



## Exercice1 : Simulation de l’additionneur :

### Additionneur complet à un bit :

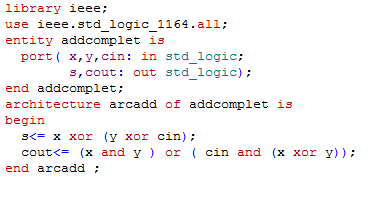


* Table de vérité :

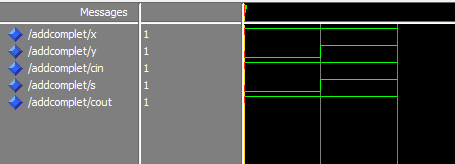
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Xi** | **Yi** | **Ci** | **Si** | **Ci+1** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

* Les équations de Si et Ci+1 :

* Code VHDl flot de données pour un additionneur complet :



* Simulation :



* Code VHDL structurelle pour un additionneur complet :

Pour ce code on doit utiliser deux composants intermédiaires : demi-additionneur, porte OU

* Demi-additionneur

Symbole :





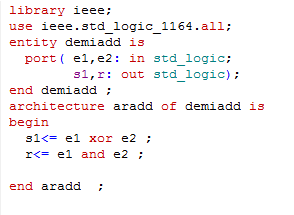


Table de vérité :

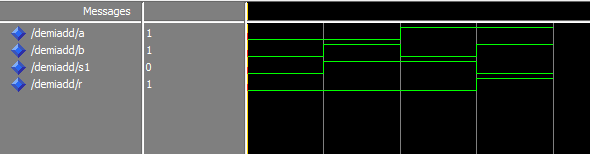
|  |  |  |  |
| --- | --- | --- | --- |
| **e1** | **e2** | **S1** | **r** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Les équations logiques :

Code VHDL :



Simulation :



* Porte OU :

Symbole :

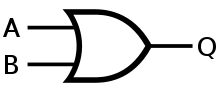
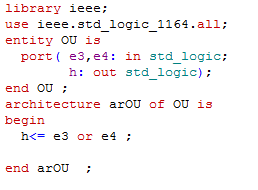


Table de vérité :

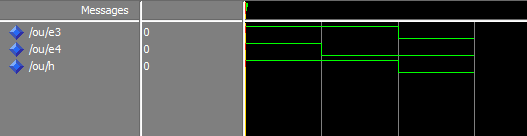
|  |  |  |
| --- | --- | --- |
| **A** | **B** | **Q** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Les équations logiques :

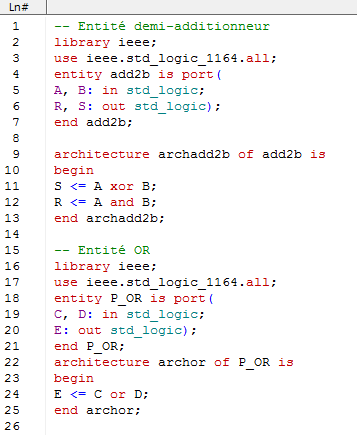
Code VHDL :

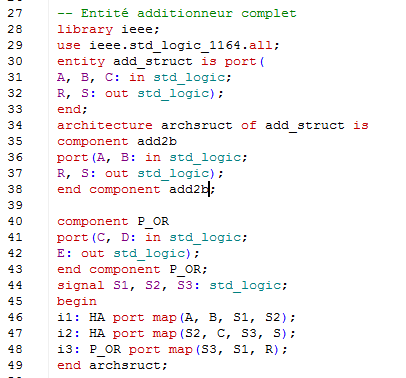


Simulation :

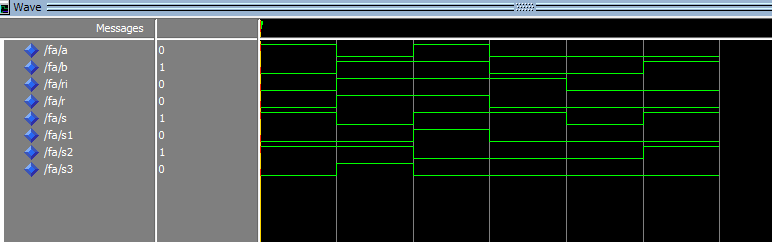


* Code VHDL principal :



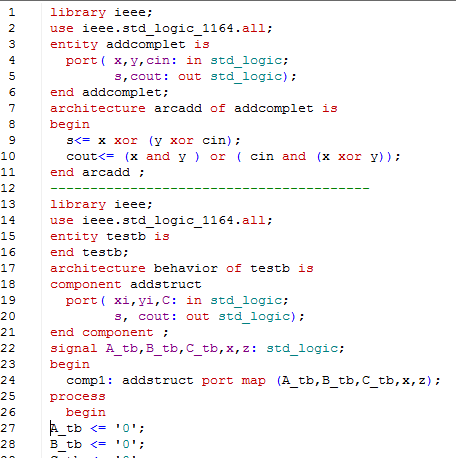


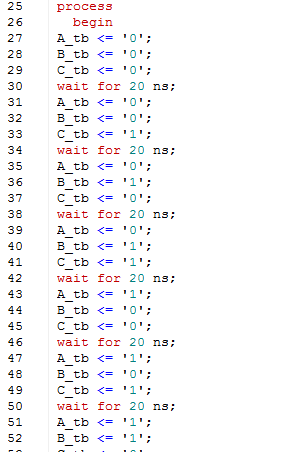
* Simulation :

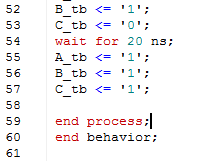


### Test Bench de l’additionneur complet a 1 bit :

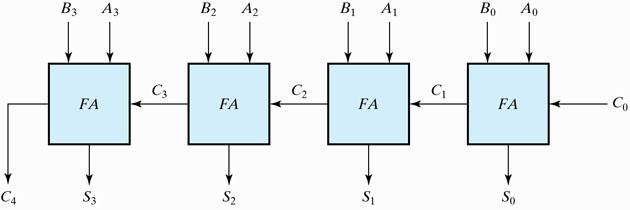
* Code :



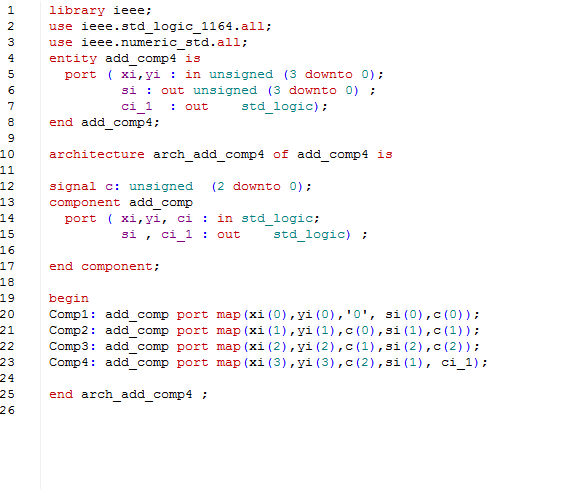




### 2.Additionneur 4 bits en utilisant un additionneur à 1 bit :



* Code VHDL :



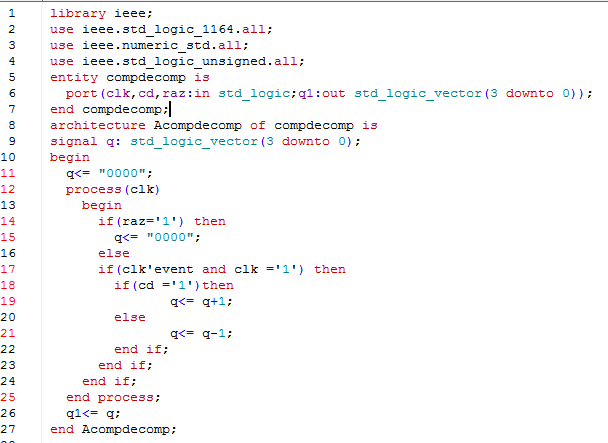
## Exercice2 :



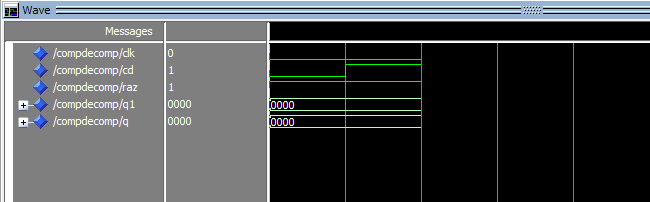


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Inputs | | | Outputs | | | |
| RAZ | CD | CLK | Q(3) | Q(2) | Q(1) | Q(0) |
| 1 | X | X | 0 | 0 | 0 | 0 |
| 0 | 1 |  | Incrémentation | | | |
| 0 | 0 |  | Décrementation | | | |

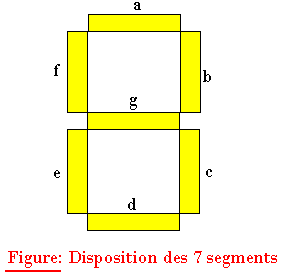
* Code en VHDL



* Simulation :



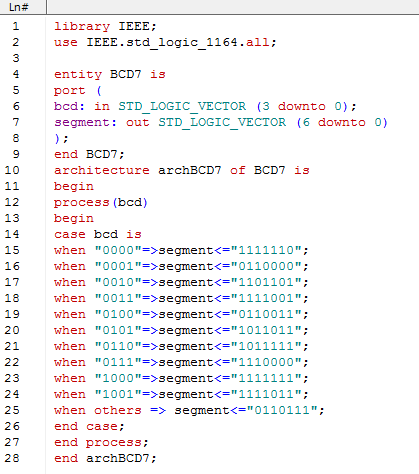
## Exercice3 : BCD vers 7-segments :



* Table de vérité :

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Entrées | | | | Sorties | | | | | | | |
| a3 | a2 | a1 | a0 | a | b | c | d | e | f | g | Affichage |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 2 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 3 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 4 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 5 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 6 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 7 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 8 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 9 |

* Code VHDL :



* Simulation :

