

1. Demi-additionneur

Un demi-additionneur réalise l'addition de deux bits A et B et produit la somme S et une retenue C.

Ecrire la table de vérité de S et C. Proposez un circuit et validez-le expérimentalement en Hades.

Une fois le circuit enregistré sous le nom DemiAdd, on utilisera la fonction « edit>create symbol » pour créer un composant réutilisable du même nom que le fichier. Cela crée un fichier .sym , éditable en mode texte et qui représente le graphisme associé au composant.

2. Additionneur

En utilisant au moins deux demi-additionneurs de la question 1 construire un additionneur à trois entrées A B C in et sortie S Cout. Il suffit dans le menu contextuel de cliquer sur « create subsystem » et de sélectionner le fichier hds correspondant.

Vérifiez le comportement et créer un nouveau symbol Add.

3. Additionneur 4 bits, 8 bits

A partir de 4 additionneurs 1 bit Add construire en cascade un additionneur 4 bits. Créez un composant Add4. On pourra ensuite utiliser ce composant avec en entrée deux io Hex Switch et en sortie un io Hex display afin de construire un additionneur avec interface.

Vérifiez le comportement avec Hades.

Construire un additionneur 8 Bits avec 2 entrées en hexa et deux sorties en hexa plus une retenue.

4. Additionneur décimal

Proposez un additionneur décimal de deux nombres compris entre 0 et 99 qui affiche le résultat en décimal avec une retenue.

Il faudra sans doute passer par la table de vérité d'un Add10 pour des nombres compris entre 0 et 9.

Question ouverte : Comment améliorer les temps de propagation de la retenue ????