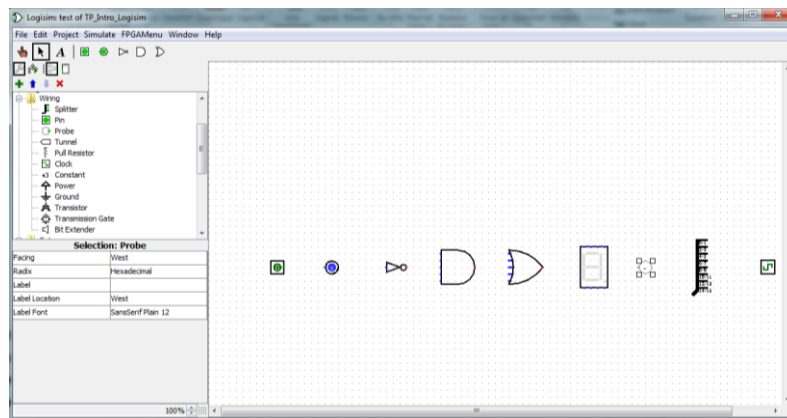


TP d'introduction au logiciel Logisim

Edition et simulation de circuits numériques

Polytech Nice Sophia



Téléchargez et installez la version Polytech de Logisim par les commandes suivantes (Ubuntu 16.04)

- `sudo add-apt-repository ppa:polytech-nice/logisim-evolution`
- `sudo apt update`
- `sudo apt install logisim-evolution`

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 « edit viewed circuit's subcircuit appearance »
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.

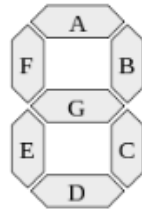
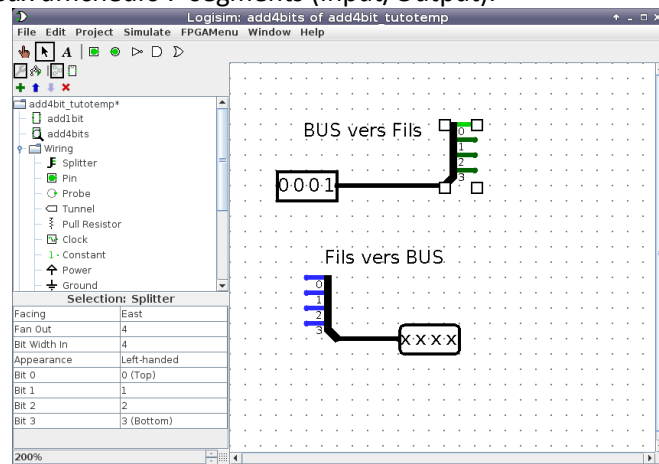


Figure 1 Afficheur utilisant 7-segments notés de A à G

- Ecrire le table de vérité du décodeur,
- 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.
 - renseigner le nom des 4 bits d'entrées (Inputs),
 - puis les 7 signaux de sortie (Outputs),
 - puis le remplissage de la table de vérité et la génération du layout (onglet Table).
- Terminer les autres étapes de l'édition
- puis instancier ce décodeur dans un nouveau circuit qui utilisera :
 - un générateur pseudo-aléatoire de 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 probes (wiring) pour vérifier la valeur de chaque bus de 4-bits en hexadécimal,
 - deux afficheurs 7-segments (Input/Output).



Première simulation

- 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).
- Visualiser le chronogramme temporel de la simulation : menu simulate-> chronogram. Ajoutez les objets à visualiser : le générateur random, les 2 probes, et l'horloge. Cochez 'enable time simulation' et choisissez sysclk comme horloge puis Start Chronogram.

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 Git :

https://github.com/MIAOU-Polytech/SI3_PEP_Project

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