

Série de TP N° 3

1. Découvrir les différentes connexions des bascules RS et D prédéfinie dans Logisim (**Dossier memory**). Effectuer des simulations de fonctionnement (Tester les 4 types d'activation de CK).
2. Construire et simuler un registre à entrées parallèles sortie parallèles de 4 bits (utiliser 4 bascules D). Ensuite, simuler le fonctionnement d'un registre équivalent de 4 bits prédéfini dans Logisim (**Dossier memory**). Stocker dans ce registre le nombre binaire 0111. CK sera active à front montant.
3. Addition parallèle : Simuler  $i = i + 1 \rightarrow (i++)$  sur 8 bits  
Composants prédéfinis à utiliser:  
**Dossier arithmitic** : Adder de 8 bits.  
**Dossier memory** : Register de 8 bits pour le stockage de i.  
**Dossier wiring** : Clock CK pour le registre i. à chaque impulsion de CK, il y a une incrémentation de i.
4. Traduire sous la forme d'un logigramme la séquence d'instructions, ci-dessous, qui représente le calcul factoriel d'un entier n. Pour des raisons de simplicité le codage des variables « n », « i » et « Fact » sera sur 8 bits au lieu du codage réel sur 32 bits.

```
int i, n, Fact;  
Fact = 1;  
n = 6;  
for (i = 2 ; i < n; i++)  
    Fact = Fact*i;  
printf(" %d ", Fact);
```

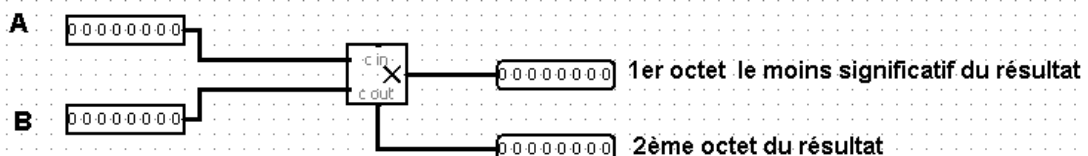
Le  $i++$  sera assuré par le logigramme de l'exercice 3

**Dossier arithmitic** : Additionneur de 8 bits.

**Dossier memory** : 3 registres de 8 bits chacun pour le stockage de n, de Fact et de i,

**Dossier wiring** : Clock CK pour les registres de i et Fact, (pour le registre de n lire la remarque ci-dessous).

**AxB sur 8 bits le résultat est sur 16 bits**



**Remarque:** Initialisation ou modification du contenu d'un registre sans pin d'entrée et sans CK.

Le contenu d'un registre est affiché en Hexa dans la boîte qui le représente :



Dans cet exemple le registre stocke la valeur 6 = 00000110

Pour initialiser un registre par une valeur, ou modifier sa valeur (au cours de la simulation):

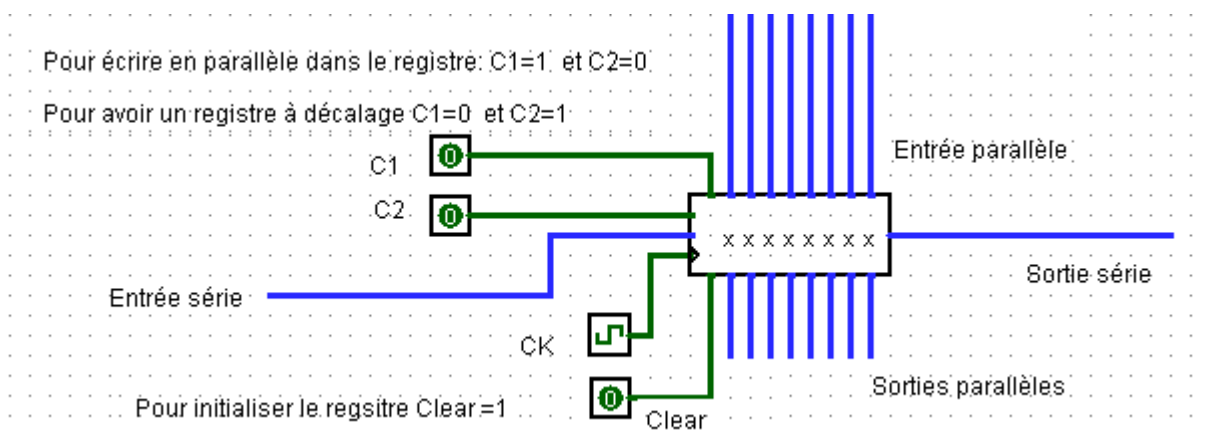


Ponter la valeur, puis utiliser le pavé numérique de votre clavier.

**5. Cet exercice permet d'avoir une autre solution de l'exercice 5 de la série de TD N°4 :**

Communication Parallèle/ série et série/parallèle entre deux registres à décalage de 8 bits à front montant, prédéfinies dans Logisim (**Dossier memory : Shift register**), nommés R1 (Emetteur) et R2 (Récepteur).

Description d'un registre à décalage :



- a. Effectuer la communication en série sur 8 bits de A = 11101010 de R1 vers R2 :
- Stockage de la valeur de A, à envoyer, dans le registre R1: entrées parallèles-sorties parallèles (C1=1 et C2=0), cliquez sur CK pour écrire A dans le registre. Ensuite, mettre C1=0.
  - Au niveau R2 : Mettre C1=0 et C2=1 pour avoir Entrée série et sorties parallèles.

A chaque clique sur CK, R1 est décalé à droite (Emission des bits de A en série) : les bits sortent bit/bit sur sa sortie série en même temps R2 est décalé à droite (Réception des bits de A en série) en recevant les bits au niveau de son entrée série. Il fonctionne donc à entrée série –sorties Parallèles.

R1 et R2 doivent être commandés par la même horloge CK.

- b. Indiquer le nombre de périodes de CK pour le transfert de A de R1 vers R2.