INTRODUCTION AUX MICROPROCESSEURS

STRUCTURE DE BASE D'UN CALCULATEUR

D'un point de vue matériel, un système informatique minimal est constitué d'un processeur, des mémoires et d'un ensemble des circuits d'interfaces.

Le microprocesseur (MPU : *Micro-Processing Unit*) élément de base, est un circuit intégré à très grande échelle d'intégration (VLSI : Very Large-Scale Integration) capable d'exécuter des instructions (opérations élémentaires) chargées dans la mémoire de programme. Ainsi, il remplit donc les fonctions d'une unité centrale de traitement (CPU : *Central Processing Unit)* en un seul boîtier.

Toutes les informations que le microprocesseur utilise, sont stockées dans des mémoires; le fonctionnement du microprocesseur est entièrement conditionné par le contenu de celles-ci. Ces mémoires contiennent deux types d'informations : le **programme** et les **données**, nécessaires pour la réalisation d'une tâche précise.

- Le programme (ensemble des ordres élémentaires à exécuter) est placé de façon définitive dans des mémoires qui ne pourront qu'être lues par le microprocesseur. Ce sont des mémoires non volatiles, de type **ROM** (Read Only Memory), **PROM, EPROM, Flash** …
- Les données ou OPERANDES proviennent le plus souvent, d'un calcul effectué par le microprocesseur ou d'un périphérique d'entrée, (clavier, disque, ...). Elles sont stockées dans des mémoires qui peuvent être lues et écrites, appelées **RAM** (Random Acces Memory).

Dans un système à microprocesseur (Figure 1), l'interface d'entrée-sortie (E/S) permet d'assurer la liaison entre l'unité centrale de traitement et l'environnement extérieur (périphériques). Le microprocesseur échange les informations avec les composants qui lui sont associés (mémoire et périphériques E/S) au moyen d'un ensemble des lignes de connexions appelé ***bus***.

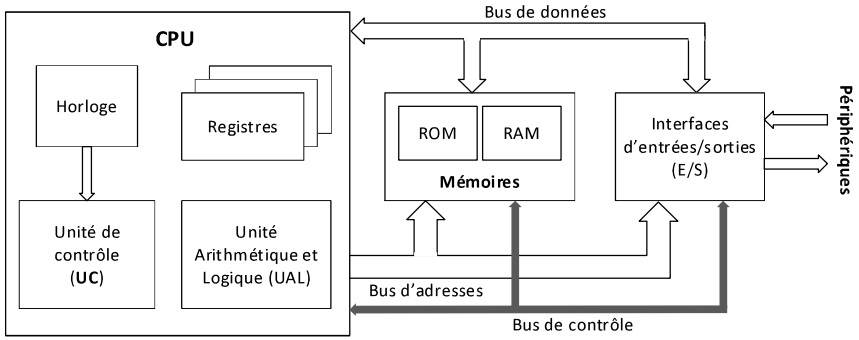


Figure 1 : Schéma bloc d'un système à microprocesseur

Un bus est un ensemble de fils qui assurent la transmission du même type d'information. On distingue trois types de bus :

- **Le bus de données** : bidirectionnel, assure le transfert des informations entre le microprocesseur et son environnement. Le nombre de lignes du bus de données définit la capacité de traitement du microprocesseur ; selon le microprocesseur la largeur du bus peut être de 8 bits, 16 bits, 32 bits, 64 bits …
- **Le bus d'adresses :** unidirectionnel, permet la sélection de la case contenant l'information à traiter dans un *espace* adressable. L'espace adressable peut avoir 2^n emplacements, avec *n* est le nombre de lignes du bus d'adresses.
- **Le bus de contrôle** : constitué de quelques lignes, assure la synchronisation du flux d'informations entre le microprocesseur et les circuits aux qu'il s'adresse.

ARCHITECTURE D'UN CPU

Le microprocesseur est constitué des unités fonctionnelles suivantes :

- Une unité Arithmétique et Logique (UAL ou ALU en anglais),
- Des registres,
- Une unité de contrôle (CU).

Unité arithmétique et logique

Cet organe, interne au microprocesseur, permet la réalisation des opérations arithmétiques (Addition, Soustraction…) et logiques (AND,

OR, XOR…). Outre les opérations arithmétiques et logiques, l'ALU réalise aussi les opérations de décalage et de rotation. Le registre d'état lié étroitement à l'ALU, met à la disposition du programmeur (à travers ces indicateurs « Flags ») des renseignements supplémentaires sur le résultat de quelques opérations (résultat nul, négatif, dépassement…). Citons quelques indicateurs que nous retrouvons dans la plupart des microprocesseurs :

- Le bit de retenue (**C** : carry), mis à 1 si le résultat d'une opération dépasse la taille du registre de destination.
- Le bit de retenue intermédiaire (**AC** : Auxiliary-Carry), retenue sur le quatrième bit (utilisé pour le calcul en BCD).
- Le bit de signe (**S** ou **N**), mis à 1 si le résultat d'une opération arithmétique est négatif.
- Le bit zéro (**Z**) **:** mis à 1 si le résultat d'une opération est nul.
- Le bit de débordement (**OV** : overflow), mis à 1 si une opération arithmétique a écrasé le bit de signe.

Pour monter l'utilité de ces indicateurs, prenons l'exemple d'une instruction conditionnelle :

Si (**A** = **B**) alors

Traitement 1

Sinon

Traitement 2

La comparaison entre **A** et **B** est traitée par le microprocesseur de la manière suivante :

- Soustraire **B** de **A**
- Si l'indicateur **Z = 1** (c-à-d **A – B = 0**) il exécute le **Traitement 1** et saute le *traitement 2*.
- Sinon (**Z = 0**) il saute le *traitement 1* et exécution du **Traitement 2.**

Toutes les instructions du saut conditionnel, se basent sur le test sur ces indicateurs.

Les registres

Il existe deux types de registres : les registres à usage général, et les registres d'adresses (ou pointeurs).

Les registres à usage général (Registres de travail)

Ce sont des mémoires rapides à l'intérieur du microprocesseur ; ils permettent à l'UAL de manipuler des données à vitesse élevée.

L'adresse d'un registre est associée à son nom (Exemple : **A**, **BX**, **WREG**…).

Les registres d'adresses (pointeurs)

Ce sont des registres sont utilisés pour l'adressage de la mémoire. Parmi ces registres, nous pouvons citer :

- Le compteur de programme **PC** : le microprocesseur utilise ce registre pour repérer l'exécution du programme. Celui-ci contient toujours l'adresse de la prochaine instruction à exécuter.
- Le pointeur de pile (Stack Pointer **SP**) : pointe toujours le sommet de la pile. La pile est une partie de la mémoire de données de type **LIFO** (Last In First Out) utilisée pour sauvegarder l'adresse de retour d'un sous-programme et/ou des variables locales.
- Les registres pointeurs de données ou d'index : utilisés pour l'adressage indirect de la mémoire de données.

L'unité de contrôle (UC)

Elle permet de séquencer le déroulement des instructions. Elle effectue la recherche en mémoire des instructions, le décodage et l'exécution de l'instruction recherchée. Elle est composée essentiellement d'un :

- *Registre d'instruction* (RI) : recoit le code de l'instruction à exécuter.
- *Décodeur d'instruction* : permet de déterminer le type de l'instruction à exécuter.
- *Bloc logique de contrôle (ou séquenceur)* : Il organise toutes les étapes d'exécution des instructions au rythme d'une horloge et élabore tous les signaux de synchronisation internes et externes du microprocesseur.

FONCTIONNEMENT DU MICROPROCESSEUR

Le programme est une suite d'instructions stockées dans la mémoire de programme. Pour exécuter ces instructions de manière séquentielle, le microprocesseur utilise un registre appelé compteur de programme. Ce registre contient l'adresse de l'instruction à exécuter.

Jeu d'instructions

Les microprocesseurs sont capables d'effectuer un certain nombre d'opérations élémentaires. Cet ensemble d'opérations élémentaires est appelé **jeu d'instructions**. Une instruction au niveau machine doit fournir à l'unité centrale toutes les informations nécessaires pour déclencher une telle opération élémentaire. Elle comporte en général plusieurs champs ; le premier champ contient le code de l'opération (***Code-Op*** ou ***Op-Code*** en anglais) ; les autres champs peuvent comporter des données ou l'identificateur des opérandes. Sur certaines machines toutes les instructions ont la même longueur, sur d'autres cette longueur est variable.

Format d'une instruction :

Code de l'opération	Opérande1 ,	Opérande2

Exemple :

Additionner le contenu du registre A avec le contenu de la case mémoire dont l'adresse est mentionnée dans le champ 3

ADDITIONNER	Registre A ,	@ Case mémoire

Cycle d'exécution d'une instruction

Lors de son exécution, une instruction est décomposée en mini-opérations élémentaires. Celles-ci sont généralement au nombre de trois : *Recherche de l'instruction*, *Décodage* et *Exécution*.

Recherche de l'instruction (Fetch)

Le contenu du compteur de programme est placé sur le bus d'adresses (c'est l'unité de contrôle qui établit la connexion). L'unité de contrôle (UC) émet un ordre de lecture de la case mémoire dont le contenu sera ensuite acheminé à travers le bus de données au registre instruction (**RI**).

Remarque : Dès la mise sous tension ou après un RESET, le compteur de programme est initialisé par une valeur (adresse) fixée par le constructeur du microprocesseur, appelée Vecteur RESET.

Décodage (Decode)

Le registre d'instruction (**RI**) contient le **code opératoire**. L'unité de contrôle décode le contenu de **RI** pour savoir la nature de l'opération à effectuer (addition, rotation, ...) et incrémente le compteur de programme (PC) pour pointer sur l'instruction suivante.

Exécution (Execute)

Le cycle d'exécution varie en fonction de l'architecture et le type de l'instruction. Mais d'une manière générale c'est l'ALU qui exécute l'instruction en cours et positionne les indicateurs du registre d'état.

La Figure 2 illustre la séquence de déroulement de ces étapes. Chaque instruction est équivalente à une suite de mini-opérations exécutées dans un ordre précis. C'est ainsi qu'un microprocesseur procède à l'exécution de chaque instruction.

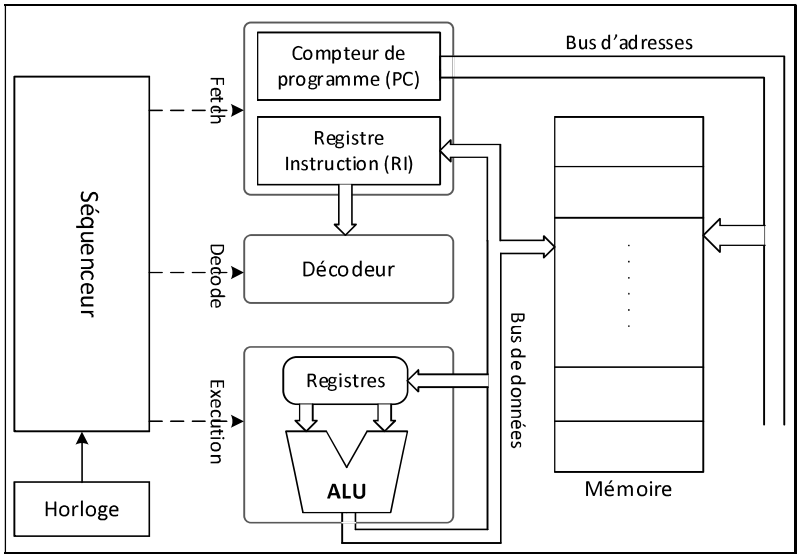


Figure 2 : Etapes d'exécution d'une instruction

Ces micro-opérations (fetch, decode, execute) sont cadencées au rythme d'horloge qui pilote le séquenceur. La durée de traitement d'une instruction s'appelle cycle d'instruction ou ***cycle machine***.

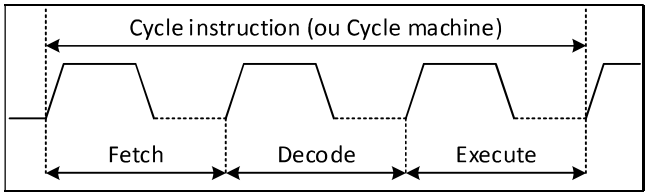


Figure 3 : Cycle d'exécution d'une instruction

Le nombre de périodes d'horloge nécessaires à l'exécution d'une instruction dépend de l'architecture du processeur et du mode d'adressage. Le microcontrôleur PIC que nous allons étudier, exécute toutes les instructions (hormis les instructions de saut) sur quatre périodes d'horloge.

DIFFERENTES ARCHITECTURES

Les différentes unités sont organisées suivant deux architectures :

- **L'architecture Von Neuman** (du nom d'un des savants qui a contribué à la mise au point des premiers ordinateurs). La mémoire de programme, la mémoire de données et les périphériques d'entrées/sorties partagent le même bus d'adresses et de données.
- **L'architecture Harvard**, sépare systématiquement la mémoire de programme de la mémoire de données : l'adressage de ces mémoires est indépendant. Ce type d'architecture favorise l'accès simultané aux mémoires de programme et de données.

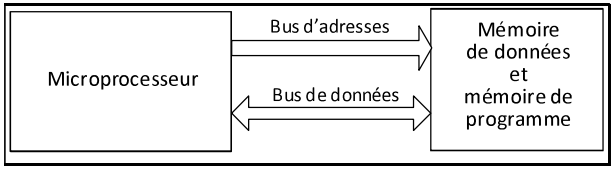


Figure 4 : Architecture Von Neumann

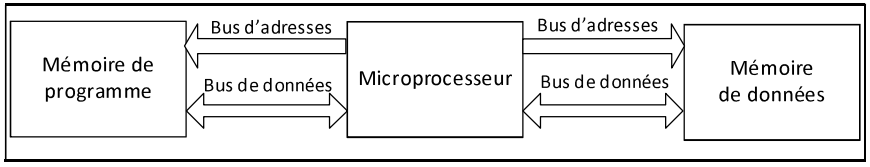


Figure 5 : Architecture Harvard

Au niveau du jeu d'instructions, les microprocesseurs se réparties en deux grandes catégories appelées CISC et RISC :

- **Architecture CISC** (Complex Instruction Set Computer) : une instruction peut effectuer plusieurs opérations élémentaires (faire par exemple une opération arithmétique avec chargement du résultat dans la mémoire). La longueur de l'instruction et le temps d'exécution varient d'une instruction à l'autre.
- **Architecture RISC** (Reduced Instruction Set Computer) : les processeurs RISC possèdent un jeu d'instruction réduit où chaque instruction effectue une opération élémentaire. Seules les instructions *Load* et *Store* accèdent à la mémoire. La plupart des instructions ont la même taille et s'exécutent sur un seul cycle machine.

LES PROCESSEURS SPECIALISES

Les microcontrôleurs

Au début de la commercialisation des microprocesseurs, un système minimum était obligatoirement constitué de plusieurs circuits intégrés.

Les microcontrôleurs sont apparus suite au progrès considérable de l'intégration des composants. Un microcontrôleur regroupe dans un même boîtier les éléments essentiels d'un système à microprocesseur, à savoir, la CPU, les mémoires RAM et ROM et quelques interfaces d'entrées-sorties (PORTS, Timers, ADC ...). Il s'agit d'un dispositif de contrôle, dont sa conception met l'accent sur la réduction du coût de développement des applications embarquées.

Les processeurs de traitement de signal (DSP)

Les DSP (Digital Signal Processor) sont des processeurs optimisés pour l'exécution des applications de traitement numérique de signal (filtrage numérique, convolution, transformée de Fourrier rapide…). Les DSP sont employés dans les modems, les téléphones mobiles, les appareils multimédia …).

LES MEMOIRES

Les mémoires sont des circuits intégrés à grande échelle d'intégration, capables de sauvegarder des informations *binaires* de façon permanente ou temporaire. Elles sont liées étroitement aux microprocesseurs puisqu'elles constituent l'élément de stockage de premier niveau.

On distingue deux types de mémoires :

- Les mémoires vives (RAM : Random Access Memoy) : Ce sont des mémoires volatiles, le maintien de l'information dépend de la présente de l'alimentation. Toute coupure du courant provoque la perte des informations. Elles sont utilisées pour stocker généralement les données temporaires.
- Les mémoires mortes (ROM : Read Only Memory) : gardent les informations même en absence d'alimentation. Ces mémoires contiennent des informations figées (souvent des programmes) et que l'accès ne se fait qu'en lecture seule.

Schéma synoptique d'une mémoire

Une mémoire est accessible via, un bus d'adresses, un bus de données et les signaux de lecture et d'écriture.

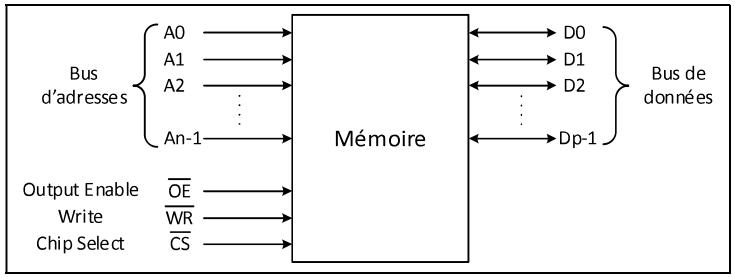


Figure 6 : schéma de connexion d'une mémoire

La capacité de la mémoire est donnée par la relation $C = 2^n$ avec n est le nombre de lignes d'adresses. Elle est exprimée en KBits ou KOctets (1K = 1024).

Cycles de Lecture / Ecriture

Cycle de lecture : Le cycle de lecture commence à l'instant où le microprocesseur met l'adresse de la case mémoire sur le bus, cette adresse

doit rester stable pendant la durée totale du cycle t_{RC} (Read Cycle Time). Le circuit de décodage met la ligne $\overline{OE}$ (Output Enable) à l'état bas indiquant ainsi la présence d'une adresse stable sur le bus. Le mémoire répond après une durée appelée temps d'accès t_{AA} (Adress Acces time) en plaçant le contenu de la case sur le bus de données. Le paramètre t_{OE} (Output Enable Time) définit le temps nécessaire à la mémoire pour mettre une donnée stable sur le bus. A partir de cet instant la donnée peut être récupérée. Après le retour de de la ligne $\overline{OE}$ à l'état haut, le bus de données maintien la donnée pendant un temps t_{OD} avant de passer à l'état haute impédance.

Cycle d'écriture : Le cycle d'écriture est similaire à celui de lecture. Le microprocesseur fournit l'adresse et la donnée à mémoriser. Après la stabilisation de l'adresse, la ligne $\overline{WR}$ est forcée à l'état bas. Pendant la durée de cette impulsion t_{PW}, l'adresse et la donnée doivent être maintenues stables.

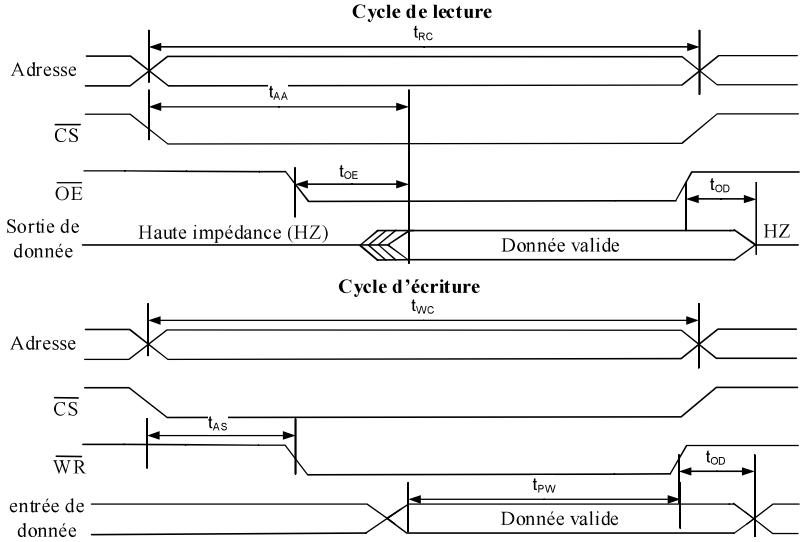


Figure 7 : cycles de lecture/écriture d'une mémoire

QUESTIONS ET EXERCICES CORRIGES

Questions

1. Que signifie RAM et ROM ? Quel est le rôle de chacune d'elles dans un calculateur ?
2. Que veut dire une mémoire non volatile ?
3. Que signifie CPU ? Expliquer sa fonction dans un calculateur ?
4. Citer les trois composants essentiels dans un calculateur.
5. Citer les différents bus dans un système à microprocesseur.
6. Citer les trois parties essentielles d'un microprocesseur.
7. Donner la capacité d'adressage d'un microprocesseur ayant 16 lignes d'adresses.
8. Quel est le rôle des registres internes du microprocesseur ?
9. Quel est le rôle du compteur de programme ?
10. Qu'appelle-t-on l'emplacement, où le microprocesseur cherche à exécuter la première instruction après la mise sous tension ?

Réponses

1. RAM : Random Access Memory (mémoire à accès aléatoire) ; utilisé pour le stockage temporaire des données lors de l'exécution du programme.
   a) ROM : Read Only Memory (mémoire à lecture seule), stocke en permanence les instructions que le microprocesseur doit exécuter.
2. Une mémoire non volatile conserve les données même en absence d'alimentation.
3. CPU : Central Processing Unit (unité centrale de traitement), peut être considéré comme le cerveau du calculateur. Il exécute le programme et contrôle tous les autres circuits du calculateur.
4. CPU, Mémoires et interfaces d'entrées/sorties (E/S).
5. Bus d'adresses : utilisé pour repérer l'emplacement mémoire auquel le microprocesseur veut accéder.
   b) Bus de données : véhicule l'information entre le microprocesseur et les autres composants.
   c) Bus de contrôle : synchronise le flux d'informations avec tous les composants adressables.
6. Unités de contrôle (UC), l'unité arithmétique et logique (UAL) et les registres internes.
7. $2^{16} = 65536\ cases$
8. Stockage temporaire des données. D'autres registres utilisés pour l'adressage des mémoires.

9. Repère l'évolution de l'exécution du programme. Il contient toujours l'adresse de la prochaine instruction à exécuter.
10. Vecteur RESET, il est placé au début ou bien à la fin de l'espace adressable. Selon l'architecture du processeur, cette case peut contenir la première instruction à exécuter ou son adresse.

Exercice

Le schéma de connexion de deux mémoires avec un microprocesseur est illustré à la Figure 8.

1. Quelle est l'architecture de ce microprocesseur ?
2. Quelle est la largeur du bus d'adresses et du bus de données.
3. Donner la capacité de chaque mémoire en kbits et en koctets.
4. On veut adresser la mémoire RAM1 à partir de l'adresse 0000H et la ROM1 à partir de l'adresse E000H.
   a) Compléter le Tableau 1, en indiquant dans la troisième colonne l'adresse la plus basse et l'adresse la plus haute de chaque mémoire en hexadécimal.

Tableau 1 : plage d'adressage des mémoires

		Adr. en Hexa.	A15 …	A12	A11 … A8	A7 … A4	A3 … A0
RAM1	Adr. basse						
	Adr. haute						
ROM1	Adr. basse						
	Adr. haute						

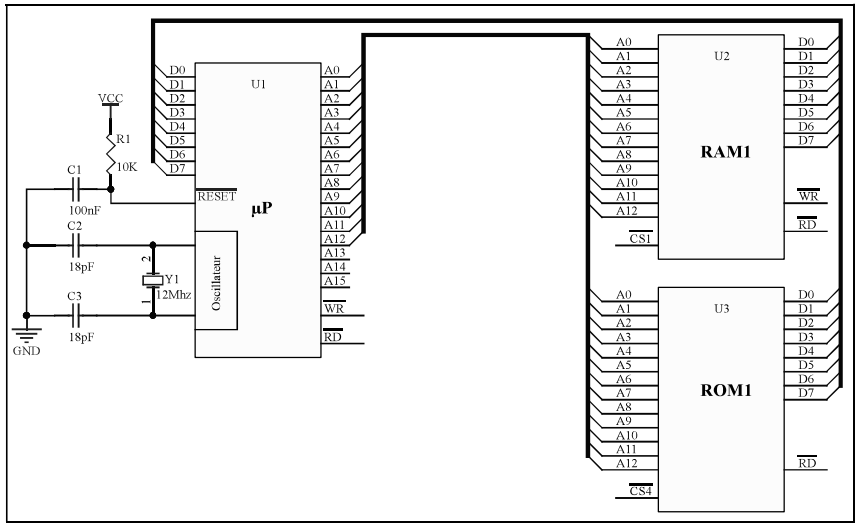


Figure 8 : schéma de connexion du microprocesseur avec deux mémoires

b) Donner les équations logiques des lignes $\overline{CS1}$ et $\overline{CS4}$ en fonction des lignes d'adresses A15, A14 et A13.

c) Compéter alors le schéma de décodage d'adresse en utilisant des opérateurs logiques.

5. On veut étendre la capacité mémoire de cette carte en ajoutant deux mémoires RAM2 et ROM2 identiques aux précédentes.

a) Donner l'adresse de base des nouvelles mémoires, si on veut que deux mémoires de même type soient adjacentes.

b) 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.

c) Donner le schéma de décodage d'adresses de toutes les mémoires en utilisant le circuit intégré 74138.

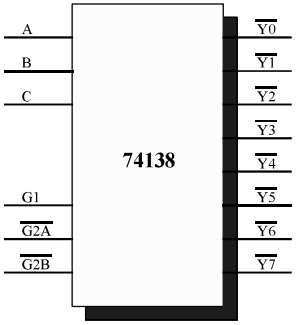


Tableau 2 : table de vérité du 74138

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

6. On suppose que le programme utilisateur commence au début de la mémoire de programme (ROM2), et que le vecteur RESET est situé à l'adresse FFFEH.

Donner le contenu des cases mémoires d'adresses FFFEH et FFFFH (ce microprocesseur stocke les données selon le format *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).

Solution

1. Architecture Van Neumann : puisque les mémoires partagent le même bus d'adresses et bus de données.
2. 16 bits d'adresses et 8 bits de données.
3. Les deux mémoires ont la même capacité, car elles ont le même nombre de lignes d'adresses.

$$C = 2^{13} = 8192 \text{ Octets} = 8 \text{ koctets} = 64 \text{ kbits}$$

4. RAM1 et ROM1

a) Détermination des adresses

Tableau 3 : plage d'adressage des mémoires

		Adr. en Hexa	A15 ...	A12	A11 .. A8	A7 ... A4	A3 ... A0
RAM1	Adr. basse	0000H	0 0 0	0	0 0 0 0	0000	0000
	Adr. haute	1FFFH	0 0 0	1	1 1 1 1	1111	1111
ROM1	Adr. basse	E000H	1 1 1	0	0 0 0 0	0000	0000
	Adr. haute	FFFFH	1 1 1	1	1 1 1 1	1111	1111

b) Vous remarquez bien que la RAM1 est sélectionnée pour la combinaison (A15,A14,A13)=(0,0,0).

Donc $CS1 = \overline{A15} \cdot \overline{A14} \cdot \overline{A13}$ d'où $\overline{CS1} = \overline{\overline{A15} \cdot \overline{A14} \cdot \overline{A13}}$

$$\overline{CS1} = A15 + A14 + A13$$

De même la ROM1, elle est sélectionnée quand (A15,A14,A13)=(1,1,1), donc $CS4 = A15.A14.A13$

d'où $\overline{CS4} = \overline{A15.A14.A13}$

c) Schéma de décodage d'adresses

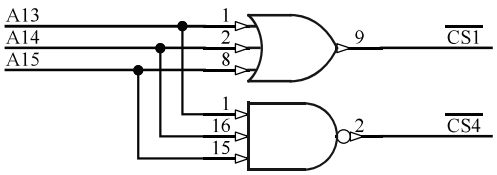


Figure 9 : circuit de décodage d'adresses

5. Ajout des mémoires

a) Adresses de base des nouvelles mémoires

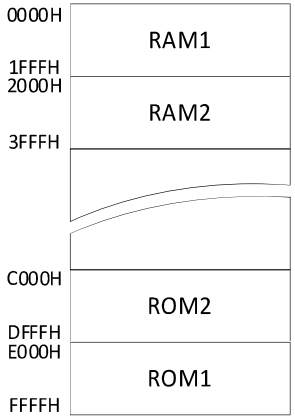
L'adresse de la dernière case de la RAM1 est 1FFFH, la RAM2 est placée juste après, donc son adresse de base est 1FFFH+1 = 2000H.

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

L'adresse la plus haute de la RAM2 = adresse de base + la capacité de la mémoire -1=2000H + 2000H -1= 3FFFH.

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

L'adresse la plus haute = C000H+2000H-1=DFFFH



Espace adressable

b) Equations de sélection des nouvelles mémoires :

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

2000H = 0010000000000000 et 3FFFH = 0011111111111111. Les bits qui ne changent pas sont (A15,A14,A13) = (0,0,1), nous pouvons déduire que $CS2 = \overline{A15} \cdot \overline{A14} \cdot A13$ d'où $\overline{CS2} = \overline{\overline{A15} \cdot \overline{A14} \cdot A13}$

De même pour la ROM2. En comparant les adresses de début et de fin, nous remarquons que les bits qui ne changent pas sont (A15,A14,A13) = (1,1,0), alors $CS3 = A15 \cdot A14 \cdot \overline{A13}$

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

c) Schéma de décodage d'adresse : D'après la table de vérité du circuit 74138, le lignes de sélection $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$, et $\overline{CS4}$ correspondent bien aux sorties $\overline{Y0}$, $\overline{Y1}$, $\overline{Y6}$ et $\overline{Y7}$.

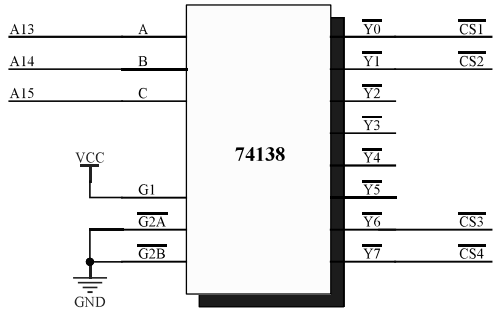


Figure 10 : circuit de décodage avec le circuit 74138

6. 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. Vous devez stocker la valeur 00H à l'adresse FFFEH et la valeur C0H à l'adresse FFFFH.