Etude et Modélisation des Composants Actifs

Rapport de projet Pierre Papier Ciseaux

Benne Cécile
Bougeard Matthieu
23 janvier 201

Table des matières

Introduction

Dans le cadre de l'étude des composants actifs, nous sommes amenés à réaliser un TP sur le logiciel Microwind. Nous sommes libres de choisir un sujet de développement. Nous avons choisi de réaliser le célèbre jeu « Pierre Papier Ciseaux ». Cela nous semble être un sujet d'étude original à finalité ludique.

Dans le TP, nous avons utilisé les composants mis à disposition dans les bibliothèques associées au logiciel. Cependant, nous avons vérifié au préalable le comportement de chacune des portes (NAND, NOR et porte inverseur).

Description logique du jeu

Nous commençons par décrire le jeu en termes d'entrées/sorties de cette façon :

- Trois entrées par joueur : une par choix de jeu
    - Une pour Pierre
    - Une pour Papier
    - Une pour Ciseaux
- Trois sorties :
    - Une pour la gagne du joueur 1
    - Une pour la gagne du joueur 2
    - Une pour le match nul
- Nous utilisons les règles classiques du jeu :
    - la pierre l'emporte sur les ciseaux
    - les ciseaux l'emportent sur le papier
    - le papier l'emporte sur la pierre

Cela donne donc une valeur en sortie fonction des combinaisons d'entrée. Il y a six entrées (Pi1, Pa1, Ci1, Pi2, Pa2, Ci2). On obtient donc $2^6 = 64$ combinaisons.

Nous comprenons qu'il sera compliqué de partir de cette modélisation pour développer notre jeu. En effet, les équations sont difficiles à simplifier et cela impliquera l'utilisation d'un grand nombre de portes logiques (chose malaisée sous Microwind).

Nous choisissons alors de coder chaque combinaison par un code sur 2 bits :

(A1,A2) codera le choix du joueur 1
(B1,B2) codera le choix du joueur 2
(R1,R2,R3) codera le résultat du jeu

Tels que

R1 est à 1 si le joueur 1 gagne
R2 est à 1 si le joueur 2 gagne
R3 est à 1 s'il y a match nul

	A1	A2		B1	B2	R1	R2	R3
Rien	0	0	Rien	0	0	0	0	0
Rien	0	0	Pierre	0	1	0	0	0
Rien	0	0	Papier	1	0	0	0	0
Rien	0	0	Ciseaux	1	1	0	0	0
Pierre	0	1	Rien	0	0	0	0	0
Pierre	0	1	Pierre	0	1	0	0	1
Pierre	0	1	Papier	1	0	0	1	0
Pierre	0	1	Ciseaux	1	1	1	0	0
Papier	1	0	Rien	0	0	0	0	0
Papier	1	0	Pierre	0	1	1	0	0
Papier	1	0	Papier	1	0	0	0	1
Papier	1	0	Ciseaux	1	1	0	1	0
Ciseaux	1	1	Rien	0	0	0	0	0
Ciseaux	1	1	Pierre	0	1	0	1	0
Ciseaux	1	1	Papier	1	0	1	0	0
Ciseaux	1	1	Ciseaux	1	1	0	0	1

Tableau 1 : Combinaisons admises

On obtient le jeu d'équations :

$$R1 = \overline{A1}.A2.B1.B2 + A1.\overline{A2}.\overline{B1}.B2 + A1.A2.B1.\overline{B2}$$

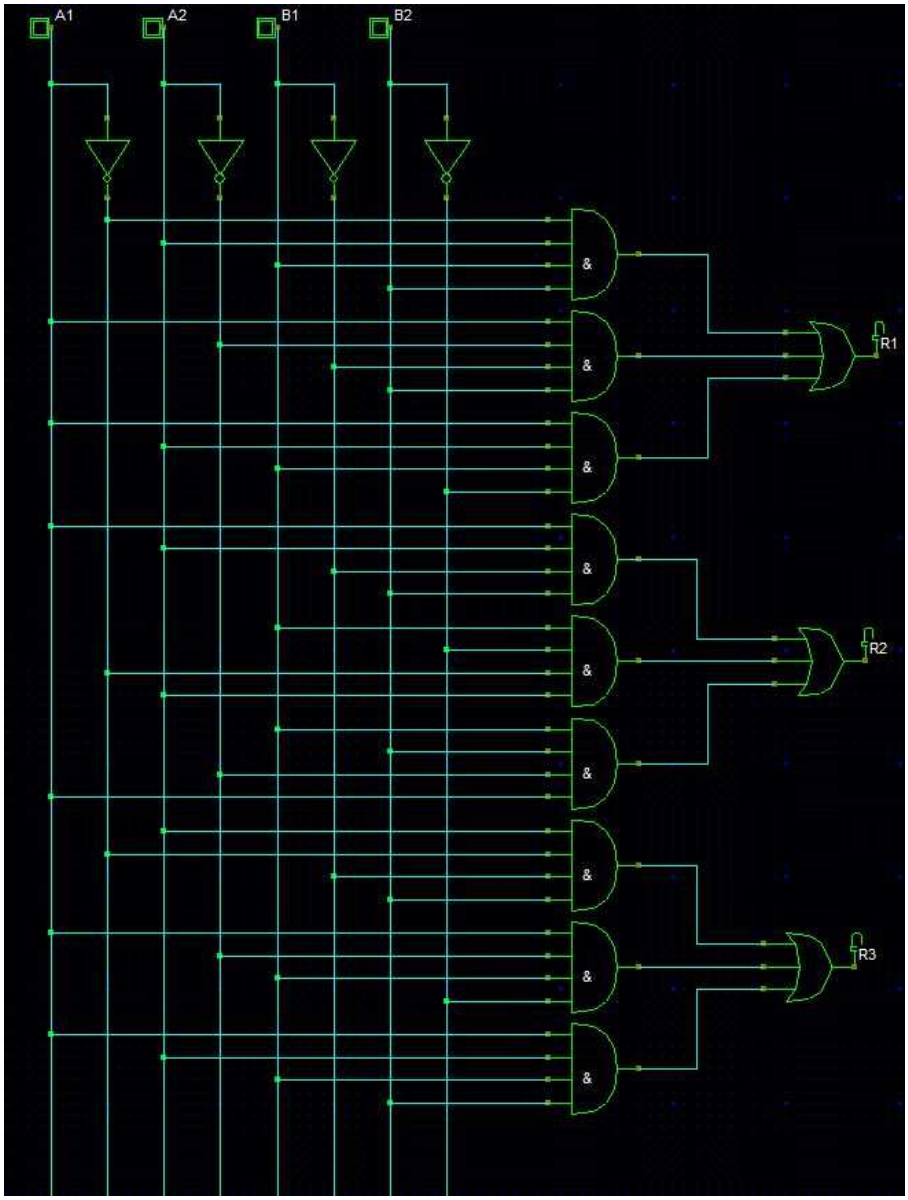
$$R2 = \overline{A1}.A2.B1.\overline{B2} + A1.\overline{A2}.B1.B2 + A1.A2.\overline{B1}.B2$$

$$R3 = \overline{A1}.A2.\overline{B1}.B2 + A1.\overline{A2}.B1.\overline{B2} + A1.A2.B1.B2$$

Ces équations impliquent l'utilisation de 3 x 3 portes NAND et 3 portes NOR, c'est-à-dire beaucoup moins que dans la précédente modélisation.

Câblage en portes logiques

On peut câbler ce jeu d'équations directement en porte AND et OR. On utilise Dsch35 :



On peut ainsi tester le bon fonctionnement logique de cet agencement grâce à l'outil de simulation de Dsch35.

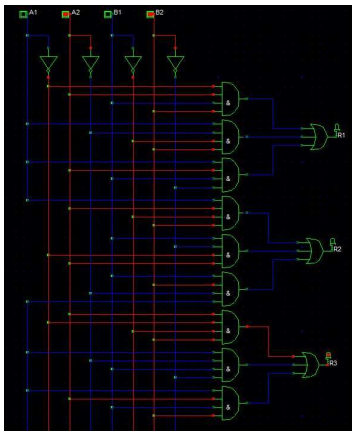


Figure 1 : test du match nul

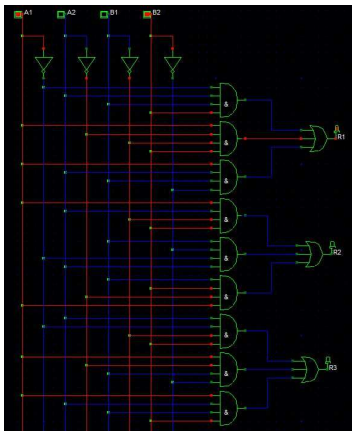


Figure 2: test de la victoire du joueur 1

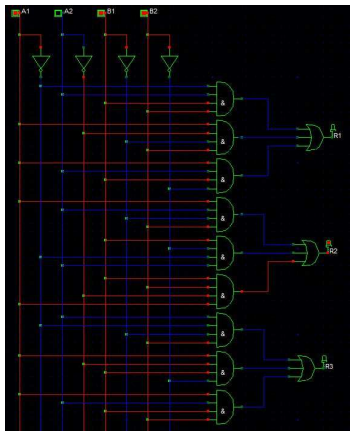


Figure 3 : test de la victoire du joueur 2

Nous validons la logique et pouvons passer à l'implémentation en couches.

Implémentation sous Microwind

Nous avons puisé dans une bibliothèque de portes logiques pré-câblées. Nous nous sommes permis ce raccourci car nous avions abordé les problématiques inhérentes à cet exercice lors des premiers TPs, avant le début du projet. Il s'agit donc d'assembler les portes logiques et d'optimiser ce montage à la recherche d'un compromis vitesse d'exécution / consommation.

Pour chaque sortie, nous avons créé une matrice de base permettant de créer chaque signal d'entrée. Puis nous avons implémenté avec des portes logiques NAND 4 entrées et NOR 3 entrées.

Pour simuler les 8 entrées et les 12 sorties nécessaires, nous avons créé une matrice 8x12. Chaque colonne correspond aux quatre entrées A1, A2, B1 et B2 et leur complément. La première colonne est donc A1, la deuxième correspond donc au complément d'A1. En modifiant juste les connexions, nous avons ainsi pu réaliser les trois sorties.

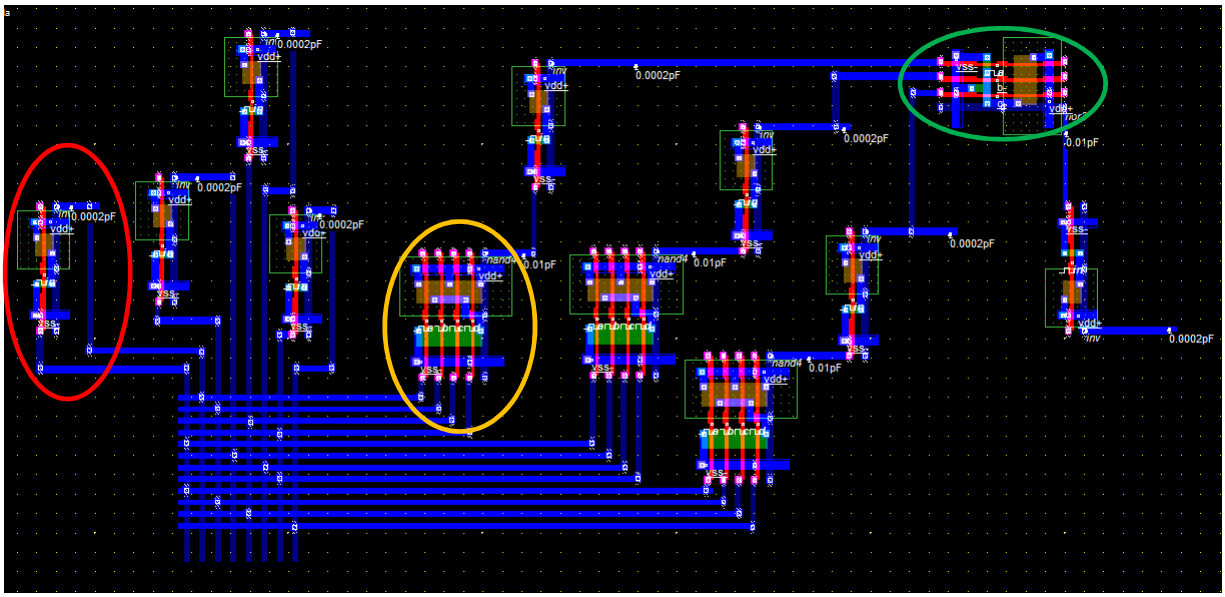


Figure 4: Schéma Microwind de la logique de la sortie de R1.

En rouge est entourée une porte NON. Elle et ses quatre voisines permettent d'inverser chacune des entrées.

En orange est entourée une porte NAND. La porte AND n'étant pas disponible dans la bibliothèque, nous avons utilisé une NAND suivie d'une NON. On obtient alors le fonctionnement logique d'une AND.

En vert est entourée la porte NOR, suivie d'une porte inverseur, ce qui permet d'obtenir une porte OR.

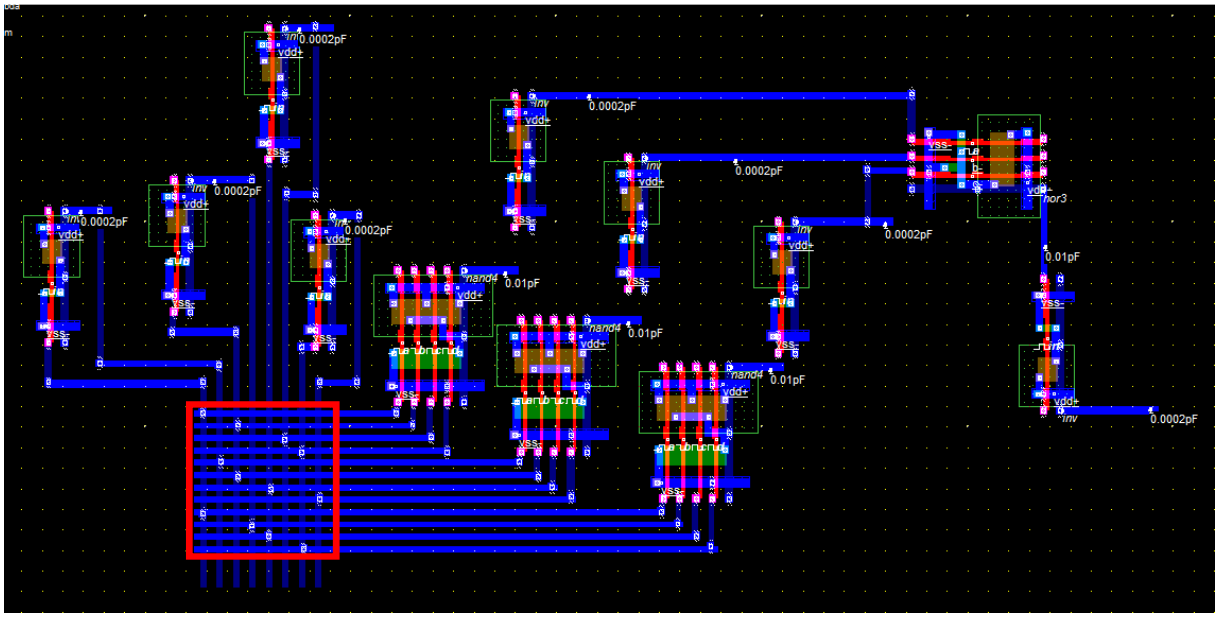


Figure 5 : Schéma sous Microwind de la logique de la sortie R2

L'analyse de cette partie est en tout point similaire à celle présentée en Figure 4. L'avantage de cette configuration en matrice donne la possibilité de configurer par placement de contacts sur la matrice (encadré en rouge sur la Figure 5).

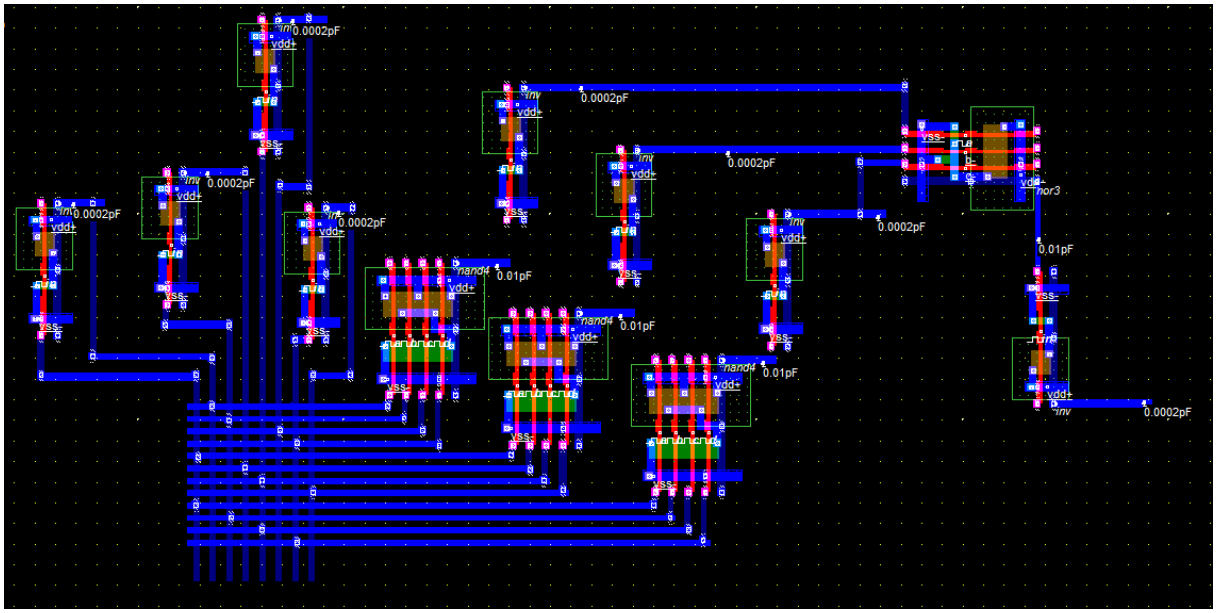


Figure 6 : Schéma sous Microwind de la logique de la sortie R3

Simulations du comportement :

En entrée du système, on met des horloges de fréquences différentes ce qui nous permettra d'avoir les différentes combinaisons d'entrées.

Figure 7: Simulation de la sortie R1

Testons une sortie après l'autre. Sur la Figure 7, nous observons la sortie R1 ainsi que chacun des termes en AND. Chacun des rails rouges mets en évidence l'état 1 de chacun de ces termes. Ils se retrouvent projetés sur la sortie R1.

On constate que R1 ne passe pas à 1 à chaque fois qu'il le devrait. Le cercle rouge met en évidence ce dysfonctionnement que l'on pourrait expliquer en remarquant que les pics sur R1 sont d'amplitude très faible ; suffisamment faible pour ne pas être pris en compte et être considérés comme un zéro.

On remarque également un léger décalage temporel entre le moment où la sortie de la porte inverseur passe à 1 et le moment où R1 passe à 1.

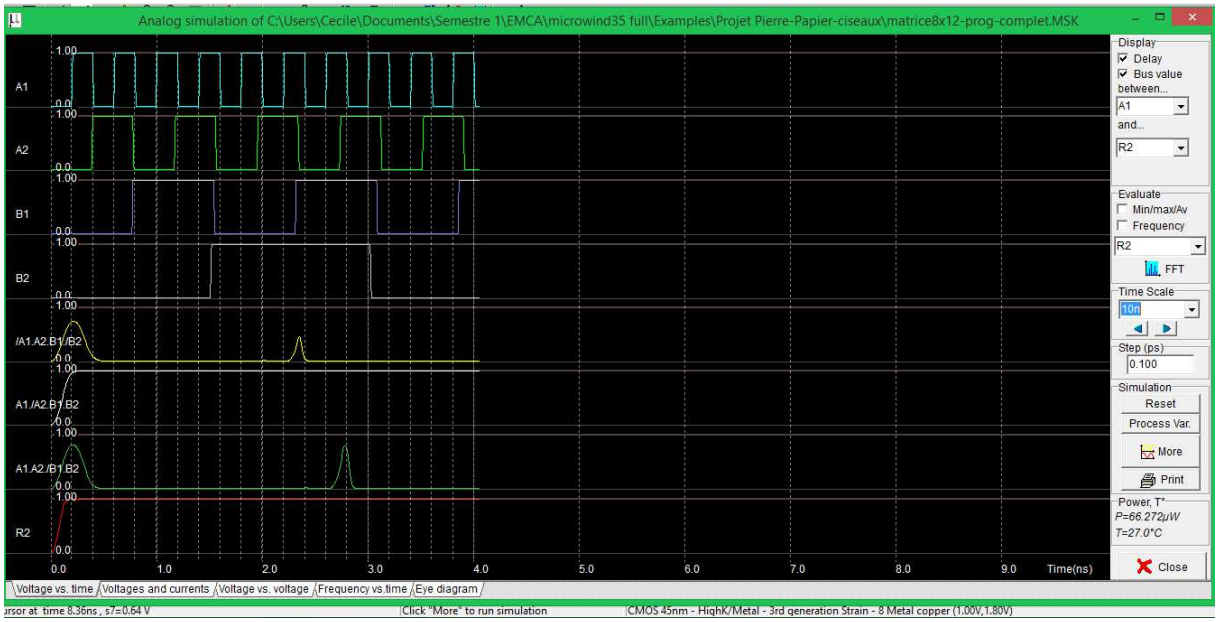


Figure 8: Simulation de la sortie R2

On peut observer une erreur : la sortie R2 est toujours à 1. Après simulation, on peut voir que c'est la deuxième porte NAND qui est toujours à 1. Après test, il s'avère que l'erreur vient seulement d'un contact oublié. La sortie se mettait donc automatiquement à 1, c'est-à-dire à la tension d'alimentation de la porte. Cette erreur corrigée, nous passons à la simulation du montage entier :

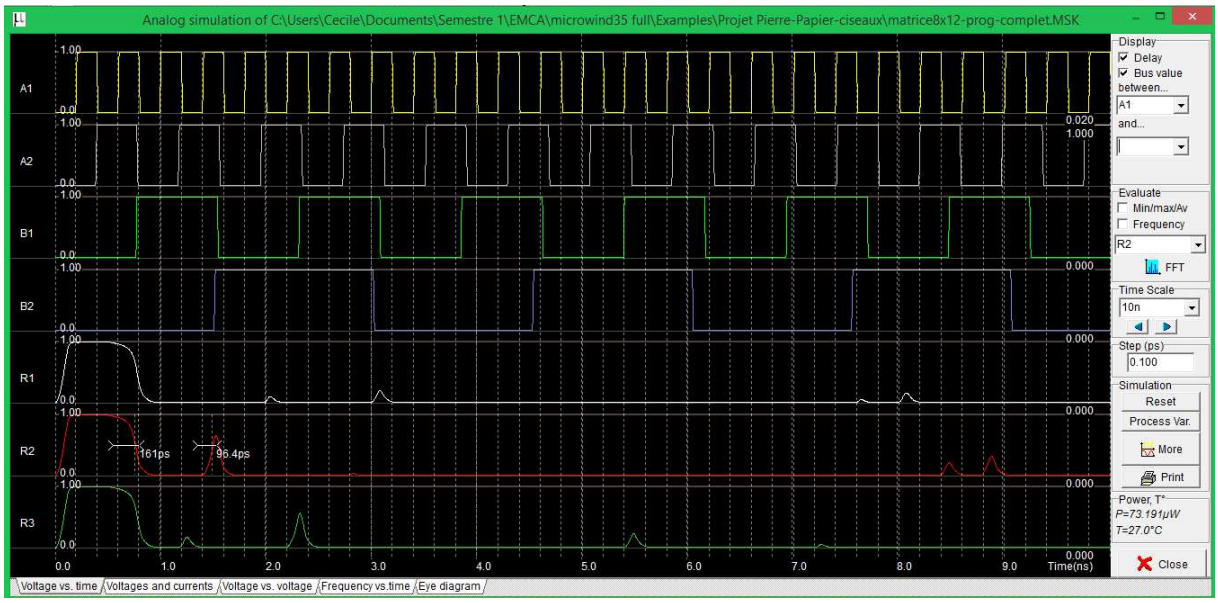


Figure 9 : Simulation des trois sorties R1, R2 et R3.

Nous sommes confrontés au même problème : l'amplitude des pics n'est pas suffisamment importante pour que tous ressortent. Ajoutant à cela les temps de propagation, l'objet final a un fonctionnement altéré. Pour un cycle d'entrée, nous devrions observer, sur chaque sortie, 3 pics ; ce n'est pas le cas.

Pour pallier ce problème, nous pourrions diminuer la fréquence de commutation des entrées.

Il est intéressant d'analyser comment optimiser cette implémentation. Nous observons par exemple que :

- la largeur des canaux doit être très petite devant leur longueur
- les liaisons doivent être les plus courtes possibles pour limiter les temps de propagation
- limiter la surface diminue les capacités parasites

Le logiciel permet de simuler le fonctionnement sous des températures différentes. Testons à 70°C :

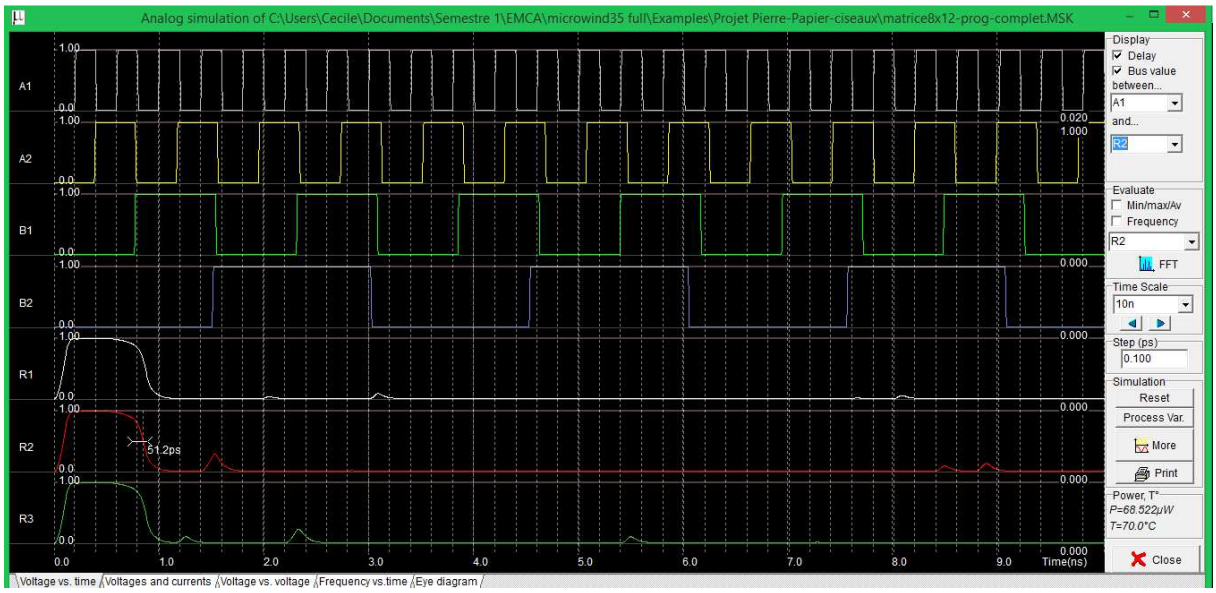


Figure 10 : Simulation sous 70°C

On constate l'augmentation du temps de réponse. Ceci se traduit sur la Figure 10 par l'abaissement de l'amplitude des pics.

Conclusion

Ce projet a été le premier d'une discipline que nous n'avions jamais abordée auparavant. Il a donc été particulièrement intéressant puisqu'il nous a permis de mettre en pratique les connaissances acquises les mois et années passés en semi-conducteur, puis en EMCA. Il nous a également permis de prendre en compte les soucis de fabrication : temps de commutation dans les composants, vitesse d'exécution, dérive en température…

La première difficulté que nous avons rencontrée dans ce sujet a été le nombre de combinaisons possibles pour le jeu et la réalisation de la table de vérité. Heureusement, une simplification du code réalisée en associant deux variables uniquement par joueur et une combinaison pour chaque coup a permis de diminuer fortement la logique. Pour réaliser notre logique câblée sous Microwind, nous aurions pu passer par le schéma électronique. Cependant, dans un souci de faciliter les corrections des bugs, nous avons préféré réaliser nous-même la logique en associant des portes NAND, NOR et NO.