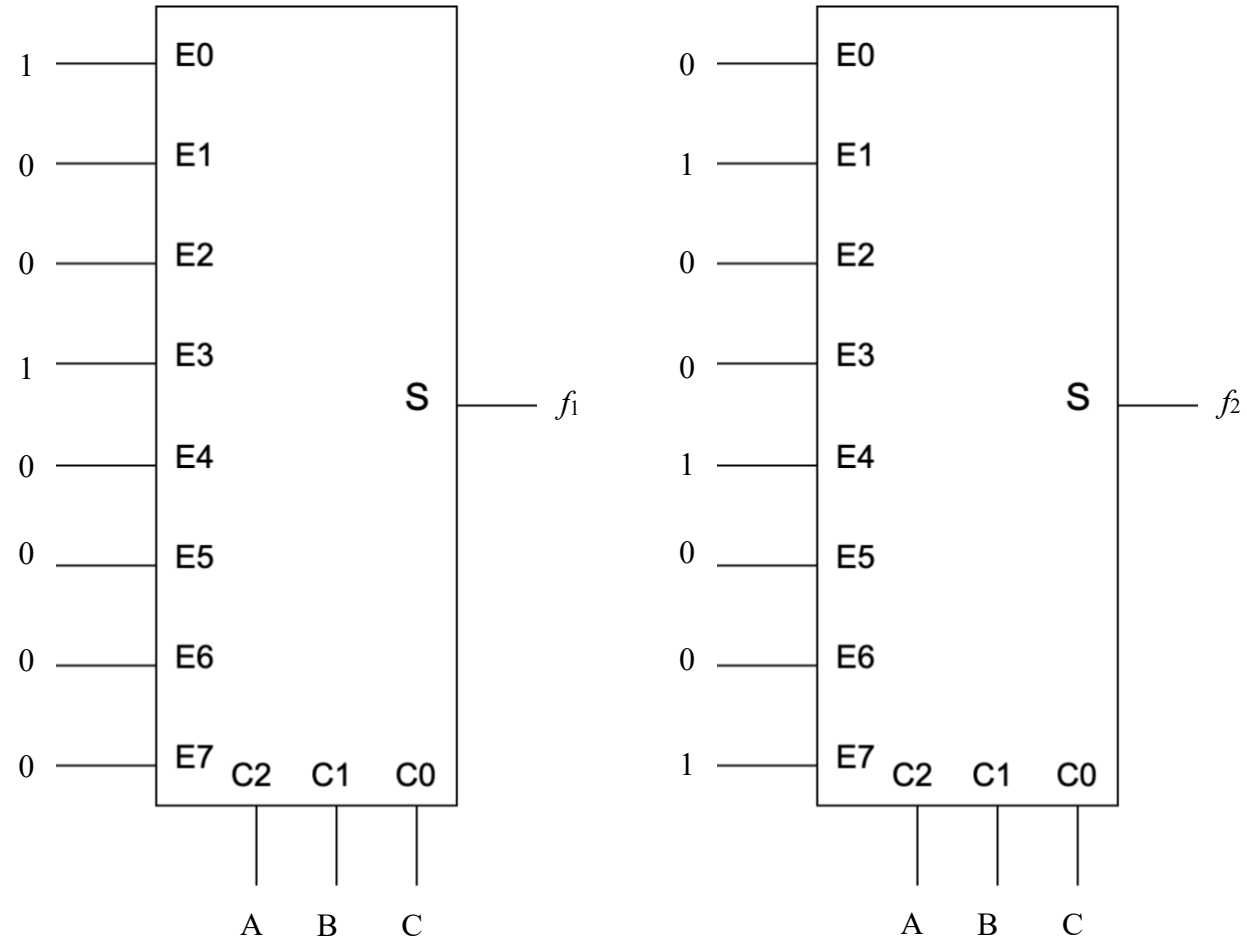
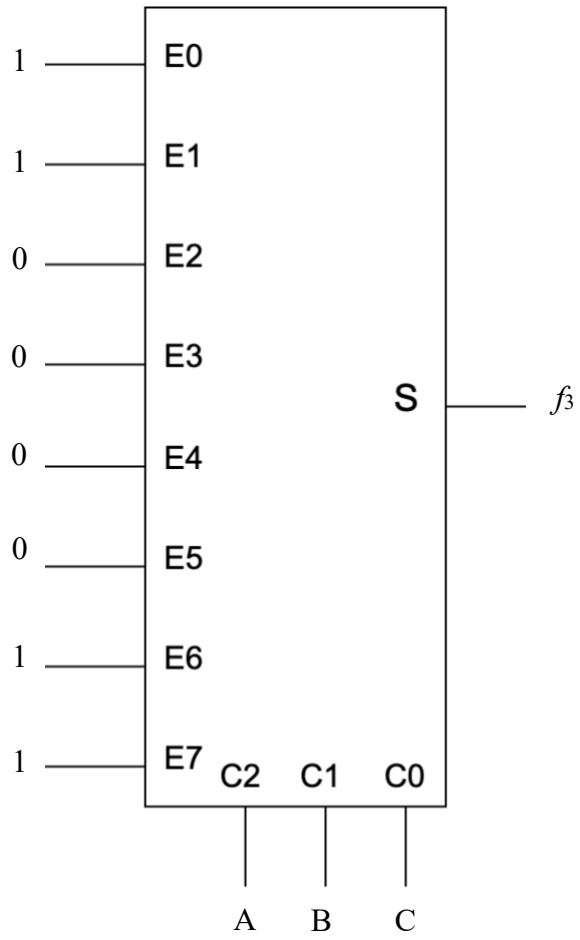
# *Correction PCD4 ELP111 : Circuits combinatoires*

## *Section 3.1 : MUX à tout faire*

Attention aux bits de poids fort et de poids faible !






## *Section 3.2 : Décodeur binaire / démultiplexeur*

Soit un **décodeur binaire** à 2 entrées $A_0$ et $A_1$ et à 4 sorties $S_0$, $S_1$, $S_2$, et $S_3$ tel qu'une seule sortie soit active à la fois et que le rang de la sortie active correspond à la valeur binaire présentée en entrée (ex : $A_0$=$A_1$=0 => $S_0$ =1).

1. Etablir la table de vérité.
2. Réaliser ce décodeur à l'aide d'un démultiplexeur.

Table de vérité

| **Entrées** | | **Sorties** | | | |
|---|---|---|---|---|---|
| A1 ($2^1$) | A0 ($2^0$) | $S_3$ | $S_2$ | $S_1$ | $S_0$ |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

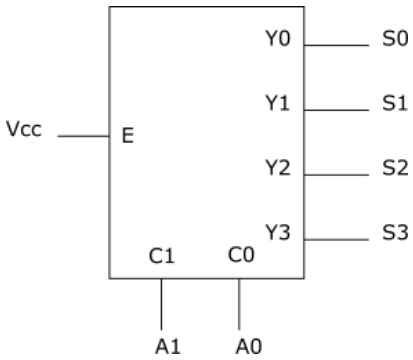



**Figure A - Vue externe du demux avec les branchements nécessaires à la réalisation du décodeur.**

Nous utilisons donc le demux en branchant Vcc (c'est à dire '1') sur son entrée, $A_0$ sur C0, et $A_1$ sur C1. Les sorties du demux, numérotées Y0 à Y3 dans la vue externe de la figure A, vont permettre de récupérer les sorties S0 à S3. Cet exercice illustre la section 2.4.2.1. mais avec une entrée active à 1 dans cet exercice.

Attention aux bits de poids fort et de poids faible ! Pour le codage binaire naturel, les indices les plus faibles correspondent toujours aux bits de poids les plus faibles et sont positionnés à droite. Exemple : $A_1A_0$

## *Section 3.3 : comparateur de mots binaires*

On souhaite réaliser un **comparateur** de deux mots de $n$ bits (figure 1) $A = (A_n \cdots A_i \cdots A_1)$ et $B = (B_n \cdots B_i \cdots B_1)$

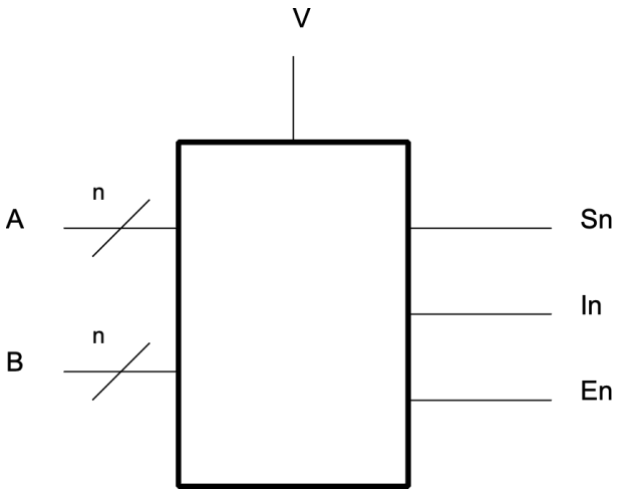



Figure. Symbole du comparateur

Fonctionnement du comparateur :

- Si l'entrée de validation ($V$) vaut 0, alors les trois sorties (Egal : $E_n$), (Supérieur : $S_n$) et (Inférieur : $I_n$) sont égales à 0.
- Si l'entrée de validation ($V$) vaut 1, alors :
  - $S_n = 1$ ssi $A > B$
  - $I_n = 1$ ssi $A < B$
  - $E_n = 1$ ssi $A = B$

## Comparateur élémentaire

Dans un premier temps, on souhaite réaliser un comparateur élémentaire de deux mots de 1 bit ( $n = 1$ ).

1. Etablir les équations des sorties $S_1$, $I_1$, et $E_1$ en fonction des entrées $A_1$, $B_1$, et $V$.
2. Dessiner le schéma de ce comparateur à partir d'opérateurs élémentaires (INV, NAND, AND, NOR, OR à 2, 3 ou 4 entrées).

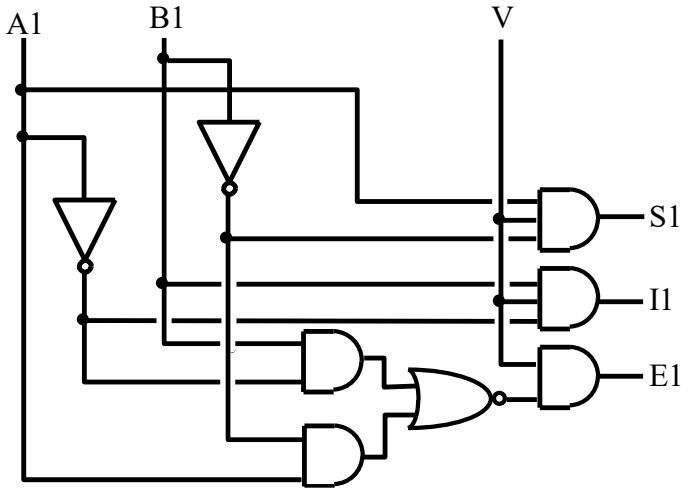
$S_1 = V.A_1.\overline{B_1}$

$I_1 = V.\overline{A_1}.B_1$

$E_1 = V.(A_1.B_1 + \overline{A_1}.\overline{B_1}) = V.(\overline{A_1 \oplus B_1})$

Un exemple de réalisation à partir d'une NOR pour $E_1$ :

$$E_1 = V.\overline{(A_1.\bar{B}_1 + \bar{A}_1 B_1)}$$




## Extension à 2 mots de 2 bits

On souhaite maintenant étendre l'amplitude du comparateur à deux mots de 2 bits.

1. Etablir les équations de $S_2$, $I_2$, et $E_2$ en fonction des bits $A_2$, $A_1$, $B_2$, $B_1$, et de l'entrée de validation $V$
2. Concevoir le schéma de ce comparateur en associant des comparateurs élémentaires et un minimum d'opérateurs (NAND, AND, NOR, OR à 2, 3 ou 4 entrées).

Comparaison de 2 mots de 2 bits : $A > B$ ssi [( $A_2 > B_2$ ) ou ( $A_2 = B_2$ et $A_1 > B_1$ )]

$S_2 = V(A_2.\overline{B_2} + A_2 B_2 A_1 \overline{B_1} + \overline{A_2}\,\overline{B_2} A_1 \overline{B_1}) = V(A_2.\overline{B_2} + A_1 \overline{B_1}(\overline{A_2 \oplus B_2}))$

$I_2 = V.(\overline{A_2}.B_2 + \overline{A_1} B_1(\overline{A_2 \oplus B_2}))$

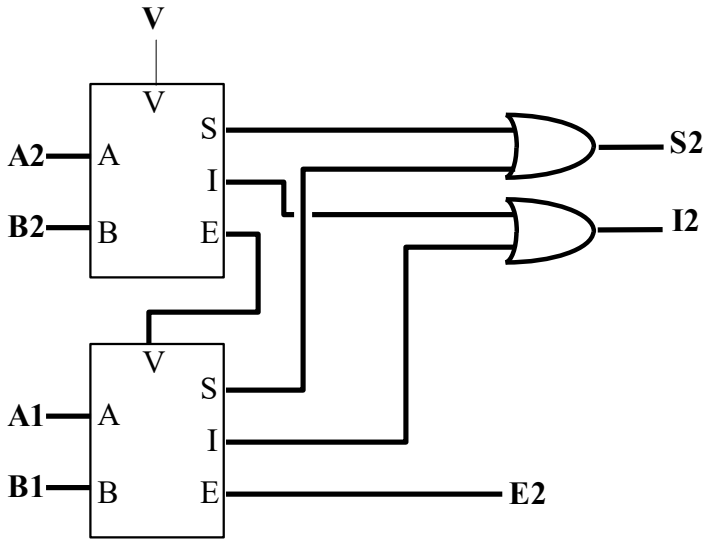
$E_2 = V.(\overline{A_2 \oplus B_2})(\overline{A_1 \oplus B_1})$

Remarque : on pourrait dresser la table de vérité et les tableaux de Karnaugh ; on obtiendrait alors :

$$S_2 = V(A_2.\overline{B_2} + A_1\overline{B_1}\,\overline{B_2} + A_1A_2\overline{B_1})$$
$$I_2 = V.(\overline{A_2}.B_2 + \overline{A_1}B_1B_2 + \overline{A_1}\,\overline{A_2}B_1)$$
$$E_2 = V.(\overline{A_2 \oplus B_2})(\overline{A_1 \oplus B_1})$$

Les 2 solutions sont correctes, mais la seconde ne profite pas au maximum de la possibilité d'utiliser les comparateurs 1 bit.

Réalisation :




## Généralisation (optionnel)

A partir des équations trouvées précédemment, établir les relations de récurrence ci-dessous :

$$S_n = \mathrm{f}(S_{n-1}, A_n, B_n, V)$$
$$I_n = \mathrm{g}(I_{n-1}, A_n, B_n, V)$$
$$E_n = \mathrm{h}(E_{n-1}, A_n, B_n, V)$$

On déduit de la question précédente :

$$S_n = V.A_n.\overline{B_n} + S_{n-1}(\overline{A_n \oplus B_n})$$
$$I_n = V.\overline{A_n}B_n.+I_{n-1}(\overline{A_n \oplus B_n})$$
$$E_n = E_{n-1}(\overline{A_n \oplus B_n})$$

Pour $E_n$, le signal de validation (V) est déjà présent dans $E_{n-1}$.

Idem pour $I_{n-1}$, et $S_{n-1}$.

## *Section 3.4 : Fonction logique et assurance*

1) Donner la table de vérité de la fonction logique f telle que f=1 si la police d'assurance n°15 est délivrée. On posera les variables d'entrée suivantes :

- P=1 si la police d'assurance n°10 est souscrite,
- S=1 si la personne est de sexe féminin,

- M=1 si la personne est mariée,
- A=1 si la personne est âgée de moins de 25 ans.

| P | S | M | A | f |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

| PS \ MA | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | 1 | 1 |
| 01 | | 1 | 1 | 1 |
| 11 | | 1 | 1 | 1 |
| 10 | | | 1 | 1 |

$$f = SA + M$$