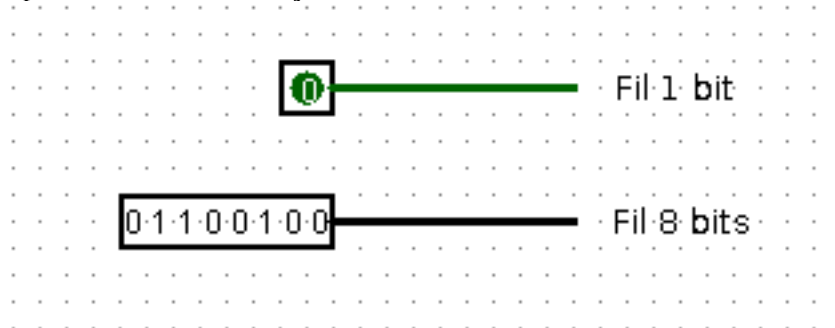


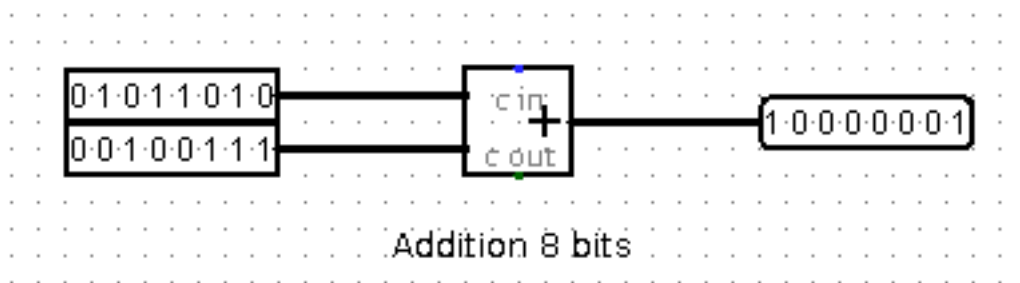
Annexe : composantes de logisim

Voici une liste des différentes composantes (ou type de composante) pouvant être utiles. De manière générale le fonctionnement est assez simple (comme d'habitude, les options de chaque composante se trouvent dans la fenêtre en bas à gauche). Les schémas sont issus du fichier tuto.circ disponible sur le site.

1. Les fils peuvent représenter un bit (0/1) ou plusieurs (jusqu'à 32). En général on peut le paramétrer pour les différents objets à l'aide de « Data bits ».

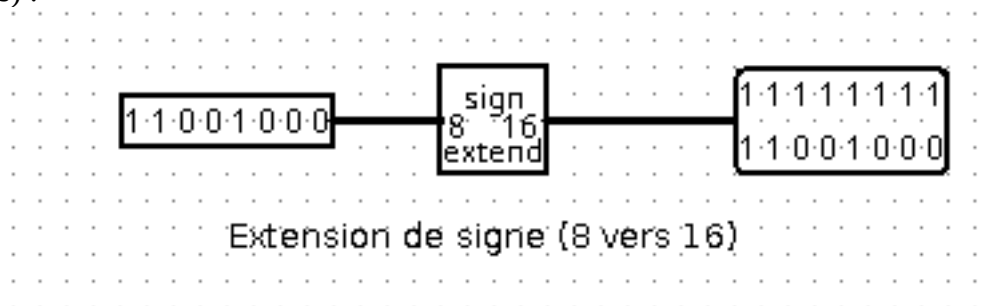


2. Il existe bcp de composantes complexes déjà faites, par exemple un additionneur :

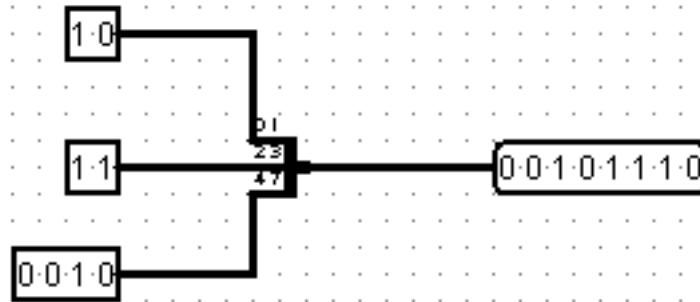


Le résultat des entrées de gauche ressort par la droite (dans son orientation par défaut). Autre exemple : le décalage de bit (shifter) : ici vous pouvez régler le décalage vers la droite/gauche ou arithmétique.

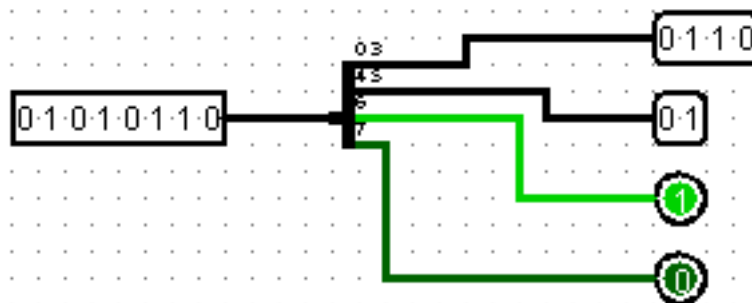
3. Vous pouvez faire des extension (ext. de signe, à zéro, à un ou même réduire le nombre de bits) :



4. Vous pouvez scinder un fil en ses différentes sous parties, ou bien regrouper plusieurs fils dans un seul à l'aide du « splitter »

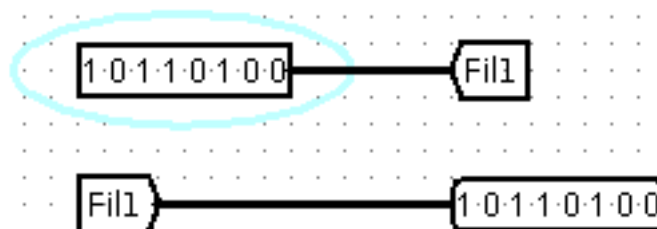


2-2-4 bits regroupés dans un fil à 8 bits

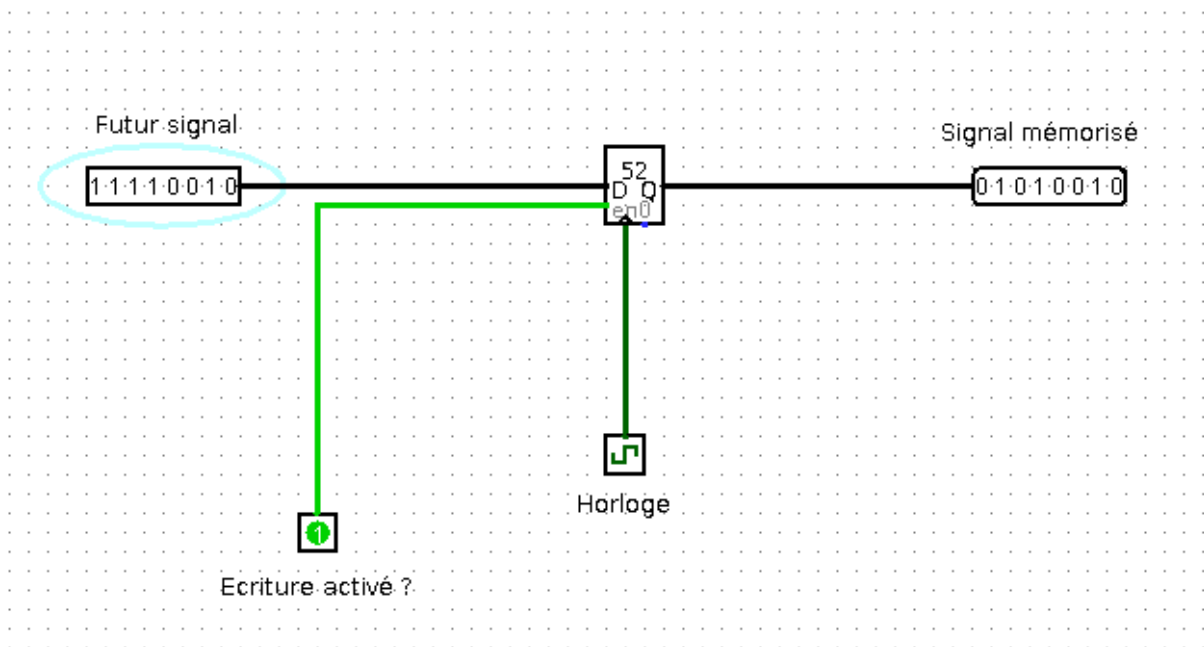


8 bits scinder en 1-1-2-4

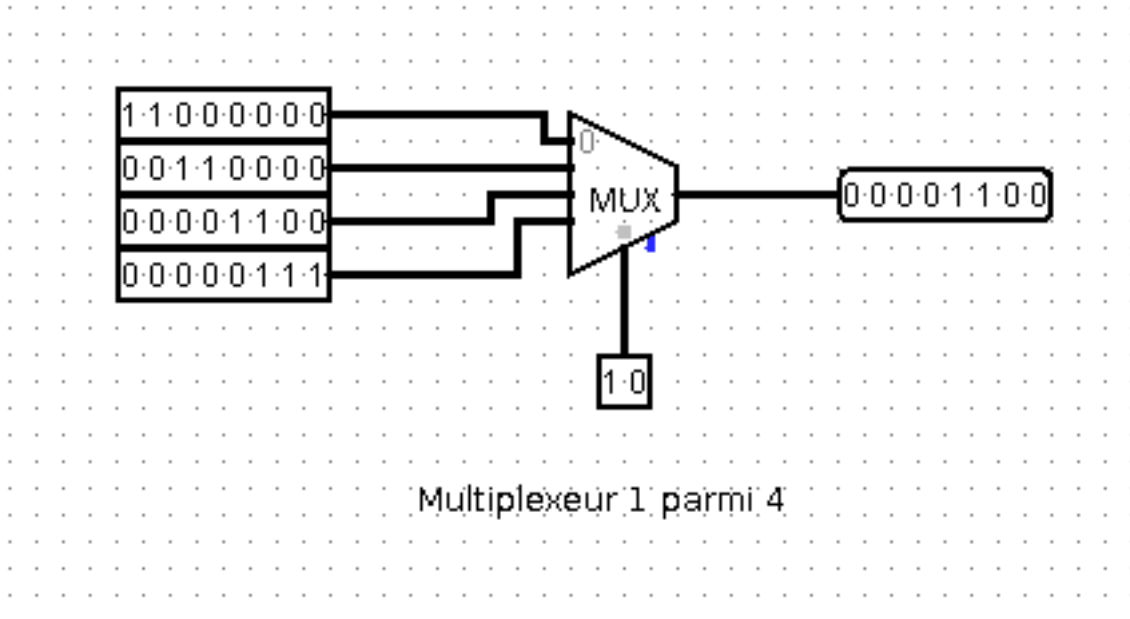
5. Vous pouvez faire des tunnels pour faire circuler les fils de façon plus propre. Il faut diriger une entrée vers un « tunnel » et donner un nom (label) au tunnel. Ensuite, il suffit de créer un autre tunnel avec le même label pour récupérer la valeur du fil.



6. Le registre est un type de mémoire (memory/register). Il contient quatre entrées et une sortie. La sortie correspond à la valeur contenu par le registre (Q). Les entrées correspondent
1. au futur signal D (Q sera mis à D sur un front montant d'horloge)
 2. à l'activation de l'écriture (si 0, on n'écrit rien dans le registre, si 1 écriture sur front montant).
 3. à l'horloge
 4. reset : pour remettre la valeur à zéro (pas présent sur le schéma).



7. Le multiplexeur fonctionne comme vu en cours :



8. La mémoire ROM (read-only memory) sera utilisée, on ne peut pas écrire dedans (sauf manuellement). On indique l'adresse par une variable d'entrée (A) et ressort la donnée contenue en mémoire (D).

