC_{in}	C_{out}
000	0
001	0
010	0
011	1
100	0
101	1
110	1
111	1

2.3.2 Item B

Para o item B, a lógica segue a mesma do item 2A. Basta primeiramente programar o circuito XOR entre B e C , e reutilizá-lo com a saída deste e a entrada A . A implementação segue na figura 7.

Novamente, em todo encontro de duas portas inversoras, estas foram retiradas visto a redundância. A tabela verdade do circuito segue:

ABC_{in}	S
000	0
001	1
010	1
011	0
100	1
101	0
110	0
111	1

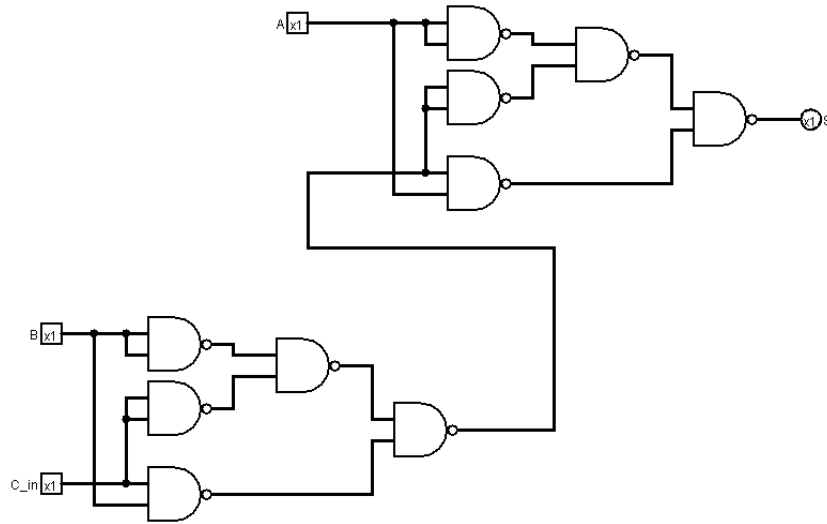


Figura 7: S construído com NANDs

2.4 Exercício 4

O exercício 4 encarrega o trabalho de construir (novamente usando apenas NANDs) um circuito lógico Y tal que $Y = D_0\bar{S}_0\bar{S}_1 + D_1S_0\bar{S}_1 + D_2\bar{S}_0S_1 + D_3S_0S_1$.

O circuito, então, foi construído primeiramente recebendo ambas entradas S_0 e S_1 , processando-as para 4 saídas diferentes (das quais apenas uma está ativa em cada das quatro possibilidades de entrada). Em seguida, associa-se esta saída com a entrada D equivalente por um simples AND, e, finalmente, aplica-se um OR nas quatro saídas para obter-se a saída final. O circuito final está representado na figura 8.

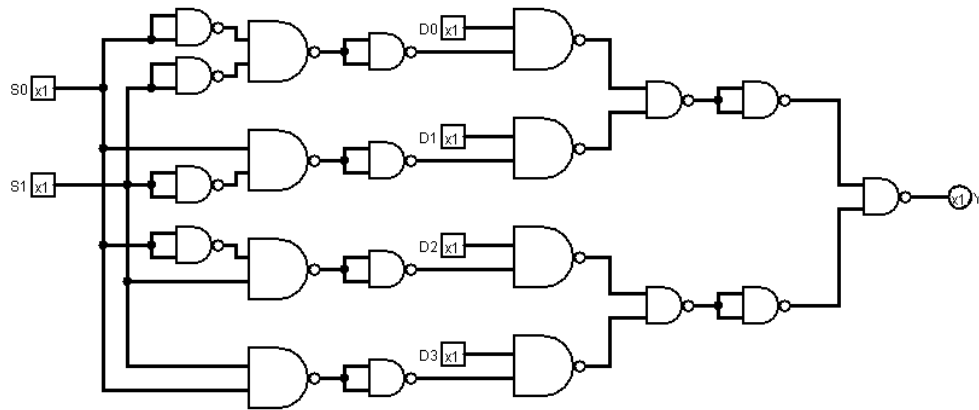


Figura 8: Y construído com NANDs

A tabela verdade de 6 variáveis é demasiadamente complexa para adicionar ao relatório. Contudo, como o circuito se trata de um multiplexador, existe uma forma de simplificar a tabela e ainda representar toda a informação necessária:

S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

Quando a entrada for 00, teremos que a saída do sistema será o valor de D_0 . Quando for 01, será D_1 , e assim por diante. Ou seja, S_0 e S_1 determinam qual dos D terá o valor transmitido para Y . Assim, nota-se que apenas os valores S e o valor D equivalente influenciam na saída

Y , enquanto os outros restantes se tornam irrelevantes.

3 Conclusão

As implementações dos circuitos muito facilita na visualização dos efeitos de portas lógicas, além de introduzir alguns conceitos ainda não vistos e familiarizar sistemas comuns, como o somador (implementado na figura 3 e 6) e o multiplexador (implementado na figura 8).