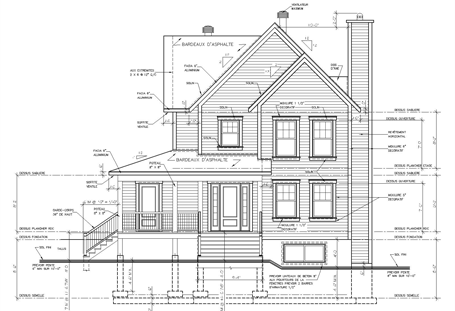
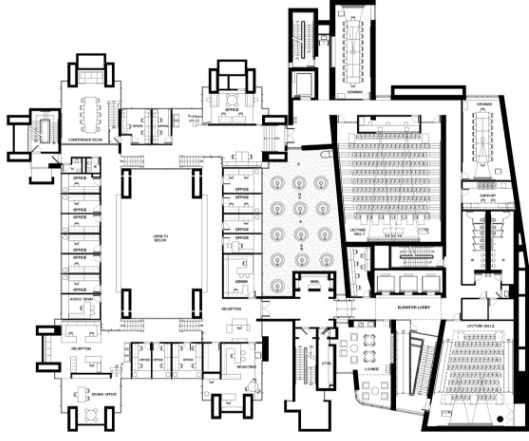
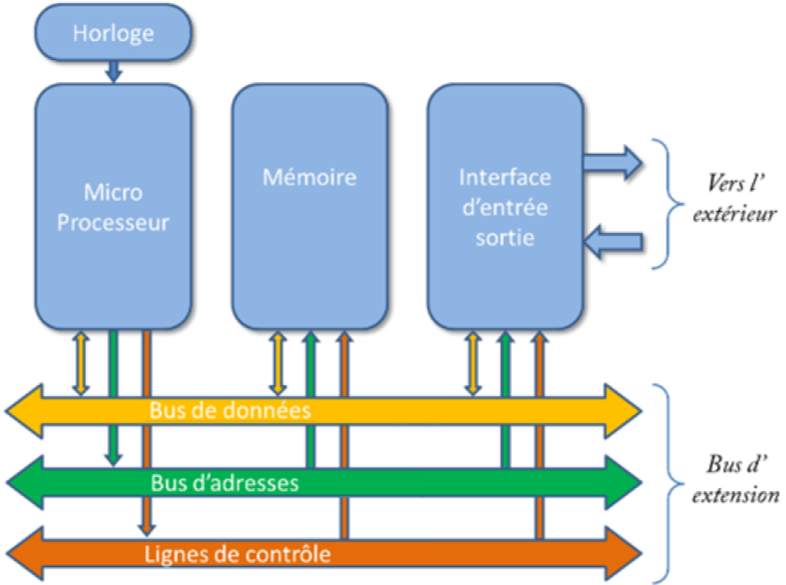
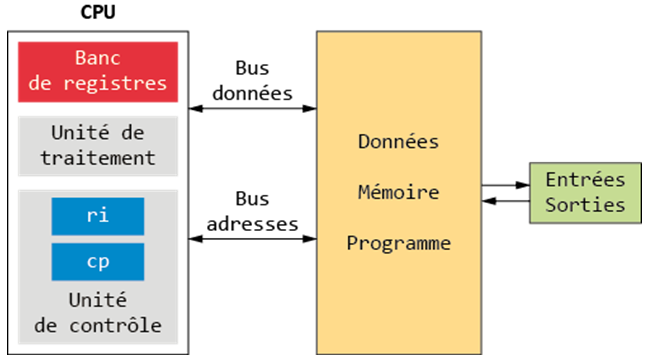
|  |  |  |
| --- | --- | --- |
| Numérique et Sciences Informatiques | | |
| 1h30 | **Architecture de Von Neumann** |  |
| Objectif : Comprendre l'architecture des ordinateurs. | | |
| **Matériel :** | | |

Architecture d'un bâtiment : plans



Architecture d'une machine informatique : organisation entre le processeur, la mémoire et les entrées/sorties

L'architecture des ordinateurs est conforme à un schéma qui a peu évolué depuis son origine en 1945, le modèle créé par **John Von Neumann** :



**Un peu d'histoire**

En 1945, à l'université de Pennsylvanie (USA), les ingénieurs américains John Eckert et John Mauchly achèvent la construction du premier calculateur numérique électronique programmable connu : l'E.N.I.A.C. (Electronic Numerical Integrator and Computer), initialement destiné à remplacer un service de calcul mécanique de tir d'artillerie.

Volumineux (30 tonnes et 30 m3), utilisant 18 000 tubes électroniques, l'E.N.I.A.C. est de cent à mille fois plus rapide que les appareils électromécaniques.

Inauguré en 1946 et utilisé jusqu'en 1955, il sert d'abord à l'armée américaine pour des calculs balistiques, puis il aide à résoudre des problèmes de physique nucléaire et de météorologie.

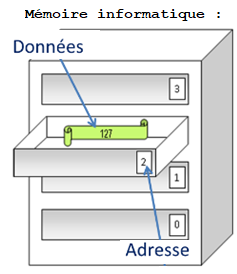
Le mathématicien [John Von Neumann](https://www.universalis.fr/encyclopedie/john-von-neumann/) comprend que pour améliorer le rendement de cet appareil, il faut remplacer la [programmation](https://www.universalis.fr/encyclopedie/programmation/) externe (cartes perforées) par des programmes enregistrés dans une mémoire interne de la machine, John Von Neumann publie, en 1946, un rapport décrivant précisément cette future machine, ce que nous appelons aujourd'hui un ordinateur.

De retour à l'Institute for Advanced Studies de l'université de Princeton, il met en chantier l'ordinateur I.A.S. (initiales de son institut), pendant que Eckert et Mauchly construisent le Binac (Binary Automatic Computer), puis l'Univac-I (Universal Automatic Computer).

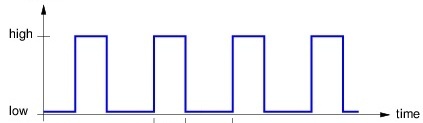




**Le modèle de Von Neumann est constitué de quatre parties distinctes**

* Le **CPU** (Central Processing Unit), plus communément appelé le **processeur**.
* La **mémoire** (mémoire vive) où sont stockés les données et les programmes.
* Les **entrées-sorties** (I/O : Input/Output) pour échanger avec l'extérieur.
* Les **bus** (fils électriques) pour relier les différents composants.

**Echange des informations dans l'architecture de Von Neumann**

* Les échanges entre la **mémoire** et les **registres** (cases mémoires du **CPU**) se font par les **bus** selon une chronologie cadencée par une **horloge** (= signal électrique de forme rectangulaire).
* Un programme est enregistré dans la **mémoire**.
* L'**adresse** (un nombre entier codé en binaire) de l'instruction en cours de traitement est stockée dans un registre interne au CPU (processeur) nommée **CP ou PC** (**P**rogram **C**ounter).
* La valeur de cette instruction (un entier codé en binaire) est stockée dans un autre registre interne au CPU nommée **RI ou IR** (**I**nstruction **R**egister).
* Le CPU dispose aussi d'un **banc de registres** (petite mémoire interne) pour placer les données en cours d'exécution.
* **L'unité de traitement** **ALU** (**A**rithmetic **L**ogic **U**nit) effectue les opérations arithmétiques (+, -, x, / ...) et logiques (AND, OR ...) sur les données.

**Cadencement du processeur**

Le **CPU** dispose d'une **horloge** qui cadence l'exécution des instructions.

La fréquence d'horloge d'un circuit numérique synchrone est la fréquence du signal d'horloge qui le cadence (nombre de **cycles par seconde**). Cette information est indiquée en [hertz](https://fr.wikipedia.org/wiki/Hertz) (Hz).

Lorsqu'on parle d'un processeur cadencé à 3 GHz, cela signifie qu'il y a 3 milliards de cycles par seconde.

Jusqu'en 2004 la fréquence des processeurs augmentait linéairement mais depuis elle stagne.

En effet la chaleur produite par la rapidité de travail du processeur risque de détériorer ses circuits électroniques (température maximale du silicium des transistors est d'environ 175°C).

Depuis 2004 les constructeurs augmentent le nombre de cœurs présents sur un CPU pour augmenter ses performances (un cœur est principalement composé d'une ALU et d'une unité de contrôle).

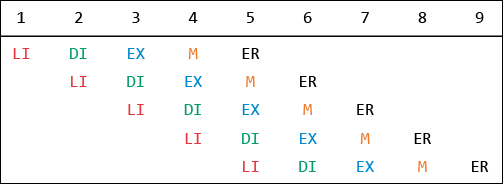
**Cycle**

Dans un processeur chacune des **5 actions** suivantes est exécutée lors d'**1 cycle** :

* LI : lire l'instruction en assembleur.
* DI : décoder l'instruction en code machine.
* EX : exécuter l'opération dans l'ALU.
* M : accéder à la mémoire en lecture ou en écriture si besoin.
* ER : écrire le résultat dans les registres de l'ALU.

Pour gagner du temps, le processeur n'exécute pas les instructions de manière **séquentielle** (les unes après les autres) car il est capable de réaliser **simultanément** des actions qui ne monopolisent pas les même ressources. C'est le principe du **pipeline d'instructions**.

Exemple : l'exécution de 5 instructions demande 9 cycles.



→ pour un processeur cadencé à 3 GHz → 3 milliards de cycles/s → 3 milliards x 5/9 = 1,66 milliards d'instruction/s exécutées.

**Les différentes mémoires informatiques présentent dans un PC**

Les **mémoires vives** (**registres**, **mémoire centrale** et **mémoire cache**) : leurs contenus est perdu lorsque le composant n'est plus alimenté en énergie électrique.

Un **registre** est un emplacement mémoire interne au processeur permettant de stocker des opérandes (données qui entrent dans une opération arithmétique) et des résultats intermédiaires lors des opérations effectuées dans l’ALU.

NB : La plupart des PC actuels ont des registres de 64 bits (32 bits sur la génération précédente).

La **mémoire centrale** (**RAM lente**) contient les programmes en cours et les données qu’ils manipulent.

RAM : Random Access Memory = mémoire à accès direct.

RAM lente : SDRAM.

NB : sur un PC le programme à exécuter sera chargé du disque dur vers la mémoire centrale.

La **mémoire cache** (**RAM rapide**) permet de stocker les instructions et données en cours d’utilisation, ce qui augmente la rapidité de traitement.

En effet le processeur, qui est très rapide, va souvent se resservir des données qui viennent d'être utilisées, d'où l'intérêt de les stocker dans une mémoire très rapide d'accès.

RAM rapide : SRAM plus chère que SDRAM.

Les **mémoire persistantes** (**ROM**, **SSD** et **HDD**) : leurs contenus est conservé même lorsque le composant n'est plus alimenté en énergie électrique.

ROM : Read Only Memory

SSD : Solid State Drive.

HDD : Hard Drive Disk = disques magnétiques.

Dans un PC, la ROM contient le **BIOS** (Basic Input/Output System) qui effectue des opérations de bases au démarrage :

* Initialisation de tous les composants de la carte mère et de certains périphériques.
* Identification de tous les périphériques internes et externes.
* Initialisation de l'ordre de priorité de traitement des périphériques d'entrées.
* Démarrage du système d'exploitation présent sur le premier périphérique disponible.
* ...

**Informations :**

Originellement, l'expression **mémoire morte**(ROM) désignait une **mémoire persistante** dont le contenu était fixé lors de sa programmation, elle pouvait être lue par le processeur mais ne pouvait pas être reprogrammée sans procédure spéciale.

Ces mémoires sont les [UVPROM](https://fr.wikipedia.org/wiki/UVPROM), les [PROM](https://fr.wikipedia.org/wiki/PROM), les [EPROM](https://fr.wikipedia.org/wiki/EPROM) et les [EEPROM](https://fr.wikipedia.org/wiki/EEPROM).

**La machine de Turing, l'ancêtre de la machine de Von Neumann**

Vidéo pour expliquer la machine de Turing (10min24) : <https://www.youtube.com/watch?v=Kf-IHn1PvaQ>

Vidéo sur Turing (5min36) : <https://www.youtube.com/watch?v=zHTBHiOy6eY>

**D'autres architectures ...**

Vidéo sur l'ordinateur Quantique (5min29) : <https://www.youtube.com/watch?v=rNdWOXQ8V4A>

Mais ça existe un ordinateur Quantique (2min32) ? :

<https://www.arte.tv/fr/videos/093442-000-A/l-ordinateur-quantique-de-google/>

L'ordinateur Quantique pour aller plus loin (25min54) : <https://www.youtube.com/watch?v=2aCS5mEeiwg>

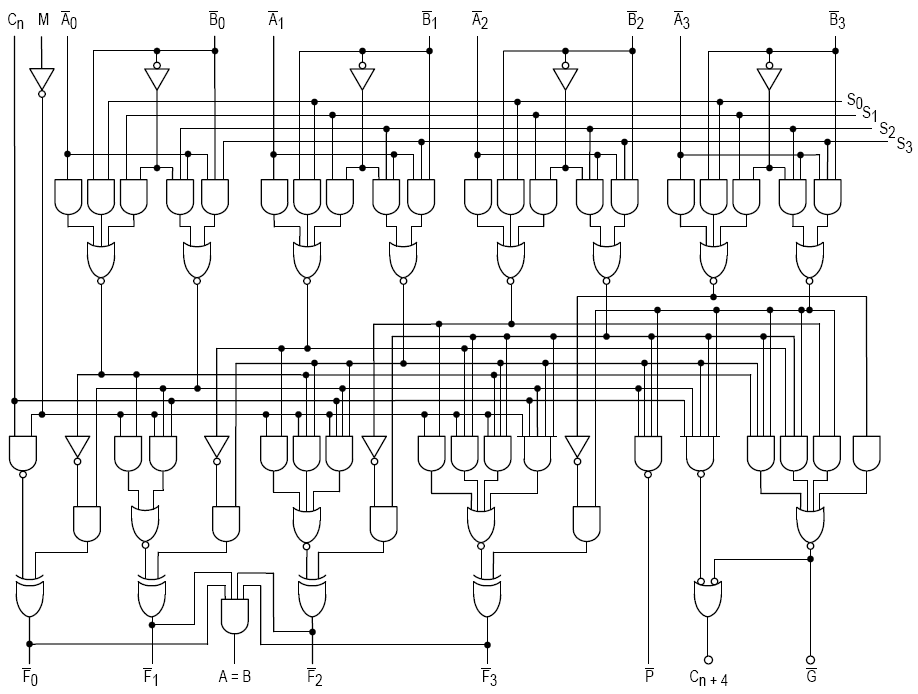
Automate cellulaire (12min48) : <https://www.youtube.com/watch?v=lkdnOuzHdFE>

Automate cellulaire suite (15min47) : <https://www.youtube.com/watch?v=CfRSVPhzN5M>

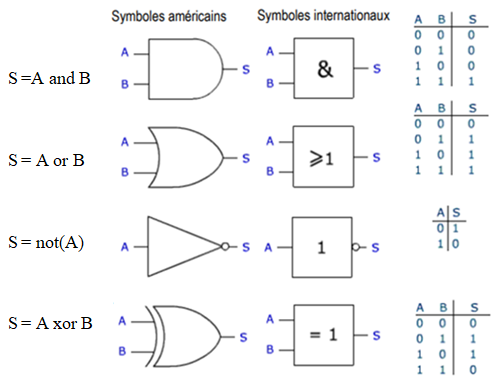
Vidéo calculateur Analogique (0min42) : <https://www.youtube.com/watch?v=gVPKGclBIZQ>

**Les circuits combinatoires**

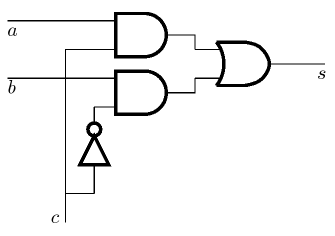
**Une ALU (Arithmetic Logic Unit) est composée de circuits combinatoires, exemple :**



Les circuits combinatoires sont constitués de portes logiques (NOT, AND, OR, XOR) :



Les circuits combinatoires permettent de réaliser des fonctions booléennes (on a déjà vu l'algèbre de Boole).

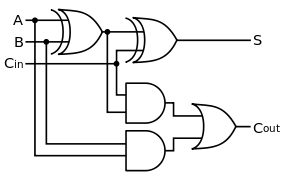
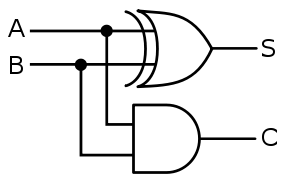
Exemple, le circuit combinatoire suivant permet de réaliser la fonction booléenne suivante :

**S = (a and c) or (b and not(c))**

Le résultat en sortie dépend des entrées.

|  |
| --- |
| Remarque (hors programme) : pour réaliser des circuits séquentiels, les sorties peuvent devenir des entrées, c'est comme cela qu'on réalise des mémoires, exemple d'une mémoire de 1 bit (bascule D) :  Les bascules (Flip-flop) |

**Exemple de l'additionneur :**



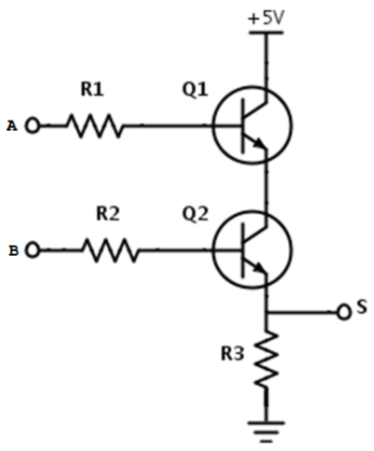
S = A xor B

C = A and B (C représente la retenue de l'addition).

Exemple 1 : 1 + 0 = (01)2 = (1)10 S = 1 xor 0 = 1 et C = 1 and 0 = 0

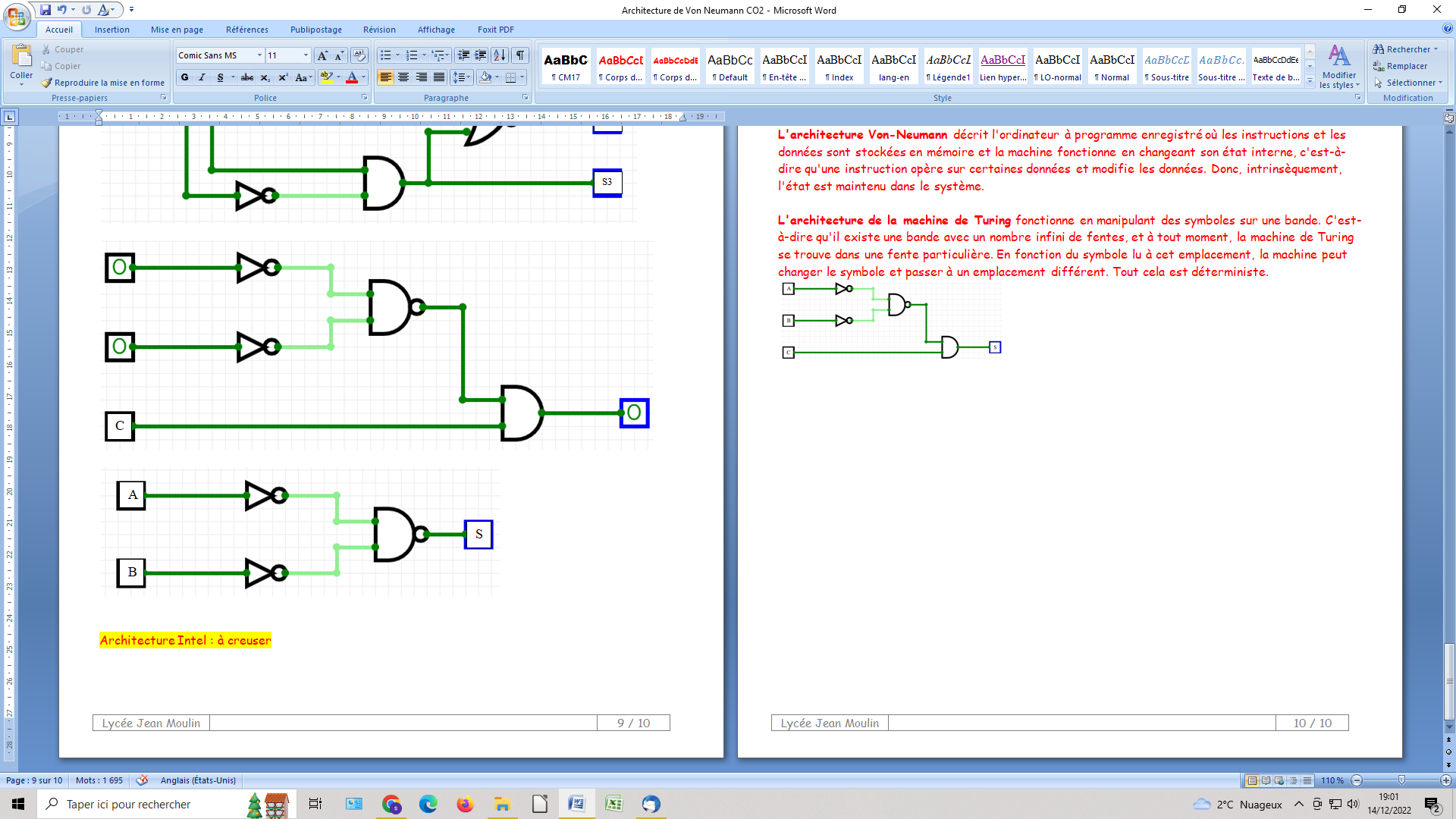
Exemple 2 : 1 + 1 = (10)2 = (2)10 S = 1 xor 1 = 0 et C = 1 and 1 = 1

Les portes logiques sont constituées de transistors, exemple du AND :



Il faut mettre 5V sur les A et B pour que les transistors Q1 et Q2 laissent passer le courant vers la sortie → S = 5V

* Donnez l'équation logique de S avec le circuit suivant :

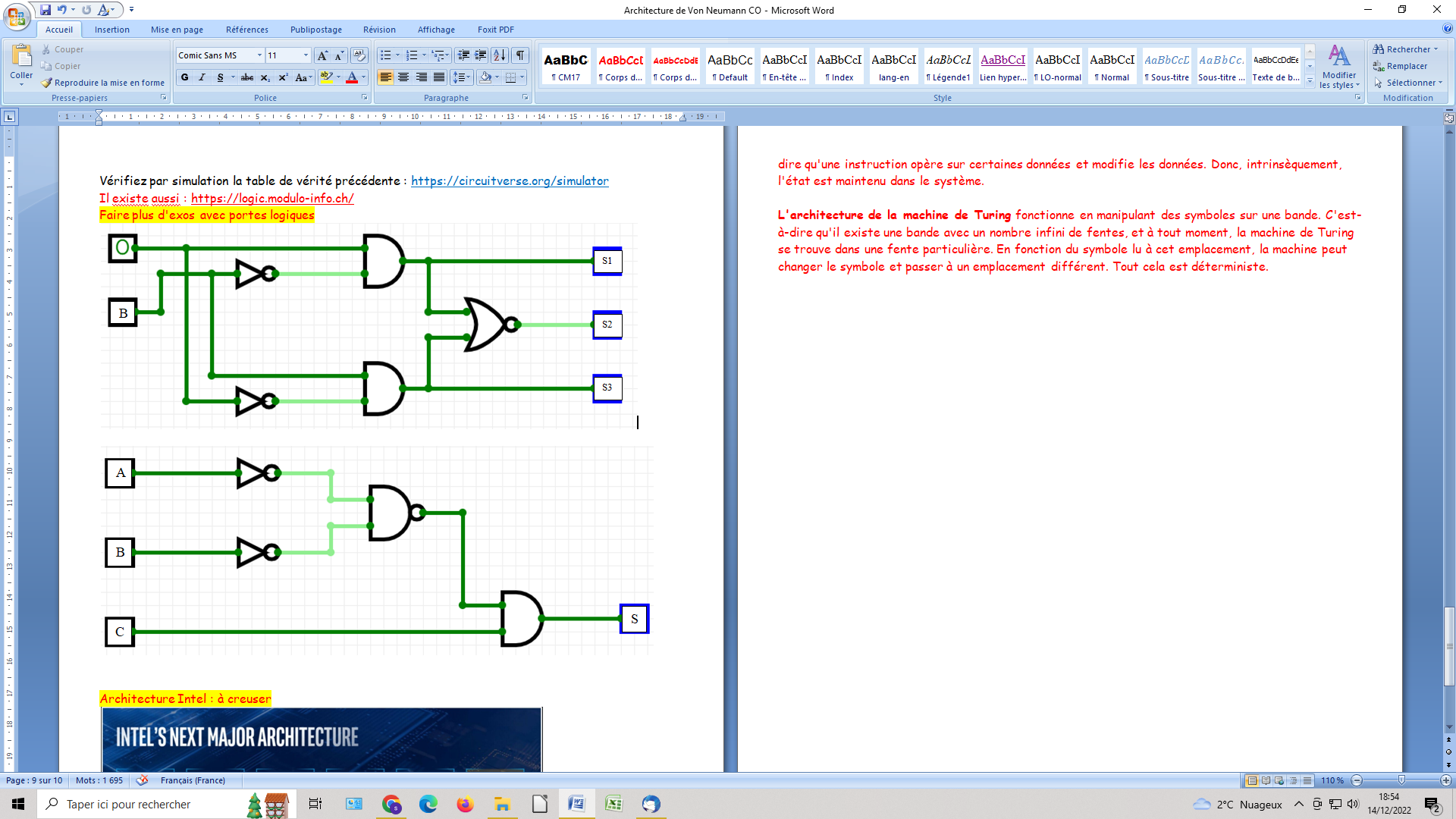


Complétez la table de vérité du circuit précédent :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | not A | not B | S |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |

Vérifiez par simulation la table de vérité précédente : https://circuitverse.org/simulator

* Donnez l'équation logique de S avec le circuit suivant :

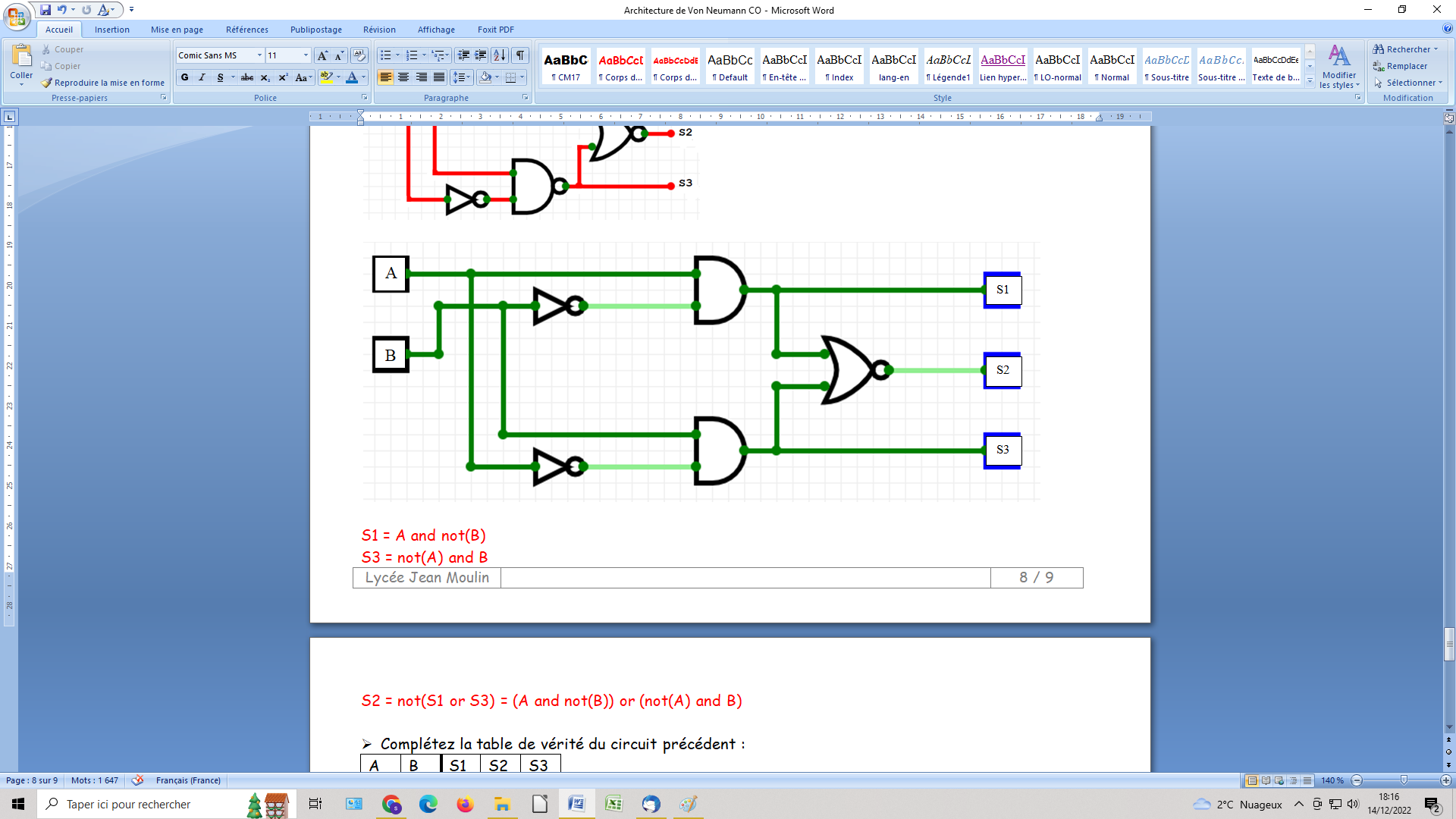


Complétez la table de vérité du circuit précédent :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | C | not A | not B |  | S |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 |

Vérifiez par simulation la table de vérité précédente : https://circuitverse.org/simulator

* Donnez les équations logiques de S1, S2 et S3 avec le circuit suivant :



* Complétez la table de vérité du circuit précédent :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | S1 | S2 | S3 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |

Vérifiez par simulation la table de vérité précédente : https://circuitverse.org/simulator