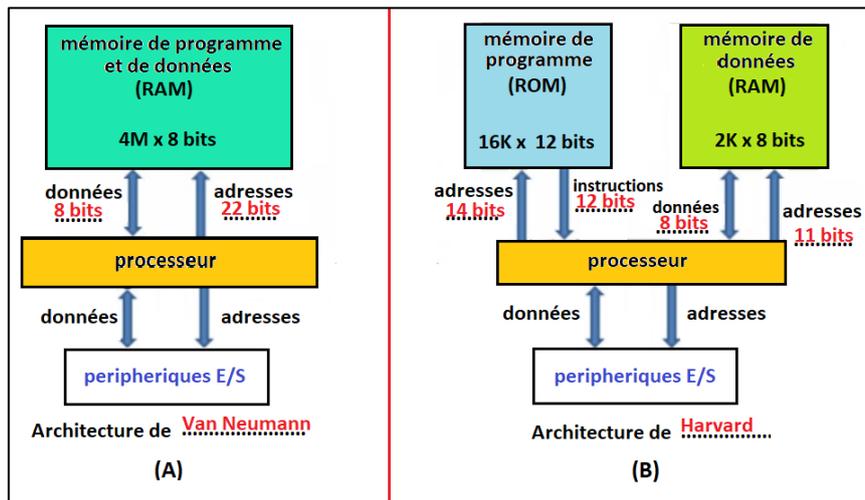


## Corrigé de la série de TD N° 7-8

### Corrigé de l'exercice 01 :

- Le schéma bloc (A) présente une architecture de Van Neumann, puisque, il utilise une seule mémoire pour les données et le programme.  
Le schéma bloc (B) présente une architecture de Harvard, puisque, il utilise deux mémoires séparées : une mémoire pour les données et une autre pour le programme.
- Déterminer la taille de chaque bus (adresses, données, instructions).



- On peut trouver le microcontrôleur dans : les véhicules, les robots, les appareils médicaux, les émetteurs-récepteurs radio mobiles, les distributeurs automatiques et les appareils électroménagers.
- Les microcontrôleurs peuvent avoir une architecture Harvard ou Van Neuman.
- Les composants qu'on peut trouvés dans un microcontrôleur sont :
  - Les Timers (temporisateurs)
  - Les convertisseurs analogique/numérique et les convertisseurs numérique/analogique.
  - Les ports d'entrée/sortie.
  - Module de communication série.
  - Watch dog
  - EEprom

### Corrigé de l'exercice 02 :

#### Les réponses aux questions de cours

- Le PIC16F84 est conçu selon une architecture RISC, puisqu'il a un jeu instructions constitué seulement de 35 instructions.
- La taille des instructions du PIC16F84 est 14 bits, et la taille ses données mémoire est de 8 bits.
- Les circuits qui sont indispensables au fonctionnement du microcontrôleur sont :  
Le circuit de reset : il faut que l'entrée  $\overline{MCLR}$  soit à un niveau de tension haut.  
L'horloge externe, pour cadencer le fonctionnement du processeur.
- Si l'horloge externe du microcontrôleur PIC16F84 est dotée d'un quartz de 8Mhz.

$$F_{cyc} = \frac{F_{osc}}{4}, \text{ le temps de cycle est défini par : } T_{cyc} = \frac{1}{F_{cyc}} = \frac{1}{2 \cdot 10^6} = 0,5 \cdot 10^{-6} = 0,5 \mu s$$

- La pile du PIC16F84 est de structure LIFO (Last In First Out), elle contient 8 niveaux.
- La taille de la mémoire programme du PIC16F84 est  $1K \cdot 14$  bits.
- Le PIC16F84 peut gérer 4 interruptions périphériques qui sont :  
RB0/INT, Timer0, RB4..7 et fin d'écriture EEPROM.

8. Le bit GIE (Global Interrupt Enable du registre INTCON<7>) permet d'autoriser toutes les interruptions du PIC16F84 lorsqu'il est positionné à '1'.
9. Si l'interruption TIMER0 est autorisée (T0IE =1), l'interruption se produit, lors du débordement du registre TMR0 (lorsqu'il passe de FFH à 00H).
10. L'instruction **RETFIE** (RETurn From Interrupt) annonce la fin de sous-programme d'interruption et oblige le processeur à retourner au programme principal.

### Corrigé de l'exercice 03 :

Répondez par (vrai) ou (faux) aux affirmations suivantes, et corrigez les affirmations fausses

1. Dans une architecture Harvard, le processeur peut facilement lire/écrire des données et accéder aux instructions à tout moment. **(vrai)**
2. Lors de la mise sous tension du PIC16F84, le processeur pointe vers l'adresse 0000H de la mémoire du programme. **(vrai)**
3. Un niveau de tension haut sur la broche  $\overline{\text{MCLR}}$  provoque un reset du PIC16F84. **(faux)**  
Un niveau de tension bas sur la broche  $\overline{\text{MCLR}}$  provoque un reset du PIC16F84.
4. Si un périphérique provoque une interruption, le processeur se positionne à l'adresse 0004H. **(vrai)**
5. Les registres TRISA et TRISB déterminent respectivement le sens du PORTA et du PORTB. **(vrai)**
6. Le mode SLEEP est utilisé pour minimiser la consommation d'énergie dans les applications alimentées par batterie. **(vrai)**
7. Une interruption sur la broche RB0/INT peut faire sortir le PIC du mode SLEEP (réveiller le PIC). **(vrai)**
8. Dès la mise sous tension du PIC16F84 le registre TMR0 s'incrémente automatiquement. **(vrai)**
9. La pile du PIC16F84 permet la sauvegarde de l'adresse du PC de retour lors des appels des sous-programmes (CALL) ou lorsqu'une interruption survient. **(vrai)**
10. Le programmeur doit effacer le drapeau qui a signalé l'interruption après l'avoir traitée. **(vrai)**
11. L'instruction RETFIE permet de restaurer le contenu du registre PC à partir de la pile pour permettre au processeur de reprendre l'exécution du programme qu'il avait interrompu. **(vrai)**
12. Le PIC16F84 peut sauvegarder des données mémoire en pile. **(faux)**  
La pile du PIC16F84 est utilisé pour sauvegarder les adresses de retour vers le programme, lorsqu'une interruption se produit ou lors d'un appel de sous-programme par l'instruction 'CALL'.
13. Dans une architecture de Van Neumann, la vitesse d'exécution est plus rapide car le processeur récupère simultanément les données et les instructions. **(faux)**  
Dans une architecture de Van Neumann, La vitesse d'exécution est plus lente car il ne peut pas récupérer les données et les instructions en même temps.

### Corrigé de l'exercice 04 :

Nous constatons que les adresses des boutons poussoirs sur le PORTB (RB0, RB2 et RB4) ont les mêmes adresses que les LEDs sur le PORTA (RA0, RA2 et RA4). Pour afficher l'état des boutons, on lit le PORTB et on transfère son contenu directement sur le PORTA.

```

org    0x000          ; Adresse de départ après reset
goto  init          ; Adresse 0: initialiser

init
; INITIALISATION DES PORTS
Clrf   PORTA        ; initialisation des PORTS A et B
Clrf   PORTB
BSF    STATUS, RP0  ; Sélection de la bank1 pour manipuler les registre TRISA et TRISB
Clrf   TRISA        ; PORTA programmé en sortie
Movlw  0x0FF
Movwf  TRISB        ; PORTB programmé en entrée

```

```
        BCF    STATUS, RP0    ; sélection de la bank0
        goto   start
; PROGRAMME PRINCIPAL
Start
        Movf   PORTB, 0       ; W ← PORTB
        Movwf  PORTA         ; PORTA ← W
        goto   start         ; boucler
END                          ; directive de fin de programme
```