

UNIVERSITE D'ALGER  
ECOLE NATIONALE POLYTECHNIQUE

4/73

- DEPARTEMENT ELECTRONIQUE -

3 ex

المدرسة الوطنية للعلوم الهندسية

THES'E DE FIN D'ETUDES

ECOLE NATIONALE POLYTECHNIQUE  
BIBLIOTHEQUE

ETUDE ET REALISATION  
EN CIRCUITS INTEGRES DE  
L'INTEGRATEUR NUMERIQUE  
INCREMENTAL,  
ELEMENT DE BASE DE  
L'ANALYSEUR DIFFERENTIEL  
NUMERIQUE

المدرسة الوطنية للعلوم الهندسية  
ECOLE NATIONALE POLYTECHNIQUE  
BIBLIOTHEQUE

JURY

President: Mr. BENTCHICOU Maitre de Conference  
Assesseurs: Mme. G. MONDON Ingenieur  
: Mr. RAJAGOPALAN. Docteur Ingenieur  
Invites: M. Y. FLAMANT Professeur à la faculté des sciences  
D'alger  
M. A. BOUKRI Du Departement Electronique au cnr

Proposée par  
Mme: G. MONDON

Etudiée par  
Mr. S. MEDJOU

PROMOTION. 1973



*A mes parents.*

*C'est à partir d'opérations élémentaires  
que les problèmes scientifiques les plus  
compliqués sont résolus.*

*Ramenée à sa plus simple expression  
la Condition humaine sera alors  
meilleure.*

R e m e r c i e m e n t s

*Durant deux années Mme. G. Mondon a essayé de nous faire découvrir ce vaste domaine des machines informatiques,*

*Pour ses enseignements*

*Pour ses conseils*

*Pour son aide précieuse,*

*profondément je la remercie.*

*Je remercie aussi Mr. Boukri du Département Electronique au CNI,  
Mr. Dupin , Responsable T.V., Mr. Driouche, Technicien.*

# S O M M A I R E

## 0 . Introduction

### 1<sup>ere</sup> Partie: Etude théorique

#### 1 . Calcul incrémental

1.1 Principe du calcul incrémental	
1.1.1 Représentation numérique.....	4
1.1.2 Représentation numérique incrémentale....	5
1.1.3 Représentation numérique incrémentale à éléments unitaires.....	5
1.2 L'opération d'intégration	
1.2.1 La variable d'intégration est le temps... 6	6
1.2.2 La variable d'intégration est une fonction du temps : -problème purement numérique -problème en temps réel .....	7
1.3 Algorithmes de calcul.....	
1.3.1 La variable est le temps.....	9
1.3.2 La variable est une fonction du temps... 10	10
1.4 Echelle de l'intégrale .....	12

#### 2 . L'intégrateur Numérique incrémental

2.1 Condition suffisante pour obtenir des incréments unitaires.....	14
2.2 Organisation de calcul .....	14
2.3 Exemple d'intégration par rapport au temps....	17
2.4 Intégration par rapport à une variable autre que le temps.....	19
2.5 Symbole.....	20

#### 3 . L'analyseur différentiel Numérique ( DDA)

3.1 Représentation des nombres dans le DDA	
3.1.1. Représentation à incréments binaires....	21
3.1.2. Représentation à incréments ternaires... 22	22
3.1.3. Exemple de fonctionnement dans le système à incréments binaires.....	23

3.1.4. Exemple de fonctionnement dans le système à incréments ternaires.....	24
3.2 Programmation.....	26
3.3 Mise à l'échelle d'un problème résolu sur DDA .	29
3.4 Erreurs en calcul par DDA .....	29
3.5 Réalisation en MO S de l'intégrateur .....	30

2° Partie: Réalisation Pratique de l'intégrateur  
Numérique incrémental à Circuits  
Intégrés.

1. <u>Schéma synoptique</u> .....	31
2. <u>Notes sur les circuits intégrés fondamentaux constituant l'intégrateur.</u>	
2.1 L'additionneur - soustracteur .....	32
2.2 Le registre mémoire SN 7475 N .....	33
2.3 Le comparateur binaire SN 7485 N .....	34
2.4 Le compteur Décompteur SN 74193 N.....	34
2.5 Flip-Flop JK SN 7476 N .....	35
2.6 Flip-Flop JK SN 7473 N .....	35
3. <u>Schéma de l'unité principale à circuits intégrés et fonctionnement.</u>	
3.1 Schéma .....	35
3.2 Fonctionnement.....	35
4. <u>Le séquenceur</u> .....	38
5. <u>L'horloge</u> .....	39
6. <u>Système d'entrée sortie</u> .....	40
7. <u>Schéma général</u> .....	41
8. <u>Circuits intégrés utilisés</u> .....	41
9. <u>Simulation sur 1130</u> .....	
9.1 Organigramme.....	41
9.2 Programmation et commentaire.....	43

Conclusion

Bibliographie

-- INTRODUCTION --

Les calculateurices analogiques sont des machines simples, rapides, bon marché, de programmation et d'emploi aisés. Elles ont cependant deux défauts inhérents à leur nature :

- Elles n'intègrent que par rapport à une variable indépendante unique, homogène au temps et par conséquent leur emploi est limité à des problèmes linéaires temporels. Pour des problèmes plus compliqués, il faut, au minimum adjoindre des multiplieurs. Ceux-ci restent chers et délicats, ils sont souvent la cause du mauvais fonctionnement de la machine. Pour résoudre des problèmes plus généraux, il faut des générateurs de fonctions et d'autres organes non linéaires.

- Elles sont relativement peu précises. Les amplificateurs ont un gain limité. De plus des bruits de différentes origines : instabilité des composants, enregistreurs, limitent la précision d'une façon irrémédiable. Au contraire les calculatrices numériques universelles sont précises mais d'un certain point de vue, lentes. La précision est limitée par l'encombrement, le prix de revient, le temps de calcul. Elle est indépendante des bruits des circuits lorsque ceux-ci sont bien étudiés. Le seul "bruit" à envisager est celui des erreurs d'arrondi. Les calculatrices numériques résolvent des équations différentielles au prix d'un travail important d'analyse numérique et d'une programmation importante, ce qui revient très cher en temps et en argent.

- Pour certains problèmes la précision des calculatrices analogiques est insuffisante et le temps d'opération des calculatrices numériques universelles trop long. C'est actuellement le cas, par exemple, des calculatrices pour simulateurs de vol qui doivent travailler avec une grande précision en "temps réel". Une combinaison des qualités de souplesse et rapidité de l'une avec la précision de l'autre semble être le meilleur compromis. Une telle machine mixte est dénommée Analyseur différentiel Numérique\*. En effet - comme dans le calculateur numérique l'information est digitalisée et par conséquent la précision de cette représentation des grandeurs peut être aussi élevée que l'on veut : il suffit pour cela d'augmenter le nombre de bits pris en considération.

Comme dans le calculateur analogique l'opération de base est l'intégration. Il s'emploie dans une certaine mesure de la même manière en réunissant par un câblage manuel ou télécommandé, divers blocs qui figurent les opérateurs élémentaires de l'analyse. Par contre la variable indépendante ne sera pas uniquement le temps.

Considérons par ailleurs un calculateur numérique suivant, en temps réel, un processus physique quelconque : le calculateur reçoit les informations en provenance de plusieurs capteurs, les traite suivant un modèle mathématique adapté à la machine par le programmeur, et agit sur le processus de sortie par divers organes. Mais à chaque itération le calcul se fait sur la totalité de la représentation numérique des grandeurs, alors que dans le DDA on considère que les grandeurs utilisées varient de façon continue et que par conséquent il suffit d'effectuer, à chaque itération, un calcul de variations. On va voir que cette approche permet dans bien des cas une amélioration importante du rapport performance/Prix.

Pour illustrer ce point, considérons le calcul de  $\sin \theta$  et  $\cos \theta$  avec 13 bits significatifs. La méthode utilisée dans le calculateur universel consiste à considérer les trois premiers termes des développements limités ce qui est suffisant pour obtenir 13 bits significatifs, en supposant  $0 \leq \theta \leq \frac{\pi}{4}$ .

$$\sin \theta = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \dots (-1)^p \frac{\theta^{2p+1}}{(2p+1)!} \approx \theta \left[ \left( \frac{1}{5!} \theta^2 - \frac{1}{3!} \theta^2 + 1 \right) \right]$$

$$\cos \theta = 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \dots (-1)^p \frac{\theta^{2p}}{2p!} \approx 1 + \left( \frac{\theta^2}{4!} - \frac{1}{2!} \right) \theta$$

On constate que la première expression nécessite 4 multiplications et 2 additions la seconde 3 multiplications et 2 additions. Si l'on considère qu'une Multiplication équivaut en temps à 13 additions on arrive à un équivalent de  $7 \times 13 + 2 \cdot 2 = 95$  additions

Si l'on suppose maintenant que l'on connaît  $\sin \theta_0$  et  $\cos \theta_0$  et que l'on désire avoir  $\sin \theta$  et  $\cos \theta$  avec  $\theta = \theta_0 + \Delta \theta$  on obtient :

$$\sin \theta = \sin(\theta_0 + \Delta \theta) = \sin \theta_0 \cos \Delta \theta + \cos \theta_0 \sin \Delta \theta \approx \sin \theta_0 + \Delta \theta \cos \theta_0$$

$$\cos \theta \approx \cos \theta_0 - \sin \theta_0 \cdot \Delta \theta$$

En supposant  $\Delta\theta \leq 2^{-13}$  rad. En prenant  $\Delta\theta = 2^{-13}$  ce qui est le poids du dernier bit significatif, on constate que le calcul de  $\sin\theta$  et  $\cos\theta$  se ramène à 2 additions puisqu'une multiplication par une puissance de 2 est une addition. Cet avantage de 47,5 à 1 en faveur du DDA ne correspond naturellement à un avantage réel que lorsqu'on a à calculer ( $\sin\theta$ ,  $\cos\theta$ ) de façon permanente ; lorsqu'on doit faire le calcul pour une valeur isolée de  $\theta$ , le calculateur universel garde toute sa valeur puisque le DDA peut avoir à faire  $\frac{\pi/4}{2^{-13}} = \pi \times 2^{11} = 6450$  itérations soit 12900 additions. Dans ce cas l'avantage est de  $\frac{12900}{95} = 136$  à 1 en faveur du calculateur universel.

Cet exemple a donc montré que le DDA peut procurer des avantages importants mais uniquement si on l'emploie à la tâche pour laquelle il a été conçu, à savoir le calcul de variations ou calcul incrémental.

Toutes les applications relevant de ce type de calcul (et elles sont, au moins potentiellement, innombrables) peuvent donc avoir à gagner de l'utilisation du DDA objet de notre étude et dont la conception ne peut se faire qu'après avoir bien cerné cette méthode de calcul si particulière qu'est le calcul incrémental.

Grâce aux énormes possibilités offertes par les circuits intégrés la réalisation de l'élément de base du DDA devient relativement aisée.

La première partie de notre travail sera consacrée à une étude théorique à travers laquelle le calcul incrémental nous permettra de définir l'opérateur essentiel de ce calculateur, l'intégrateur Numérique incrémental dont la réalisation pratique constituera la seconde partie.

-----

CHAPITRE 1.

CALCUL INCREMENTAL

Dans un calculateur incremental l'information que l'on traite est mesurée par les variations des grandeurs plutôt que par ces grandeurs elles-mêmes.

1.1. Principe du calcul incremental:

Dans les calculateurs numériques les grandeurs qui interviennent se présentent sous forme de nombres contenus dans des registres. L'élément de base d'un registre est la cellule de mémoire. Il s'agit d'un organe pouvant présenter plusieurs états stables (Tore, bascule,...) auxquels on attache arbitrairement une valeur numérique (chiffre). En groupant de telles cellules il est possible de représenter des nombres. Avec n cellule de poids  $2^0, 2^1, \dots, 2^{n-1}$  on peut représenter tous les entiers de 0 à  $2^n - 1$ . On définit la capacité N d'un registre comme étant le plus grand nombre qu'il est possible de représenter dans ce registre. Il ne faut pas dépasser la capacité de ce registre. On définit alors une échelle E. Considérons l'évolution dans le temps d'une fonction Y(t). Il importe donc de préjuger de l'excursion maximale de cette fonction,  $Y_{max}$ .

On choisit l'échelle E en imposant  $|Y|_{max} \cdot E \leq N$  (1)

Tels qu'ils sont représentés les registres ne peuvent contenir que des nombres entiers. Il y correspond des valeurs de la grandeur représentée qui sont espacées d'une quantité Q appelée Quantum telle que :

$$Q = \frac{1}{E} \quad (2)$$

La vitesse de calcul est limitée, les résultats des opérations demandent un certain temps. On utilisera les résultats à des moments bien déterminés où l'on est sûr que l'opération est terminée. Cette façon de procéder entraîne un échantillonnage dans le temps. Le plus souvent celui-ci est de période T. Celle ci est liée à la vitesse du calculateur.

1.1.1. Représentation numérique:

La représentation numérique de y(t) s'obtient dès lors en prenant à chaque instant d'échantillonnage le multiple entier de Q approchant au mieux selon un critère donné (excès ou défaut) la grandeur réelle. Dans le cas de la figure 1-1, c'est par défaut. Comme ces valeurs ne peuvent être utilisées qu'aux instants d'échantillonnage on a une représentation sous forme de saies.

Remarque: L'erreur ne dépasse jamais un quantum  $|Y(t_n) - Y^*(t_n)| \leq Q$  (3)

1.1.2. Représentation numérique incrémentale: L'information est mesurée par les variations ou 'incréments des grandeurs plutôt que par ces grandeurs.

Définition de l'incrément  $\Delta y^*$  : il est défini aux instants d'échantillonnage par :

$y^*(t_n) = y^*(t_{n-1}) + \Delta y^*(t_n)$  (4)  $\Delta y^*(t_n)$  constitue la représentation numérique incrémentale de  $y(t)$  - Voir figure :1-1

Nous avons :  $\Delta y^*(t_n) = y^*(t_n) - y^*(t_{n-1})$  (5)

Relations entre les modes de représentation.

$y^*(t_n) = y^*(t_0) + \sum_{j=1}^n \Delta y^*(t_j)$  (6)

Le passage à la représentation numérique complète  $y^*(t_n)$  est aisé il suffit de faire la somme algébrique des incréments successifs à partir de  $y^*(t_0)$ .

1.1.3. Représentation numérique incrémentale à éléments unitaires:

\* Définition: Les incréments comportaient généralement un certain nombre de quantités. Dans la R.N.I unitaire on impose que les incréments ne soient formés que d'un seul quantum valant donc + Q ou - Q. On conçoit qu'une telle contrainte impose certaines conditions dans le choix de Q et T.

\* Condition suffisante de compatibilité: La variation maximum d'une grandeur y représentée par ce mode à incréments unitaires est par définition égale à Q pour une période d'échantillonnage

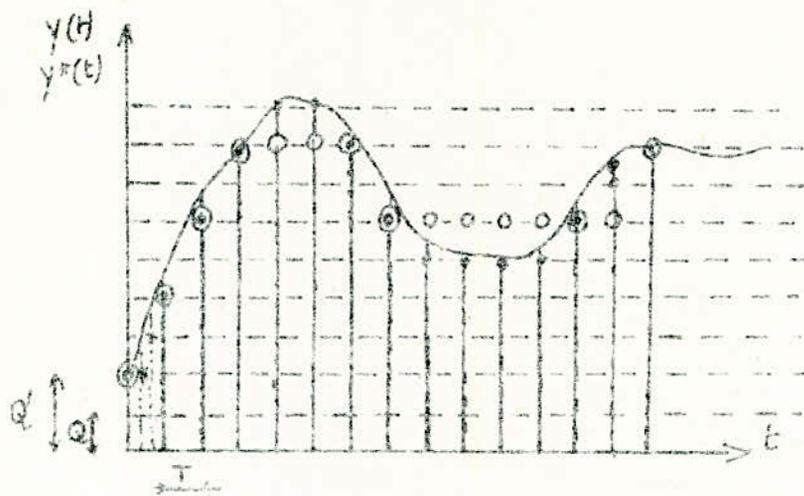
$\left| \frac{dy}{dt} \right|_{max} \leq \frac{Q}{T}$  condition suffisante on peut agir sur Q et T

Action sur Q: En augmentant Q il est possible d'obtenir des incréments unitaires FIG.1.

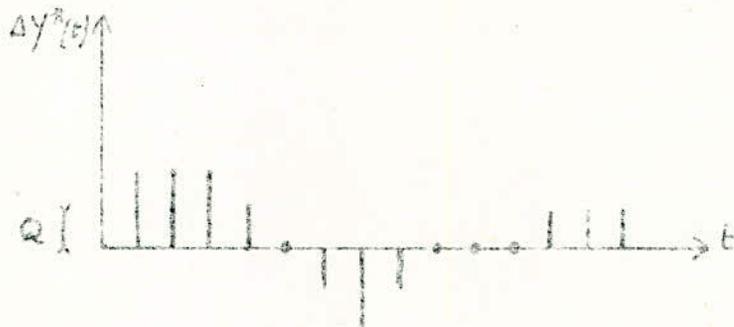
Action sur T: En diminuant T on peut obtenir une représentation à incréments unitaires (voir fig.1.1.d.)

Rappelons que T dépend de la vitesse du calculateur.

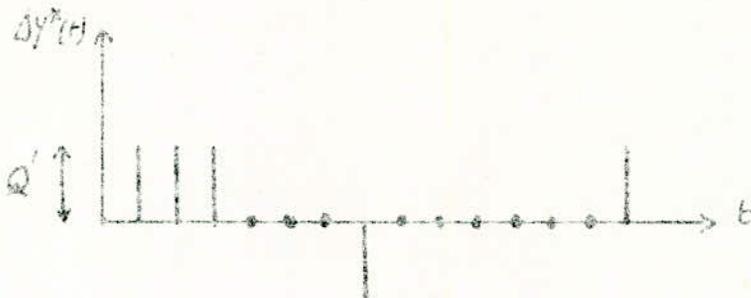
\* Conclusions: Dans la plupart des cas la précision est imposée (Q) on ne pourra alors recourir qu'à l'action sur T pour satisfaire la condition de compatibilité. Il s'ensuit que les performances des calculateurs numériques incrémentaux à I.U. sont très étroitement liées à la vitesse de calcul. Heureusement du fait de l'utilisation d'I.U. la structure des calculateurs se simplifie considérablement et les temps d'opération seront très courts.



a)



b)



c)



d)

Figure 1.1 Représentation Numérique et incrémentale

- cas des incréments non unitaires
- cas des incréments unitaires.

1.2. L'opération d'intégration:

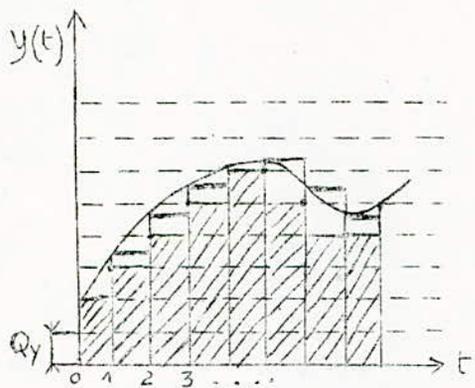
Etant donné le rôle particulier joué par le temps dans les calculateurs incrémentaux on distingue deux genres d'intégrales:

- La variable d'intégration est le temps
- La variable d'intégration est une fonction du temps.

1.2.1. La variable d'intégration est le temps: Pour calculer numériquement l'intégrale on l'approche généralement par une somme. Nous nous limiterons à la méthode des "rectangles" (de Riemann) considérons la fonction  $y(t)$  on se propose de déterminer  $I(t) = \int_0^t y(t) dt$  on rappelle que  $I(t)$  est l'aire entre  $y(t)$  et l'axe  $t$ .

Approcher l'intégrale  $\leftrightarrow$  approcher l'aire. Seules les valeurs de l'intégrale aux instants d'échantillonnage sont à considérer.

$$I(t_n) = \int_0^{t_n} y(t) dt.$$



- Sans quantification d'amplitude
- /// quantification d'amplitude

. A chaque  $\Delta t = T \rightarrow$  un accroissement d'aire  $T \times y(t_{i-1})$

On aura  $I(t_n) \approx \sum_{i=1}^n T y_{i-1}$  où  $y_{i-1} = y(t_{i-1})$

$$I(t_n) \approx T \sum_1^u y_{i-1}$$

. Avec la quantification d'amplitude

$$I(t_n) \approx T \sum_1^n y_{i-1}^*$$

Posons  $S_n^* = \sum_1^n y_{i-1}^* \Rightarrow I(t_n) \approx T \cdot S_n^*$  donc  $S_n^*$  constitue à un facteur près une valeur approchée de l'intégrale  $\int_0^{t_n} y(t) dt$

Dans l'intégrateur numérique que nous étudierons c'est  $S_n^*$  que l'on calculera.

1.2.2. La variable d'intégration est une fonction du temps: On distingue alors

2 problèmes:

- Le problème purement numérique
- Le problème en temps réel.

\* Problème numérique: On désire connaître la valeur numérique  $I(a)$  de l'intégrale

$$\int_a^b y(X) dX$$

Etant donné la limitation de vitesse des opérateurs numériques on devra

établir le calcul de cette intégrale dans le temps. La méthode générale consiste à assimiler la variable d'intégration  $X$  au temps en posant  $X = Ct$

$$I(a) = C \int_0^{a/c} y(Ct) dt$$

on est ramené au problème où la variable d'intégration

est le temps.

\* Problèmes traités en temps réel (voir figure 12.) On s'intéresse à  $\int_0^X y(X) dX$

$X$  est une grandeur physique liée à un phénomène que l'on observe et sur lequel on agit

Tout phénomène physique déroulant dans le temps est une  $f(t)$  et  $Y$  est une  $f(t)$  travers  $X$ . On peut s'intéresser à l'intégrale de Stieljes  $\int_0^X Y(t) dX(t)$ ,  $y(t)$  et

$X(t)$  peuvent être indépendants. Le cas de l'intégrale de Stieljes est plus général.

$$dX(t) = \frac{dX(t)}{dt} \cdot dt \Rightarrow I(t) = \int_0^X y(t) \frac{dX(t)}{dt} \cdot dt$$

on est encore ramené au pro

où la variable est le temps et on pose:

$$K(t) = y(t) \frac{dX(t)}{dt} \Rightarrow I(t_n) \approx T \cdot \sum_{i=1}^n K_i - 1$$

On ne dispose pour le calcul de l'intégrale que de  $y(t)$  et de  $X(t)$  ou si l'on fait usage d'un calculateur numérique de  $y^*(t)$  et de  $X^*(t)$ . Il nous faut cependant connaître la valeur de  $K(t)$  à chaque instant d'échantillonnage.

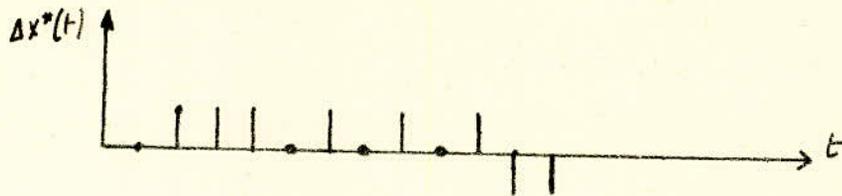
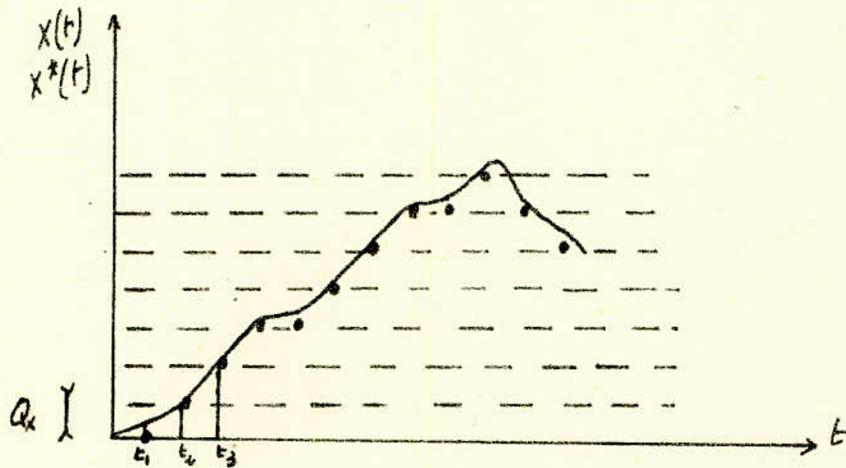
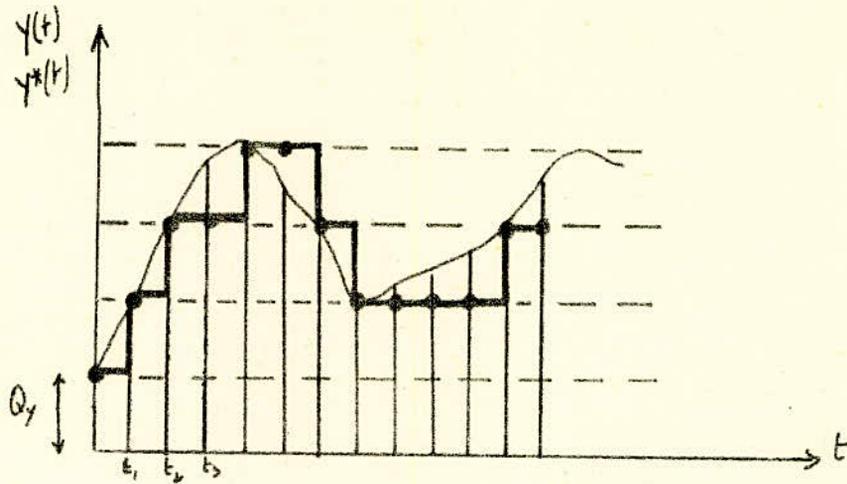


Figure 1-2: Representation de  $Y(t)$ ;  $Y^*(t)$ ;  $X(t)$ ;  $X^*(t)$  et  $\Delta X^*(t)$

- La variable d'intégration  $X$  est une fonction du temps  
cas des problèmes traités en temps réel •

à  $t = t_i$   $K_i$  peut être obtenu par les approximations suivantes :

$$\left. \frac{dX(t)}{dt} \right|_{t=t_i} \approx \frac{\Delta X_i^*}{T} \quad \Delta X_i^* \text{ est l'incrément de } X^* \text{ pour } t = t_i \Rightarrow$$

$$K_i \approx y_{i-1}^* \cdot \frac{\Delta X_i^*}{T} \Rightarrow I(t_n) \approx T \cdot \sum_1^n y_{i-1}^* \cdot \frac{\Delta X_{i-1}^*}{T}$$

$$I(t_n) \approx \sum_{i=1}^n y_{i-1}^* \cdot \Delta X_{i-1}^*$$

Cette approximation est donc une somme de produits.

Si les incréments  $\Delta X^*$  sont unitaires cette somme de produits se ramène à une somme. En multipliant et en divisant le 2ème membre par  $Q_x \Rightarrow$

$$I(t_n) \approx Q_x \sum_{i=1}^n y_{i-1}^* \cdot \frac{\Delta X_{i-1}^*}{Q_x} \quad \text{posons} \quad S_n^{**} = \sum_{i=1}^n y_{i-1}^* \cdot \frac{\Delta X_{i-1}^*}{Q_x}$$

\* rappelle que la variable d'intégration est une fonction  $X(t)$

$$I(t_n) \approx Q_x \cdot S_n^{**} \quad \text{comme } \Delta X_{i-1}^* \text{ est unitaire donc } \frac{\Delta X_{i-1}^*}{Q_x} \text{ vaudra } +1$$

- 1 ou 0.

Les valeurs numériques  $y_{i-1}^*$  ne peuvent donc être multipliés que par un facteur unitaire ou nul et la somme des produits approchant l'intégrale se ramène à une simple somme dans le cas des incréments unitaires. Dans l'intégrateur numérique (plus loin) c'est cette somme que l'on calculera  $S_n^{**}$

### 1.3. Algorithmes de calcul.

Nous avons dans le schéma fonctionnel un registre Y dans lequel on placera les valeurs successives de  $Y^*$  est un registre S (registre de sortie) où apparaitront les valeurs la somme  $S^*$  ou  $S^{**}$  selon le cas. On peut faire une addition ( $S + Y$ ) ou ( $S - Y$ ) entre les registres S et Y. On peut transférer le résultat de cette opération dans S. Noter que si l'on commence les calculs à un instant d'échantillonnage le résultat ne sera vraiment disponible, dans le registre S, qu'à l'instant d'échantillonnage suivant.

1.3.1. La variable est le temps: On désire qu'à chaque instant d'échantillonnage le contenu du registre S soit égal à la valeur de  $S^*$ , relative au même instant d'échantillonnage si  $[S]_n$  est le contenu du registre S à un instant  $t_n$  quelconque il faut que l'on ait :

$$[S]_n = S_n^* = \sum_1^u y_{i-1}^*$$

$$t_1 \rightarrow S_1^* = y_0^*$$

$$t_2 \rightarrow S_2^* = y_0^* + y_1^*$$

$$t_n \rightarrow S_n^* = \sum_1^n y_{i-1}^*$$

De plus considérons que le contenu de Y soit égal à chaque instant d'échantillonnage à la valeur de  $Y^*$  à cet instant, c'est à dire que l'on a  $\forall n \rightarrow [Y]_n = Y_n^*$

A l'instant  $t_0$  on dispose de la valeur  $y_0^*$  dans le registre Y. A ce même instant le registre S est vide. Puisque l'on désire disposer à  $t_1$  de  $y_0^*$  dans S, on commence les opérations de transfert et d'addition dès l'instant  $t_0$ ; si on avait commencé à  $t_1$ , le résultat n'aurait été disponible qu'à l'instant  $t_2$ .

a  $t = t_1$  on a dans le registre  $[Y] \rightarrow y_1^*$  et  $y_0^*$  dans  $[S]$ ; par addition et transfert

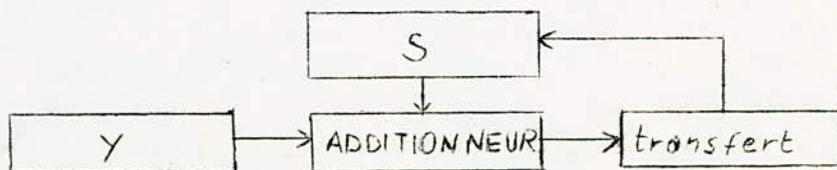
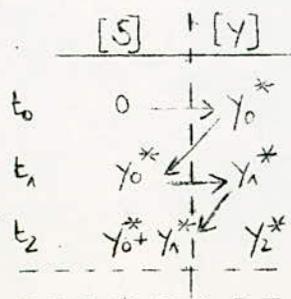
Il faut que la période d'échantillonnage soit supérieure au temps addition-transfert.

On a à l'instant  $t_2 \rightarrow [S_2] = y_0^* + y_1^* \Rightarrow [S_n] = S_n^*$

Nous pouvons donc résumer l'algorithme de calcul comme suit :

\* A chaque instant d'échantillonnage y compris l'instant  $t_0$  on commence par additionner le contenu du registre Y au contenu de S puis on transfère le résultat dans le registre S afin de former  $S^*$  pour l'instant d'échantillonnage suivant.

$$A \ t = t_n \rightarrow [S]_n + [Y]_n = [S]_{n+1} = S_{n+1}^*$$



1.3.2. La variable d'intégration est une fonction X du temps:

Dans ce cas <sup>c'est</sup> la somme  $S^{**}$  définie plus haut que l'on se propose de former dans le registre S

$$\Rightarrow \forall n \quad [S_n] = S_n^{**} = \sum_{i=1}^n y_{i-1}^* \frac{\Delta X_{i-1}^*}{Q_x}$$

Par exemple à

$$t = t_1 \quad \longrightarrow \quad S_1^{**} = y_0^* \frac{\Delta X_0^*}{Q_x}$$

$$t = t_2 \quad S_2^{**} = y_0^* \frac{\Delta X_0^*}{Q_x} + y_1^* \frac{\Delta X_1^*}{Q_x}$$

Dans le cas de la figure 1-3.

$$[S_1] = 0 \quad \text{car} \quad \Delta X_0^* = 0$$

$$[S_2] = 0 \quad \text{car} \quad \Delta X_1^* = 0$$

$$[S_3] = 0 + 0 + y_2^* \frac{\Delta X_2^*}{Q_x} = y_2^* \quad \text{car} \quad \Delta X_2^* = Q_x$$

$$[S_4] = y_2^* + y_3^* \quad \text{car} \quad \Delta X_3^* = Q_x$$

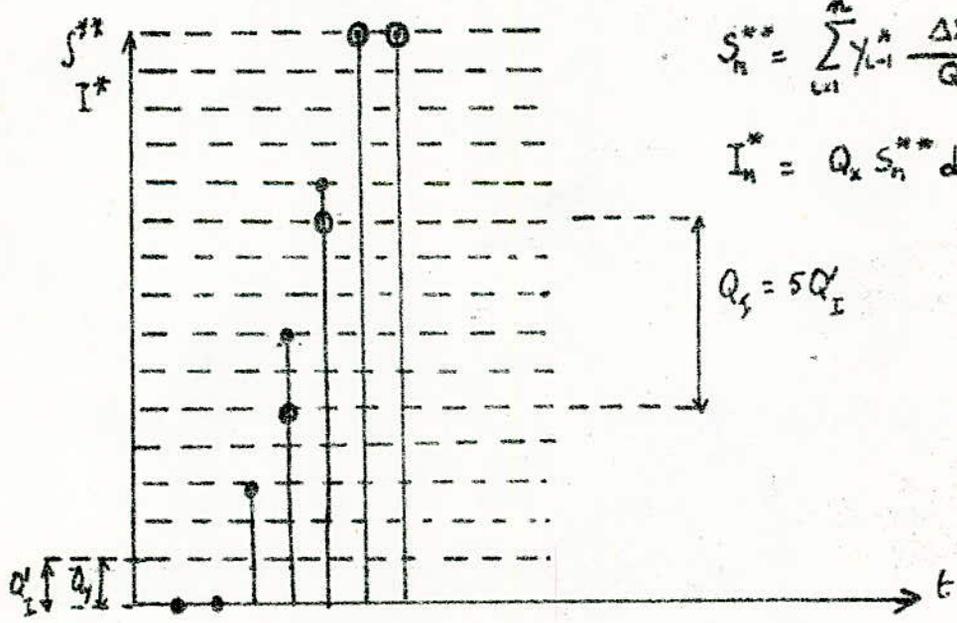
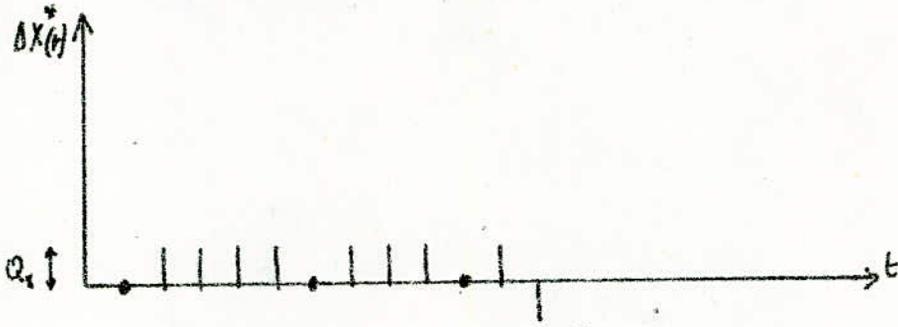
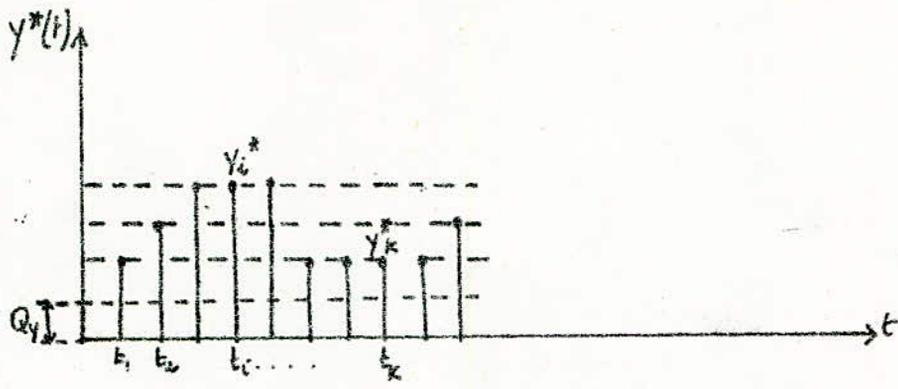
Au départ le registre est vide tant que  $X^*$  ne varie pas le registre S doit rester vide

Dès que l'on constate la présence d'un incrément (unitaire) de  $X^*$  [ $\Delta X^* = Q_x$ ]

On commence les opérations d'addition et de transfert :

à  $t_2$   $\Delta X_2^* = Q_x$  donc à  $t_2$  on commence add. et transfert

à  $t = t_3$   $S_3^{**} = y_2^*$  figure bien dans le registre S comme on le souhaite.



$$S_n^{**} = \sum_{k=1}^n y_{k-1}^* \frac{\Delta x_{k-1}^{**}}{Q_x}$$

$$I_n^* = Q_x S_n^{**} \text{ de quatem } Q_I^*$$

$$Q_s = 5Q_I$$

Figure 13

• Aspect de  $y^*$ ,  $\Delta x^{**}$  et  $s^{**}$  telles qu'elles apparaissent dans le calculateur au cours  $t$  du temps.

à cet instant ( $t = t_3$ ) nouvel incrément  $\Delta X_3^* = Q_x$   $[S_3] = y_2^*$  et

$$[Y]_3 = y_3^*$$

addition et transfert à  $t = t_4$   $[S]_4 = y_2^* + y_3^* = S_4^{**}$

D'une manière générale on aura  $[S]_n = S_n^{**}$ .

L'Algorithme de calcul est le suivant :

A chaque instant d'échantillonnage on examine s'il existe ou non un incrément de  $X^*$

$$1) \exists \Delta X_i^* = Q_x > 0 \quad t = t_i$$

On commence à additionner le contenu de registre Y au contenu du registre S puis on transfère le résultat dans S afin de former  $S^{**}$  pour l'instant d'échantillonnage suivant  $[S]_i + [y]_i = [S]_{i+1} = S_{i+1}^{**}$

$$2) \Delta X_i^* = 0 \quad \text{à } t = t_j$$

Il importe que le contenu de S soit inchangé jusqu'au temps suivant

$$[S]_j = [S]_{j+1} = S_{j+1}^{**}$$

$$3) \exists \Delta X_i^* = -Q_x < 0 \quad \text{à } t = t_K$$

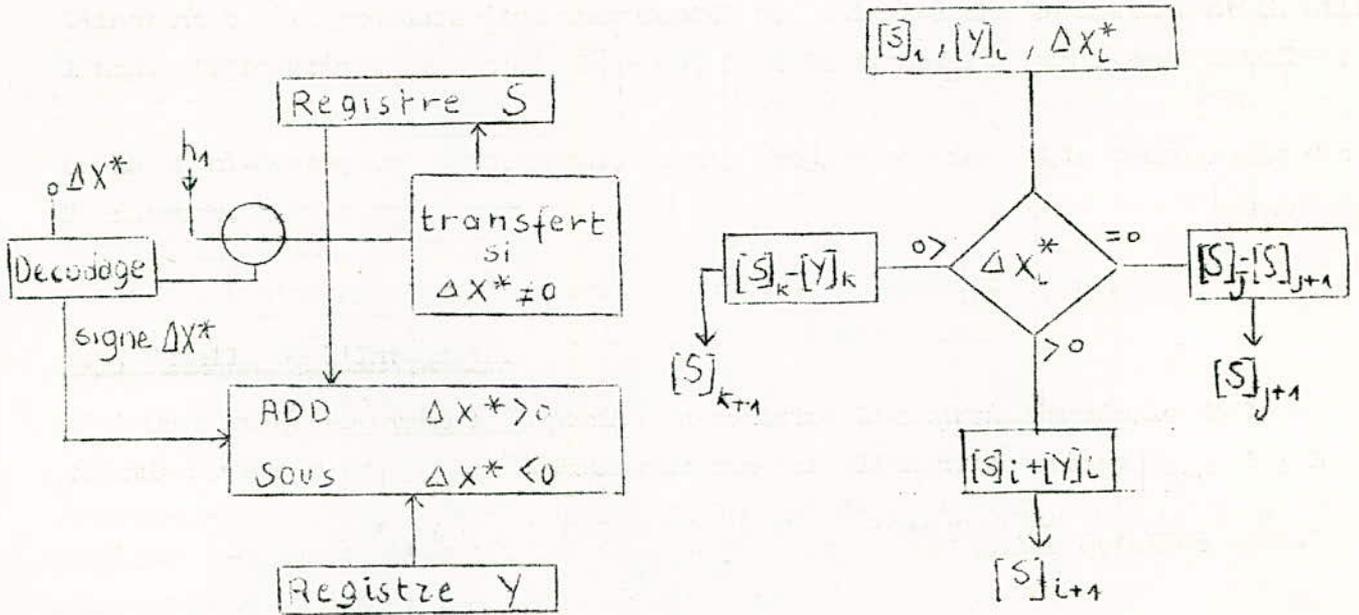
Dans ce cas on soustrait le contenu de Y du contenu S puis on transfère le résultat dans S afin de former  $S^{**}$  pour le temps suivant

$$[S]_K - [y]_K = [S]_{K+1} = S_{K+1}^*$$

La réunion des 3 cas s'écrit :

$$[S]_n + \left( \frac{\Delta X_n^*}{Q_x} \right) [y]_n = [S]_{n+1} = S_{n+1}^{**}$$

La figure 1-3 montre l'aspect de  $\Delta X^*$ ,  $y^*$  et  $S^{**}$  telles qu'elles apparaissent dans le calculateur au cours du temps.



• Schema fonctionnel de l'Integrateur Numerique

C'est le signe de l'increment  $\Delta X^*$  qui determine si l'on doit faire une addition ou une soustraction. D'autre part l'ordre de transfert n'est donné que si un increment se présente. Pratiquement l'ordre de transfert est donné périodiquement par un signal d'horloge et cet ordre est inhibé en l'absence d'incrément.

Dans le cas où la variable est le temps il suffit de simuler en permanence dès l'instant  $t_0$  la presence d'un incrément  $> 0$ . Ainsi à chaque instant d'échantillonnage (y compris  $t_0$ ) on a  $[S] + [y]$  et le transfert dans S sera possible.

Le schéma ci-dessus est fonctionnel, on n'y prejuge en rien de la technologie des operateurs.

1.4. Echelle de l'Intégrale.

On a vu que N étant la capacité du registre l'excursion maximale de la fonction y étant  $|y|_{\max}$  on définissait une echelle E telle que  $|y|_{\max} E \leq N$

- Cas où  $X = f(t)$  alors  $I(t_n) = \int_0^{t_n} y(t) dX(t)$  est liée à  $S_n^{**}$  calculée dans l'intégrateur numérique par :

$$I(t_n) = Q_X S_n^{**} \rightarrow I_n^* = Q_X S_n^{**} \text{ ou bien}$$

$$I_n^* = Q_X \sum y_{i-1}^* \frac{\Delta X_{i-1}^*}{Q_X}$$

Au quantum  $Q_Y$  de  $y \rightarrow Q_I^* = Q_X Q_Y$  de  $I^*$  (fig.1-3). On en déduit la capacité maximale de  $S$  c'est à dire  $N_S$

Si  $|X|_{\max}$  et  $|y|_{\max}$  sont les valeurs absolues maximum de  $X(t)$  et  $y(t)$

$$\left| \int y(t) dX(t) \right| < |X|_{\max} |y|_{\max} \quad |X|_{\max} \ll N_X Q_X, |y|_{\max} \ll N_Y Q_Y$$

$$, |I|_{\max} \ll N_X N_Y Q_X Q_Y = N_X N_Y Q_I^* \quad \boxed{N_S = N_X N_Y}$$

On peut aussi écrire

$$|I|_{\max} E_I^* \leq N_S \quad E_I^* = \frac{1}{Q_I^*} \text{ avec } Q_I = N Q_I^*, N \text{ entier}$$

On aura une représentation à incréments unitaires de  $I^*$ . Dans le cas de la figure 1-3  $Q_I = 5 Q_I^*$

- Cas où  $t$  est la variable :  $I_n^* = T \sum y_{i-1}^* = T S_n^*$  de quantum

$Q_I^* = T Q_Y$  avec  $Q_I = N Q_I^* = N T Q_Y$  on aura une représentation à incréments unitaires de  $I_n^*$ .

L'Intégrateur Numérique Incremental.

On désire que les résultats apparaissent sous forme numérique incrementale (increments unitaires).

2.1. Condition suffisante pour obtenir des Increments unitaires dans la R.N.I:

Pour que  $I^*$  ne comporte que des I.u. il suffit que  $Q_I$  soit supérieur à

$$|\Delta I^*|_{\max}$$

$$I^* = Q_X S^{**} \text{ on peut déterminer } |\Delta I^*|_{\max} \text{ à partir de } |\Delta S^{**}|_{\max}$$

$$I_n^* = Q_X \sum_{i=1}^n y_{i-1}^* \frac{\Delta X_{i-1}^*}{Q_X}, \quad S_n^{**} = \sum_{i=1}^n y_{i-1}^* \frac{\Delta X_{i-1}^*}{Q_X}$$

$$S_{n+1}^{**} - S_n^{**} = y_n^* \frac{\Delta X_n^*}{Q_X}$$

$$|\Delta S^{**}|_{\max} = |y^*|_{\max} \Rightarrow |\Delta I^*|_{\max} = Q_X |y^*|_{\max}$$

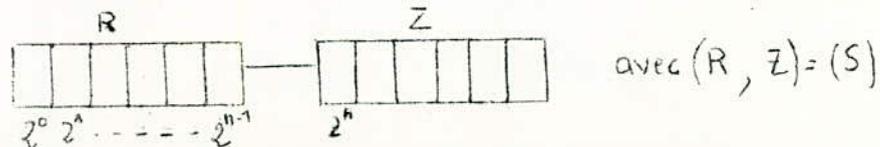
Or  $|y^*|_{\max} \leq N_y Q_y \Rightarrow |\Delta I^*|_{\max} \leq N_y Q_y Q_X$  et la condition suffisante pour obtenir des increments unitaires dans la representation numérique de l'intégrale est

$$Q_I \gg N_y Q_y Q_X \text{ ou encore } E_I \leq \frac{E_x E_y}{N_y}$$

La plus petite valeur du quantum sera  $Q_I = N_y \cdot Q_y \cdot Q_X$  representation la plus précise.

2.2. Organisation du calcul.

Reportons nous au schéma fonctionnel paragraphe 1.3.2 et imaginons que S se divise en 2 registres partiels R et Z. S est un registre binaire.



La capacité du registre R est  $N_R = 2^0 + 2^1 + \dots + 2^{n-1} = 2^n - 1$ . R est entièrement rempli par  $(2^n - 1)$  quanta  $Q_y$ . Si on ajoute alors un quanta  $Q_y$  à [S] apparition d'une unité dans la première cellule Z (poids  $2^n$ ) tandis que R est à 0 fig.2.1. Dès lors le contenu du registre Z peut être considéré comme une nouvelle représentation  $S^{***}$  de  $S^{**}$  de quantum  $Q_Z = 2^n \cdot Q_y$   $Q_Z = (N_R + 1)Q_y$ .

Le contenu de Z peut aussi être considéré comme une nouvelle représentation numérique de  $I^*$  c'est à dire  $I^{**}$  ou Z pour rappeler que l'intégrale se trouve dans le registre Z, avec  $Q_I = (N_R + 1) Q_X Q_y$  si  $N_R = N_y$   $Q_I = (N_y + 1) Q_X Q_y$  ce qui satisfait à l'obtention d'incréments unitaires et diffère peu du cas optimal car  $N_y \gg 1$ .

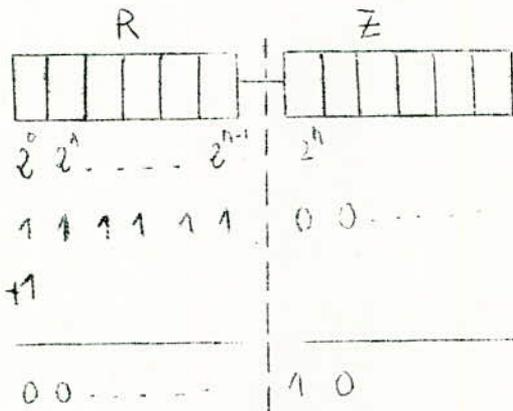


FIG. 2.1

nombre représenté	$S^{**}$	$S^{***}$	$I^{**}$
$2^n - 1$	$(2^n - 1)Q_y$	0	0
+ 1	+ $Q_y$		
$2^n$	$2^n Q_y$	$Q_Z$	$Q_I$

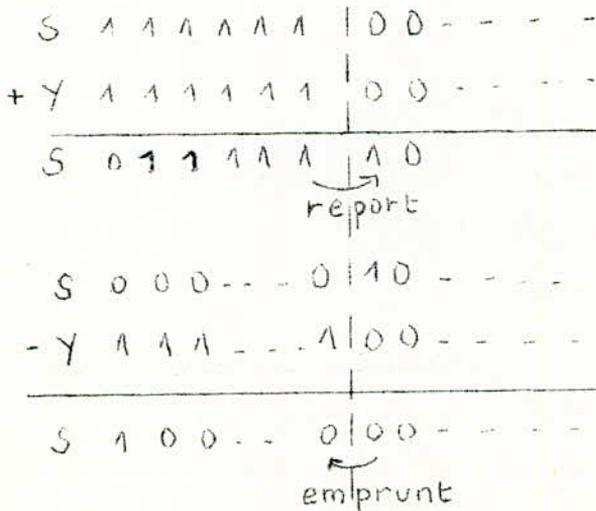


FIG 2.2

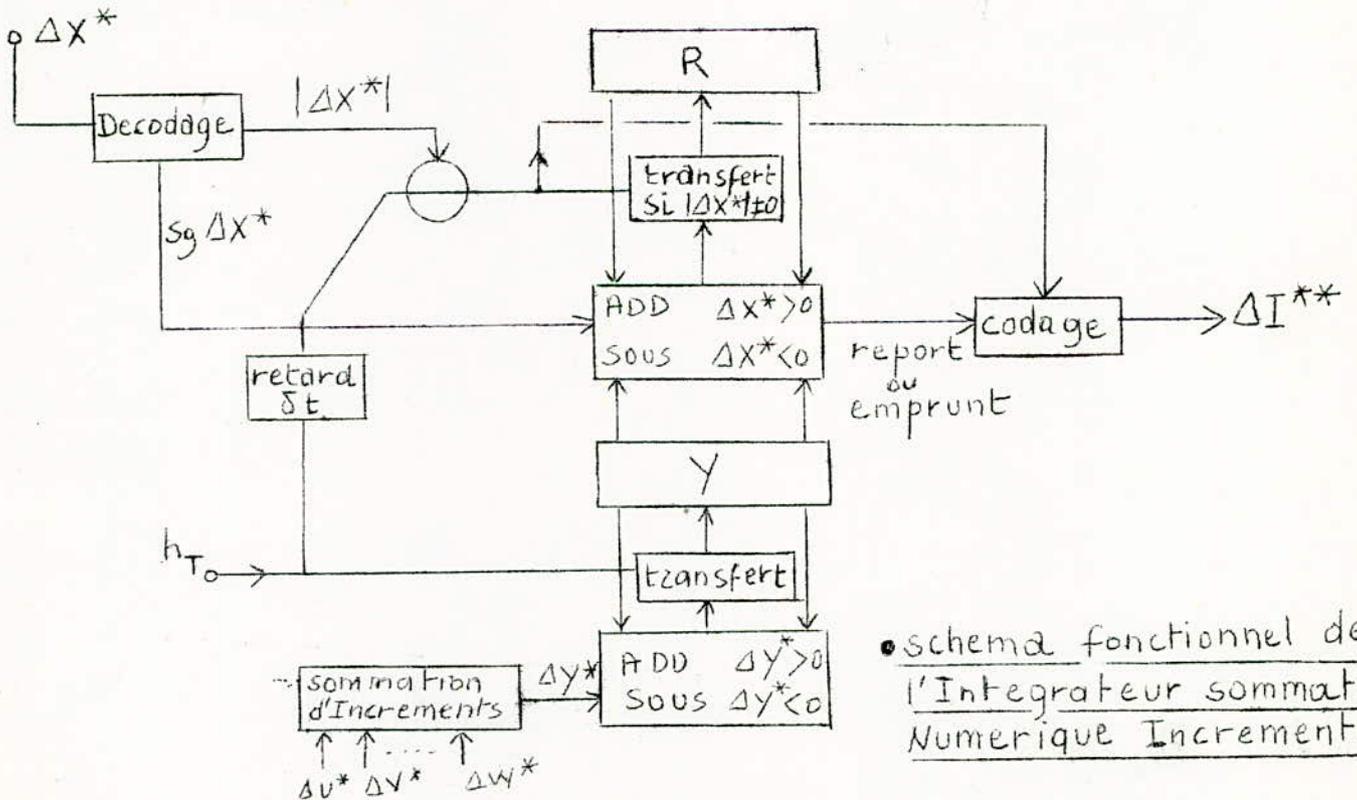
nombre représenté	valeurs $S^*$ et $Y^*$	$S^{***}$	$I^{**}$	$R^*$
$2^n - 1$	$(2^n - 1)Q_y$	0	0	$(2^n - 1)Q_y$
+ $2^n - 1$	+ $(2^n - 1)Q_y$			
$2^n + (2^n - 2)$	$[2^n + (2^n - 2)]Q_y$	$Q_Z$	$Q_I$	$(2^n - 2)Q_y$
$2^n$	$2^n Q_y$	$Q_Z$	$Q_I$	0
$-(2^n - 1)$	$-(2^n - 1)Q_y$			
1	$Q_y$	0	0	$Q_y$

D'après les exemples de la figure 2.2. le contenu de Z ne peut varier que d'un quantum  $Q_Z$  à la fois lors d'une addition  $[S] + [y]$  ou  $[S] - [y]$ . Les variations du contenu de Z dépendent exclusivement de la présence d'un report (add) ou d'un emprunt (sous) dans l'add- sous au niveau de la dernière cellule de y et R.

$$\text{Un report} \quad \begin{cases} \Delta S^{***} = (N_{y+1}) Q_y = \Delta Z^+ \text{ (increment positif)} \\ \Delta I^{**} = Q_I = Q \Delta Z^+ \end{cases}$$

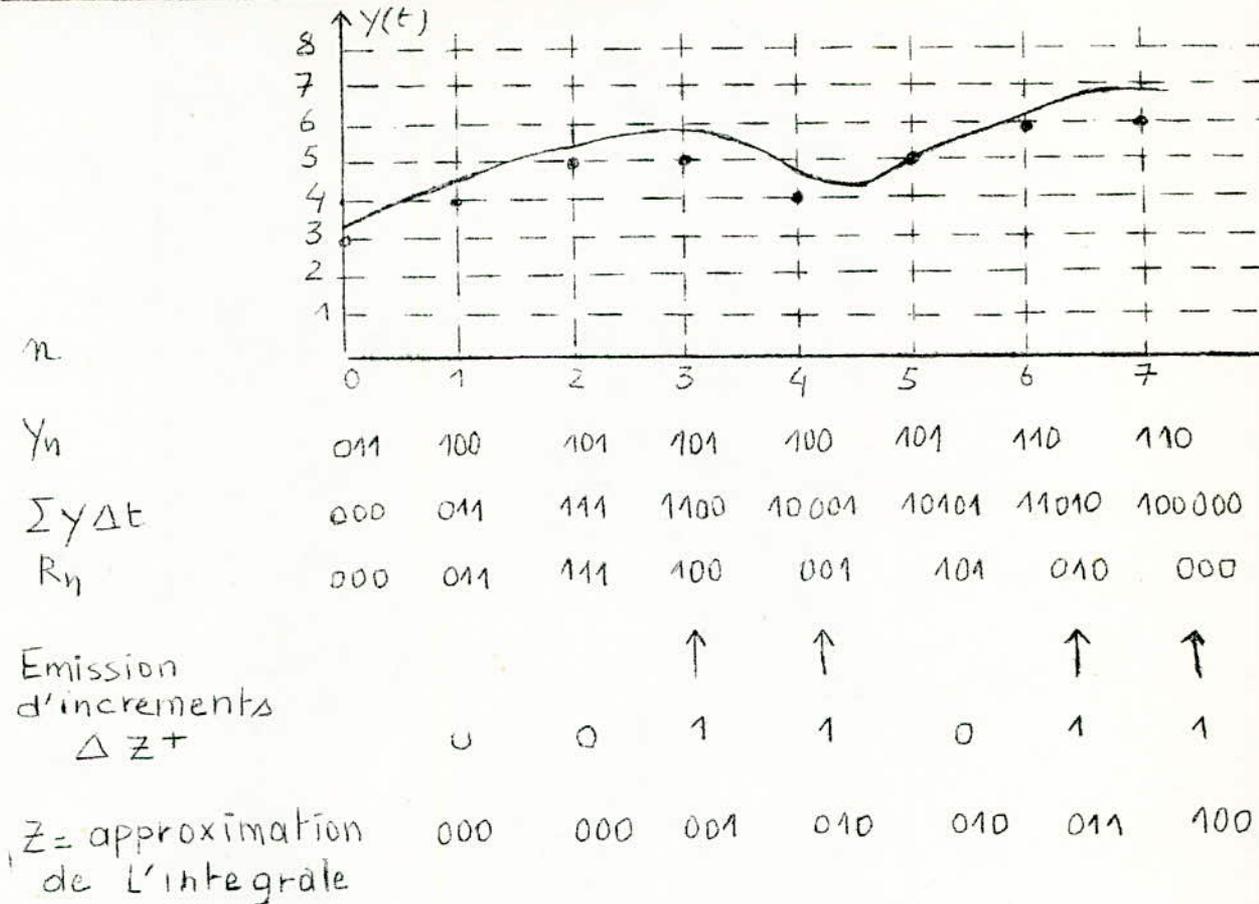
$$\text{Un emprunt} \quad \begin{cases} \Delta S^{***} = - (N_{y+1}) Q_y = \Delta Z^- \text{ increment negatif} \\ \Delta I^{**} = - Q_I = - Q_X \Delta Z^- \end{cases}$$

Ces reports ou emprunts peuvent constituer une sortie I.u.  $\Rightarrow Z$  n'intervient pas dans la formation de l'increment de sortie. Si l'on ne s'intéresse qu'à la partie incrementale de l'intégrale on peut supprimer Z.



Il peut y avoir plusieurs increments  $\Delta y^*$  au cours d'une iteration il convient alors de les ajouter avant de les introduire dans l'intégrateur.

2.3. Exemple d'intégration par rapport au temps



Loi de recurrence:

$$\begin{aligned} R_{n+1} &= R_n + y_n \\ y_{n+1} &= y_n + \Delta y_n \end{aligned}$$

La figure ci-dessus montre la variation d'une fonction du temps  $y(t)$ . Les instants d'échantillonnage  $0, 1, \dots, n$  sont. La valeur analogique  $y'_n$  issue de l'échantillonnage est digitalisée en un mot  $y_n$  de 3 bits par un convertisseur analogique digital que l'on suppose fournir une approximation par défaut de  $y'_n$  (On se limite à  $y > 0$  pour simplifier le raisonnement). Supposons de plus que la période d'échantillonnage soit suffisamment brève pour que  $y$  ne change pas de plus d'un quantum en plus ou en moins de l'échantillonnage  $n$  à l'échantillonnage  $n+1$  : en d'autres termes seul le bit le moins significatif change dans cet intervalle de temps.

L'expression numérique de l'intégrale est approximée par la "méthode des rectangles", qui consiste par exemple à effectuer la somme des aires des rectangles définis sur la figure. En partant des valeurs initiales  $y_0 = 3$  et 0 pour la somme des aires des rectangles :

$\sum y \Delta t = 0$  pour  $n = 0$ , on incremente  $y_0$  de 1 bit pour obtenir  $y_1$ , la première aire est obtenue en ajoutant  $y_0$  à l'aire initiale. La loi de recurrence est donc :

$$I \begin{cases} y_{n+1} = y_n + \Delta y_n \\ (\sum y \Delta t)_{n+1} = (\sum y \Delta t)_n + y_n \end{cases}$$

Si l'on utilise des registres à trois bits pour contenir  $y_n$  (stocké dans  $y$ ) et  $(\sum y \Delta t)_n$  (stocké dans  $R$ ) on voit que  $R$  va déborder du côté des poids les plus élevés : On dit qu'il emet un "increment" que l'on note  $\Delta Z^+$ . La figure donne des exemples d'émission de tels increments. On peut maintenant accumuler ces increments dans un registre qui sera appelé  $Z$  et l'on constate sur la figure que ce registre contient un nombre binaire (poids des bits  $2^5, 2^4, 2^3$ ) qui est la représentation binaire de la somme des aires. Les bits de poids plus faibles restent naturellement dans  $R$  et l'on appelle  $R_n$ , reste à l'iteration  $n$ , le contenu de ce registre. Si l'on considère maintenant des nombres négatifs on constate que l'accumulation des valeurs négatives de  $y_n$  produit également un débordement du registre  $R$  accumulant ces valeurs, débordement que l'on qualifie de négatif (emprunt) par opposition au précédent qui était un débordement positif et que l'on note  $\Delta Z^-$ .

Un tel intégrateur numérique est incremental, ternaire.

- Il est incremental en ce sens qu'à chaque iteration il reçoit un incrément d'amplitude nulle ou égale à un quantum pour "mettre à jour" la fonction  $y$  intégrée de plus par le jeu du débordement du registre  $R$ , il emet un increment.

- Il est ternaire parce que l'increment qu'il reçoit peut avoir 3 valeurs : + 1, 0, -1 ainsi que l'increment qu'il emet.

Ce fait est fondamental car il permet la mise en serie des intégrateurs.

En résumé l'intégrateur numérique incremental et ternaire a le fonctionnement suivant :

- Il accumule les échantillons digitaux  $y_n$  (Kbits) de la fonction  $y$  à intégrer, selon le principe de la méthode des rectangles.

- Il accepte une "mise à jour" de  $y_n$  par incrementation ternaire.

- L'accumulation précédente se faisant dans un registre à Kbits, il produit des increments ternaires.

- Ces increments étant de la même nature que les increments acceptés peuvent être stockés dans l'accumulateur d'un autre intégrateur; cet accumulateur contient une approximation de l'intégrale cherchée.

Au lieu de I on peut écrire :

$$I' \quad \left[ \begin{array}{l} y_{n+1} = y_n + \Delta y_n \\ R_{n+1} = R_n + y_n \end{array} \right. \quad \text{Méthode des rectangles inférieurs}$$

$$II \quad \left[ \begin{array}{l} y_{n+1} = y_n + \Delta y_n \\ R_{n+1} = R_n + y_{n+1} \end{array} \right. \quad \text{Méthode des rectangles supérieurs.}$$

#### 2.4. Intégration par rapport à une variable autre que le temps.

Dans le cas de l'intégration par rapport au temps, le nombre  $y_n$  est multiplié par l'incrément de temps  $\Delta t$ , constamment égal au quantum de temps (ou encore une iteration) et ajouté à l'ancienne valeur de  $\sum y \Delta t$ .

On a vu que cette multiplication, puis addition se ramenait à une simple addition.

Dans le cas de l'intégration par rapport à une variable quelconque  $X$ , d'incrément  $\Delta X$ , l'intégration par la méthode des rectangles conduit à  $\sum y \Delta X$  : On voit donc qu'à chaque itération il convient d'effectuer une multiplication suivie d'une addition.

Cependant si  $X$  est du même type que  $\Delta t$  c'est à dire si  $\Delta X$  ne dépasse pas en valeur absolue un quantum au cours d'une iteration du calcul, la simplification précédente a encore lieu, si bien que le même type d'intégrateur peut servir à cette fonction.

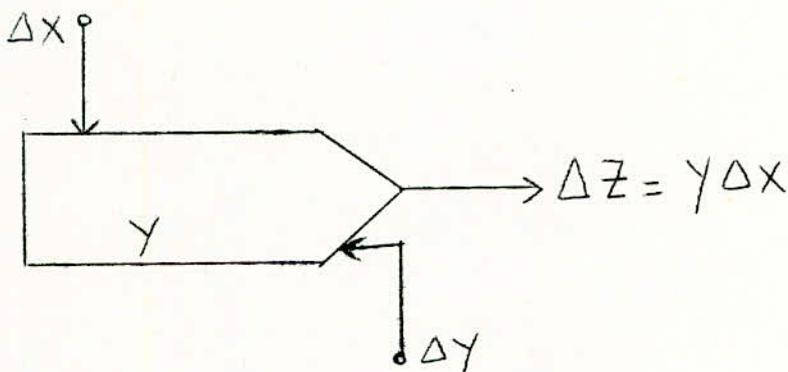
Il faut cependant remarquer que si dans le premier cas iteration et incrément de temps se confondent, il peut ici y avoir plusieurs increments  $\Delta X$  au cours d'une iteration:

Si cela est le cas, il faut additionner ces increments avant de les introduire dans l'intégrateur précédemment défini.

2.5. Symbole.

Ecrivons symboliquement  $(Z)(R) = \sum y \Delta X \Rightarrow \Delta[(Z)(R)] = y \Delta X$ , or nous approchons l'intégrale par le contenu de registre  $Z \Rightarrow \Delta(Z) = y \Delta X$ . Avec la sortie incrementale  $\Delta Z = y \Delta X$ .

La représentation symbolique de l'intégrateur numérique incrementale est :



CHAPITRE 3.

L'ANALYSEUR DIFFERENTIEL NUMERIQUE (DDA)

3.1. Représentation des nombres dans le D.D.A.

Pour réaliser un appareil universel il est nécessaire de savoir enregistrer des y négatifs et d'interpréter les  $\Delta X$  et  $\Delta y$  négatifs. Jusqu'à présent deux représentations sont adoptées. La représentation à increments binaires et celle à increments ternaires. Dans le 1er cas les increments ne peuvent avoir que deux états  $\pm 1$  dans le second par contre ils peuvent avoir 3 états différents: 0, + 1, -1. La signification du  $\Delta$  tant bien entendu définie par l'échelle que l'on adopte durant la programmation de la machine.

3.1.1. Représentation à increments binaires: Les nombres sont compris entre 0 et 2. Il est cependant arbitraire que les nombres négatifs soient compris entre 0 et 1 ou entre 1 et 2. - Voir table ci-dessous. Nous adoptons 4 bits pour la représentation des nombres.

La colonne I indique le nombre décimal équivalent au nombre binaire qui apparaît dans la machine; La colonne II la représentation binaire du nombre; la colonne III sa signification dans le système decimal. Nous pouvons considérer le bit zéro en tête comme signe négatif et le bit 1 comme signe positif: colonne III ou inverser la convention: colonne IV. Il est toujours admis que les nombres soient normalisés c'est à dire que leur valeur absolue doit être inférieure à 1 avant d'être introduits dans la machine.

Lorsque y ou R contiennent 0,000 cela est interprété comme -1. Lorsqu'ils contiennent 1111 cela est interprété comme + 1. et inversement

Equivalent decimal	Représentation machine	Signification du nombre binaire	Convention inverse
1,875	1.111	0,875	-0,875
1,750	1.110	0,750	-0,750
1,625	1.101	0,625	-0,625
1,500	1.100	0,500	-0,500
1,375	1.011	0,375	-0,375
1,250	1.010	0,250	-0,250
1,125	1.001	0,125	-0,125
1,000	1.000	0,000	0.000
0,875	0.111	-0,125	+0.125
0,750	0.110	-0,250	+0,250
0,625	0.101	-0,375	+0,375
0,500	0.100	-0,500	+0,500
0,375	0.011	-0,625	+0,625
0,250	0.010	-0,750	+0,750
0,125	0.001	-0,875	+0,875
0,000	0.000	-1.000	+1,000

Le plus grand nombre positif 1.111 diffère du nombre le plus négatif 0,000 par un simple bit en effet

$$\begin{array}{r}
 1.111 \\
 1 \text{ bit} \quad \underline{\quad 1} \\
 (1) \ 0 \ 000 \\
 \longleftarrow \text{retenue}
 \end{array}$$

Notons aussi que l'on peut représenter exactement -1 et non + 1 pour la première convention et + 1 et non -1 pour la seconde.

3.1.2. Représentation à increments ternaires : figure 3.2. Dans ce cas lorsque y ou R contiennent :

0,000 cela est interprété comme 0. Nous pouvons aussi avoir la convention inverse c'est à dire :

0,111	+ 1	0,111	- 1
1,000	- 1	1,000	+ 1

Equivalent décimal	représentation machine	signification du nombre binaire	convention inverse
0,875	0,111	0,875	- 0,875
0,750	0,110	0,750	0,750
0,625	0,101	0,625	- 0,625
0,500	0,100	0,500	- 0,500
0,375	0,011	0,375	- 0,375
0,250	0,010	0,250	- 0,250
0,125	0,001	0,125	- 0,125
0,000	0,000	0,000	0,000
1,875	1,111	-0,125	0,125
1,750	1,110	-0,250	0,250
1,625	1,101	-0,375	0,375
1,500	1,100	-0,500	0,500
1,375	1,011	-0,625	0,625
1,250	1,010	-0,750	0,750
1,125	1,001	-0,875	0,875
1,000	1,000	-1,000	1,000

La plus employée.

FIG 3.2

Insistons sur le fait que cette distribution des nombres binaires dans les deux systèmes a été établie de telle sorte que l'on ait pour :

- Le premier cas (ou inversement)      0000  $\rightarrow$  - 1      1111  $\rightarrow$  + 1      increments binaires

Pour le second cas nous devons pouvoir représenter  $-1,0, +1$ . Ce sera donc une simple représentation binaire fractionnaire avec la convention des compléments à 2. Avec ce système, plus généralement si l'on utilise des mots de 4 bits (bit signe compris) on pourra représenter les nombres compris entre  $-1$  et  $+1-2^{-3}$ . L'addition ou la soustraction dans ce système donne le résultat correct en valeur absolue et en signe, dans ce même système, et ce, sans correction ni complémententation. Pour fixer les idées et distinguer entre le premier et le second système considérons un fonctionnement simple.

3.13. Exemple de fonctionnement dans le système à increments binaires:

Prenons  $y = \text{constante} = 0,250$  ; R étant initialisé à zéro et supposons que  $\Delta X = \text{constante}$ .

(suite d'impulsions périodiques) - Le fonctionnement de la machine pour le système à increments binaire, est décrit dans la figure 3.3. ci-dessous :

Fig 3.3

It	R	y	R + y	Overflow	Signification décimale
	1.000				0,000
1		1.000	0.010	1	0,250 -0,750
2	0.010	1.010	1.100	0	- 0,750 +0,250 +0,500
3	1.100	1.010	0.110	1	0,500 0,250 -0,250
4	0.110	1.010	0.000	1	-0,250 0,250 -1,000
5	0.000	1.010	1.010	0	-1,000 0,250 0,250
6	1.010	1.000	0.100	1	0,250 0,250 -0,500
7	0.100	1.010	1.110	0	-0,500 0,250 0,750
8	1.110	1.010	1.000	1	0,750 0,250 0,000

A la 8ème iteration le registre R retourne à sa valeur initiale.

Examinons en détail la première iteration nous savons que nous devons avoir comme résultat  $R = 0,250 = \frac{1}{4}$ . En fait le registre R contient  $-\frac{3}{4}$  et il y a eu un overflow. Si nous interprétons cet overflow comme étant + 1 en faisant la somme:  $\text{Overflow} + R = 1 - \frac{3}{4} = \frac{1}{4}$  résultat exact.

Considérons la 2ème iteration: Nous devons avoir  $(\text{overflow})_{n-1} + (\text{overflow})_n +$

$$R_n = \frac{1}{2}$$

Or  $1 + 0 + 0,500 = 1 + \frac{1}{2}$ . Ce qui n'est pas le résultat exact. Si nous interprétons l'absence d'overflow comme étant - 1 alors nous obtiendrons un résultat exact. Globalement au bout de ce cycle nous aurons  $5 - 3 = 2$  overflows. Ce phénomène explique la dénomination d'increments binaires.

Dans ce système l'overflow 0 n'existe pas de la même façon que pour l'overflow 1 ou - 1 il est représenté par une suite sequentielle d'overflows (1,0,1,0...) (1, -1, 1, -1....) En effet l'addition repetitive de zero nous donne :

(R)	[y]	[R] + [y]	Overflow	Signification décimale
1.000	1.000	0.000	1	0,000 0,000 -1,000
0.000	1.000	1.000	0	-1,000 0,000 0,000
1.000	1.000	0.000	1	0,000 0,000 -1,000

### 3.14. Exemple de fonctionnement dans le système ternaire.

Supposons  $y = 0,250 = 0,010$

$R_0 = 0,000$

$\Delta X = T$  fonctionnement périodique.

Les séquences de fonctionnement sont indiquées sur la figures 3.4.

	[R]	[y]	[R]+[y]	Overflow	Signification décimale
1	0,000	0,010	0,010	0	0,000 0,250 0,250
2	0,010	0,010	0,100	0	0,250 0,250 0,500
3	0,100	0,010	0,110	0	0,500 0,250 0,750
4	0,110	0,010	1,000	+1	0,750 0,250 -1,000

A la 4ème séquence le registre R est à - 1.

Règle générale: A chaque fois, lors d'une addition  $[R]+[y]$ , qu'il y a un report dans le bit signe nous changeons le 1 en zéro et ce 1 représentant un overflow dont le signe sera positif si  $\Delta X$  et y ont même signe et négatif si  $\Delta X$  et y ont des signes différents à la 4ème i teration nous aurons alors :

$$R = 1.000 \longrightarrow 0.000 \quad (+1) \longrightarrow \text{overflow}$$

└───────────────────────────────────┘ nouvelle valeur de R

La fin du cycle correspond au retour à la valeur initiale de R.

Les nombres négatifs sont considérés d'une manière légèrement différente.  
 Considérons le même fonctionnement que précédemment mais avec:

$$\begin{cases} y = -\frac{1}{4} = 1.110 \\ R_0 = 0.000 \end{cases}$$

	[R]	[y]	[R+y]	Overflow	signification décimale
1	0.000	1.110	1.110 0,110	-1	0,000 -0,250 -0,250 +0,750
2	0.110	1.110	0.100 On ignore le report	0	0,750 -0,250 0,500
3	0.100	1.110	0010 On ignore le report	0	0,500 -0,25 0,250
4	0010	1.110	0000 On ignore le report	0	0,250 -0,250 0,000

fig 3.5

Le véritable report, au delà du bit signe est ignoré. Il n'y a d'overflow que lorsqu'un report apparaît dans le bit signe uniquement.

Nous deduisons de ce qui précède que le résultat final est obtenu au bout d'un nombre d'iteration moitié en fonctionnement ternaire qu'en fonctionnement binaire.

### 3.2. Programmation d'un DDA.

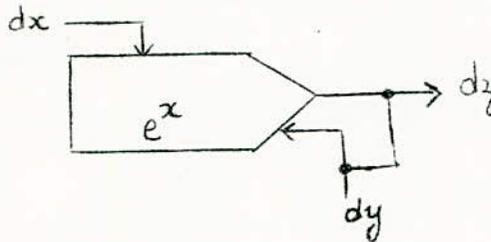
Le DDA ne comporte aucun système d'instruction ni de programme emmagasiné. La programmation est réalisée de façon générale en reliant les unités, que sont les intégrateurs, les uns aux autres selon la particularité du problème. Les entrées et sorties seront alors reliées à un tableau de branchement. Pour programmer un problème sur DDA il convient de décomposer l'ensemble des équations en un ensemble de générations de  $y$  dx.

Exemple:

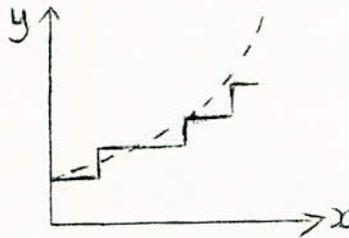
\* Génération de la fonction  $y = y_0 e^x$ .

Avant de commencer le calcul on introduit  $y_0$  dans le registre Y.

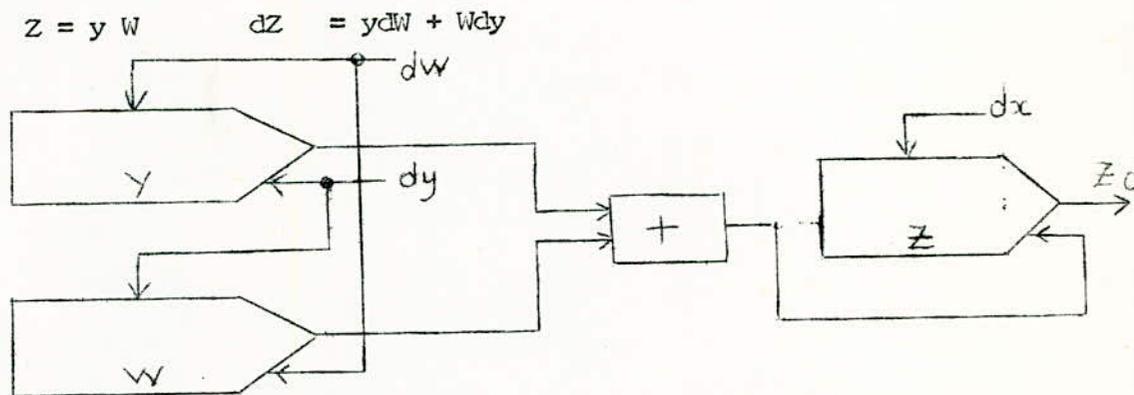
$$y = \frac{dy}{dx} \Rightarrow dy = y dx \quad \text{d'où}$$



Si  $dx$  est un débit d'impulsion donné,  $dz$  sera un débit  $e^x$  et si le registre Y est accessible on pourra y trouver à chaque instant la fonction  $y = e^x$ ,  $x$  étant à chaque instant le nombre total d'impulsions depuis le début de l'intégration. L'exponentielle obtenue sera alors de la forme :



\* Multiplication: Soit à effectuer le produit de deux fonctions  $y$  et  $w$  de la variable  $x$ .



Note: Dans DDA les résultats de calcul ne peuvent être recueillis que dans le registre d'un intégrateur.

Aussi  $Z = yw$  sera dans le registre d'un troisième intégrateur.

\* Considérons l'équation différentielle non linéaire.

$$\frac{d^2 x}{dy^2} + x \frac{dx}{dy} + by^2 = a \sin cy$$

Ecrivons la sous la forme :

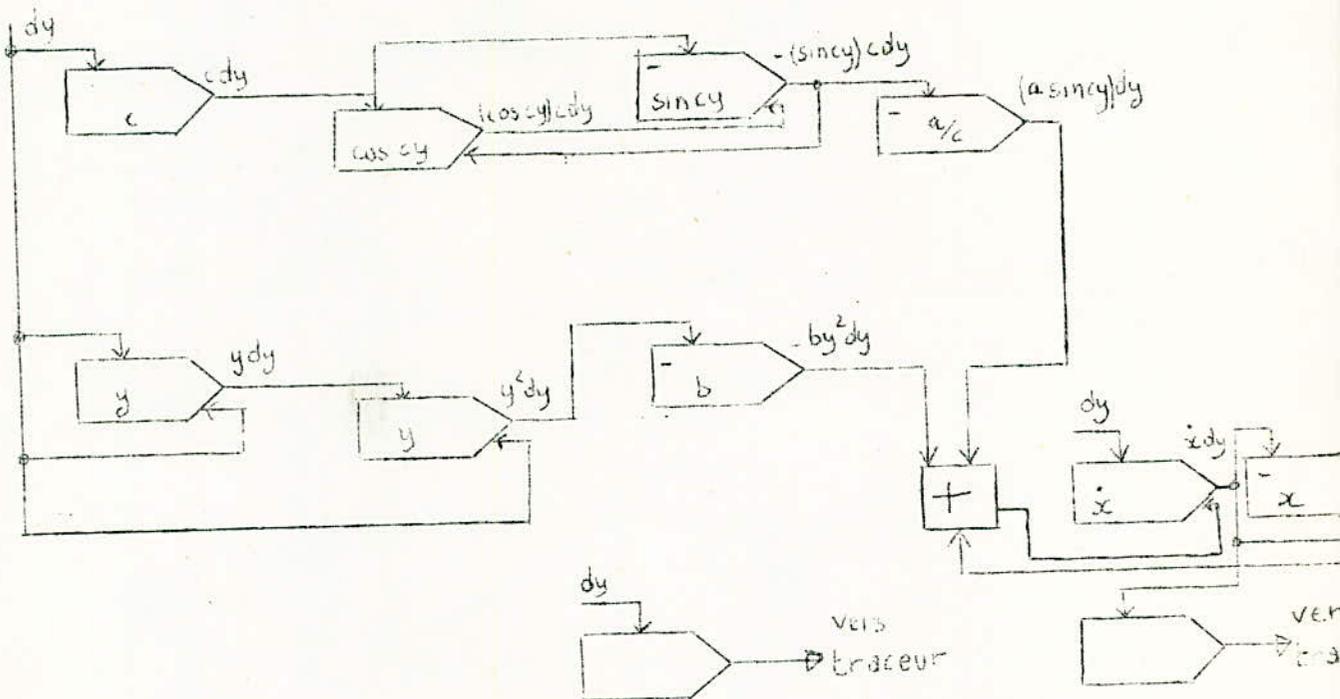
$$\frac{d^2 X}{dy^2} dy + x \frac{dx}{dy} dy + B y^2 dy = a \sin cy dy$$

$$\frac{d^2 x}{dy^2} dy + x dx + by^2 dy = (a \sin cy) dy$$

$$d \frac{dx}{dy} = \dot{x} \quad \dot{x} = (a \sin cy - by^2) dy - x dx. \quad (1)$$

$$d(\sin cy) = c \cos cy dy$$

$d(\cos cy) = -c \sin cy dy$  le diagramme flow de l'équation (1) sera alors:



La fonction cosinus ou sinus sera le résultat d'une génération à partir d'une équation différentielle telle que  $\frac{d^2 y}{dx^2} + y = 0$

### 3.3. Mise à l'échelle d'un problème.

On a vu que les nombres machines peuvent être soit exprimés dans le système à increments binaires soit dans le système à increments ternaires. Cela implique nécessairement une procédure de mise à l'échelle pour passer des variables problèmes aux variables-machines.

Les variables-machines (lettres majuscules) sont liées aux variables-problème par les relations suivantes:

$$y = Y 2^\alpha \quad \delta \text{ correspond à la place du 1 du quantum } Qy$$

$$dx = \Delta X 2^\gamma \quad \text{par exemple } Qy = 0001 ; \quad \delta = 4$$

$$dy = \Delta Y 2^\delta$$

$$dz = \Delta Z 2^\varepsilon \quad \gamma \text{ et } \varepsilon \text{ caractérisent les increments } \Delta X \text{ et } \Delta Z.$$

Le poids des increments générés est fonction des poids des increments admis:

$$\varepsilon = \gamma + \alpha \quad \text{puisque } dz = y dx = Y \Delta X 2^{\gamma + \alpha}$$

$\alpha$  est choisi de telle façon que le registre  $Y$  ne déborde pas pour cela il suffit de satisfaire la relation  $|y_{\max}| < 2^\alpha$ . Lorsque l'on réalise un schéma DDA il faut que les facteurs d'échelle aux deux extrémités d'une connexion soient identiques. Lorsque la sortie de l'intégrateur  $i$  est reliée à l'entrée  $dy$  de l'intégrateur  $j$  on doit avoir  $\varepsilon_i = \delta_j$ ; si elle est reliée à l'entrée  $dx$  de l'intégrateur  $k$  on doit réaliser  $\varepsilon_i = \gamma_k$ . Lorsque le schéma est simple et le problème bien connu il est facile de déterminer à la main les facteurs d'échelle. Lorsque le schéma est complexe, le DDA comportant un grand nombre d'intégrateurs, on aura recours à un programme de détermination et de correction automatiques des facteurs d'échelle par l'utilisation d'un ordinateur.

3.4. Erreurs en calcul par DDA: De même que les résultats fournis par un ordinateur doivent être accueillis, malgré le nombre impressionnant de chiffres "significatifs" avec la confiance que peut seule donner une étude d'analyse numérique de même, pour un schéma DDA déterminé il est impératif de se livrer à une étude détaillée des erreurs surtout si l'on ne dispose pas de moyen pour contrôler les résultats. Si l'on considère le fonctionnement d'un intégrateur au point de vue des erreurs on peut distinguer deux composantes:

- Erreur de méthode
- Erreur d'arrondi.

\* Erreur de méthode: On utilise un algorithme d'intégration numérique. Le processus d'intégration par la méthode des rectangles pourra être amélioré par celui des trapèzes où au lieu d'utiliser à chaque addition  $y_i$  on utilise  $y_i + \frac{\Delta y_i}{2}$ . On aura alors

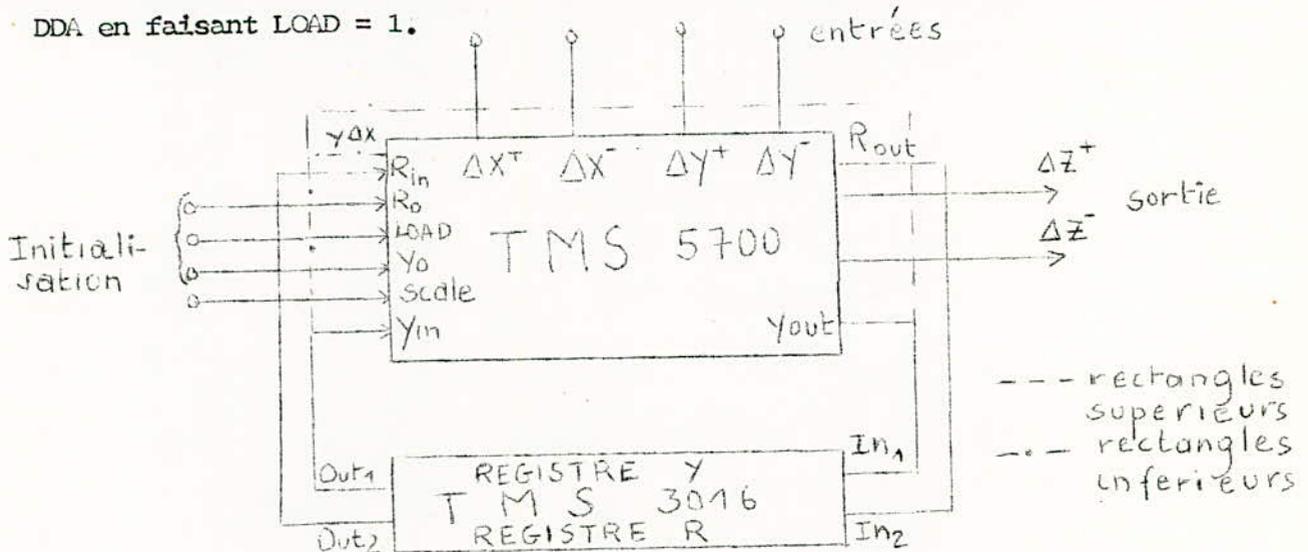
$$y_{n+1} = y_n + \Delta y_n$$

$$R_{n+1} = R_n + y_n + \frac{1}{2} \Delta y_n$$

Néanmoins dans les deux cas il y a troncature du développement en série de l'intégrale cherchée.

\* Erreur d'arrondi : L'increment  $\Delta Z$  mis ne représente pas parfaitement la variation de l'intégrale; il subsiste un "reste" dans le registre R.

3.5. Réalisation en MOS de l'intégrateur: Les techniques d'intégration à grande échelle permettent maintenant d'avoir l'opérateur de base du DDA sous forme de deux boites de réalisation MOS: L'élément TMS 5700 et l'élément TMS 3016. Ce dernier est un registre double 16 bits. Le premier est composé de deux additions soustractions série Y et R d'un circuit élaborant l'increment  $\Delta Z^+$ ; un autre  $\Delta Z^-$  d'un circuit de chargement des conditions initiale  $y_0$  et  $R_0$ , valeurs qui sont introduites dans les registres Y et R à l'initialisation du DDA en faisant LOAD = 1.



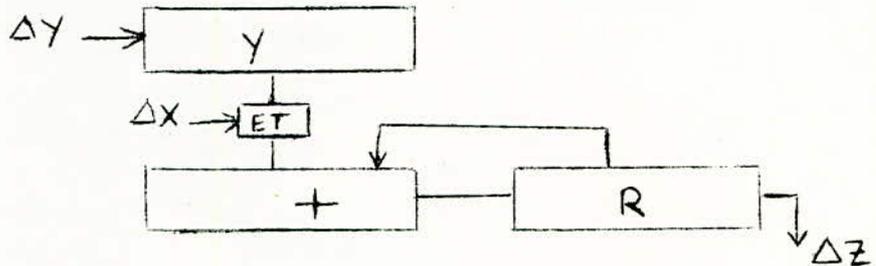
Le débit "scale" correspond à l'increment de un quantum pour  $y_n$  c'est à dire  $\Delta y_n$ .

2° Partie : Réalisation Pratique de L'intégrateur  
 Numérique incrémental à circuits intégrés.

1°/ Schéma Synoptique :

Nous avons vu lors de l'étude théorique que le fonctionnement de l'intégrateur numérique incrémental nécessitait :

- Une mise à jour de l'échantillon digital  $Y_n$  de la fonction  $Y$  à intégrer par incrementation ternaire : lorsqu'un incrément  $\Delta Y^*$  arrive nous devons avoir  $Y_{n+1} = Y_n + \Delta Y_n$ .
- Une accumulation de ces échantillons digitaux dans un registre  $R$  : lorsqu'un incrément  $\Delta X^*$  arrive nous devons avoir  $R_{n+1} = R_n + Y_n$ . Cette accumulation se traduit alors par une émission d'incrément  $\Delta Z^*$  correspondants au débordement de  $R$  ce qui aboutit au schéma de principe suivant :



qui peut se traduire par le schéma synoptique ci-après :

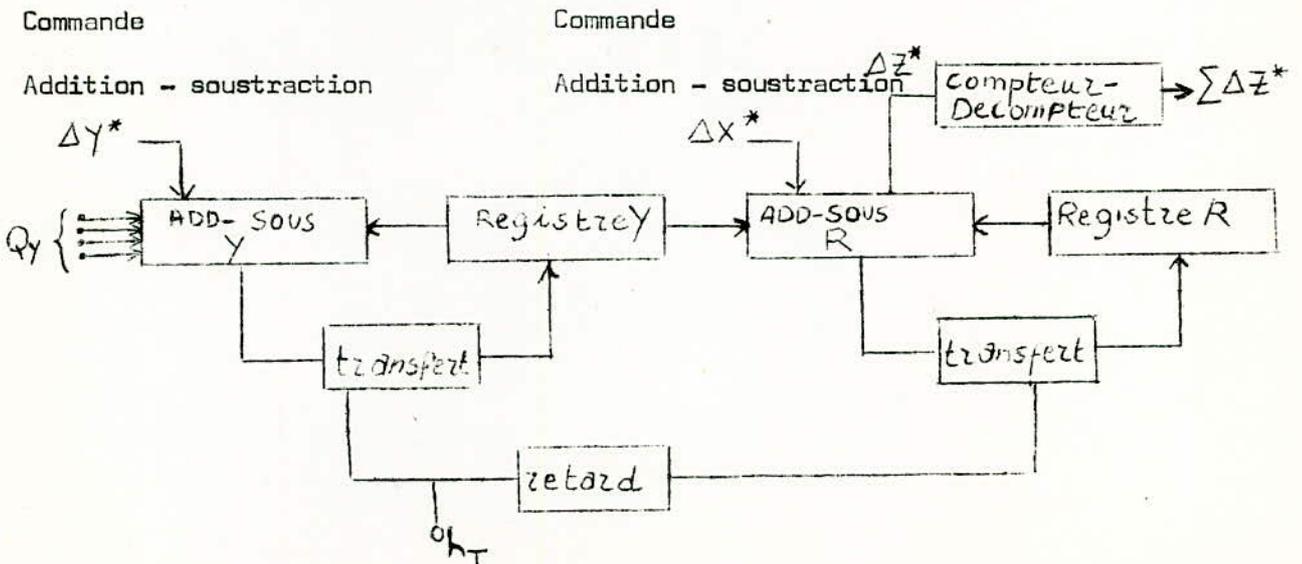
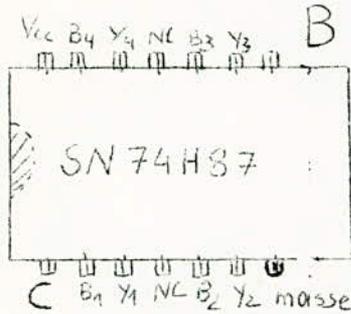


Schéma que nous proposons de réaliser avec les circuits intégrés accomplissant les fonctions d'addition, soustraction de mise en mémoire, de comptage - décomptage .

2°) Notes sur les circuits intégrés fondamentaux.

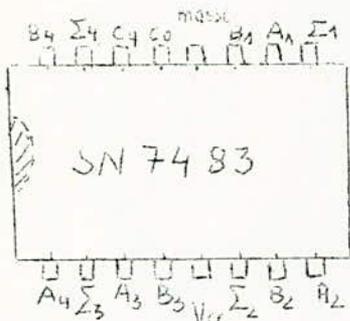
2.1. L'additionneur soustracteur 4 bits: Il est réalisé à l'aide de deux boîtiers qui sont le SN 74 H 87 N générateur de complément 4 bits et le SN7483 additionneur quatre bits.

\* Implantation et fonctionnement du SN74H87.



contrôle des entrées		sorties				
B	C	y <sub>1</sub>	y <sub>2</sub>	y <sub>3</sub>	y <sub>4</sub>	
0	0	$\overline{B}_1$	$\overline{B}_2$	$\overline{B}_3$	$\overline{B}_4$	Soustraction
1	0	1	1	1	1	
0	1	B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>	B <sub>4</sub>	addition
1	1	0	0	0	0	Conservation

\* Implantation et fonctionnement du SN7483: C'est un additionneur parallèle 4 bits. Les bits des deux nombres entrent simultanément et les bits somme sont formés simultanément. Le nombre de cellules constituant l'additionneur est égal au nombre de bits à additionner. Chaque cellule délivrant un bit somme et un bit retenue qui se propage en cascade.



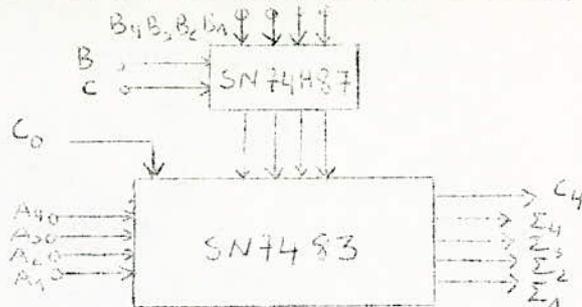
A <sub>i</sub>	B <sub>i</sub>	C <sub>i-1</sub>	Σ <sub>i</sub>	C <sub>i</sub>
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Tableau de vérité d'une cellule

$$\Sigma_i = A_i \oplus B_i \oplus C_{i-1}$$

$$C_i = A_i B_i + (A_i + B_i) C_{i-1}$$

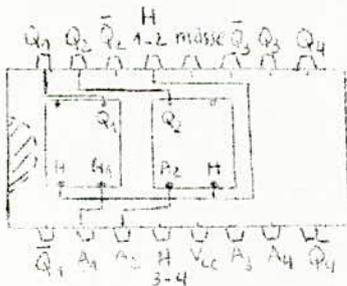
\* Fonctionnement de l'ensemble SN 74 H 87 - 74 83.



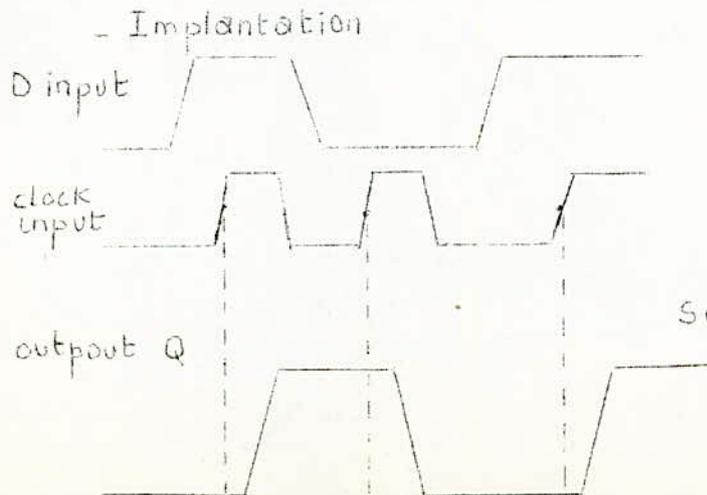
Les entrées B et C commandent l'opération à effectuer : addition ou soustraction. Dans le cas où l'on fait la soustraction le nombre  $A_4 A_3 A_2 A_1$  est ajouté au complément à 1 de  $B_4 B_3 B_2 B_1$  c'est à dire  $\overline{B_4} \overline{B_3} \overline{B_2} \overline{B_1}$ .

Donc pour avoir le résultat en complément à 2 il suffit d'ajouter un 1 au résultat ce qui revient à avoir une retenue initiale  $C_0 = 1$ .

**2.2. Le registre Mémoire SN 74 75:** La fonction mémoire 4 bits est réalisée à l'aide de 4 bascules D; chaque couple étant commandé par le même signal d'horloge. Les données présentes à l'entrée ne sont transmises en sortie que sur un ordre d'un signal horloge. Pour avoir un fonctionnement correct il est nécessaire que l'information encadre le signal horloge.



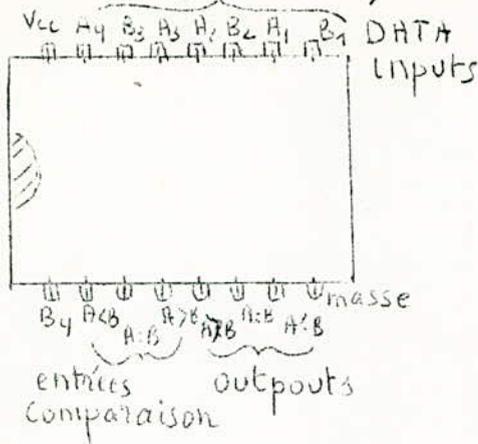
$t_n$	$t_{n+1}$
$A_i$	Q
1	1
0	0



- table de vérité

sequences de fonctionnement

2.3. Le comparateur binaire 4 bits SN7485: C'est un comparateur parallèle. Il indique si  $A < B$  ou  $A > B$  ou  $A = B$ .

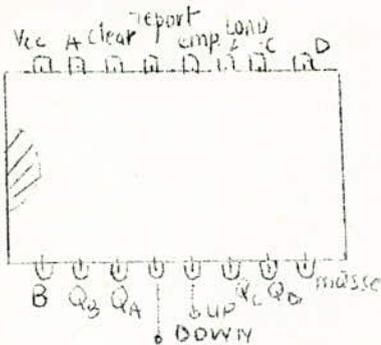


Les entrées comparaison doivent être reliées au Vcc si on fait simplement la comparaison de deux nombres de 4 bits.

2.4. Le compteur décompteur SN 74193.

C'est un compteur décompteur asynchrone à retenue parallèle. Le SN 74193 est conçu pour compter en binaire pur.

\* Implantation.



\* Fonctionnement : voir figure II.1.

Il peut être chargé parallèlement. Les données appliquées sur les entrées ABCD sont placées dans le compteur lorsque l'entrée "LOAD" est au niveau logique "0". L'action de cette entrée "LOAD" est indépendante de l'état du compteur et des entrées comptage ou décomptage.

Les sorties des autres bascules master Slave changent d'état sur une transition d'un niveau logique "0" à un niveau "1" de l'une quelconque des entrées comptage ou décomptage. En effet un étage inverseur commande les entrées de déclenchement des bascules, à partir des entrées UP, DOWN.

Figure II.1

Compteur-decompteur SN74193

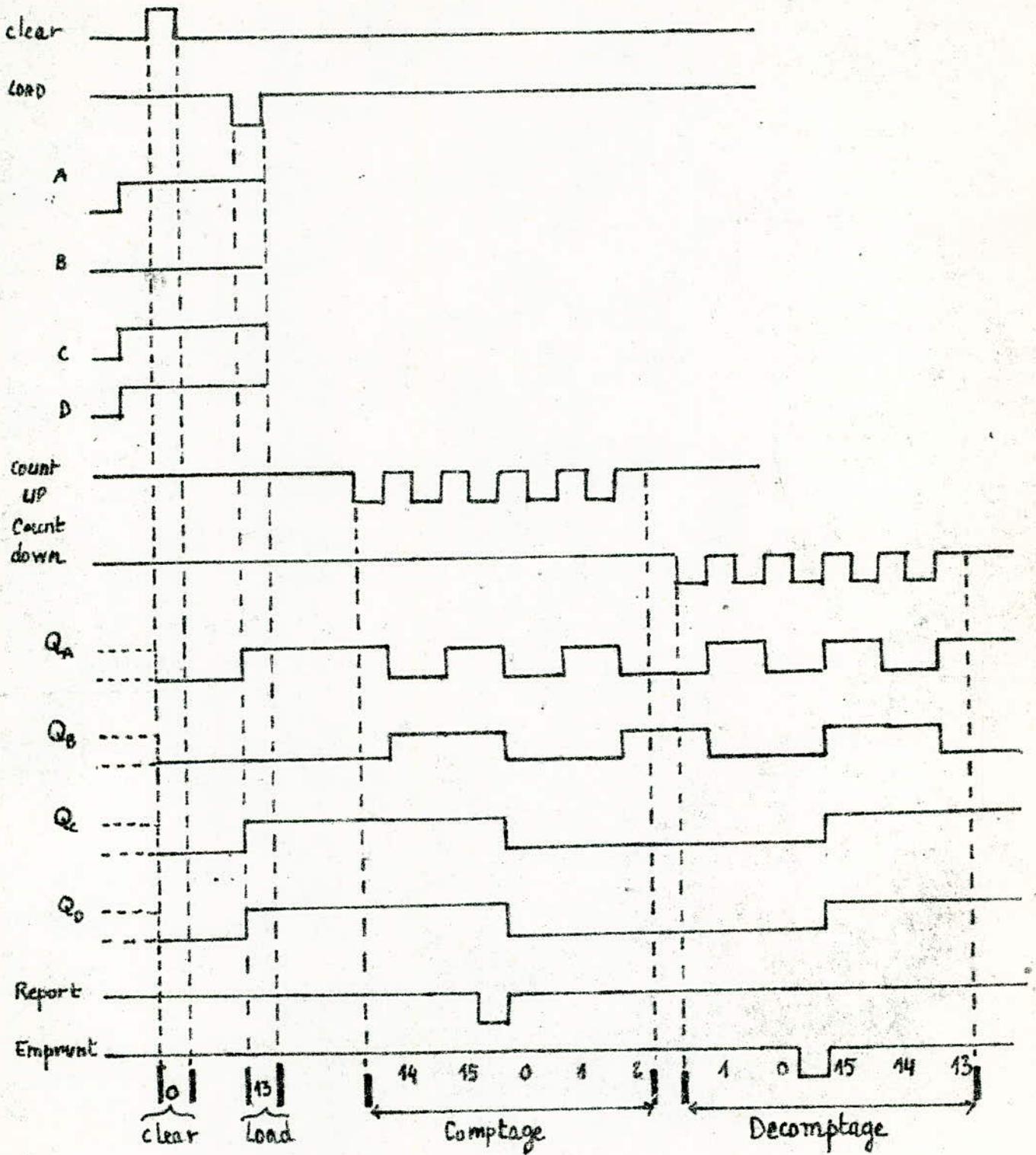


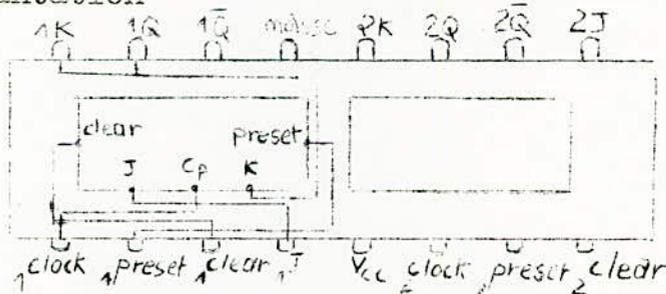
Figure II.1

Dans ce type de compteur il n'y a pas de contrôle de sens du comptage. Ce sens est déterminé par l'entrée qui reçoit les impulsions d'horloge. Celle ne recevant pas d'impulsions doit être laissée au niveau "1". L'entrée de Remise à zéro (RAZ) : clear, agit au niveau "1". Elle est indépendante de l'état de l'entrée de chargement "LOAD" ou des entrées de comptage UP et DOWN.

2.5 Flip Flop JK Maître esclave SN 7476 : Le SN 7476 est une double bascule JK synchrones avec deux entrées de forçage Clear et Preset.

Une transition du niveau logique "1" au niveau logique "0" sur preset place la sortie Q de la bascule au niveau "1". Une transition du niveau "0" au niveau "1" sur Clear place la sortie Q de la bascule au niveau "0". Clear et preset sont indépendantes de l'horloge. Dans cette présente étude nous le ferons fonctionner en forçage dans le système Marche-Arrêt.

Implantation



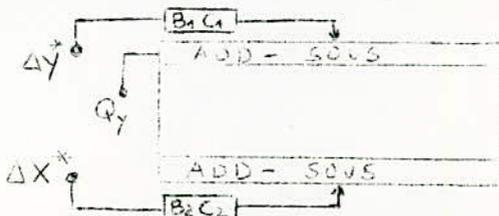
$t_n$		$t_{n+1}$
J	K	Q
0	0	$Q_n$
0	1	0
1	0	1
1	1	$\bar{Q}_n$

2.6 Flip Flop JK Maître esclave SN 7473

C'est aussi une double bascule JK mais avec une seule entrée de forçage "Clear" qui permet par l'application d'un niveau bas de placer Q à 0 indépendamment de l'entrée horloge. Il sera utilisé comme compteur jusqu'à 4 dans le sequenceur.

3°/ Schéma de l'Unité principale à C.I et fonctionnement  
3.1 : schéma voir figure II.2

oo schéma peut être représenté symboliquement par :



$$\sum \Delta Z^* = S_n^{***} = S_n^{**} - R_n \neq S_n^{**} = \frac{I^*}{Q_x}$$

3.2 Fonctionnement : Le schéma a été conçu de telle façon que les données et les résultats soient des nombres binaires représentés par l'allumage ou l'extinction de lampes. Nous utiliserons comme détecteur de reports et d'emprunts, dont le résultat global à un facteur près constituera une valeur approchée de l'intégrale, un compteur décompteur.

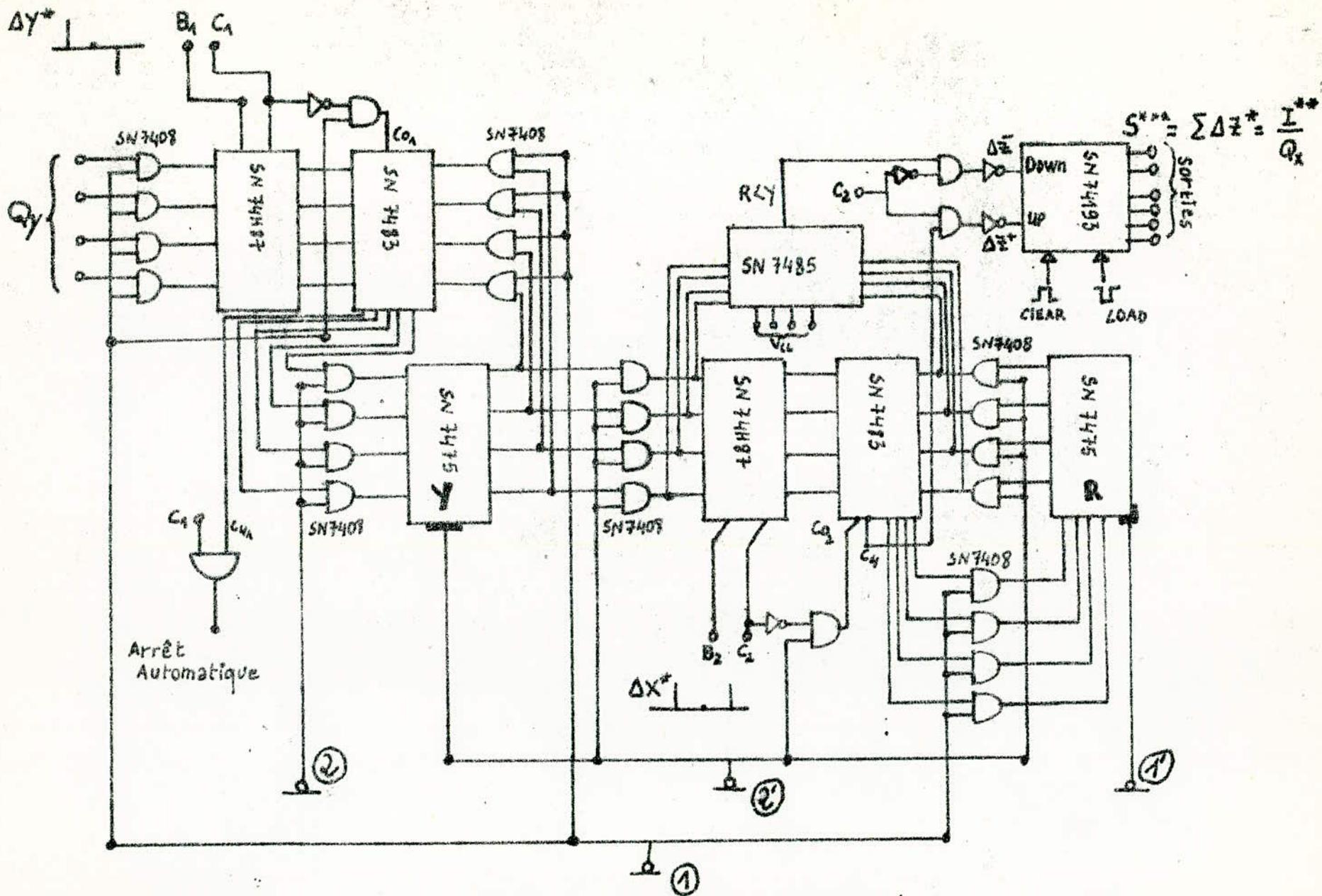
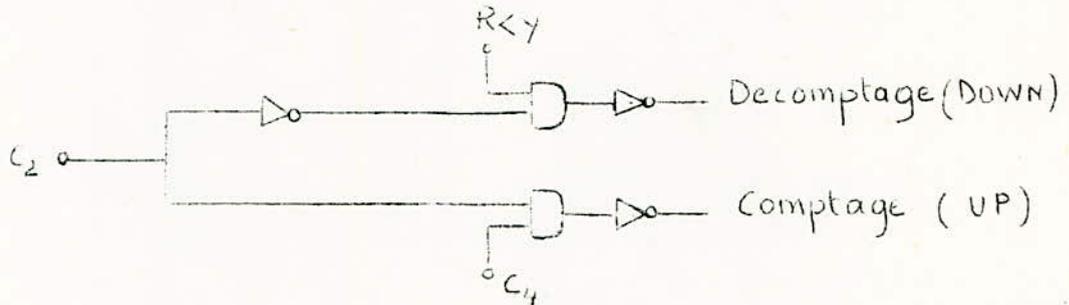


Figure II.2 schema de l'unité principale

Il y a comptage si lors d'une addition (Y+R) il y a un débordement de R ce qui revient à avoir une reteme à la sortie de l'additionneur-soustracteur correspondant .

Il y a décomptage si lors d'une soustraction (R-Y) il y a un emprunt de R ce qui revient à avoir  $R < Y$  . Le circuit logique permettant d'avoir ce fonctionnement est le suivant .



$C_4$  reteme de l'addition (Y + R)

$B_2, C_2$  constituant la commande de l'additionneur soustracteur R

$B_1, C_1$  Constituant la commande de l'additionneur soustracteur Y

En fait l'état de repos des entrées UP et DOWN du compteur décompteur est le niveau "1" . On mettra un inverseur à la sortie de chaque porte

$$U_P = \overline{C_2 \cdot C_4} \quad \text{DOWN} = (R < Y) \cdot \overline{C_2}$$

Les données sont : - L'increment  $\Delta Y^*$  représenté par la commande  $B_1, C_1$

- L'increment  $\Delta X^*$  " " " "  $B_2, C_2$

- QUANTUM  $Q_Y$  représenté par un nombre binaire  $A_4 A_3 A_2 A_1$  .

- L'intialisation du compteur -Décompteur, ABCD

### Phases de fonctionnement

Phase 1 :-addition Y

-Les informations à la sortie de l'additionneur soustracteur R sont transmises à l'entrée du registre R .

Phase 1' :-Les informations présentes à l'entrée du registre R sont transmises à la sortie

Phase 2 :-Les informations à la sortie de l'add-Sous-Y sont transmises à l'entrée du registre memoire Y

Phase 2' : Addition R .

En supposant avoir à l'entrée  $Q_Y$ , un fonctionnement en additionneur des deux add-sous-traduisons sous forme de tableau l'evolution du système en initialisant à zéro .

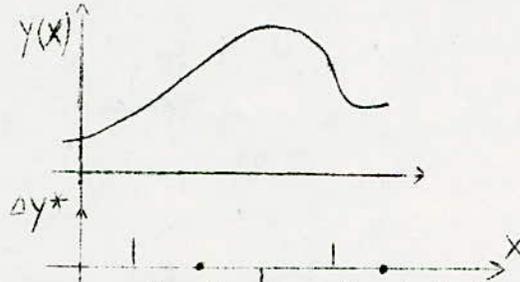
	ADD1	[Y]	ADD2	[R]
1-1'	Qy	0	0	0
2-2'		Qy	Qy	
"	2Qy	2Qy	2Qy	2y
"	3Qy	3Qy	6Qy	3Qy
"	4Qy	4Qy	10Qy	6Qy
"	5Qy	5Qy	15Qy	10Qy

Dans ce cas particulier notre système réalisé alors  $R_{n+1} = R_n + Y_n$ .

$$Y_{n+1} = Y_n + Qy$$

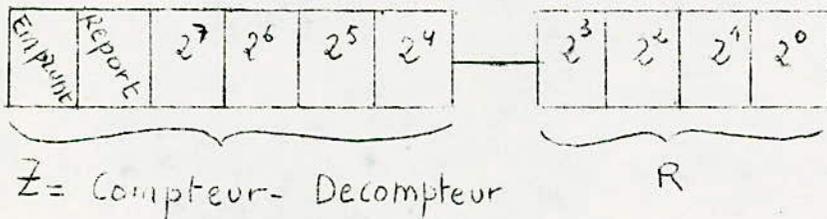
et s'arrêtera automatiquement lorsque une retenue se produit à la sortie de l'additionneur soustracteur R cela revient à avoir une valeur maximum de la fonction Y égale à 1111 dans le cas où l'échelle E est égale à 1.

En résumé : Avec ce système nous pouvons, par accumulation d'increments, calculer l'intégrale de toute fonction dont la forme est la suivante.



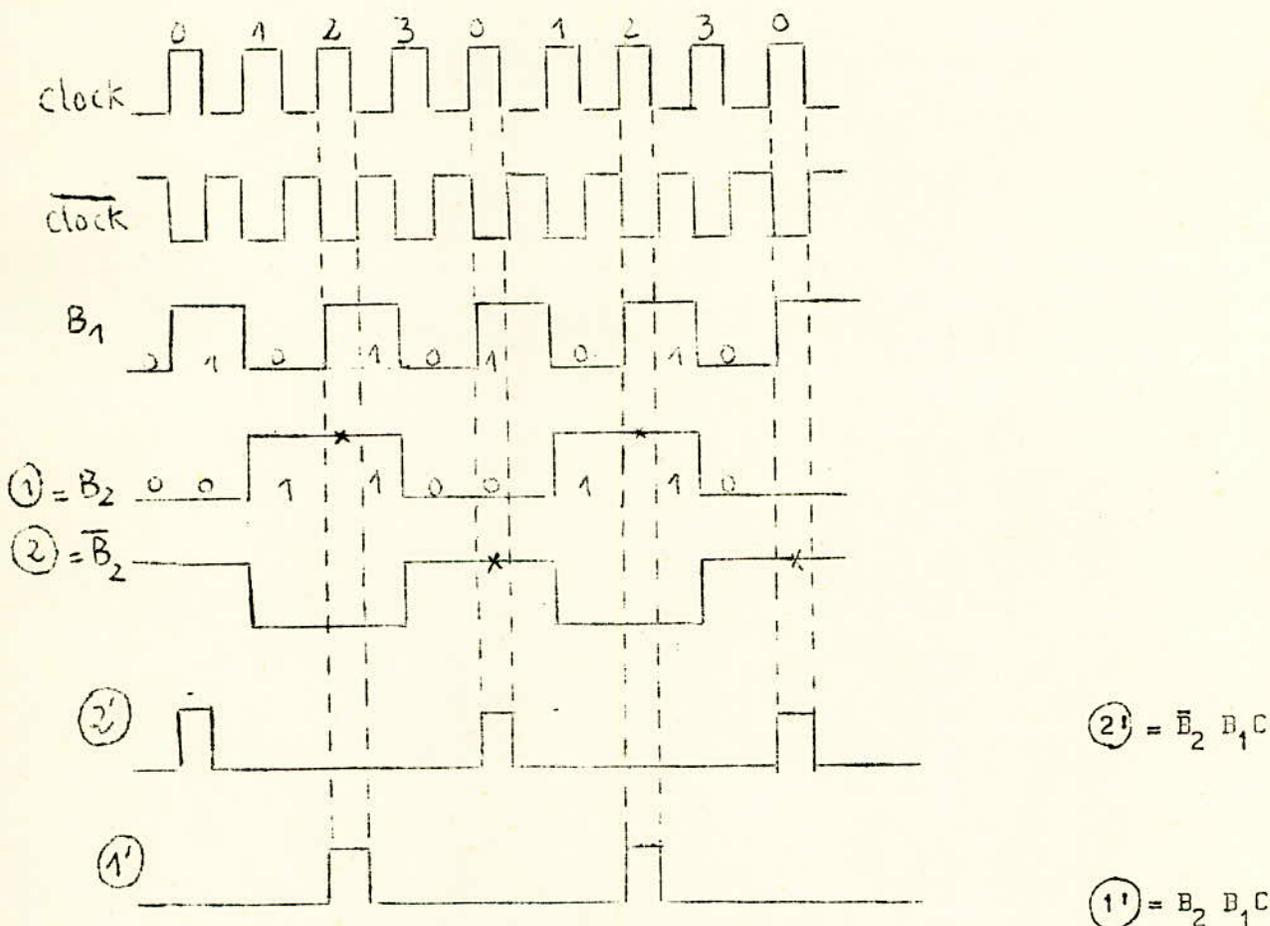
X étant du même type que le temps c'est à dire qu'il est unitaire. Y et R sont des nombres positifs et il y aura soit addition soit soustraction soit conservation selon que  $Y^*$  et  $\Delta X^*$  sont positifs, négatifs ou égaux à zéro.

Le résultat sera le contenu de E c'est à dire  $S_n^{***} = \sum \Delta S_n^{***} = \sum \Delta Z^*$   
 $S_n^{***}$  n'est autre qu'un "tronçage" de  $S_n^{**} = \sum_{i=1}^n Y_i^* \cdot \frac{\Delta X_{i-1}^*}{QX}$



$$S = (Z, R)$$

4°/ Le séquenceur : Voir figure II.3



Nous voyons <sup>d'après</sup> le schéma précédent que nous avons besoin de 4 signaux d'horloge.  
 Les signaux 1 et 2 étant complémentaires ; Quant à (1') et (2') ils doivent être encadrés respectivement par (1) et (2) ; le bon fonctionnement des registres memoires SN 74 75 nécessitant l'encadrement du signal horloge par le signal information. Ces signaux sont obtenus à l'aide d'un compteur synchrone par 4, ainsi que des circuits auxiliaires que sont ce SN 7410 et le SN 7404.

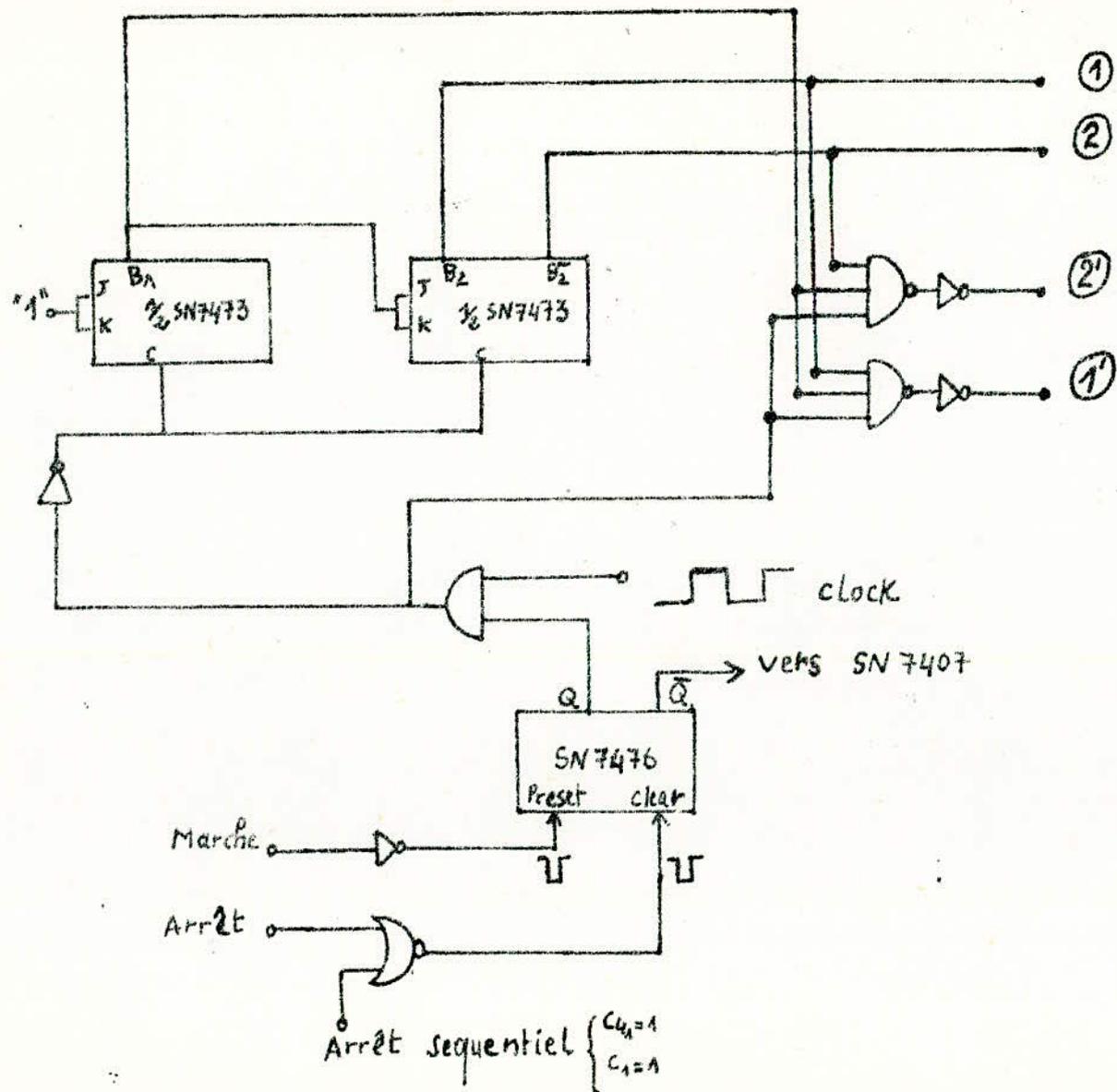


figure II.3- séquenceur : ① =  $B_2$     ② =  $\bar{B}_2$     ①' =  $B_2 B_1$  clock    ②' =  $\bar{B}_2 B_1$  clock

Synthèse du compteur par 4 : B1 et B2 étant les sorties des bascules JK

$(B_2 B_1)_t$	$(B_2 B_1)_{t+1}$
0 0	0 1
0 1	1 0
1 0	1 1
1 1	0 0

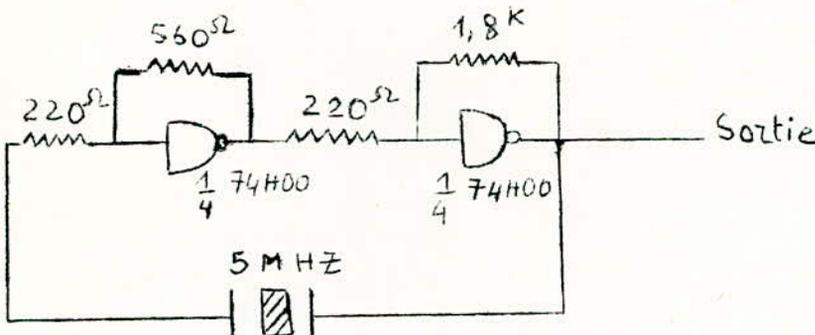
$$\text{donc } (B_1)_{t+1} = \bar{B}_1 t \Rightarrow J_1 = K_1 = 1$$

$$(B_2)_{t+1} = (B_1 \oplus B_2) t \quad J_2 = K_2 = B_1$$

Le séquenceur proprement dit est commandé par un système marche-arrêt. Le signal horloge n'est transmis aux entrées des bascules JK du séquenceur que si le niveau de sortie du SN 7476 est "1". Pour qu'il le soit il suffit de passer du niveau "0" au niveau "1" sur l'entrée Preset : Entrée Marche Manuelle. L'Entrée clear est commandée soit par un arrêt manuel soit par un arrêt automatique.

Voir figure II. 3

5°/ L'horloge : c'est pendant le top 2<sup>e</sup> que s'effectue le transfert des contenus des registres Y et R vers l'entrée de l'additionneur soustracteur ou la comparaison. La durée de ce top est égale à celle du signal d'horloge ; donc la fréquence de l'horloge doit être telle que  $\frac{1}{2}$  temps de réponse de l'ensemble SN 7475 - 7408 - 74H - 7583 ( < 100ns) pour un signal d'horloge symétrique.



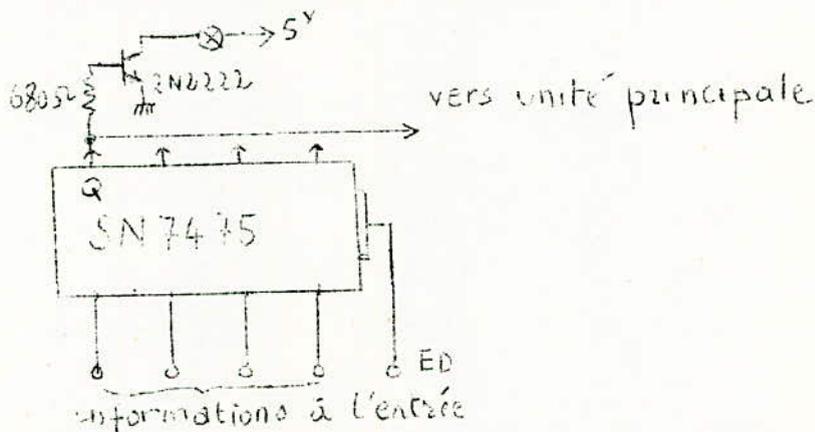
6°/ SYSTEME D'ENTREE SORTIE - Voir Figure II. 4

=====

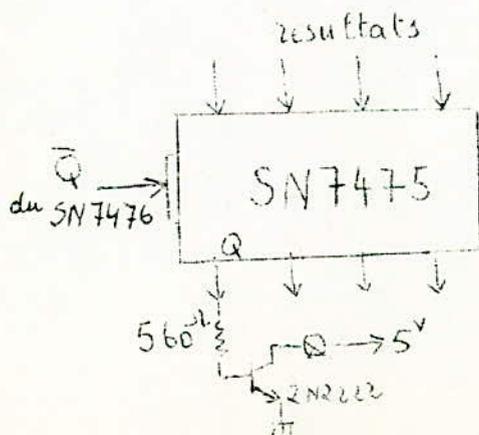
\* Visualisation des Informations à l'entrée : quantum  $Q_1$ ; commande  $B_1-C_1$

$B_2-C_2$ ; Initialisation et forçage du compteur. L'affichage des données commandé par l'entrée ED (entrée données) se fait manuellement à l'aide d'interrupteurs. La sortie Q du registre SN 7475 attaqué à travers une résistance de  $680\Omega$  la base d'un transistor 2N2222 chargé par une lampe  $6,5\text{ V} - 70\text{ ma}$ .

Lorsque  $Q = 1$  la lampe s'allume. Le schéma est le suivant :



\* Visualisation des Résultats: Tant que le système évolue les comptages ou décomptages seront à l'entrée du SN 7475 de sortie et ce n'est que lorsqu'il y a arrêt automatique ou manuel c'est à dire que la sortie complémentaire du SN 7476 est vraie ( $\bar{Q} = 1$ ) qu'il y a transfert des résultats sur lampes. Notons cependant que le niveau de l'information à la sortie est plus faible que celui de celle à l'entrée car elle aura traversé un certain nombre de circuits aussi la résistance de base des transistors de sortie sera-t-elle plus faible.



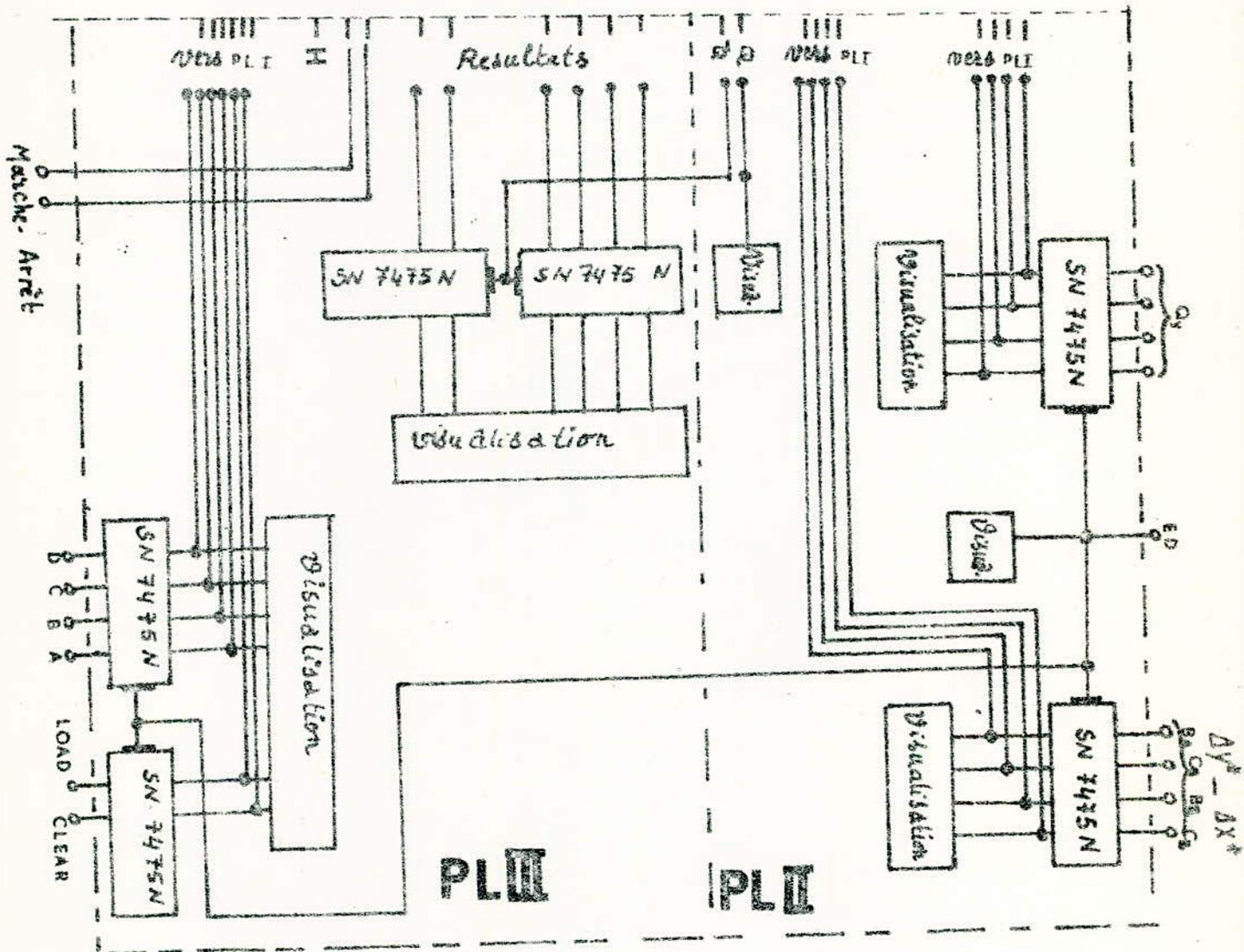


Fig II-4 ENTREE-SORTIE

7°/ SCHEMA GENERAL - Voir figure II. 4

8°/ Circuits intégrés Utilisés

\* Intégrateur Incremental PLI

2/SN 7483

2/SN74487

7+1/SN 7408  
4

5/6/SN 7404

2/SN 7475

1/SN 7485

1/SN74193

\* Séquenceur PLI

1/SN 7473

1/SN 7476

2/3SN7410

4/SN 7404  
6

1/SN 7408  
4

1/SN 7402  
4

\* Horloge PLI

2 SN74400  
4

\* Entrée sortie

2 SN 7475 PL II

4 SN 7475 PL III

\* Caractéristiques des circuits intégrés Utilisés

Tension d'alimentation  $S^V$

- Fan-out 10

- Une entrée en l'air correspondant au niveau logique "1"/

- Boitier enfichable TO 116

9°/ Simulation sur IBM 1130

9.1 : Organigramme : Nous traduisons le fonctionnement de notre système sous forme d'organigramme donné par la figure II. 5

L'organigramme se subdivise en deux parties l'une reposant sur le test de  $\Delta Y^*$  et l'autre sur celui de  $\Delta X^*$ . Les données sont  $\Delta Y^*$ ,  $\Delta X^*$ ,  $Q_Y$ , l'initialisation des registres R, Y et t du compteur décompteur.

Selon que  $\Delta Y^*$  est positif, négatif ou nul nous aurons :

- soit  $Y_N = Y_N + Q_Y$  dans ce cas s'il y a report, le calcul est terminé

- soit  $Y_N = Y_N - Q_Y$

- soit  $Y_N = Y_N$

De même pour  $\Delta X^*$  nous aurons

- soit  $R_N = R_N + Y_N$  s'il y a débordement ceci se traduit par un comptage

- Soit  $R_N = R_N - Y_N$  si  $R_N < Y_N$  il y aura décomptage.

- Soit  $R_N = R_N$

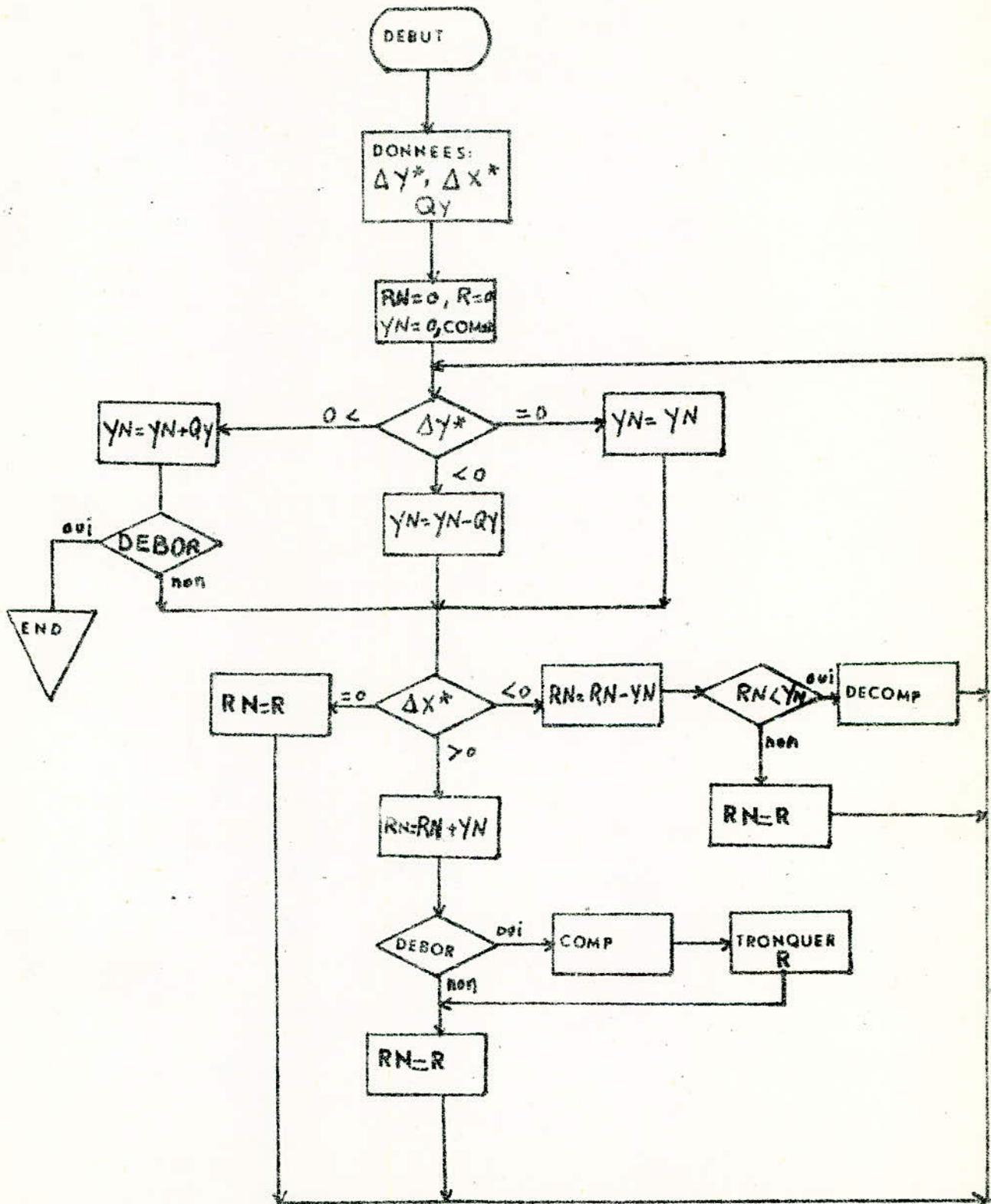


FIG II.5. ORGANIGRAMME-

## 9.2 PROGRAMMATION ET COMMENTAIRES

```

*SIMU L-DD A-MAYA *IMPRESSION DUTITRE
DEB  LDX  L1 MES1
      STX  L1 BAH
      BS1  L  IMPR
      LDX  L1 40 *MISE A BLANC DE LA ZONE D'IMPRESSION DES RESUL-
SUITE LD  L BLANC-TATS . -TATS.
      STO  L1 INCRY
      MDX  1 -1
      B    SUITE
DEBU  LDX  L1 MES *IMPRESSION DE TOUS LES RESULTATS D'UNE ITERATION
      STX  L1 BAH
      BSI  L  IMPR
      LDX  L1 INCRY
      LD   L  1
      STO  L  ZONE
      LD   L  Y *TRADUCTION ET TEST DE L'INCREMENT Y
      BSI  L  TRADU TRADUCTION EN CODE D'IMPRESSION .
      BP   L  ADD1 INCREMENT POSITIF FAIRE ADDITION 1
      BZ   L  CONS1 INCREMENT NUL CONSERVATION
      LD   L  Y *INCREMENT NEGATIF FAIRE SOUSTRACTION ,TRADUC-
      S    L  QY -TION ET STOCKER RESULTAT DANS ZONE D'IMPRES-
      STO  L  Y -SION
      LDX  L1 SOU1
      LD   L  1
      STO  L  ZONE
      BSI  L  TRADU
      B    L  ENCOR PASSER AU TEST DE L'INCREMENT X
CON1  LD   L  Y *CONSERVER LE RESULTAT ,LE TRADUIRE ET LE STOCKER
      STO  L  Y DANS LA ZONE D'IMPRESSION
      LDX  L1 CON1
      LD   L  1
      STO  L  ZONE
      LD   L  Y
      BSI  L  TRADU
      B    L  ENCOR PASSER AU TEST DE X
ADD1  LD   L  Y *FAIRE ADD1,TRADUIRE RESULTAT ET LE STOCKER
      A    L  QY DANS LA ZONE D'IMP RESSION,DEMANDER L'IMPRESSION
      DE TOUS LES RESULTATS STOCKES .

```

	STO	L	Y	
	LDX	L1	ADD1	
	LD	L	1	
	STO	L	ZONE	
	LD	L	Y	
	BSI	L	TRADU	
	BSI	L	IMPR	
	SRT	L	4	* Test de la retenue
	BNZ	L	FIN	Oui Brancher à fin programme
ENCOR	LDX	L1	INCRX	Traduire increment $\Delta X$ , le stocker dans zone d'impression, le tester
	LD	L	1	
	STO	L	ZONE	
	LD	L	ZINCR	
	BSI	L	TRADU	
	BF	L	ADD2	: Positif faire addition 2
	BE	L	CONS2	: Nul conserver le resultat
	LD	L	RN	* Negatif faire soustraction 2, preparer impression
	S	L	Y	
	STO	L	RN	
	LDX	L1	SCU2	
	LD	L	1	
	STO	L	ZONE	
	LD	L	RN	
	BSI	L	TRADU	* TEST de décomptage
	BN	L	DECOM	: Resultat de la soustraction negatif faire un décomptage de 1 unité
DECOM	B	L	DEBU	
	LD	L	COMPT	* Décomptage et préparation de l'impression du resultat de ce décomptage
	S	L	UN	
	STO	L	COMPT	
	LDX	L1	DECO	
	LD	L	1	
	STO	L	ZONE	
	LD	L	COMPT	
	BSI	L	TRADU	
	B	L	DEBU	: Passer à l'iteration suivante
CONS2	LD	L	RN	
	LDX	L1	CON2	
	LD	L	1	
	STO	L	ZONE	
	LD	L	RN	
	BSI	L	TRADU	
	B	L	DEBU	: Passer a l'iteration suivante
ADD2	LD	L	RN	* Addition 2
	A	L	Y	
	STO	L	R	
DEBOR	LD	L	R	* Test de debordement de l'addition 2.
	SRT	L	4	

	BE	NUL	Pas de debordement aller a nul
	AND	L MASK	Débordement
	SLT	4	
	STO	L R	* Comptage , et preparation a l'impression de ce comptage
	LD	L COMPT	
	A	L UN	
	STO	L COMPT	
	LDX	L1 COMP1	
	LD	L 1	
	STO	L ZONE	
	LD	L COMPT	
NUL	BSI	L TRADU	: Preparation de l'impression de addition 2 .
	LD	L R	
	STO	L RN	
	LDX	L1 AD2	
	LD	L 1	
	STO	L ZONE	
	LD	L RN	
	BSI	L TRADU	
	B	L DIEBU	: Passer a l'iteration suivante.
FIN	EXIT		
TRADU	DC	0	
	LIBF	BINDC	
	DC	BATA	
	LIBF	SPEED	
	DC	/ 0000	
ZONE	DC	BATA	
	DC	0	
	DC	6	
	B	I TRADU	BINAIRE → Code carte : sous-Programme
IMPR	DC	0	BINDC
	LIBF	PRINT1	Code carte → EBC DIC : SPEED
	DC	/2000	
BAH	DC	0	
	DC	ERROR	
	LIBF	PRINT 1	
	DC	/0000	
	B	*-3	
	B	I IMPR	Ces deux sous-programmes sont assemblés dans un même sous-programme TRADU que nous avons écrits nous même le Sous programme <u>IMPR</u> : Permet l'impression du résultats stocké dans la zone voulue.
MASK	DC	/0000	
RN	DC	0	
R	DC	0	
Y	DC	0	
COMPT	DC	0	
UN	DC	1	
QY	DC	/1	
YINCR	DC	/4	
YINCR	DC	4	

ERROR	DC	0
	B	I ERROR
BLANC	DC	/4040
MES1	DC	FMES1-*
	EBC	.YINR * XINCR * ADD1 * CONS1 * SOU.
	EBC	.1 * COMPT * ADD2 * CONS2 * SOUS2.
	EBC	. * DECOMT *.
FMES1	EQU	*
MES	DC	FMES-*
INCRY	BSS	3
	DC	/4040
INCRX	BSS	3
	DC	/4040
AD1	BSS	3
	DC	/4040
CON1	BSS	3
	DC	/4040
SOU1	BSS	3
	DC	/4040
COMP1	BSS	3
	DC	/4040
AD2	BSS	3
	DC	/4040
SOU2	BSS	3
	DC	/4040
DECO	BSS	3
	DC	/4040
CON2	BSS	3
	DC	/4040
FMES	EQU	*
BATA	BSS	6
	END	DEB

// XEQ

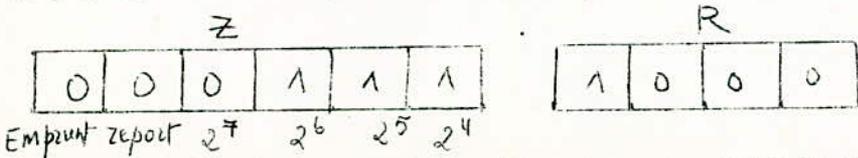
YINCR \* XINCR \* ADD1 \* CONS1 \* SOU 1 \* COMPT \* ADD2 \* CONS2 \* SOUS2 \* DECOMP \*

+00004		+00001		
+00004	+00004	+00001		+00001
+00004	+00004	+00002		+00001
+00004	+00004	+00002		+00003
+00004	+00004	+00003		+00003
+00004	+00004	+00003		+00006
+00004	+00004	+00004		+00006
+00004	+00004	+00004		+00010
+00004	+00004	+00005		+00010
+00004	+00004	+00005		+00015
+00004	+00004	+00006		+00015
+00004	+00004	+00006	+00001	+00005
+00004	+00004	+00007	+00001	+00005
+00004	+00004	+00007	+00001	+00012
+00004	+00004	+00008	+00001	+00012
+00004	+00004	+00008	+00002	+00004
+00004	+00004	+00009	+00002	+00004
+00004	+00004	+00009	+00002	+00013
+00004	+00004	+00010	+00002	+00013
+00004	+00004	+00010	+00003	+00007
+00004	+00004	+00011	+00003	+00007
+00004	+00004	+00011	+00004	+00002
+00004	+00004	+00012	+00004	+00002
+00004	+00004	+00012	+00004	+00014
+00004	+00004	+00013	+00004	+00014
+00004	+00004	+00013	+00005	+00011
+00004	+00004	+00014	+00005	+00011
+00004	+00004	+00014	+00006	+00009
+00004	+00004	+00015	+00006	+00009
+00004	+00004	+00015	+00007	+00008
+00004	+00004	+00016	+00007	+00008

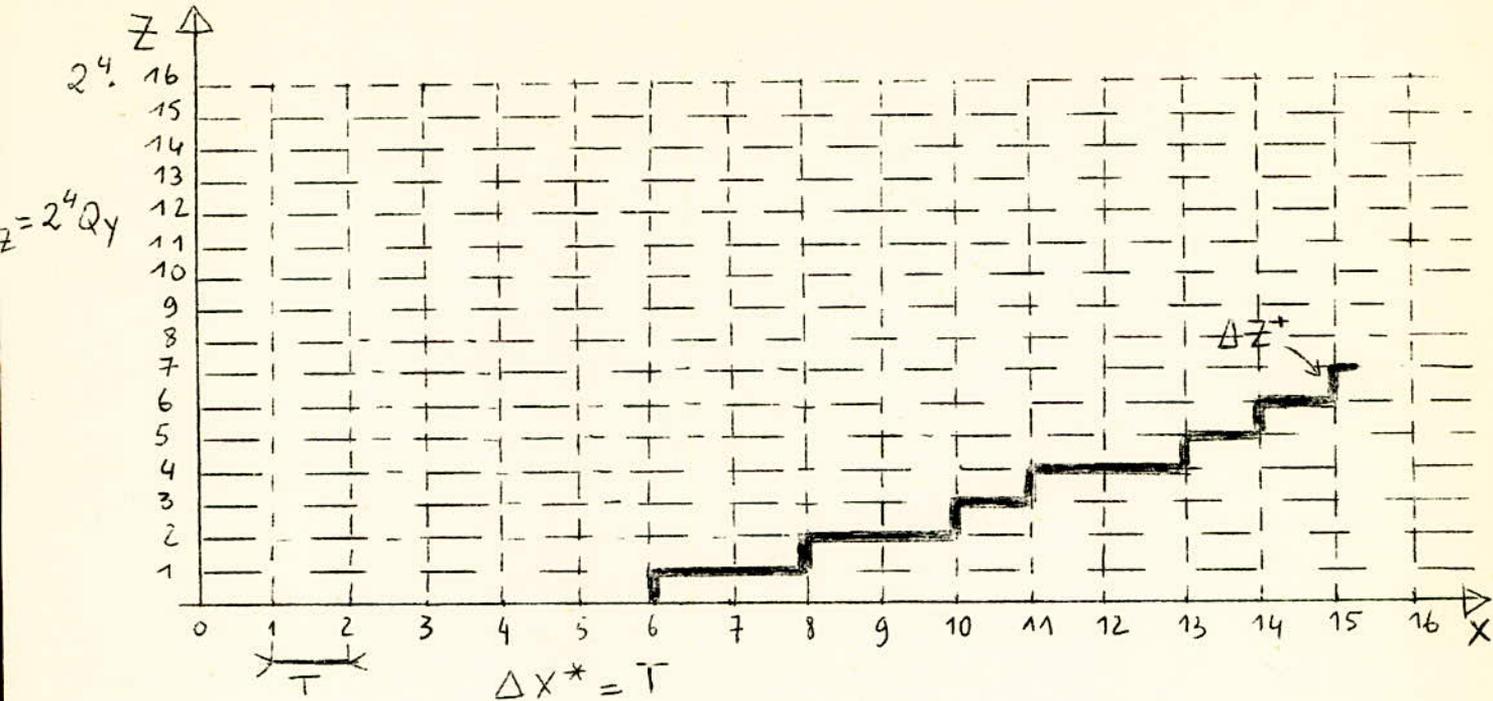
La programmation est faite en ASSEMBLEUR (langage machine). En introduisant comme données  $Q_y = 1$  (nombre de 4 bits),  $\Delta Y^* > 0$ ,  $\Delta X^* > 0$ , initialisation à zero le système s'est arrêté pour  $Y = 16$  et a donné 7 comptages et  $R = 1000$  (8) tel devra être le résultat de notre opérateur avec  $QY = 1$ , initialisation à zero et les commandes suivantes :

- $B_1 C_1$  addition
- $B_2 C_2$  addition

Il y a eu 7 increments positifs  $\Delta Z^+$  représentés par 7 comptages :



Si au lieu d'avoir un compteur -décompteur on avait un convertisseur incremental-analogique on aurait eu dans le cas de l'exemple programmé :



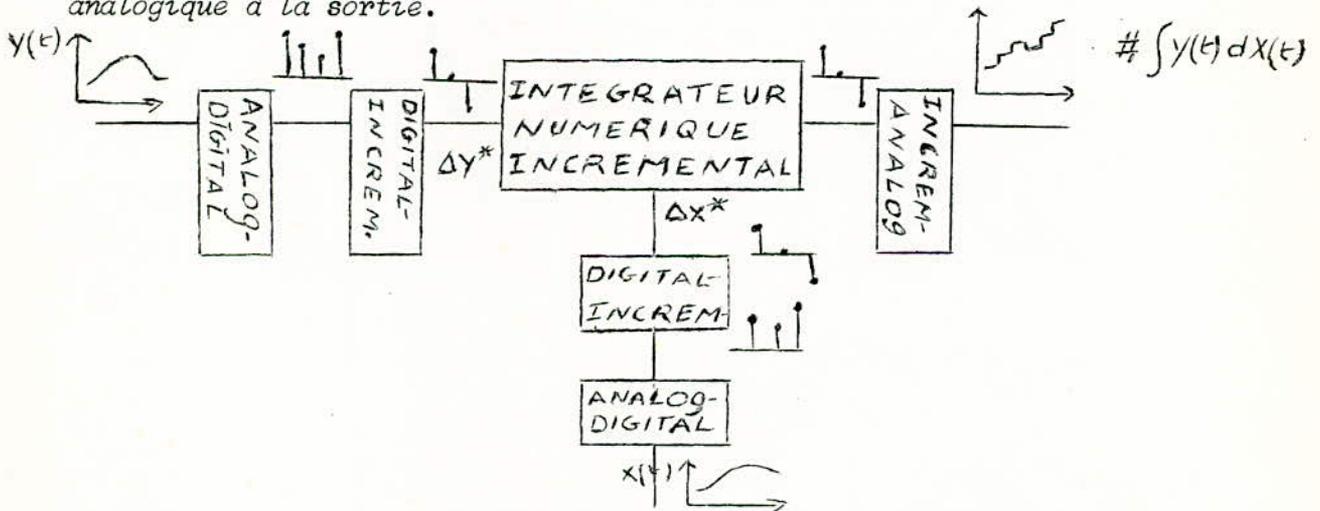
ce programme n'a en fait un intérêt réel que dans le cas où la commande de l'intégrateur se fait automatiquement selon l'évolution des fonctions  $Y(t)$  et  $X(t)$ . Il faut alors prévoir une logique d'attaque des commandes  $B_1, C_1$  et  $B_2, C_2$  et introduire des circuits complémentaires permettant de travailler avec  $Y$  et  $R$  en tant que nombres algébriques et non pas uniquement positifs.

--- Conclusion ---

Si d'un point de vue général l'organisation de calcul de l'intégrateur numérique incremental se fait autour de deux registres Y et R, néanmoins cette simplicité théorique ne se retrouve pas dans sa réalisation pratique. Cette dernière fut pourtant proposée de telle sorte qu'elle soit la plus simple possible tout en profitant de la souplesse d'utilisation des circuits intégrés TTL. Pour que toute fonction puisse être intégrée par rapport au temps ou une fonction du temps par la méthode incrementale il faudrait :

1°/ introduire des circuits complémentaires de correction ce qui permettra de travailler avec des nombres algébriques.

2°/ prévoir un convertisseur analogique-digital puis digital -incremental aux entrées  $\Delta Y^*$  et  $\Delta X^*$  et un convertisseur incremental analogique à la sortie.



Il semble bien qu'actuellement les performances des calculateurs incrementaux ne puissent être égalées dans les applications en temps réel où les DDA serie assurent en général, dans de bien meilleures conditions que ne peuvent le faire les calculateurs universels, le calcul des lignes trigonométriques d'un certain nombre d'angles mesurés par des capteurs. Lorsque vitesse et précision sont nécessaires (NASA) les DDA parallèles s'imposent. Cependant il est plus pratique pour les besoins courants de programmer un DDA sur une calculatrice Universelle. Elle place les valeurs initiales, charge les pistes programme et recueille les résultats lorsqu'ils ne sont pas destinés à une sortie immédiate sous forme incrementale.

BIBLIOGRAPHIE :

- Florine: Calculateurs incrémentaux-Revue d'automatisme-*  
*Monroe : Digital processes for sampled data systems*  
*H. Boucher: Machines informatiques - Hardware -*  
*J. Lehouchu: Mise en oeuvre du TMS 5700 Element principal*  
*d'analyseur différentiel numérique -Texas*  
*Instruments France -*

