

Laboratoire d'Électronique numérique**Titulaire : Greta Van Vinckenroy****Manipulation n°11****Titre : Réalisation d'une unité centrale de micro-ordinateur**Groupe : **Bloc ALU**

Rédacteur : Dethier Grégoire

Binôme : Daniel, Ronick

Date manipulation :

Rapport remis le :

*Commentaires :**Note : /20*

Catégorie technique

Section Electronique – Finalité Electronique Appliquée

Bloc1 (1EA)

Année académique 2021 – 2022

Table des matières

Introduction :	3
Schéma de la machine de Von Neuman :	3
Objectif :	4
Le CPU	4
Consignes données :	4
Schéma bloc :	5
Schéma simulation (Multisim) :	6
Schéma simulation test (Multisim) :	7
Liste des composants utilisés :	8
Montage sur breadboard :	10
Légende :	10
Problèmes rencontrés et solutions :	11
Conclusion :	11
Bibliographie :	12

Introduction :

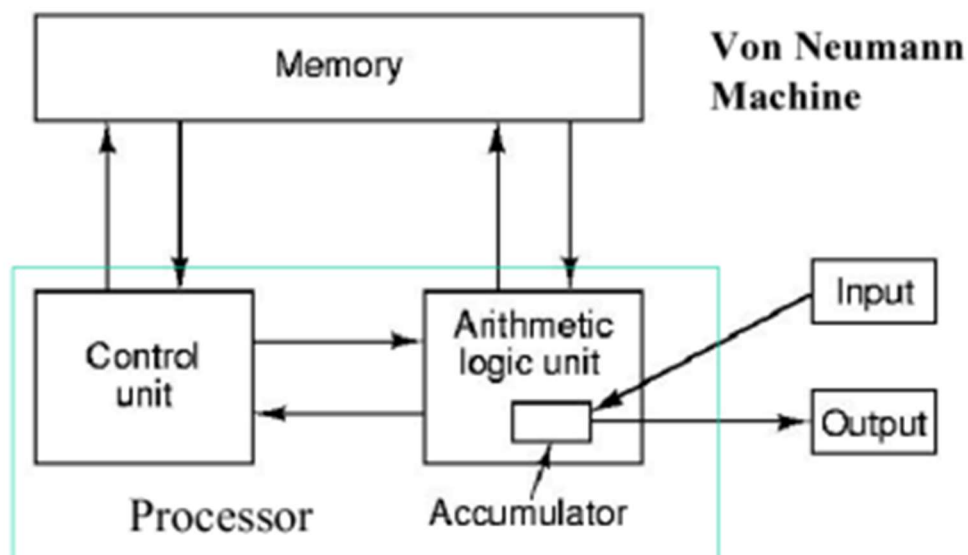
Premièrement, nous avons en groupe divisé le micro-ordinateur en 4 différents blocs (CU, ALU, OSC et I/O).

Nous avons eu le Bloc ALU (Arithmétique and Logic Unit), dans un premier temps on a dû déterminer la structure générale de notre bloc, quels composants utiliser et comment les connecter entre eux.

Après avoir établi un schéma qui semble correct sur papier, nous avons procédé au montage du schéma en simulation sur multisim, chaque bloc fait sa partie de son côté, lorsque toutes les simulations fonctionnent individuellement, on peut enfin communiquer avec les autres blocs pour connecter tous les schémas ensemble.

La dernière étape est de tester le montage sur breadboard en connectant les montages des 4 blocs ensemble et de tester les opérations.

Schéma de la machine de Von Neuman :



Objectif :

Réalisation de l'unité centrale d'un microordinateur d'architecture Von Neumann.

Pour rappel, l'ordinateur de type Von Neumann utilise une structure de stockage unique pour conserver à la fois les instructions et les données requises ou générées par le calcul.

(Un ordinateur d'architecture d'Harvard, sépare physiquement la mémoire de données et la mémoire programme. L'accès à chacune des deux mémoires s'effectue via deux bus distincts.)

Il y a donc plusieurs parties distinctes dans un micro-ordinateur :

Le CPU (central processing unit) regroupe

- L'unité arithmétique et logique (UAL ou ALU en anglais) ou unité de traitement : son rôle est d'effectuer les opérations de base ;
- L'unité de contrôle, chargée du séquençage des opérations ;
- Les accumulateurs, servant au stockage temporaire des données de/depuis la mémoire et les entrées/sorties

La mémoire qui contient à la fois les données et le programme qui dira à l'unité de contrôle quels calculs faire sur ces données. De manière générale, la mémoire se divise entre mémoire volatile (programmes et données en cours de fonctionnement) et mémoire permanente (programmes et données de base de la machine).

Les dispositifs d'entrée-sortie, qui permettent de communiquer avec le monde extérieur.

Sans oublier les éléments indispensables tels l'**oscillateur** et l'**alimentation**...

Consignes données :

- Les opérandes a et b ont deux bits.
- l'UAL doit pouvoir réaliser quatre opérations :
 - au niveau logique: $s = a \text{ AND } b$, $s = a \text{ OR } b$, $s = \text{NOT } a$
 - au niveau arithmétique : $s = a + b$.
- Ce bloc contient des accumulateurs (registres) pour stocker les variables de travail et assurer une resynchronisation des timings - nombre minimum à définir
- interconnexion avec le BLOC_IO par un bus de données, muni de pull-ups.

Schéma bloc :

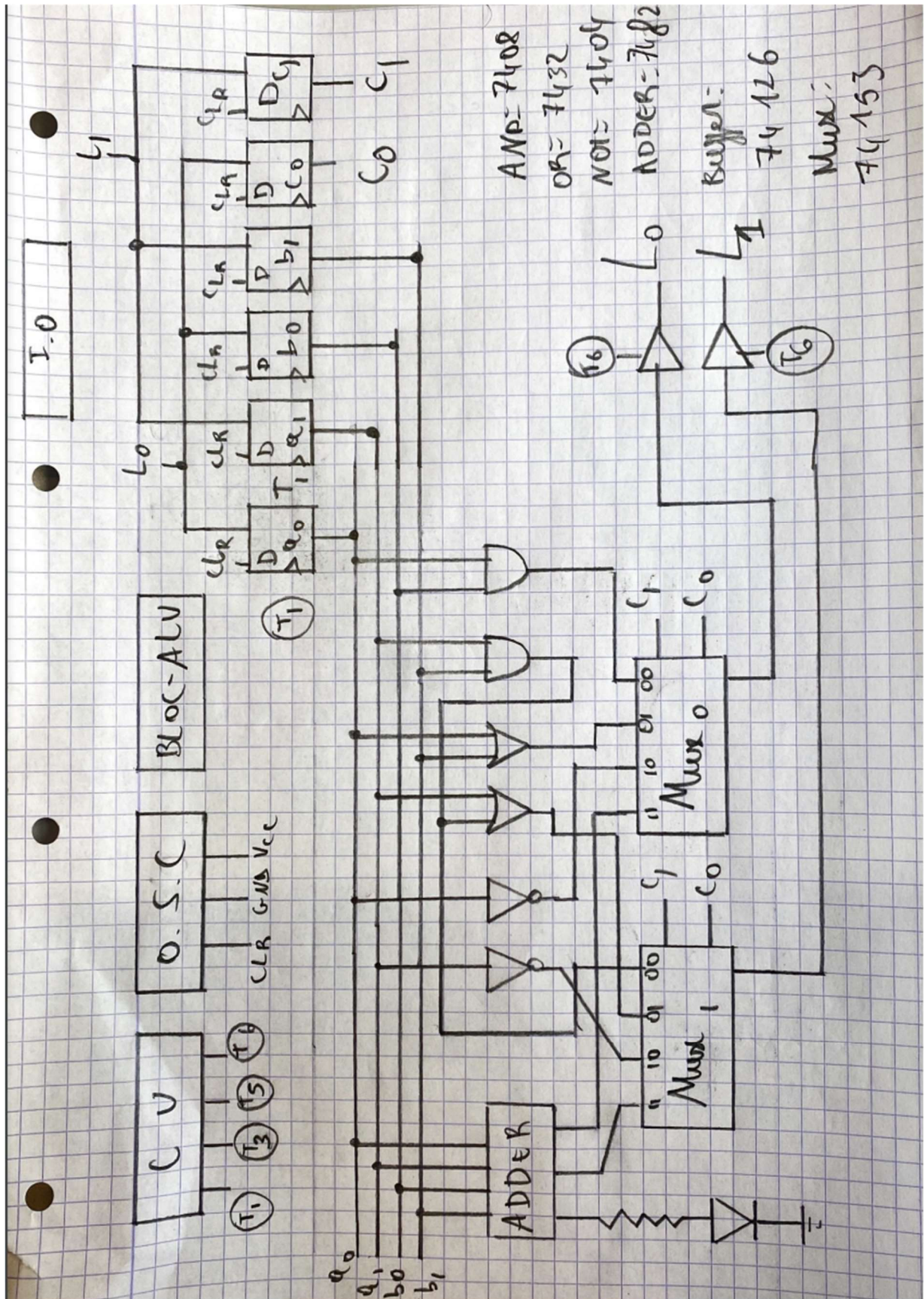


Schéma simulation (Multisim) :

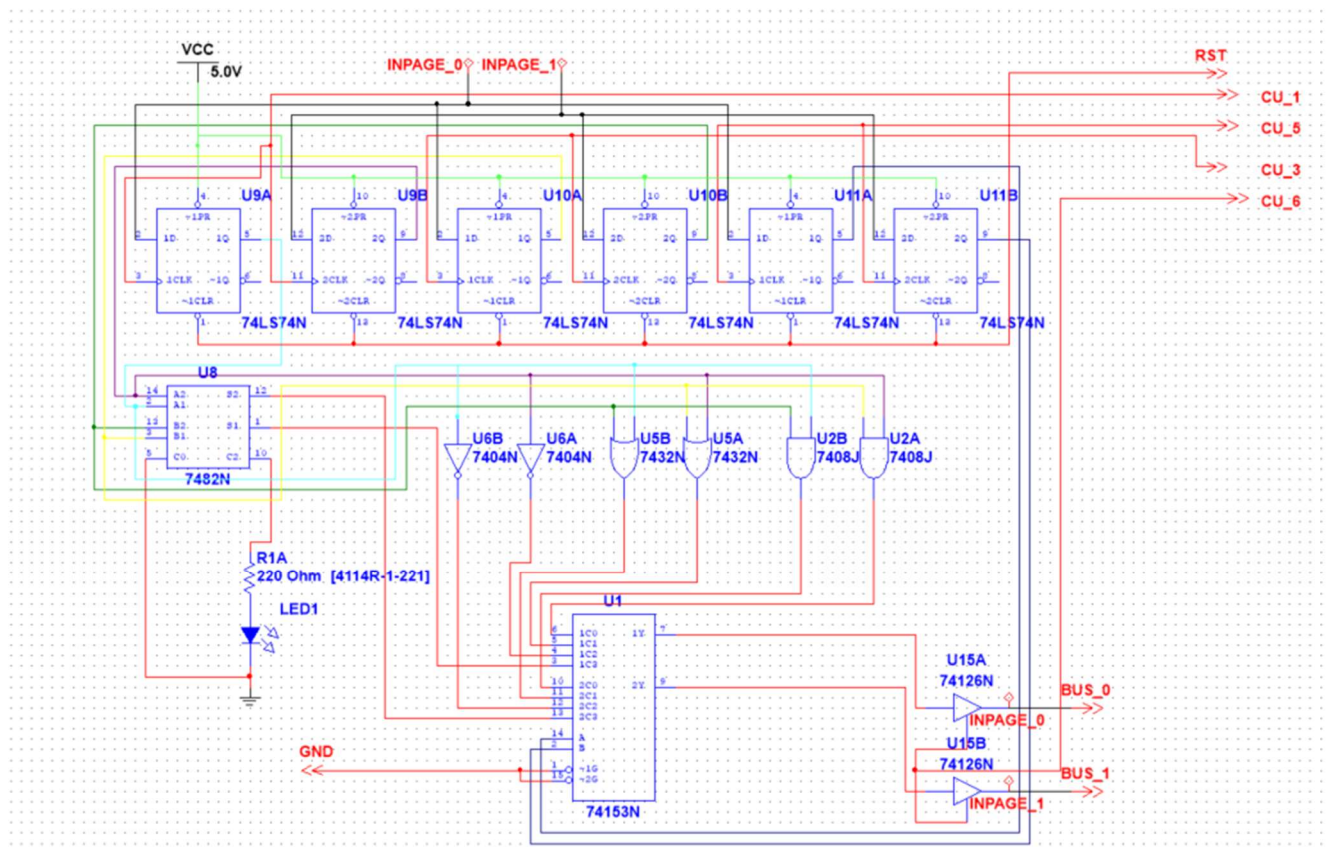
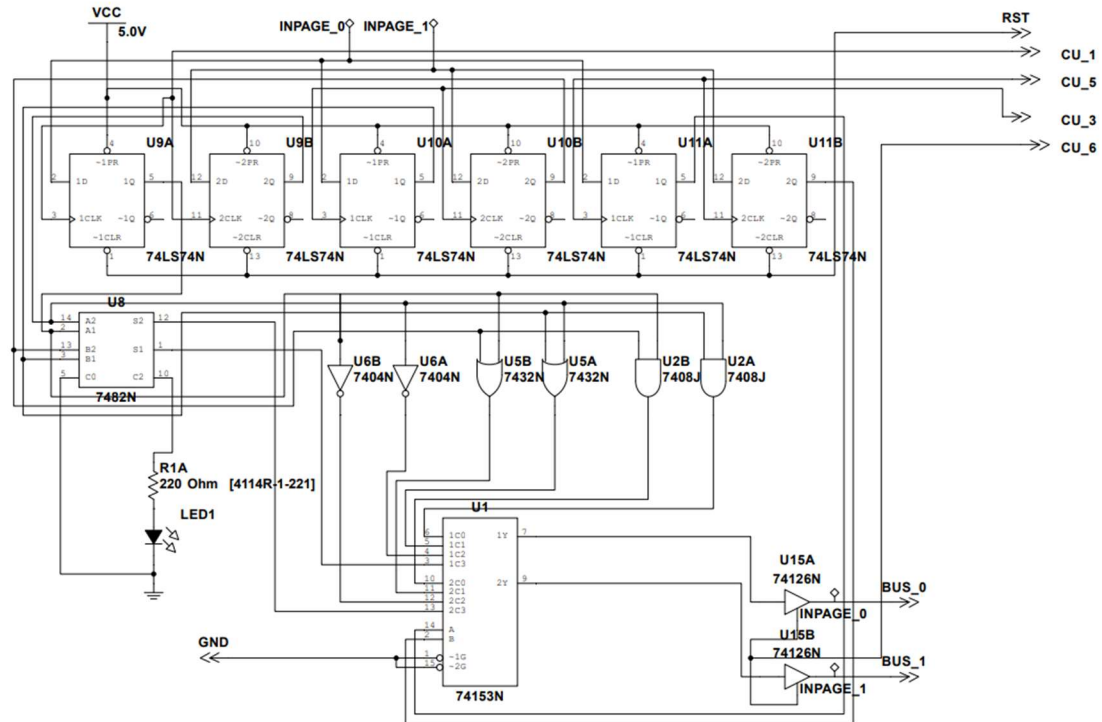
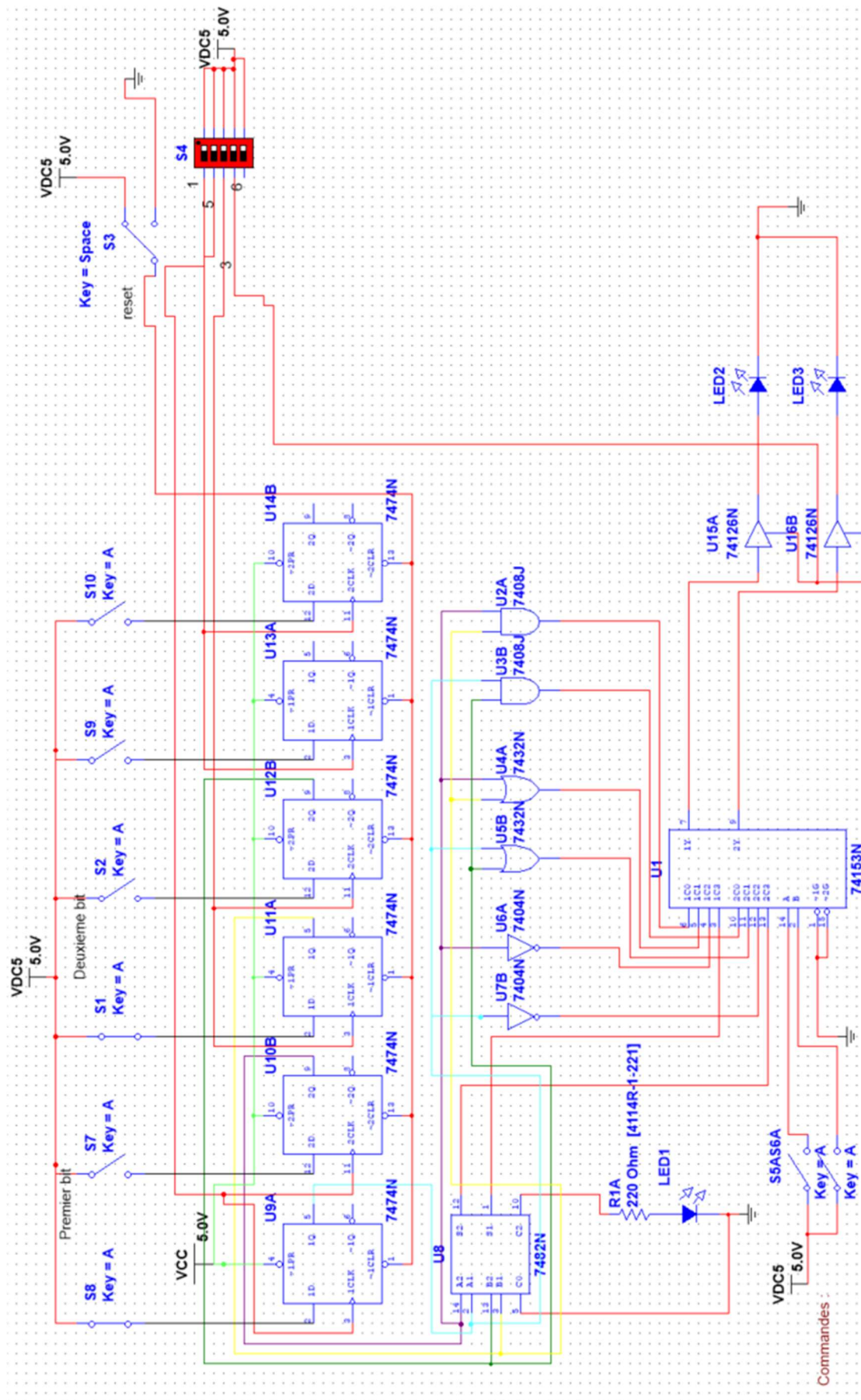


Schéma simulation test (Multisim) :



Liste des composants utilisés :

- **7474: D-Type Positive Edge-Triggered Flip-Flop (x3)**

Registre de stockage des variables de travail reçues du Bloc IO. Les trois bits de poids le plus faible (a0,b0,c0) et les trois bits de poids le plus fort (a1,b1,c1) arrivent en série sur deux bus (Bus « L0A » et « L1A »).

Nous utiliserons 6 bascules qui stockeront successivement les trois paires de bits. Le Bloc CU nous enverra les trois premiers coups de clock pour pouvoir réceptionner et enregistrer les bits paire par paire (Bus « ALUA », « ALUB », « ALUC »).

- **7432: 2- input OR Gates**

Réalisation de l'opération logique OU.

- **7408: 2-input AND Gates**

Réalisation de l'opération logique ET.

- **7404: HEX INVERTER**

Réalisation de l'opération logique NON, uniquement pour a0 et a1.

Nous aurions pu nous passer de ce composant. En effet, il nous suffisait d'utiliser les sorties /Q des bascules Da0 et Da1.

- **7482: 2-Bit Binary Full Adder**

Réalisation de l'opération arithmétique d'addition.

- **74126: Buffer**

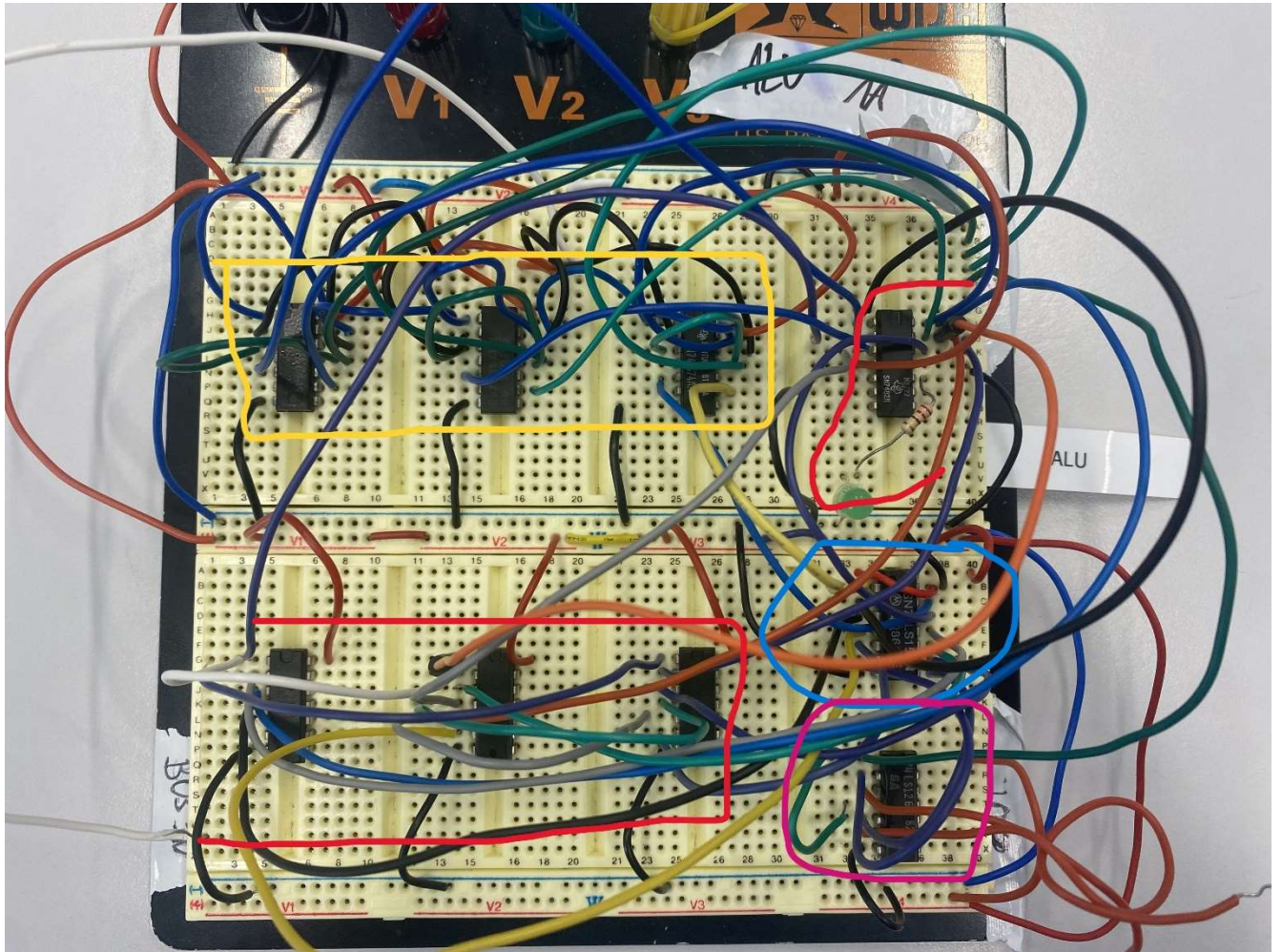
Haute impédance qui nous sert à protéger nos sorties quand le BLOC_IO nous transmet des données.

Dès que tous les bits de donnée sont arrivés, le buffer reçoit le 4^{ème} coup de clock (Bus : « ALUS ») du CU et les données peuvent repartir vers le Bloc IO.

- **74153: Multiplexer**

Sélection de l'opération à exécuter grâce au bit de commande c0 et c1 qui nous viennent aussi du Bloc IO.

Montage sur breadboard :



Légende :

Jaune : bascules, registre a décalage

Rouge : système arithmétique (adder, or, ou, et)

Bleu : Multiplexer

Rose : Buffer

Problèmes rencontrés et solutions :

Le principe de communiquer avec les autres groupes est de bien choisir les composants pour ne pas avoir de problèmes et avoir les mêmes noms de câbles d'entrée/sortie de bus afin que les pages multisim s'accordent. On aurait pu perdre un temps précieux en niant la communication entre les différents groupes.

On a eu du mal à trouver la manière donc les différentes informations allaient arriver dans le bon ordre à l'entrée des bascules mais on s'est accordé avec le CU pour qu'on reçoive des coups de clocks aux moments désirés, ce qui amènera les différents bits aux bons moments.

Conclusion :

Premièrement, on a dû faire un schéma et le câblage après avoir étudié la structure d'un micro-ordinateur. Le fait de devoir travailler en groupe est un gros avantage si les groupes savent bien communiquer entre eux, cela a permis à l'ensemble de la classe de comprendre comment fonctionne un micro-ordinateur.

Ensuite on a commencé à apprendre à utiliser Multisim qui est un logiciel de simulation, un logiciel très utile qui nous a permis de nous introduire à la simulation de circuits électroniques.

Pour ce dernier labo de numérique, le professeur nous a laissé travailler en autonomie, une façon de travailler qui nous sera très utiles pour nos projets futurs. Nous avons dû trouver les composants correspondants, manipuler un nouveau logiciel, communiquer afin d'avoir les mêmes entrées/sorties, etc...

Finalement, on a pu tester nos simulations des quatre blocs ensembles et le tout fonctionne, nos opérations sont correctes et le résultat est logique, la dernière étape est de câbler le tout et tout connecter pour voir si ça fonctionne comme en simulation.

Bibliographie :

Exemple de construction d'un bloc ALU :

<https://www.youtube.com/watch?v=mOVOS9AigFs>

Site construction micro-ordi: <https://eater.net/8bit>

Vidéo ytb: <https://www.youtube.com/watch?v=HyznrdDSSGM>

Tuto comment utiliser Multisim :

<https://www.youtube.com/watch?v=ujKN7MUHYys>

Texas instruments – datasheets:

<https://www.ti.com/>

NI : (ou trouver multisim)

<https://www.ni.com/fr-be.html>