

CORRECTION

Section : S	Option : Sciences de l'ingénieur	Discipline : Génie Électrique
La mémorisation d'une donnée numérique		
Domaine d'application : Traitement programmé de l'information	Type de document : Cours	Classe : Terminale
		Date :

I - Identification de la fonction

Introduction : Tout système numérique a besoin de mémoriser les données sur lesquelles il travaille, dans le but de différer leur traitement.

Rappel : une donnée logique est constituée d'un seul bit et elle est mémorisable en utilisant une bascule.

Définition : Une donnée numérique est un ensemble de données logiques (sous forme d'un mot binaire de n bits) et sera mémorisé dans un registre. Un registre est un montage composé d'un ensemble de bascules.

II - Les registres

Un registre est caractérisé par :

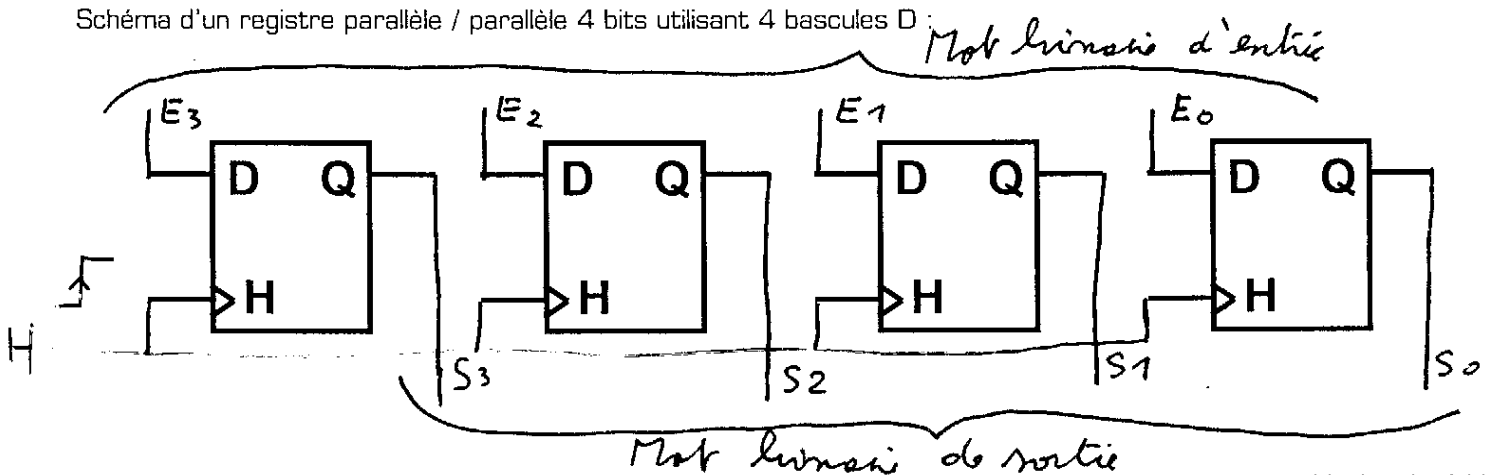
- * sa capacité de mémorisation (nombre de bits)
- * son mode d'écriture (série ou parallèle)
- * son mode de lecture (série ou parallèle)

II - 1 - Le registre à entrée parallèle et à sortie parallèle

Ce registre est aussi appelé « registre parallèle / parallèle ».

Le mode parallèle signifie que l'ensemble des bits du mot binaire sont mémorisés simultanément dans le registre, en une seule opération.

Schéma d'un registre parallèle / parallèle 4 bits utilisant 4 bascules D :



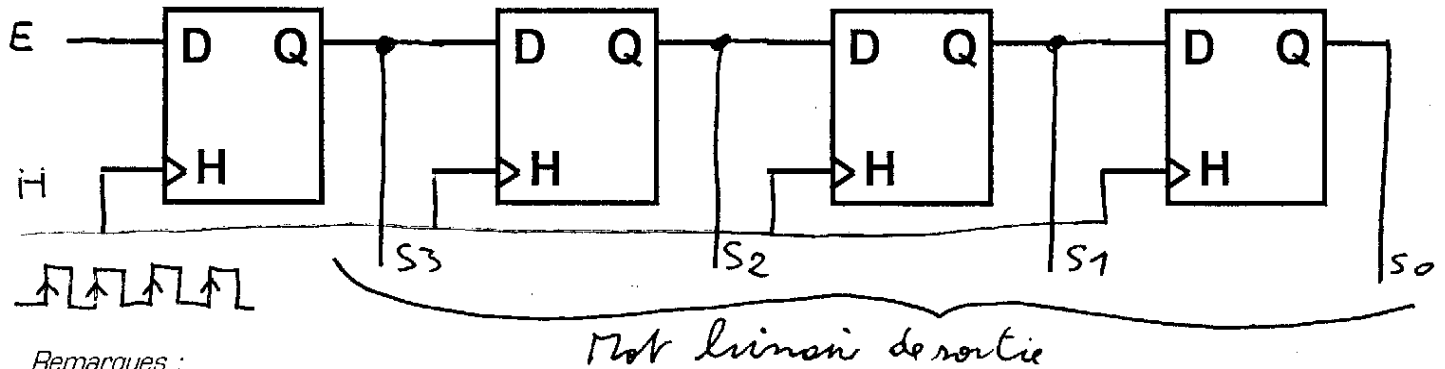
Le chargement du registre s'effectue de manière globale à chaque front actif de l'horloge. Le mot binaire de 4 bits mémorisé est disponible à tout moment sur les sorties Q des bascules.

II - 2 - Le registre à entrée série et à sortie parallèle

Ce registre est aussi appelé « registre série / parallèle ».

Le mode série signifie que le mot binaire se mémorise bit par bit appliqué bit à bit sur l'entrée de la première bascule jusqu'à ce que le registre soit plein.

Schéma d'un registre série / parallèle 4 bits utilisant 4 bascules D :



Remarques :

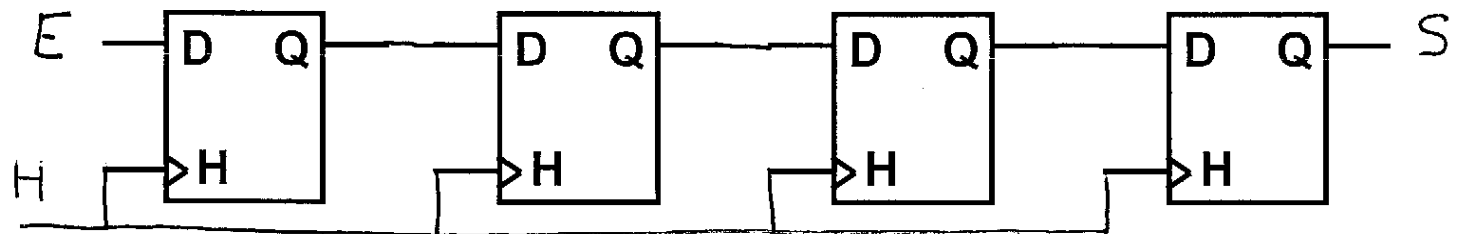
- * chaque sortie Q d'une bascule est reliée à l'entrée D de la bascule suivante
- * à chaque front actif de l'horloge les bits mémorisés dans le registre se décalent d'un rang
- * un tel registre est appelé un **registre à décalage**
- * pour un registre à décalage à n bits il faudra attendre n impulsions sur l'horloge avant que le mot binaire soit entièrement mémorisé

II - 3 - Le registre à entrée série et à sortie série

Ce registre est aussi appelé « registre série / série ».

C'est un registre à décalage possédant une seule entrée de donnée et une seule sortie.

Schéma d'un registre série / série 4 bits utilisant 4 bascules D :



Remarques :

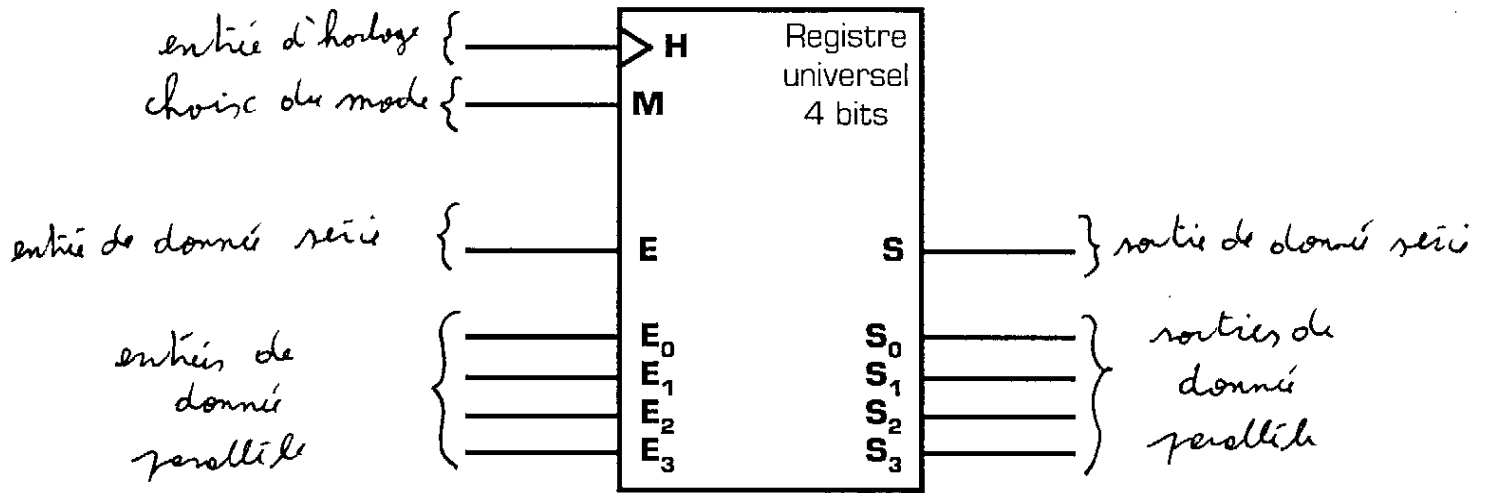
- * le chargement du registre s'effectue bit à bit en 4 étapes, en envoyant 4 impulsions sur l'entrée d'horloge
- * la lecture du mot binaire mémorisé s'effectue également en 4 étapes, avec les 4 impulsions suivantes arrivant sur H
- * le premier bit entré dans le registre sera le premier bit sortie

II - 4 - Le registre universel

Ce registre regroupe dans un seul circuit les différents types précédents, en permettant les modes de fonctionnement suivants :

- * chargement en série ou en parallèle
- * lecture en série ou en parallèle

symbole d'un registre universel 4 bits :



L'entree M permet de determiner le mode de fonctionnement utilise :

M	Mode de fonctionnement	Remarques
0	SÉRIE	Les données sont lues bit à bit sur l'entree E, et sont sorties du registre bit à bit sur la sortie S
1	PARALLÈLE	Le registre est chargé d'un coup à partir des entrees E ₀ à E ₃ . La donnée memorisee est disponible à tout moment sur les sorties S ₀ à S ₃ .

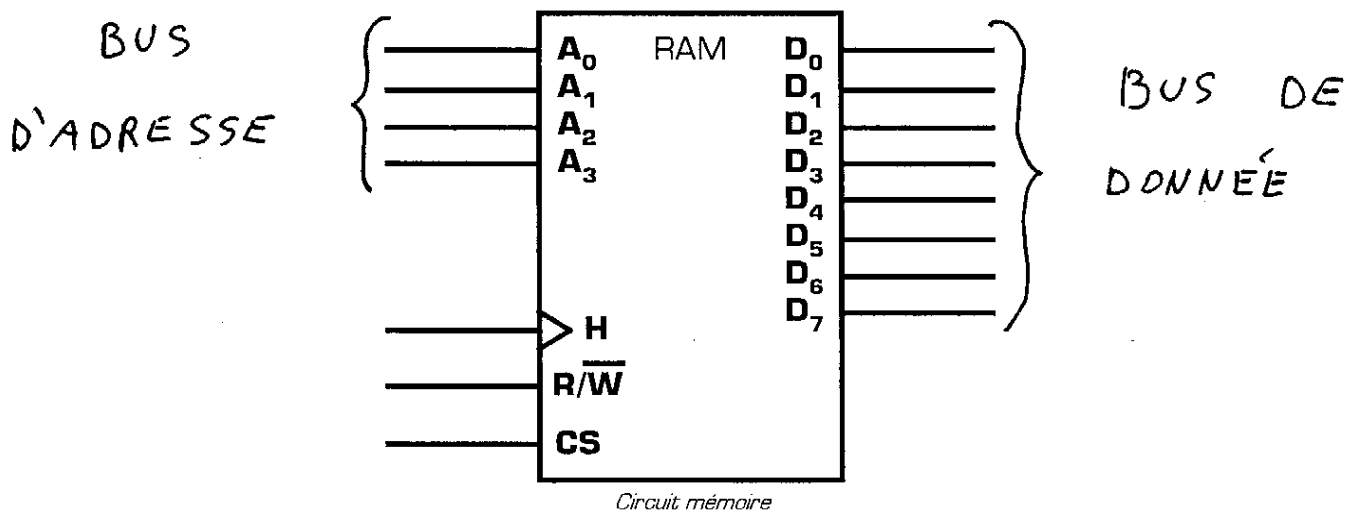
III - Les memoires

La somme d'informations que doit traiter tout systeme automatique necessite l'utilisation de circuits ayant la capacite de conserver un grand nombre de donnees numeriques. Ces circuits capables de recevoir et de restituer les informations portent le nom de « *memoire* ».

III - 1 - Structure des circuits memoire

Dans un circuit memoire, chaque information numerique est memorisee dans un registre, accessible à une adresse precise. Une memoire est donc un circuit compose d'un ensemble de registres, accessible individuellement grâce à une adresse.

Symbole d'un circuit memoire :



Remarques :

Le bus d'adresse : les entiers A_0 à A_3 permettent de définir l'adresse d'un des registres, et constituent le bus d'adresse.

Le bus de données : les lignes D_0 à D_7 constituent le bus de données | permettant d'écrire ou de lire les informations numériques dans le circuit mémoire. | Ces lignes sont bidirectionnelles :

- * ce sont des entrées dans le cas d'une écriture dans la mémoire
- * ce sont des sorties dans le cas d'une lecture

L'entrée R/\bar{W} : elle permet de définir l'action à réaliser.

R/\bar{W}	Sens de circulation des données
1	lecture d'une donnée dans la mémoire (mémoire \rightarrow extérieur)
0	écriture d'une donnée dans la mémoire (extérieur \rightarrow mémoire)

L'entrée CS : elle permet de valider ou d'inhiber le circuit

CS	Etat du boîtier mémoire
1	le circuit mémoire est validé
0	le circuit mémoire n'est pas validé (déconnexion du bus de données)

L'entrée d'horloge H : elle permet de déclencher l'action pré-programmée sur les autres entrées.

Dans l'exemple du circuit mémoire donnée en bas de la page 3, on remarque que :

- * le bus de données est composé de 8 bits : cela signifie que les mots binaires mémorisés ont une taille de 8 bits, ce circuit mémorise donc des octets.
- * Le bus d'adresse est composé de 4 bits, il permet donc d'accéder à $2^4 = 16$ registres internes différents
- * La capacité de ce boîtier mémoire est donc de 16 octets

III - 2 - Caractéristiques des circuits mémoire

Un circuit mémoire est caractérisé par :

- * La taille des mots binaires qu'il peut mémoriser [exemple : 8 bits]
- * La quantité de mots binaires qu'il peut mémoriser [exemple : 16 ko]

Rappel concernant les préfixes utilisés pour désigner les mémoires de grande capacité :

Nom	Préfixe	Quantité équivalente
1 kilo octet	1 ko	2^{10} octets = 1024 octets
1 méga octet	1 Mo	2^{10} ko = 1024 ko = 2^{20} octets
1 giga octet	1 Go	2^{10} Mo = 1024 Mo = 2^{30} octets
1 téra octet	1 To	2^{10} Go = 1024 Go = 2^{40} octets
1 péta octet	1 Po	2^{10} To = 1024 To = 2^{50} octets
1 exa octet	1 Eo	2^{10} Po = 1024 Po = 2^{60} octets
1 zetta octet	1 Zo	2^{10} Eo = 1024 Eo = 2^{70} octets
1 yotta octet	1 Yo	2^{10} Zo = 1024 Zo = 2^{80} octets

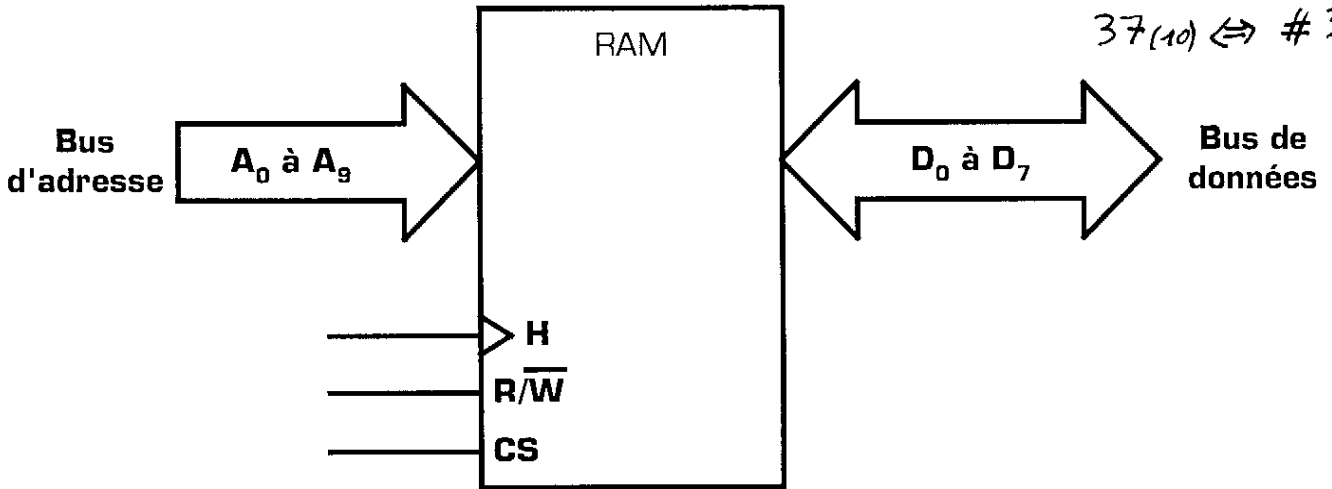
Le plus chaque circuit mémoire possède une ou plusieurs entrée[s] de validation du boîtier, noté parfois E [pour enable = validation] parfois CS [pour Chip Select = sélection du boîtier]. Ces entrées peuvent être active au niveau haut ou au niveau bas [voir le symbole du circuit ou la documentation constructeur pour le savoir]. *Rappel :*

III - 3 - Emploi de l'hexadécimal pour désigner les adresses mémoire

Prenons comme exemple le circuit mémoire suivant possédant un bus d'adresse de 10 bits : $0110_{(2)} \Leftrightarrow \%0110$

$F2_{(16)} \Leftrightarrow \$F2$

$37_{(10)} \Leftrightarrow \#37$



Cette mémoire mémorise des octets car son bus de données est sur *8 bits*.....
 Elle possède *10* lignes d'adresse [A₀ à A₉] permettant d'adresser $2^{10} = 1024$ registres internes de 8 bits chacun. La capacité de ce boîtier mémoire est donc de *1 ko*.....
 L'adresse du 1^{er} registre est : *%00,0000,0000* \equiv *\$000*.....
 L'adresse du 1024^{ème} registre est : *%11,1111,1111* \equiv *\$3FF*.....
 L'espace mémoire adressable par ce boîtier mémoire va donc de l'adresse *\$000*..... à l'adresse *\$3FF*....., ce qui représente *1024* adresses différentes.

Avantage de l'hexadécimal pour représenter les adresses mémoire :

- * Avec très peu de chiffres l'hexadécimal permet de représenter les grands espaces mémoire, ce qui n'est pas le cas en binaire.
- * La conversion entre le binaire et l'hexadécimal est immédiate (il suffit de regrouper les bits par paquets de 4), ce qui n'est pas le cas du décimal.

Pour toutes ces raisons l'hexadécimal sera toujours le système de numération utilisé pour exprimer les adresses mémoires dans le plan mémoire d'un système numérique.

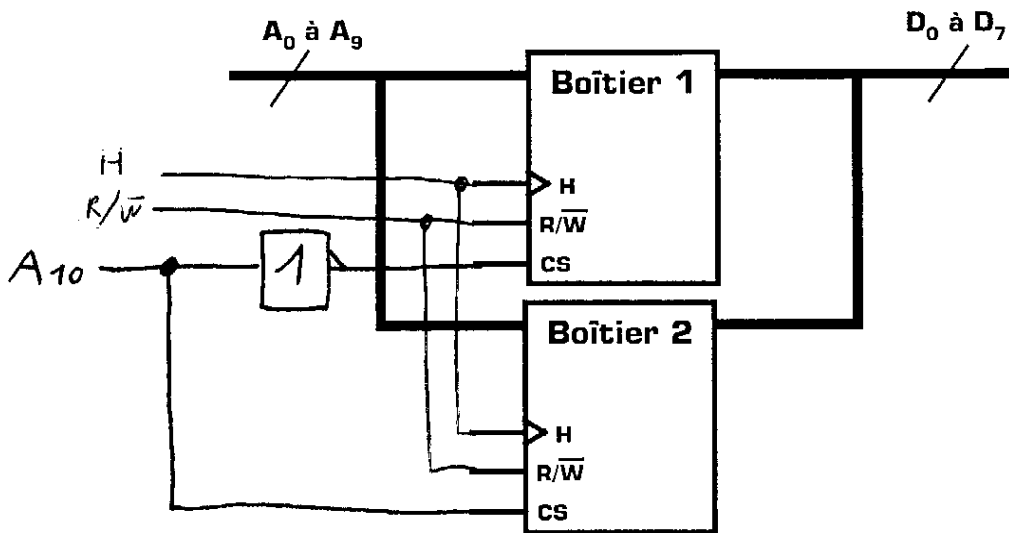
III - 4 - Utilisation de plusieurs boîtier mémoire pour augmenter la capacité

Le boîtier mémoire précédent a une capacité de 1 ko. Si on veut une mémoire de 2 ko il faudra alors utiliser 2 boîtiers différents :

$2 \text{ ko} = 2 \times 1024 \text{ octets} = 2048 \text{ octets} = 2^{11} \text{ octets}$

Il faut 11 lignes d'adresse pour adresser 2048 adresses différentes. L'idée est alors la suivante : on va ajouter un 11^{ème} bit sur le bus d'adresse [noté A₁₀] qui va permettre de sélectionner un boîtier ou l'autre selon qu'il est à 0 ou à 1 :

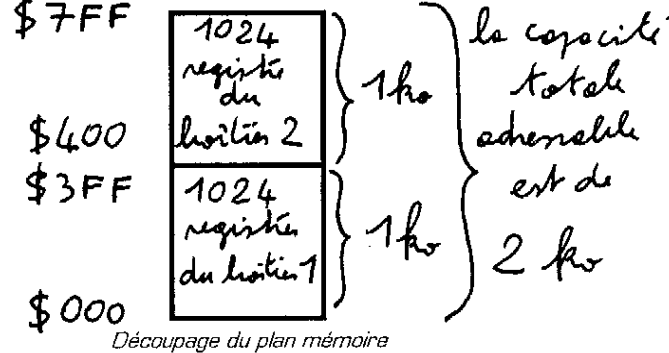
A ₁₀	Boîtier mémoire sélectionné
0	Boîtier 1 sélectionné
1	Boîtier 2 sélectionné



Réalisation d'une mémoire de 2 ko avec deux boîtiers de 1 ko chacun

Conséquence :

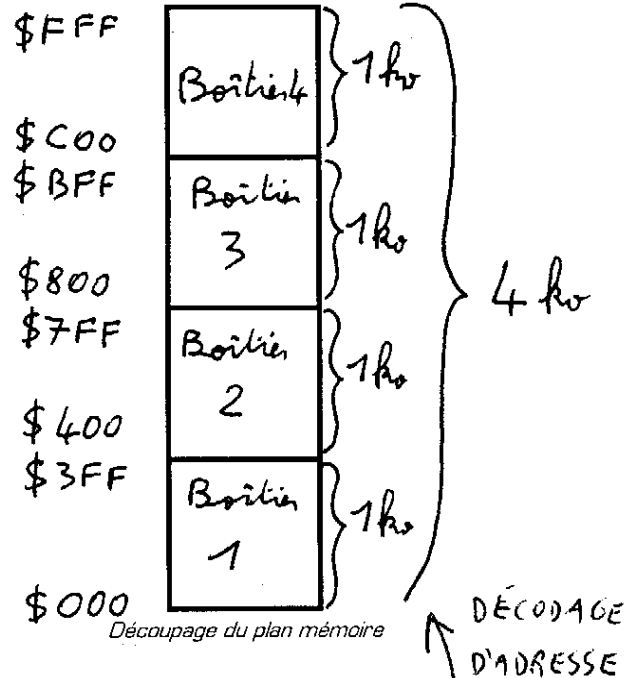
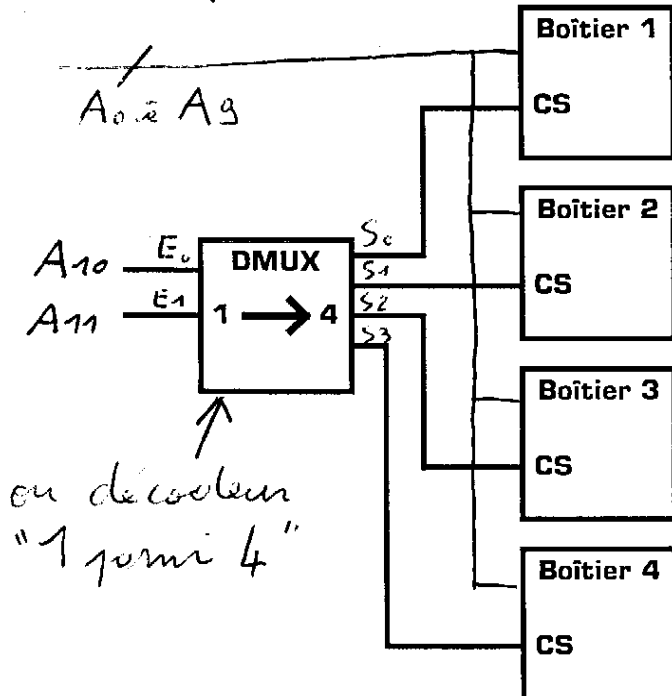
Les adresses $00000000000_{(2)}$ à $01111111111_{(2)}$ correspondent aux 1024 registres du boîtier 1 et les adresses $10000000000_{(2)}$ à $11111111111_{(2)}$ correspondent aux 1024 registres du boîtier 2. On peut alors représenter l'espace mémoire par le plan mémoire ci-contre. L'espace mémoire adressable va maintenant de l'adresse $000_{(16)}$ à l'adresse $7FF_{(16)}$, ce qui représente bien une capacité de 2 ko [2048 adresses].



Découpage du plan mémoire

Autre exemple : comment adresser 4ko en utilisant 4 boîtiers identiques de 1 ko chacun ?

La solution consiste à utiliser un démultiplexeur 1 vers 4, dont les 2 entrées d'adresse seront ajoutées au bus d'adresse :



Découpage du plan mémoire

Sélection des boîtier mémoire en fonction des bits de poids fort A10 et A11 du bus d'adresse :

A11	A10	Boîtier sélectionné	Intervalle des adresse de chaque boîtier en binaire
0	0	Boîtier 1	de $\%00000000000$ à $\%001111111111$
0	1	Boîtier 2	de $\%01000000000$ à $\%011111111111$
1	0	Boîtier 3	de $\%10000000000$ à $\%101111111111$
1	1	Boîtier 4	de $\%11000000000$ à $\%111111111111$

Retrouvez d'autres cours sur le site ressource

www.gecif.net

Téléchargez librement sur Gecif.net :

- ✍ **des cours et des TP de Génie Electrique**
- ✍ **des exercices et des évaluations avec corrections**
- ✍ **des ressources Automgen, ISIS Proteus et Flowcode**
- ✍ **des QCM pour réviser les cours et vous entraîner**
- ✍ **des logiciels d'électronique pour les installer chez vous**
- ✍ **des dossiers techniques de systèmes originaux**
- ✍ **des fiches pratiques sur tous les domaines des sciences de l'ingénieur**
- ✍ **des sujets de BAC**
- ✍ **et bien plus encore sur Gecif.net !**