IESAM – INSTITUTO DE ESTUDOS SUPERIORES DA AMAZÔNIA

CAPÍTULO 3
Aritmética Binária

3.1 – Introdução

No Capítulo 2 estudamos a conversão entre os sistemas decimal, binário, octal e hexadecimal. Nesta Unidade estudaremos as operações aritméticas de soma, subtração, multiplicação e divisão de binários, além de conceitos como complemento a 1 e a 2 e a sinalização dos números binários. Essas funções lógicas aritméticas constituem a Unidade Lógica e Aritmética (ULA) que é um bloco funcional fundamental em um microprocessador.

3.2 – Adição no Sistema Binário

Para efetuarmos a adição no sistema binário, devemos agir como numa adição convencional no sistema decimal, lembrando que, no sistema binário temos apenas dois algarismos. Temos então:

0 + 0 = 0, vai 0
1 + 0 = 1, vai 0
0 + 1 = 1, vai 0
1 + 1 = 0, vai 1

Convém observar que no sistema decimal 1 + 1 = 2 e no sistema binário é representado o número 2_{10} por 10_2.

Assim sendo: $1 + 1 = 10_2$.

Já temos aqui a primeira regra de transporte para a próxima coluna:

1 + 1 = 0 e transporta 1 (vai um)

<u>Exemplo 1</u>

Para exemplificar, vamos somar os números binários:

$11_2 + 10_2 =$

$$\text{“vai um”}\quad \begin{array}{cccc} & 1 & & \\ + & & 1 & 1 \\ & & 1 & 0 \\ \hline & 1 & 0 & 1 \end{array} \quad \Longrightarrow \quad 11_2 + 10_2 = 101_2$$

Exemplo 2

Some os seguintes binários: 110_2 e 111_2

$$
\begin{array}{rcccc}
 & 1 & 1 & & \\
\text{“vai um”} & & 1 & 1 & 0 \\
 & + & 1 & 1 & 1 \\
\hline
 & 1 & 1 & 0 & 1
\end{array}
\quad \Rightarrow \; 110_2 + 111_2 = 1101_2
$$

Exercícios Propostos

a) $1000_2 + 1001_2$
b) $10001_2 + 11110_2$
c) $101_2 + 100101_2$
d) $110_2 + 1001011_2$
e) $10101_2 + 1001001_2$

3.3 – Subtração no Sistema Binário

A subtração requer um pouco de atenção. Quando subtraímos números às vezes temos que fazer um empréstimo da próxima coluna à esquerda. Esse caso ocorre quando temos que subtrair 1 de 0. Observe as operações:

0 – 0 = 0, empresta 0
1 – 1 = 0, empresta 0
1 – 0 = 1, empresta 0
0 – 1 = 1, empresta 1

Exemplo 3

Subtraia os seguintes binários: 111_2 e 100_2

$$
\begin{array}{rccc}
 & 1 & 1 & 1 \\
- & 1 & 0 & 0 \\
\hline
 & 0 & 1 & 1
\end{array}
\qquad 111 - 100 = 011
$$

Exemplo 4

Subtraia os binários 1000 e 111.

Resolvendo por partes:

$$\begin{array}{rcccc} & 1 & 0 & 0 & 0 \\ - & & 1 & 1^{1} & 1 \\ \hline & & & & 1 \end{array}$$

0 –1 = 1
e empresta 1

$$\begin{array}{rcccc} & 1 & 0 & 0 & 0 \\ - & & 1^{1} & 1^{1} & 1 \\ \hline & & & 0 & 1 \end{array}$$

$\underline{0 – 1 = 1}$ - 1 = 0
empresta 1 → emprestado

$$\begin{array}{rcccc} & 1 & 0 & 0 & 0 \\ - & {}^{1} & 1^{1} & 1 & 1 \\ \hline & & 0 & 0 & 1 \end{array}$$

$\underline{0 – 1 = 1}$ – 1 = 0
empresta 1 → emprestado

$$\begin{array}{rcccc} & 1 & 0 & 0 & 0 \\ - & {}^{1} & 1 & 1 & 1 \\ \hline & 0 & 0 & 0 & 1 \end{array}$$

1 –1 = 0
↓
emprestado

$$1000_2 – 111_2 = 0001_2$$

Exercícios Propostos

a) 1100_2 - 1010_2
b) 10101_2 - 1110_2
c) 11110_2 - 1111_2
d) 1011001_2 - 11001_2
e) 100000_2 - 11100_2

3.4 – Multiplicação no Sistema Binário

As regras da multiplicação de binários são iguais às regras da multiplicação de decimais.

0 x 0 = 0
0 x 1 = 0
1 x 0 = 0
1 x 1 = 1

Exemplo 5

Multiplique os binários 11 e 11.

$$
\begin{array}{rcccc}
 & & & 1 & 1 \\
 & & x & 1 & 1 \\ \cline{3-5}
\text{Produtos Parciais} & & & 1 & 1 \\
 & + & 1 & 1 & \\ \cline{2-5}
 & 1 & 0 & 0 & 1
\end{array}
$$

Exemplo 6

Multiplique os binários 101 e 111.

$$
\begin{array}{rcccccc}
 & & & & 1 & 1 & 1 \\
 & & & x & 1 & 0 & 1 \\ \cline{2-7}
 & & & & 1 & 1 & 1 \\
\text{Produtos Parciais} & + & & 0 & 0 & 0 & \\
 & & 1 & 1 & 1 & & \\ \cline{2-7}
 & 1 & 0 & 0 & 0 & 1 & 1
\end{array}
$$

Exercícios Propostos

a) 1100_2 x 101_2
b) 10101_2 x 111_2
c) 11110_2 x 11_2
d) 1011001_2 x 110_2
e) 100000_2 x 10_2

3.5 – Divisão no Sistema Binário

A divisão é análoga à uma divisão de decimais, trabalhando com multiplicação e subtração.

Exemplo 7

Divida o binário 1100 por 10.

$$
\begin{array}{ccccc|ccc}
 & 1 & 1 & 0 & 0 & \angle & 1 & 0 \\
- & 1 & 0 & & & 1 & 1 & 0 \\ \cline{1-3}
 & 0 & 1 & 0 & & & & \\
 & - & 1 & 0 & & & & \\ \cline{2-4}
 & & 0 & 0 & & & &
\end{array}
$$

Exemplo 8

Divida os binários 110 e 11.

$$\begin{array}{rccccc} & 1 & 1 & 0 & \angle & 1\;1 \\ - & 1 & 1 & & & 1\;0 \\ \hline & 0 & 0 & 0 & & \end{array}$$

Exercícios Propostos

a) $11001_2 \div 111_2$
b) $10101_2 \div 11_2$
c) $1100_2 \div 10_2$
d) $11110_2 \div 111_2$
e) $100000_2 \div 10_2$

3.6 – Representação de Números Binários com Sinal

Antes de iniciarmos o assunto, vamos relembrar alguns tópicos importantes da subtração de dois números binários:

0 – 0 = 0
0 – 1 = 1 e empresta 1
1 – 0 = 1
1 – 1 = 0

Por ser diferente da adição, a subtração exigiria, em princípio, um circuito diferente, específico, para ser realizada. Mas se houver um jeito de representarmos números negativos em binário, a subtração seria transformada em uma simples adição, pois:

$$A - B = A + (-B)$$

Portanto, o problema deixa de ser o projeto de um circuito subtrator e passa a ser a representação de números negativos em binário.

Os computadores primitivos usavam o chamado “sistema de sinal-magnitude” para representar números binários com sinal. Nesta convenção o MSB era o bit do sinal e o resto da palavra era sempre o próprio valor absoluto do número; se o MSB=0, o número era positivo, e se o MSB=1, o número era negativo. Por exemplo:

$$+3_{10} = 0011_2$$

$$-3_{10} = \underbrace{1}_{\text{Bit de sinal}}\underbrace{011}_{\text{Bits de magnitude}}{}_2$$

Embora a representação do sistema sinal-magnitude seja direto, calculadoras e computadores não o utilizam normalmente porque a implementação em circuito é mais complexa do que em outros sistemas. O sistema mais amplamente usado para representação de números binários com sinal é o sistema de complemento a 2. Existem duas formas de se encontrar o complemento a 2 de um número binário: encontrando, primeiramente, o complemento a 1 e depois o complemento a 2 ou de forma direta encontrar logo o complemento a 2.

3.6.1 – Complemento a 1

O complemento a 1 de um número binário se faz simplesmente trocando os bits zeros por uns e os bits uns por zeros, veja o exemplo:

1	0	1	1	0	Número binário original
↓	↓	↓	↓	↓	
0	1	0	0	1	Complemento a 1 do binário original

3.6.2 – Complemento a 2

O complemento a 2 de um número binário é formado tomando-se o complemento a 1 do número e adicionando-se 1 na posição do bit menos significativo. Veja o exemplo:

	1	0	1	0	1	1	Equivalente binário do número 43.
	0	1	0	1	0	0	Complemento a 1 de 43.
+						1	Soma-se 1 para formar o complemento a 2.
	0	1	0	1	0	1	Complemento a 2 do binário 43.

Logo, diz-se que 010101 é a representação em complemento a 2 de 101011.

3.6.3 – Complemento a 2 de forma direta

Existe uma técnica muito simples que permite encontrarmos o complemento a 2 de um número binário sem a necessidade de se fazer os dois passos vistos anteriormente (complemento a 1 e soma com bit 1), a saber: efetuamos a leitura do da palavra binária qualquer da direita para a esquerda até encontrarmos o primeiro bit “1” e a seguir invertemos o valor lógico de todos os bits à esquerda do primeiro “1”. Vejamos um exemplo:

10110

Primeiro bit 1

← Sentido da leitura

Complemento a 2: 01010

3.6.4 – Representação de Números com Sinal Usando Complemento a 2

O sistema de complemento a 2 para representar números com sinal funciona do seguinte modo:

- Se o número é positivo, a magnitude é a forma binária direta e um bit de sinal 0 é colocado na frente do bit mais significativo, veja Figura 9.7.a;
- Se o número é negativo, a magnitude é representada na forma de seu complemento a 2, e um bit de sinal 1 é colocado na frente do bit mais significativo, veja Figura 9.7.b.

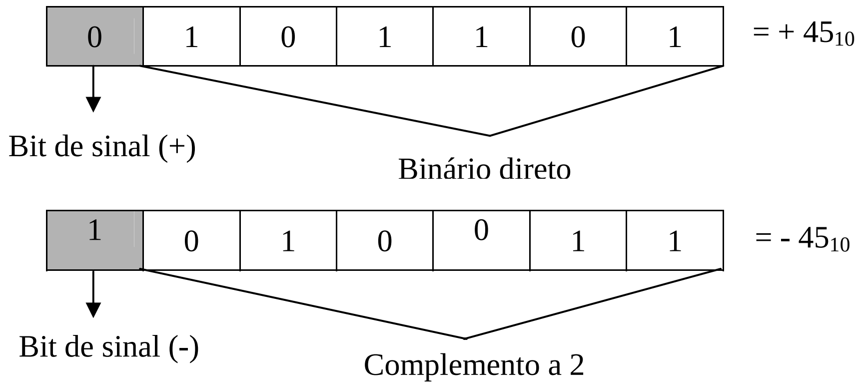

Figura 9.7 – Representação de números binários com sinal no complemento a 2.

O sistema de complemento a 2 é usado para representar números com sinal porque, conforme veremos, ele permite realizar a operação de subtração efetuando na verdade uma adição. Isto é importante, pois significa que um computador digital pode usar os mesmos circuitos tanto para somar como para subtrair, deste modo economizando em *hardware*.

3.7 – Exemplos de Adição e Subtração no Sistema de Complemento a 2

Caso I: Dois Números Positivos – A adição de dois números positivos é bastante direta e segue as regras de adição já vistas anteriormente. Considere a soma entre +10 e +5:

+10	→	0	1	0	1	0	(1ª parcela)
+5	→	0	0	1	0	1	(2ª parcela)
		0	1	1	1	1	(soma = +15)

Bits de sinal

- Os bits de sinal são iguais a zero, indicando que as parcelas e o resultado são positivos;
- As parcelas são escritas de modo a terem o mesmo número de bits, isto sempre deve ser feito no sistema de complemento a 2.

Caso II: Um Número Positivo e um Outro Menor e Negativo – Vamos considerar a operação entre +10 e –5.

Primeiramente, devemos encontrar o complemento a 2 de $+5_{10} = 0101_2$. Usando qualquer um dos métodos vistos anteriormente, encontramos que $-5_{10} = 1011_2$.

+10 →		0	1	0	1	0	(1ª parcela)
–5 →		1	1	0	1	1	(2ª parcela)
	1	0	0	1	0	1	(soma = +5)

Este *carry* é desconsiderado.

Bits de sinal

- O bit de sinal da segunda parcela é igual a 1, indicando ser um número negativo;
- O resultado do bit de sinal é 0, indicando que o mesmo é positivo. O *carry* ("vai um") gerado na última posição da adição é sempre descartado. Observe que a operação de adição é feita, também, sobre os bits de sinal.

Caso III: Um Número Positivo e um Outro Maior e Negativo – Considera a operação entre –10 e +5.

Novamente, primeiro devemos encontrar o complemento a 2 do número negativo, isto é, $-10_{10} = 0110_2$.

–10 →	1	0	1	1	0	(1ª parcela)
+5 →	0	0	1	0	1	(2ª parcela)
	1	1	0	1	1	(soma = -5)

Bits de sinal

- Como era de se esperar, o bit de sinal do resultado é igual a 1, indicando resultado negativo;
- Como o resultado é negativo, ele está representado em complemento a 2, de modo que os últimos 4 bits (bits de magnitude), 1011, de fato representam o complemento a 2 do resultado. Para encontrar a magnitude verdadeira, basta encontrar o complemento a 2 de 1011, que é 0101 = +5. Logo, 11011 representa –5.

Caso IV: Dois Números Negativos – Considere a operação entre –10 e –5.

A operação de soma se dará entre os complementos a 2 de ambos os números:

–10 →		1	0	1	1	0	(1ª parcela)
–5 →		1	1	0	1	1	(2ª parcela)
	1	1	0	0	0	1	(soma = -15)

Este *carry* é desconsiderado.

Bits de sinal

- Como o resultado é negativo, a sua magnitude está na forma de complemento a 2. Portanto, devemos encontrar o complemento a 2 do resultado para encontrar a magnitude verdadeira. Logo, 11111 representa –15.

Caso V: Dois Números Iguais em Magnitude Mas de Sinais Contrários – Considere a operação entre +10 e –10.

–10	→	1	0	1	1	0	(1ª parcela)
+10	→	0	1	0	1	0	(2ª parcela)
	1	0	0	0	0	0	(soma = +0)

Este *carry* é desconsiderado.

Bits de sinal

- O resultado é, como esperado, +0.

3.8 – *Overflow* Aritmético

Considere a soma entre +10 e +7:

+10	→	0	1	0	1	0	(1ª parcela)
+7	→	0	0	1	1	1	(2ª parcela)
		1	0	0	0	1	

Magnitude incorreta

Sinal incorreto

A resposta tem um bit de sinal negativo, o que obviamente está errado visto que estamos somando números positivos. O erro é causado porque a magnitude (17) precisa de 5 e não 4 bits como as parcelas tinham e, portanto, ocorreu um *overflow* na posição do bit de sinal. Esta condição somente ocorre quando somamos dois números positivos ou dois números negativos, e ela sempre produz um resultado incorreto. A ocorrência de *overflow* pode ser detectada examinando-se o bit de sinal do resultado e comparando-o com os bits de sinal dos números que estão sendo adicionados. Nos computadores, um circuito especial é usado para detectar qualquer condição de *overflow* para indicar que a resposta está errada.

3.9 – Bibliografia

1 – Sistemas Digitais – *Princípios e Aplicações* –Tocci e Widmer – LTC Editora, 7ª edição;
2 – Elementos de Eletrônica Digital – Capuano e Idoeta – Editora Érica – 19ª Edição;
3 – Sistemas Digitais – Fundamentos e Aplicações – Thomas L. Floyd - Ed. Bookman – 9ª Edição;
4 – Capítulo III – Circuitos Digitais Combinacionais – Prof. F.C.C de Castro, PUCRS.