

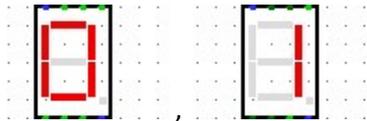
Exercice:

Affichage de 0 ou 1

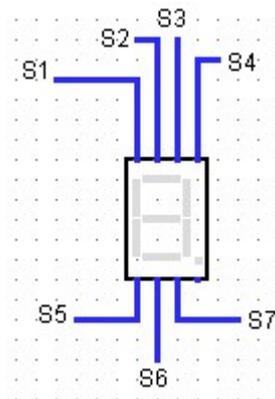
outils:

- 7-segments display
- pin (entrée 0 ou 1)
- Des portes logiques

Objectif faire afficher un 0 si on rentre 0 et 1 si on rentre 1 comme sur cette exemple:



- 1) Placer un 7-segments display et un pin et tester le 7-segment pour connaître le rôle de chaque entrée du 7-segment.



- 2) Remplir la table de vérité suivante et Ecrire les fonctions booléennes pour chaque entrée du 7-segment pour répondre à l'objectif

x	S1	S2	S3	S4	S5	S6	S7
0							
1							

- 3) Créer le circuit pour répondre à l'objectif

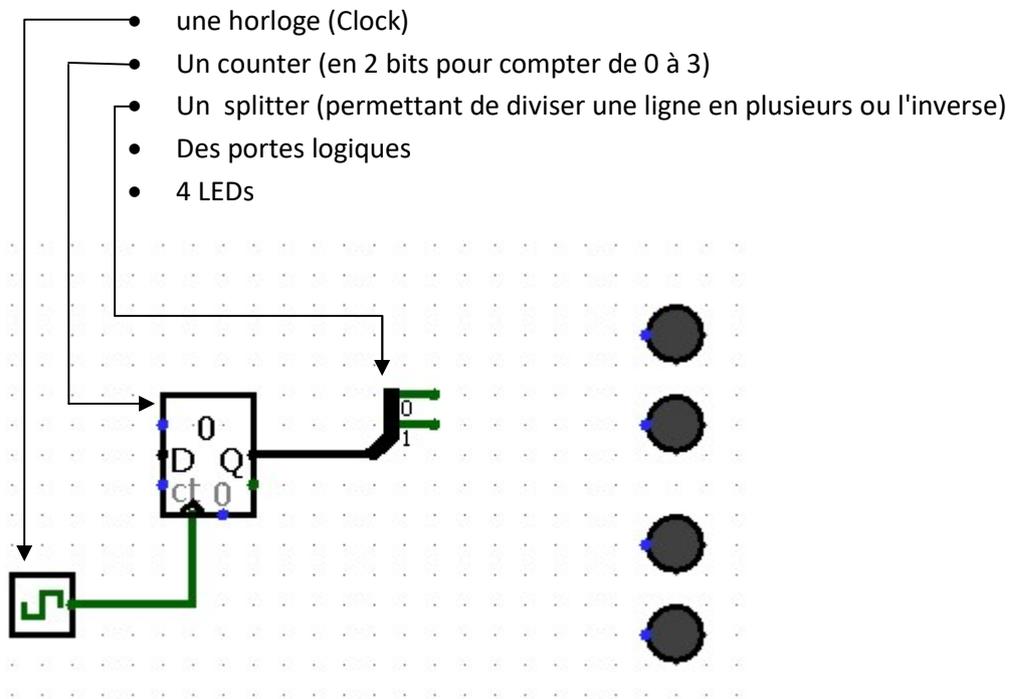
MINI PROJETS

Pour chaque mini projet vous devez rendre les fonctions de sortie (simplifiées si possible) et le circuit

Mini projet 1: logisim_Projet1_DecorationNoel_NomEleve

Décoration de Noël

Matériels:



Objectif:

1) Dans une première étape faire allumer successivement 4 leds commandées par une horloge. (Il ne doit y avoir qu'une seule led allumée à la fois)

a) Remplir la table de vérité de S1, S2, S3 et S4 représentant chaque led pour répondre au problème

x	Y	S1	S2	S3	S4
0	0				
0	1				
1	0				
1	1				

b) Ecrire les expressions booléennes de S1, S2, S3 et S4 puis faire le circuit correspondant.

2) Choisir un allumage qui allume 2 leds en même temps puis une 3ème puis la 4ème puis les éteints toutes

3) Complément Facultatif :

Si jamais vous êtes en avance par rapport aux autres vous pouvez faire un circuit pour changer de guirlandes ou plus subtilement d'allumage.

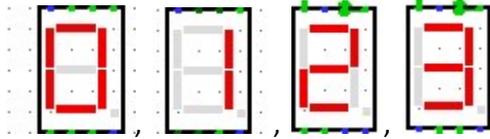
Mini projet 2: logisim_Projet2_Affichage_NomEleve

Affichage digitale (0 , 1, 2 ou 3)

outils:

- 7-segments display
- pin (entrée 0 ou 1)
- Des portes logiques

Objectif faire afficher un 0 si on rentre 0 et 1 si on rentre 1 , etc ... comme sur cette exemple:



- 1) Placer un 7-segments display et deux pins (ou une clock 2 bit) et tester le 7-segment
- 2) Remplir la table de vérité suivante et Ecrire les fonctions booléennes pour chaque entrée du 7-segment pour répondre à l'objectif

x	Y	S1	S2	S3	S4	S 5	S6	S7
0	0							
0	1							
1	0							
1	1							

- 3) Créer le circuit pour répondre à l'objectif

Mini projet 3: TPlogisim_Projet3_Comparateur_NomEleve (mini-projet)

Comparateur

Objectif:

- 1) Dans une première étape faire un comparateur binaire. Il y a deux entrées a et b representant deux nombres d'un seul bit (a et b ne peuvent prendre que les valeurs 0 ou 1). Il y a trois sorties(des leds par exemple) :une qui s'allume si $a=b$, une pour $a < b$ et une pour $a > b$
- 2) Ensuite vous ferez un comparateur entre un nombre de deux bits et un nombre de un bit

Matériels:

- Des entrée un ou deux bits
- Des portes logiques
- 3 LEDs

Mini projet 4: logisim_Projet4_PetitExosl_NomEleve (mini-projet)

Quelques exercices

Exercice 1

Dessinez un circuit à trois entrées et une sortie réalisant la fonction majorité (la sortie est à 0 si et seulement si le nombre d'entrées étant à 0 est majoritaire). Testez-le.

Exercice 2

Dessinez un circuit à deux entrées et quatre sorties réalisant la fonction carré (d'un nombre sur deux bits). Testez-le.

Matériels:

- Des entrée un ou deux bits
- Des portes logiques
- 3 LEDs

Mini projet 5: TPlogisime_Projet4_Des_NomEleve (mini-projet)

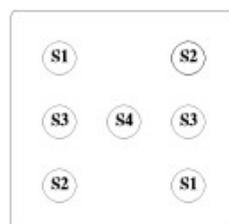
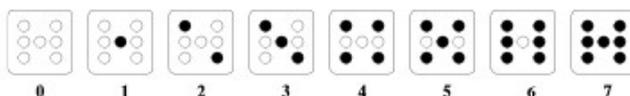
Les dés entre guillemets

On veut construire un circuit logique destiné à afficher un entier compris entre 0 et 7 à la manière d'un numéro tel qu'il apparaît sur un dé à jouer (à 8 faces). La figure ci-dessous représente les états de chacune des sept leds en fonction de chaque entier.

On peut remarquer que la led supérieure gauche est toujours dans le même état que la led inférieure droite (toutes deux sont éteintes pour 0 et 1 mais sont allumées pour 2, 3, 4, 5, 6 et 7).

Notre circuit n'a donc besoin que d'une seule sortie s_1 qui sera reliée à ces deux leds. Pour les mêmes raisons, on n'a besoin que d'une sortie s_2 pour les leds supérieure droite et inférieure gauche, ainsi que d'une sortie s_3 pour les leds de centre gauche et de centre droit. Enfin, une sortie s_4 sera reliée à led centrale.

Notre circuit a trois entrées e_2, e_1 et e_0 qui codent l'entier à afficher sous forme binaire, e_2 étant le bit de poids le plus fort. Par exemple, pour coder l'entier 6, on a $e_2=1$, $e_1=1$ et $e_0=0$



- 1) Ecrivez la table de vérité (à 3 entrées et 4 sorties) des fonctions booléennes d'affichage d'un entier tel que décrit ci-dessus.
- 2) Déterminez les fonctions logiques de chaque bit de sortie si.
- 3) Dessinez le circuit correspondant en reliant les 4 sorties à sept leds comme décrit ci-dessus

Mini projet 6: logisim_Projet6_Additionneur_NomEleve (mini-projet)

- 1) Construire un circuit additionneur 1 bit qui prend en entrée deux nombres de 1 bit et donne en sortie leur somme sur deux bits (deux sorties 1 bit)
- 2) Construire un circuit additionneur 2 bit qui prend en entrée deux nombres de 2 bit et donne en sortie leur somme sur 3 bits donc 4 entrées 1 bits et 3 sorties 1 bit
- 3) Si vous avez le temps faites un soustracteur 2 bits, tenir compte du complément à 2

Faire une fiche projet

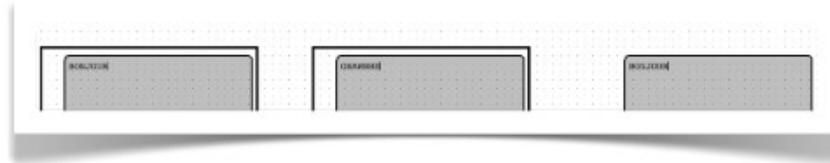
Description précise du projet, description des outils, table de vérité, simplification, circuit et commentaires.

- Des entrée un et deux, 3 bits
- splitter
- Des portes logiques

Ave César

Objectif

Obtenir un circuit permettant de chiffrer un texte. Le chiffre à utiliser est celui appelé « chiffre de César » ou « rot13 » chez les informaticiens. Son principe est très élémentaire puisqu'il s'agit de la simple permutation (A N) (B O) ... (M Z). Ce chiffre fait partie de la famille des chiffres à décalage qui n'offrent aucune résistance cryptographique, mais sont simplement mentionnés pour des raisons historiques ou pédagogiques. Rot13 n'est qu'une des variantes du chiffre de César qui est en réalité paramétré par le décalage de l'alphabet (donc 25 chiffres de César possible). La particularité de rot13 est qu'il est involutif. On cherchera donc à vérifier cette propriété par exemple en chiffrant puis chiffrant à nouveau et à la suite le texte d'entrée comme dans la figure suivante :



Éléments utiles

Les éléments pouvant servir sont :

- keyboard (permettant de saisir le texte à chiffrer)
- tty (pour affiche le résultat)
- portes arithmétiques diverses
- constantes
- bit extender (pour convertir les données 7 bits en 8 bits et vice-versa)

Réalisation

La difficulté ici est essentiellement de bien comprendre comment l'arithmétique est réalisée. Diverses constructions peuvent être obtenues :

1. dans un premier temps, à obtenir un circuit convertissant une donnée 7 bits en une donnée 8 bits équivalente (sous réserve que l'entrée est un caractère alphabétique majuscule)
2. modifiez le circuit de sorte que les caractères autres que les lettres majuscules soient conservés à l'identique (via des opérateurs de comparaison, etc)
3. modifiez le circuit de sorte que le chiffre s'applique sur les lettres alphabétiques majuscules et minuscules
4. faites du circuit de chiffrement un sous-circuit en rajoutant un autre circuit (Project ➤ Add circuit) et en faisant du nouveau le circuit principal (sélectionnez-le et faites apparaître le menu contextuel Set As Main Circuit), puis ajoutez keyboard et tty(s) pour saisir un texte et voir son chiffrement et déchiffrement opérer (via une horloge manuelle ou non).

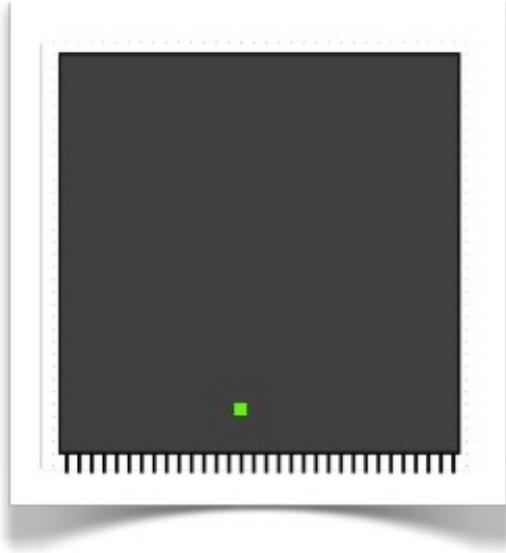
Plus compliqué mais dans le même esprit mais aussi beaucoup plus intéressant:

Un projet pourrait être de réaliser un circuit réalisant le chiffrement DES (Data Encryption Standard). Cet algorithme de chiffrement est utilisé pour chiffrer les mots de passe de presque tous les systèmes d'exploitation depuis plus de 30 ans. Attention : c'est assez difficile à réaliser mais intéressant car pour obtenir un circuit raisonnable, il faut faire « tourner » les données à l'intérieur du circuit.

Contrôleur de pixel

Objectif

Obtenir le déplacement d'un pixel à l'écran via un joystick. Le cadencement des mises à jour s'effectuera à l'aide d'une horloge. L'écran pourrait ressembler à



Une vidéo de l'effet final est consultable ici : <http://youtu.be/YYtBWIpfeIQ>

Éléments utiles

Les éléments pouvant servir sont :

- splitter (permettant de combiner n-lines parallèles 0/1 en une seule ou l'inverse)
- joystick (2 bits)
- led matrix (32x32)
- clock
- counter (2 pour représenter les « coordonnées »)
- decoder

Réalisation

La difficulté ici est essentiellement de bien comprendre comment les données peuvent être sérialisées/parallélisées/projetées, etc

1. dans un premier temps, chercher à obtenir le déplacement « vertical »
 2. dans un second temps, le déplacement « horizontal »
-

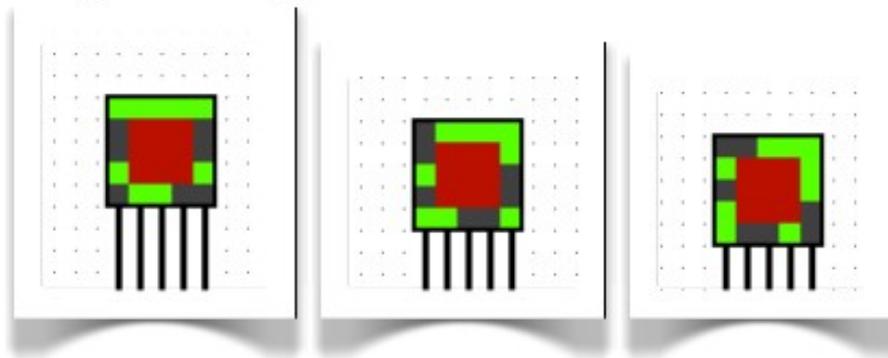
Le chenillard

Objectif

Obtenir un chenillard.

1. son entrée sera un mot de 16 bits représentés en sortie par une boucle de pixel allumés ou éteints
2. piloté par une horloge (automatique ou manuelle), à chaque top les pixels « tourneront » (dans un sens choisi à l'avance)

Une séquence de trois tops pourrait donner l'effet suivant :



Éléments utiles

Bien que tout autre élément puisse être considéré, une réalisation assez simple utilise les éléments suivants :

- splitter (permettant de combiner n-lines parallèles 0/1 en une seule ou l'inverse)
- shift register (un registre à décalage)
- register (un registre de 16 bits pour l'entrée)
- led matrix (5x5)
- clock

Réalisation

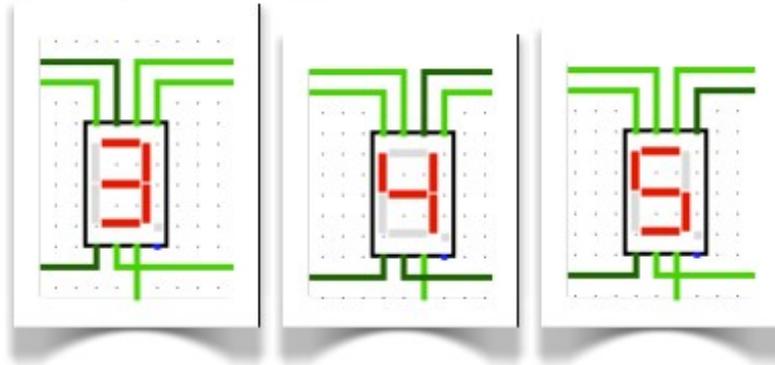
1. dans un premier temps obtenir un circuit permettant de faire tourner le chenillard dans un sens fixé à l'avance
2. dans un second temps, permettre de sélectionner le sens de rotation du chenillard dans les deux sens (en utilisant le pilotage du décaleur — facile)
3. dans un troisième temps, implémenter un troisième circuit de chenillard qui ne produise pas une simple « boucle » (mais un effet graphique un peu plus évolué, type parcours selon une « courbe de Peano », attention à obtenir un parcours de longueur 16), et qui permette de sélectionner l'un de ces trois scénarios à volonté

L'afficheur numérique

Objectif

Obtenir un afficheur de chiffre décimal.

1. son entrée sera un compteur de 4 bit qui sera interprété comme une valeur décimale entre 0 et 9 (compris)
2. piloté par une horloge (automatique ou manuelle), à chaque top on incrémente la valeur du compteur, celle-ci étant immédiatement affichée
3. Une séquence de trois tops pourrait donner l'effet suivant :



Éléments utiles

De nombreux autres éléments peuvent être employés, mais une réalisation simple utilise :

- counter
- register (un registre de 4 bits)
- splitter (permettant de combiner n-lignes o/1 parallèles en une seule ou l'inverse)
- 7-segment display (un afficheur 7 segments)
- des portes logiques élémentaires diverses et variées
- clock

Réalisation

1. dans un premier la réalisation peut-être la plus simple possible (i.e. utilisant un systématisme sans défaut pour synthétiser le circuit)
2. dans un second temps, il peut être intéressant de travailler à la simplification du circuit en diminuant le nombre de portes

Pour que le projet soit intéressant, on pourra faire un compteur plus évolué sur des nombres plus grand ou même une horloge.