

Architecture

Bon alors, c'est quoi un ordinateur ?

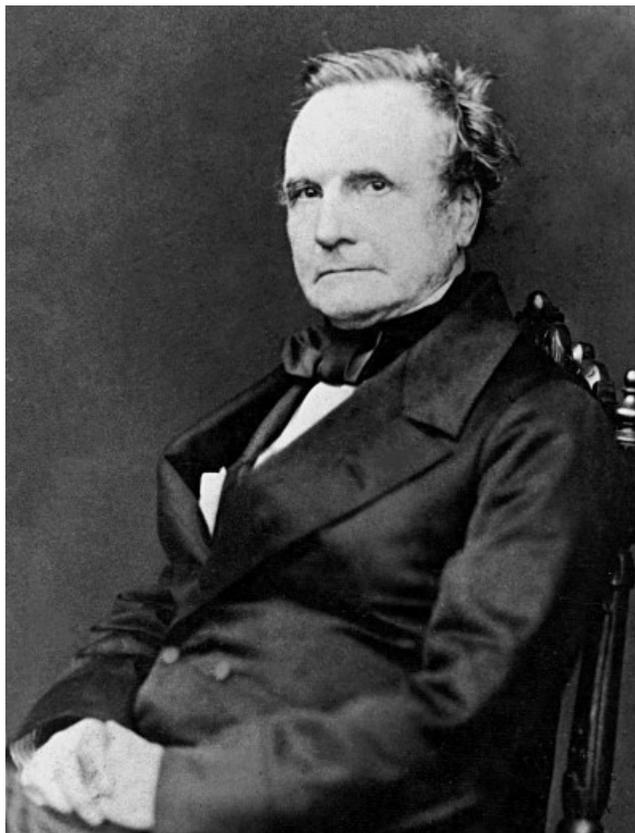
Et bien, c'est à cette question (qu'on pourrait croire bête) que nous allons répondre aujourd'hui.

1 - Un peu d'histoire

De quand date le premier ordinateur de l'histoire ?

Tout dépend de ce qu'on considère être un ordinateur.

En 1834, le mathématicien anglais Charles Babbage fut l'un des précurseurs de l'informatique.



Charles Babbage - Image du domaine public USA

Il fut le premier à énoncer le principe d'un ordinateur. Il a travaillé sur le principe d'une machine à calculer qui incorpore les cartes perforées des machines à tisser Jacquard (inventées en 1801 par le lyonnais Joseph Marie Jacquard).

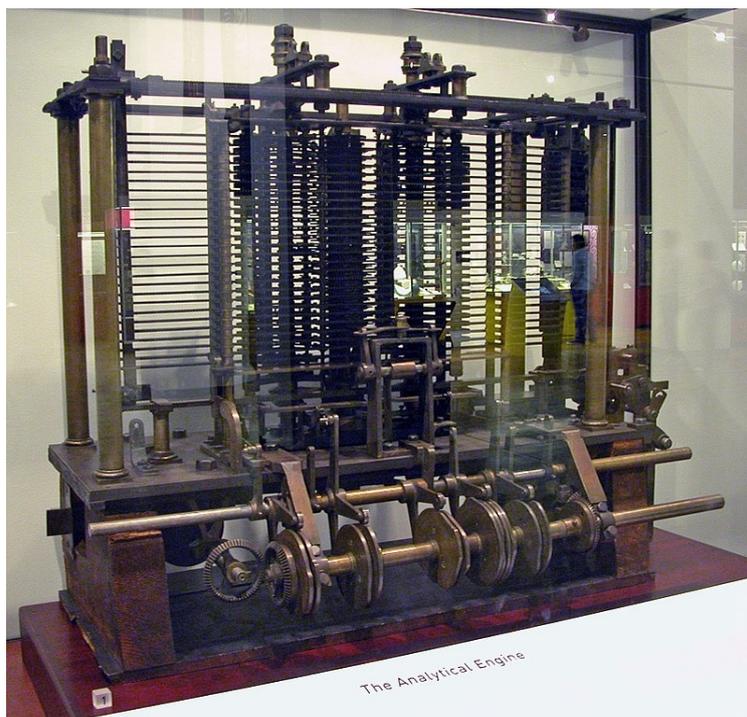


Météier Jacquard [Musée des Arts et Métiers](#) [CC BY-SA]

Partant de ce principe, Charles Babbage imagine un calculateur mécanique qui recevrait instructions et données à partir de la lecture des perforations sur les fiches.

Son idée était de produire des tables nautiques, astronomiques et mathématiques exactes. A l'époque, beaucoup de tables comportaient des erreurs à cause d'erreurs de recopies ou d'erreurs de calcul humain.

Il passa le reste de sa vie à travailler sur les détails de cette machine analytique, tentant de la faire fonctionner.

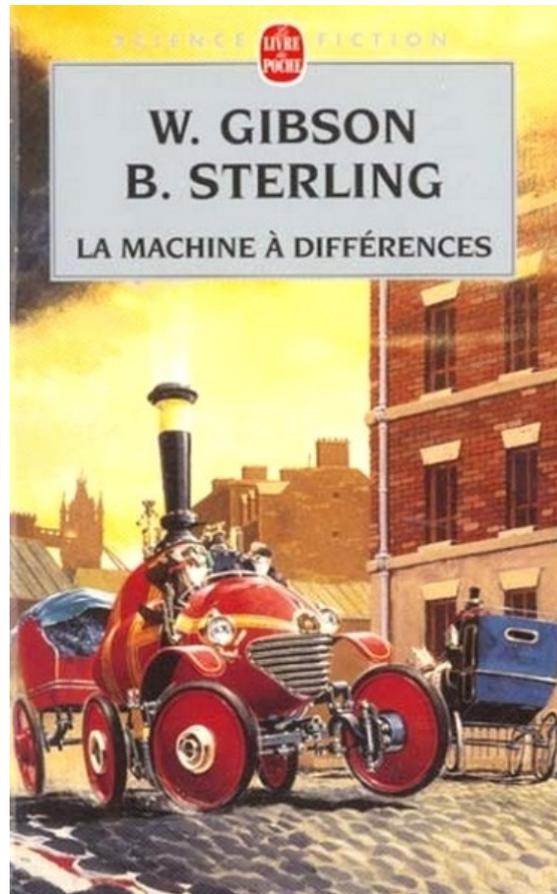


Une partie de la machine analytique, construit par Charles Babbage, exposée au Science Museum de Londres.

[Musée des Arts et Métiers](#) (photo Bruno Barral) [CC BY-SA]

Même s'il ne parvint pas à la construire de son vivant, elle fût bien construite à partir des plans de Babbage mais en 1991 (avec l'équivalent des technologies d'époque). Avec un peu plus de chance, réussir à construire sa machine. La théorie était bonne et fonctionnelle.

L'Histoire aurait sans doute été bien différente si la machine avait vu le jour. La Machine à différences, un roman de science-fiction est construit autour de cette uchronie : que ce serait-il passé si Charles Babbage avait produit une machine à différence ?



Couverture de la Machine à Différences de William Gibson et Bruce Sterling

Pendant ces nombreuses années de recherche, Charles Babbage entre en contact avec de nombreuses personnes.

Parmi elles figure Ada Lovelace, de son nom complet Augusta Ada King, comtesse de Lovelace, née Ada Byron. Il s'agit de la fille du poète Lord Byron, l'un des poètes anglais les plus connus de cette époque. Elle ne connut pas son père puisqu'il quitta l'Angleterre pour toujours alors qu'elle n'avait qu'un an.

Eduquée par sa mère, elle-même cultivée et intelligente, Ada Lovelace parvient à devenir une jeune fille particulièrement douée et éduquée en mathématiques, ce qui est très inhabituel à l'époque : la maîtrise des sciences étant un symbole de puissance et de contrôle, on écartait les femmes de cette connaissance.

Lors de ses premières correspondances avec Charles Babbage, elle est à l'époque une jeune fille de 17 ans.



Ada Lovelace, née Ada Byron

Ada Lovelace est immédiatement fascinée par la machine théorique de Charles Babbage. Même après son mariage, elle continuera ses études de mathématiques, parvenant même à obtenir comme tuteur le célèbre mathématicien Auguste De Morgan.

Elle étudiera la machine avec passion et parviendra à créer le premier programme de l'histoire ! En collaboration avec Charles Babbage, elle mettra au point un algorithme permettant de lancer un calcul très complexe sur la machine.

Elle est aujourd'hui considérée comme la première programmeuse de l'histoire. Aucun humain avant elle n'avait conçu de programme potentiellement exécutable sur une machine réelle. Son programme comportait d'ailleurs la première boucle non bornée (TANT QUE) non purement théorique de l'histoire. Son programme était donc non purement séquentiel, contrairement aux instructions envoyées aux machines Jacquard.

Une autre idée révolutionnaire qu'on lui attribue : l'idée d'encoder autre chose que des nombres avec des nombres ! Maintenant que vous avez vu qu'on pouvait encoder un texte, une image ou une relation à l'aide d'une suite de nombres, cela vous paraît peut-être évident, mais comme toujours, tant que personne n'en a eu l'idée, une invention n'existe pas. Elle aurait ainsi eu l'idée d'utiliser la machine pour créer des morceaux de musique.

Alternant entre haut et bas, elle tentera de créer des algorithmes permettant de gagner aux courses hippiques (pour subventionner la machine !) mais dilapida malheureusement son argent. Elle mourut jeune, à l'âge de 36 ans.

Voici une illustration de son programme : [image du programme](#) (Cette œuvre est dans le domaine public dans tous les pays pour lesquels le copyright a une durée de vie de 100 ans ou moins après la mort de l'auteur).

Pour l'anecdote, on peut voir son portrait sur les hologrammes d'authentification des produits Microsoft.

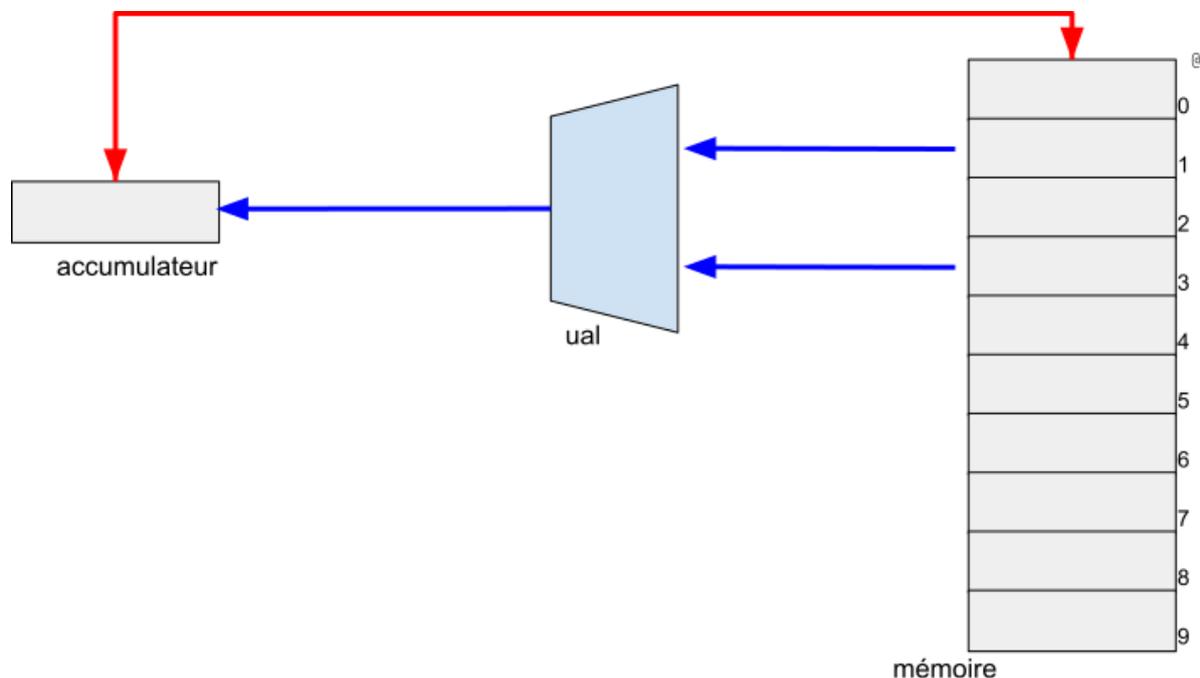
2 - Présentation M10

Aujourd'hui, nous allons vous offrir un ordinateur. Un vrai. Mais en papier.

M-10 est une petite machine informatique constituée

- d'une mémoire qui contient des données,
- d'une unité arithmétique et logique (UAL) capable de réaliser une opération telle une addition,
- d'un registre nommé accumulateur qui recevra le résultat des opérations réalisées par l'UAL.

La mémoire est composée de 10 "cases". Ces cases sont numérotées de 0 à 9. Ce numéro est l'adresse de la case.



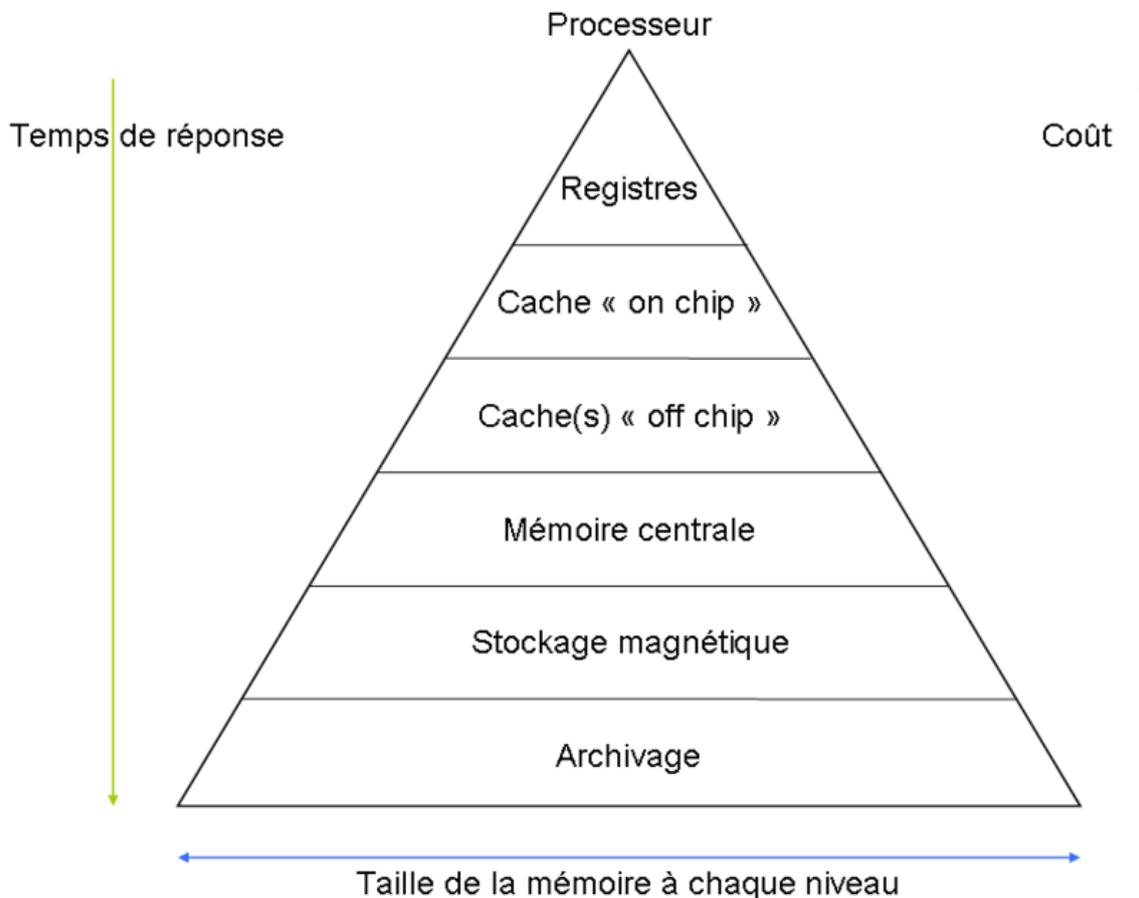
La structure de la machine M10

Registre

Qu'est-ce qu'un registre ?

Il s'agit d'une zone-mémoire interne au processeur. L'accès à cette mémoire est donc très rapide puisqu'elle est au sommet de la hiérarchie des mémoires. Il s'agit d'une zone-mémoire qui est réellement physiquement très proche du processeur.

Dans les architectures qualifiées de load-store, les programmes transfèrent d'abord des données de la mémoire centrale (grand stockage mais peu rapide) vers des registres (faible stockage mais rapide), puis effectuent des opérations sur ces registres, et enfin transfèrent le résultat en mémoire centrale.

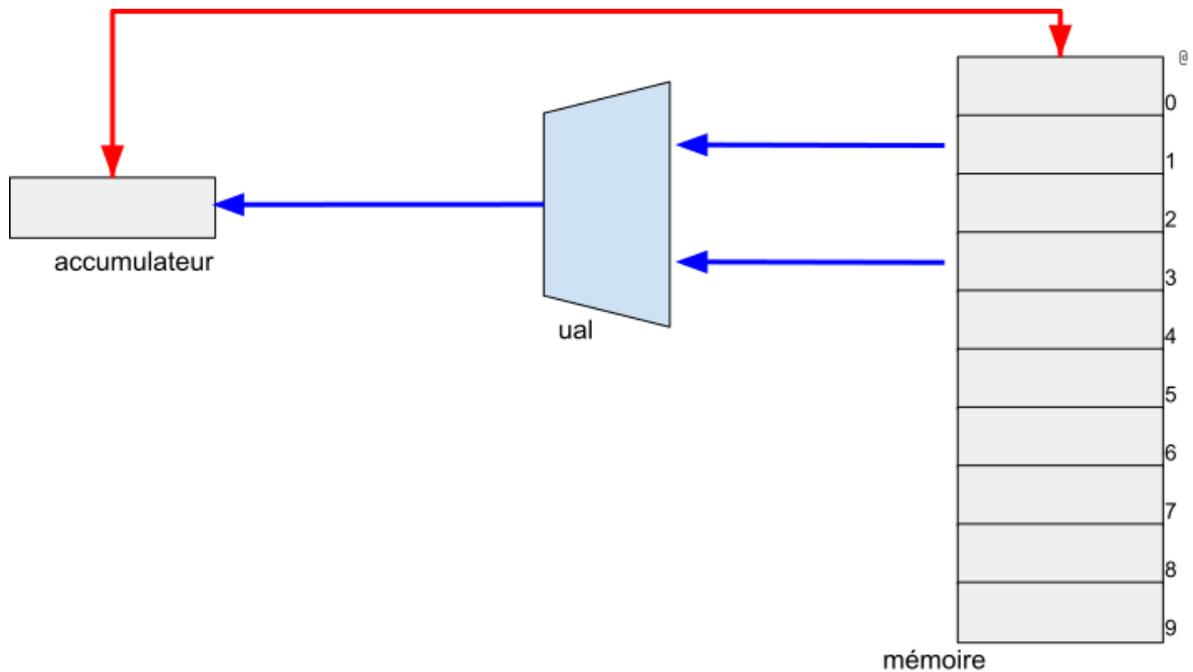


Par Original téléversé par [Poil](#) sur [Wikipédia français](#). — Auteur: [Stéphane](#) 3 avr 2005 à 03:54 (CEST)Source PPT disponible sur demande., [CC BY-SA 3.0](#), [Lien](#)

On voit donc apparaître plusieurs zones de stockage différentes

- Registres : les fameuses mémoires proches du processeur
- Caches on-chip : des zones mémoires assez rapides situées directement sur la carte à puce (chip) du microprocesseur. On y stocke les résultats précédents de façon à pouvoir les retrouver sans avoir à tout recalculer.
- Caches off-chip : des zones mémoires assez rapides situées en dehors de la carte du processeur lui-même.
- Mémoire centrale : une mémoire bien plus grande mais également beaucoup plus longue en terme d'accès.
- ...

Voilà. On peut reprendre.



La structure de la machine M10

Jeux d'instructions

Notre machine possède deux jeux d'instructions :

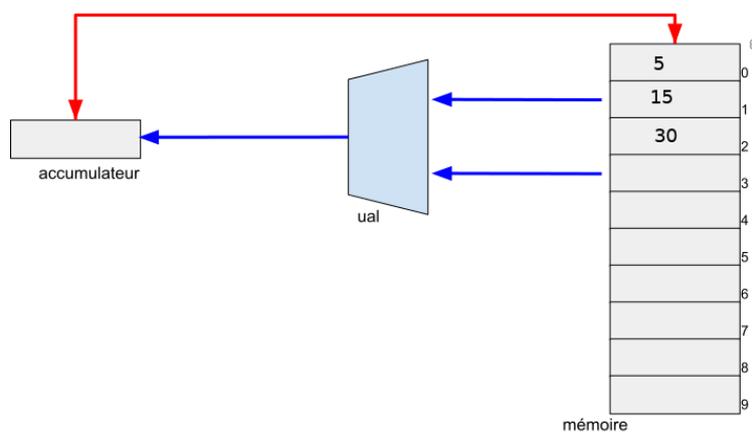
1. les instructions arithmétiques et logiques réalisées par l'UAL (+, -, *, ET, OU ...) à partir des deux données reçues,
2. les instructions de transfert entre la mémoire et l'accumulateur.

Voyons une première instruction qu'on notera ADD et qui permet de faire une addition.

Un tout petit rikiki programme : ADD @1 @2.

Que fait-il ?

1. Il commande à l'UAL de faire une addition. Pour cela :
2. Il place le contenu de la case-mémoire 1 à l'entrée de l'UAL.
3. Il place le contenu de la case-mémoire 2 sur l'autre entrée de l'UAL.
4. Il additionne les deux entrées et place le résultat dans l'accumulateur



01° Que contient l'accumulateur après exécution du programme (ADD @1 @2) ?

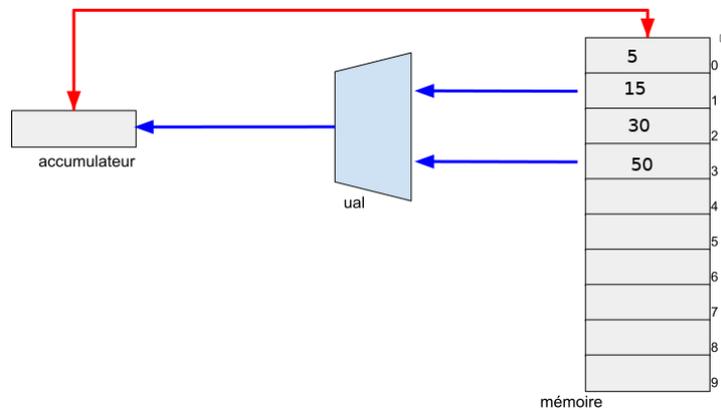
Une fois le résultat dans l'accumulateur, comment parvenir à le placer en mémoire ?

Et bien imaginons une instruction STR @3 (STR pour STORE from REGISTER) qui permet de placer le contenu de l'accumulateur dans la case-mémoire 3.

D'ailleurs, imaginons aussi l'inverse : LDR @5 (LDR pour LOAD to REGISTER) qui permet de placer le contenu de la case-mémoire 5 dans l'accumulateur.

02° Exécuter le programme suivant en utilisant la mémoire initiale représentée ci-dessous. Que contiennent les mémoires à la fin ? On remarquera qu'on part à chaque fois chercher deux valeurs en mémoire centrale.

ADD @2 @3 STR @4 ADD @3 @4 STR @5

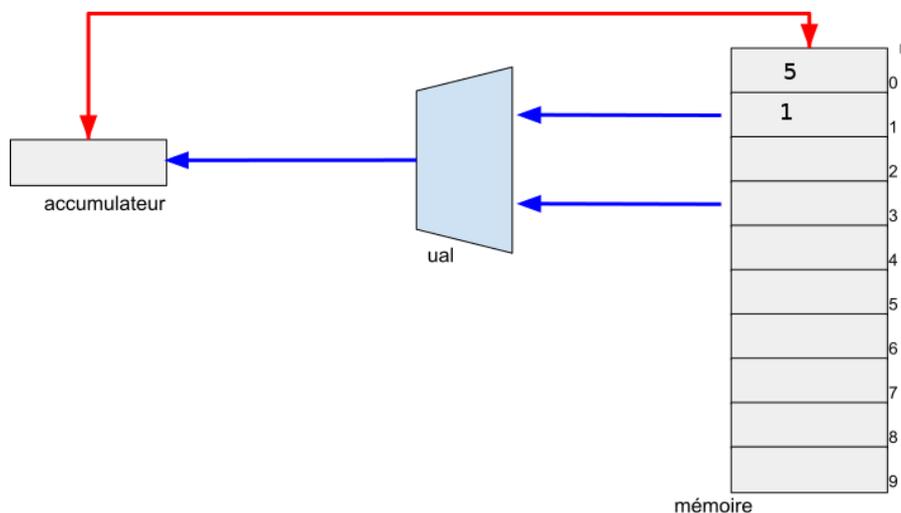


Faisons une liste des instructions disponibles :

1. ADD @2 @3 : additionne les contenus des zones-mémoires et place le résultat dans l'accumulateur
2. SUB @2 @3 : Calcule contenu 2 - contenu 3 et place le résultat dans l'accumulateur
3. STR @5 : copie le contenu du registre-accumulateur en zone-mémoire 5.
4. LDR @5 : copie le contenu de la zone-mémoire 5 dans l'accumulateur.

03° Exécuter le programme suivant en utilisant la mémoire initiale représentée ci-dessous. Que contiennent les mémoires à la fin ?

SUB @0 @1 STR @0 SUB @0 @1 STR @0

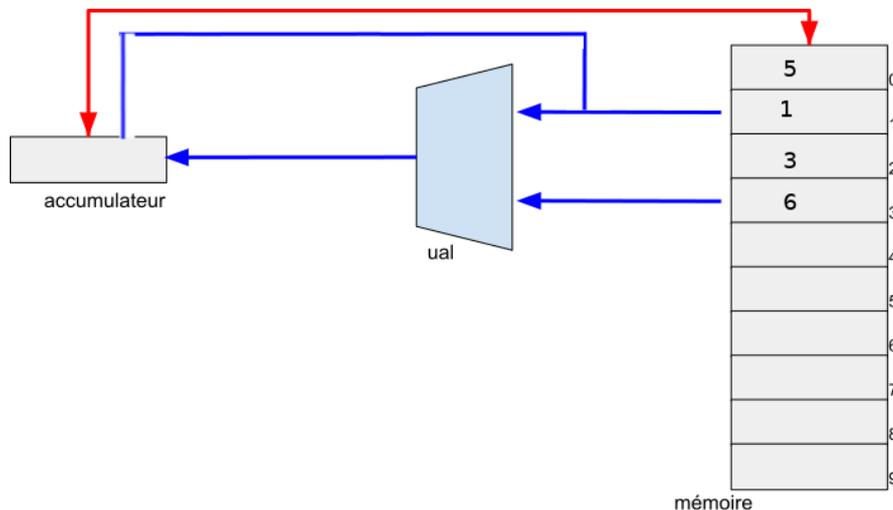


Le problème est que l'accès à la mémoire centrale est plutôt lent. Or ici, on ne fait que stocker le résultat pour aller ensuite le reprendre...

On pourrait au contraire utiliser une instruction utilisant le résultat contenu dans l'accumulateur (identifié avec une adresse du type @A) pour aller directement le replacer à l'entrée de l'UAL.

04° Exécuter le programme suivant en utilisant la mémoire initiale représentée ci-dessous. Que contiennent les mémoires à la fin ? Vous devriez constater qu'on effectue une somme, mais en ne faisant qu'un seul appel à la mémoire centrale à chaque étape.

```
SUB @0 @1 SUB @A @2 SUB @A @3 STR @0
```



Il reste toujours pas mal de questions en suspens. Quatre exemples :

1. Où sont stockés les programmes là-dedans ? Ça doit être forcément dedans puisque le programme fait partie de la machine. Mais où ?
2. Comment fait la machine pour faire une addition ?
3. Si on veut récupérer des données externes (clavier ?), on récupère à partir de quelle provenance ?
4. Si on veut envoyer des informations vers l'extérieur (écran ?), on envoie vers quelle destination ?

3 - Machine de Turing

La théorie de la calculabilité est un domaine de la logique mathématique et de l'informatique théorique. Elle permet d'identifier la classe des fonctions qui peuvent être calculées à l'aide d'un algorithme (c'est à dire en utilisant un nombre fini d'instructions et un nombre fini de zones mémoires).

Alan Turing (1912-1954) est un mathématicien et cryptologue britannique. Nous allons voir que ses recherches fondent la science informatique.



Alan Turing

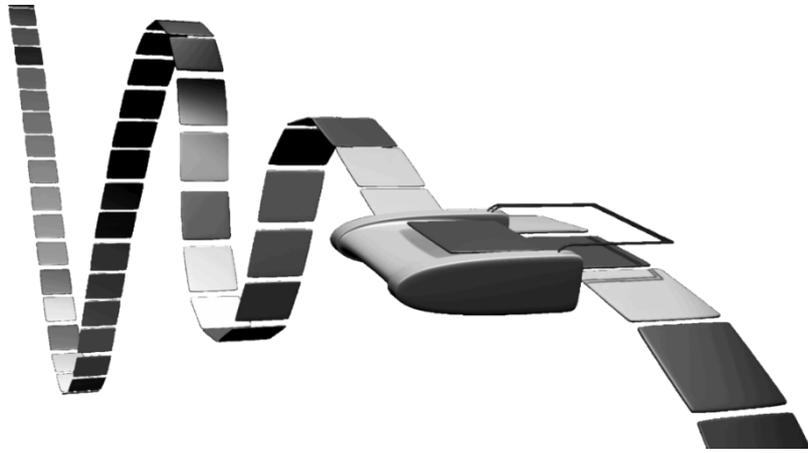
Il va notamment répondre à l'une des questions fondamentales de la théorie de la décidabilité : un programme a qui on fournit le code d'un autre programme ainsi que les données qui devront être traitées peut-il à tous les coups prédire si le programme va s'arrêter et fournir une réponse ou s'il va tourner en boucle ? On nomme cela le problème de l'arrêt (et il sera étudié en Terminale NSI).

Pour répondre à cette question, Alan Turing présente en 1936 une machine imaginaire, nommée depuis machine de Turing.

En informatique théorique, une machine de Turing est une machine automatique virtuelle permettant d'exécuter des algorithmes. Elle fut d'ailleurs inventée avant le premier ordinateur.

La machine de Turing se compose :

1. d'un ruban infini divisé en cases consécutives qui peuvent contenir un ensemble fini de symboles (initialement toutes les cases contiennent un symbole signifiant VIDE)
2. d'une tête de lecture-écriture qui se positionne autour du ruban et qui permet donc de lire l'une des cases ou d'y écrire un nouveau contenu
3. d'un registre d'état permettant d'enregistrer un nombre fini d'états (au départ, il contient un code signifiant "ETAT DE DEPART")
4. Une table d'action qui indique ce qu'on doit faire lorsque la machine lit tel ou tel symbole sur la case lue par la tête de lecture.



Vue d'artiste de la machine de Turing (domaine public, publié sur Wikipedia)

A l'aide d'une expérience de pensée et de démonstrations mathématiques, Alan Turing va alors montrer que le problème de l'arrêt est indécidable : un programme ne peut pas déterminer à coup sûr qu'un autre programme dont il peut lire le code et les entrées va s'arrêter.

De façon plus générale, la machine de Turing permet maintenant de définir précisément ce qu'est un algorithme puisqu'il s'agit d'un ensemble d'instructions qui peuvent être exécutées (au moins en esprit) sur cette machine virtuelle.

Alan Turing est donc vu comme l'un des pères de l'informatique.

Il joua un rôle majeur dans la cryptanalyse de la machine Enigma utilisée par les armées allemandes. Ceci fut classé secret défense pendant de nombreuses années après la 2nd Guerre Mondiale

Sans la découverte du moyen de comprendre les communications cryptées par Enigma, la guerre aurait sans doute duré plus longtemps. Et comme les Nazis recherchaient également à produire la bombe atomique, l'enjeu de la rapidité était considérable.

Après la guerre, il participa à l'élaboration des premiers ordinateurs, aux études initiales sur l'intelligence artificielle et à certaines études liant informatique et biologie.

Le rôle fondamental d'Alan Turing lors de la guerre ne sera révélé qu'en 1970. Jusqu'à cette date, son implication dans l'effort de guerre était classifiée secret-défense. Pourtant, il sera poursuivi en justice pour son homosexualité en 1952 et fut retrouvé mort, empoisonné au cyanure, chez lui en 1954. Il n'avait alors que 41 ans.

