



Prova de Digital – Teoria – Sexto Período – 2014/1
Professor: Adriano Martins Moutinho

Nota:

PERMITIDO USO DE CALCULADORA E
PROVA A LAPIS. SEM CONSULTA!

Nome: _____ Turma: _____

1) Usando assembly x86 de 16 bits, faça um programa que troque o conteúdo das posições de memória 2000:2000 e 2000:1000: (1.0 pontos)

2) Usando assembly x86 de 16 bits, faça um programa que escreva na pilha e gere os números na sequência decimal abaixo, desde 0 até FFFF: (1.0 pontos)

0,4,16,64,256,1024,4096,...

3) Usando assembly x86 de 16 bits, faça um programa que escreva na pilha números múltiplos de 21 (decimal) desde 0 até 256 (decimal): (1.0 pontos)

4) Usando assembly x86 de 16 bits, faça:

- a) Um programa capaz de somar dois números (A e B) de **16 bits**. (0.5 ponto)
- b) Um programa capaz de somar dois números (A e B) de **64 bits**. (1.5 ponto)

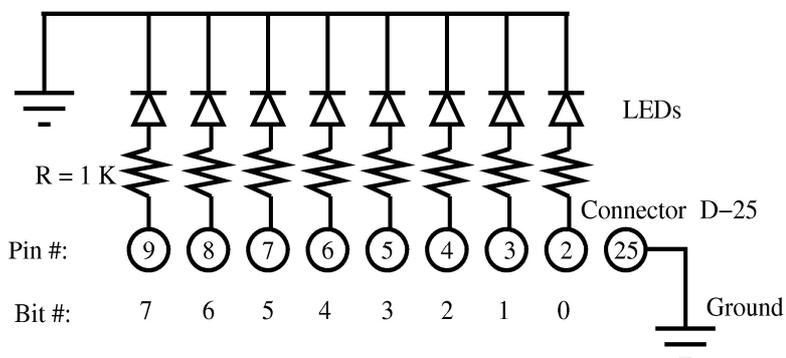
Em ambos os casos, o primeiro número (A) inicia na posição 2000:2000, o segundo (B) inicia no primeiro endereço após o primeiro e o resultado (A+B) deve ser armazenado no local do primeiro (A).

5) Traduza os programas abaixo (com valores em decimal) para o assembly de x86 (16 bits), mantendo o mesmo funcionamento. Considere que a variável NUM tem seu valor inicial na posição de memória 1000:1000 e seu valor final deve também ser gravado nesta posição (2 pontos).

a) Se NUM diferente de 21 então
NUM recebe o valor 13
Senão NUM recebe o NUM + 4

b) Se NUM igual a 13 então
NUM recebe NUM - 1
Senão (Se NUM = 9 então
NUM recebe o valor 2
Senão NUM = NUM + 1)

6) Oito LEDs estão conectados à porta paralela como indicado na figura ao lado. Usando assembly x86 de 16 bits, faça um programa acenda os leds um a um, desde o mais significativo até o menos (repetindo o processo em loop infinito). Escreva call 2000 onde houver necessidade de uma rotina de delay para observar-se o resultado. (2 pontos)



DICA: Instrução OUT DX,AL (onde AL é o dado para acender os leds e DX tem que ser 378)

7) A questão 5 necessita de um delay com, aproximadamente, 1500 M laços. Escreva-o em assembly x86 de 16 bits, terminado a rotina com **RETURN**: (1 pontos)