

TPC1

Resultados dos exercícios propostos:

1. ^(A)Efectue as seguintes conversões:

| | Valor a converter | Resultado | | Valor a converter | Resultado |
|-------------|-------------------------------|----------------------------------|--|------------------------|-------------------|
| a) decimal | 1101.01 ₂ | 13.25 | | 10.01 ₂ | 2.25 |
| b) octal | 110 111 011 101 ₂ | 6735 ₈ | | 11 111.11 ₂ | 37.6 ₈ |
| c) hexadec. | 10 1100 1011.001 ₂ | 2CB.2 ₁₆ | | | |
| d) binário | 0xFF1F | 1111 1111 0001 1111 ₂ | | | |
| e) ternário | 174 | 20110 ₃ | | | |

2. ^(A)Converta o número **-233** para uma representação binária usando 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 |

3. ^(A)Converta para decimal o valor em binário (usando apenas 10-bits) **10 0111 0101₂**; 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 | 0+ | 0+ | 0+ | 64+ | 32+ | 16+ | 0+ | 4+ | 0+ | 1 | 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 |

4. ^(A)A maioria das pessoas apenas consegue contar até 10 com os seus dedos...

- Com este método, até quanto é possível contar usando ambas as mãos? → 1023
- Um dos dedos na extremidade da mão é o bit do sinal (em compl p/ 2). Qual a gama de valores que é possível representar com ambas as mãos? → [-2⁹, 2⁹]

5. ^(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 direcção, como é óbvio).

6. ^(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 | $] -2^{11}, 2^{11}[$ | $[-2^{11}, 2^{11}[$ | $[-2^{11}, 2^{11}[$ |

7. ^(A) Efectue os seguintes cálculos usando aritmética binária de 8-bits em complemento para 2:

- a) $4 + 120$ Res.: $0000\ 0100_2 + 0111\ 1000_2 = 0111\ 1100_2$
b) $70 + 80$ Res.: $0100\ 0110_2 + 0101\ 0000_2 = 1001\ 0110_2$ *overflow (devia ser >0)*
c) $100 + (-60)$ Res.: $0110\ 0100_2 + 1100\ 0100_2 = 0010\ 1000_2$
d) $-100 - 27$ Res.: $1001\ 1100_2 - 0001\ 1011_2 = 1000\ 0001_2$