

# Étude Booléenne

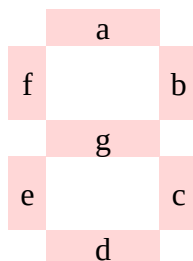
## Décodeur 7 segments

Voici les chiffres qu'on veut obtenir sur un afficheur 7 segments. Contrairement à ce qui est fait habituellement, nous utiliserons toutes les combinaisons des quatre bits entrants, soit seize chiffres pour décoder en hexadécimal. Le décodage décimal est inclus puisque les combinaisons de 10 à 15 ne sont pas utilisées si un nombre Décimal Codé Binaire arrive en entrée du décodeur.

### Visuel des chiffres et noms des segments

0 1 2 3 4 5 6 7 8 9 A b c d E F

La disposition des segments est :



## Table de vérité du décodeur

| Bits (indexés par poids) |   |   |   | Chiffre | Segments |   |   |   |   |   |   |
|--------------------------|---|---|---|---------|----------|---|---|---|---|---|---|
| 3                        | 2 | 1 | 0 |         | a        | b | c | d | e | f | g |
| 0                        | 0 | 0 | 0 | 0       | 1        | 1 | 1 | 1 | 1 | 1 | 0 |
| 0                        | 0 | 0 | 1 | 1       | 0        | 1 | 1 | 0 | 0 | 0 | 0 |
| 0                        | 0 | 1 | 0 | 2       | 1        | 1 | 0 | 1 | 1 | 0 | 1 |
| 0                        | 0 | 1 | 1 | 3       | 1        | 1 | 1 | 1 | 0 | 0 | 1 |
| 0                        | 1 | 0 | 0 | 4       | 0        | 1 | 1 | 0 | 0 | 1 | 1 |
| 0                        | 1 | 0 | 1 | 5       | 1        | 0 | 1 | 1 | 0 | 1 | 1 |
| 0                        | 1 | 1 | 0 | 6       | 1        | 0 | 1 | 1 | 1 | 1 | 1 |
| 0                        | 1 | 1 | 1 | 7       | 1        | 1 | 1 | 0 | 0 | 0 | 0 |
| 1                        | 0 | 0 | 0 | 8       | 1        | 1 | 1 | 1 | 1 | 1 | 1 |
| 1                        | 0 | 0 | 1 | 9       | 1        | 1 | 1 | 1 | 0 | 1 | 1 |
| 1                        | 0 | 1 | 0 | A       | 1        | 1 | 1 | 0 | 1 | 1 | 1 |
| 1                        | 0 | 1 | 1 | B       | 0        | 0 | 1 | 1 | 1 | 1 | 1 |
| 1                        | 1 | 0 | 0 | C       | 0        | 0 | 0 | 1 | 1 | 0 | 1 |
| 1                        | 1 | 0 | 1 | D       | 0        | 1 | 1 | 1 | 1 | 0 | 1 |
| 1                        | 1 | 1 | 0 | E       | 1        | 0 | 0 | 1 | 1 | 1 | 1 |
| 1                        | 1 | 1 | 1 | F       | 1        | 0 | 0 | 0 | 1 | 1 | 1 |

### Segment a

Ce segment s'allume pour ces combinaisons :

0000 0010 0011 0101 0110 0111 1000 1001 1010 1110 1111

Ce qui correspondant à la Somme De Produits booléenne suivante (0 complémentés deviennent 1) :

$$\text{SegA} = \overline{A_3}\overline{A_2}\overline{A_1}\overline{A_0} + \overline{A_3}\overline{A_2}A_1\overline{A_0} + \overline{A_3}\overline{A_2}A_1A_0 + \overline{A_3}A_2\overline{A_1}\overline{A_0} + \overline{A_3}A_2A_1\overline{A_0} + \overline{A_3}A_2A_1A_0 + A_3\overline{A_2}\overline{A_1}\overline{A_0} + A_3\overline{A_2}\overline{A_1}A_0 + A_3\overline{A_2}A_1\overline{A_0} + A_3\overline{A_2}A_1A_0$$

Il faut un total de 11x4 transistors<sup>1</sup> pour les portes AND et 11 transistors pour la porte OR soit 55 au total.

Les combinaisons manquantes pour éteindre sont moins nombreuses : 0001 0100 1011 1100 1101

et correspondent au Produit de Sommes booléen suivant et qui doit obtenir un zéro final (il faut donc chercher à annuler les bits, c'est-à-dire complémenter les 1) :

$$\text{SegA} = (A_3 + A_2 + A_1 + \overline{A_0})(A_3 + \overline{A_2} + A_1 + A_0)(\overline{A_3} + A_2 + \overline{A_1} + \overline{A_0})(\overline{A_3} + \overline{A_2} + A_1 + A_0)(\overline{A_3} + \overline{A_2} + A_1 + \overline{A_0})$$

Moins de portes logiques utilisées. Il y a 5x4 transistors pour les portes OR et 5 pour la porte AND. Soit un total de 25 transistors. On peut faire mieux avec un tableau de Karnaugh :

<sup>1</sup> Il faut 1 transistor par entrée de porte logique AND ou de porte logique OR. Il faut 1 transistor par porte NOT qui ne sont pas comptées ici car les inversions de bits se font une fois pour tous les décodeurs de segments.

|                               |    | A <sub>1</sub> A <sub>0</sub> |    |    |    |
|-------------------------------|----|-------------------------------|----|----|----|
| SDP segA                      |    | 00                            | 01 | 11 | 10 |
| A <sub>3</sub> A <sub>2</sub> | 00 | 1                             |    | 1  | 1  |
|                               | 01 |                               | 1  | 1  | 1  |
|                               | 11 |                               |    | 1  | 1  |
|                               | 10 | 1                             | 1  |    | 1  |

Le tableau est cylindrique suivant les deux axes. On groupe les cases adjacentes pour faire les plus gros groupes possibles tout en étant de taille en puissance de deux, avec recouvrement autorisé.

D'après les groupements dans le tableau pour lesquels on ne conserve que les variables qui sont constantes à l'intérieur du groupe :

$$\text{SegA} = \overline{A_2}\overline{A_0} + \overline{A_3}A_1 + A_2A_1 + \overline{A_3}A_2A_0 + A_3\overline{A_2}\overline{A_1}$$

Ce qui fait 2+2+2+3+3=12 transistors pour les portes AND et 5 pour la porte OR. Soit 17 transistors. On essaie aussi le produit de sommes, pour lequel on cherche à obtenir un zéro final.

|                               |    | A <sub>1</sub> A <sub>0</sub> |    |    |    |
|-------------------------------|----|-------------------------------|----|----|----|
| PDS segA                      |    | 00                            | 01 | 11 | 10 |
| A <sub>3</sub> A <sub>2</sub> | 00 |                               | 0  |    |    |
|                               | 01 | 0                             |    |    |    |
|                               | 11 | 0                             | 0  |    |    |
|                               | 10 |                               |    |    | 0  |

D'après les groupements qui sont ici les plus gros possibles, le produit de sommes pour lequel on veut annuler le résultat (donc il faut complémenter les 1 pour en faire des 0), est :

$$\text{SegA} = (\overline{A_2+A_1+A_0})(\overline{A_3+A_2+A_1})(A_3+A_2+A_1+\overline{A_0})(\overline{A_3+A_2+A_1+\overline{A_0}})$$

Ce qui fait 3+3+4+4=14 transistors pour les portes OR et 4 pour la porte AND, soit 18 transistors. Au final, à un transistor près, la SDP minimisée est la plus économe en transistors.

## Segment b

Ce segment s'allume pour ces combinaisons :

0000 0001 0010 0011 0100 0111 1000 1001 1010 1101

Ce qui correspond à la Somme De Produits booléenne suivante :

$$\text{SegB} = \overline{A_3}\overline{A_2}\overline{A_1}\overline{A_0} + \overline{A_3}\overline{A_2}\overline{A_1}A_0 + \overline{A_3}\overline{A_2}A_1\overline{A_0} + \overline{A_3}\overline{A_2}A_1A_0 + \overline{A_3}A_2\overline{A_1}\overline{A_0} + \overline{A_3}A_2A_1\overline{A_0} + A_3\overline{A_2}\overline{A_1}\overline{A_0} + A_3\overline{A_2}\overline{A_1}A_0 + A_3\overline{A_2}A_1\overline{A_0} + A_3\overline{A_2}A_1A_0$$

Il faut un total de 10x4 transistors pour les portes AND et 10 transistors pour la porte OR soit 50 transistors au total.

Les combinaisons manquantes sont moins nombreuses : 0101 0110 1011 1100 1110 1111

et correspondent au Produit de Sommes booléen suivant et qui doit obtenir un zéro final (il faut donc chercher à annuler les bits, c'est-à-dire complémenter les 1) :

$$\text{SegB} = (A_3 + \bar{A}_2 + A_1 + \bar{A}_0)(A_3 + \bar{A}_2 + \bar{A}_1 + A_0)(\bar{A}_3 + A_2 + \bar{A}_1 + \bar{A}_0)(\bar{A}_3 + \bar{A}_2 + A_1 + A_0)(\bar{A}_3 + \bar{A}_2 + \bar{A}_1 + A_0)(\bar{A}_3 + \bar{A}_2 + \bar{A}_1 + \bar{A}_0)$$

Moins de portes logiques utilisées. Il y a 6x4 transistors pour les portes OR et 6 pour la porte AND. Soit un total de 30 transistors. On peut faire mieux avec un tableau de Karnaugh :

|                               |    | A <sub>1</sub> A <sub>0</sub> |    |    |    |
|-------------------------------|----|-------------------------------|----|----|----|
|                               |    | 00                            | 01 | 11 | 10 |
| A <sub>3</sub> A <sub>2</sub> | 00 | 1                             | 1  | 1  | 1  |
|                               | 01 | 1                             |    | 1  |    |
|                               | 11 |                               | 1  |    |    |
|                               | 10 | 1                             | 1  |    | 1  |

Le tableau est cylindrique suivant les deux axes. On groupe les cases adjacentes pour faire les plus gros groupes possibles tout en étant de taille en puissance de deux, avec recouvrement autorisé.

D'après les groupements dans le tableau pour lesquels on ne conserve que les variables qui sont constantes à l'intérieur du groupe :

$$\text{SegB} = \bar{A}_2\bar{A}_0 + \bar{A}_3\bar{A}_1 + A_3A_1A_0 + \bar{A}_3A_2\bar{A}_0 + \bar{A}_3A_2A_1$$

Ce qui fait 2+2+3+3+3=13 transistors pour les portes AND et 5 pour la porte OR. Soit 18 transistors. On essaie aussi le produit de sommes, pour lequel on cherche à obtenir un zéro final.

|                               |    | A <sub>1</sub> A <sub>0</sub> |    |    |    |
|-------------------------------|----|-------------------------------|----|----|----|
|                               |    | 00                            | 01 | 11 | 10 |
| A <sub>3</sub> A <sub>2</sub> | 00 |                               |    |    |    |
|                               | 01 |                               | 0  |    | 0  |
|                               | 11 | 0                             |    | 0  | 0  |
|                               | 10 |                               |    | 0  |    |

D'après les groupements qui sont ici les plus gros possibles, le produit de sommes pour lequel on veut annuler le résultat (donc il faut complémentar les 1 pour en faire des 0), est :

$$\text{SegB} = (\bar{A}_3 + \bar{A}_2 + A_0)(\bar{A}_3 + \bar{A}_1 + A_0)(\bar{A}_2 + \bar{A}_1 + A_0)(A_3 + \bar{A}_2 + A_1 + \bar{A}_0)$$

Ce qui fait 3+3+3+4=13 transistors pour les portes OR et 4 pour la porte AND, soit 17 transistors. Au final, à un transistor près le PDS minimisé est le plus économe en nombre de transistors.

## Segment c

Ce segment s'allume pour ces combinaisons :

0000 0001 0011 0100 0101 0110 0111 1000 1001 1010 1011 1101

Ce qui correspond à la Somme De Produits booléenne suivante :

$$\text{SegC} = \bar{A}_3\bar{A}_2\bar{A}_1\bar{A}_0 + \bar{A}_3\bar{A}_2\bar{A}_1A_0 + \bar{A}_3\bar{A}_2A_1\bar{A}_0 + \bar{A}_3\bar{A}_2A_1A_0 + \bar{A}_3A_2\bar{A}_1\bar{A}_0 + \bar{A}_3A_2\bar{A}_1A_0 + \bar{A}_3A_2A_1\bar{A}_0 + \bar{A}_3A_2A_1A_0 + A_3\bar{A}_2\bar{A}_1\bar{A}_0 + A_3\bar{A}_2\bar{A}_1A_0 + A_3\bar{A}_2A_1\bar{A}_0 + A_3\bar{A}_2A_1A_0 + A_3A_2\bar{A}_1\bar{A}_0$$

Il faut un total de 12x4 transistors pour les portes AND et 12 transistors pour la porte OR soit 60 transistors au total.

Les combinaisons pour éteindre le segment sont moins nombreuses : 0010 1100 1110 1111

et correspondent au Produit de Sommes booléen suivant et qui doit obtenir un zéro final (il faut donc chercher à annuler les bits, c'est-à-dire complémenter les 1) :

$$\text{SegC} = (A_3 + A_2 + \bar{A}_1 + A_0)(\bar{A}_3 + \bar{A}_2 + A_1 + A_0)(\bar{A}_3 + \bar{A}_2 + \bar{A}_1 + A_0)(\bar{A}_3 + \bar{A}_2 + \bar{A}_1 + \bar{A}_0)$$

Moins de portes logiques utilisées. Il y a 4x4 transistors pour les portes OR et 4 pour la porte AND. Soit un total de 20 transistors. On peut faire mieux avec un tableau de Karnaugh :

|                               |    | A <sub>1</sub> A <sub>0</sub> |    |    |    |
|-------------------------------|----|-------------------------------|----|----|----|
|                               |    | 00                            | 01 | 11 | 10 |
| A <sub>3</sub> A <sub>2</sub> | 00 | 1                             | 1  | 1  |    |
|                               | 01 | 1                             | 1  | 1  | 1  |
|                               | 11 |                               | 1  |    |    |
|                               | 10 | 1                             | 1  | 1  | 1  |

Le tableau est cylindrique suivant les deux axes. On groupe les cases adjacentes pour faire les plus gros groupes possibles tout en étant de taille en puissance de deux, avec recouvrement autorisé.

D'après les groupements dans le tableau pour lesquels on ne conserve que les variables qui sont constantes à l'intérieur du groupe :

$$\text{SegC} = \bar{A}_3 A_2 + \bar{A}_1 A_0 + A_3 A_2 + \bar{A}_3 A_0 + \bar{A}_3 A_1$$

Ce qui fait 2+2+2+2=8 transistors pour les portes AND et 5 pour la porte OR. Soit 13 transistors. On essaie aussi le produit de sommes, pour lequel on cherche à obtenir un zéro final.

|                               |    | A <sub>1</sub> A <sub>0</sub> |    |    |    |
|-------------------------------|----|-------------------------------|----|----|----|
|                               |    | 00                            | 01 | 11 | 10 |
| A <sub>3</sub> A <sub>2</sub> | 00 |                               |    |    | 0  |
|                               | 01 |                               |    |    |    |
|                               | 11 | 0                             |    | 0  | 0  |
|                               | 10 |                               |    |    |    |

D'après les groupements qui sont ici les plus gros possibles, le produit de sommes pour lequel on veut annuler le résultat (donc il faut complémenter les 1 pour en faire des 0), est :

$$\text{SegC} = (\bar{A}_3 + \bar{A}_2 + A_1 + A_0)(\bar{A}_3 + A_2 + A_1)(A_3 + A_2 + \bar{A}_1 + A_0)$$

Ce qui fait 4+3+4=11 transistors pour les portes OR et 3 pour la porte AND, soit 14 transistors. Au final, à un transistor près, la SDP minimisée est la plus économe en nombre de transistors.

## Segment d

Ce segment s'allume pour ces combinaisons :

0000 0010 0011 0101 0110 1000 1001 1011 1100 1101 1110

Ce qui correspondant à la Somme De Produits booléenne suivante :

$$\text{SegD} = \overline{A_3}\overline{A_2}\overline{A_1}\overline{A_0} + \overline{A_3}\overline{A_2}A_1\overline{A_0} + \overline{A_3}\overline{A_2}A_1A_0 + \overline{A_3}A_2\overline{A_1}A_0 + \overline{A_3}A_2A_1\overline{A_0} + \overline{A_3}A_2\overline{A_1}\overline{A_0} + \overline{A_3}A_2\overline{A_1}A_0 + A_3\overline{A_2}A_1A_0 + A_3A_2\overline{A_1}\overline{A_0} + A_3A_2\overline{A_1}A_0 + A_3A_2A_1\overline{A_0}$$

Il faut un total de 11x4 transistors pour les portes AND et 11 transistors pour la porte OR soit 55 transistors au total.

Les combinaisons pour éteindre le segment sont moins nombreuses : 0001 0100 0111 1010 1111 et correspondent au Produit de Sommes booléen suivant et qui doit obtenir un zéro final (il faut donc chercher à annuler les bits, c'est-à-dire complémentérer les 1) :

$$\text{SegD} = (A_3+A_2+A_1+\overline{A_0})(A_3+\overline{A_2}+A_1+A_0)(A_3+\overline{A_2}+\overline{A_1}+\overline{A_0})(\overline{A_3}+A_2+\overline{A_1}+A_0)(\overline{A_3}+\overline{A_2}+\overline{A_1}+\overline{A_0})$$

Moins de portes logiques utilisées. Il y a 4x4 transistors pour les portes OR et 4 pour la porte AND. Soit un total de 20 transistors. On peut faire mieux avec un tableau de Karnaugh :

|                               |    |                               |    |    |    |
|-------------------------------|----|-------------------------------|----|----|----|
|                               |    | A <sub>1</sub> A <sub>0</sub> |    |    |    |
|                               |    | 00                            | 01 | 11 | 10 |
| A <sub>3</sub> A <sub>2</sub> | 00 | 1                             |    | 1  | 1  |
|                               | 01 |                               | 1  |    | 1  |
|                               | 11 | 1                             | 1  |    | 1  |
|                               | 10 | 1                             | 1  | 1  |    |

Le tableau est cylindrique suivant les deux axes. On groupe les cases adjacentes pour faire les plus gros groupes possibles tout en étant de taille en puissance de deux, avec recouvrement autorisé.

D'après les groupements dans le tableau pour lesquels on ne conserve que les variables qui sont constantes à l'intérieur du groupe :

$$\text{SegD} = \overline{A_2}\overline{A_1}\overline{A_0} + \overline{A_3}\overline{A_2}A_1 + \overline{A_2}\overline{A_1}A_0 + \overline{A_2}A_1\overline{A_0} + \overline{A_3}\overline{A_2}A_0 + \overline{A_3}A_1$$

Ce qui fait 3+3+3+3+3+2=17 transistors pour les portes AND et 6 pour la porte OR. Soit 23 transistors. On essaie aussi le produit de sommes, pour lequel on cherche à obtenir un zéro final.

|                               |    |                               |    |    |    |
|-------------------------------|----|-------------------------------|----|----|----|
|                               |    | A <sub>2</sub> A <sub>3</sub> |    |    |    |
|                               |    | 00                            | 01 | 11 | 10 |
| A <sub>0</sub> A <sub>1</sub> | 00 |                               | 0  |    |    |
|                               | 01 | 0                             |    | 0  |    |
|                               | 11 |                               |    | 0  |    |
|                               | 10 |                               |    |    | 0  |

D'après les groupements qui sont ici les plus gros possibles, le produit de sommes pour lequel on veut annuler le résultat (donc il faut complémentérer les 1 pour en faire des 0), est :

$$\text{SegD} = (A_3+A_2+A_1+\overline{A_0})(A_3+\overline{A_2}+A_1+A_0)(\overline{A_2}+\overline{A_1}+\overline{A_0})(\overline{A_3}+A_2+\overline{A_1}+A_0)$$

Ce qui fait 4+4+3+4=15 transistors pour les portes OR et 4 pour la porte AND, soit 19 transistors. Finalement, le PDS minimisé est le plus économe en nombre de transistors.

## Segment e

Ce segment s'allume pour ces combinaisons :

0000 0010 0110 1000 1010 1011 1100 1101 1110 1111

Ce qui correspondant à la Somme De Produits booléenne suivante :

$$\text{SegE} = \bar{A}_3\bar{A}_2\bar{A}_1\bar{A}_0 + \bar{A}_3\bar{A}_2A_1\bar{A}_0 + \bar{A}_3A_2A_1\bar{A}_0 + A_3\bar{A}_2\bar{A}_1\bar{A}_0 + A_3\bar{A}_2A_1\bar{A}_0 + A_3\bar{A}_2A_1A_0 + A_3A_2\bar{A}_1\bar{A}_0 + A_3A_2\bar{A}_1A_0 + A_3A_2A_1\bar{A}_0 + A_3A_2A_1A_0$$

Il faut un total de 10x4 transistors pour les portes AND et 10 transistors pour la porte OR soit 50 transistors au total.

Les combinaisons pour éteindre sont moins nombreuses : 0001 0011 0100 0101 0111 1001

et correspondent au Produit de Sommes booléen suivant et qui doit obtenir un zéro final (il faut donc chercher à annuler les bits, c'est-à-dire complémenter les 1) :

$$\text{SegE} = (A_3+A_2+A_1+\bar{A}_0)(A_3+A_2+\bar{A}_1+\bar{A}_0)(A_3+\bar{A}_2+A_1+A_0)(A_3+\bar{A}_2+A_1+\bar{A}_0)(A_3+\bar{A}_2+\bar{A}_1+\bar{A}_0)(\bar{A}_3+A_2+A_1+\bar{A}_0)$$

Moins de portes logiques utilisées. Il y a 6x4 transistors pour les portes OR et 6 pour la porte AND. Soit un total de 30 transistors. On peut faire mieux avec un tableau de Karnaugh :

| SDP segE                      |    | A <sub>1</sub> A <sub>0</sub> |    |    |    |
|-------------------------------|----|-------------------------------|----|----|----|
|                               |    | 00                            | 01 | 11 | 10 |
| A <sub>3</sub> A <sub>2</sub> | 00 | 1                             |    |    | 1  |
|                               | 01 |                               |    |    | 1  |
|                               | 11 | 1                             | 1  | 1  | 1  |
|                               | 10 | 1                             |    | 1  | 1  |

Le tableau est cylindrique suivant les deux axes. On groupe les cases adjacentes pour faire les plus gros groupes possibles tout en étant de taille en puissance de deux, avec recouvrement autorisé.

D'après les groupements dans le tableau pour lesquels on ne conserve que les variables qui sont constantes à l'intérieur du groupe :

$$\text{SegE} = \bar{A}_2\bar{A}_0 + A_1\bar{A}_0 + A_3A_1 + A_3A_2$$

Ce qui fait 2+2+2+2=8 transistors pour les portes AND et 4 pour la porte OR. Soit 10 transistors. On essaie aussi le produit de sommes, pour lequel on cherche à obtenir un zéro final.

| PDS segE                      |    | A <sub>1</sub> A <sub>0</sub> |    |    |    |
|-------------------------------|----|-------------------------------|----|----|----|
|                               |    | 00                            | 01 | 11 | 10 |
| A <sub>3</sub> A <sub>2</sub> | 00 |                               | 0  | 0  |    |
|                               | 01 | 0                             | 0  | 0  |    |
|                               | 11 |                               |    |    |    |
|                               | 10 |                               | 0  |    |    |

D'après les groupements qui sont ici les plus gros possibles, le produit de sommes pour lequel on veut annuler le résultat (donc il faut complémenter les 1 pour en faire des 0), est :

$$\text{SegE} = (A_3 + \bar{A}_0)(A_3 + \bar{A}_2 + A_1)(\bar{A}_3 + A_2 + A_1 + \bar{A}_0)$$

Ce qui fait 2+3+4=9 transistors pour les portes OR et 3 pour la porte AND, soit 12 transistors. Finalement, la SDP minimisée est la plus économique en nombre de transistors.

## Segment f

Ce segment s'allume pour ces combinaisons :

0000 0100 0101 0110 1000 1001 1010 1011 1110 1111

Ce qui correspondant à la Somme De Produits booléenne suivante :

$$\text{SegF} = \bar{A}_3\bar{A}_2\bar{A}_1\bar{A}_0 + \bar{A}_3A_2\bar{A}_1\bar{A}_0 + \bar{A}_3A_2\bar{A}_1A_0 + \bar{A}_3A_2A_1\bar{A}_0 + A_3\bar{A}_2\bar{A}_1\bar{A}_0 + A_3\bar{A}_2\bar{A}_1A_0 + A_3\bar{A}_2A_1\bar{A}_0 + A_3\bar{A}_2A_1A_0 + A_3A_2\bar{A}_1\bar{A}_0 + A_3A_2\bar{A}_1A_0 + A_3A_2A_1\bar{A}_0$$

Il faut un total de 10x4 transistors pour les portes AND et 10 transistors pour la porte OR soit 50 transistors au total.

Les combinaisons pour éteindre sont moins nombreuses : 0001 0010 0011 0111 1100 1101

et correspondent au Produit de Sommes booléen suivant et qui doit obtenir un zéro final (il faut donc chercher à annuler les bits, c'est-à-dire complémentérer les 1) :

$$\text{SegF} = (A_3 + A_2 + A_1 + \bar{A}_0)(A_3 + A_2 + \bar{A}_1 + A_0)(A_3 + A_2 + \bar{A}_1 + \bar{A}_0)(A_3 + \bar{A}_2 + \bar{A}_1 + \bar{A}_0)(\bar{A}_3 + \bar{A}_2 + A_1 + A_0)(\bar{A}_3 + \bar{A}_2 + A_1 + \bar{A}_0)$$

Moins de portes logiques utilisées. Il y a 6x4 transistors pour les portes OR et 6 pour la porte AND. Soit un total de 30 transistors. On peut faire mieux avec un tableau de Karnaugh :

|                               |    | A <sub>1</sub> A <sub>0</sub> |    |    |    |
|-------------------------------|----|-------------------------------|----|----|----|
|                               |    | 00                            | 01 | 11 | 10 |
| A <sub>3</sub> A <sub>2</sub> | 00 | 1                             |    |    |    |
|                               | 01 | 1                             | 1  |    | 1  |
|                               | 11 |                               |    | 1  | 1  |
|                               | 10 | 1                             | 1  | 1  | 1  |

Le tableau est cylindrique suivant les deux axes. On groupe les cases adjacentes pour faire les plus gros groupes possibles tout en étant de taille en puissance de deux, avec recouvrement autorisé.

D'après les groupements dans le tableau pour lesquels on ne conserve que les variables qui sont constantes à l'intérieur du groupe :

$$\text{SegF} = \bar{A}_3\bar{A}_1\bar{A}_0 + \bar{A}_3A_2\bar{A}_1 + A_2A_1\bar{A}_0 + A_3A_2 + A_3A_1$$

Ce qui fait 3+3+3+2+2=13 transistors pour les portes AND et 5 pour la porte OR. Soit 18 transistors. On essaie aussi le produit de sommes, pour lequel on cherche à obtenir un zéro final.



|                               |    |                               |    |    |    |
|-------------------------------|----|-------------------------------|----|----|----|
|                               |    | A <sub>1</sub> A <sub>0</sub> |    |    |    |
| PDS segF                      |    | 00                            | 01 | 11 | 10 |
|                               | 00 | 0                             | 0  | 0  | 0  |
| A <sub>3</sub> A <sub>2</sub> | 01 |                               |    | 0  |    |
|                               | 11 | 0                             | 0  |    |    |
|                               | 10 |                               |    |    |    |

D'après les groupements qui sont ici les plus gros possibles, le produit de sommes pour lequel on veut annuler le résultat (donc il faut complémentérer les 1 pour en faire des 0), est :

$$\text{SegF} = (A_3 + A_2 + \bar{A}_0)(A_3 + A_2 + \bar{A}_1)(A_3 + \bar{A}_1 + \bar{A}_0)(\bar{A}_3 + \bar{A}_2 + A_1)$$

Ce qui fait 3+3+3+3=12 transistors pour les portes OR et 4 pour la porte AND, soit 16 transistors. Finalement, le PDS minimisé est le plus économe en nombre de transistors.

## Segment g

Ce segment s'allume pour ces combinaisons :

0010 0011 0100 0101 0110 1000 1001 1010 1011 1100 1101 1110 1111

Ce qui correspond à la Somme De Produits booléenne suivante :

$$\text{SegG} = \bar{A}_3\bar{A}_2A_1\bar{A}_0 + \bar{A}_3\bar{A}_2A_1A_0 + \bar{A}_3A_2\bar{A}_1\bar{A}_0 + \bar{A}_3A_2\bar{A}_1A_0 + \bar{A}_3A_2A_1\bar{A}_0 + \bar{A}_3A_2A_1A_0 + A_3\bar{A}_2\bar{A}_1\bar{A}_0 + A_3\bar{A}_2\bar{A}_1A_0 + A_3\bar{A}_2A_1\bar{A}_0 + A_3\bar{A}_2A_1A_0 + A_3A_2\bar{A}_1\bar{A}_0 + A_3A_2\bar{A}_1A_0 + A_3A_2A_1\bar{A}_0 + A_3A_2A_1A_0$$

Il faut un total de 13x4 transistors pour les portes AND et 13 transistors pour la porte OR soit 65 transistors au total.

Les combinaisons pour éteindre sont bien moins nombreuses : 0000 0001 0111

et correspondent au Produit de Sommes booléen suivant et qui doit obtenir un zéro final (il faut donc chercher à annuler les bits, c'est-à-dire complémentérer les 1) :

$$\text{SegG} = (A_3 + A_2 + A_1 + A_0)(A_3 + A_2 + A_1 + \bar{A}_0)(A_3 + \bar{A}_2 + \bar{A}_1 + \bar{A}_0)$$

Beaucoup moins de portes logiques utilisées. Il y a 3x4 transistors pour les portes OR et 3 pour la porte AND. Soit un total de 15 transistors. On peut faire mieux avec un tableau de Karnaugh :

|                               |    |                               |    |    |    |
|-------------------------------|----|-------------------------------|----|----|----|
|                               |    | A <sub>1</sub> A <sub>0</sub> |    |    |    |
| SDP segG                      |    | 00                            | 01 | 11 | 10 |
|                               | 00 |                               |    | 1  | 1  |
| A <sub>3</sub> A <sub>2</sub> | 01 | 1                             | 1  |    | 1  |
|                               | 11 | 1                             | 1  | 1  | 1  |
|                               | 10 | 1                             | 1  | 1  | 1  |

Le tableau est cylindrique suivant les deux axes. On groupe les cases adjacentes pour faire les plus gros groupes possibles tout en étant de taille en puissance de deux, avec recouvrement autorisé.

D'après les groupements dans le tableau pour lesquels on ne conserve que les variables qui sont constantes à l'intérieur du groupe :

$$\text{SegG} = \overline{A_2}A_1 + A_2\overline{A_1} + A_1\overline{A_0} + A_3$$

Ce qui fait 2+2+2+1=7 transistors pour les portes AND et 4 pour la porte OR. Soit 11 transistors. On essaie aussi le produit de sommes, pour lequel on cherche à obtenir un zéro final.

|          |    |          |    |    |    |
|----------|----|----------|----|----|----|
|          |    | $A_1A_0$ |    |    |    |
| PDS segG |    | 00       | 01 | 11 | 10 |
|          | 00 | 0        | 0  |    |    |
| $A_3A_2$ | 01 |          |    | 0  |    |
|          | 11 |          |    |    |    |
|          | 10 |          |    |    |    |

D'après les groupements qui sont ici les plus gros possibles, le produit de sommes pour lequel on veut annuler le résultat (donc il faut complémentérer les 1 pour en faire des 0), est :

$$\text{SegG} = (A_3 + A_2 + A_1)(A_3 + \overline{A_2} + \overline{A_1} + \overline{A_0})$$

Ce qui fait 3+4=7 transistors pour les portes OR et 2 pour la porte AND, soit 9 transistors. Finalement, le PDS minimisé est le plus économe en nombre de transistors.

## Expressions booléennes minimisées du décodeur 7 segments

Après minimisation des sommes de produits et produits de sommes, les plus économes en transistors sont ces expressions-ci :

$$\text{SegA} = \overline{A_2}\overline{A_0} + \overline{A_3}A_1 + A_2A_1 + \overline{A_3}A_2A_0 + A_3\overline{A_2}\overline{A_1}$$

$$\text{SegB} = (\overline{A_3} + \overline{A_2} + A_0)(\overline{A_3} + \overline{A_1} + \overline{A_0})(\overline{A_2} + \overline{A_1} + A_0)(A_3 + \overline{A_2} + A_1 + \overline{A_0})$$

$$\text{SegC} = \overline{A_3}A_2 + \overline{A_1}A_0 + A_3\overline{A_2} + \overline{A_3}A_0 + \overline{A_3}\overline{A_1}$$

$$\text{SegD} = (A_3 + A_2 + A_1 + \overline{A_0})(A_3 + \overline{A_2} + A_1 + A_0)(\overline{A_2} + \overline{A_1} + \overline{A_0})(\overline{A_3} + A_2 + \overline{A_1} + A_0)$$

$$\text{SegE} = \overline{A_2}\overline{A_0} + A_1\overline{A_0} + A_3A_1 + A_3A_2$$

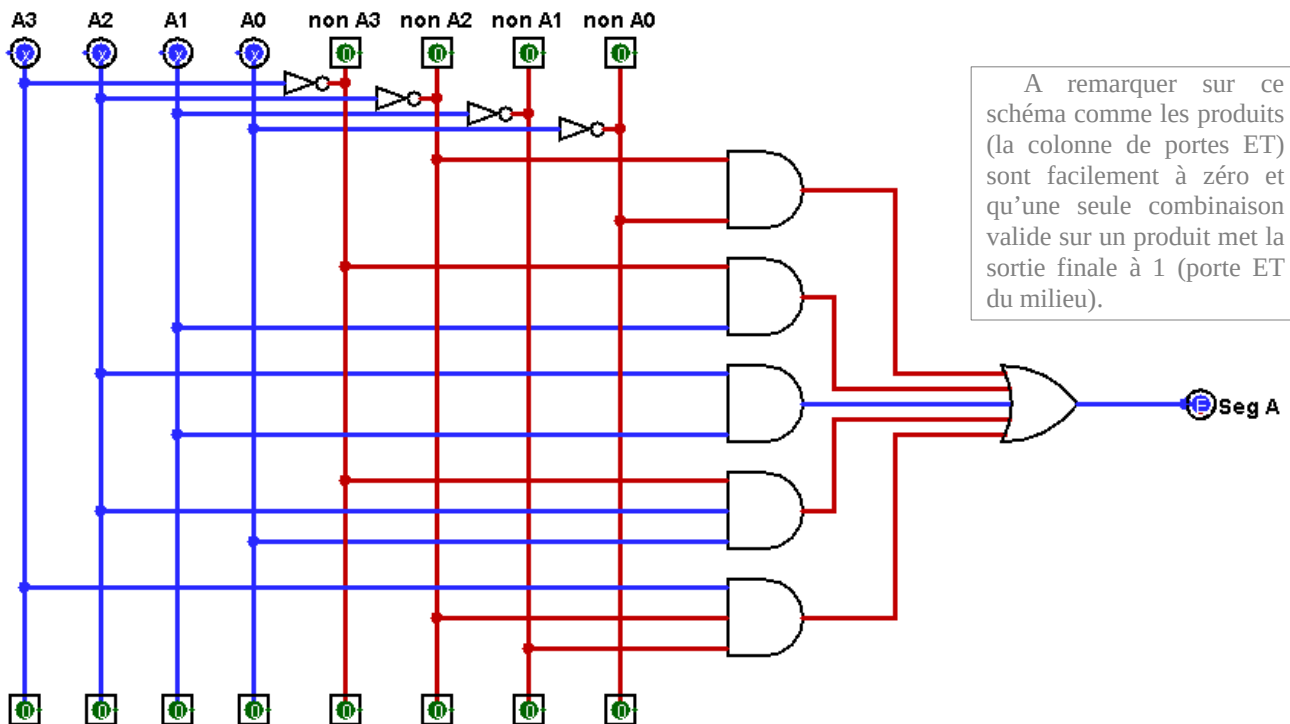
$$\text{SegF} = (A_3 + A_2 + \overline{A_0})(A_3 + A_2 + \overline{A_1})(A_3 + \overline{A_1} + \overline{A_0})(\overline{A_3} + \overline{A_2} + A_1)$$

$$\text{SegG} = (A_3 + A_2 + A_1)(A_3 + \overline{A_2} + \overline{A_1} + \overline{A_0})$$

Il est temps maintenant de faire les routages de tous des décodeurs, c'est-à-dire de faire les schémas des câblages entre les signaux d'entrée  $A_3, A_2, A_1, A_0$  et de leurs compléments, aussi en entrées  $\overline{A_3}, \overline{A_2}, \overline{A_1}$  et  $\overline{A_0}$  et les portes ET des sommes de produits ou les portes OU des produits de sommes. Les SDP sont constitués d'une ligne de portes ET chacune étant connectée à ses signaux d'entrée spécifiques pour effectuer les produits qui sont ensuite acheminés vers une unique porte OU pour en faire la somme. Inversement, les PDS sont constitués d'une ligne de portes OU chacune étant connectée à ses signaux d'entrée spécifiques pour effectuer les sommes qui sont ensuite acheminés vers une unique porte ET pour en faire le produit. Les routages qui sont montrés ci-après sont à reproduire dans le logiciel Digital Logic Sim pour pouvoir en faire une simulation.

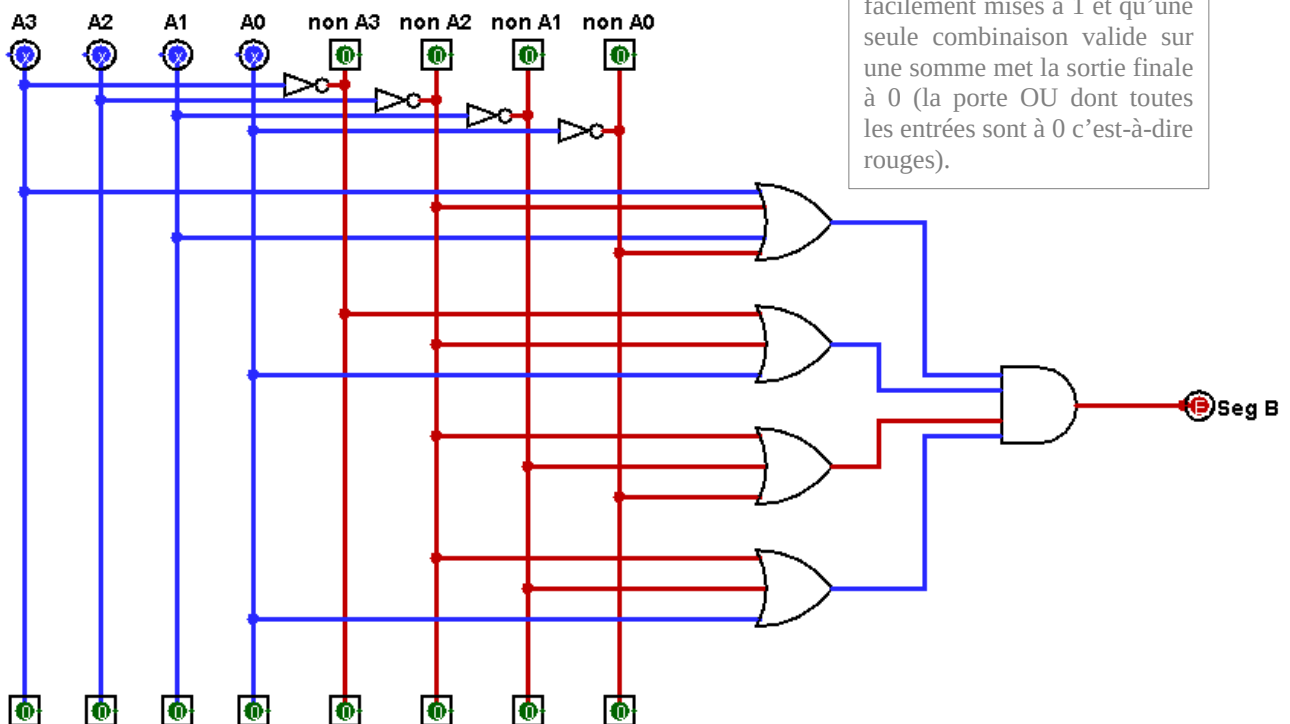
## Routage du décodeur du segment a

La somme de produits SegA donne ce schéma :



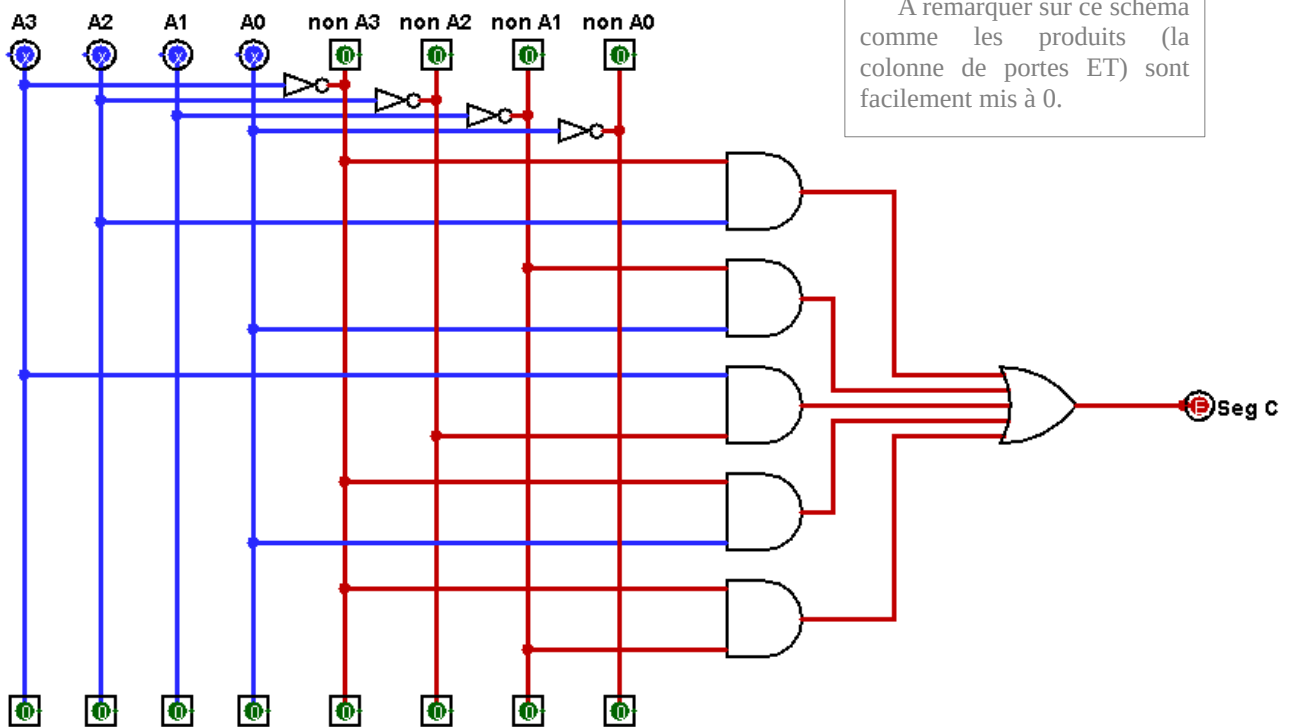
## Routage du décodeur du segment b

La somme de produits SegB donne ce schéma :



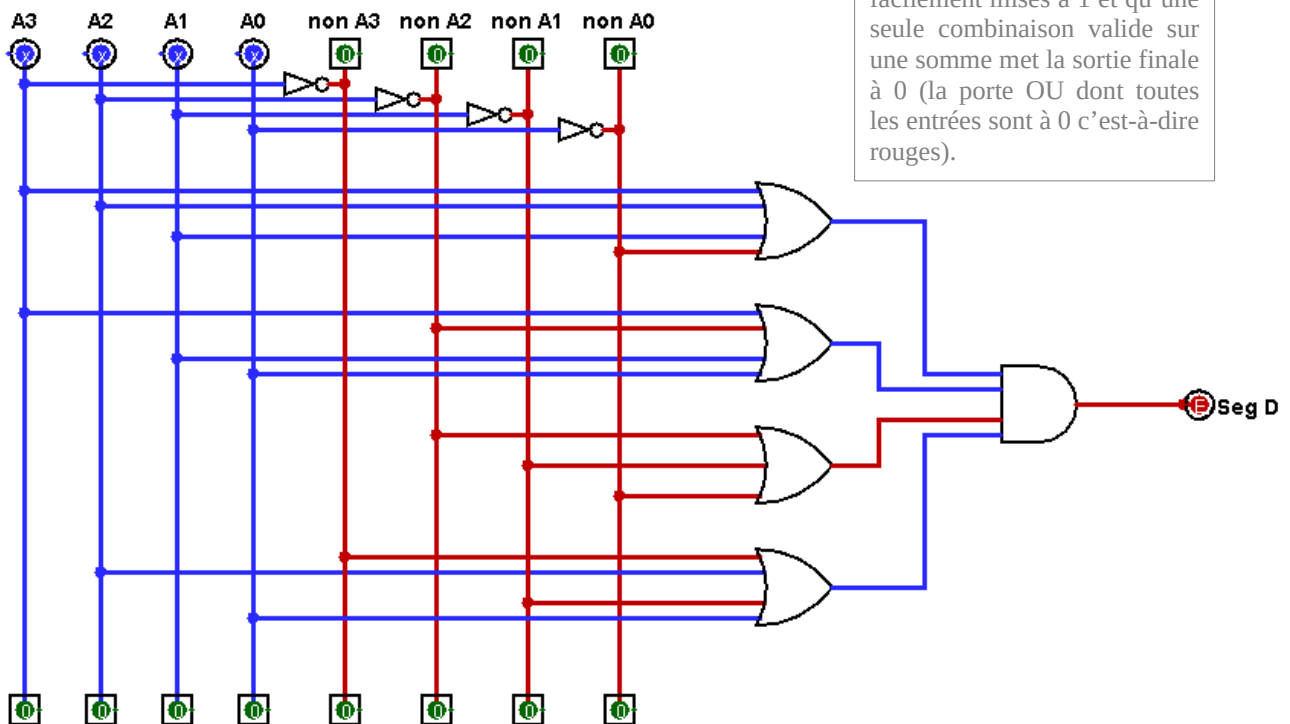
## Routage du décodeur du segment c

La somme de produits SegC donne ce schéma :



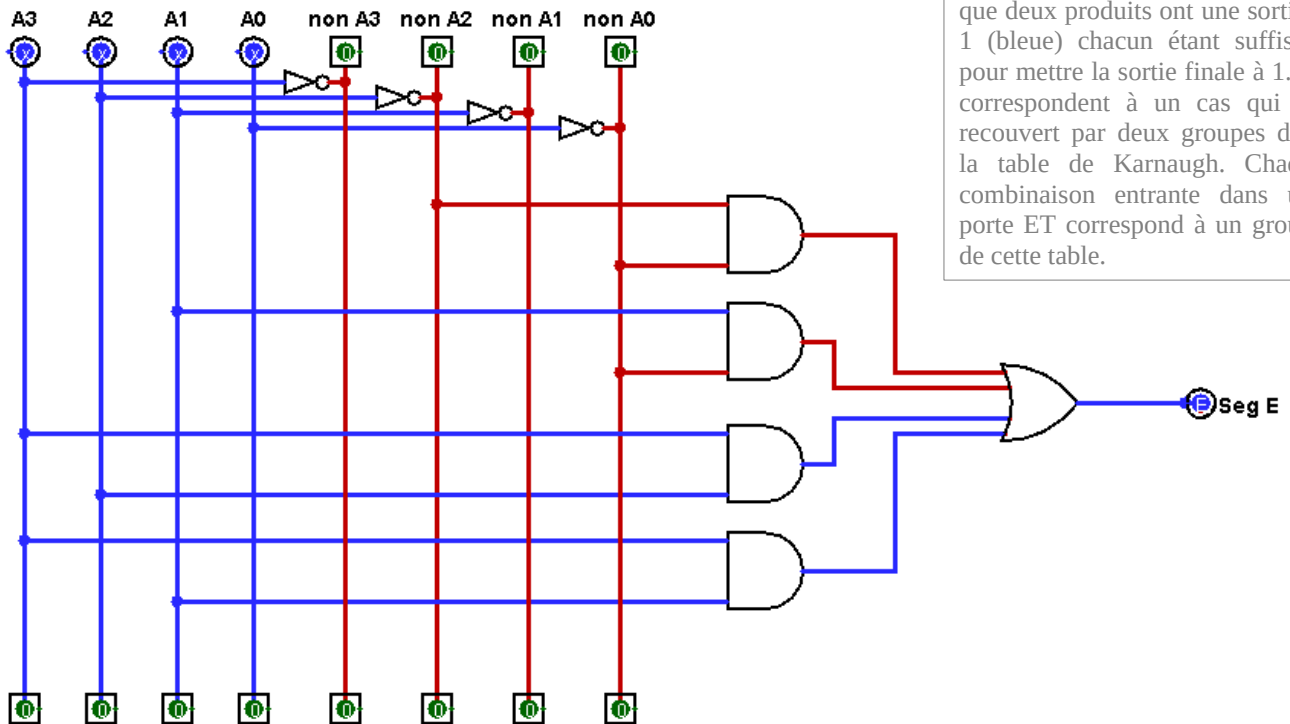
## Routage du décodeur du segment d

La produit de sommes SegD donne ce schéma :



## Routage du décodeur du segment e

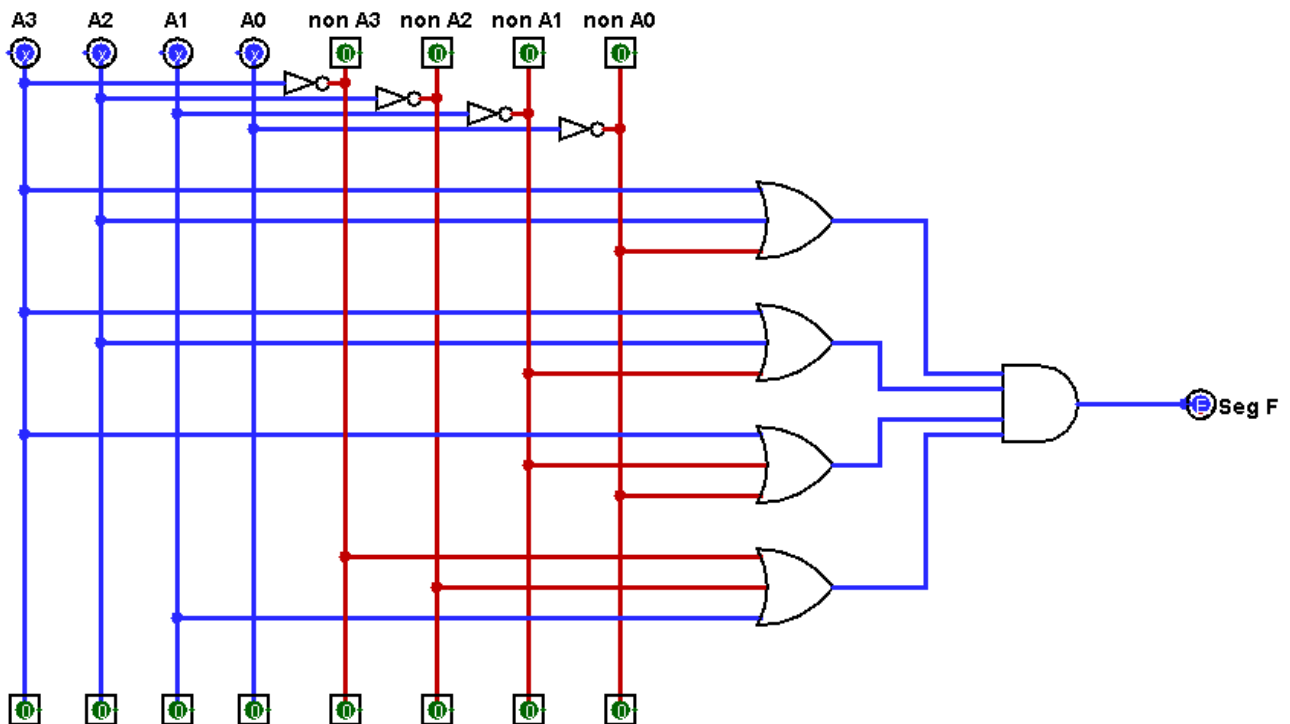
La somme de produits SegE donne ce schéma :



A remarquer sur ce schéma que deux produits ont une sortie à 1 (bleue) chacun étant suffisant pour mettre la sortie finale à 1. Ils correspondent à un cas qui est recouvert par deux groupes dans la table de Karnaugh. Chaque combinaison entrante dans une porte ET correspond à un groupe de cette table.

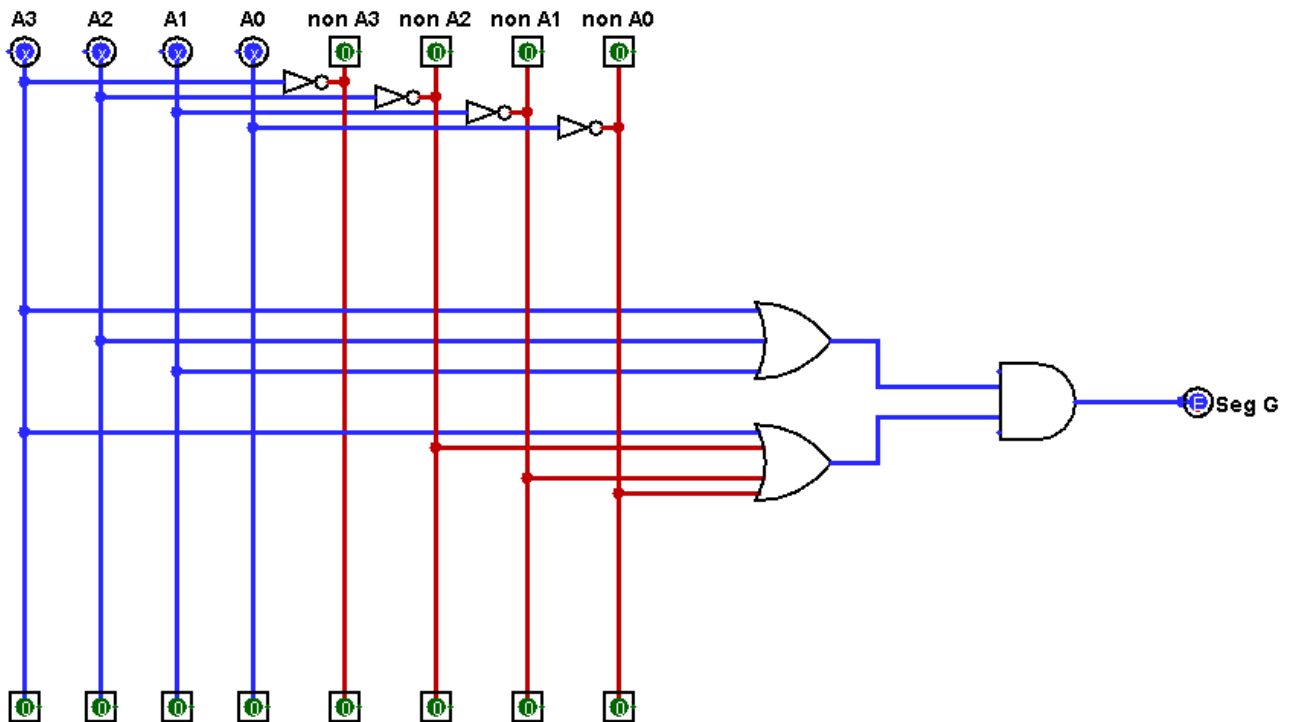
## Routage du décodeur du segment f

La produit de sommes SegF donne ce schéma :



## Routage du décodeur du segment g

Le produit de sommes SegG donne ce schéma :



Les fils bleus représentant un bit à 1, on constate que pour ces 7 routages ayant les quatre bits d'entrée à 1 qu'ils allument les bons segments pour former le F.