

Nome:

Nº:

Notas:

1. Apresente sempre a **justificação da solução**, incluindo os cálculos que efetuar (se precisar de mais espaço utilize o verso da folha).
2. **Não são permitidas** máquinas de calcular.

1. Considere um sistema de computação operando sobre inteiros, representados com 8 bits, em complemento para 2:
 - a) **Calcule** o resultado da soma de 0xFC com 0x16. Apresente os resultados em decimal e hexadecimal.
 - b) **Calcule** o resultado, numa representação sem sinal, da divisão do valor decimal 35 por 4. Apresente o resultado em binário (com 8 bits) e em decimal. Apresente os cálculos que efetuar.
2. Considere a representação de números reais usando uma versão adaptada para 8 bits da norma IEEE 754, com 4 bits para o expoente, 3 para a mantissa e 1 para o sinal.
 - a) **Represente** nesse formato os valores 4.5 e 0.375. Apresente os resultados em binário.
 - b) **Verifique** se é possível representar exatamente o resultado da soma dos dois valores anteriores. Justifique a resposta comentando o resultado obtido.
3. Considere um sistema de computação com uma unidade de processamento IA-16 (*little endian*), que se encontra prestes a iniciar a execução da instrução `addw (%ax), %bx` de `prog`, cujo estado (parcial) é dado por:

Registos: %ax=0x8018 %ip=0x4014 %bx=0x0020 Memória (dados): 0x8014: 0x10 0x15 0xab 0x00 0x8018: 0x01 0x02 0x00 0x04	Memória (código): // objdump -d prog ... 0x4014: 83 10 addw (%ax), %bx ...
-----------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------

- a) **Indique**, cronologicamente, toda a informação que passa pelos barramentos de endereços, dados e de controlo durante a execução da instrução, incluindo as fases de busca e decodificação da instrução.

*Address Bus**Data Bus**Control Bus* *Time*

- b) **Indique**, justificando, quais os registos modificados e apresente o seu conteúdo após a execução da instrução.

Nome:

Nº:

4. Considere o seguinte excerto de código originado pela compilação de um programa escrito na linguagem C.

```

400000 <func>:
...
400009: 8d 04 12      lea    (%edx,%edx,1),%eax
40000c: 39 ca         cmp    %ecx,%edx
40000e: 7d 03         jge    400013 <func+0x13>
400010: 8d 04 4a      lea    (%edx,%ecx,2),%eax
400013: ...

```

- a) **Apresente**, as linhas de código C que possam ter originado, respetivamente, cada uma das duas instruções `lea` presentes neste programa.
- b) **Indique**, justificando, uma estrutura de controlo em C que possa ter gerado este código.
- c) **Explique** como é possível codificar a instrução `jge` com apenas 2 bytes, uma vez que o destino do salto é um endereço de 32 bits.

Notas de apoio (recomendação IEEE 754)

Normalized	±	0 < Exp < Max	Any bit pattern
Denormalized	±	0	Any nonzero bit pattern
Zero	±	0	0
Infinity	±	1 1 1 ... 1	0
Not a number	±	1 1 1 ... 1	Any nonzero bit pattern

Sign bit

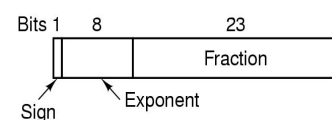
Valor decimal de um fp em binário:

precisão simples, normalizado:

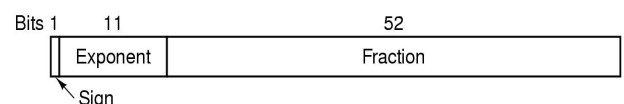
$$V_n = (-1)^S * (1.F)_2 * 2^{E-127}$$

precisão simples, subnormal:

$$V_{sub} = (-1)^S * (0.F)_2 * 2^{-126}$$



(a)



Valor do excesso (expoente com e bits):

$$\text{excesso} = 2^{(e-1)} - 1$$