



----- I<sup>a</sup>-PARTE -----

1- (a) Um micro com **32 linhas de endereços**, quantos endereços é capaz de endereçar (o valor deve vir especificado em K ou múltiplos)?

(b) Uma RAM de **32 Kbytes** selecciona uma gama de endereços desde **4000H** até .... **H** ?

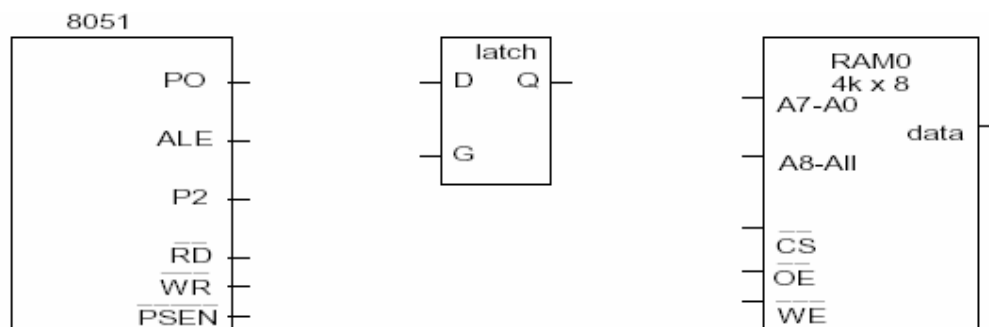
2- Para escrever as rotinas de controlo duma aplicação pode-se escrever o código em linguagem **ASM** ou em **C** por exemplo. Identifique um conjunto de situações onde é mais aconselhado usar uma e outra situação.

3- Escreva um pequeno segmento de código **ASM** para alterar o banco de registos corrente para o **banco 2**:

(a) Usando instruções de bytes (operando bytes em vez de bits)

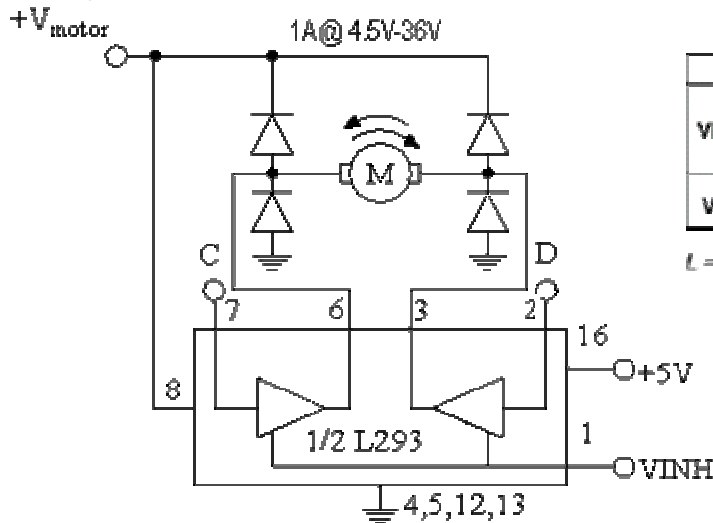
(b) Usando instruções de bits.

4- Complete o seguinte diagrama desenhando as linhas de endereço, controlo e linhas de dados para que o 8051 possa comunicar com um modulo de RAM externa. A memória **RAM0** deve responder ao espaço de endereçamento de **0x8000-0xBFFF**. Por favor, identifique cada linha apropriadamente.





1- Implemente em Assembler MCS8051 um controlador de um motor DC ligado às rodas dum robot. No interface entre o micro da família 'C52 e os motores DC é usado o driver **L293** do tipo "H-Bridge". O circuito básico do driver **L293** é mostrado na figura seguinte. Da tabela de verdade, verifica-se que a direcção do motor pode ser controlado pelo pino **C** e **D**. **VINH** inibe/desinibe a alimentação do motor para regular a velocidade, por isso usamos este pino para a sinalização **PWM**.



|          | INPUTS       | FUNCTION                |
|----------|--------------|-------------------------|
| VINH = H | C = H; D = L | Turn Right              |
|          | C = L; D = H | Turn Left               |
|          | C = D        | Fast Motor Stop         |
| VINH = L | C = X; D = X | Free Running Motor Stop |

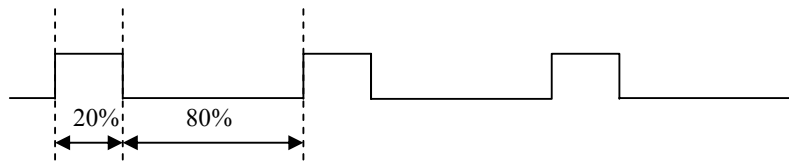
L = Low H = High X = Don't Care

**Use of L293 forms an H-Bridge for bidirectional DC motor driver**

O interface entre um micro da família '89C52 e o motor DC é efectuado pelo porto **PI** que por sua vez está ligado ao driver **L293** que alimenta o motor. O sinal **C** e **D** são usados no controlo de direcção do motor. Existe também um botão de emergência "EXIT" que faz parar bruscamente os motores.

Para implementar o controlador é preciso programar em Assembler MCS8051 um conjunto de rotinas.

- Complete o esquema das ligações de hardware que vai utilizar entre o micro, o **L293** e os botão .
- Implementar a rotina "PWM" que produz um sinal de onda quadrada ("duty cycle" de 50%) de período igual a 256 us.
- Implemente a rotina "DIREITA" que faz o motor rodar para a direita, para isso cria um pulso PWM com um "duty cycle" de 20% (i.é, 20% do tempo ON e 80% OFF).



- Implemente a rotina "ESQUERDA" que faz o motor rodar para a esquerda, para isso cria um pulso PWM com um "duty cycle" de 80%.
- Implementar a rotina "STOP" que faz o motor parar rapidamente.
- Desenvolva o programa principal que irá fazer as chamadas da rotinas criadas anteriormente. Deve incluir as inicializações das interrupções, a rotina DIREITA e ESQUERDA são activadas por interrupções externas activadas por nível e por último deve ter em atenção também o botão EXIT.

**Boa Sorte !!!**