

STRUCTURE DE BASE D'UN MICROPROCESSEUR

1/ Automatisation des opérations en chaîne

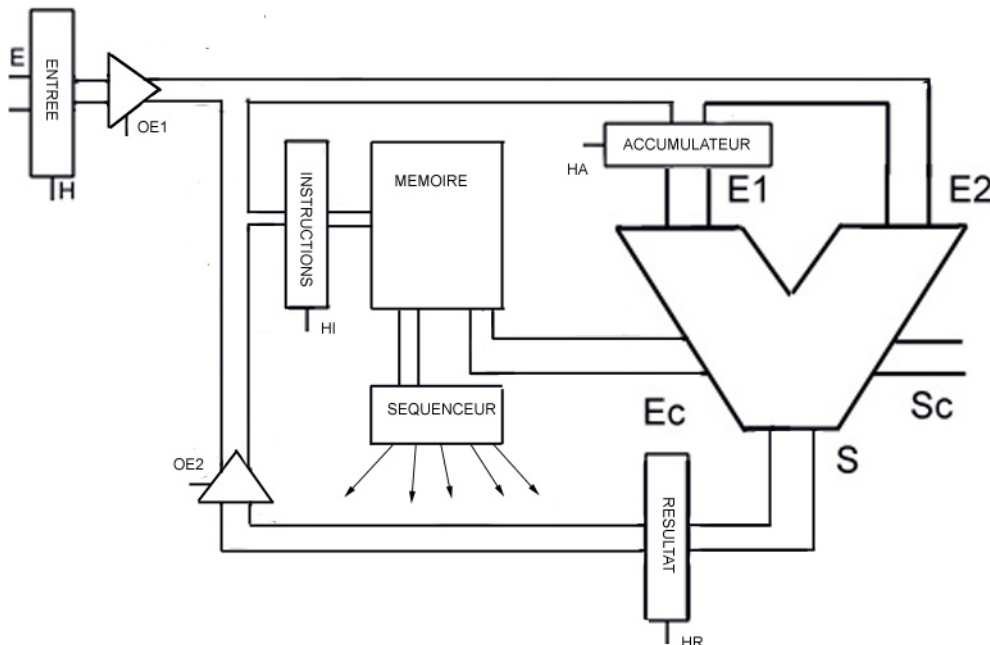
Nous venons de faire un pas vers les microprocesseurs, il faut maintenant automatiser notre précédent schéma. Deux fonctions sont à distinguer :

1/ La fonction du programmeur

Le programmeur qui est chargé de décomposer le travail à faire exécuter par le microprocesseur en une succession de tâches élémentaires que le microprocesseur sait exécuter.

2/ La fonction du concepteur (ou du circuit)

Le concepteur du microprocesseur va donner du savoir faire à son circuit en automatisant l'exécution de chaque tâche. L'ensemble des tâches que le microprocesseur sait exécuter s'appelle alors le **jeu d'instructions**. Un code est attribué à chaque instruction (**code opération** ou **code-op**), ce code est dirigé vers l'entrée d'adresse d'une mémoire qui va alors fournir à l'UAL le code de configuration et aux séquenceurs les nombres qui vont leur permettre de générer la séquence des micro-instructions



Dans le schéma les séquences sont devenues automatiques, le séquenceur va délivrer sa séquence définie par l'instruction, mais dans cette séquence à un certain moment il faut que le nombre sur lequel doit porter l'opération (la **donnée** ou **opérande**) soit présenté sur E ou les instructions ne sont pas toutes de durée identiques, il va donc falloir également automatiser l'arrivée des instructions et des données.

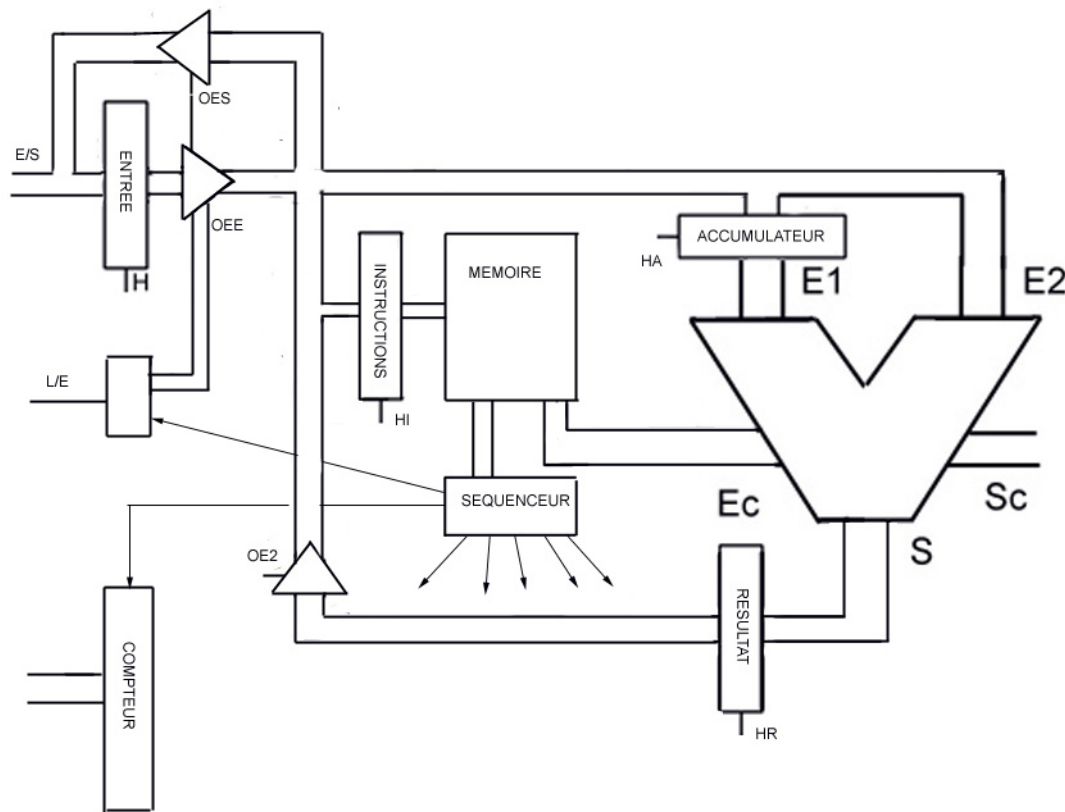
Le programmeur va placer dans une mémoire la succession des instructions (code-op) et des données (opérandes), c'est la mémoire de programme. Le microprocesseur va être muni d'un compteur qui fournira les adresses à la mémoire, le séquenceur sera chargé d'incrémenter ce compteur au moment opportun. Les sorties de ce compteur constituent le **bus d'adresses** du microprocesseur. Les lignes E vont servir à entrer les informations et aussi sortir les résultats, c'est le **bus de données**, le bus externe qui les alimente sera donc bidirectionnel

Le bus de données est bidirectionnel :

Une sortie du microprocesseur informera les circuits externes du sens de transfert sur le bus de données, on dira que le bus est :

en écriture c'est à dire que le sens de transfert va du micro vers l'extérieur

en lecture, le sens de transfert va de l'extérieur vers le microprocesseur.



Un microprocesseur est destiné à dialoguer avec d'autres éléments extérieurs que la mémoire de programme. Ce sont les circuits dits périphériques, on dira un périphérique. Supposons que le microprocesseur soit destiné à réguler la température d'un four, le programme de régulation sera logé dans une mémoire programmable, mais il faudra que le microprocesseur soit capable de lire la température fournie par un capteur à sortie numérique et qu'il soit en mesure de fournir au système de chauffage une commande numérique définissant la puissance de chauffage ou sa modification.

2/ Notion d'adressage

Tout élément pouvant fournir ou recevoir un nombre binaire au microprocesseur devra être connecté au bus de données

Un fournisseur d'information sera dit « **accessible en lecture** », c'est à dire que le nombre qu'il est susceptible de fournir devra être contenu dans un registre à sorties 3états. En général cette sortie est en haute impédance mais lorsqu'on applique un état actif sur sa ligne OE (Output Enable – Validation de la sortie) il émet son nombre binaire sur le bus de données. L'état actif sur OE devra être généré lorsque le microprocesseur fournira l'adresse de celui-ci sur son bus d'adresse et qu'il placera son bus de données en entrée, en lecture

Un accepteur d'information sera dit « **accessible en écriture** », c'est à dire que le nombre qu'il est susceptible de recevoir devra être stocké dans un registre à entrées parallèles qui sont connectées au bus de données. Le nombre binaire qui lui est destiné sera présenté sur le bus de données. Une impulsion sur son entrée d'horloge fera pénétrer le nombre dans le registre et donc dans le circuit. L'impulsion d'horloge devra être généré lorsque le microprocesseur fournira l'adresse de celui-ci sur son bus d'adresse et qu'il placera son bus de données en sortie, en écriture

3/ Compteur d'adresses et bus d'adresses

Avec 8bits on ne peut générer que 255 nombres différents donc, même les microprocesseurs les plus rudimentaires ne peuvent se contenter de 255 adresses. Avec 16 bits on peut générer 65536 nombres (64K). On dira qu'un microprocesseur ayant un bus d'adresse de 16 bits à une **capacité d'adressage** de 64K

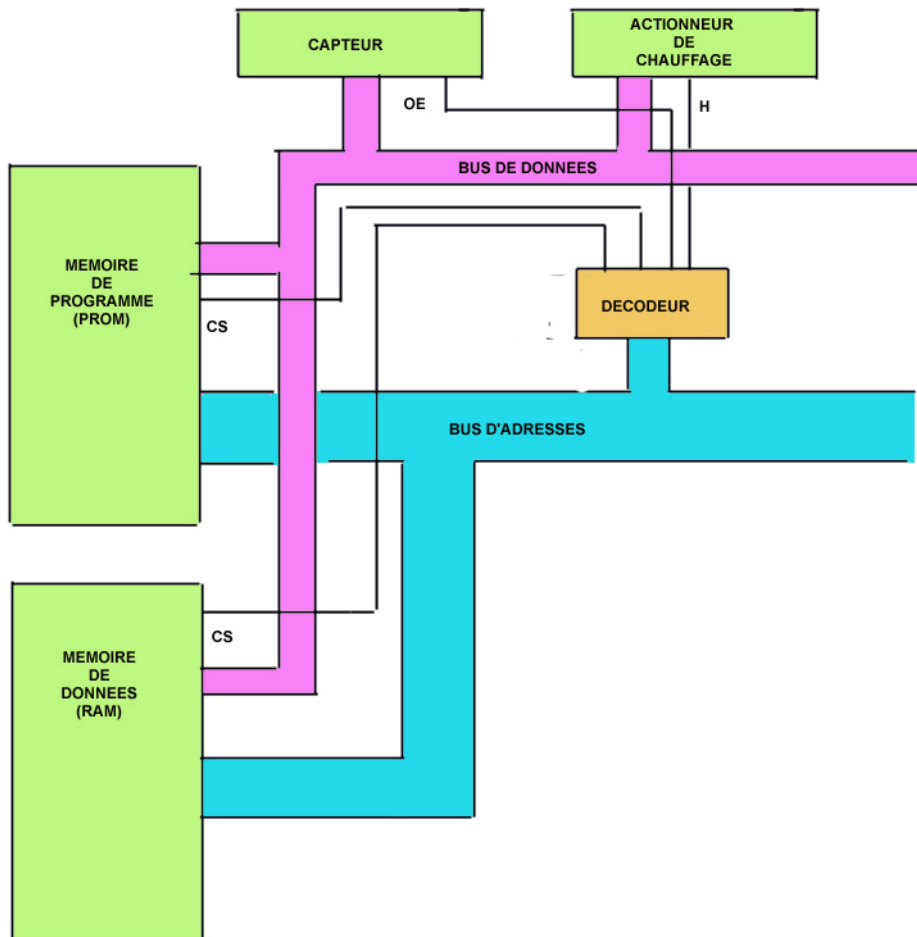
(le 6800 avait un bus d'adresses de 16 bits, il avait une capacité d'adressage de 64K

le 68000 a un bus de 24 bits sa capacité d'adressage est 2^{24} soit 16M

le 68020 a un bus de 32 bits sa capacité d'adressage est 2^{32} soit 4G)

Le microprocesseur ne va pas placer sur son bus d'adresses des adresses successives en effet prenons l'exemple du système de régulation, le microprocesseur va lire la température en plaçant sur son bus l'adresse du capteur, puis il va dans son programme en mettant l'adresse de la mémoire de programme qui va comparer cette valeur à une valeur de consigne en mettant l'adresse de la valeur de consigne puis il va passer dans le programme chauffer plus ou dans le programme chauffer moins et envoyer la commande au système de chauffage etc..

L'architecture du système peut être représenté par le schéma ci-dessous :



Dans l'exemple précédent le bus d'adresses va voir des adresses très diverses, il faut donc que la génération d'adresse soit beaucoup plus sophistiquée que nous l'avons suggéré.

3-1 le compteur

Lorsque des sauts dans le programme sont nécessaires, il va falloir soit charger une nouvelle valeur dans le compteur soit ajouter ou soustraire un nombre au contenu du compteur. Charger une nouvelle valeur suppose un compteur prépositionnable mais comme le bus interne est de 8 bits le chargement devra se faire en deux temps. D'autre part l'addition ou la soustraction d'un nombre au contenu du

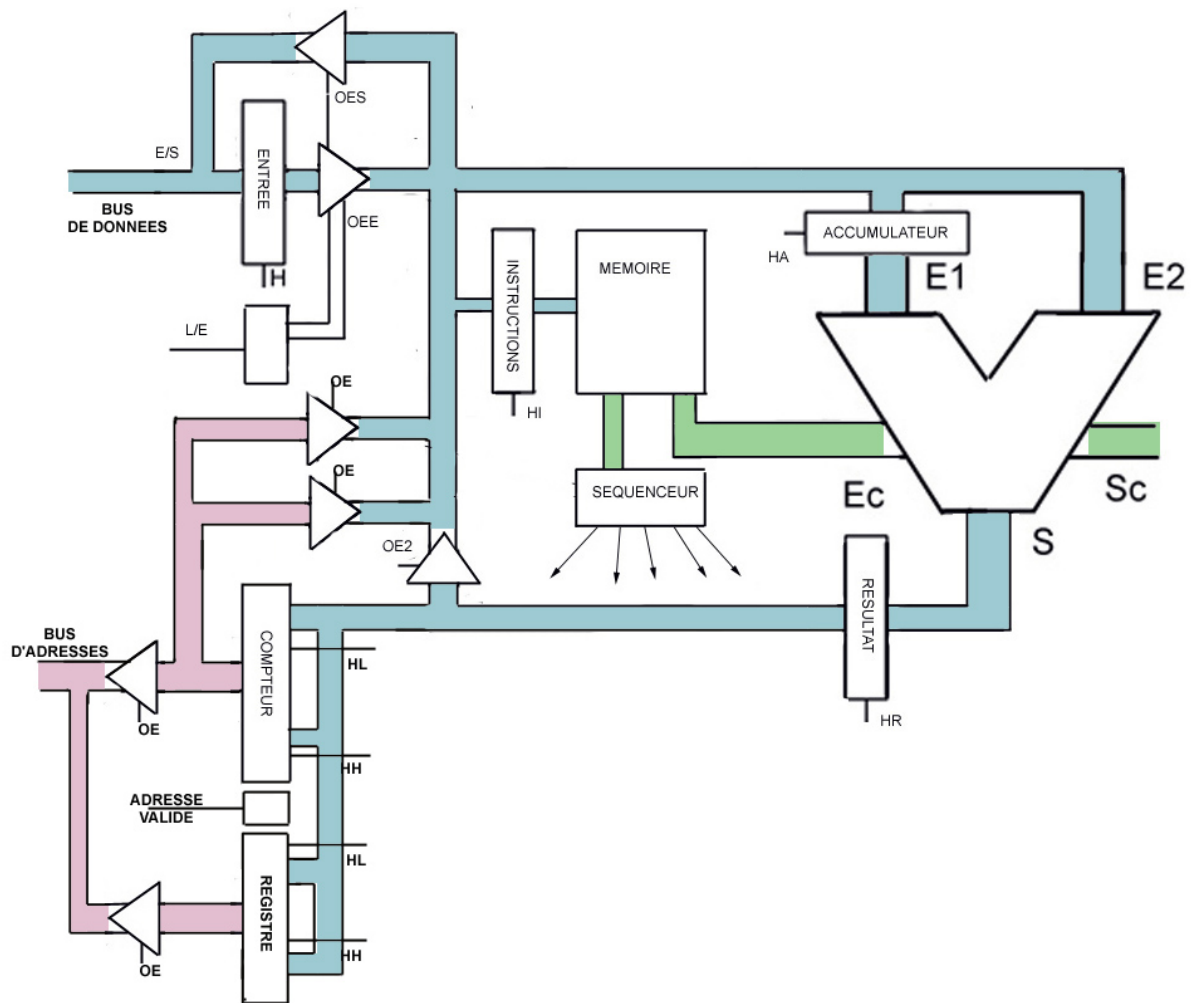
compteur ne peut se faire qu'à l'aide de l'UAL, les valeurs modificatrices sont fournies par le programme mais il faudra que l'UAL puisse lire le contenu du compteur

3-2 un registre

Lorsque l'adresse d'un capteur doit être placée sur le bus d'adresse, il est nécessaire de conserver dans le compteur l'adresse de la prochaine instruction à exécuter il faut donc que cette adresse soit stockée dans un registre qui va momentanément se substituer au compteur

3-3 Adresse valide

Nous avons vu que les modifications de l'adresse se font généralement en deux temps il y a donc des moments où l'adresse est en cours d'évolution, le danger est alors que les décodeurs d'adresse soient trompés. Pour éviter cela le microprocesseur doit informer les décodeurs lorsque l'adresse est valide. Le bit VMA (Validation Memory Address) est destiné à cet usage.



Nous avons déjà évoqué le rôle de la sortie Sc de l'UAL avec les bits N, Z, V, C, ces bits vont être stockés dans un registre. Nous les appellerons bit d'état.

Dans l'exemple déjà évoqué, lorsque le microprocesseur à lu la valeur du capteur il va lui soustraire la valeur de consigne, trois cas peuvent se présenter

Les deux valeurs sont identiques, le résultat de la soustraction est 0 (8 fois 0) le bit Z va passer à 1 dans ce cas, puisque la puissance de chauffe est bonne, le microprocesseur doit enclencher une temporisation puis à nouveau interroger le capteur

La grandeur de consigne est plus grande que celle du capteur, la soustraction va donner un résultat négatif, le bit N va être à 1 le microprocesseur va sauter dans le programme

consistant à augmenter la puissance de chauffage, il enclenchera la temporisation puis interrogera à nouveau le capteur

La grandeur de consigne est plus faible, le résultat de la soustraction est positif, le bit N est à 0, le microprocesseur saute dans le programme chauffer moins puis va dans la temporisation après quoi il interrogera à nouveau le capteur.

Ces trois cas nous montre le rôle des bits Z et N dont les états vont « conditionner » la suite du programme.

Le registre qui va contenir ces bits s'appellera le **registre de codes conditions ou CCR**, il sera lui aussi relié au bus de données de façon à pouvoir être envoyé à l'extérieur du microprocesseur. Nous obtenons maintenant un schéma interne de notre microprocesseur assez complet à notre échelle.

