

ARCHITECTURES DES ORDINATEURS II

Introduction et Rappels

EQUIPE PEDAGOGIQUE :

Chargé de cours : Mr M.S AYACHE

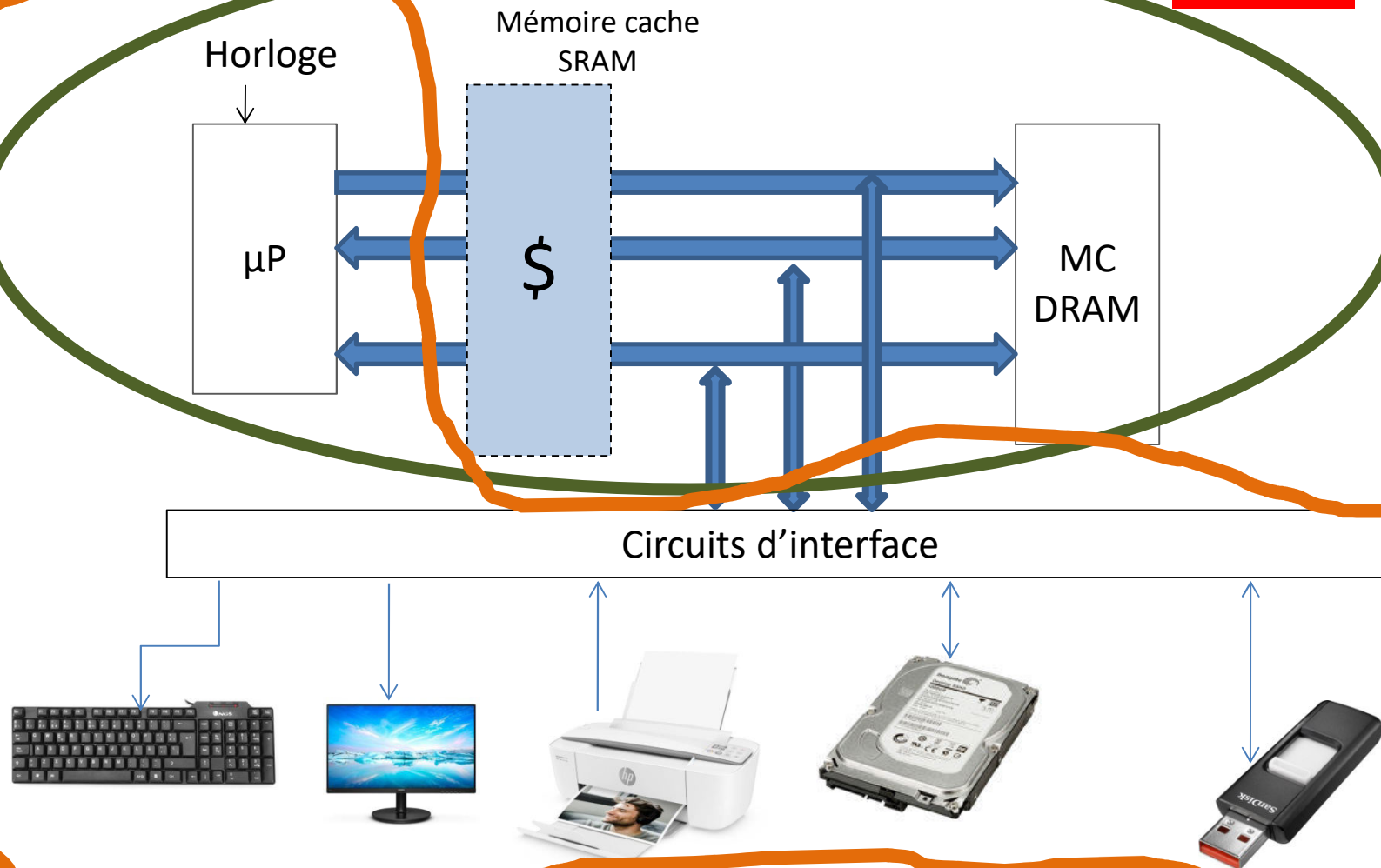
Chargée de TP : Mme S. BOUCHENE

Chargée de TD : Mr M.S AYACHE & Melle I. SETITRA

RAPPEL : Schéma synoptique d'un micro-ordinateur

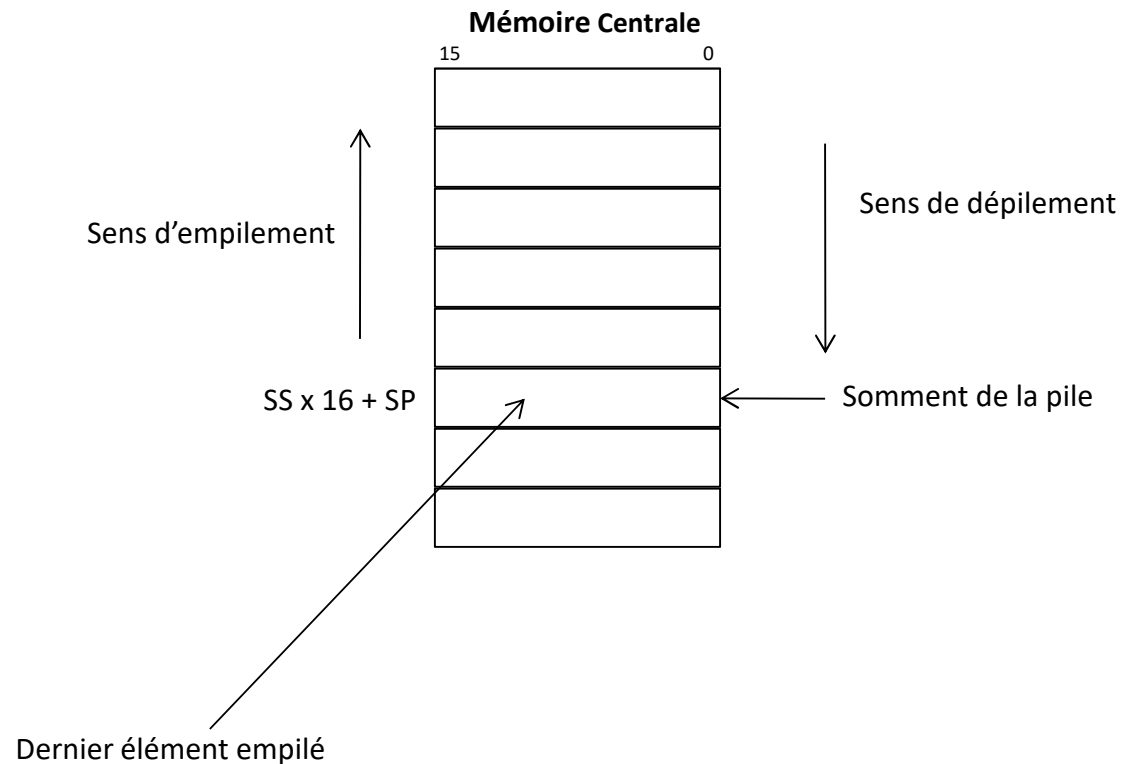
ARCHI I

ARCHI II



LA PILE (STACK):

La pile est une partie de la mémoire RAM dont l'organisation est LIFO. La pile est caractérisée par un sommet de pile et ses adresses évoluent contrairement aux adresses de la mémoire traditionnelle. Pour le microprocesseur 8086, tous les accès à la pile se font sur 16 bits et l'adresse physique du sommet de la pile est $SS \times 16 + SP$.



MODES D'UTILISATION DE LA PILE:

a) La pile en tant que pile


Dans ce mode d'utilisation, les accès à la pile sont tous séquentiels. Deux types d'utilisation sont possibles: explicite et implicite.


a.1 Utilisation explicite:

sert à empiler ou dépiler explicitement des données de la pile. Les instructions qui permettent une telle utilisation sont:

PUSH SRC :	empiler	POP DST :	dépiler
PUSHF :	empiler flags	POPF :	dépiler flags

ACTIONS:

PUSH SRC  $(SP) \leftarrow (SP) - 2$
 $((SP)+1: (SP)) \leftarrow (SRC)$

PUSHF  $(SP) \leftarrow (SP) - 2$
 $((SP)+1: (SP)) \leftarrow (PSW)$

POP DST  $(DST) \leftarrow ((SP)+1: (SP))$
 $(SP) \leftarrow (SP) + 2$

POPF  $(PSW) \leftarrow ((SP)+1: (SP))$
 $(SP) \leftarrow (SP) + 2$

EXEMPLE:

```
mov ax, 0400h
mov ss , ax
mov sp , 0100h
mov bx , 1234h
push bx
push ax
.
.
pop ax
pop bx
```

adresse	15	0
0400:00FC	0400	
0400: 00FE	1234	
0400:0100		

← Sommet de la pile

a) après push

adresse	15	0
0400:00FC	0400	
0400: 00FE	1234	
0400:0100		

ax = 0400H

bx= 1234H

← Sommet de la pile

b) après pop

a.2 Utilisation implicite:

Ce type d'utilisation est rencontré dans les appels et retours des procédures NEAR et FAR et dans les appels et retours des interruptions logicielles et matérielles.

Appels et Retour des procédures *(on ne traite que le cas des appels directs)*

Type d'appel

Intrasegment

Syntaxe

CALL NEAR PTR nom_proc

Action automatique

$(SP) \leftarrow (SP) - 2$
 $((SP)+1:SP) \leftarrow IP_{\text{retour}}$

Intersegment

CALL FAR PTR nom_proc

$SP \leftarrow (SP) - 2$
 $((SP)+1:SP) \leftarrow CS_{\text{retour}}$
 $(SP) \leftarrow (SP) - 2$
 $((SP)+1:SP) \leftarrow IP_{\text{retour}}$

Type de Retour

intrasegment

Syntaxe

RET

Action automatique

$(IP) \leftarrow ((SP)+1:SP)$
 $(SP) \leftarrow (SP) + 2$

RET E.C

même chose que ci-dessus
 $(SP) \leftarrow (SP) + E.C$

intersegment

RET

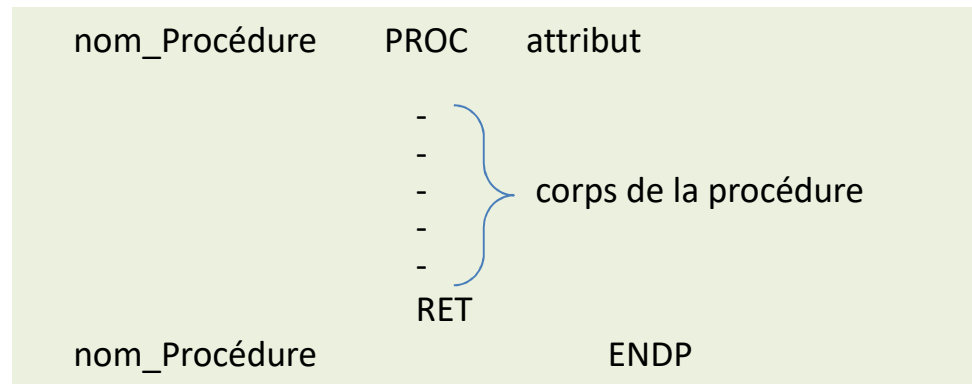
$(IP) \leftarrow ((SP)+1:SP)$
 $(SP) \leftarrow (SP) + 2$
 $(CS) \leftarrow ((SP)+1:SP)$
 $(SP) \leftarrow (SP) + 2$

RET E.C

même chose que ci-dessus
 $(SP) \leftarrow (SP) + E.C$

*** E.C : Expression Constante

SYNTAXE DE DECLARATION DES PROCEDURES



Attribut : NEAR ou FAR (par défaut Near)

REMARQUES:

- ☐ Si l'attribut est NEAR , **la procédure est intrasegment** ceci signifie que l'appel et la procédure doivent être dans le même code segment et donc il n'y a que IP qui change.
- ☐ Si l'attribut est FAR , **la procédure est intersegment** ceci signifie que l'appel et la procédure sont dans des code segments différents et donc IP et CS changent.
- ☐ L'action de l'instruction RET dépend de l'attribut de la procédure.
- ☐ Même si n-1 appels à une procédure sont intrasegment et un appel à cette même procédure est intersegment, l'attribut de cette procédure doit être déclaré FAR.

EXEMPLE:

Programme Principal

```
.  
.   
mov ax, 0400h  
mov ss, ax  
mov sp, 0100h  
.   
.   
3500:0020 CALL FAR PTR usthb  
.   
.
```

Procédure intersegment

```
Usthb PROC FAR  
8000:0600.  
.   
.   
RET  
Usthb ENDP
```

adresse	15	0
0400:00FC	0025	
0400: 00FE	3500	
0400:0100		

← Sommet de la pile

$IP_{proc} \leftarrow 0600H$

$CS_{proc} \leftarrow 8000H$

a) effet du CALL

adresse	15	0
0400:00FC	0025	
0400: 00FE	3500	
0400:0100		

$IP_{retour} = 0025H$

$CS_{retour} = 3500H$

← Sommet de la pile

b) effet du RET

b) La pile en tant que mémoire

Ce type d'utilisation permet de voir la pile comme étant une partie de la mémoire DRAM, et en conséquence, de pouvoir effectuer des accès directs dans cette structure de données qui est la pile.

Pour ce faire, on utilise le registre BP . Ce dernier permet d'accéder à n'importe quel espace dans le stack segment y compris l'espace de la pile. Ce type d'utilisation trouve son intérêt dans les traitements récurifs ainsi que dans le passage des paramètres aux procédures.

EXEMPLE

Programme principal

```
mov ax , 0800H
mov ss , ax
mov sp , 0200H
Mov si, 1234H
Mov di , 6789H
Push si
Push di
Call near ptr archi
.
```

Procédure intrasegment

```
archi PROC NEAR
    mov bp , sp
    mov bx , [bp+2]
    mov cx , [bp+4]
    .
    .
    .
    ret
archi ENDP
```

adresse	15	0
0800:01FA	IP _{retour}	
0800:01FC	6789	
0800: 01FE	1234	
0800:0200		

← Sommet de la pile

BP ← 01FAH
bx ← 6789H
cx ← 1234H