

Engenharia da Computação

Matemática Computacional

Lógica e Aritmética Binária

Aula 2

Prof. Luis Gonzaga de Paulo

Conversa inicial

Olá! Vamos dar início a mais uma aula de **Matemática Computacional!**

Nesta aula, abordaremos as operações lógicas binárias, que são realizadas pelos circuitos eletrônicos mais básicos dos computadores. São elas: Not (Não ou negação), And (E), Or (Ou), Xor (Ou Exclusivo) e, finalmente, Shift (Deslocamento). Também conheceremos as operações aritméticas binárias de adição, subtração, multiplicação e divisão. Essas operações são básicas e de grande importância, pois através delas são construídas todas as demais operações realizadas pelos computadores.

Vamos começar?

Para saber mais sobre o que será estudado na aula de hoje, acesse o material *on-line* e assista ao vídeo de introdução que o professor Luis Gonzaga preparou para você!

Contextualizando

Você já parou para pensar em como as operações de um computador são executadas somente no sistema binário? Se as informações armazenadas e manipuladas pelos computadores são compostas de bits, e os circuitos eletrônicos de um computador operam somente com os dígitos zero e um, então é possível que todos os cálculos computacionais, por mais complexos que sejam, sejam realizados com a lógica binária? Responder a essas e outras questões de grande importância para o entendimento do ambiente computacional é a nossa missão nesta aula.

Veja o que o professor Luis Gonzaga tem a dizer sobre isso no vídeo que está disponível no material *on-line*.

Antes de começar, que tal aproveitar os recursos à disposição e inteirar-se do que se trata a aula? Leia o capítulo 3 do livro da bibliografia básica -

TANENBAUM, A. S. **Organização estruturada de computadores**. 5ª ed. São Paulo: Editora Pearson, 2007 – disponível na Biblioteca Virtual, que versa sobre portas lógicas e aritmética binária.

Lógica binária

Um computador digital dos tempos atuais é um infindável aglomerado de circuitos eletrônicos interligados, cuja ultra veloz combinação é capaz de realizar as mais complexas operações matemáticas por meio da programação. Tais circuitos são denominados “portas lógicas”, pois são capacitados a realizar operações binárias básicas, que compõem o que se denomina **lógica binária**. Essa lógica binária vem a ser, então, o conjunto de operações realizadas no nível de bits, fundamentada em comparações sobre o estado de uma ou mais situações - ou entradas das portas lógicas, que alteram ou determinam uma nova situação, estado ou um novo valor – a saída das portas lógicas. A seguir, veremos quais são as operações lógicas.

Not

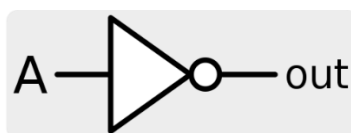
É a operação de **negação** (Não) ou **inversão**, também chamada de **complemento**. Utiliza somente um operador: um único bit. Como funcionamento do circuito eletrônico – porta lógica, apresenta na saída o inverso do valor representado na sua entrada. Essa porta lógica é denominada circuito ou porta lógica NOT, e nas expressões lógicas é representada por um til “~” ou por uma barra horizontal “-” colocada sobre o valor negado inverso, por exemplo \bar{A} , que se lê “A barra” ou “A barrado” ou “A negado”. A tabela verdade, quer seja a representação dos estados de entrada e saída da porta lógica NOT, é a seguinte:

Figura 1 – Tabela Verdade NOT

A	Out
0	1
1	0

Já a representação da porta lógica que realiza essa operação é a seguinte:

Figura 2 – Porta lógica NOT



Importante:

A porta lógica NOT é comumente associada às entradas e/ou saídas das demais portas lógicas, quando então é representada apenas por um pequeno círculo.

And

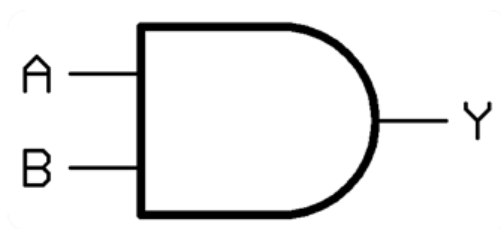
É a operação de conjunção binária, isto é, uma operação semelhante à multiplicação. Essa operação, como resultado, apresenta um valor “1” na saída da porta lógica quando todos os operandos – entradas da porta lógica – apresentam um valor “1”. Nos demais estados, a saída é “0”. Esta operação é realizada pelo circuito ou porta lógica AND, que é representada pelo símbolo “&”, chamado “e comercial” ou, em inglês, *ampersand*. A tabela verdade para essa operação é:

Figura 3 – Tabela verdade AND

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

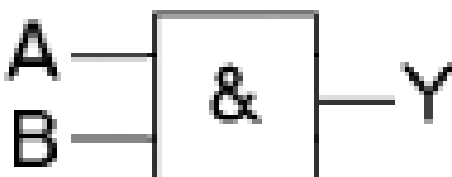
E a porta lógica que realiza essa operação é representada como:

Figura 4 - Porta lógica AND



Ou, também, deste outro modo (padrão IEEE):

Figura 5 - Porta lógica AND (padrão IEEE)



Or

É a operação de disjunção binária, que é semelhante à soma ou adição. Nesta operação, há um valor “1” na saída quando qualquer dos operandos ou entradas da porta lógica tem valor “1”. A operação OR é assinalada nas expressões lógicas com o sinal de adição, o “+”. As figuras a seguir representam o circuito ou porta lógica OR:

Figura 6 - Porta lógica OR

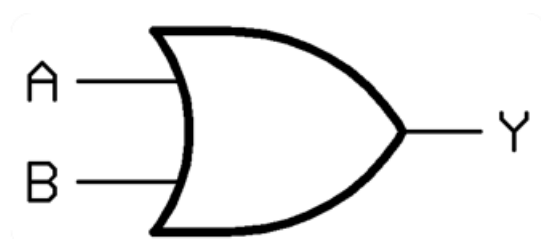


Figura 7 - Porta lógica OR (padrão IEEE)

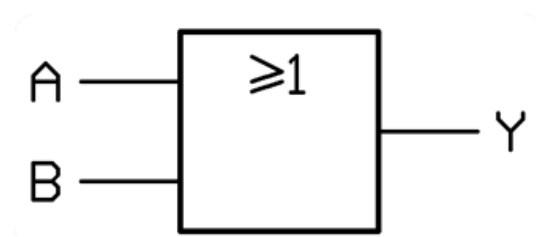


Tabela verdade:

Figura 8 - Tabela verdade OR

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Xor

Esta é a operação lógica binária de disjunção exclusiva, a qual tem por finalidade detectar uma desigualdade entre os valores apresentados na entrada. Somente apresenta um valor “1” na saída quando os operandos (entradas) têm valores diferentes. Corresponde ao circuito ou porta lógica XOR, e é representada pelo sinal de adição contornado por um círculo: \oplus

A porta lógica correspondente é representada das seguintes formas:

Figura 9 - Porta lógica XOR

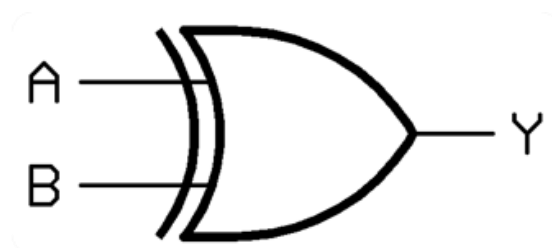


Figura 10 - Porta lógica XOR (padrão IEEE)

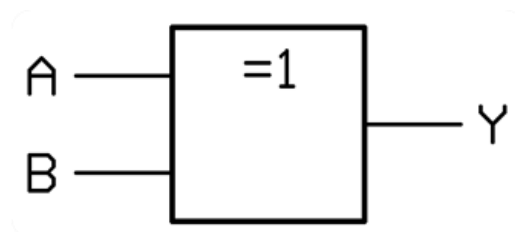


Tabela verdade:

Figura 11 - Tabela verdade XOR (padrão IEEE)

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Shift

O deslocamento, ou rotação de bits, é uma operação binária que efetua a divisão ou multiplicação por 2 e seus múltiplos. Em um número binário, deslocando-se os bits da direita para a esquerda em uma posição, multiplica-se o operador por 2. Sendo o deslocamento da esquerda para a direita, divide-se o operador por 2. Nos computadores, o circuito eletrônico responsável por tal operação é chamado **registrador de deslocamento**. A operação é demonstrada no exemplo da figura abaixo:

Figura 12 - Deslocamento de bits

Em binário	0	1	0	0	1	1	0	1	<<	1	0	0	1	1	0	1	0
Em decimal					77				x2				154				
Em binário	0	1	0	0	1	1	0	1	>>	0	0	1	0	0	1	1	0
Em decimal					77				/2				38				

Importante:

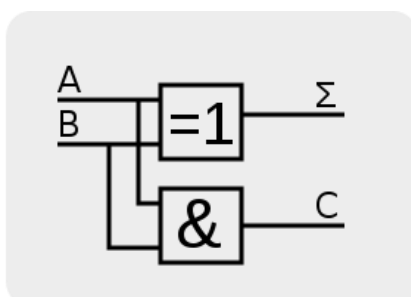
Os bits nas posições deslocadas são preenchidos com “0” e a operação de divisão só apresenta resultados inteiros.

Para tirar as dúvidas, assista ao vídeo que está disponível no material *on-line!*

Adição binária

É uma operação semelhante à soma decimal, incluindo o “vai-um” – neste caso, denominado *Carry Out* - com a diferença a favor de utilizar apenas os dígitos zero (0) e um (1) do sistema de numeração binário. A adição binária é a operação típica de uma porta lógica “OR”, que, por isso mesmo, é representada pelo sinal de adição “+” nas expressões lógicas. Na composição do computador, um circuito somador simples, denominado meio somador, emprega duas portas lógicas (OR e AND) para executar a soma de um único bit, como mostrado na figura a seguir:

Figura 13 - Circuito meio somador



As regras para a operação de adição binária são simples, a saber:

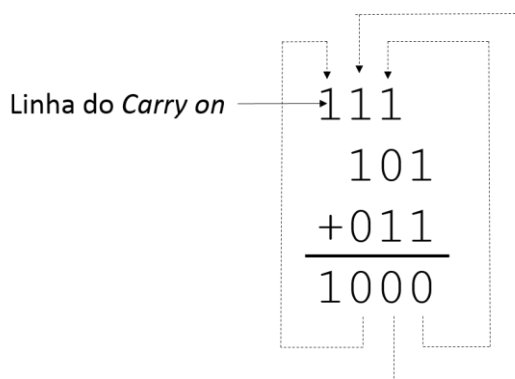
$0 + 0 = 0$
 $0 + 1 = 1$
 $1 + 0 = 1$
 $1 + 1 = 0$ e “vai um”: *Carry Out* = 1
 $1 + 1 + 1 = 1$ e “vai um”: *Carry Out* = 1

Os resultados obtidos nesta operação – nas saídas do circuito mostrado – são apresentados a seguir, na tabela verdade da operação.

Figura 14 - Tabela verdade da adição binária

A	B	Σ	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

Exemplo: $101 + 011$:



Como no exemplo, não utilizamos a soma binária para dois bits apenas: as somas são realizadas com diversos valores. Então vamos exercitá-la!

Exercício 1

Somar os valores **decimais** 65 e 68.

"Vai um"→	1	1
	65	1000001
	+ 68	+ 1000100
	<hr/>	<hr/>
	133	10000101

Primeiro faz-se a conversão para o sistema binário e, em seguida, soma-se bit a bit, observando as regras apresentadas.

Exercício 2

Somar os valores **octais** 3705 e 3717.

"Vai um"→	1 1	111 1111
	3705	11111000101
	+ 3717	+ 11111001111
	<hr/>	<hr/>
	7624	111110010100

Da mesma forma, primeiro faz-se a conversão para o sistema binário e, em seguida, soma-se bit a bit, observando as regras apresentadas.

Exercício 3

Somar os valores **hexadecimais** 89F e 457.

"Vai um"→	1	1111
	89F	10001001111
	+ 457	+ 1000101011
	<hr/>	<hr/>
	CF6	110011110110

Seguem-se as mesmas etapas: conversão e soma, bit a bit, de acordo com as regras.

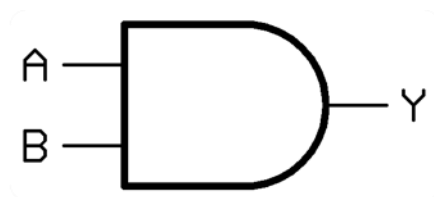
Para fixar esses conceitos e dominar o assunto, acesse a lista de exercícios no AVA e resolva-os. Avalie com seus colegas de turma as dificuldades encontradas e compartilhe os resultados obtidos.

Aproveite e acesse o material *on-line* para mais um vídeo do professor Luis Gonzaga!

Multiplicação binária

A multiplicação binária segue o mesmo modelo da multiplicação decimal, porém com a utilização apenas nos dois dígitos, zero e um. É uma operação típica de uma porta “AND” mostrada na figura a seguir:

Figura 15 - Porta AND



Para estruturar a operação, recomenda-se que o número maior seja colocado acima do número menor. Assim a operação fica mais simples. A tabela verdade da operação é a mesma da porta AND, como segue:

Figura 16 - Tabela verdade da multiplicação binária

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

As regras para a operação de subtração binária são simples, a saber:

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

Os resultados de cada multiplicação de um bit são colocados em linhas seguidas, deslocando-se um dígito para a esquerda, como mostrado no exemplo a seguir:

$\begin{array}{r} 6 \\ + 5 \\ \hline 30 \end{array}$	$\begin{array}{r} 110 \\ 101 \\ \hline 110 \\ + 000 \\ \hline 110 \\ \hline 11110 \end{array}$
--	--

Exercício 1

Realizar em modo binário a operação decimal 22 x 7:

$\begin{array}{r} 22 \\ \times 7 \\ \hline 154 \end{array}$	$\begin{array}{r} 10110 \\ 111 \\ \hline 10110 \\ + 10110 \\ \hline 10110 \\ \hline 10011010 \end{array}$
---	---

Quer saber mais? Então acesse o material *on-line* e assista ao vídeo que está disponível para você!

Subtração binária

A subtração binária segue o mesmo processo da subtração decimal, incluindo o “pede emprestado” para o dígito de maior valor (à esquerda), quando

se subtrai um valor maior de um valor menor, isto é, 1 (um) de 0 (zero). É a operação típica de uma porta XOR e também pode ser realizada através da “soma com complemento de base”.

A tabela verdade da operação de subtração binária é a seguinte:

Figura 17 - Tabela verdade da subtração binária

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

As regras para a operação de subtração binária são:

- $0 - 0 = 0$
- $0 - 1 = 1$ e “pede emprestado” 1 ao dígito da esquerda
- $1 - 0 = 1$
- $1 - 1 = 0$

Exemplo: realizar a subtração 5 - 3 em binário:

$$\begin{array}{r}
 101 \\
 - 011 \\
 \hline
 010
 \end{array}$$

“empresta” um →

Exercício 1

Realizar a subtração 68 – 65 em binário.

$$\begin{array}{r} 68 \\ - 65 \\ \hline 3 \end{array} \qquad \begin{array}{r} 1000100 \\ 1000001 \\ \hline 11 \end{array}$$

Exercício 2

Realizar a subtração 22 - 7 em binário.

$$\begin{array}{r} 22 \\ - 7 \\ \hline 15 \end{array} \qquad \begin{array}{r} 10110 \\ 111 \\ \hline 1111 \end{array}$$

No material *on-line*, você encontrará um vídeo com mais explicações sobre isso. Acesse!

Divisão binária

O processo de divisão binária segue o mesmo modelo da divisão decimal, operando com os mesmos quatro elementos:

- O dividendo, que é o valor a ser dividido;
- O divisor, que é o número que deve estar contido **n** vezes no dividendo;
- O quociente – valor de **n** que se procura, isto é, o resultado da divisão;
- O resto, caso a divisão não seja inteira, isto é: dividendo – (divisor * quociente). **O resto sempre deve ser menor que o quociente.**

Exemplo: realizar a seguinte divisão: 42/ 6

101010/110

$$\begin{array}{r}
 101010 \overline{)110} \\
 \underline{-110} \quad 111 \\
 1001 \\
 \underline{-110} \\
 0110 \\
 \underline{110} \\
 000
 \end{array}$$

Resultado: $42 / 6 = 7$

Exercício 1

Realizar a divisão decimal $37 / 4$ usando o sistema binário.

$$\begin{array}{r}
 100101 \overline{)100} \\
 \underline{100} \quad 1001 \\
 0101 \\
 \underline{-100} \\
 1
 \end{array}$$

Resultado: $37 / 4 = 9$, resta 1.

Note, neste caso, dois aspectos importantes:

1. A divisão não é inteira, isto é, há um resto;
2. Depois da primeira operação, quando “100” – dividendo – é dividido por “100” – divisor, ao fazer a descida do valor “1” do dividendo, ainda não se tem um valor que possa ser dividido por “100”. Então, adicionamos um “0” ao quociente e descemos mais um dígito. No caso é “0”, o que faz ser ainda insuficiente o valor, pois “010” é menor que “100”. Então repetimos este passo: acrescentamos um zero ao quociente e descemos mais um dígito, formando assim o valor 0101 que, desta forma, pode ser dividido por 100, restando então o valor 1.

Agora as explicações são por conta do professor Luis. Assista ao vídeo que está disponível no material *on-line*!

Trocando ideias

Vimos as operações com números inteiros e positivos. E como seriam as operações com os números que contém casas decimais? E valores negativos? Ou operações de subtração que resultam em números negativos? Leia o material complementar, procure na bibliografia, na internet, e discuta com os colegas de turma no fórum “Lógica e Aritmética Binária”. Acesse o Ambiente Virtual de Aprendizagem!

Na prática

Como vimos, as operações básicas de um computador são realizadas por circuitos relativamente simples. As operações mais complexas são subdivididas ou realizadas em etapas a partir dessas operações elementares, através de **circuitos combinacionais**. Como os computadores fazem essas operações à altíssimas velocidades, é possível realizar grandes quantidades destas operações em curto espaço de tempo.

Para assimilar o conteúdo apresentado e habilitar-se a executar os cálculos com agilidade, é necessário exercitar bastante. Por isso são disponibilizadas duas listas de exercícios sobre os assuntos abordados. Faça os exercícios e utilize o fórum da aula para compartilhar as experiências e as dúvidas com os colegas de turma. E faça uso do canal de Tutoria para esclarecer as dúvidas e superar as dificuldades.

Acesse o material *on-line* e assista ao vídeo que o professor Luis preparou para você!

Síntese

Muito bem! Chegamos ao final da segunda aula de Matemática Computacional!

Nesta aula estudamos as operações lógicas elementares que possibilitam o funcionamento dos computadores, a saber: NOT, AND, OR, XOR e SHIFT. Também estudamos como são realizadas as operações aritméticas básicas: Adição, Multiplicação, Subtração e Divisão com o sistema binário, para valores inteiros positivos.

Até a próxima!

Para as considerações finais do professor Luis, assista ao vídeo que está disponível no material *on-line*!

Referências

MACEDO, L. R. D., CASTANHEIRA, N. P., ROCHA, A. **Tópicos de Matemática Aplicada**. Curitiba: Intersaberes, 2013.

LEITE, A. E., CASTANHEIRA, N. P. **Teoria dos Números e Teoria dos Conjuntos**. Curitiba: Editora Intersaberes, 2014.

GUIMARÃES, C.H.C. **Sistemas de numeração**: aplicação em computadores digitais. Rio de Janeiro: Editora Interciência, 2014.

TANENBAUM, A. S., WETHERAL, D. **Redes de computadores**. 5ª ed. São Paulo: Pearson Educational, 2009.

TANENBAUM, A. S. **Sistemas operacionais modernos**. 3ª ed. São Paulo: Pearson Educational, 2013.

_____. **Organização estruturada de computadores**. 5ª ed. São Paulo: Editora Pearson, 2007.