



CHAPITRE N°9 : Pile

PLAN :

I) Définition	p1
II) Empilement d'une donnée	
1) Présentation de la pile	p1
2) Présentation du fonctionnement du push	p1
3) Mécanisme	p2
4) Programme de push	p2
III) Dépilement	
1) Présentation	p3
2) Mécanisme	p3
3) Programmation	p3
IV) Remarques	p4

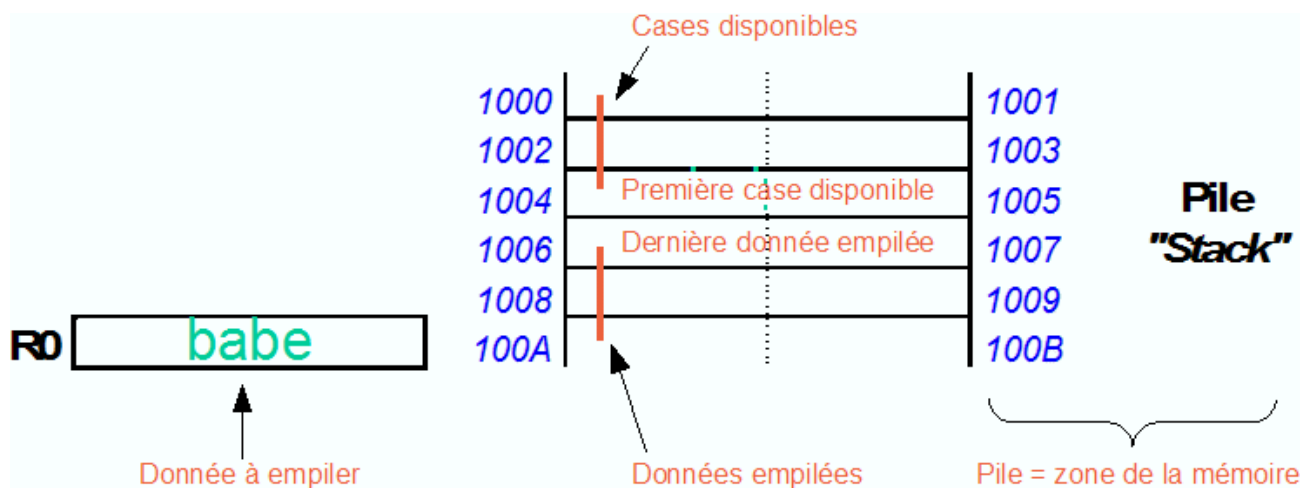
I) Définition

PILE ("STACK") = zone de la mémoire où stocker temporairement des informations par empilage et dépilage.

LIFO = "Last In - First Out" : dernier entré - premier sorti

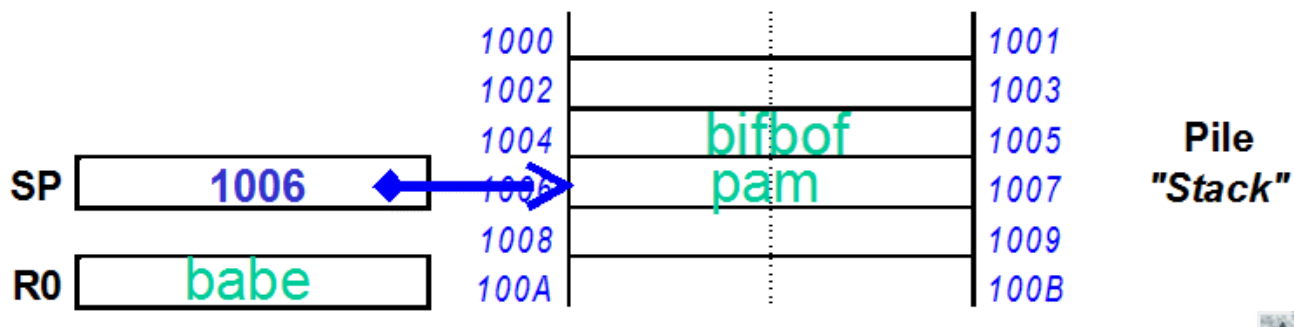
II) Empilement d'une donnée

1) Présentation de la pile

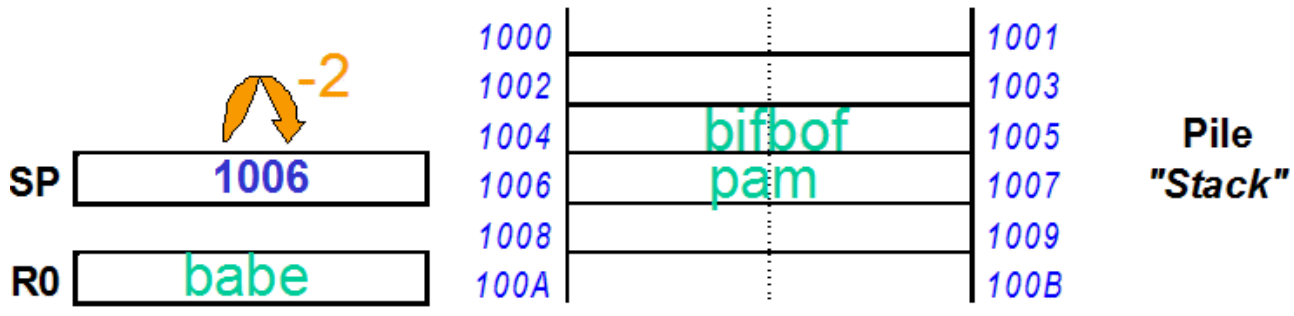


2) Présentation du fonctionnement du push

SP est le pointeur de pile (= Stack Pointer). Il pointe donc sur le sommet de la pile (= Top of Stack), donc sur la dernière donnée empilée.

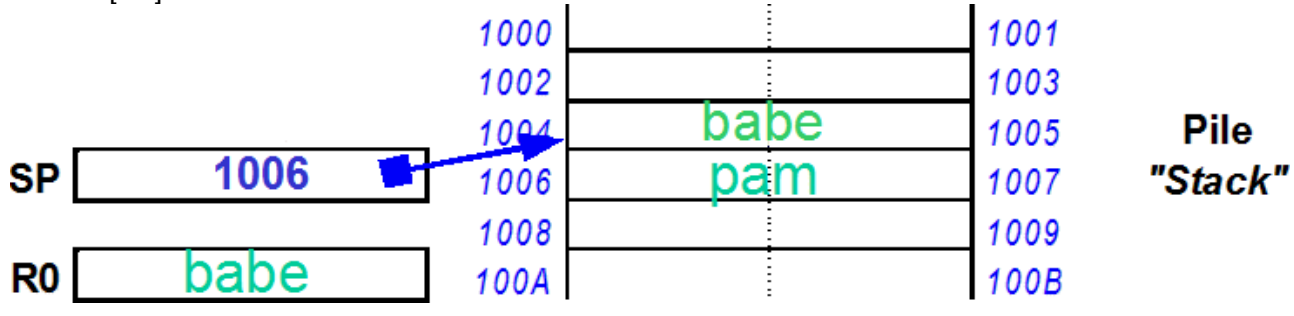


3) Mécanisme



SP pointe donc provisoirement sur la première case disponible (1004 : bifbof).

R0 -> M[SP] :



SP pointe à nouveau sur la dernière donnée empilée.

4) Programme de push

Une instruction spéciale n'est pas nécessaire.

```
ADQ -2, SP ; // décrémente SP;
              // SP pointe ensuite sur la 1ère case disponible
STW R0, (SP) ; // stocke le contenu de R0
                // dans la case pointée par SP
                // donc la 1ère disponible
```

Utilisation du mode "basé pré-décrémenté" :

```
STW R0, -(SP) ; // décrémente SP, puis
                 // stocke le contenu de R0
                 // ds la case pointée par SP
```

Remarque :

Pour dépiler des bytes, on utilise STB R, -(SP)
 SP sera décrémenté que de 1 dans ce cas.

Dans beaucoup de machines, il existe une instruction spécifique et optimisée pour SP : PuSh Word.

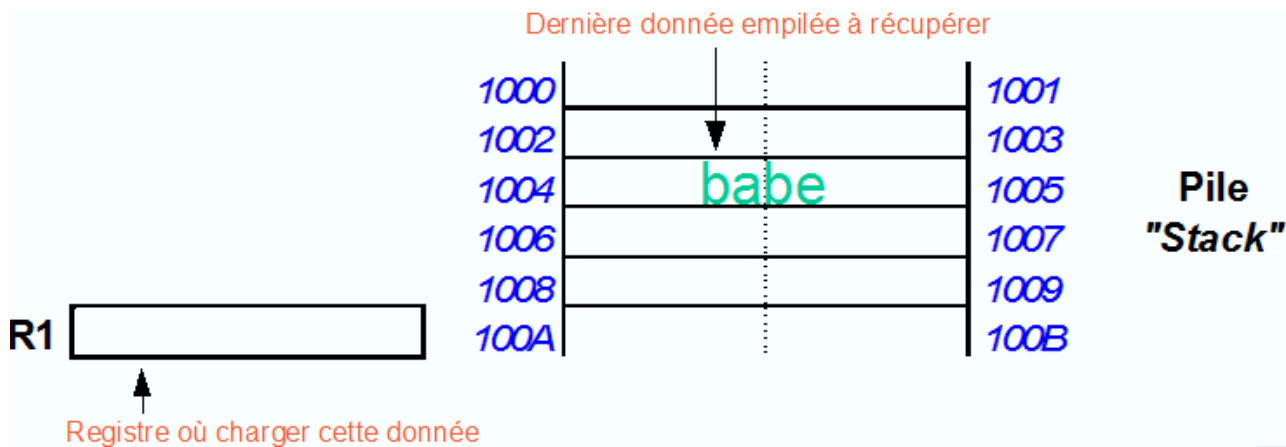
```
PSW R0 ; // décrémente SP, puis
          // stocke le contenu de R0
```

// ds la case pointée par SP



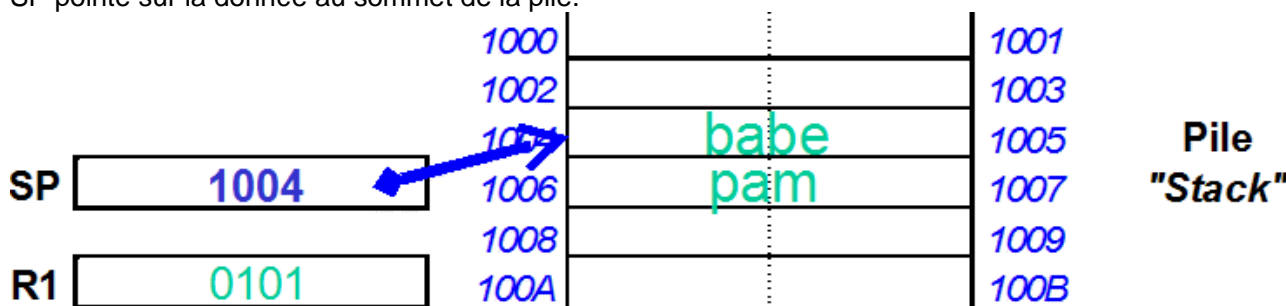
III) Dépilement

1) Présentation

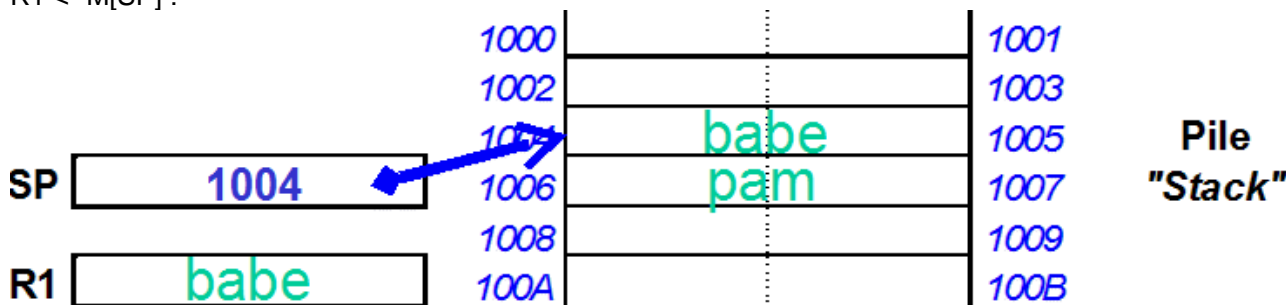


2) Mécanisme

SP pointe sur la donnée au sommet de la pile.



R1 <- M[SP] :



SP <- SP + 2

SP pointe donc sur la dernière donnée empilée avant celle qui vient d'être récupérée (sur pam), donc SP pointe sur le sommet de pile, puisque la case au-dessus de pam est à nouveau disponible.



3) Programmation

Une instruction spéciale n'est pas nécessaire.

```
LDW R0, (SP) ; // charge R0 avec le sommet de pile
ADQ 2, SP ; // incrémente SP
```

Utilisation du mode "basé pré-décrémenté" :

```
LDW R0, (SP)+ ; // charge R0 avec le sommet de pile puis
                // incrémente SP
```



Remarque :

Pour dépiler des bytes, on utilise `LDB R, (SP) +`
SP sera incrémenté que de 1 dans ce cas.

Dans beaucoup de machines, il existe une instruction spécifique et optimisée pour SP : PoP Word ou Pull Word.

```
PPW R0 ; // charge R0 avec le sommet de pile puis
        // incrémente SP
```



IV) Remarques

Ⓞ Initialisation :

La base de la pile doit être spécifiée avant usage.

```
LDW SP, #2048 // initialise SP
```

Ⓞ Débordement :

Lorsque l'on empile trop d'informations, la pile déborde ("stack overflow"), et écrase les données juste au dessus.

Ⓞ La pile est utilisée pour stocker des informations en LIFO :

- informations lors de l'appel et du retour des sous-programmes
- informations locales à un bloc
- données intermédiaires lors du calcul.

Retour



CHARDON Marion
[Webmestre](#)