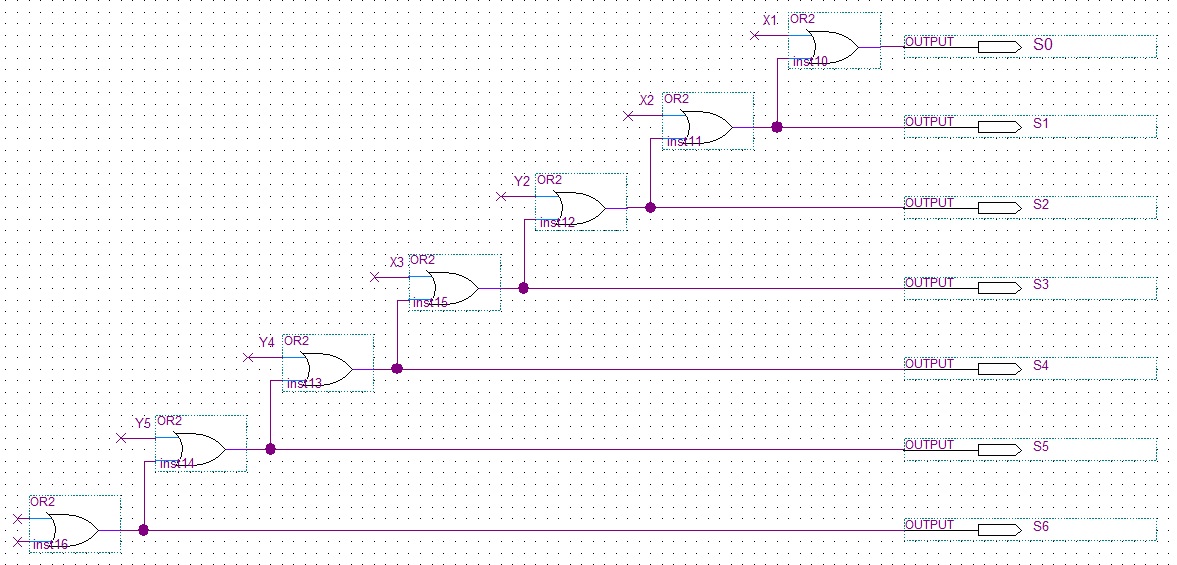
TP3 - Circuits logiques combinatoires

# Exercice 1

|  |  |  |  |
| --- | --- | --- | --- |
| X0 | X1 | X2 | S |
| 0 | 0 | 0 | 000 0000 |
| 0 | 0 | 1 | 000 0001 |
| 0 | 1 | 0 | 000 0011 |
| 0 | 1 | 1 | 000 0111 |
| 1 | 0 | 0 | 000 1111 |
| 1 | 0 | 1 | 001 1111 |
| 1 | 1 | 0 | 011 1111 |
| 1 | 1 | 1 | 111 1111 |

*Tableau exercice 1*

Comme le montre le tableau ci-dessus, en fonction de la valeur en binaire sur 3 bits entrée, on obtient un certain nombre de 1 en sortie. On a donc fait en sorte que si la sortie Si vaut 1 toutes les sorties inférieures à Si soient à 1, pour cela, nous avons utilisé des portes OU placer de la même façon que sur le schéma ci-dessous.



*Schéma position porte OU*

# Exercice 2

1. Table de vérité du circuit COMP1 :

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | Sup | Inf |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

On en déduit les expressions booléennes suivantes :

Sup = ~X . Y

Inf = X . ~Y

1. Le circuit EXP2 admet en entrée les résultats de 2 circuits COMPn : on les nomme SupX et InfX d’une part, et SupY et InfY d’autre part. La sortie Sup de EXP2 vaut 1 si X > Y, Inf vaut 1 si X < Y. On a établi les tables de Karnaugh respectives de Sup et Inf :

* Sup :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SupY InfY / SupX InfX | **00** | **01** | **11** | **10** |
| **00** | 0 | 0 | / | 1 |
| **01** | 0 | 0 | / | 0 |
| **11** | / | / | / | / |
| **10** | 1 | 1 | / | 1 |

On en déduit l’expression suivante :

Sup = SupY . ~InfX + SupX

* Inf :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SupY InfY / SupX InfX | **00** | **01** | **11** | **10** |
| **00** | 0 | 0 | / | 1 |
| **01** | 0 | 0 | / | 0 |
| **11** | / | / | / | / |
| **10** | 1 | 1 | / | 1 |

On en déduit l’expression suivante :

Inf = InfY . ~SupX + InfX

# Exercice 3

1. On propose le jeu de tests suivant :

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | Z | Résultat attendu |
| 0000 | 0000 | 0000 | 000000 |
| 1111 | 1111 | 1111 | 101101 |
| 1010 | 0110 | 1000 | 011000 |
| 0010 | 0100 | 1011 | 010001 |
| 1001 | 1101 | 0011 | 011001 |

1. Le nombre maximum que l’on peut obtenir est 1111 + 1111 + 1111 = 101101. Le résultat devra donc être codé sur 6 bits.
2. On utilise deux circuits ADD4, le premier additionnant X et Y et le second additionnant ce résultat avec Z. Si les deux ADD4 produisent une retenue, le bit S5 est à 1 ; s’il n’y a qu’une retenue, c’est le bit S4, sinon les 2 sont à 0.