

**Laboratoire de systèmes numériques / Architecture des ordinateurs**

Auteurs : A. Lescourt, Prof. F. Vannel, Prof. A. Upegui (Hepia)

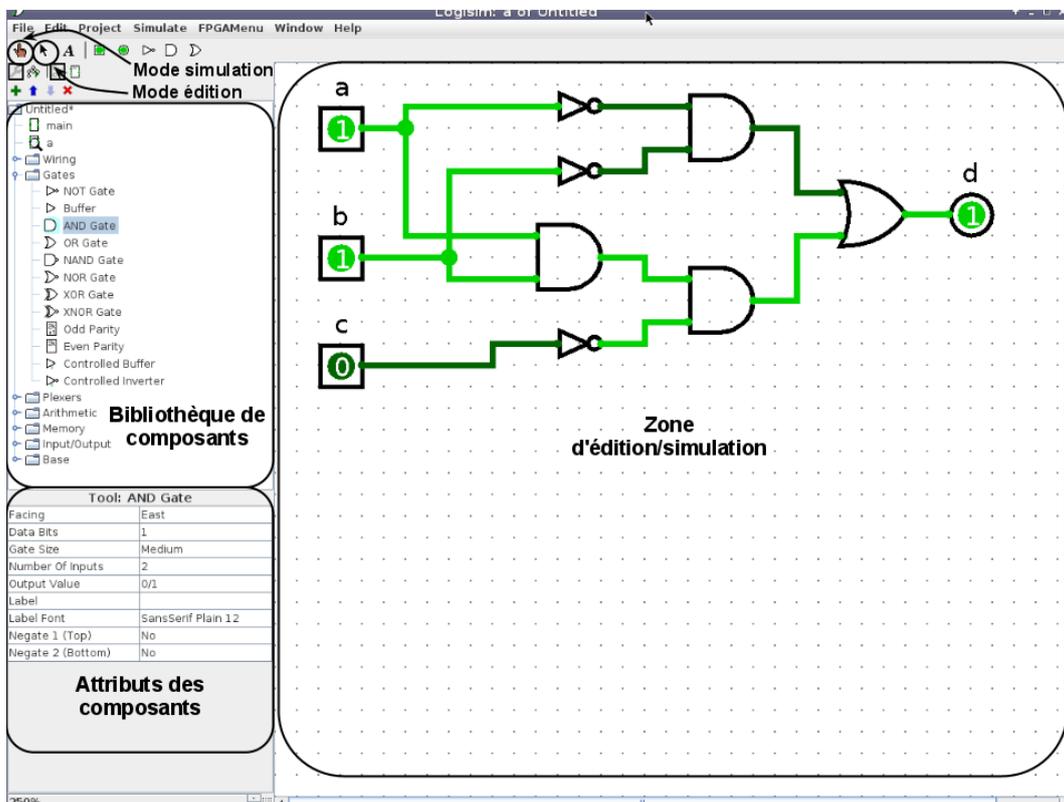
Modifications : Y. Saugy (HEIG-VD)

**Introduction à l'utilisation de logisim**

**1 Introduction**

Logisim est un logiciel open-source permettant de concevoir et de simuler des circuits logiques. Ce document est un tutoriel qui décrit comment établir un système numérique à l'aide de cet éditeur de schéma. Nous expliquerons les démarches nécessaires afin de concevoir, simuler et implémenter un projet sur une carte Altera EPM 25p-25p.

Il existe différentes façons de décrire formellement les systèmes numériques : des langages de description du matériel (HDL), des tables de vérité, des graphes d'états, ou des schémas. Logisim permet uniquement de travailler sur des schémas. Le premier chapitre expliquera comment réaliser un premier schéma.



**FIGURE 1.** Interface de Logisim

Une des particularité de Logisim est de pouvoir éditer et simuler son circuit en même temps. Nous expliquerons plus tard dans ce document comment simuler un circuit, puis comment l'implémenter sur la carte du laboratoire.

## 2 Mode édition

1. Pour utiliser le mode édition, il faut simplement sélectionner la flèche comme indiqué en haut de la figure 1.
2. On peut alors choisir un composant dans la bibliothèque sur la gauche. Pour l'ajouter dans son schéma, il suffit de cliquer sur le composant désiré, puis de cliquer sur le schéma.
3. Chaque composant que vous utiliserez aura des attributs modifiables dans la zone inférieure gauche de Logisim. Par exemple si l'on pose une porte AND, on peut modifier le nombre de signaux qu'elle prend en entrée, ou encore mettre un inverseur sur une de ses entrées.
4. Il est aussi possible de faire des copier/coller d'un ou plusieurs composants. Dans ce cas, les composants conserveront aussi tous les attributs préalablement définis.
5. Voici un descriptif des éléments que vous allez avoir besoin pour ce laboratoire :
  - Pour les entrées, l'élément Pin de Wiring.
  - Pour les sorties, l'élément Pin de Wiring avec l'attribut `output?=yes`.
  - Les portes logiques sont présentes dans le répertoire Gates.
  - Le splitter de Wiring.
  - Le ground et power de Wiring.
6. Une fois que l'on a posé tous les composants, il faut alors les connecter. Pour cela il suffit de placer le curseur avec la souris sur un des ports à connecter et, en gardant pressé le bouton gauche de la souris, le déplacer jusqu'au port de destination.

## 3 Additionneur 1 bit

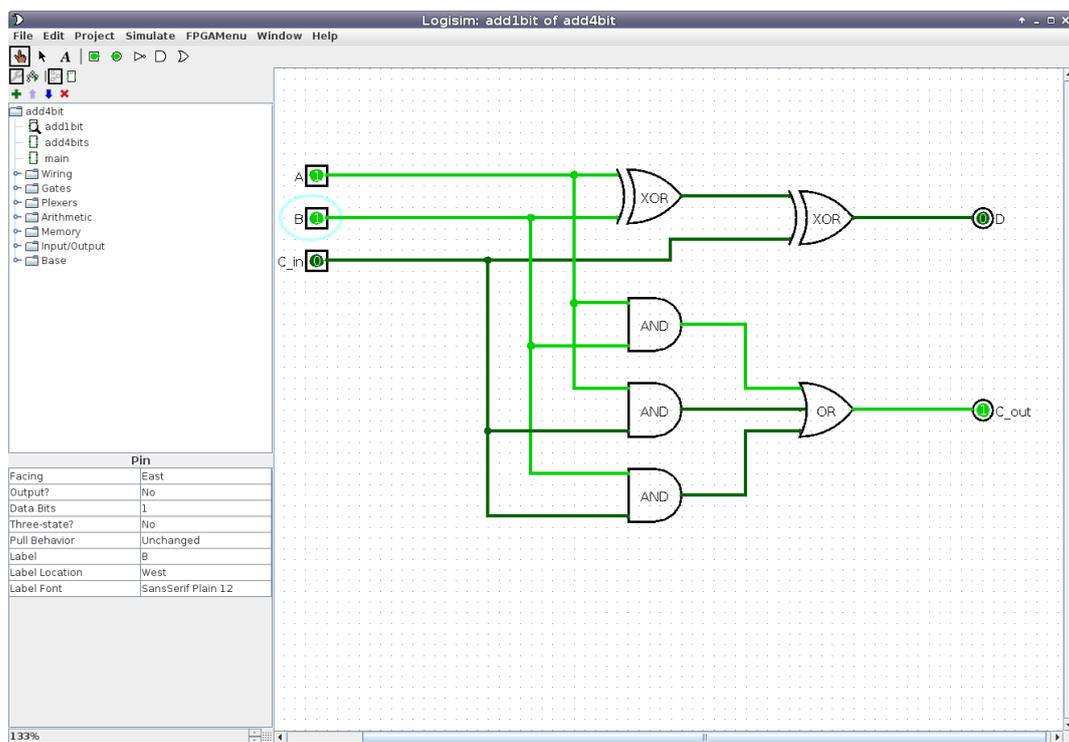


FIGURE 2. Additionneur 1 bit

Tous les circuits réalisés dans Logisim peuvent être réutilisés dans d'autres circuits. Afin de créer un nouveau circuit, il faut aller dans Project -> Add circuit... -> nommer le circuit.

### 3.1 Add1bit

Réaliser le schéma Figure 2 ci dessus. Nommez le Add1bit.

Remarques :

- Le circuit en cours d'édition est celui qui comporte une petite loupe en dessous du nom du projet.
- Ne prenez pas en compte la couleur des fils ni la valeur des Pin d'entrées (ces dernières sont un X bleu par défaut).
- Vous pouvez changer l'orientation des composants en modifiant l'attribut Facing.

## 4 Mode simulation

Logisim est capable de simuler le circuit en affichant les valeurs des signaux directement sur le schéma. L'utilisateur peut alors définir les valeurs des bits en entrée et observer la réaction du design.

1. Pour utiliser le mode simulation, il faut sélectionner la main en haut à gauche de Logisim (cf figure 1)
2. Il est alors possible de contrôler l'état des différentes entrées en cliquant directement dessus. Le X bleu des Pin d'entrées représente l'état haute impédance. Dans ce laboratoire, nous travaillerons uniquement avec des états haut ou bas. Pour supprimer cet état de haute impédance, il faut modifier les attributs de ces Pin d'entrées de façon à ce que la ligne Three-state) soit égale à No.
3. En cliquant sur une entrée, la valeur doit alterner entre '0' ou '1'.
4. Voici un descriptif des couleurs utilisées pour les signaux en mode simulation :

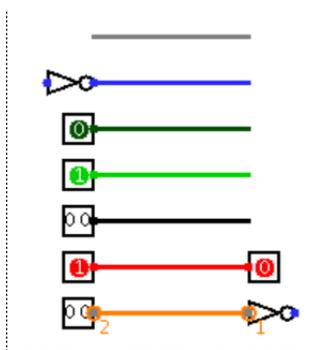


FIGURE 3. Couleurs des fils en simulation

- **Gris** : La taille du fil est inconnue. Le fil n'est relié à aucune entrée ou sortie.
  - **Bleu** : Le fil comporte une valeur, cependant elle est inconnue.
  - **Vert foncé** : Le fil comporte la valeur '0'.
  - **Vert clair** : Le fil comporte la valeur '1'.
  - **Noir** : Le fil comporte plusieurs bits (BUS).
  - **Rouge** : Le fil comporte une erreur.
  - **Orange** : Les composants reliés au fil n'ont pas la bonne taille.
5. Testez le bon fonctionnement de votre additionneur 1 bit.

## 5 Design hiérarchique

La méthodologie de design que l'on vient d'utiliser est valable pour la conception de systèmes numériques plutôt simples, c'est-à-dire avec un nombre de portes logiques plutôt bas. Lorsque l'on vise des systèmes plus compliqués on risque de voir le nombre de portes et de connexions exploser. Dans ce cas, le risque d'introduire des erreurs devient très important.

La clé pour gérer correctement une complexité plus grande est d'utiliser le design hiérarchique. Grâce au design hiérarchique on peut travailler à différents niveaux d'abstraction. D'abord on décrit des blocs de base à l'aide des portes logiques, pour ensuite utiliser ces blocs de base comme parties d'un système plus large. Dans le cas de notre additionneur 1 bit, on va en utiliser quatre pour construire un additionneur 4 bits. Ce nouveau bloc pourrait ensuite faire lui aussi partie d'un autre système plus grand.

Pour créer un design hiérarchique en incluant l'additionneur 1 bit que l'on a déjà conçu, il faudra suivre les pas suivants :

1. Créez un nouveau circuit comme déjà expliqué dans la section 3 et nommez le `Add4bits`. Pour passer de l'édition d'un circuit à l'autre, il suffit de double-cliquer sur le nom de celui désiré dans le menu de gauche.
2. Il est alors possible d'ajouter un sous circuit `add1bit` de la même manière que l'utilisation d'un composant quelconque. On clique sur `Add1bit` dans le menu indiqué sur la figure 4, puis on le place en cliquant sur le circuit.
3. Si le circuit `Add1Bit` a été créé correctement, alors il devrait être représenté par un petit bloc, avec sur sa gauche trois points bleus correspondant aux entrées et deux points rouges sur sa droite correspondant aux sorties.

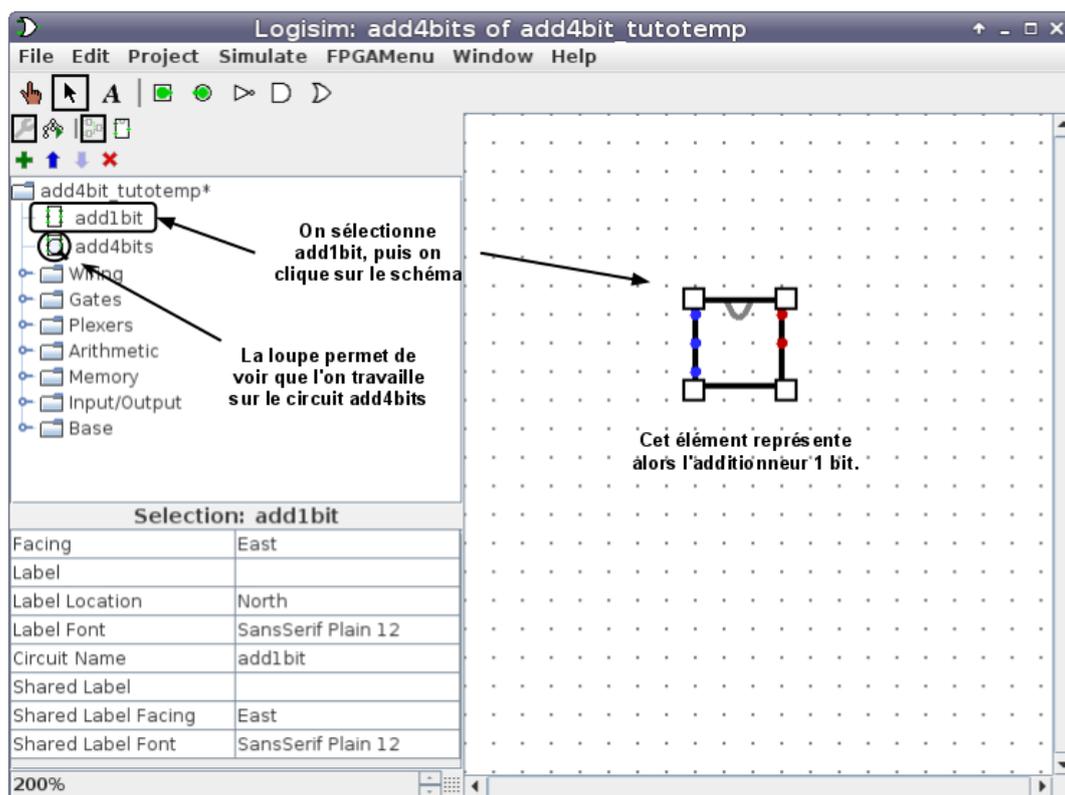
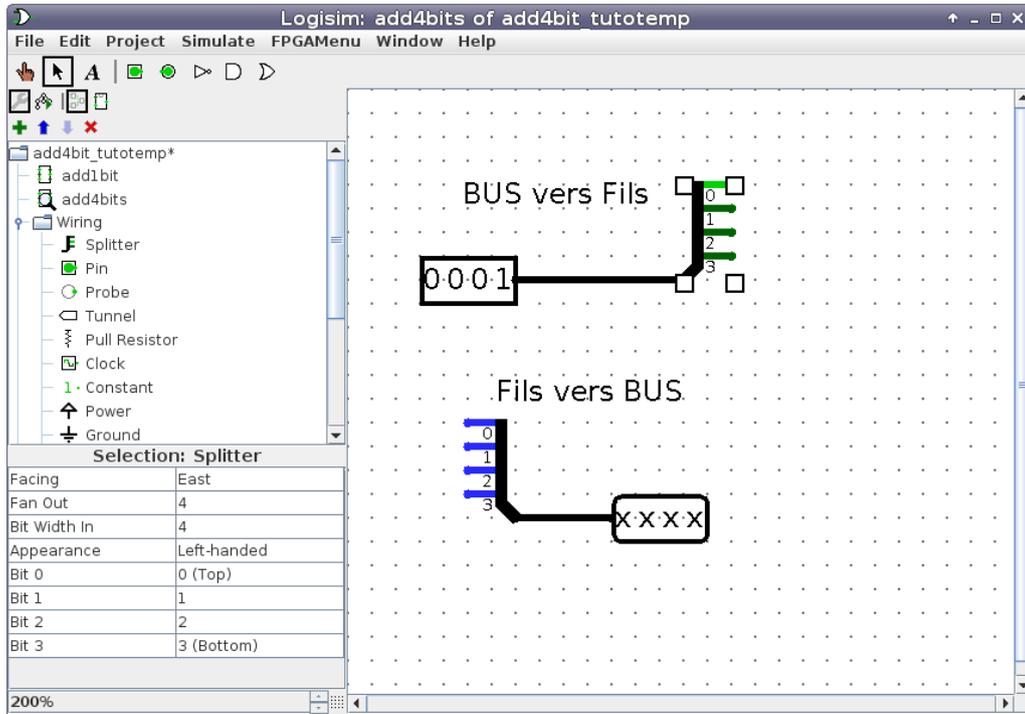


FIGURE 4. Sous circuit

4. Si les sorties apparaissent en bleu et non en rouge sur le schéma, vérifiez que vous avez bien affecté l'attribut `output?=yes` dans les pins de sorties.
5. Pour l'implémentation de l'additionneur 4 bits, il vous faut 4 additionneurs 1 bit, donc complétez le schéma en incluant les ports d'entrée et sortie. Une des différences entre les

circuits pour les additionneurs 1 et 4 bits, est le fait que dans le premier cas les entrées et sorties étaient toutes des fils indépendants, tandis que dans le deuxième cas, on a des bus de données à l'entrée et à la sortie. Par exemple, pour définir l'entrée A comme un bus de 4 bits, il faut ajouter un élément `pin` et définir sa taille via l'attribut `Data bits = 4`.

6. Lorsque l'on tire un fil de l'une de ces entrées, ce n'est plus un simple signal mais un bus de 4 bits. Pour pouvoir connecter les éléments de ce bus aux entrées des additionneurs 1 bits, on va devoir séparer les différents fils du bus afin de pouvoir les traiter un par un. L'élément `splitter` de `wiring` permet d'effectuer ces conversions dans les deux sens : d'un bus de 4 bits vers 4 fils, et de 4 fils vers un bus de 4 bits.



**FIGURE 5.** Exemples splitters

Il faut définir les tailles d'entrées et de sorties du `splitter` via les attributs `Fan out` et `Bit Width In`. Dans notre cas on définit les deux valeurs à 4.

Note : Le bit de poids faible est indexé à 0 en sortie du `splitter`.

## 6 Additionneur 4 bits

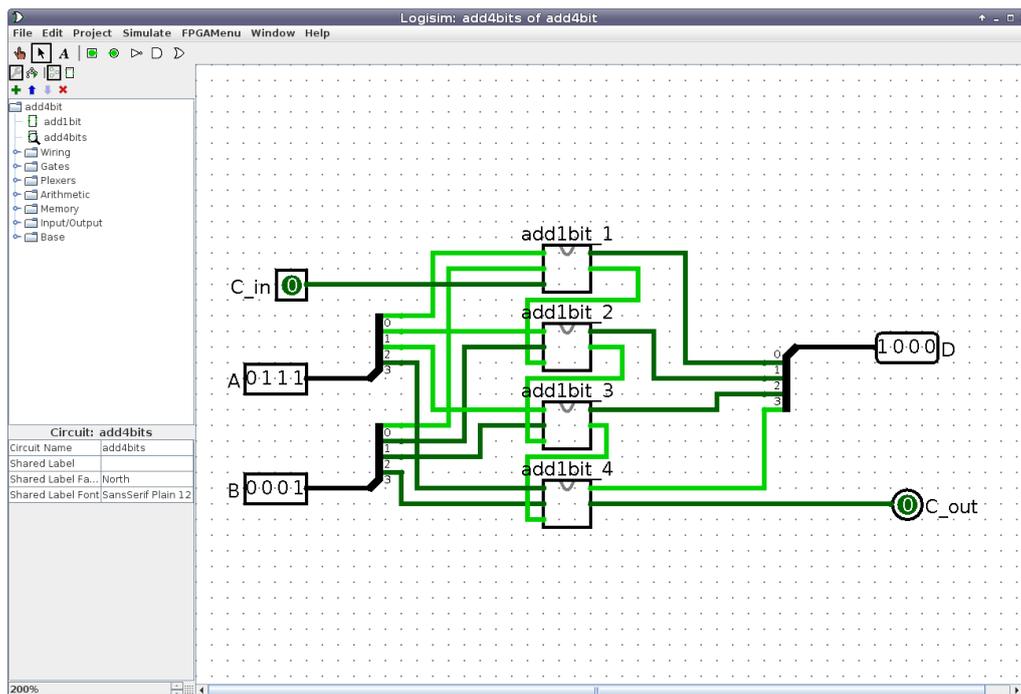


FIGURE 6. Additionneur 4 bits

Réalisez cet additionneur 4 bits puis vérifiez son bon fonctionnement en simulation.

## 7 Chronogramme

En plus de pouvoir simuler en temps réel un schéma logique, Logisim peut représenter cette simulation sous la forme d'un chronogramme. Si cette visualisation est peu adaptée aux systèmes purement combinatoire, elle est très pratique dans le cas des systèmes logiques séquentiels (utilisation d'horloges, ex : compteur, registres...). En effet, le chronogramme permet de voir les états de chaque signal au temps  $t$ .

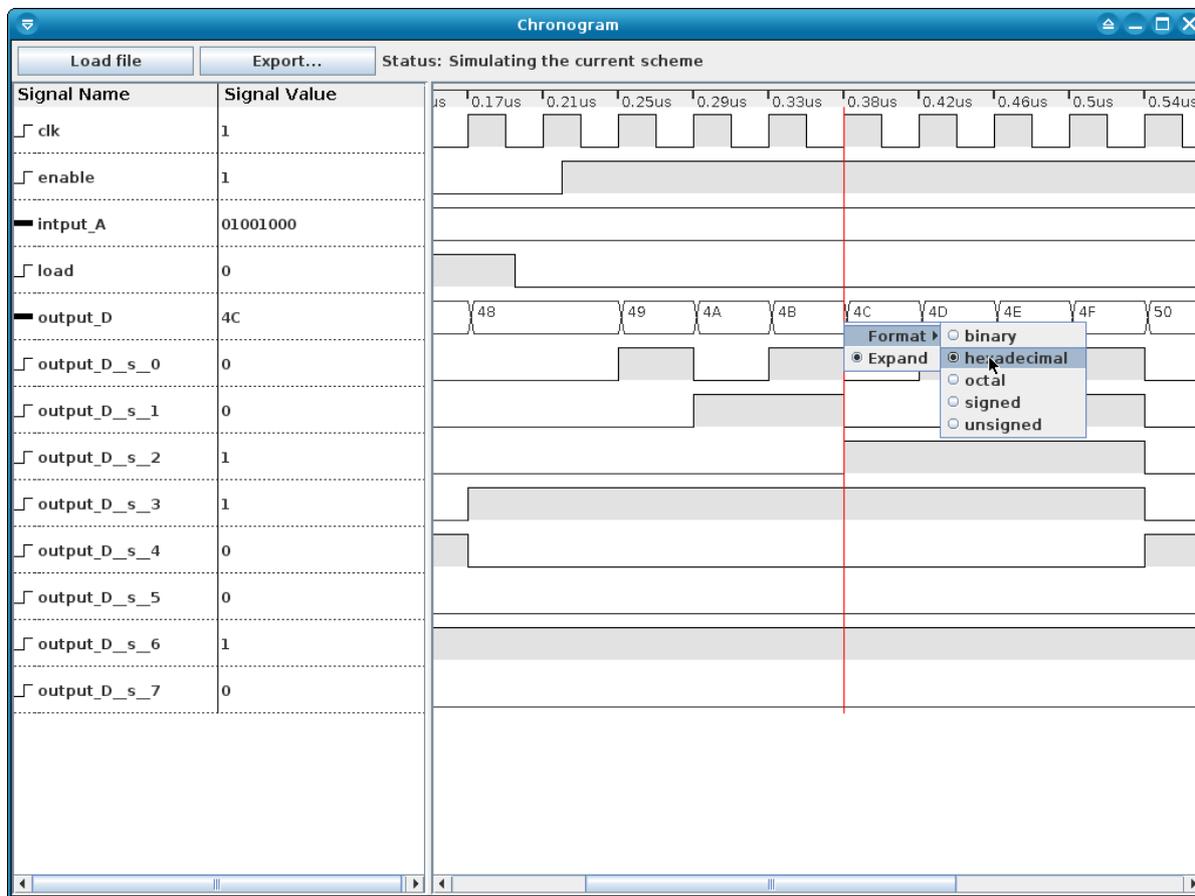


FIGURE 7. Chronogramme

### 7.1 Configuration du Chronogramme

Pour pouvoir afficher le chronogramme, il faut définir une fréquence d'échantillonnage qui indiquera au système à quels moments il doit afficher les valeurs de chaque signal. Cette fréquence doit être défini par une horloge d'échantillonnage ajouté au circuit simulé, qui ne doit être reliée à aucun composant.

#### 7.1.1 Horloge d'échantillonnage

Pour créer l'horloge d'échantillonnage il faut :

- Aller dans le liste de composant, prendre Wiring -> clock et l'ajouter au schéma.
- Aller dans les attributs de cette horloge et la nommer sysclk (Attribut Label)
- Assurez vous que les attributs High Duration et Low duration sont bien définis à un coup d'horloge.
- **Ne reliez cette horloge à aucun composants !**

- Pour définir la fréquence de cette horloge : `Simulate -> Tick Frequency`. Prenez par exemple 4Hz.

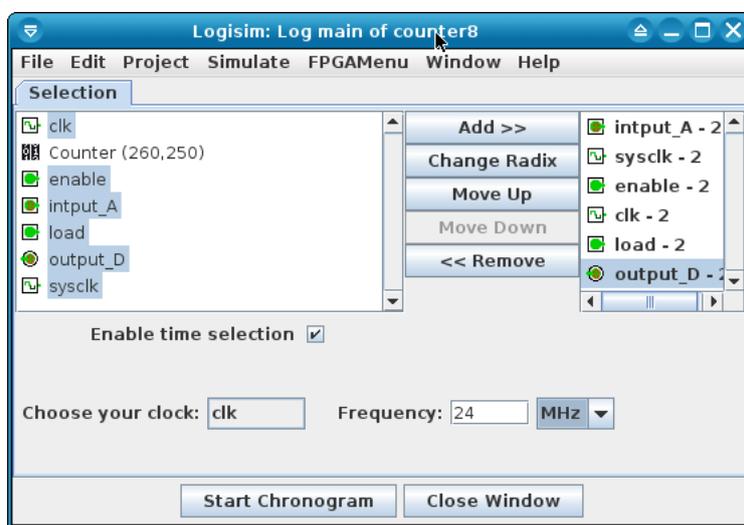
### 7.1.2 Ajout d'une horloge interne

L'horloge interne de votre système doit être plus lente que `sysclk`. Elle s'ajoute de la même façon que l'horloge d'échantillonnage : `Wiring -> clock`. Usuellement, on nome l'horloge de son système `clk`.

Pour définir sa fréquence, il faudra modifier ses attributs afin de régler combien de ticks cette horloge nécessite avant de changer d'état. Par exemple si vous avez choisi une fréquence d'échantillonnage de 4Hz, mettez les attributs `High duration` et `Low duration` de votre horloge à 4 ticks. Votre horloge changera alors d'état une fois par seconde.

### 7.1.3 Sélection des signaux

Il faut ensuite définir quels signaux nous voulons afficher dans le chronogramme.



**FIGURE 8.** *Sélection des signaux*

- Ouvrez le menu `Simulate` puis sélectionnez `Chronogram`. La liste des signaux disponibles apparaît.
- Sélectionnez les signaux que vous voulez et cliquez sur `Add >>`
- Il est indispensable que l'horloge d'échantillonnage `sysclk` soit aussi ajoutée.

Il est possible (mais pas indispensable) d'afficher une base de temps, selon la fréquence d'un signal d'horloge. Il faut activer la case `Enable time selection`, puis choisir l'horloge de votre système ainsi que sa fréquence.

Démarrez le chronogramme via le bouton `Start Chronogram`, puis organisez les fenêtres afin d'être capable de voir à la fois le schéma et le chronogramme.

## 7.2 Utilisation du chronogramme

A chaque changement d'état de l'horloge `sysclk`, le chronogramme se mettra à jour. Vous pouvez le faire manuellement en cliquant sur le composant en mode simulation.

Il est aussi possible de démarrer et d'arrêter l'horloge : `Simulate -> Ticks Enabled` (Raccourci : `Ctrl+k`)

Contrôles du chronogramme :

- Valeur à l'instant  $t$  : En cliquant sur un signal dans la partie droite du chronogramme, la valeur des données à cet instant sera affichée à côté du nom du signal.
- Scrollez avec la molette sur le chronogramme pour zoomer/dézoomer (focus sur la zone sélectionnée).
- Click droit sur un bus ->`Format` pour changer son format d'affichage.
- Click droit sur un bus ->`Expand` afficher chaque signal composant le bus.

### 7.2.1 Sauvegarde et chargement

Une fois la simulation terminée, vous pouvez exporter ce chronogramme dans un logFile via le bouton `Export`. Créez un fichier `.txt` ou `.log` à l'emplacement souhaité.

Le chargement d'un de ces chronogrammes se fait via le bouton `Load External File`. Dans ce cas, le chronogramme ne prend plus en compte les signaux du schéma courant. Il affiche simplement l'état des signaux tels qu'ils étaient lors de l'export. Par ailleurs, si vous voulez utiliser le chronogramme uniquement pour visualiser un fichier préalablement exporté, il faut simplement lancer le chronogramme avant de charger le fichier : (`Simulate -> Chronogram -> Start Chronogram -> Load file`).

## 8 Programmation de la carte EPM 25p-25p

La version du laboratoire de Logisim permet de programmer directement la carte EPM 25p-25p. Il faudra alors associer les Pin en entrées à des boutons, et ceux des sorties à des LEDs.

### 8.1 Connexion

La carte EPM 25p-25p doit être alimentée avec une alimentation 5 volts. Sa programmation se fait via le module USB-Blaster. Ces deux éléments doivent être branchés comme sur la figure 9.

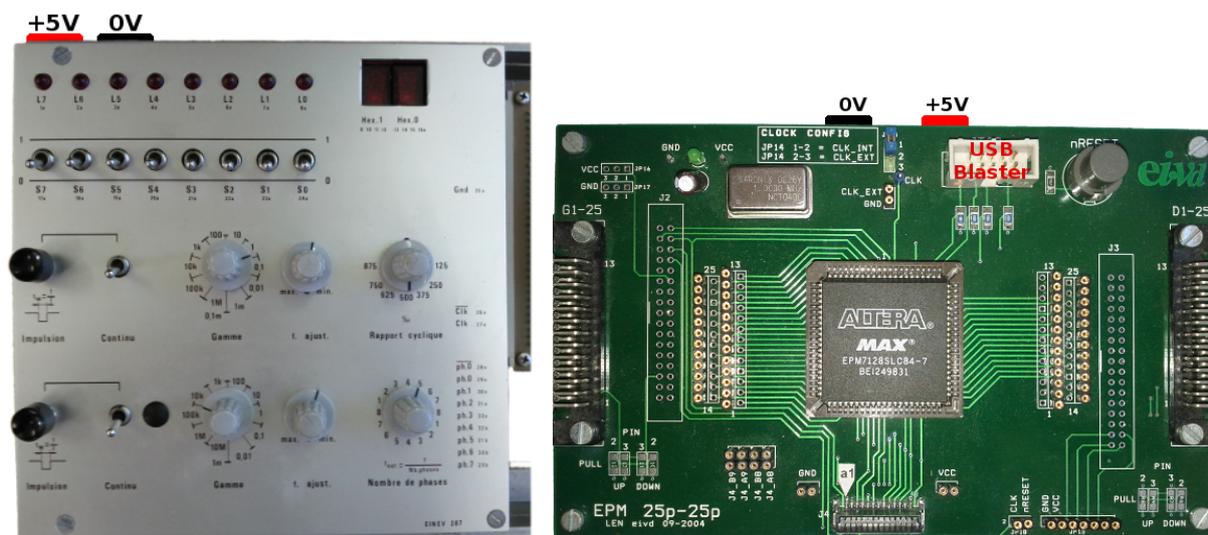


FIGURE 9. Connexion de la carte EPM 25p-25p

### 8.2 Configuration

1. Avant de pouvoir programmer la carte, il est nécessaire d'annoter chaque entrée/sortie en leur affectant l'attribut Label. On utilisera les termes A, B et D pour respectivement les deux bus d'entrées et le bus de sorties. Les retenues seront annotées avec C\_in et C\_out. Il faut aussi annoter tous les composants avec des identifiants uniques. Les quatre additionneurs 1 bits auront alors pour label add1bit\_X ou X est un numéro unique.
2. On peut ensuite ouvrir l'interface de programmation via FPGAMenu -> FPGA Commander.

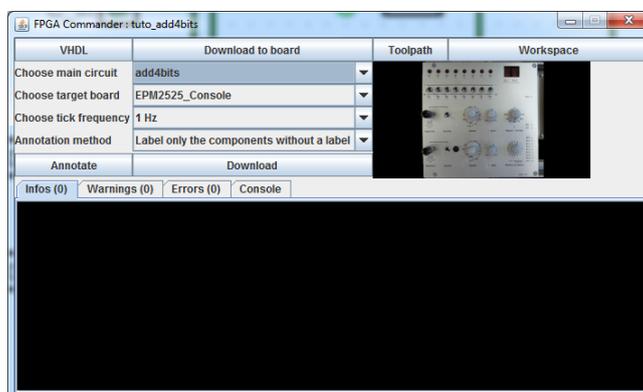
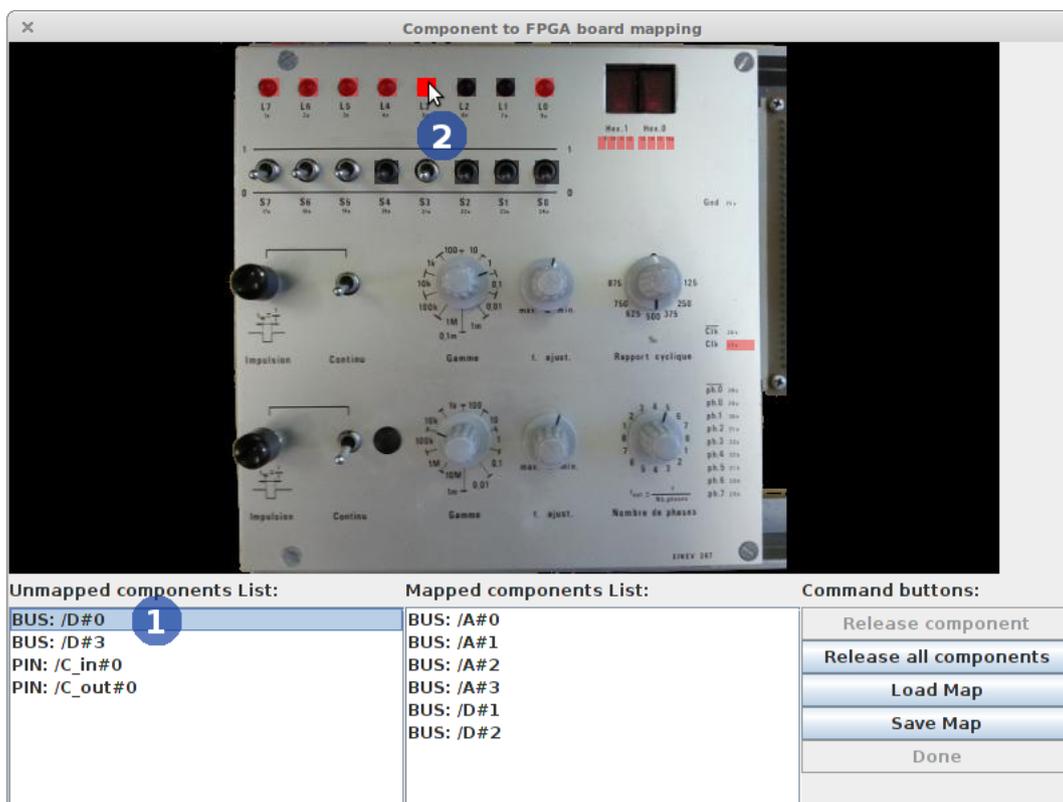


FIGURE 10. FPGA Commander

3. Plusieurs paramètres doivent alors être définis :

- **Toolpath** : Il faut s'assurer qu'il pointe vers  
C:\EDA\Altera\Quartus\v10\_0\quartus\bin.
  - **Workspace** : Il doit être défini vers D:\ARO1
  - **Choose main circuit** : Le circuit top level que nous voulons programmer. Dans notre cas c'est Add4bits.
  - **Choose target board** : La carte que nous allons utiliser. Il faut choisir la carte du laboratoire EPM2525\_Console.
  - Les autres paramètres sont laissés à leur valeur par défaut.
4. Afin de programmer la carte EPM 25p-25p, Logisim va tout d'abord générer plusieurs fichiers .vhd. Il faut pour cela cliquer sur **Design Rule Check**, puis sur ce même bouton qui sera devenu **Generate HDL**. Tous les fichiers générés sont affichés dans l'onglet Infos de la console.
  5. Il faut ensuite associer les entrées/sorties de l'additionneur aux boutons et aux LEDs de la carte. Cliquez sur **Ressource check and map** pour afficher l'interface permettant ce mapping.



**FIGURE 11.** Console Board Mapping

6. Toutes les entrées/sorties que l'on a défini sont visibles dans la zone **Unmapped components list**. Si l'on veut par exemple mettre le bit 0 de l'entrée A sur l'interrupteur tout en bas à droite de la carte, il faut sélectionner PIN: /A<0>, puis cliquer sur l'interrupteur dans l'image. Si le mapping est possible, lorsque l'on passe le curseur sur l'interrupteur, ce dernier devient rouge.
7. Placez les éléments de la manière suivante (cf figure 12) :
  - Les 4 bits de l'entrée A sur les 4 premiers switches (S0 à S3).
  - Les 4 bits de l'entrée B sur les 4 switches suivants (S4 à S7).
  - L'entrée C\_in sur l'horloge (Clk)
  - Les 4 bits de la sortie D sur les 4 premières LEDs en bas à droite (L0 à L3).
  - La sortie C\_out sur la LED L7



**FIGURE 12.** *Position des entrées/sorties*

- Une fois que toutes les entrées et sorties ont été définies, cliquez sur Done
8. Il ne reste plus qu'à télécharger le code sur la carte via le bouton Download.