

Architecture des ordinateurs

TD 1

Exercice n°1:

$$X_1 = 01100$$

$$X_2 = 01101$$

$$X_3 = 10111$$

$$X_4 = 10100$$

a) bit de poids fort = 0 $\Rightarrow$ nombre positif = pas besoin du complément
= 1 $\Rightarrow$ — négatif = besoin —

$$011011 = \oplus 27$$

$$101010 = -22 \quad \text{car: complément à 2 (car 1 à gauche)}$$

$\rightarrow$ complément à 1 ($x \rightarrow \bar{x}$) puis +1
(ce qui donne complément à 2)

$$\begin{array}{r} 101010 \rightarrow 010101 \\ + \quad \quad \quad 1 \\ \hline 010110 \rightarrow 22 \end{array}$$

intervalle: avec n le nombre de bits:

$$[(-2^{n-1}), (2^{n-1} - 1)]$$

b) $X_1 = 01100_2 \Rightarrow (8+4)_2 = 12_{10}$

$$X_2 = 01101_2 \Rightarrow (8+4+1)_2 = 13_{10}$$

$$\begin{aligned} X_3 = 10111_2 &\Rightarrow \bar{X}_3 = 01000_2 \\ \bar{X}_3 + 1 &= 01001_2 = X_3 = -(8+1) = -9 \end{aligned}$$

$$X_4 = 10100|_2, \quad \overline{X}_4 = 01011|_2, \quad X_4 = -(4+8) = -12$$

$$\overline{X}_{4+1} = 01100|_2$$

c) $X_2 - X_1 \Leftrightarrow X_2 + X_4 \quad | \quad X_4 = -(X_1)$

$$\begin{array}{r} 01101 \\ + 10100 \\ \hline 00001 = 1 \end{array}$$

$$\underline{X_1 - X_2} = X_1 + (-X_2)$$

$$-X_2 = 10011 \quad (\overline{x}_2 + 1)$$

$$\begin{array}{r} X_1 - X_2 = 11111 = -1 \\ = \begin{array}{r} 00000 \\ 1 \\ \hline 1 \end{array} \end{array}$$

$$\underline{X_1 + X_3} = \begin{array}{r} 01100 \\ + 10111 \\ \hline 10011 = 3 \end{array}$$

$$\underline{X_1 + X_2} = \overset{\text{xxx}}{\textcircled{-7}} \quad (25 \text{ normalerent})$$

$$\begin{array}{r} 01100 \\ + 01101 \\ \hline 11001 = -7 \\ \text{complément:} \\ 00110 \\ + 1 \\ \hline 00111 = 7 \end{array}$$

$$\begin{array}{r} X_3 + X_4 \\ 10111 \\ + 10100 \\ \hline 101011 = \overset{\text{xxx}}{\textcircled{+11}} \quad (-21 \text{ normalerent}) \end{array}$$

d) Overflow

e)

bit poids fort du 1^{er} (ex: X_2) bit de poids fort du 2^{eme} (ex: X_1)

$$V = \overline{a_{m-1}} \overline{b_{m-1}} S_{m-1} + a_{m-1} b_{m-1} \overline{S_{m-1}}$$

bit de poids fort du résultat ($X_2 - X_1$)

$-X_2 = 10100$

$X_1 = 01100$

$X_2 = 01101$

$-X_2 = 10111$

$X_3 = 10111$

$X_4 = 10100$

$X_2 - X_1 = 00001$

$X_1 - X_2 = 11111$

$X_1 + X_3 = 00011$

$X_1 + X_2 = 11001$

$X_3 + X_4 = 01011$

$V_{Op1} = 1 \times 0 \times 0 + 0 \times 1 \times 1 = 0$

$V_{Op2} = 1 \times 0 \times 1 + 0 \times 1 \times 0 = 0$

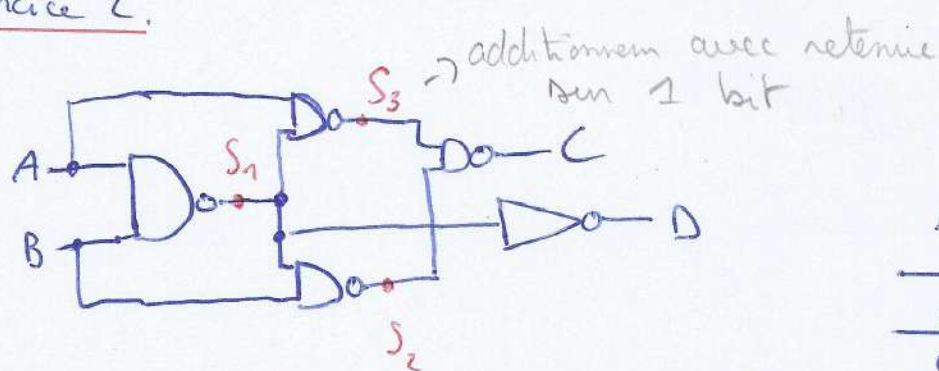
$V_{Op3} = 1 \times 0 \times 0 + 0 \times 1 \times 1 = 0$

$V_{Op4} = 1 \times 1 \times 1 + 0 \times 0 \times 0 = 1$

$V_{Op5} = 0 \times 0 \times 0 + 1 \times 1 \times 1 = 1$

Quand $V = 1$; on était en overflow

Exercice 2.



A	B	S_1	S_2	S_3	C	D
0	0	1	1	1	0	0
0	1	1	0	1	1	0
1	0	1	1	0	1	0
1	1	0	1	1	0	1

$S_1 = (A \cdot B)$

$S_2 = (\overline{S_1} \cdot B) = B \cdot (\overline{A \cdot B}) = \overline{A \cdot B} \cdot B$

$S_3 = (\overline{S_1} \cdot A) = A \cdot (\overline{A \cdot B}) = \overline{A \cdot B} \cdot A$

simple

$C = S_3 \cdot S_2 = (\overline{A \cdot B} \cdot A) \cdot (\overline{A \cdot B} \cdot B) = \overline{A \cdot B} \cdot A \cdot B$

$D = \overline{S_1} = \overline{A \cdot B} = \overline{AB}$

$C = \overline{AB} \cdot A \cdot B$
 $= \overline{AB} \cdot A \cdot B = \overline{AB} \cdot A \cdot B = (\overline{A} + \overline{B}) \cdot A \cdot B = \overline{A} \cdot A \cdot B + \overline{B} \cdot A \cdot B = \overline{A} \cdot B + \overline{B} \cdot A$

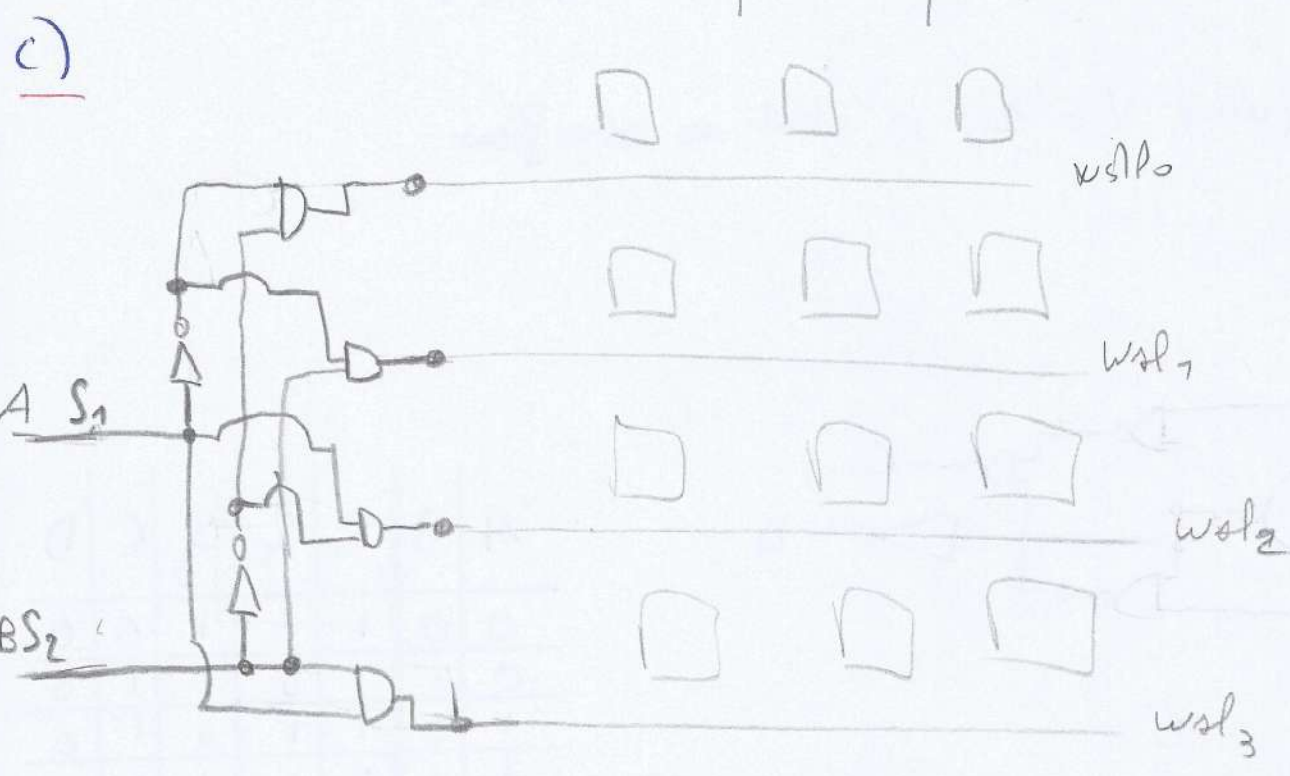
Exercice 3.

a) 2 fils pour adresser 4 mots mémoire.

A	B	
0	0	→ mot 0
0	1	→ — 1
1	0	→ — 2
1	1	→ — 3

Circuit : décodeur

b) 3 bits → 3 fils



"0" représente par fil + inverseur.
 porte AND car $A \text{ et } B = 0$.

d) d1) Si on veut écrire: $C_S = 1$
 $OE = 0$
 $R/\overline{W} = 0$

Si on veut lire: $C_S = 1$
 $OE = 1$
 $R/\overline{W} = 1$

d2) On les branche sur le CLK

d3) Voir feuille

d4) sur les fils WSL

e) sur D_0, D_1, D_2 . En écrivant.

f) Sur Q_0, Q_1, Q_2 . Sur écriture

g) g1) multiplexeur: plusieurs entrées; 1 sortie, qui est l'une des entrées. Signal de sélection

g2) Select°: WSL
entrées: Q_2, Q_1, Q_0

i) bascules: 12
ou: 3
et: 22

Pour 1024 octets:

lignes d'adresse: 10
fils de données: 8
bascules: 8192 (8×1024)
et: $1024 \times 10 + 2 = 10242$ (3)
ou: 8

ADD TD1

Exercice 4.

1) $01110 \rightarrow 14 \rightarrow \text{da'g} : 11100 \rightarrow 28 \rightarrow \text{da'd} : 00111 \rightarrow 7.$
 $01111 \rightarrow 15 \quad \text{---} \quad 11110 \rightarrow 30 \quad \text{---} \quad 00111 \rightarrow 7$
 $11110 \rightarrow 30 \quad \text{---} \quad 11100 \rightarrow 28 \quad \text{---} \quad 01110 \rightarrow 14$
 $11111 \rightarrow 31 \quad \text{---} \quad 11110 \rightarrow 30 \quad \text{---} \quad 01111 \rightarrow 15$

da'g: division par 2

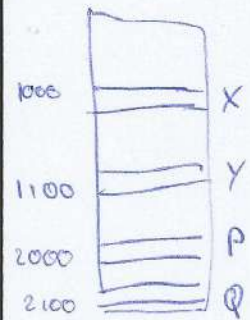
da'd: multiplier par 2

sans si le 1 en poids fort disparaît

2) Il faut pouvoir conserver le bit de poids fort (avec un accumulateur)

ADD TD2

Exercise 1:

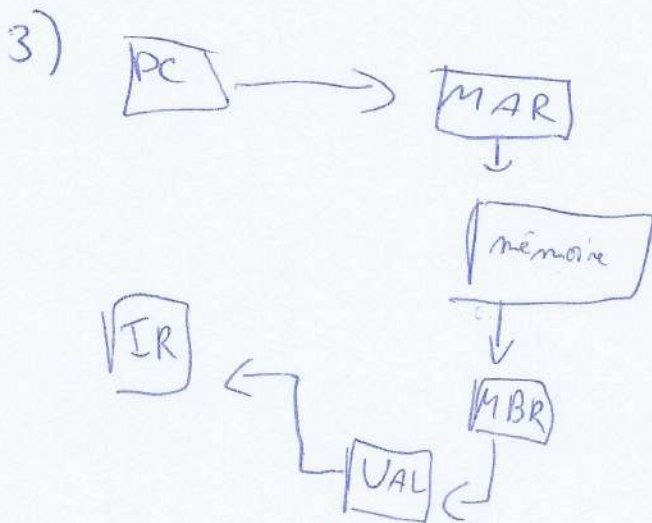


1)

LDA 2000	0	LDA 2100	0
ADD 2100	1	ADD 2000	1
STA 1000	2	STA 1000	2

2)

CMP #4	3	CMP #4	3
BRZ 9	4	BRZ 8	4
LDA #2	5	LDA 2100	5
ADD 2100	6	ADD #2	6
^{STA} 1100	7	BRA 9	7
BRA 11	8	LDA #6	8
LDA #6	9	STA 1100	9
STA 1100	10	STOP	10
STOP	11		



64 32 16 8 4 2 1

a) 0 1 1 1 0 $\rightarrow$ 14 $\rightarrow$ da'g : 1 1 1 0 0 $\rightarrow$ 28 $\rightarrow$ da'd $\rightarrow$ 0 0 1 1 1 $\rightarrow$ 7
 0 1 1 1 1 $\rightarrow$ 15 . da'g $\rightarrow$ 1 1 1 1 0 $\rightarrow$ 30 . da'd $\rightarrow$ 0 0 1 1 1 $\rightarrow$ 7 Δ 16 8 4 2 1
 1 1 1 1 0 $\rightarrow$ 30 . da'g $\rightarrow$ 1 1 1 0 0 $\rightarrow$ 28 da'd $\rightarrow$ 0 1 1 1 0 $\rightarrow$ 14
 1 1 1 1 1 $\rightarrow$ 31 da'g $\rightarrow$ 1 1 1 1 0 $\rightarrow$ 30 da'd $\rightarrow$ 0 1 1 1 1 $\rightarrow$ 15

da'g : nbre₁₀ $\times$ 2 da'd : nbre₁₀ $\div$ 2

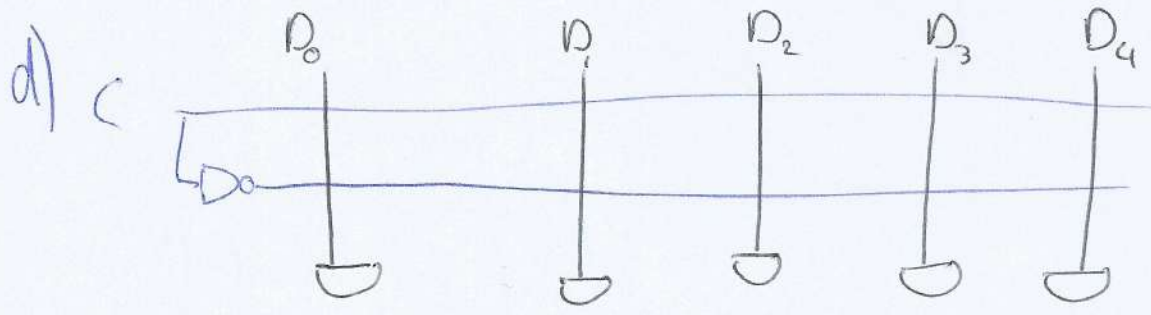
Si assez de bits.

da'd : diviseur (par 2)

da'g : multiplicateur (par 2)

Donc réserve d'assez de bits pour
conserver les 1

b) En binaire ~~D₀ sur D₄ ce qui conserve le 1 et reste dans la logique~~
conserver le bit de poids fort



CMP#4
BRZ

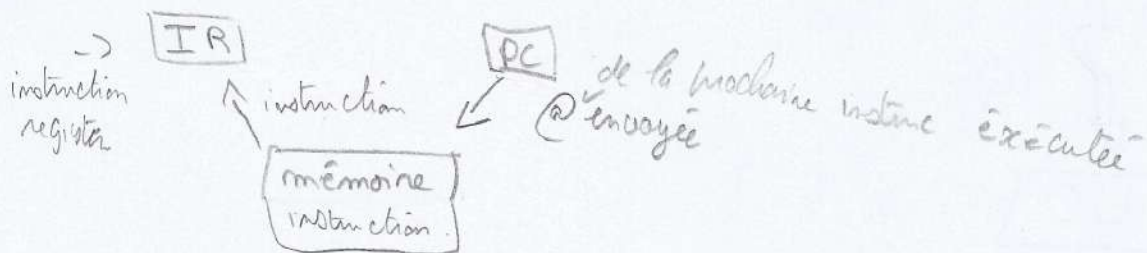
1)

		AD0		TD3							
	move	ADD	SUB	MUL	OR	AND	XOR	NOT	CMP		
const. const	x	x	x	x		x		reg			
mem. mem	x	x	x	x		x					
reg. reg	✓	✓	✓	✓		✓			✓		
const. mem	✓	x	x	x		x					
const. reg	✓	✓	✓	✓		✓			✓		
mem. const	x	x	x	x		x					
mem. reg	✓	x	x	x		x					
reg. const	x	x	x	x		x					
reg. mem	✓	x	x	x		x			✓		

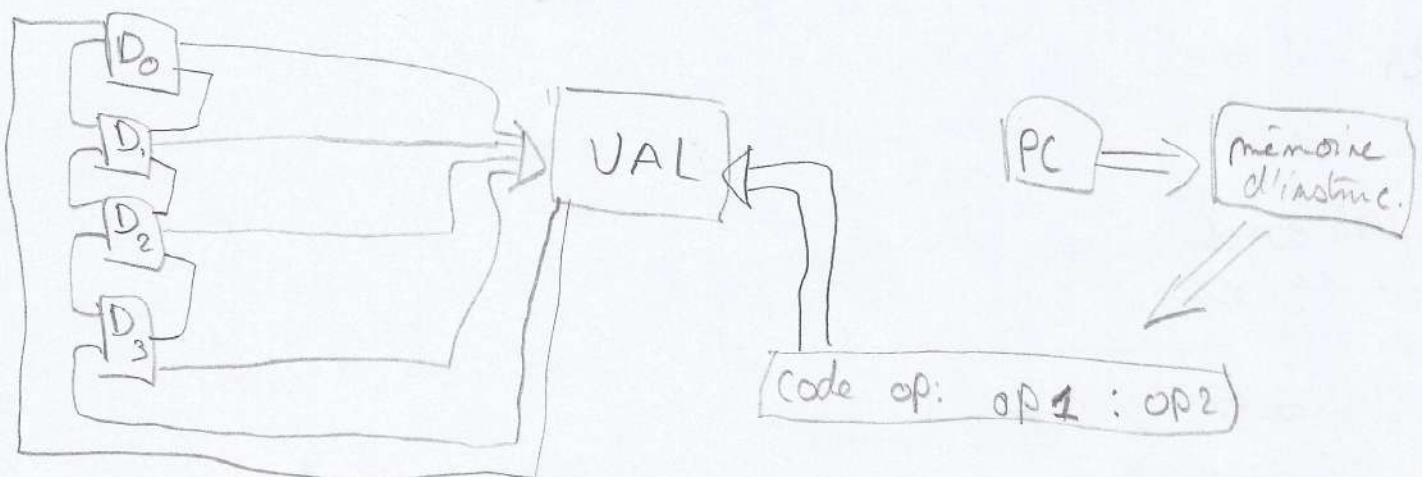
Bra const → BRC reg
 Bra registre → BRN const
 BRZ const → BRN reg
 BRZ Reg
 BRC const

2) Fetch pour récupérer l'instruction
Exécute

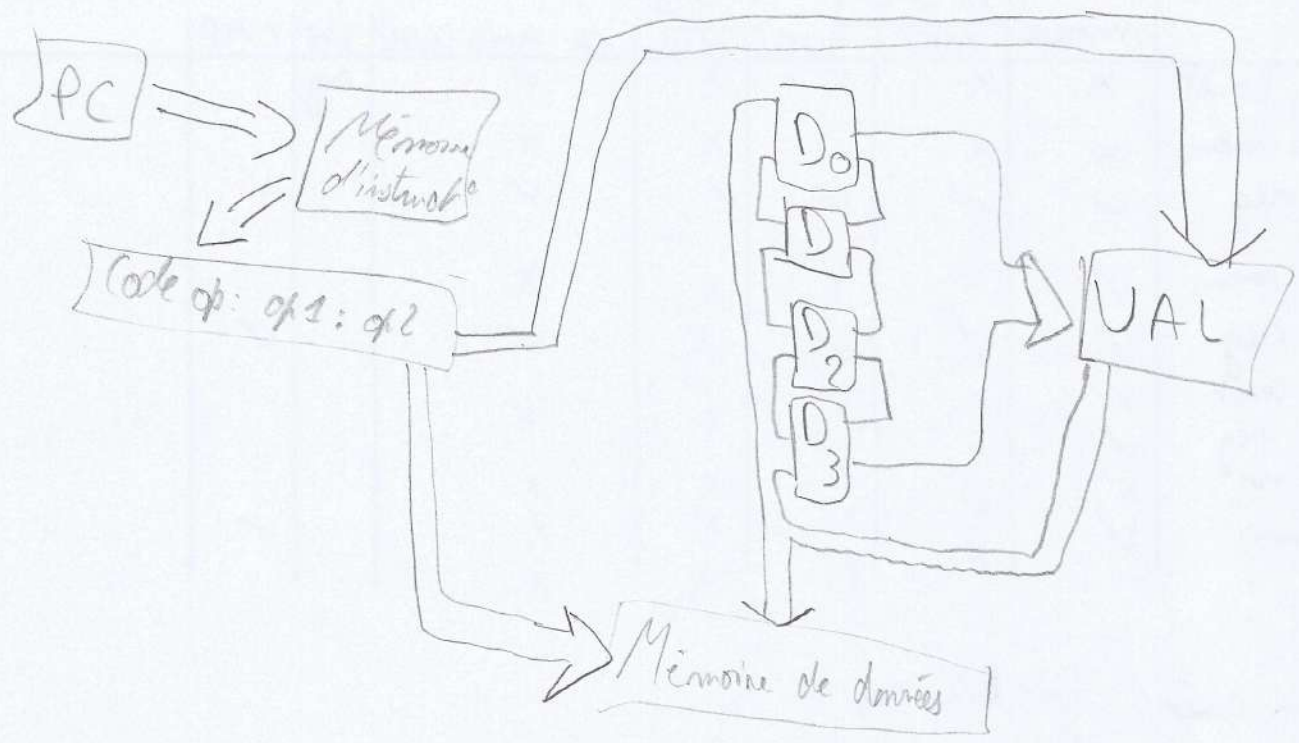
3)



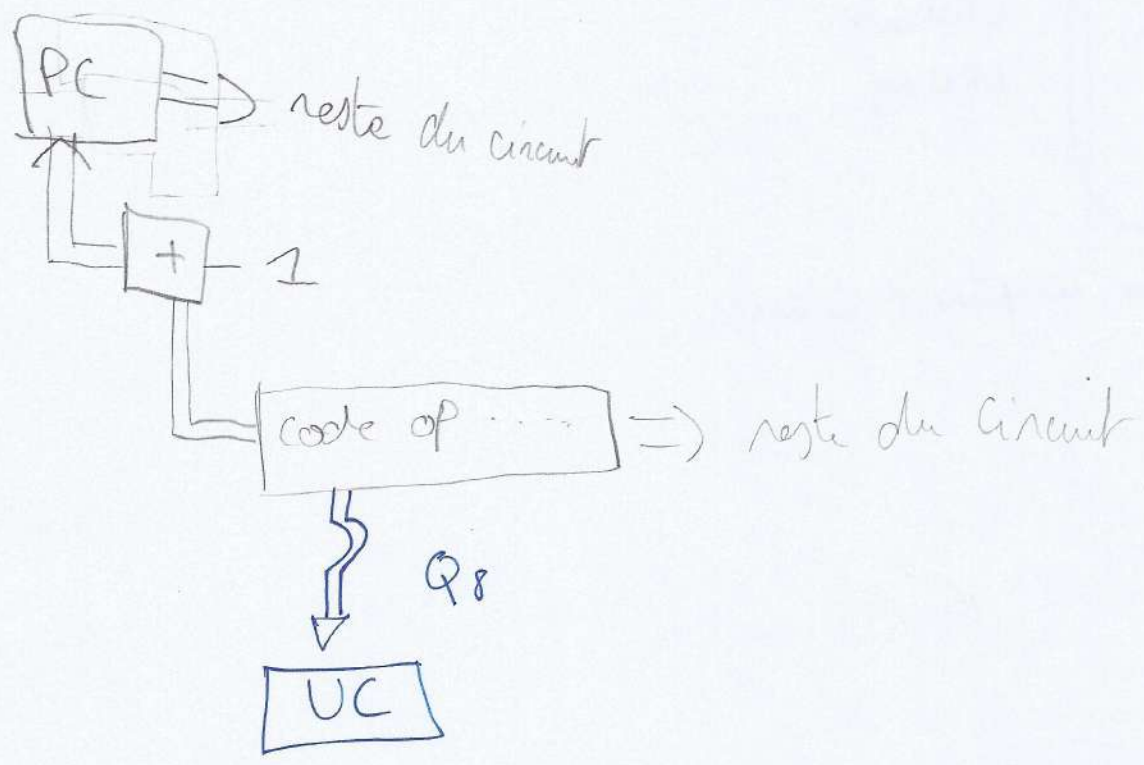
4)



6)



7)

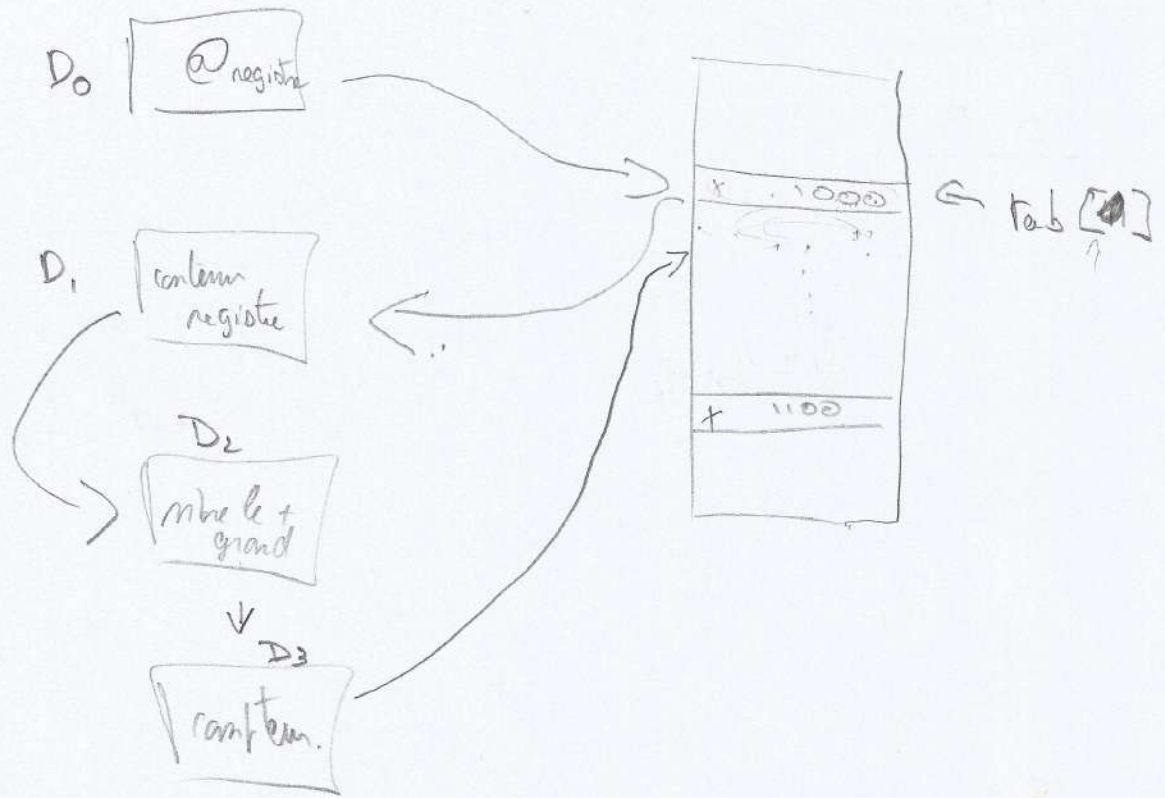


8) On a 31 codes donc il faut 5 bits.

9) ADD #302, D3.

L'instruction, pointée par le registre PC, est lue dans la mémoire d'instructions et transférée dans le registre IR. La partie code est une version de l'instruction d'addition, elle est dirigée vers le séquenceur pour organiser le chemin des données. Le 1^{er} opérande est la val. immédiate, 302 envoyé à l'UAL depuis l'IR. Le 2^{ème} ——— le contenu d'un registre dont le nom figure dans l'opérande. L'UC demande à un décodeur de sélectionner le registre et demande le transfert de son contenu vers l'UAL.

TD3
10)



= val immediate

Move #1000, D₀

Move 1000, D₂

Move (D₀), D₁

Move #100, D₃

(1000)

15.

15.

TD 4

Exo 1

a)

Petit boustiflé

3	2	1	0
7	6	5	4
11	10	9	8
15	14	13	12
19	18	17	16

Gros boustiflé

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19

b) Petit boustiflé

	C	V	L
a	v	u	D
		10	p
00	00	00	IF
00	00	01	04

age : 0... 11111
00 00 00 , F

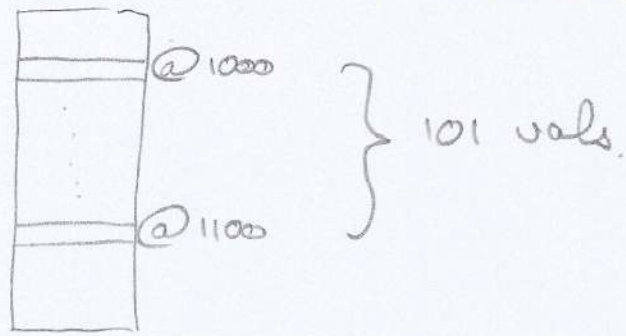
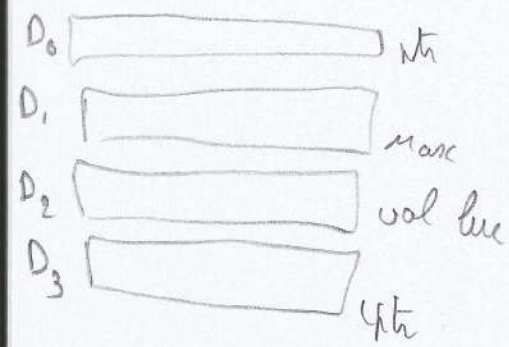
n° de série : $260 = 256 + 4$
= bimaine...

Gros boustiflé

Parait mais dans le bon sens

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19

comparateur numérique.



Move 1000, D_1
Move #1001, D_0

Move (D_0), D_2

Move #100, D_3

CMP D_2 , D_1

BRN 2

BRA 2

Move D_2 , D_1

ADD #1, D_0

Sub #1, D_3

BRZ 2

BRA -7

STOP

: val immédiate

#1000 = 1000

(D_0) = contenu
de l'adresse
contenue dans
 D_0

On saute 2 étages
si flags actives

Le calcul s'effectue dans l'UAL. Une fois terminé, l'UC demande le transfert (via le décodeur) du résultat dans le registre D_3 où le résultat sera écrit. Le PC s'incrémente de 1 et les flags sont reset.
La mem. n'est pas sollicitée ici

CMP D_2 D_4

FETCH.

La partie code est envoyée vers l'instructeur de comparaison, elle est dirigée vers l'UC pour organiser le chemin de données.

Les 1er et 2e opérande sont des contenus de registres dont les noms sont précisés dans l'opérande. L'UC demande à un décodeur de sélectionner le registre D_2 et ordonne le transfert de son contenu vers l'UAL.

De m pour D_4 .

Dans l'UAL, la soustraction entre l'opérande 2 et l'opérande 1 s'effectue. Les drapeaux sont mis à jour. L'UC demande à

Le PC s'incrémente de 1

La mémoire n'est pas sollicitée

Overflow	Négatif
V	N
Zero	retenu
Z	C