

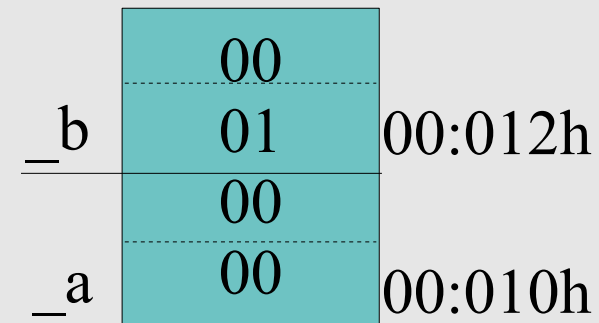
::: Architecture :::

semaine n°2

Premières instructions et modes d'adressages

Modifier les valeurs des registres R0 à R7 tel que le ferait le programme suivant.
À droite deux variables « a » et « b » en mémoire.

```
MOV R0, #10h
MOV R1, #20h
MOV WR2, WR0
MOV DR4, DR0
MOV R0, _b
MOV WR2, _b
```



R4 : 01	R5 : 02	R6 : 03	R7 : 04
R0 : 00	R1 : 00	R2 : 00	R3 : 00

Modifier les valeurs des registres R0 à R7 tel que le ferait le programme suivant.
À droite deux variables « a » et « b » en mémoire.

```
MOV R0, #10h
MOV R1, #20h
MOV WR2, WR0
MOV DR4, DR0
MOV R0, _b
MOV WR2, _b
```

	00	
<u>_b</u>	01	00:012h
	00	
<u>_a</u>	00	00:010h

R4 : 10h	R5 : 20h	R6 : 10h	R7 : 20h
R0 : 01h	R1 : 20h	R2 : 01h	R3 : 00h

Les opérations logiques :

- ANL - Et logique, bit à bit

AND RegD, RegS

AND RegD, #direct

RegD \leftarrow RegD & RegS

RegD \leftarrow RegD & direct

- ORL – OU logique, bit à bit
- XRL – OU EXCLUSIF, bit à bit

Etiquettes et directives

- Voir exercice page 14

Etiquettes et directives

ORG 4000h

NB: DB 5

TAB1: DW 4000h, 1, -8, 25

TAB2: DD 0,259

NB :	05h	00:4000h
TAB1 :	40h	00:4001h
	00h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	00h	00:4009h
	00h	00:400Ah
	00h	00:400Bh
	00h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Le mode implicite : Registre à registre

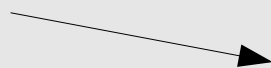
- La source et la destination sont des registres
 - MOV R0, R1
 - ADD R2, R4
- La source et la destination en en général la même taille, mais il existe des instructions pour passer d'une taille à une autre :
MOVH DR0, WR6 : copie WR6 dans les poids forts de DR0 sans modifier les poids faibles :

MOVH DR0, WR6



DR0 : 00 01 02 03

WR6 : 04 05



DR0 : 04 05 02 03

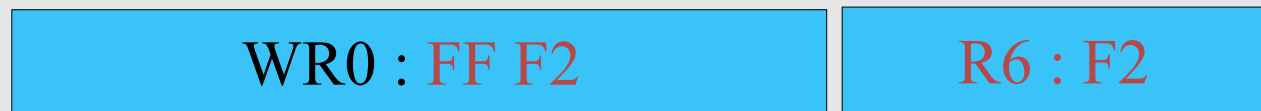
WR6 : 04 05

Le mode implicite : Register à registre

- MOVZ WR_x, R_x
 - Copie R_x dans les poids faible et force les poids forts à 0



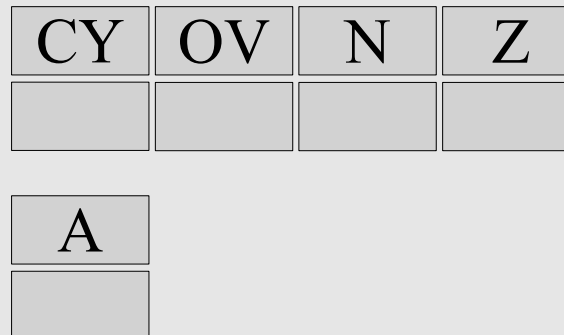
- MOVS WR_x, R_x
 - Copie R_x dans les poids faible et force les poids forts avec le signe de R_x.



Le mode implicite : Registre à registre (exo)

DR8	R8	R9	R10	R11
DR4	R4	R5	R6	R7
DR0	R0 15h	R1 FBh	R2 96h	R3 08h

MOV DR4,DR0
 MOV R0,R2
 MOVZ WR2,R6
 ADD R1,R7
 MOVS WR10,R5



NB :	05h	00:4000h
TAB1 :	40h	00:4001h
	00h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	00h	00:4009h
	00h	00:400Ah
	00h	00:400Bh
	00h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Le mode implicite : Registre à registre (exo)

DR8	R8	R9	R10 FFh	R11 FBh
DR4	R4 15h	R5 FBh	R6 96h	R7 08h
DR0	R0 96h	R1 03h	R2 00h	R3 96h

MOV DR4,DR0 2 / 2
 MOV R0,R2 2 / 1
 MOVZ WR2,R6 2 / 1
 ADD R1,R7 2 / 1
 MOVS WR10,R5 2 / 1

CY	OV	N	Z
1	0	0	0
A			
FBh			

NB :	05h	00:4000h
TAB1 :	40h	00:4001h
	00h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	00h	00:4009h
	00h	00:400Ah
	00h	00:400Bh
	00h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Rq : calcul de l'overflow

Add R1,R7

```
  0xFB  0' 1 1 1 1 1 0 1 1
+ 0x08  0' 0 0 0 0 1 0 0 0
```

```
  1 0 0 0 0 0 0 1 1
```

retenue: 1 1 1 1 1 0 0 0

en vert : $CY = 1$

en rouge : $OV = XOR(R,R-1) = 0$

Le mode immédiat : Registre et valeur directe

- Mov Rx,#direct8
 - Mov WRx, #direct16
 - Il n'y a pas de mode immédiat pour 32b.
-
- Charge la valeur « direct » dans le registre de destination.
 - La valeur est indiquée en décimal ou en hexadécimal si « h » est ajouté à la fin

Le mode immédiat : Registre et valeur directe (exo)

DR8	R8	R9	R10	R11
DR4	R4	R5	R6	R7
DR0	R0	R1	R2	R3

```
MOV    R7,#10
MOV    WR4,#0FF45h
MOVO   DR0,#TAB2
SUB    WR2,#TAB1
```

CY	OV	N	Z

```
INC    R1,#4
```

CY	OV	N	Z

```
CMP    R7,#8Ah
```

CY	OV	N	Z

NB :	05h	00:4000h
TAB1 :	40h	00:4001h
	00h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	00h	00:4009h
	00h	00:400Ah
	00h	00:400Bh
	00h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Le mode immédiat : Registre et valeur directe (exo)

DR8	R8	R9	R10	R11
DR4	R4 FFh	R5 45h	R6	R7 0Ah
DR0	R0 FFh	R1 FF 03	R2 40 00	R3 09 08

```
MOV    R7,#10
MOV    WR4,#0FF45h
MOVO   DR0,#TAB2
SUB    WR2,#TAB1
```

```
INC    R1,#4
```

```
CMP    R7,#8Ah
```

CY	OV	N	Z
0	0	0	0
CY	OV	N	Z
#0	#0	0	0
CY	OV	N	Z
1	1	1	0

NB :	05h	00:4000h
TAB1 :	40h	00:4001h
	00h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	00h	00:4009h
	00h	00:400Ah
	00h	00:400Bh
	00h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Rq : calcul de l'overflow

Cmp R7,#8Ah

0x0A	0' 0 0 0 0 1 0 1 0
- 0x8A	0' 1 0 0 0 1 0 1 0

	0' 0 0 0 0 1 0 1 0
+	1' 0 1 1 1 0 1 1 0

	1 1 0 0 0 0 0 0 0
retenue:	0 1 1 1 1 1 1 0

en vert : $CY = 1$

en rouge : $OV = XOR(R,R-1) = 1$

Le mode immédiat : Registre et valeur directe (exo)

DR8	R8	R9	R10	R11
DR4	R4 FFh	R5 45h	R6	R7 0Ah
DR0	R0 FFh	R1 FF 03	R2 40 00	R3 09 08

MOV	R7,#10	3	/	2
MOV	WR4,#0FF45h	4	/	2
MOVO	DR0,#TAB2	4	/	4
SUB	WR2,#TAB1	4	/	3
INC	R1,#4	2	/	1
CMP	R7,#8Ah	3	/	2

NB :	05h	00:4000h
TAB1 :	40h	00:4001h
	00h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	00h	00:4009h
	00h	00:400Ah
	00h	00:400Bh
	00h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Le mode direct : Registre à mémoire

Lecture

- Mov Rx, *Adresse*
- Mov WRx, *Adresse*
- Mov DRx, *Adresse*

Ecriture

- Mov *Adresse*, Rx
- Mov *Adresse*, WRx
- Mov *Adresse*, DRx

- Charge le nombre d'octets indiqué par la taille de la destination, à partir de l'adresse mémoire passée en source.
- L'adresse est une valeur numérique ou une étiquette.

Le mode direct : Register à mémoire

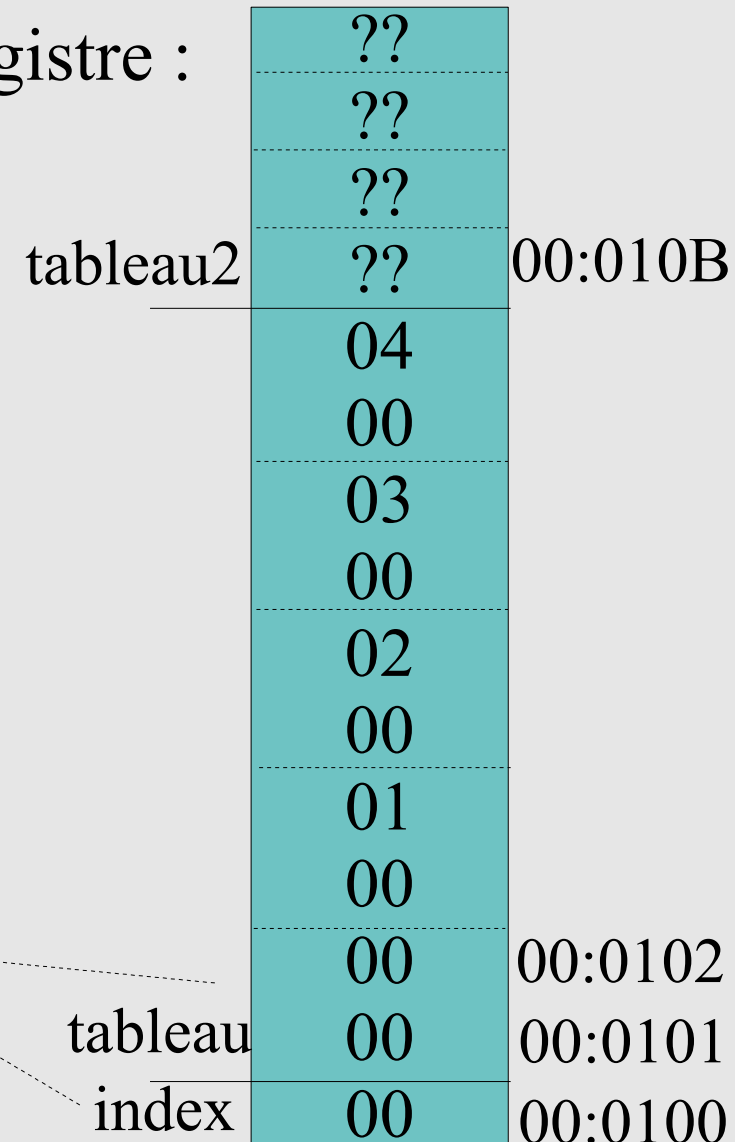
- Charger une variable mémoire dans un registre :
 - MOV R0, index

R0	R1	R2	R3
00	C8	61	1A

- MOV WR2, tableau

R0	R1	R2	R3
03	C8	00	00

WR2



Le mode adressage direct : Registre à mémoire

EXO

Le MCS251-Modes Adressage

Le mode adressage direct : Registre à mémoire (exo)

DR8	R8	R9	R10	R11
DR4	R4	R5	R6	R7
DR0	R0	R1	R2	R3

```
MOV DR0,TAB2
MOV WR2,TAB1
MOV WR2,TAB1+2
MOV WR4,400Fh
MOV R6,4005h
MOV R7,NB
MOV TAB2,DR4
```

CY	OV	N	Z

NB :	05h	00:4000h
TAB1 :	40h	00:4001h
	00h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	00h	00:4009h
	00h	00:400Ah
	00h	00:400Bh
	00h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Le MCS251-Modes Adressage

Le mode adressage direct : Registre à mémoire (exo)

DR8	R8	R9	R10	R11
DR4	R4	R5	R6	R7
DR0	R0 00	R1 00	R2 00	R3 00

```
MOV DR0,TAB2
MOV WR2,TAB1
ADD WR2,TAB1+2
MOV WR4,400Fh
MOV R6,4005h
MOV R7,NB
MOV TAB2,DR4
```

NB :	05h	00:4000h
TAB1 :	40h	00:4001h
	00h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	00h	00:4009h
	00h	00:400Ah
	00h	00:400Bh
	00h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Le mode adressage direct : Registre à mémoire (exo)

DR8	R8	R9	R10	R11
DR4	R4	R5	R6	R7
DR0	R0 00	R1 00	R2 40h	R3 00

```
MOV DR0,TAB2
MOV WR2,TAB1
ADD WR2,TAB1+2
MOV WR4,400Fh
MOV R6,4005h
MOV R7,NB
MOV TAB2,DR4
```

NB :	05h	00:4000h
TAB1 :	40h	00:4001h
	00h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	00h	00:4009h
	00h	00:400Ah
	00h	00:400Bh
	00h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Le mode adressage direct : Registre à mémoire (exo)

DR8	R8	R9	R10	R11
DR4	R4	R5	R6	R7
DR0	R0 00	R1 00	R2 40h	R3 01

```
MOV DR0,TAB2
MOV WR2,TAB1
ADD WR2,TAB1+2
MOV WR4,400Fh
MOV R6,4005h
MOV R7,NB
MOV TAB2,DR4
```

NB :	05h	00:4000h
TAB1 :	40h	00:4001h
	00h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	00h	00:4009h
	00h	00:400Ah
	00h	00:400Bh
	00h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Le MCS251-Modes Adressage

Le mode adressage direct : Registre à mémoire (exo)

DR8	R8	R9	R10	R11
DR4	R4 01	R5 03	R6	R7
DR0	R0 00	R1 00	R2 40h	R3 01

```
MOV DR0,TAB2
MOV WR2,TAB1
ADD WR2,TAB1+2
MOV WR4,400Fh
MOV R6,4005h
MOV R7,NB
MOV TAB2,DR4
```

NB :	05h	00:4000h
TAB1 :	40h	00:4001h
	00h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	00h	00:4009h
	00h	00:400Ah
	00h	00:400Bh
	00h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Le MCS251-Modes Adressage

Le mode adressage direct : Registre à mémoire (exo)

DR8	R8	R9	R10	R11
DR4	R4 01	R5 03	R6 FFh	R7
DR0	R0 00	R1 00	R2 40h	R3 01

```
MOV DR0,TAB2
MOV WR2,TAB1
ADD WR2,TAB1+2
MOV WR4,400Fh
MOV R6,4005h
MOV R7,NB
MOV TAB2,DR4
```

NB :	05h	00:4000h
TAB1 :	40h	00:4001h
	00h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	00h	00:4009h
	00h	00:400Ah
	00h	00:400Bh
	00h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Le MCS251-Modes Adressage

Le mode adressage direct : Registre à mémoire (exo)

DR8	R8	R9	R10	R11
DR4	R4 01	R5 03	R6 FFh	R7 05
DR0	R0 00	R1 00	R2 40h	R3 01

```
MOV DR0,TAB2
MOV WR2,TAB1
ADD WR2,TAB1+2
MOV WR4,400Fh
MOV R6,4005h
MOV R7,NB
MOV TAB2,DR4
```

NB :	05h	00:4000h
TAB1 :	40h	00:4001h
	00h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	00h	00:4009h
	00h	00:400Ah
	00h	00:400Bh
	00h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Le MCS251-Modes Adressage

Le mode adressage direct : Registre à mémoire (exo)

DR8	R8	R9	R10	R11
DR4	R4 01	R5 03	R6 FFh	R7 05
DR0	R0 00	R1 00	R2 40h	R3 01

```
MOV DR0,TAB2
MOV WR2,TAB1
ADD WR2,TAB1+2
MOV WR4,400Fh
MOV R6,4005h
MOV R7,NB
MOV TAB2,DR4
```

NB :	05h	00:4000h
TAB1 :	40h	00:4001h
	00h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	01h	00:4009h
	03h	00:400Ah
	FFh	00:400Bh
	05h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Le mode indirect par registre : Registre à mémoire

- Par rapport au mode direct, l'adresse mémoire n'est pas une donnée statique (étiquette) mais une donnée dynamique : un registre.

Lectures

Mov Rx,@WRx

Mov WRx,@WRx

~~Mov DRx,@WRx~~

Écritures

Mov @WRx, Rx

Mov @WRx, WRx

~~Mov @WRx,DRx~~

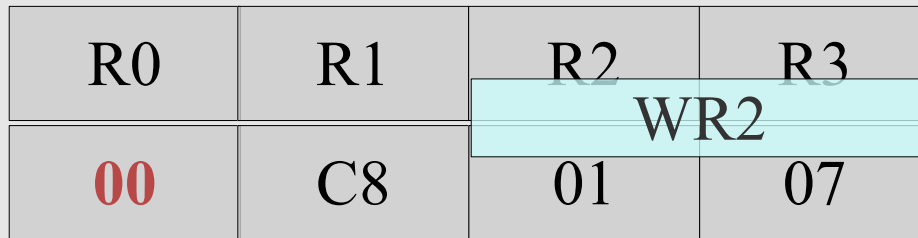
- Dans ce mode, le registre précédé d'un « @ » correspond à un pointeur, comme en C.
- **Il faut distinguer la taille du pointeur ici (WR – 16b) de la taille des données lues/écrites, ici 8 puis 16 puis 32b.**

Le mode indirect par registre : Registre à mémoire

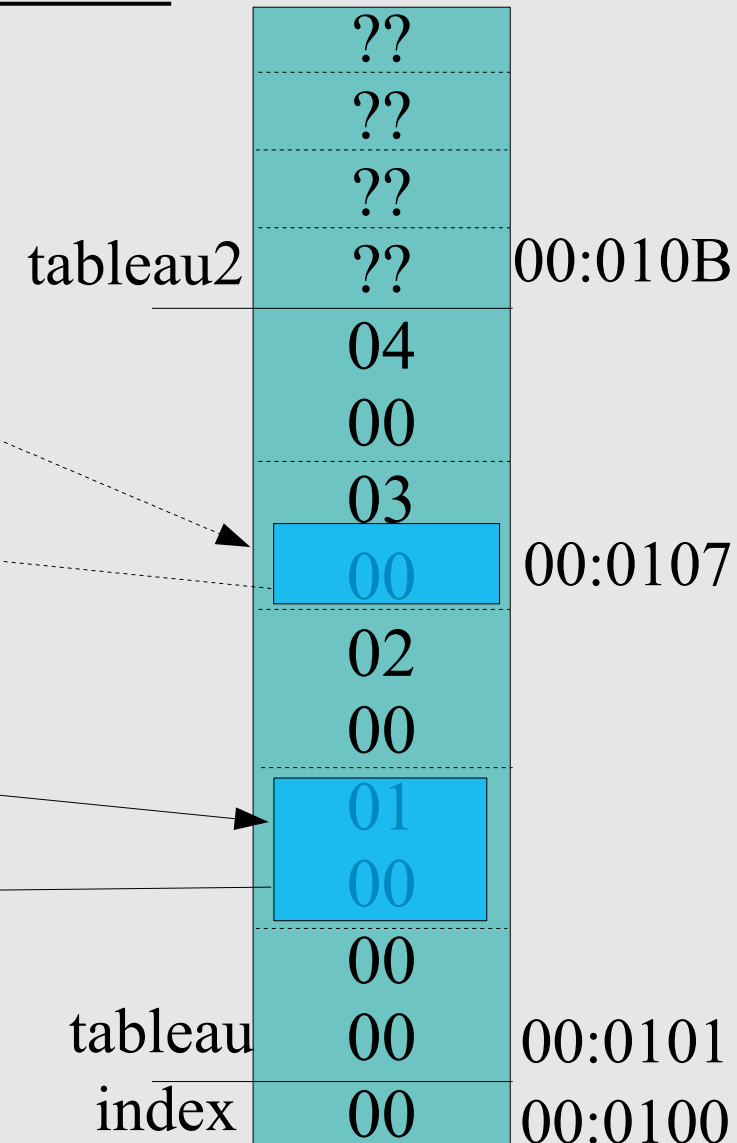
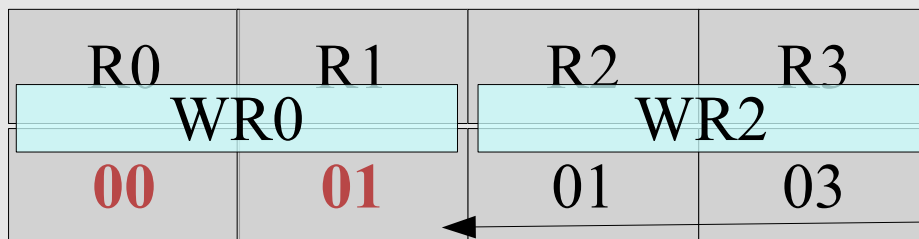
- **Un pointeur sera généralement de 16 bits (taille standard d'une adresse sur le MCS251)**
- Il existe toutefois des pointeurs 32b (puisque la mémoire totale peut faire 24bits) et des pointeurs courts 8b (pour l'accès aux registres SFR de configuration).

Le mode indirect par registre : Registre à mémoire

- MOV R0, @WR2



- MOV WR0, @WR2



Mode indirect par registre (exo)

DR4	R4	R5	03	R6	R7
DR0	R0	R1	00	R2	40h
			00		R3
					01

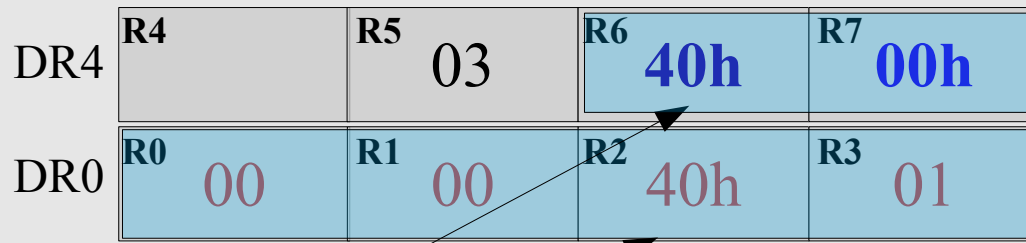
```
MOV WR6,@DR0
MOV R4,@WR6
MOV @WR2,WR4
```

```
CMP R5,@DR0
```

CY	OV	N	Z

NB :	05h	00:4000h
TAB1 :	40h	00:4001h
	00h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	01h	00:4009h
	03h	00:400Ah
	0Fh	00:400Bh
	05h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Mode indirect par registre (exo)



```
MOV WR6,@DR0
MOV R4,@WR6
MOV @WR2,WR4
```

```
CMP R5,@DR0
```

CY	OV	N	Z

NB :	05h	00:4000h
TAB1 :	40h	00:4001h
	00h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	01h	00:4009h
	03h	00:400Ah
	0Fh	00:400Bh
	05h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Mode indirect par registre (exo)

DR4	R4 05h	R5 03	R6 40h	R7 00
DR0	R0 00	R1 00	R2 40h	R3 01

```
MOV WR6,@DR0
MOV R4,@WR6
MOV @WR2,WR4
```

```
CMP R5,@DR0
```

CY	OV	N	Z

NB :	05h	00:4000h
TAB1 :	40h	00:4001h
	00h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	01h	00:4009h
	03h	00:400Ah
	0Fh	00:400Bh
	05h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Mode indirect par registre (exo)

DR4	R4 05h	R5 03	R6 40h	R7 00
DR0	R0 00	R1 00	R2 40h	R3 01

```
MOV WR6,@DR0
MOV R4,@WR6
MOV @WR2,WR4
```

```
CMP R5,@DR0
```

CY	OV	N	Z

NB :	05h	00:4000h
TAB1 :	05h	00:4001h
	03h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	01h	00:4009h
	03h	00:400Ah
	0Fh	00:400Bh
	05h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Mode indirect par registre (exo)

DR4	R4 05h	R5 03	R6 40h	R7 00
DR0	R0 00	R1 00	R2 40h	R3 01

```
MOV WR6,@DR0
MOV R4,@WR6
MOV @WR2,WR4
```

```
CMP R5,@DR0
```

CY	OV	N	Z
1	0	1	0

NB :	05h	00:4000h
TAB1 :	05h	00:4001h
	03h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	01h	00:4009h
	03h	00:400Ah
	0Fh	00:400Bh
	05h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Le mode indexé par registre : Registre à mémoire

- Comme le mode Indirect par registre, mais il est possible en plus d'ajouter un offset (adresse fixe de départ)

LECTURES

- `Mov Rx, @WRx+direct16`
- `Mov WRx, @WRx+direct16`
- ~~`Mov DRx, @WRx+direct16`~~

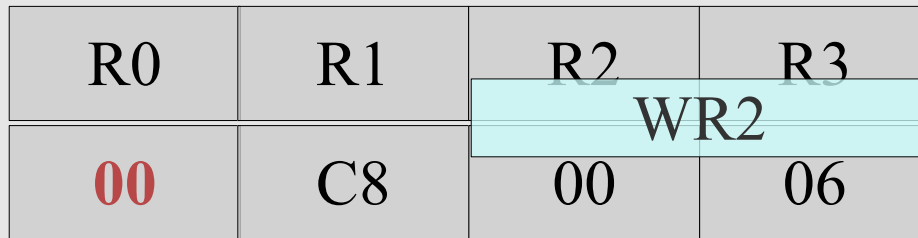
ECRITURES

- `Mov @WRx+direct16, Rx`
- `Mov @WRx+direct16, WRx`
- ~~`Mov @WRx+direct16, DRx`~~

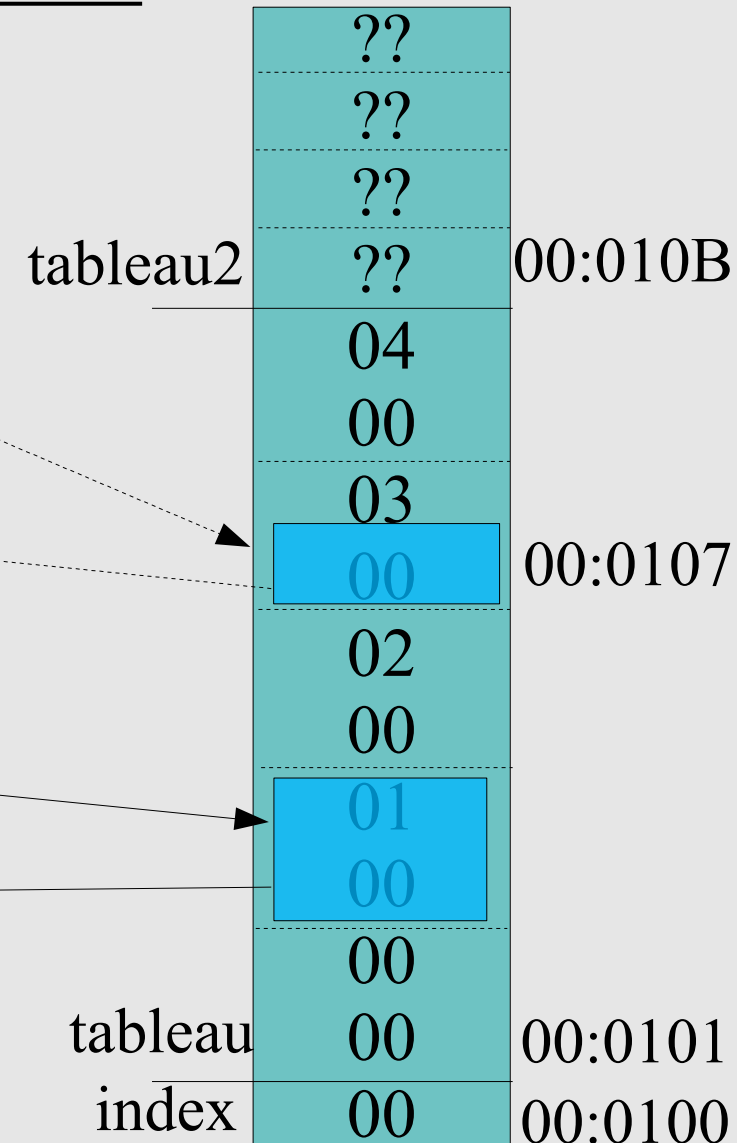
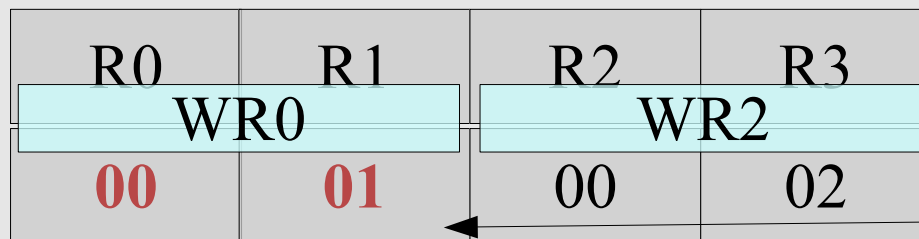
- Mode très utile pour le parcours d'un tableau : `direct16` est alors l'adresse de début du tableau et `WRx` représente l'élément accédé dans le tableau.

Le mode indexé par registre : Register à mémoire

- MOV R0, @WR2+tableau



- MOV WR0, @WR2+tableau



Mode indexé par registre (exo)

DR8	R8	R9	R10	R11
DR4	R4	R5	R6	R7
DR0	R0	R1	R2	R3

```
MOV WR0,#2
MOV WR2,#TAB2
MOV WR10,@WR0+TAB1
MOV WR4,@WR2+4
MOV WR6,@WR2+6
```

NB :	05h	00:4000h
TAB1 :	05h	00:4001h
	03h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	01h	00:4009h
	03h	00:400Ah
	0Fh	00:400Bh
	05h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Mode indexé par registre (exo)

DR8	R8	R9	R10	R11
DR4	R4	R5	R6	R7
DR0	R0 00	R1 02	R2	R3

```
MOV WR0,#2
MOV WR2,#TAB2
MOV WR10,@WR0+TAB1
MOV WR4,@WR2+4
MOV WR6,@WR2+6
```

NB :	05h	00:4000h
TAB1 :	05h	00:4001h
	03h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	01h	00:4009h
	03h	00:400Ah
	0Fh	00:400Bh
	05h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Mode indexé par registre (exo)

DR8	R8	R9	R10	R11
DR4	R4	R5	R6	R7
DR0	R0 00	R1 02	R2 40h	R3 09h

NB :	05h	00:4000h
TAB1 :	05h	00:4001h
	03h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	01h	00:4009h
	03h	00:400Ah
	0Fh	00:400Bh
	05h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

```

MOV WR0,#2
MOV WR2,#TAB2
MOV WR10,@WR0+TAB1
MOV WR4,@WR2+4
MOV WR6,@WR2+6
    
```


Mode indexé par registre (exo)

DR8	R8	R9	R10 00	R11 01
DR4	R4	R5	R6	R7
DR0	R0 00	R1 02	R2 40h	R3 09h

NB :	05h	00:4000h
TAB1 :	05h	00:4001h
	03h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	01h	00:4009h
	03h	00:400Ah
	0Fh	00:400Bh
	05h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

```

MOV  WR0,#2
MOV  WR2,#TAB2
MOV  WR10,@WR0+TAB1
MOV  WR4,@WR2+4
MOV  WR6,@WR2+6
    
```

Mode indexé par registre (exo)

DR8	R8	R9	R10	R11
			00	01
DR4	R4	R5	R6	R7
	00	00		
DR0	R0	R1	R2	R3
	00	02	40h	09h

```

MOV  WR0,#2
MOV  WR2,#TAB2
MOV  WR10,@WR0+TAB1
MOV  WR4,@WR2+4
MOV  WR6,@WR2+6
    
```

NB :	05h	00:4000h
TAB1 :	05h	00:4001h
	03h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	01h	00:4009h
	03h	00:400Ah
	0Fh	00:400Bh
	05h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

Mode indexé par registre (exo)

DR8	R8	R9	R10	R11
			00	01
DR4	R4	R5	R6	R7
	00	00	01h	03h
DR0	R0	R1	R2	R3
	00	02	40h	09h

```

MOV  WR0,#2
MOV  WR2,#TAB2
MOV  WR10,@WR0+TAB1
MOV  WR4,@WR2+4
MOV  WR6,@WR2+6
    
```

NB :	05h	00:4000h
TAB1 :	05h	00:4001h
	03h	00:4002h
	00h	00:4003h
	01h	00:4004h
	FFh	00:4005h
	F8h	00:4006h
	00h	00:4007h
	19h	00:4008h
TAB2 :	01h	00:4009h
	03h	00:400Ah
	0Fh	00:400Bh
	05h	00:400Ch
	00h	00:400Dh
	00h	00:400Eh
	01h	00:400Fh
	03h	00:4010h

La gestion des conditions

- *CMP Dest, Source* compare une destination qui est toujours un Registre avec une source qui peut être un registre, une valeur directe, le contenu d'une variable mémoire ou le contenu d'un pointeur.

- Ex :
CMP R0, #2
CMP WR0, label
CMP R0, @WR2
CMP WR2, WR3

La comparaison est en fait une soustraction dont le résultat n'est pas mémorisé dans la destination.

La comparaison positionne les flags utilisés dans les instructions de sauts conditionnels.

Les instructions de saut

- Saut à une adresse absolue *JMP Label*
- Saut conditionnel à une adresse relative *Jx Label*
 - Contrairement à JMP, l'adresse du saut est calculée à partir de l'adresse courante d'exécution.
 - Les saut ne s'exécute que si la condition précisée est remplie.
 - JZ : Saut si R11 (A) = 0
 - JNZ : Saut si R11 (A) != 0
 - JC : Saut si Flag Carry = 1
 - JNC : Saut si Flag Carry = 0
- Les conditions sont vérifiées ou non à partir de l'état des flags.
- Les flags sont en général positionnés par l'instruction de comparaison **CMP**

Les instructions de saut

- JE : Saut lorsque Source et Destination sont égales
- JNE : Saut lorsque Source et Destination sont différentes
- JGE : Saut lorsque Destination \geq Source
- JSGE : Saut lorsque Destination \geq Source (signé)
- JG : Saut lorsque Destination $>$ Source
- JSG : Saut lorsque Destination $>$ Source (signé)
- JLE : Saut lorsque Destination \leq Source
- JSLE : Saut lorsque Destination \leq Source (signé)
- JL : Saut lorsque Destination $<$ Source
- JSL : Saut lorsque Destination $<$ Source (signé)

Passage du C à l'assembleur :

Si Condition

Alors

Action1

Sinon

Action2

Finsi

==>

If1: CMP Condition

J« cond. inverse » Else1

Then1:

Action1

JMP EndIf1

Else1:

Action2

EndIf1:

Passage du C à l'assembleur :

Tantque Condition Faire

Action

Fait

=>

While: CMP Condition
J« Cond. Inverse » **EWhile**

Action

JMP **While**

EWhile: