PRESENTATION MINI-PROJET

Module Architecture – M2101

Louis Lecardonnel et Baptiste Masoud
IUT d'Orsay, 2020

Description des champs et instructions créés

Champs

MUX0 : Taille 1 bit, adresse début 3, valeur défaut 0. Le multiplexeur prend pour sortie son entrée i0 si ce champ est égal à 0, et i1 si égal à 1. L'entrée i0 est le champ DATA, i1 est la valeur stockée dans le registre A.

ALU : Taille 4 bits, adresse début 4, valeur défaut 0000. L'ALU effectuera l'opération correspondant aux différentes valeurs que ce champ peut prendre. Voir table de l'ALU.

Registre A : Taille 1 bit, adresse début 8, valeur défaut 0. Permet de stocker la valeur sortant de l'ALU dans le registre A quand le champ est égal à 1 lors de l'écriture dans les registres.

Registre B : Taille 1 bit, adresse début 9, valeur défaut 0. Permet de stocker la valeur sortant de l'ALU dans le registre B quand le champ est égal à 1 lors de l'écriture dans les registres.

MUX1 : Taille 1 bit, adresse début 10, valeur défaut 0. Le multiplexeur prend pour sortie son entrée i0 si ce champ est égal à 0, et i1 si égal à 1. L'entrée i0 est la valeur stockée dans le registre B, i1 est la valeur stockée dans le registre C.

Registre C : Taille 1 bit, adresse début 11, valeur défaut 0. Permet de stocker la valeur sortant de la mémoire de données dans le registre C quand le champ est égal à 1 lors de l'écriture dans les registres. Il faut noter que la valeur de la mémoire de données correspond à l'adresse du registre B quand MUX1 = 0

JMP : Taille 1 bit, adresse début 12, valeur défaut 0. Permet le saut à l'adresse correspondante au champ DATA dans la mémoire programme quand JMP = 1.

JMPPZ : Taille 1 bit, adresse début 13, valeur défaut 0. Permet le saut à l'adresse correspondante au champ DATA dans la mémoire programme quand JMPPZ = 1 et l'indicateur N de l'ALU = 0.

JMPN : Taille 1 bit, adresse début 14, valeur défaut 0. Permet le saut à l'adresse correspondante au champ DATA dans la mémoire programme quand JMPN = 1 et l'indicateur N de l'ALU = 1.

JMPNZ : Taille 1 bit, adresse début 115, valeur défaut 0. Permet le saut à l'adresse correspondante au champ DATA dans la mémoire programme quand JMPNZ = 1 et l'indicateur Z de l'ALU = 0.

JMPZ : Taille 1 bit, adresse début 16, valeur défaut 0. Permet le saut à l'adresse correspondante au champ DATA dans la mémoire programme quand JMPZ = 1 et l'indicateur Z de l'ALU = 1.

DATA : Taille 16, adresse début 17, valeur défaut 0000 0000 0000 0000. Permet de donnée au programme un nombre codé sur 16 bits ou une adresse codée également sur 16 bits. Il faut noter que l'entrée i0 de l'ALU = DATA si MUX0 = 0.

Instructions

La valeur d'un champ est sa valeur par défaut si aucune valeur n'est explicitement mentionnée pour ce même champ.

LOAD_A #valeur : Registre A = 1, ALU = 0001 (Opération : i0), opérande DATA en valeur immédiate

LOAD_B #valeur : Registre B = 1, ALU = 0001 (Opération : i0), opérande DATA en valeur immédiate

LOAD_A_B : Registre A = 1, ALU = 0010 (opération : i1)

LOAD_B_A : Registre B = 1, MUX0 = 1, ALU = 0001 (opération : i0)

LOAD_C_ADRB : Registre C = 1

INC_B : Registre B = 1, ALU = 0011 (opération : i1 +1)

ADD_A_C : Registre A = 1, MUX0 = 1, MUX1 = 1, ALU = 0101 (opération : i0 + i1)

DIV_B #valeur : Registre B = 1, ALU = 1001 (opération i1 / i0), opérande DATA en valeur immédiate

JMPNZ #adresse : JMPNZ = 1, opérande DATA en adresse de saut

JMPZ #adresse : JMPZ = 1, opérande DATA en adresse de saut

JMPN #adresse : JMPN = 1, opérande DATA en adresse de saut

JMP #adresse : JMP = 1, opérande DATA en adresse de saut

CMP_B #valeur : ALU = 1010 (opération i1 CMP i0), opérande DATA en valeur immédiate

NOP : ALU = 0000 (opération NOP)

LOAD_A_C : Registre A = 1, MUX1 = 1, ALU = 0010

CMP_C_A : MUX1 = 1, MUX0 = 1, ALU = 1010 (opération : i1 CMP i0)

2 programmes réalisés

MOYENNE.ASM

Programme en assembleur moyenne.asm qui permet de calculer la moyenne entière des valeurs présentes dans la mémoire des données.

```
        LOAD_A #0
        LOAD_B #0
boucle  LOAD_C_ADRB
        ADD_A_C
        INC_B
        CMP_B #4
        JMPNZ boucle
        LOAD_B_A
        DIV_B #4
        LOAD_A_B
```

MAX.ASM

Programme en assembleur max.asm qui permet de chercher le maximum des valeurs présentes dans la mémoire de données et la copie dans le registre A.

```
boucle  LOAD_C_ADRB
        INC_B
        CMP_C_A
        JMPN boucle
        LOAD_A_C
        JMP boucle
```

Amélioration du circuit

adressage direct pour le registre C

Pour rendre possible le support d'une instruction telle que LOAD_C [1A5F] utilisant l'adressage direct, on pourrait par exemple ajouter sur le bit 14 du code opératoire un troisième MUX ("MUX2") qui prendrait comme entrée la sortie de l'ALU et la sortie du MUX1, et dont la sortie serait connectée à l'entrée A de la mémoire des données, comme ci-dessous :

On peut voir que, quand MUX2 a pour valeur 0, la valeur renvoyée à la mémoire de données est celle renvoyé par MUX1, tandis que quand MUX2 a comme valeur 1, la valeur renvoyée à la mémoire de données est la valeur provenant de la sortie de l'ALU, donc de DATA si MUX0 = 0 et que ALU = 0001.