TPC1

Resultados dos exercícios propostos

1. (A) Converta cada um dos valores para os seguintes sistemas:

	Valor a converter	Resultado	Valor a converter	Resultado
a) decimal	1101.01_2	13.25	10.11_2	2.75
b) octal	$111\ 110\ 011\ 101_2$	7635_8	$11\ 011.11_2$	33.6_8
c) hexadecimal	$10\ 1010\ 1011.011_2$	0x2ab.6	72.25	0x48.4
d) binário	0xfc2f	$1111\ 1100\ 0010\ 1111_2$	36.0625	$1\ 0100.0001_2$
e) ternário	24	220_3	174	20110_3

3. (A) Preencha a tabela abaixo com a gama de valores representáveis usando 5 bits em um dos sistemas de representação propostos.

Representação	Intervalo
Binário sem sinal, inteiros	[0 , 2^5−1] -> [0 , 31]
Binário sem sinal, 1 *bit* fracionário	[0 , 15.5]
Binário sem sinal, 3 *bits* fracionários	[0 , 3.875]
Sinal + Amplitude, inteiros	[−15 , 15]
Sinal + Amplitude, 1 *bit* fracionário	[−7.5 , 7.5]
Sinal + Amplitude, 3 *bits* fracionários	[−1.875 , 1.875]

4. (A) Efetue as seguintes operações aritméticas em binário usando apenas 8 bits:

00110011_2 + 01110101_2	10101000_2
011100.11_2 + 000011.01_2	100000.00_2
01001001_2 + 11010001_2	*Overflow* no resultado
0x4c + 0x2b	01001100_2 + 00101011_2 = 01110111_2
672_8 + 703_8	*Overflow* na codificação de cada operando

5. (A) Codificação binária para as divisões de um prédio de 15 andares, com 6 apartamentos por andar:

Para representar o andar usamos `sinal+amplitude` com 4 bits.
Para representar o apartamento usamos inteiros positivos com 3 bits.
Temos um máximo de 8 divisões por apartamento, logo usamos 3 bits, com a seguinte codificação:
`000` – sala; `001` a `011` – quarto; `100` – cozinha; `101` a `111` - casa de banho.
Total: 10 bits.
O piso -5, apartamento 3, quarto 2, codifica-se como: `1101 011 010`

6. **(A)** Converta o número **–233** para uma representação binária com 10-bits, com as seguintes representações:

Bit#	9	8	7	6	5	4	3	2	1	0
Valor	*512*	*256*	*128*	*64*	*32*	*16*	*8*	*4*	*2*	*1*
Sinal & Ampl	1	0	1	1	1	0	1	0	0	1
Compl p/ 1	1	1	0	0	0	1	0	1	1	0
Compl p/ 2	1	1	0	0	0	1	0	1	1	1
Excesso 2^{n-1}	0	1	0	0	0	1	0	1	1	1

7. **(A)** Converta para decimal o valor em binário (usando apenas 10-bits) **10 0111 0101$_2$**; pode-se apresentar o resultado de uma de 2 maneiras: **(i)** escreve-se em cada célula o valor que cada bit (na codificação especificada) tem em decimal, sabendo-se que o resultado na coluna da direita toma em conta o bit do sinal (quando exista) e o seu valor é a soma desses valores, ou **(ii)** escreve-se em cada célula o valor que cada bit (na codificação especificada) tem no sistema de numeração binário, sabendo-se que o resultado na coluna da direita toma em conta o bit do sinal (quando exista) e o seu valor é a soma do produto dos bits indicados, pelo seu valor.

Bit#	9	8	7	6	5	4	3	2	1	0	Resultado
Valor	*512*	*256*	*128*	*64*	*32*	*16*	*8*	*4*	*2*	*1*	
Codif em bin	***1***	***0***	***0***	***1***	***1***	***1***	***0***	***1***	***0***	***1***	
Int s/ sinal	512+	0+	0+	64+	32+	16+	0+	4+	0+	1=	629
Sinal & Ampl	–	(0+	0+	64+	32+	16+	0+	4+	0+	1)=	–117
Compl p/ 1	–	(256+	128+	0+	0+	0+	8+	0+	2+	0)=	–394
Compl p/ 2	–	(256+	128+	0+	0+	0+	8+	0+	2+	1)=	–395
Excesso 2^{n-1}	**512+**	0+	0+	64+	32+	16+	0+	4+	0+	**1–512=**	117

Bit#	9	8	7	6	5	4	3	2	1	0	Resultado
Valor	*512*	*256*	*128*	*64*	*32*	*16*	*8*	*4*	*2*	*1*	
Codif em bin	***1***	***0***	***0***	***1***	***1***	***1***	***0***	***1***	***0***	***1***	
Int s/ sinal	1	0	0	1	1	1	0	1	0	1	629
Sinal & Ampl	–	0	0	1	1	1	0	1	0	1	–117
Compl p/ 1	–	1	1	0	0	0	1	0	1	0	–394
Compl p/ 2	–	1	1	0	0	0	1	0	1	1	–395
Excesso 2^{n-1}	**0**	0	0	1	1	1	0	1	0	1	117

8. **(R)** Executar código num computador de **6-bits**; um inteiro "*short*" é codificado com 3-bits.

```
short sy = -3;
int y = sy;
int x = -17;
unsigned ux = x;
```

Expressão	**Decimal**	**Binário**
Zero	0	00 0000
--	-6	11 1010
--	18	01 0010
ux	47	10 1111
y	-3	11 1101
x>>1 *	-9	11 0111
TMax	31	01 1111
-Tmin	-(-32)	*overflow*
Tmin+Tmin	-64	*overflow*

* Ver-se-á mais tarde porque razão este resultado é assim.
Sugestão para estudantes **B**: analisar (e tentar compreender) como é que as operações de deslocamento de bits em C se comportam, e quais as diferenças entre deslocamento para a esquerda e deslocamento para a direita (para além da direção, como é óbvio).

9. **(R)** Qual a gama de valores inteiros nas representações binárias de **(i)** sinal e amplitude, **(ii)** complemento para 2, e **(iii)** excesso 2^{n-1}, para o seguinte número de bits:

	(i)	(ii)	(iii)
a) 6 bits	$]-2^5,\ 2^5[$	$[-2^5,\ 2^5[$	$[-2^5,\ 2^5[$
b) 12 bits	$]-2^{11},\ 2^{11}[$	$[-2^{11},\ 2^{11}[$	$[-2^{11},\ 2^{11}[$

10. **(A)** Efetue os seguintes cálculos usando aritmética binária de 8-bits em complemento para 2:

a) 16 + 110 Res.: $0001\ 0000_2 + 0110\ 1110_2 = 0111\ 1110_2$
b) 70 + 80 Res.: $0100\ 0110_2 + 0101\ 0000_2 = 1001\ 0110_2$ *overflow* (devia ser >0)
c) 80 + (–60) Res.: $0101\ 0000_2 + 1100\ 0100_2 = 0001\ 0100_2$
d) –98 – 29 Res.: $1001\ 1110_2 - 0001\ 1101_2 = 1000\ 0001_2$