Circuits logiques combinatoires et séquentiels

Guy Bégin

16 novembre 2022

Mémoires

Objectifs

- Pouvoir distinguer entre mémoire volatile et non-volatile
- Pouvoir distinguer entre mémoire volatile statique et dynamique
- Connaître l'organisation typique d'une mémoire et le fonctionnement de l'adressage
- Comprendre le fonctionnement d'un bus de données
- Pouvoir interpréter les cycles d'écriture et de lecture
- Pouvoir implémenter une fonction combinatoire arbitraire à l'aide d'une mémoire ROM
- Pouvoir utiliser un tableau de correspondance
- Être familier avec les différents types de mémoires non-volatiles

Mémoires

- Une mémoire est utilisée pour stocker des valeurs binaires à plus ou moins long terme.
- Généralement, l'information stockée dans la mémoire sera lue et acheminée dans des registres pour être traitée par un circuit logique de traitement.
- Les résultats du traitement seront typiquement stockés de nouveau dans la mémoire.
- Constituée d'un grand nombre de cellules permettant chacune de stocker un bit, la mémoire est dotée de mécanismes permettant d'accéder aux cellules pour en faire la lecture ou l'écriture.
- On distingue les mémoires non-volatiles (en anglais, *Read Only Memories*, (ROM)) et les mémoires volatiles (en anglais, *Random Access Memories*, (RAM)).

Mémoires non-volatiles

- Dans une mémoire ROM, les données sont stockées une fois pour toute.
- Elles y demeurent même après que la mémoire ait été mise hors tension.
- Dans le cycle de vie des données, il y a donc une écriture initiale, mais autant de lectures qu'on le souhaite.
- Il ne peut pas y avoir de récriture.

Mémoire ROM

- Une mémoire ROM est considérée comme un dispositif programmable, dans le sens que le processus d'écriture initial demande une action particulière, une procédure spécifique au niveau matériel.
- On verra plus loin d'autres dispositifs logiques programmables.
- La programmation d'une ROM se fait en agissant sur des connexions dites fusibles.
- Initialement, le fusible est comme un fil qui permet au signal de passer.
- En le programmant, le fusible devient un circuit ouvert qui ne laisse plus passer le signal.

Mémoires volatiles

- Une mémoire RAM stocke l'information de façon temporaire.
- En principe, le contenu est conservé tant que la mémoire est maintenue sous tension. Mais la réalité est un peu plus complexe, comme on le verra plus loin.
- L'opération d'écriture permet de stocker des valeurs, et la lecture permet d'extraire l'information de la mémoire.

RAM statique

- Une mémoire RAM statique consiste en un ensemble de loquets qui permettent de conserver des données binaires.
- L'information est maintenue tant que la mémoire est alimentée.

RAM dynamique

- Les mémoires RAM dynamiques stockent l'information sous la forme d'une charge capacitive au sein des transistors du circuit intégré.
- Comme cette charge se disperse au fil du temps, la mémoire doit être rafraîchie régulièrement, en y récrivant périodiquement à très court intervalle (millisecondes) la même valeur qui est déjà stockée.
- Les mémoires dynamiques consomment beaucoup moins que les mémoires statiques et offrent des capacités de stockage largement supérieures, car une cellule de mémoire comporte beaucoup moins d'éléments (essentiellement un transistor par cellule).
- En contrepartie, les temps d'accès aux mémoires statiques sont nettement meilleurs et on n'a pas à se préoccuper de rafraîchissement.

Adressage

- Les cellules des mémoires sont organisées en petits groupes appelés mots, de façon à ce que l'on puisse accéder à chaque groupe indépendamment.
- Toutes les cellules d'un mot sont lues ou écrites ensemble.
- Cet accès individuel aux mots, appelé adressage, est une caractéristique de flexibilité essentielle.
- Le temps d'accès aux données est le même, quel que soit l'endroit dans la mémoire où un mot en particulier est stocké.
- Les mots sont généralement constitués d'un nombre de bits multiple de huit : 8, 16 ou 32 bits sont des tailles de mots courantes.
- Un groupe de huit bits est appelé octet.

Décodeur d'adresses

- L'adressage se fait au moyen d'un décodeur d'adresses, qui est simplement un décodeur binaire tel que vu précédemment.
- Le nombre de bits d'adresse détermine la capacité (en nombre de mots) de la mémoire : pour k adresses, on aura 2^k mots distincts.
- Les tailles de mémoire sont souvent exprimées au moyen de multiplicateurs : K (kilo) correspondant à 2^{10}, M (mega) correspondant à 2^{20} ou G (giga) correspondant à 2^{30}.

Schéma d'une mémoire

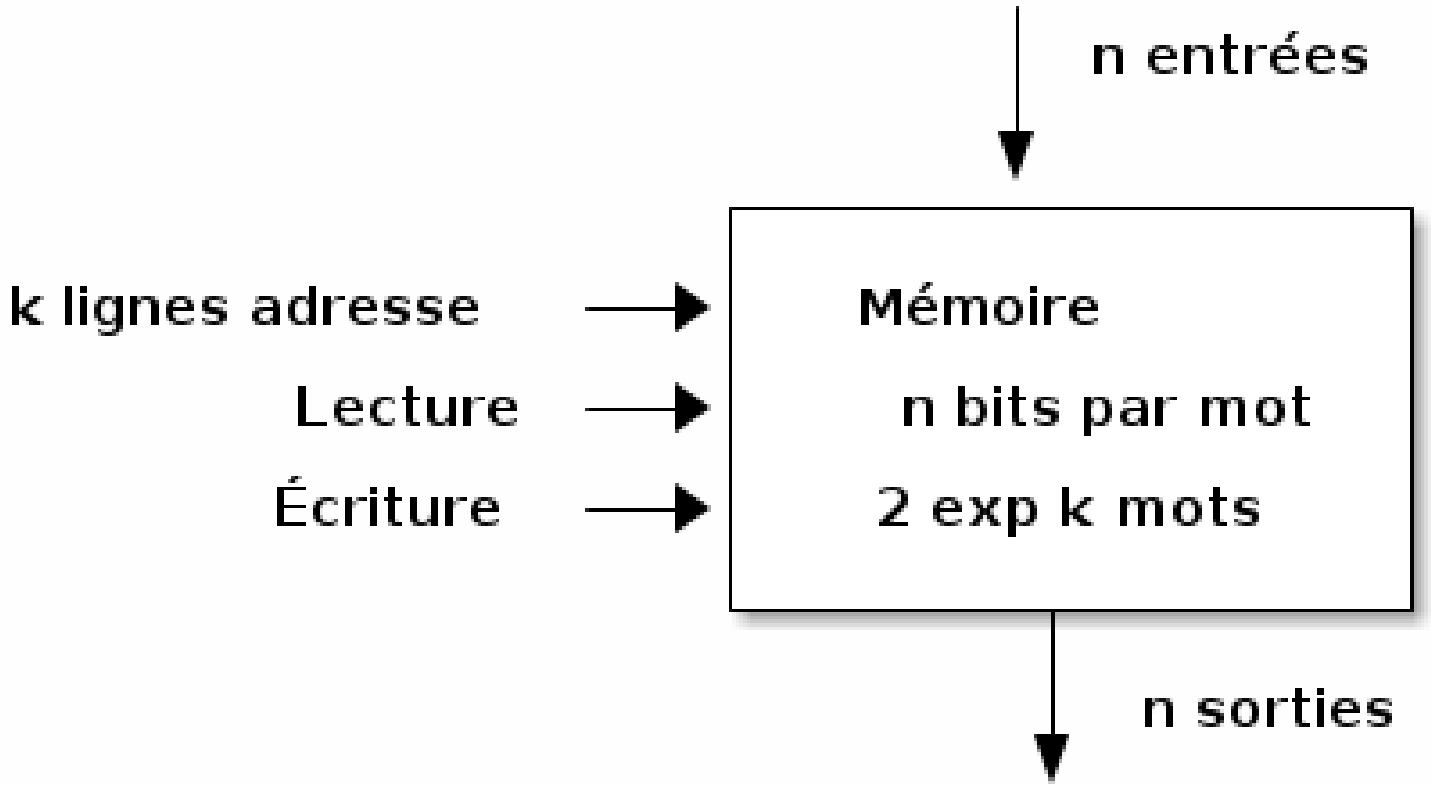

Figure 1 – Schéma d'une mémoire

Lecture et écriture

- L'opération choisie, écriture ou lecture, est commandée par une ou des entrées à cet effet.
- L'accès à un espace-mémoire (un mot) se fait selon une séquence bien précise.

Écriture

Pour une écriture :

1. Les bits d'adresse du mot sont appliqués aux lignes d'adresses.
2. Les données à écrire sont appliquées aux lignes d'entrée.
3. On active l'entrée de commande `Écriture`.
4. Les données de l'entrée sont alors stockées dans la case-mémoire adressée.

Lecture

Pour une lecture :

1. Les bits d'adresse du mot sont appliqués aux lignes d'adresses.
2. On active l'entrée de commande `Lecture`.

Les données présentes dans la case-mémoire adressée sont ensuite disponibles à la sortie de la mémoire.

Signal de contrôle combiné

- Les mémoires disponibles sur le marché optent souvent pour une combinaison des signaux de contrôle, avec un seul signal qui détermine le sens de l'action, comme on peut le voir dans le tableau 1.
- Le signal `Enable`, parfois appelé `Chip select`, permet d'activer une mémoire dans un ensemble où plusieurs mémoires sont utilisées.

Signaux de contrôle d'une mémoire

Table 1 – Signaux de contrôle d'une mémoire

Enable	Lecture/écriture	Action
0	X	Aucune
1	0	Écriture
1	1	Lecture

Bus de données

- Pour acheminer les données lues ou à écrire dans la mémoire, on utilise des tampons émetteur-récepteurs de bus, organisés en vecteur, pour créer un bus de données qui permet un aller-retour des données, selon le sens de l'action.
- Cela permet de diminuer de moitié le nombre de connexions nécessaires pour l'échange des données.
- Un signal dérivé des signaux `Lecture/écriture` et `Chip select (CS)` est typiquement utilisé pour commander l'entrée de contrôle (voir figure 2).

Bus de données, 8 bits

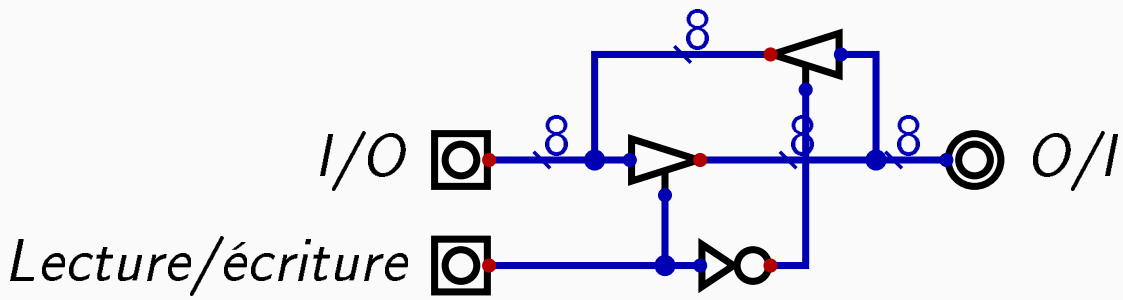

Figure 2 – Bus de données, 8 bits

Chronogrammes

- La figure 3 présente un chronogramme qui décrit l'opération d'écriture dans une mémoire RAM.
- Les valeurs d'adresses sont d'abord présentée aux entrées d'adressage.
- Le signal `L/not E` est amené au niveau bas, en même temps que le signal `CS` est activé (au niveau bas).
- Après un court délai, les données sont mises sur le bus de données et seront écrites dans la mémoire.
- On peut voir que les lignes du bus de données sont en mode haute impédance lorsque le bus est inactif (situation représentée symboliquement sur l'illustration par un signal situé entre les niveaux 0 et 1).

Mémoire RAM, chronogramme pour l'écriture

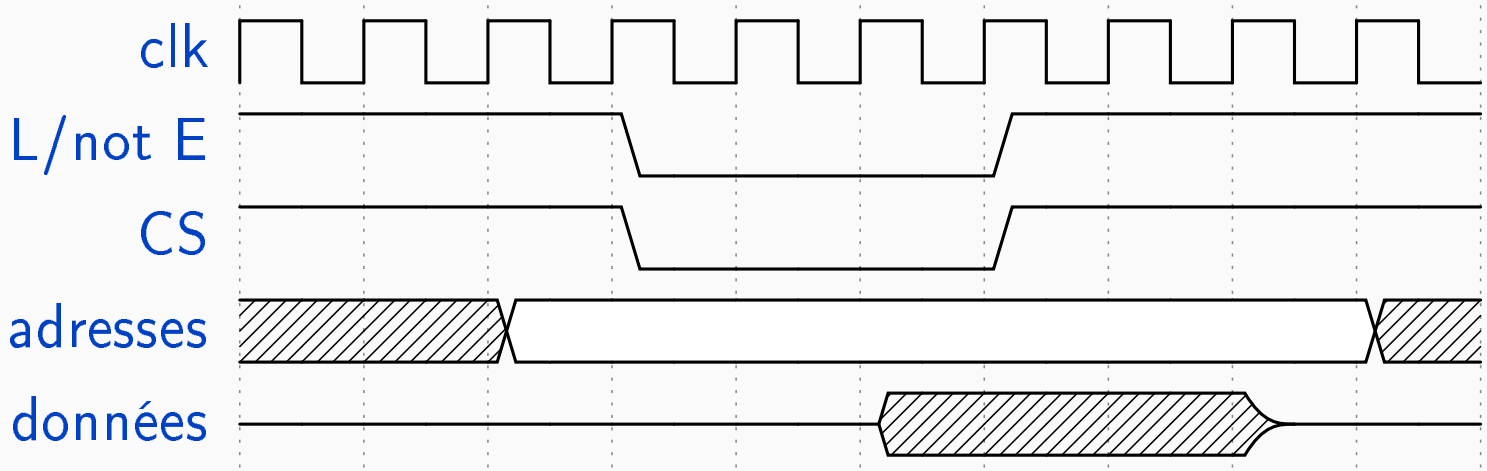

Figure 3 – Mémoire RAM, chronogramme pour l'écriture

Mémoire RAM, chronogramme pour la lecture

- La figure 4 présente un chronogramme qui décrit l'opération de lecture d'une mémoire RAM.
- Les valeurs d'adresses sont d'abord présentée aux entrées d'adressage.
- Le signal `L/not E` est maintenu au niveau élevé en même temps que le signal `CS` est activé (au niveau bas).
- Après un court délai, les données sont disponibles sur le bus de données.

Mémoire RAM, chronogramme pour la lecture

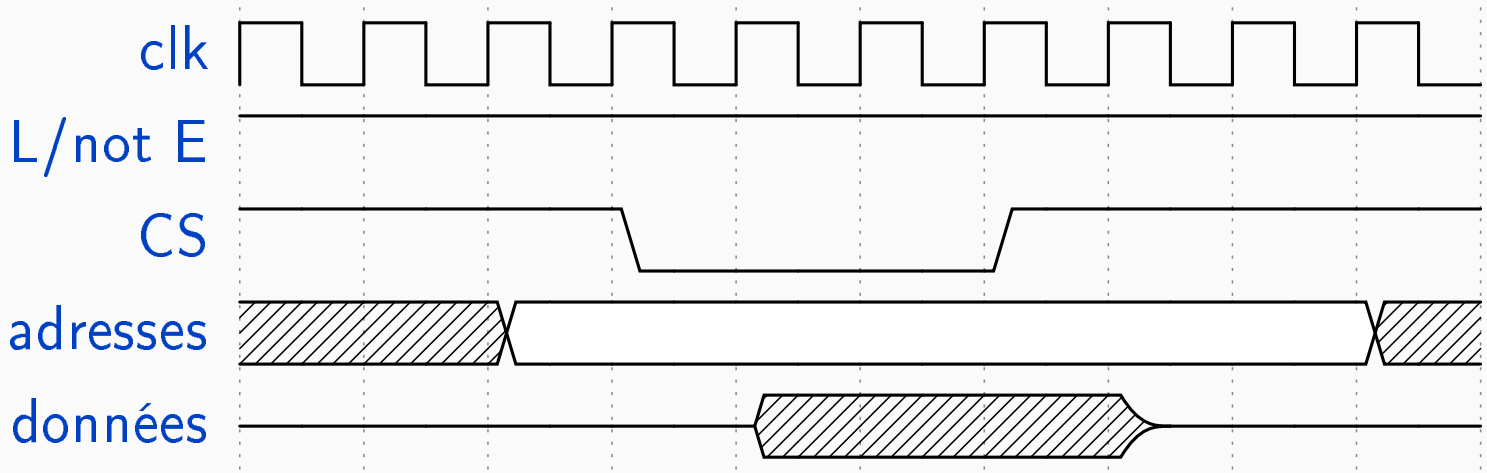

Figure 4 – Mémoire RAM, chronogramme pour la lecture

Cellule de base

- La cellule de base d'une mémoire RAM qui permet de stocker un bit est illustrée à la figure 5.
- Elle est construite autour d'un loquet SR et de portes logiques pour le contrôle.
- Une mémoire complète de m mots de taille n bits sera constituée d'une matrice de format $m \times n$ de telles cellules, avec un décodeur d'adresses pour sélectionner quel mot sera affecté par l'opération choisie.

Cellule mémoire RAM

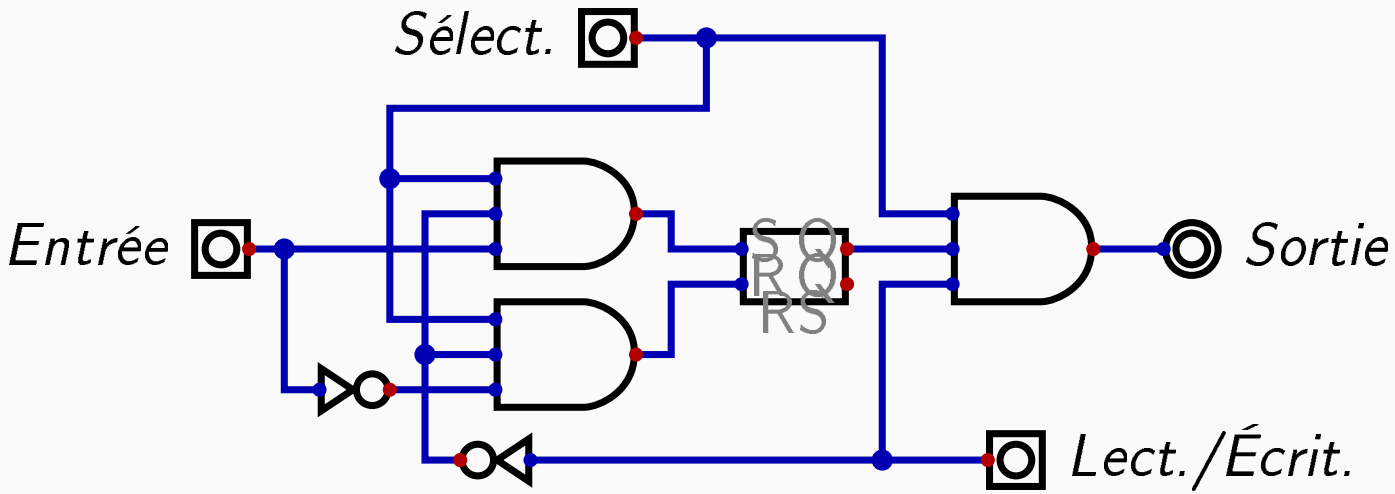

Figure 5 – Cellule mémoire RAM

Mémoires mortes

- Dans son mode d'utilisation normal, une mémoire morte peut seulement être lue.
- Il n'est donc pas nécessaire de préciser l'opération qui sera effectuée.
- Il y aura donc des entrées pour les adresses et un signal de contrôle de type `CS`.
- La figure 6 montre l'essentiel d'une mémoire ROM de 16 mots de 4 bits.
- Cette relativement petite mémoire comporte ainsi 64 intersections programmables, permettant de définir la valeur des 16 mots de mémoire de 8 bits chacun.

Modèle d'une mémoire ROM

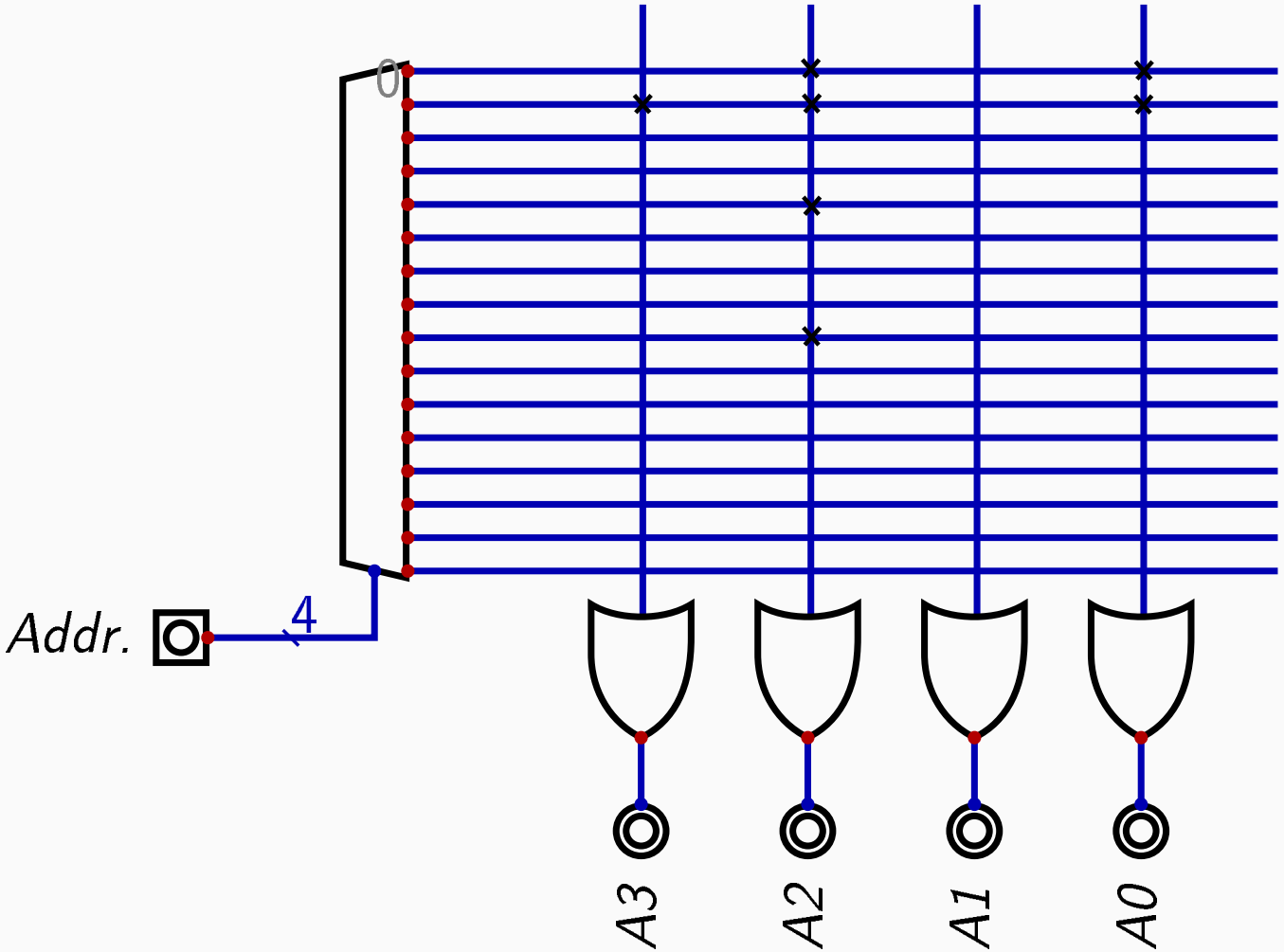

Figure 6 – Modèle d'une mémoire ROM

Mémoires mortes ... 2

- Un décodeur d'adresse permet de sélectionner quel mot sera lu, et la sortie est disponible sur les lignes $A_3, \ldots, A_0$.
- Pour simplifier la représentation de ce genre de configuration, on utilise une schématisation symbolique compacte pour les portes OU de sortie, dans laquelle chacune des 16 lignes horizontales représente en fait 16 entrées d'une porte OU.
- La présence d'une croix à l'intersection d'une ligne horizontale et d'une ligne verticale signifie que le signal de la ligne horizontale est connecté à une des entrées de la porte.
- Avec cette schématisation, on peut voir que les deux premiers mots stockés dans la mémoire illustrée dans l'exemple seraient 0101 et 1101.
- La même schématisation compacte s'emploie aussi pour des portes ET.

Implémentation de fonctions combinatoires

- Comme on l'a vu précédemment, un décodeur permet de générer les 2^k minterms possibles avec ses k entrées.
- Regrouper avec une porte OU les minterms d'une fonction permet d'implémenter cette fonction.
- Une mémoire ROM permet de faire exactement cela sans avoir rien à ajouter, car elle est munie d'un décodeur d'entrées et la porte OU de sortie fait déjà partie de la ROM.

Implémentation de fonctions combinatoires . . . 2

- On peut donc interpréter le fonctionnement d'une mémoire ROM de k bits d'adresse et avec des mots de taille m comme un dispositif qui permet, pour les entrées qui sont ses adresses, de mettre en oeuvre m fonctions combinatoires différentes (une par bit de mot) de k entrées.
- Par exemple, la sortie A_2 de la mémoire de la figure 6 implémente la fonction

$$A_2 = \sum(0, 1, 4, 8)$$

exprimée en somme de minterms.

Tableau de correspondance

- Cette approche qui consiste à réaliser une fonction logique combinatoire au moyen d'une mémoire qui spécifie, pour chaque combinaison d'entrée possible, une valeur de sortie, est largement utilisée dans les composants programmables.
- On parle alors de tableau de correspondance (en anglais, *LookUp Table*, (LUT)).
- Il s'agit ni plus ni moins que de stocker en mémoire le tableau de vérité de la fonction à réaliser.
- Dans les composants programmables, on utilise plutôt des mémoires RAM pour les tableaux de correspondance, afin que la configuration des fonctions puisse être changée selon l'application.

Catégories de mémoires ROM

- On distingue quatre grandes approches technologiques pour réaliser des mémoires mortes.
- Leurs usages typiques sont surtout déterminés par la façon de les configurer (on dit couramment *programmer*, même s'il s'agit d'un intervention au niveau du matériel).

Programmation par masque

- Dans la programmation par masque, la mémoire est programmée lors de la fabrication de la puce.
- Le fabricant se base sur un tableau de vérité fourni par le client pour établir des connexions qui seront implémentées (ou pas) dans le procédé de fabrication via des masques qui empêchent la déposition de matériau conducteur sur les couches du circuit intégré.
- Cette approche convient à la production de masse à grand volume.

Programmation sur mesure

- Dans la programmation sur mesure, on utilise un type de mémoire qui comporte initialement des connexions entre toutes les sorties du décodeur et toutes les entrées des portes OU de sortie (la mémoire en configuration initiale comporte des 1 partout).
- La programmation, qui peut se faire chez le développeur au moyen d'un dispositif de programmation (ou programmeur) spécialement conçu à cette fin, consiste à supprimer les connexions qui ne sont pas nécessaires en envoyant des impulsions à forte tension pour faire fondre les fusibles de connexions spécifiques.
- Un fusible fondu (pas de connexion) correspondant à un bit 0 dans la mémoire.
- Cette programmation est bien entendu irréversible : impossible de reconnecter une fois que le fusible est fondu.

PROM

- Avec un PROM (pour *Programmable ROM*), il est possible d'effacer la configuration dans son ensemble en soumettant la puce à une lumière ultraviolet pendant un certain temps.
- Ce bombardement énergétique permet de décharger les grilles flottantes des dispositifs qui implémente les connexions.
- La mémoire PROM peut être reconfigurée de nouveau.
- Avec la programmation électrique, il est possible de reconfigurer l'ensemble d'une mémoire dite EEPROM (*Electrically Erasable PROM*) en la soumettant à un signal électrique d'effacement.

Programmation *flash*

- Les ROM à programmation *flash* sont semblables aux EEPROM, mais la reconfiguration peut se faire adresse par adresse.
- Il est notamment possible de reconfigurer une mémoire sans la retirer de son circuit. Fonctionnellement, ces mémoires sont à mi-chemin entre la mémoire RAM et la mémoire sur disque.
- Les mémoires *flash* tendent d'ailleurs de plus en plus à remplacer ce dernier type dans les systèmes portables : téléphones, ordinateurs portables, etc.