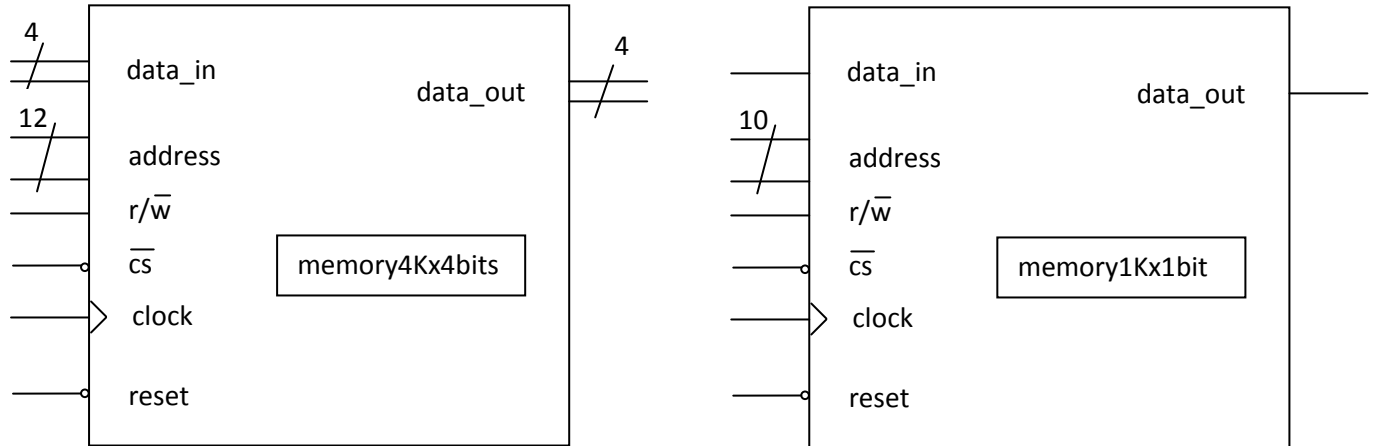


TP n°2 – Construction de Mémoire

Le but de ce TP est de construire en VHDL une mémoire de 4K x 4 bits à partir de boîtiers élémentaires de 1K x 1 bit.



Question 1

Vérifier la cohérence entre la taille des mémoires annoncées et les tailles des bus de données et d'adresses.

Question 2

Combien vous faut-il de boîtier 1K pour réaliser la mémoire principale. Proposez un câblage.

Question 3

Coder et tester en VHDL votre solution.

On vous fournit les composants suivants dans l'archive **memory-student.tgz** :

- **memory1bit.vhdl** : boîtier mémoire générique $2^{**}N \times 1$ bit
- **tb_memory1bit.vhdl** : test du boîtier mémoire $2^{**}N \times 1$ bit
- **decodeurBCD.vhdl** : décodeur décimal de nombres sur N bits
- **tb_decodeurBCD.vhdl** : test du décodeur
- **multiplexeur.vhdl** : multiplexeur $2^{**}N \rightarrow 1$
- **tb_multiplexeur.vhdl** : test du multiplexeur

Annexe - Complément de cours VHDL

1. Utilisation des composants génériques :

Pour utiliser un composant générique, on rajoute une clause **generic map** avant la clause port map à l'instanciation du composant.

Exemple d'une mémoire de 512 bits ($512 = 2^9$) :

```
u_memory : memory1bit generic map (N=>9) port map (clock => une_horloge, .....);
```

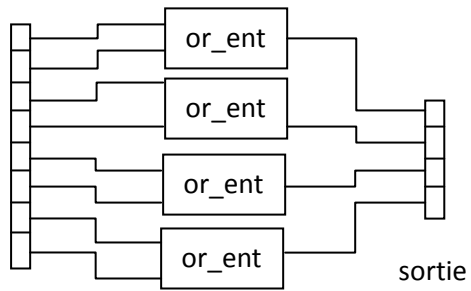
Exemple d'une décodeur décimal sur 4 bits ($512 = 2^9$) :

```
u_decodeurBCD : decodeurBCD generic map(N=>4) port map(code=> un_code, ...);
```

2. Génération en boucle de composants

Pour créer un grand nombre d'instances de composants, on peut utiliser un boucle **for ... generate**.

Exemple : OU d'un vecteur de 8 bits 2 à 2 (à partir du composant or_ent vu en cours)



entree

```
u_ors : for i in 0 to 3 generate
```

```
    u_or : or_ent port map(a => entree(i*2), b => entree (i*2+1), s => sortie(i))
```

```
end generate ;
```