

1. Objectif

Le but de ce TP est la découverte du logiciel LOGISIM. Il est fortement conseillé de prendre des notes durant la séance de TP. **Vous pouvez utiliser l'aide du logiciel pour obtenir des informations sur le mode d'emploi du logiciel et des composants utilisés.** Si après avoir cherché la réponse à une question, vous n'avez pas la solution, faites appel à l'enseignant responsable du TP.

2. Introduction


Pour commencer, allez sous moodle dans la section : BUT 1A → S2 → R204 – Communication et fonctionnement bas niveau → Logiciel LOGISIM. Téléchargez la version du logiciel souhaitée (Windows, MAC ou Linux) sur le bureau puis lancez la. Si vous ne parvenez pas à lancer LOGISIM, consultez l'aide fournie sur les différentes versions ou demandez de l'aide à votre chargé de TP.

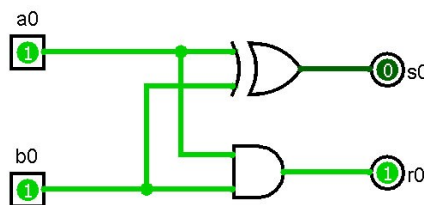
L'aide (en Anglais) intégrée dans le logiciel LOGISIM est très complète. Vous pouvez vous baser dessus pour compléter votre apprentissage de l'utilisation du logiciel. Éventuellement, vous pouvez chercher d'autres ressources sur Google : <https://www.google.com/search?q=logisim+pdf>


Pour information, le logiciel LOGISIM est téléchargeable librement sur sourceforge à l'adresse suivante : <https://sourceforge.net/projects/circuit/files/2.7.x/2.7.1/>

Il est écrit en Java et nécessite une version de Java à jour. Il est disponible aussi bien pour Windows, Mac, que Linux. Sous linux, pour le lancer, il faut ouvrir une fenêtre terminal et taper : `java -jar <chemin vers le fichier.jar téléchargé>`. Le site web du logiciel est : <http://www.cburch.com/logisim/>

3. Demi-additionneur

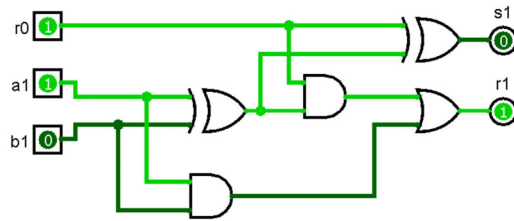
1. Dans la barre d'outils de LOGISIM cliquez sur l'icône représentant la flèche  pour passer en **mode édition** de circuits puis réalisez le circuit ci-dessous correspondant à un demi-additionneur.



2. Dans la barre d'outils de LOGISIM cliquez sur l'icône représentant la main  pour passer en **mode simulation** et cliquez alternativement sur les entrées **a0** et **b0** pour choisir les valeurs des bits à additionner. Vérifier le résultat pour les quatre cas possibles (0+0, 1+0, 0+1, 1+1).

4. Additionneur complet

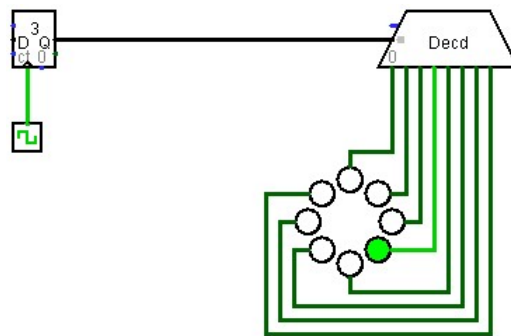
1. Réalisez le circuit ci-dessous correspondant à un additionneur complet.



2. Passez en mode simulation et cliquez alternativement sur les entrées **a1**, **b1** et **r0** pour choisir les valeurs des bits et de la retenue à additionner. Vérifier à chaque fois le résultat.

5. Roue lumineuse

On se propose de réaliser un circuit simulant une roue lumineuse.



Pour cela on utilisera les composants suivants :

- Une horloge (située dans le dossier Wiring de LOGISIM) Clock
- Un compteur (Counter situé dans le dossier Memory de LOGISIM) Counter
Configurez le champ **Data Bits** du compteur sur 3 (bits) de façon à produire de façon répétitive la séquence : 0,1,2,3,4,5,6,7,0,1....
- Un décodeur (situé dans le dossier Plexers de LOGISIM) Decoder
Configurez le champ Select Bits à 3 et le champ Facing à South
- 8 LED (disponibles dans le dossier Input/Output de LOGISIM) LED
Configurez le champ **On Color** à #00ff00 (couleur verte lorsque la diode est allumée) et le champ **Off Color** à #ffffff (couleur blanche lorsque la diode est éteinte). Configurez le champ **Facing** de chaque diode en fonction de sa position dans la roue (East, West, North ou South).

Pour simuler le fonctionnement de la roue, on utilisera cette fois-ci le mode de simulation automatique qui va faire avancer l'horloge automatiquement avec une fréquence donnée.

- Allez dans le menu **Simulate** et cliquez sur **Tick Frequency**. Choisir 32 Hz
- Allez de nouveau dans le menu **Simulate** et cliquez cette fois-ci sur **Ticks Enabled** (ou CTRL-K) pour lancer la simulation. La roue devrait se mettre à tourner
- Allez dans le menu Simulate et testez d'autres fréquences de simulation
- Observez le fonctionnement du compteur et le fonctionnement du décodeur
- Pour arrêter la simulation appuyez sur CTRL-K

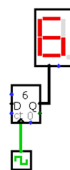
6. Horloge numérique

On se propose de réaliser un circuit permettant d'afficher une horloge numérique. Pour cela, nous aurons besoin des composants suivants :

- 1 horloge (Clock : disponible dans le dossier Wiring)
- 6 afficheurs de type Hex Digit Display (disponibles dans le dossier Input/Output)
- 6 compteurs (Counter : disponibles dans le dossier Memory)
- 5 inverseurs (NOT Gate : disponibles dans le dossier Gates)
- 1 porte ET (AND Gate : disponible dans le dossier Gates)
- 2 Splitters (disponibles dans le dossier Wiring)

Etape 1 : Conception du chiffre des unités des secondes

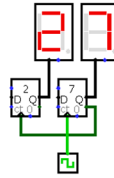
- Prenez un composant compteur et configurez son nombre de bits de données (Data Bits) à 4. Ceci permet de disposer d'un compteur qui va compter de 0 à 15.
- Toutefois nous allons nous servir de ce compteur pour contrôler seulement le chiffre des unités des secondes (nous nous occuperons du chiffre des dizaines par la suite). Par conséquent, il nous faut un compteur qui compte seulement de 0 à 9. Pour cela configurez sa valeur maximale (Maximum value) à 9. Si la valeur ne veut pas changer, désélectionnez le compteur puis resélectionnez le (bug d'affichage).
- Maintenant nous voulons que ce compteur repasse à 0 à chaque fois qu'il atteint 9. Pour cela configurez « Action On Overflow » sur « Wrap around ».
- Enfin configurez « Trigger » sur « Rising Edge ». Ceci permettra au compteur de s'incrémenter chaque fois qu'il recevra un front montant (passage de 0 à 1) sur son entrée « ct ».
- Reliez la sortie de l'horloge à l'entrée « ct » du compteur. Ainsi chaque fois que l'horloge passera de 0 à 1, le compteur sera incrémenté.
- Prenez un des afficheurs Hex Digit Display et reliez la sortie « Q » du compteur à l'entrée en bas à gauche de l'afficheur. Configurez « Background » de l'afficheur Hex sur « ffffff00 »
- Allez dans le menu « Simulate » et configurez « Tick Frequency » sur 2 Hertz. Ceci permettra à l'horloge de changer de valeur 2 fois par seconde (0→1 puis 1→0). Par conséquent elle produira un front montant (0→1) toutes les secondes. Ainsi notre afficheur s'incrémentera toutes les secondes.
- Appuyez sur CTRL-K pour activer la simulation et vérifiez que le chiffre des unités des secondes fonctionne correctement. Appuyez de nouveau sur CTRL-K pour arrêter la simulation.



Etape 2 : Conception du chiffre des dizaines des secondes

- Prenez un deuxième composant compteur et configurez de la même façon que le premier compteur sauf que cette fois-ci vous fixerez sa valeur maximale à 5 au lieu de 9. En effet les secondes vont de 00 à 59 donc la valeur maximale du chiffre des dizaines est 5. **Remarque** : vous pouvez faire un copier-coller du premier compteur et changer juste la valeur maximale.
- De même, faites un copier-coller de l'afficheur Hex et reliez la sortie Q du deuxième compteur à l'entrée en bas à gauche du deuxième afficheur Hex.
- Maintenant nous voulons que ce deuxième compteur s'incrémente chaque fois que le premier compteur repasse par 0. Or il se trouve que la sortie du compteur située juste sous la sortie « Q » nous permet de

savoir quand le compteur atteint sa valeur maximale. Ainsi, chaque fois que le premier compteur atteint la valeur 9, cette sortie passe à 1. Puis quand le compteur repasse à 0 elle revient à 0 et reste à 0 jusqu'à ce que le compteur atteigne de nouveau la valeur 9. Nous allons donc nous servir de cette sortie du premier compteur pour déclencher l'incréméntation du deuxième compteur. Pour cela reliez la sortie située juste en dessous de la sortie « Q » du premier compteur à l'entrée « ct » du deuxième compteur.



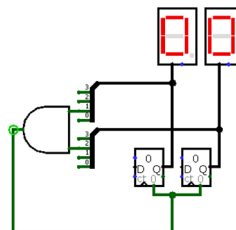
- Appuyez sur CTRL-K pour activer la simulation et observez bien le déroulement des secondes. Pourquoi cela ne fonctionne-t-il pas correctement ? Proposez une solution pour résoudre le problème (**pensez à utiliser un inverseur...**).
- Après avoir corrigé le problème, activez de nouveau la simulation et vérifiez que le déroulement des secondes s'effectue correctement.

Etape 3 : Conception de l'affichage des minutes

- En vous inspirant de ce que vous venez de faire pour les secondes, réalisez l'affichage des minutes puis lancez une simulation pour vérifier le bon fonctionnement des minutes. Vous pouvez augmenter « Tick Frequency » à 128 pour ne pas avoir à attendre longtemps pendant la simulation. **Remarque** : vous pouvez faire un copier-coller de la partie des secondes puis modifier juste ce qu'il faut.

Etape 4 : Conception de l'affichage des heures

- En vous inspirant de ce que vous venez de faire pour les minutes, réalisez l'affichage des heures. Attention toutefois, pour l'affichage des heures certaines adaptations s'imposent. En effet l'affichage des heures doit revenir à 00 chaque fois qu'il atteint 24. Autrement dit dès que les 2 compteurs qui contrôlent l'affichage des heures atteignent 24, ces compteurs doivent repasser à 0, si bien qu'on n'aura même pas le temps de voir le 24 s'afficher.
- Pour cela vous allez avoir besoin de réinitialiser ces 2 compteurs à 0, chaque fois que le compteur des dizaines des heures atteint 2 **ET** le compteur des unités des heures atteint 4.
- Comme $(2)_{10} = (0010)_2$ et $(4)_{10} = (0100)_2$, vous allez vous servir d'un premier composant splitter pour récupérer le bit de rang 1 (2^{ème} bit en partant de la droite car les rangs commencent à 0) du chiffre des dizaines. Vous vous servirez aussi d'un deuxième composant splitter pour récupérer le bit de rang 2 (3^{ème} bit en partant de la droite) du chiffre des unités des heures.
- Enfin vous relierez ces 2 bits aux deux entrées d'une porte ET et vous vous servirez de la sortie de la porte ET pour réinitialiser ces 2 compteurs à 0 en la reliant aux entrées 0 (en bas à droite) des deux compteurs.











7. Contrôle de feux de circulation

On se propose de réaliser un circuit permettant de contrôler les feux de circulation d'une intersection. On suppose que cette intersection est contrôlée par 2 feux tricolores FEU₁ et FEU₂. FEU₁ est composé de 3 ampoules : **R**₁(Rouge), **O**₁(Orange) et **V**₁(Vert). De même FEU₂ est composé de 3 ampoules : **R**₂(Rouge), **O**₂(Orange) et **V**₂(Vert).

On considère une échelle de temps composée de 4 instants : T₀ à T₃. Pour simuler ces différents instants, nous allons utiliser le composant **Counter** (compteur) de LOGISIM disponible sous le dossier **Memory**. Ce composant permet de produire une séquence répétitive de valeurs successives. Réglez son attribut « **Data Bits** » sur 2 de manière à ce qu'il produise la suite de valeur 0,1,2,3,0,1,2,3,0,1...

Connectez ce compteur à une horloge de façon à ce que ce compteur soit incrémenté à chaque cycle d'horloge.

Le tableau suivant indique l'état que doit avoir chaque ampoule à chaque instant, et donc à chaque valeur du compteur (désignons par **b₁b₀** les 2 bits de ce compteur) :

Instant	b ₁	b ₀	R ₁	O ₁	V ₁	R ₂	O ₂	V ₂	FEU ₁	FEU ₂
T ₀	0	0	0	0	1	1	0	0		
T ₁	0	1	0	1	0	1	0	0		
T ₂	1	0	1	0	0	0	0	1		
T ₃	1	1	1	0	0	0	1	0		

D'après cette table, nous voyons que b₁ et b₀ constituent les entrées de notre circuit alors que R₁, O₁, V₁, R₂, O₂, V₂ constituent les sorties du circuit. En utilisant la table de vérité ci-dessus, trouvez les expressions de R₁, O₁, V₁, R₂, O₂ et V₂ en fonction de b₁ et b₀.

R ₁ =	O ₁ =	V ₁ =	R ₂ =	O ₂ =	V ₂ =
------------------	------------------	------------------	------------------	------------------	------------------

Dessinez le circuit. On utilisera les composants **LED** (sous le dossier **Input/output**) de LOGISIM pour représenter les ampoules des deux feux (on les configurera avec les bonnes couleurs). On utilisera également le composant **Splitter** (sous le dossier **Wiring**) pour décomposer la sortie du compteur en 2 bits distincts (b₁, b₀). Enfin on utilisera une horloge reliée à l'entrée ct du compteur pour simuler l'avancement du temps.

Testez votre circuit et vérifiez qu'il fonctionne correctement.

8. BONUS 1 : Amélioration du contrôle de feux de circulation

On se propose de réaliser un circuit permettant de contrôler les feux de circulation d'une intersection un peu plus réaliste que le précédent car le laps de temps des feux rouge et vert est normalement plus long que celui de l'orange. On considère cette fois-ci une échelle de temps composée de 8 instants : T_0 à T_7 . Pour simuler ces différents instants, nous allons ici aussi utiliser le composant **Counter** (compteur) de LOGISIM disponible sous le dossier **Memory**. Réglez cette fois-ci son attribut « **Data Bits** » sur 3 de manière à ce qu'il produise la suite de valeur 0,1,2,3,4,5,6,7,0,1,2,3,4,5,6,7,0,1...

Connectez ce compteur à une horloge de façon à ce que ce compteur soit incrémenté à chaque cycle d'horloge.

Le tableau suivant indique l'état que doit avoir chaque ampoule à chaque instant, et donc à chaque valeur du compteur (désignons par $b_2b_1b_0$ les 3 bits de ce compteur) :

Instant	b_2	b_1	b_0	R_1	O_1	V_1	R_2	O_2	V_2	FEU ₁	FEU ₂
T_0	0	0	0	0	0	1	1	0	0		
T_1	0	0	1	0	0	1	1	0	0		
T_2	0	1	0	0	0	1	1	0	0		
T_3	0	1	1	0	1	0	1	0	0		
T_4	1	0	0	1	0	0	0	0	1		
T_5	1	0	1	1	0	0	0	0	1		
T_6	1	1	0	1	0	0	0	0	1		
T_7	1	1	1	1	0	0	0	1	0		

D'après cette table, nous voyons que b_2 , b_1 et b_0 constituent les entrées de notre circuit alors que R_1 , O_1 , V_1 , R_2 , O_2 , V_2 constituent les sorties du circuit. En utilisant cette table, vérifiez les expressions suivantes de O_1 , V_2 , et V_1 en fonction de b_2 , b_1 et b_0 :

$O_1 = b_0 \text{ et } b_1 \text{ et (non } b_2)$	$V_2 = (\text{non } (b_0 \text{ et } b_1)) \text{ et } b_2$	$V_1 = (\text{non } (b_0 \text{ et } b_1)) \text{ et (non } b_2)$
---	---	---

Trouvez, vous-mêmes, directement d'après la table, les expressions de R_1 , R_2 et O_2 en fonction de b_2 , b_1 et b_0 .

$R_1 =$	$R_2 =$	$O_2 =$
---------	---------	---------

Dessinez le circuit. On utilisera les composants **LED** (sous le dossier **Input/output**) de LOGISIM pour représenter les ampoules des deux feux (on les configurera avec les bonnes couleurs). On utilisera également le composant **Splitter** (sous le dossier **Wiring**) pour décomposer la sortie du compteur en 3 bits distincts (b_2 , b_1 , b_0). Enfin on utilisera une horloge reliée à l'entrée ct du compteur pour simuler l'avancement du temps.

Testez votre circuit et vérifiez qu'il fonctionne correctement.

9. BONUS 2 : Autre version utilisant un décodeur

On se propose de réaliser un circuit équivalent au circuit précédent mais en utilisant cette fois-ci :

- 1 décodeur 3x8 : composant **Decoder** dans le dossier **Plexers** de LOGISIM. Configurer son champ **Select Bits** sur 3
- 2 portes OR à 4 entrées
- 2 portes OR à 3 entrées
- Un compteur 3 bits comme précédemment
- 1 horloge

On remarque que si on relie la sortie du compteur à l'entrée du décodeur, alors les sorties du décodeur vont s'activer à tour de rôle 0,1,2,3,4,5,6,7,0,1,...

On remarque d'après la table de vérité précédente que :

- V_1 s'allume aux instants T_0 , T_1 et T_2 , c'est-à-dire lorsque la sortie 0 ou 1 ou 2 du décodeur est à 1
- O_1 s'allume à l'instant T_3 , c'est-à-dire lorsque la sortie 3 du décodeur est à 1
- R_1 s'allume aux instants T_4 à T_7 , c'est-à-dire lorsque la sortie 4 ou 5 ou 6 ou 7 du décodeur est à 1
- V_2 s'allume aux instants T_4 , T_5 et T_6 , c'est-à-dire lorsque la sortie 4 ou 5 ou 6 du décodeur est à 1
- O_2 s'allume à l'instant T_7 , c'est-à-dire lorsque la sortie 7 du décodeur est à 1
- R_2 s'allume aux instants T_0 à T_3 , c'est-à-dire lorsque la sortie 0 ou 1 ou 2 ou 3 du décodeur est à 1

Réalisez le circuit en utilisant uniquement les composants indiqués et testez.