1/71
2ex

UNIVERSITE D'ALGER
ECOLE NATIONALE POLYTECHNIQUE

Département Télécommunications

المدرسة الوطنية للعلوم الهندسية
ECOLE NATIONALE POLYTECHNIQUE
BIBLIOTHEQUE

THESE DE FIN D'ETUDES

Electronique

Etude et Réalisation d'une Unité de Calcul à Circuits Intégrés

JURY

Président : **Mr A. BENTCHIKOU** Maître de Conference

Promoteur : **Mme G. MONDON** Ingénieur

Assesseur : **Mr P. FLORET** Ingénieur

Invité **Mr Y. FLAMANT** Professeur à l'Université d'Alger

Proposée par :
Mme G. MONDON

Etudiée et Réalisée par :
Mr B. Ahmed Bey

PROMOTION 71

UNIVERSITE D'ALGER
ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT TELECOMMUNICATIONS

THESE DE FIN D'ETUDES

ETUDE ET REALISATION D'UNE UNITE DE CALCUL A CIRCUITS INTEGRES

JURY : Président : Mr. A. BENTCHIKOU : Maître de Conférences
Promoteur : Mme. G. MONDON : Ingénieur
Assesseur : Mr. P. FLORET : Ingénieur
Invité : Mr Y. FLAMANT : Professeur à l'Université d'Alger.

Proposée par :
Mme. G. MONDON

Etudiée et réalisée par :
Mr. B. AHMED-BEY

PROMOTION 1971

à mes parents

AVANT-PROPOS

Je tiens à remercier vivement Mme Georgette MONDON pour tous les conseils judicieux qu'elle a bien voulu me donner durant toute l'étude du projet et qui m'ont permis d'aboutir à sa réalisation.

Je ne manquerai pas d'exprimer aussi ma gratitude et ma reconnaissance à tous mes professeurs et assistants qui ont contribué à ma formation.

Que Mr.Y.FLAMANT trouve ici mes vifs remerciements pour l'aide précieuse qu'il m'a apportée lors du tirage du circuit imprimé.

Mes remerciements vont également à Mr.DRIOUCHE technicien au laboratoire de Télécommunications pour sa précieuse collaboration lors de la réalisation pratique.

B.AHMED-BEY

SOMMAIRE

Pages

INTRODUCTION 1

1ère PARTIE: NOTIONS FONDAMENTALES 3

Chap 1. Organisation générale d'un calculateur numérique 4

Chap 2. Eléments de logique 9

Chap 3. Système de numération et de codage 11

Chap 4. Circuits logiques intégrés 16

2ème PARTIE: CIRCUTS DE CALCULS 21

Chap 1. Additionneur-soustracteur binaire 22

Chap 2. Multiplieur binaire 30

Chap 3. Diviseur binaire 35

Chap 4. Comparateur binaire 39

3ème PARTIE: SYSTEME D'ENTREE - SORTIE 42

Chap 1. Etude des régistre SN 7475 N 43

Chap 2. Circuits de commande de sortie des opérations 46

4ème PARTIE: CONSTRUCTION DE L'UNITE DE CALCUL 49

ANNEXE 1 : UNITE ARITHMETIQUE ET LOGIQUE (Cas des nombres représentés en valeur absolue et signe) 54

ANNEXE 2 : UNITE ARITHMETIQUE ET LOGIQUE EN VIRGULE FLOTTANTE; 67

BIBLIOGRAPHIE 68

INTRODUCTION

Il existe actuellement deux grandes familles de calculateurs électroni--ques :

- CALCULATEURS ANALOGIQUES:ce sont des machines électroniques servant à l'étude des phénomènes où interviennent des grandeurs physiques à varia--tion continue.Ils permettent de réaliser des similateurs c'est-à-dire de des appareils qui sont en correspondence étroite avec le phénomène qu'ils représentent.Mais ils sont si étroitement spécialisés qu'ils ne se prê--tent pratiquement qu'à un seul genre d'étude,voire même une seule étude particulière.Par exemple,les analyseurs différentiels servent à résoudre les équations différentielles alors que les calculateurs de tir permet--tent de déterminer la position d'une cible à une certaine époque.Leur précision est très limitée.

- CALCULATEURS NUMERIQUES: ils sont à la base de la " révolution informati--que ".Ils mettent en oeuvre les principes fondamentaux du calcul arith--métique et assurent ainsi des fonctions logiques plus ou moins importan--tes et qui font d'ailleurs que l'on voit en eux des émules du cerveau humain,capables d'activités apparentées à l'intelligence.Un des attributs des calculateurs numériques est leur universalité;Parler des machines à traduire,à établir des factures et bulletins,etc,c'est laisser croire qu'il existe un grand nombre de types différents.Or,de par sa construction même,le propre d'un calculateur numérique ou" ordinateur "est de ne pouvoir accomplir certaines opérations arithmétiques et logiques très élémen--taires.C'est la combinaison ordonnée d'une multitude de telles opérations successives qui lui permet de traiter tel ou tel problème.

L'attribut majeur de l'ordinateur est cependant sa rapidité d'action qui a véritablement révolutionné le traitement de l'information.Cette rapidité il la doit à l'électronique qui a permis d'éliminer totalement dans son fonctionnement interne les éléments mécaniques doués d'inertie.

En dix ans,nous avons vu se succéder trois générations de calculateurs électroniques :

- 1ère génération : numération décimale,tubes,mémoires à tambour magnétique (IBM 650).

2ème génération : numération décimale,transistors,mémoires à ferrite (IBM 1401).

3ème génération : numération à la fois décimale et binaire,mémoire à ferrite dont la technologie est diversifiée,circuits intégrés.

De ce fait,les opérations y sont effectuées à une vitesse prodigieuse. Avec les semi-conducteurs,sous forme de diodes ou de transistors,une multiplication ne prenait guère qu'une milliseconde,il y a peu de temps encore.Avec les circuits intégrés des micromodules on atteint la microseconde.La durée se réduit aux limites de l'imaginable car actuellement on chiffre les commutations en nanosecondes.

A sa souplesse d'adaptation à des problèmes de traitements variés,à sa remarquable rapidité d'exécution des opérations arithmétiques et logiques,à sa capacité de mémorisation,l'ordinateur ajoute une qualité précieuse : Fiabilité.

Il est à peine besoin de mentionner les services irremplaçables des calculateurs électroniques dans le domaine scientifique et technique.

Cependant,notre étude ne se basera que sur la partie centrale d'un calculateur numérique : L'UNITE DE CALCUL.Celle-ci permetd'effectuer une des opérations arithmétiques quelconques sur des nombres de quatre bits.

L'étude de cette unité de calcul se divisera en quatre parties :

1. Notions fondamentales
2. Circuits de calcul
3. Système d'entrée-sortie
4. Construction de l'unité de calcul.

1 ère PARTIE

NOTIONS FONDAMENTALES

CHAPITRE 1 : ORGANISATION GENERALE D'UN CALCULATEUR NUMERIQUE

1.Généralités:

1.1.Définition d'un algorithme:

On appelle algorithme l'ensemble des prescriptions précises qui indi--quent d'exécuter,dans un ordre déterminé,une certaine succession d'opéra--tions permettant d'obtenir la résolution de tous les problèmes d'un type donné.Il se présente comme un mode opératoire.Pour exécuter un algorithme, on définit d'abord un mode de codage de l'information.

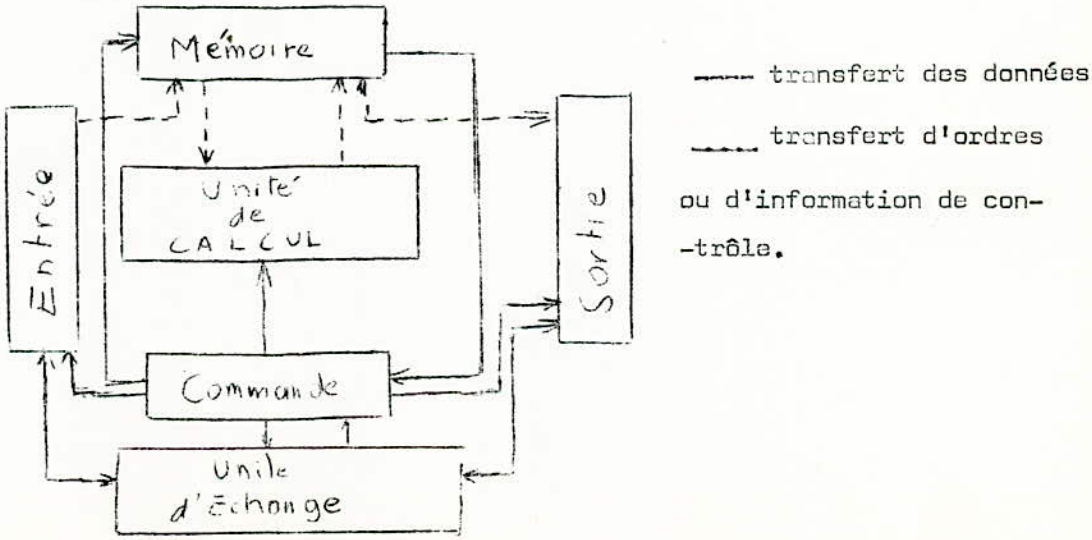
1.2.Programme et structure d'une instruction:

On appelle programme la suite d'instructions nécessaires à l'exécution d'une opération.Ces instructions indiquent à chaque intant les opérations que doit exécuter le calculateur.Ce programme est enregistré dans le calcula--teur avant son utilisation et peut être modifié suivant les besoins et les applications à traiter.

Une instruction est une opération de base que le calculateur est capa--ble de déclencher en une seule fois.On distingue des instructions à plu--sieurs ad esses(1,2,3,et 4).L'instruction à une simple adresse se compose de deux mots:le premier étant réservé au code opératoire(indiquant la nature de l'opération à effectuer) et le second à l'adresse de l'opérande.

2.Blocs fonctionnels d'un calculateur numérique:

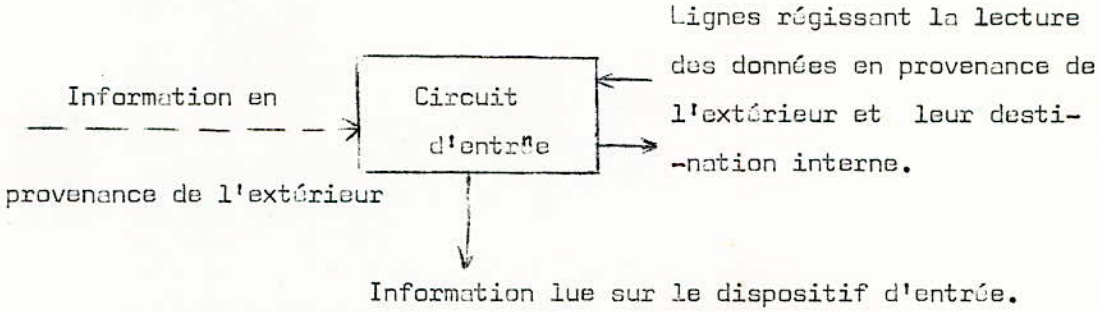
D'une façon générale,le bloc-diagramme d'un calculateur numérique peut se présenter comme suit :



Les fonctions d'un calculateur sont exercées par six organes principaux: les circuits d'entrée-sortie,la mémoire,l'unité de calcul,les circuits de com--mande et de contrôle et l'unité d'échange.

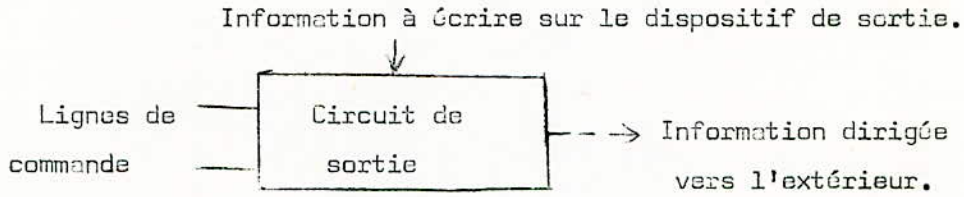
2.1 Les circuits d'entrée:

Ils permettent au calculateur d'ingérer l'information en provenance de l'extérieur et servent de traducteur du langage extérieur vers le langage in--terne à la machine.



2.2. Les circuits de sortie:

Ils jouent le même rôle que les circuits d'entrée et délivrent l'infor--mation traitée vers l'extérieur.



2.3. La mémoire:

Elle sert à contenir le programme et les données.Elle se présente comme un certain nombre de positions ayant un accès unique.Chaque position est affec--tée d'un numéro constituant l'adresse de l'information qui s'y trouve enre--gistrée.L'adressage de la mémoire permet la localisation et la sélection des informations.

Dans une mémoire,il y a deux fonctions à réaliser:

- stockage de l'information
- sélection de la position choisie extérieurement.

D'autre part,cette mémoire est caractérisée par:

- sa vitesse de fonctionnemnt qui se mesure par le temps de cycle de la mé--moire : c'est le temps minimum qui sépart deux opérations consécutives à des adresses arbitraires soit de lecture soit d'écriture.

- le temps d'accès séparant l'instant où l'on présente une demande de lecture à l'entrée de l'instant auquel l'information est disponible sur les connections de sortie. Ce temps d'accès est fonction de la technologie employée.

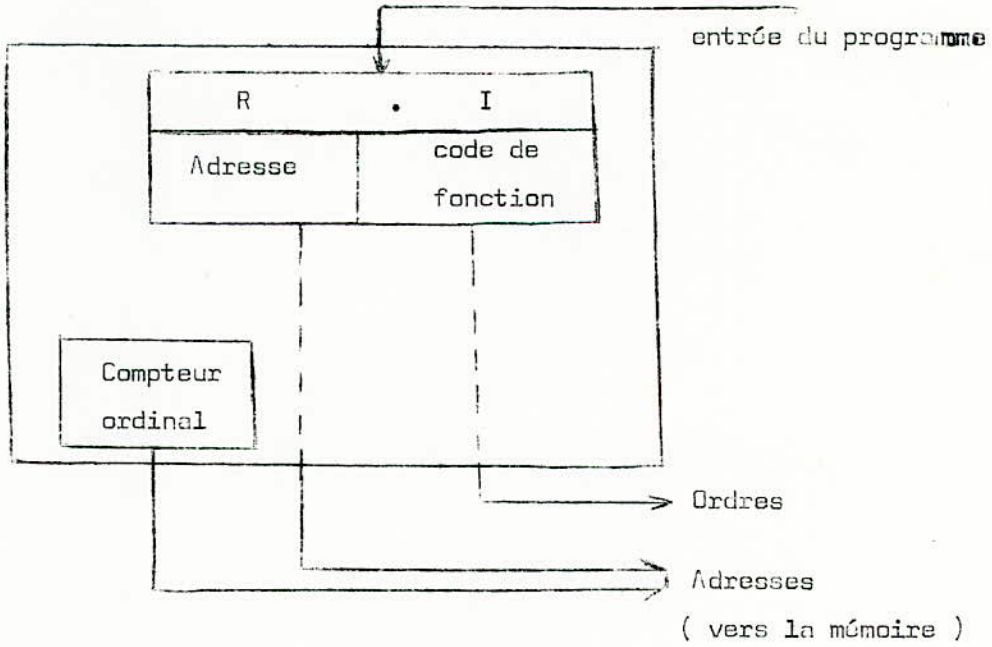
Le support de l'information en mémoire est constitué, dans la plupart des machines actuelles, par des ferrites ou tores magnétiques.

2.4. Unité de calcul:

C'est le bloc où s'effectue le calcul proprement dit (voir pagraphe 3).

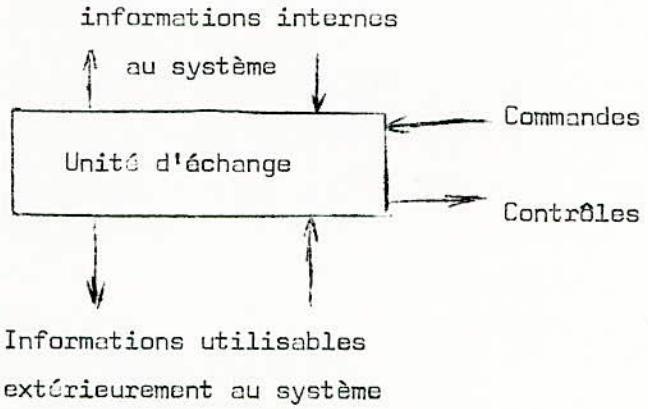
2.5. Unité de commande et de contrôle:

C'est le bloc qui organise et coordonne le déroulement d'un programme. Il comporte une entrée instruction, envoyée de la mémoire, placée dans un régistre instruction R.I. Celui-ci contient l'adresse et le code de fonction. A partir de ces informations, le bloc de commande et de contrôle fournit des ordres à la mémoire et aux autres organes leur indiquant les opérations qu'il y a lieu d'exécuter et les adresses des informations sur lequelles portent ces opérations. Un compteur ordinal permet de retenir à chaque instant le numéro de la position de lá mémoire d'où a été extraite l'instruction en cours de traitement. Le bloc de commande et de contrôle peut être représenté comme suit :



2.6.L'unité d'échange:

C'est une unité qui ne modifie pas l'information mais opère un change--ment de nature physique sur le mode de codage de l'information.Par exemple,si un calculateur est construit pour calculer en numération binaire,il faut évidemment que les données parviennent en binaire dans les circuits de calcul. Mais l'opérateur les fournit généralement en numération décimale.L'unité d'échange réalise cette conversion binaire-décimale.Elle peut se présenter comme suit :

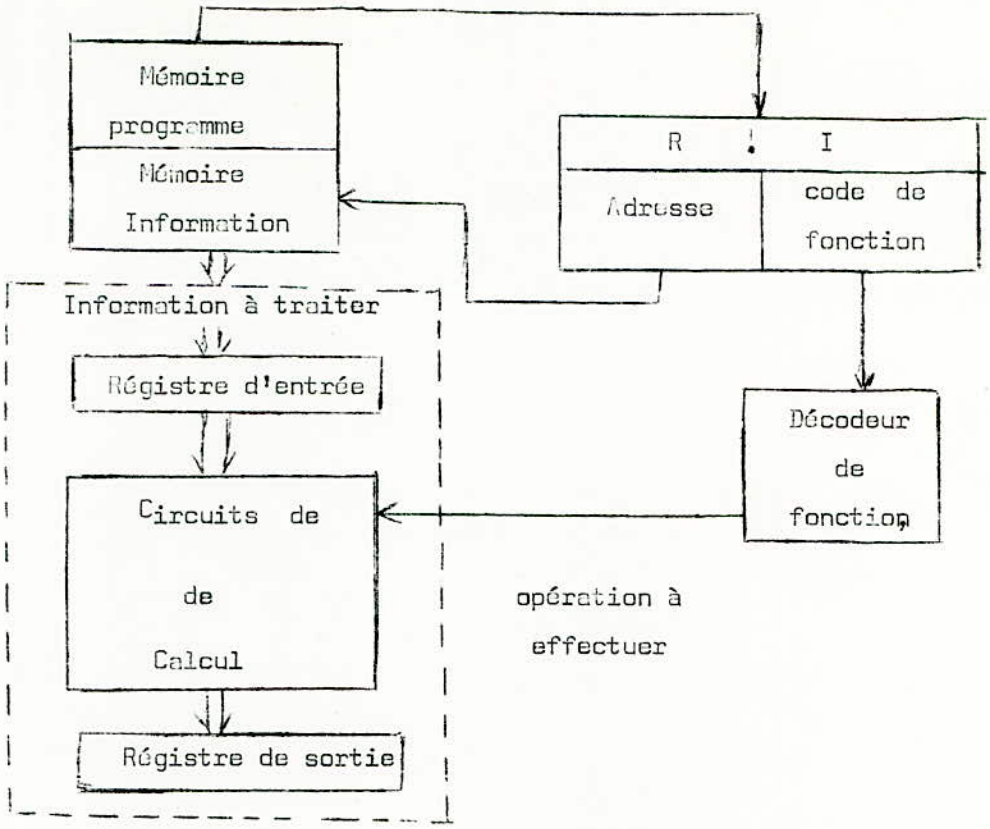


3.L'unité de calcul :

L'unité de calcul est la partie centrale du calculateur numérique qui effectue les calculs nécessaires au déroulement d'un programme donné.Elle doit être en mesure de résoudre les quatre opérations élémentaires:addition,sous--traction,multiplication et division.Ces opérations se réduisent en général à une succession d'additions.L'unité de calcul comprend un régistre d'entrée qui reçoit les informations à traiter sous forme d'un ou de deux mots simul--tanés contenant les opérandes,un régistre de sortie qui emmagasine le résul--tat de l'opération et les circuits de calcul.

Le schéma de l'unité de calcul est représenté comme suit :

Suivant le code opératoire,délivré par l'unité de commande,l'unité de calc



Suivant le code opératoire délivré par l'unité de commande,l'unité de ca calcul met en coordination toutes les portes nécessaires à l'opération deman--dée.La construction de cette unité est indépendante de la nature des opéra--tions à effectuer.Seul le nombre des régistres utilisés et leur fonctionnement diffère suivant la complexité des opérations.

On distingue deux types d'unité de calcul:

- l'unité de calcul série :comporte un seul circuit de calcul pour chaque fonction;tous les éléments binaires successifs des nombres traités passent successivement en série dans ce circuit unique.C'est une machine lente mais économique.

- l'unité de calcul parallèle:comporte autant de circuits de calcul à fonctionnement simultané qu'il y a d'éléments binaires dans les mots traités. C'est une machine rapide mais très coûteuse.

CHAPITRE 2 . ELEMENTS DE LOGIQUE

1.Rappels sur l'algèbre de BOOLE :

1.1.Postulats:

$$a + a = a \qquad a \times a = a$$

$$a + 0 = a \qquad a \times 0 = 0$$

$$a + 1 = 1 \qquad a \times 1 = a$$

1.2.Propriétés:

$$\bar{\bar{a}} = a \qquad a(b+c) = ab + ac$$

$$a + \bar{a} = 1 \qquad (a + b)(a + \bar{b}) = a$$

$$a.\bar{a} = 0 \qquad a + \bar{a}.b = a + b$$

1.3.Théorèmes de DE MORGAN:

* $$\overline{a_1 a_2 \ldots a_n} = \sum_{i=1}^{n} \bar{a}_i$$

* $$\overline{a_1 + a_2 + \ldots + a_n} = \prod_{i=1}^{n} \bar{a}_i$$

2.Opérations arithmétiques:

Addition	Soustraction
0 + 0 = 0	0 - 0 = 0
0 + 1 = 1	0 - 1 = 1 (retenue 1)
1 + 0 = 1	1 - 0 = 1
1 + 1 = 0 (retenue 1)	1 - 1 = 0

3.Niveaux logiques:

En général,les circuits logiques comportent des bornes d'entrée et des bornes de sortie.Les tensions appliquées aux bornes d'entrées et recueillies aux bornes de sortie peuvent prendre deux niveaux logiques:

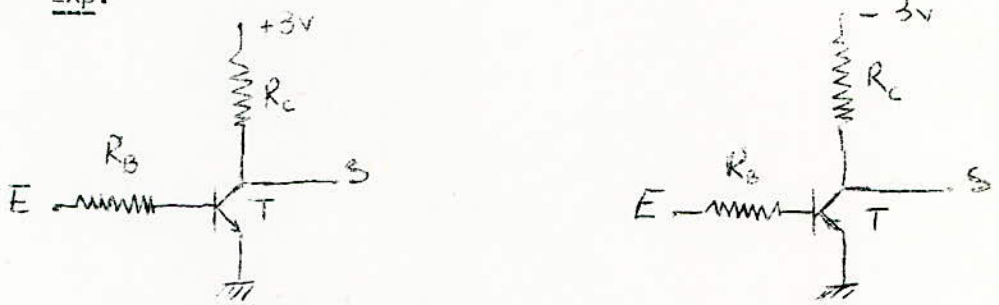
- niveau haut H auquel correspond le plus fort voltage
- niveau bas B " " " faible voltage.

La présence du niveau H caractérise la logique dite positive.Ainsi le niveau H exprime le "1" logique et le niveau bas B exprime le "0" logique.

Si le 0 logique correspond au niveau H et le 1 logique au niveau B,la logique est dite négative.

En général, avec les circuits logiques intégrés, on préfère utiliser la logique positive. En effet, la borne de sortie des circuits logiques intégrés est, dans la plupart des cas, constituée par le collecteur d'un transistor NPN, monté en émetteur commun (à la masse) et saturable. Dans ces conditions, le niveau bas de sortie correspond à la tension de saturation V_{CE} très voisine de 0 volt. Ce 0 Volt évoque donc le "0" logique et on est tenté de faire coïcider ces 2 zéros. Pour des transistors PNP, le 0 Volt correspond au niveau haut.

Exp:



a) Transistor NPN b) Transistor PNP

Si E=+3V(H) $\begin{cases} \text{T est saturé} \\ S=V_{CEsat}=+0,3V(B) \end{cases}$

Si E=-3V(B) $\begin{cases} \text{T est saturé} \\ S=V_{CEsat}=-0,3V(H) \end{cases}$

Si E=0,3V(B) $\begin{cases} \text{T est bloqué} \\ S=+3V\ (H) \end{cases}$

Si E=-0,3V(H) $\begin{cases} \text{T est bloqué} \\ S=-3V\ (B) \end{cases}$

Par application des théorèmes de De Morgan, un circuit réalisant la fonction ET en logique positive, réalise la fonction OU en logique négative et inversement. Il en est de même pour les fonctions NOR et NAND.

Remarque:

Dans toute la suite de l'étude, on raisonnera toujours avec la logique positive.

CHAPITRE 3. SYSTEME DE NUMERATION ET DE CODAGE

1. Généralités:

Dans le système de numération décimale, auquel nous sommes habitués, la représentation physique d'un chiffre qui peut prendre une valeur quelconque de 0 à 9, nécessite l'emploi d'un organe susceptible de prendre dix états distincts caractérisant chacun l'un des dix chiffres de numération. Mais les organes des calculateurs numériques qui sont de nature électrique, magnétique, ou électronique ne permettent en effet de distinguer que deux états. Ainsi les premières machines utilisaient des relais mécaniques. Un relai ne peut être qu'ouvert ou fermé. Plutard, les relais ont été remplacés par des tubes électroniques et des transistors qui ne peuvent également prendre que deux états distincts: un tube ou un transistor est saturé ou bloqué. L'arithmétique et les calculs effectués par la machine seront donc tributaires de cette nature particulière des circuits. Ceux-ci ne connaissant et ne prenant que deux états on leur attribue les valeurs '0' et '1' qui correspondent à un circuit ouvert ou fermé. C'est de cette constatation qu'est née l'idée d'utiliser dans les calculateurs numériques le système de numération à base 2 ou " système binaire ".

2. Système binaire:

On appelle bit ou digit l'un des symboles 0 ou 1 utilisés pour représenter un nombre sous sa forme binaire. La base du système binaire étant 2, on affecte à chaque bit les poids correspondants $2^0, 2^1, \ldots, 2^n$, le poids 2^0 correspond au bit le plus à droite. Ainsi, le nombre 110110 s'écrit :

$$110110 = 1.2^5 + 1.2^4 + 0.2^3 + 1.2^2 + 1.2^1 + 0.2^0 = 54$$

$$\text{soit} \quad (110110)_2 = (54)_{10}$$

Dans le cas des nombres fractionnaires, on attribue aux bits de la partie fractionnaire des poids inférieurs à 1 soit $2^{-1}, 2^{-2}, \ldots, 2^{-n}$. On utilise un point pour séparer la partie entière de la partie fractionnaire. Ainsi, le nombre 110.11 s'écrit :

$$110.11 = 1.2^2 + 1.2^1 + 0.2^0 + 1.2^{-1} + 1.2^{-2} = 6,75$$

$$\text{soit : } (110.11)_2 = (6,75)_{10}$$

D'une façon générale, dans un système de numération à base 2, un nombre quelconque s'écrit :

$$X_n X_{n-1} \ldots X_0 . X_{-1} X_{-2} \ldots X_{-m} = X_n 2^n + \ldots + X_0 2^0 + X_{-1} 2^{-1} + \ldots + X_{-m} 2^{-m}$$

Remarque:

Outre le système binaire,ontrouve les systèmes de numération à base 3 (ternaire),à base 8 (octal) et à base 16 (sexadécimal).Notons que le système à base 60 est encore utilisé dans la mesure du temps et de la température.

3.Conversion décimal-binaire:

Etant donné que les nombres traités dans les calculateurs numériques sont des nombres binaires,il est nécessaire de pouvoir convertir un nombre déci--mal en un nombre binaire.

Dans le cas des petits nombres,on soustrait successivement d'un nombre quelconque les plus hautes puissances de 2 et on attribue le bit 1 à ces puis--sances si elles existent et le bit 0 si elles n'existent pas.

Exp: soit à convertir le nombre décimal 29 .

$$29 - 2^4 = 13 \qquad 13 - 2^3 = 5 \qquad 5 - 2^2 = 1 \qquad 1 - 2^0 = 0$$

d'où : $2^4\ 2^3\ 2^2\ 2^1\ 2^0$

1 1 1 0 1

$$(29)_{10} = (11101)_2$$

Dans le cas des grands nombres,il suffit de diviser un nombre donné suc--cessivement par 2.La succession des restes donne le nombre binaire.Le dernier reste correspond au plus fort poids.

Exp:

53 | 2
13 26 | 2
1 0 13 | 2
1 6 | 2
0 3 | 2
1 1 | 2
1 0

$$(53)_{10} = (110101)_2$$

Pour convertir un nombre décimal fractionnaire,on décompose ce nombre en une partie entière et une partie décimale dont on calcule séparément leur représentation binaire.Puis les deux parties converties seront juxtaposées côte à côte.

Pour des nombres fractionnaires de faible longueur,on retranche successi--vement du nombre à convertir les plus hautes puissances négatives de 2 jusqu'à épuisement du reste ou de la précision désirée.

Exp:

$$0.943 - (1.2^{-1}) = 0.443$$
$$0.443 - (1.2^{-2}) = 0.193$$
$$0.193 - (1.2^{-3}) = 0.068$$
$$0.068 - (1.2^{-4}) = 0.004$$

$$(0.943)_{10} = (0.1111)_2$$

Pour des nombres fractionnaire de grandes longueur,on multiplie par 2 la partie décimale.Si un 1 apparaît à gauche du point décimal,on place un 1 juste après le point décimal du nombre binaire en formation.Si après multiplication par 2 un 0 apparaît à gauche du point décimal du nombre à convertir,on marque un 0 à droite du point décimal du nombre en formation.

Exp:

0.843 x 2 =	1.686	0.1
0.686 x 2 =	1.372	0.11
0.372 x 2 =	0.744	0.110
0.744 x 2 =	1.488	0.1101

soit $(0.843)_{10} = (0.1101)_2$ par défaut.

4.Représentation des nombres quelconques :

Il existe deux modes de représentation des nombres:

4.1.Représentation en virgule fixe :

La convention de représentation des nombres entiers consiste à supposer que la virgule séparant la partie entière de la partie fractionnaire d'un nombre est toujours située immédiatement à droite du dernier chiffre de ce nombre.Si l'ordre de grandeur des informations traitées reste toujours le même,il est possible de supposer la virgule placée à un endroit quelconque du mot.Dans ce cas les circuits de calcul sont indépendants de l'endroit de la virgule.Pour un câblage donné de l'unité de calcul,on pourra considérer que la virgule peut être à un endroit ou à un autre;mais une fois fixée,toute la suite du programme devra se dérouler en respectant la même convention.

4.2.Représentation en virgule flottante:

Pour éviter les difficultés de cadrage des informations et également pour accroître la capacité des régistres,certains calculateurs sont équipés d'unité arithmétique fonctionnant en virgule flottante.Le principe consiste à séparer en deux termes différents l'ordre de grandeur et les chiffres significatifs d'un nombre,ce qui aboutit à une représentation par mantisse et exposant.

Une grandeur A s'écrit sous la forme : $A = M \times b^E$

La mantisse M est toujours normalisée dans un intervalle donné, délimité par deux valeurs extrêmes M_0 et M_1 telles que $M_0/M_1 = b$. En pratique, il est commode de choisir b égal à une puissance entière de la base de numération dans laquelle M est exprimée.

En décimal : $A = M \times 10^E$ avec $0.1 \leq M < 1$

exp: $272 = 0.272 \times 10^3$

En binaire pur : $A = M \times 2^E$ avec $0.5 \leq M < 1$

exp: $12 = 0.11 \times 2^{100} = 0.75 \times 2^4$

La représentation est dite ' normalisée ' lorsque la valeur de l'exposant E a été ajustée de façon que la mantisse M soit comprise dans l'intervalle de variations qui lui a été assigné.

5. Représentation des nombres algébriques:

Il existe plusieurs systèmes de représentation des nombres algébriques ayant chacun leurs avantages spécifiques.

5.1. Représentation en valeur absolue et signe:

Elle consiste à associer à la représentation des valeurs absolues, un élément binaire égal à 0 pour les nombres positifs et à 1 pour les nombres né--gatifs.

Exp: $(+10)_{10} = (0\ 1010)$

$(-4)_{10} = (1\ 0100)$

Comme en algèbre ordinaire, on effectue les opérations sur les modules et les signes sont traités àséparément.

Considérons un nombre supérieur à un, possédant m bits pour la partie fractionnaire et (n-1) bits pour la partie entière, le signe étant affecté à la position n. Le nombre s'écrit sous la forme :

$$2^n \ 2^{n-1} 2^{n-2} \ldots 2^0 \ 2^{-1} 2^{-2} \ldots 2^{-m}$$
$$X_n \ X_{n-1} X_{n-2} \ldots X_0 . X_{-1} X_{-2} \ldots X_{-m}$$

La valeur algébrique du nombre X s'obtient par la formule :

$$X = (-1)^{X_n} \sum_{k=-m}^{n-1} X_k 2^k \quad \text{avec} \quad \begin{array}{ll} X_n = 0 & \text{pour } X > 0 \\ X_n = 1 & \text{pour } X < 0 \end{array}$$

5.2. Représentation des nombresnégatifs par le complément vrai :

Le principe consiste à changer les 0 par les 1 et les 1 par les 0 et à ajouter un 1 au plus faible poids;

Exp: X = -7 = 1 0111 CV = 0 1000 + 1 = 0 1001

La somme du nombre et de son complément vrai donne un nombre formé uni--quement de 0.En effet,si n est le nombre d'éléments binaires traités dans le système,cette somme donne (0) et un 1 à la position (n+1) qui n'existe pas.

$CV + X = 2^4$.C'est pour cette raison qu'on donne au complément vrai le nom de complément à 2^n.

5.3.Représentation par le complément restreint:

Le complément restreint ou complément à 1 s'obtient en changeant tous les 0 par les 1 et tous les 1 par les 0.

Exp: 100101 CR = 011010

Le complément à 1 est égal au complément à 2 moins 1.

Avec ces deux dernières représentations,on peut réduire la soustraction à une addition du complément .

Exp:

	Complément vrai	Complément restreint
11100	11100	11100
- 10100	+ 01100	+ 01011
01000	101000	100111
	Laisser tomber la retenue	1
		01000

6.Codage:

Pour représenter les nombres décimaux,on utilise le code binaire pur. Toutefois,ce code a l'inconvevient d'entrainer l'écriture de très longues suites de bits 0 ou 1 pour les nombres importants.On a donc très généralemnt recours aujourd'hui à un système mixte:le système décimal codé binaire(BCD)dans lequel chaque chiffre est représenté séparément par son équivalent binaire:

$(57)_{10} = (101\ 111)(BCD)$

En fait,le 8 et le 9 s'écrivent en binaire 1000 et 1001 et 4 positions binaires sont nécessaires dans le système BCD : $(57)_{10}$=(0101 0111)(BCD).

Cependant,les calculateurs doivent être capables de représenter et de traiter bien d'autres informations que des valeurs numériques.Ainsi utilise-t--on les codes 2 parmi 5,le code de Gray,le code biquinaire,etc.

Le choix de la combinaison de 0 et de 1 qui représentera une information donnée est en principe arbitraire.Il est toutefois nécessaire que les codes soient normalisés pour les différents organes d'un même calculateur,pour les différents calculateurs d'un même constructeur ou pour l'ensemble des cons--tructeurs.

CHAPITRE 4. CIRCUITS LOGIQUES INTEGRES

1. Circuits logiques:

On appelle circuits logiques, des circuits qui permettent d'accomplir certaines opérations relevant de l'algèbre de Boole qui est liée étroitement à l'arithmétique binaire. On distingue deux types de circuits logiques: les circuits logiques combinatoires et les circuits logiques séquentiels.

1.1. Circuits logiques combinatoires:

Ce sont des circuits dont les commandes antérieures qui ont été appliquées sur les bornes d'entrée sont sans influence sur le résultat observé, à l'instant considéré, sur les bornes de sortie.

Ces circuits comportent :

- des circuits élémentaires: NON, OU, ET (NOT, OR, AND)
- des circuits du 1er degré: NON-OU, NON-ET (NOR, NAND)
- et des circuits complexes : OU EXCLUSIF, ET-OU-NON, Circuits d'addition et de retenue.

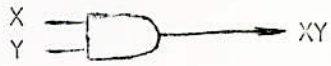
Tableau de vérité :

X	Y	NON $\overline{X}$	NON $\overline{Y}$	ET XY	OU X+Y	NON-OU $\overline{X+Y}$	NON-ET $\overline{XY}$	OU-EXCLUSIF $X \oplus Y$
0	0	1	1	0	0	1	1	0
0	1	1	0	0	1	0	1	1
1	0	0	1	0	1	0	1	1
1	1	0	0	1	1	0	0	0

Symboles:

Fonction NON :

Fonction ET:



Fonction OU :

Fonction NON-OU :

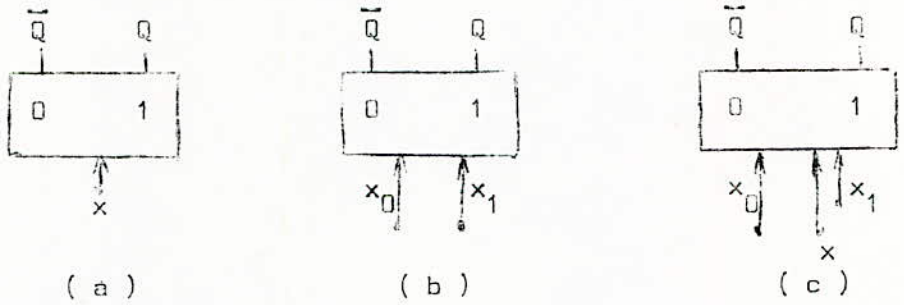
Fonction NON-ET :

OU EXCLUSIF :

1.2.Circuits logiques séquentiels:

Ces circuits délivrent des niveaux de sortie qui dépendent des commandes qui avaient été appliquées à l'entrée,antérieurement à l'instant considéré.Ils possèdent la qualité de mémorisation des états antérieurs.L'élément de base en est la BASCULE.Celle-ci possède deux états stables et est susceptible de chan--ger d'état sous l'effet de signaux de commande appropriés.Les signaux de com--mande sont généralement des impulsions délivrées par une HORLOGE.

Chaque bascule possède 2 sorties complémentaires Q et $\overline{Q}$ et une,deux ou trois entrées.Pour la bascule à une entrée ou bascule à commande symétrique(a) l'impulsion x commande le changement d'état quelque soit l'état initial.Pour la bascule à deux entrées(b),l'impulsion x_1 provoque le passage dans l'état 1 si la bascule se trouvait initialement dans l'état 0 et est sans effet si elle se trouvait dans l'état 1.L'impulsion x_0 commande de même l'état 0.Enfin ces deu deux modes de commande sont combinés dans la bascule à trois entrées(c).



L'association de plusieurs bascules de même type forme un REGISTRE. Celui-ci est capable de mémoriser un mot.Suivant la nature du couplage entre ces bascules et suivant leurs entrées et sorties,on aura des registres compteurs, des régistres de décalage ou des régistres arithmétiques:Additionneur-Soustrac--teur,accumulateurs.

2.Classification des circuits logiques intégrés:

Deux grandes classes peuvent être dégagées :

- les circuits fonctionnant en saturation
- et les circuits fonctionnant hors saturation.

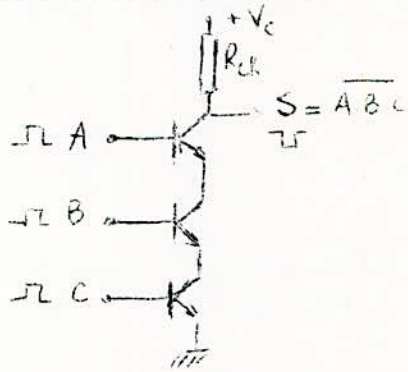
Pour les circuits fonctionnant hors saturation,la commutation des élé--ments se fait entres deux points situés dans la zone d'amplification ou entre le point de blocage et un point situé dans la zone d'amplification.Les circuits réalisés sont très rapides du fait de la réduction du temps de blocage et de la non saturation des transistors.

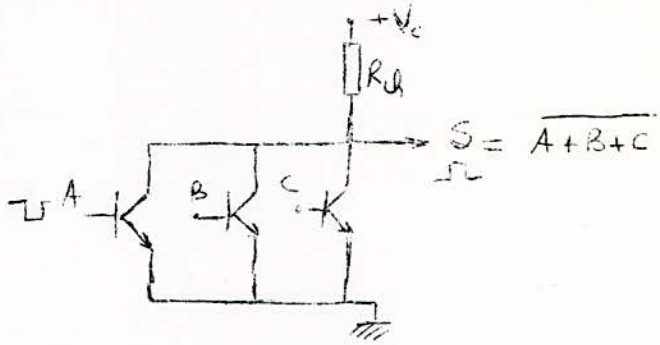
Parmi les circuits fonctionnant en saturation,on peut distinguer :

- les circuits où la fonction logique est assurée par des résistances,
- ceux où elle est assurée par des transistors,
- et ceux où elle est assurée par des diodes.

2.1.Circuits DCTL(Direct Coupled Transistor Logic):

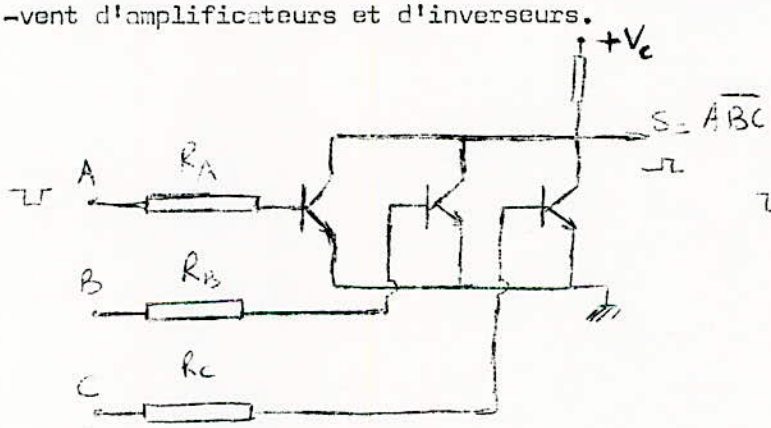
Ces circuits comportent uniquement des transistors et des résistances de charge.Suivant le couplage des transistors,on réalise soit la fonction NOR soit la fonction NAND.

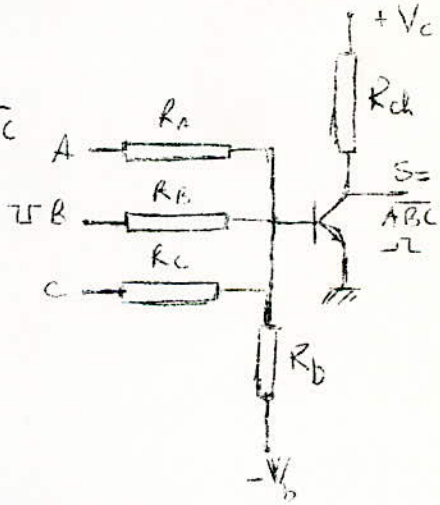




2.2.Circuits RTL(Resistor Transistor Logic):

L'entrée se fait à travers des résistances de base.Les transistors ser--vent d'amplificateurs et d'inverseurs.

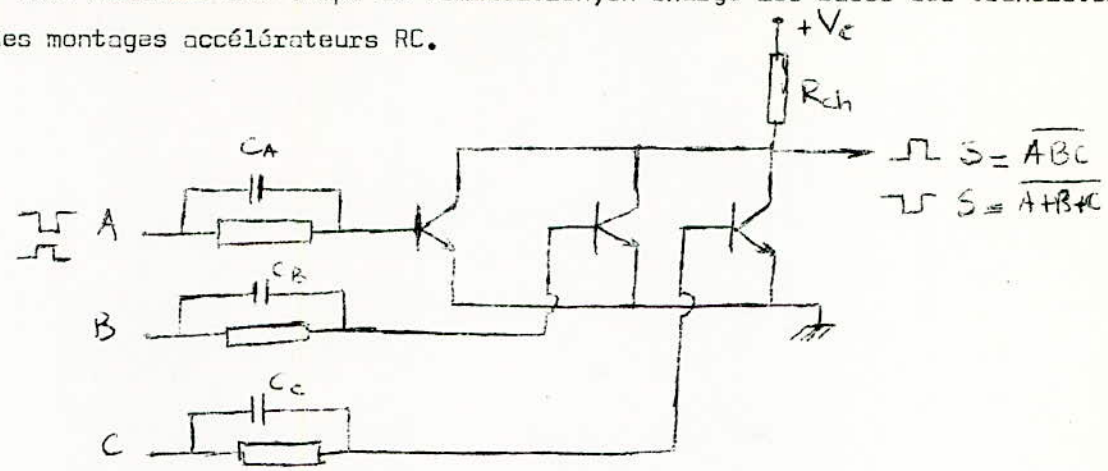




Les circuits RTL sont moins sensibles aux variations des courants de base des transistors que les circuits DCTL.Les grandes amplitudes possibles des impultions de commutation donne une grande immunité aux bruits.L'inconvé--nient des circuits RTL,c'est que les résistances de base augmentent les temps de commutation.

2.3.Circuits RCTL(resistor Capacitor Transistor Logic):

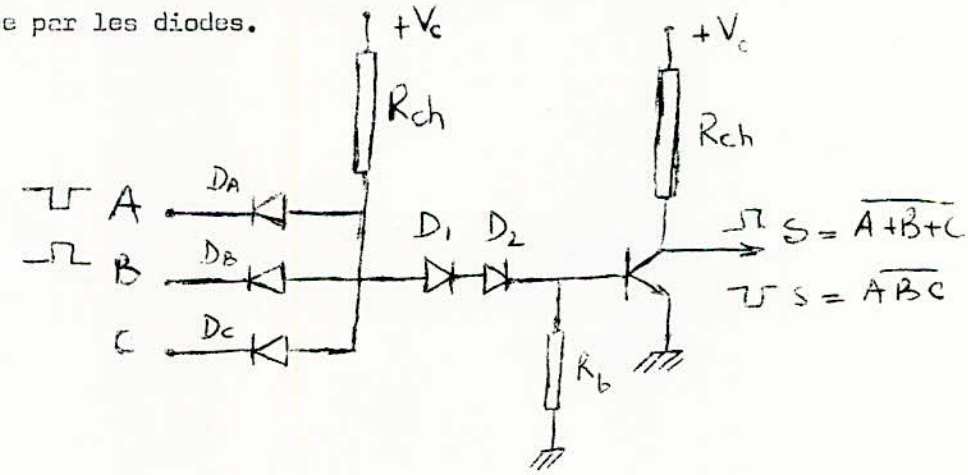
Pour diminuer les temps de commutation,on charge les bases des transistors par des montages accélérateurs RC.



Inconvenient:en hautes fréquences,les capacités représentant des court--circuits,laissent passer les composantes HF du bruit,ce qui rend les circuits RCTL sensibles aux bruits HF.

2.4.Circuits DTL(Diode Transistor Logic):

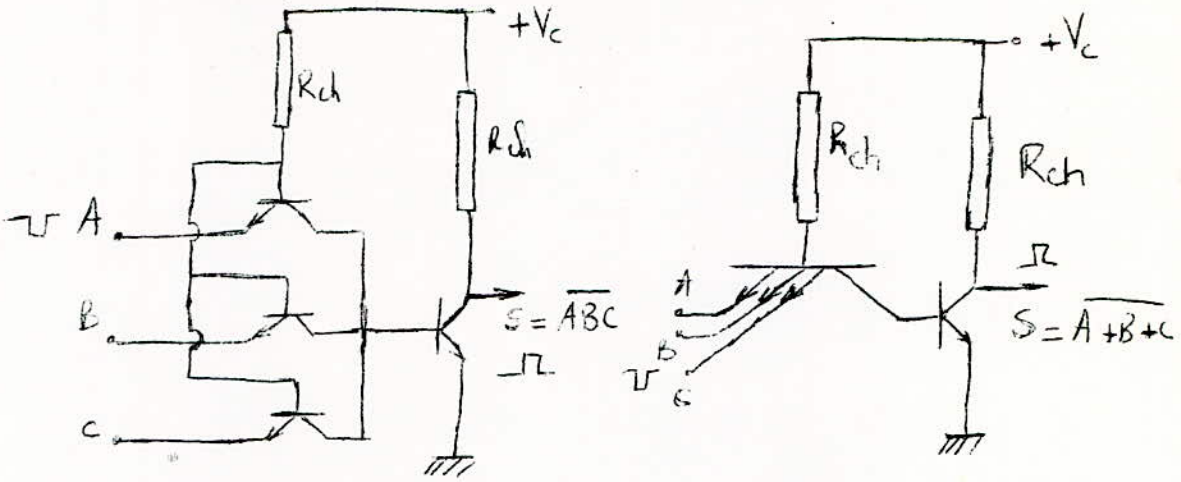
Ils comportent des diodes et des transistors.La fonction logique est assurée par les diodes.



Les diodes D_A,$D_B$et D_C réalisent la fonction ET en logique positive et la fonction OU en logique négative.Le transistor réalise la fonction de complé--mentation.Les circuits DTL présente une excellente immunité au bruit.

2.5.Circuits TTL(Transistor Transistor Logic):

ce sont des circuits dérivés du DTL.Les diodes d'entrée D_A,$D_B$et D_C sont remplacées par les diodes de jonctions émettrices des transistors et les diodes de transposition D_1 et D_2 par la jonction collectrice.



Le circuit obtenu est très rapide du fait que le transistor à émetteur multiple est pratiquement toujours saturé;par conséquent,il n'introduit aucun retard de commutation.

Avantages:

- la construction des circuits logiques est simplifiée
- l'absence de la résistance de base R_b permet d'augmenter la vitesse de commutation.

Inconvénients:

-l'amplification d'un transistor à émetteur multiple donne la possibilité de naissance de l'effet " current robbing ",ce qui limite la valeur de l'en--trance.

- l'immunité au bruit est faible par rapport aux autres circuits.

Remarque:

Pour la réalisation de l'unité de calcul,on utilisera des circuits TTL.

2ème PARTIE

CIRCUITS DE CALCULS

CHAPITRE 1 . ADDITIONNEUR-SOUSTRACTEUR BINAIRE

Dans toute machine numérique,les quatre opérations arithmétiques fonda--mentales que sont l'addition,la soustraction,la multiplication et la division peuvent être réalisées à partire d'une seule:l'addition.En effet,la soustraction peut être ramenée à l'addition du complément(vrai ou restreint);la multiplica--tion n'est qu'une suite d'addition.Quant à la division,ce n'est qu'une répéti--tion de soustractions donc d'additions.Il est plus simple,lorsqu'on parle de circuits arithmétiques,de raisonner sur les additionneurs.

Comme notre étude concerne seulement les machines parallèles,nous allons détailler l'additionneur parallèle.

Dans un additionneur parallèle,tous les bits des deux nombres entrent simultanément et tous les bits de la somme sont formés simultanément.Ainsi,le nombre de cellules est égal au nombre de bits des deux nombres à additionner. Chaque cellule délivre un bit somme et un bit retenue qui se propage en cascade. Ce mode de fonctionnement est très rapide mais très coûteux au point de vue circuits.Pour des nombres de quatre bits,le schéma synoptique de l'additionneur parallèle est donné par la figure 1

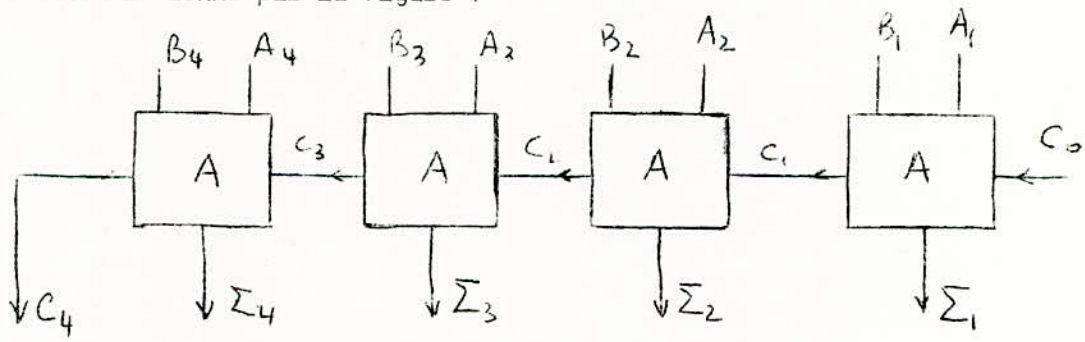


Fig 1.Additionneur parallèle 4 bits.

Cette juxtaposition de quatre étages(cellules) suppose que les signaux de retenue issus d'un étage sont présentés à l'étage suivant.Donc,la rapidité de l'additionneur est limitée par le temps de propagation de la retenue.Nous examinerons le problème de la retenue au paragraphe 2.

1.Additionneur binaire à retenue en cascade:

1.1.Etude d'une cellule:

On suppose qu'on désire effectuer la somme de deux bits A_i et B_i d'ordre i et d'une retenue éventuelle C_{i-1} d'ordre immédiatement inférieur(i-1).

A_i	B_i	C_{i-1}	$\sum_i$	C_i
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

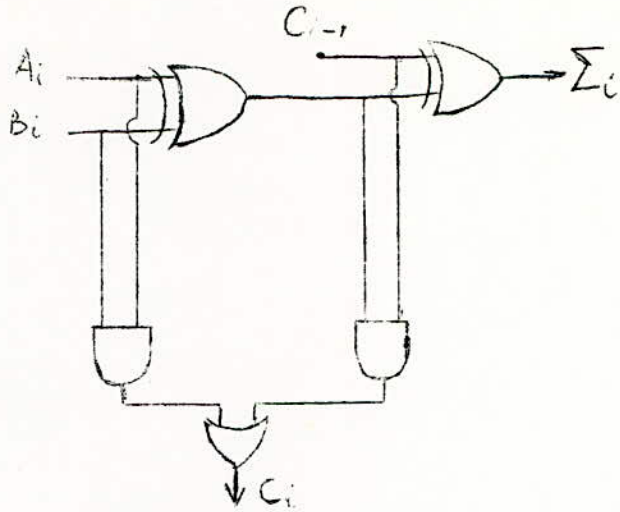


Fig 2.Tableau de vérité et schéma logique d'une cellule.

Le tableau de vérité nous donne :

$$\sum_i = \overline{A}_i B_i \overline{C}_{i-1} + A_i \overline{B}_i \overline{C}_{i-1} + \overline{A}_i \overline{B}_i C_{i-1} + A_i B_i C_{i-1}$$

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

$$\boxed{C_i = \overline{A}_i B_i + C_{i-1}(\overline{A_i \oplus B_i}) = A_i B_i + A_i C_{i-1} + B_i C_{i-1}}$$

$$\overline{C}_i = \overline{A}_i \overline{B}_i \overline{C}_{i-1} + \overline{A}_i \overline{B}_i C_{i-1} + \overline{A}_i B_i \overline{C}_{i-1} + A_i \overline{B}_i \overline{C}_{i-1}$$

Multiplions les deux membres par ($A_i + B_i + C_{i-1}$) :

$$\overline{C}_i (A_i + B_i + C_{i-1}) = \overline{A}_i \overline{B}_i C_{i-1} + \overline{A}_i B_i \overline{C}_{i-1} + A_i \overline{B}_i \overline{C}_{i-1}$$

Le bit somme peut s'écrire alors :

$$\sum_i = A_i \overline{C}_i + B_i \overline{C}_i + C_{i-1} \overline{C}_i + A_i B_i C_{i-1}$$

Le schéma logique d'une cellule délivrant le bit somme $\sum_i$ et le bit retenue C_i est donné par la figure 3.

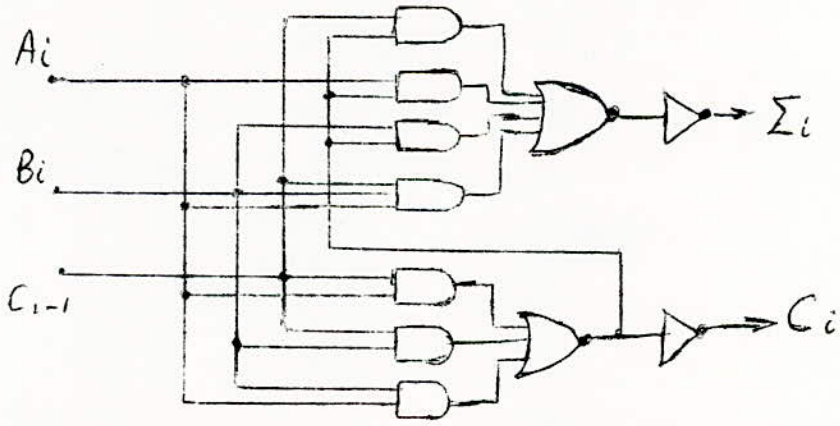


Fig 3.Schéma logique d'une cellule d'addition.

ADDITIONNEUR BINAIRE 4 BITS

SN 7483 N

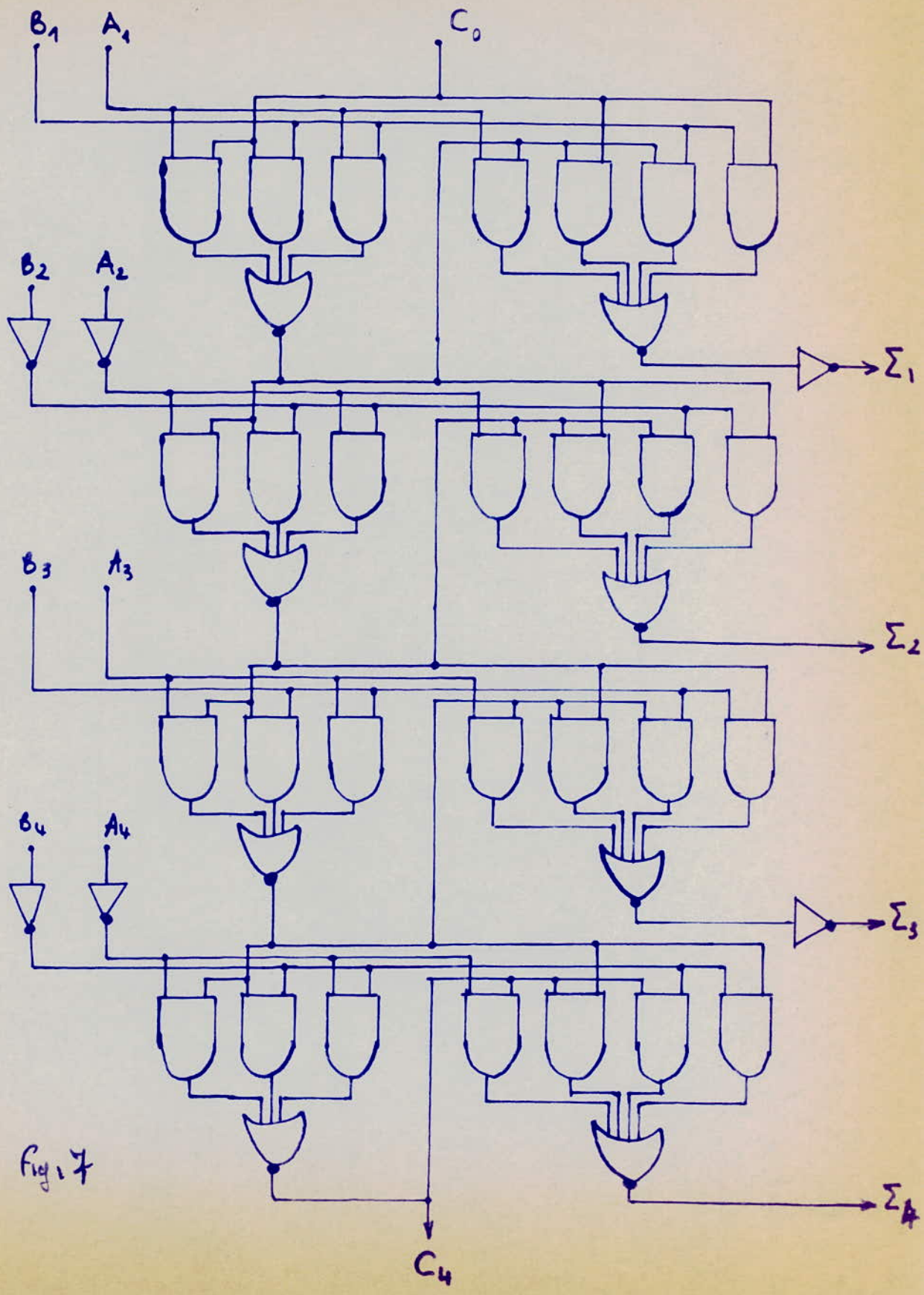


Fig. 7

1.2. Etude de la cellule immédiatement supérieure:

Normalement, la cellule immédiatement supérieure est identique à la première. On profite du fait que celle-ci peut fournir le bit retenue complémen--té. Si on complémente les deux bits A_{i+1} et B_{i+1} on aura :

$$C_{i+1} = \overline{A}_{i+1}\overline{B}_{i+1} + \overline{A}_{i+1}\overline{C}_i = + \overline{B}_{i+1}\overline{C}_i$$

$$C_{i+1} = \overline{A_{i+1}B_{i+1} + A_{i+1}C_i + B_{i+1}C_i}$$

Le bit somme s'écrit :

$$\Sigma_{i+1} = \overline{A_{i+1}B_{i+1}C_i + A_{i+1}\overline{C}_{i+1} + B_{i+1}\overline{C}_{i+1} + C_i\overline{C}_{i+1}}$$

On n'aura pas à complémenter ni le bit retenue, ni le bit somme à la sortie des opérateurs NOR. Le schéma logique de la seconde cellule est donné par la figure 4.

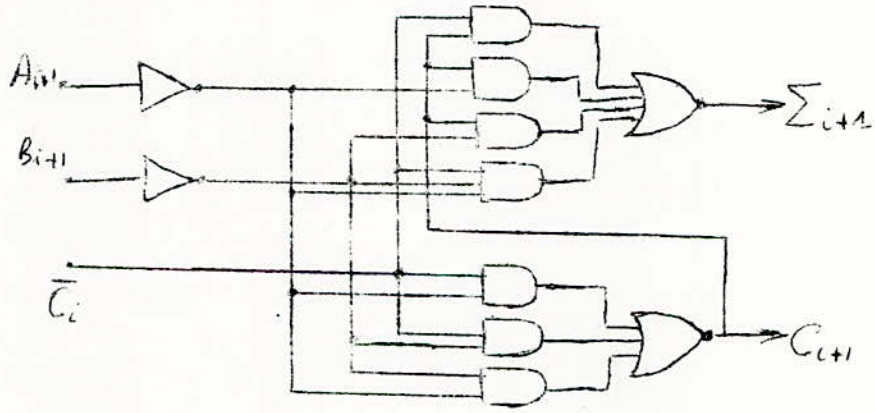


Fig 4. Schéma logique de la seconde cellule.

L'association de ces deux cellules nous donne un additionneur à deux bits. Il est représenté par la figure 5.

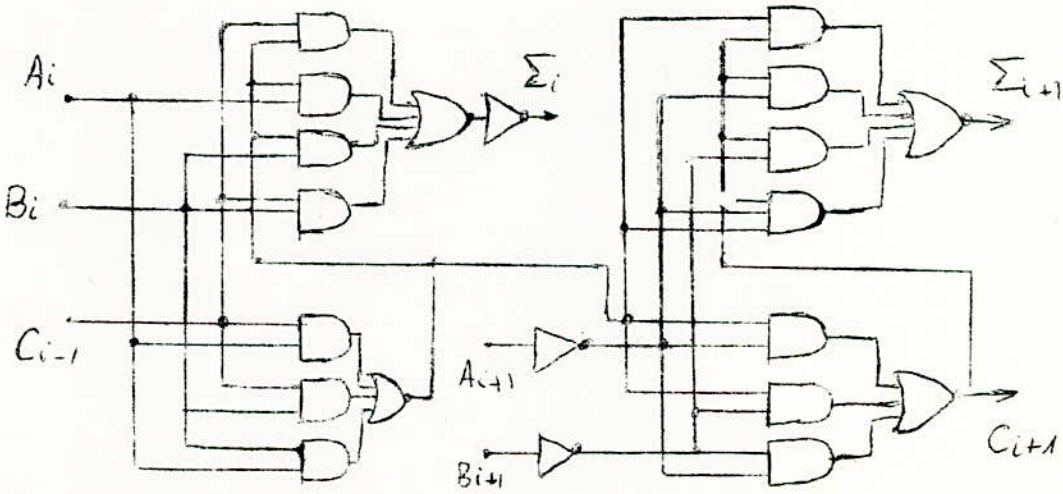


Fig 5. Additionneur parallèle 2 bits.

1.3.Additionneur complet 4 bits :

Le schéma synoptique de l'additionneur parallèle 4 bits est donné par la figure 6.

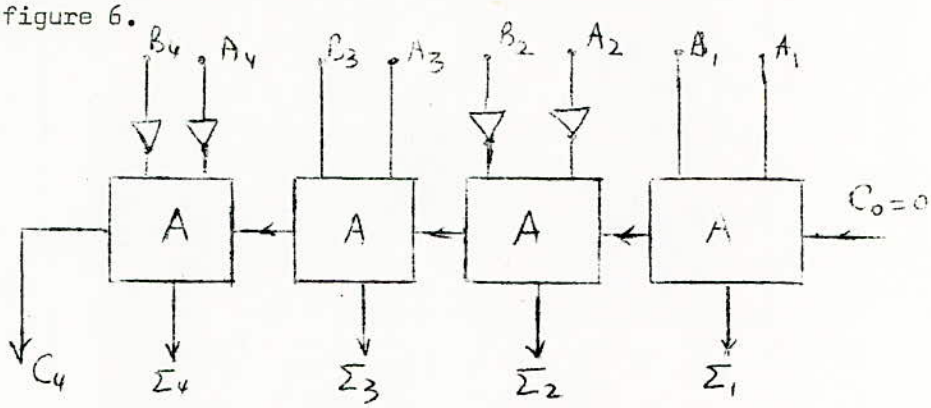


Fig 6.Additionneur parallèle 4 bits.

On aboutit ainsi au circuit intégré: SN 7483 N capable d'effectuer la somme de deux nombres à 4 bits.Le circuit est représenté par la figure 7.La figure 8 donne le brochage du SN 7483 N .

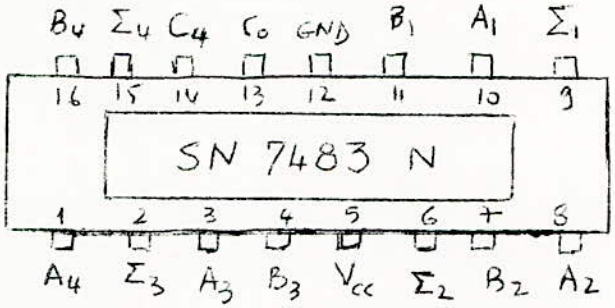


Fig 8.Brochage du circuit intégré SN 7483 N .

2.Problème de la retenue: retenue anticipée:

On a vu qu'une cellule d'ordre i ne peut effectuer la somme de deux bits A_i et B_i d'ordre i que si la retenue délivrée par la cellule d'ordre(i-1)soit présente.et que le temps d'attente soit égal au temps de propagation de cette retenue.Ainsi le temps d'addition est fonction du nombre de cellules placées en cascade et la retenue de l'addition se trouve réduite.Pour améliorer la vitesse de l'addition,on utilise la méthode de la retenue anticipée.

Le tableau de vérité de l'addition binaire donne :(voir fig 2)

$$C_i = A_iB_i + A_iC_{i-1} + B_iC_{i-1}$$

$$\text{ou} \quad C_i = A_iB_i + (A_i + B_i)\,C_{i-1}$$

Il y a une retenue générée par la cellule i si :

1^{er}cas: les A_i et B_i valent 1 quelque soit la valeur de la retenue C_{i-1}

2^{ème}cas:A_i (ou B_i) et C_{i-1} valent 1 .

Le 1er cas traduit le fait qu'au niveau de la cellule d'addition,il y a une élaboration d'une retenue quand A_i et B_i ont la valeur 1;on dit qu'il y a génération d'une retenue propre à la cellule.Ainsi A_iB_i est appelé terme de génération.

Le 2^{ème}cas fait intervenir la notion de propagation de la retenue à travers la cellule et fait intervenir la somme $P_i = A_i + B_i$,appelé terme de propagation.La retenue C_i peut s'écrire alors :

$$\boxed{C_i = G_i + P_iC_{i-1}}$$

On suppose qu'on a un additionneur à 4 cellules avec les termes de gé--nération et de propagation suivants :

1^{ère}cellule :	G_0	P_0
2^{ème}cellule :	G_1	P_1
3^{ème}cellule :	G_2	P_2
4^{ème}cellule :	G_3	P_3

Pour les différentes cellules,on peut écrire :

1^{ère}cellule: $C_1 = G_0 + P_0C_0$

2^{ème}cellule: $C_2 = G_1 + P_1C_1 = G_1 + P_1(G_0+P_0C_0)$

3ème cellule:$C_3 = G_3 + P_2C_2 = G_2 + P_2(G_1+P_1(G_0+P_0C_0)$

4^{ème}cellule: $C_4 = G_3 + P_3C_3=$

$$\boxed{C_4 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_0}$$

Cette expréssion montre qu'il est possible de remplacer un étage d'ad--dition à 4 bits à retenue en cascade(retenue qui doit se propager à travers 4 niveaux logiques) par un circuit qui ne requiert qu'un seul niveau logique. Ce circuit est représenté par la figure 9.

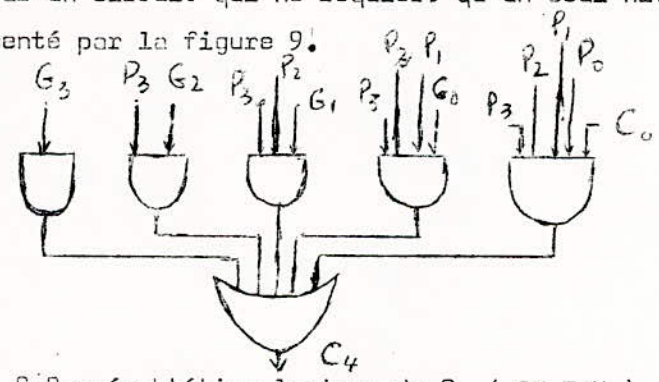


Fig 9.Représentation logique de C_4 (SN 74H).

A l'aide de ce circuit de retenue anticipée,on a amélioré de 4 fois la vitesse d'addition,au prix d'une légère complexité du circuit.

Additionneur 4 bits SN 7483 N à retenue anticipée:

Cet additionneur élabore une retenue C_4 soit à partir d'une retenue propre à la cellule et apparaissant en sortie du SN 7483 N, soit à partir de la retenue présente à l'entrée (et se propageant à travers l'étage) grâce à la sortie P du circuit intégré SN 7454 N puisque :

$$P = \overline{\overline{A_1}\overline{B_1} + \overline{A_2}\overline{B_2} + \overline{A_3}\overline{B_3} + \overline{A_4}\overline{B_4}} = (A_1+B_1)(A_2+B_2)(A_3+B_3)(A_4+B_4)$$

P traduit bien un terme de propagation $\boxed{C_4 = G + C_0 P}$

Donc, la retenue C_0 au lieu de se propager à travers 4 cellules du l'additionneur SN 7483 N, transite par un seul niveau lequel a été choisi dans la série T.T.L rapide (SN 74H52 N) pour réduire le temps de propagation à 10 ns au maximum pour 4 bits.

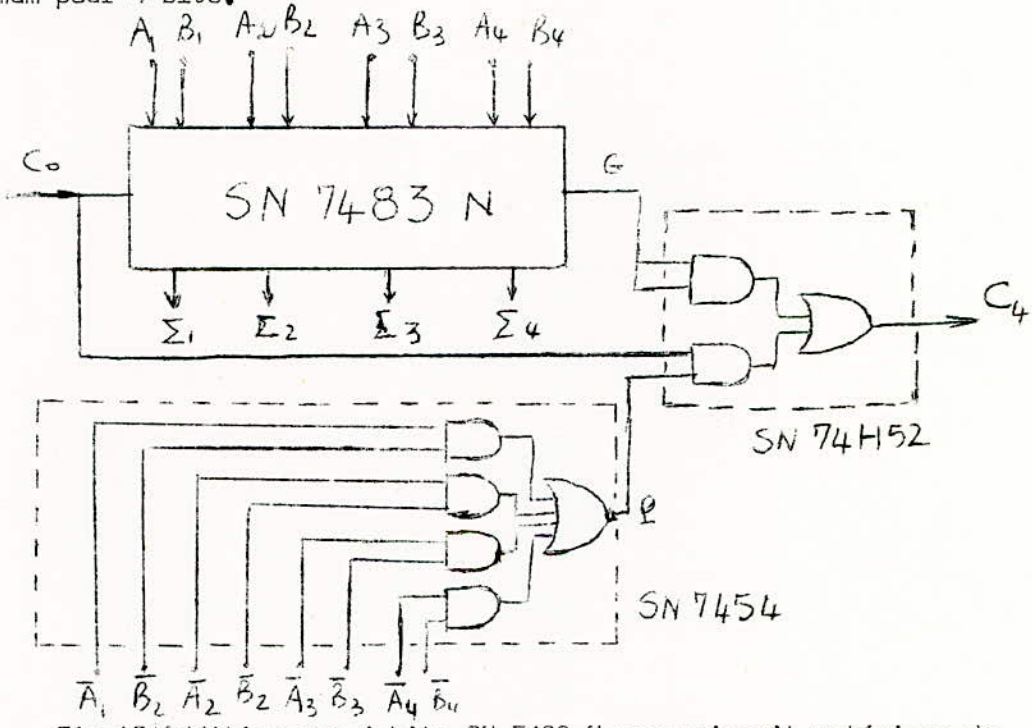


Fig 10. Additionneur 4 bits SN 7483 N avec circuit extérieur de retenue anticipée.

3. Soustracteur binaire:

Par soustracteur, on entend un dispositif effectuant la différence de 2 bits A_i et B_i d'ordre i et d'une retenue C_{i-1} d'ordre (i-1) immédiatement inférieur. Le tableau de vérité de la figure 11 donne:

$$\Sigma_i = \overline{A}_i B_i \overline{C}_{i-1} + A_i \overline{B}_i \overline{C}_{i-1} + \overline{A}_i \overline{B}_i C_{i-1} + A_i B_i C_{i-1}$$

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

A_i	B_i	C_{i-1}	$\sum_i$	C_i
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

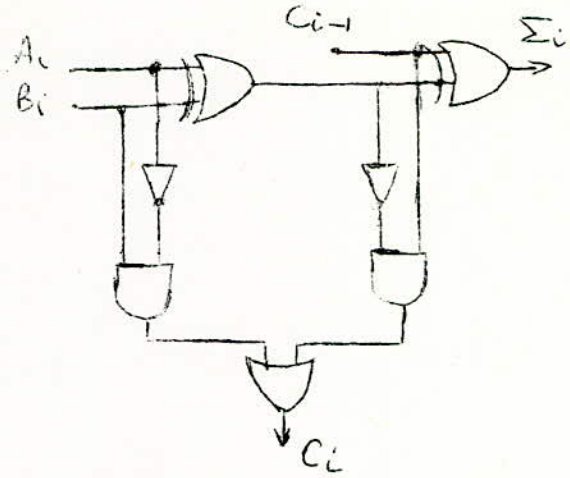


Fig 11.Tableau de vérite et schéma logique d'un soustracteur.

Le bit retenue s'écrit :

$$C_i = \overline{A}_i B_i + C_{i-1} \overline{(A_i \oplus B_i)} = \overline{A}_i B_i + (A_i + B_i)\, C_{i-1}$$

D'après les équations logiques,on remarque que la soustraction sans re--tenue $\sum_i$ des 2 bits A_i et B_i est identique à la somme sans repport de ces mêmes bits.La seule différence réside dans l'expression logique de la retenue C_i.De ce fait,on peut grouper les deux circuits logiques en un seul qui fait, suivant une commande d'opération,soit une addition soit une soustraction.Le schéma de l'additionneur-soustracteur est donné par la figure 12.

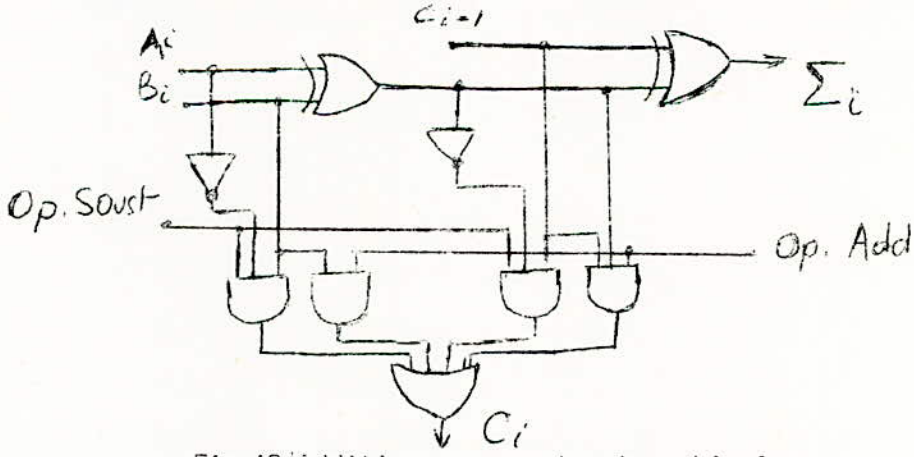


Fig 12.Additionneur-soustracteur binaire.

Etant donné que la soustraction peut être ramenée à une addition,on va essayer d'élaborer un circuit logique de complémentation.Pour cela,on impose deux entrées de contrôle B et C telles que :

$$B = 0 \begin{cases} C = 0 & \text{le bit est complémenté à 1} \\ C = 1 & \text{le bit n'est pas complémenté} \end{cases}$$

$$B = 1 \begin{cases} C = 0 & \text{le bit est égal à 1} \\ C = 1 & \text{Le bit est égal à 0} \end{cases}$$

Tableau de vérité:

B	C	X	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

La fonction de complémentation s'écrit:

$$Y = \overline{B}.\overline{X}.\overline{C} + \overline{B}.C.X + B.\overline{C}.\overline{X} + B.\overline{C}.X$$

$$Y = \overline{C}.\overline{X} + \overline{B}.C.X + B.\overline{C}.X$$

$$Y = \overline{C}(\overline{X} + X.B) + \overline{B}.C.X$$

d'où :

$$Y = C(\overline{B}.X) + \overline{C}(\overline{\overline{B}.X})$$

$$\boxed{Y = C \oplus \overline{\overline{B}.X}}$$

On en déduit le schéma logique de la fonction Y :

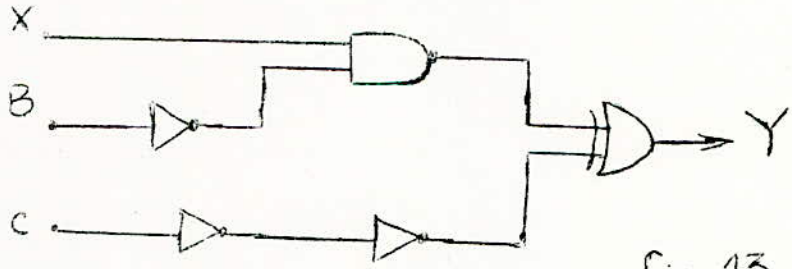


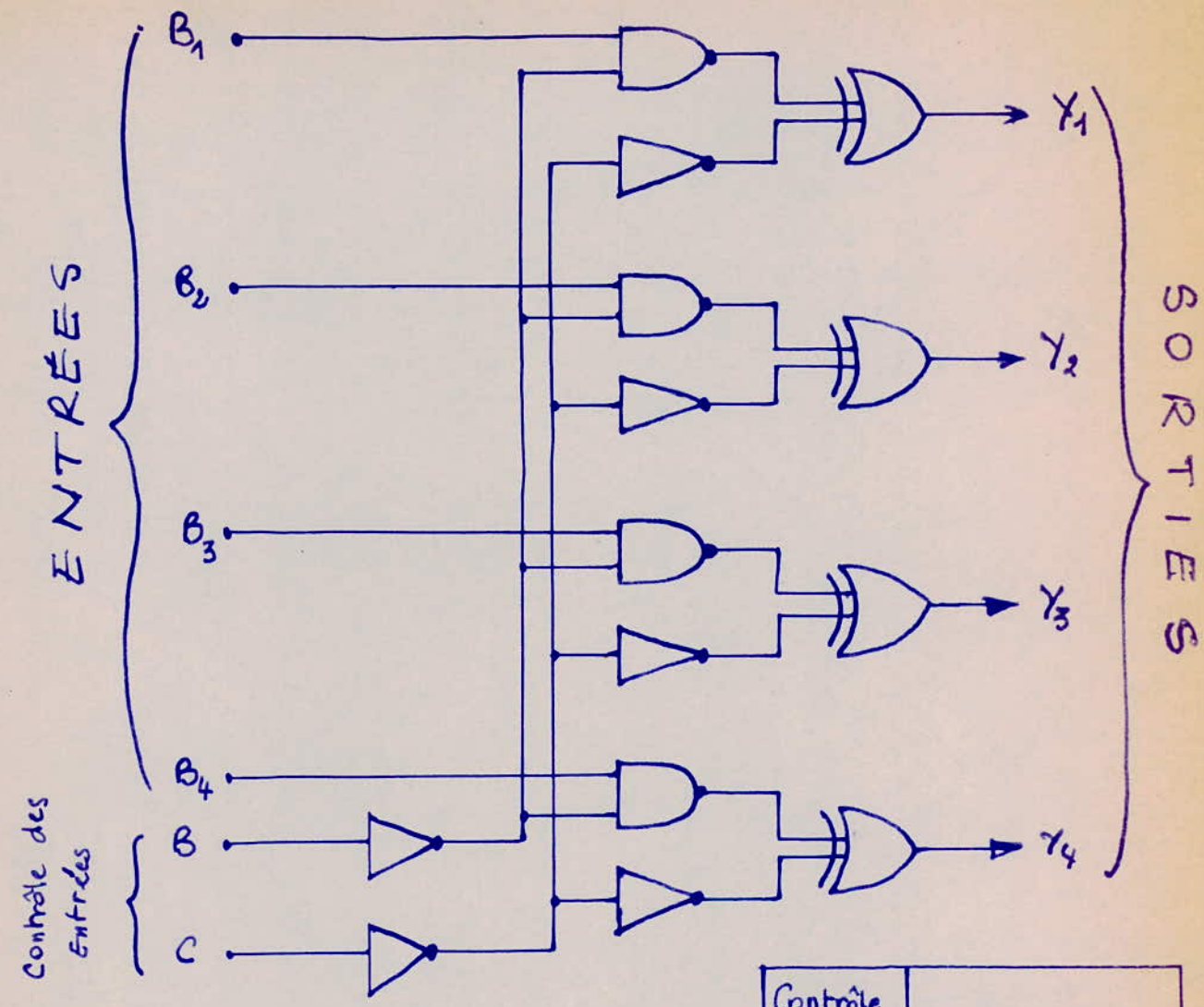
Fig. 13.

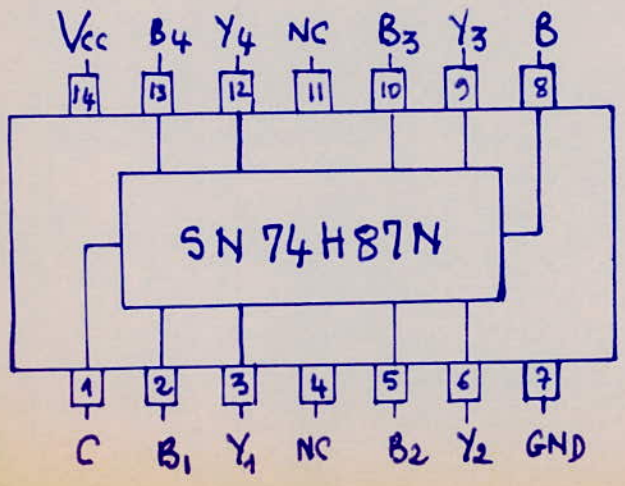
Pour des nombres de 4 bits,on aura 4 circuits logiques identiques au précédent.On aboutit ainsi au générateur de complément:le SN 74H87 N.Le circuit complet est représenté par la figure 13.

Le circuit intégré SN 74H87 N donne le complément à 1 d'un nombre binaire à 4 bits.Pour avoir l'addition en complément à 2,il suffit d'ajouter un 1 au résultat;ce qui revient à avoir une retenue initiale égale à 1.Le circuit logique complet de l'additionneur-soustracteur parallèle à 4 bits est représenté par la figure 14 .

GENERATEUR DE COMPLEMENT

SN 74H87 N





Contrôle des Entrées		SORTIES			
B	C	Y_1	Y_2	Y_3	Y_4
0	0	$\overline{B_1}$	$\overline{B_2}$	$\overline{B_3}$	$\overline{B_4}$
1	0	1	1	1	1
0	1	B_1	B_2	B_3	B_4
1	1	0	0	0	0

fig: 13

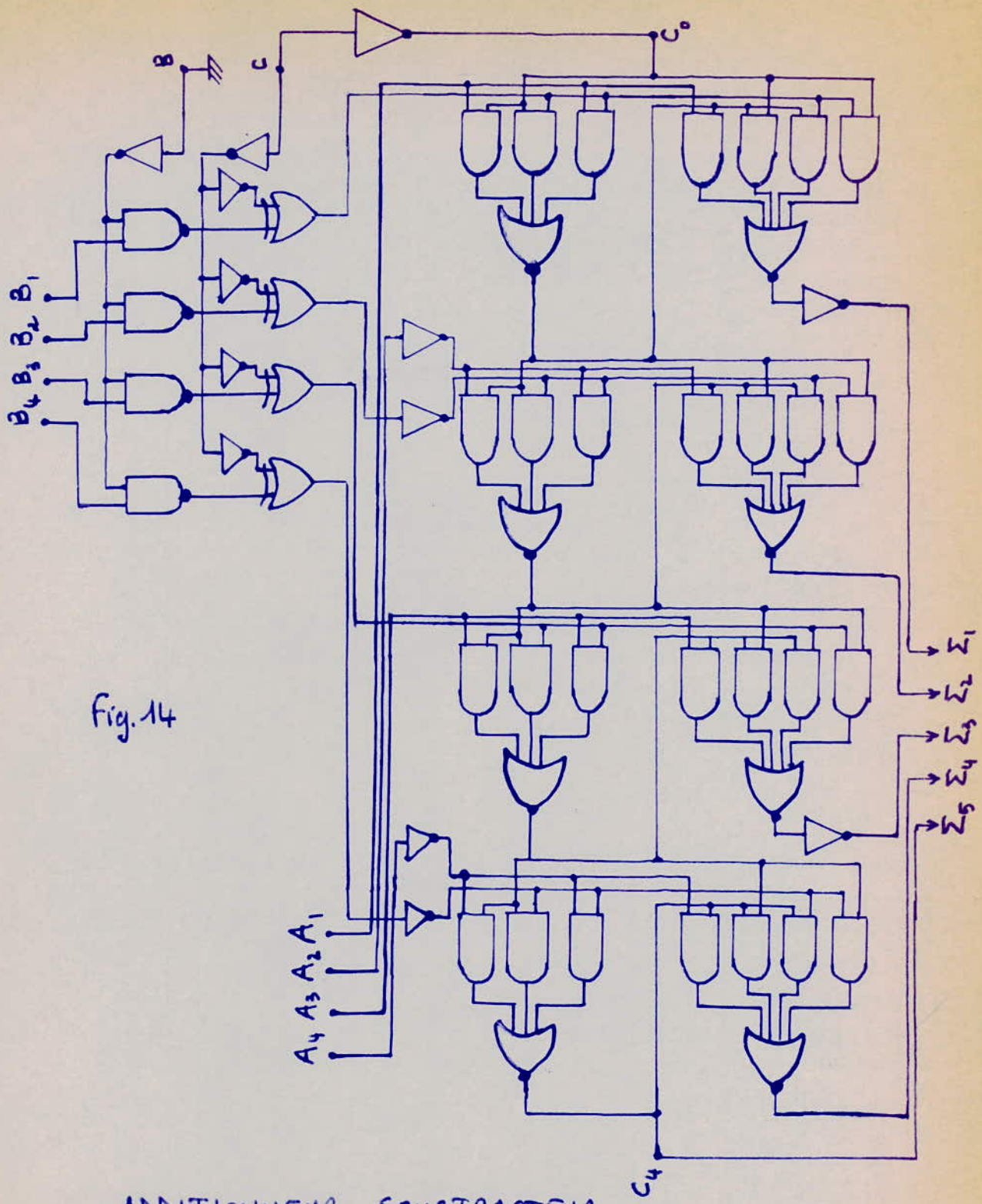


Fig. 14

ADDITIONNEUR - SOUSTRACTEUR
(4 BITS)
(SN7483N + SN74H87N)

CHAPITRE 2 . MULTIPLIEUR BINAIRE

La présence ou l'absence d'un code opératoire pour la multiplication de nombres entiers marque une frontière entre les puissances des unités arithmétiques des ordinateurs. A partir des opérations d'additions, de décalage logique et de branchement conditionnel, on peut évidemment programmer la multiplication mais cela oblige à appeler plusieurs fois dans l'unité arithmétique une quantité inscrite en mémoire, ce qui allonge beaucoup le temps d'exécution.

La multiplication "cablée" peut se faire de plusieurs manières; par additions répétées, par additions et décalages de façon dite simultanée. Ce dernier mode de multiplication vise avant tout la rapidité. Le produit se forme en très peu de temps d'horloge au prix évidemment d'une complication des circuits.

1. Multiplication par additions répétées:

En général, un additionneur complet est organisé pour recevoir trois éléments binaires (2 opérandes et la retenue issue de l'étage précédent) et pour fournir deux éléments binares (la somme et la retenue). La propagation de cette re retenue alloge considérablement l'addition. Pour éviter ce retard, on peut faire l'addition en ne prenant les retenues en considération qu'au cycle suivant de l'addition. On aura donc un régistre supplémentaire destiné à recevoir le mot constitué par l'ensemble des retenues à transfér^er d'une addition à la suivante. Mais le contenu du régistre supplémentaire ne doit être pris en considération qu'après avoir été décalé d'un rang vers les poids les plus forts.

Exp: soit à multiplier $(11)_{10} = (1011)_2$ par $(13)_{10} = (1101)_2$

1011	
1101	
1011	x 1
0000	x 0
01011.	
0000	RR
1011	Rés
1011	x 1
100111	
001000	RR
001000	Décl.Gh
0110111	Rés
1011	x 1
1101111	
0010000	RR
0010000	Décl.Gh
01001111	Rés

01001111	Rés
00100000	RR
00100000	Décl.Gh
000001111	Rés
001000000	RR
001000000	Décl.Gh
0010001111	Rés

$$(10001111)_2 = 1.2^7 + 1.2^3 + 1.2^2 + 1.2^1 + 1.2^0$$

$$(10001111)_2 = 128 + 8 + 4 + 2 + 1 = (143)_2$$

2.Multiplication par addition et décalage:

La multiplication de deux nombres de n bits donne un résultat de 62n bits. Il faut donc prévoir deux régistres pour contenir le résultat.L'unité de calcul utilise trois régistres pour la multiplication:

- Régistre R qui contient le multiplicande
- Régistre Q qui contient le multiplicateur
- Régistre accumulateur qui élabore chaque somme partielle.

Le schéma de principe est donné par la figure 15.

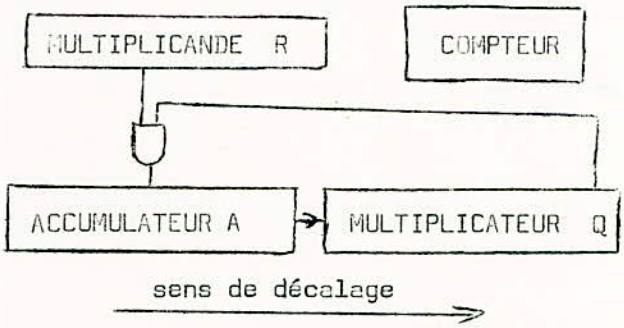


Fig.15

Le résultat de la multiplication apparaît dans l'ensemble des deux ré--gistres A et Q qu'on désigne par AQ.

Le fonctionnemnt se fait en deux temps :

- Selon que le bit du plus faible poids du multiplicateur est 1 ou 0 , le multiplicande ou 0 est transmis et additionné au contenu de l'accumulateur. Pendant le transfert,A n'est pas effacé ou si cela est,réinscrit automatiquement.

- Le contenu de AQ est décalé d'un rang vers la droite.Le bit du plus faible poids du multiplicateur est perdu et le bit du plus faible poids du ré--sultat partiel s'écrit dans la position du plus fort poids (qui vient d'être libéré par le décalage à droite) du régistre A.

L'opération se continue avec tous les bits du multiplicateur.Simultané--ment à ces opérations d'additions et de décalages,un compteur progresse d'une unité après chaque décalage.L'opération se termine après n décalages,n étant le nombre de bits du multiplicateur.

Exp: soit à multiplier :

$$A = (13)_{10} = (1101)_2$$

par $$B = (11)_{10} = (1011)_2$$

Disposition initiale

Registre A

1	1	0	1

Accumulateur

0	0	0	0	0

Registre B

1	0	1	1

Sens du décalage →

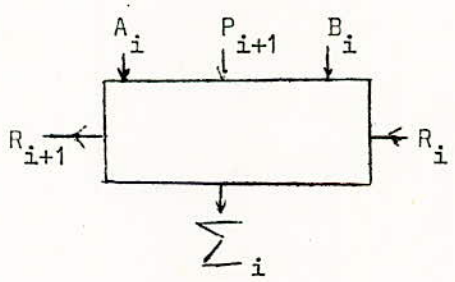
1er pas	1.Add	0 1 1 0 $\boxed{1}$	1 0 1 1	× 1
	2.Dcl	0 0 1 1 0	$\boxed{1}$ 1 0 1	
2ème pas	1.ADD	1 0 0 1 1	$\boxed{1}$ 1 0 1	× 1
	2.Dcl	0 1 0 0 1	1 $\boxed{1}$ 1 0	
3ème pas	1.Add	0 1 0 0 1	1 $\boxed{1}$ 1 0	× 0
	2.Dcl	0 0 1 0 0	1 1 $\boxed{1}$ 1	
4ème pas	1.Add	1 0 0 0 1	1 1 1 1	× 1
	2.Dcl	0 1 0 0 0	1 1 1 $\boxed{1}$	

$$A \times B = (10001111)_2 = (143)_{10}$$

3.Multtiplieur cellulaire:

Etant donné que dans l'algorithme de la multiplication par addition et décalage tous les pas sont identiques(Add-Dcl),on effectue sur une ligne le produit $(A.B_i)$ et on ajoute à ce produit le résultat partiel précédemment dé--calé.Chaque cellule d'une ligne possède donc 4 entrées:A_i,B_i,R_i et le bit re--tenue partiel décalé P_{i+1}; et 2 sorties:le bit somme et le bit retenue R_{i+1}.

La cellule se présente comme suit:



$$\sum_i = B_i(A_i + P_{i+1} + R_i) + \overline{B}_i P_{i+1}$$

$$\boxed{\sum_i = (B_i(A_i \oplus R_i)) + P_{i+1}}$$

$$\boxed{R_{i+1} = B_i(A_i P_{i+1} + A_i R_i + R_i P_{i+1})}$$

Fig.16

Fig 17 : Réseau cellulaire du Multiplieur 4 bits

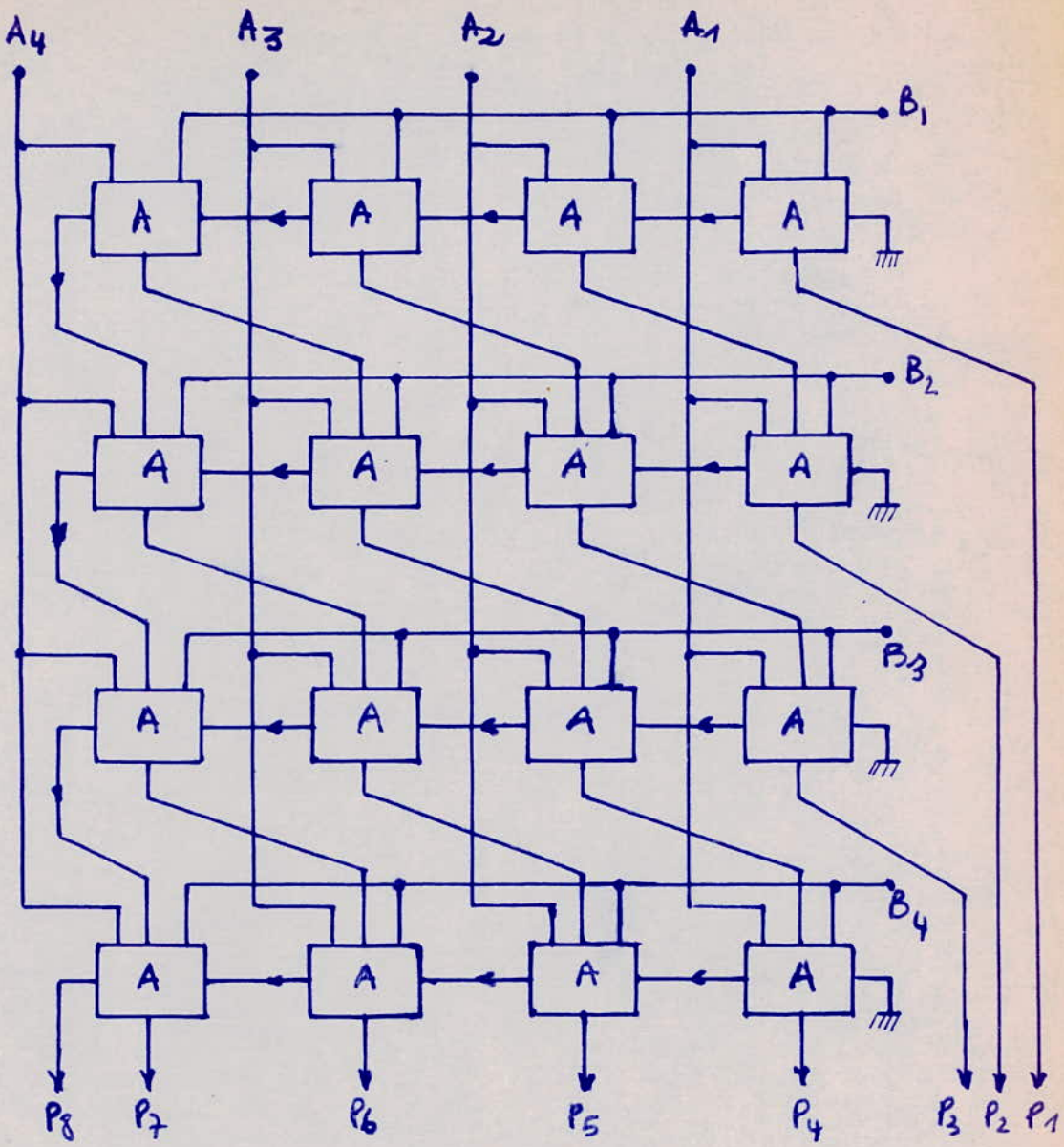
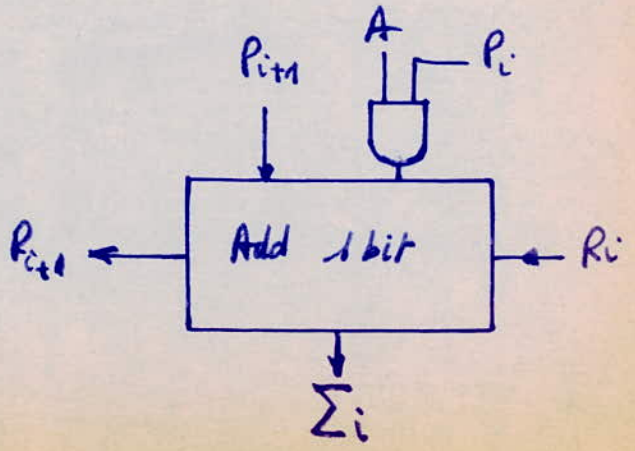


Fig 18 :
Schéma logique d'une Cellule A.



MULTIPLIEUR CELLULAIRE

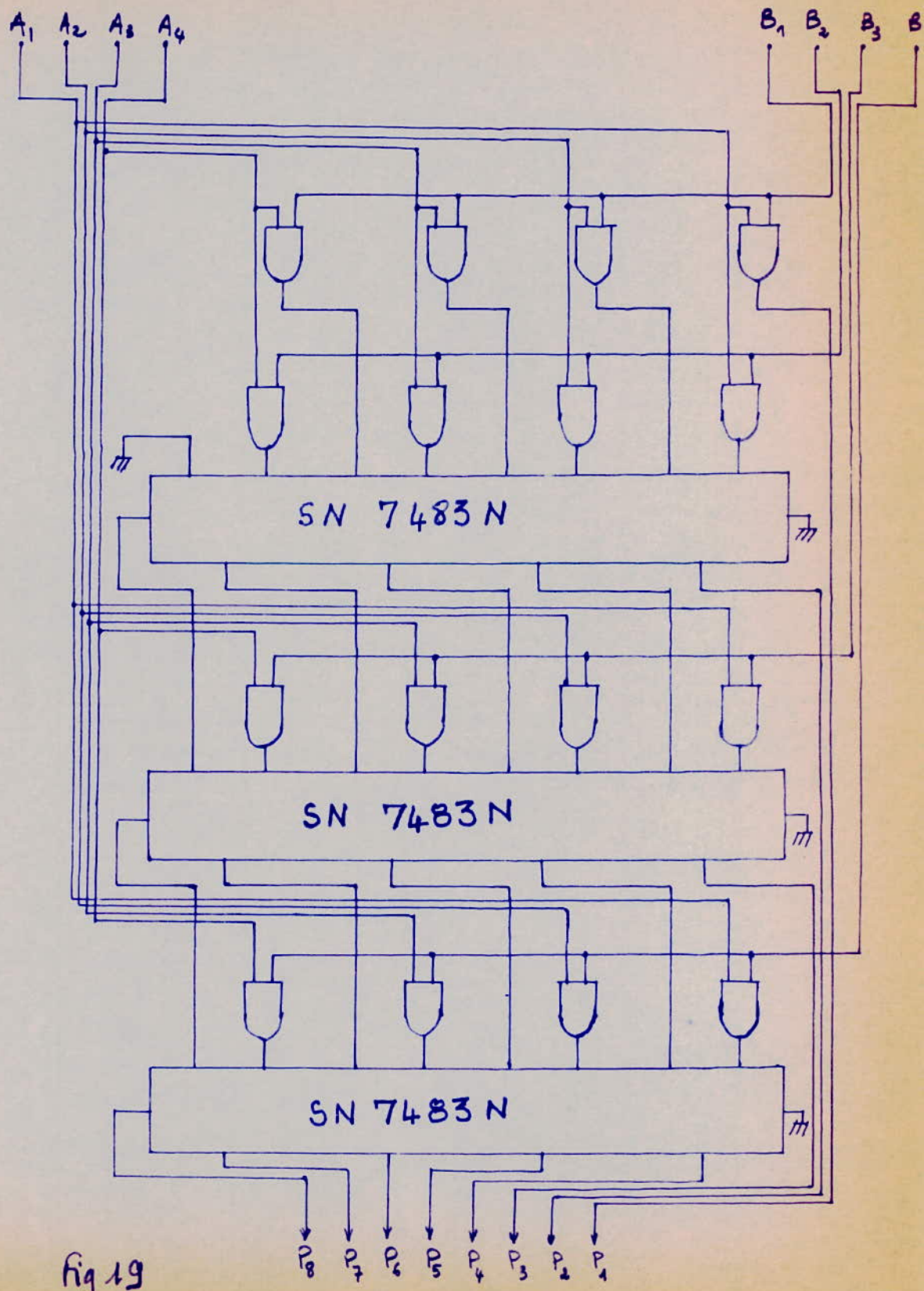


fig 19

La commande B_i peut se matérialiser par une porte ET.Etant donné que pour les cellules de la première ligne,les produits partiels et les retenues sont nuls,elles se réduisent simplement à des portes ET.

Or,si le bit B_i du multiplicateur est égal à 1,la cellule fonctionne en étage d'additionneur d'un poids précisé.Si B_i=0,la cellule transmet le bit du même poids du résultat partiel P_{i+1} d'ordre inférieur décalé.Le réseau cellulaire est donné par la figure 17.

La cellule peut se présenter comme suit:

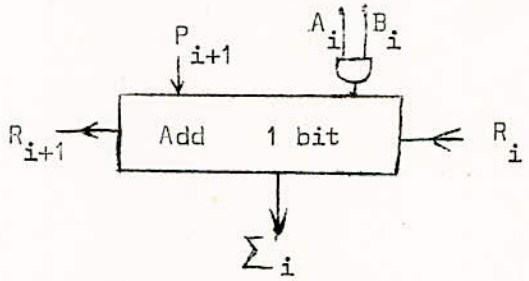


Fig.18

On peut grouper les 4 cellules d'addition 1 bit pour former l'addition--neur complet (4 bits):le SN 7483 N .Le schéma global du multiplieur cellulaire est donné par la figure 1 .

On remarque aussi qu'on peut grouper les 4 portes ET en un seul circuit intégré:le SN7408 N .

4.Autres méthodes:

De nombreuses méthodes sont mises en œuvre pour diminuer le temps d'exécution de la multiplication.

4.1.Méthode de BOOTH :

Elle est applicable à des nombres représentés en valeur et signe.Les nombres négatifs sont représentés par leur complément à 2 et la multi--plication s'effectue en considérant le bit signe comme un bit ordinaire. La machine examine successivement les poids du multiplicateur pris 2 à 2 en commençant par les plus faibles poids.Suivant les 4 possibilités: 00,01,10 et 11,elle effectue les opérations suivantes:

00 → Décalage du produit d'un rang à droite

01 → Add du multiplicande puis Décl d'un rang à droite du produit

10 → Sous " " " " "

11 → Décl du produit partiel d'un rang à droite.

4.2.Multiplication multidigitale :

Avec cette méthode,on peut faire la multiplication simultanée de plusieurs bits du multiplicateur par le multiplicande.Suivant que les deux derniers bits du multiplicateur sont : 00,01,01 et 11,la machine effectuera les opérations suivantes:

00 → Pas d'addition du multiplicande

01 → Addition du multiplicande,Décalage à droite de deux rangs

10 → Addition de 2 fois le multiplicande ,décalage de deux rangs à droite.

11 → Addition de 3 fois le multiplicande,décalage de deux rangs à droite.

Si l'on dispose au préalable dans deux régistres des valeurs doubles ou triples du multiplicande ,on diminue le nombre d'additions et on augmente la vitesse de la multiplication.

CHAPITRE 3 .DIVISEUR BINAIRE

Comme pour la multiplication,la division de deux nombres de n bits donne pour résultat un mot de 2n bits:le Quotient et le Reste.Il existe deux méthodes.

1.Méthode par soustractions successives:

C'est l'opération inverse de la multiplication par addition répétées. Pour des nombres entiers,elle consiste à retrancher le diviseur du dividende jusqu'à celui-ci devient négatif ou nul.Le comptage du nombre d'opérations fournit le quotient.Le reste est ce qu'est devenu le dividende.

Cette méthode est simple mais très lente.

2.Méthode par Add/Sous et décalage:

L'unité de calcul comporte trois régistres comme pour la multiplication:

- Régistre Diviseur D
- Régistre Quotient Q
- Régistre totaliseur qui contient le dividende T.

Le schéma synoptique du diviseur est donné par la figure 20.

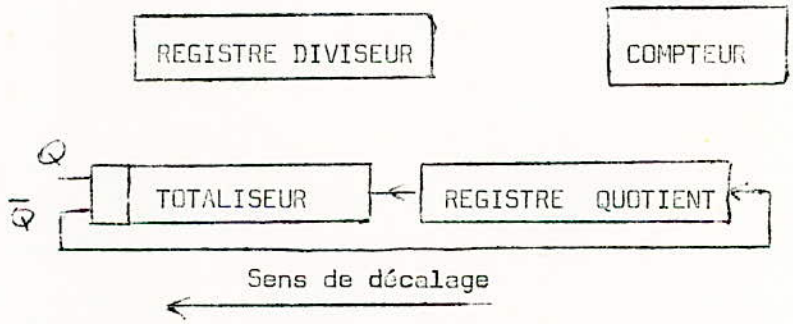


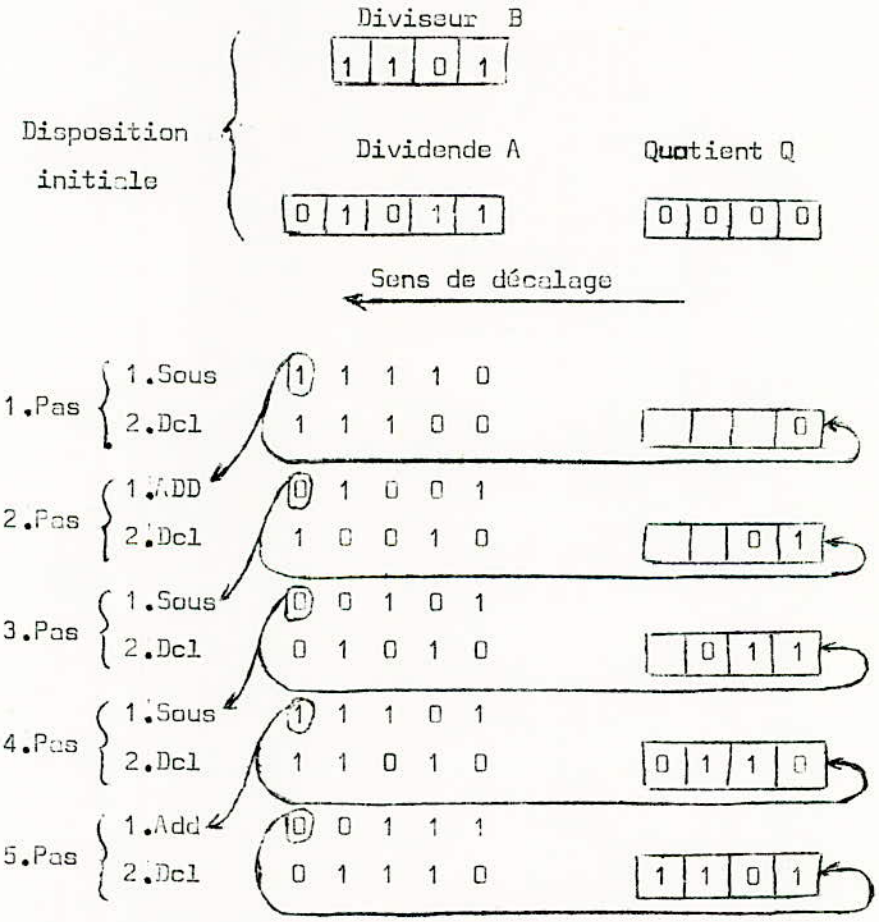
fig 20

On suppose que le diviseur et le dividende sont inférieurs à l'unité et que le diviseur est supérieur au dividende.

Lorsque la soustraction du diviseur au dividende est impossible,le bit quotient est égal à 0.Il faut donc diviser le diviseur par la base(2) et essayer à nouveau de soustraire.Or,la multiplication par 2 revient à un décalage vers la gauche.On décale alors l'ensemble Dividende-Quotient d'une position vers la gauche.Ce décalage libère la case la plus à droite du régistre quotient;ce qui permettra d'y inscrire le résultat de la soustraction suivante.La division s'arrête en comptant le nombres de bits établis 0 ou 1 du quotient,ce qui re--vient à compter les décalages opérés.Plutôt que d'effectuer un décalage vers

la gauche,ce décalage sera circulaire.Le bit de la position la plus à gauche du régistre T s'inscrit dans la position la plus à droite du régistre Q.De plus, lorsque la soustraction est impossible,le bit résultat est égal à 1 (résultat négatif) alors que le bit quotient doit être égal à 0.Donc,on doit complémen--ter le bit de la position la plus à gauche du régistre T.Or,un élément du ré--gistre (ou Bascule) possède une sortie directe et une sortie complémentée. Il suffit de relier la sortie complémentée à la position la plus à droite du régistre quotient.

Exp : soit à diviser $A = (0,1011)_2 = (0,6875)_{10}$ par $B=(0,1101)_2 = (0,8125)_{10}$



soit $A/B = (0,1101)_2 = (0,8125)_{10}$ par défaut .

On remarque que le 1 commande l'addition et le 0 commande la soustraction.

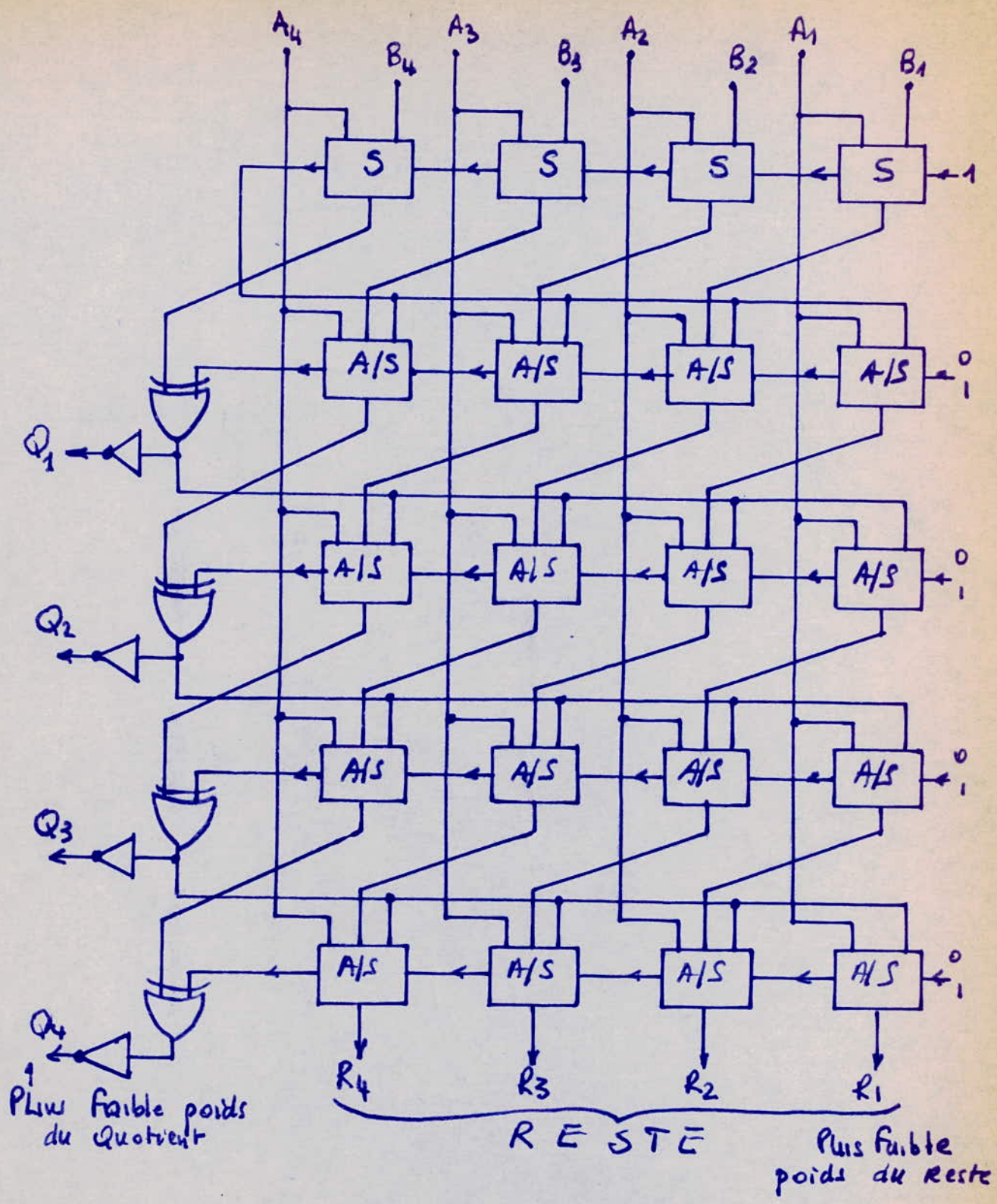


Fig 22. Réseau Cellulaire du Diviseur binaire 4bits

3 .Diviseur cellulaire:

Etant donné que dans l'algorithme de la division par Add/Sous et décala--ge,tous les pas sont identiques,le réseau sera constitué de plusieurs cellules chacune d'elles pourra effectuer soit une addition soit une soustraction sui--vant que le reste partiel est négatif(1) ou positif(0).Le réeultat décalé vers les plus forts poids.

Or, on sait que le nbit de la soustraction est identique au bit de l'ad--dition.De plus,le bit Q de la position la plus à gauche du régistre T comman--de soit une addition(Q=1) soit une soustraction(Q=0).

Une cellule peut se présenter comme suit :

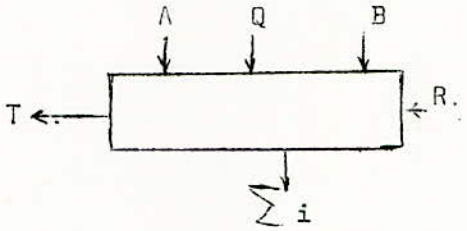


Fig.21

Pour une addition,la retenue d'ordre supérieur R_{i+1} est égal à :

$$R_{i+1} = A_i B_i + A_i R_i + R_i B_i$$

Pour la soustraction R_{i+1} s'écrit :

$$R_{i+1} = \overline{A}_i B_i + \overline{A}_i R_i + R_i B_i$$

La cellule délivre donc une retenue T telle que :

$$\boxed{T = \underbrace{Q(A.B + A.R + B.R)}_{\text{addition}} + \underbrace{\overline{Q}(\overline{A}.B + \overline{A}.R + B.R)}_{\text{soustraction}}}$$

ou

On remarque que les cellules de la première ligne n'effectuent qu'une soustraction.Le réseau cellulaire est représenté par la figure 22.

Pour satisfaire à ces conditions,on utilise le bloc Additionneur-Sous--tracteur dont l'entrée C est reliée à la sortie Q.Le schéma détaillé de la cellule est donné par la figure 23.

Le divideur cellulaire complet pour des nombresde 4 bits est représenté par la figure 24.

DIVISEUR CELLULAIRE (4 bits)

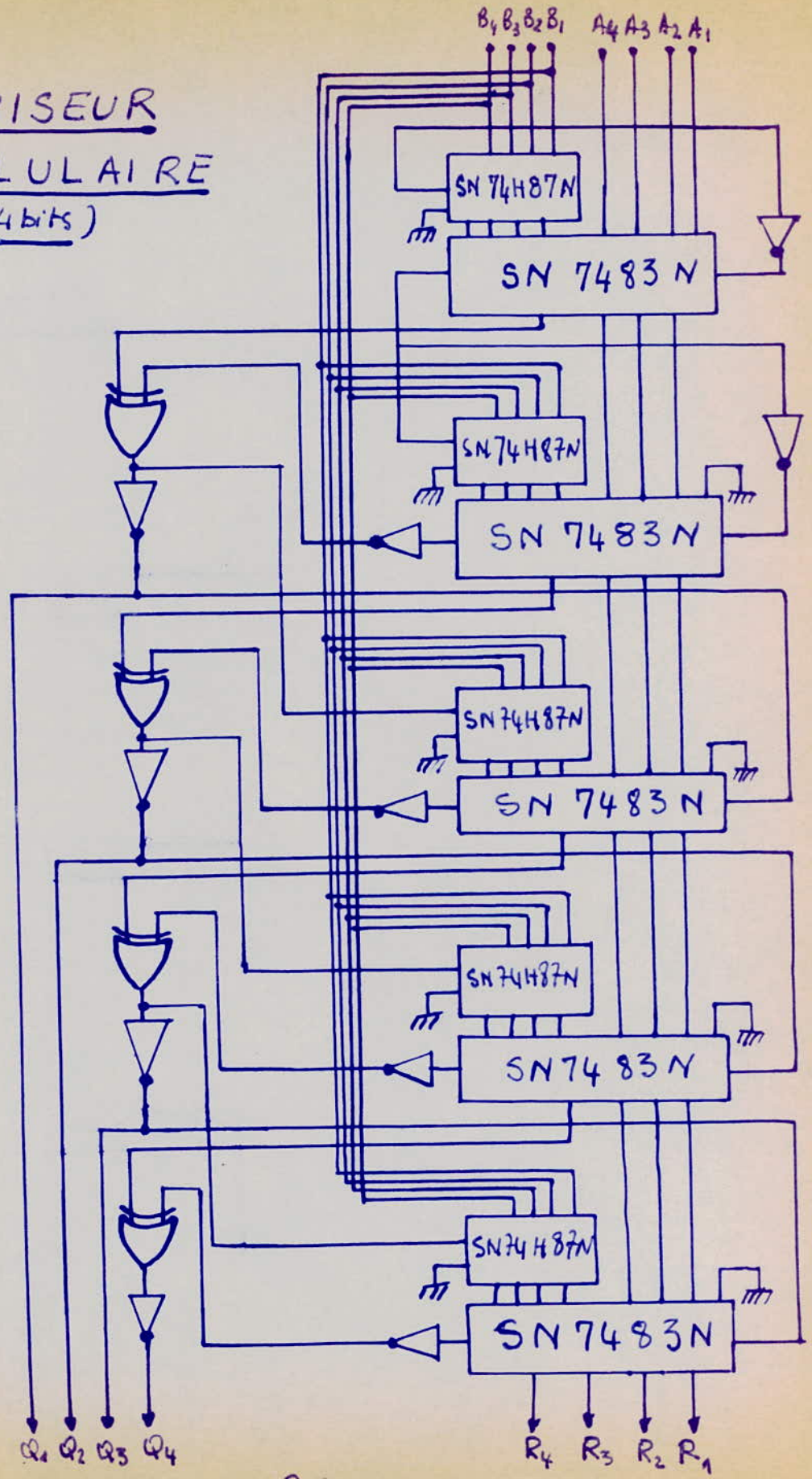


Fig 24

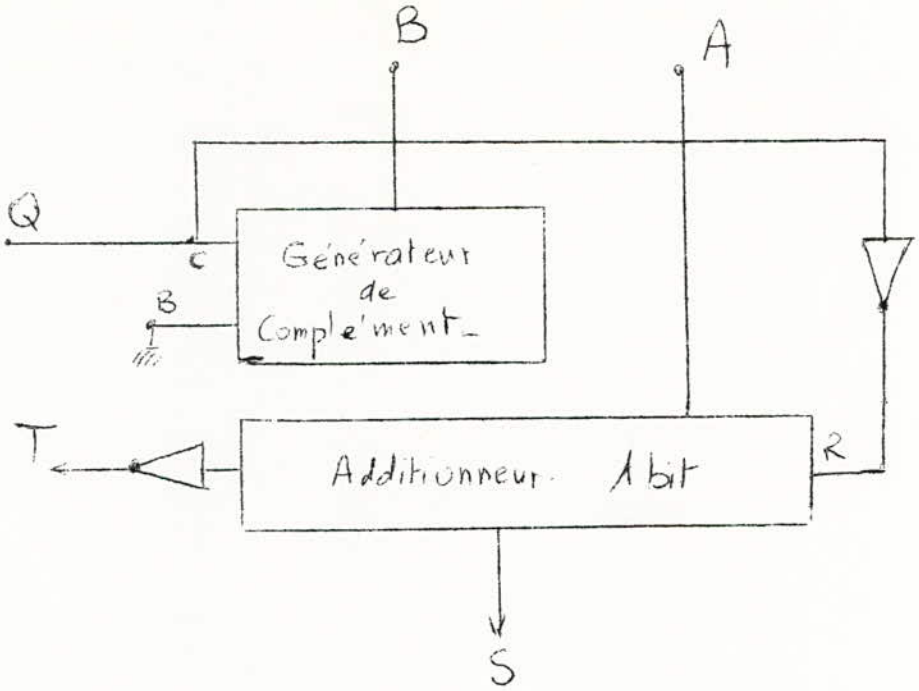


Fig 23

CHAPITRE 4. COMPARATEUR BINAIRE

Par comparateur, on entend un dispositif susceptible de comparer deux nombres binaires A et B entre eux. Il doit être capable de déterminer si A est inférieur, égal ou supérieur à B. On distingue deux types de comparateurs: le comparateur parallèle et le comparateur série. Dans le premier cas, il s'agit de comparer deux nombres qui restent figés dans deux régistres; dans le second cas, l'information contenue dans les deux régistres est décalée et les deux nombres sont comparés poids par poids.

Etant donné que notre calculateur numérique est du type parallèle, on étudiera simplement les comparateurs parallèles.

Les comparateurs parallèles sont conçus suivant deux méthodes:

1.1 ère méthode: utilisation des circuits de coïncidence:

1.1. Circuit de coïcidence:

Ce circuit réalise la fonction inverse du circuit OU EXCLUSIF, c'est-à-dire qu'il délivre un 1 en sortie lorsque les deux entrées sont identiques.

Tableau de vérité:

A	B	$A \oplus B$	$A \odot B$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

$$A \odot B = \overline{A \oplus B} = A.B + \overline{A}.\overline{B} =$$
$$= (A + \overline{B})(\overline{A} + B)$$

Fig 25. Tableau de vérité du circuit de coïcidence.

1.2. Comparateur parallèle :

On concidère deux nombres binaires $A_n A_{n-1} \ldots A_2 A_1$ et $B_n B_{n-1} \ldots B_2 B_1$. On réalise la coïcidence de tous les bits:

$$X_1 = A_1 \odot B_1 \quad , \quad X_2 = A_2 \odot B_2 \quad , \quad \ldots , \quad X_n = A_n \odot B_n$$

La fonction logique de la sortie du comparateur s'écrit:

$$Z = X_n . X_{n-1} \ldots X_1 = (A_n \quad B_n)(A_{n-1} \quad B_{n-1}) \ldots (A_1 \quad B_1)$$

D'où :

$$Z = 1 \quad \text{si} \quad A = B$$
$$Z = 0 \quad \text{si} \quad A \neq B$$

COMPARATEUR PARALLELE

Utilsant les circuits de coïcidence.

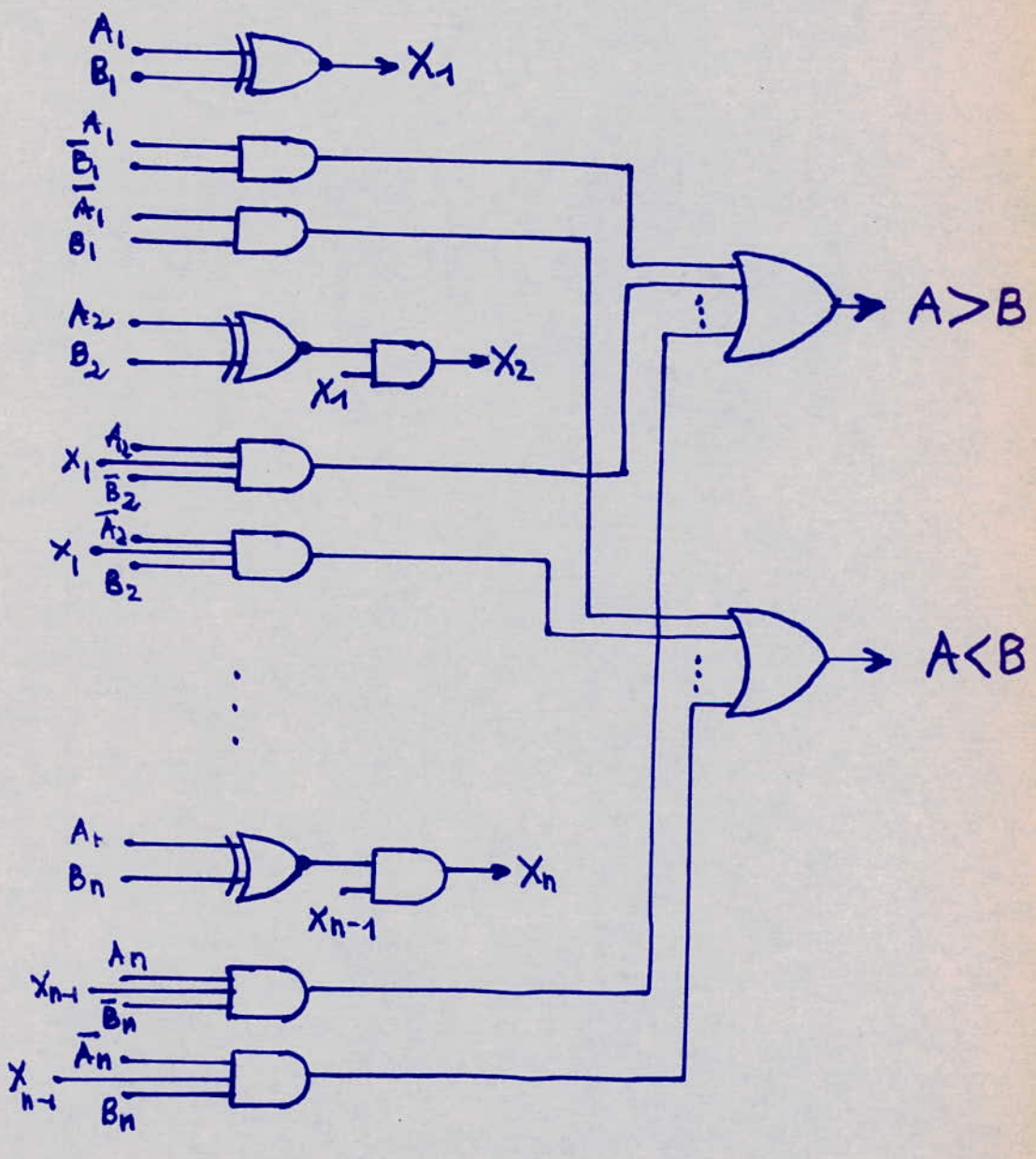


fig 27.

Pour établir le schéma logique du comparateur, on suppose que la sortie Z répond aux conditions suivantes;

$$Z = 0 \text{ si } A < B, \quad Z = 1 \text{ si } A = B \text{ et } Z = \phi \text{ si } A > B$$

Tableau de vérité pour n = 2 :

A_2	A_1	B_2	B_1	Z
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	ϕ
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	ϕ
1	0	0	1	ϕ
1	0	1	0	1
1	0	1	1	0
1	1	0	0	ϕ
1	1	0	1	ϕ
1	1	1	0	ϕ
1	1	1	1	1

Tableau de Karnaugh:

B_2B_1 \ A_2A_1	00	01	11	10
00	1	ϕ	ϕ	ϕ
01	0	1	ϕ	ϕ
11	0	0	1	0
10	0	0	ϕ	1

$$Z = \overline{B}_1\overline{B}_2 + A_1A_2 + A_1\overline{B}_2 + A_2\overline{B}_1$$

$$Z = (A_1 + \overline{B}_1)(A_2 + \overline{B}_2)$$

Fig 26.

Il est possible de généreriser pour des nombres d'ordre n :

$$Z = (A_1 + \overline{B}_1)(A_2 + \overline{B}_2)\ldots(A_n + \overline{B}_n)$$

Avec cette fonction, on vérifie simplement que le nombre A est inférieur au nombre B. Dans le cas où l'on ne fait aucune hypothèse sur A et B, on compare alors successivement tous les poids des deux nombres en commençant par les poids les plus élevés à l'aide d'un circuit de coïcidence. Si les nombres sont identiques, tous les poids sont successivement examinés. S'ils sont différents, il existe une position où deux bits ne coïcident pas. Il n'est pas alors nécessaire de continuer la comparaison des plus faibles poids, mais de s'assurer lequel des deux bits est égal à 1. Si $B_i = 1$ et $A_i = 0$, $A > B$; si $A_i = 1$ et $B_i = 0$ alors $A < B$. Le schéma du comparateur est donné par la figure 27.
Pour des nombres binaires très grands, le circuit devient très complexe. Surtout on sera limité par l'entrance des portes.

COMPARATEUR PARALLELE PAR SOUSTRACTION DE B

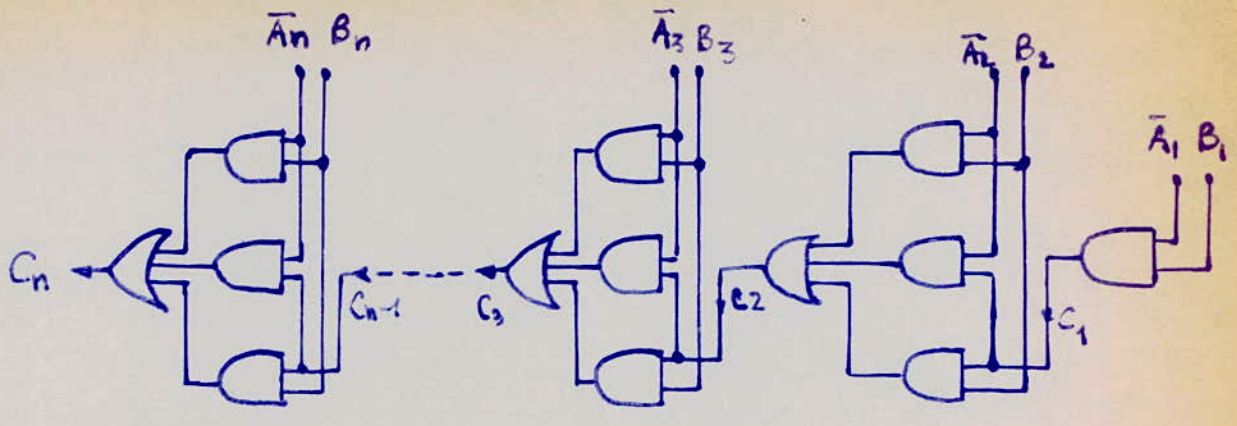


Fig: 28

si $C_n = 1$ $B > A$
si $C_n = 0$ $B < A$ ou $B = A$.

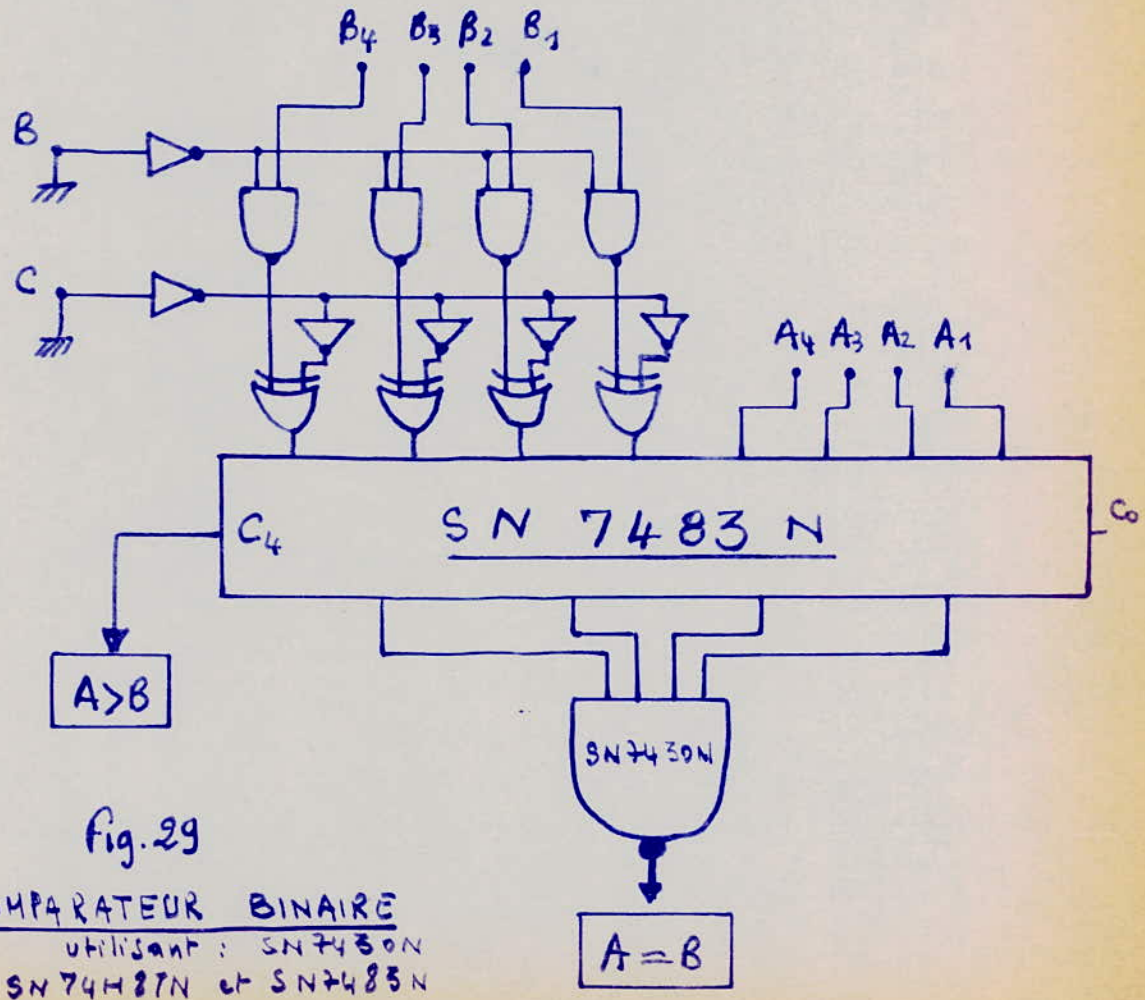


Fig. 29

COMPARATEUR BINAIRE
utilisant : SN7430N
SN74H87N et SN7483N

2. 2ème méthode:Comparaison par soustraction:

Elle consiste à soustraire un des deux opérandes et d'examiner l'existence éventuelle d'une retenue au delàsdu plus fort poids du module du résultat. On suppose qu'on désire effectuer(A - B).

Si $A > B$ la retenue au delà du plus fort poids est égale à 1
Si $A < B$ " " " " " 0

Donc,l'opération consiste à effectuer la différence sans se préoccuper du résultat;mais on tient compte simplement des retenues dans chaque étage.

On sait que,dans le cas d'une soustraction,le bit retenue s'écrit:

$$C_i = \bar{A}_i B_i + \bar{A}_i C_{i-1} + B_i C_{i-1}$$

Le schéma du comparateur est donné par la figure 28.

Une autre méthode consiste à additionner au nombre A le complément à 1 du nombre B.Pour des nombres de 4 bits,on utilise l'additionneur 4SN 7483 N et le générateur de complément SN 74H87 N .

Si les deux nombres sont égaux,tous les bits'somme' $_i$ sont égaux à 1.En reliant tous ces bits à un opérateur NAND (SN 7430 N) à 4 entrées,le signal logique 0 apparaît sur la sortie de cet opérateur.Par contre,le niveau logique 1 apparaît sur la sortie C_4 de l'additionneur quand le nombre A est supérieur au nombre B.Le bloc ADDITIONEUR-SOUSTRACTEUR fonctionne en soustrac--teur;donc: C = 0 et B = 1 .

Le schéma logique du comparateur binaire est donné par la figure 29.

3ème PARTIE

SYSTEME D'ENTREE-SORTIE

CHAPITRE 1 . ETUDE DES REGISTRES : SN 747SN

1.Généralités sur les bascules:

Dans l'unité de calcul,il est nécessaire de conserver en mémoire,pendant un certain temps,les données présentes à l'entrée d'un circuit.La réalisation de ce ciacuit repose sur l'utilisation des bascules électroniques.On distingue plusieurs types de bascules suivant le nombre de ses entrées : RS,RST,DT,ou D, T,JKT ou JK.

1.1.Bascule T :

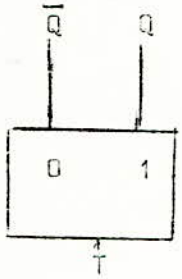
Elle possède une entrée T et deux sorties Q et $\overline{Q}$.Elle change d'état lorsqu'on applique un 1 sur son entrée quel que soit son état antérieur et con--serve cet état indéfiniment.Elle ne change pas d'état lorsqu'on applique un 0 sur son entrée T .

Tableau de vérité :

Q_t	T	Q_{t+1}
0	0	0
0	1	1
1	0	1
1	1	0

$$Q_{t+1} = \overline{Q}_t T + Q_t \overline{T}$$

$$\boxed{Q_{t+1} = Q_t \oplus T}$$

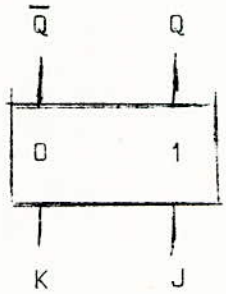


1.2.Bascule JK:

Elle possède deux entrées J et K et deux sorties Q et $\overline{Q}$.L'entrée J est destinée à la remise à 1 et la borne K à la remise à 0.

Tableau de vérité :

Q_t	J	K	Q_{tx+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0



$$Q_{t+1} = \overline{Q}_t J\overline{K} + \overline{Q}_t JK + Q_t \overline{J}K + Q_t J\overline{K}$$

$$\boxed{Q_{t+1} = J\,\overline{Q}_t + \overline{K}\,Q_t}$$

1.3. Bascule RS :

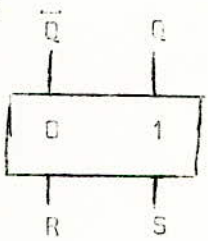
C'est une bascule JK avec une condition interdite RS = 0 .

Ta bleau de vérité:

Q_t	R	S	Q_{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
1	0	0	1
1	0	1	1
1	1	0	0

$$Q_t = \overline{Q}_t \overline{R}S + Q_t \overline{R}\,\overline{S} + Q_t \overline{R}S$$

$$\boxed{Q_{t+1} = S + \overline{R}\,Q_t}$$



1.4. Bascule RST :

C'est une bascule RS à laquelle on a ajouté une borne T. Lorsqu'on ap--plique un 1 en T, la bascule change d'état quel que soit son état an--térieur. Les conditions interdites sont :

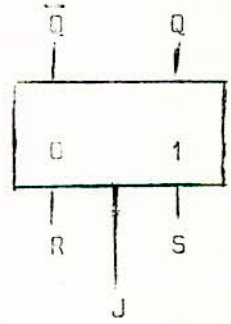
$$RST = 0 \quad RS = 0 \quad ST = 0 \quad RT = 0$$

Tableau de vérité:

Q_t	R	S	T	Q_{t+1}
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	1	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	0

$$Q_{t+1} = \overline{Q}\,\overline{R}\,\overline{S}\,T + \overline{Q}_t \overline{R}\,S\,\overline{T} + Q_t \overline{R}\,\overline{S}\,\overline{T} + Q_t \overline{R}\,S\,\overline{T}$$

$$\boxed{Q_{t+1} = S + T\overline{Q}_t + \overline{R}\,\overline{T}\,Q_t}$$



2. Etude d'une cellule de mémoire: bascule D:

Elle possède deux entrées D et H et deux sorties Q et $\overline{Q}$. Le fonctionnement de la bascule est le suivant:

au repos, on a Q qui est égal à 1 ou à 0 et donc $\overline{Q}$ est égal à 1 ou 0. Le signal d'horloge H n'est pas appliqué (H=0). Si D=1 et H=0, il ne pas--se rien. Mais si on maintient D à 1 et qu'on envoie une impulsion po--sitive sur H, après cette impulsion on aura Q=1 et $\overline{Q}$=0 quel que soit l'état de Q avant l'impulsion. Après l'impulsion, on peut changer la valeur de D, il ne se passe rien. Mais si on maintient D à 0 et qu'on envoie une impulsion sur H, après l'impulsion on aura Q=0 et $\overline{Q}$=1.

MEMOIRE 4 BITS : SN 7475N

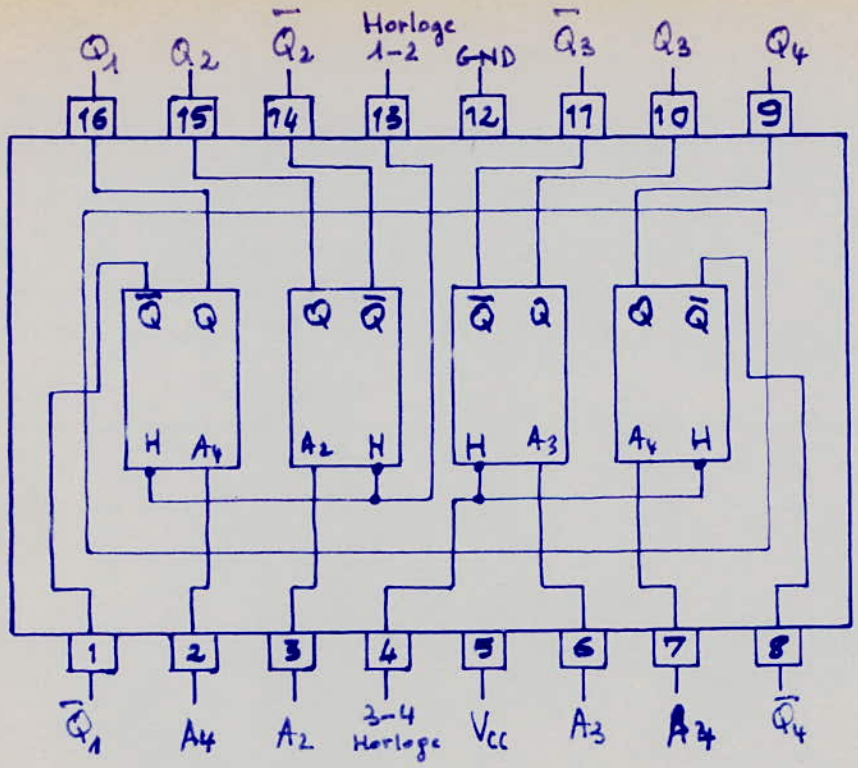


tableau de vérité

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

fig 35 : Brochage du SN 7475N. Schéma logique d'une cellule de mémoire et sa séquence.

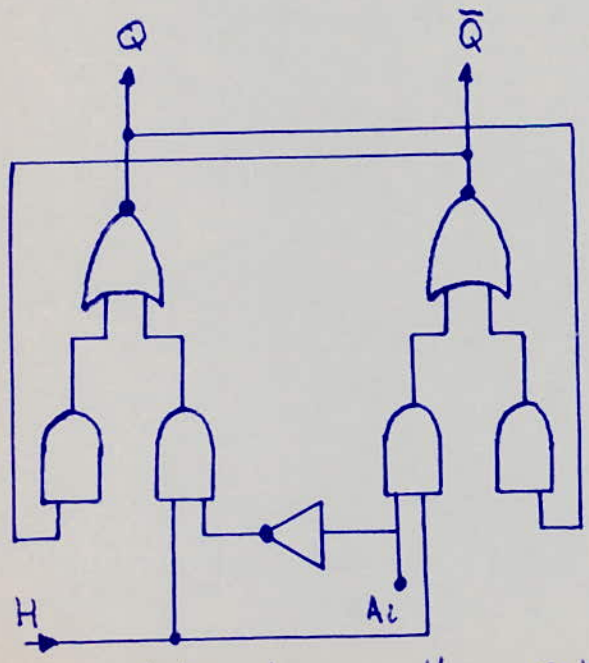
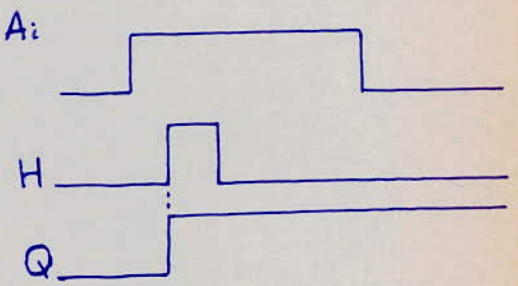


Schéma logique d'une cellule de mémoire



Séquence

Si on recommence l'opération en laissant D à 0, Q restera à 0 car il y était déjà.On aura donc,après l'impulsion d'horloge,sur la borne Q ce que l'on avait sur D avant l'impulsion.

Le changement d'état de D en dehors des impulsions d'horloge n'a aucune importance;la valeur de Q avant l'impulsion d'horloge n'influe pas sur le ré--sultat final.Ainsi,tant que l'horloge est au niveau haut,l'information sur l'entrée D ne peut être modifiée.

L'information présente sur l'entrée D est enregistrée au moment du front de montée de l'impulsion d'horloge et cette information doît être maintenue pendant tout le temps où l'horloge reste au niveau logique 1.

Le schéma logique de la cellule de mémoire est donné par la figure 35. Le circuit intégré SN 7474 N comprend deux cellules de mémoire.Il est capable de mémoriser un mot de deux bits.Le schéma du SN 7474 N est représenté par la figure 36 .

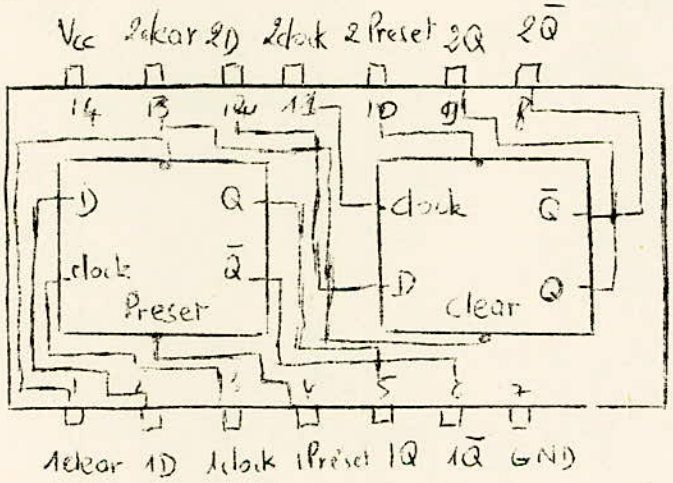


Tableau de vérité

t_n	t_{n+1}	
D	Q	$\bar{Q}$
0	0	1
1	1	0

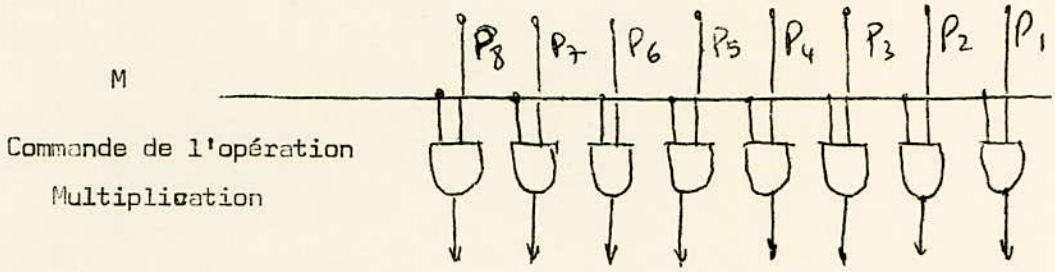
Fig 36 . Circuit intégré SN 7474 N

Etant donné que notre unité de calcul opère sur des mots de 4 bits, on utilise alors le circuit intégré SN 7475 N .Ce circuit comprend 4 bascules reliées deux à deux.Chaque couple est commandé par le même signal d'horloge. Le schéma logique du SN 7475 N est donné par la figure 35 .

CHAPITRE 2:CIRCUITS DE COMMANDE DE SORTIE DES OPERATIONS

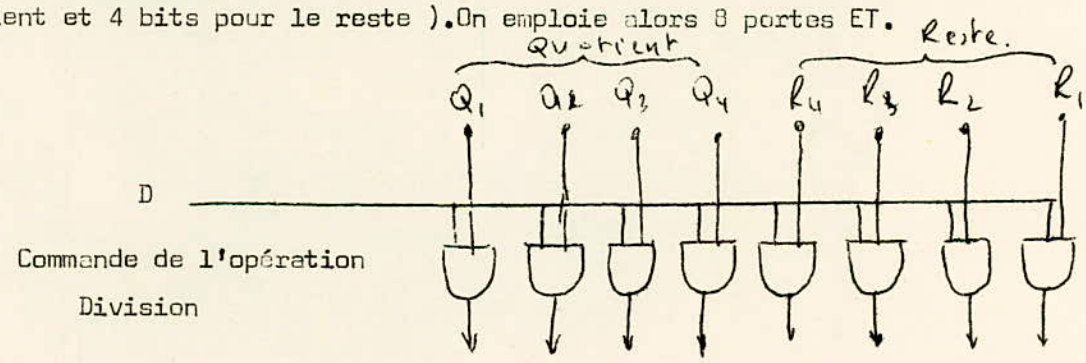
Ces circuits permettent de transférer le résultat de l'opération des circuits de calcul vers les régistres de sortie.Ce transfert s'effectue suivant une commande d'opération.Il peut se matérialiser par des portes ET.

Pour la multiplication,on utilise 8 portes ET.Une commande de multiplication ouvre les 8 portes et le résultat est transmis aux régistres de sortie.



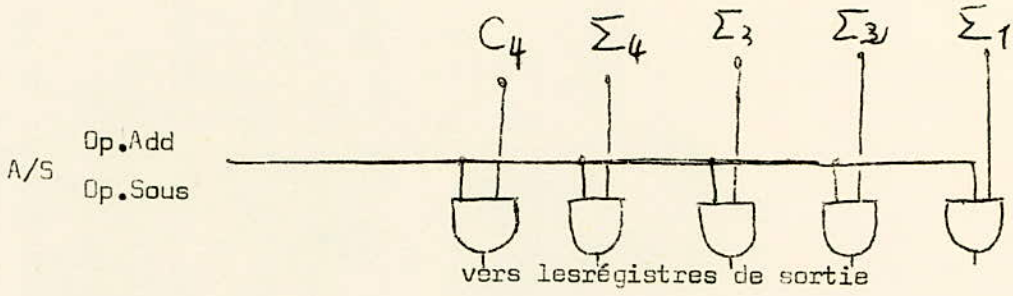
vers les régistres de sortie

On sait que la division donne un résultat de 8 bits(4 bits pour le quotient et 4 bits pour le reste).On emploie alors 8 portes ET.



vers les régistres de sortie

L'addition ou la soustraction donne un résultat de 5 bits au maximum. Pour l'addition,le résultat apparaît en clair .Mais pour la soustraction ,il faut complémenter la retenue d'ordre supérieur C_4 .Ceci provient du fait qu'on a réduit la soustraction à une addition de complément à 2.On utilise 5 portes ET .

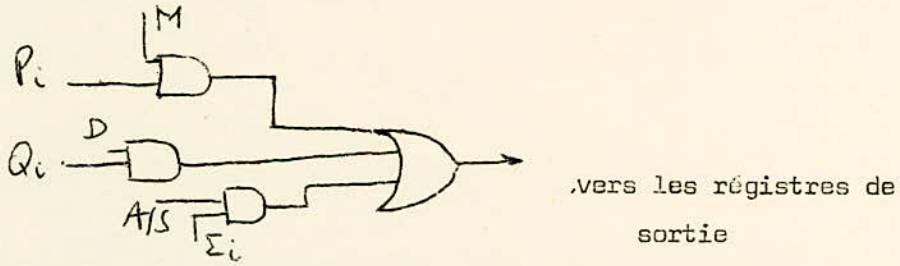


Pour effectuer une addition ou une soustraction,on doit agir sur deux commandes :

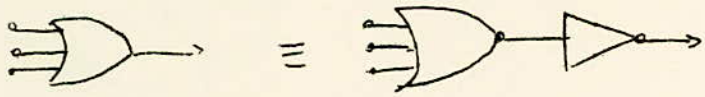
- une commande A/S permettant d'ouvrir les 5 portes ET ,
- et la commande C de l'étage Additionneur - Soustracteur.

(C=0 Op. Sous , C=1 Op. Add)

Suivant la nature de l'opération,les registres de sortie emmagasinent le résultat de l'opération désirée.On peut réunir les bits de même poids à l'aide d'une porte OU à 3 entrées au maximum.

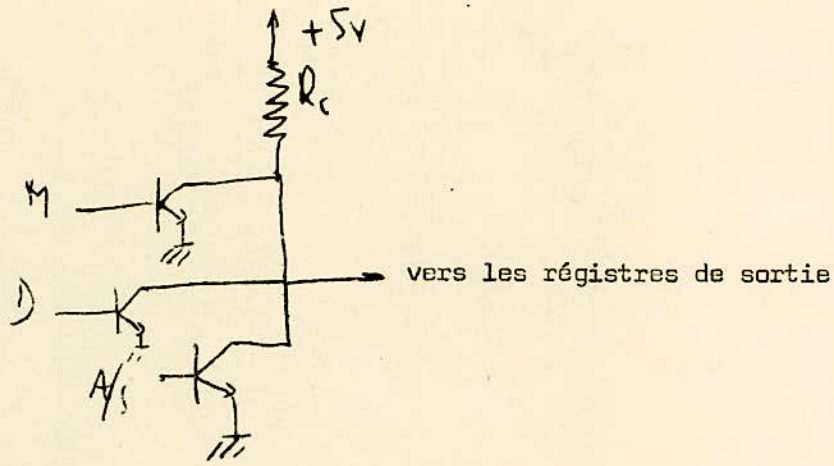


Etant donné qu'on n'a pas commandé des portes OU,on peut réaliser une porte OU à l'aide d'une porte NOR suivie d'un inverseur:



Cette méthode utilise beaucoup de portes et complique la circuiterie. Pour réduire au munimum le nombre de portes utilisées ,on utilise une mé--thode moderne celle du OU CABLE .

Si l'on regarde,la constitution d'une porte ET du circuit intégré SN 7409 N,on constata que le collecteur du transistor de sortie est à l'état ouvert.On recueille sur le collecteur le bit résultat d'une opéra--tion quelconque.En reliant les collecteurs de sortie entre-eux à travers une résistance de charge,on réalise le OU CABLE.



Pour les opérations,on utilise le circuit intégré SN 7409 N qui comporte 4 portes ET et dont le brochage est représenté par la figure 37 .

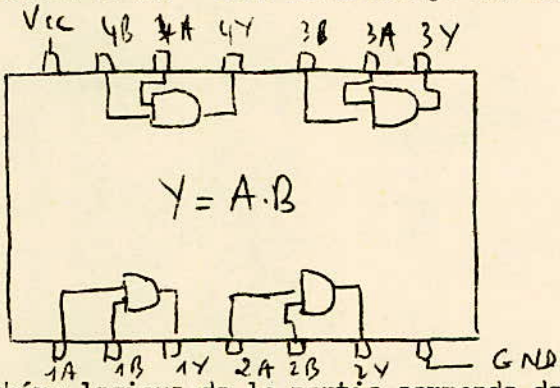


Fig. 37

Le schéma logique de la partie commande de sortie des opérations est donné par la figure 38 .

CIRCUITS DE COMMANDE DES OPERATIONS

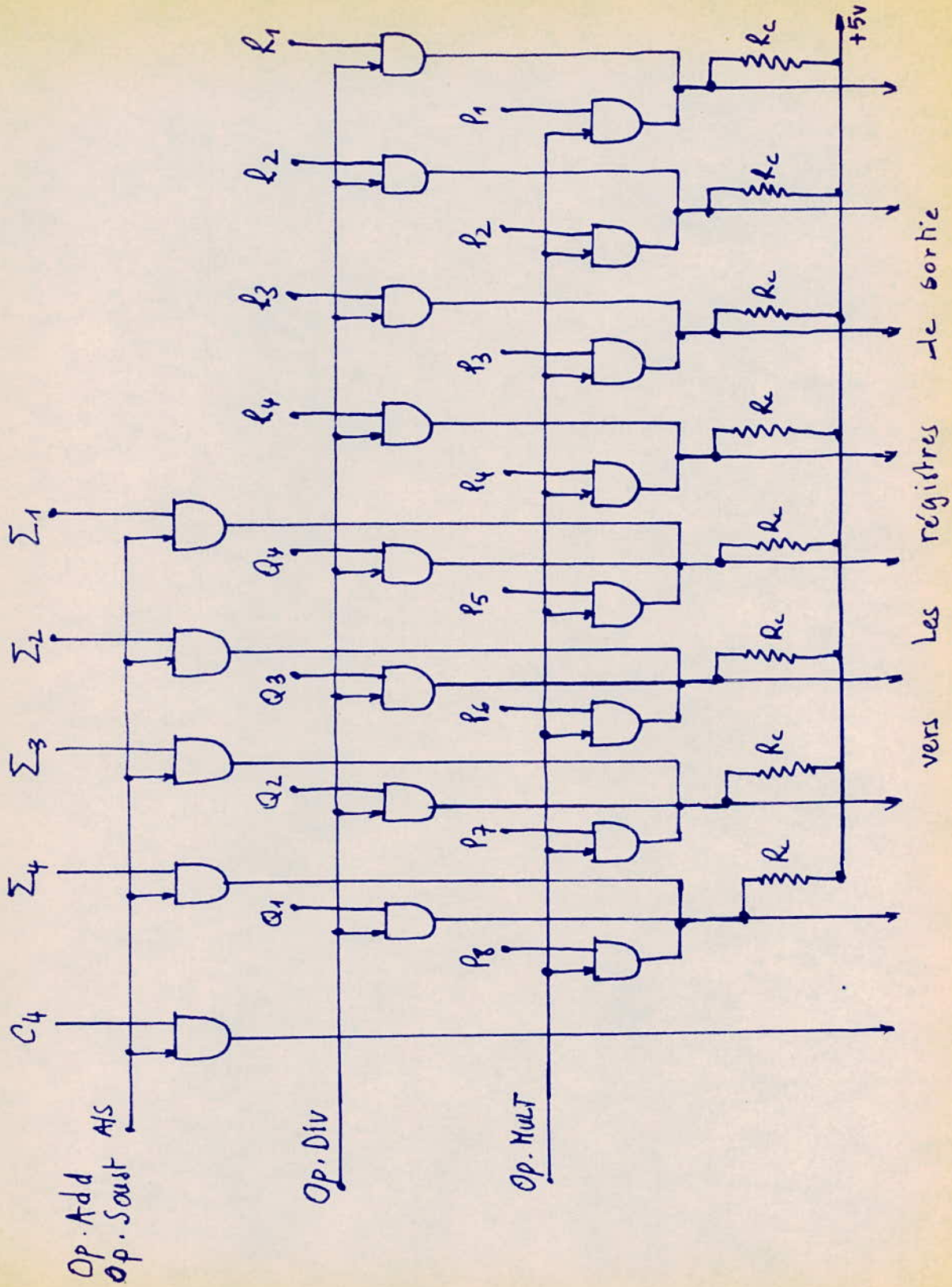


Fig 38

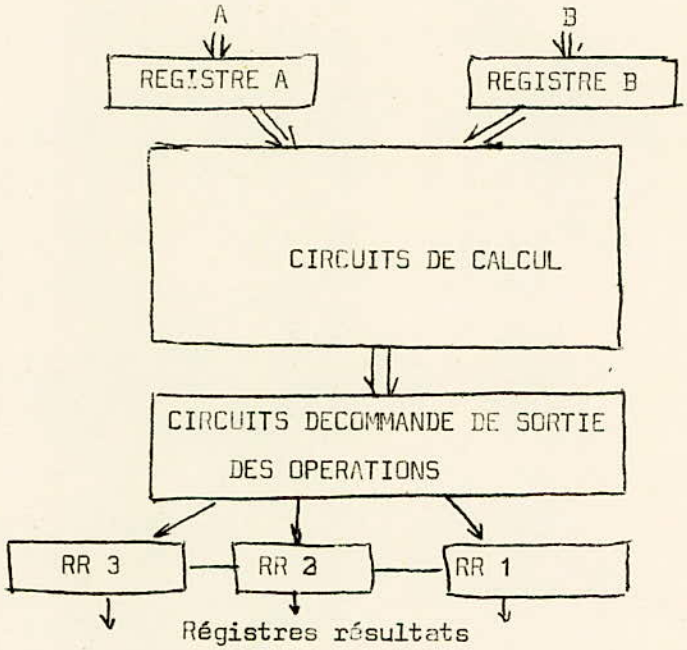
4 ème PARTIE

CONSTRUCTION DE L'UNITE DE CALCUL

1.Caractéristiques de l'unité de calcul:

L'unité de calcul qu'on veut réaliser est du type parallèle.Elle doit effectuer les 4 opération élémentaires : addition, soustraction,multiplica--cation et division;sur des nombres d'une longueur de 4 bits au maximun. Les nombres à traiter sont des nombres binaires positifs représentés en vir--gule fixe.Le résultat est donné es code binaire.

Le schéma synoptique de l'unité de calcul est donné par la figure 39



Le régistre résultat RR 1 emmagasine le reste de la division $R_4R_3R_2R_1$ et les 4 bits $P_4P_3P_2P_1$ de la multiplication.Le régistre RR 2 emmagasine le quotient $Q_1Q_2\ Q_3Q_4$, les 4 bits $P_8P_7P_6P_5$ de la multiplication et les 4 bits $\Sigma_4\Sigma_3\Sigma_2\Sigma_1$ de l'addition ou de la soustraction.RR 3 reçoit simplement le bit C_4 .

L'aspect extérieur de la maquette est représenté par la figure 40.Pour la construction de cette unité,on utilisera uniquement des circuits intégrés du type T.T.L.

2.Caractéristiques des circuits intégrés T.T.L:

- Tension d'alimentation 5 Volts (4,5 V à 5,5 V)
- Courant d'entrée au niveau logique 1 = 40 mmA à 2,4V;1mA à 5,5V.
- Courant de sortie au niveau logique 0=1,6 mA à 5,5 V .
- Fan-out (ou sortance) = 10 .

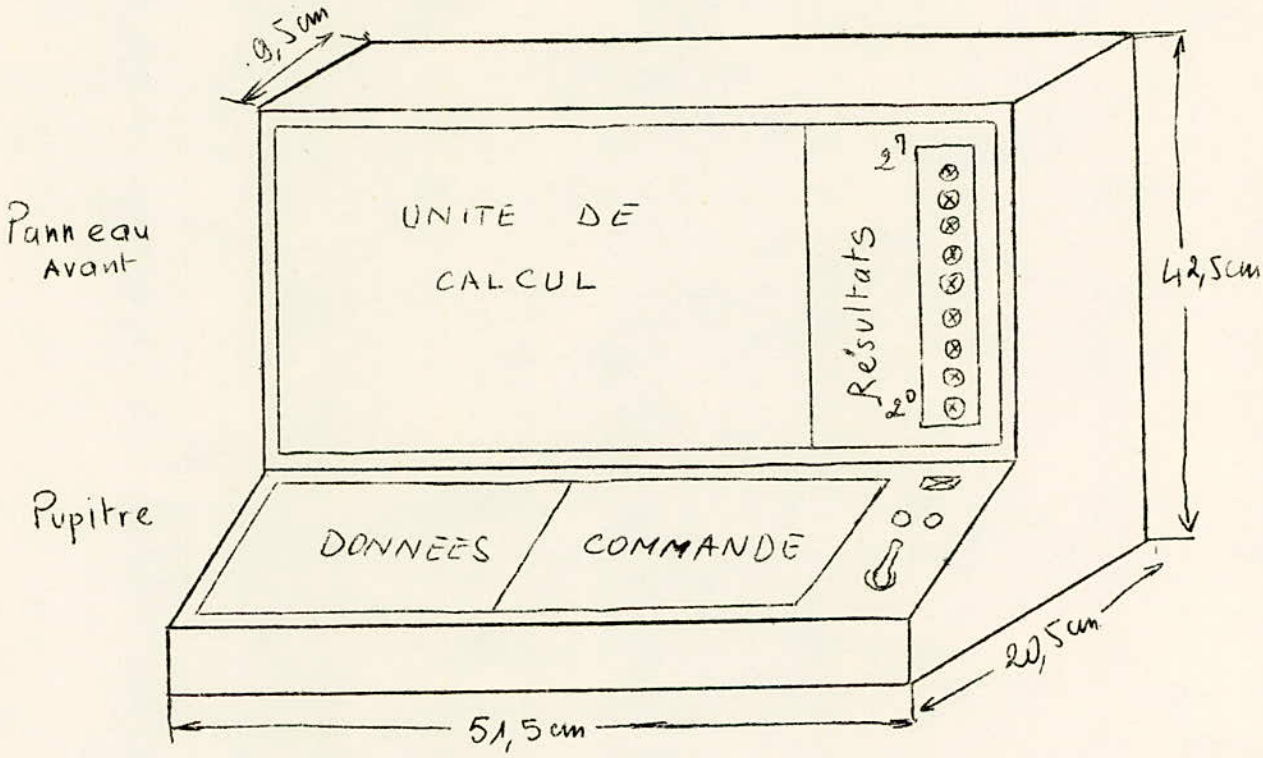


fig.40. Vue générale de la maquette.

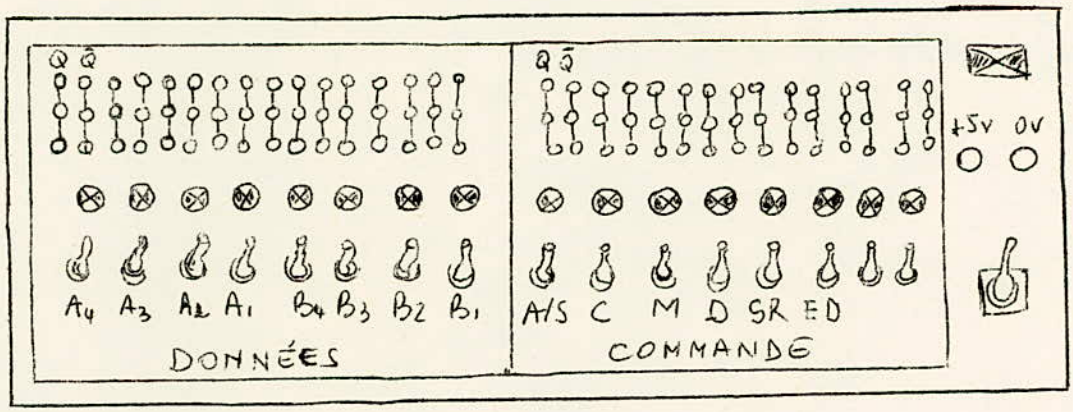


fig.41. Vue de face du pupitre.

SR : sortie du Résultat
ED = Entrée des Données.

- une entrée en l'air correspond au niveau logique 1 .
- Boîtier enfichable type TO 116

3.Circuits intégrés utilisés:

3.1.Addition-Soustraction:

- 1 SN 7483 N
- 1 SN 74H87 N
- 1/6 SN 7404 N .

3.2.Multiplieur :

- 3 SN 7483 N
- 4 SN 7408 N .

3.3.Diviseur :

- 5 SN 7483 N
- 5 SN74H87 N
- 2 SN 7404 N
- 1 SN 74 86 N

3.4.Circuits de commande de sortie des opérations:

- 5 SN 7409 N

3.5.Régistres d'entrée:

-2 SN 7475 N

3.6.Registre de sortie:

- 3 SN 7475 N

3.7.Commande :

- 2 SN 7404 N

4.Visualisation des données et du résultat:

L'affichage des données se fait manuellement à l'aide d'interrupteurs.La sortie 1Q du registre SN 7475 N attaque la base d'un transistor 2N 2222 à tra--vers une résistance de 680 Ω .Le collecteur est chargé par une lampe (6,5V-70mA) Le schéma est donné par la figure 42.?

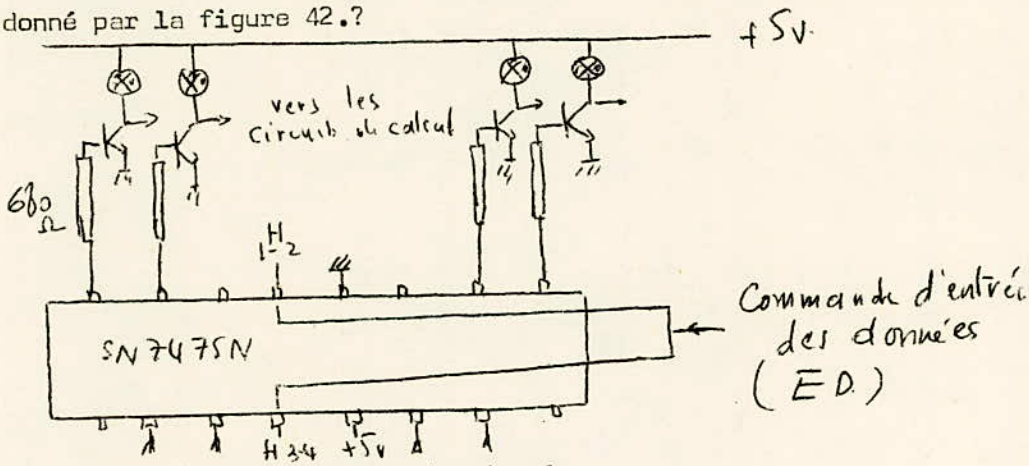
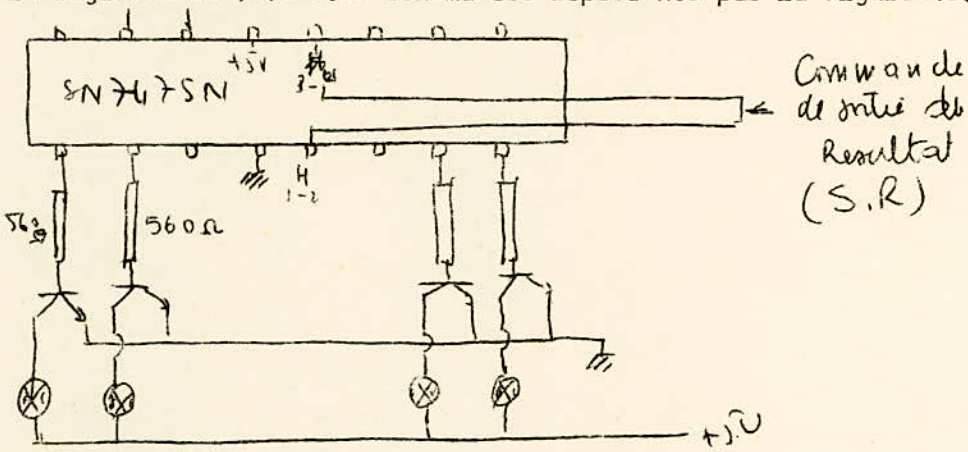


Fig 42 . Visualisation des données;

Pour la visualisation du résultat,on utilise le même montage,sauf qu'ici ceux-sont les bits délivrés par le circuit de commande de sortie des opérations qui attaquent le registre SN 7475 N.Le schéma est représenté par la figure 43.



5.Câblage:

Les différents logiques sont réalisés sur circuit imprimé à double face.

Différentes plaques :

* Plaques IA IB comprend le bloc A/S,le multiplieur,le diviseur,le bloc de commande de sortie des opérations et les 3 régistres de sortie.

  Plaque IA : face avant sur laquelle sont fixés les circuits intégrés.

  Plaque IB : envers (face de soudure).

* Plaque IIA IIB visualisation des données.

* Plaque IIIA IIIB Commande de sortie des opérations.

* Plaque IV Amplificateurs des lampes de sortie .

Le schéma global de l'unité de calcul est représenté par la figure 44.

6.Alimentation :

On utilise une alimentation stabilisée délivrant 5 Volts nécessaire pour l'alimentation des circuits intégrés et des lampes.

ANNEXE 1

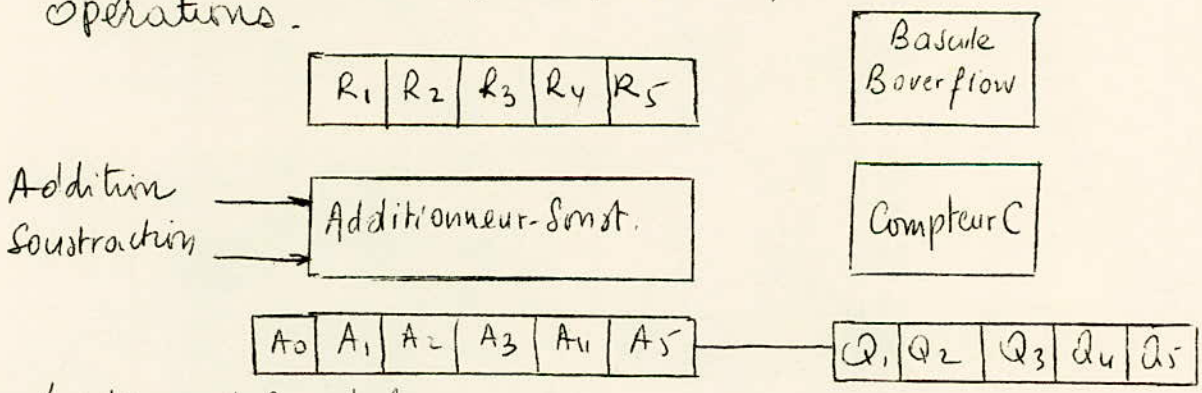
UNITE ARITHMETIQUE ET LOGIQUE

(Cas des nombres représentés en valeur absolue et signe)

Organisation de l'unité arithmétique :

Elle comprend :

- un registre accumulateur A qui contient un des 2 opérandes en début d'opération ainsi que les résultats intermédiaires en cours d'opération.
- Un registre multiplicateur - quotient Q relié au précédent pour former le registre AQ qui contient le résultat de la multiplication et celui de la division.
- Un registre R qui contient l'autre opérande pendant toute la durée de l'opération.
- Un additionneur - soustracteur du type parallèle.
- Un compteur de décalages C.
- Une bascule B overflow qui indique l'invalidité des opérations.



Abréviations et Symboles :

(A) = Contenu du registre A ; (C) contenu du compteur C

(A) ⇒ B transfert du contenu de A dans B.

(C) - 1 ⇒ C compteur C diminué d'une unité.

(A) + (B) ⇒ Somme logique du contenu de A et B puis transférée dans B.

B op = Bascule opération.

B op inv = Bascule opération invalide.

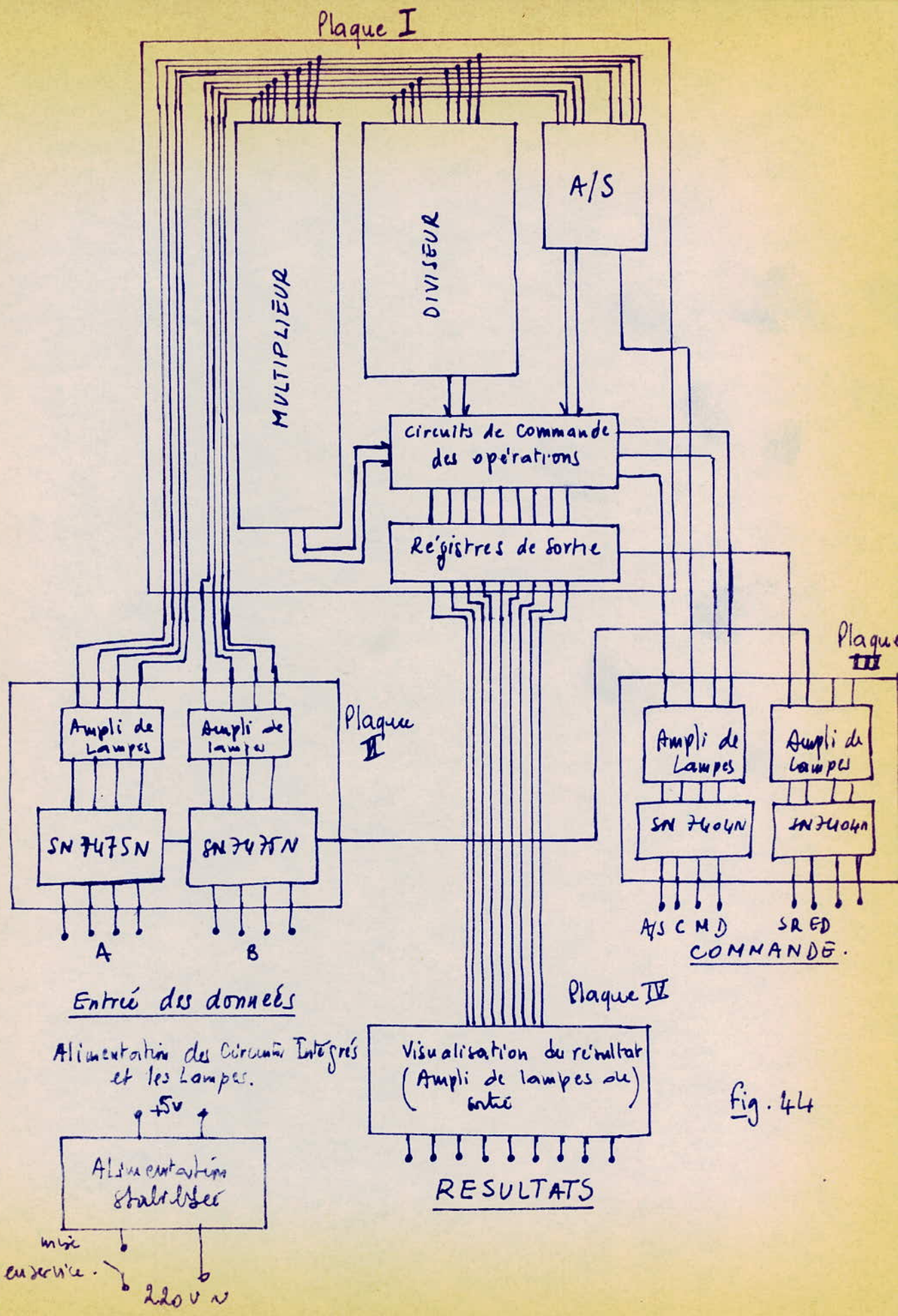

Plaque I
MULTIPLIEUR
DIVISEUR
A/S
Circuits de Commande des opérations
Régistres de Sortie
Plaque II
Ampli de Lampes
Ampli de lampes
SN 7475N
SN 7475N
A
B
Entrée des données
Plaque III
Ampli de Lampes
Ampli de Lampes
SN 7404N
SN 7404N
A/S CMD
SRED
COMMANDE.
Plaque IV
Visualisation du résultat (Ampli de lampes de sortie)
RESULTATS
Alimentation des Circuits Intégrés et les Lampes.
+5v
Alimentation stabilisée
mise en service.
220 v ~

fig. 44

ADDITIONNEUR - SOUSTRACTEUR.

Les nombres traités ont une longueur fixe de 5 bits. Les bits A_1 et R_1 sont réservés aux signes des 2 nombres. Pour convension, on attribue au bit 0 le signe plus (+) et au bit 1 le signe moins (−). La comparaison des signes des 2 nombres est donnée par le circuit OU EXCLUSIF qui délivre un 0 si les 2 signes sont identiques et un 1 s'ils sont différents. Ce résultat de la comparaison est mémorisé dans une bascule signe Bsgn. Suivant le code opération (addition ou soustraction) et le contenu de Bsgn, l'unité arithmétique effectue les opérations suivant le tableau suivant :

R_1	A_1	Bsgn	Opération effectuée	Opération sur les modules.
0	0	0	Addition	Addition des modules
0	1	1		Soustraction des modules
1	0	1		Soustraction des modules
1	1	0		Addition des modules.
0	0	0	Soustraction	Soustraction des modules
0	1	1		Addition des modules
1	0	1		Addition des modules.
1	1	0		Soustraction des modules

Le tableau donne :

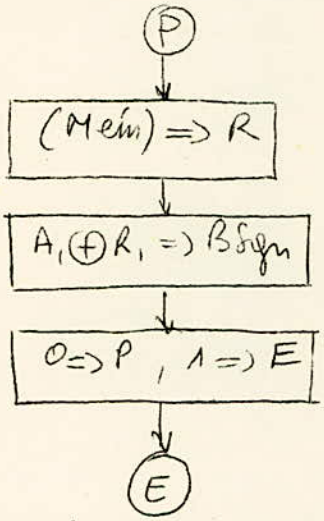
$$\boxed{\begin{aligned} \text{Add mod} &= \overline{\text{Bsgn}} \,.\, \text{Add} + \text{Bsgn} \,.\, \text{Soust} \\ \text{Soust mod} &= \text{Bsgn} \,.\, \text{Add} + \overline{\text{Bsgn}} \,.\, \text{Soust} \end{aligned}}$$

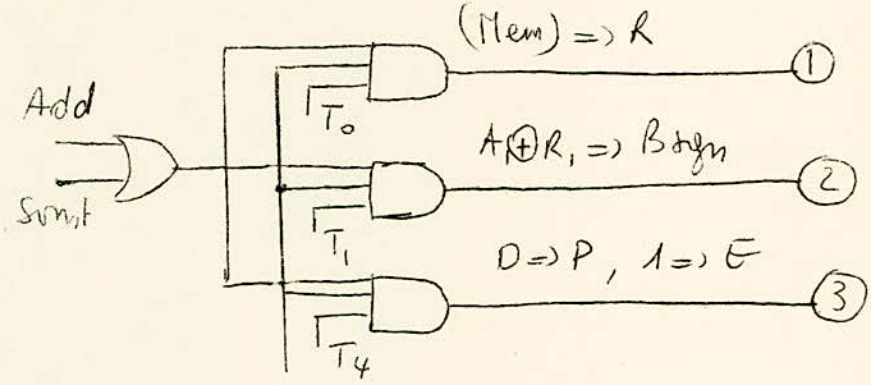
Le résultat de l'opération apparait dans le registre accumulateur. S'il n'existe pas de report au delà du plus fort poids du module, le résultat est exact. S'il existe un report dans la position A_0, il faut :

- changer le signe du registre accumulateur
- complémenter à 2 le module du résultat.

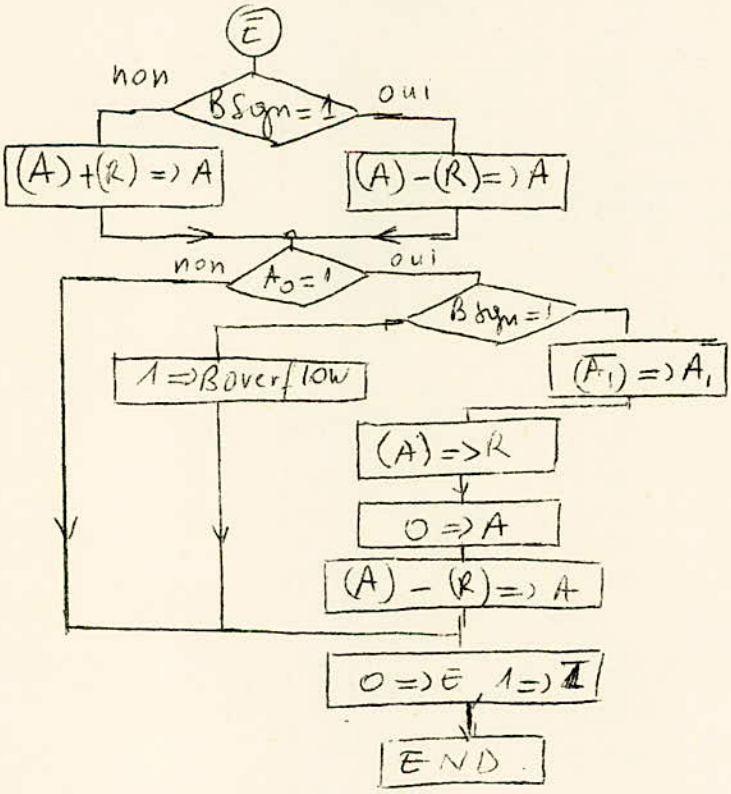
Cette complémentation à 2 s'obtient par un transfert du contenu de A (moins le signe) dans R, par une remise à 0 des 4 positions du module de l'accumulateur et par une soustraction du contenu de R au contenu de A. Une bascule B overflow signal un résultat invalide.
Pour effectuer une opération, on procède en 2 cycles :

1- Cycle de préparation : il est identique pour les 2 opérations.

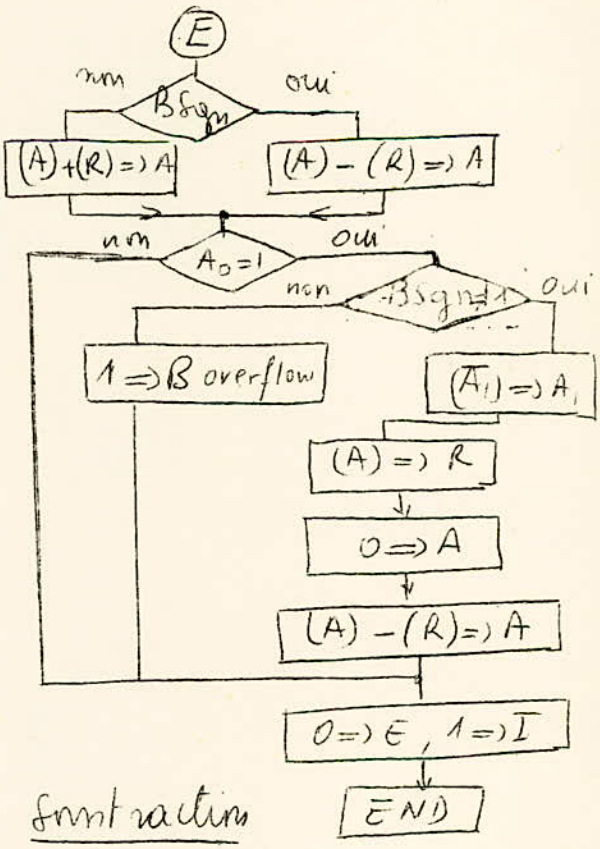




2- Cycle d'Exécution :

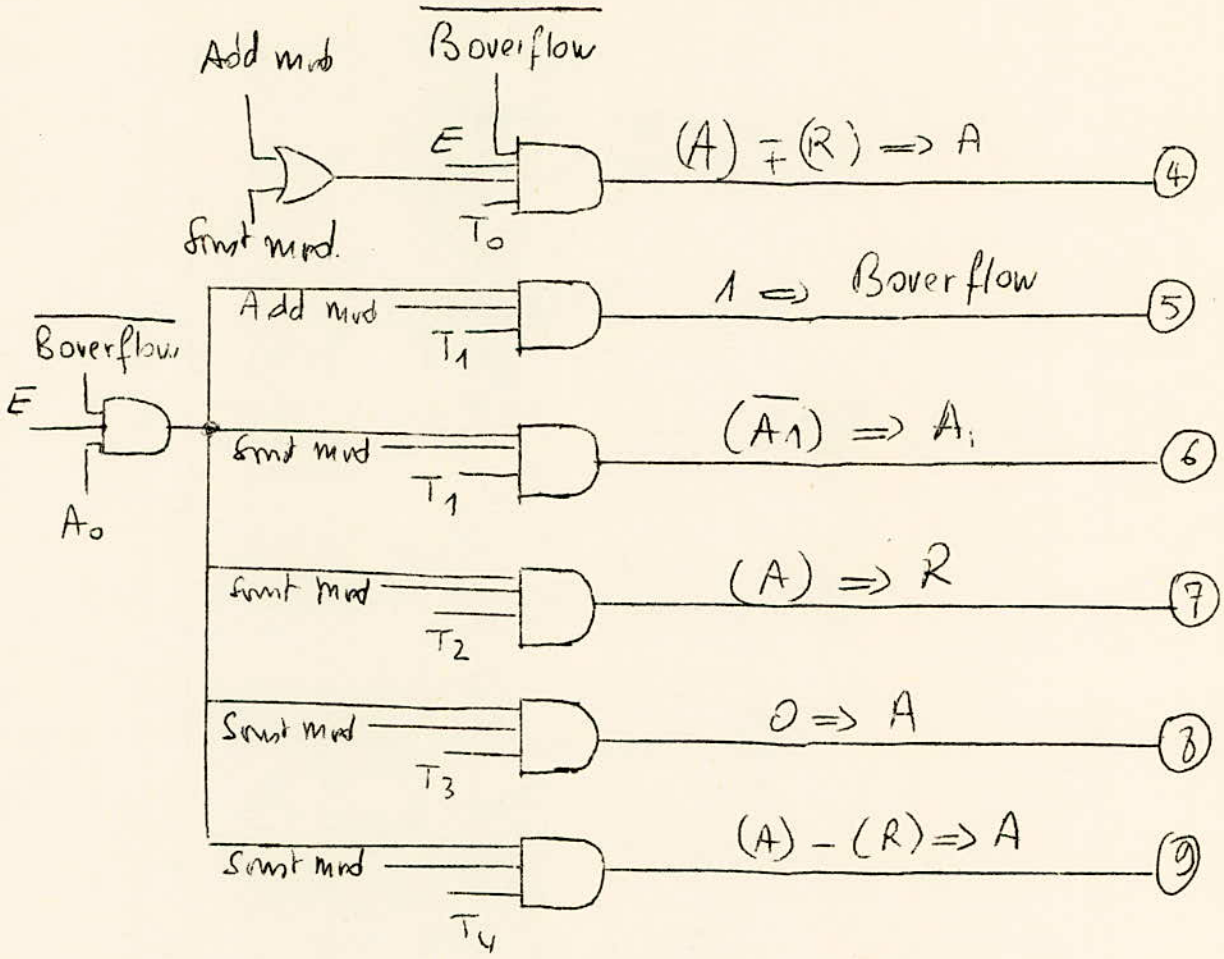


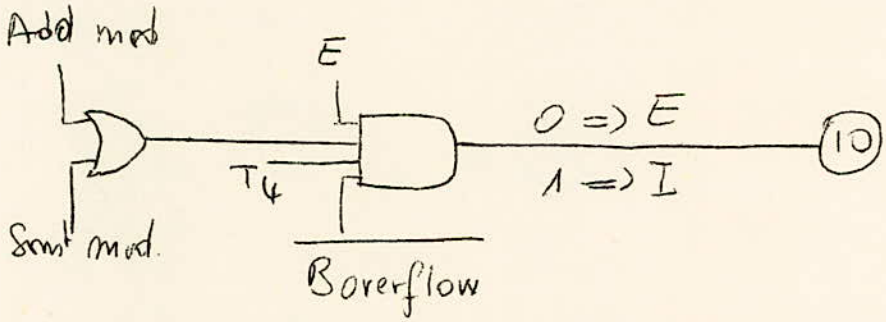
Addition



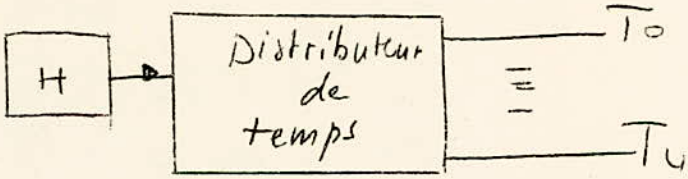
Soustraction

Réalisation du cycle d'exécution :

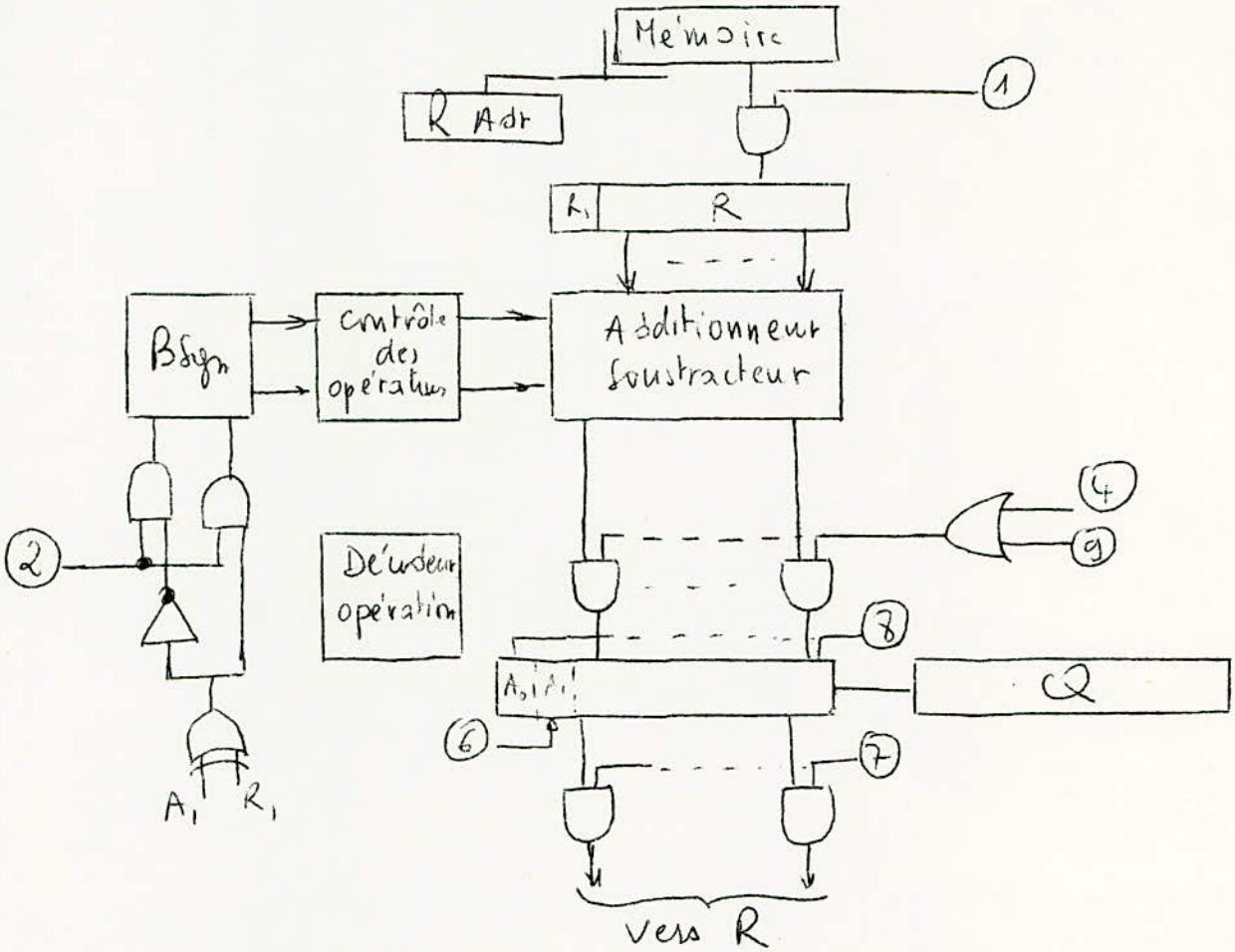




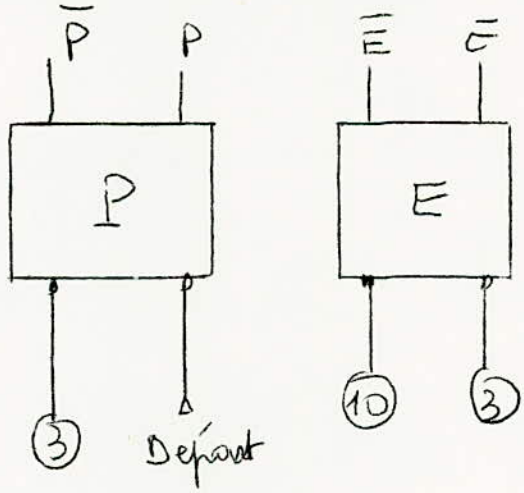
Les signaux T_0 à T_4 sont délivrés par un distributeur de temps commandé par une horloge H.



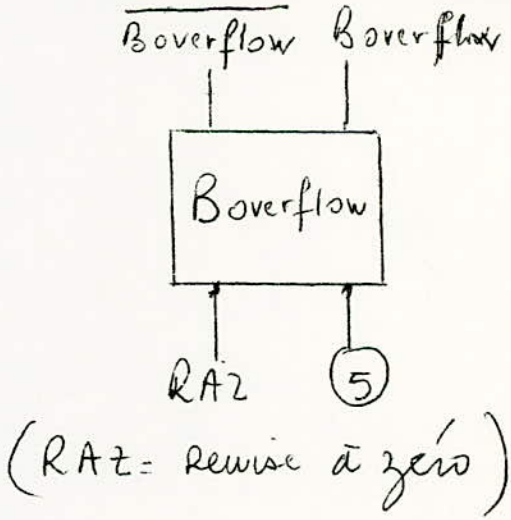
Le schéma général de l'additionneur - soustracteur avec ses circuits associés se présente comme suit :



Bascules P et E des cycles de préparation et d'exécution



Bascule de débassement de capacité de l'accumulateur



(RAZ = Remise à zéro)

MULTIPLIEUR BINAIRE

La multiplication de 2 nombres représentés en valeur absolue et signe s'effectue en deux temps :
— signe du résultat
— produit des deux modules.

Le signe du résultat est le OU EXCLUSIF des signes du multiplicateur et du multiplicande.

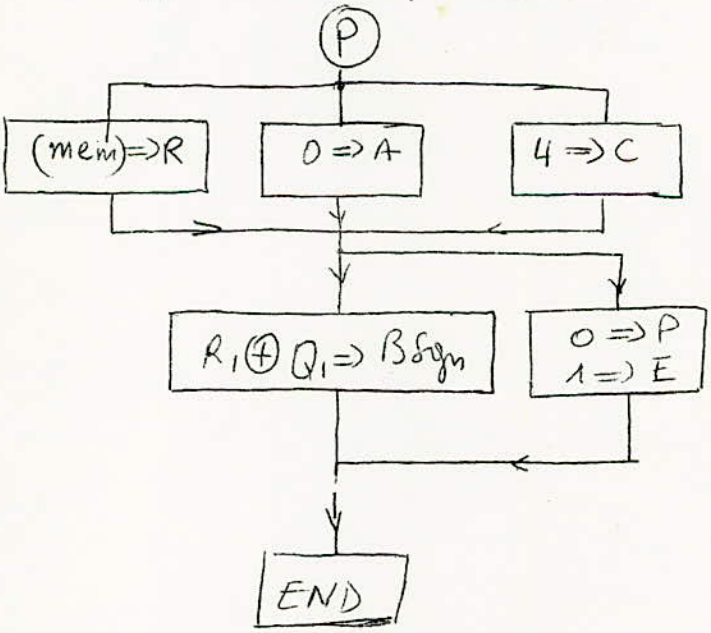
$$B_{sgn} = A_1 \oplus R_1$$

Un compteur C contient le nombre de décalages. Celui-ci est égal au nombre de bits du multiplicateur.

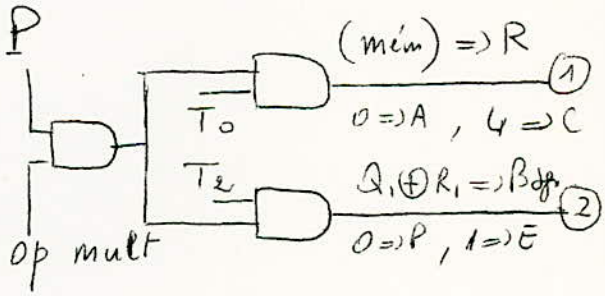
Si $Q_5 = 0$ on ne fait aucune opération
Si $Q_5 = 1$ on additionne le contenu du registre R à celui de A.

Pour effectuer la multiplication de 2 nombres, on procède en 2 cycles :

1- Cycle de Préparation :

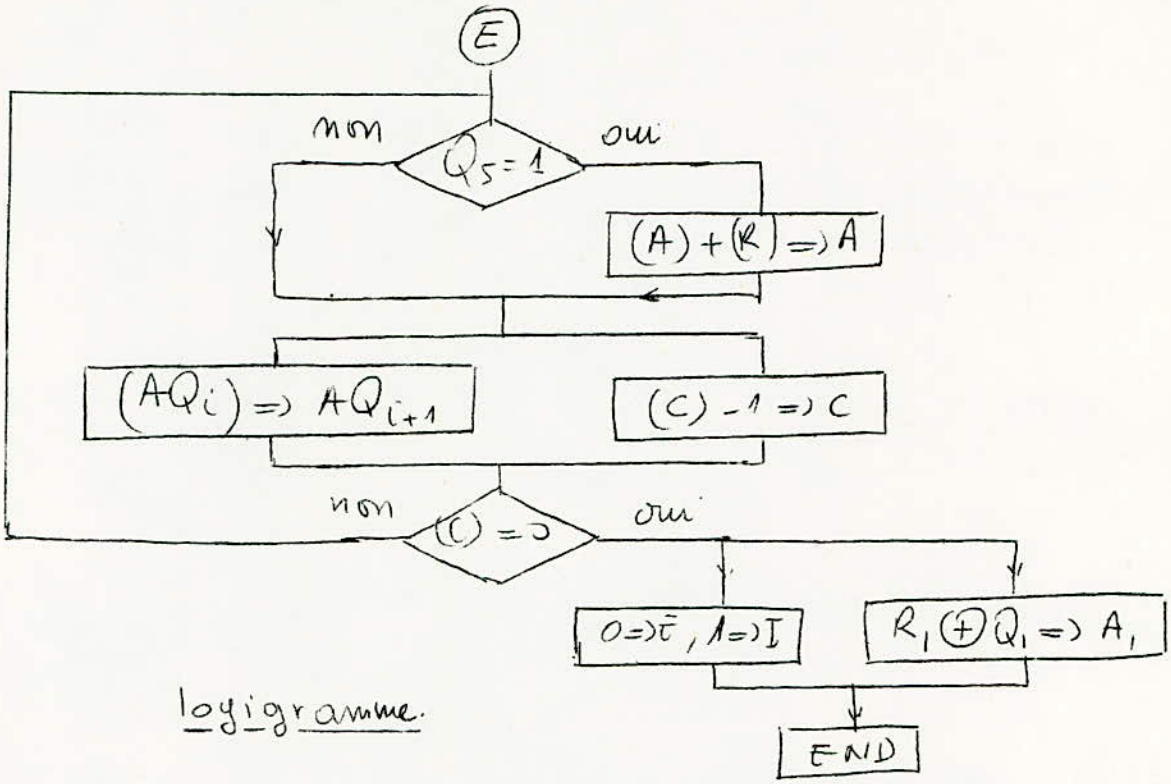


Logigramme.

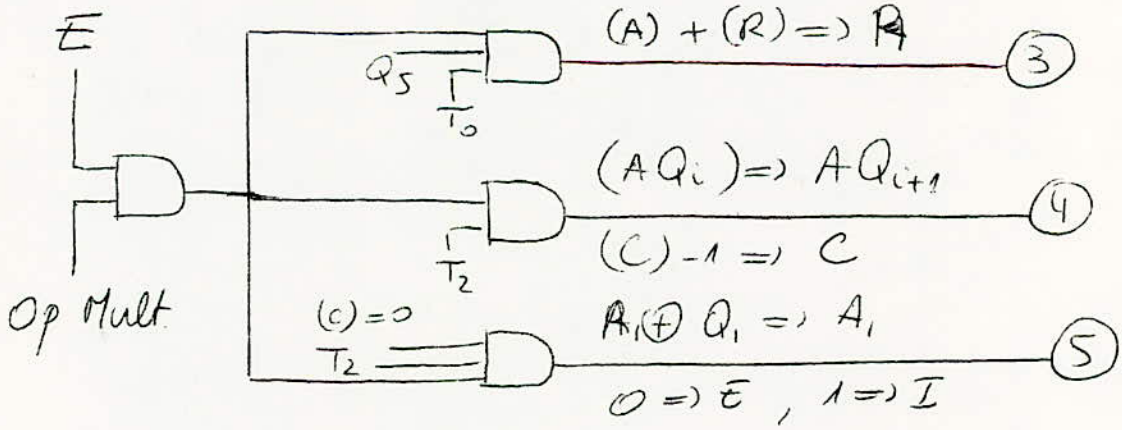


Réalisation du cycle de préparation

2. Cycle d'exécution :

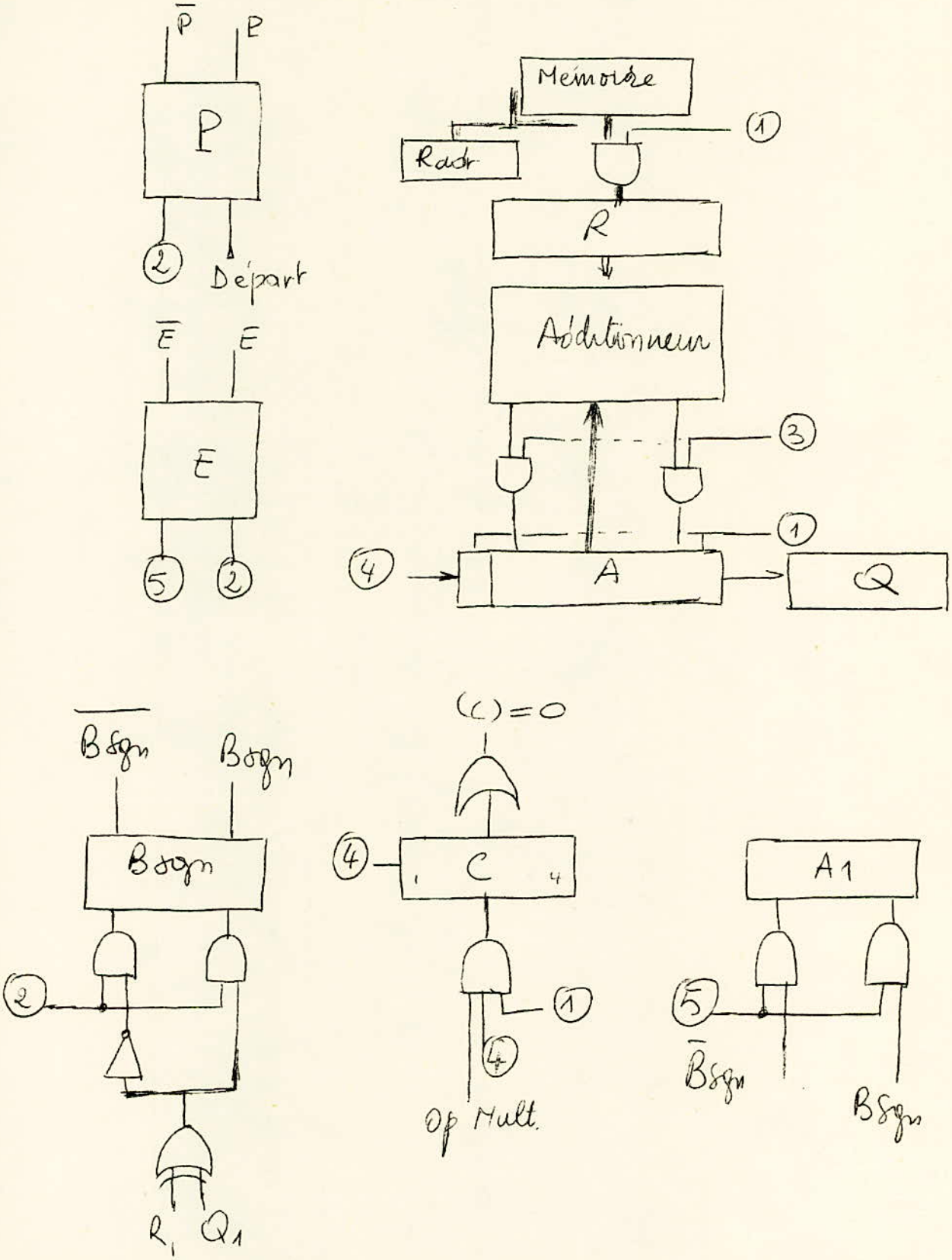


logigramme.



Réalisation du cycle d'exécution

Schéma complet du multiplieur



DIVISEUR BINAIRE

Comme pour la multiplication, le signe du résultat est élaboré avant le début de l'opération et pris en mémoire par la bascule signe. On effectue ensuite la soustraction du module du diviseur des plus forts poids du module du dividende et des restes successifs préalablement décalés d'un rang à gauche. On restaure le dividende chaque fois que la soustraction du diviseur donne un résultat négatif ($A_0 = 1$).

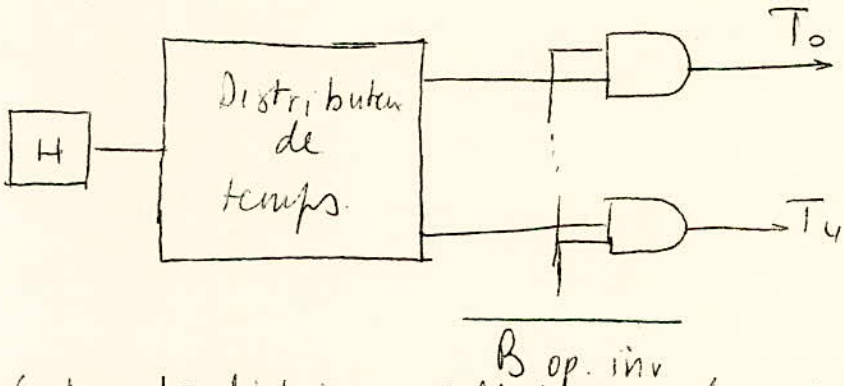
Cette restauration s'obtient par une addition du diviseur à chaque reste partiel négatif. Le signe de chaque reste partiel impose la nature de l'opération suivante ainsi que le digit correspondant du quotient.

Si $A_0 = 1$, le bit quotient est égal à 0 et l'opération suivante est une addition.

Si $A_0 = 0$, le bit quotient est égal à 1 et l'opération suivante est une soustraction.

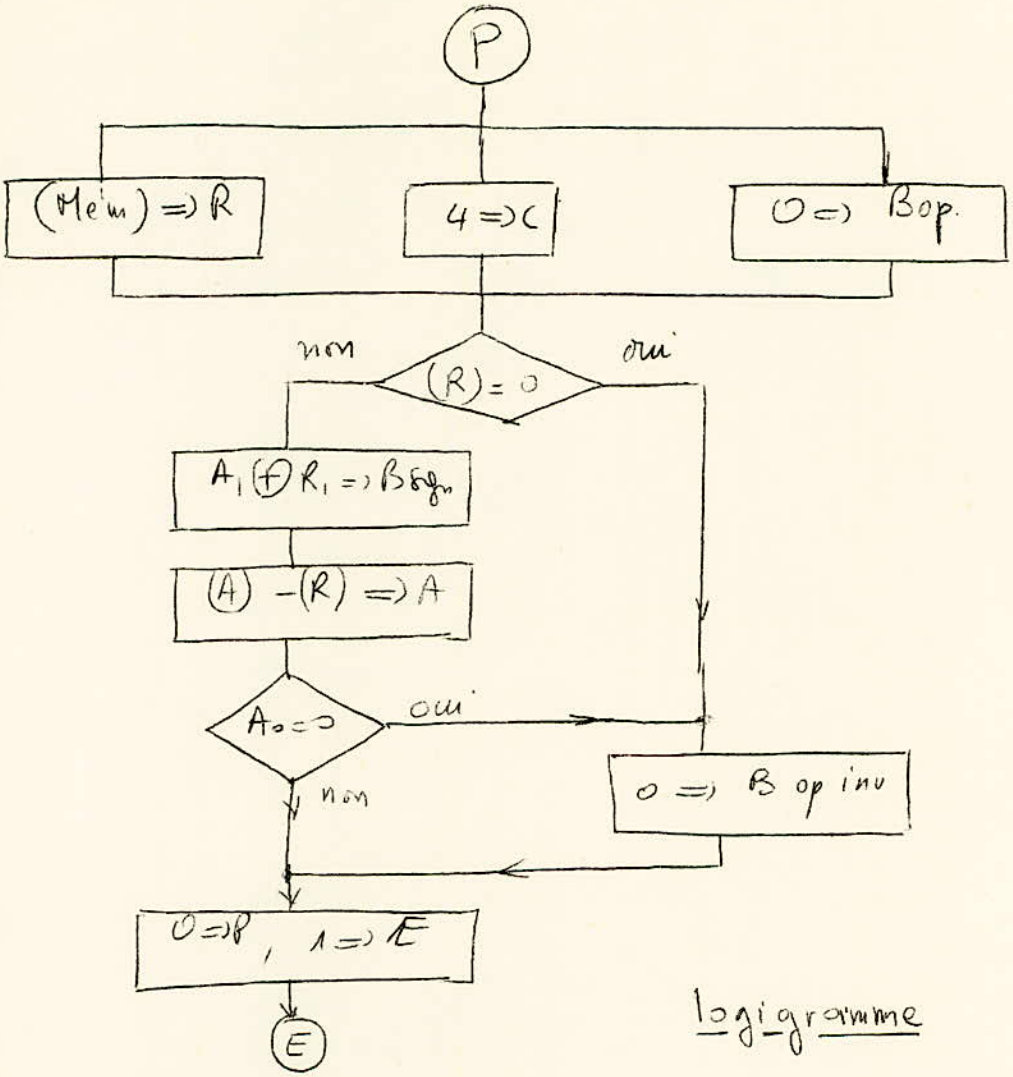
Une bascule Bop inv indique une division invalide.

Les signaux T_0 à T_4 sont délivrés par le distributeur de temps lorsque l'opération est valide.
Les impulsions T_0 à T_4 sont inhibés si Bop inv = 1

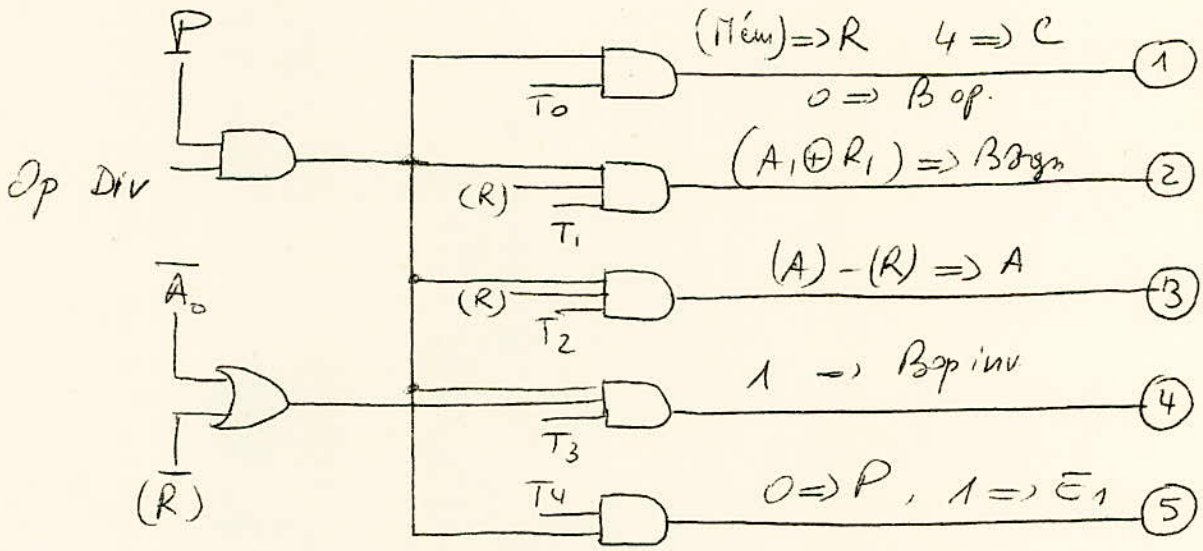


L'opération de division s'effectue en deux cycles.

1 - Cycle de préparation :

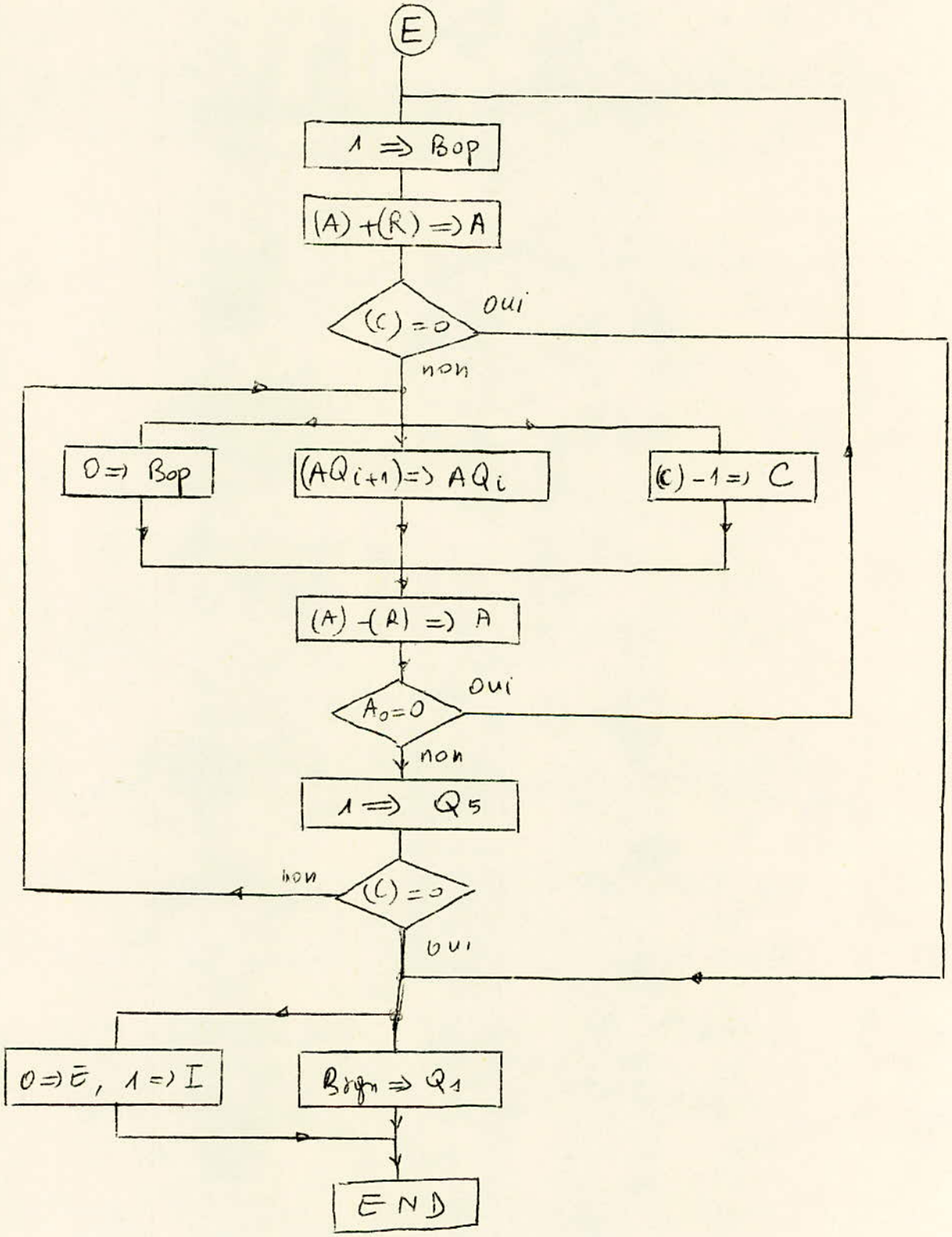


logigramme

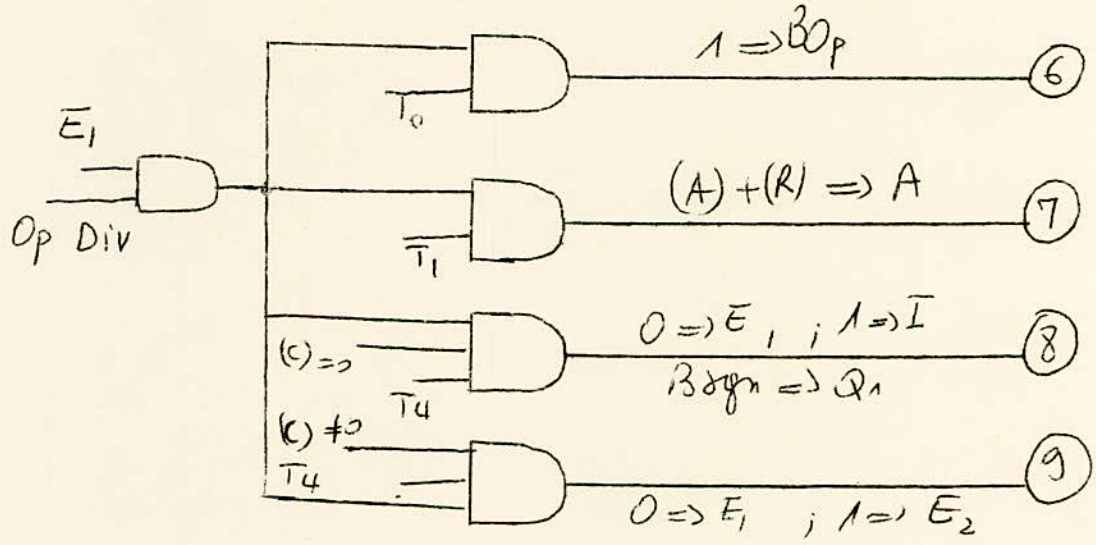


Réalisation du cycle de préparation

2. Cycle d'exécution :



Réalisation du Cycle d'exécution :



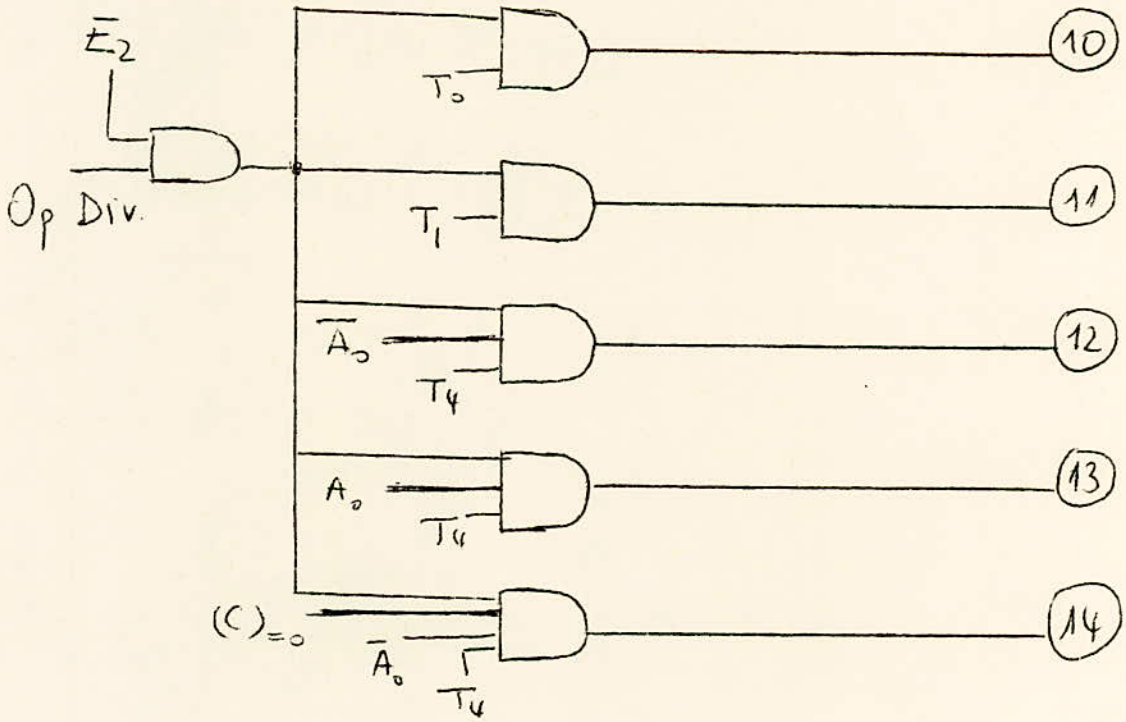
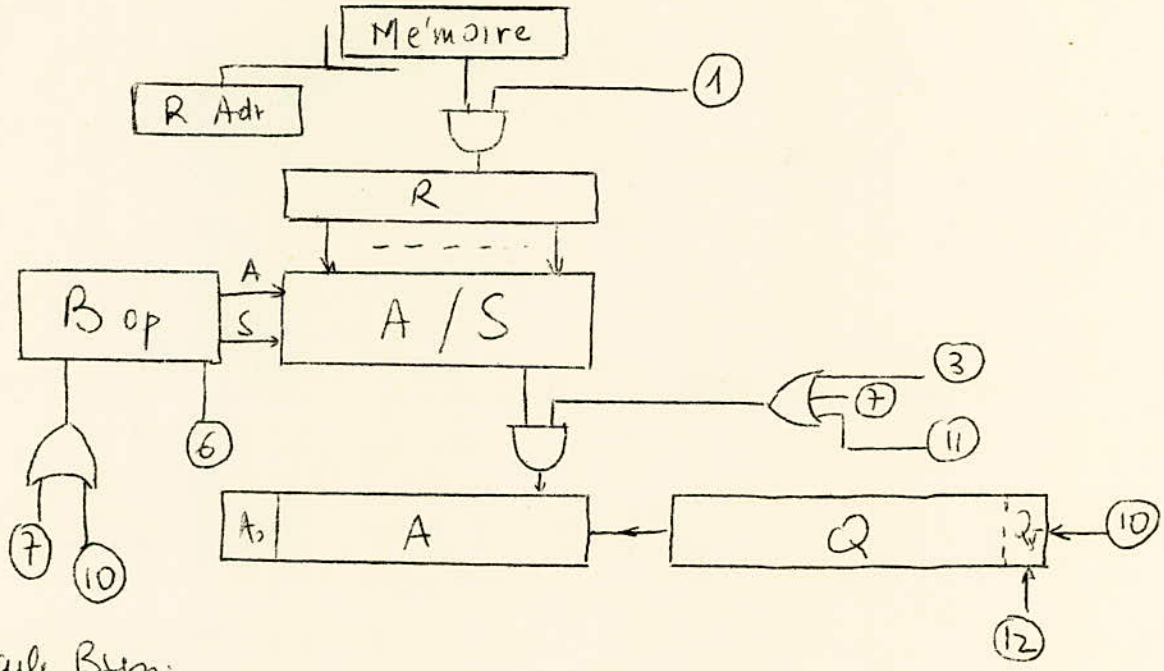
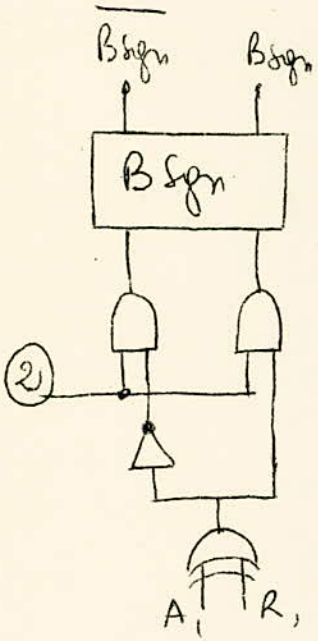
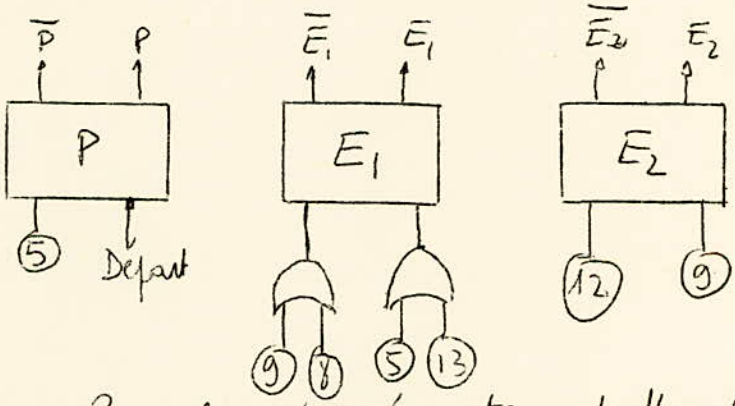


Schéma logique du diviseur :

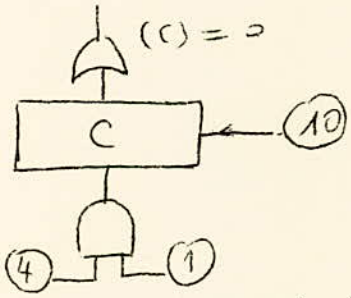


Bascule Bsgn :

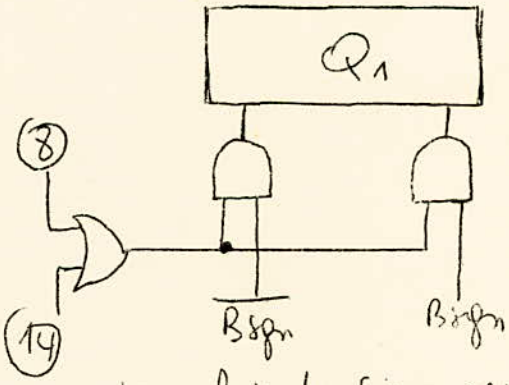




Bascules de préparation et d'exécution



Compteur de décalage



transfert du signe resultat dans le registre Quotient.

ANNEXE 2. U-A et L en Virgule flottante

En virgule flottante, les nombres sont représentés sous forme exponentielle comportant 2 parties : la mantisse et l'exposant.
L'UAL doit réaliser des opérations sur les mantisses et des opérations sur les exposants. Étant donné que l'unité de calcul étudiée peut traiter des nombres inférieurs à 1, on utilise cette unité pour la manipulation des mantisses. Il faut donc un bloc supplémentaire pour le traitement des exposants.

1. Addition soustraction : soient : $A = a.2^{\alpha}$ et $B = b.2^{\beta}$

On commence d'abord par comparer les exposants α et β en calculant $\alpha - \beta = d$. On ajuste le plus faible sur le plus grand en décalant la mantisse de d rangs à droite. On effectue ensuite l'add ou la soust. des mantisses. L'exposant du résultat est l'exposant le plus élevé. Enfin, on normalise la mantisse si cela est nécessaire.

Exp : 1er cas : $\alpha > \beta$ $\quad A \pm B = (a \pm b\,2^{-d})\,2^{\alpha}$

2e cas : $\alpha < \beta$ $\quad A \pm B = (a.2^{-d} \pm b)\,2^{\beta}$

2. Multiplication et division :

Pour effectuer la multiplication de 2 nombres en virgule flottante, on multiplie les mantisses et on fait l'addition des exposants.

$$A \times B = (a.b).\,2^{\alpha+\beta}$$

Pour effectuer la division on divise les mantisses et on fait la soustraction des exposants.

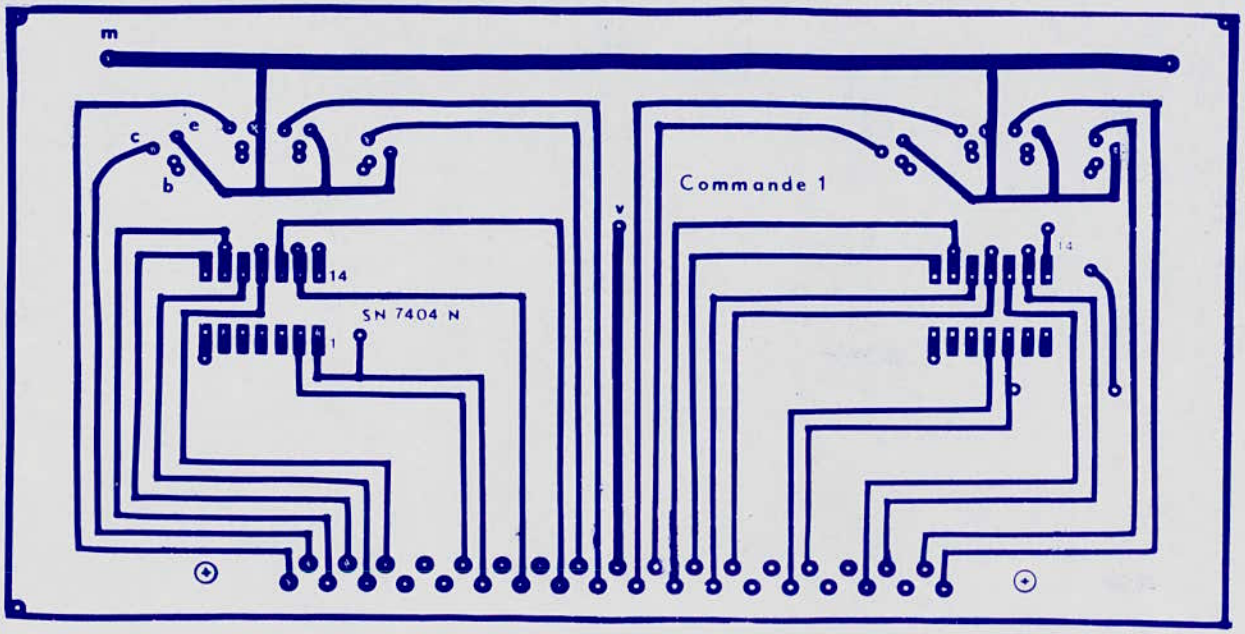
$$\frac{A}{B} = \left(\frac{a}{b}\right).\,2^{\alpha-\beta}$$

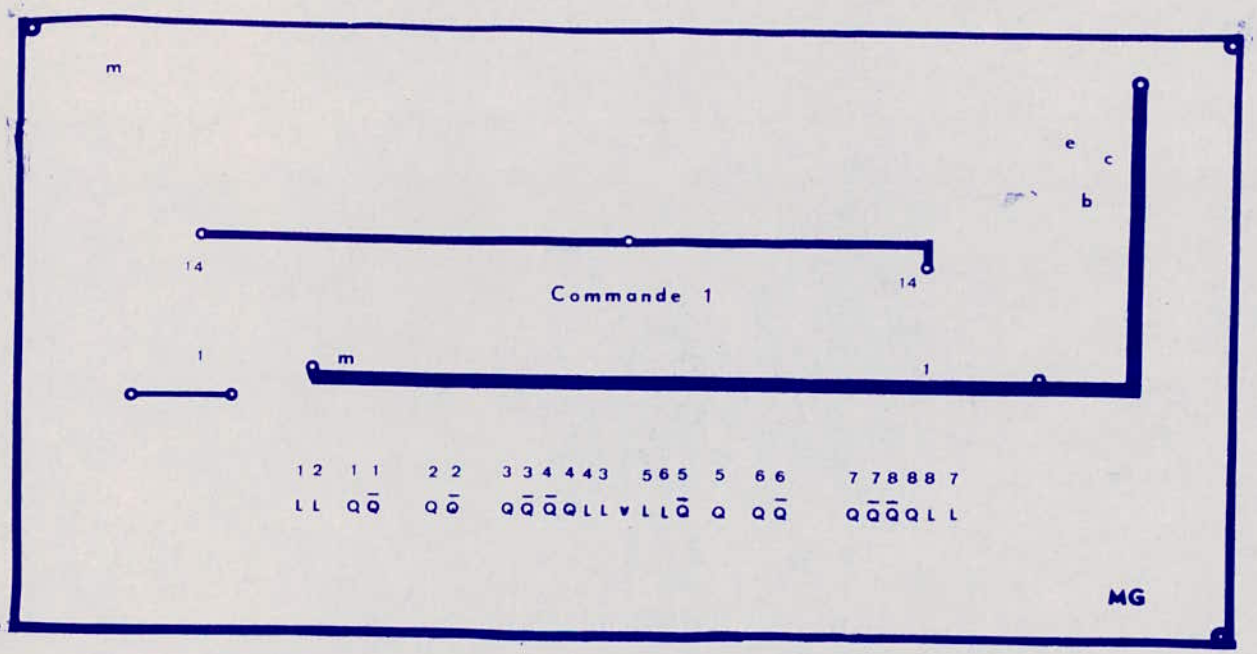
En résumé, l'unité arithmétique et logique (UAL) en virgule flottante comprendra :

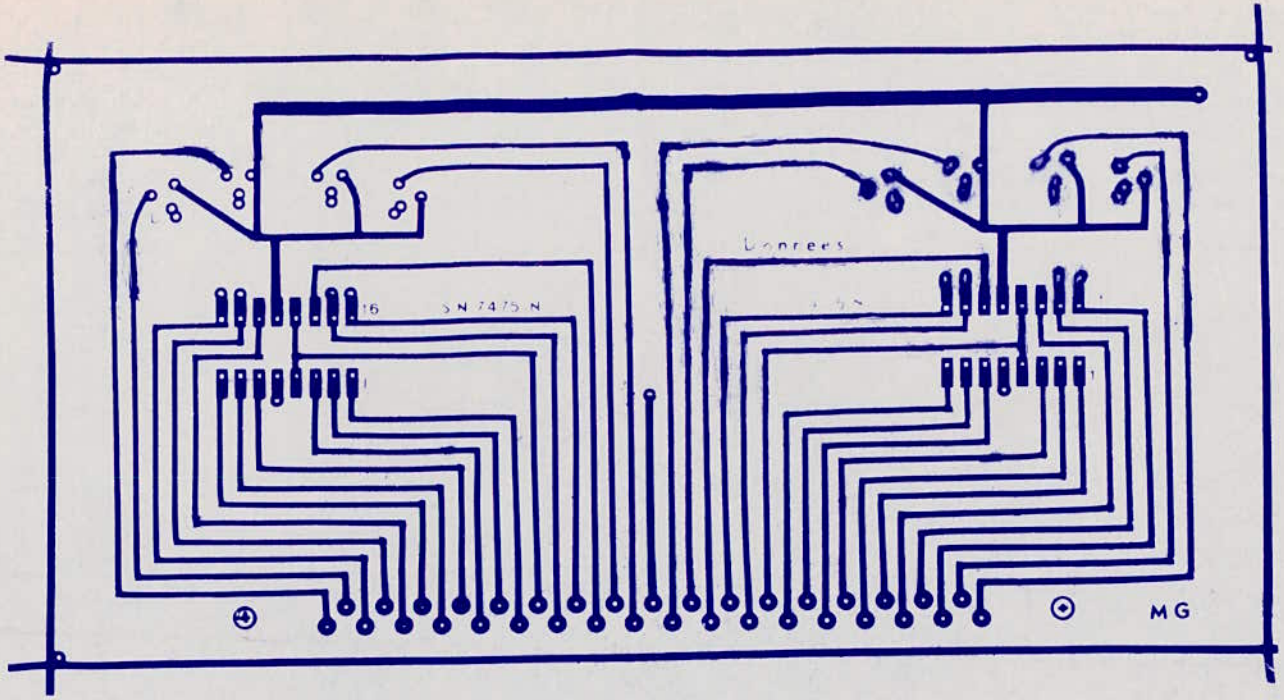
- une unité opérant sur les mantisses
- un add-soust pour les opérations sur les exposants
- un dispositif de comparaison des exposants
- et un dispositif de contrôle de la normalisation du résultat.

BIBLIOGRAPHIE

- A.PROFIT : Structure et technologie des ordinateurs.
- BOULAYE ; Logique et organe des calculatrices numériques.
- R.LYON-CAEN : Circuits logiques intégrés.
- A.PETIT-CLERC :Traité des ordinateurs (tome I et II).
- J.SLOSIAR et N.LANNELONGUE : technologie des composants II. Calculs des bobines.Bobinage.Micromodules.Circuits intégrés.
- P.VOLMER : Calculateurs numériques ;
- H.BOUCHER :Organisation et fonctionnement des machines arithmétiques.
- Revue : Electronique Industrielle (1968-69-70).


m
c
e
b
v
Commande 1
14
SN 7404 N
1
14


m
e
c
b
14
14
Commande 1
1
m
1
12 11 22 3344443 565 5 66 7788887
LL QQ̄ QQ̄ QQ̄Q̄QLLVLLQ̄ Q QQ̄ QQ̄Q̄QLL
MG


SN 7475 N
MG

e c
b

Visualisation des Données

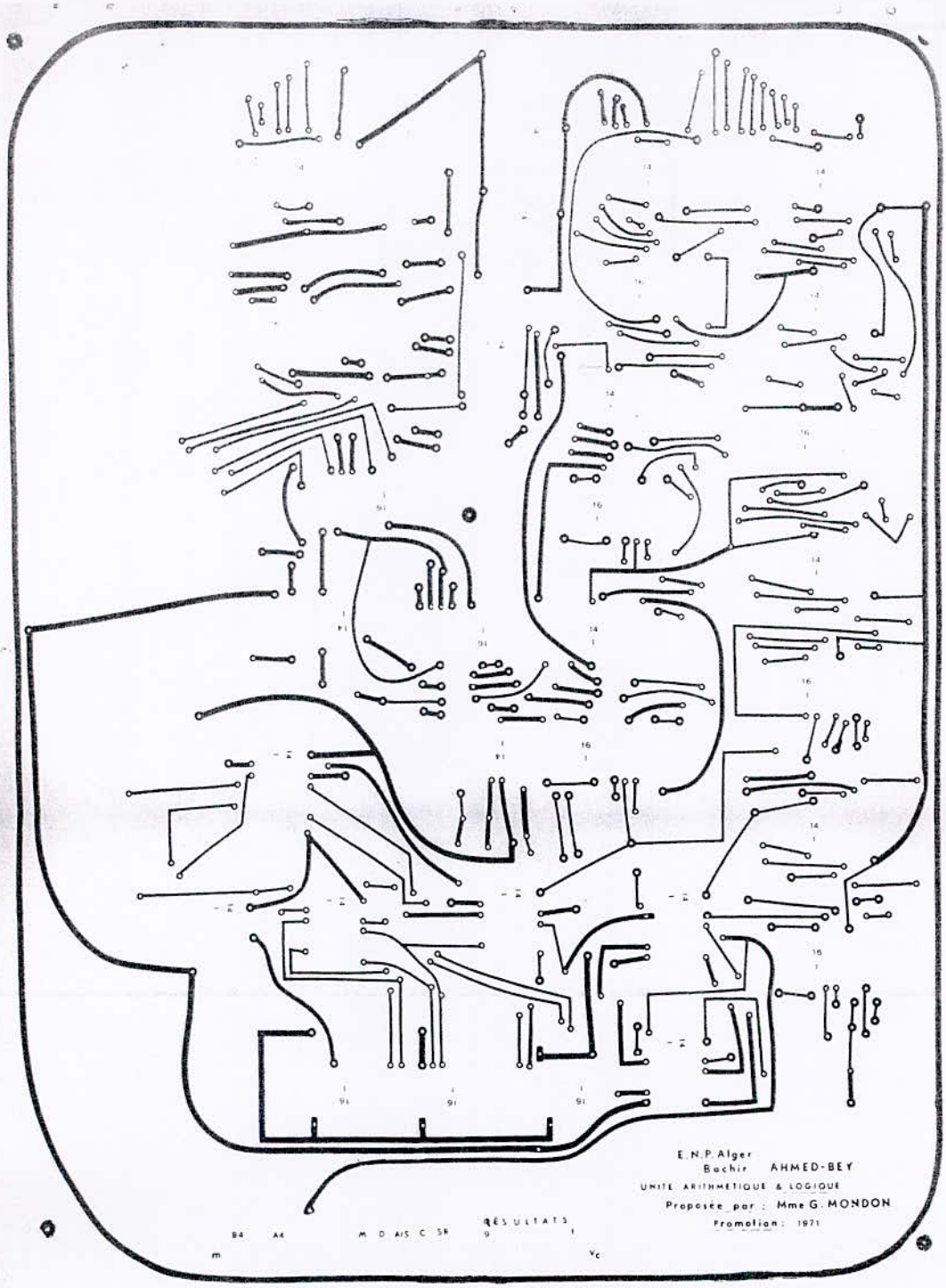
16 16

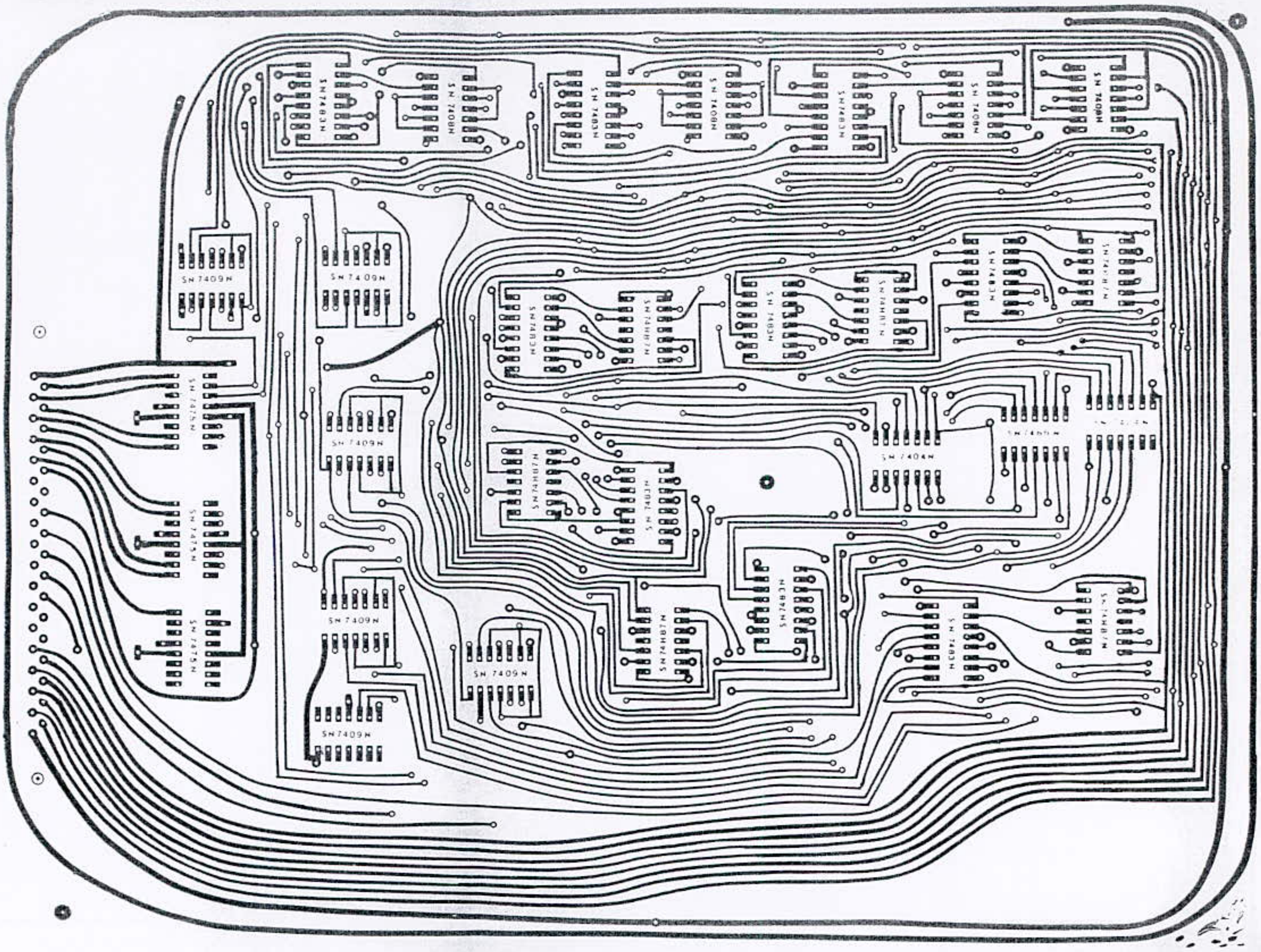
Registres d'entrées

1 1

2 1 1 2 2 1 1 2 3 4 4 3 4 3 4 3 v 6 5 6 5 6 5 5 6 7 8 8 7 7 8 8 7
L L Q Q Q Q D D D D Q H Q Q Q L L L L Q Q Q H Q D D D D Q Q Q Q L L

MG


E.N.P. Alger
Bachir AHMED-BEY
UNITE ARITHMETIQUE & LOGIQUE
Proposée par : Mme G. MONDON
Promotion : 1971
RÉSULTATS


SN 7409N
SN 7409N
SN 7409N
SN 7409N
SN 7409N
SN7409N
SN 7404N