

TD – CH1 : INTRODUCTION AUX MICROPROCESSEURS

Exercice 1 :

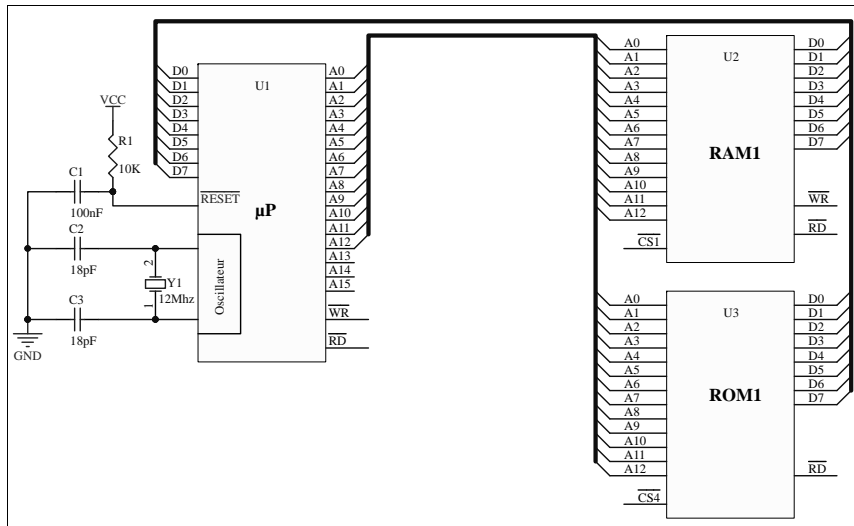
- Donner les avantages de l'architecture Harvard
- Qu'appelle-t-on l'emplacement, où le microprocesseur cherche à exécuter la première instruction après la mise sous tension (POR : Power On Reset) ?
- De quoi est composé un système à microprocesseur ?
- Le registre d'état du microcontrôleur PIC16F877 comporte les indicateurs suivants :
 - C (Carry) : inducteur de dépassement de capacité ; ce bit est inversé en cas d'un emprunt.
 - DC (Decimal Carry) : indicateur de dépassement sur 4 bits.
 - Z (Zero) : indique que le résultat est nul.

Déterminer l'état de ces indicateurs dans les cas suivants :

38H + 2FH ; 9FH + 61H ; 1BH – 20H

Exercice 2 :

On donne le schéma de connexion de deux mémoires avec un microprocesseur de 16 bits d'adresses et 8 bits de données.



- Quelle est l'architecture de ce microprocesseur ?
- Donner la capacité de chaque mémoire en kbits et en octets.
- On veut adresser la mémoire RAM1 à partir de l'adresse 0000H et la ROM1 à partir de l'adresse E000H.
 - Compléter le tableau suivant, en indiquant dans la troisième colonne l'adresse la plus basse et l'adresse la plus haute de chaque mémoire en hexadécimal.

		Adr. en Hexa	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
RAM1	Adr. basse																	
	Adr. haute																	
ROM1	Adr. basse																	
	Adr. haute																	

- Donner les équations logiques des lignes $\overline{CS1}$ et $\overline{CS4}$ en fonction des lignes d'adresses A15, A14 et A13.
 - Compéter alors le schéma de décodage d'adresse en utilisant des opérateurs logiques.
- On veut étendre la capacité mémoire de cette carte en ajoutant deux mémoires RAM2 et ROM2 identiques aux précédentes.
 - Donner l'adresse de base des nouvelles mémoires, si l'on veut que deux mémoires de même type soient adjacentes.
 - Donner les équations logiques des lignes de sélection $\overline{CS2}$ et $\overline{CS3}$ (correspondant respectivement aux mémoires RAM2 et ROM2) en fonction des lignes d'adresses A15, A14 et A13.
 - Donner le schéma de décodage d'adresses de toutes les mémoires en utilisant le circuit intégrées 74138.

A	B	C	G1	G2A	G2B	$\overline{Y0}$	$\overline{Y1}$	$\overline{Y2}$	$\overline{Y3}$	$\overline{Y4}$	$\overline{Y5}$	$\overline{Y6}$	$\overline{Y7}$
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	1	0	0	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

- On suppose que le programme utilisateur commence au début de la mémoire programme (ROM), et que le vecteur RESET est situé à l'adresse FFFEH. Donner le contenu des cases mémoires d'adresses FFFEH et FFFFH (ce microprocesseur a une architecture little-endian, c.-à-d. que l'octet le plus faible est placé à l'adresse basse et l'octet le plus fort est placé à l'adresse haute).

Exercice 3 :

On donne ci-après, le schéma électronique d'une carte à base d'un microprocesseur générique à architecture Harvard.

PARTIE A

- Quel est le rôle du circuit composé de R1 et C1 ?
- Donner la capacité de la mémoire de données. Déterminer l'adresse la plus basse et l'adresse la plus haute.
- Donner l'adresse de la mémoire de programme en kbits et kmots (1 mot = 16 bits).
- Donner les adresses de bases de deux circuits d'interface U4 et U5.

PARTIE B

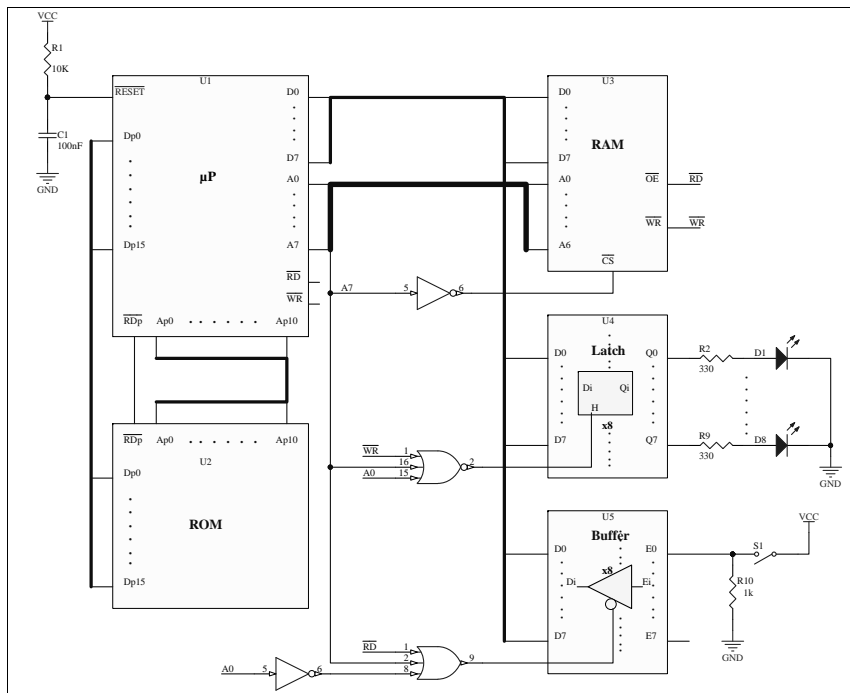
On donne dans le tableau suivant quelques instructions et le code machine correspondants:

Code mnémotique	Description	Code machine
movlw k	$W \leftarrow k$: charger la valeur k dans W	0000 1110 kkkk kkkk
movfw A	$W \leftarrow (A)$: le contenu de la case mémoire d'adresse A est chargé dans le registre W	0110 1111 AAAA AAAA

movwf A	(A) ← W : le contenu du registre W est chargé de la case mémoire d'adresse A	0101 0000 AAAA AAAA
andfw A	W ← W and (A), si W = 0 alors le flag Z ← 1	0001 0100 AAAA AAAA
goto A	Saut à l'adresse A (PC ← A)	1110 0AAA AAAA AAAA
JZ déplacement	si Z = 1, saut à l'adresse PC=PC+déplacement	1111 0100 AAAA AAAA
JnZ déplacement	si Z = 0, saut à l'adresse PC=PC+déplacement,	1111 1000 AAAA AAAA

On connecte à la broche 0 de circuit U5 un interrupteur S1 et huit diodes LEDs à la sorties du circuit U4.

- Traduire l'algorithme suivant en assembleur
Repete toujours
 si S1 est fermé alors
 Toutes les LEDs allumées
 sinon
 Toutes les LEDs éteintes
 Finsi
Fin Repete
- Donner la codification en binaire des instructions, si on suppose que ce code commence à l'adresse 0200H.



Corrigé

Exercice 1 :

- Les avantages de l'architecture Harvard :
 - Accès simultané aux mémoires programme et données : donc exécution plus rapide des instructions.
 - Limite le problème de modification accidentelle des programmes
 - Bien adaptée pour une mise en œuvre sur FPGA

De même cette architecture présente inconvénients tels que :

- Plus difficile à mettre en œuvre
 - Augmentation de nombre des broches du circuit
- Cet emplacement est appelé « le vecteur RESET ».
 - Un système à microprocesseur comprend : un microprocesseur, des mémoires RAM et ROM et des périphériques.
 -

38H : 00111000	9FH : 10011111	1BH : 00011011
+ 2FH : 00101111	+ 61H : 01100001	- 20H : 00100000
= 67H : 01100111	= 100H : 100000000	= FBH : 11111011
C = 0, DC = 1, Z = 0	C = 1, DC = 1, Z = 1	C = 0, DC = 1, Z = 0

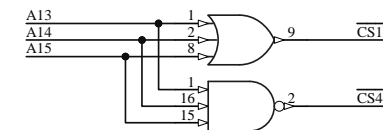
Exercice 2 :

- Architecture Van Neuman : puisque les mémoires ont le même bus d'adresses et ont le même bus de données.
- Les deux mémoires ont la même capacité, car elles ont le même nombre de lignes d'adresses. $C = 2^{13} = 8192 \text{ Octets} = 8 \text{ koctets} = 64 \text{ kbits}$
- RAM1 et ROM1
 - Détermination des adresses

	Adr. en Hexa	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
RAM1	Adr. basse	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Adr. haute	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ROM1	Adr. basse	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	Adr. haute	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

- Vous remarquez bien que la RAM1 est sélectionnée pour la combinaison $(A15, A14, A13) = (0, 0, 0)$. Donc $\overline{CS1} = \overline{A15} \cdot \overline{A14} \cdot \overline{A13}$ d'où $\overline{CS1} = \overline{A15} \cdot \overline{A14} \cdot \overline{A13} = \overline{CS1} = A15 + A14 + A13$
 De même la ROM1 est sélectionnée pour $(A15, A14, A13) = (1, 1, 1)$ donc $\overline{CS4} = A15 \cdot A14 \cdot A13$ d'où $\overline{CS4} = A15 \cdot A14 \cdot A13$

- Schéma de décodage



4. Ajout des mémoires

a. Adresses de base des nouvelles mémoires

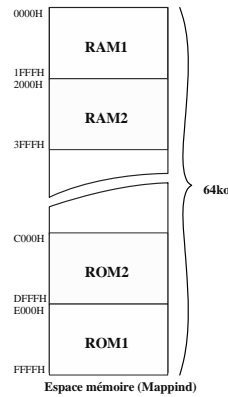
L'adresse de la dernière case de la RAM1 est 1FFFH, la RAM2 est placée juste après, donc son adresse de base est 1FFFH+1 = 2000H.

La capacité de la mémoire est égale à 8k = 8192 = 2000H.

L'adresse la plus haute de la RAM1 = adresse de base + la capacité de la mémoire -1 = 2000H + 2000H -1 = 3FFFH.

La mémoire ROM2 est placée avant la ROM1. Donc son adresse de base = adresse de base de la ROM1 - sa capacité = E000H - 2000H = C000H.

Pour l'adresse la plus haute = C000H+2000-1=DFFFH



b. Equations de sélection des nouvelles mémoires :

Pour la RAM2, si on compare les adresses basse et haute :

2000H = 0010000000000000 et 3FFFH = 0011111111111111. On remarque les bits qui ne changent pas sont (A15,A14,A13) = (0,0,1) on déduit facilement que

$$CS2 = \overline{A15} \cdot \overline{A14} \cdot A13 \text{ d'où } \overline{CS2} = A15 \cdot A14 \cdot \overline{A13}$$

De même pour la ROM2. En comparant les adresses de début et de fin, on remarque que les bits qui ne changent pas sont (A15,A14,A13) = (1,1,0), on déduit alors que

$$CS3 = A15 \cdot A14 \cdot \overline{A13} \text{ d'où } \overline{CS3} = \overline{A15} \cdot \overline{A14} \cdot A13$$

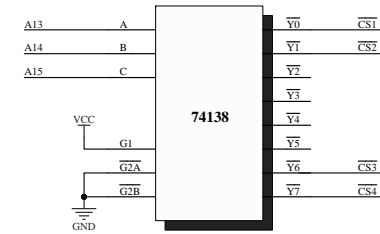
c. Schéma de décodage d'adresse : On donne dans ce tableau les combinaisons de sélection des mémoires qu'on a déterminé dans 3.b) et 4.b)

A15	A14	A13	CS1	CS2	CS3	CS4
0	0	0	0	1	1	1
0	0	1	1	0	1	1
1	1	0	1	1	0	1
1	1	1	1	1	1	0

Choisissons à partir du tableau de fonctionnement du circuit 74138, les quatre combinaisons du tableau précédent, c-à-d pour (C,B,A)=(0,0,0) et (0,0,1) et (1,1,0) et (1,1,1).

G2A	G2B	G1	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

En comparant les deux tableaux, il suffit de connecter A15, A14, et A13 aux entrées C, B et A, et les sorties Y0, Y1, Y6 et Y7 aux lignes de sélection CS1, CS2, CS3 et CS4. Il ne faut pas oublier de connecter les entrées de validation G1 à VCC et G2A, G2B à la masse.



5. Le programme commence au début de la mémoire de programme, c'est-à-dire à l'adresse C000H, il faut donc charger cette valeur dans le vecteur RESET.

On charge donc à l'adresse FFEH la valeur 00H et à l'adresse FFFFH la valeur 00H.

Exercice 3 :

PARTIE A

1. Ce circuit a pour rôle de générer une impulsion négative à la mise sous tension. Cette impulsion permet d'initialiser le microprocesseur.
2. La mémoire de données a 7 lignes d'adresse (A0 à A6) d'où C = 2⁷ = 128 octets. Cette mémoire est sélectionnée lorsque la ligne d'adresse A7 = 1. D'où l'adresse la plus basse = 80H et l'adresse la plus haute = FFH.
3. La mémoire programme comporte 11 lignes d'adresses (Ap0 à Ap10), et chaque case à une largeur de 16 bits (Dp0 à Dp15). D'où Cp = 2¹¹ = 2048 mots = 32kbits
4. Le circuit U4 est sélectionné quand A7 = 0 et A0 = 0 donc son adresse de base = 00H. De même le circuit U5 est sélectionné quand A7 = 0 et A0 = 1 donc son adresse de base = 01H.

PARTIE B

	Code assembleur	adresse	Code machine
Boucle :	movlw 01H ; Masque	0200H	0000 1110 0000 0001
	andfw 01H ; 01H : adresse de U5	0201H	0001 0100 0000 0001
	JZ Led_off ; sauter à Led_off	0202H	1111 0100 0000 0011
	movlw FFH ; LEDs allumées	0203H	0000 1110 1111 1111
	movwf 00H ; 00H : adresse de U4	0204H	0101 0000 0000 0000
Led_off :	goto Boucle ; revenir au début	0205H	1110 0010 0000 0000
	movlw 00H ; LEDs éteintes	0206H	0000 1110 0000 0000
	movwf 00H ; 00H : adresse de U4	0207H	0101 0000 0000 0000
	goto Boucle ; revenir au début	0208H	1110 0010 0000 0000

JZ Led_off : c-à-d. on doit sauter à l'adresse 206H = PC + déplacement. A ce moment PC contient la valeur 0203H (0202H + 1 : incrémentation de PC lors de décodage). D'où déplacement = 206H-203H=03H.