

19/81

Université des Sciences et de la Technologie
Houari Boumediène

Ecole Nationale Polytechnique

Département Électricité

Projet de fin d'études: Ingéniorat en Électronique

ETUDE D'UNE U.A.L D'ADDITION
EN VIRGULE FLOTTANTE.

المدرسة الوطنية للعلوم الهندسية
المكننة

ECOLE NATIONALE POLYTECHNIQUE
BIBLIOTHEQUE

Proposé par: M^r B.CHEKHOVTSOV

Etudié par: M^{rs} HASNAOUI Mohamed
AITHAMMOU Mnd Oubélaïd

Promotion: Janvier 1981

By: H. A. H. H. H.

المدرسة الوطنية للعلوم الهندسية
بمصر
—
ECOLE NATIONALE D'INGENIEURIE
BIBLIOTHEQUE

REMERCIEMENTS



Nous tenons à remercier M^rB.CHEKHOVTSOV notre promoteur dont les conseils et les encouragements nous ont été très utiles dans l'élaboration de ce travail.

Nous remercions aussi M^r BEREKSI des conseils qu'il nous a prodigués. Nos remerciements vont aussi à tous ceux qui nous ont aidé de près ou de loin à l'élaboration de ce travail.

Que tous les professeurs de l'E.N.P.A et de l'U.S.T.H.B qui ont contribué à notre formation veuillent bien trouver ici l'expression de notre sincère gratitude.

Que tous ceux qui ont contribué au tirage et à la reproduction du présent ouvrage trouvent ici l'expression de notre profonde reconnaissance.

AIT HAMMOU M.O.

HASNAOUI M.

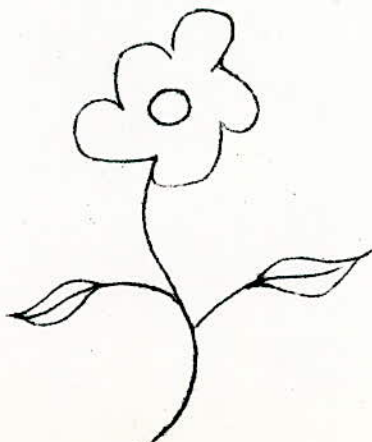
~ Dédicaces ~

- à mon père, à ma mère.
- à ma femme.
- à mes frères et sœur.
- à toute ma famille et amis.

AÏT HAMMOU. M. oubélaïd.

- à mon père, à ma mère.
- à ma grand'mère.
- à mes frères et sœurs.
- à toute ma famille et amis.

HASNAOUI Mohamed.



SOMMAIRE

I-INTRODUCTION:

II-REPRESENTATION DES NOMBRES DANS LES MACHINES CALCULATRICES:

2-1. Notion de base.

2-2. Le système binaire.

III-ARITHMETIQUE BINAIRE:

IV-ALGORITHME DE L'ADDITION EN VIRGULE FLOTTANTE:

1^a). Alignement des exposants.

2^o). Addition des mantisses.

3^o). Normalisation du résultat.

V-L'UNITE ARITHMETIQUE ET LOGIQUE:

5-1. Partie opératoire.

5-2. Etude et réalisation du séquenceur.

VI-METHODES DE SIMPLIFICATION DES CIRCUITS LOGIQUES:

6-1. Méthode des noeuds.

6-2. Codage rationnel des états.

VII-L'ORDINATEUR SIMULOG.

VIII-CONCLUSION:

IX-ANNEXE:

-Bibliographie.

I) Introduction:

Depuis fort longtemps l'homme a cherché à construire une machine capable de le remplacer pour effectuer ses calculs numériques.

Né il y a à peine une trentaine d'années, l'ordinateur a atteint aujourd'hui tous les domaines de la recherche scientifique, de l'industrie et même de la vie de tous les jours, s'imposant partout où l'on a besoin de calculer de gérer ou de prévoir.

Les ordinateurs traitent de l'information digitale, c'est à dire de l'information fondamentalement discontinue comme des chiffres ou des caractères alphabétiques.

Toutes les machines actuelles travaillent en binaire (système de numération à base 2) qui a été introduit au XVII^e siècle par LEIBNITZ.

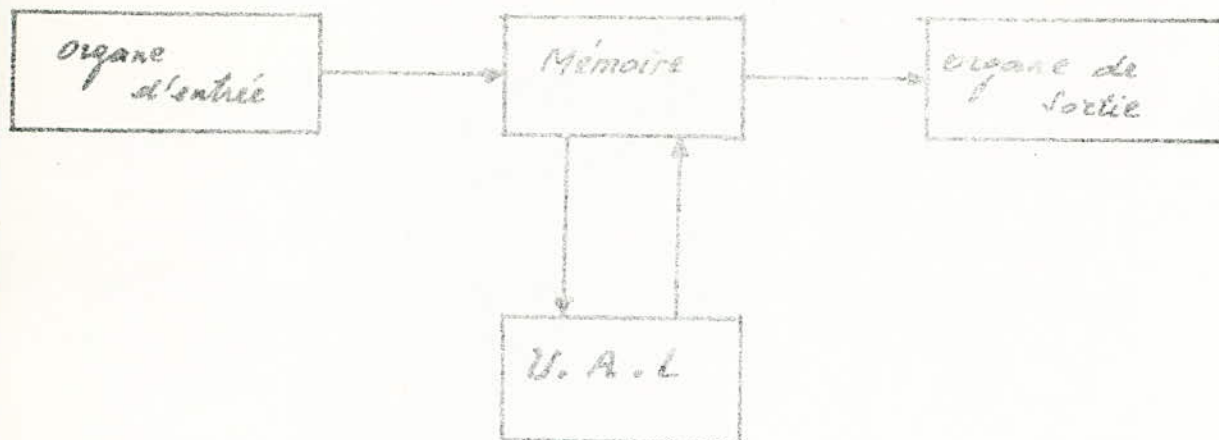
Sous la forme générale un ordinateur comprend:

- Un organe d'entrée pour l'introduction des opérandes et les symboles des opérations arithmétiques à effectuer. (Clavier, Lecteur de cartes, etc....).

- Un Organe de Calcul qui effectue les opérations arithmétiques et logiques demandées.

- Un Organe de sortie qui permet de communiquer avec l'extérieur. Il peut être une imprimante, un traceur de courbes, etc....

- La mémoire qui différencie un ordinateur d'une machine à calculer de bureau. Grâce à la mémoire, il est possible de n'utiliser qu'un seul organe de calcul et de conserver les résultats intermédiaires qui peuvent être nécessaires pour la suite des calculs.

Structure d'un ordinateur :

Notre travail se limite à l'étude d'une unité arithmétique et logique capable d'effectuer l'addition de 2 nombres binaires représentés en virgule flottante.

I. Représentation des nombres dans les machines calculatrices:

La numération décimale est employée et utilisée pour les travaux habituels, tandis que l'utilisation des machines digitales amène à travailler avec le système binaire ou système à base 2.

2.1. Notion de base:

Un nombre entier N s'exprime dans un système de numération à base B sous la forme suivante:

$$N = a_n B^n + a_{n-1} B^{n-1} + \dots + a_1 B^1 + a_0 B^0.$$

et s'écrit en numération de position:

$$a_n a_{n-1} a_{n-2} \dots a_1 a_0.$$

Un nombre fractionnaire compris entre 0 et 1 peut être représenté sous la forme suivante:

$$N = a_{-1} B^{-1} + a_{-2} B^{-2} + \dots \quad \text{Et s'écrit en numération}$$

de position sous la forme suivante:

$$0, a_{-1} a_{-2} a_{-3} \dots$$

La position " i " occupée par le symbole " a_i " indique le poids attaché à ce symbole. Par exemple dans le système décimal, a_0 représente le chiffre des unités (poids 1), a_1 représente le chiffre des dizaines (poids 10) et a_2 celui des centaines etc...

Les nombres écrits dans différentes bases utilisent les chiffres employés en base 10 augmentés de symboles nouveaux lorsque la base est plus grande que 10. Si la base est inférieure à 10, on prend le nombre de chiffres nécessaires en partant de zéro.

2.2. Le système binaire:

Introduit par LEIBNITZ au XVII^e siècle, le système binaire ou système à base 2 est la numération qui apparaît comme la plus évidente dans les machines à calculer électroniques. Les machines utilisent uniquement des systèmes à deux états d'équilibre (0 ou 1).

Cette restriction permet de faire correspondre aux valeurs "0" et "1" les états d'un commutateur "tout ou rien" (Un transistor fonctionnant en commutation).

Dans les machines calculatrices électroniques on utilise deux formes de représentation:

- La représentation en virgule fixe.
- La représentation en virgule flottante.

2.2.1. Représentation des nombres en virgule fixe:

C'est la façon la plus naturelle d'écrire un nombre binaire. Les nombres sont considérés comme des entiers, le soin de placer la virgule est laissé au programmeur. Le premier bit à gauche représente en général le signe. Conventionnellement on prend:

- "0" pour les nombres positifs.
- "1" pour les nombres négatifs.

a). Représentation des nombres positifs:



b). Représentation des nombres négatifs:

Les nombres négatifs peuvent être représentés dans l'une des trois conventions suivantes:

- signe et valeur absolue (code droit):



- Complément à 2 ou complément vrai:

Si un nombre binaire est négatif, on obtient son complément à 2 en prenant le complément de tous les bits de la valeur absolue et en ajoutant "1" au résultat.

$$[A]_{cv} = 2^n - A$$

n représente le nombre de bits de la valeur absolue.

Complément à 1 ou complément Restreint:

Si un nombre est négatif, on obtient son complément restreint en prenant le complément de tous les bits de la valeur absolue.

$$[A]_{CR} = (2^n - 1) - A$$

n représente le nombre de bits de la valeur absolue de A .

2.2.2. Représentation des nombres en virgule flottante:

La représentation en virgule flottante utilise deux champs mémoire, l'un contenant la mantisse et l'autre l'exposant. C'est donc une représentation semi-logarithmique.

Dans chacun de ces champs, la représentation des nombres se fait suivant les mêmes règles qu'en virgule fixe, c'est à dire:

- signe et valeur absolue.
- Complément à 2.
- Complément à 1.

Un nombre A représenté en virgule flottante s'écrit:

$$A = M \cdot B^E$$

où B représente la base du système considéré. Dans notre cas (système binaire), le nombre A sera représenté de la manière suivante:

$$A = M \cdot 2^E$$



signe



MANTISSE



signe



EXPOSANT

Si le premier bit de la valeur absolue de la mantisse est significatif (c'est à dire 1), on dit que le nombre est normalisé.

2.3. Comparaison des deux représentations:

a) Taille des nombres:

La gamme de représentation des nombres est plus grande en virgule flottante qu'en virgule fixe pour un même nombre de bits de registre. Pour illustrer ceci, considérons un registre de $(m+n)$ bits.

a.1. Cas de la virgule fixe:

Le plus petit nombre qu'on peut représenter

sera:

$$\pm, \underbrace{00 \dots 01}_{m+n \text{ bits}} = 2^{-(m+n)}$$

et le plus grand nombre:

$$\pm, \underbrace{11 \dots 11}_{m+n} = 2^{m+n} - 1$$

Donc en virgule fixe la gamme de représentation d'un nombre A est la suivante:

$$\boxed{2^{-(m+n)} \leq |A| \leq 2^{m+n} - 1} \quad \text{①}$$

a.2. Cas de la virgule flottante:

Dans ce cas, il faudrait que les nombres soient normalisés c'est à dire que: $\frac{1}{2} < |M| < 1$

Le plus petit nombre qu'on peut représenter aura pour mantisse:

$$M = \underbrace{0.10 \dots 0}_n = +2^{-1}$$

et pour exposant:

$$E = \underbrace{1.11 \dots 1}_m = -(2^m - 1)$$

Il sera égal à:

$$2^{-1} \cdot 2^{-(2^m - 1)} = 2^{-2^m}$$

Le plus grand nombre qu'on peut représenter aura pour mantisse:

$$M = 0.111 \dots 1 = 1 - 2^{-n}$$

et pour exposant:

$$E = 0.11 \dots 1 = +(2^m - 1)$$

Ce nombre sera égal à : $(1-2^{-n}) \cdot 2^{+(2^m-1)}$

Donc en virgule flottante la gamme de représentation d'un nombre A est la suivante :

$$2^{-2^m} \leq |A| \leq (1-2^{-n}) \cdot 2^{+(2^m-1)} \quad (2)$$

En comparant les relations (1) et (2), on voit bien que la taille des nombres qu'on peut représenter en virgule flottante est plus grande qu'en virgule fixe, d'où une plus grande précision des calculs en virgule flottante.

b.) Temps d'exécution des opérations arithmétiques :

Malgré les difficultés de la gestion de la virgule fixe qui amène à effectuer plusieurs opérations de décalage, les calculs en virgule fixe sont plus rapides qu'en virgule flottante où existent les problèmes de normalisation et d'alignement.

c.) Matériel :

Dans le cas de la virgule flottante, nous devons traiter séparément les mantisses et les exposants, d'où utilisation de beaucoup plus de matériel qu'en virgule fixe.

Malgré l'utilisation de beaucoup plus de matériel et le temps d'exécution relativement long, on préfère la représentation en virgule flottante lorsqu'une grande précision est exigée, ce qui est le cas des calculs scientifiques.

4. Arithmétique binaire :

4.1. Addition et soustraction de deux nombres dans le code vrai :

Pour additionner deux nombres dans le code vrai, on ajoute le deuxième nombre au premier, l'addition portant sur tous les bits y compris ceux de signe, et on laisse tomber les reports produits.

Pour soustraire deux nombres dans le code vrai, on ajoute le complément à 1 (complément restreint) du deuxième

nombre au premier nombre et on ajoute ensuite "1" au résultat. Tous les bits, y compris ceux de signe sont ajoutés, on laisse tomber les reports produits.

3.2. Addition et soustraction dans le code restreint.

En addition on ajoute le deuxième nombre au premier, les additions portent sur les bits y compris ceux de signe. Si un report se produit, on ajoute "1" au résultat.

En soustraction, on ajoute le complément restreint du deuxième nombre au premier, les additions portent sur tous les bits y compris ceux de signe. Si un report se produit, on ajoute "1" au résultat.

3.3. Circuits d'addition.

3.3.1. Semi-additionneur combinatoire.

On appelle semi-additionneur, un circuit électronique capable d'exécuter l'addition logique de deux bits. Il est défini par la table de vérité suivante, où a et b représentent les bits à additionner et s et r la somme et la retenue.

a	b	s	r
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Les équations de s et r sont lues directement sur la table de vérité; nous avons :

$$s = \bar{a}b + a\bar{b} = a \oplus b$$

$$r = ab$$

Le schéma logique est donné à la figure ①.

3.3.2. Additionneur complet combinatoire.

C'est un circuit permettant l'addition logique de deux bits compte tenu d'une retenue précédente. Il est défini par la table de vérité suivante; où a et b représentent les bits à additionner, r_{i-1} la retenue précédente

et s et r représentant la somme et la retenue présentes.

a	b	r_{i-1}	s	r
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

Les équations logiques de s et r sont :

$$s = (\bar{a}b + a\bar{b})\bar{r}_{i-1} + r_{i-1}(\bar{a}b + ab)$$

$$s = (a \oplus b)\bar{r}_{i-1} + r_{i-1}(a \oplus b)$$

$$s = a \oplus b \oplus r_{i-1}$$

$$r = ab + r_{i-1}(a+b)$$

Le schéma logique est donné à la figure ②.

3.3.3. Accumulateur totaliseur à 1 bit.

C'est un additionneur complet à 1 bit et 3 séquences utilisant une bascule du type T dont la table de vérité est donnée à la figure suivante où Q_1 est l'état futur :

Q_1	T	0	1
0	0	0	1
1	1	1	0

$$Q_1 = Q \oplus T$$

A l'aide de cette table nous remarquons que la bascule T change d'état à chaque "1" logique. C'est cette propriété qui sera utilisée pour réaliser l'additionneur Accumulateur à 1 bit de la figure ③.

L'addition de 3 bits s'effectuera en 3 cycles :

- pour $a_i=1$, $b_i=0$ et $r_{i-1}=0$, la bascule T passera à la première impulsion à "1" et le conservera après les 3 temps d'addition.

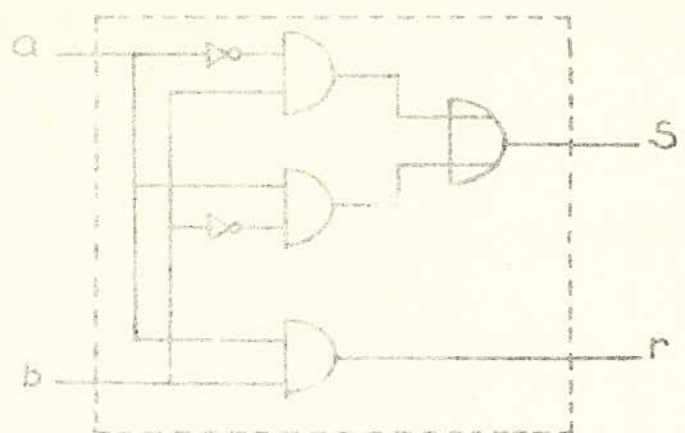
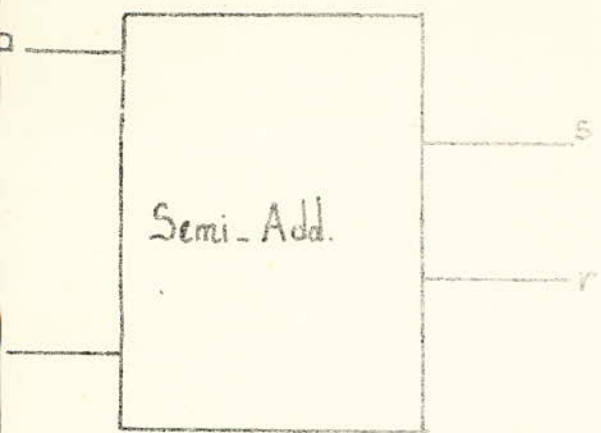


Fig1. Semi-additionneur.

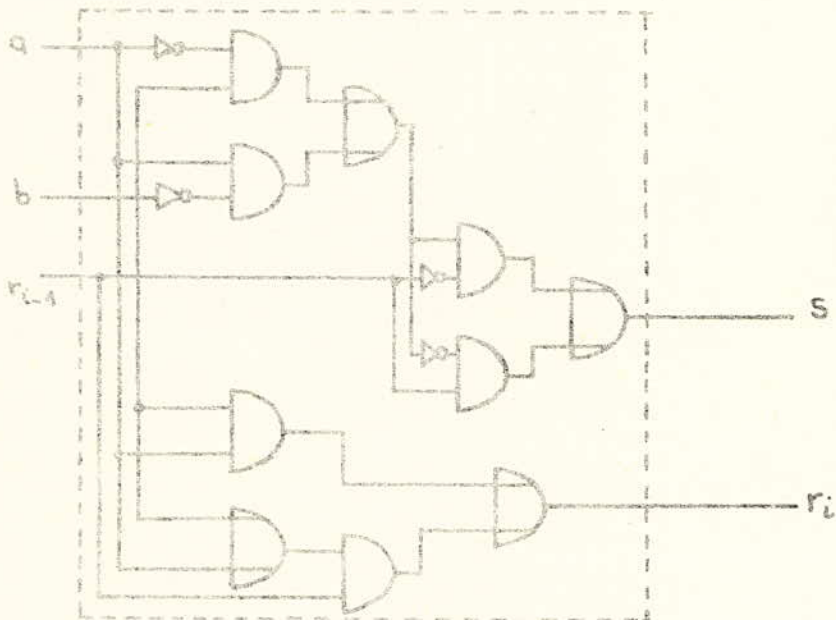
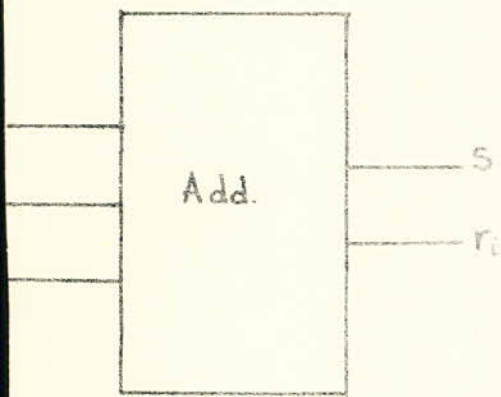


Fig2. Additionneur Complet combinatoire

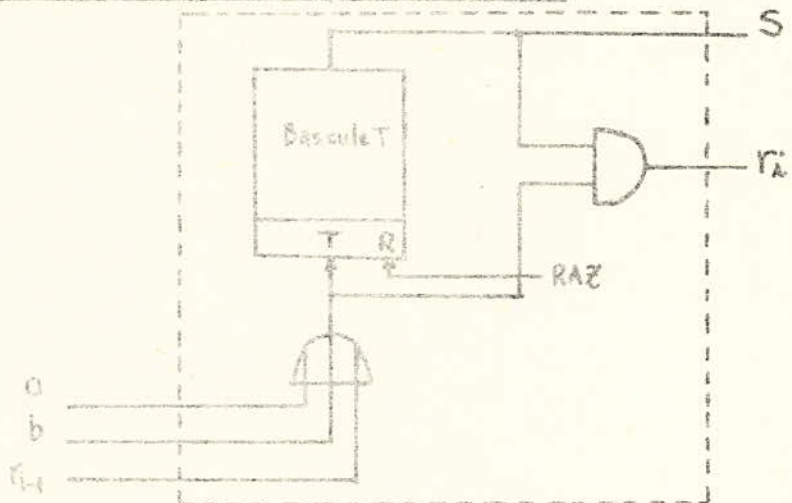


Fig3. Additionneur totaliseur à 1 bit et 3 sequences

- Si les bits à additionner sont: $a_i=1$, $b_i=1$ et $r_{i-1}=0$, la bascule T prendra après la 1^{ère} impulsion ($a_i=1$) l'état 1 et reviendra après la deuxième impulsion ($b_i=1$) à l'état zéro pour élaborer un signal de report à la sortie R_i (sortie de report).

- Si l'entrée de comptage de la bascule T reçoit simultanément toutes les trois impulsions ($a_i=b_i=r_{i-1}=1$), l'impulsion de report sera délivrée à la deuxième impulsion et la bascule T passera à l'état 1 à la troisième impulsion ($r_{i+1}=1$).

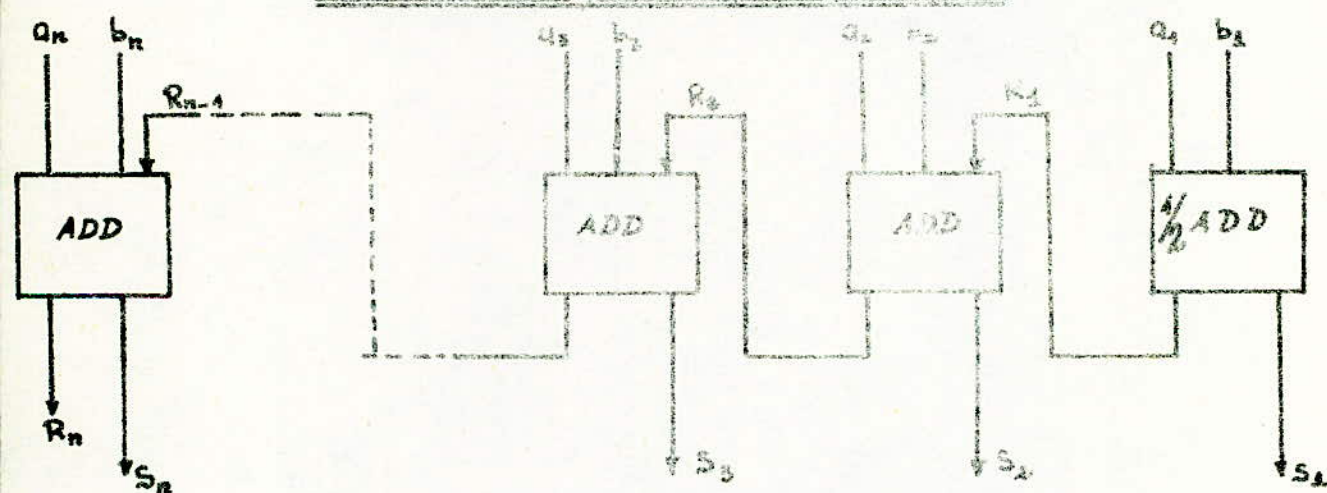
Il est évident que la rapidité de fonctionnement d'un tel additionneur est d'environ trois fois inférieure à celle d'un additionneur complet combinatoire qui mesure l'addition des trois bits pendant un seul cycle.

3.3.4. Additionneur parallèle à report sériel:

Il existe deux types:

- avec des additionneurs complets combinatoires.
- avec des accumulateurs totalisateurs.

a) Additionneur parallèle combinatoire:



On peut constater que les retards de la formation de la somme à la sortie du poids le plus fort de l'additionneur sera constitué pour un seul cas (le report du 1^{er} bit au n^{ème} bit) par la somme des retards introduits dans tous les bits précédents et le retard propre de formation de la somme.

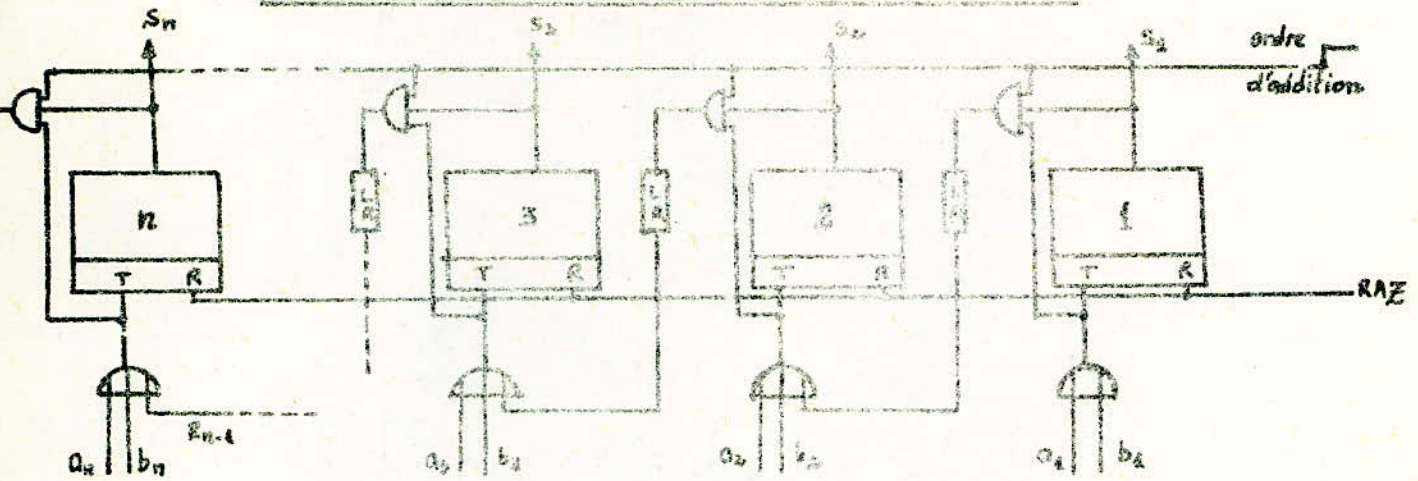
$$T = (n-1) T_R + T_s$$

T : retard maximal dans l'additionneur parallèle.

T_R = retard à la formation de la retenue.

T_S = retard à la formation de la somme.

b) Accumulateur totaliseur à plusieurs bits :



Les nombres à additionner doivent être présentés aux entrées parallèles des bits en séquence dans le temps, en assurant l'intervalle de temps séparant l'arrivée des 2 bits suffisant pour former le signal de report dans les bits de l'additionneur et le transmettre à travers les lignes à retard (RAZ). Le retard T_{LR} est réglé de manière à remplir la condition :

$$T_{LR} \geq T_i + T_r$$

où T_i est la durée maximale de l'impulsion de bit attaquant l'entrée du totaliseur et T_r est le retard maximum de basculement.

Cet accumulateur totaliseur est caractérisé par une faible rapidité de fonctionnement :

$$T = (T_{LR} + T_S)(n-1) + T_S$$

T : retard maximum de formation de la somme.

T_S : retard du signal dans la porte de formation du report.

La remise à zéro est obtenue par impulsion sur les entrées "R" des bascules.

3.3.5. Additionneur série :

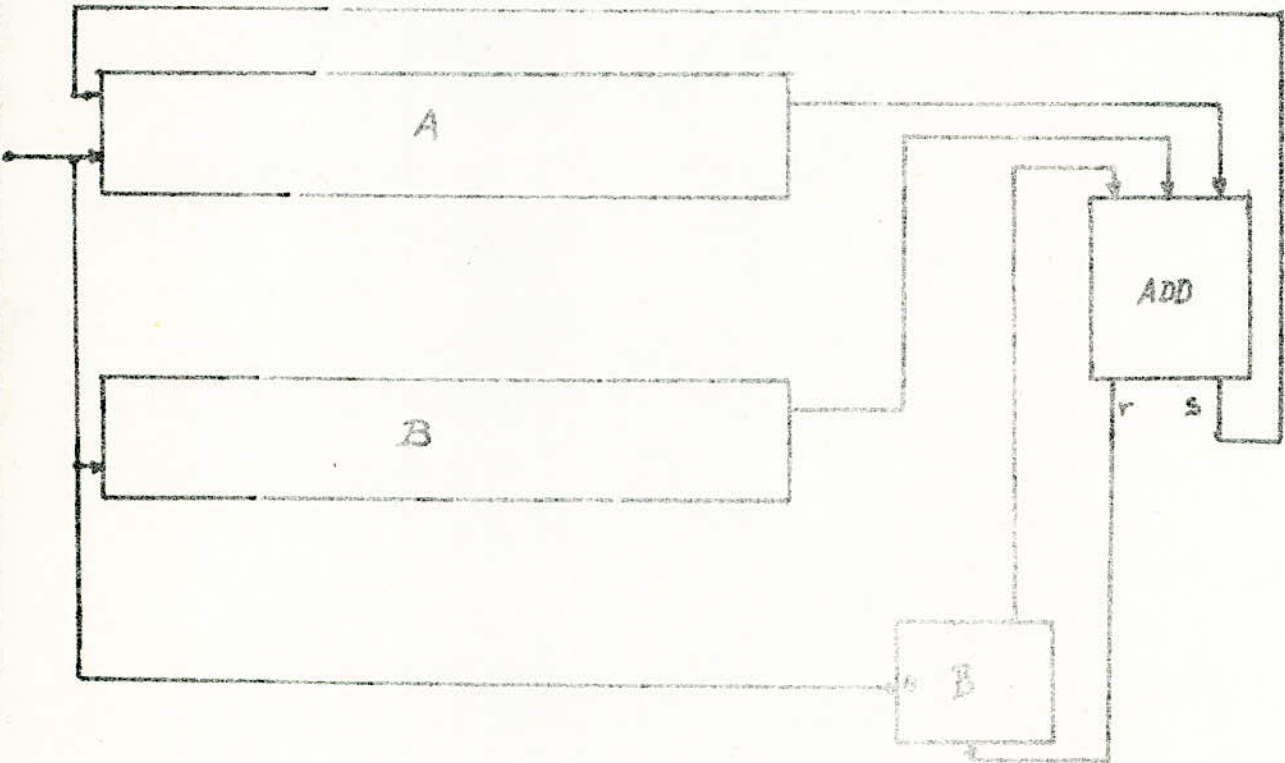
Dans un additionneur série, les chiffres des nombres à additionner parviennent rang par rang aux entrées d'un additionneur complet (RAZ). La somme et le report apparaissent

également rang par rang à ses sorties. Le report éventuel se doit être stocké pendant le temps d'une impulsion afin qu'il puisse en être tenu compte à l'addition du rang suivant.

L'addition de deux nombres binaires de n bits nécessite n impulsions.

Le schéma de principe est le suivant :

Les registres A et B sont des registres à décalage à droite.



Le registre A contient avant l'opération, l'un des opérandes. Après l'opération, il contient le résultat de celle-ci. Il a donc une double fonction. On le désigne souvent sous le nom d'Accumulateur.

IV. ALGORITHME DE L'ADDITION EN VIRGULE FLOTTANTE:

Soit à additionner deux nombres A et B représentés en virgule flottante, ayant respectivement pour mantisses M_A et M_B et pour exposant E_A et E_B .

L'opération se déroulera en trois phases:

- Alignement des exposants et dénormalisation de l'une des mantisses.

- Addition des mantisses.

- Normalisation du résultat.

1°) Alignement des exposants:

On fait la différence des exposants:

$$E_A - E_B = E.$$

Deux cas peuvent se présenter:

$$E = 0 \quad \text{ou bien} \quad E \neq 0.$$

- si $E = 0$. L'opération d'addition est exécutée en additionnant directement les mantisses et l'exposant du résultat sera E_A ou E_B .

- si $E \neq 0$: on a deux cas:

a) $E_A > E_B$. On dénormalise la mantisse M_B en la décalant à droite et en ajoutant 1 à E_B à chaque décalage d'1 position jusqu'à obtention $E_A = E'_B$.

b) $E_A < E_B$. On dénormalise la mantisse M_A en la décalant à droite et en ajoutant 1 à E_A à chaque décalage d'une position jusqu'à obtention $E'_A = E_B$.

à la fin de cette étape nous aurons à additionner deux nombres ayant pour mantisses M'_A et M'_B et pour exposant E'_A et E'_B .

2°) Exécution de l'opération d'addition:

Le résultat de l'addition aura pour mantisse

$$M_R = M'_A + M'_B$$

et pour exposant:

$$E_R = \text{Max}(E_A; E_B)$$

39) Normalisation du résultat:

Si l'exposant du résultat est dénormalisé on le décale à gauche jusqu'à normalisation et on retranche 1 à l'exposant du résultat à chaque décalage d'une position.

N.B.:

Après l'exécution de l'opération d'addition, le résultat peut dépasser la capacité ou devenir inférieur à la valeur minimale qu'on peut représenter. On dit qu'il y a débordement. Dans ces deux cas la machine émet un signal pour l'indiquer.

Cet algorithme est représenté sous forme de diagramme à la figure 5.

V. L'UNITÉ ARITHMÉTIQUE ET LOGIQUE:

Une unité arithmétique et logique (U.A.L.) est un circuit électronique qui réalise des opérations arithmétiques et logiques sur des opérandes binaires appliqués à ses entrées.

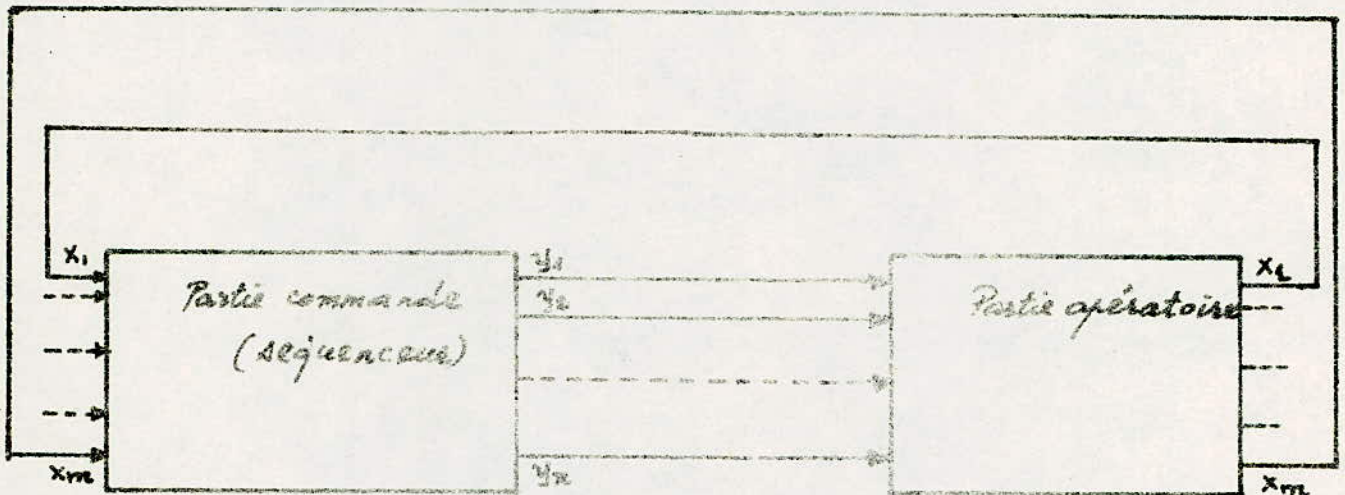
Elle est caractérisée par la taille du mot binaire qu'elle traite et le nombre d'opérations différentes qu'elle exécute. La sélection se fait par des commandes extérieures.

L'Unité Arithmétique et logique comporte deux parties:

a). La partie opératoire qui exécute les opérations arithmétiques et logiques (addition, décalages etc...). Elle comporte un accumulateur totaliseur et des registres pour stocker les opérandes.

b). La partie commande ou séquenceur dont le but principal est de produire des signaux destinés à positionner les circuits pour faire exécuter les différentes phases de l'instruction.

Schématiquement on peut représenter l'U.A.L. de la manière suivante:



Les " y_i " sont les signaux de commande et les " x_i " sont les signaux d'état.

L'élément de traitement de l'information est la micro-opération. Par exemple: Addition, décalage, transfert etc...

Le traitement de l'information se fait par un signal de commande. A chaque micro-opération correspond un signal de commande y_i . L'ensemble des micro-opérations qui peuvent être exécutées en même temps sont appelées micro-instructions. (Y_i)

Le passage d'une micro-instruction à une autre se

défini par la fonction de transfert d_{it} telle que :

$$Y_i \xrightarrow{d_{it}} Y_t$$

d_{it} est une fonction booléenne qui tient compte des deux conditions :

a) - Condition d'Orthogonalité :

$$d_{ij} \cdot d_{jk} = 0 \quad \text{si } j \neq k.$$

Celle-ci montre qu'on peut réaliser qu'une seule micro-instruction à un instant donné.

b) - Condition de plénitude : qui signifie qu'elle sera réalisée sans faute.

$$\sum_{i,t=1}^n d_{it} = 1.$$

Ceci montre que l'ensemble des micro-instructions et des fonctions de transfert nous donne un micro-programme de traitement de l'information de l'ordinateur.

5.1. Partie opératoire :

Le schéma synoptique de la partie opératoire est donné à la figure 4.

La partie opératoire comporte :

- Deux accumulateurs totalisateurs pour traiter séparément les mantisses et les exposants.

- Le premier nombre de mantisse M_1 et d'exposant E_1 se trouve dans les accumulateurs ACCM et ACCE.

Le second nombre à additionner se trouve dans les registres RegM₂ et RegE₂. Quand on réalise la différence et l'alignement des exposants, l'ACCE qui contenait l'exposant E_1 contiendra après l'opération le résultat $E_1 - E_2$. Donc on risque de perdre l'exposant E_1 . Pour remédier à cela, on utilise un registre auxiliaire RegE₁ pour mémoriser l'exposant E_1 avant l'exécution de la différence et de l'alignement des exposants.

A. Accumulateur pour le traitement des exposants et registres :

Le schéma est donné à la figure 5.

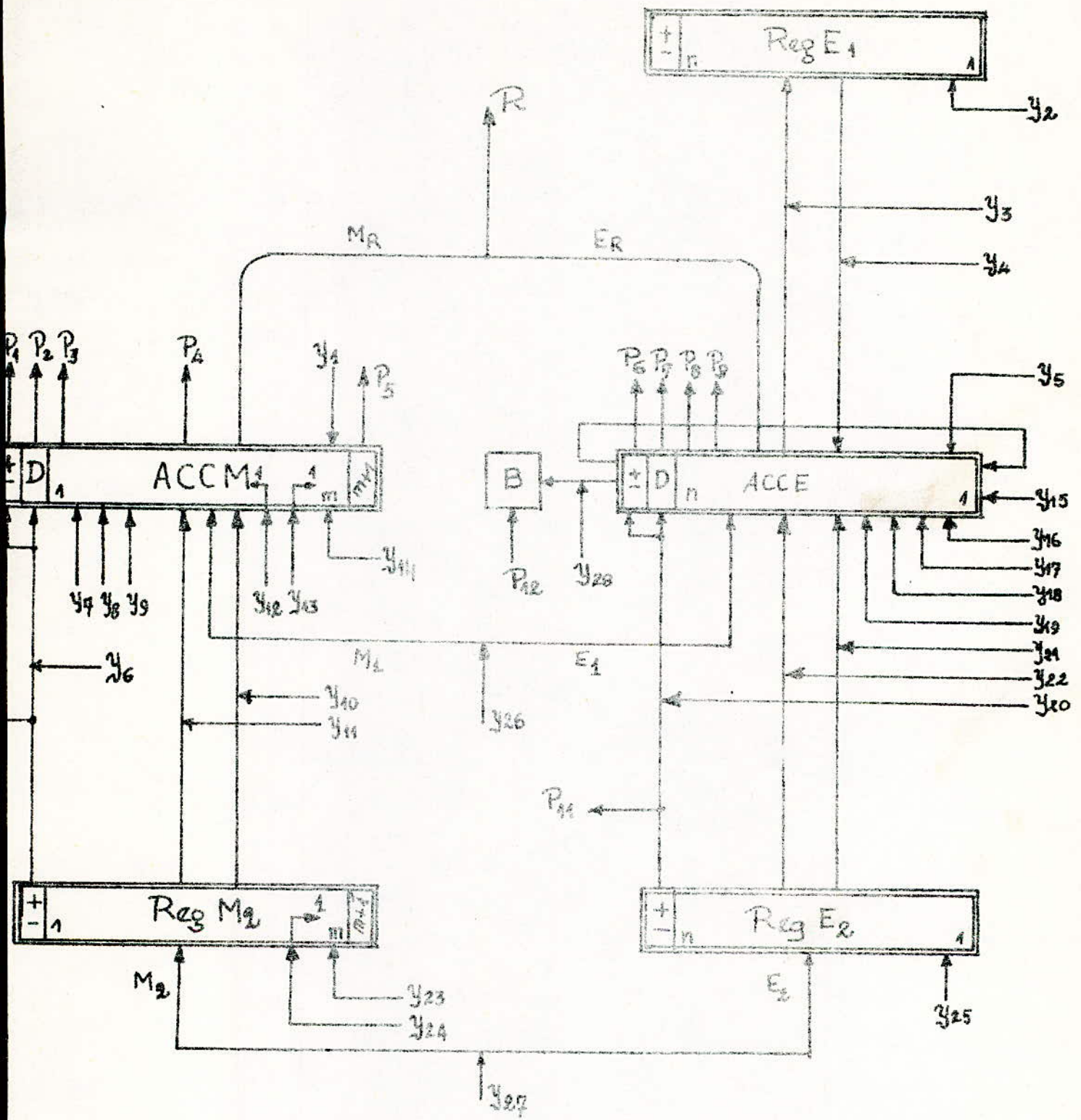


Fig 4. Schéma synoptique de la partie opératoire de l'unité arithmétique et logique d'addition en virgule flottante.

Liste des micro-opérations: μ_i et des tests: P_i

$\mu_1: ACCM[m] := 1$

$\mu_2: Reg E_1 := 0$

$\mu_3: Reg E_1 := ACCE$

$\mu_4: ACCE := Reg E_1$

$\mu_5: \text{Addition en ACCE.}$

$\mu_6: \text{Signe ACCM} := \text{signe Reg M}_2$

$\mu_7: ACCM[1] := 1$

$\mu_8: ACCM := 0$

$\mu_9: ACCM := [ACCM]_{inv}$

$\mu_{10}: ACCM := Reg M_2$

$\mu_{11}: ACCM := [Reg M_2]_{inv}$

$\mu_{12}: ACCM = G_1(ACCM).$

$\mu_{13}: ACCM = D_1(ACCM).$

$\mu_{14}: \text{Addition en ACCM}$

$\mu_{15}: ACCE := ACCE + 1$

$\mu_{16}: ACCE := ACCE - 1$

$\mu_{17}: ACCE := 0$

$\mu_{18}: ACCE := [ACCE]_{inv}$

$\mu_{19}: ACCE := -E_{max}$

$\mu_{20}: \text{Signe ACCE} := \text{Signe Reg E}_2$

$\mu_{21}: ACCE := |Reg E_2|$

$\mu_{22}: ACCE := [Reg E_2]_{inv}$

$\mu_{23}: Reg M_2 := 0$

$\mu_{24}: Reg M_2 := D_1(Reg M_2)$

$\mu_{25}: Reg E_2 := 0$

$\mu_{25}: ACCM := M_1 ; ACCE := E_1$

$\mu_{27}: Reg M_2 := M_2 ; Reg E_2 := E_2$

$\mu_{28}: B := \text{Signe ACCE}$

$P_1 = \text{Signe ACCM}$

$P_2 = ACCM[D]$

$P_3 = ACCM[1]$

$P_4 = 1 \text{ si } ACCM = 0$

$P_5 = ACCM[m+1]$

$P_6 = \text{Signe ACCE}$

$P_7 = ACCE[D]$

$P_8 = 1 \text{ si } ACCE = 0$

$P_9 = 1 \text{ si } |ACCE| > m$

$P_{10} = \text{Signe Reg M}_2$

$P_{11} = \text{Signe Reg E}_2$

$P_{12} = B$

$\mu_{29} = \text{Signal de débordement}$

L'Accumulateur totaliseur est du type parallèle à report série. Il est réalisé à l'aide de bascule "T". Il nécessite moins de matériel que l'additionneur du type combinatoire.

L'accumulateur totaliseur sera bouclé sur lui même pour ajouter une retenue éventuelle car nous travaillons en code restreint.

- La remise à zéro de l'accumulateur se fait par application de l'impulsion (y_{17}) sur les entrées "Reset" des bascules.

- Etant dans un registre mémoire, l'exposant E_1 sera chargé dans l'ACCE par application d'une impulsion de commande (y_{26}).

- Le passage en code restreint du contenu de l'ACCE est obtenu par l'impulsion de commande (y_{18}) qui appliquée aux entrées "T" des bascules fait changer l'état de celles-ci.

- En appliquant les signaux de commande y_5 et y_{20} ou y_{22} nous faisons l'addition du contenu de l'ACCE avec celui du registre $RegE_2$ en code droit ou en code restreint.

- Pour ajouter "1" au contenu de l'ACCE, on applique simultanément une impulsion y_{15} sur l'entrée "T" de la dernière bascule et une impulsion y_7 qui réalise l'addition.

- Pour retrancher "1" au contenu de l'ACCE, il suffit d'ajouter "1" en code restreint ce qui revient à appliquer l'impulsion y_{16} sur les entrées T de toutes les bascules sauf la dernière et d'appliquer le signal y_7 qui réalise l'addition.

- Le signe de l'ACCE est mémorisé dans le Bistable B par application de l'impulsion y_{20} .

- Le contenu de l'ACCE est transféré dans le registre $RegE_1$ par application du signal y_3 .

- La remise à zéro de $RegE_1$ se fait en appliquant une impulsion y_2 sur les entrées "Reset" des bascules.

- Le transfert du contenu du registre $RegE_1$ dans l'ACCE se fait en appliquant une impulsion de commande y_{14} .

- La remise à zéro du registre $RegE_2$ se fait en appliquant une impulsion y_{25} sur les entrées "Reset" de toutes les bascules.

- Chargement du $RegE_2$ en parallèle par l'exposant E_2

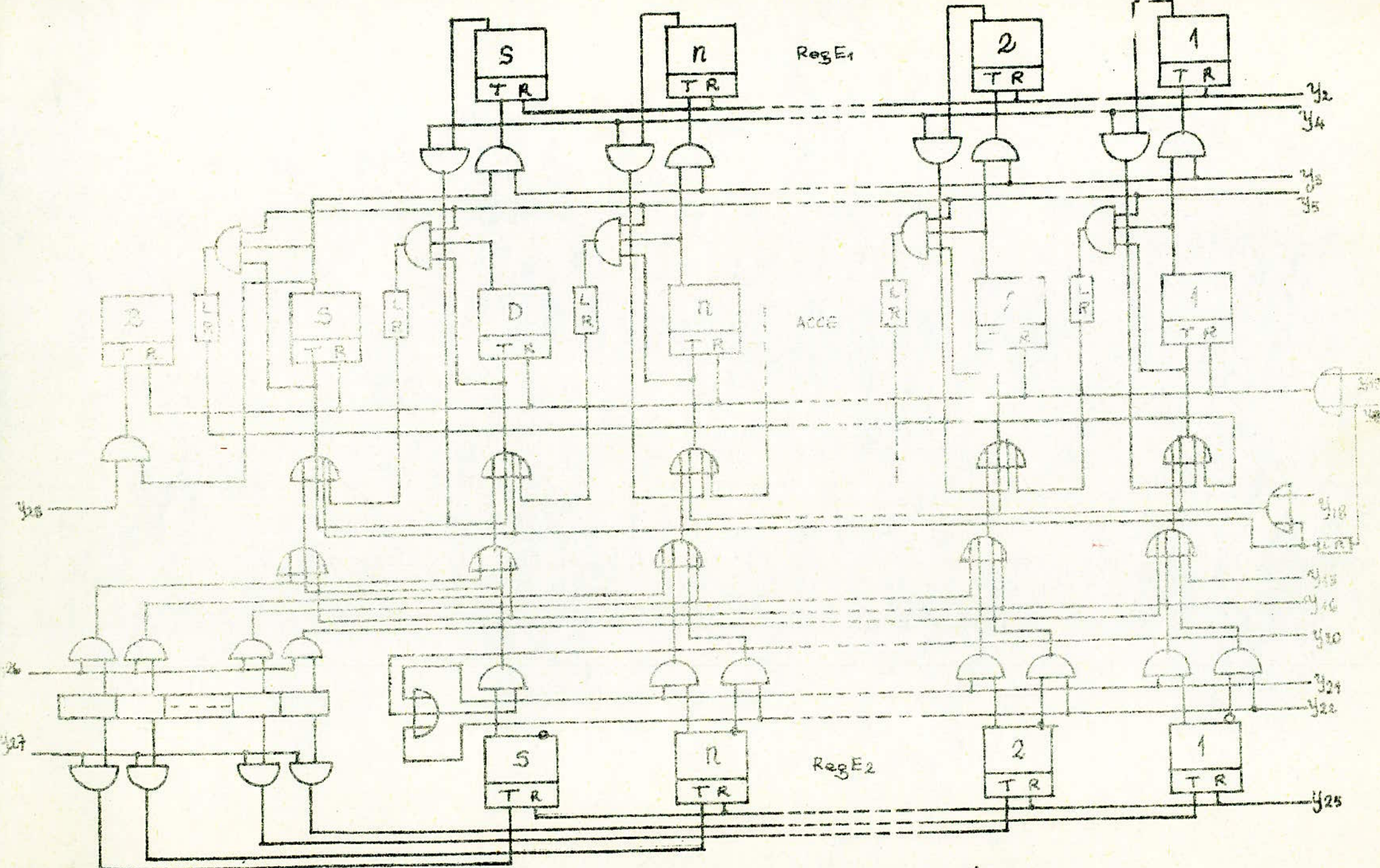


Fig 5 . Partie opératoire pour traiter les exposants.

contenu dans un module mémoire.

- Possibilité de transférer le contenu ou le contenu inversé du Registre E_2 dans l'ACCE par les impulsions Y_{51} ou Y_{52}
- Possibilité de transférer le signe des Registres E_2 dans la bascule de signe de l'ACCE par l'impulsion Y_{50} .

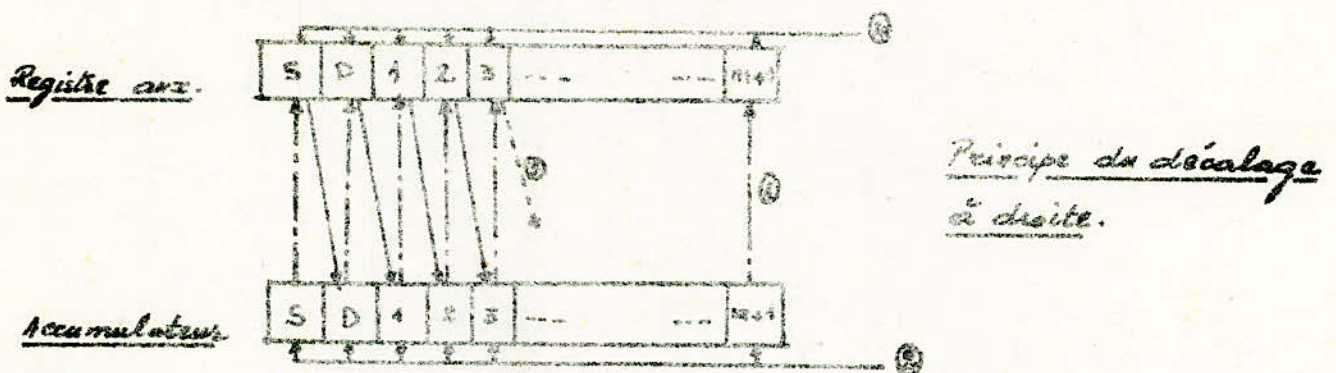
B. Partie traitant les mantisses:

Le schéma est donné à la figure 5 bis.

Nous utiliserons un accumulateur totaliseur à $m+1$ bits pour prévoir l'arrondissement du résultat.

Puisque nous ne pouvons pas réaliser des registres à décalage à droite ou à gauche à l'aide de bascules T, nous utiliserons un registre auxiliaire. Dans ce cas là le décalage à droite ou à gauche se fera en 4 étapes:

- 1°) - Transfert du contenu de l'accumulateur dans le registre auxiliaire.
- 2°) - Remise à zéro de l'accumulateur.
- 3°) - Transfert du contenu du registre auxiliaire (Décalé d' k position soit à droite ou bien à gauche) dans l'accumulateur.
- 4°) - Remise à zéro du registre auxiliaire.



- La remise à zéro de l'accumulateur se fait par l'application de l'impulsion Y_8 sur les entrées "Reset" des bascules.

- Avant l'opération la mantisse du 2^e opérande se trouve dans un registre mémoire, son chargement dans l'ACCUM se fait à l'aide de l'impulsion Y_{16} .

- Le décalage à gauche du contenu de l'ACCM se fait à l'aide de l'impulsion Y_{96} .

- Le décalage à droite de la position du contenu de l'ACCM se fait à l'aide de l'impulsion Y_{93} .

Dans ces deux cas le premier bit de l'ACCM doit être égal à 1. Ceci est assuré par l'impulsion Y_9 .

- Le registre M_2 est un registre à décalage à droite. Il peut être réalisé à l'aide de bascules M ou R . Son chargement par la mantisse du deuxième opérande se fait à l'aide du signal Y_{27} .

- Le contenu du registre M_2 est décalé à droite d'une position par l'application du signal Y_{24} .

- Le registre M_2 est mis à zéro par application de l'impulsion Y_{23} sur les entrées "Reset" de toutes les bascules.

- On peut transférer le contenu du Registre M_2 dans l'ACCM à l'aide de l'impulsion Y_{10} .

- Le contenu inversé du registre M_2 est transféré dans l'ACCM à l'aide de l'impulsion Y_{11} .

- L'addition en accumulateur M est obtenue par application du signal Y_{14} .

- Le signe du registre M_2 est transféré dans la bascule de signe de l'ACCM à l'aide de l'impulsion Y_8 .

- Le bit d'ordre (m) de l'ACCM est forcé à "1" à l'aide du signal Y_1 .

Dans la figure 5 bis, nous n'avons pas montré comment se fait le chargement de l'ACCM et du registre M_2 par manque de place, mais ce chargement est réalisé de la même manière que pour l'ACC et le registre E_2 , c'est-à-dire à l'aide des signaux Y_{56} et Y_{57} . (fig 5).

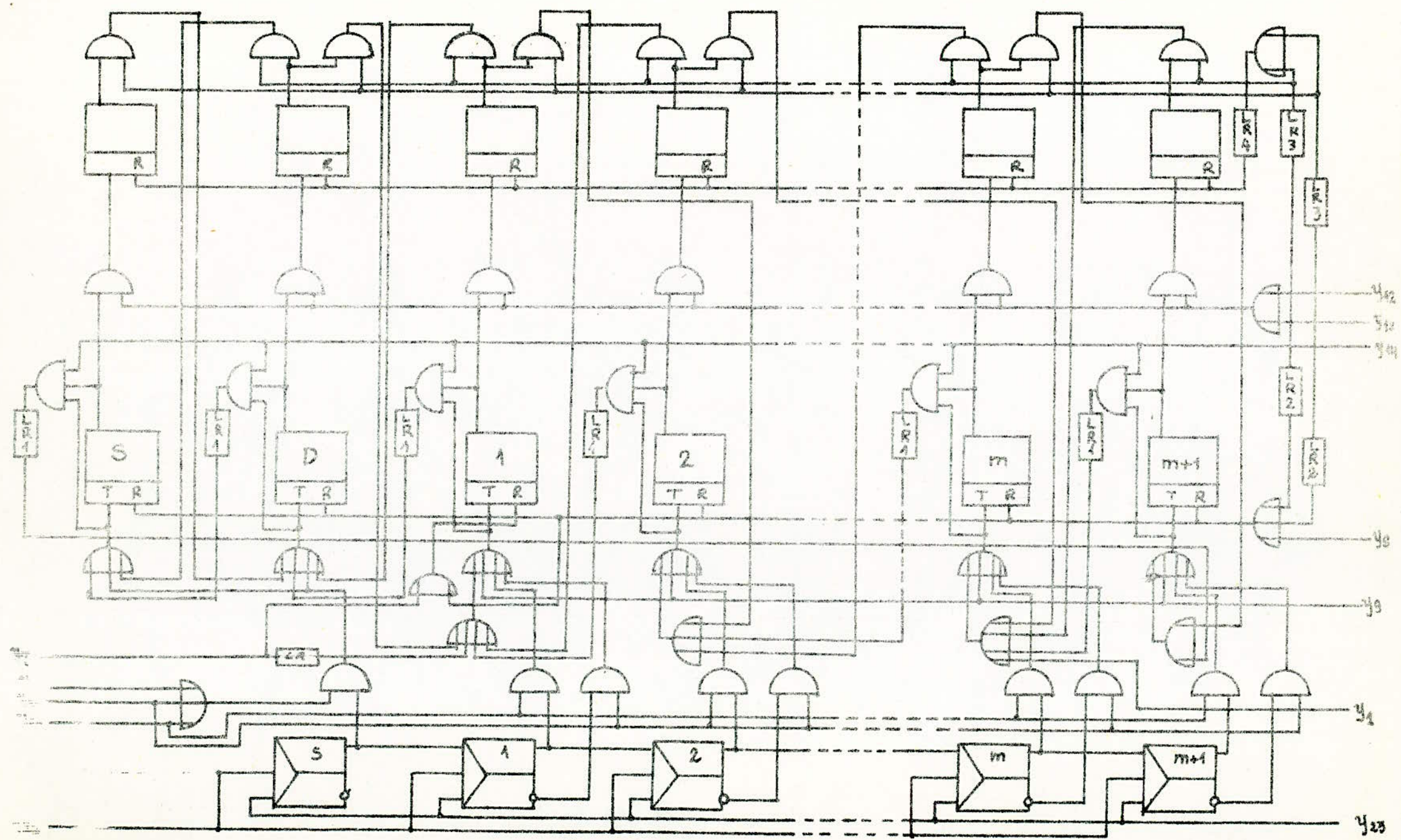


Fig 5bis Partie opératoire pour traiter les mantissas.

ion des signaux d'état "X_i":

Pour bien illustrer ceci, nous avons utilisé des accumulateurs et des registres de 8 bits.

- X₁ = P₆: signe de l'ACCE.

Ce signal est disponible à la sortie "Q" de la bascule de signe de l'ACCE.

- X₂ = P₁₁: signe du registre E₂. Ce signal est disponible à la sortie "Q" de la bascule de signe du reg E₂.

- X₃ = P₉ = 1 si ACCE > m = 8 (nombre de bits de l'ACCM).

ACCE > 8 ⇒ P₉ = 1

ACCE < 8 ⇒ P₉ = 1

h	g	f	e	d	c	b	a
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	1
0	0	0	0	0	1	1	0
0	0	0	0	0	1	1	1

On aura:

P₉ = hgfē (dcbā + dcb̄ā + dēbā + dēb̄ā + d̄cb̄ā + d̄cb̄ā + d̄cb̄ā + d̄cb̄ā)

P₉ = hgfē (dēb̄ + d̄ēb̄ + d̄cb̄ + d̄cb̄)

P₉ = hgfē (d̄ē + d̄c) = hgfēd̄ ⇒ **P₉ = h+g+f+e+d**

où h, g, f, e et d sont les états des cinq dernières bascules de l'ACCE.

- X₄ = P₈ = 1 si ACCE = 0

⇒ P₈ = hgfēd̄c̄b̄ā = P₉ · c̄b̄ā

- X₅ = P₁₂ = Etat de la Bascule B (B: bascule qui sert à mémoriser le signe de l'ACCE pour les calculs intermédiaires).

Ce signal P₁₂ = X₅ est disponible à la sortie "Q" de la bascule B.

- X₆ = P₁ = signe ACCM: Disponible à la sortie "Q" de la bascule de signe de l'ACCM.

- X₇ = P₁₀ = signe RegM₂: Disponible à la sortie "Q" de la bascule de

Signe du registre M₂.

- $X_9 = P_5$ = Etat de la 9^{ème} Bascule de l'ACCM. Disponible à la sortie "Q" de celle-ci.

- $X_{10} = \bar{P}_1 P_2 + P_1 \bar{P}_2 = P_1 \oplus P_2$.

P_1 : état de la bascule de signe de l'ACCM

P_2 : état de la bascule de débordement de l'ACCM

- $X_{10} = \bar{P}_6 P_7$.

P_6 : état de la Bascule de signe de l'ACC

P_7 : état de la bascule de débordement de l'ACC

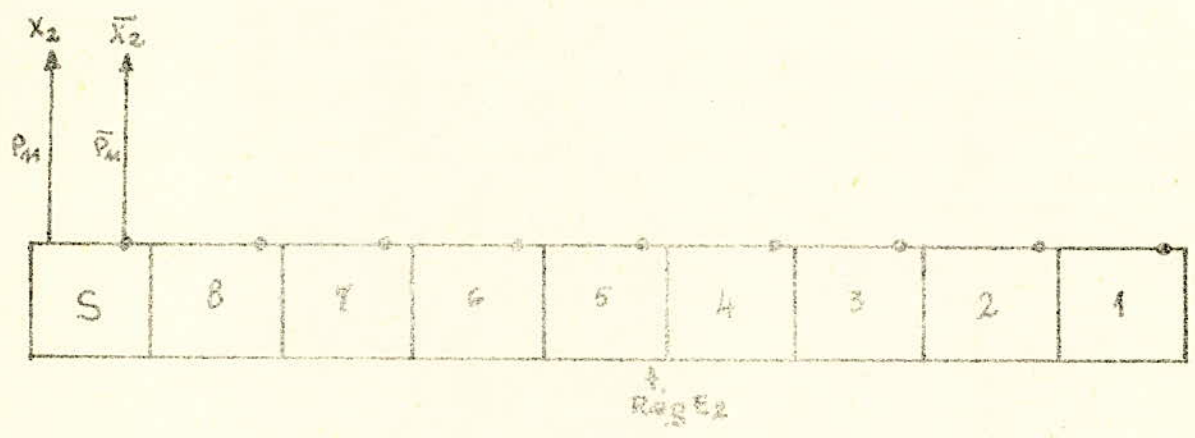
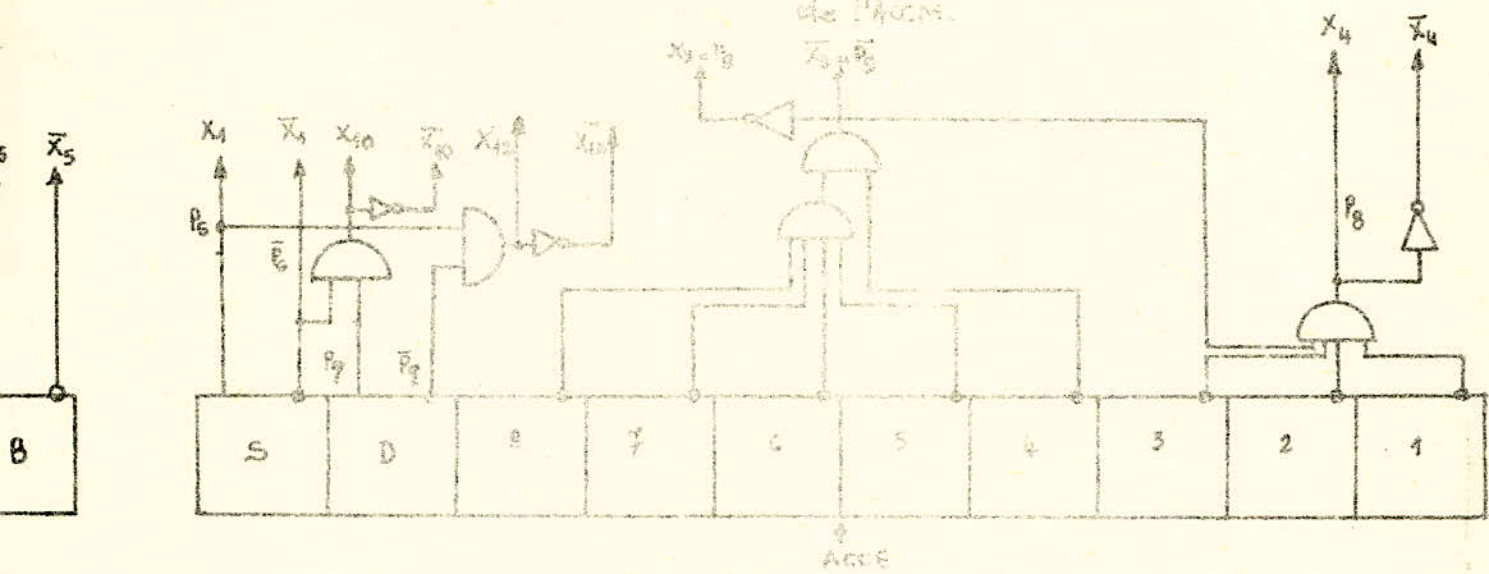
- $X_{11} = P_3$ = Etat de la première Bascule de l'ACCM

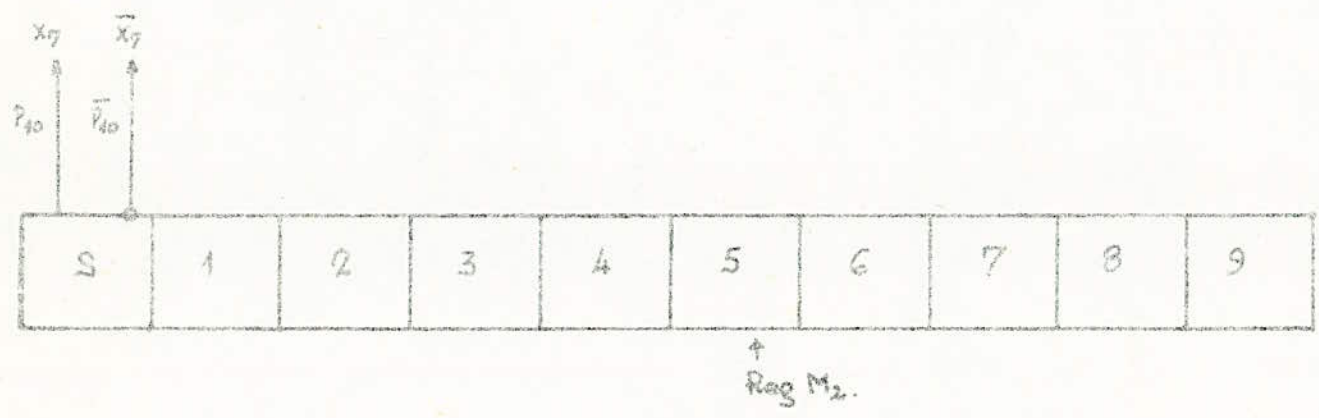
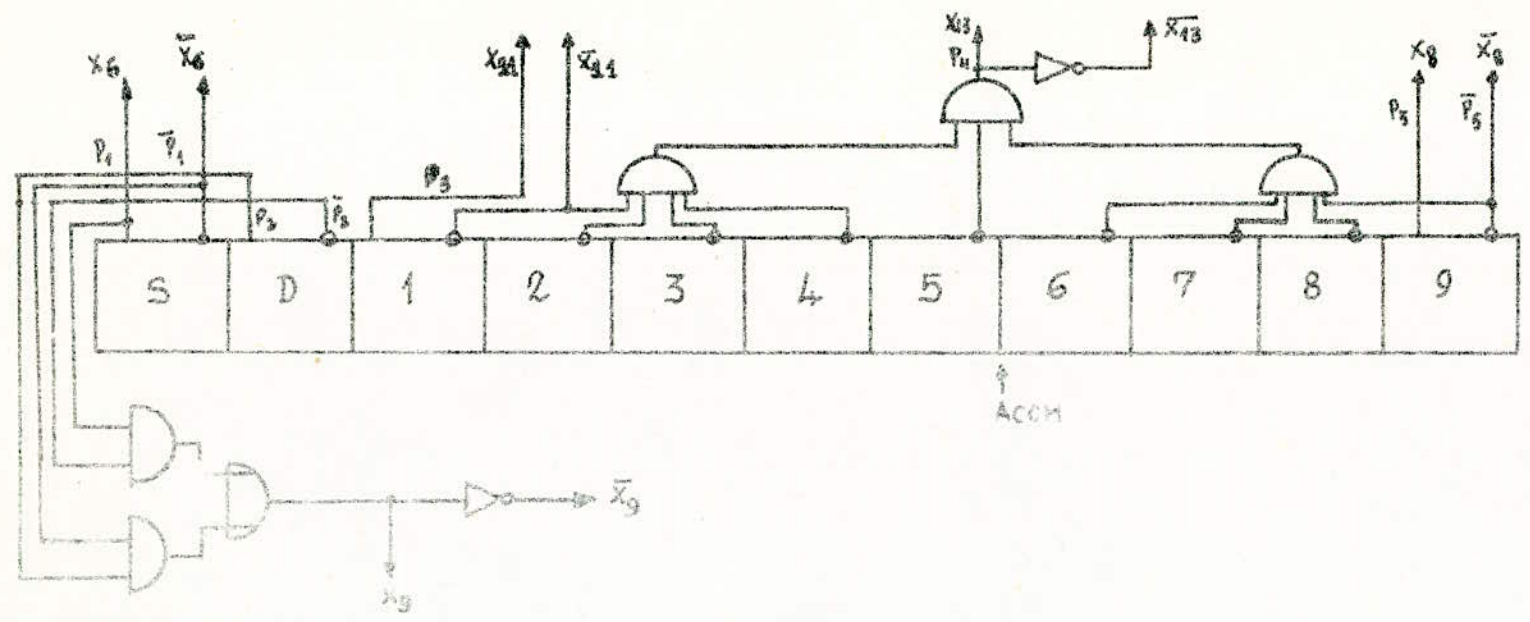
- $X_{12} = P_6 \bar{P}_7$.

- $X_{13} = P_4 = 1$ si ACCM = 0

$$P_4 = \bar{k}' \bar{k}' \bar{g}' \bar{f}' \bar{e}' \bar{d}' \bar{c}' \bar{b}' \bar{a}'$$

où a', b', c', d', e', f', g', h', k' représentant des états des bascules de l'ACCM.





5.2. Etude et réalisation du séquenceur:

5.2.1. Rappels sur les systèmes logiques:

La plupart des systèmes logiques se ramènent à deux types de base: Les systèmes synchrones et les systèmes asynchrones.

a). Systèmes logiques asynchrones:

Ce sont principalement des automatismes industriels comportant des moteurs dont la commande de mise en marche et la durée de marche dépendent souvent de variables asynchrones de l'horloge. Par exemple:

- Fin de course.

- Océdeur de niveau etc...

b). Systèmes logiques synchrones:

Dans de tels systèmes toutes les variables logiques sont synchrones de l'horloge, et une opération logique quelconque a une durée égale à une ou plusieurs périodes de l'horloge.

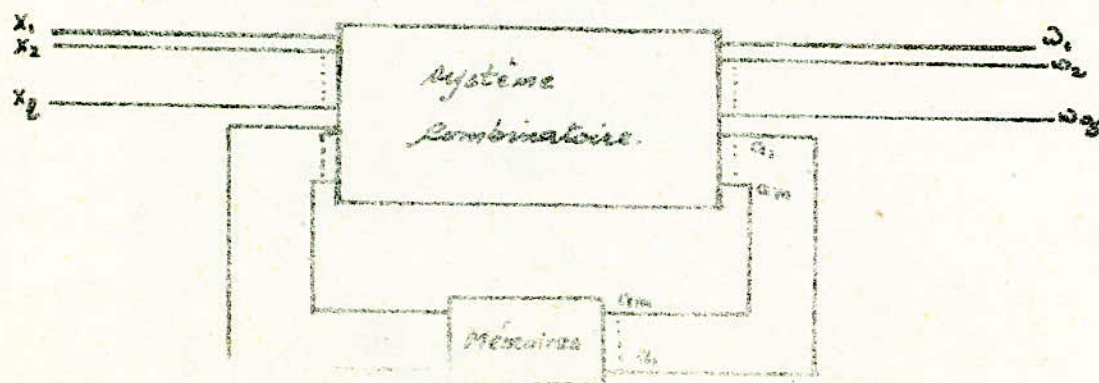
Par exemple:

- 1 impulsion pour le chargement parallèle d'un registre et 4 impulsions pour le décalage sur quatre positions.

Ce sont des ensembles logiques réalisant une ou plusieurs fonctions de l'arithmétique binaire; des logiques assurant le séquençement des appareils de mesure. Les systèmes ne comprennent en général que des circuits intégrés logiques avec éventuellement quelques entrées asynchrones mais synchronisées sur l'Horloge.

5.2.2. Représentation des systèmes séquentiels synchrones:

Un système séquentiel synchrone peut être représenté par le schéma suivant:



$A = \{a_1, a_2, \dots, a_m\}$: Ensemble des états du système.

$X = \{x_1, x_2, \dots, x_p\}$: Ensemble des entrées principales.

$W = \{w_1, w_2, \dots, w_q\}$: Ensemble des sorties du système.

$a(t+1) = \delta[A(t), X(t)]$. L'état futur est lié à l'état présent et à l'entrée présente par la fonction de transition δ . Nous pouvons représenter la sortie sous deux formes

a). Représentation de Mealy:

$W(t) = \lambda_1[A(t), X(t)]$. La sortie présente est liée à l'entrée présente et à l'état présent par la fonction de sortie λ_1 .

b). Représentation de Moore:

$W(t) = \lambda_2[A(t)]$. La sortie présente est liée uniquement à l'état présent par la fonction de sortie λ_2 .

Remarque: Nous pouvons passer de la représentation de Mealy à celle de Moore et inversement.

5.2.3. Etude du séquenceur:

Comme nous l'avons vu, le rôle du séquenceur est d'élaborer les signaux de commande "y" pour réaliser les différentes micro-opérations.

L'état a_i du séquenceur caractérise l'état d'avancement dans l'organigramme.

- Il passe explicitement de l'état a_i à l'état a_{i+1}
- Il peut effectuer des sauts à d'autres positions (rupture de séquence).
- Il peut rester à une position donnée en attente d'un événement.

1. Méthodes pour réaliser le séquençement dans le temps

a). Le séquenceur connaît le temps de réponse des différents organes commandés, il fait alors intervenir des délais appropriés entre les différentes commandes successives de manière à laisser aux opérations le temps de se réaliser.

b). Le séquenceur reçoit des différents organes du

calculateurs des signaux indiquant qu'ils ont terminé les opérations commandées et se libèrent éventuellement pour tester leurs états. Le séquenceur ne laisse une nouvelle opération se réaliser qu'après avoir été assuré que les opérations le précédant ont été terminées et s'être assuré que l'organe qui doit l'effectuer est libre.

Il existe plusieurs méthodes pour la réalisation du séquenceur. Nous allons exposer celle qui répond le mieux à l'étude des automates finis de grande capacité.

2. Ordinoqramme:

C'est le schéma représentant le déroulement détaillé de toutes les micro-opérations pour la réalisation de l'addition en virgule flottante.

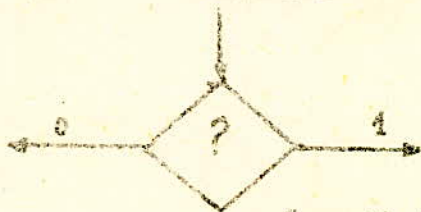
Nous adopterons les conventions suivantes:

- Un rectangle représentera une micro-instruction. Celle-ci est indiquée par l'ensemble des μ -opérations inscrites à l'intérieur du rectangle.

- Un losange ou sommet conditionnel qui signifie qu'en fonction de la valeur "0" ou "1" d'une fonction logique bien précise, il faut réaliser telle opération ou telle autre.



Schéma d'une instruction



Sommet conditionnel.

L'Ordinoqramme (Compte-Schéma d'Algorithme) satisfait aux conditions suivantes:

- Les entrées et les sorties de sommets se relient l'un vers l'autre à l'aide de segments orientés de la sortie vers l'entrée.
- Chaque sortie n'est reliée qu'avec une seule entrée.
- N'importe quelle entrée se relie au moins avec une sortie.
- Il n'importe quel sommet nous avons au moins un chemin qui nous relie au sommet initial et au sommet final.
- L'une des sorties de sommet conditionnel peut être reliée à l'entrée de celui-ci. Il s'appellera alors "Sommet d'attente". Ceci n'est pas admissible pour un sommet d'opérateur.

- Dans chaque sommet conditionnel on inscrit un élément de l'ensemble $X = \{x_1, x_2, \dots\}$ (ensemble des conditions logiques). Quand on a un même test il est permis d'inscrire un même élément de l'ensemble X dans des sommets conditionnels différents.

- Dans chaque sommet d'opérateur on inscrit une micro-instruction γ_i . Il est permis d'inscrire la même micro-instruction dans des sommets d'opérateurs différents quand on a la même μ -instruction à exécuter dans les différents cas.

L'ordinogramme (Graphe-Schéma d'Algorithme) est donné à la figure 6.

3. Obtention de GSA marqué (ordinogramme marqué):

Il est obtenu à partir de l'ordinogramme représenté à la figure 6 en codant toute les entrées de sommets qui viennent après les sommets d'opérateurs (instructions) de la manière suivante:

a). Par le symbole a_1 nous codons l'entrée du sommet qui suit le sommet initial et l'entrée du sommet final.

b). Les entrées de tous les sommets qui suivent les sommets d'opérateurs sont codés par des symboles a_i .

c). L'entrée d'un sommet n'est marquée que par un seul symbole.

d). Les entrées de sommets différents sont codées par des symboles différents sauf pour le sommet final.

Le schéma de l'ordinogramme marqué est donné à la figure n°7.

4. Composition du graphe des transitions de l'automate (diagramme de flux):

Le GSA marqué sert de base pour l'obtention du diagramme de flux. Dans notre problème nous utilisons la représentation de Mealy. Lors du codage de l'ordinogramme nous avons utilisé les symboles (a_1, \dots, a_n) qui représentent les états de l'automate de Mealy. Nous procédons de la façon suivante

- Trouver tous les chemins qui permettent de passer de l'état a_m à l'état a_s sans passer par un état intermédiaire a_j .

- Relier par une flèche les transferts de a_m à a_s et mettre en correspondance les signaux d'entrée $X(a_m - a_s)$ qui

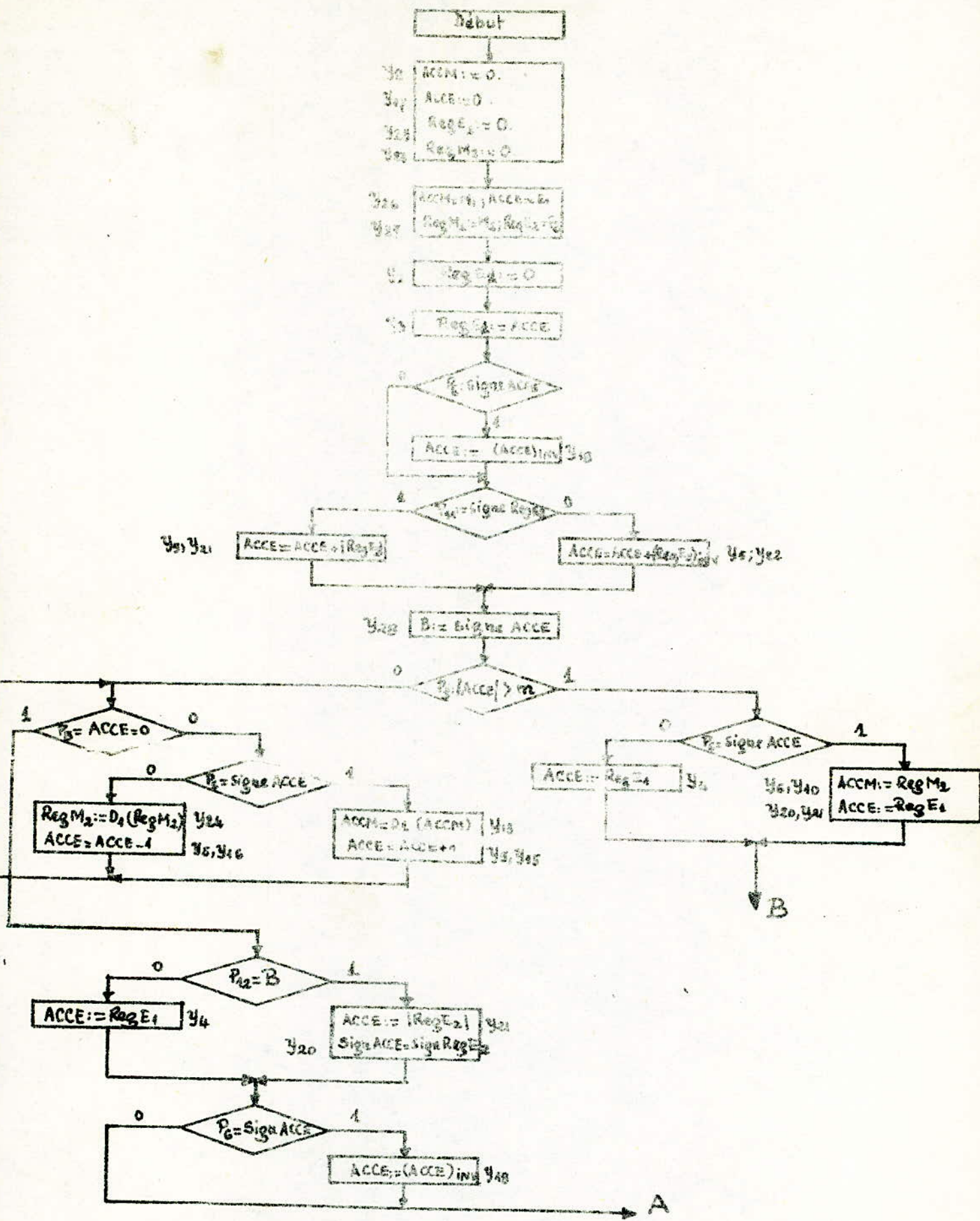


fig 6. Ordigramme de l'Addition en virgule Flottante. ①

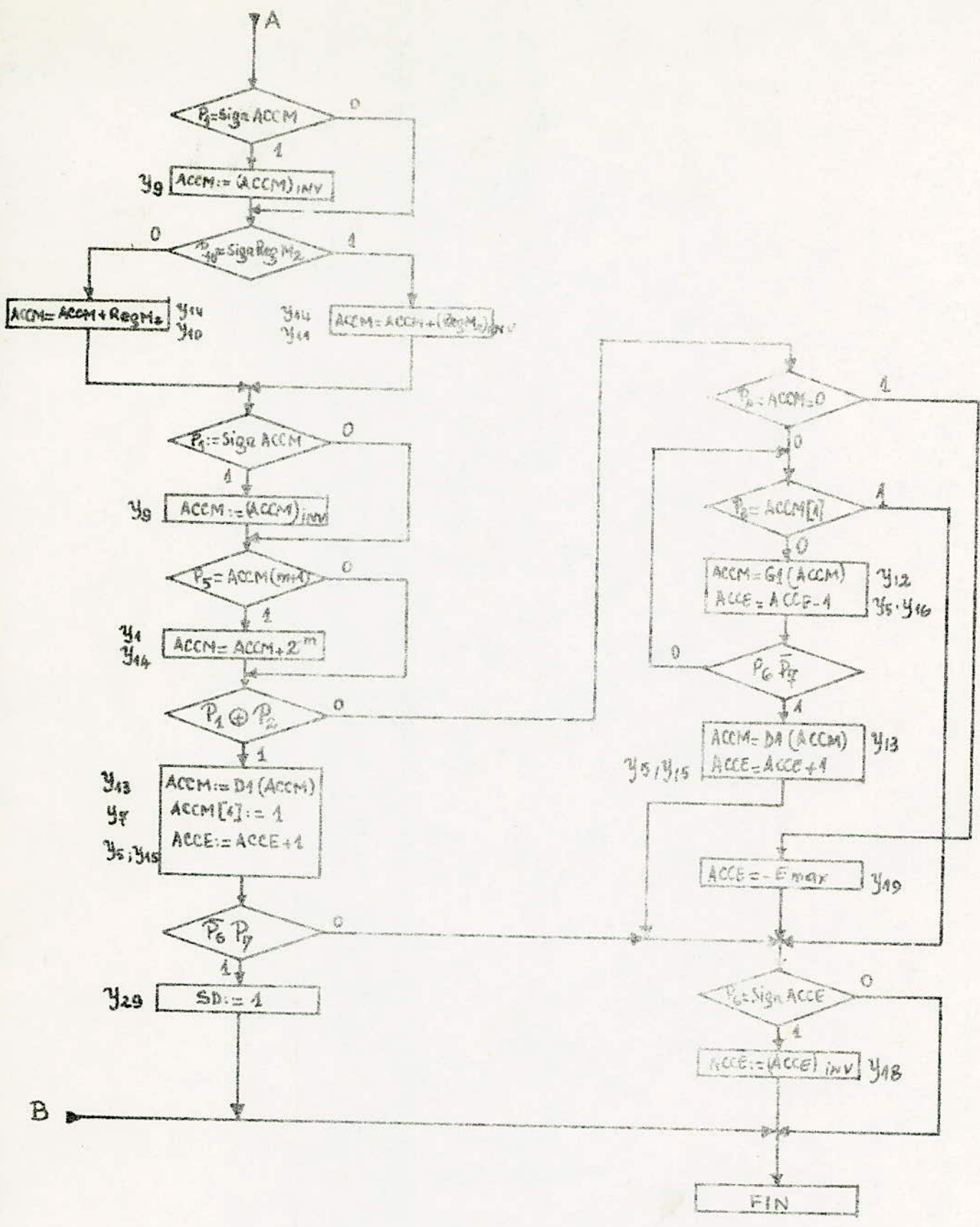


Fig. 6. Ordinoqramme de l'Addition en virgule flottante ②.

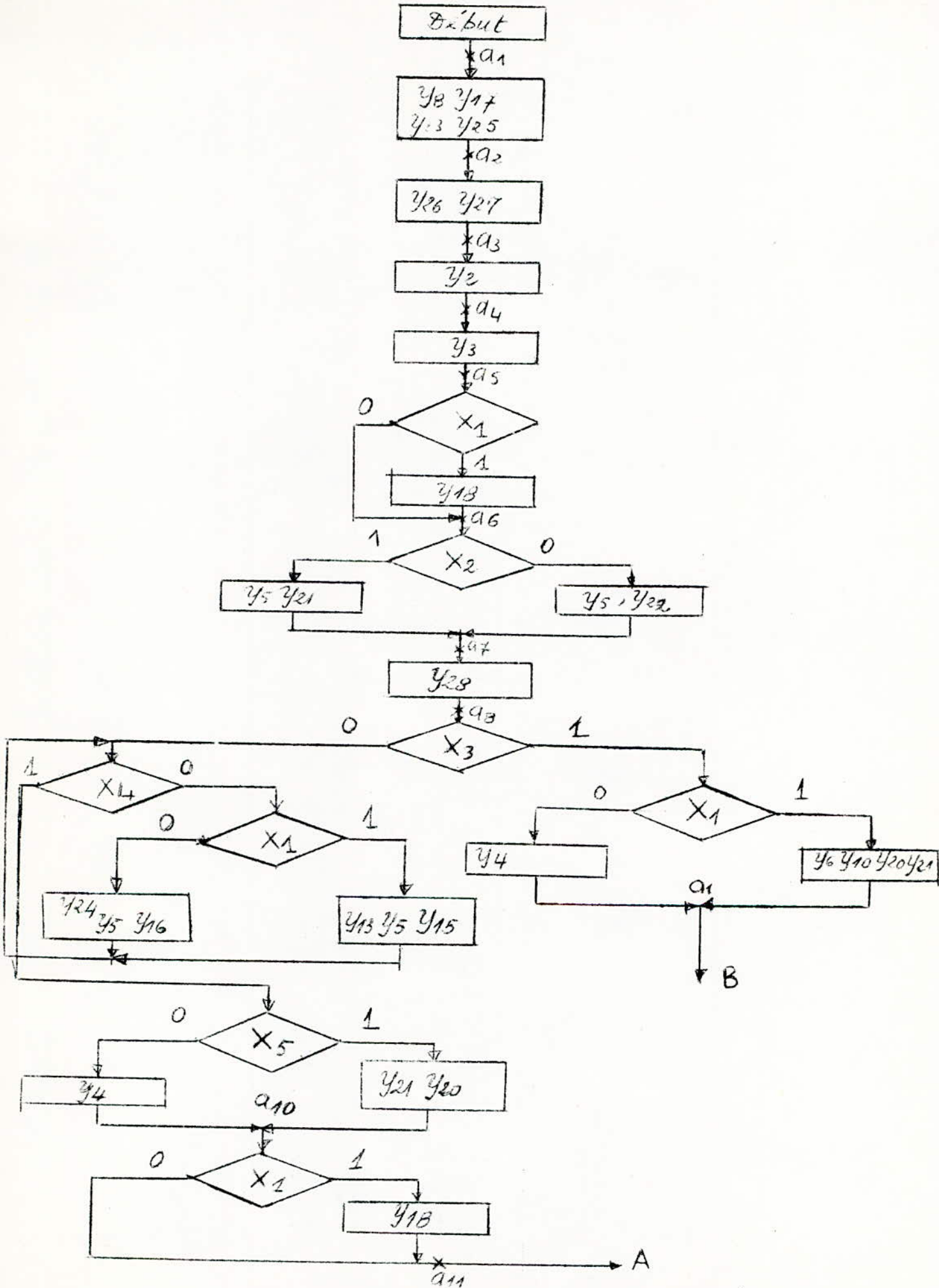


Fig 7. Ordinogramme marqué ①.

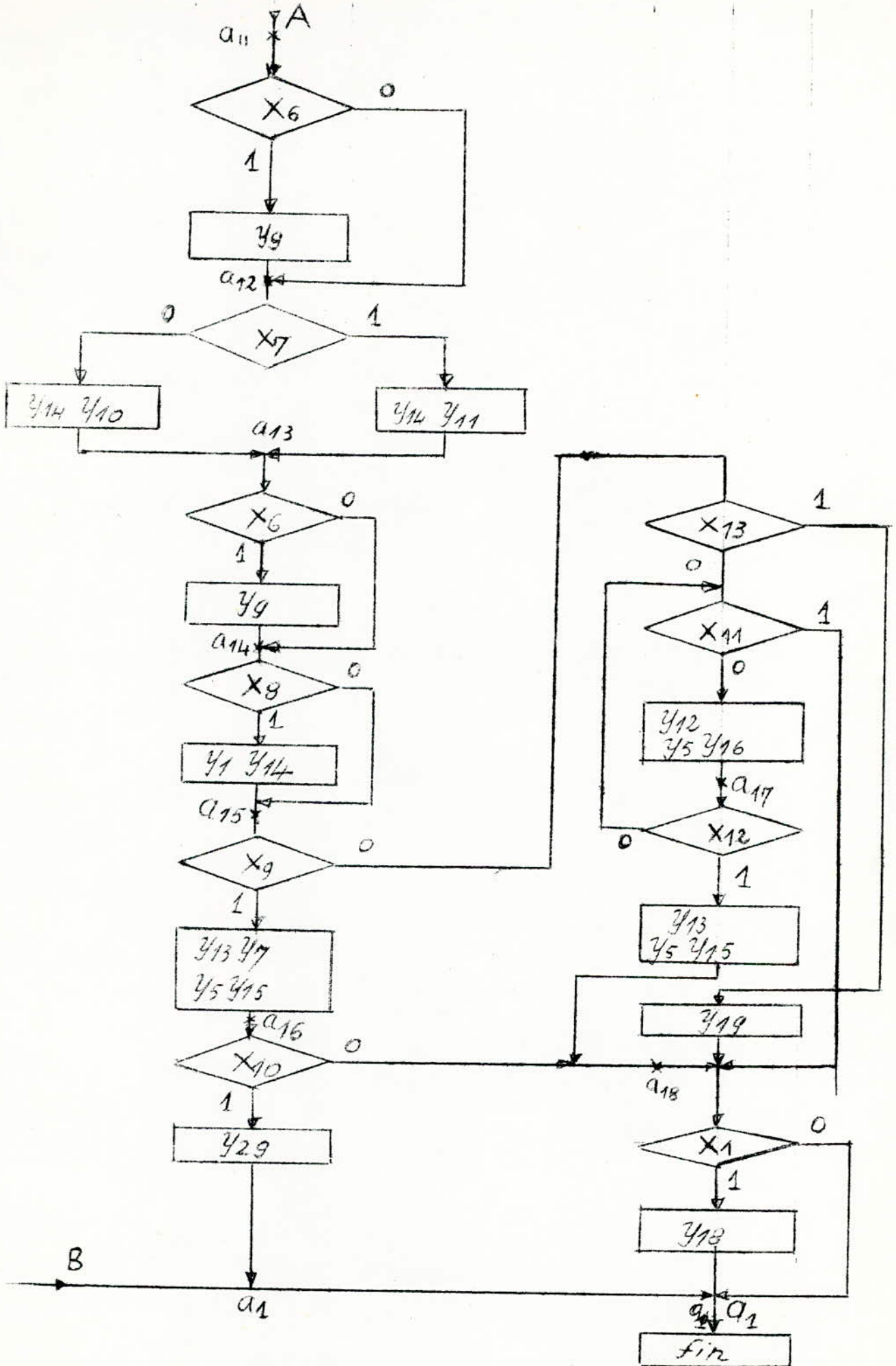


Fig 7. Ordinateur programme marqué (2)

permettent le transfert avec les signaux de sortie. Ceci est illustré par le schéma suivant:



X: entrées qui permettent la Trans. est.

W: Sorties.

Le graphe des transitions (diagramme de fluence) de l'automate de Mealy est donné à la figure 5.

La synthèse d'un système logique synchronisé par une horloge et comprenant au maximum trois variables et deux entrées est relativement facile en utilisant la méthode d'Huffman par exemple, mais ceci n'est pas le cas de notre problème car nous avons 18 états à coder. Il nous faut donc cinq variables. Rappelons à titre indicatif la méthode d'Huffman.

- Diagramme de fluence.
 - Table des états primitive.
 - contraction des états →
- réduction des états redondants (stable)
 - réduction des lignes.
- Choix des variables secondaires et leur codage
 - Matrice d'excitation.
 - Schéma.

5. Table structurale inverse:

C'est la représentation tabulaire de l'automate. Elle comporte toutes les transitions $Q_m \rightarrow Q_s$, les fonctions d'entrée et de sortie. Elle comporte trois autres colonnes (qui n'apparaissent pas dans le diagramme de fluence).

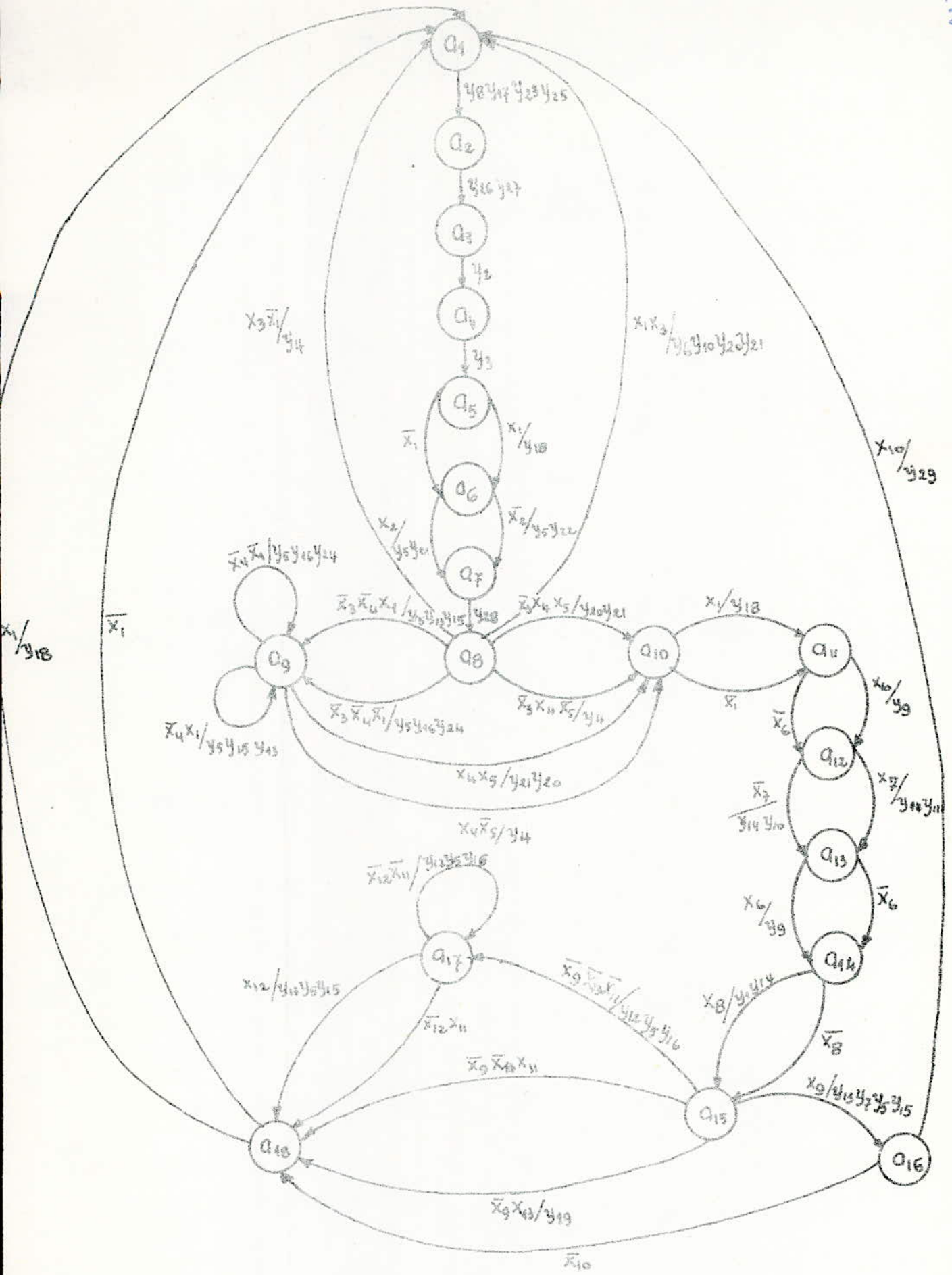
$C(Q_m)$: code de l'état initial.

$C(Q_s)$: code de l'état final.

et $\tilde{F}(Q_m, Q_s)$ qui est l'ensemble des fonctions d'excitation qui changent l'état des bascules pour qu'on ait la transition $Q_m \rightarrow Q_s$.

Pour obtenir cette table nous procéderons de la manière suivante:

*. Dans la colonne Q_m nous notons tous les états qui viennent vers l'état Q_i et à chaque état de la colonne Q_m , nous faisons correspondre toutes les entrées et les sorties X_i et Y_j que l'on a



Figs . Diagramme de Fluence.

lors de la transition $a_m \rightarrow a_i$, et enfin dans la dernière colonne on note les fonctions d'excitation obligatoires pour avoir la transition $a_m \rightarrow a_i$.

Pour chaque état a_i de la colonne a_s on procède de la même façon.

* Codage des états.

Notre système comporte 18 états. Pour compter de 0 à 18, il nous faut cinq bascules. Et chaque état i nous faisons correspondre le nombre binaire i . Rappelons qu'il existe des méthodes pour le codage rationnel des états qui permettent de réduire le nombre de bascules des mémoires élémentaires, donc de simplifier le schéma de l'automate. L'une de ces méthodes sera étudiée par la suite.

* Fonctions d'excitation.

Pour passer de l'état a_m à l'état a_s , nous avons certains éléments mémoires qui changent d'état. Dans cette colonne on note la fonction obligatoire qui permet d'avoir la transition.

Exemple :

$$C(a_4) = 00110 \quad \text{et} \quad C(a_5) = 01000.$$

Si nous utilisons comme élément mémoire des bascules R_5 ; pour passer de l'état a_m à l'état a_s il faut:

- mettre à 1 la bascule 5
- Remettre à zéro les bascules 3 et 4.

nous aurons : $F(a_m, a_s) = E_5, R_3, R_4$

Si nous utilisons des bascules T nous aurons:

$$F(a_m, a_s) = T_2, T_3, T_4.$$

La table structurale inverse est donnée aux pages 39 et 40.

Les équations des sorties F et S et des signaux Y_i sont données aux pages 41 et 42.

Table Structural inverse:

a_m	Code(a_m)	a_s	Code(a_s)	$X(a_m, a_s)$	$Y(a_m, a_s)$	$\tilde{F}(a_m, a_s)$
a_8	01000	a_1	00001	$X_1 X_3$	$y_6 y_{10} y_{20} y_{21}$	$R_2 S_5$
a_3	01000			$\bar{X}_1 X_3$	y_4	$R_2 S_5$
a_{16}	10000			X_{10}	y_{29}	$R_1 S_5$
a_{18}	10010			\bar{X}_1	—	$R_1 R_4 R_5$
a_{18}	10010			X_1	y_{18}	$R_1 R_4 R_5$
a_1	00001	a_2	00010	—	$y_8 y_{17} y_{23} y_{25}$	$R_5 S_4$
a_2	00010	a_3	00011	—	$y_{26} y_{27}$	S_5
a_3	00011	a_4	00100	—	y_2	$R_4 R_5 S_3$
a_4	00100	a_5	00101	—	y_3	S_5
a_5	00101	a_6	00110	X_1	y_{18}	$R_5 S_4$
a_5	00101			\bar{X}_1	—	$R_5 S_4$
a_6	00110	a_7	00111	X_2	$y_5 y_{22}$	S_5
a_6	00110		0	X_2	$y_5 y_{21}$	S_5
a_7	00111	a_8	01000	—	y_{28}	$R_3 R_4 R_5 S_2$
a_3	01000	a_9	01001	$\bar{X}_3 \bar{X}_4 X_1$	$y_5 y_{13} y_{15}$	S_5
a_3	01000			$\bar{X}_3 \bar{X}_4 \bar{X}_1$	$y_{15} y_{16} y_{24}$	S_5
a_9	01001			$\bar{X}_4 \bar{X}_1$	$y_5 y_{16} y_{24}$	—
a_9	01001			$\bar{X}_4 X_1$	$y_5 y_{13} y_{15}$	—
a_8	01000	a_{10}	01010	$\bar{X}_3 X_4 X_5$	$y_{20} y_{21}$	S_4
a_8	01000			$\bar{X}_3 X_4 \bar{X}_5$	y_4	S_4
a_9	01001			$X_4 X_5$	$y_{20} y_{21}$	$R_5 S_4$
a_9	01001			$X_4 \bar{X}_5$	y_4	$S_5 S_4$
a_{10}	01010	a_{11}	01011	X_1	y_{18}	S_5
a_{10}	01010			\bar{X}_1	—	S_5
a_{11}	01011	a_{12}	01100	\bar{X}_6	—	$R_4 R_5 S_3$
a_{11}	01011			X_6	y_9	$R_4 R_5 S_3$
a_{12}	01100	a_{13}	01101	X_7	$y_{11} y_{14}$	S_5
a_{12}	01100			\bar{X}_7	$y_{10} y_{14}$	S_5
a_{13}	01101	a_{14}	01110	X_6	y_9	$R_5 S_4$
a_{13}	01101			\bar{X}_6	—	$R_5 S_4$

Table Structurale inverse (suite):

a_m	Code(a_m)	a_s	Code(a_s)	$X(a_m, a_s)$	$Y(a_m, a_s)$	$\tilde{F}(a_m, a_s)$
a_{14}	01110	a_{15}	01111	x_8	$y_1 y_{14}$	S_5
a_{14}	01110			\bar{x}_8	—	S_5
a_{15}	01111	a_{16}	10000	x_9	$y_5 y_7 y_{13} y_{15}$	$R_2 R_3 R_4 R_5 S_1$
a_{15}	01111	a_{17}	10001	$\bar{x}_{13} \bar{x}_9 \bar{x}_{11}$	$y_5 y_{12} y_{16}$	$R_2 R_3 R_4 S_1$
a_{17}	10001			$\bar{x}_{11} \bar{x}_{12}$	$y_5 y_{12} y_{16}$	—
a_{15}	01111	a_{18}	10010	$\bar{x}_{13} \bar{x}_9 \bar{x}_{11}$	—	$R_2 R_3 R_5 S_1$
a_{15}	01111			$x_{13} \bar{x}_9$	y_{19}	$R_2 R_3 R_5 S_1$
a_{16}	10000			\bar{x}_{10}	—	S_4
a_{17}	10001			$x_{11} \bar{x}_{12}$	—	$R_5 S_4$
a_{17}	10001			x_{12}	$y_5 y_{13} y_{15}$	$R_5 S_4$

Nous utiliserons des bascules R-S. Les T_i représentent les états des bascules, et en introduisant un signal d'horloge "h", nous obtenons les fonctions d'excitation et de sortie suivantes.

$$R_1 = h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_{10} \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5.$$

$$R_2 = h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_3 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5.$$

$$R_3 = h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5.$$

$$R_4 = h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_{13} X_9 X_{11}.$$

$$R_5 = h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_9 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_{13} X_9 X_{11} \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_{13} X_9 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_{11} X_{12} \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_{12}.$$

$$S_1 = h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5.$$

$$S_2 = h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5.$$

$$S_3 = h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5.$$

$$S_4 = h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_3 X_4 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_4 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_{10} \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_{11} X_{12} \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5.$$

$$S_5 = h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_{10} \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_{10} \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_3 X_4 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5.$$

$$y_1 = h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_8$$

$$y_2 = h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5.$$

$$y_3 = h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5.$$

$$y_4 = h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_1 X_3 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_3 X_4 X_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_4$$

$$y_5 = h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_1 X_3 X_4 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_4 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_9 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_{13} X_9 X_{11} \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_{11} \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_{12}.$$

$$y_6 = h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_1 X_3$$

$$y_7 = h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_9.$$

$$y_8 = h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5.$$

$$y_9 = h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_6 \vee h \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 X_6.$$

6. Minimisation du circuit:

Les expressions des fonctions d'excitation et de sortie correspondant à un certain transfert sont obtenues par les combinaisons de variables par lesquelles sont codés les états de l'automate. Dans notre cas, elles sont fonction de cinq variables (x_1, x_2, x_3, x_4 et x_5) et nous allons faire correspondre à chaque état une case de la table de Karnaugh et codé cet état d'une autre manière.

Exemple: $1.01 = \underbrace{\bar{x}_1 \bar{x}_2 x_3}_{\alpha_1} \underbrace{x_4 x_5}_{\beta_3} = \alpha_1 \beta_3.$

et en posant $\alpha_i \beta_j = \beta_i$ nous aurons: $\beta_{11} = \alpha_1 \beta_3.$

a) Table de Karnaugh pour cinq variables:

$x_4 x_5$	$x_1 x_2 x_3$	000	001	011	010	110	111	101	100
00	β_0	α_0	α_1	α_2	α_3	α_4	α_5	α_6	α_7
01	β_1								
11	β_2			α_{15}					
10	β_3								

On peut "à l'aide" de ce "retour" le circuit qui permet de réaliser les α_i et β_j . Le schéma du précodeur de retour est donné à la figure 3.

b) Expression des fonctions d'excitation et de sortie en utilisant le précodeur de retour:

Les α_i et β_j sont directement à la sortie du précodeur.

$$R_1 = \alpha_4 \beta_0 x_{10} + \alpha_4 \beta_2 \quad R_2 = \alpha_2 \beta_3 x_3 + \alpha_2 \beta_3.$$

$$R_3 = \alpha_3 \beta_3 + \alpha_3 \beta_3$$

$$R_4 = \alpha_4 \beta_2 + \alpha_0 \beta_3 + \alpha_1 \beta_3 + \alpha_2 \beta_3 + \alpha_3 \beta_3 + \alpha_4 \beta_3 + \alpha_5 \beta_3 + \alpha_6 \beta_3 + \alpha_7 \beta_3 + \alpha_8 \beta_3 + \alpha_9 \beta_3 + \alpha_{10} \beta_3 + \alpha_{11} \beta_3 + \alpha_{12} \beta_3 + \alpha_{13} \beta_3 + \alpha_{14} \beta_3 + \alpha_{15} \beta_3$$

$$R_5 = \alpha_0 \beta_2 + \alpha_0 \beta_3 + \alpha_1 \beta_2 + \alpha_1 \beta_3 + \alpha_2 \beta_2 x_4 + \alpha_2 \beta_3 + \alpha_3 \beta_2 + \alpha_3 \beta_3 x_5 + \alpha_4 \beta_2 x_3 \bar{x}_9 x_{10} + \alpha_4 \beta_3 x_4 \bar{x}_9 + \alpha_4 \beta_2 x_{11} \bar{x}_{12} + \alpha_4 \beta_1 x_{12}.$$

$$S_1 = \alpha_3 \beta_3 \quad S_2 = \alpha_0 \beta_3 \quad S_3 = \alpha_0 \beta_3 + \alpha_2 \beta_3.$$

$$S_4 = \alpha_0 \beta_2 + \alpha_1 \beta_2 + \alpha_2 \beta_0 x_3 x_4 + \alpha_2 \beta_1 x_4 + \alpha_3 \beta_1 + \alpha_4 \beta_0 x_{10} + \alpha_4 \beta_1 x_{12} + \alpha_0 \beta_2 x_1 \bar{x}_{12}.$$

$$S_5 = \alpha_2 \beta_0 x_3 + \alpha_4 \beta_0 x_{10} + \alpha_4 \beta_2 + \alpha_0 \beta_3 + \alpha_2 \beta_0 + \alpha_3 \beta_2 + \alpha_2 \beta_0 \bar{x}_3 \bar{x}_4 + \alpha_2 \beta_2 + \alpha_3 \beta_0 + \alpha_3 \beta_0.$$

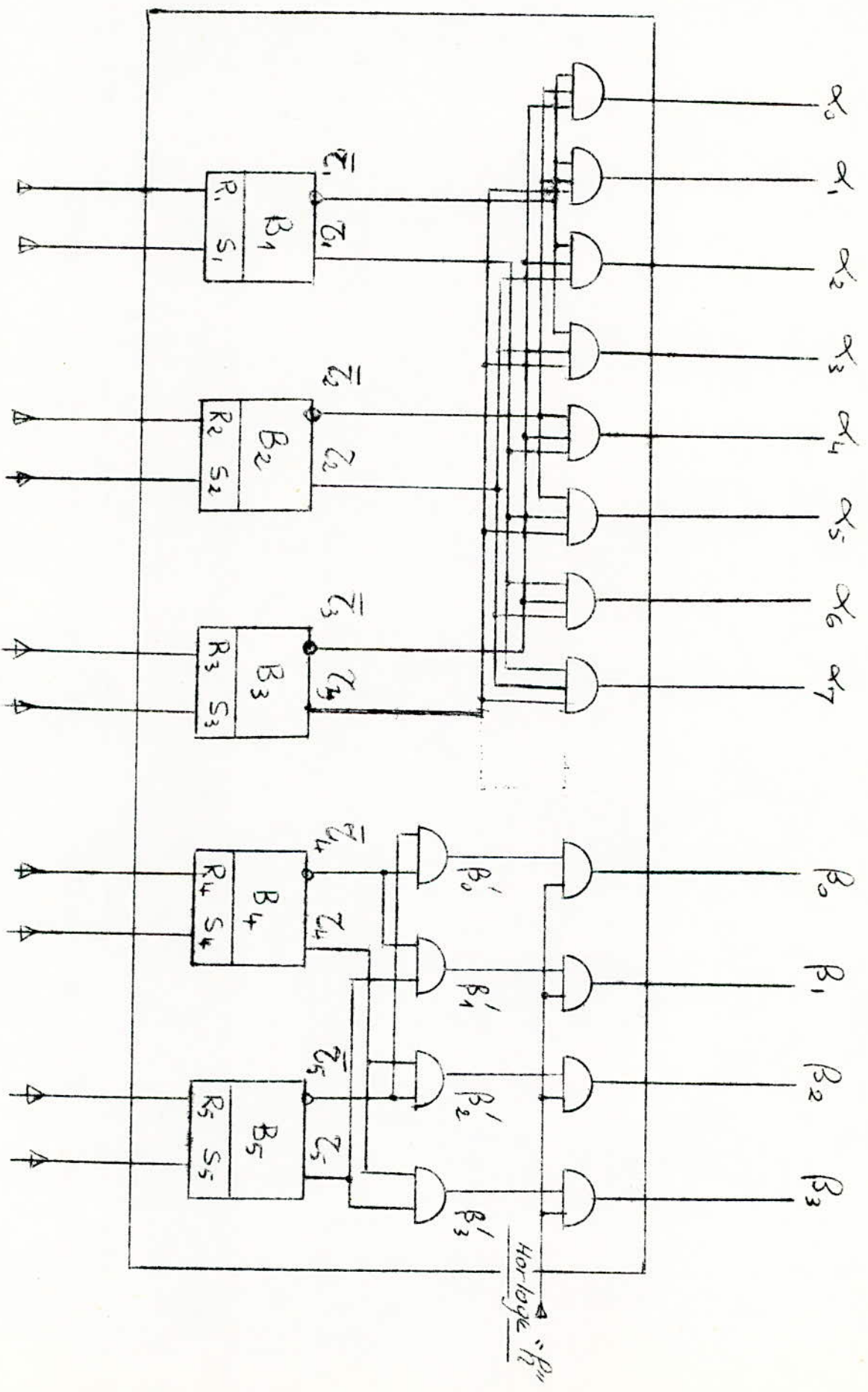


Fig. 9. Précodeur de retour.

$$\begin{aligned}
 y_1 &= \alpha_3 \beta_2 x_8 & y_2 &= \alpha_0 \beta_3 & y_3 &= \alpha_1 \beta_0 \\
 y_4 &= \alpha_2 \beta_0 \bar{x}_1 x_3 + \alpha_2 \beta_0 \bar{x}_2 x_4 \bar{x}_5 + \alpha_2 \beta_2 x_6 \bar{x}_5 \\
 y_5 &= \alpha_1 \beta_2 + \alpha_3 \beta_2 x_9 + \alpha_4 \beta_1 \bar{x}_{12} + \alpha_4 \beta_2 \bar{x}_{11} \bar{x}_{12} + \alpha_2 \beta_0 \bar{x}_1 \bar{x}_3 \bar{x}_4 + \alpha_2 \beta_1 \bar{x}_4 + \\
 &+ \alpha_3 \beta_3 \bar{x}_{13} \bar{x}_9 \bar{x}_{11} \\
 y_6 &= \alpha_2 \beta_0 x_1 x_3 & y_7 &= \alpha_1 \beta_3 x_9 & y_8 &= \alpha_0 \beta_1 \\
 y_9 &= \alpha_2 \beta_3 x_6 + \alpha_1 \beta_0 x_6 & y_{10} &= \alpha_2 \beta_0 \bar{x}_1 \bar{x}_3 + \alpha_3 \beta_0 \bar{x}_7 \\
 y_{11} &= \alpha_3 \beta_0 x_7 & y_{12} &= \alpha_3 \beta_2 \bar{x}_{13} \bar{x}_9 \bar{x}_{11} + \alpha_4 \beta_1 \bar{x}_{11} \bar{x}_{12} \\
 y_{13} &= \alpha_2 \beta_0 \bar{x}_3 \bar{x}_4 x_1 + \alpha_2 \beta_2 \bar{x}_4 x_2 + \alpha_3 \beta_2 x_9 + \alpha_4 \beta_1 x_2 \\
 y_{14} &= \alpha_3 \beta_0 + \alpha_2 \beta_2 x_8 \\
 y_{15} &= \alpha_2 \beta_0 \bar{x}_3 \bar{x}_4 x_1 + \alpha_2 \beta_2 \bar{x}_4 x_1 + \alpha_3 \beta_3 x_9 + \alpha_4 \beta_0 x_{12} \\
 y_{16} &= \alpha_2 \beta_0 \bar{x}_3 \bar{x}_4 \bar{x}_9 + \alpha_2 \beta_0 \bar{x}_4 \bar{x}_{11} + \alpha_3 \beta_3 \bar{x}_{13} \bar{x}_9 \bar{x}_{11} + \alpha_4 \beta_2 \bar{x}_{11} \bar{x}_{12} \\
 y_{17} &= \alpha_0 \beta_2 & y_{18} &= \alpha_0 \beta_2 x_9 + \alpha_4 \beta_1 x_1 + \alpha_2 \beta_2 x_2 & y_{19} &= \alpha_3 \beta_3 x_4 \bar{x}_9 \\
 y_{20} &= \alpha_2 \beta_0 x_1 x_3 + \alpha_2 \beta_0 \bar{x}_3 x_1 x_5 + \alpha_2 \beta_2 x_4 x_5 \\
 y_{21} &= \alpha_2 \beta_0 x_1 x_3 + \alpha_1 \beta_2 x_2 + \alpha_2 \beta_0 \bar{x}_3 x_4 x_5 + \alpha_2 \beta_2 x_4 x_5 \\
 y_{22} &= \alpha_1 \beta_2 \bar{x}_2 & y_{23} &= \alpha_0 \beta_2 & y_{24} &= \alpha_2 \beta_0 \bar{x}_3 \bar{x}_4 \bar{x}_1 + \alpha_2 \beta_2 \bar{x}_4 \bar{x}_1 \\
 y_{25} &= \alpha_0 \beta_2 & y_{26} &= y_{27} = \alpha_0 \beta_2 & y_{28} &= \alpha_1 \beta_3 \\
 y_{29} &= \alpha_4 \beta_0 x_{10}
 \end{aligned}$$

En utilisant le schéma ci-dessus nous avons un gain de matériel par rapport au schéma réalisé directement avec les fonctions obtenues. Nous avons aussi une amélioration du point de vue vitesse.

Pour simplifier le schéma encore plus nous coderons les $\alpha_i \beta_j$ de la manière suivante.

$$\begin{aligned}
 \alpha_i \beta_0 &= A_i & \alpha_i \beta_1 &= B_i & \alpha_i \beta_2 &= C_i \\
 \text{et } \alpha_i \beta_3 &= D_i
 \end{aligned}$$

Le schéma qui permet d'obtenir les A_i ; B_i ; C_i et D_i est donné à la figure 10 (partie C).

En utilisant ces simplifications nous obtenons les équations suivantes :

$$\begin{aligned}
 R_1 &= A_4 x_{10} + C_4 & S_1 &= D_3 \\
 R_2 &= A_2 x_3 + D_3 & S_2 &= D_1 \\
 R_3 &= D_1 + D_3 & S_3 &= D_0 + D_2 \\
 R_4 &= C_4 + D_0 + D_2 + D_3 x_9 + D_5 \bar{x}_{12} \bar{x}_9 \bar{x}_1 \\
 S_4 &= B_0 + B_1 + B_3 + B_2 x_4 + B_4 x_{12} + A_2 \bar{x}_3 x_1 + A_4 \bar{x}_{10} + B_3 x_9 \bar{x}_{12}
 \end{aligned}$$

$$R_5 = B_0 + B_1 + B_3 + D_0 + D_1 + D_2 + B_2 X_4 + B_4 X_{12} + B_4 X_{11} \bar{X}_{12} + D_3 X_9 + D_3 \bar{X}_9 X_{13}$$

$$S_5 = A_1 + A_3 + C_0 + C_1 + C_2 + C_3 + C_4 + A_2 X_3 + A_2 \bar{X}_3 \bar{X}_4 + A_4 X_{10}$$

$$y_1 = C_3 X_8 \quad y_2 = D_0 \quad y_3 = A_1 \quad y_4 = A_2 \bar{X}_1 X_3 + A_2 \bar{X}_3 X_4 \bar{X}_5 + B_2 X_4$$

$$y_5 = C_1 + A_1 X_1 \bar{X}_3 \bar{X}_4 + B_2 \bar{X}_4 + D_3 X_9 + D_3 \bar{X}_9 \bar{X}_{11} \bar{X}_{13} + B_4 X_{12} + B_4 X_{11} \bar{X}_{12}$$

$$y_6 = A_2 X_1 X_3 \quad y_7 = D_3 X_9 \quad y_8 = B_0 \quad y_9 = (B_3 + D_2) X_6$$

$$y_{10} = A_2 X_1 X_3 + A_3 \bar{X}_7 \quad y_{11} = A_3 X_7 \quad y_{12} = B_4 \bar{X}_{11} \bar{X}_{12} + D_3 \bar{X}_9 \bar{X}_{11}$$

$$y_{13} = A_2 X_1 \bar{X}_3 \bar{X}_4 + B_2 \bar{X}_4 X_1 + D_3 X_9 + B_4 X_2 \quad y_{14} = D_3 + C_3 X_8$$

$$y_{15} = A_2 X_1 \bar{X}_3 \bar{X}_4 + B_2 X_1 \bar{X}_4 + D_3 X_9 + B_4 X_{12}$$

$$y_{16} = A_2 \bar{X}_1 \bar{X}_3 \bar{X}_4 + B_2 \bar{X}_1 \bar{X}_4 + D_3 \bar{X}_9 \bar{X}_{11} \bar{X}_{13} + B_4 \bar{X}_{11} \bar{X}_{12}$$

$$y_{16} = y_{12} + A_2 \bar{X}_1 \bar{X}_3 \bar{X}_{11} + B_2 \bar{X}_1 \bar{X}_4 \quad y_{17} = A_1$$

$$y_{18} = (B_1 + C_2 + C_4) X_1 \quad y_{19} = D_3 X_4 \bar{X}_9$$

$$y_{20} = A_2 X_1 X_3 + A_2 \bar{X}_3 X_4 X_5 + B_2 X_{11} X_5 \quad y_{21} = y_{20} + C_1 X_2$$

$$y_{22} = C_1 \bar{X}_2 \quad y_{23} = B_0 \quad y_{24} = B_2 \bar{X}_1 \bar{X}_4 + A_2 \bar{X}_1 \bar{X}_3 \bar{X}_{11}$$

$$y_{25} = y_{23} = B_0 \quad y_{26} = y_{27} = C_0 \quad y_{28} = D_1 \quad y_{29} = A_4 X_{10}$$

Le schéma du séquenceur est donné à la figure n° 10 (parties ①, ② et ③).

I. METHODES DE SIMPLIFICATION DES CIRCUITS LOGIQUES

Avec l'introduction du pré-décodage de retour nous avons pu simplifier le schéma du séquenceur. Pour les automatismes ayant un G.S.A assez long, il existe des méthodes de décomposition du circuit qui permettent d'alléger le schéma et économiser ainsi du matériel.

6.1. Méthode des nœuds:

1) - Définition d'un nœud:

Si l'entrée d'un sommet conditionnel est liée à deux autres sommets ou moins et que l'un ou plusieurs d'entre eux est un sommet conditionnel, cette entrée est appelée: NOEUD. Le nœud n'est pas un état de l'automate, il est introduit pour alléger la composition du circuit.

2) - Ordinnogramme marqué avec nœuds:

Sur l'ordinnogramme marqué de la figure 7, nous notons tous les nœuds qui s'y trouvent. Nous avons obtenu.

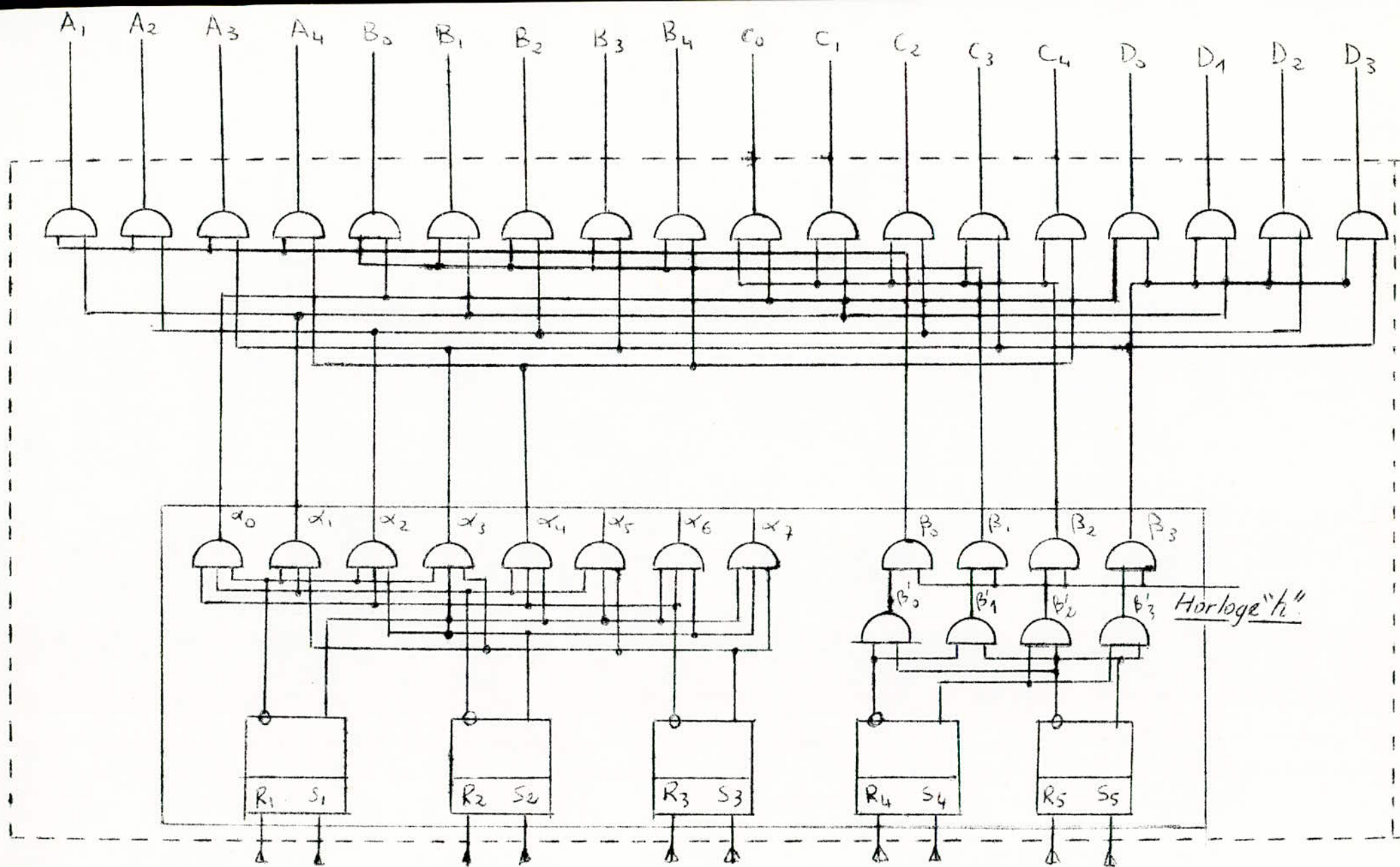


Fig 10 Schéma de l'automate (partie 1).

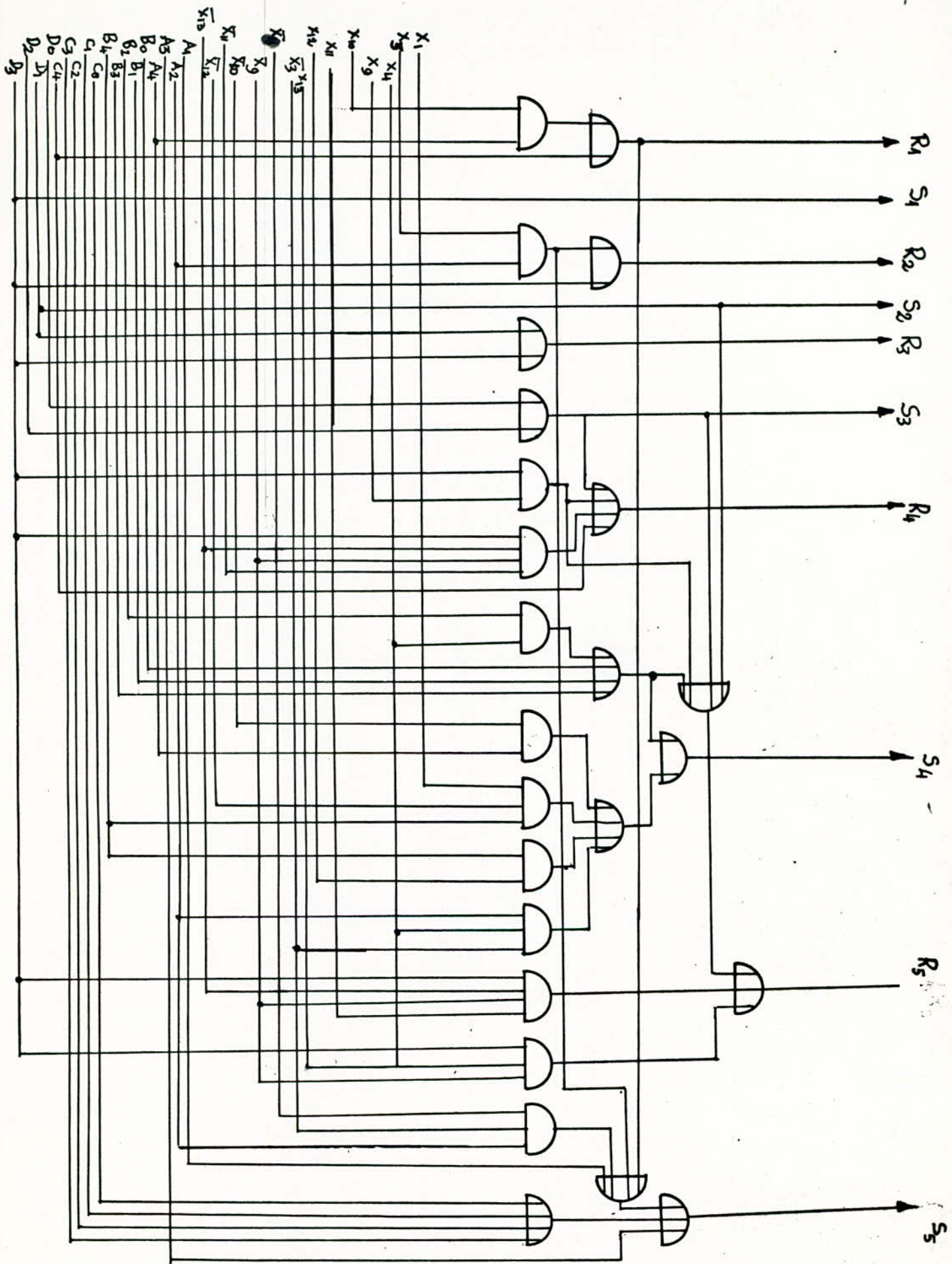


Fig. 10 Schéma de l'automate. (partie ③)

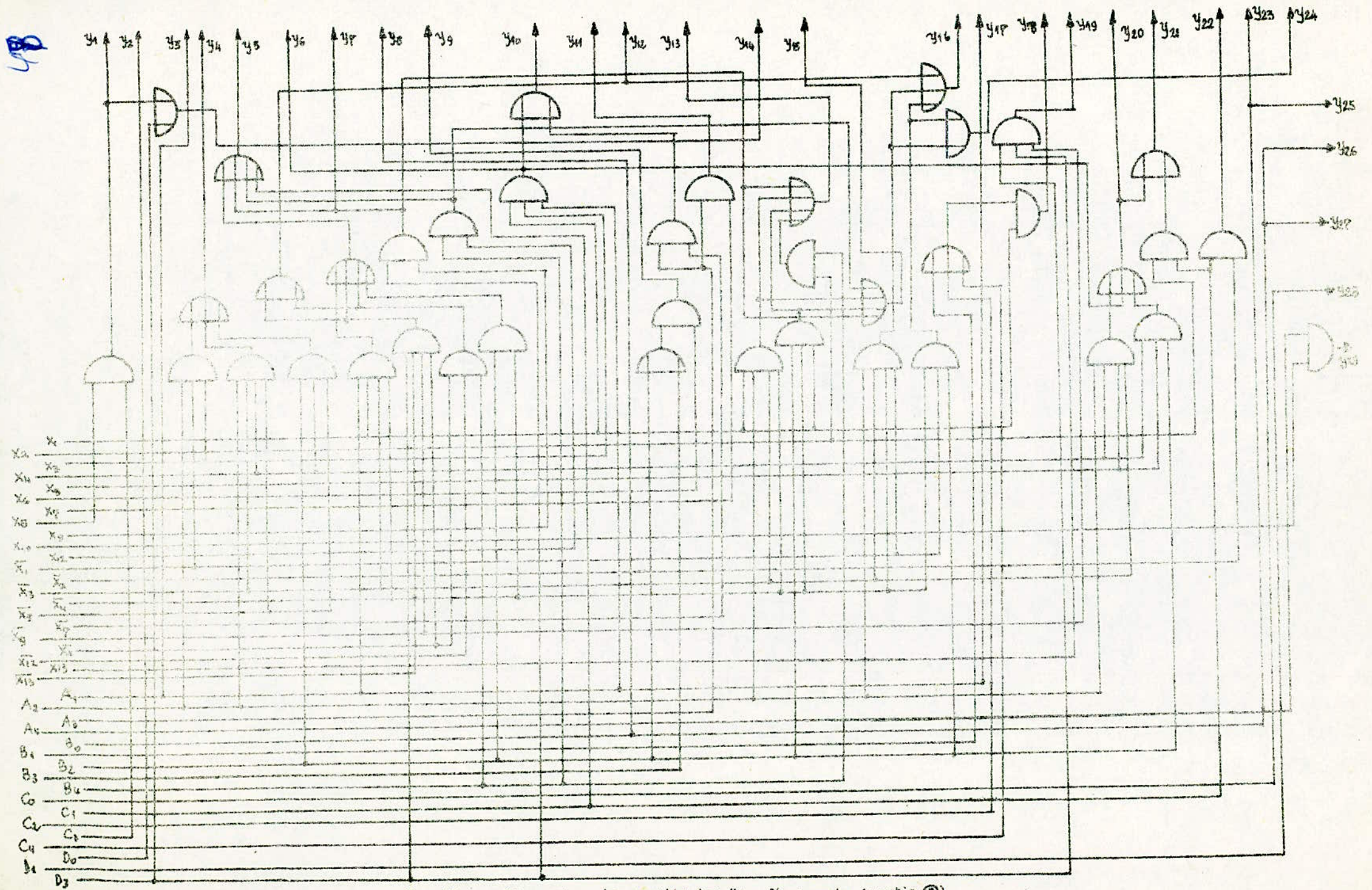


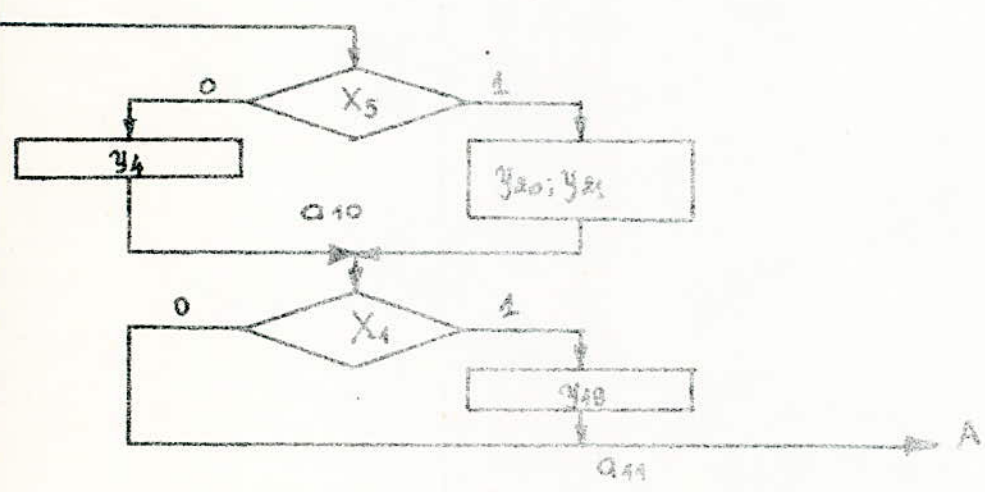
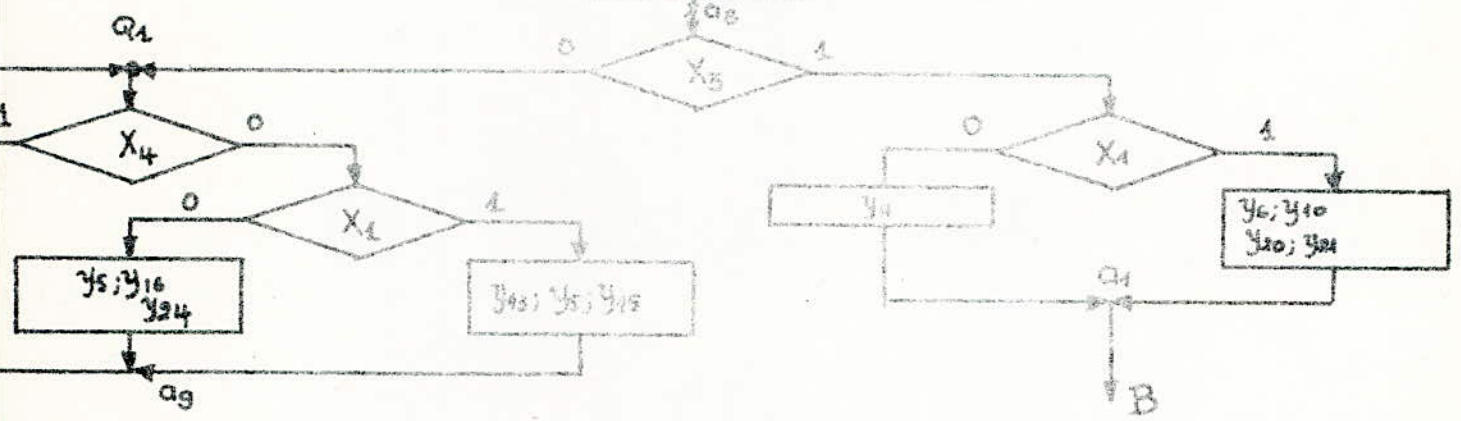
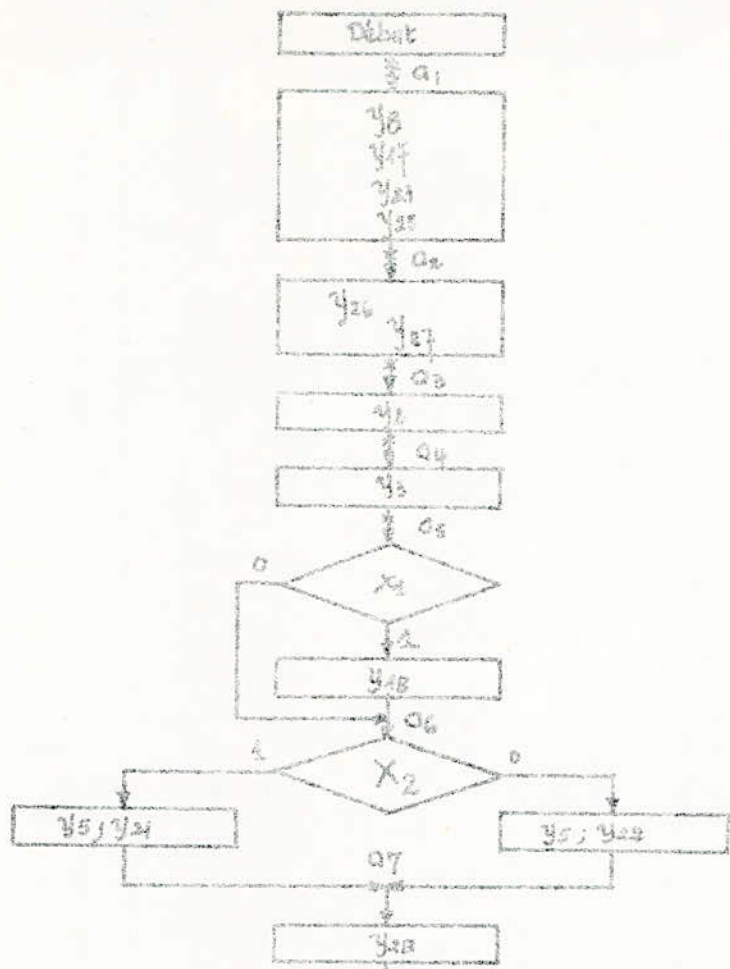
Fig 10 . Signaux de sortie "y_i" de l'automate (partie ②)

trois nœuds Q_1, Q_2 et Q_3 . L'Andimogramme est représenté dans les figures 50 et 51 qui suivent.

4. Table structurale inverse:

La table est du même type que celle étudiée précédemment, mais dans celle-ci nous représenterons les chemins de transfert aux nœuds et aux états (les nœuds ne sont pas codés).

Q_m, Q_n	$C(Q_m)$	$Q_3 \text{ ou } Q_2$	$C(Q_n)$	$X(Q_m, Q_n)$	$Y(Q_m, Q_n)$	$\tilde{F}(Q_m, Q_n)$
Q_8	01000	Q_1	—	\bar{X}_3	—	—
Q_9	01001	Q_1	—	—	—	—
Q_{15}	01111	Q_2	—	$\bar{X}_3 \bar{X}_{13}$	—	—
Q_{17}	10001	Q_2	—	\bar{X}_{12}	—	—
Q_{18}	10010	Q_3	—	—	—	—
Q_{17}	10001	Q_3	—	X_{18}	$Y_5 Y_{13} Y_{15}$	—
Q_{16}	10000	Q_3	—	\bar{X}_{10}	—	—
Q_2	—	Q_3	—	X_{11}	—	—
Q_8	01000	Q_1	00001	$X_1 X_3$	$Y_6 Y_{10} Y_{10} Y_{14}$	$R_2 S_5$
Q_8	01000	Q_1	00001	$\bar{X}_4 X_3$	Y_6	$R_2 S_5$
Q_{16}	10000	Q_1	00001	X_{10}	Y_{19}	$R_1 S_5$
Q_3	—	Q_1	00001	\bar{X}_1	—	$R_1 R_2 R_3 R_4 S_5$
Q_3	—	Q_1	00001	X_1	Y_{18}	$R_1 R_2 R_3 R_4 S_5$
Q_1	00001	Q_2	00010	—	$Y_8 Y_{17} Y_{23} Y_{25}$	$S_4 R_5$
Q_2	00010	Q_3	00011	—	$Y_{26} Y_{27}$	S_5
Q_3	00011	Q_4	00100	—	Y_2	$S_3 R_4 R_5$
Q_4	00100	Q_5	00101	—	Y_3	S_5



a_m, Q_m	$C(a_m)$	a_s, Q_s	$C(a_s)$	$X(a_m, a_s)$	$Y(a_m, a_s)$	$\tilde{F}(a_m; a_s)$
a_5 a_5	00101 00101	a_6 a_6	00110 00110	\bar{x}_4 x_4	— y_{18}	$S_4 R_5$ $S_4 R_5$
a_6 a_6	00110 00110	a_7 a_7	00111 00111	\bar{x}_2 x_2	$y_5 y_{22}$ $y_5 y_{21}$	S_5 S_5
a_7	00111	a_8	01000	—	y_{23}	$S_2 R_3 R_4 R_5$
Q_1 Q_1	— —	a_9 a_9	01001 01001	$\bar{x}_4 \bar{x}_4$ $x_4 \bar{x}_4$	$y_5 y_{16} y_{24}$ $y_5 y_{13} y_{15}$	$R_1 S_2 R_3 R_4 S_5$ $R_1 S_2 R_3 R_4 S_5$
Q_2 Q_2	— —	a_{10} a_{10}	01010 01010	$x_4 x_5$ $x_4 \bar{x}_5$	$y_{20} y_{21}$ y_{14}	$R_1 S_2 R_3 S_4 R_5$ $R_1 S_2 R_3 S_4 R_5$
a_{10} a_{10}	01010 01010	a_{11} a_{11}	01011 01011	\bar{x}_1 x_1	— y_{18}	S_5 S_5
a_{11} a_{11}	01011 01011	a_{12} a_{12}	01100 01100	x_6 \bar{x}_6	y_9 —	$R_4 R_5 S_3$ $R_4 R_5 S_3$
a_{12} a_{12}	01100 01100	a_{13} a_{13}	01101 01101	\bar{x}_7 x_7	$y_{10} y_{14}$ $y_{11} y_{14}$	S_5 S_5
a_{13} a_{13}	01101 01101	a_{14} a_{14}	01110 01110	\bar{x}_6 x_6	— y_9	$S_4 R_5$ $S_4 R_5$
a_{14} a_{14}	01110 01110	a_{15} a_{15}	01111 01111	\bar{x}_8 x_8	— $y_{10} y_{14}$	S_5 S_5
a_{15}	01111	a_{16}	10000	x_9	$y_{15} y_{17} y_{13} y_{15}$	$S_1 R_2 R_3 R_4 R_5$
Q_2	—	a_{17}	10001	\bar{x}_{11}	$y_5 y_{12} y_{16}$	$S_1 R_2 R_3 R_4 S_5$
a_{15}	01111	a_{18}	10010	$\bar{x}_9 x_{13}$	y_{19}	$S_1 R_2 R_3 R_5$

4). Equations aux nœuds:

Pour chaque groupe menant à un nœud Q_n nous utilisons autant de circuits "ET" que de lignes qu'occupe ce groupe dans la table structurale. Les sorties des circuits "ET" sont liées par un "OU" pour avoir les équations aux nœuds $\Phi(Q_n)$.

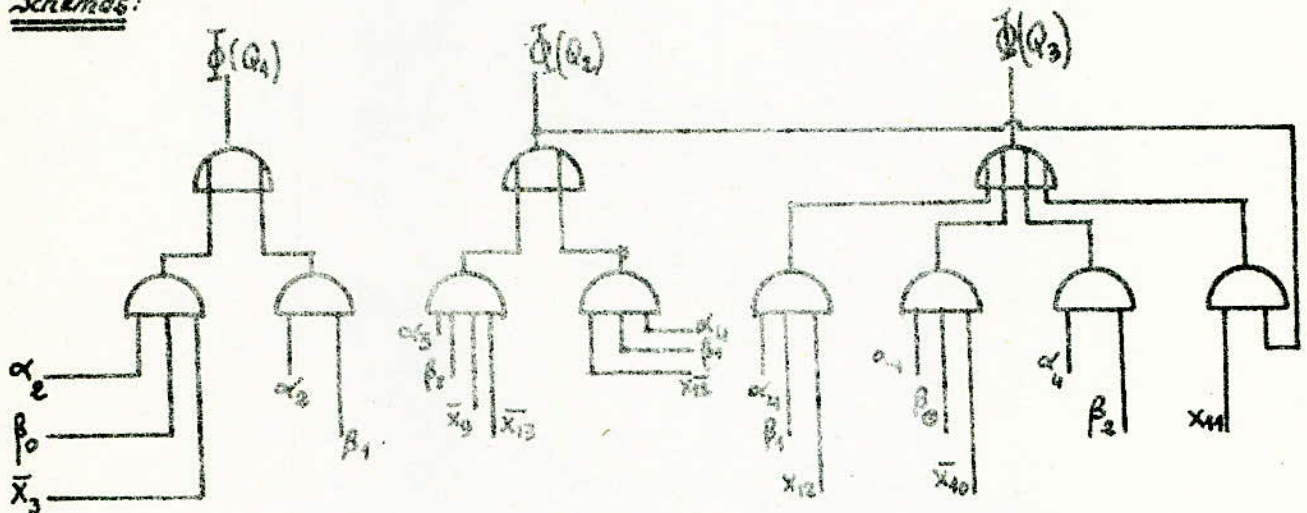
$$\Phi(Q_1) = \alpha_2 \beta_0 \bar{x}_3 + \alpha_2 \beta_1 + \Phi(Q_2) \bar{x}_4 (x_1 + \bar{x}_2)$$

$$\Phi(Q_1) = \alpha_2 \beta_0 \bar{x}_3 + \alpha_2 \beta_1$$

$$\Phi(Q_2) = \alpha_3 \beta_3 \bar{x}_9 \bar{x}_{13} + \alpha_4 \beta_1 \bar{x}_{12}$$

$$\Phi(Q_3) = \alpha_0 \beta_2 + \alpha_4 \beta_1 \bar{x}_{12} + \alpha_4 \beta_0 \bar{x}_{10} + \Phi(Q_2) \bar{x}_{11}$$

Schémas:



5). Fonctions d'excitation et de sortie:

$$R_1 = \Phi(Q_2) + \Phi(Q_3) + \alpha_4 \beta_0 \bar{x}_{10}$$

$$S_1 = \alpha_3 \beta_3 \bar{x}_9 + \Phi(Q_2) \bar{x}_{11} + \alpha_3 \beta_3 \bar{x}_9 \bar{x}_{13}$$

$$R_2 = \alpha_2 \beta_0 \bar{x}_3 + \Phi(Q_1) + S_1$$

$$S_2 = \alpha_1 \beta_2 + \Phi(Q_2)$$

$$R_3 = \Phi(Q_2) + \alpha_1 \beta_2 + \Phi(Q_3) + S_2 = S_2 + S_2 + \Phi(Q_3)$$

$$S_3 = \alpha_0 \beta_2 + \alpha_1 \beta_2$$

$$R_4 = \Phi(Q_3) + \alpha_0 \beta_2 + \alpha_1 \beta_2 + \Phi(Q_2) \bar{x}_4 + \alpha_2 \beta_1 + \alpha_3 \beta_3 \bar{x}_9 + \Phi(Q_2) \bar{x}_{11}$$

$$R_4 = \alpha_0 \beta_2 + \alpha_1 \beta_2 + \alpha_3 \beta_3 \bar{x}_9 + \Phi(Q_2) \bar{x}_4 + \Phi(Q_2) \bar{x}_{11} + \Phi(Q_3)$$

$$S_4 = \alpha_0 \beta_2 + \alpha_1 \beta_2 + \Phi(Q_2) \bar{x}_4 + \alpha_2 \beta_0$$

$$R_5 = \alpha_0 \beta_2 + \alpha_0 \beta_2 + \alpha_1 \beta_2 + \alpha_1 \beta_2 + \Phi(Q_2) \bar{x}_4 + \alpha_3 \beta_3 + \alpha_3 \beta_2 + \alpha_3 \beta_2 \bar{x}_9 + \alpha_3 \beta_3 \bar{x}_9 \bar{x}_{13}$$

$$S_5 = \alpha_2 \beta_0 \bar{x}_3 + \Phi(Q_3) + \alpha_0 \beta_2 + \alpha_1 \beta_2 + \alpha_1 \beta_2 + \alpha_2 \beta_1 \bar{x}_4 + \alpha_2 \beta_2 +$$

$$\begin{aligned}
 & + \alpha_3 \beta_0 + \alpha_3 \beta_2 + \Phi(Q_2) \bar{x}_{11} + \alpha_4 \beta_0 x_{10} \\
 S_5 = & \alpha_2 \beta_0 x_3 + \alpha_0 \beta_2 + \alpha_1 \beta_0 + \alpha_1 \beta_2 + \alpha_2 \beta_3 + \alpha_3 \beta_0 + \alpha_3 \beta_2 + \alpha_4 \beta_2 \bar{x}_4 + \Phi(Q_3) + \\
 & + \Phi(Q_2) \bar{x}_{11} + \alpha_4 \beta_0 x_{10} \\
 y_1 = & \alpha_3 \beta_2 x_8 \quad y_2 = \alpha_0 \beta_3 \quad y_3 = \alpha_1 \beta_0 \\
 y_4 = & \alpha_2 \beta_0 \bar{x}_4 x_3 + \Phi(Q_2) x_4 \bar{x}_5. \\
 y_5 = & \alpha_4 \beta_2 x_{12} + \alpha_1 \beta_2 + \Phi(Q_2) \bar{x}_4 + \alpha_3 \beta_3 x_9 + \Phi(Q_2) \bar{x}_{11}. \\
 y_6 = & \alpha_2 \beta_0 x_4 x_3 \quad y_7 = \alpha_3 \beta_3 x_9 \quad y_8 = \alpha_0 \beta_2 \\
 y_9 = & \alpha_2 \beta_3 x_6 + \alpha_3 \beta_2 x_6 = (\alpha_2 \beta_3 + \alpha_3 \beta_2) x_6. \\
 y_{10} = & \alpha_2 \beta_0 x_1 x_3 + \alpha_3 \beta_0 \bar{x}_7 \quad y_{11} = \alpha_4 \beta_0 x_7 \quad y_{12} = \Phi(Q_2) \bar{x}_{11}. \\
 y_{13} = & \alpha_4 \beta_2 x_{12} + \Phi(Q_2) x_4 \bar{x}_4 + \alpha_3 \beta_3 x_9 \quad y_{14} = \alpha_3 \beta_0 + \alpha_3 \beta_2 x_8. \\
 y_{15} = & \alpha_4 \beta_2 x_{12} + \Phi(Q_2) x_4 \bar{x}_4 + \alpha_3 \beta_3 x_9 \\
 y_{16} = & \Phi(Q_2) \bar{x}_4 \bar{x}_4 + \Phi(Q_2) \bar{x}_{11} \quad y_{17} = \alpha_0 \beta_2 \\
 y_{18} = & \Phi(Q_2) x_2 + \alpha_1 \beta_2 x_2 + \alpha_2 \beta_2 x_2 = x_2 (\alpha_1 \beta_2 + \alpha_2 \beta_2 + \Phi(Q_2)). \\
 y_{19} = & \alpha_3 \beta_3 \bar{x}_9 x_{13} \quad y_{20} = \alpha_2 \beta_0 x_4 x_3 + \Phi(Q_2) x_4 x_5. \\
 y_{21} = & \alpha_2 \beta_0 x_1 x_3 + \alpha_1 \beta_2 x_2 + \Phi(Q_2) x_4 x_5. \\
 y_{22} = & \alpha_1 \beta_2 \bar{x}_2 \quad y_{23} = \alpha_0 \beta_2 \quad y_{24} = \Phi(Q_2) \bar{x}_4 \bar{x}_4 \quad y_{25} = \alpha_0 \beta_1 \\
 y_{26} = & \alpha_0 \beta_2 = y_{27} \quad y_{28} = \alpha_4 \beta_3 \quad y_{29} = \alpha_4 \beta_0 x_{10}.
 \end{aligned}$$

Nous ne représenterons pas le schéma de l'automate car la méthode des nœuds nous a donné des expressions presque identiques aux précédentes.

6.2. Codage rationnel des états:

- L'un des problèmes essentiels dans les méthodes de synthèse des automates est le codage des états, car en utilisant différents codes nous obtenons des expressions différentes des fonctions d'excitation et de sortie.

- Il existe plusieurs méthodes de codage rationnel, mais la plupart d'entre elles sont très complexes, nous allons exposer celle qui a été mise au point par M^r MOROZ (Savant Soviétique).

6.2.1. Définitions:

* Distance entre codes: La distance entre les codes $C(a_m)$ et $C(a_s)$ est égal au nombre d'éléments mémoire qui changent d'état lors de la transition $a_m \rightarrow a_s$.

$$L_{ms} = |K_m - K_s|^2$$

* Fonction poids: Elle est égale à la somme de toutes les distances entre codes des différentes transitions de l'automate. Elle caractérise la complexité du circuit. $W = \sum \text{tous}$.

* Automate connexe: on dit qu'un automate est connexe s'il peut être représenté par un graphe connexe.

* Graphe connexe: Un graphe $G(X, U)$ est fortement connexe si quelque soient x_i et x_j appartenant à X , il existe un chemin formé d'arcs appartenant à U liant x_i et x_j , tel que $f_i(x_i)$ est un sommet.

6.2.2. Algorithme de la méthode:

Nous allons illustrer cet algorithme par un exemple.

Soit le diagramme de flux suivant:



①. Nous composons la matrice M qui contient toutes les paires différentes (α_i, β_i) où α_i a la transition $q_{\alpha_i} \rightarrow q_{\beta_i}$ de l'automate. (Une ligne de la matrice correspond à une flèche.)

$$M = \begin{pmatrix} \alpha_1 - \beta_1 \\ \alpha_2 - \beta_2 \\ \vdots \\ \alpha_R - \beta_R \end{pmatrix}$$

dans notre exemple

$$M = \begin{pmatrix} 1 - 2 \\ 2 - 4 \\ 2 - 5 \\ 3 - 2 \\ 4 - 3 \\ 4 - 5 \\ 5 - 4 \\ 5 - 1 \end{pmatrix}$$

②. Nous permutoisons les lignes de la matrice M pour réaliser la condition suivante:

$$\{\alpha_i, \beta_i\} \cap \{\alpha_1, \beta_1, \alpha_2, \beta_2, \dots, \alpha_{i-1}, \beta_{i-1}\} \neq \emptyset \quad \emptyset = \text{ensemble vide}$$

Cette condition signifie qu'un élément de la ligne i se trouve dans n'importe quelle ligne précédente. Dans notre exemple cette condition est réalisée. Pour la ligne 4 nous avons $\alpha_4 = 3; \beta_4 = 2$

$$\{3, 2\} \cap \{1, 2, 4, 5\} = \{2\} \neq \emptyset.$$

③. Codage des états de la première ligne de la matrice M :

$$K_{\alpha_1} (\text{code de } \alpha_1) = 000 \dots 0; \quad K_{\beta_1} = 00 \dots 01$$

Dans notre exemple: $\alpha_1 = 1$ et $\beta_1 = 2$

$$K_{\alpha_1} = 000 \quad \text{et} \quad K_{\beta_1} = 001$$

④. Nous barrons la première ligne car les 2 états a_{α_1} et a_{β_1} sont codés, nous obtenons une nouvelle matrice M' :

$$M' = \begin{array}{|c} 2-4 \\ 2-5 \\ 3-2 \\ 4-3 \\ 4-5 \\ 5-4 \\ 5-1 \end{array}$$

⑤. Nous noterons un élément de la première ligne qui n'est pas codé par le symbole γ .

Dans notre exemple $\gamma = 4$, car l'élément 2 est déjà codé.

Composons la matrice M_γ qui est formée par toutes les lignes de la matrice M' qui contiennent γ .

* $B_\gamma = \{\gamma_1, \gamma_2, \dots, \gamma_f\}$ est l'ensemble des éléments de la matrice M_γ qui correspondent aux états déjà codés.

* Nous codons les éléments $\gamma_1 \dots \gamma_f$.

$$B_4 = \{2\} \quad K_2 = 001$$

$$M_4 = \begin{array}{|c} 2-4 \\ 4-3 \\ 4-5 \\ 5-4 \end{array}$$

⑥. Nous notons tous les codes $C_{\gamma_f}^1$ qui sont adjacents aux codes de γ_i et qui ne sont pas occupés par le codage des autres états de l'automate.

$D_\gamma^1 = \bigcup_{f=1}^F C_{\gamma_f}^1$ est l'ensemble des codes qui sont adjacents aux codes γ_f .

si $D_\gamma^1 = \emptyset$, composer le nouvel ensemble $D_\gamma^2 = \bigcup_{f=1}^F C_{\gamma_f}^2$.

$C_{\gamma_f}^2$ est l'ensemble des codes qui ont la distance codée 2. ($K_{\gamma_f} = 2$).

si $D_\gamma^2 = \emptyset$, dans ce cas il faut composer D_γ^3 et ainsi de suite jusqu'à obtenir $D_\gamma^n \neq \emptyset \{n=1, 2, \dots, n\}$.

$$D_\gamma^n = \{K_{\gamma_1}, \dots, K_{\gamma_g}, \dots, K_{\gamma_n}\}$$

Dans notre exemple $C_2^1 = \{101, 011\}$.

les codes adjacents à 001 sont : 011, 101 et 000 ; mais 000 est déjà utilisé pour coder l'état 1, donc : $D_4^1 = \{101, 011\}$.

⑦. Trouver pour chaque code de D_γ la distance codée K_{γ_f} et tous les codes utilisés $K_{\gamma_f} \{f=0, 1, \dots, n\}$.

$$W_{101} = |K_{\gamma_g} - K_{\gamma_f}|^2 = |101 - 001|^2 = 1$$

$$W_{011} = |101 - 001|^2 = 1.$$

⑨. trouver $W_g = \sum_{f=1}^F W_{gf} \quad g=1, \dots, G.$

$W_{12}; W_{22} \quad W_{101} = W_{011} = 1.$

Si l'automate a la transition de l'état a_{gf} à l'état a_{ge} et inversement $a_{ge} \rightarrow a_{gf}$, alors W_{gf} doit apparaître deux fois.

⑩. De l'ensemble D_g^z nous prenons le code K_g pour lequel W_g est minimum et nous codons l'élément γ (qui correspond à l'état a_γ) par le code $K_{g\gamma}$.

Dans notre exemple $W_g = 1$ dans les deux cas nous pouvons coder K_H par 101 ou 011

101 0	00	01	11	10
0	1	2		
1		4		

⑪. Barrons les lignes de M' dans lesquelles deux états dans déjà codés. S'il ne reste aucune ligne on passe à ⑫, et s'il y a une ligne au moins qui reste on revient à ⑤ et on refait les points ⑤ à ⑩.

$$M^2 = \begin{array}{|l} 2-5 \\ 3-2 \\ 4-3 \\ 4-5 \\ 5-4 \\ 5-1 \end{array}$$

on revient donc au point ⑤.

⑫. $\gamma = 5$

⑬. $\rightarrow M_5 = \begin{array}{|l} 2-5 \\ 4-5 \\ 5-4 \\ 5-1 \end{array}$

; $B_5 = \{2, 4, 1\}$.

⑭. $K_2 = 001$; $K_4 = 101$; $K_1 = 000$.

* codes adjacents à 2 non déjà utilisés : 011

$C_2^1 = \{011\}$.

* codes adjacents à 4 non déjà utilisés : 111; 100

$C_4^1 = \{111; 100\}$.

* codes adjacents à 1 non déjà utilisés : 010; 100

$C_1^1 = \{010; 100\}$.

$$D_5^1 = \{010; 100; 011; 111\}$$

⑧. Pour chaque code de D_5^1 nous trouvons les fonctions poids entre les codes de D_5^1 et tous les codes utilisés dans B_5 . c'est à dire dans la matrice M_5 .

$$W_{011} = |011-001|^2 + |011-101|^2 + |011-101|^2 + |011-000|^2 =$$

$$W_{011} = 1 + 2 + 2 + 2 = 7.$$

$$W_{100} = |100-001|^2 + |100-101|^2 + |100-101|^2 + |100-000|^2 =$$

$$W_{100} = 2 + 1 + 1 + 1 = 5.$$

$$W_{111} = |111-001|^2 + |111-101|^2 + |111-101|^2 + |111-000|^2$$

$$W_{111} = 2 + 1 + 1 + 3 = 7.$$

$$W_{010} = |010-001|^2 + |010-101|^2 + |010-101|^2 + |010-000|^2.$$

$$W_{010} = 2 + 3 + 3 + 1 = 9.$$

⑩. Min de $W_g = 5$

Nous codons l'élément 5 qui correspond à l'état a_5 par 100.

⑪. De la matrice M^1 on barre les lignes dans lesquelles les deux états sont codés. On obtient la matrice M^3 .

$$M^3 = \begin{vmatrix} 3 & -2 \\ 4 & -3 \end{vmatrix}$$

puisque'il nous reste deux lignes dans la matrice on passe au point ⑤. et on continue jusqu'à ⑪.

⑤. $Y = 3$

$$⑥. M_3 = \begin{vmatrix} 3 & -2 \\ 4 & -3 \end{vmatrix} \rightarrow B_3 = \{2-4\}. \quad K_2 = 001$$

$$K_4 = 101.$$

$$⑦. C_2^1 = \{011\} \quad \text{et} \quad C_4^1 = \{111\}.$$

$$\Rightarrow D_3^1 = \{011; 111\}.$$

$$⑧ \text{ et } ⑨. W_{011} = |011-001|^2 + |011-101|^2 = 1 + 2 = 3.$$

$$W_{111} = |111-001|^2 + |111-101|^2 = 2 + 1 = 3.$$

⑩. $W_{011} = W_{111}$ donc on peut coder l'état 3 par 011 ou 111.

On prend par exemple $K_3 = 011$.

⑪. Nous barrons les lignes de M^3 dans lesquels deux états sont déjà codés. Il ne reste aucune ligne car tous les états sont codés. Nous passons au point ⑫.

⑫. On définit la fonction poids $W = \sum t_{ms}$ où $t_{ms} = |K_m - K_s|^2$.

Nous avons le tableau suivant:

états	codes
1	000
2	001
3	011
4	101
5	100

$$W = t_{12} + t_{24} + t_{25} + t_{45} + t_{54} + t_{43} + t_{32} + t_{51} = 1+1+2+2+1+1+1+1 = 10.$$

Le nombre $k = \frac{W}{P}$ est une évaluation de la qualité du codage dans cet algorithme.

P = Nombre de transitions. dans notre il est égal à 8.

$$\Rightarrow k = \frac{10}{8} = 1,25.$$

Remarque: Pour des automates ayant un nombre d'états compris entre 10 et 120 on choisit k tel que: $1,4 < k < 2,1$

Le codage est adjacent lorsque nous avons $k = 1$.

⑨. Fin

Conclusion: L'intérêt de cette méthode réside dans le fait qu'elle est exploitable par ordinateur et pourrait être utilisée pour la réalisation d'automates complexes en vue d'une économie de matériel et d'amélioration de la vitesse d'exécution.

Pour le codage direct que nous avons utilisé dans notre problème nous obtenons:

$$W = 75$$

$$P = 40 \Rightarrow k = \frac{W}{P} = 1,875.$$

D'après la remarque faite plus haut, nous considérons que le codage direct que nous avons utilisé est assez bon.

VII. L'ORDINATEUR SIMULOG:

Cet ordinateur est conçu à des fins pédagogiques. (Compréhension du principe de fonctionnement d'un ordinateur). Le langage de programmation est du type "assembleur" c'est à dire qu'une instruction en code mnémorique est traduite en une instruction en langage machine.

Les entrées des instructions et la sortie du résultat s'effectuent manuellement. Nous allons étudier l'unité centrale qui constitue l'équipement de base de cet ordinateur. Cette unité centrale est constituée de:

- La mémoire centrale.
- L'organe de traitement.
- Les organes de commande.

7.1. Etude des différents modules constituant l'unité centrale:

a). La mémoire centrale:

C'est une association de 16 modules mémoires. Chacun d'eux étant un registre à 8 bits à décalage à droite. Le chargement à l'aide d'un autre registre et la sortie du contenu se font en série. Une impulsion sur l'entrée d'Horloge H fait décaler le contenu d'un bit vers la droite. Pour le chargement et pour le transfert il nous faut une série de 8 impulsions. Chaque module mémoire peut être chargé manuellement en valeurs binaires par application d'un signal "0" aux entrées $\bar{5}$ des mémoires élémentaires (bascules).

Chaque sortie du décodeur d'adresses est reliée à l'entrée d'Horloge H d'une mémoire.

b). Organe de traitement:

Cet organe de l'ordinateur effectue les opérations arithmétiques et logiques. Il comprend: L'accumulateur, l'unité de calcul et le registre d'index.

- L'unité de calcul: Sa partie principale est un additionneur complet du type combinatoire. L'addition de deux nombres binaires se fait en série à l'aide de 8 impulsions d'Horloge. Toutes les instructions (opérations) sont codées. Si par exemple nous avons un signal "1" à l'entrée ADD on effectue la somme des bits parvenant de l'accumulateur et d'un module mémoire en tenant compte d'une retenue éventuelle. Pour la soustraction, au contenu de l'accumulateur nous ajoutons

le complément vrai du contenu d'un module mémoire.

- La multiplication de deux nombres se fait par additions successives.

- La division se fait par soustractions successives.

- L'accumulateur :

Comme le module mémoire, l'accumulateur est un registre à décalage à droite constitué par 8 mémoires élémentaires.

Les quatre dernières mémoires élémentaires peuvent être chargées en parallèle. L'accumulateur sert à mémoriser l'un des opérandes et le résultat.

- Registre d'Index : Il est réalisé à l'aide du compteur avec commutateur dont la partie principale est un compteur à 4 bits. On l'utilise pour le branchement conditionnel. Si le contenu d'un registre d'index est supérieur à 0, le programme se branche à la nouvelle adresse et le contenu du compteur est diminué d'un bit. À l'aide d'un signal de commande, le commutateur électronique permet la commutation simultanée des 4 bits du compteur.

c) Organes de Commande :

Ils élaborent les signaux de commande pour un déroulement correct du programme. Ces organes de commande sont :

- Le synchroniseur.

- Le registre d'instructions

- Les décodeurs d'opération et d'adresse et le compteur ordinal.

* Synchroniseur : C'est un générateur d'impulsions de fréquence 5 Hz muni d'un compteur à 4 bits. On peut émettre des impulsions individuelles à l'aide d'un bouton poussoir. On peut aussi émettre une série de 8 impulsions en actionnant le bouton poussoir.

* Registre Instructions :

Son rôle est de conserver l'instruction pendant son interprétation. Il comprend deux parties. 4 bits pour le code de l'opération à effectuer et 4 bits pour l'adresse de l'opérande. Il est réalisé à l'aide d'un module mémoire. Pour traiter et exécuter une instruction il nous faut deux cycles de 8 impulsions.

Pendant le 1^{er} octet d'impulsions le contenu du module mémoire d'adresse indiquée par le compteur d'adresses instructions

Mémoire	Fonction	Code machine
0	$RE_1 \rightarrow ACC$	0011 1100
1	$ACC - RE_2 \rightarrow ACC$	1000 1101
2	$ACC \rightarrow RX.$	1111 0000
3	$MB \rightarrow ACC$	0011 1111
4	$ACC = ACC/2.$	0101 0000
5	$BRX > 0 \text{ goto } 4.$	1011 0100
6	$ACC + MA \rightarrow ACC.$	0111 1110
7	$ACC \rightarrow MB.$	0010 1111.
8	$EA \rightarrow ACC$	0011 1100
9	$ACC \rightarrow EB.$	0010 1101
10	FIN	0000 0000
11		
12	EA	XXXX XXXX
13	EB	XXXX XXXX
14	MA	XXXX XXXX
15	MB	XXXX\ XXXX

Les résultats sont dans les registres du nombre B

$MR \rightarrow MB.$

$ER \rightarrow EB.$

Si le résultat est négatif on l'obtient dans le complément à 2ⁿ.

VIII. CONCLUSION:

Cette étude nous a permis de comprendre quelques méthodes utilisées pour la conception d'automates synchrones à plusieurs états. Elle nous a permis d'avoir une idée sur le fonctionnement des machines calculatrices.

Nous aurions aimé faire la réalisation pratique de cette U.A.L. d'addition en virgule flottante pour connaître nos possibilités de réalisation et avoir une idée de ses performances. Ceci n'a malheureusement pas été possible en raison du manque de matériel à l'école.

Ce travail ne constitue pas une fin en soit, des améliorations sont possibles; par exemple en ajoutant certaines instructions on pourrait réaliser d'autres opérations. Il pourrait servir de base pour tous qui s'intéressent aux automatismes.