

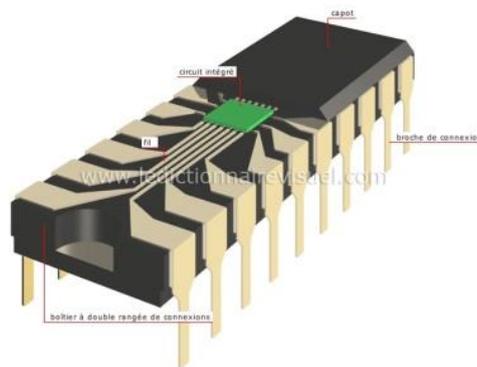


TRAVAUX PRATIQUES Electronique Numérique (M33)

SMP6 –Parcours 1

Pr. Aziz Amari

Année universitaire 2019-2020



N.B. Un travail bien préparé permet une meilleure compréhension du $J\mathcal{P}$ et également une meilleure rédaction. Un compte rendu de travaux pratiques n'est pas seulement une suite de résultats mais comprend aussi des commentaires et des conclusions.

Plan

Quelques Conseils	2
Fiche de Connaissances	3
TP N°1 : Câblage des Portes Logiques	5
TP N°2 : Portes Logiques en mode interactif sous Logisim.....	9
TP N°3 : Additionneur et Soustracteur.....	15
TP N°4 : Comparateur Binaire.....	22
TP N°5 : Montages à Bascules	26

Quelques Conseils ...

Préparation du TP

La préparation du TP est très importante.

- Vous devez, avant de venir en séance de TP, avoir préparé votre travail.
- Une séance de TP n'est pas une séance presse bouton, ni un empilement de courbes sans légende ni commentaire.
- De plus, la préparation est individuelle et ne doit pas se faire en prenant exemple sur le travail d'un collègue.

Mise au point du montage

Pour éviter les erreurs de câblage, il est très important de dessiner le montage en indiquant les **numéros de broches**.

Votre *comportement en TP*, c'est à dire votre *autonomie*, la façon dont vous résolvez les problèmes, votre *efficacité* est prise en compte dans la note.

Dans un TP il y a toujours :

- Une **approche théorique (AT)**
 - Rechercher les éléments de cours et les TD portant sur les mêmes notions.
 - Préparer par écrit vos équations, vos calculs et vos montages.
 - Ceci doit être fait chez vous
- Une **approche expérimentale (AE)**
 - Rechercher les éléments de cours et les TD portant sur les mêmes notions.
 - Relisez vos notes relatives à l'utilisation des composants, des matériels et des logiciels.
 - La partie **AE** ne peut être réalisée qu'en salle disposant du matériel.
 - Pour un travail en binôme, il faut répartir les tâches.
- Une **synthèse**
 - La synthèse ne peut être réalisée que sur place quand vous disposez :
 - De vos résultats théoriques,
 - De vos résultats expérimentaux.
 - C'est **LA PARTIE LA PLUS IMPORTANTE DU TP** pour laquelle il faut dégager du temps...

Compte rendu

Le compte-rendu doit comporter pour chaque TP :

- Une **partie théorique**,
- Des **schémas** avec des symboles normalisés et le brochage (**numéros de broches**),
- Une **analyse** des résultats obtenus.

Enfin, la **présentation en elle-même** de tout travail est importante.

Le contrôle

Le contrôle aura lieu à la fin des TP. Il dure **1h30** et sera individuel.

Brochage du CI numérique

I. Bases des CI

CI est l'abréviation de Circuit Intégré. Un CI est un assemblage de plusieurs transistors, FET, résistances, etc., dans un seul ensemble. Les avantages de l'utilisation du CI sont la simplification du câblage et la réduction de la taille des circuits.

Dans les circuits numériques, "1" représente généralement "élevé" tandis que "0" représente généralement "faible" ; la Figure A. 1 ci-dessous représente un signal numérique idéal. La production de masse a considérablement réduit les coûts de CI numériques grâce à des percées dans le traitement des semi-conducteurs.

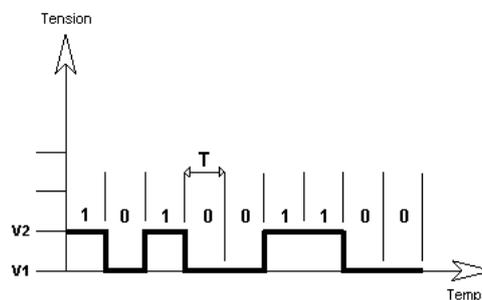


Figure A. 1. Représentation numérique d'une tension.

II. Broches des CI numériques

L'emballage du **CI** numérique fait appel au **Dual In line Package (DIP)** qui est un boîtier de CI qui le connecte au monde extérieur.

Le boîtier DIP est un boîtier dont les pattes sont numérotées dans un ordre précis à partir d'un repère ou d'un marquage sur le boîtier. Les pattes (ou broches) sont les connexions électriques vers le circuit intégré lui-même et sont espacées de 2,54mm. Les deux rangées de pattes sont espacées de 7,62mm.

Il faut tout d'abord voir le repère du boîtier **DIP** qui indique la broche 1. Ce repère peut avoir la forme d'un **petit rond creux** ou d'une **encoche** en forme de demi-cercle sur le côté du boîtier. A partir du repère, pour la numérotation, il faut tourner dans le sens inverse des aiguilles d'une montre quand on regarde le boîtier par le haut (Figure A. 2).

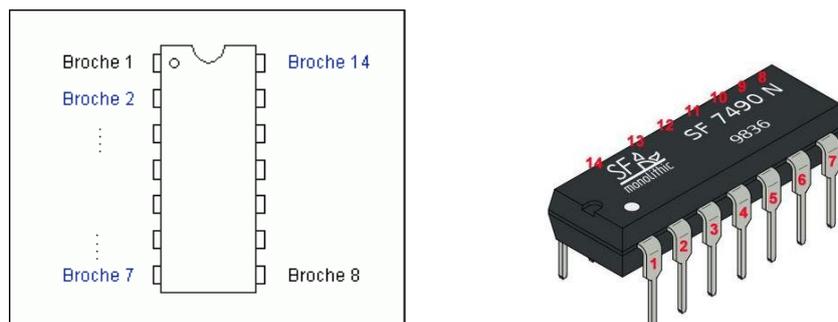


Figure A. 2. Exemple de boîtier DIP 14.

Brochage des LEDS

Une diode électroluminescente (abrégé en LED, de l'anglais : **L**ight-**E**mitting **D**iode est une diode qui émet de la lumière, ses caractéristiques sont proches de celles d'une diode de signal. Elle ne laisse passer le courant que dans un seul sens. Les LED doivent être alimentées via une résistance afin de limiter le courant qui les traverse.

Deux façons de vérifier la polarité des LEDs (Figure A. 3) :

- Les deux broches n'ont pas la même longueur : La broche la plus longue correspond à l'ANODE. C'est cette broche qui devra être connectée au '+'
- La partie 'lumineuse' de la LED est 'arrondie' d'un côté et 'plate' de l'autre. Le côté 'plat' correspond à la CATHODE. La CATHODE doit être connectée à une résistance afin de protéger la LED.

Il est **INDISPENSABLE** de bien respecter cette règle. Sinon, les LEDs ne s'allumeront pas !

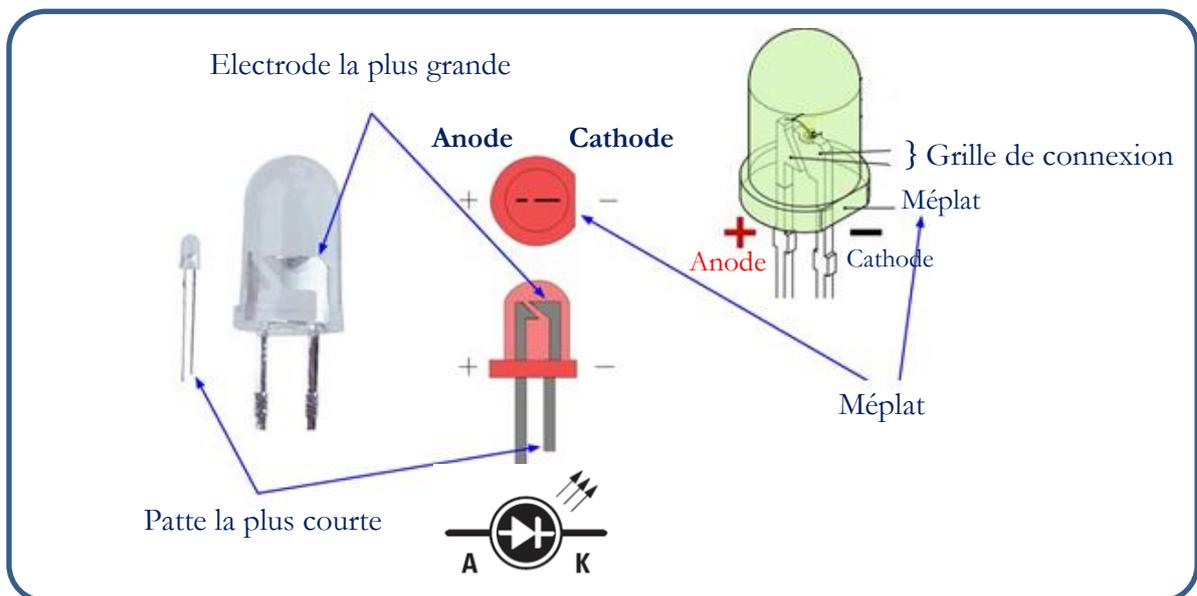


Figure A. 3. Brochage des LEDs (Light Emitting diode)

T.P. N° 1
Câblage des Portes Logiques

T.P. N° 1
Câblage des Portes Logiques
SMP6 - P1
Module d'Electronique Numérique

Objectif de ce T.P.

- Identifier, utiliser, et brancher un circuit intégré logique.
- Réalisation pratique d'un montage logique.
- Réaliser un logigramme réel en utilisant des portes logiques électroniques.

Préparation

Lisez attentivement le sujet pour chaque question :

- Déterminer les résultats que vous pensez obtenir lors de la séance.
- Préparer les schémas de câblage en indiquant les **numéros des broches** sur le schéma.

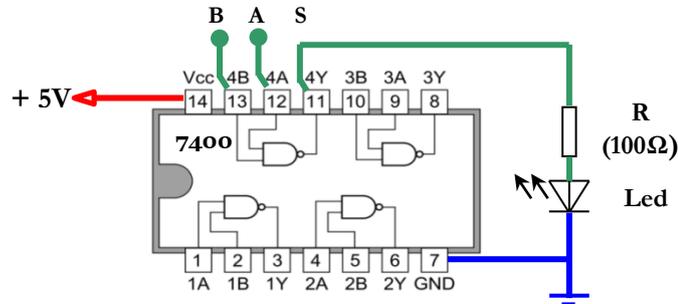
Avant de commencer un-câblage de circuits logiques il est nécessaire de faire un **schéma clair et complet** des circuits logiques physiques et des liaisons qui interviennent dans la fonction à réaliser.

Il faut donc y indiquer les **numéros des broches** de connexion des différents **CI** (Circuits Intégrés) et ne pas oublier les broches de **masse** et **d'alimentation** des **CI**.

I. Portes élémentaires

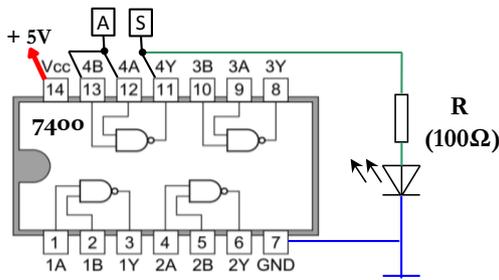
Afin de se familiariser avec le Circuit Intégré **74LS00** (voir brochage en page 8), on veut réaliser quelques opérateurs de base à l'aide de ce circuit **seul**.

1. Afin de se familiariser avec la **plaque d'essai**, réaliser et tester ce montage :



2. Réaliser les montages suivants, remplir leur **TV** expérimentalement¹ puis déduire leurs fonctions.

1. $S=f(A)$



TV

A	S

Fonction réalisée :

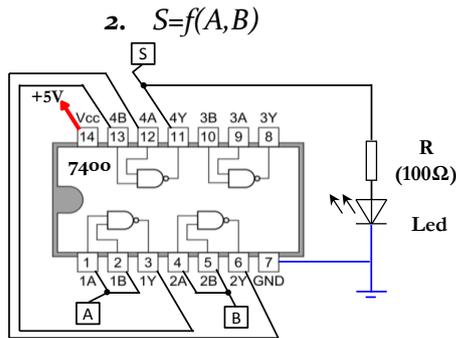
.....
.....

¹ **Remarque : Alimentation des circuits**

Les états, ou niveaux, sont caractérisés par des valeurs de tensions dont les limites sont précisées.

ÉTAT 1 : Niveau Haut (**H**igh), présence de tension : **2,4 à 5 V** pour la technologie TTL ;

ÉTAT 0 : Niveau Bas (**L**ow), absence de tension : **0 à 0,8 V** pour la technologie TTL.

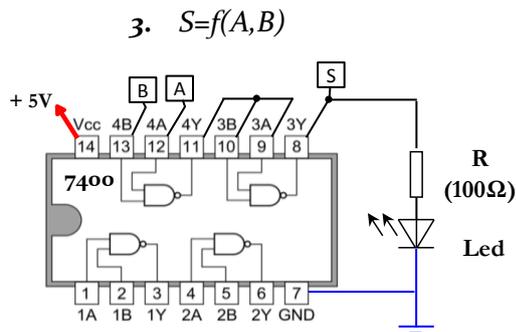


TV

A	B	S

Fonction réalisée :

.....
.....



TV

A	B	S

Fonction réalisée :

.....
.....

3. Déduire les expressions de ces fonctions.

II. Opérateur OU EXCLUSIF

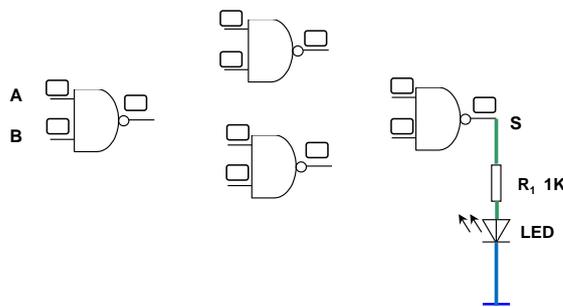
II.1. Définition :

Le OU EXCLUSIF (XOR) est une fonction logique qui **compare** deux bits de données. Elle vaut "1" si les bits sont différents et "0" si non.

II.2. Opérateur Ou exclusif à deux entrées :

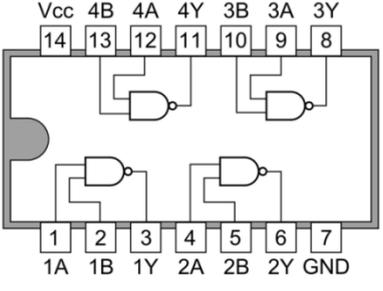
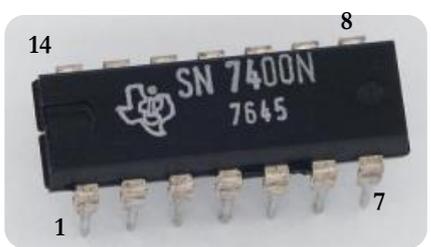
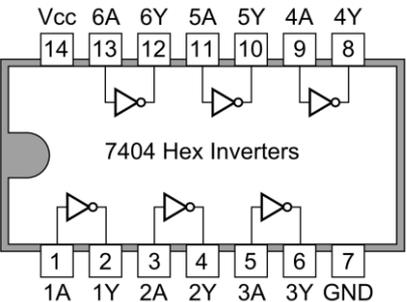
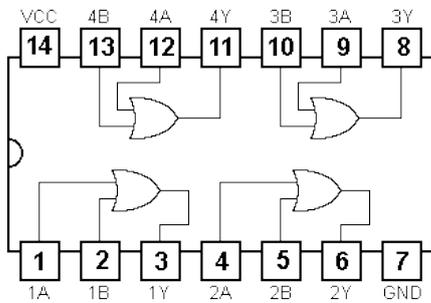
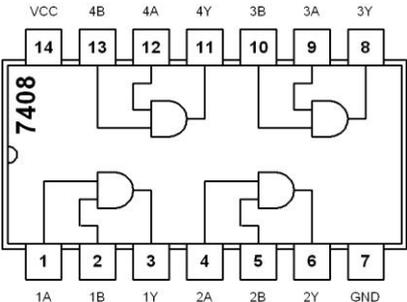
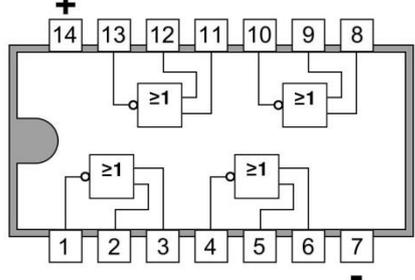
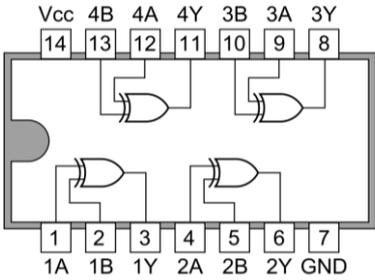
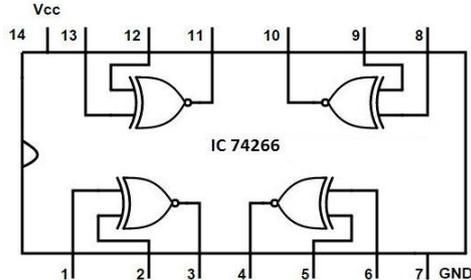
Symbole US	Symbole UE	Notation	Table de vérité		
			A	B	S
		$S = A \oplus B$ (lire A ou exclusif B)	0	0	
			0	1	
			1	0	
			1	1	

1. Compléter la Table de vérité de l'opérateur XOR ci-dessus.
2. Etablir ses équations selon la 1^{ère} puis la 2^{ème} forme canonique ?
3. Transformer ces équations afin de réaliser un OU exclusif en utilisant un seul CI 74LS00.
4. Compléter le logigramme de la fonction XOR suivant, en indiquant les numéros des broches.



5. Câbler le circuit et procéder aux essais après vérification du professeur. Prenez soin de relier d'abord les broches d'alimentation et de masse du CI 74LS00.
6. Relever la **table de vérité** du circuit réalisé.

BROCHAGES des Circuits Intégrés

NAND : 74LS00	C.I : 74LS00
	
NOT : 74LS04	OR : 74LS32
	
AND : 74LS08	NOR : 74LS02
	
XOR : 74LS86	XNOR : 74LS266
	

T.P. N° 2
Portes Logiques en mode
interactif sous Logisim

T.P. N° 2

Portes Logiques en mode interactif sous Logisim

SMP6 -P1

Module d'Electronique Numérique

OBJECTIF :

- Maitriser le logiciel **Logisim**.
- Créer et valider des logigrammes d'une complexité quelconque.
- Simulation sous le logiciel **Logisim** pour vérifier le fonctionnement des circuits.

I. PRISE EN MAIN DES OUTILS DE SIMULATION SOUS 'LOGISIM'

Présentation

Logisim est un logiciel **open-source** permettant de concevoir et de simuler des circuits logiques. Une des avantages de ce logiciel est de pouvoir **éditer** et **simuler** un circuit en même temps. La dernière version de **Logisim** peut être téléchargée à l'adresse : <http://reds-data.heig-vd.ch/logisim-evolution/logisim-evolution.t.jar>

Lorsque vous démarrez Logisim, vous verrez une fenêtre semblable à l'illustration suivante (Fig. 2. 1 Erreur ! Source du renvoi introuvable.).

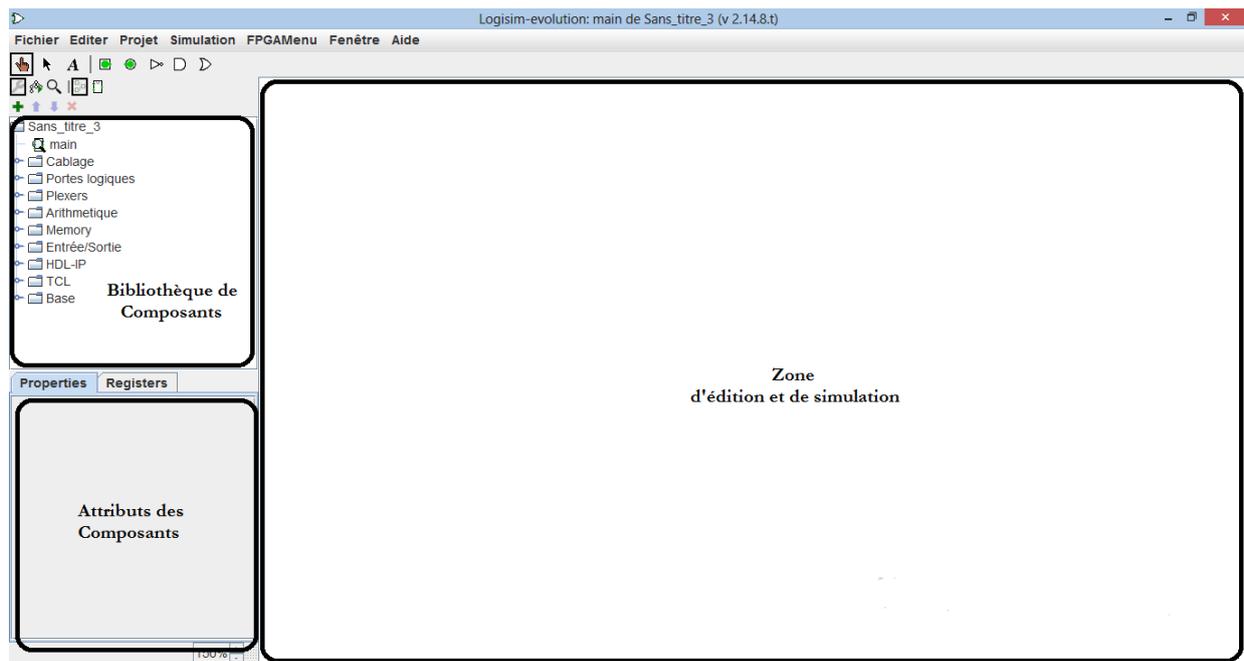


Fig. 2. 1. Interface de Logisim

Le logiciel se présente comme une table de manipulation. L'espace de Logisim est divisé en **trois parties** appelés le **panneau de navigation**, la **table des attributs**, et la **zone de travail** (ou zone d'édition et de simulation). Au-dessus de ces trois parties se trouvent la **barre de menu** et la **barre d'outils**.

A **l'ouverture de Logisim**, une fenêtre demande l'introduction du *nom d'utilisateur* (voir Fig. 2. 2). Lors de la première utilisation, il faut ajouter un utilisateur comme en Fig. 2. 3:

1. Cliquer sur *Change user*.
2. Introduire votre prénom et votre nom, séparé par un **_**, dans la case *Add new user*.
3. Cliquer sur *Add*.
5. Cliquer sur *Accept conditions*.



Fig. 2. 2. Fenêtre de Login



Fig. 2. 3. Ajout d'un utilisateur de Logisim

La liste des *utilisateurs* est enregistrée par la suite sur la machine. Pour les utilisations ultérieures, il suffit de cliquer sur *Accept conditions* pour sélectionner l'utilisateur actif ou de le sélectionner dans la liste.

II. CONSTRUCTION DU SCHEMA

A. Mode édition :

1. Pour utiliser le **mode édition**, il faut simplement sélectionner la *flèche* :  comme indiqué en **Erreur ! Source du renvoi introuvable.** On peut alors choisir un composant dans la **bibliothèque** sur la gauche.
2. Pour l'ajouter dans la **zone d'édition et de simulation**, il suffit de cliquer sur le *composant désiré*, puis de cliquer sur la **zone d'édition et de simulation** à l'endroit où on veut le placer.
3. Chaque *composant* que vous utiliserez aura des **attributs** modifiables dans la zone inférieure gauche de Logisim (**Erreur ! Source du renvoi introuvable.**). En effet, chaque composant possède un ou plusieurs **ports** d'entrée/sortie permettant la connexion avec autres composants.
 - Pour **changer le nombre de ports**, il suffit de modifier la valeur de l'attribut *Nombres d'entrée (Number Of Inputs)* du composant.
 - Pour **changer le nombre de bits** sur les ports d'entrée et sortie, il suffit de modifier la valeur de l'attribut *Data Bits* du composant.
4. Il est possible de faire des *copier/coller* d'un ou plusieurs composants. Dans ce cas, les composants **conserveront** tous les **attributs** préalablement définis.
5. Pour **ajouter une connexion** (un fil) entre deux composants, cliquez sur un des ports à connecter et déplacez la souris jusqu'au port de destination.
6. Une porte logique ou un circuit a au moins une **broche d'entrée** (*input pin*) permettant de définir les valeurs en entrée du circuit.
 - Vous trouverez la broche d'entrée () dans le menu en haut de Logisim.
7. Une porte ou un circuit a au moins une **broche de sortie** (*output pin*) permettant de vérifier les valeurs en sortie et donc le bon fonctionnement du circuit.
 - Vous trouverez la broche de sortie () dans le menu en haut de Logisim.
8. Pour changer l'orientation d'un composant, modifiez la valeur de l'attribut *Facing* de celui-ci.

B. Mode simulation

Pour activer Le **mode simulation** il faut sélectionner la main :  dans le menu en haut à gauche (**Erreur ! Source du renvoi introuvable.**). Ce mode permet de vérifier le bon fonctionnement d'un circuit, en affichant directement sur le circuit les valeurs des signaux (broches et fils).

1. Pour **définir des valeurs** sur une broche d'entrée (*input pin*), il suffit de cliquer sur celle-ci. Sa valeur devrait alterner entre '0' et '1'. Si la taille de la broche est supérieure à 1, il faut cliquer sur chaque bit pour en changer sa valeur.
2. En fonction des valeurs sur les entrées, les différents fils et sorties changeront de valeur.
3. En mode de simulation, Logisim utilise un **ensemble de couleurs** pour dénoter les valeurs sur les fils :

- **Gris** : Le fil **n'est relié** à aucune entrée ou sortie.
- **Bleu** : Le fil comporte une valeur, cependant elle est **inconnue**.
- **Vert foncé** : Le fil a valeur '0'.
- **Vert clair** : Le fil a 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**.

III. **NOTRE 1^{ER} CIRCUIT SUR LOGISIM**

A. Circuit 1 : Construction directe à partir d'un Logigramme

Il s'agit de **réaliser** et **simuler** le logigramme d'un circuit composé de deux inverseurs (porte **NOT**), deux portes **OR**, et une porte **AND**.

1. La première des choses à faire c'est **d'ajouter** les portes logiques puis modifier le nombre d'entrées (Fig. 2. 4).

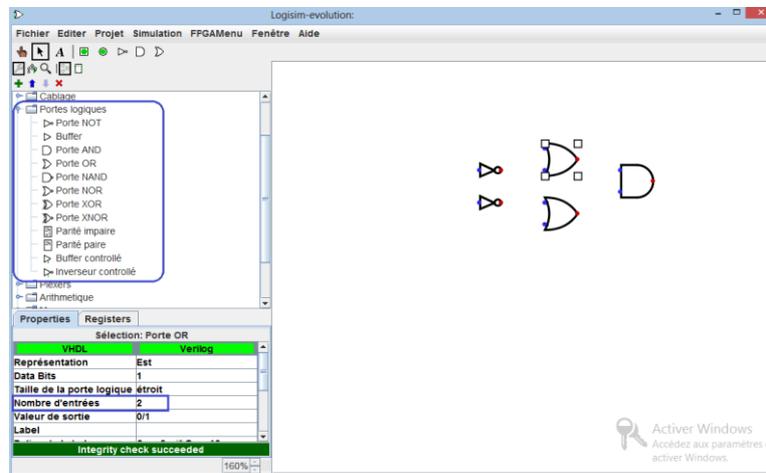


Fig. 2. 4. Ajout des portes logiques

2. **Ajouter** deux **broche d'entrée** () avec **label A** pour la première puis **B** pour la deuxième (Fig. 2. 5).
3. **Ajouter** ensuite une **broche de sortie** () avec **label S**.
4. Relier les différents éléments du circuit comme illustré sur la Fig. 2. 6.

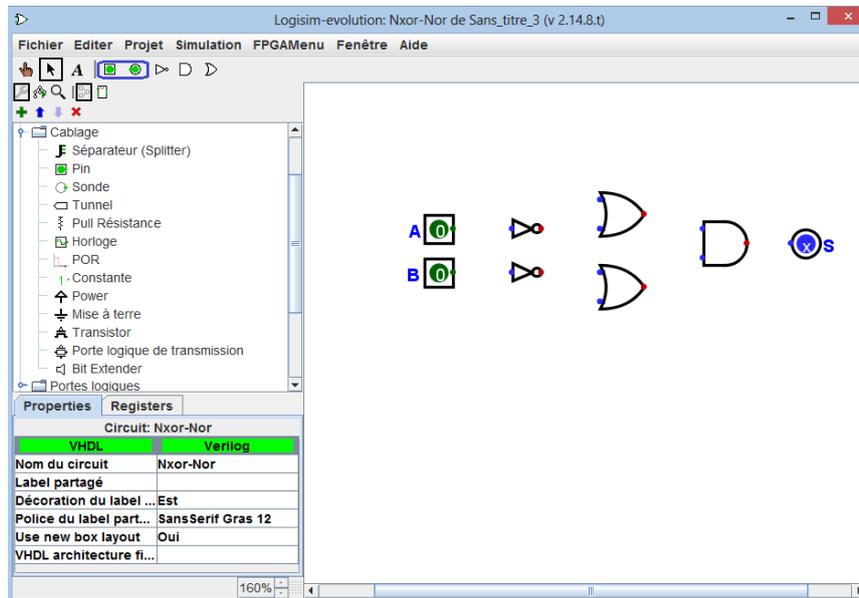


Fig. 2. 5. Ajout des broches d'entrées et de sorties.

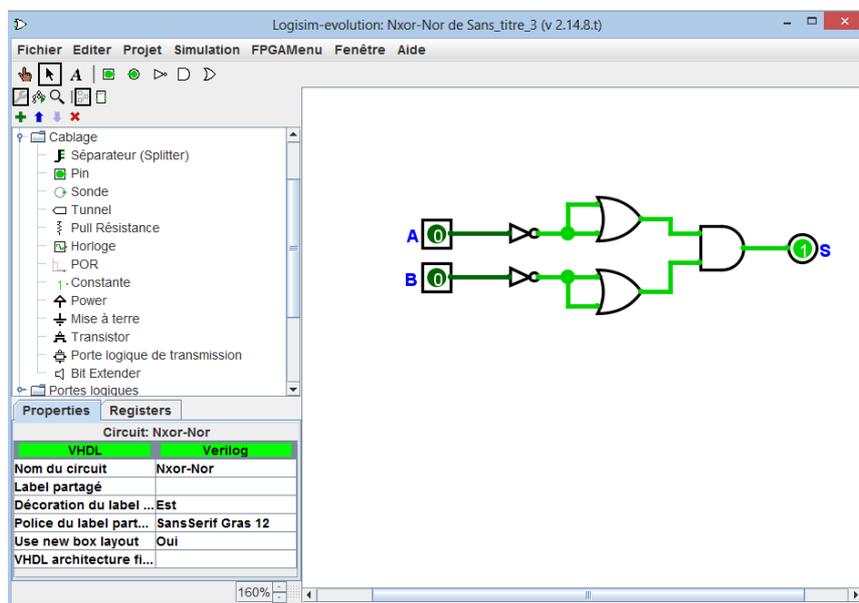


Fig. 2. 6. Connexion des portes.

5. Tester le circuit pour établir la table de vérité correspondante.
6. Quelle est la **fonction** obtenue.
7. Retrouver l'expression simplifiée de cette fonction en sélectionnant *Project --> Analyser circuit* dans Logisim.

B. Circuit 2 :

a) Construction à partir d'une Table de Vérité

1. Ouvrir un nouveau fichier puis sélectionner *Project --> Analyser circuit*.
2. Ajouter deux variables d'entrées : **A** et **B** dans l'onglet *Entrées (Inputs)* –voir Fig. 2. 7- et une variable de sortie : **S** dans l'onglet *Sorties (Outputs)*.
3. Dans l'onglet *Table*, cliquer sur les 'x' de la sortie S de telle sorte à avoir une table qui correspond à la fonction **Non-Ou-Exclusive (XNOR)**.
4. Afficher le résultat obtenu dans l'onglet *Expression* ; puis donner les deux formats simplifiés de cette expression à partir de l'onglet *Karnaugh*.

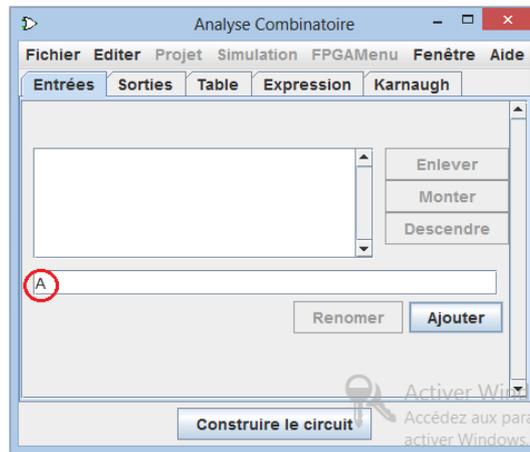


Fig. 2. 7. Ajouter des variables.

5. Cliquer sur **construire le circuit** puis nommer votre circuit et activer l'utilisation des portes **NAND** à 2 entrées uniquement ;
6. Afficher et commenter le résultat ainsi obtenu.
7. Tester le circuit obtenu en passant en mode simulation : 🖱️.
8. Transformer les expressions ainsi obtenus afin de réaliser un **Non-OU-exclusif** en utilisant un **minimum** de portes **NOR** (en pratique on veut utiliser un seul circuit CI 74LS00).

b) Reconstruction à partir d'un logigramme

1. Ouvrir un nouveau fichier et placer le **minimum** de portes **NOR** (trouvé dans question 9) dans la zone de travail, placer les deux entrées **A** et **B** et la sortie **S** puis relier les différents éléments du circuit pour réaliser la fonction **XNOR**.
2. Choisir l'icône représentant une main : 🖱️ dans la barre d'outils (passage en mode « simulation») et cliquer alternativement sur chaque entrée ; chaque clic modifie la valeur de l'entrée correspondante.
3. L'onglet *Table* de la fenêtre affichée par le menu *Projet --> Analyser circuit*, construit la *table de vérité* du circuit au fur et à mesure. Représenter cette table.
4. Donner l'expression ainsi obtenue dans l'onglet *Expression*.
5. Commenter les résultats en les comparant avec ceux obtenus dans la partie **B. a)**.

T.P. N° 3
Additionneur et Soustracteur

T.P. N° 3

Additionneur et Soustracteur

SMP6 -P1

Module d'Electronique Numérique

OBJECTIF :

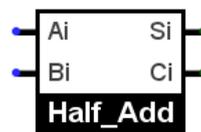
- Réaliser un **additionneur** à partir des **demi-additionneurs**.
- Concevoir un $\frac{1}{2}$ **soustracteur** puis un **soustracteur Complet**
- Concevoir un **soustracteur Complet** en se basant sur le circuit d'additionneur.
- Simulation sous le logiciel Logisim pour vérifier le fonctionnement de ces circuits.

I. ETUDE D'UN ADDITIONNEUR

On va étudier le mode de fonctionnement d'une partie d'une calculatrice et notamment les opérations de base. Le principe de base de cette partie consiste à **additionner** des nombres.

I.1. Etude d'un demi-additionneur (Half Adder) logique

Un demi-additionneur est un circuit logique à deux entrées à additionner : a_i et b_i et deux sorties : la somme S_i et la retenue de sortie C_i de poids directement supérieur.



1. Etablir le **logigramme** qui permet de faire l'addition de deux nombres de **1-bit**.
2. Ouvrir un nouveau fichier sous **Logisim**, afin de créer un nouveau circuit, il faut aller dans *Project* -> *Ajouter circuit...* -> nommer le circuit comme suit : **Half_Add**.
3. Effectuer la **saisie du schéma** structurel électronique correspondant à votre logigramme sur **Logisim**.
4. Ajouter les **entrées** a_i , b_i et les **sortie** S_i et C_i , à partir de la barre d'outils.
5. Procéder à la **simulation** avec Logisim en cliquant alternativement sur chaque entrée.
6. Pour **afficher les résultats**, sélectionner *Project* --> *Analyser circuit*.
7. Montrer la **table de vérité**, les expressions et le tableau de Karnaugh de ce circuit.

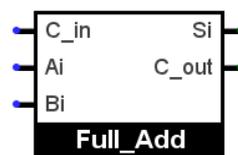
Ce sous circuit réalisé dans Logisim peut être réutilisé maintenant dans d'autres circuits. Le sous circuit créé devient un **composant disponible dans la bibliothèque**.

Remarque :

Le circuit en cours d'édition est celui qui comporte une petite loupe en dessous du nom du projet.

I.2. Additionneur complet (Full Adder)

Pour la prise en compte de la retenue des bits de poids inférieurs, un circuit additionneur doit donc comporter trois entrées a_i , b_i et C_{in} et deux sorties ; le résultat est S_i et une retenue C_{out} .

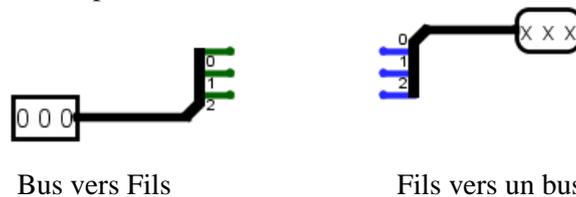


1. Ajouter un nouveau sous circuit dans le même projet de Logisim, il faut aller dans *Project* -> *Ajouter circuit...* -> nommer le circuit comme suit : **Full_Add**.

2. Construire l'**additionneur complet** à partir de deux **demi-additionneurs** (*Half_Add*) et réaliser le montage à l'aide de Logisim. Il est possible d'ajouter un sous circuit *Half_Add* de la même manière que l'utilisation d'un composant quelconque. On clique sur *Half_Add* dans le menu des composants, puis on le place en cliquant sur le schéma.
3. Teste de fonctionnement du montage réalisé.
 - a. **Vérifier** toutes les **combinaisons possibles** d'opérandes et assurez-vous d'obtenir les bons résultats arithmétiques en sortie.
 - b. Dresser la **table de vérité**, les **expressions obtenues** et les **tableaux de Karnaugh** de vos résultats.
4. Réalisation du montage additionneur de **deux nombres A et B de 3 bits** chacun :
 - a. Ajouter un nouveau sous circuit dans le même projet de Logisim, il faut aller dans Project -> Ajouter circuit... -> nommer le circuit comme suit : *Add_3bits*.
 - b. Réaliser le montage d'addition de **A** et **B** en utilisant le nombre nécessaire d'instances du circuit *Full_Add* ainsi conçus. Tester le circuit.

Remarques :

- Pour définir l'entrée **A** ou **B** comme un bus de **3 bits**, il faut ajouter un élément *Pin* et définir sa taille via l'attribut *Data bits = 3*. Lorsque l'on tire un fil de l'un de ces entrées, ce n'est plus un simple signal mais un **bus de 3 bits**.
- Pour pouvoir **connecter** les éléments de ce bus aux entrées des additionneurs, on va devoir séparer les différents fils du bus afin de pouvoir les traiter un par un. L'élément *séparateur (splitter)* de câblage permet d'effectuer ces conversions dans les deux sens : d'un bus de 3 bits vers 3 fils, et de 3 fils vers un bus de 3 bits.
- Il faut définir les tailles d'entrées et de sorties du *séparateur* via les **attributs** *Largeur de bits en entrée (Bit Width In)* et *Ventilation en sortie (Fan out)*. Dans notre cas on définit les deux valeurs à 3. (Le bit de poids faible est indexé à 0)



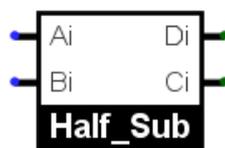
- c. Indiquer où nous devons placer un **indicateur** qui vaut **1** lorsque j'aurais un débordement (**dépassement de capacité**) pour l'**addition** de **A** et **B** et **0** sinon.
- d. A l'aide de C.I. **4008** sous *Workbench* ou **74LS83** sous *ISIS* (Voir Fig. 3. 1 et Fig. 3. 2).
–Voir brochage sur Fig. 3. 3 et Fig. 3. 4. (*Facultatif*)

II. ETUDE D'UN SOUSTRACTEUR

On va étudier dans cette partie les opérations de **soustraction** binaire en utilisant le **complément à 2**.

II.1. Etude d'un demi-soustracteur (Half Subtractor)

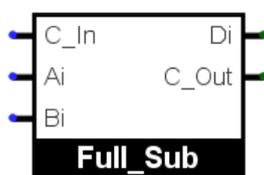
Un demi-soustracteur est un circuit logique à deux entrées et deux sorties. On veut réaliser la soustraction de deux entrées a_i et b_i . Les deux sorties sont la **différence** D_i et la **retenue** sortante C_i .



1. Ouvrir un nouveau projet sous Logisim, ajouter un nouveau sous circuit, il faut aller dans *Project -> Ajouter circuit...* -> nommer le circuit comme suit : **Half_Sub**.
2. Sélectionner *Project --> Analyser circuit*. Ajouter deux variables d'entrées : a_i et b_i dans l'onglet *Entrées (Inputs)* et deux variables de sortie : D_i et C_i dans l'onglet *Sorties (Outputs)*.
3. Dans l'onglet *Table*, cliquer sur les 'x' des sorties de telle sorte à avoir la **table de vérité** qui correspond à ce $\frac{1}{2}$ soustracteur
4. Montrer les résultats obtenus dans les onglets *Expression* et *Karnaugh*.
5. En se basant sur les résultats obtenus, placer un minimum de portes pour réaliser ce $\frac{1}{2}$ soustracteur.
6. Procéder à la simulation avec Logisim.

II.2. Etude d'un Soustracteur complet

Pour généraliser cette structure afin de décrire la soustraction de nombres de taille supérieure à **1 bit**, il faut introduire une variable supplémentaire C_{in} qui représente une **retenue entrante (Carry in)**.



1. Ajouter un nouveau sous circuit dans le même projet de Logisim, il faut aller dans *Project -> Ajouter circuit...* -> nommer le circuit comme suit : **Full_Sub**.
2. Sélectionner *Project --> Analyser circuit*. Ajouter deux variables d'entrées : a_i , b_i et c_{in} dans l'onglet *Entrées (Inputs)* et deux variables de sortie : D_i et C_{out} dans l'onglet *Sorties (Outputs)*.
7. Montrer les résultats obtenus dans les onglets *Expression* et *Karnaugh*.
3. Transformer les expressions trouvées dans l'onglet *Expression* afin de construire un **soustracteur complet** à partir de **deux demi-soustracteurs (Half_Sub)**.
4. Réaliser le montage et procéder à la **simulation**.
 - a. Vérifier toutes les combinaisons possibles d'opérandes et assurez-vous d'obtenir les bons résultats arithmétiques en sortie.
 - b. Redresser la **table de vérité** de vos résultats.
5. Etablir le montage d'un soustracteur de **deux nombres A et B de 4 bits** chacun.
 - a. En utilisant un nombre nécessaire du **soustracteur complet (Full_Sub)** ainsi conçus. Tester le circuit.
 - e. A l'aide de C.I. **4008** sous *Workbench* ou **74LS83** sous *ISIS* (Voir Fig. 3. 1 et Fig. 3. 2). - Voir brochage sur Fig. 3. 3 et Fig. 3. 4. (*Facultatif*)

III. ETUDE D'UN ADDITIONNEUR-SOUSTRACTEUR

Dans un processeur de calcul, on ne trouve pas de circuit soustracteur. Pour ce faire, on exploite les propriétés du **complément à 2** (CA_2) et le fait que le bit de poids faible de l'additionneur n'a pas de retenue d'entrée ($C_{in}=0$). En effet, effectuer $A - B$ en CA_2 , est équivalent à $A + CA_1(B) + I$.

Pour effectuer la deuxième addition (+1), il suffit d'injecter un **1** en guise de retenue entrante ($C_{in}=1$) dans l'additionneur de poids faible. On peut donc supposer que l'on dispose d'un **signal de contrôle E** qui vaut **0** lorsque l'on veut faire une **addition**, et **1** lorsque l'on veut faire une **soustraction**.

1. Etablir la relation liant l'entrée de contrôle E aux entrées de l'additionneur C_{in} et B_i .
2. Dans le projet de l'additionneur déjà réalisé, sélectionner *Project -> Ajouter circuit... ->* et créer un sous circuit sous le nom : *Add_Sub*.
3. Réaliser le montage de l'**additionneur-soustracteur** à partir de *Full_Add* et un minimum de portes, pour le cas de deux mots A et B de **2 bits** chacun. Procéder à la **simulation**.
4. En **complément à 2** sur n bits, quel est le **plus petit entier (négatif)** représentable, et quel est le **plus grand entier (positif)** représentable ?
5. Montrez que si A et B sont de **signes opposés**, alors il ne peut pas y avoir de **dépassement de capacité** dans le calcul de $A - B$ en CA_2 sur n bits.
6. Dédurre de la question précédente une **condition nécessaire** pour qu'il y ait dépassement de capacité.
7. Complétez le tableau ci-dessous, dans lequel on ne s'intéresse qu'au calcul effectué par le *Full_Add* le plus à gauche. *L'indicateur O (Overflow)* doit prendre la valeur **1** en cas de **dépassement de capacité** ou de **résultat faux**, la valeur **0** sinon.

a_{n-1}	b_{n-1}	C_{n-1}	S_{n-1}	C_n	O
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

8. En vous référant à la table de vérité ci-dessus, exprimez O en fonction de C_{n-1} et C_n . Justifiez votre réponse.
9. Complétez le circuit de l'additionneur-soustracteur afin qu'il produise la sortie O . Tester le circuit.

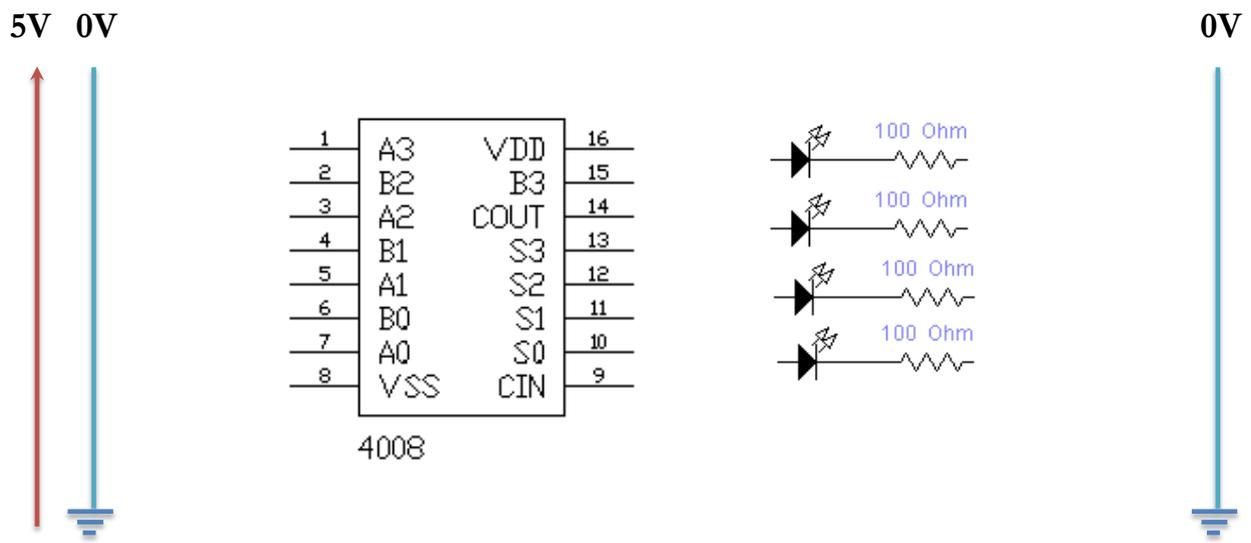


Fig. 3. 1

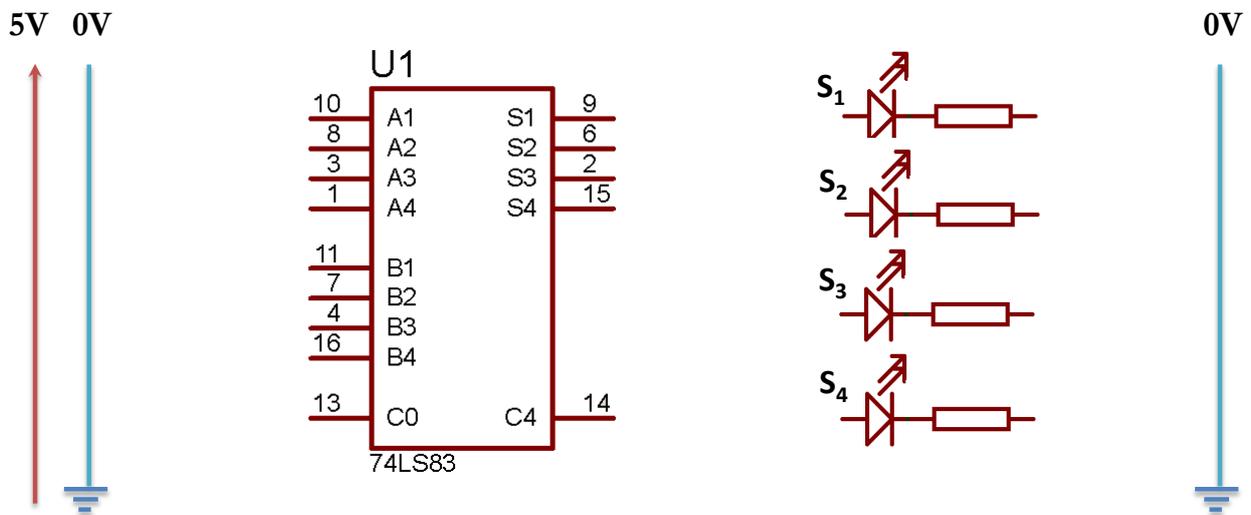


Fig. 3. 2

BROCHAGES des Circuits Intégrés

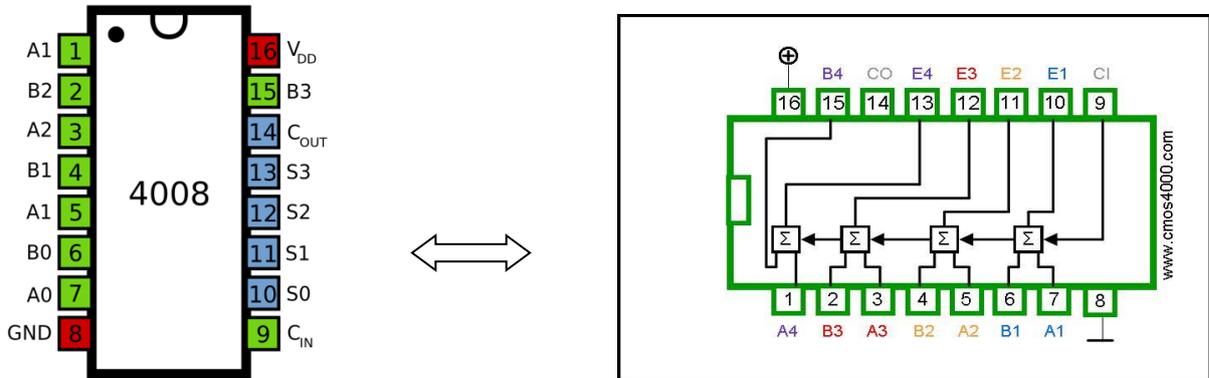


Fig. 3. 3. Brochage du circuit intégré 4008

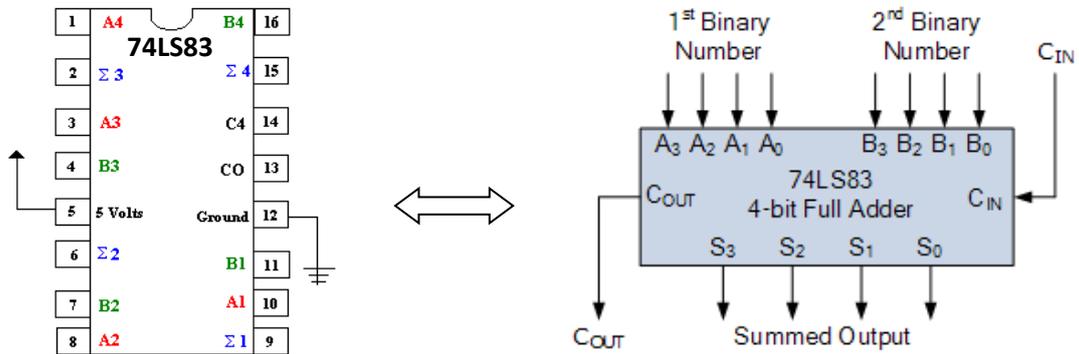


Fig. 3. 4. Brochage du circuit intégré 74LS83

T.P. N° 4
Comparateur Binaire

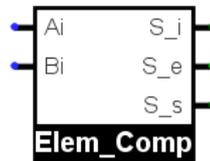
T.P. N° 4
Comparateur Binaire
SMP6 -P1
Module d'Electronique Numérique

Objectif :

- Réaliser un **comparateur** de deux nombres de **n bits** chacun.
- Etre capable d'utiliser le logiciel Logisim afin de réaliser et vérifier le fonctionnement de ces circuits.

I. Comparateur élémentaire

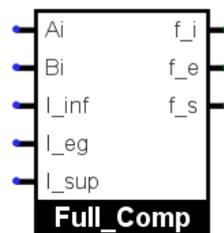
On souhaite mettre au point un circuit permettant de comparer deux entiers naturels codés en binaire. Dans un premier temps, on commence par réaliser un **comparateur élémentaire** de deux mots $A = a_i$ et $B = b_i$ de **1 bit**.



1. Etablir les **équations** des sorties S_i , S_e , et S_s en fonction des entrées a_i et b_i .
2. Avec les portes logiques appropriées, établir le **logigramme** de ce comparateur.
3. Ouvrir un nouveau projet sous Logisim, ajouter un nouveau sous circuit, il faut aller dans *Project -> Ajouter circuit...* -> nommer le circuit comme suit : **Elem_Comp**.
4. Effectuer la saisie du **schéma structurel** électronique correspondant à ce logigramme sur Logisim.
5. Procéder à la **simulation** avec Logisim en vérifiant toutes les combinaisons possibles d'opérandes.

II. Comparateur complet

On souhaite maintenant étendre l'amplitude du comparateur à deux mots de **2 bits** $A = a_1 a_0$ et $B = b_1 b_0$, ce comparateur dit complet (**Full comparator**) doit donc disposer d'**entrées supplémentaires** ($I_<$, $I_=>$, et $I_>$) permettant la prise en compte, en **cas d'égalité**, du résultat de la comparaison des bits de **poinds inférieurs**.



1. Etablir les **équations** de la comparaison finale f_I , f_E et f_S en fonction des entrées $S_<$, $S_=>$, $S_>$, $I_<$, $I_=>$, et $I_>$.
2. Dans le même projet, sélectionner *Project -> Ajouter circuit...* -> et créer un sous circuit sous le nom : **Full_Comp**.

3. Concevoir le schéma d'un **comparateur complet** de **2 mots** d'un **seul bit** chacun en associant le comparateur élémentaire (*Elem_Comp*) et un minimum de **portes élémentaires**.
4. Concevoir le **schéma** d'un comparateur de **2 mots A et B** de **2 bits** chacun en utilisant les **comparateurs complets** (*Full_Comp*) déjà conçus.
5. Procéder à la **simulation** sous Logisim en vérifiant toutes les combinaisons possibles d'opérandes.
6. Compléter le schéma de câblage de la Fig. 4. 1 pour comparateur **2 mots** de **4 bits** à l'aide de **C.I. 74LS85** (voir Brochage et TV en Fig. 4. 2 et Tableau. 4. 1)

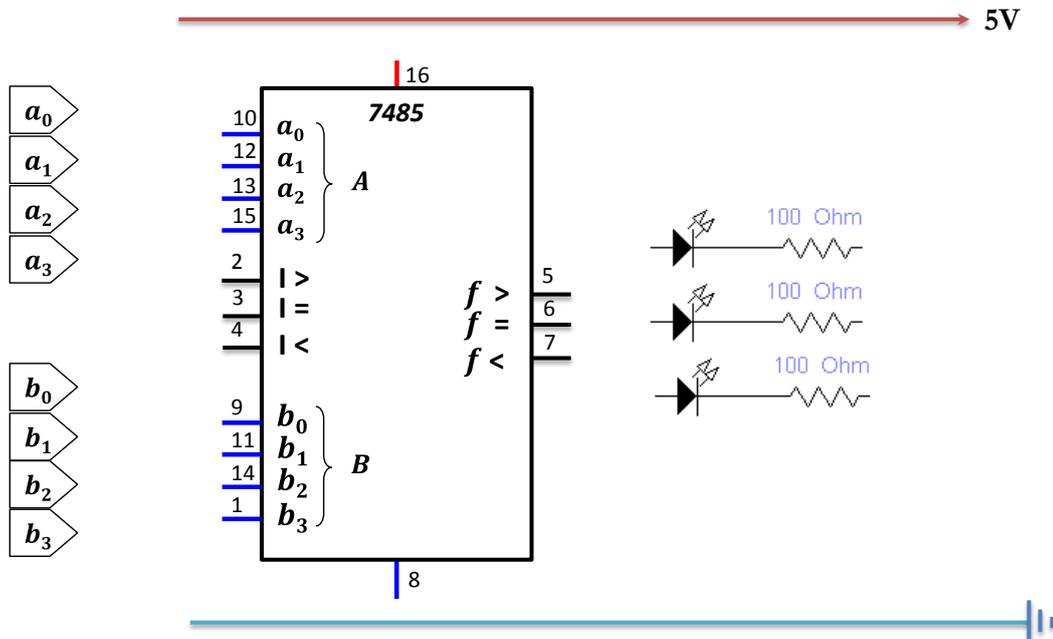


Fig. 4. 1

7. Etablir le montage de comparaison de deux nombres de **5 bits** chacun, $A = a_4a_3a_2a_1a_0$ et $B = b_4b_3b_2b_1b_0$ à l'aide des **comparateurs complets** déjà conçus sur Logisim ou encore à l'aide des C.I. 74LS85 sur Proteus Isis.

BROCHAGE des Circuits Intégrés

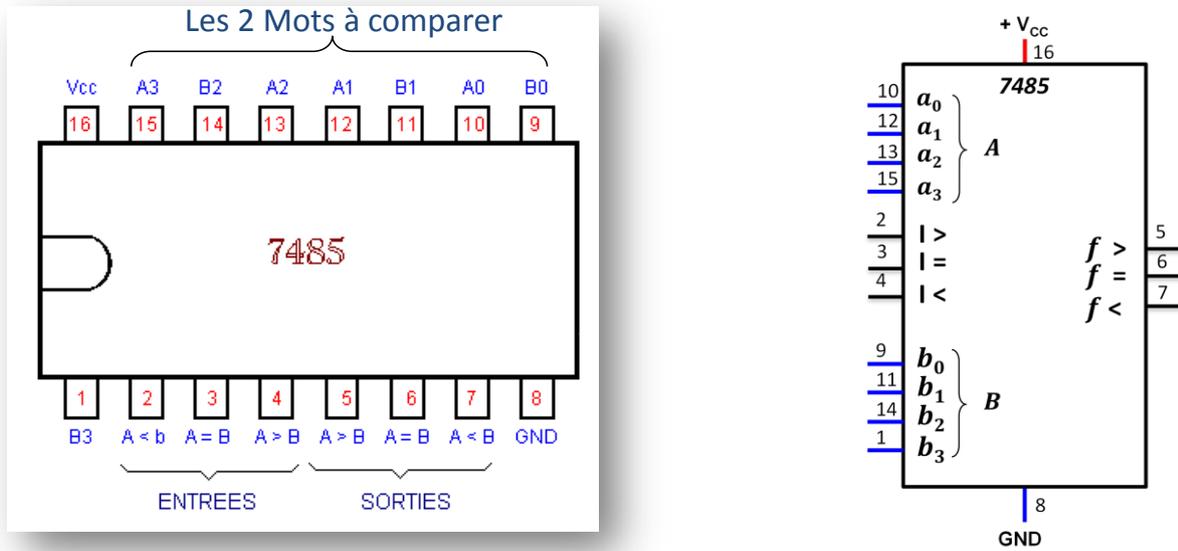


Fig. 4. 2. Brochage du circuit intégré 7485

Tableau. 4. 1. Table de vérité du circuit intégré 7485

Entrées des nombres				Entrées cascadables			Sorties		
A3, B3	A2, B2	A1, B1	A0, B0	A > B	A < B	A = B	A > B	A < B	A = B
A3 > B3	X	X	X	X	X	X	1	0	0
A3 < B3	X	X	X	X	X	X	0	1	0
A3 = B3	A2 > B2	X	X	X	X	X	1	0	0
A3 = B3	A2 < B2	X	X	X	X	X	0	1	0
A3 = B3	A2 = B2	A1 > B1	X	X	X	X	1	0	0
A3 = B3	A2 = B2	A1 < B1	X	X	X	X	0	1	0
A3 = B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	1	0	0
A3 = B3	A2 = B2	A1 = B1	A0 < B0	X	X	X	0	1	0
A3 = B3	A2 = B2	A1 = B1	A0 = B0	1	0	0	1	0	0
A3 = B3	A2 = B2	A1 = B1	A0 = B0	0	1	0	0	1	0
A3 = B3	A2 = B2	A1 = B1	A0 = B0	0	0	1	0	0	1
A3 = B3	A2 = B2	A1 = B1	A0 = B0	X	X	1	0	0	1
A3 = B3	A2 = B2	A1 = B1	A0 = B0	1	1	0	0	0	0
A3 = B3	A2 = B2	A1 = B1	A0 = B0	0	0	0	1	1	0

T.P. N° 5
Montages à Bascules

T.P. N° 5
Montages à Bascules
SMP6 -P1
Module d'Electronique Numérique

Rappel

Pour certains opérateurs, l'état de la sortie dépend non seulement de la combinaison appliquée à l'entrée (**logique combinatoire**) mais aussi de l'état précédent des sorties du circuit : ils sont dits **séquentiels** et ont un effet « **mémoire** ».

La logique séquentielle est donc une logique combinatoire avec une mémorisation des sorties. Cette mémorisation est réalisée par ce qu'on appelle une **bascule** ; c'est un organe de mémorisation unitaire (mémorisation d'une seule donnée).

Objectif

Le but de ce TP est de concevoir en simulation et expérimentation les différentes bascules RS, D et JKH, ainsi qu'un compteur asynchrone à partir de portes logiques combinatoires.

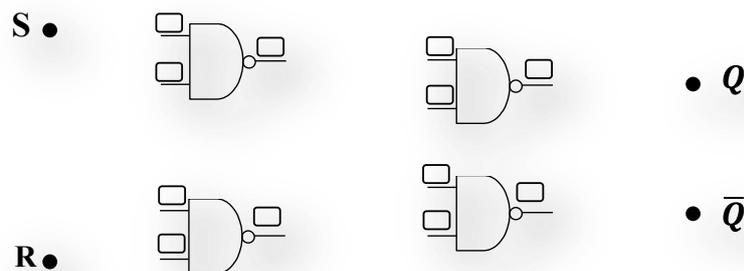
I. Réalisation pratique d'une bascule RS

Une bascule RS peut être réalisée par un circuit intégré ou par l'association d'opérateurs NON-OU (**NOR**) ou NON-ET (**NAND**).

A- RS asynchrone

Nous commençons ce TP par la réalisation d'une bascule asynchrone, c'est à dire une bascule dont les sorties **réagissent directement** sur les modifications des entrées, après délais de commutation.

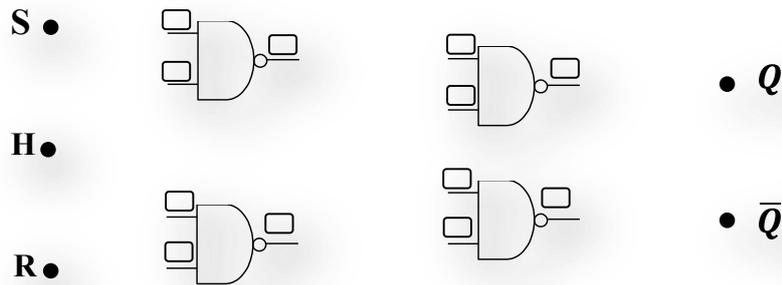
1. Compléter ce logigramme de la bascule **RS asynchrone** conçu avec des portes **NAND** :



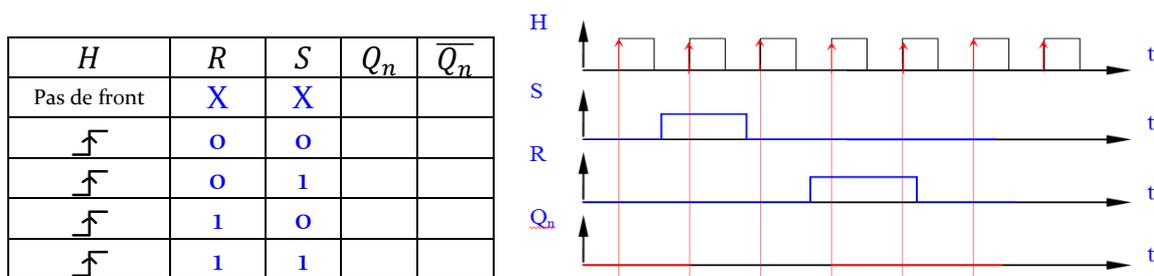
2. Indiquer le **brochage** du CI **74LS00** à utiliser sur le schéma du montage précédant (voir Fig. 5. 1).
3. Réaliser le montage de cette bascule à l'aide du CI **74LS00**.
4. Tester l'ensemble des **combinaisons possibles** et déduire la **table de vérité** simplifiée.
5. Prenant le cas où **R** et **S** sont au **niveau haut (R=S=1) -cas anormale d'exploitation-**, que se passe-t-il si maintenant R et S sont ramenées à **0 en même temps**. Réexpliquer ce phénomène théoriquement.

B- RSH

1. Compléter le schéma de la bascule **R-S-H (synchrone)** suivant :



2. Indiquer le **brochage** du CI **74LS00** sur le schéma du montage.
3. **Réaliser** ce montage à l'aide du CI **74LS00**.
4. Tester l'ensemble des **combinaisons possibles** et déduire la **table de vérité simplifiée**.
5. Compléter le **chronogramme** suivant en analysant le fonctionnement du montage :



II. Réalisation pratique d'une bascule D

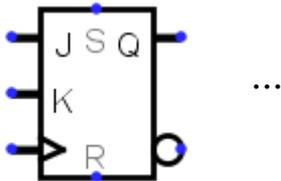
1. Donner le schéma d'une **basculer D synchrone** à l'aide de la bascule **RSH** seulement.
2. Donner le **logigramme** de ce montage, en indiquant le **brochage** du composant utilisé.
3. **Réaliser** ce montage et **tester** l'ensemble des **combinaisons possibles**.
4. Donner le **chronogramme** du montage et la **table de vérité simplifiée**.

III. Simulation sous Logisim d'une bascule JKH

1. Donner le **schéma** d'une bascule **JKH** à base d'une bascule **RSH**.
2. Investiguez les caractéristiques d'opération du C.I. **74LS73** (Voir Fig. 5. 2).
3. Ouvrir un nouveau projet sous Logisim, ajouter un nouveau sous circuit, il faut aller dans *Project -> Ajouter circuit...* -> nommer le circuit comme suit : **JKH**.
 - a. Réaliser le logigramme de la bascule **JKH** à base d'une bascule **S-R Flip-Flop** se trouvant dans la bibliothèque de composant (*Memory*).
 - b. Simuler le circuit et déduire la **table de fonctionnement** de cette bascule.
4. Déterminer **J** et **K** de manière à ce que cette bascule se comporte comme une **basculer D**.
5. Vérifier la **table de vérité** de ce pseudo bascule D.
6. Peut-on faire le montage inverse (réaliser une bascule JK à partir d'une bascule D) ?

IV. Simulation sous Logisim d'un Compteur asynchrone

1. Dans le même projet, sélectionner *Project* -> *Ajouter circuit...* -> et créer un sous circuit sous le nom : *Comp_Asynch_mod_8*.
 - a. Compléter le montage ci-après pour réaliser un *compteur asynchrone modulo 8* en utilisant des bascules **JKH** de la bibliothèque de Logisim. La sortie de ce compteur sera **codée en binaire naturel**.



- b. Ajouter un **afficheur hexadécimal** pour visualiser la sortie du compteur.
2. **Simuler** ce compteur.
 3. Tracer le **chronogramme** des sorties du compteur.
 4. Quelle est la **période** de la sortie de la dernière bascule JKH par rapport à celle de la première ?
 5. Proposer une **application** possible de ce compteur ?
 6. Modifier le montage de telle sorte qu'on puisse réaliser un *compteur décimal*.

BROCHAGE des Circuits Intégrés

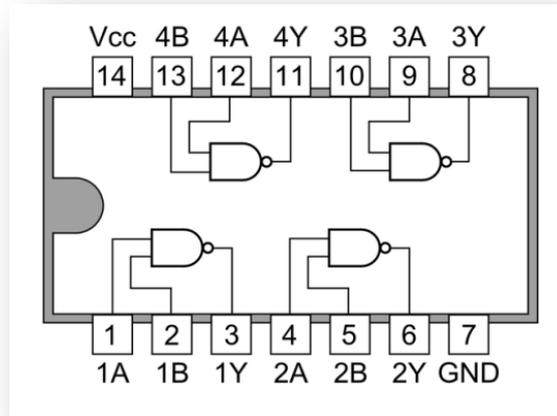


Fig. 5. 1. Brochage du circuit intégré 74LS00

Input				Output	
$\overline{\text{CLR}}$	CLK	J	K	Q	$\overline{\text{Q}}$
L	X	X	X	L	H
H	↓	L	L	Q ₀	$\overline{\text{Q}}_0$
H	↓	H	L	H	L
H	↓	L	H	L	H
H	↓	H	H	Toggle	Toggle
H	H	X	X	Q ₀	$\overline{\text{Q}}_0$

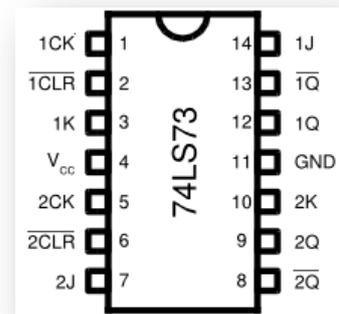


Fig. 5. 2. TV et Brochage du circuit intégré 74LS73

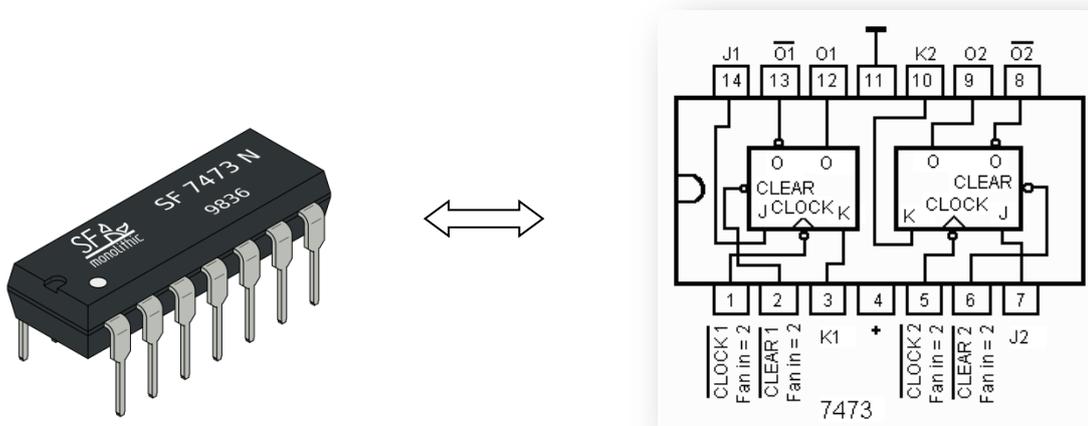


Fig. 5. 3. Image et structure interne du circuit intégré 74LS73