

NOM:	Prénom:	N° Groupe:
------	---------	------------



Le corrigé détaillé est après le sujet

ESIAL 1^{ÈRE} ANNÉE
MODULE PFSI

ÉPREUVE DE CONSOLIDATION

Mardi 13 juin 2006

Les quatre parties sont indépendantes.

*Aucun document autre que
la documentation fournie
n'est autorisé;*

L'épreuve est individuelle;

Durée: 2 heures

1. CODAGE DE L'INFORMATION (RÉPONDRE SUR LA FEUILLE ET AU DOS)

Le format IEEE754 est le suivant : (<signe> <exp> <mantisse>)

1.1. Coder au format IEEE 754 simple précision les nombres de **R** suivants:

a/ +0,0

b/ $-\infty$

c/ 3,1415926535

1.2 Quelle est le **plus grand nombre** avant $+\infty$ que l'on peut coder (qui ne soit pas un NaN) ?

1.3. Quel est le **plus petit** avant +0 (on tiendra compte de la dénormalisation pour les nombres très petits) ?

1.4 Quelle **résolution relative** a t-on ? (différence relative entre deux nombres codables consécutifs)

1.5 Quelle **erreur relative maximale** a t-on ?

1.6 Quelle **erreur relative moyenne** a t-on lorsque l'on code des nombres réels quelconques ?

1.7. On suppose que la mémoire d'un système informatique "Big Endian" contient les octets suivants :

45	FF	FC	51	FF	12	34	00
----	----	----	----	----	----	----	----

Ces octets sont interprétés comme des entiers codés selon les formats ci-dessous, donner les valeurs représentées :

- a) Ces **Non signés** codés sur 16 bits :
- b) **Signés** codés sur 16 bits en **code complément à deux** :
- c) **Signés codés par biais** sur 16 bits (préciser la valeur du biais) :

NOM:	Prénom:	N° Groupe:
-------------	----------------	-------------------

2. ANALYSE ET SYNTHÈSE D'OPÉRATEUR LOGIQUE (RÉPONDRE AU DOS DE CETTE FEUILLE)

On considère l'opérateur ALU (Arithmetic & Logic Unit) dont les spécifications suivent :

Interface:

Nom	Sens	Taille en bits	Commentaire
P	entrée	1	commande de passage de <u>B</u>
E	entrée	1	commande de sortie nulle
I	entrée	1	retenue d'entrée
<u>A</u>	entrée	4	opérande A en code binaire
<u>B</u>	entrée	4	opérande B en code binaire
<u>S</u>	sortie	4	résultat en code binaire

Comportement :

PE	<u>S</u>
00	<u>A</u> # I
-1	<u>0</u>
10	<u>B</u>

2.1. Dessiner la **représentation symbolique** (interface) de cet opérateur.

2.2. Dessiner le **schéma synoptique de la structure** de cet opérateur par une décomposition en tranches de 1 bit de large.

2.3. Dessiner la **représentation symbolique (interface) d'une tranche**.

2.4. Spécifier le **comportement d'une tranche** à l'aide d'un tableau donnant pour chaque valeur de la commande PE l'expression algébrique des sorties en fonction des autres entrées.

2.5. En déduire les **polynômes booléens réduits** (i.e. forme normale disjonctive réduite) de chacune des sorties d'une tranche en utilisant une table de Karnaugh.

2.6. Dessiner la **structure interne d'un opérateur MYST** de spécifications suivantes en utilisant l'opérateur AU et éventuellement **un seul** opérateur classique parmi : NOT, AND, OR, MUX, ADD, LFF, DFF, REG. On utilisera une machine de Moore simplifiée.

Interface :

Nom	Sens	Taille en bits	Commentaire
k	entrée	1	Horloge (« clocK »)
T	entrée	1	Commande d'incréméntation ("Toggle")
R	entrée	1	Commande de remise à zéro ("Reset")
L	entrée	1	Commande de chargement ("Load")
<u>D</u>	entrée	4	Entrée de donnée ("Data")
<u>Q</u>	sortie	4	Donnée contenue dans l'opérateur

Comportement:

Événement	Action	Commentaires
R=1 et $k \uparrow$	$Q \leftarrow 0$	Remise à zéro
L=1 et $k \uparrow$	$Q \leftarrow D$	Chargement
L=0 et T=1 et $k \uparrow$	$Q \leftarrow Q \# 1$	Incréméntation

(Lorsque k ne monte pas et **pour les autres événements**, Q garde sa valeur).

2.7. Quel est cet opérateur MYST ?

NOM:	Prénom:	N° Groupe:
------	---------	------------

3. MICROPROGRAMMATION (RÉPONDRE SUR LA FEUILLE)

On considère la machine décrite dans la documentation (identique à celle vue en cours)

3.1. Donner toutes les **actions** (i.e. qui changent l'état de la machine, c'est à dire affectations des cases mémoire, des compteurs, des registres ou des bascules d'indicateurs) nécessaires pour effectuer *l'instruction complète* suivante :

ADI #iv, R2, R4

En particulier, cette instruction "ADd Immediate":

*charge le registre R4 la somme de iv et R2 ;

* modifie les indicateurs du registre d'état en fonction de la somme.

3.2. Déterminer la **valeur numérique** (*en binaire*) de chacun des **champs** du code de micro-instruction (**MIC**, « Micro-Instruction Code ») (répondre sur la feuille du sujet).

Les champs dont la valeur est sans importance pourront être marqués -.

cycle	ALU + A			REG FILE			QV	μV			C BUS			X	SR	BC	PC		IR	μPC
	P	L	E	S	L	E	E	μVC	E	W	ADS	L	L	CC	T	E	L	R*		
0																				
1																				
2																				
3																				
4																				
5																				
6																				
7																				

7

I CODAGE (suite)

2

on suppose que la machine est "big endian" comme celle du TP.

	S		
45FF	0	000101111111	
FC51	1	111110001010001	
FF12	1	111111100010010	
3400	0	011010000000000	

Hexa	Décimal non signé
45FF	17919
FC51	64593
FF12	65298
3400	13312

valeur = code

c

Signé biaisé

45FF	→ 17919 - biais =	-14848
FC51	→ 64593 - biais =	31826
FF12	→ 65298 - biais =	32531
3400	→ 13312 - biais =	-19455

valeur = code - biais

biais = $2^{N-1} - 1$

biais = $2^{16-1} - 1 = 32767$

signé code C52

b

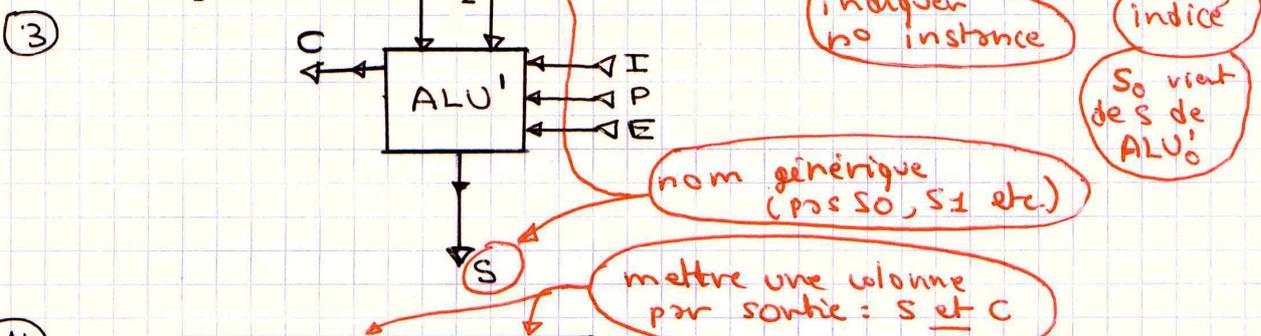
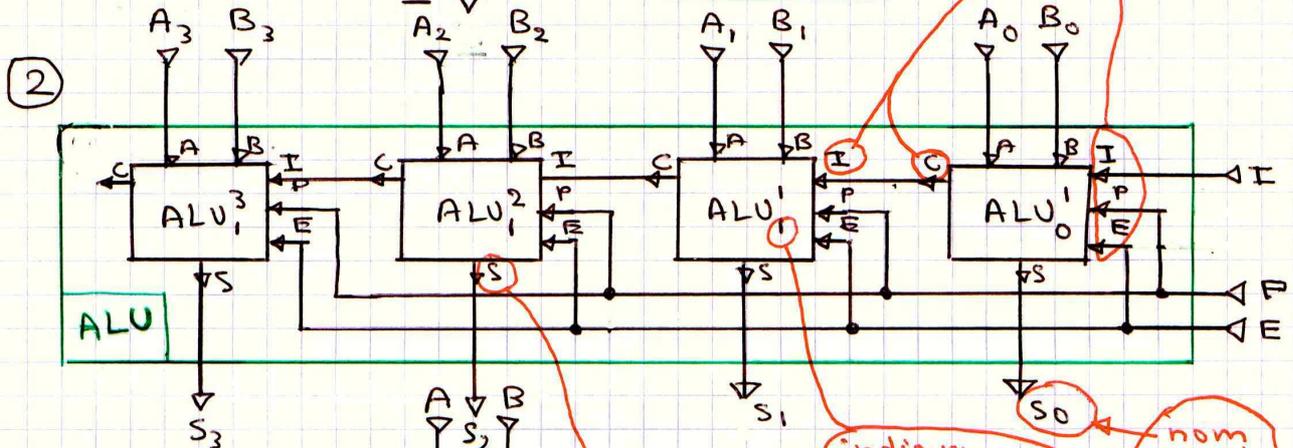
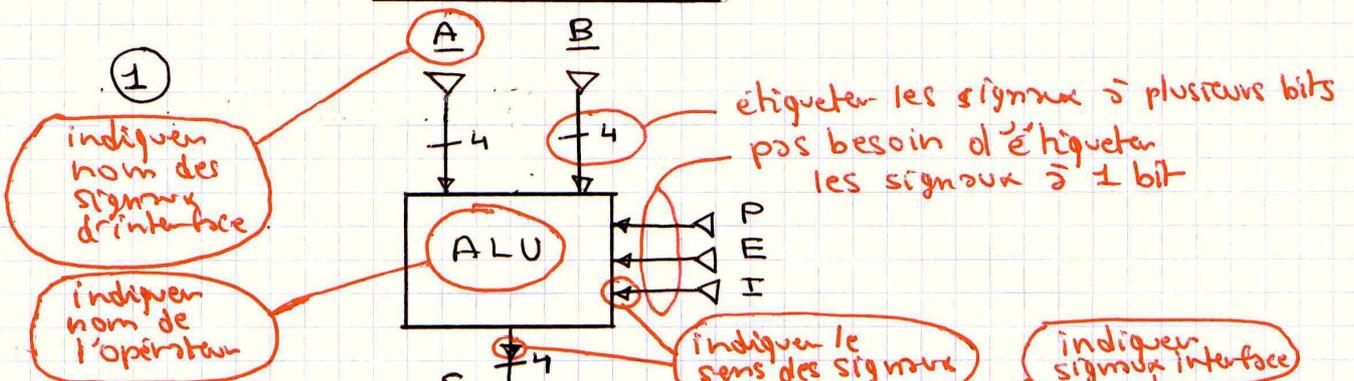
45FF	→ signe +	→ aucun changement	17919
FC51	→ signe -	→ $64593 - 2^{16} =$	-943
FF12	→ signe -	→ $65298 - 2^{16} =$	-238
3400	→ signe +	→ aucun changement	13312

indiqué par bit de signe à gauche
0 ⇒ +, 1 ⇒ -

$\left\{ \begin{array}{l} \text{valeur} = \text{code si } + \\ \text{valeur} = \text{Complément } \rightarrow 2 \text{ de code si } - \end{array} \right.$
 complément $\rightarrow 2$ de code = $\overline{\text{code}} + 1$
 = $\text{code} - 2^N$

II LOGIQUE

(3)



④

PE	S	C
00	$A \oplus I$	$A \cdot I$
11	\emptyset	—
10	B	—

indiquer que C ne sert pas ici
 mettre des expressions algébriques
 mettre une ligne par valeur d'entrée de commande

④

II LOGIQUE (suite)

⑦ Compteur binaire synchrone rechargeable avec remise à zéro synchrone.

ordre binaire réfléchi

⑤

PE \ IAB	000	001	011	010	110	111	101	100
00	0	0	1	1	0	0	1	1
01	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
10	0	1	1	0	0	1	1	0

valeurs commandés

symétrie / médiane

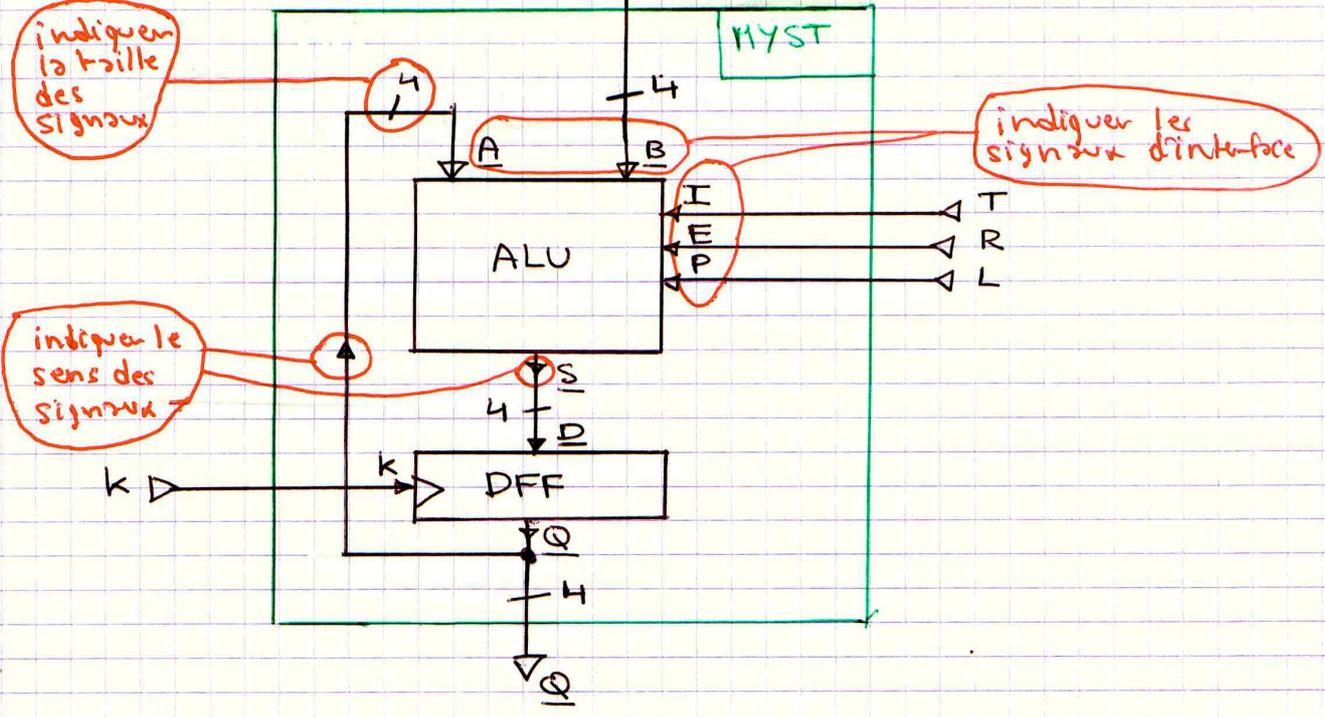
$$S = \bar{P} \cdot \bar{E} \cdot \bar{I} \cdot A \quad \textcircled{1}$$

$$+ \bar{P} \cdot \bar{E} \cdot I \cdot \bar{A} \quad \textcircled{2}$$

$$+ P \cdot \bar{E} \cdot B \quad \textcircled{3}$$

$$C = A \cdot I \quad \text{directement d'après } \textcircled{4}$$

⑥



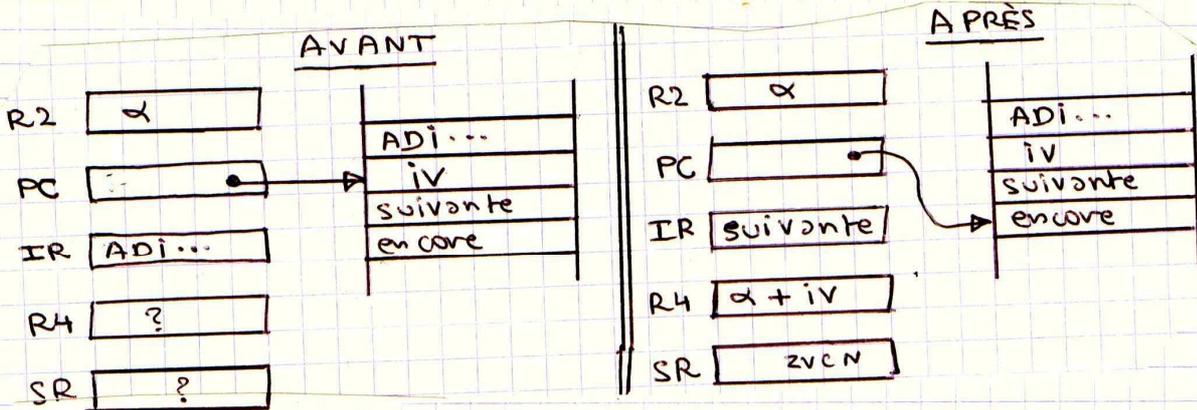
5

III MICRO PROGRAMMATION

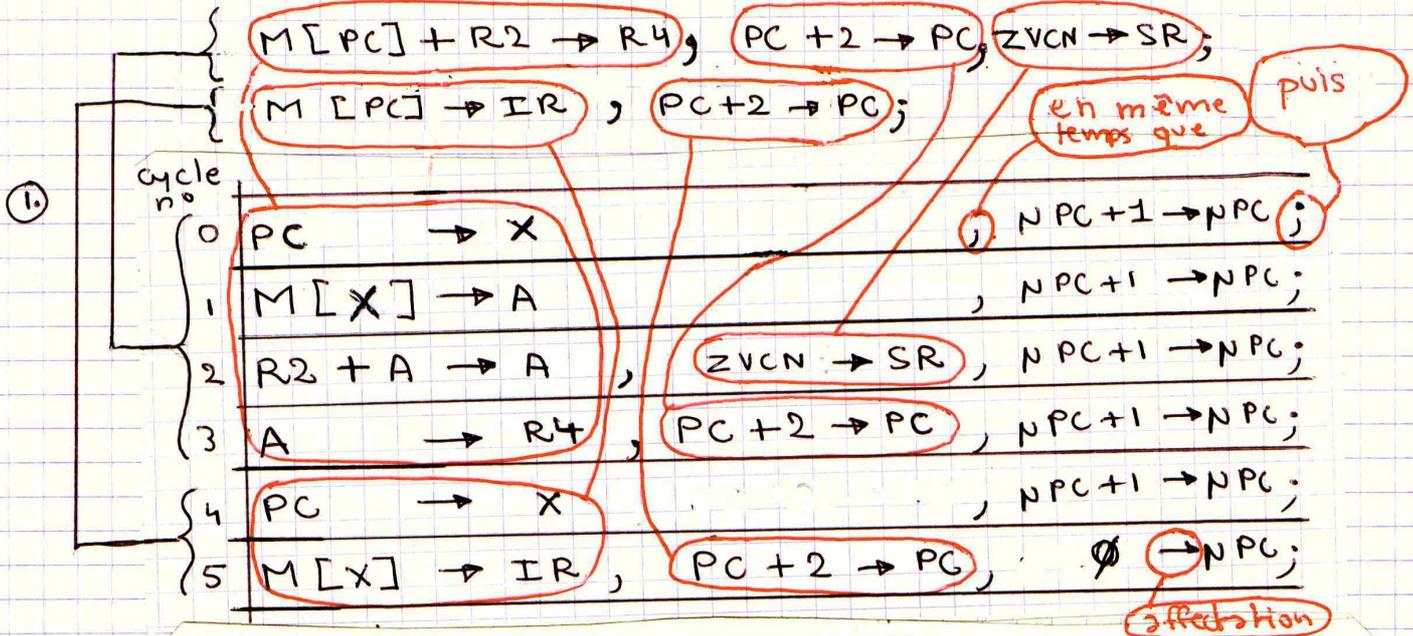
iv = "Immediate Value" suit le code d'instruction et pointée par PC.

Explications:
(Ressemble à

LDW R4, #iv
cf. TD)



On veut faire globalement:

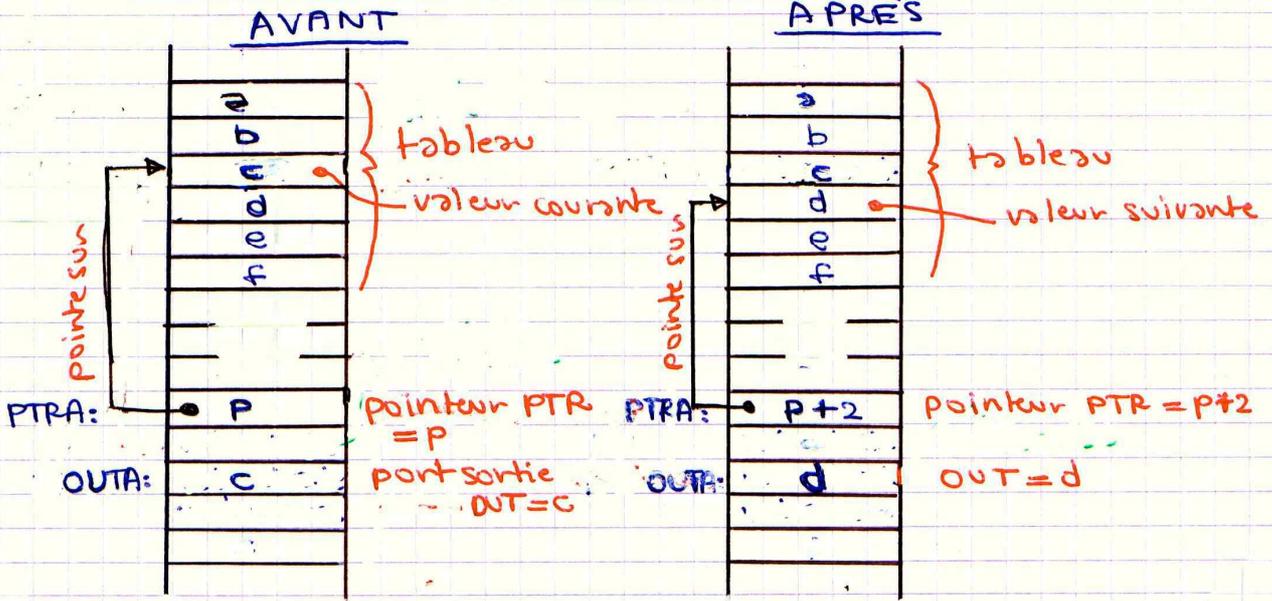


2.

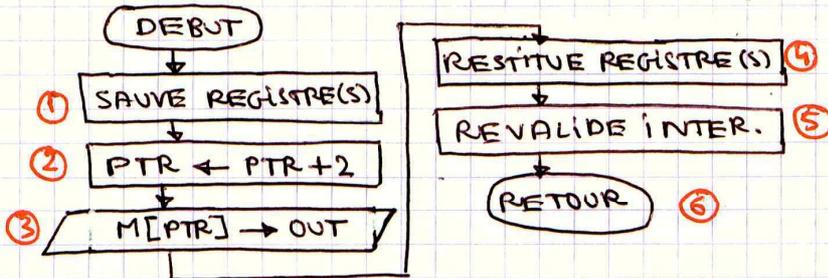
cycle	ALU + A			REG FILE			QV	μV	CBUS			X	SR	BC	PC	IR	μPC	
	P	L	E	S	L	E	E	μVC	E	W	ADS	L	L	CC	T	E	L	R*
0	---	0	---	0	0	0	0	--	0	0	0	1	0	0000	0	1	0	0
1	0000	1	0	---	0	0	0	--	0	0	1	0	0	0000	0	0	0	0
2	0001	1	0	0010	0	1	0	--	0	0	0	0	1	0000	0	0	0	0
3	---	1	0	1001	0	0	0	--	0	0	0	0	0	0000	1	0	0	0
4	---	0	---	0	0	0	0	--	0	0	0	1	0	0000	0	1	0	0
5	---	0	---	0	0	0	0	--	0	0	1	0	0	0000	1	0	1	1

6

IV PROGRAMMATION MACHINE



ORGANIGRAMME



ADRS	HEXA	CODE BINAIRE	ASSEMBLEUR	COMMENTAIRE
			EQ SP, R15	SP remplacé par R15
			EQ OUTA, 0xF004	OUTA ——— 0xF004
			EQ PTR, 0xF000	PTRA ——— 0xF000
A000	614F	0110 0001 0100 1111	STW R1, -(SP)	save R1 dans la pile
A002	61D0	0110 0001 1101 0000	LDW R1, @PTR	R1 ← PTR = M[PTRA]
A004	F000			change le pointeur dans R1
A006	3102	0011 0001 0000 0010	ADQ 2, R1	R1 ← = PTR+2
A008	6150	0110 0001 0101 0000	STW R1, @PTR	R1 = PTR+2 → M[PTRA] = PTR
A00A	F000			savegarde le nouveau pointeur
A00C	61A1	0110 0001 010 0001	LDW R1, (R1)	R1 ← valeur suivante
A00E	6150	0110 0001 0101 0000	STW R1, @OUTA	valeur suivante → OUT = M[OUTA]
A010	F004			
A012	61BF	0110 0001 011 1111	LDW R1, (SP)+	restitue R1
A014	0700	0000 111 0000 0000	ENI	revalide les interruptions
A016	0300	0000 011 0000 0000	RTI	retourne au prog. principal

①
②
③
④
⑤
⑥