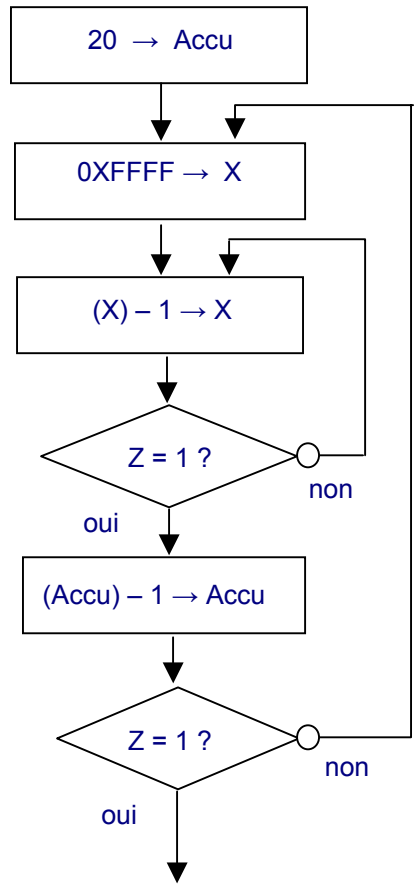
NOTION DE SOUS-PROGRAMME

1/ Qu'est-ce qu'un sous-programme

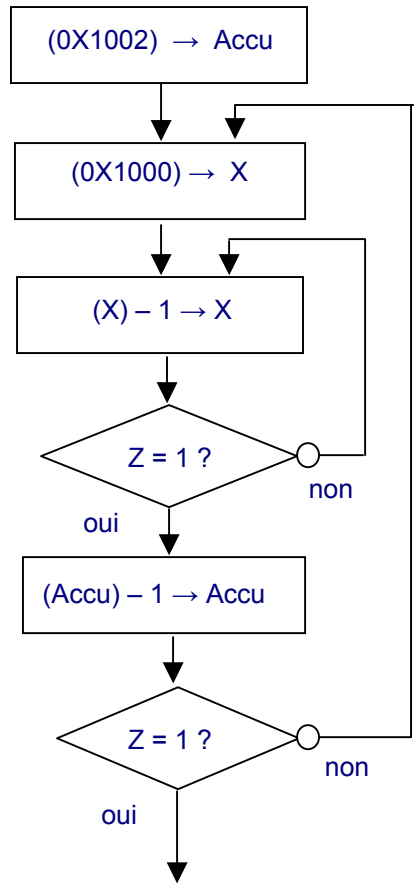
Dans un système piloté par un microprocesseur certaines portions de programme sont utilisés plusieurs fois à quelques valeurs près. Par exemple la temporisation que nous avons étudié dans la leçon précédente pourrait être utilisée à divers moment du programme général.
Il serait intéressant de la rendre plus générale, notamment permettre que la durée puisse être adaptée facilement. On la dira paramètrable.

Reprenons notre programme :




```
Debut :  LDAA   # 0X14      mettre 20 dans l'accu A
Tempo    LDX    # 0XFFFF    mettre 65535 dans l'index
Dec      DEX                décrémentation de l'index
         BNE    Dec         Si l'index ≠ 0 décrémenter à nouveau
         DECA               Si l'index est vide décrémenter l'accu
         BNE    Tempo       Si l'acccu ≠ 0 recharger l'index
         Suite
```

Nous voyons que ce qui fixe la durée de la temporisation sont les nombres qui sont chargés dans l'accumulateur , ici 20 en décimal , 14 en Hexadécimal et 65535 (0XFFFF) dans l'index. Conserver ces valeurs dans le programme le rend rigide, il ne peut que durer 10 secondes.
Si nous chargeons dans ces registres au lieu de nombres, le contenu d'adresse mémoire, nous pourrions avant d'entrer dans ce programme charger des valeurs dans ces cases mémoire le programme deviendrait plus souple.
Supposons que nous ayons une mémoire **RAM logée de l'adresse 0 à 0X1FFF**. Convenons que l'index (X) registre 16 bits sera chargé par le contenu des adresses 0X1000 et 1001, et l'Accu par le contenu de 0X1002. Pour utiliser notre temporisation il nous faudra préalablement charger ces trois cases mémoire. La mémoire de programme qui est une mémoire morte, **PROM, est placée de l'adresse 0XE000 à l'adresse 0XFFFF**. Le programme devient :




```
Debut :  LDAA    0X1002     mettre (0X1002) dans l'accu A
Tempo    LDX     # 0X1000   mettre(0X1000) dans l'index
Dec      DEX                décrémentation de l'index
         BNE     Dec        Si l'index ≠ 0 décrémenter à nouveau
         DECA                Si l'index est vide décrémenter l'accu
         BNE     Tempo      Si l'acccu ≠ 0 recharger l'index
         Suite
```

2/ Problèmes posés par l'appel à notre programme

Représentons notre mémoire de programme qui est une mémoire PROM de la façon ci-dessous :

Adresse		
	0XE000	1er programme
	0XE080	demande de temporisation
	0XE1B0	
	0XE200	2ème programme
	0XE424	demande de temporisation
	0XE480	
	0XE500	Temporisation

Nous voyons que le premier programme appel la temporisation depuis l'adresse 0XE080 et le second depuis l'adresse 0XE424 mais le problème se pose pour le retour . En effet si la demande de

temporisation est un saut à l'adresse 0XE500 qui s'écrira en assembleur JMP 0XE500, instruction en 3 octets, l'adresse de retour dans le premier programme 0XE083 et dans le second 0XE427 , il n'est donc pas possible à la fin du programme de temporisation de mettre l'adresse de retour en clair. Nous allons devoir avant d'entrer dans le programme de temporisation charger non seulement les valeurs pour la tempo mais aussi l'adresse de retour. Par exemple pour le premier programme on stockera 0XE083 en 0X1003 et 0X1004 et pour le second 0XE427 à ces mêmes adresses.
Les dernières instructions de notre temporisation seront alors

LDX	0X1003	Chargement dans l'index du contenu de 1003 et 1004
JMP	0,X	Saut à l'adresse 0 par rapport à l'index

3/ La pile et le pointeur de pile

Les microprocesseurs sont munis d'un pointeur de pile, registre de 16bits qui pointe le sommet d'une zone de mémoire RAM l'adresse de cette zone est définie par le concepteur du système et sa profondeur par le programmeur. La pile sera employée notamment pour les appels à sous-programme et pour la gestion des interruptions .

Dés les premières instructions le pointeur de pile doit être chargé avec l'adresse de la pile (l'adresse la plus haute) pour certains microprocesseur le pointeur de pile est chargé avec le contenu d'une adresse en PROM à la mise sous tension ou lors d'un Reset . Dans le cas du 6800 seul le compteur de programme est chargé à la mise sous tension avec le contenu des adresses FFFE et FFFF ce qui oblige à avoir de la mémoire morte à ces adresses.

Pour accéder à notre programme de Temporisation dont la première instruction est précédée de l'étiquette Debut, il nous suffira (après avoir chargé les paramètres de la temporisation) de placer l'instruction Bsr (Branch to Subroutine) Tempo ou Jsr (Jump to Subroutine) Tempo suivant l'amplitude du saut. Avant de placer dans le compteur de programme l'adresse de « Debut » le microprocesseur place dans la pile l'adresse de retour (l'adresse de l'instruction qui suit l'appel à sous programme). Il range les deux octets d'adresse en se décrémentant si bien qu'après « empilement » de l'adresse le pointeur contient la valeur initiale –2. A la fin de notre sous-programme il suffira de placer l'instruction RTS (Return from Subroutine) , le micro processeur aidé par le pointeur de pile récupère l'adresse de retour qu'il place dans son compteur de programme, et le pointeur de pile s'incrémente et retrouve sa valeur initiale