

INF1600

Travail pratique 5

Unité de contrôle et gestion de la pile

Travail réalisé par :

El Harami, Mehdi (2113402)

Ouazani Chahdi, Rita (2178393)

Groupe Laboratoire : 02

Architecture des micro-ordinateurs (INF1600)

Département de Génie Informatique et Génie Logiciel

Polytechnique Montréal

BARÈME

TP 5		/4,00
		/4,00
Q1	/0,50	
Q2	/0,50	
Q3	/1,00	
Q4	/1,00	
Q5	/1,00	
Bonus	/0,25	

Section 3

Q1) Le programme permet de calculer la valeur à l'adresse de $n+1$. On peut traduire cela à l'aide de la formule mathématique :

$$vA(n+1) = 2(vAn) - (vA(n-2))$$

Où :

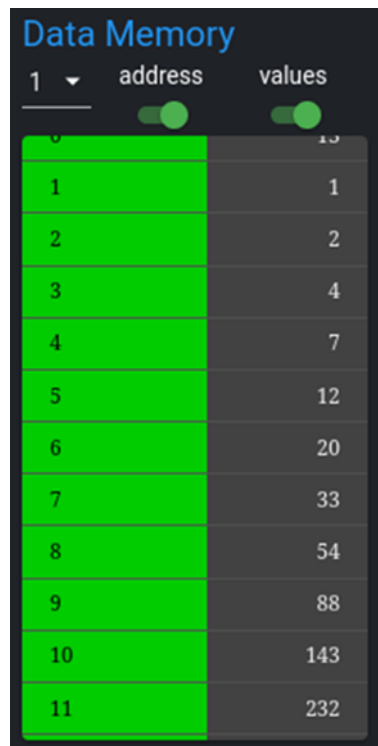
$vA(n+1)$: valeur à l'adresse de $(n+1)$

vAn : valeur à l'adresse de n

$vA(n-2)$: valeur à l'adresse de $(n-2)$

Ce qui nous permet de constater une similarité avec la suite de Fibonacci. En effet, on obtient les termes-1 de la suite de Fibonacci à partir du 3^{ème} terme.

Capture d'écran de nos résultats



1	address	values
1	1	1
2	2	2
3	3	4
4	4	7
5	5	12
6	6	20
7	7	33
8	8	54
9	9	88
10	10	143
11	11	232

Suite de Fibonacci

1
1
2
3
5
8
13
21
34
55
89
144
233

Et
ainsi
de
suite...

Q2)

Ligne 3 : ld r10, (r10)

$IR \leftarrow Mi[PC] : PC \leftarrow PC + 1;$

$Mi \leftarrow R[10];$

$R[10] \leftarrow Md[Mi];$

Ligne 8 : sub r14, r11, r10

$IR \leftarrow Mi[PC] : PC \leftarrow PC + 1;$

$Mi \leftarrow R[11] - R[10];$

$R[14] \leftarrow Mi;$

Ligne 16 : mv r2, r1

$IR \leftarrow Mi[PC] : PC \leftarrow PC + 1;$

$Md \leftarrow R[2];$

$R[1] \leftarrow Md;$

Ligne 20 : add r11, r11, r21

$IR \leftarrow Mi[PC] : PC \leftarrow PC + 1;$

$Mi \leftarrow R[11] + R[21];$

$R[11] \leftarrow Mi;$

Q3)

Ligne 3 :

Instruction à la ligne 3 :

	do_branch	sel_reg_data	op_alu	wr_reg	wr_mem	wr_pc	wr_ir
Cycle 1	0	x	x	0	0	1	1
Cycle 2	x	2	x	1	0	0	0
Cycle 3	x	x	x	1	0	0	0

Ligne 8 :

Instruction à la ligne 8 :

	do_branch	sel_reg_data	op_alu	wr_reg	wr_mem	wr_pc	wr_ir
Cycle 1	0	x	x	0	0	1	1
Cycle 2	x	2	2	0	0	0	0
Cycle 3	x	x	x	1	0	0	0

Ligne 16 :

Instruction à la ligne 16 :

	do_branch	sel_reg_data	op_alu	wr_reg	wr_mem	wr_pc	wr_ir
Cycle 1	0	x	x	0	0	1	1
Cycle 2	x	1	x	0	1	0	0
Cycle 3	x	x	x	1	0	0	0

Ligne 20 :

Instruction à la ligne 20 :

	do_branch	sel_reg_data	op_alu	wr_reg	wr_mem	wr_pc	wr_ir
Cycle 1	0	x	x	0	0	1	1
Cycle 2	x	0	2	0	0	0	0
Cycle 3	x	x	x	1	0	0	0

Q4)

.text

ldi r0, 250

st (r0), r0

ldi r0, 251

st (r0), r0

ldi r0, 252

st (r0), r0

ldi r0, 253

st (r0), r0

ldi r0, 254

st (r0), r0

ldi r0, 255

st (r0), r0

ldi r31, 256

ldi r0, 2

ldi r1, 3

ldi r2, 4

ldi r3, 5

ldi r4, 6

ldi r5, 7

ldi r29, 1

sub r31, r31, r29

st 0(r31), r0

sub r31, r31, r29

st 0(r31), r1

sub r31, r31, r29

st 0(r31), r2

sub r31, r31, r29

st 0(r31), r3

sub r31, r31, r29

st 0(r31), r4

sub r31, r31, r29

st 0(r31), r5

stop

Q5)

.text

ldi r0, 250

st (r0), r0

ldi r0, 251

st (r0), r0

ldi r0, 252

st (r0), r0

ldi r0, 253

st (r0), r0

ldi r0, 254

st (r0), r0

ldi r0, 255

st (r0), r0

ldi r31, 250

ldi r29, 1

ld r1, 0(r31)

add r31, r31, r29

ld r2, 0(r31)

add r31, r31, r29

ld r3, 0(r31)

add r31, r31, r29

ld r4, 0(r31)

add r31, r31, r29

ld r5, 0(r31)

add r31, r31, r29

ld r6, 0(r31)

add r31, r31, r29

stop

Question bonus

Pour pouvoir émuler l'instruction `ret`, il faudrait modifier le chemin de données en ajoutant un bus allant de la mémoire de données jusqu'au programme counter (PC).