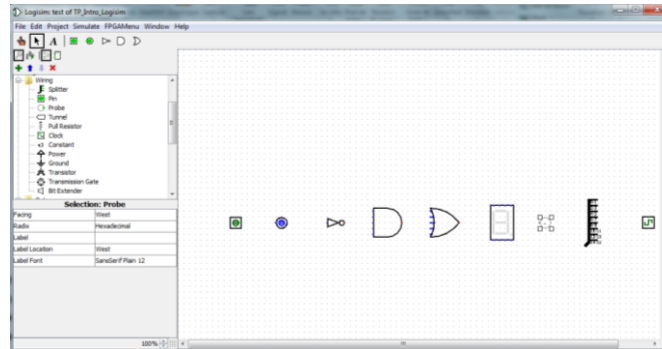


## TP d'introduction au logiciel Logisim

### Architecture des ordinateurs / Polytech Nice Sophia



Téléchargez et installez la version 3.5 de Logisim depuis [lms.univ-cotedazur.fr](https://lms.univ-cotedazur.fr) ou bien <https://github.com/reds-heig/logisim-evolution>

**Vous devez disposer d'une installation de java récente ( $\geq 11$ ).**

#### Environnement

L'environnement Logisim est composé de 5 zones graphiques :

- **La feuille d'édition** au centre,
- **La frame interactive**, en haut à gauche, proposant par défaut de visualiser la bibliothèque de composants et votre environnement de travail contenant vos propres circuits (par défaut uniquement le circuit main),
- **La barre de menus**,
- **La frame informative**, en bas à gauche, donnant des indications sur le composant sélectionné et permettant de paramétrer celui-ci,
- **Les boutons de la barre d'édition** principale en haut.

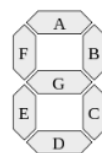
#### Edition d'un nouveau circuit (cas général)

L'édition de circuit sous logisim se fait en 6 étapes :

1. Cliquer sur le bouton + (Add Circuit) et nommer votre composant,
2. Réaliser l'édition du diagramme schématique du circuit à partir de portes logiques ou de composants déjà réalisés,
3. Ajouter des pins d'entrée/sortie pour votre composant en les dotant d'un label,
4. Réaliser l'édition de l'apparence du circuit (son interface réutilisable) en plaçant les pins d'entrée/sortie sur la périphérie du composant. Pour cela vous devrez cliquer sur le bouton 12 « éditer le dessin de sous circuit du circuit affiché »
5. Sauvegarde de votre composant qui apparaîtra dans votre répertoire de travail,
6. Instanciation du composant réalisé au sein d'un composant de plus haut-niveau.

#### Réalisation de votre premier composant

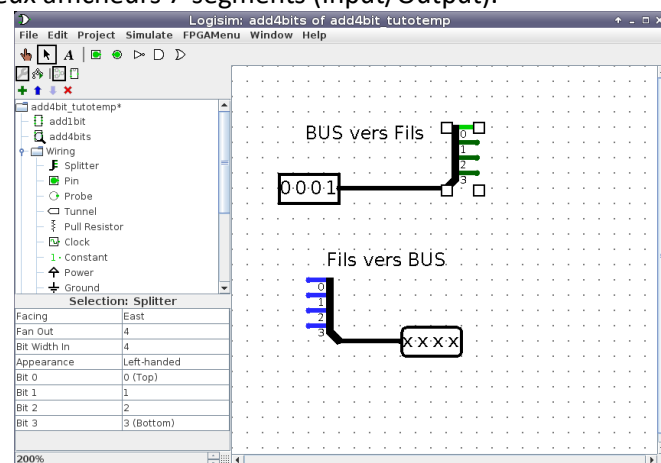
Vous allez commencer par réaliser un premier circuit simple, le décodeur 7-segments à partir des fichiers disponibles sur moodle.



*Figure 1 Afficheur utilisant 7-segments notés de A à G*

Ce circuit décode une donnée sur 4 bits allant de 0 à 15 pour afficher le symbole hexadécimal correspondant sur l'afficheur 7 segment illustré précédemment.

- Ecrire le table de vérité du décodeur en dessinant les représentations des 16 premiers symboles hexadécimaux de 0 à F,
- Suivre les 6 étapes précédentes, mais en passant par le menu Project-> Analyse circuit qui permet de générer directement le schéma associé à la table de vérité.
  - Si cette fonction n'est pas disponible, lancer logisim depuis un shell :  
`java -jar logisim-evolution.jar -analyze`
  - renseigner le nom des 4 bits d'entrées (Inputs) : **e3 e2 e1 e0**,
  - puis les 7 signaux de sortie (Outputs) : **a b c d e f g**,
  - puis le remplissage de la table de vérité et la génération du layout (onglet Table).
- Terminer les autres étapes de l'édition en cliquant sur : générer, OK, oui.
- Profiter de cet outil pour observer les formes simplifiées des équations logiques de chaque signal (variable logique).
- puis instancier ce décodeur dans un nouveau circuit qui utilisera :
  - un compteur sur 8 bits en guise de donnée d'entrée,
  - son horloge (bibliothèque wiring) avec le label sysclk,
  - un splitter (wiring) pour séparer ce mot de 8 bits en 2 mots de 4 bits à envoyer à 2 instances de votre décodeur (voir figure ci-dessous),
  - deux afficheurs 7-segments (Input/Output).



### Première simulation et tests

- Vérifier le comportement de ce circuit en passant en mode simulation (menu simulate-tick enabled ou en cliquant directement sur l'horloge pour une simulation pas à pas).
- Vous allez maintenant automatiser le test de votre circuit décodeur en remplissant le vecteur de test fourni sur moodle. Ce fichier énumère les tests unitaires de votre circuit. Lisez attentivement l'organisation de ce fichier, vous aurez à en utiliser dans le projet.
  - Compléter la fin des tests,
  - lancer le test en mode GUI en chargeant le fichier de test depuis le menu simulation -> vecteurs de test
  - lancer ensuite le même vecteur de test depuis le shell avec la commande :  
`java -jar logisim-evolution.jar -testvector circuit_name test_vector_name.txt file_name.circ`

### Réalisation du projet

Vous êtes maintenant prêts pour démarrer le projet **PARM** (Polytech ARM-based embedded processor). La documentation nécessaire à la réalisation du projet ainsi que les fichiers communs qui sont mis à votre disposition sont disponibles sur Moodle.

Vous pouvez dès à présent vous organiser en groupes de 4 étudiants et commencer à lire la documentation pour vous répartir le travail.