Processeu NONO 1et 2

Cédric BOIS Benjamin SIENTZOFF

13 décembre 2014

Table des matières

Introduction

Dans le cadre du cours intitulé *Architecture des ordinateurs*, nous devons recréer un processeur Nono-1. Par la suite, ce processeur sera modifier pour devenir Nono-2. Ce rapport retrace comment nous avons réalisé ces processeurs MIPS.

Les circuits électroniques présentés sont produits avec le logiciel *Logisim*. Ces circuits et les différents fichiers permettant notamment de programmer le processeur sont fournis avec la version numérique de ce rapport. Les images RAM peuvent être directement chargées dans la RAM des processeurs Nono. Ces images correspondent aux programmes compilés pour ces architectures et peuvent être exécutés directement dans *Logisim*.

Dans une première partie, nous présentons les différents sous-circuits composants le processeur Nono-1. Une seconde partie présente sont fonctionnement global et les modifications apportées à Nono-1 pour implémenter les fonctions de Nono-2.

1 Réalisation de Nono-1

1.1 Opcode des instructions

Nono-1 et Non-2 sont des processeurs utilisant l'assembleur MIPS. Les instructions disponibles sur Nono-1 sont présentés au tableau de la figure 1.1. On remarque que les instructions reconnues sont relativement restreintes. Ces instructions sont de trois formats différents comme on peut le voir à la figure 1.1 [1].

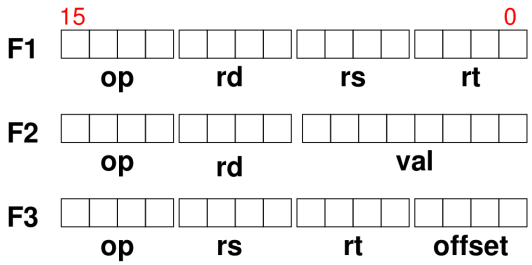

FIGURE 1 – Formats des instructions

Le Format F1 Le format F1 est composé de quatre paquets de bits. Le premier est sur quatre bits, il correspond au code de l'instruction, et c'est le cas pour tous les formats d'instructions. Les trois paquets suivants, sur quatre bits. Ce format est utilisé typiquement pour des opérations faisant intervenir trois registres. Le premier correspond à la destination du résultat et les deux suivants aux registres contenant les opérantes.

Le Format F2 Le format F2 est composé de trois paquets de bits. Le premier est sur quatre bits, il correspond au code de l'instruction. Les 4 bits suivants correspondent à un nom de registre et les 8 derniers à une valeur immédiate. Ce format est typiquement utilisé pour l'instruction *li*.

Le Format F3 Le format F3 peut être divisé en quatres parties. C'est le format utilisé pour les sauts. Les quatres bits correspondent à l'opcode de l'instruction. Le paquet des quatres bits et le suivant constitué des quatres autres bits suivant correspondent à des noms de registres. Enfin les derniers bits (au nombre de quatre) correspondent à un offset. Pour les sauts, cela correspond à l'adresse de l'étiquette où effectuer le saut.

Le choix des opcode n'a pas était fait au hasard, en effet, en regardant le nombre de sauts et le nombre d'opérations faisant appellent à l'unité arithmétique et logique du processeur, on s'apperçoit qu'il peuvent consistuer deux groupes. On a donc coupé les opcode en trois groupes. Le premier correspond aux opcodes qui commence par un 1, ce sont les instructions qui font appelle à l'UAL.
Le second groupe, les opcodes commencent par un 0, correspondent aux saut. Enfin le dernier groupe est composé des autres instructions, les opcodes 0000 et

1. Tiré du sujet du projet rédigé par M. Frédéric GOUALARD

Instruction et paramètres	Format	Opcode
`add` r_d, r_s, r_t	F_1	1000
`sub` r_d, r_s, r_t	F_1	1001
`or` r_d, r_s, r_t	F_1	1010
`and` r_d, r_s, r_t	F_1	1011
`not` r_d, r_s	F_1	1100
`shl` r_d, r_s, r_t	F_1	1101
`shr`$_d$, r_s, r_t	F_1	1110
`li` r_d, *val*	F_2	1111
`halt`	F_1	0000
`b` *offset*	F_3	0001
`beq` r_s, r_t, *offset*	F_3	0010
`bne` r_s, r_t,*offset*	F_3	0011
`bge` r_s, r_t, *offset*	F_3	0100
`ble` r_s, r_t, *offset*	F_3	0101
`bgt` r_s, r_t, *offset*	F_3	0110
`blt` r_s, r_t, *offset*	F_3	0111

FIGURE 2 – *Opcode* des différentes instruction du processeur NONO 1

1111. Le tableau des instructions, leur format et les opcodes correspondants est présenté à la figure 1.1.

Maintenant qui vous avons définit nos opcodes, il est temps de concevoir les circuits électroniques composants le processeur NONO 1. Commençons par l'Unité Arithmétique et Logique.

1.2 L' unité arithmétique et logique

L'Unité Arithmétique et Logique, abrégé UAL, permet de faire des calculs basiques (additions, divisions, décallages de bits, etc.). Elle effectue les calcules sur huit bits. L'UAL a trois entrées. La première sur trois bits permet de préciser le code l'opérateur à effectuer. Les deux autres entrées sur hiut bits sont les opérantes de l'opération demandée.
En sortie, sur *ouput* on peut lire le résultat de l'opération. Il y a également quatres drapeaux comme détaillés ci-dessous.

- CF pour *Carrie Flag* est le drapeau levé lorsque que l'opération génère une retenue.
- ZF pour *Zero Flag* qui est armé lorsque que le résultat comporte uniquement des 0.
- OF pour *Over Flow* qui est un drapeau levé lorsque on dépasse la capacité des nombres représentés.
- SF pour *Sign Flag* qui est levé lorsque le résultat a son bit de poids fort à 1, c'est un nombre signé.

Le schéma électronique de l'UAL est présenté à la figure 1.2. La partie qui décode le signal *ctrlUAL* correspond au tableau de Karnaugh de la figure 1.2

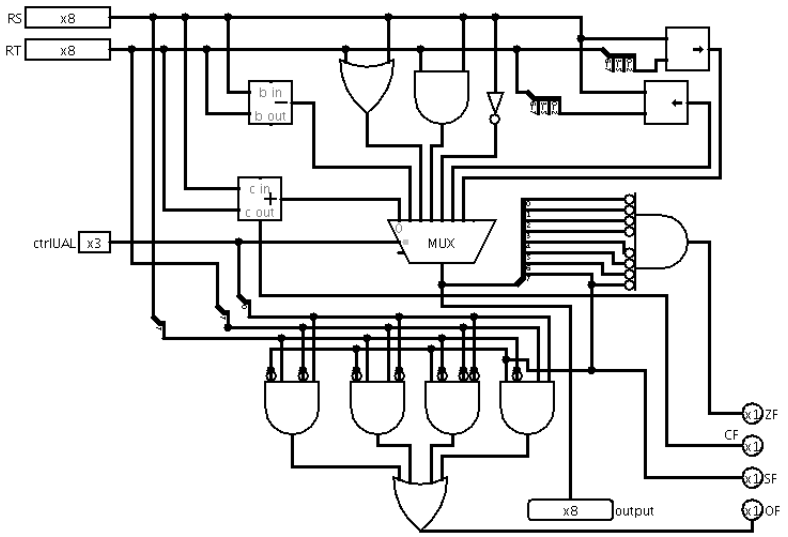

FIGURE 3 – Schéma électronique de l'Unité Arithmétique et Logique

b3b2 \ b1b0	0	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	0	1
10	1	1	1	1

FIGURE 4 – Tableau de Karnaugh pour le décodage de *ctrlUAL*

duquel on extrait l'équation :

$$b3.\neg(b2) + b3.b2\neg(b1) + b3.b1\neg(b0) \quad (1)$$

Cette équation nous permet alors de réaliser le circuit électronique décodant *ctrlAUL*.

1.3 Le contrôleur de saut

Le second circuit éléectronique composant le processeur est le contrôleur de saut. Son rôle est de mettre à jour PC à jour en fonction du résultat de l'opération que vient d'effectuer l'UAL pour le cycle suivant. Le saut est déterminé en fonction des indicateurs que le circuits à en entrées, c'est-à-dire SF et ZF.

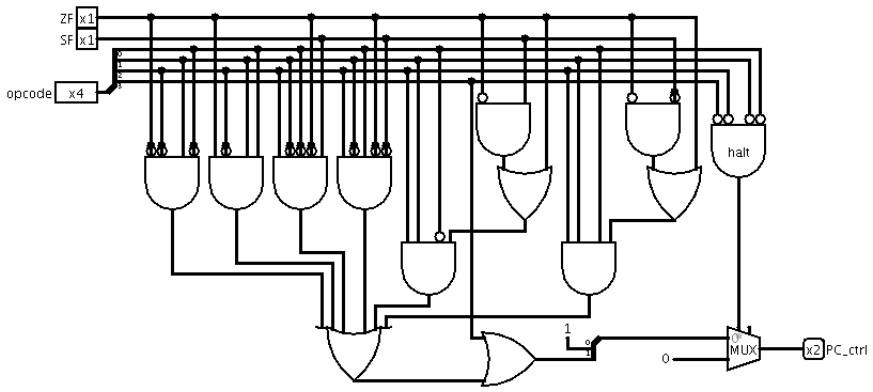

FIGURE 5 – Schéma électronique du contrôleur de sauts

Le schéma électronique du contrôleur de saut est à la figure 1.3.

1.4 Le décodeur d'instructions

intro, explications

1.5 La sélection des registres

intro, explications

1.6 Le banc de registres

intro, explications

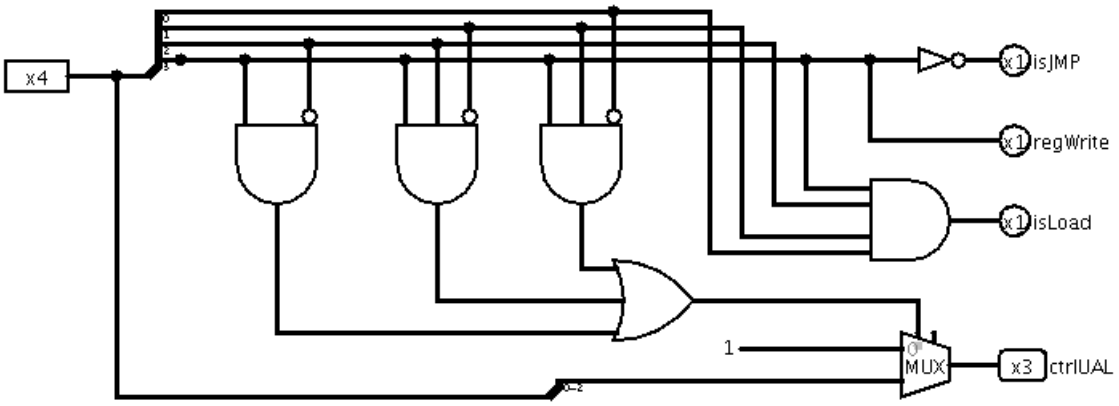

FIGURE 6 – Schéma électronique pour le décodeur d'instructions

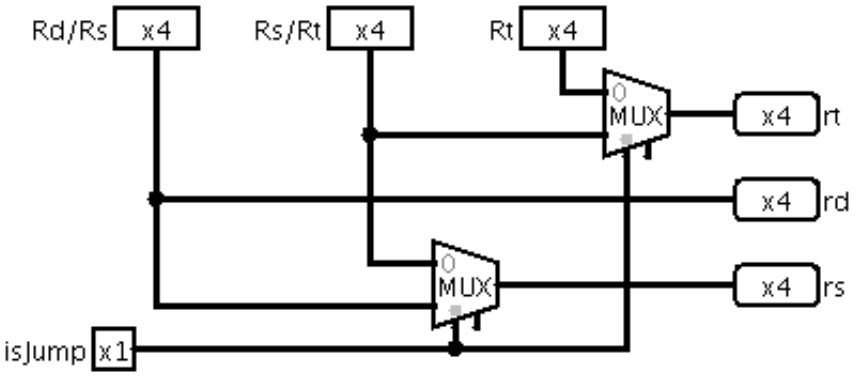

FIGURE 7 – Schéma électronique pour la sélection de registres

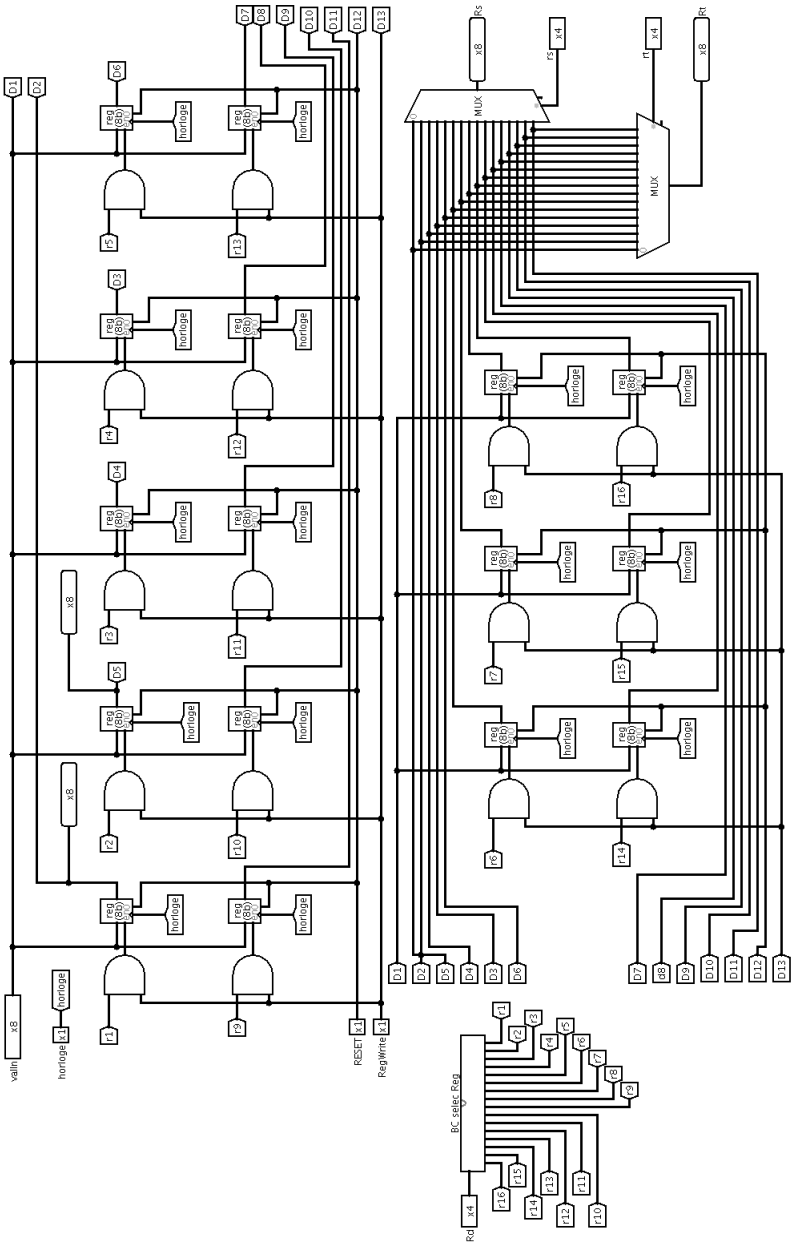

FIGURE 8 – Schéma électronique pour le banc de registres

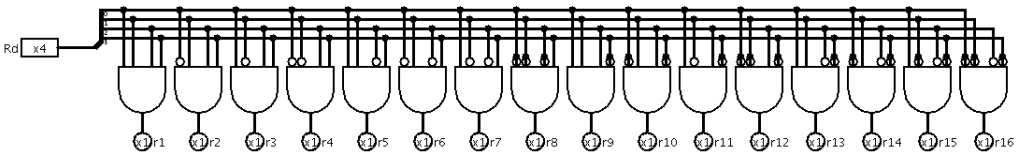

FIGURE 9 – Schéma électronique pour le sélecteur du banc de registres

2 Processeurs Nono-1 et Nono-2

2.1 Nono-1

2.2 Nono-2

Conclusion

je conclu