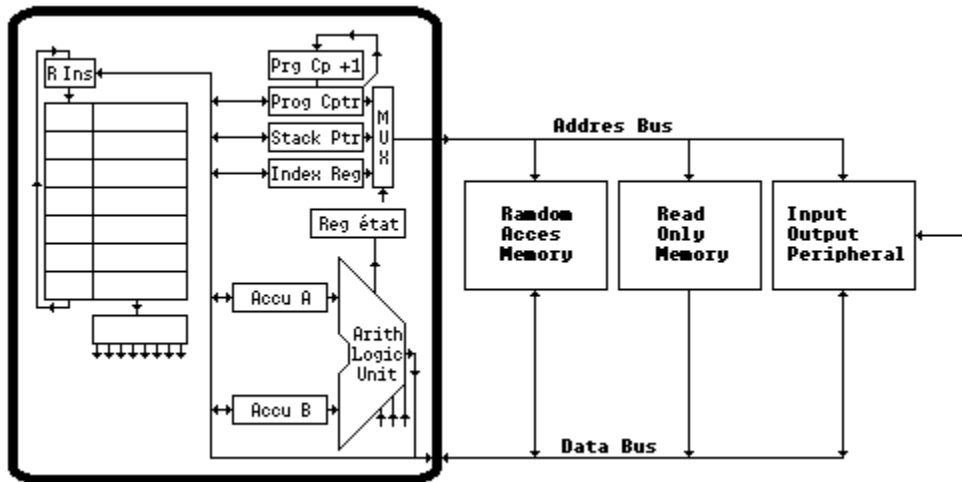


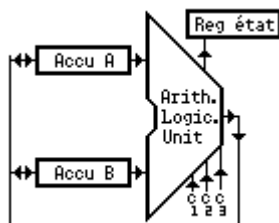
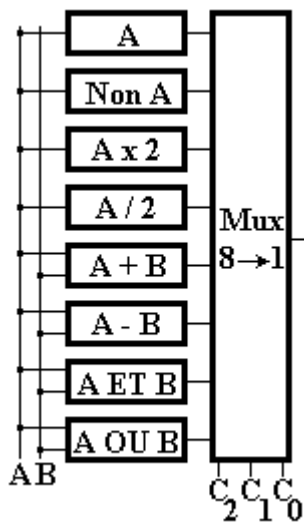
A l'intérieur d'un microprocesseur

Voici le synoptique général d'un microprocesseur très simple (6800 de Motorola). Nous allons progressivement détailler chacune de ses parties.



1) L'unité arithmétique et logique (U.A.L.)

11) Réalisation d'une Unité Arithmétique et Logique



12) Les différents traitements faits par une U.A.L

121) Traitements arithmétiques

Addition, soustraction, multiplication, multiplication et division par 2, complémentation à 1 et à 2.

122) Traitements divers

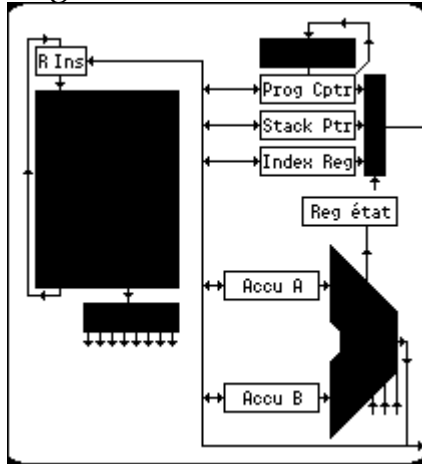
rotations à droite ou à gauche...

123) Fonctions logiques pour traiter les données

ET, OU, PAS, OU_EX, ET_EX, NON_ET, NON_OU .

124) Tests logiques

=, >, <, >=, <=, vide?, non-vide?

2) Les registres internes**21) Quelle est leur fonction ?**

Ce sont des registres qui stockent les données internes du microprocesseur sous forme binaire (avec des "0" ou des "1").

211) Programm Counter (Compteur de programme)

Il sert pour aller chercher l'instruction suivante en mémoire, pour effectuer les branchements conditionnels et pour appeler les sous-programmes.

212) Stack-pointer (pointeur de pile)

Il sert à adresser la pile pour y stocker les adresses de retour de sous-programme, et pour passer des données dans le cadre d'une programmation structurée en assembleur.

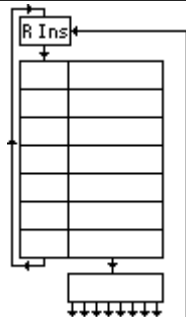
213) Index-register (registre index)

C'est un registre d'indexation pour adresser la mémoire par adressage relatif et par adressage indexé.

22) Ah quelle salade !

A l'intérieur d'un microprocesseur, les données sont stockées sous forme binaire, mais ce peut-être:

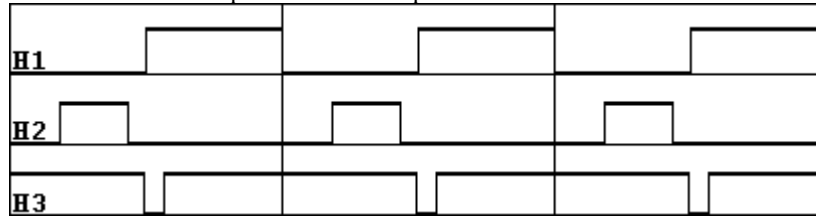
- Des adresses pour adresser la mémoire.
- Des instructions de programmation.
- Des données binaires à traiter.

3) Unité de contrôle

Seule l'U.A.L. fait un travail effectif, la mémoire et les registres mémorisent les données. L'unité de contrôle est le chef d'orchestre qui fait défiler et passer les données dans l'U.A.L. au rythme de l'horloge. Son travail consiste à ouvrir et fermer les portes aux différentes données: "Entrez donc dans la salle d'attente, votre client n'est pas encore arrivé, quand ce sera le cas je vous présenterai à son excellence l'U.A.L.. L'U.C. a une structure d'automate à pile (Voir le cours d'intelligence artificielle - Automates et grammaires).

4) L'horloge interne

C'est le métronome qui bat la mesure pour le chef d'orchestre

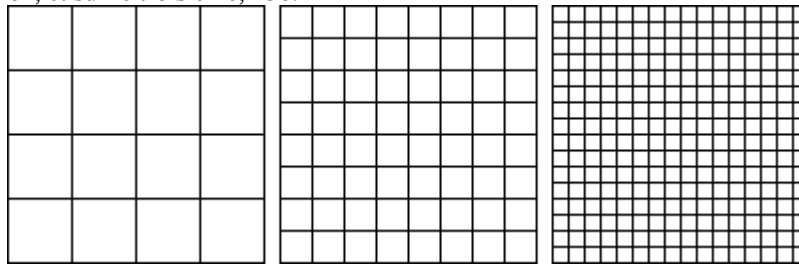


41) Histoire des fréquences d'horloge

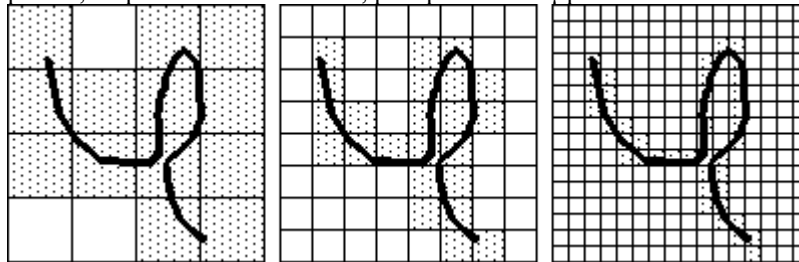
- Le premier μ p que j'ai acheté en kit (SC-MP) pédalait à 1 Mhz et avait besoin de 12 cycles d'horloges pour faire certaines instructions.
- Ensuite j'ai travaillé avec un 6800 qui pédalait lui aussi à 1Mhz, mais était plus rapide.
- L'IBM PC XT pédale à 4,77Mhz.
- Le goupil G4 vers 8 Mhz.
- Les premiers AT 12 Mhz, puis 16.
- Maintenant on est rendu à 25 ou 33Mhz pour les 386 et 486.
- Avec les doubleurs de fréquence, on atteint 50 et même 66Mhz.

42) Electronique sub-micronique: Montée en vitesse des μ p

Depuis quelques temps, l'électronique réussit à intégrer des transistors dont la taille est inférieure au micron. Ce gain de taille permet d'accélérer les μ p en diminuant la longueur des connections. Voici par exemple, intégrés sur le même wafer(gaufre), des μ p dont le côté est divisé par deux à chaque étape. Sur le premier wafer on intègre 16 μ p, sur le second 64, et sur le troisième, 256.



L'augmentation de la densité d'intégration induit une plus grande immunité aux défauts des wafers. Voici par exemple un défaut que l'on va reproduire trois fois identiquement. On remarque que dans le premier cas presque tous les μ p sont perdus, et que dans le dernier cas, presque tous les μ p sont bons.



5) La structure pipe-line

Le principe du travail à la chaîne

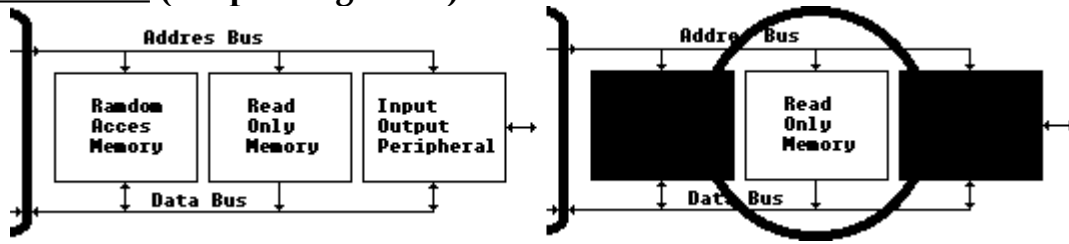
Soit le cas d'une chaîne de montage de 100 postes, où une voiture avance toutes les 6 minutes. Pendant 6 minutes, chacun des 100 ouvriers monte un module. Pour monter une voiture, il faut 6×100 minutes (10 heures), pourtant, toutes les 6 minutes, une voiture sort de l'usine.

Application à l'informatique

La structure pipe-line, c'est le principe du travail à la chaîne appliqué à l'ordinateur: Pendant que l'UAL effectue un calcul, d'autres registres préparent l'instruction suivante, alors que certains n'ont pas encore rangé le résultat de l'opération précédente.

A l'intérieur d'un microprocesseur

1) La mémoire (ci-après à gauche)

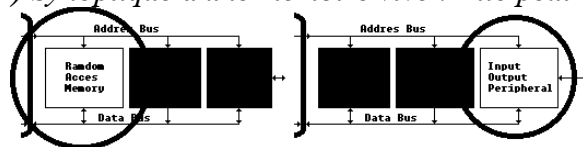


11) Synoptique d'une mémoire: C'est une table

111) Synoptique d'une mémoire morte : Elle ne peut qu'être lue. (ci-avant à droite)

Les flèches indiquent que si je fournis une adresse sur le bus adresse, je récupère une donnée sur le "data bus".

112) Synoptique d'une mémoire vive : Elle peut se lire et s'écrire (ci-après à gauche)



Les flèches indiquent que si je fournis une adresse sur le bus adresse, soit je récupère une donnée sur le "data bus"(lecture), soit je lui en fournis une.

12) Certains périphériques apparaissent au up comme une mémoire (ci-avant à droite)

Les flèches indiquent que si je fournis une adresse sur le bus adresse, soit, sur le "data bus", je récupère une donnée en provenance du périphérique (input - Ex: lecture au clavier), soit je lui en fournis une (output - Ex: écriture sur l'imprimante).

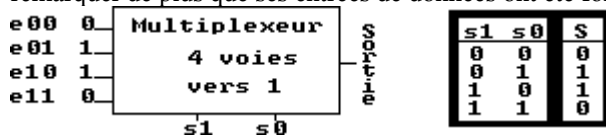
13) Notion de mémoire morte

Si je photographie un jeu d'échec j'ai une mémoire, car je puis toujours répondre à des questions sur le contenu des cases du jeu d'échec, mais je suis bien sûr dans l'impossibilité de changer le contenu de ces cases. C'est pourquoi on dit que je suis en présence d'une mémoire morte: Elle est "programmée une fois pour toute: On peut la lire (l'interroger), mais on ne peut pas l'écrire (changer le contenu de ses adresses).

14) Utilisation d'un multiplexeur pour faire une mémoire ROM

141) Schéma synoptique d'un multiplexeur 4 vers 1. (ci-après à gauche)

Nous voici donc en présence d'un multiplexeur à une sortie, quatre entrées de données, et deux entrées de sélection. Il faut remarquer de plus que ses entrées de données ont été forcées à 1 ou à 0.



142) Fonctionnement en mémoire

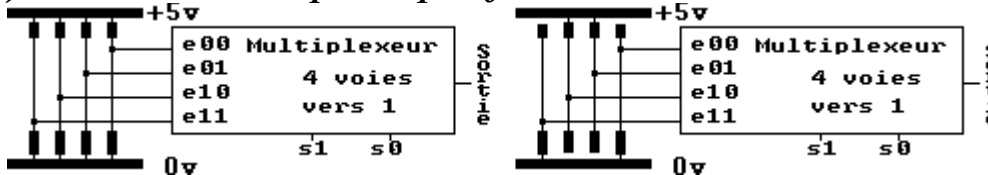
Si on ne touche pas aux entrées de données, on peut considérer les entrées de sélection comme des adresses et si on fait varier les entrées de sélection, en sortie on obtient le contenu de notre mémoire.

Si, sur les entrées de sélection on met: "00" alors le multiplexeur valide vers la sortie la voie OO qui a été initialisée à 0, et on obtient 0 en sortie. Etc ... On peut ainsi sonder notre mémoire in extenso et en remplir sa table de vérité. On obtient la table ci-avant à droite.

Ainsi au travers de cet exemple nous avons vu comment on peut obtenir une porte élémentaire: ET, OU, PAS, à partir d'une mémoire.

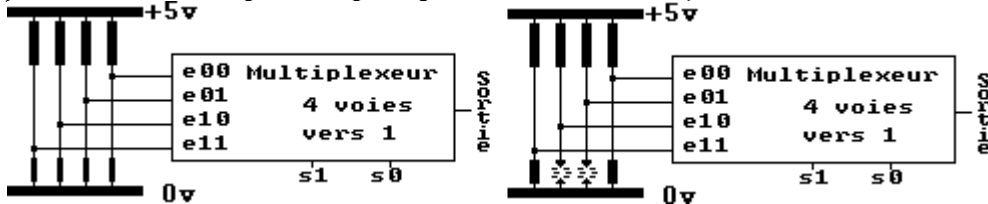
L'entrée des données est définitivement câblée. Nous avons ici une mémoire que l'on peut seulement lire (Read Only Memory).

143) Utilisation d'un multiplexeur pour faire une PROM



Au départ le fabricant livre une mémoire non programmée: A gauche, on peut voir que les entrées sont connectées à un pont alimenté entre Le 5v et la masse et qui fournit un niveau intermédiaire et bâtarde. Par programmation on fusionne une des connections des ponts qui fixent les entrées et on obtient la figure ci-avant à droite.

144) Utiliser un multiplexeur pour faire une REPRON, EEPROM



Au départ, le fabricant livre un multiplexeur dont les entrées sont forcées à 0 par un pont diviseur dissymétrique (Figure de gauche). La mémoire ne contient donc que des 0. Par programmation, en envoyant de grosses impulsions négatives, on piège des électrons sur certaines des résistances connectées à la masse. Ceux-ci, bloquent le canal conducteur en repoussant les charges + qui voudraient passer, l'entrée se trouve ainsi forcée à 1 (figure de droite).

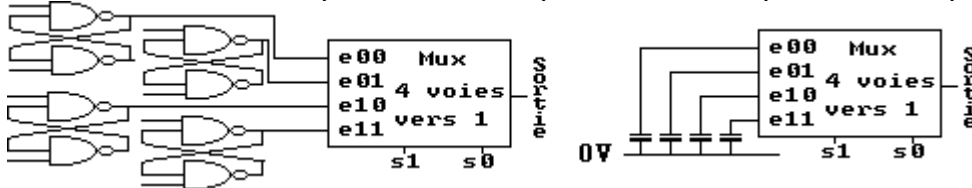
15) Différentes technologies de mémoires vives

151) Notion de mémoire vive

Quand on peut changer la valeur inscrite dans une mémoire on dit que c'est une mémoire vive.

1511) Notion de mémoire vive statique: (ci-après à gauche)

L'information est mémorisée par une bascule RS qui est une structure qui fonctionne en permanence.



1512) Notion de mémoire vive dynamique: (ci-avant à droite)

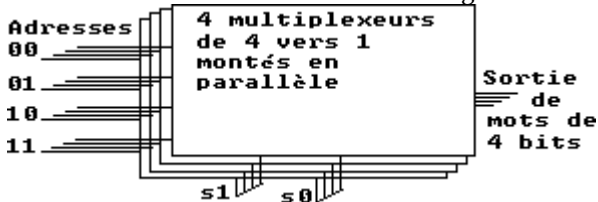
L'information est mémorisée par une capacité qui voit sa charge varier lentement au fil du temps. Ce type de mémoire doit partiellement être immobilisée et relue systématiquement pour rafraîchir l'information qu'elle contient. Dans ce cas l'ordinateur est immobilisé. Ces mémoires, plus faciles à construire, sont bon marché, mais moins pratiques.

16) Regroupement en parallèle des mémoires

Mémoire morte de 4 adresses de 4 bits:

Les entrées de données servent à la programmation.

Les entrées de sélection servent à l'adressage.



On peut regrouper des mémoires ensemble pour additionner leurs capacités. Il existe différents montages, dont le montage en parallèle présenté ici.

Les adresses: Elles sont reliées ensemble et ainsi les différentes mémoires reçoivent la même demande d'adresse.

Les sorties: Elles sont présentées en parallèle.

Dans l'exemple de ce chapitre on a utilisé quatre multiplexeurs de quatre bits pour obtenir une mémoire de 4 adresses de 4 bits.

17) Regroupement en série des mémoires

171) Notre but: Obtenir une mémoire de 8 adresses de 4 bits

Dans ce cas, cette mémoire de 8 adresses sera commandée par 3 entrées de sélection: S_2 , S_1 et S_0 , et elle a 4 bits d'information en sortie.

172) Ce dont nous disposons

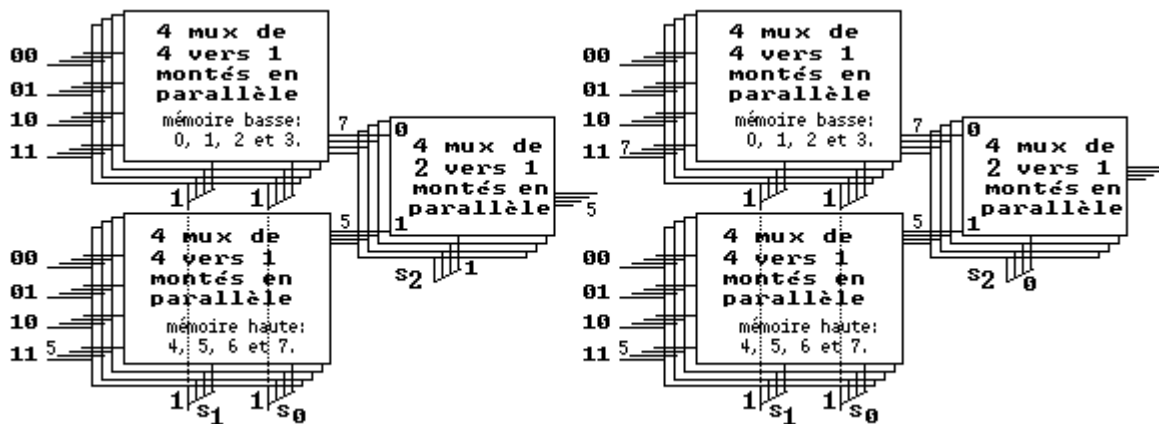
Nous avons vu comment fabriquer une mémoire de 4 adresses de 4 bits. Dans ce cas, cette mémoire de 4 adresses est commandée par 2 entrées de sélection: S_1 et S_0 , et elle fournit 4 bits d'information en sortie.

173) A nous de câbler

Prenons 2 de ces mémoires et connectons leurs 4 sorties sur 4 multiplexeurs de 2 vers 1. Commandons ce multiplexeur par S_3 , sachant que nos deux mémoires sont déjà commandées par S_1 et S_0 .

174) Si je mets 3 en entrée (figure de droite)

J'ai $S_2=0$, $S_1=1$ et $S_0=1$, Les deux mémoires (basse et haute) fournissent chacune une information (3 et 7), mais les 4 multiplexeurs de 2 vers 1, ne laissent passer que '7', l'information de la mémoire basse (car $S_2=0$).



175) Si je mets 7 en entrée (figure de gauche)

J'ai $S_2=1$, $S_1=1$ et $S_0=1$, Les deux mémoires (basse et haute) fournissent toujours la même information (3 et 7), mais les 4 multiplexeurs de 2 vers 1, ne laissent passer que '5', l'information de la mémoire haute (car $S_2=0$).

J'ai $s_2=1$, $s_1=1$ et $s_0=1$, Les deux mémoires fournissent information, mais les 4 multiplexeurs de 2 vers 1, ne vont laisser passer que l'information de la mémoire haute (car $s_2=1$).

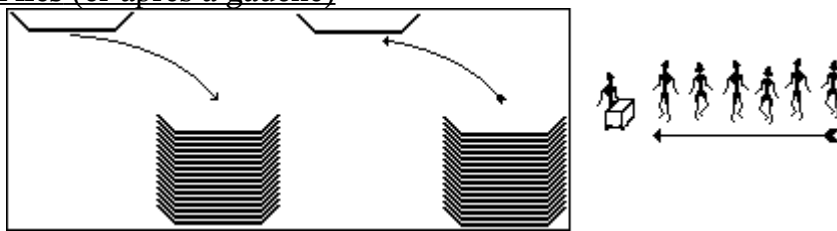
Conclusion

Il faut donc retenir trois choses de ce travail

- La structure interne d'une mémoire est simple et ne recèle aucune intelligence.
- En groupant (série ou parallèle) des mémoires on obtient n'importe quelle taille.
- Le fonctionnement synoptique d'une mémoire est le suivant:

On lui fournit une adresse binaire. Elle rend en sortie une information binaire sur N bits qui provient du contenu de l'adresse demandée en entrée.

18) Piles (ci-après à gauche)



Quand j'empile des assiettes, la dernière empilée sera la première reprise. En anglais on appelle cette structure: "FIFO"(First In, First Out).

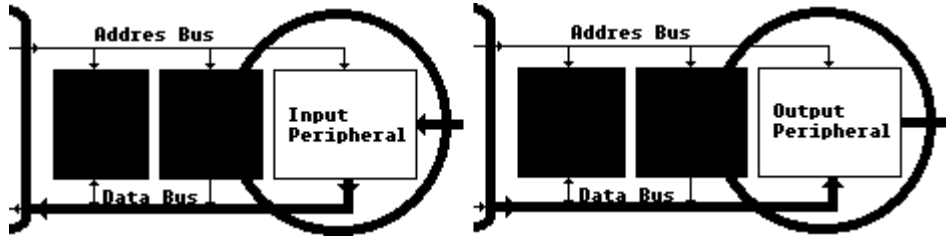
19) Files (ci-avant à droite)

Au guichet d'une administration, le premier arrivé est le premier servi. En anglais, on appelle cette structure: "LIFO (Last In, First Out).

2) Les circuits d'entrée-sortie série ou parallèle

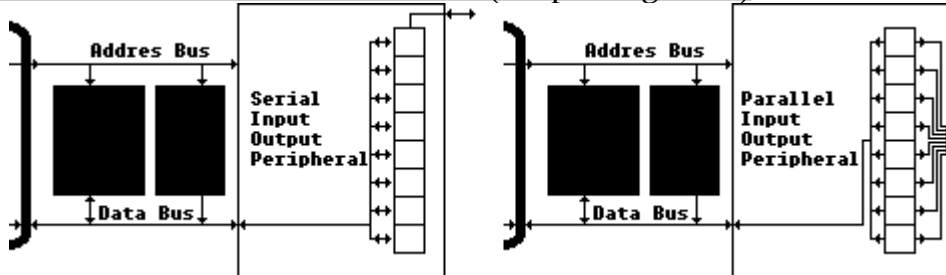
21) Synoptique: Ils apparaissent comme des adresses mémoires

Les circuits vers des périphérique d'entrée apparaissent comme des mémoires que l'on peut lire (Figure de gauche).



Les circuits vers des périphérique de sortie apparaissent comme des mémoires que l'on peut écrire (Figure de droite).

22) Les entrées sorties en mode série (ci-après à gauche)

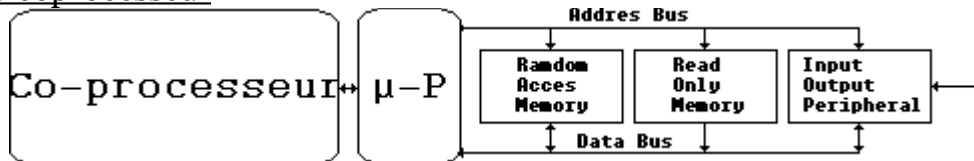


Les données, en provenance de l'intérieur du μ p, arrivent en parallèle par le "data bus". Elles sont sérialisées dans le coupleur de périphérique (ACIA), et sortent bit à bit vers l'extérieur.

23) Les entrées sorties en mode parallèle (ci-avant à droite)

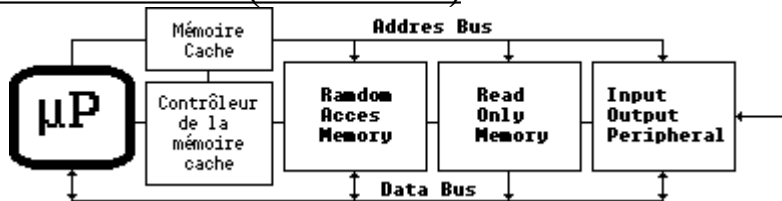
Les données, en provenance de l'intérieur du μ p, arrivent en parallèle par le "data bus". Elles sont transmises par le coupleur de périphérique (PIA), et continuent de sortir en parallèle vers l'extérieur.

2) Le coprocesseur



C'est une super Unité arithmétique et logique
Il est jusqu'à cent fois plus rapide que l'U.A.L.
Mais le gain effectif va de 10 à 20.

4) La mémoire-cache (état d'attente)



41) Historiquement elle apparaît avec les 386

Les 386 et suivants, sont tellement rapides, que la mémoire RAM ne peut plus répondre à temps à leur demande : Ils sont obligés d'attendre.

42) Elle évite le wait-state

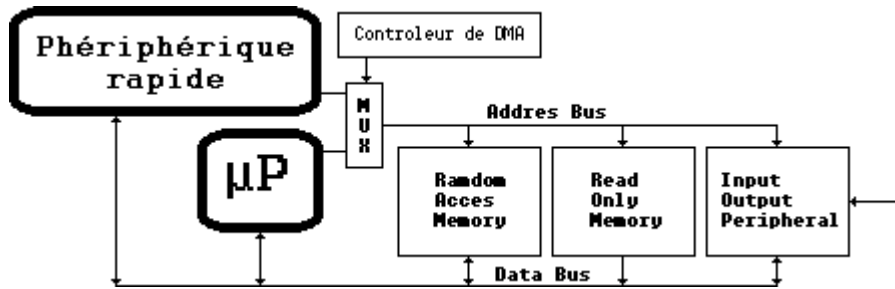
Cet état d'attente du μ p, est préjudiciable à la vitesse. En anglais il se traduit mot à mot par "wait-state".

43) Principe de fonctionnement

On utilise une "mémoire cache", bien moins grande, mais plus rapide que la RAM. Elle sert à dupliquer les adresses les plus utilisées de la RAM. Quand le μ p présente une requête, si la donnée est disponible dans la mémoire-cache, on peut lui fournir rapidement, sinon on lui fournit, avec un peu de retard, la donnée de la RAM classique.

5) Les boîtiers de DMA (Direct Memory Acces)

Ils permettent aux périphériques un accès direct à la mémoire
Synoptique du câblage



6) Les contrôleurs de disque

Rôle

7) Les bus

71) Les bus de la micro

L'ancien standard ISA de IBM

Pour le PC XT de IBM 4,7 Mhz, 1981 8 bits.

Pour le 286 8 Mhz, 16 bits. La bande passante est de 16 MB/s

Puis le bus PCI 32 bits

Pour le Pentium, 33 Mhz.

La bande passante est de 132 MB/s (i.e. 7 ou 8 fois celle du bus ISA)

Utilisé au début pour la carte écran, généralisé aujourd'hui (son, modem, réseau).

Le bus AGP

Bus pour la vidéo et les échanges mémoire ↔ processeur (DMA).

AGP 1X 66 Mhz : BP = 264 MB/s

AGP 2X 66 Mhz : BP = 512 MB/s (front montants et descendants).

AGP 4X 66 Mhz : BP = 1070 MB/s (Dédoublé sur les fronts).

Calcul de la bande passante BP d'un bus

La référence initiale est le PC XT avec un bus de 8 bits (ou un byte), alors l'unité est le byte par seconde, i.e. B/s.

$BP = \text{Fréquence en hz} * \text{Largeur du bus en bits} / 8$

Ou, avec un byte = 8 bits :

$BP = \text{Fréquence en hz} * \text{Largeur du bus en bytes}$.

Note : en hardware, on utilise des kilos de 1000 plutôt que 1024, ce qui peut expliquer les petites différences de calcul.

72) Les bus professionnels

Destination multiprocesseurs

Hautes performances

Débits des infos

Largeur du bus

Interchangeabilité des processeurs

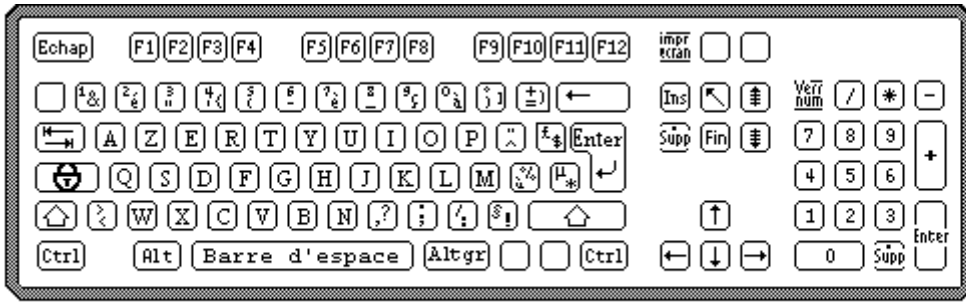
Hautes performances de fiabilité

Ce n'est pas notre domaine

Les coûts sont très élevés car on privilégie les performances

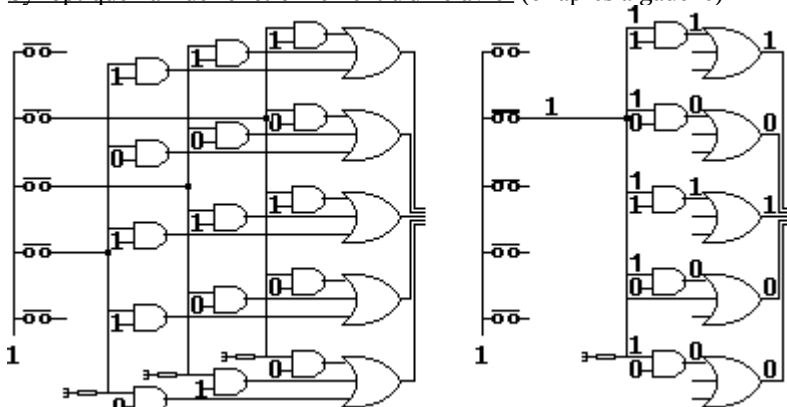
Les périphériques d'entrée

1) Le clavier



11) Synoptique naïf du périphérique type en entrée : Le clavier

Synoptique naïf de fonctionnement d'un clavier (ci-après à gauche)



Quand l'utilisateur tape sur une des touches du clavier (ci-avant à droite), celle-ci fournit un "1" de validation qui vient débloquer la rangée des "ET", et le code de la touche apparaît à leur sortie. La rangée de "OU" collecte cette information, et en sortie on obtient: "10100".

12) Conversion ASCII -> binaire d'un caractère numérique en entrée

121) La fonction logique ET utilisée en masque

Un exemple de masquage:

Dans le code ASCII, on représente les chiffres comme suit:

0 = 0110000

1 = 0110001

2 = 0110010

3 = 0110011 etc.

Pour transcoder depuis le code ASCII vers le binaire, il faut réussir à masquer le début du mot: 011.... pour ne garder que la fin ...0000. Ceci est possible en utilisant le ET logique entre deux mots binaires.

122) Utilité

C'est assez souvent que l'on doit masquer une partie d'un mot, pour n'en utiliser qu'une partie.

123) Rappel sur la table de vérité du ET (ci-après à gauche)

On voit bien que si le masque est à 0, il y a masquage, sinon l'info est transmise intégralement.

e2	e1	S	Masque	Data	Result
0	0	0	0	0	0
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	1	1	1

Pour plus de précision, voir le cours de logique sur l'U.A.L..

13) Lecture d'une variable de type "entier" au clavier

Si l'utilisateur de l'ordinateur frappe le nombre "1992" au clavier, On reçoit la liste suivante de caractères ASCII: "31 39 39 32" exprimée en hexadécimal. Par masquage des 4 bits de poids fort par le filtre "0F" en hexa (ou 0000 1111 en binaire), on obtient la liste "1 9 9 2" en hexa(ou 0000 0001, 0000 1001, 0000 1001, 0000 0010 en binaire). Il faut ensuite tenir compte des poids de chacun des chiffres qui composent ce nombre, et multiplier "1" par 1000, "9" par 100, "9" par 10, et "2" par 1 et additionner le tout.

2) Clavier

21) Synoptique réel de fonctionnement d'un clavier

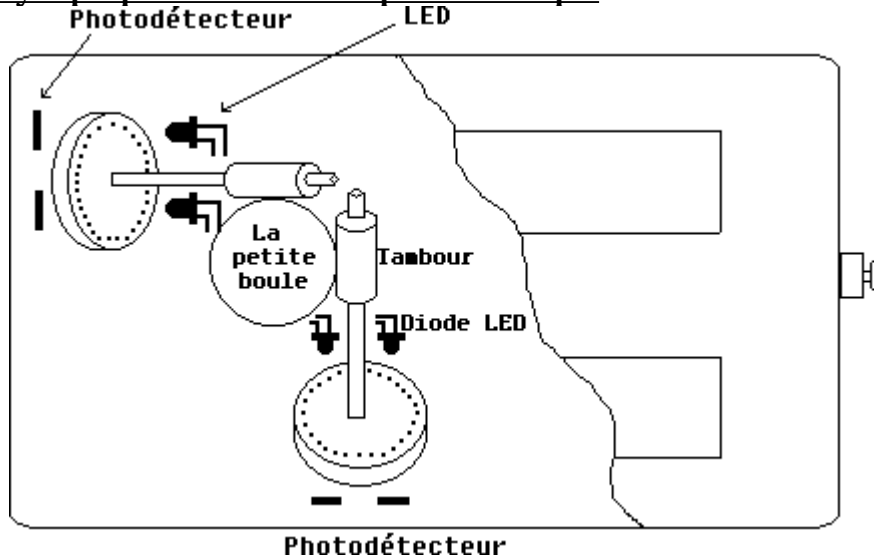
Il est bien compliqué, retenons qu'il comporte des mémoires internes, que chacun doit initialiser pour pouvoir taper un texte dans sa langue ... et arrêtons-nous là.

22) Le programme DOS keyb fr

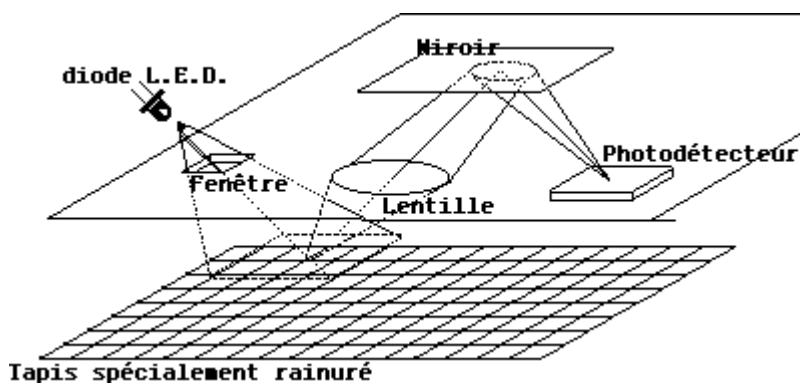
Lors de la mise sous tension, l'ordinateur exécute le programme "autoexec.bat", qui doit comporter l'instruction "keybfr" (dos 2) ou "keyb fr" (dos 3) pour initialiser le clavier.

3) Souris, track-ball et tablette graphique

31) Synoptique d'une souris opto-mécanique



32) Synoptique d'une souris optique



4) Crayon optique

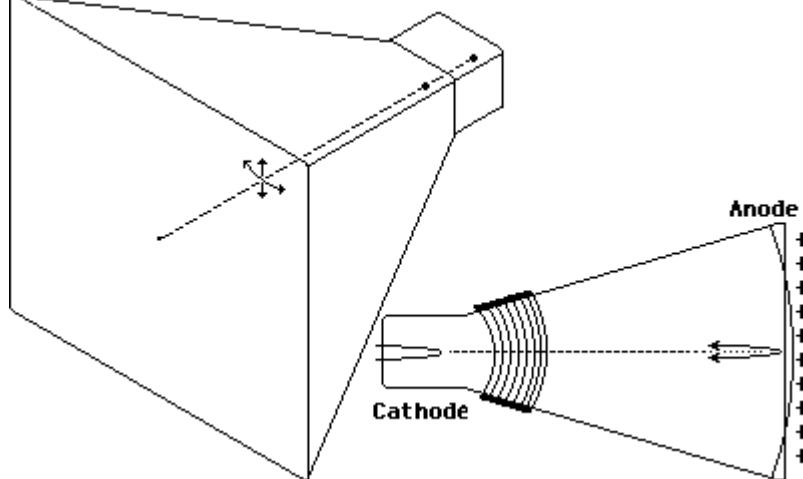
5) Digitaliseur-Scanner

Les périphériques de sortie

1) Ecran

11) Synoptique général d'un écran(mur de pixels)

Il faut bien être persuadé qu'un écran est un mur de pixels disposés selon une matrice de lignes et de colonnes. Dessiner une courbe sur cet écran se ramène forcément à allumer des points. (Voir le matricage de lettres par des points en graphisme).



12) Synoptique naïf de fonctionnement d'un écran: ci-avant à droite

13) Conversion binaire-> ASCII d'un caractère numérique en sortie

131) La fonction logique OU utilisée en incrustation

132) Un exemple d'incrustation:

Dans le code ASCII, on représente les chiffres comme suit:

0 = 0110000

1 = 0110001

2 = 0110010

3 = 0110011 etc.

Pour transcoder depuis le code binaire vers l'ASCII, il faut faire l'opération inverse, c'est à dire ajouter au début du mot: 011.... tout en gardant la fin ...0000 .

Ceci est possible en utilisant le OU logique entre deux mots binaires.

133) Utilité : Je l'ai utilisé de temps en temps.

134) Rappel sur la table de vérité du OU :

e2	e1	S	Patron	Data	Result	
0	0	0	0	0	0	Transmission
0	1	1	0	1	1	
1	0	1	1	0	1	Incrustation
1	1	1	1	1	1	

On voit que si on ajoute un motif, celui-ci est incrusté dans les données de sortie. Pour plus de précision, voir le cours de logique sur l'U.A.L..

14) Impression d'une variable de type "entier" à l'écran

Quand un ordinateur a pour tâche d'imprimer "1992" à l'écran, il dispose de la représentation binaire de ce nombre. Il faut d'abord le convertir en décimal au moyen de divisions successives par 10(10) cad "0000 1010(2)" on obtient alors la liste "1 9 9 2(16)". Pour la transformer en ASCII, il faut incruster le masque "0011 000".

15) Balayage vectorisé par opposition au balayage par points

Il faut garder présent à l'esprit qu'une image de 1000x1000 points contient 1 million de pixels et que, pour afficher cette image 25 fois par seconde, il faut transférer les données à une fréquence de 25 Méga pixels. Le fait de doubler la dimension de l'image induit un quadruplement de la fréquence de transfert. Pour palier cet inconvénient, et éviter l'effet d'escalier, les systèmes de haut niveau utilisent un balayage par segments: Le spot pour dessiner une courbe du point 1,1 au point 100, 100 va effectivement faire ce trajet. Pour ce type d'affichage, il faut utiliser des écrans à déflexion par champs électriques qui coûtent plus cher que les écrans de télévision classiques, car les plaques déflexrices doivent être prévues à l'intérieur du tube, alors que les bobines des écrans de télévision se tiennent à l'extérieur, et peuvent donc être rajoutées après.

16) Dessiner des courbes en mode graphique

Dans le cadre de logiciel comme "paint", le nombre de courbes potentiellement dessinables est tel qu'il est impossible de le prévoir. La génération des courbes doit donc se faire par soft au moyen d'algorithmes subtils.

17) Ecran utilisé en mode graphique

Si un écran est utilisé en mode graphique, l'ordinateur doit alors mémoriser la totalité des pixels. Exemple: Pour mémoriser un écran EGA de 320 par 200, il faut enregistrer 64 000 informations.

18) Ecran utilisé en mode texte

Contrairement au mode graphique, le mode texte permet une compression de l'information. Il suffit de mémoriser les caractères à afficher, et leur expansion en pixel se fait par hardware ou par soft au moment de l'affichage. C'est ainsi que même un écran VGA graphique se mémorise au moyen de 80x25 informations.

19) Technologie

191) Ecran Cathodique

Synoptique général (Cathode, anode, focalisation du pinceau)

Loi de Laplace (F B I)

Les trois couleurs élémentaires RVB

Les différentes normes d'écrans(

CGA Common Graphic Adaptator: 320x200

EGA Enhanced Graphic Adaptator: 640x350 (16 couleurs)

Hercule 720x348

VGA

SVGA Super VGA

TIGA Texas Instrument Graphic Adaptator

Les différents nombres de couleurs RVB 4,16,256,65536

192) Ecran non-cathodique

Passif(à cristaux liquides)

Nématique en hélice

Nouveau nématique

Biréfringence contrôlée

Actif

Led

Plasma

Electroluminescence

2) Imprimantes

21) Le spooler et le buffer d'imprimante

22) Les différents mode d'impression: NLQ, draft, postscript.

23) Divers types d'imprimantes

231) A impact(TTY, chaîne, marguerite)

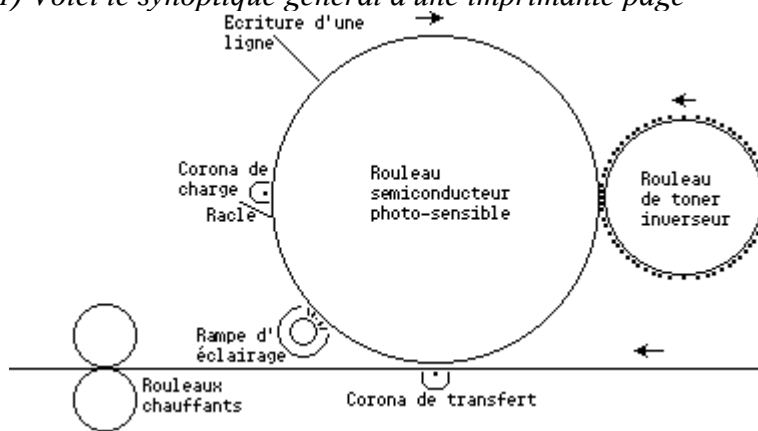
232) Matricielles(8,24)

233) A jet d'encre, bulle d'encre

234) Laser

24) Fonctionnement d'une imprimante page (laser)

241) Voici le synoptique général d'une imprimante page



242) Le tambour

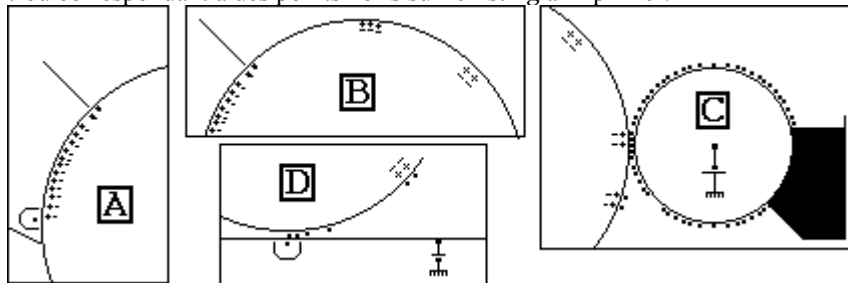
C'est un cylindre de 30 cm de long, de 10 cm de diamètre recouvert d'un semi-conducteur photosensible (sélénium)

243) Chargement électrostatique du tambour (Figure B)

Le corona de charge est un fil de tungstène qui court sur toute la longueur du cylindre, au ras de sa surface. Sur le dessin, comme il est vu de profil, il apparaît comme un point. Porté à un potentiel très négatif (8000 volt) il charge par influence la surface du cylindre, et fait apparaître des charges + sur son bord.

244) Écriture sur le tambour (Figure B)

Par un balayage point par point, on vient écrire une ligne du tambour. Là où la feuille à imprimer est noire, on passe sans rien faire. Si elle est blanche, on éclaire le tambour. Le point lumineux qui tombe sur la surface du sélénium la rend photoconductrice. Les paires électron-trou créés par l'influence du corona se recombinent. Une surface éclairée redevient électrostatiquement neutre. Sur le haut du tambour, après impression lumineuse, ne subsiste que quelques paires électron-trou correspondant à des points noirs sur le listing à imprimer.



245) Révélation de l'image par les grains de toner (Figure C)

Le toner est une encre solide qui se présente sous forme d'une poussière noire (d'encre!). Le rouleau encreur porte à sa surface des grains de toner chargés négativement. Quand le cylindre de sélénium passe devant eux, ils se précipitent vers les charges + et viennent se coller à sa surface.

246) Dépose des particules sur le papier (Figure D)

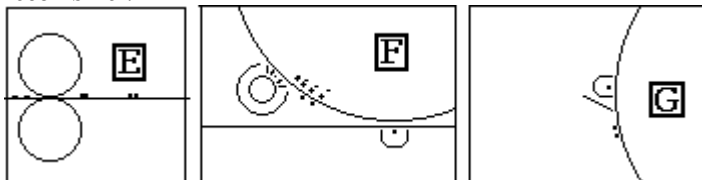
Le papier est chargé négativement par friction. Le corona de décharge est porté à un potentiel très négatif. Les particules de toner deviennent plus attirées par le corona de décharge que par les charges + du cylindre: Elles chutent sur le papier où elles se trouvent très bien et s'endorment.

247) Fixation de l'image (Figure E).

Le papier, vecteur de l'image passe entre deux galets presseurs porté à une température élevée. Les grains de toner sont spécialement étudiés pour fondre à la chaleur, et imbiber le papier machine. Notre feuille sort donc un peu chaude de l'imprimante page.

248) Remise à zéro pour le tour suivant (Figure F)

En éclairant toute la surface du tambour par une lampe, on le rend conducteur, et toutes les paires électron-trou peuvent se recombinaison.



Le semi-conducteur redevient électrostatiquement neutre.

249) Chasse aux récalcitrants (Figure G)

Fixée sur le corona de charge, juste avant de recommencer le cycle, une racle de plastique fait tomber les grains de toner récalcitrants.

3) Table traçante

31) Principe de commande en X,Y.

Les déplacements en x sont assurés par le bras. Les déplacement en Y proviennent d'un moteur qui commande l'avance ou le recul du papier.

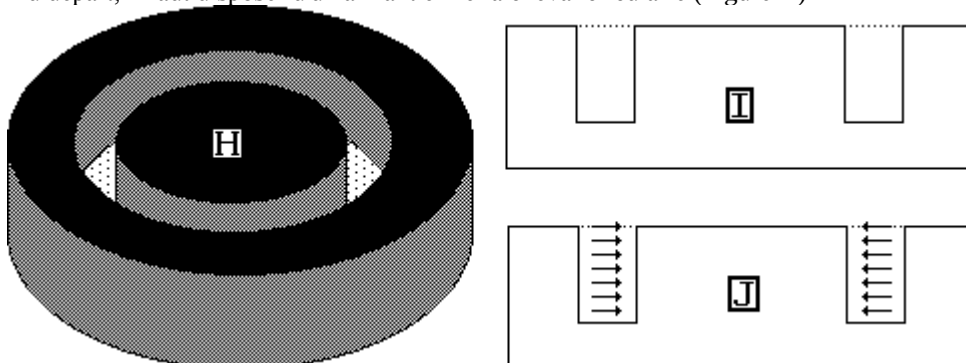
32) Utilisation d'une imprimante en table traçante

La tête d'écriture de l'imprimante sert pour les x. Si l'imprimante est capable d'aller en avant et en arrière on peut faire une vraie table traçante. Sinon on ne peut l'utiliser qu'en enregistreur.

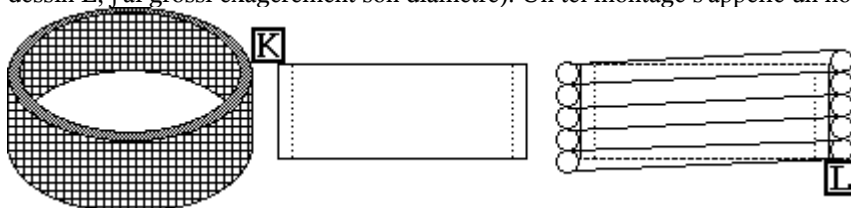
4) Haut-parleur

41)Principe de fonctionnement

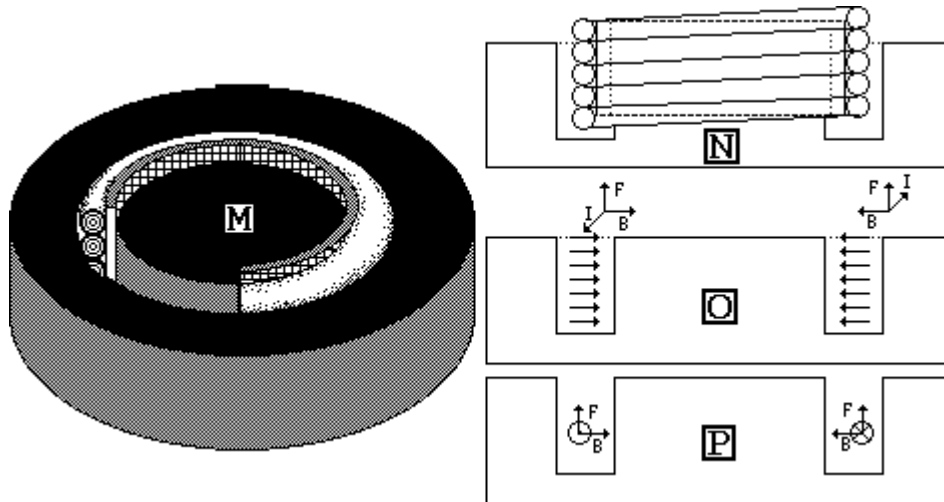
Au départ, il faut disposer d'un aimant en fer à cheval circulaire (Figure H)



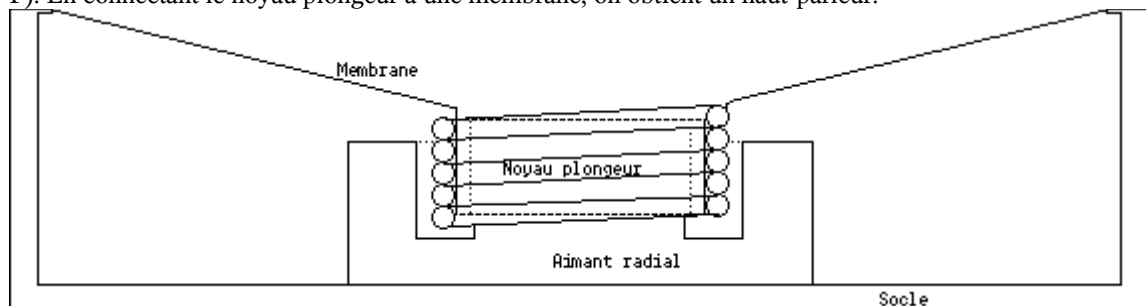
Vu de profil on obtient la coupe I. Dans un tel aimant les lignes de champ sont centripètes(elle vont vers le centre (Figure J). Ensuite il faut un cylindre de plastique où de carton (figure K), sur lequel on bobine un fil électrique (Pour faciliter le dessin L, j'ai grossi exagérément son diamètre). Un tel montage s'appelle un noyau plongeur.



Ensuite le noyau plongeur est introduit dans l'aimant, pour que ses spires baignent et coupent le champ magnétique. On obtient le montage décrit en M et N.



Sachant la loi de Laplace... (Figure O et P), on en déduit que les forces travaillent à faire bouger le noyau plongeur (Figure P). En connectant le noyau plongeur à une membrane, on obtient un haut-parleur.



42) Utilisation en tout ou rien (digital)

43) Utilisation en analogique

44) Quelques notions d'acoustique

441) Fréquence d'un signal

C'est le nombre de fois où celui-ci se reproduit identique à lui-même. C'est la fréquence d'un son qui détermine sa note. Si le son se reproduit plus souvent, sa note monte, sinon elle descend. Doubler la fréquence d'un son fait monter sa note d'une octave.

442) Amplitude d'un signal

C'est sa puissance, son volume.

443) Enveloppe

Diagramme de son amplitude en fonction du temps

443) Harmonique

Fréquence de vibration multiple de la fondamentale.

444) Timbre

Ce sont les harmoniques d'un signal qui détermine son timbre

Des mémoires de masse en entrée/sortie

Disquettes

Disque dur, disques amovibles et streamers

Mémoires à bulles.

Des mémoires de masse en entrée

Disque-optique et CD-ROM

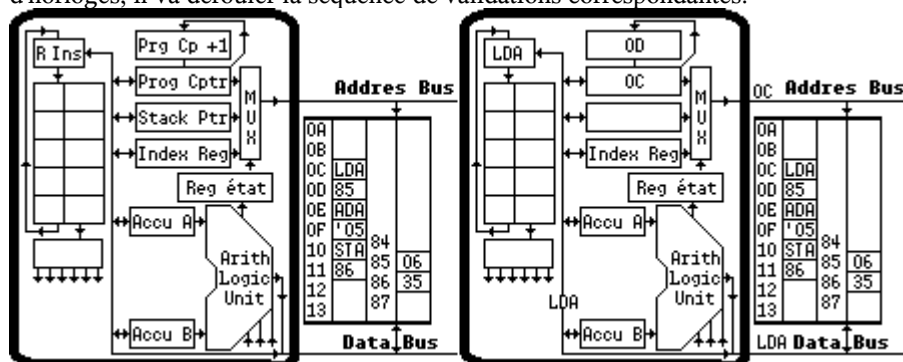
DAT

La machinerie interne du microprocesseur

Dans la mémoire, à partir de l'adresse 0C, un petit programme qui additionne deux nombres. A l'adresse 85 on trouve le premier de ces nombres. L'adresse 86 servira de lieu de rangement du résultat.

Au départ (ci-après à droite),

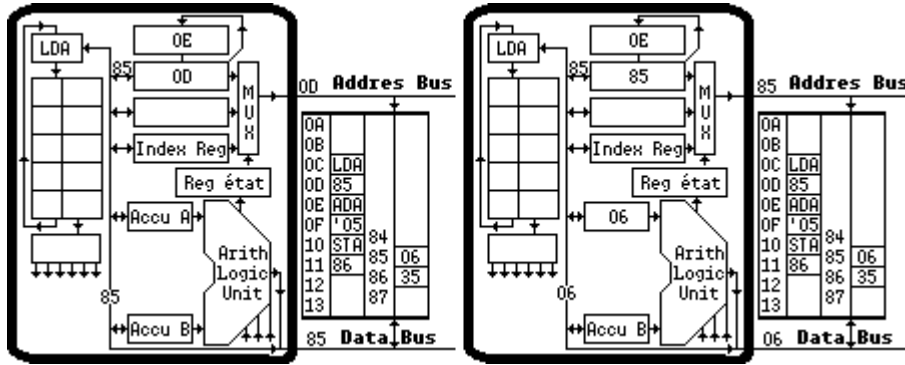
Tout est accroché au Programm Counter qui pointe sur l'adresse de début du programme:0C. Le multiplexeur permet au PC d'adresser la RAM, qui fournit en sortie le code hexadécimal du mnémonique 'LDA'. Le bus de données est aiguillé pour que cette information entre dans le μ p. Seul le registre de contrôle reçoit une impulsion de chargement. Maintenant il est initialisé à l'adresse du début du micro-programme correspondant à l'instruction 'LDA'. Au rythme des impulsions d'horloges, il va dérouler la séquence de validations correspondantes.



Séquence de l'instruction 'LDA'

Initialisation

(Ci-après à gauche). Le premier travail du séquençage de l'instruction 'LDA' (load accumulateur A), est de valider PC+1 vers PC et donner une impulsion de chargement au PC. Celui-ci se trouve incrémenté, et passe à 0D et adresse la mémoire, qui fournit le contenu 85. Le bus de données est validé vers l'intérieur du μ p, et une impulsion de chargement sur le compteur de programme le remplit avec la donnée 85, qui est maintenant interprétée comme une adresse.

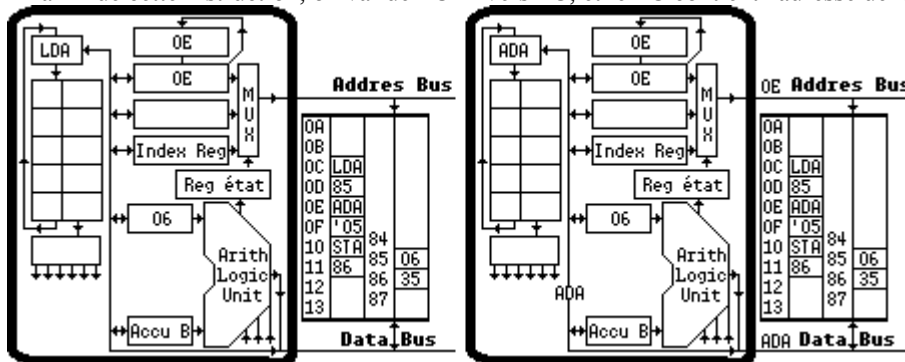


Charger le contenu de l'adresse 85

(Ci-avant à droite)Le contenu du PC adresse la mémoire qui fournit la donnée 06. Cette donnée est aiguillée vers le µp et chargée dans l'accumulateur A.

Passage à l'instruction suivante

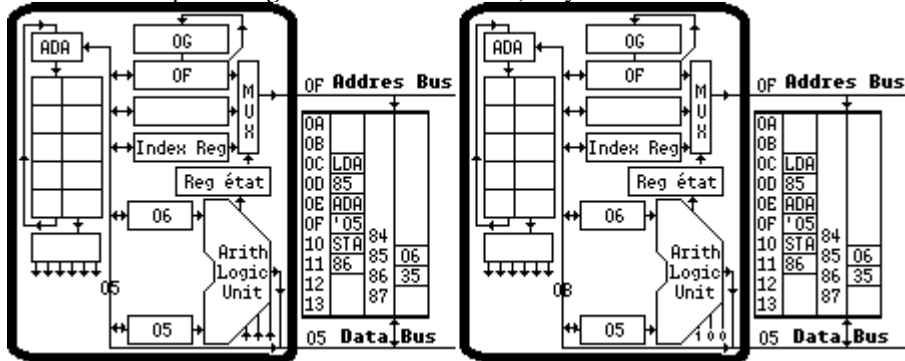
A la fin de cette instruction, on valide PC+1 vers PC, et le PC contient l'adresse de la prochaine instruction à exécuter.



En validant le PC vers le bus on adresse la mémoire pour récupérer dans le registre instruction le mnémonique 'ADA'.

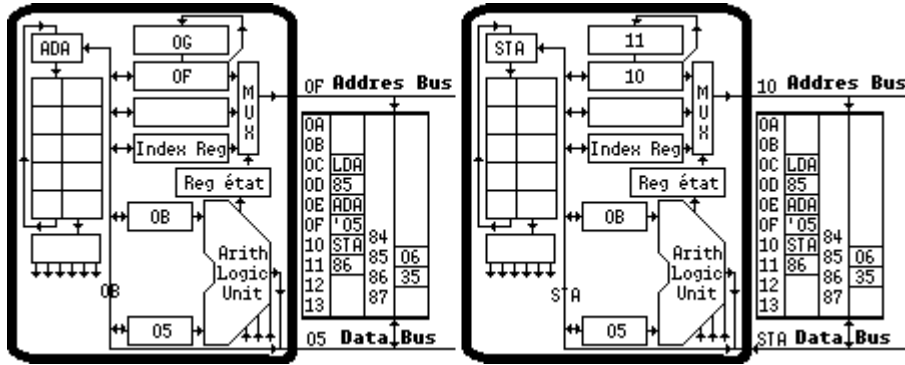
Exécution de l'instruction 'ADA'

En validant le PC vers le bus, on adresse la mémoire, et on récupère sur le bus de données le contenu de 0F qui est 05. En donnant un coup d'horloge sur l'accumulateur B, on y fait rentrer 05.



L'unité de contrôle fournit à l'U.A.L. la validation 100 qui la fait fonctionner en additionneur. Le résultat de l'addition des deux mots hexadécimaux 06 et 05 donne 0B qui est présenté sur tout le bus de données.

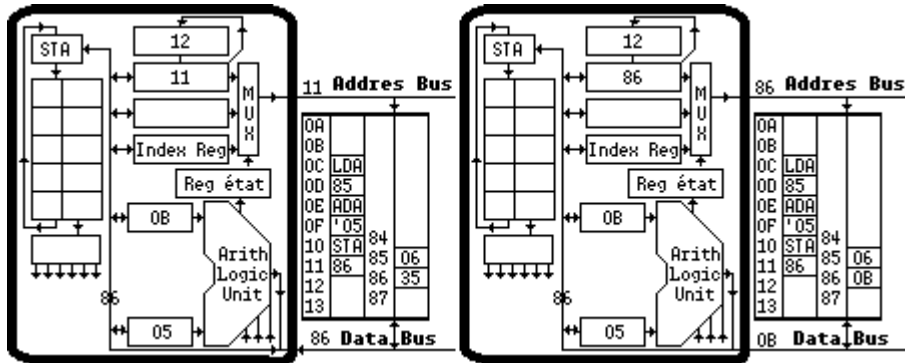
Un petit coup d'horloge sur l'accumulateur A permet de le charger avec le résultat : 0B. Ensuite, la suite de l'U.A.L. est désactivée et elle cesse de fournir une information au bus de données.



A la fin de l'instruction 'ADA', le contenu de PC+1 est validé vers PC, et adresse la mémoire qui retourne l'instruction suivante : 'STA'.

Exécution de l'instruction 'STA'

Puis le PC est incrémenté, et adresse la mémoire pour aller chercher l'adresse 86 où ranger le résultat 0B qui est dans l'accumulateur A.



Dans le PC on trouve 86 qui est l'adresse du lieu de rangement. On valide cette adresse vers la mémoire qui sera utilisée ici en écriture. On valide aussi l'accumulateur B vers le bus des données, et au top d'horloge, l'adresse 86 de la mémoire se charge de l'information 0B.

Suite

Ensuite PC+1 va dans PC et on va voir dans l'adresse 12 l'instruction suivante, et ça continue ... mais nous on arrête !