

Correcção do Teste-Ensaio
29/04/2025

1)

a) $65536 = 2^{16}$ lugares, logo são necessários 16 bits no total.

b b b b b b b b b b b b b b b b

Linhas (7 bits)

Colunas (9 bits)

$$\downarrow \\ 2^7 = 128$$

$$\downarrow \\ 2^9 = 512$$

Como existem 4 vezes mais colunas que linhas, são precisos 7 bits para representar as linhas e 9 bits para representar as colunas ($128 \times 512 = 65536$).

Para codificar a linha 48, coluna 16, tem-se:

0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0

b) Um registo de um processador IA-32 ocupa 4 bytes e cada carácter ASCII ocupa 1 byte. Logo, é possível representar desde '0' '0' '0' '0' a '9' '9' '9' '9'.

Logo a afirmação é verdadeira.

2)

a) O código assembly descreve uma estrutura de controlo de fluxo do C do tipo while (cond) {<código>}, pois a condição que faz decidir se o salto é tomado ou não é a comparação `cmpl $0, %eax`, que vem antes do código a ser executado (caso o salto não seja tomado, isto é, caso o resultado da comparação não seja maior ou igual).

b)

```
movl $30, %eax    // i = 30
L1:
cmpl $100, %eax    // i <= 100?
jg L2              // se i > 100, salte para L2
addl %eax, %ebx     // b += i
addl $2, %eax       // i += 2
jmp L1
L2:
```


3) $0x8040: 03$ $addw \%ax, -8(\%bp)$
 $0x8041: 45$
 $0x8042: f8$ $\%ax = 0xffff; \%bp = 0x4008$

$0x4000: 0x30$
 $0x4001: 0x00$
 $0x4002: 0x20$
 $0x4003: 0xff$
 $0x4004: 0x30$
 $0x4005: 0x00$
 $0x4006: 0x40$
 $0x400f: 0xff$

a) Data bus: $03 \ 45 \ f8 \ ?? \ 0x30 \ 0x00 \ 0x0f \ 0x10$

- 1º - Ler IP ($0x8040$ e $0x8041$, pois nesta arquitectura lê-se sempre 2 bytes).
 - 2º - Colocar no barramento de dados os valores lidos (03 e 45).
 - 3º - Ler próximo endereço (pois é preciso saber o offset, neste caso é 8 e está no endereço $0x4042$, mas como se lê dois bytes, também se lê o endereço $0x4043$).
 - 4º - Colocar no barramento de dados os valores lidos ($f8$ e $??$).
 - 5º - Depois de fazer as contas ($0x4008 + 8 = 0x4000$), lê-se o endereço $0x4000$ e o endereço $0x4001$.
 - 6º - Colocar no barramento de dados os valores lidos ($0x30$ e $0x00$).
 - 7º - Depois de fazer as contas ($ax + 0x30 = 0xffff + 0x0030 = 0x100f$), coloca-se o resultado no barramento de dados, em formato little-endian, que neste caso corresponde a $0x0f$ e $0x10$.
 - 8º Depois incrementa-se o IP + tamanho da instrução = $0x8043$, para apontar para a próxima instrução a ser executada.
- (Além destes passos, também passa informação no control bus e no address bus).

b) O registo IP foi alterado (de $0x8040$ para $0x8043$).
 Na memória, foram alterados as células $0x4000$ (de $0x30$ para $0x0f$) e $0x4001$ (de $0x00$ para $0x10$).

4)

a) A afirmação é falsa. Com estas condições, não é possível representar o valor 360 nem o valor 320 .
 O maior número positivo possível a ser representado é $2^{n-1} - 1 = 2^{8-1} - 1 = 2^7 - 1 = 128 - 1 = 127$.

b) $addb \%al, \%ah$ $\%al = 0xd3; \%ah = 0x61$

$$\begin{array}{r} \%al = 0xd3 = 1101 \ 0011 \\ + \%ah = 0x61 = 0110 \ 0001 \\ \hline 1 \ 0011 \ 0100 \end{array}$$

↓
 Como os números
 a somar são de sinais
 opostos, não se conta este
 último transporte.

$$\text{Resultado} = 0011 \ 0100_2 = 52_{10}$$

5)

a)

1	0	0	0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---

 $\begin{matrix} S & E+7 & F \end{matrix}$

$$E = 2^{(4-1)} - 1 = 2^3 - 1 = 8 - 1 = 7$$

$$V = (-1)^S * (0.F) * 2^E = (-1)^1 * 0.0000001_2 * 2^{-6} = -0.0000001_2 * 2^{-6} = -1 * 2^{-7} * 2^{-6} = -1 * 2^{-13} = -2^{-13}$$

b) valor máximo:

0	1	1	1	0	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---

 $\begin{matrix} S & E & F \end{matrix}$

$$E = 1110_2 = 14$$

$$V = (-1)^S * (1.F) * 2^{E-7} = (-1)^0 * (1.1111111)_2 * 2^{14-7} = 1.1111111_2 * 2^7 = 11111111_2 * 2^0 = 11111111_2 = 255$$

Logo, neste formato não é possível representar a distância média da Terra à Lua.