

TP architecture ordinateur 2020

TP2 : 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. Le premier chapitre 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.

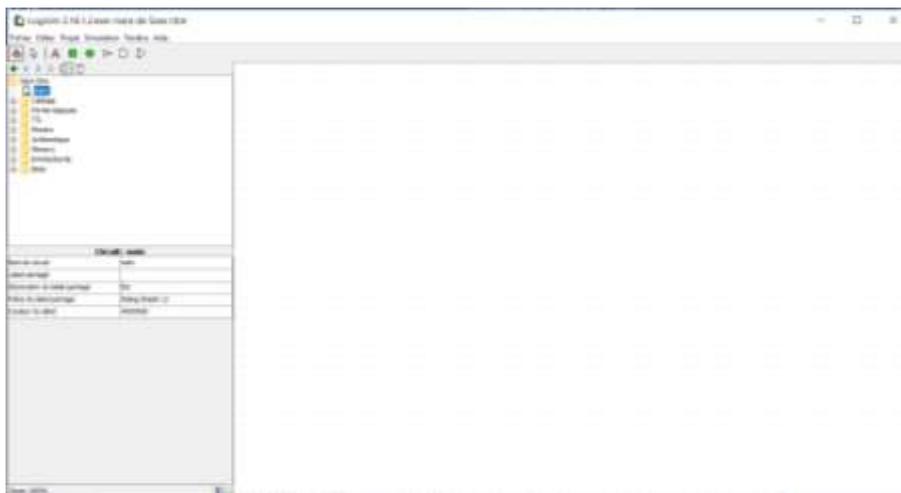
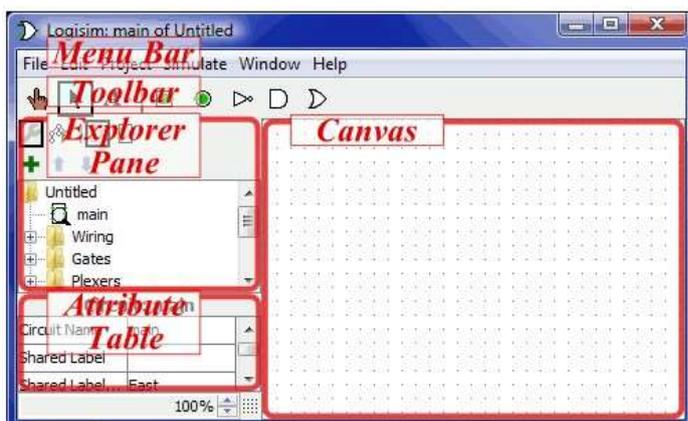


FIGURE 1. Interface de Logisim

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*.



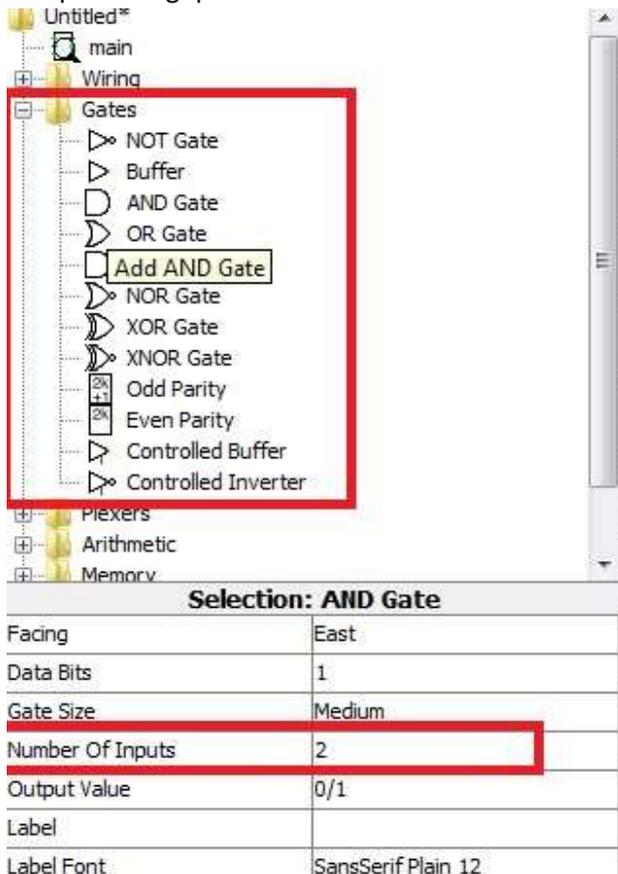
Vous pouvez rapidement jeter un œil sur les détails du panneau de navigation et la table des attributs : nous ne les examinerons pas en détail dans ce tutoriel, et vous pouvez juste les ignorer. Par ailleurs, la barre de menu s'explique d'elle-même.

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.

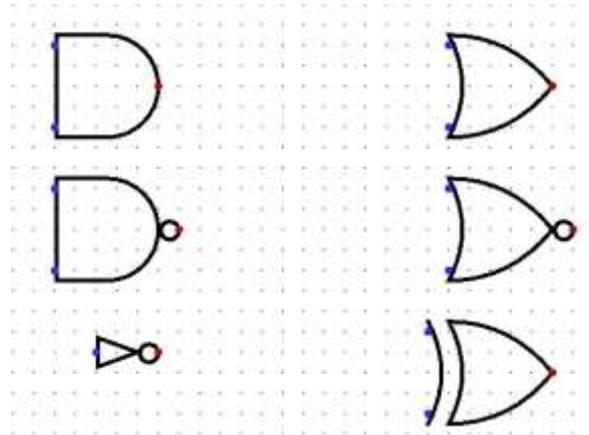
Question 1

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

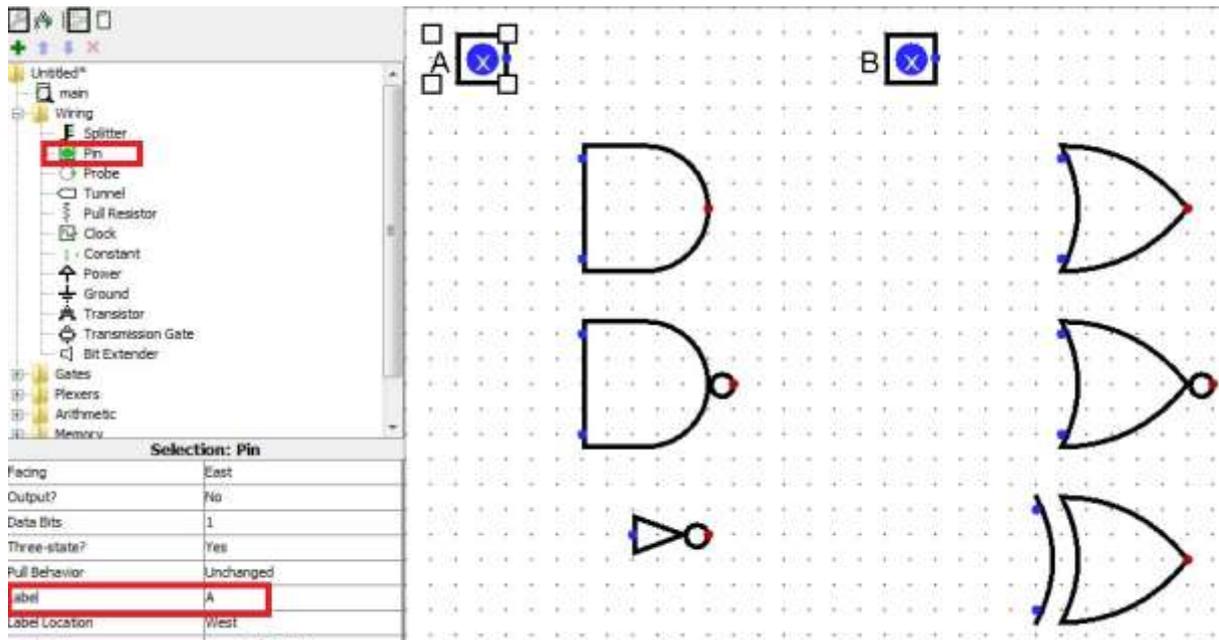
La première chose que nous allons faire c'est d'ajouter les circuits logiques puis modifier le nombre de portes logiques et leurs combinaisons.



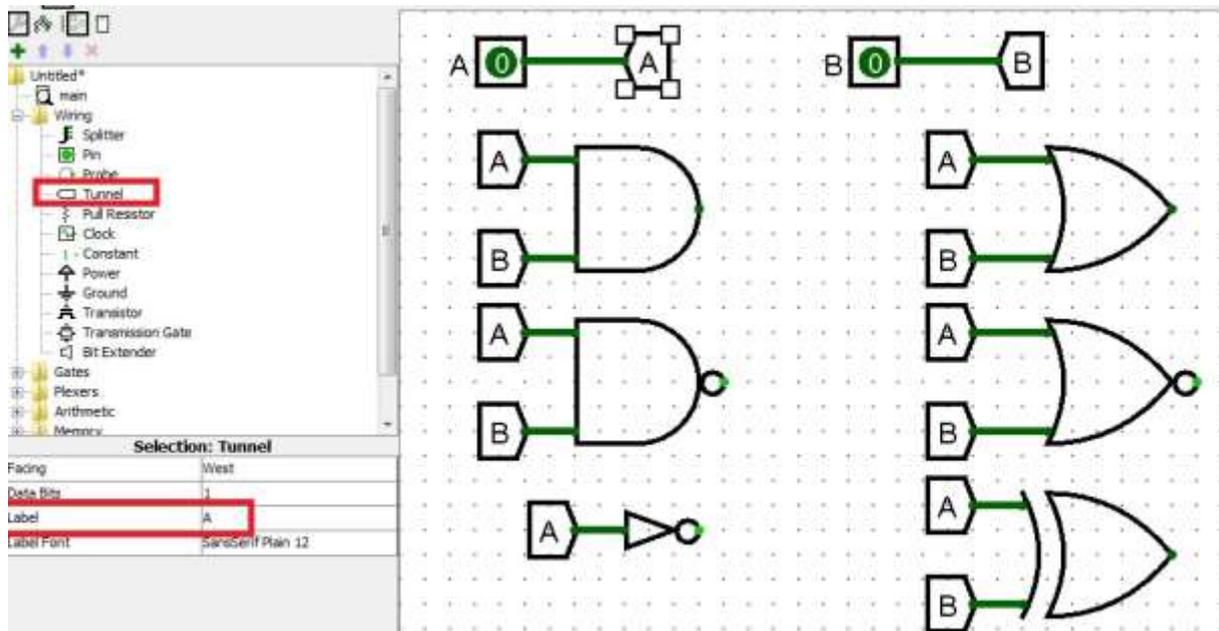
Selection: AND Gate	
Facing	East
Data Bits	1
Gate Size	Medium
Number Of Inputs	2
Output Value	0/1
Label	
Label Font	SansSerif Plain 12



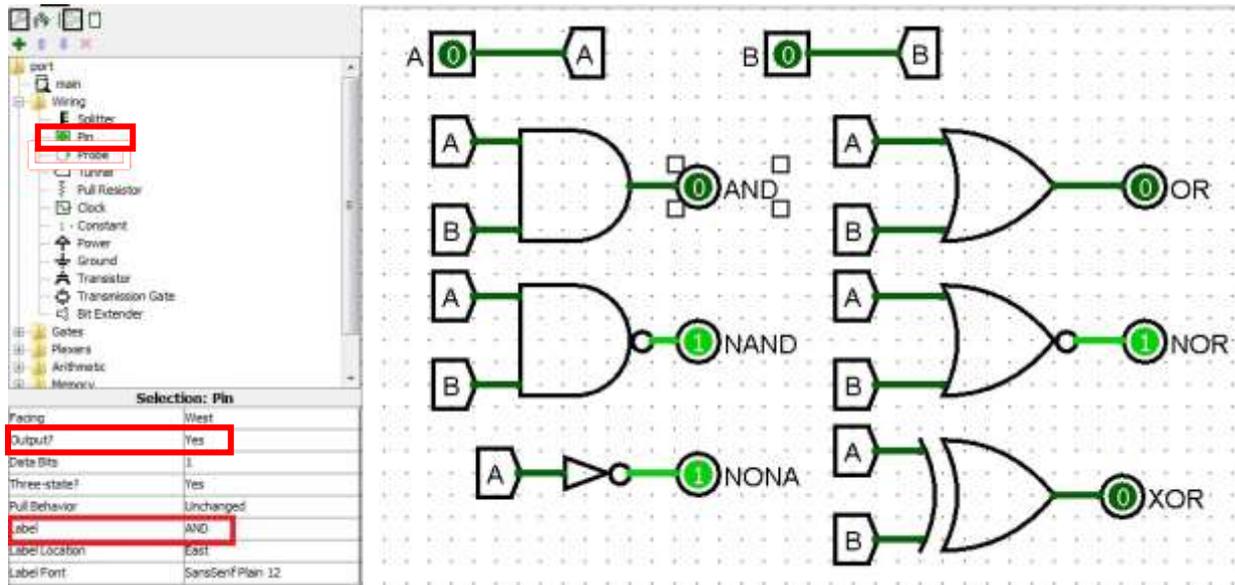
Ajouter 2 entrées « PIN » le premier avec label A et le deuxième B



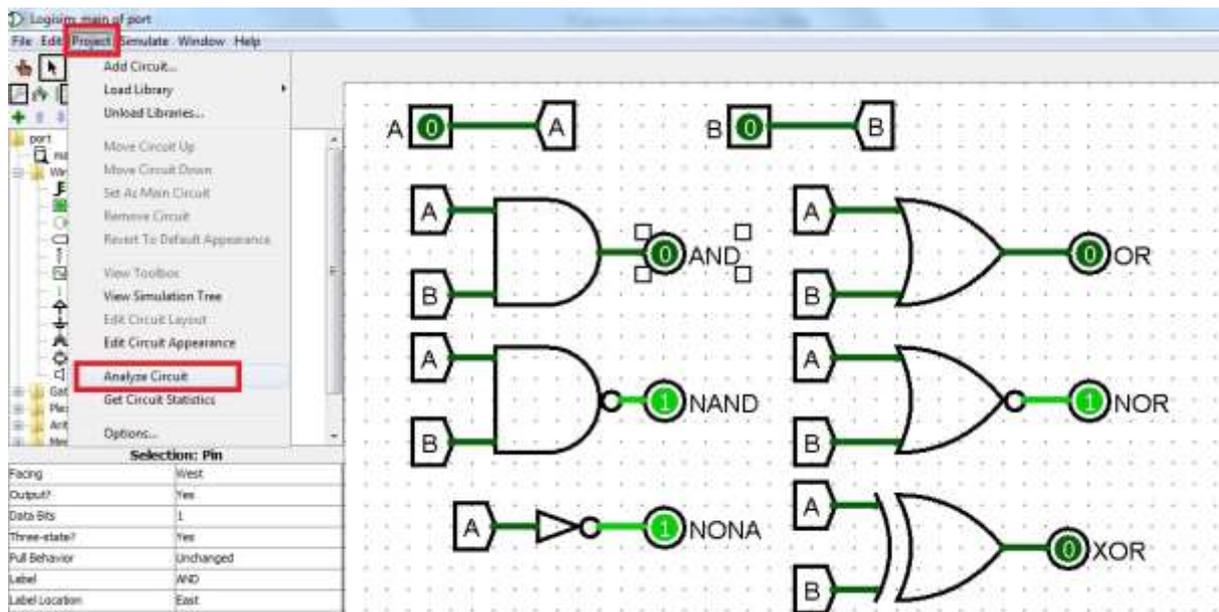
Ajouter plusieurs tunnels A pour lier entrée A avec chaque porte logique et une autre B pour l'entrée B . Conseil : pour changer l'orientation du port réaliser une rotation en choisissant nord, sud , est ou ouest dans l'attribut représentation(facing) de l'objet



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

The 'Combinational Analysis' window displays a truth table for the circuit. The table has columns for inputs A and B, and outputs for AND, OR, NAND, NOR, NONA, and XOR. The data is as follows:

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

Question 4

Simplifier l'expression suivante grâce à la méthode algébrique :

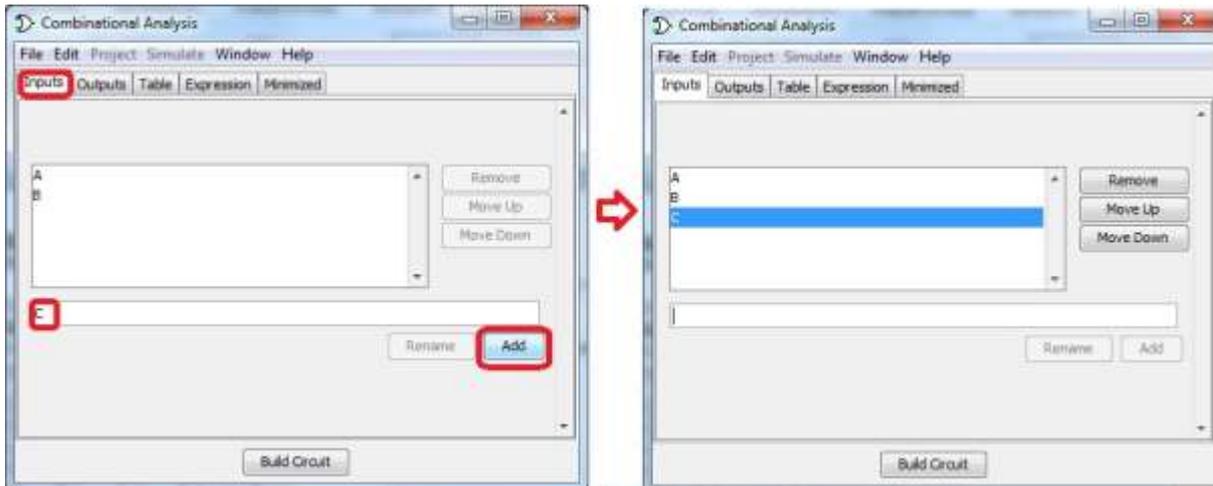
$$f(a, b, c) = (a + b + c)(a + b + \bar{c})(a + \bar{b} + \bar{c})(\bar{a} + b + c)(\bar{a} + \bar{b} + c)$$

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

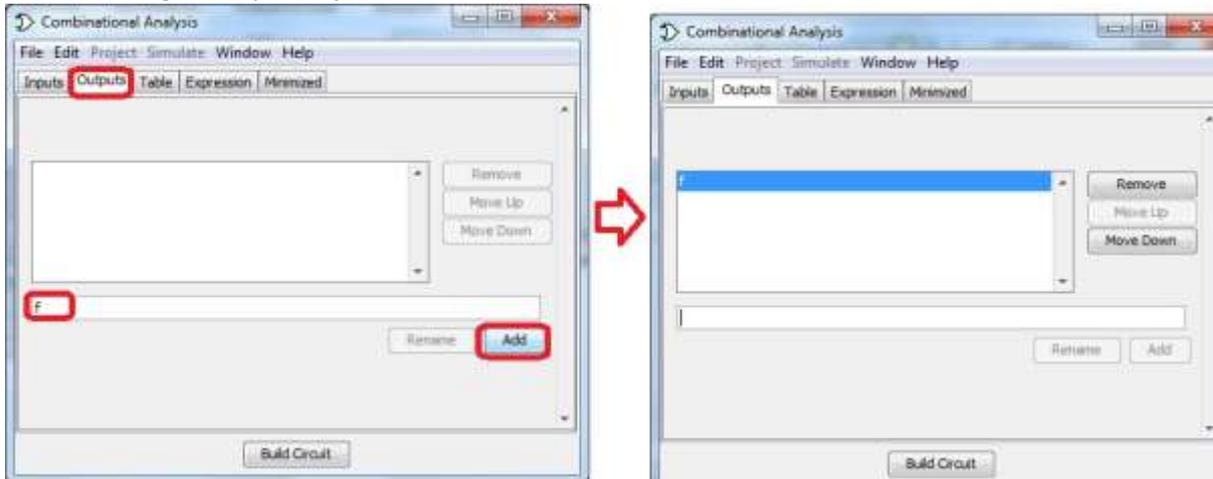
Question 5

Ouvrir nouveau fichier Après cliquer sur projet puis analyze circuit Dans

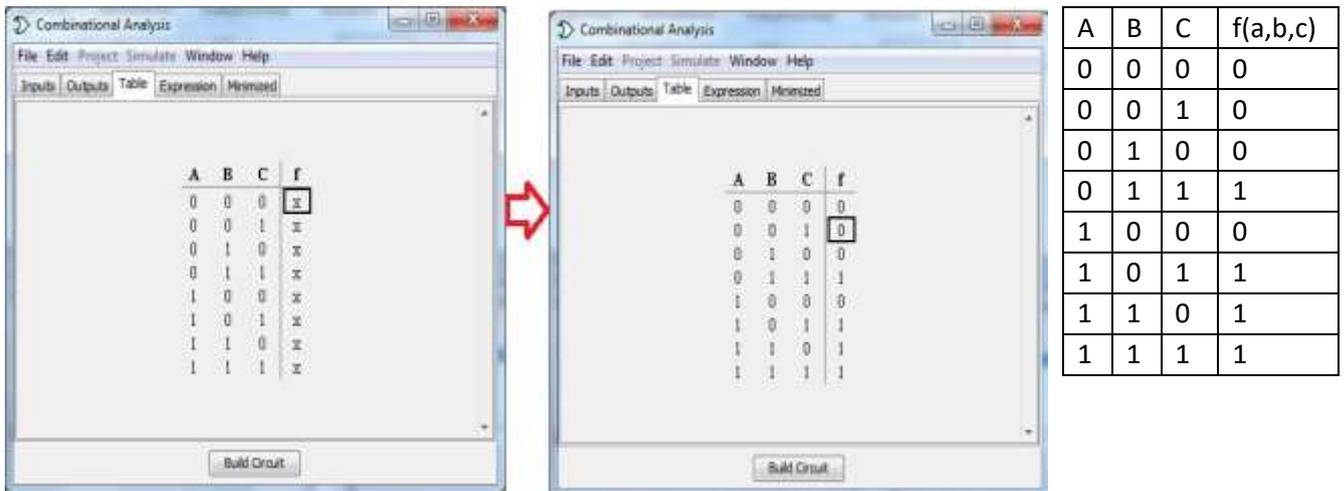
l'onglet inputs rajouter une variable d'entrée « c »



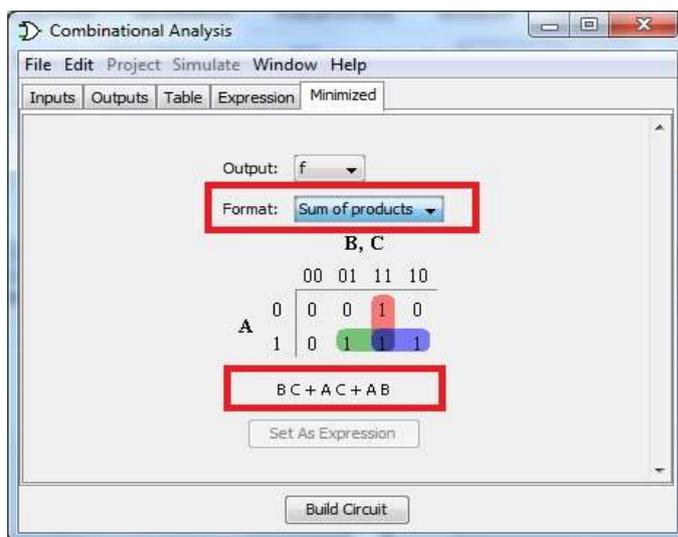
Puis dans l'onglet outputs rajouter une variable de sortie « f »



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



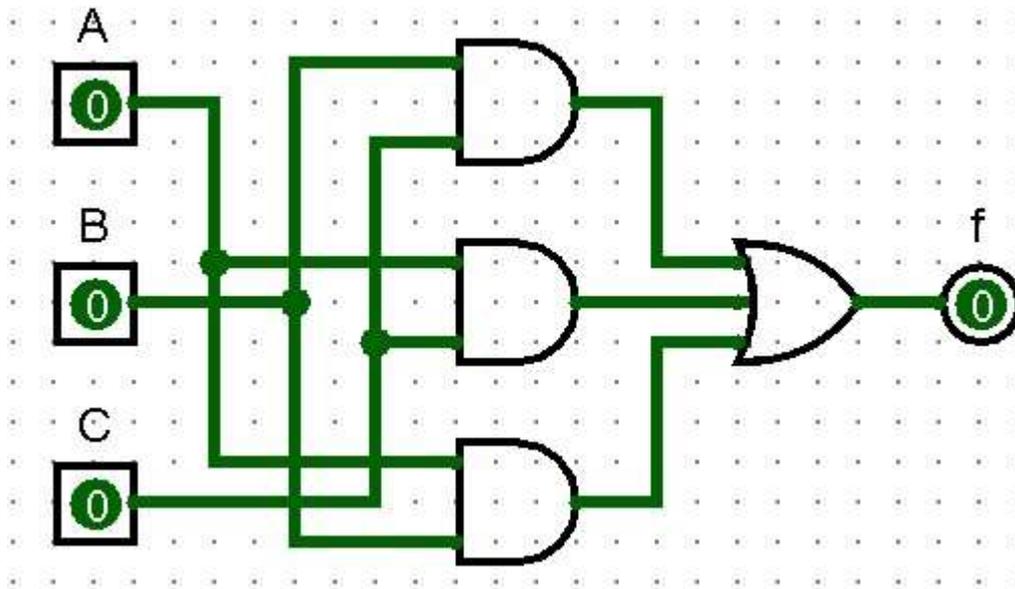
(a) simplification des tables de Karnaugh : cliquer sur l'onglet Karnaugh choisir sum of product



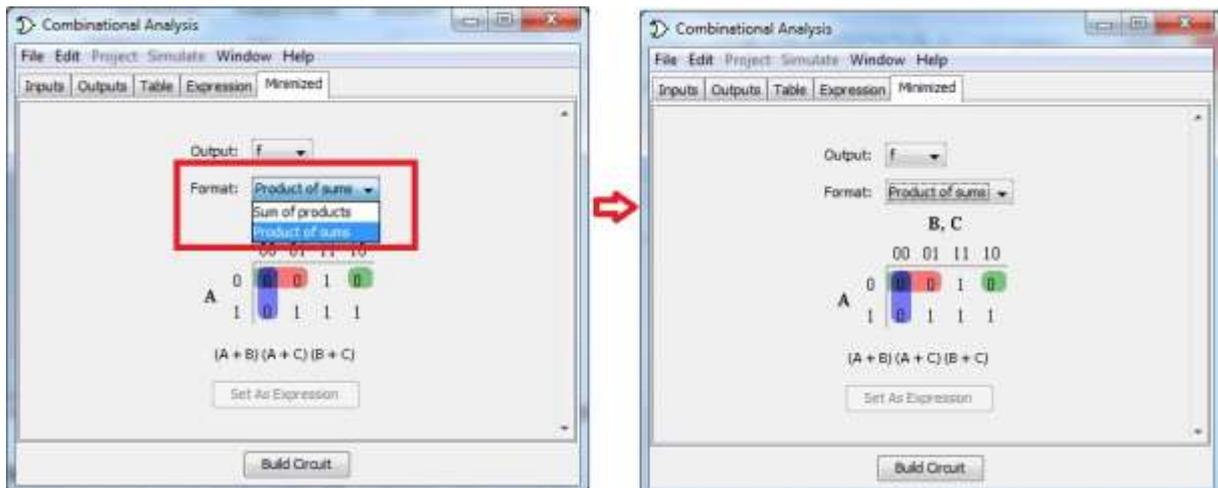
Cliquer sur build circuit puis OK puis yes



Le logiciel propose le logigramme de solution

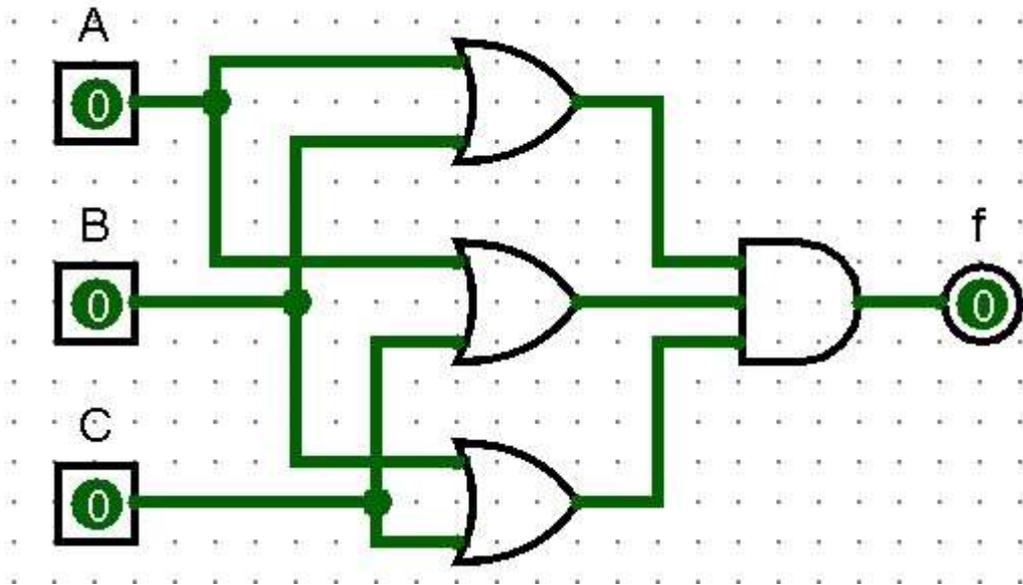


(b) Simplification des tables de Karnaugh « maxterms » clique sur l'onglet Karnaugh puis choisir product of sum



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

Selection: Pin

Facing	East
Output?	No
Data Bits	1
Three-state?	No
Pull Behavior	Unchanged
Label	A

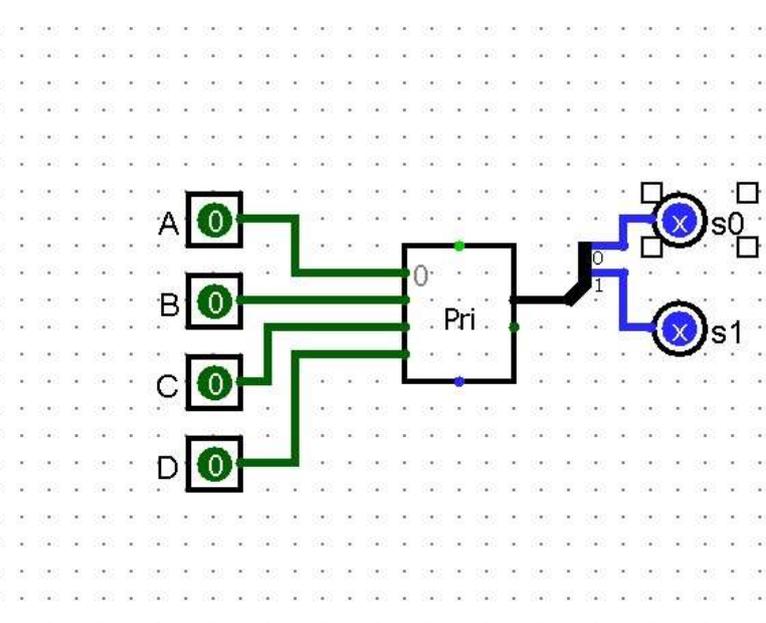
Ajouter un splitter pour pouvoir sélectionner chacun des bits de sortie

Selection: Splitter

Facing	East
Fan Out	2
Bit Width In	2
Appearance	Left-handed
Bit 0	0 (Top)
Bit 1	1 (Bottom)

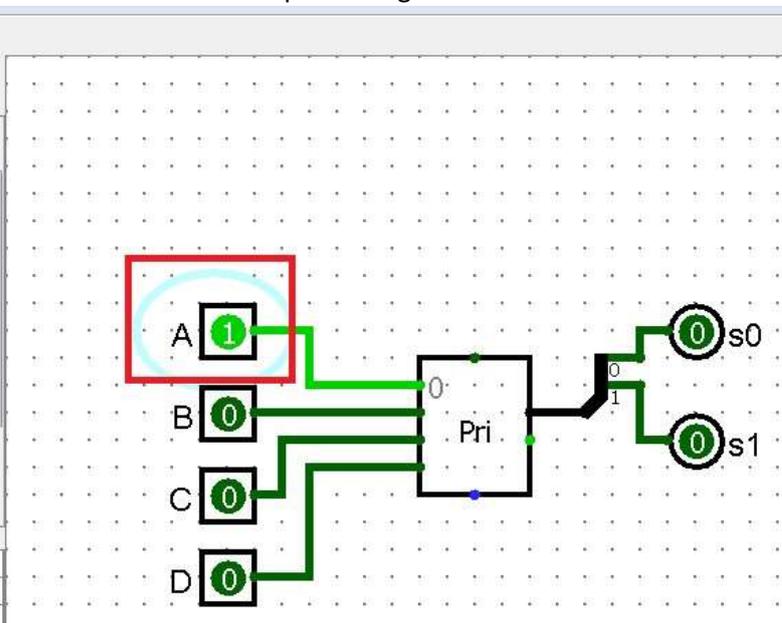
Ajouter 2 sorties S0 et S1

Selection: Pin	
Facing	West
Output?	Yes
Data Bits	1
Three-state?	Yes
Pull Behavior	Unchanged
Label	s0
Label Location	East

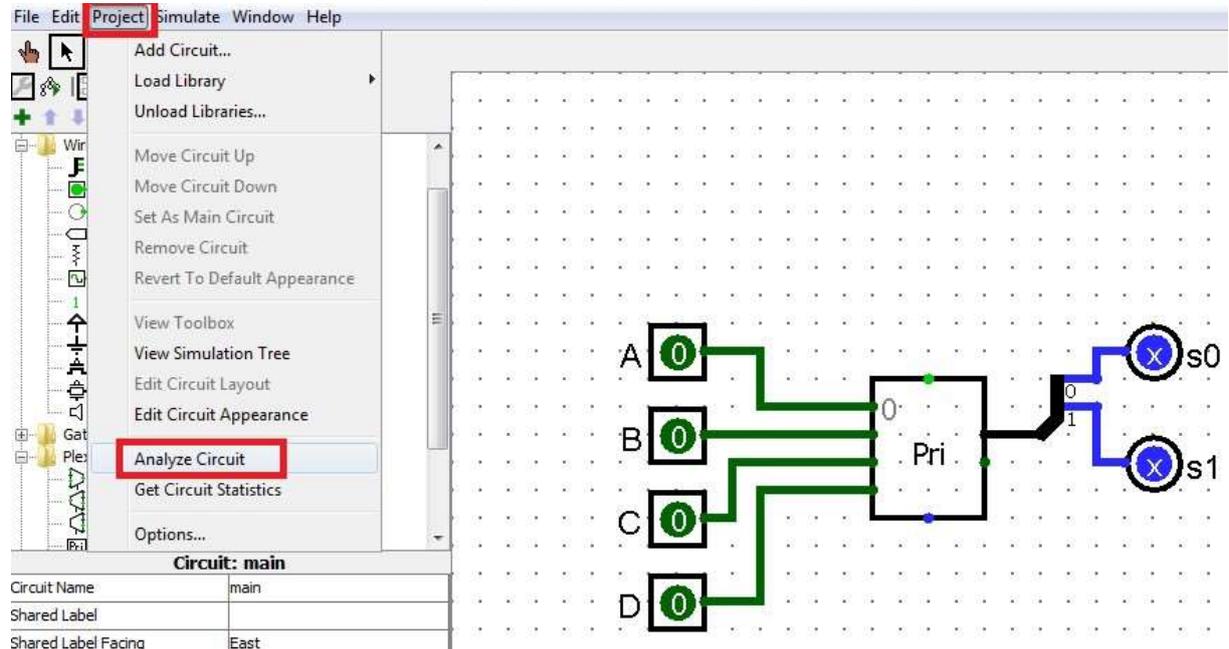


Vérifier le fonctionnement du circuit avec un clic sur la main puis changer les combinaisons

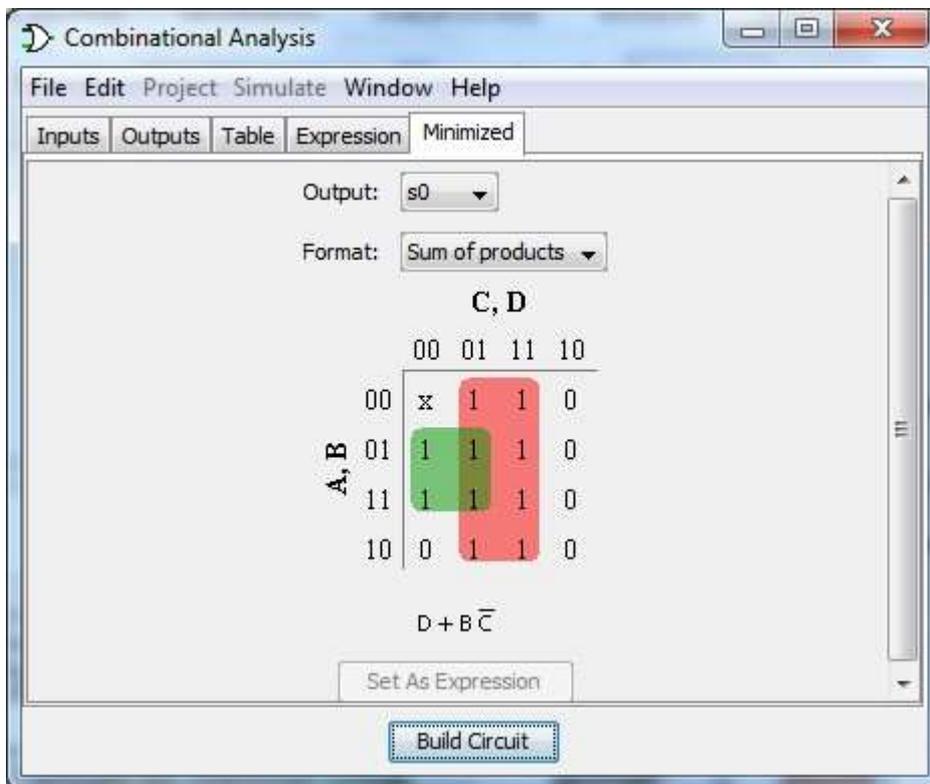
Pin	
Facing	East
Output?	No
Data Bits	1



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

