

TP- REPRESENTATION DES DONNEES

Discipline : REPRESENTATION DES DONNEES

Enseignant : HAMADOU AHMED

Année : 2020/2021

PARTIE A :

Définir aux étudiants les termes suivants :

- Ordinateur
 - Système d'exploitation
 - DOS
 - MSDOS
 - Adresse IP
-
1. Démarrer l'ordinateur.
 2. Ouvrir l'invite de commande (fenêtre noire). (Touche windows +R, tapez cmd dans la barre de recherche puis entrée).
 3. Pour savoir la version de votre système d'exploitation : tapez la commande ver, puis la touche entrée.
 4. Pour savoir le nom de votre PC, tapez hostname, puis la touche entrée.
 5. Pour savoir votre adresse IP, tapez ipconfig, puis la touche entrée.
 6. Quitter l'invite de commande en tapant exit.

PARTIE B:

Commandes MSDOS:

- Créer un repertoire: md nom_repertoire
- Déplacement vers un repertoire : cd nom_repertoire
- Déplacement vers un repertoire parent : cd..
- Créer un fichier vide : echo>nom_fichier
- Créer un fichier qui contient du texte : echo>nom_fichier saisie texte

1. Créer le repertoire INFO.
2. Dans le repertoire INFO, créer 3 sous-répertoires INFO1, INFO2 et INFO3.
3. Dans le repertoire INFO1, créer 2 sous-répertoires cours et TD.
4. Dans le repertoire INFO2, créer 2 sous-répertoires TP et DEVOIR.
5. Dans le repertoire INFO3, créer 1 sous repertoire et 2 fichiers CC et CF.

PARTIE C :

1. Quelle est l'adresse IP de votre PC ?
2. Quel est son masque de sous-réseau ?
3. Quelle est son adresse MAC ?

La représentation des nombres relatifs, des nombres réels

Thème : Représentation et traitement des données

Compétences à acquérir :

Notions abordées :

- Addition en binaire
- Représentation binaire d'un entier relatif, d'un réel
- Espace de stockage
- Langage python : affectations, séquences, tests... et boucles

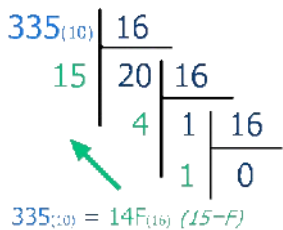
- Savoir additionner deux nombres binaires entre eux
- Savoir trouver le complément à deux d'un nombre binaire
- Se familiariser avec python et être capable d'écrire des petits programmes utilisant la boucle « while », les fonctions « input », « print », listes : savoir parcourir une liste avec l'instruction « for »

Éléments d'information et exemples.

Sitographie :

- Diaporama « Représentation des nombres en machine » : <http://mariepierrot.free.fr/lycee/1ereNSI/representationdesnombresenmachine.pdf>
- Trouver le complément à deux (vidéo 17mn) : <https://www.youtube.com/watch?v=oTIXW6AhPWQ>
- Circuit logique, l'additionneur : <https://fr.wikipedia.org/wiki/Additionneur>
- Animation sur l'additionneur logique : <http://ressources.univ-lemans.fr/AccessLibre/UM/Pedago/physique/02/electro/addi4bits.html>
- Toujours les conversions le cas de l'hexadécimal : <http://villemin.gerard.free.fr/Wwwgvm/Numerati/ConHexa.htm>
- Arithmétique en nombres à virgule flottante : <https://docs.python.org/fr/3.6/tutorial/floatpoint.html>

Conversion en hexadécimal :



$$335_{10} = 1 \times 16^2 + 4 \times 16^1 + 15 \times 16^0$$

$$= 1 \times 16^2 + 4 \times 16 + 15$$

Addition « à la main » de nombres binaires :

Retenues:		1		1			
89		1	0	1	1	0	0
+21	+	0	0	1	0	1	0
= 110	=	1	1	0	1	1	0

Fonctions de conversion natives :

`bin() ; hex() ; int()`

```
>>> bin(335)
>>> '0b101001111'
>>> hex(335)
>>> '0x14f'
>>> int('101001111',2)
>>> 335
>>> int('14F',16)
>>> 335
```

Trouver le complément à deux d'un nombre binaire :

Méthode 1

- 1) On inverse tous les bits
- 2) On ajoute 1

Méthode 2

- 1) On part du LSB (bit de poids faible)
- 2) On remonte jusqu'au premier « 1 »
- 3) On inverse tous les autres bits

Calculs binaires :

<http://python.jpweb.com/python/mesrecettespython/doku.php?id=binaire>

... voir divmod() et l'opérateur ^

Travail à réaliser

I- Les nombres relatifs

- 1) Ecrire un programme qui affiche le résultat de l'addition deux nombres exprimés en binaire, donnés par l'utilisateur.
- 2) Ecrire un programme (ou une commande !) qui affiche le complément à 2 (CP2) d'un nombre exprimés en binaire, donné par l'utilisateur.
- 3) a- Chercher « à la main » comment réaliser une multiplication en binaire.
b- Ecrire l'algorithme correspondant à cette opération,
c- ...enfin écrire le programme en python.

II- Les nombres réels

- 1) Tester dans l'interpréteur les commandes suivantes (4×10^{-12} s'écrit en python $4e-12$) :
`1+1e-15 ; 1+1e-16 ; 0.1+0.1+0.1==0.3 ; 3.11 + 2.08 ; ...`
- 2) Comment pourriez-vous expliquer les résultats obtenus (aller voir le dernier lien de la sitographie...) ?

TP architecture ordinateur 2020

TP3 : Circuits logiques avec logisim-2.16.1.2.exe

I. Introduction

Logisim est un logiciel open-source permettant de concevoir et de simuler des circuits logiques. Ce document est un tutoriel qui décrit comment établir un système numérique à l'aide de cet éditeur de schéma. Nous expliquerons les démarches nécessaires afin de concevoir, simuler une CPU simple.

Il existe différentes façons de décrire formellement les systèmes numériques : des langages de description du matériel (HDL), des tables de vérité, des graphes d'états, ou des schémas. Logisim permet uniquement de travailler sur des schémas. La première partie expliquera comment réaliser son premier schéma.

Une des particularités de Logisim est de pouvoir éditer et simuler son circuit en même temps. Nous expliquerons plus tard dans ce document comment simuler un circuit.

Etape 0: S'orienter dans l'interface

Lorsque vous démarrez Logisim, vous verrez une fenêtre semblable à l'illustration suivante.

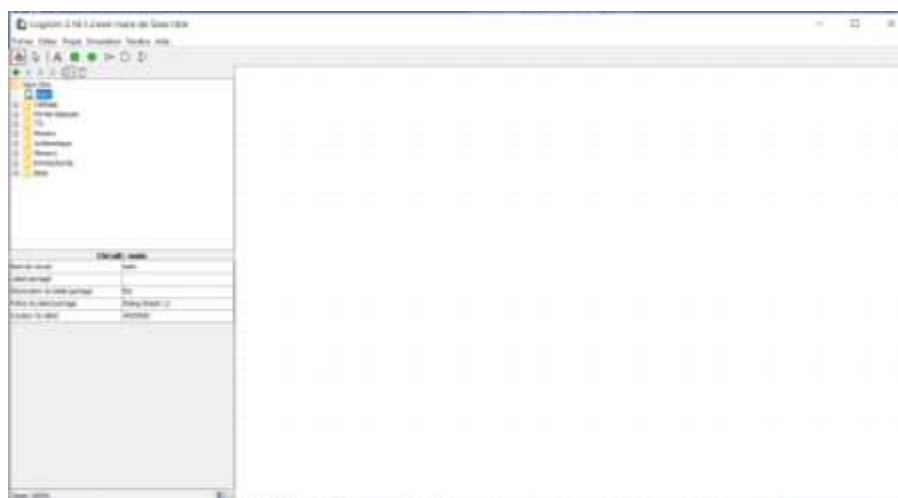
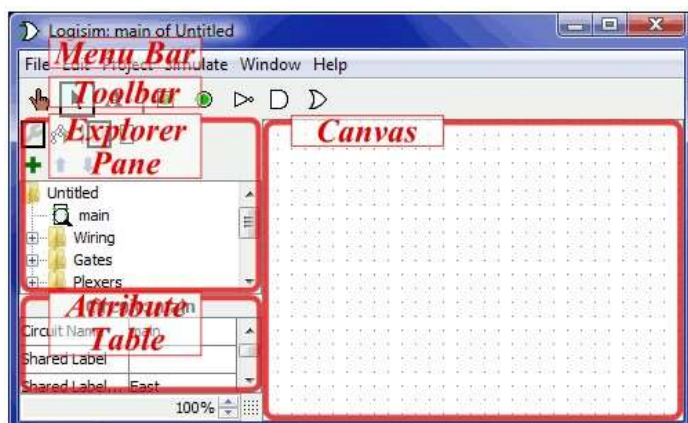


FIGURE 1. Interface de Logisim

L'espace de Logisim est divisé en trois parties appelées le *panneau de navigation*, la *table des attributs*, et la *surface de travail* (ou zone d'édition). Au-dessus de ces trois parties se trouvent la *barre de menu* et la *barre d'outils*.



Vous pouvez rapidement jeter un œil sur les détails du panneau de navigation et la table des attributs : nous ne les examinerons pas en détail dans ce tutoriel, et vous pouvez juste les ignorer. Par ailleurs, la barre de menu s'explique d'elle-même.

Il nous reste la barre d'outils et la surface de travail. La surface de travail est l'endroit où vous dessinez votre circuit ; et la barre d'outils contient les outils qui nous permettrons d'y arriver.

Question 1

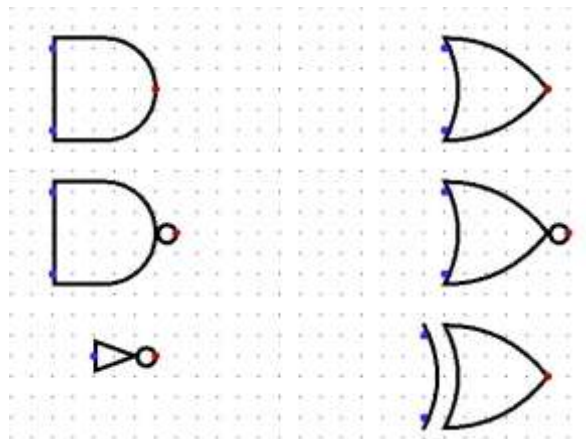
Établir la table de toutes les fonctions logiques à 2 variables (a et b) possibles. Identifier celles que vous connaissez.

La première chose que nous allons faire c'est d'ajouter les circuits logiques puis modifier le nombre de portes logiques et leurs combinaisons.

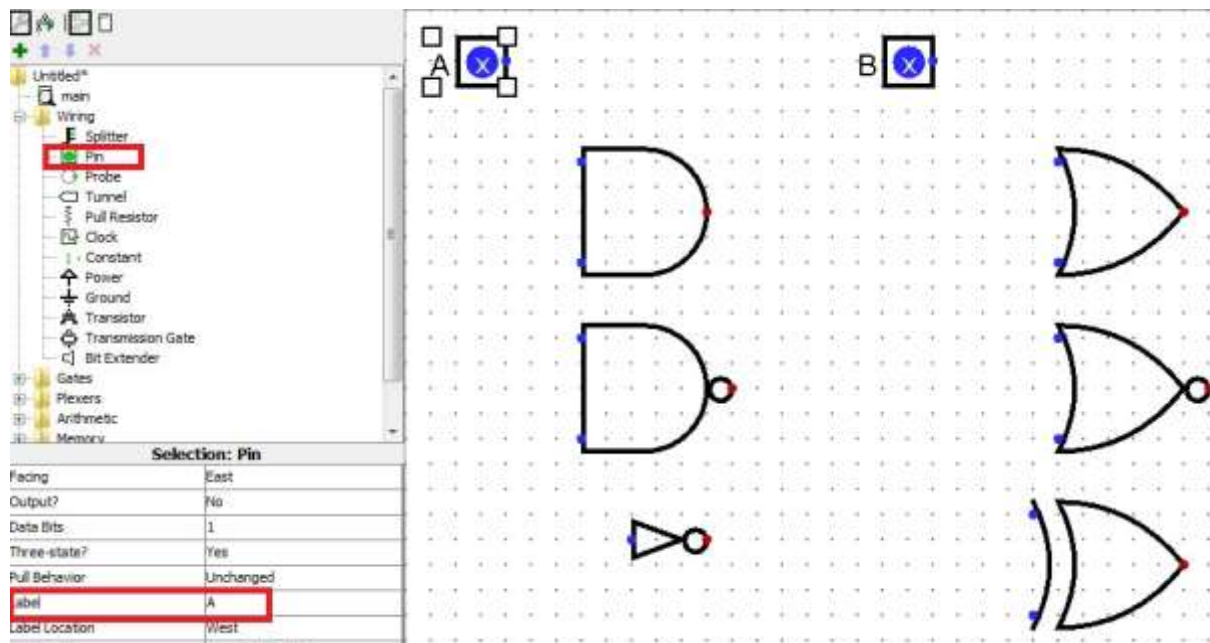


The screenshot shows the Logic Editor interface. On the left, a tree view shows the project structure: 'main' containing 'Wiring' and 'Gates'. The 'Gates' folder is expanded, showing a list of logic gates: NOT Gate, Buffer, AND Gate, OR Gate, Add AND Gate (highlighted with a red box), NOR Gate, XOR Gate, XNOR Gate, Odd Parity, Even Parity, Controlled Buffer, and Controlled Inverter. Below the tree view, a table titled 'Selection: AND Gate' displays the properties of the selected gate.

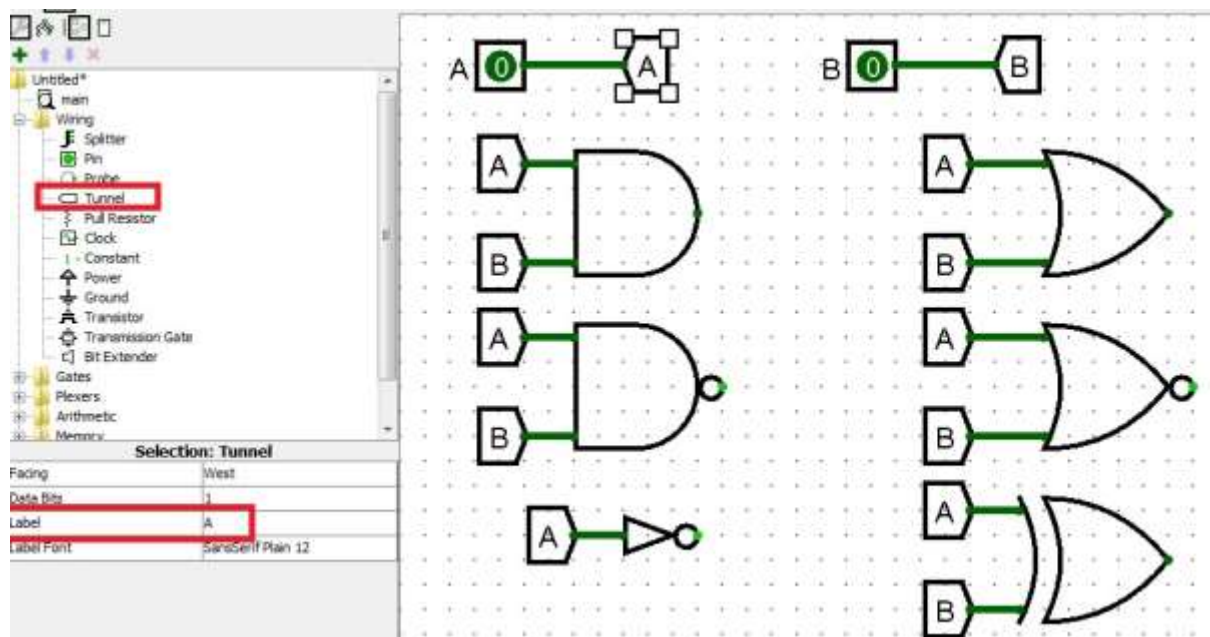
Selection: AND Gate	
Facing	East
Data Bits	1
Gate Size	Medium
Number Of Inputs	2
Output Value	0/1
Label	
Label Font	SansSerif Plain 12



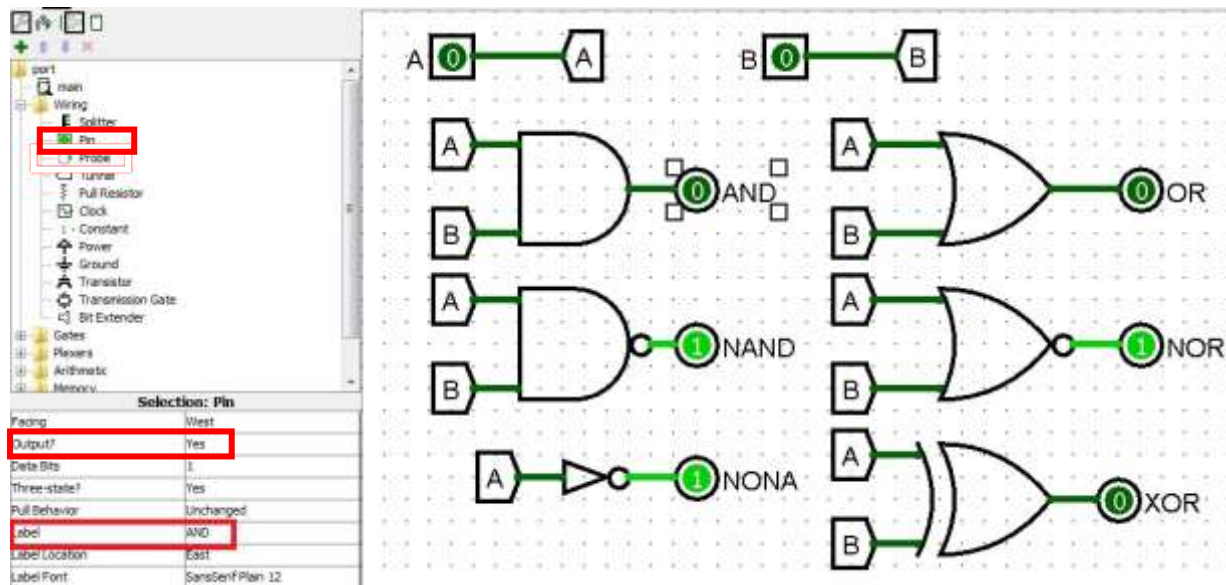
Ajouter 2 entrées « PIN » le premier avec label A et le deuxième B



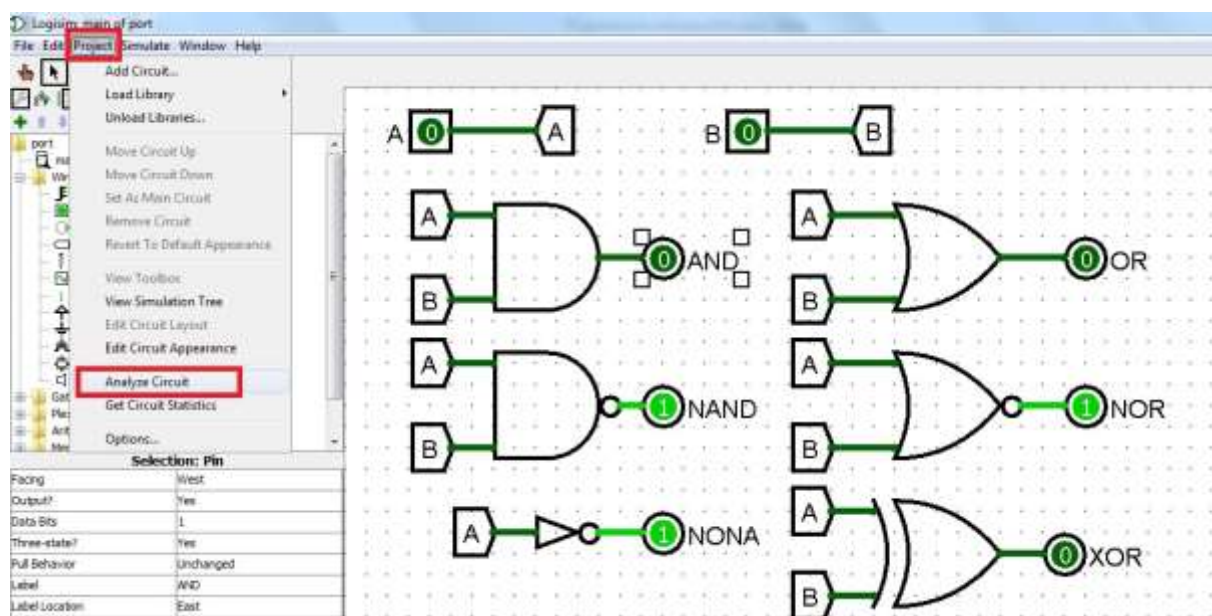
Ajouter plusieurs tunnels A pour lier entrée A avec chaque porte logique et une autre B pour l'entrée B . Conseil : pour changer l'orientation du port réaliser une rotation en choisissant nord, sud , est ou ouest dans l'attribut représentation(facing) de l'objet



Ajouter pour chaque porte logique une sortie (Pin) avec comme label le nom de la porte logique. Par exemple AND pour la sortie du port AND



Après cliquer sur projet puis analyse circuit



En cliquant sur l'onglet Table, le logiciel affiche la table de vérité de tous les ports

Combinational Analysis

File Edit Project Simulate Window Help

Inputs Outputs Table Expression Minimized

A	B	AND	OR	NAND	NOR	NONA	XOR
0	0	0	0	1	1	1	0
0	1	0	1	1	0	1	1
1	0	0	1	1	0	0	1
1	1	1	1	0	0	0	0

Build Circuit

Question 4

Simplifier l'expression suivante grâce à la méthode algébrique :

$$f(a, b, c) = (a + b + c)(a + b + \bar{c})(a + \bar{b} + \bar{c})(\bar{a} + b + c)(\bar{a} + \bar{b} + c)$$

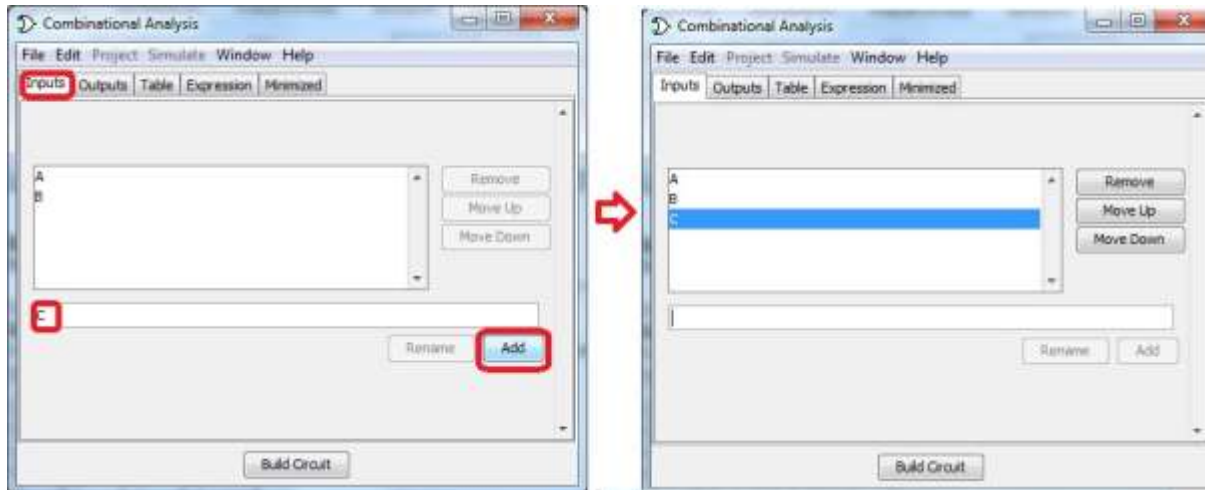
Ecrire la table de vérité équivalente puis les étapes de la question 5

Sciences de l'ingénieur

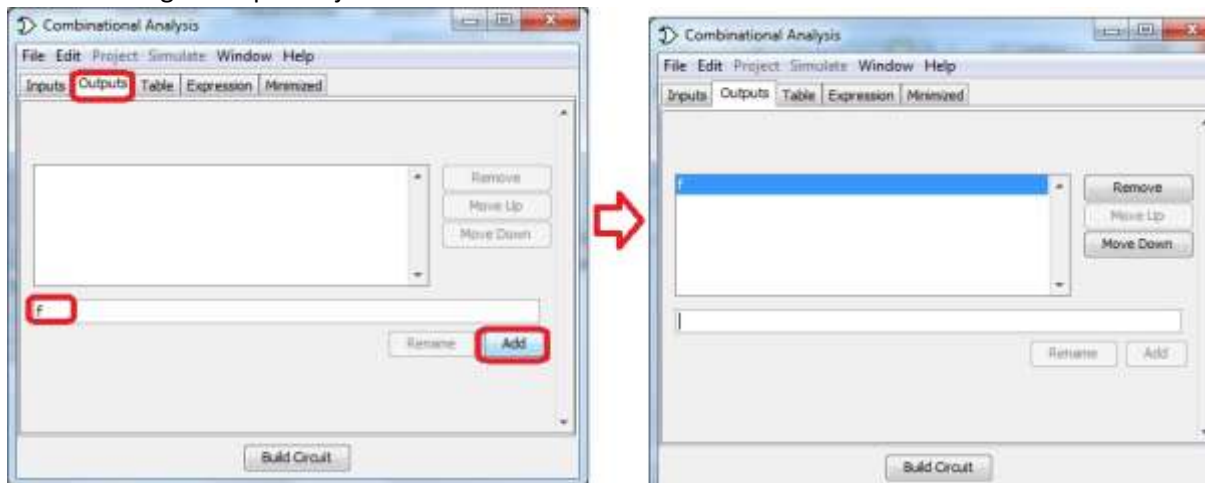
Question 5

Ouvrir nouveau fichier Après cliquer sur projet puis analyze circuit Dans

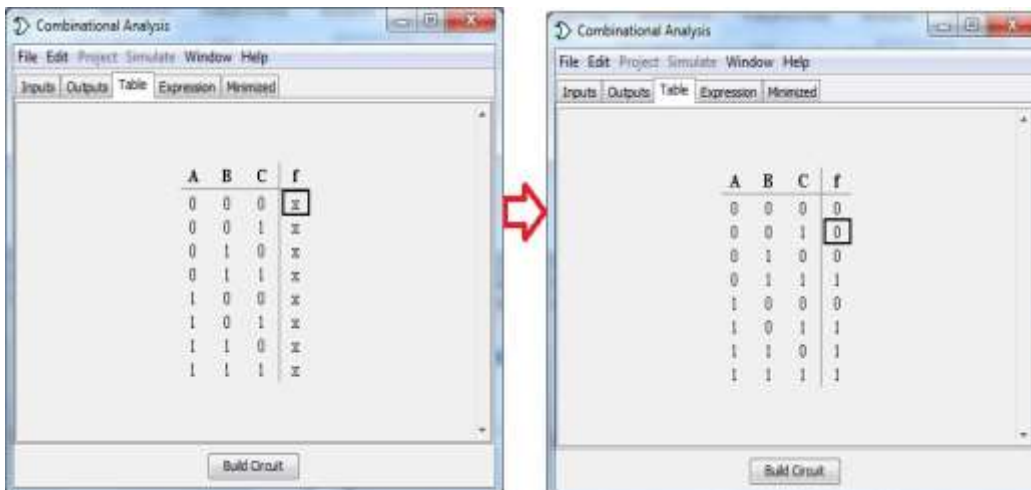
l'onglet inputs rajouter une variable d'entrée « c »



Puis dans l'onglet outputs rajouter une variable de sortie « f »

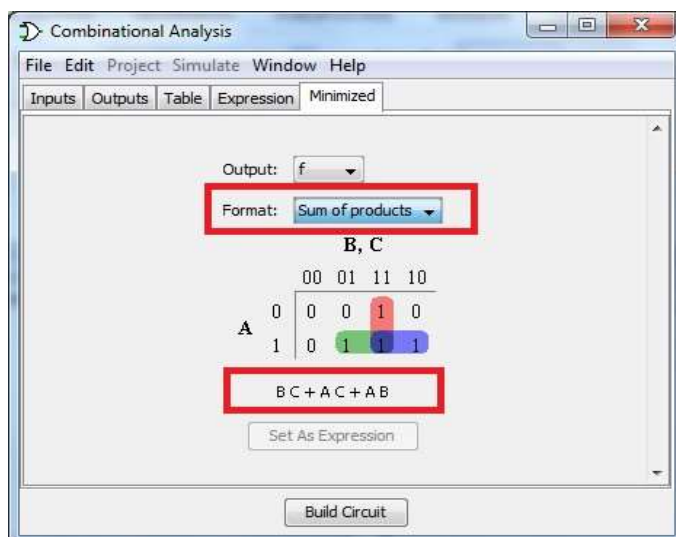


Dans l'onglet table cliquer sur x plusieurs fois dans la colonne f pour reproduire la table de vérité de la question 4.

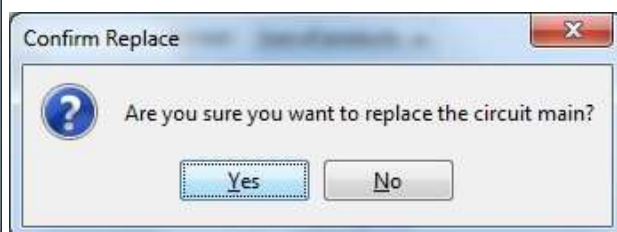
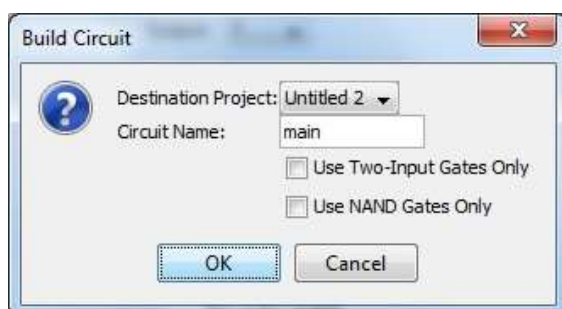


A	B	C	f(a,b,c)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

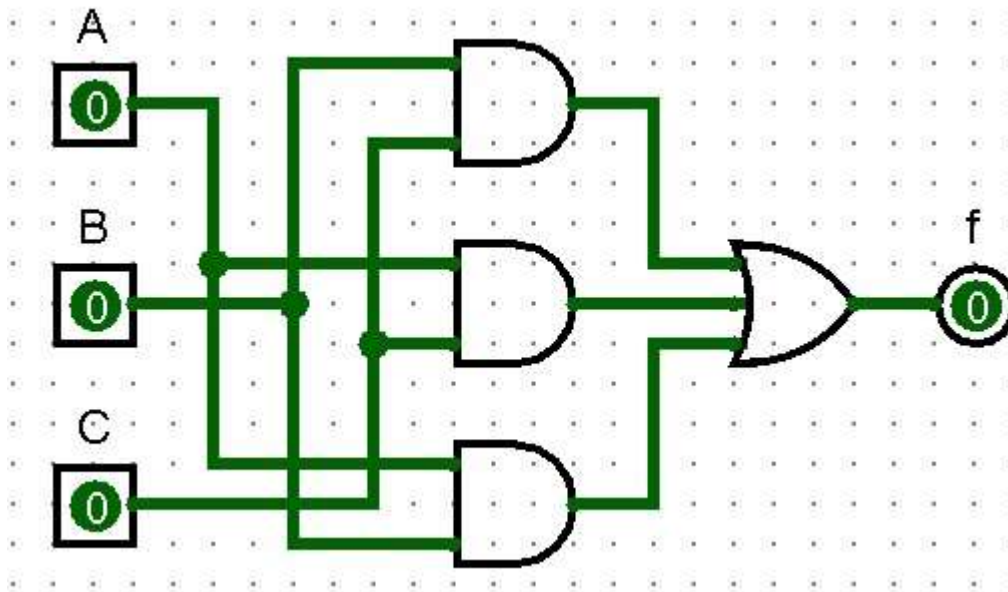
(a) simplification des tables de Karnaugh : cliquer sur l'onglet Karnaugh choisir sum of product



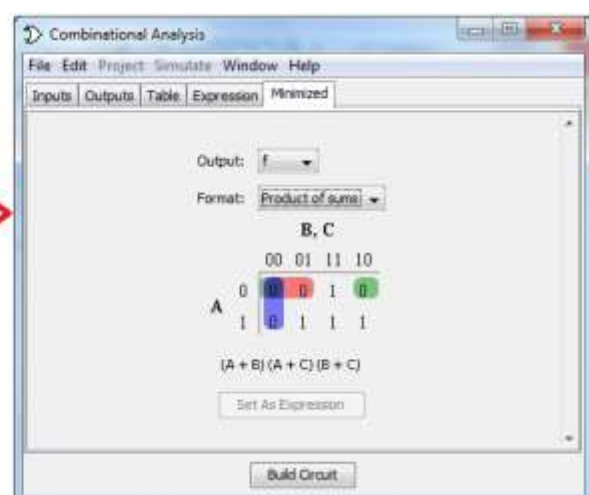
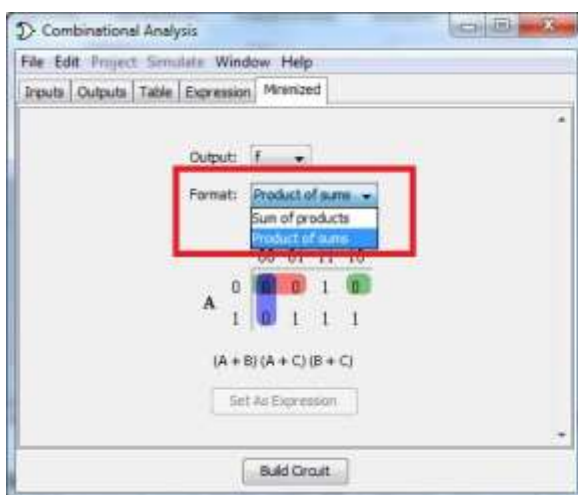
Cliquer sur build circuit puis OK puis yes



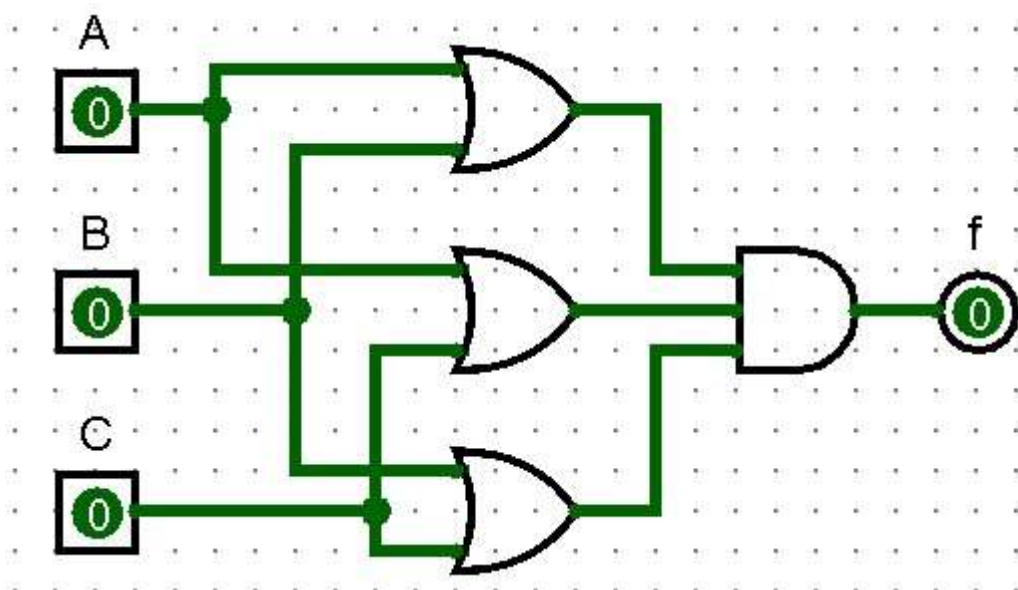
Le logiciel propose le logigramme de solution



(b) Simplification des tables de Karnaugh « maxterms » clique sur l'onglet Karnaugh puis choisir product of sum



Clique sur build circuit puis OK puis yes
Le logiciel propose le logigramme de solution



Circuits combinatoires

Question 9 (vérifier avec logisim) Utilisation des circuits logiques.

Nous souhaitons utiliser un circuit logique pour "convertir" un nombre décimal (de 0 à 3) en nombre binaire. Pour cela, nous utiliserons 4 entrées, actives de façon exclusive, pour symboliser la valeur du nombre décimal à convertir. Faire la synthèse de ce circuit. Quel est son nom ?

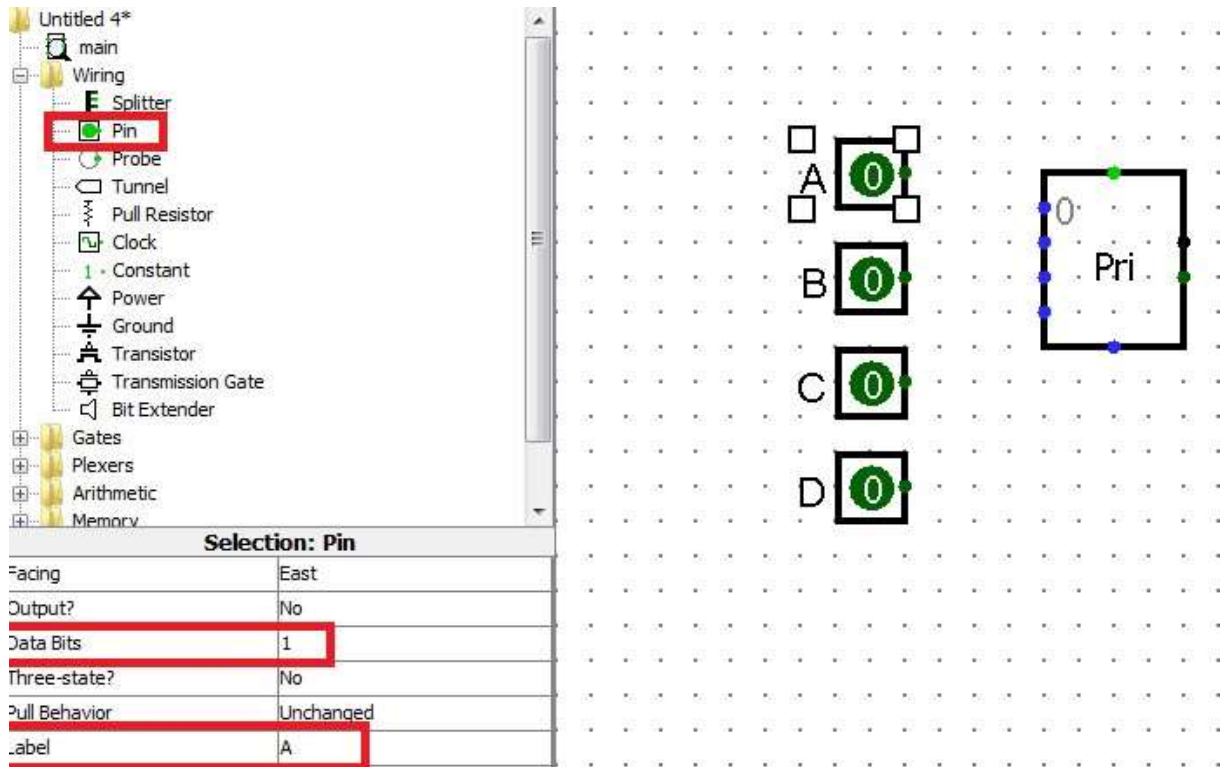
Ouvrir nouveau fichier puis ajouter un circuit « Priority Encoder » avec l'attribut select bits à 2

Untitled 4*

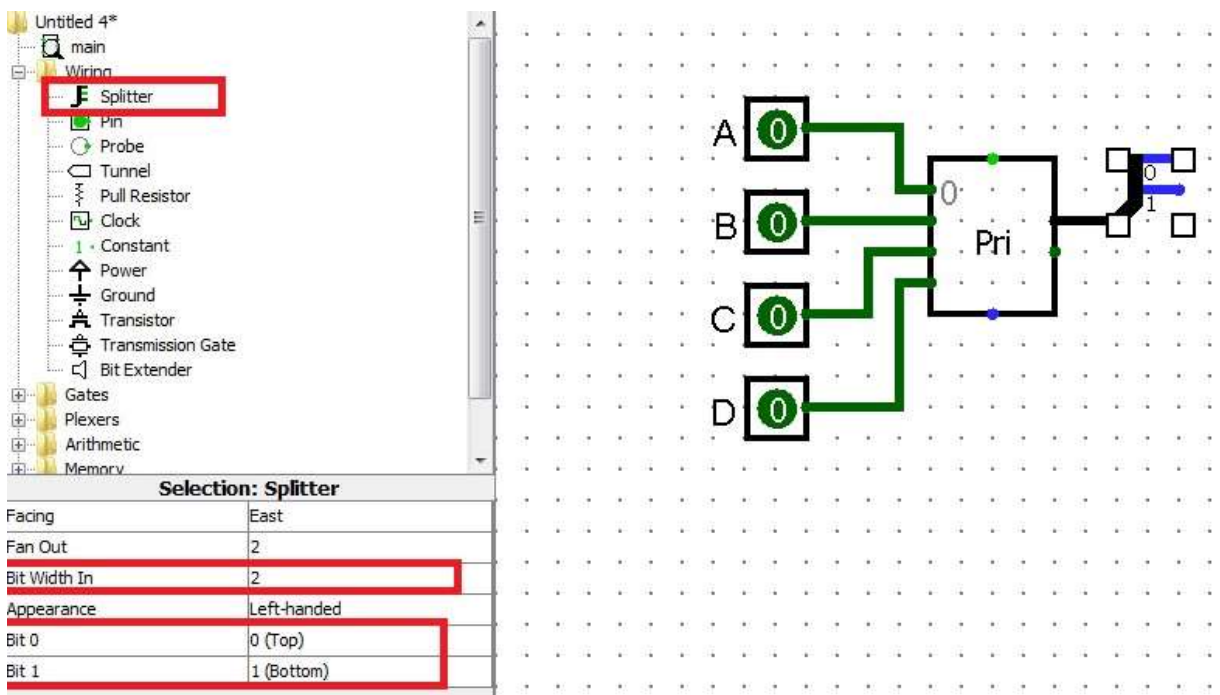
- main
- Wiring
- Gates
- Plexers
- Multiplexer
- Demultiplexer
- Decoder
- Priority Encoder**
- Bit Selector
- Arithmetic
- Memory
- Input/Output
- Base

Selection: Priority Encoder	
Facing	East
Select Bits	2
Disabled Output	Floating

Ajouter 4 entrées (PIN) A,B,C,D

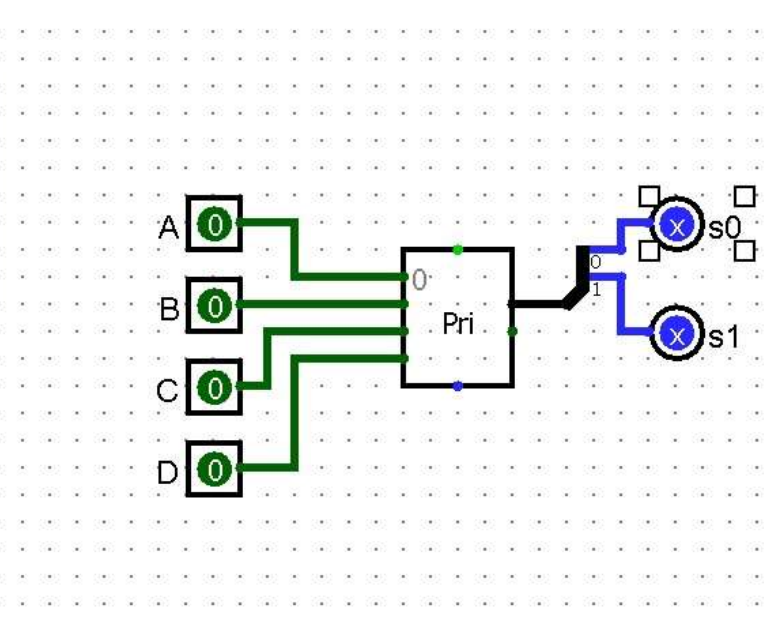
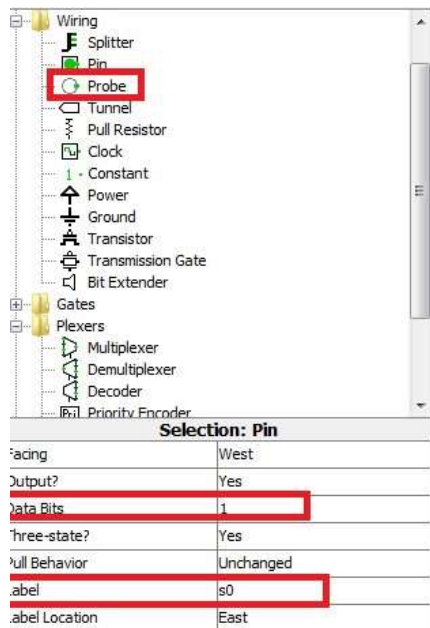


Ajouter un splitter pour pouvoir sélectionner chacun des bits de sortie

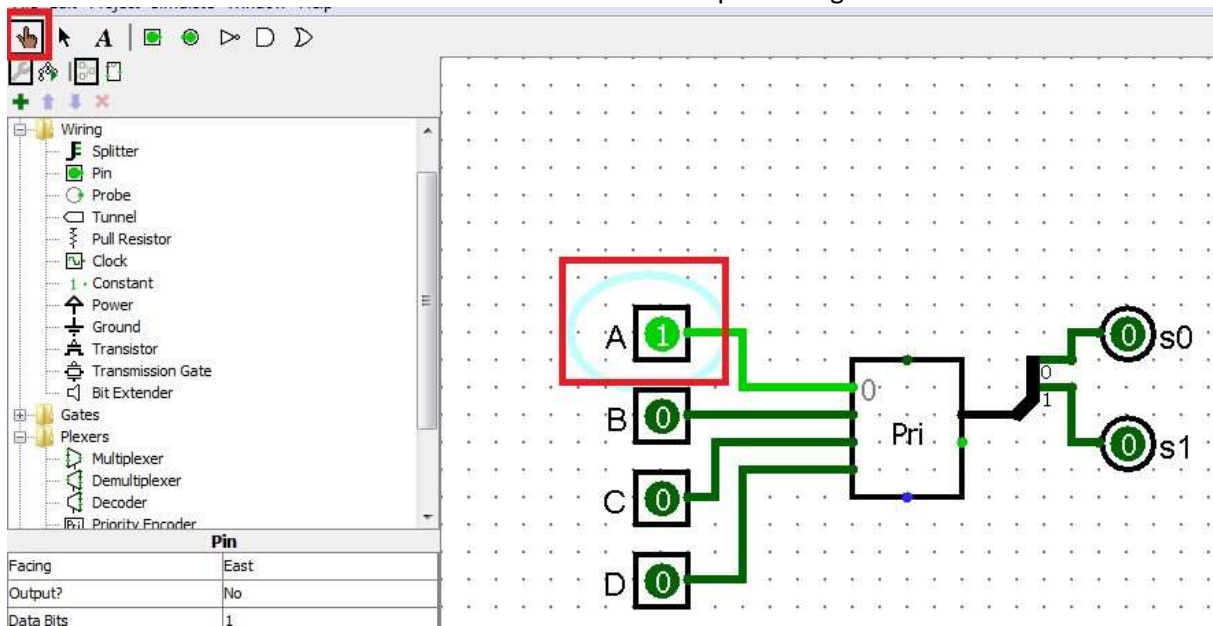


Ajouter 2 sorties S0 et S1

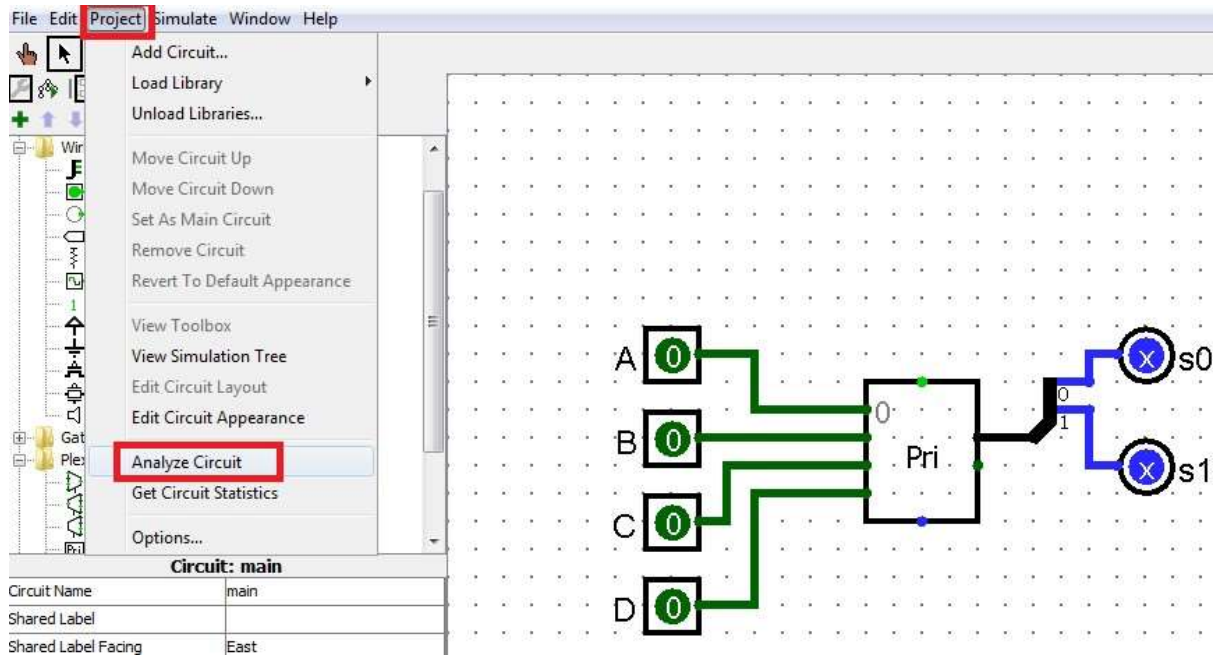
Sciences de l'ingénieur



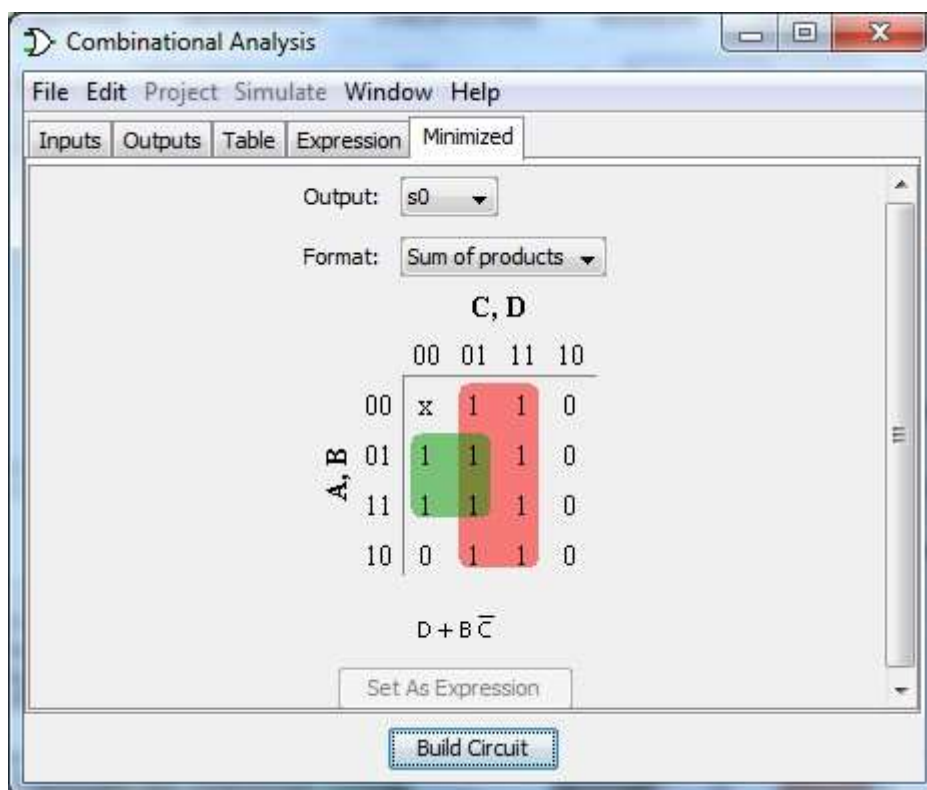
Vérifier le fonctionnement du circuit avec un clic sur la main puis changer les combinaisons



Pour réaliser ce circuit avec les portes logiques ; cliquer sur Project puis Analyze circuit



Puis choisir l'onglet Karnaugh



Cliquer sur Build circuit puis Vérifier le fonctionnement du circuit avec un clic sur la main puis changer les combinaisons

