Representações Binárias

Circuitos Lógicos

Prof. Odilson Tadeu Valle

Instituto Federal de Santa Catarina – IFSC
Campus São José
odilson@ifsc.edu.br

Conteúdo programático

1 Sistemas de numeração posicionais

2 Conversão entre sistemas de numeração

3 Códigos

4 Códigos para números negativos

5 Representação de números reais
- Representação de ponto fixo
- Representação por ponto flutuante

Conteúdo programático

1 Sistemas de numeração posicionais

2 Conversão entre sistemas de numeração

3 Códigos

4 Códigos para números negativos

5 Representação de números reais
- Representação de ponto fixo
- Representação por ponto flutuante

Sistemas de numeração posicionais

O valor atribuído ao símbolo depende da posição que ele ocupa em relação ao conjunto.

Valor absoluto

O valor absoluto de um número é o valor em si. No número 953:

- $9 = 9$
- $5 = 5$
- $3 = 3$

Valor posicional

O valor relativo ou posicional de um número depende da posição em que o algarismo se encontra. No número 953:

- $9 = 900$ (9×10^2)
- $5 = 50$ (5×10^1)
- $3 = 3$ (3×10^0)

Odômetro

- Suponha que o odômetro da figura acima possua somente duas roldanas de algarismos e que cada algarismo represente exatamente 1 km.
- Qual a quantidade máxima de quilômetros que o suposto odômetro pode representar?

Odômetro

- Suponha que o odômetro da figura acima possua somente duas roldanas de algarismos e que cada algarismo represente exatamente 1 km.
- Qual a quantidade máxima de quilômetros que o suposto odômetro pode representar?

$$
\begin{array}{ccccc}
00 & 11 & \cdots & 91 & 01 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
09 & 19 & \cdots & 99 & 09 \\
10 & 20 & \cdots & 00 & 10
\end{array}
$$

Sistemas de numeração posicionais

- Agora suponha que cada roldana tenha impresso somente os valores 0 e 1.
- Qual a quantidade máxima de quilômetros que o suposto odômetro pode representar?

Sistemas de numeração posicionais

- Agora suponha que cada roldana tenha impresso somente os valores 0 e 1.
- Qual a quantidade máxima de quilômetros que o suposto odômetro pode representar?

00
01

Sistemas de numeração posicionais

- Agora suponha que cada roldana tenha impresso somente os valores 0 e 1.
- Qual a quantidade máxima de quilômetros que o suposto odômetro pode representar?

00
01
10
11

Sistemas de numeração posicionais

- Agora suponha que cada roldana tenha impresso somente os valores 0 e 1.
- Qual a quantidade máxima de quilômetros que o suposto odômetro pode representar?

00
01
10
11
00
01
⋮

- Este sistema de numeração é conhecido como binário ("que tem aspecto dual, ou é formado por dois elementos ou partes").

Sistemas de numeração posicionais

- Usando a mesma estrutura de duas roldanas, como conseguiríamos representar um valor maior que 99 km?

Sistemas de numeração posicionais

- Usando a mesma estrutura de duas roldanas, como conseguiríamos representar um valor maior que 99 km?
- Aumentado o número de algarismos disponíveis em cada roldana.
- Por exemplo, se adotarmos os seguintes algarismos: 0, 1, ..., 8, 9, A, B, C, D, E, F, pode-se ter a seguinte representação quilométrica:

Sistemas de numeração posicionais

- Usando a mesma estrutura de duas roldanas, como conseguiríamos representar um valor maior que 99 km?
- Aumentado o número de algarismos disponíveis em cada roldana.
- Por exemplo, se adotarmos os seguintes algarismos: 0, 1, ..., 8, 9, A, B, C, D, E, F, pode-se ter a seguinte representação quilométrica:

$$\begin{array}{cccccc} 00 & 11 & \cdots & 91 & A1 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \cdots \\ 09 & 19 & \cdots & 99 & A9 & \cdots \end{array}$$

Sistemas de numeração posicionais

- Usando a mesma estrutura de duas roldanas, como conseguiríamos representar um valor maior que 99 km?
- Aumentado o número de algarismos disponíveis em cada roldana.
- Por exemplo, se adotarmos os seguintes algarismos: 0, 1, ..., 8, 9, A, B, C, D, E, F, pode-se ter a seguinte representação quilométrica:

$$
\begin{array}{cccccc}
00 & 11 & \cdots & 91 & A1 & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \cdots \\
09 & 19 & \cdots & 99 & A9 & \cdots \\
0A & 1A & \cdots & 99 & A9 & \cdots \\
0B & 1B & \cdots & 99 & A9 & \cdots \\
0C & 1C & \cdots & 99 & A9 & \cdots \\
0D & 1D & \cdots & 99 & A9 & \cdots \\
0E & 1E & \cdots & 99 & A9 & \cdots \\
0F & 1F & \cdots & 99 & A9 & \cdots \\
10 & 20 & \cdots & A0 & B0 & \cdots
\end{array}
$$

- Esse é o sistema de numeração **Hexadecimal**.

Sistemas de numeração posicionais

- Os sistemas de numeração que nos interessam são:

  Decimal É o mais utilizado. Algarismos 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

  Binário É o mais utilizado em máquinas. Algarismos 0, 1.

  Hexadecimal Relação direta com o binário. Algarismos 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F,

  Octal Relação direta com o binário. Algarismos 0, 1, 2, 3, 4, 5, 6, 7.

- Nada impede que construamos outros sistemas de numeração a partir de outras bases, por exemplo, base 5: Algarismos 0, 1, 2, 3, 4.

Conteúdo programático

1 Sistemas de numeração posicionais

2 Conversão entre sistemas de numeração

3 Códigos

4 Códigos para números negativos

5 Representação de números reais
- Representação de ponto fixo
- Representação por ponto flutuante

Conversão entre sistemas de numeração

- Como é de conhecimento geral, o sistema de numeração mais utilizado por seres humanos é o sistema decimal.
- Não tão conhecido assim, mas muito utilizado, é o sistema de numeração binário, amplamente adotado nos sistema informatizados.
- Uma pergunta que cabe é, por exemplo, quando digitamos algum número em uma calculadora, o que acontece?

Conversão entre sistemas de numeração

- Como é de conhecimento geral, o sistema de numeração mais utilizado por seres humanos é o sistema decimal.
- Não tão conhecido assim, mas muito utilizado, é o sistema de numeração binário, amplamente adotado nos sistema informatizados.
- Uma pergunta que cabe é, por exemplo, quando digitamos algum número em uma calculadora, o que acontece?
  1. a calculadora apresenta o valor digitado no visor, para termos certeza do que digitamos.
  2. internamente à calculadora este valor é convertido para binário, o sistema de numeração que ela entende. Como ocorre a conversão?
  3. a operação solicitada pelo usuário é realizada através dos circuitos aritméticos, a serem estudados posteriormente.
  4. a calculadora converte o resultado para decimal
  5. apresenta o resultado na tela.

Conversão de qualquer base para a decimal

- A regra geral para conversão de um valor em qualquer base para o valor decimal é assim expressa:

$$y = \sum_{k=0}^{N-1} a_{N-1-k} \times B^{N-1-k}$$

  onde:
  - y é um valor decimal equivalente a ser obtido
  - k é a posição do algarismo no valor sob cálculo
  - N é a quantidade de algarismos do valor sob cálculo
  - a o algarismo em si
  - B é o valor da base, 2 para binário, 8 para octal e 16 para hexadecimal.
- Por exemplo, vamos converter 11001_2 para a base decimal: $11001_2 = 1{\cdot}2^4{+}1{\cdot}2^3{+}0{\cdot}2^2{+}0{\cdot}2^1{+}1{\cdot}2^0 = 1{\cdot}16{+}1{\cdot}8{+}0{\cdot}4{+}0{\cdot}2{+}1{\cdot}1 = 25$
- Outro exemplo, vamos converter 703_8 para a base decimal. $703_8 = 7 \cdot 8^2 + 0 \cdot 8^1 + 3 \cdot 8^0 = 7 \cdot 64 + 0 \cdot 8 + 3 \cdot 1 = 451$
- Outros exemplos....

Conversão da base decimal para outras bases

- A conversão de um número da base decimal para qualquer outra base pode ser efetivada usando-se divisões sucessivas do valor decimal pela base a ser convertido, tomando-se com resultado os sucessivos restos dessa divisão.
- Por exemplo, para converter 23 para a base binária devemos fazer o seguinte procedimento:

$$\begin{array}{r|l} 23 & 2 \\ \hline 1 & 11 \end{array} \longrightarrow 23 = 2 \times 11 + 1$$

$$\begin{array}{r|l} 11 & 2 \\ \hline 1 & 5 \end{array} \longrightarrow 23 = 2 \times (2 \times 5 + 1) + 1 = 5 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

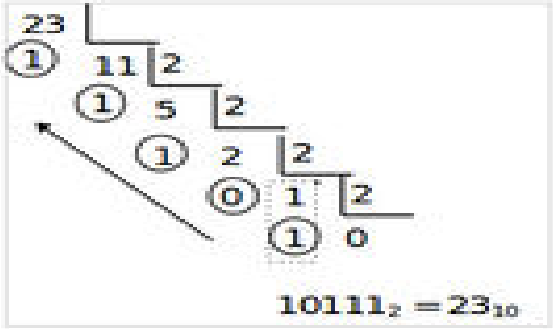
$$\begin{array}{r|l} 5 & 2 \\ \hline 1 & 2 \end{array} \longrightarrow 23 = (2 \times 2 + 1) \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

$$\begin{array}{r|l} 2 & 2 \\ \hline 0 & 1 \end{array} \longrightarrow 23 = (1 \times 2) \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

$$= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

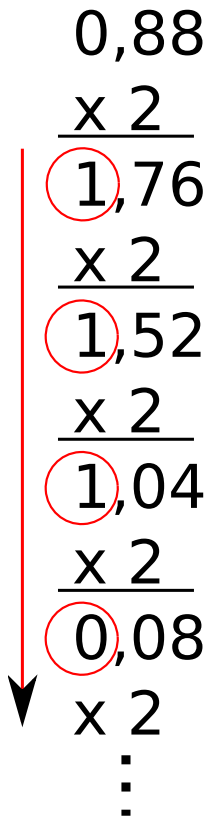
$$= 10111$$

Regra prática:



Conversão da base decimal para base binária com números fracionários

- Para números fracionários deve-se adotar dois procedimentos distintos, um para a parte inteira, já visto anteriormente, e outro para a parte fracionária.
- Exemplo: converter 23,88 para a base binária:
  1. Parte inteira (slide anterior): $23_{10} = 10111_2$
  2. Para fracionária. Ao contrário das divisões sucessivas pela base, aqui faz-se multiplicações sucessivas pela mesma e toma-se os bits, zerando-os, de cima para baixo, conforme figura. Nesse caso: $0,88_{10} = 0,1110..._2$
  3. É muito comum ocorrerem dízimas, portanto, deve-se delimitar o número de bits desejado, em nosso exemplo 4.
  4. Finalmente: $23,88_{10} = 10111,1110_2$, com 4 bits fracionários.



Conversão da base decimal para outras bases

- O procedimento para outras bases é o mesmo, por exemplo, para converter-se 23 para a base 8, utiliza-se o mesmo procedimento acima, substituindo os valores 2 por 8, que representam a mudança de base, no divisor.

Conversão entre bases de origem binária

- Para conversão entre as bases binária, octal e hexadecimal, basta fazer-se o uso das seguintes tabelas de conversão.

binário	octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

binário	hexadecimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Conversão entre bases de origem binária

- Por exemplo, para converter 730_8 para binário, deve-se tomar dígito a dígito da tabela acima e ir montando o valor binário equivalente: $7_8 = 111_2$, $3_8 = 011_2$ e $0_8 = 000_2$. Portanto o resultado da conversão é 111011000_2
- Outro exemplo, para converter 1110011_2 para octal, deve-se agrupar os bits em conjuntos de 3, a partir da direita (LSB), e tomar na tabela seus octais equivalentes: 1_2 ou $001_2 = 1_8$, $110_2 = 6_8$ e $011_2 = 3_8$
- Se desejarmos converter da base octal para a hexadecimal e vice-versa, a maneira mais fácil é primeiro a conversão da base de origem para a base binária e, em seguida, desta para a base destino. Por exemplo, para converter $FACA_{16}$ para a base octal procedemos primeiro a conversão para a base binária (1111101011001010_2) e em seguida, tomando três a três dígitos a partir do LSB, convertemos para o octal usando a tabela, resultando em 175312_8

Conteúdo programático

1 Sistemas de numeração posicionais

2 Conversão entre sistemas de numeração

3 Códigos

4 Códigos para números negativos

5 Representação de números reais
- Representação de ponto fixo
- Representação por ponto flutuante

Definição de códigos

Código

Código binário é a linguagem que determina as instruções para um computador. 1 bit (*binary digit* ou dígito binário em português) é a menor unidade de informação que pode ser armazenada ou partilhada e só pode ter 2 valores: 0 ou 1.

Código

Código é uma tabela que relaciona uma determinada grandeza, valor, caractere etc ao equivalente binário. Por exemplo, no código ASCII: A = 1000001, @ = 1000000.

Decifre a mensagem do código ASCII (www.tradutordebinario.com):
01101101011010010110110101101001011011010110100

Lista de códigos

Número decimal	Código binário conv.	Código octal	Código hexadec.	Código Gray	Código one-hot	Código Johnson	Código BCD
0	0000	00	0	0000	0000000000000001	00000000	0000
1	0001	01	1	0001	0000000000000010	10000000	0001
2	0010	02	2	0011	0000000000000100	11000000	0010
3	0011	03	3	0010	0000000000001000	11100000	0011
4	0100	04	4	0110	0000000000010000	11110000	0100
5	0101	05	5	0111	0000000000100000	11111000	0101
6	0110	06	6	0101	0000000001000000	11111100	0110
7	0111	07	7	0100	0000000010000000	11111110	0111
8	1000	10	8	1100	0000000100000000	11111111	1000
9	1001	11	9	1101	0000001000000000	01111111	1001
10	1010	12	A	1111	0000010000000000	00111111	0001 0000
11	1011	13	B	1110	0000100000000000	00011111	0001 0001
12	1100	14	C	1010	0001000000000000	00001111	0001 0010
13	1101	15	D	1011	0010000000000000	00000111	0001 0011
14	1110	16	E	1001	0100000000000000	00000011	0001 0100
15	1111	17	F	1000	1000000000000000	00000001	0001 0101

- Gray - Código de distância unitária: $\neq$ palavras adjacentes é 1 bit.
- One-hot - Utilizado na codificação de máquina de estados.
- Johnson - Utilizado na codificação de máquina de estados.
- BCD - *Binary-coded decimal*

ASCII - *American Standard Code for Information Interchange*

Código de 8 bits. Na figura são apresentados os primeiros 128 bits.

	$b_6b_5b_4$							
$b_3\ b_2\ b_1\ b_0$	000	001	010	011	100	101	110	111
0000	NULL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	“	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	‘	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

- Qual é a sequência de bits para a palavra Casa? (xxd -b teste.txt)

Conteúdo programático

1 Sistemas de numeração posicionais

2 Conversão entre sistemas de numeração

3 Códigos

4 Códigos para números negativos

5 Representação de números reais
- Representação de ponto fixo
- Representação por ponto flutuante

Números não sinalizados

- Na representação de números não sinalizados, os números são representados por um símbolo que identifica o sinal ($+$ ou $-$) e a sua grandeza. Ex:
  - $+$ 5 $=$ 5 $=$ 101
  - - 5 $=$ - 101
  - -7 $=$ - 111

Código sinal-magnitude

Como representar sinal no código binário?

Código sinal-magnitude

Como representar sinal no código binário?

- No código sinal magnitude o MSB representa o sinal: 0 = + e 1 = -.
- Ou seja, como o nome sugere a representação é composta por dois campos: sinal seguida por magnitude.
- Exemplos:
  - 0000 = +0
  - 1000 = -0
  - 110011 = -19
- Perceba que nessa representação é imprescindível a quantificação do número de bits e deixar explícito que trata-se de um número sinal-magnitude, para que não haja confusão entre os valores representados.

Código complemento de um

- Se o MSB for 0 significa que é número positivo.
- Para obter o equivalente negativo, basta inverter todos os bits. Ex:
  - 0000 = +0
  - 1111 = -0
  - 0111 = +7
  - 1000 = -7

Código complemento de dois

- Esta é a opção adotada para representar números negativos em praticamento todos os computadores e outros sistema digitais.
- A representação binária de um número negativo é obtida tomando sua representação positiva, invertendo todos os bits (complemento de um) e então adicionando um (1) a ele. Exemplo:
  - -7 (5 bits): +7 é igual a 00111 ==> 11000 + 1 = 11001

Com apresentação em 8 bits, determine a representação em complemento de dois para:

1. -0
2. -1
3. -123
4. -128

- Qual é a representação do +0 e -0?
- Como determinar o equivalente positivo de um número apresentado em complemento de dois? Faz-se seu complemento de dois.

Código complemento de dois

- Qual é o máximo valor positivo e o máximo negativo que se consegue representar com 3 bits?

Código complemento de dois

- Qual é o máximo valor positivo e o máximo negativo que se consegue representar com 3 bits?

Complemento de dois	Decimal
000	+0
001	+1
010	+2
011	+3
100	-4
101	-3
110	-2
111	-1

R: de $-2^{(n-1)}$ até $+2^{(n-1)} - 1$, onde n é o número de bits.

- Qual é o máximo valor positivo e o máximo negativo que se consegue representar com 8 bits?

Código complemento de dois: convertendo para decimal

- Para o caminho inverso, ou seja, para determinar o equivalente decimal a partir de uma representação de complemento de dois, deve-se tomar o respectivo valor, diminuir 1 e inverte-lo completamente.
- Exemplo:
  - 111001:

Código complemento de dois: convertendo para decimal

- Para o caminho inverso, ou seja, para determinar o equivalente decimal a partir de uma representação de complemento de dois, deve-se tomar o respectivo valor, diminuir 1 e inverte-lo completamente.
- Exemplo:
  - 111001: 111001 - 1 = 111000 ==> 000111 = + 7, ou seja, o valor inicial é equivalente a -7.
- Determine o equivalente decimal para as representações em complemento de dois abaixo:
  1. 10001
  2. 111101010
  3. 10001100
  4. 101111

Código complemento de dois: convertendo para decimal

- Regra prática – Para cálculo do equivalente decimal de um número representado em complemento de dois, pode-se fazer uso do modelo apresentado no exemplo:

1	0	1	1	0	1
-32	+0	+8	+4	+0	+1

- Ao somar-se todos os valores da segunda linha da tabela acima, obtém-se o valor -19.
- Determine o equivalente decimal para as representações em complemento de dois abaixo:
  1. 10001
  2. 111101010
  3. 10001100
  4. 101111

Conteúdo programático

1 Sistemas de numeração posicionais

2 Conversão entre sistemas de numeração

3 Códigos

4 Códigos para números negativos

5 Representação de números reais
- Representação de ponto fixo
- Representação por ponto flutuante

Representação de ponto fixo

- A representação segue a regra

$$S \ I \ F$$

onde:

S - sinal. 0 para positivo e 1 para negativo

I - parte inteira do valor. Representação binária tradicional. Obs.: deve-se estabelecer previamento a quatidade de bits para essa representação.

F - parte fracionária do valor. O expoente da base inicia em -1 e vai sendo decrementado. Idem observação acima.

- Por exemplo 011101,101:

0	1	1	1	0	1	1	0	1
+	1×2^4	1×2^3	1×2^2	0×2^1	1×2^0	1×2^{-1}	0×2^{-2}	1×2^{-3}
+	16	8	4	0	1	0,5	0	0,125

- Portanto o decimal equivalente é: 29,625
- Qual é a representação binária equivalente a 19,5?

Representação de ponto fixo

- A representação segue a regra

$$S \ I \ F$$

onde:

S - sinal. 0 para positivo e 1 para negativo

I - parte inteira do valor. Representação binária tradicional. Obs.: deve-se estabelecer previamento a quatidade de bits para essa representação.

F - parte fracionária do valor. O expoente da base inicia em -1 e vai sendo decrementado. Idem observação acima.

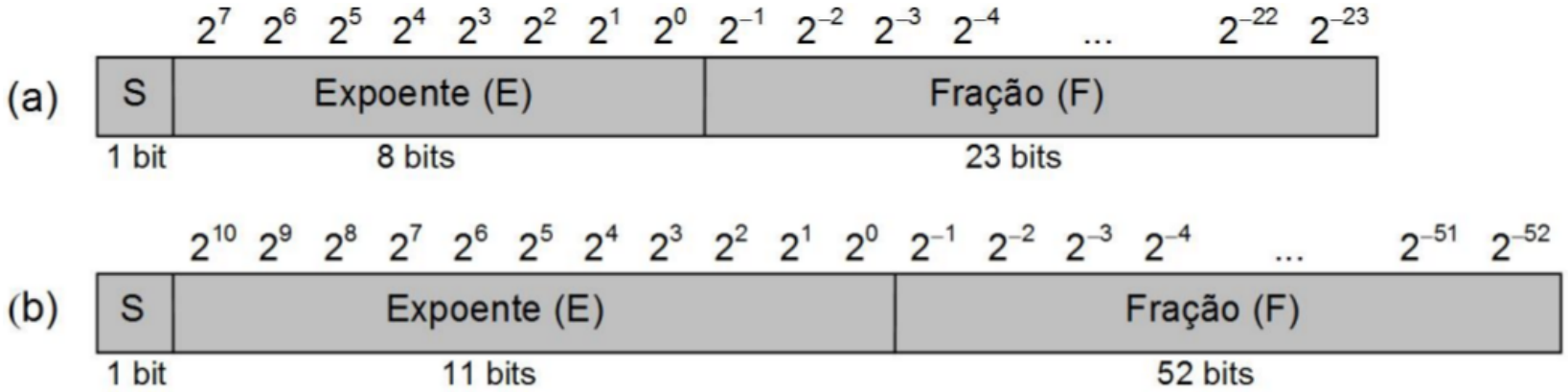
- Por exemplo 011101,101:

0	1	1	1	0	1	1	0	1
+	1×2^4	1×2^3	1×2^2	0×2^1	1×2^0	1×2^{-1}	0×2^{-2}	1×2^{-3}
+	16	8	4	0	1	0,5	0	0,125

- Portanto o decimal equivalente é: 29,625
- Qual é a representação binária equivalente a 19,5?: 10011,1.
- Como representar a vírgula na representação binária?

Representação por ponto flutuante

- Além do ponto fixo, o mais usual para representar números reais é empregar-se o padrão IEEE 754 – código para representar números com ponto flutuante.



  a) Ponto flutuante de precisão simples: 1 bit de sinal (S) + 8 bits para expoente (E) + 23 bits para fração (F)
  b) Ponto flutuante de precisão dupla: 1 bit de sinal (S) + 11 bits para expoente (E) + 52 bits para fração (F)

- Considera-se que é representado com a utilização de notação científica normalizada: exatamente 1 dígito diferente de zero antes da vírgula.

Precisão simples e dupla

- O número decimal equivalente é obtido por:
  - **precisão simples**: $y = (-1)^S \times (1 + F) \times 2^{E-127}$
  - **precisão dupla**: $y = (-1)^S \times (1 + F) \times 2^{E-1023}$

  E é o expoente deslocado e e é o expoente verdadeiro (pode ser negativo): $e = E - 127$ ($E = e + 127$) ou $e = E - 1023$ ($E = e + 1023$)
- Ex1: Encontre o valor decimal equivalente para o valor abaixo que está em ponto flutuante com precisão simples:

$$10111111110000000000000000000000$$

- Analisando o número acima obtemos: S = 1, E = 01111111 = 127, F = 10...0 = 1×2^{-1} = 0,5.
- Portanto

$$y = (-1)^1 \times (1 + 0,5) \times 2^{127-127} = -1,5$$

- Ex2: Encontre o valor decimal equivalente para o valor abaixo que está em ponto flutuante com precisão simples:

$$01000000101000000000000000000000$$

Representação por ponto flutuante - observações

- Na representação por ponto flutuante, a representação de números negativos é equivalente a representação sinal magnitude, ou seja, um número negativo tem exatamente os mesmos bits que sua contraparte positiva, com uma única diferença, que é o bit de sinal.
- As equações não produzem valor 0 (zero), portanto, uma representação especial se faz necessário para representar esse valor: S=0/1, E=0 e F=0 é equivalente a +0 e -0.
- Representação de infinito: S=0/1, E=max e F=0 é equivalente a $+\infty e - \infty$.
- Representação NaN (*not a number*) que são obtidas por números inválidos ou indeterminados (0/0, $\infty - \infty$ etc): S=0/1, E=0 e $F \neq 0$ é equivalente a NaN (erro).

Representação por ponto flutuante - exemplos

- Determinar a representação binária por ponto flutuante de precisão simples para -2,625.

Representação por ponto flutuante - exemplos

- Determinar a representação binária por ponto flutuante de precisão simples para -2,625.
  1. Encontrar o equivalente binário do número, conforme apresentado no slide 10: $-2,625_{10} = -10,101_2$
  2. Representar com notação científica normalizada: $-1,0101 \times 2^1$
  3. Extrair S, F e E: S=1, F=01010000000000000000000, E=128 (10000000), já que, $E = e + 127 ==> E = 1 + 127 ==> E = 128$.
  4. Portanto a sequência binária seria: 11000000001010000000000000000000.
- Determinar a representação binária por ponto flutuante de precisão simples para 0,75.

Representação por ponto flutuante - exemplos

- Determinar a representação binária por ponto flutuante de precisão simples para -2,625.
  1. Encontrar o equivalente binário do número, conforme apresentado no slide 10: $-2,625_{10} = -10,101_2$
  2. Representar com notação científica normalizada: $-1,0101 \times 2^1$
  3. Extrair S, F e E: S=1, F=01010000000000000000000, E=128 (10000000), já que, $E = e + 127 ==> E = 1 + 127 ==> E = 128$.
  4. Portanto a sequência binária seria: 11000000001010000000000000000000.
- Determinar a representação binária por ponto flutuante de precisão simples para 0,75.
  1. $0,75_{10} = 0,11_2 = 1,1 \times 2^{-1}$
  2. Portanto, S = 0, F = 10000000000000000000000 e E = 01111110 ($E = e + 127 = -1 + 127 = 126$)
  3. Portanto a sequência binária seria: 00111111010000000000000000000000.

Ponto flutuante versus inteiro

Ponto flutuante pode representar números muito grandes e números muito pequenos.

- Por exemplo, com os 32 bits pode-se representar:
  - de 2^{-126} até aproximadamente 2^{128} em ponto flutuante de representação simples
  - de 2^{-31} até $2^{+31} - 1$ na representação inteira, portanto, uma faixa numérica bem menor.
- Qual o preço a pagar?

Ponto flutuante versus inteiro

Ponto flutuante pode representar números muito grandes e números muito pequenos.

- Por exemplo, com os 32 bits pode-se representar:
  - de 2^{-126} até aproximadamente 2^{128} em ponto flutuante de representação simples
  - de 2^{-31} até $2^{+31} - 1$ na representação inteira, portanto, uma faixa numérica bem menor.
- Qual o preço a pagar?
- A precisão dos números em ponto flutuante pode ser menor.

Precisão – Exemplo prático

- A representação de 5.023.421.348,75 em IEEE 754 é:
  - Equivalente binário: 100101011011010110101001110100100,11
  - Notação científica normalizada: 1,00101011011010110101001110100100 11 $\times 2^{32}$
  - $\therefore$ S=0, e=32 ==> E=32+127 ==> E=10011111 e F=00101011011010110101001
  - IEEE 754: $\boxed{01001111100101011011010110101001}$
- Revertendo: 01001111100101011011010110101001
  - S=0, E=10011111=159 e F=$00101011011010110101001_2 = 0,16960632801055908 2_{10}$
  - $\therefore y = (-1)^0 \times (1 + 0,16960632801055908 2) \times 2^{159-127}$
  - Decimal: $\boxed{5.023.420.928}$
- Conclusão: $5.023.421.348,75 \neq 5.023.420.928$