



----- I^a- PARTE -----

1- (a) Um micro com **20 linhas de endereços**, quantos endereços é capaz de endereçar (o valor deve vir especificado em K ou múltiplos)?

(b) Uma RAM de **2 Kbytes** selecciona uma gama de endereços desde **2000H** até **H** ?

2- Porque é que nalguns circuitos com o microcontrolador da família 8051 se usam osciladores de frequência “esquisitas” tais como OSC=**11.059000** MHz. Explique porquê e em que situações específicas é que se justifica a sua aplicação.

3- As seguintes instruções em Assembler 8051 são válidas ou não? Se não explique porquê.

a) **ANL A, @R3**

b) **MOV C, SP.3**

c) Qual o conteúdo do acumulador depois da seguinte instrução: **ORL A, #0FFH**.

4- Escreva o código necessário que transmite o conteúdo do acumulador via **P1** se **R7** é positivo e transmite o registo **B** se **R7** é negativo. Comente as opções tomadas.

5- Desenhe um diagrama das linhas de endereço, controlo e linhas de dados para que o 8051 possa comunicar com um modulo de **ROM** externa. A memória **ROM0** deve responder ao espaço de endereçamento de **0x2000-0x8FFF**. Por favor, identifique cada linha apropriadamente.



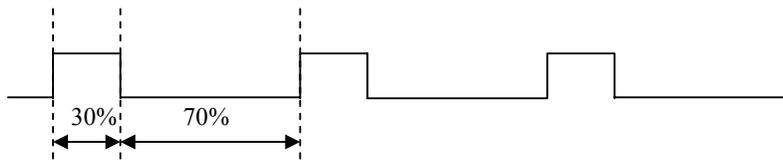
----- 2^a- PARTE -----

II-1- Implemente em Assembler MCS8051 um programa que transmita todos os números entre **0** e **255** para o porto **P0**, um a um. O próximo numero deve ser transmitido quando for detectado uma nova **interrupção externa por nível**. Depois de 255 o programa deve voltar a zero. Use a interrupção **externa 1** no modo “*edge triggered*”.

II-2- Foi delineado o seguinte ciclo para gerar a largura de pulsos dum sinal de frequência fixa.

```
main()
{
  int I, t_period, T_on;
  for (;;) // ciclo infinito
  {
    for( I =0; I < T_period; I++) //loop for one cycle,
    {
      if ( I < Ton_time)
        output = 1; // on
      else
        output = 0 ; //off
    }
  }
}
```

- (a) Traduza para linguagem **ASM8051** o código representado acima.
- (b) Comente a eficiência deste método para gerar sinais de **PWM** num contexto dum programa mais extenso e complexo.
- (c) Implemente uma nova rotina de **PWM** e um “**duty cycle**” de 50% usando agora interrupções e fazendo **T_period=128 us**.
- (d) Implemente a rotina “**DIMINUI**” que produz um pulso PWM com um “**duty cycle**” de **30%** (i.é, 30% do tempo ON e 70% OFF). Deve apresentar todas as inicializações dos registos.



Boa Sorte !!!