

37/84

2^{es}

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : ELECTRONIQUE

PROJET DE FIN D'ETUDES

SUJET

**TEST DES CIRCUITS LOGIQUES
PAR MICRO_ORDINATEUR
KIT 6802 D5E**

Proposé par :

Mr ARDIBILI

Etudié par :

S. LAISSAOUI

M. BOUCHAHLATA

Dirigé par :

Mr BOUDRAA



PROMOTION : JUIN 84



TEST DES CIRCUITS LOGIQUES
PAR MICRO-ORDINATEUR
KIT 6802 D5 E

~ Dédicaces ~

Æ

Mes parents qui m'ont tout donné; pour leur amour et leur bienveillance.

Mes frères et soeurs en témoignage de notre amour fraternel.

Tous mes amis.

~ Salim ~

Æ

mes parents.

mes frères et soeurs.

mes amis

~ Moussa ~

Remerciements

Vous remercions vivement Monsieur Abdibili, notre promoteur pour son aide et ses conseils qui nous ont été utiles à l'élaboration de ce travail.

Vous formulons l'expression de notre profonde reconnaissance à Monsieur Boudraa pour son précieux concours et ses remarques judicieuses.

Notre sincère gratitude à M^{lles} Kaoua et Fizi ainsi qu'à tous les professeurs de l'école nationale polytechnique qui ont contribué à notre formation.

INTRODUCTION

L'évolution des techniques électroniques requiert à l'heure actuelle des mesures nombreuses et répétitives qu'il serait serait onéreux et fastidieux d'exécuter avec des méthodes manuelles. La création d'instruments de mesure programmables pouvant être commandés à partir de signaux électriques et fournir les résultats sous une forme numérique a permis, en les associant entre eux, de constituer des ensembles automatiques de mesure. Ceux-ci sont capables d'exploiter les résultats sous les formes les plus diverses, d'opérer des comparaisons par rapport à des valeurs de consigne, de réaliser des opérations de tri, etc ...

L'utilisation de ces ensembles offre la possibilité entre autres de minimiser le temps de contrôle en production tout en apportant une plus grande sécurité. Le gain ainsi obtenu représente une fraction importante des coûts, amortissant ces matériels de façon extrêmement rapide.

Rentabilité :

Le coût de la main d'œuvre représente en effet 20 à 30 % des frais de production dans l'industrie électronique. Un système de test automatique en réduisant le temps de contrôle et les frais d'opérateurs spécialisés permet de réduire les coûts des opérations de test et de contrôle dans un rapport supérieur à dix.

Rapidité :

Seul un système automatique permet, avec un petit nombre de postes de travail, d'atteindre les cadences exigées actuellement dans l'industrie.

— Sécurité :

Les tests manuels dépendent trop du facteur humain surtout s'ils sont répétitifs et rapides : (fatigue, non motivation ...). Elles comportent donc un risque d'erreur important. Ce risque est totalement éliminé dans un test automatique.

De ce fait, il nous a été demandé de réaliser dans le cadre de notre projet de fin d'études, un banc d'essai pour lequel des logiciels doivent être élaboré sur microordinateur 6802 DSE dans le but de tester des circuits logiques de la famille TTL.

Dans le premier chapitre nous commençons par donner un aperçu général sur les circuits digitaux.

Dans le second chapitre, nous présentons le microordinateur MEK 6802 DSE de Motorola sur lequel nous travaillons.

Dans le troisième chapitre, on fait l'étude et la réalisation du banc d'essai, ainsi que sa éventuelle amélioration au niveau du Hardware.

Dans le quatrième chapitre, nous expliquons les différentes étapes qui nous ont permis d'élaborer les logiciels d'application et test des circuits TTL.

Enfin, dans le cinquième chapitre, nous présentons l'interprétation de nos résultats.

~ Table des matieres ~

Introduction

Chapitre I : Rappels sur les circuits logiques

I - Caractéristiques des circuits logiques digitaux.....	1
1.1 - Immunité aux bruits.....	1
1.2 - Temps de propagation et de transition.....	1
1.2.1 - Temps de transition.....	1
1.2.2 - Temps de propagation.....	2
1.3 - Consommation.....	2
II - La famille TTL.....	2
2.1 - Règle de charge (TTL).....	3
2.2 - Temps de transition des entrées (TTL).....	3
2.3 - logique à trois états.....	4
2.4 - Comparaison entre les circuits: 74XX, 74HXX, 74SXX, 74LSX...5	5
2.4.1 - Caractéristiques de la TTL.....	6
III - Des familles: ECL - MOS - SOS - I ² L.....	6
3.1 - La famille ECL.....	6
3.2 - la famille MOS.....	7
3.3 - la famille SOS.....	8
3.4 - la famille I ² L.....	8

Chapitre II : Présentation du microordinateur MEK 6802 D5E

I - Organisation générale du Kit D5E.....	9
II - Présentation des différents blocs fonctionnels.....	9
2.1 - le bloc de commande de tout le système: le MPU 6802...9	9
1.2.1 - Généralités.....	10
1.2.2 - Constitution interne.....	10
1.2.3 - jeu d'instruction du MPU.....	10

2.1.4 - Modes d'adressage du MPU.....	11
2.2 - Bloc du Keypad du system PIA, des afficheurs et de l'interface cassette.....	11
2.3 - Bloc à mémoires mortes (ROM).....	12
2.4 - Bloc à mémoires vives (RAM).....	12
2.5 - Le décodeur d'adresses.....	12
2.6 - Bloc interface série.....	13
2.7 - Bloc buffers.....	13
2.8 - Bloc interface parallèle "USER PIA".....	13
2.8.1 - Caractéristiques.....	14

Chapitre III: Etude et réalisation du banc d'essai.

I - Choix des circuits.....	15
1.1 - Présentation des circuits testés.....	15
II - Présentation du matériel (Hardware) choisi.....	20
2.1 - Synoptiques de la carte.....	20
III - Amélioration du testeur (banc d'essai).....	20
3.1 - Amélioration au niveau du matériel (Hardware).....	21
3.2 - Amélioration au niveau du logiciel (software).....	21

Chapitre IV: Logiciels d'application: Test des circuits TTL

I - Principe du test d'un circuit intégré.....	24
1.1 - Porte à une entrée - sortie.....	24
1.2 - Porte à deux entrées - sortie.....	25
1.3 - Etablissement de la table de vérité du C.I.....	26
1.4 - L'organigramme de test.....	27
II - L'organigramme général.....	27
2.1 - Les différents organigrammes et leurs programmes.....	30

III - La procédure du test d'un circuit intégré.....	33
3.1 - organigramme de test du CI : Recherche du code et de l'adresse du programme de test correspondant..	34
3.2 - Pagination.....	37
IV - Les différents circuits intégrés à tester.....	37
4.1 - Brochage et paramètres du test du CI.....	37
4.2 - Représentation de la table de vérité d'un circuit intégré séquentiel.....	50
4.3 - Tableau récapitulatif.....	53
V - Listing du programme.....	55

Chapitre V : Interprétation des résultats

I - Résultats du logiciel.....	71
--------------------------------	----

Conclusion

Bibliographie

CHAPITRE I

**RAPPELS SUR LES CIRCUITS
LOGIQUES**

RAPPELS SUR LES CIRCUITS LOGIQUES

Il existe plusieurs familles de circuits digitaux :
 (TTL , ECL , N-MOS , P-MOS , C-MOS , SOS , I²L) .
 Dans ce chapitre on examinera la famille TTL sur laquelle a porté notre travail . On présentera ses avantages et ses inconvénients .

I - Caractéristiques des circuits intégrés digitaux :

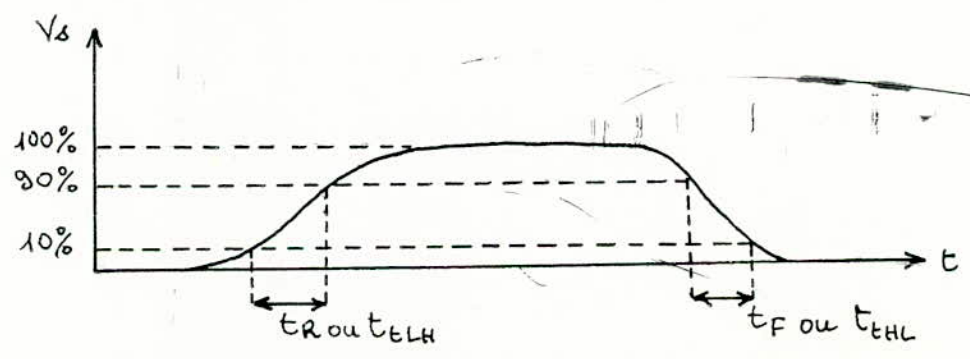
1.1 - Immunité aux bruits :

Les champs magnétiques et électriques parasites peuvent induire des tensions sur les fils de connexions entre les circuits logiques . Ces signaux (bruit) indésirables peuvent perturber les niveaux des signaux logiques .

1.2 - Temps de propagation et de transition :

1.2.1 - Temps de transition

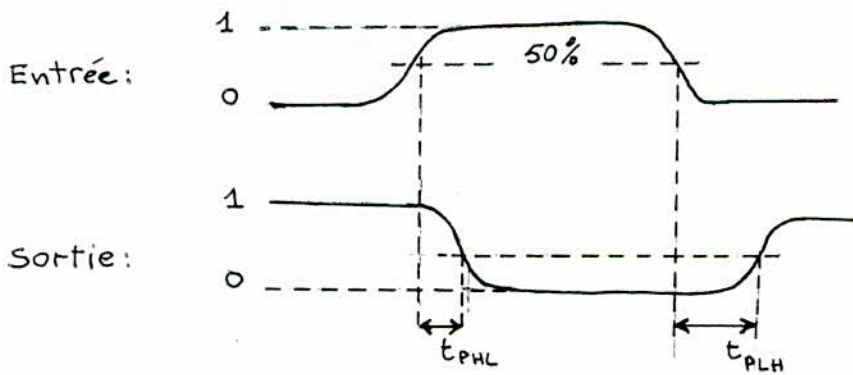
Certains circuits logiques répondent à des niveaux, alors que d'autres ne sont actifs que pour des variations brutales de tension . Pour ces derniers, il est important que les signaux d'entrée soient suffisamment rapides pour qu'ils répondent proprement . C'est pour cette raison qu'on définit le temps de montée (t_R : rise time) et le temps de descente (t_F : fall time) .



L'importance du temps de transition limite la fréquence du changement d'état et donc la rapidité du système.

1.2.2 - Temps de propagation :

Une porte logique met un certain retard pour répondre. Prenons l'exemple de l'inverseur :



En général t_{PHL} et t_{PLH} ne sont pas égaux. Les valeurs des temps de propagation sont utilisés pour définir la vitesse des circuits logiques, et l'importance du temps de propagation limite la rapidité du système.

1.3 - Consommation :

La puissance requise par un circuit intégré est une importante caractéristique qui figure toujours dans les Data Books. Elle est donnée sous plusieurs formes ;

- Puissance dissipée P_D :

plus souvent elle est indirectement spécifiée en termes de courant fournie par l'alimentation du circuit intégré.

II La famille TTL :

Jusqu'à présent la famille de circuits logiques la

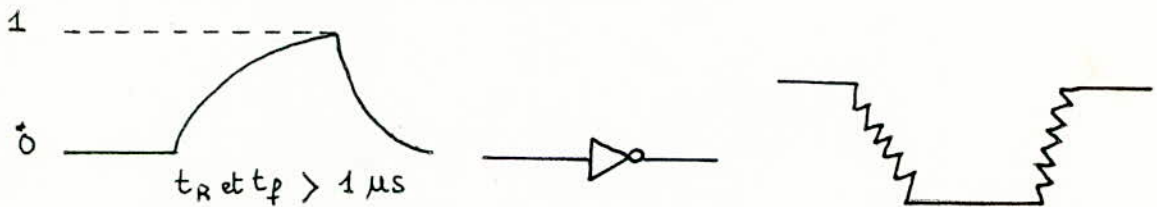
plus utilisée est la famille transistor-transistor-Logic (TTL) prononcée souvent T²L. Elle utilise des transistors bipolaires et rentre dans la catégorie des familles logiques bipolaires.

2.1 - Règle de charge (TTL):

Quand il s'agit de concevoir un système utilisant des circuits intégrés digitaux TTL, il est important de savoir comment déterminer et utiliser la sortie en vue de connaître la capacité de commande de chaque circuit.

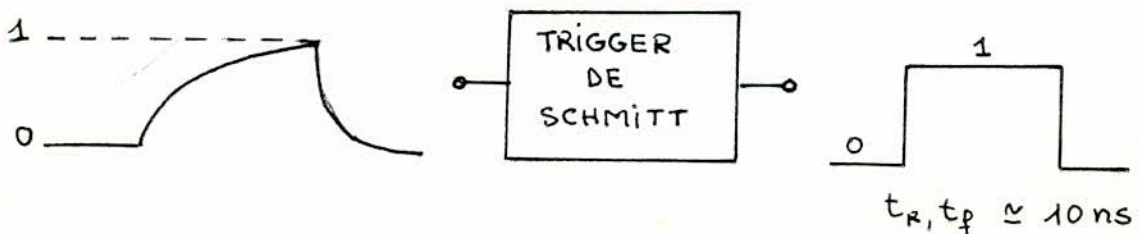
2.2 - Temps de transition des entrées (TTL):

Les signaux d'entrée des circuits TTL doivent avoir des transitions relativement rapides.



Ces oscillations peuvent causer de sérieux problèmes si ce signal de sortie viendrait à attaquer une entrée de FF, OS ou compteur.

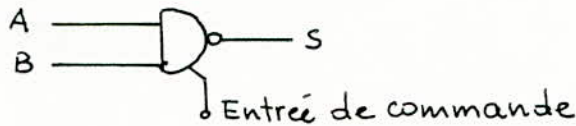
La solution consiste à intercaler le circuit appelé : TRIGGER DE SCHMITT qui produit des temps de transition à la sortie de l'ordre de 10 ns indépendamment des temps de transition à l'entrée, la sortie peut alors attaquer une entrée TTL.



Certains circuits intégrés digitaux TTL sont conçus avec des circuits Trigger de Schmitt à l'intérieur, ce qui entraîne que ces circuits peuvent manipuler des signaux à faibles temps de transition.

2.3 - Logique à trois états :

Un des nouveaux développements en circuits intégrés digitaux est un nouvel état appelé Etat haute impédance (TRI-STATE) ou TSL.



Considérons une porte NAND à 3 états et 2 entrées, on remarque que cette porte possède en plus des 2 entrées conventionnelles A et B, une troisième entrée appelée : "Entrée de commande" (control input) qui sert à mettre la sortie à l'état haute impédance.

Entrée de commande	Sortie	Courants d'entrée en A et B
BAS (non active)	la sortie se comporte comme celle d'un NAND	$I_{IH} = 40 \mu A$ $I_{IL} = -1,6 mA$
HAUT (active)	la sortie est à haute impédance	$I_{IH} = I_{IL} = 40 \mu A$ l'entrée est aussi à l'état haute impédance.

Il faut remarquer que lorsque l'entrée de contrôle est active, l'état haute impédance est établi aussi bien à l'entrée qu'à la sortie. Cette entrée est parfois appelée : Entrée ENABLE ou entrée DISABLE suivant le constructeur.

2.4 - Comparaison entre les : 74LXX, 74HXX, 74SXX, 74LSXX

- Série 74LXX :

Fabriquée pour avoir une faible puissance de dissipation au détriment du temps de propagation. Même circuit que 74XX, valeurs des résistances importantes.

Une porte NAND a : une puissance dissipée = 1 mW

une propagation moyenne = 33 ns

des applications de la série 74LXX : dans les cas où la consommation doit être faible et la vitesse non importante (BF).

- Série 74HXX :

La série 74HXX est rapide, même circuit de base que TTL 74XX mais les valeurs des résistances diminuent :

temps de propagation moyen $\approx 6 \text{ ns}$

Puissance de dissipation moyenne $\approx 23 \text{ mW}$

- Série 74SXX :

Elle possède la plus grande vitesse :

temps de propagation moyen $\approx 3 \text{ ns}$

Puissance de dissipation moyenne $\approx 23 \text{ mW/porte}$

- Série 74LSXX :

Pour diminuer la puissance dissipée, on augmente les valeurs des résistances :

temps de propagation moyen $\approx 9,5 \text{ ns}$

Puissance de dissipation moyenne $\approx 2 \text{ mW/porte}$

Tableau récapitulatif :

Séries	Retard (ns)	Puissance mW	Fréq max MHz	Fan out
74XX	9	10	35	10
74LXX	33	1	3	10
74HXX	6	23	50	10
74SXX	3	23	125	10
74LSXX	9,5	2	45	10

2.4.1 - Caractéristiques de la TTL :

En 1964, Texas instruments introduit le premier produit standard des circuits TTL : 5400 / 7400 series. La seule différence qui existe entre elles, est que la 54XX est produite pour une utilisation militaire.

Exemple :

$$\text{TTL 7400} \left\{ \begin{array}{l} 0^\circ \text{ à } 70^\circ \text{C} \\ V_{cc} \text{ varie de } 4,75\text{V à } 5,25\text{V} \end{array} \right.$$

$$\text{TTL 5400} \left\{ \begin{array}{l} -55^\circ \text{C à } 125^\circ \text{C} \\ V_{cc} \text{ varie de } 4,5\text{V à } 5,5\text{V} \end{array} \right.$$

fan out = 10

temps de propagation moyen = 9 ns

Puissance de dissipation moy ≈ 10 mW

III Les Familles : ECL, MOS, SOS, I²L :

3.1 - la famille ECL :

La famille ECL (Emitter coupled logic) a été développée dans le but d'augmenter considérablement la vitesse de commutation par rapport à celui de la TTL.

d'un de ses inconvénients est la faible marge de bruit, nous obligeant à ne pas l'utiliser dans un environnement industriel important.

- table de comparaison entre l'ECL et la TTL :

Familles	t_p (ns)	P_d (mW)	Marge de bruit (mV)	freq. max de cadence d'horloge (MHz)
7400	9	10	400	35
74L00	33	1	400	3
74H00	6	23	400	50
74S00	3	23	300	125
74LS00	9,5	2	300	45
ECL	2	25	250	200

3.2 - La famille MOS :

La technologie MOS (Métal-oxyde-semi conducteur) tire son nom de la structure de base : les électrodes (source et Drain) métalliques sont isolées par un oxyde (SiO_2) du substrat (semi conducteur). Des transistors obtenus par cette structure sont appelés des MOSFETS. Les circuits intégrés MOS sont fabriqués à base de MOSFETS. Les avantages primordiaux du MOSFET est la relative simplicité de fabrication, dimension réduite, consommation faible, cela représente environ 30% de complexité de fabrication en moins par rapport à celle de la technologie bipolaire. De plus les composants MOS occupent moins de place sur un boîtier, et les circuits intégrés MOS n'utilisent pas d'éléments résistants, ce qui fait gagner énormément d'espace.

Toutes ces raisons ont permis de pouvoir intégrer plus de circuits dans un boîtier, ceci explique pourquoi les MOS l'emportent sur les TTL en LSI. La grande densité d'intégration des CIS MOS entraîne une meilleure sécurité de fonc-

tionnement à cause de la réduction du nombre de connexions externes. Le principal inconvénient est leur relative faible vitesse de fonctionnement comparée à celles des bipolaires.

Les circuits digitaux MOSFET sont répartis en trois catégories :

- P-MOS ou MOSFETS canal P à enrichissement uniquement
- N-MOS ou MOSFETS canal N à enrichissement uniquement
- C-MOS utilise les deux : canal N et canal P.

Les CIS digitaux N-MOS et P-MOS ont une meilleure densité d'intégration que les C-MOS. Ces derniers sont les plus complexes et ont la plus faible densité d'intégration dans la famille MOS, par contre ils ont l'avantage important d'une meilleure vitesse et la plus faible puissance de dissipation.

3.3 - La famille SOS :

La famille SOS est une modification de la famille C-MOS. Au lieu d'utiliser le substrat en silicium, on utilise un substrat isolant, pour réduire les capacités associées à chaque MOSFET.

Les SOS travaillent comme les C-MOS, mais sont plus rapides à cause de la réduction de leur capacité. Cependant le SOS a une procédure de fabrication complexe, ce qui explique le fait qu'il soit cher.

3.4 - La famille I²L :

C'est la plus nouvelle famille logique bipolaire, elle est conçue essentiellement pour rivaliser les MOS dans des applications LSI.

Elle se prête aussi à une haute densité d'intégration. Quand elle opère à des temps lents (100 ns), I²L consomme 5 mW donc moins que n'importe quelle famille logique à vitesse rapide.

CHAPITRE II

PRESENTATION DU MICRO- ORDINATEUR MEK 6802 D5E

PRESENTATION DU MICROORDINATEUR MEK 6802 D5E

I - Organisation générale du kit D5E

Le kit D5E de Motorola est un microordinateur parfaitement adapté à l'initiation des techniques de programmation et les développements des systèmes associés à la famille des microprocesseurs MC 6800.

Le kit D5E est constitué par différents blocs fonctionnels qu'on représentera successivement :

- Le bloc de commande de tout le système : Le MPU 6802
- Le bloc du keypad du système PIA, des afficheurs, de l'interface cassette.
- Le bloc à mémoires mortes (ROM)
- Le bloc à mémoires vives (RAM)
- Le décodeur d'adresse
- Le bloc interface parallèle "USER PIA" : Le MC 6821
- Le bloc buffers
- Le bloc interface série

II - Présentation des différents blocs fonctionnels :

La présentation des différents blocs est donnée par :

2.1 - Le bloc de commande de tout le système : le MPU 6802

Le MC 6802 ayant fait l'objet de plusieurs travaux et thèses d'ingénierat, nous nous limiterons à un bref rappel de ce que nous pensons être indispensable à la compréhension de ce travail.

2.1.1 — Généralités :

Le 6802 est un microprocesseur monolithique de 8 bits dont les caractéristiques générales sont :

- Tension d'alimentation +5V
- Circuit d'horloge intégré
- Mémoire à Ecriture/Lecture intégrée 128 x 8 bits
- 32 octets de cette mémoire peuvent être sauvegardés
- Compatible au niveau du logiciel avec le MC 6800
- Extensible jusqu'à 64 K mots
- Entrées et Sorties compatibles avec la logique TTL
- Mots de 8 bits
- Capacité d'adressage : 64 K octets
- Possibilité d'interruption

2.1.2 — Constitution interne :

Le MC 6802 possède 3 registres de 16 bits et 3 registres de 8 bits accessibles par programme :

- Deux Accumulateurs A et B
- Un registre d'index
- Un Pointeur de Pile
- Un Registre d'état
- Un Compteur programme

2.1.3 — Jeu d'instruction du MPU :

Le MPU a 72 instructions différentes de longueur variable (un à trois octets). Le jeu d'instruction du MPU comprend les instructions suivantes :

- Arithmétique binaire et décimale
- Logique
- Décalages
- Décalages circulaires
- Chargements
- Stockages
- Branchements conditionnels et inconditionnels
- Instructions associées aux interruptions.
- Instructions de manipulation de la pile.

2.1.4 — Modes d'adressage du MPU :

Le MPU a 7 modes d'adressage possibles :

- adressage des accumulateurs
- adressage immédiat
- adressage direct
- adressage étendu
- adressage indexé
- adressage implicite
- adressage relatif

2.2 — Bloc du Keypad du system PIA, des afficheurs et de l'interface cassette :

Le clavier (ou Keypad) permet de faire rentrer des données qui seront prises en compte par le MPU à travers le système PIA, leur affichage s'effectue par l'envoi sur les lignes du port A du code 7 segments du digit à afficher.

Ce même "system PIA" est relié à un interface cassette capable de détecter ou de délivrer des informations binaires pouvant être soit lues par le microprocesseur, soit enregistrées sur cassette (sortie Mic).

2.3 — Bloc à mémoires mortes (ROM) :

C'est un bloc de mémoire à lecture seule dont la "D5 BUG" de 2K octets est indispensable pour la gestion du kit. La "USER ROM" (EPROM "U13") qui peut aller jusqu'à 2K octets est par contre optionnelle.

La prise de connexion J4 nous permet d'utiliser différents types d'EPROM qui peuvent être :

- soit une 2708 de 1K octets, tri-tensions (+5V, -5V, +12V)
- soit une 2716 de 2K octets, monotension (+5V)

2.4 — Bloc à mémoires vives (RAM)

En plus de la RAM interne du 6802, le kit possède deux autres RAM du type volatile qui sont :

- La "SCRATCH RAM" (MC 6810) composée de 128 octets et l' "USER RAM" divisée en boîtiers (U7 et U8) du type MC 2114 de capacité 1Kx4 bits.

La "SCRATCH RAM" est divisée en 3 parties dont 2 sont réservées spécialement pour la ROM moniteur de gestion "D5 BUG" (pour son fonctionnement). La 3^{ème} partie est prévue pour assurer le rôle d'une pile à l'usage de l'utilisateur.

Les deux RAM (U7 et U8) sont constituées par des cases mémoires de 4 bits. En étant placées en parallèle, elles forment le mot complet de 8 bits.

2.5 — Le décodeur d'adresses

C'est grâce à ce circuit logique du type 74LS156 que l'on arrive à simplifier au maximum l'adressage

sur la carte. Son rôle principal est de décoder les lignes d'adresse de poids forts pour former 8 sorties dont chacune sélectionnera un boîtier ou des zones mémoires.

2.6 — Bloc interface série :

Ce bloc en option sur le kit est composé d'un interface série l'"USER ACIA" (U23) et d'un générateur de fréquence (U30). Sur notre carte, il pourra être utilisé pour des applications d'Entrée/Sortie en série :

(télétype, transmission avec MODEM etc ...).

La synchronisation s'effectue par l'horloge (U30) qui génère plusieurs fréquences.

2.7 — Bloc buffers :

Ce sont des amplificateurs des lignes d'adresses et de données. Ils sont utilisés pour l'extension du kit D5 à travers un connecteur de 86 pins.

2.8 — Bloc interface parallèle "USER PIA"

Le PIA MC 6821 (U9) est un circuit d'interface permettant de connecter une unité périphérique au microprocesseur MC 6802. Ce circuit interface le MPU avec les périphériques par 2 bus de 8 lignes de données bi-directionnelles vers la périphérie et quatre lignes de commande. Pour réaliser l'interface avec de nombreux périphériques, il n'est pas nécessaire d'ajouter de circuits supplémentaires.

La configuration fonctionnelle du PIA est programmée par le MPU pendant la mise au point initial du système. Chacune des 16 lignes de donnée vers la périphérie peut être programmée pour être utilisée soit en entrée, soit en sortie. Chacune des 4 lignes de commande peut être programmée pour un des modes de fonctionnement possibles. Ceci permet une grande souplesse dans l'utilisation du PIA.

2.8.1 — Caractéristiques:

- Bus de données 8 bits bi-directionnel vers le MPU.
- 2 lignes d'interruption vers le MPU.
- 2 bus 8 bits bi-directionnels vers la périphérie.
- 2 registres de commande programmables.
- 2 registres de sens de transfert des données: (commande du sens de transfert de chaque ligne de données vers la périphérie).
- 4 lignes de contrôle de la périphérie: 2 en entrée (entrées d'interruption) et 2 soit en entrée, soit en sortie (sorties de commande).
- Système de synchronisation de demande - acquittement ("handshake") pour les opérations d'entrée et de sortie.
- Lignes vers la périphérie à 3 états et pouvant commander des transistors.
- Système d'interruption commandé par programme avec possibilité de masquage (d'inhibition).
- Possibilité de commander des circuits CMOS par les sorties d'un des bus de 8 lignes vers la périphérie
- Possibilité de commander deux charges TTL
- Compatible TTL
- Fonctionnement statique

CHAPITRE III

**ETUDE ET REALISATION DU BANC
D'ESSAI**

ETUDE ET REALISATION DU BANC D'ESSAI

Introduction :

Avant d'utiliser un circuit intégré quelconque il est important de savoir si celui-ci fonctionne. Tester un circuit intégré à plusieurs entrées manuellement et avec des moyens classiques est souvent une opération délicate, ennuyeuse et parfois très lente. Ceci nous a poussé à réaliser un testeur rapide permettant le contrôle des circuits intégrés TTL (famille 74xx) les plus utilisés au niveau de notre labo. Le testeur réalisé est adaptable sur le KIT MEK 05 de Motorola et va nous permettre de tester un large éventail de CI TTL de la famille 74xx. Une bonne souplesse d'utilisation est conférée à l'ensemble du fait de sa programmation.

I. Choix des circuits :

Notre choix a porté sur la famille 74xx (TTL) car elle est la plus utilisée. Ce sont des circuits de 14 ou 16 pins qui représentent un fort pourcentage dans cette famille.

Parmi ces circuits, on remarque deux types différents :

- Pour les circuits à 14 broches :
 - ceux alimentés en pin 7 (GND) et pin 14 (Vcc)
 - ceux alimentés en pin 4 (GND) et pin 11 (Vcc)

- Pour les circuits à 16 broches :
 - ceux alimentés en pin 8 (GND) et pin 16 (Vcc)
 - ceux alimentés en pin 5 (GND) et pin 13 (Vcc)

1.1 - Présentation des circuits testés :

Nous citons ici, la liste des différents circuits

intégrés pouvant être contrôlés par notre testeur :

74 00	Quadruple 2-input positive - NAND Gates.
74 01	Quadruple 2-input positive - NAND Gates with open-collector out puts.
74 02	Quadruple 2-input positive - NOR Gates.
74 03	Quadruple 2-input positive - NAND Gates with open-collector out puts.
74 04	Hex inverters positive logic.
74 05	Hex inverters with open-collector out puts.
74 06	Hex inverters Buffers/Drivers with open collector high voltage out puts.
74 07	Hex Buffers/Drivers with open collector high voltage out puts.
74 08	Quadruple 2-input positive - AND Gates
74 09	Quadruple 2-input positive - AND Gates with collector out puts.
74 10	Triple 3-input positive - NAND Gates
74 11	Triple 3-input positive - AND Gates
74 12	Triple 3-input positive - NAND Gates with open collector out puts.
74 13	Dual 4-input positive - NAND Schmitt triggers.
74 14	Hex Schmitt triggers inverters
74 15	Triple 3-input positive - AND Gates with open collector out puts.
74 16	Hex inverter Buffers/Drivers with open collector high voltage out puts.
74 17	Hex Buffers/Drivers with open collector high voltage out puts.
74 20	Dual 4-input positive - NAND Gates.
74 21	Dual 4-input positive - AND Gates
74 22	Dual 4-input positive - NAND Gates with open collector out puts.

74 26	Quadruple 2-input high voltage interface positive NAND Gates.
74 27	Triple 3-input positive - NOR Gates
74 28	Quadruple 2-input positive - NOR Gates
74 32	Quadruple 2-input positive - OR Gates
74 33	Quadruple 2-input positive - NOR Buffers with open collector out puts.
74 37	Quadruple 2-input positive - NAND Buffers
74 38	Quadruple 2-input positive - NAND Buffers with open collector out puts.
74 74	Flips - Flops
74 80	Dual-in-line package
74 86	Quadruple 2-input exclusive - OR Gates
74 107	Dual J-K master-slave Flip-Flops with clear
74 109	Flip-Flops Dual J-K positive - EDGE Triggered
74 110	AND-Gated J-K positive - EDGE Triggered Flip-Flops
74 111	Dual J-K Master-Slave Flip-Flops with Data Rackout
74 125	Quadruple Bus Buffer Gates with three state out puts
74 126	Quadruple Bus Buffer Gates with three state out puts
74 128	Quadruple 2-input positive NOR
74 132	Quadruple 2-input positive - NAND Schmitt Triggers
74 135	Quadruple exclusive OR/NOR Gates
74 136	Quadruple 2-input exclusive - OR Gates with open collector out puts.

En résumé on peut classer les circuits intégrés à tester de la façon suivantes :

- circuits à une seule entrée (inverseur - non inverseur).
- circuits à 2 entrées (fonctions OR, NOR, AND, NAND, XOR).
- circuits à 3 et 4 entrées (fonctions OR, NOR, AND, NAND).
- Les Buffers (Registre tampon)
- Les Flip-Flops
- Les compteurs

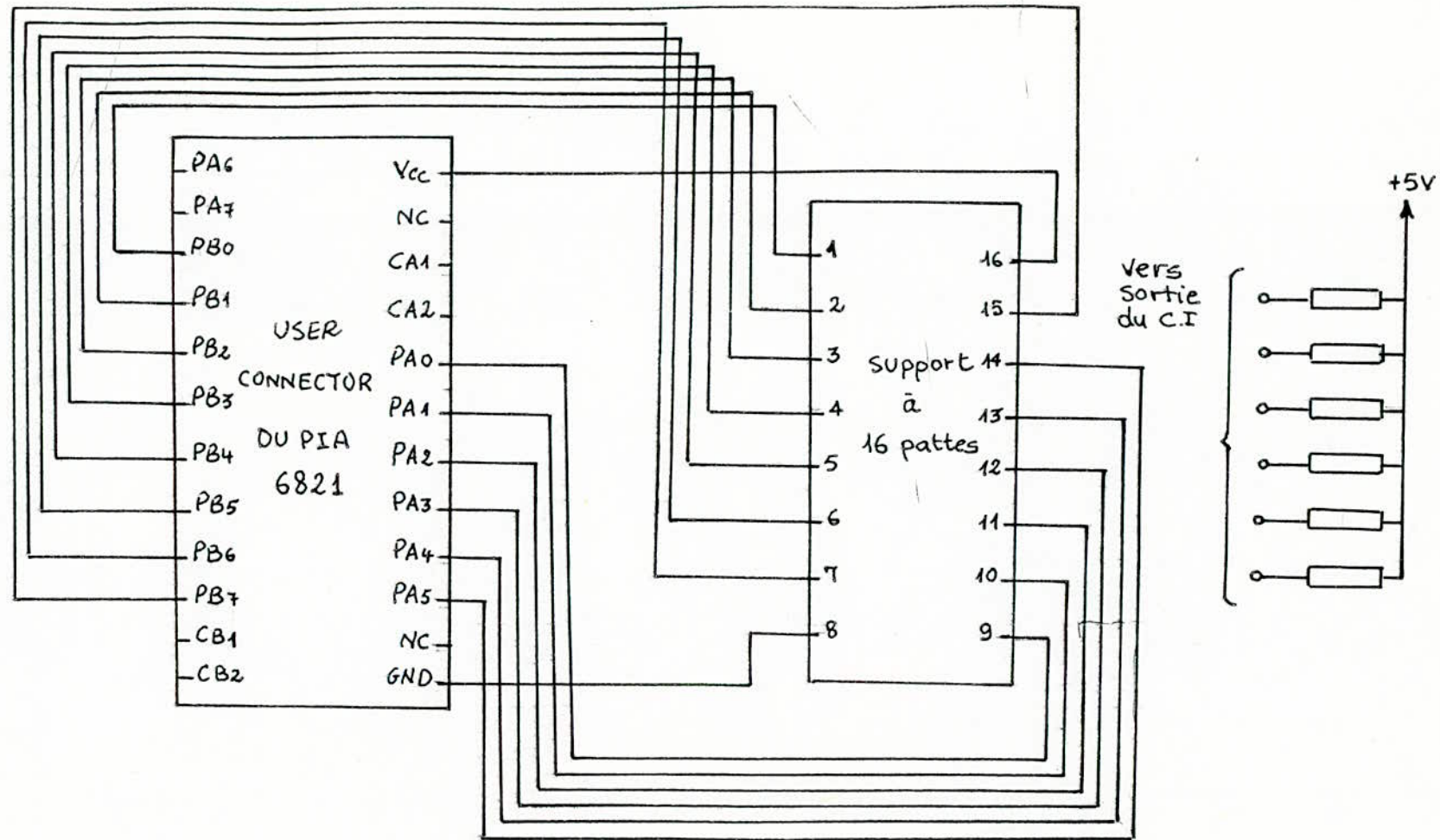


Fig.2. INTERFACE POUR C.I à 16 BROCHES

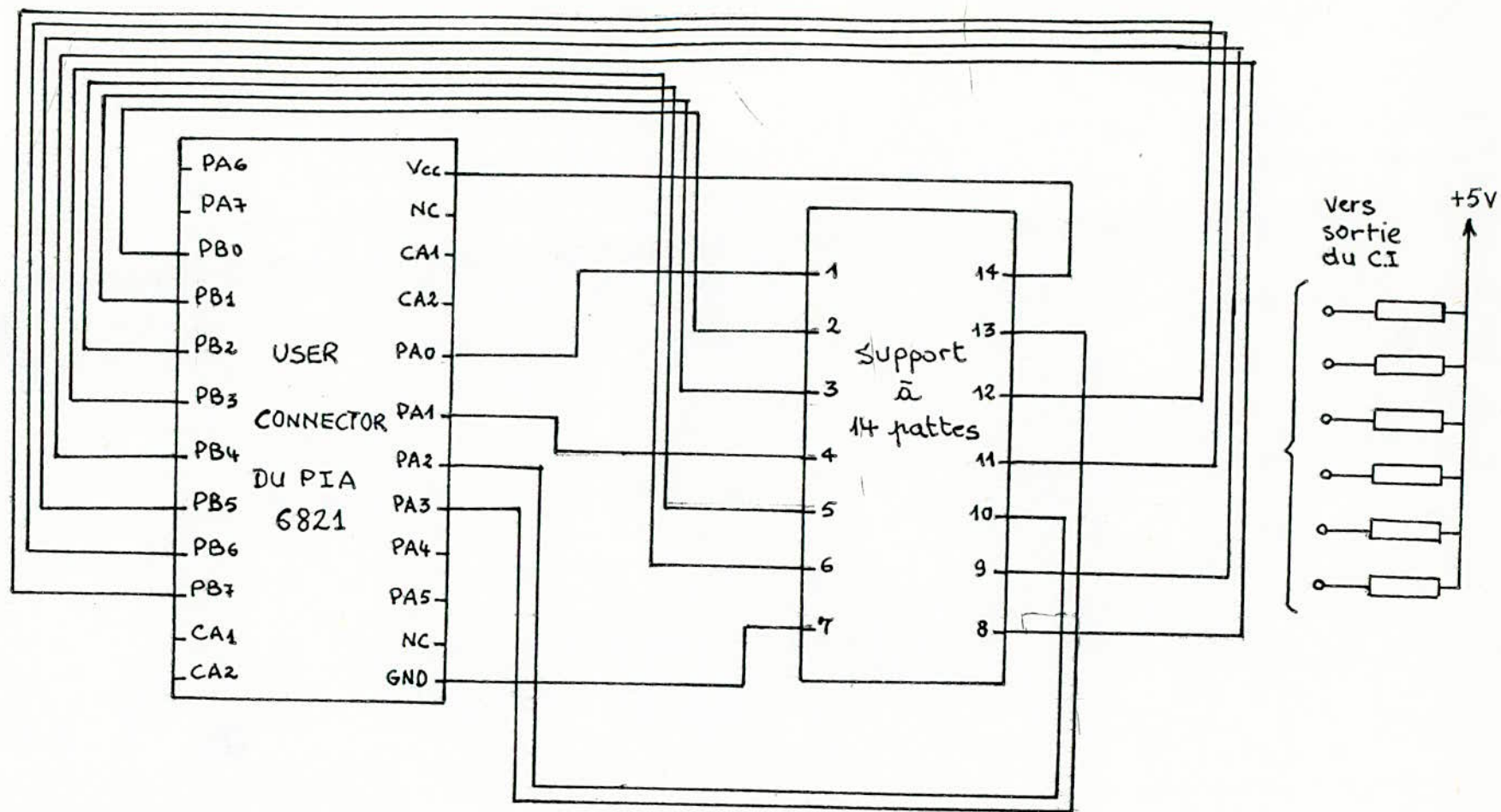


Fig.1. INTERFACE POUR CI à 14 BROCHES

II - Présentation du Matériel (Hardware) choisi :

Le Hardware qu'on a réalisé est très simple, car il prend en considération le minimum de matériel.

2.1 - Synoptiques de la carte :

- brochage adopté dans le cas d'un CI à 14 pattes :
(voir figure - 1 -)

Remarque: on peut dégager un code qui sera la base de programmation du PIA; qui est le suivant :

code:

PA ₃	PB ₇	PB ₆	PA ₂	PB ₅	PB ₄	PB ₃	PB ₂	PA ₁	PB ₁	PB ₀	PA ₀	PIA
13	12	11	10	9	8	6	5	4	3	2	1	CI

- brochage adopté dans le cas d'un CI à 16 pattes :
(voir figure - 2 -)

code:

PB ₇	PA ₅	PA ₄	PA ₃	PA ₂	PA ₁	PA ₀	PB ₆	PB ₅	PB ₄	PB ₃	PB ₂	PB ₁	PB ₀	PIA
15	14	13	12	11	10	9	7	6	5	4	3	2	1	CI

patte 16 → Vcc
patte 8 → GND

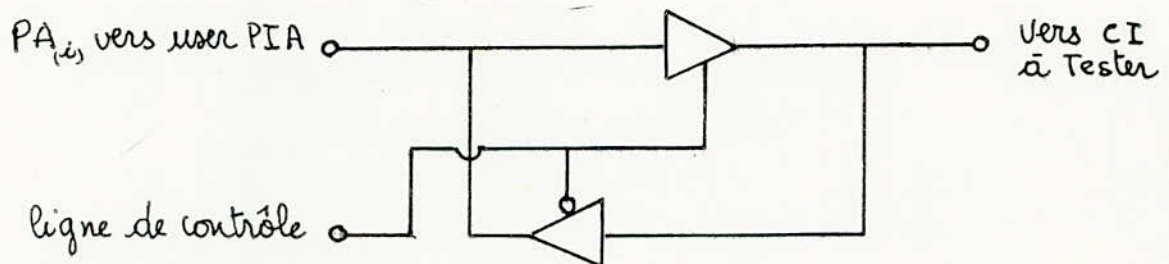
III - Amélioration du testeur (banc d'essai) :

Du fait de la simplicité de notre banc d'essai au niveau du Hardware, nous avons pensé de proposer une idée sur l'amélioration de notre testeur qui sera son extension par quelques modifications au niveau du matériel (Hardware) et du Logiciel (Software).

3.1 — amélioration au niveau Matériel (Hardware) :

On propose une carte comportant :

- quatre support pour CI, chacun correspondant à une famille parmi les cas suivants :
 - * CI à 14 broches :
 - { ceux alimentés en pin 7 (GND) et pin 14 (Vcc)
 - { ceux alimentés en pin 4 (GND) et pin 11 (Vcc)
 - * CI à 16 broches :
 - { ceux alimentés en pin 8 (GND) et pin 16 (Vcc)
 - { ceux alimentés en pin 5 (Vcc) et pin 13 (GND)
- quatre LED, chacune correspondant à un support.
- un interface assurant le dialogue entre le KIT et le CI à tester, représenté de la manière suivante :



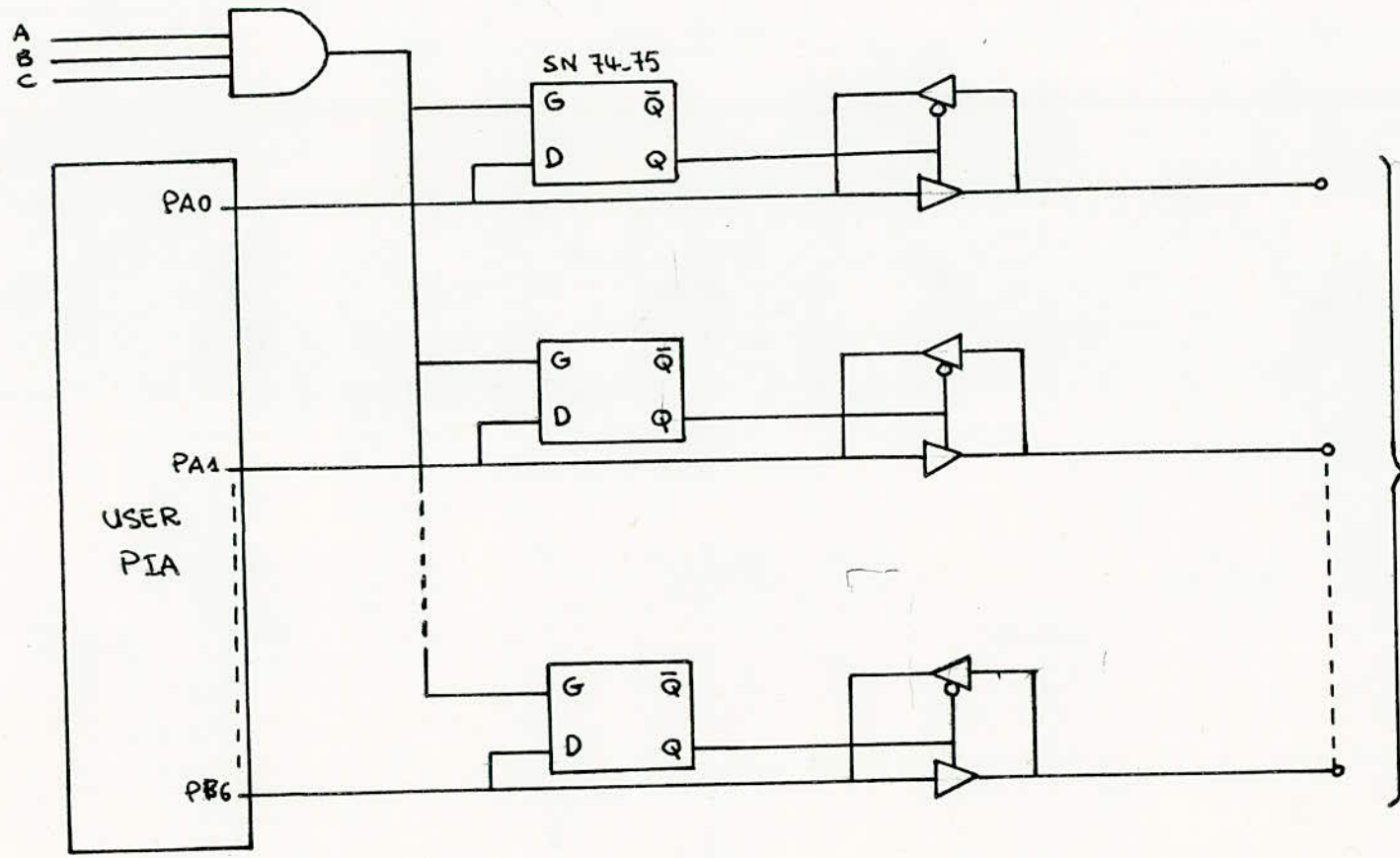
Cet interface assure la liaison CI - user PIA, dans le sens désiré suivant le pin du CI (Entrée ou sortie); à l'aide de la ligne de contrôle.

Ainsi à l'aide d'un ensemble de petit interface de ce type on réalise un interface global capable d'assurer une bonne liaison entre le PIA et le CI. [voir figure - 3 -]

3.2 — amélioration au niveau du Logiciel (software) :

On propose de faire les organigrammes suivants :

- l'organigramme pour l'allumage des LED correspondantes à chaque support où doit se poser le CI à tester.



Vers Circuit intégré à contrôler

Fig.3. SYNOPTIQUE D'INTERFAÇAGE

- l'organigramme d'affichage du numéro du CI.
- faire une routine de conversion Décimal - Hexadécimal, du fait que le kit travaille en Hexa alors que le code du CI à tester est donné en décimal.

En conclusion; la partie Hardware (interface) utilisée nous permettra d'éliminer la partie Software qui a été réservé pour la programmation du PIA en (E/S); on assure ainsi un gain du point de vue position mémoire.

Enfin, dans notre proposition on a donné les principales mesures qu'il faut prendre en considération si une étude dans ce sens peut être développée.

CHAPITRE IV

**LOGICIELS D'APPLICATION :
TEST DES CIRCUITS TTL**

LOGICIELS D'APPLICATION = TEST DES CIRCUITS TTL

I - Principe du test d'un circuit intégré :

La programmation du PIA en Entrée/Sortie est basée sur " le code de brochage " du user PIA avec le support du circuit intégré.

Pour expliquer la procédure de programmation utilisée, prenons l'exemple d'un inverseur ou un non-inverseur (Porte à 1 entrée - 1 sortie). Après une petite recherche sur les circuits intégrés possédant ces fonctions, on a adopté une règle générale pour que la majorité des circuits logiques répondent à la configuration suivante :

1.1 - Porte à 1 Entrée - 1 sortie :

13	12	11	10	9	8	6	5	4	3	2	1	Pattes C.I
E	S	E	S	E	S	S	E	S	E	S	E	E/S du C.I
S	E	S	E	S	E	E	S	E	S	E	S	PIA

- Configuration des registres de direction de transfert du user PIA; Code fixe :

.	.	.	.	13	10	4	1	Pattes CI
1	1	1	1	S	E	E	S	PIA

-DDRA-

12	11	9	8	6	5	3	2	Pattes CI
E	S	S	E	E	S	S	E	PIA

-DDR B-

On prendra pour la programmation du PIA, les affectations suivantes :

- Entrée PIA (E), affectée par "0" → Lecture
- Sortie PIA (S), affectée par "1" → Ecriture

La configuration des registres de direction de transfert du user PIA sera donc la même pour tous les circuits intégrés qui répondent au même code, on obtient ainsi pour:

- DDRA -

.	.	.	.	13	10	4	1	C.I
1	1	1	1	1	0	0	1	PIA

"F9"

- DDRB -

12	11	9	8	6	5	3	2	C.I
0	1	1	0	0	1	1	0	PIA

"66"

1.2 - Porte à 2 entrées - 1 sortie :

on procède de la même manière que précédemment, mais dans ce cas, on a deux configurations :

1	2	3	4	5	6	8	9	10	11	12	13	C.I
E	E	S	E	E	S	S	E	E	S	E	E	E/S-C.I
S	S	E	S	S	E	E	S	S	E	S	S	PIA

Première Configuration

1	2	3	4	5	6	8	9	10	11	12	13	C.I
S	E	E	S	E	E	E	E	S	E	E	S	E/S-C.I
E	S	S	E	S	S	S	S	E	S	S	E	PIA

Deuxième configuration

- Configuration des registres de direction de transfert du user PIA:

.	.	.	.	13	10	4	1	C.I	DDRA
1	1	1	1	S	S	S	S	PIA	FF

12	11	9	8	6	5	3	2	C.I	DDRBB
S	E	S	E	E	S	E	S	PIA	A5

Première Configuration

.	.	.	.	13	10	4	1	C.I	DDRA
1	1	1	1	E	E	E	E	PIA	FO

12	11	9	8	6	5	3	2	C.I	DDRBB
S	S	S	S	S	S	S	S	PIA	FF

Deuxième Configuration

La méthode a été appliquée pour les autres circuits intégrés à 3 entrées; 4 entrées...

Un regroupement selon le code entraîne donc une même programmation (Entrée/Sortie) du PIA et cela conduit à un gain en position mémoire.

1.3 - Etablissement de la table de vérité du C.I :

Une fois le sens des lignes est défini, on envoie les données (table de vérité) vers le C.I qu'on veut tester et on lie sa réponse. Pour éviter de faire deux tables (une pour l'écriture des données, l'autre pour comparer les mots lus), on fait une seule tel que chaque mot envoyé vers le C.I contiendra en même temps les données d'écriture-lecture. Les réponses seront comparées, une à une aux octets correspondants aux données déjà envoyées, pour nous informer sur l'état du circuit intégré. Autrement dit, le mot lu sera comparé directement au mot envoyé : s'ils ne sont pas identiques, le C.I est alors défectueux.

Examinons le cas du "C.I 7404"; Pour ce circuit on doit envoyer sur les deux ports du user PIA les mots suivants:

PA ₇	PA ₆	PA ₅	PA ₄	PA ₃	PA ₂	PA ₁	PA ₀	
1	1	1	1	0	1	1	0	F6
1	1	1	1	1	0	0	1	F9

- PORT A -

PB ₇	PB ₆	PB ₅	PB ₄	PB ₃	PB ₂	PB ₁	PB ₀	
1	0	0	1	1	0	0	1	99
0	1	1	0	0	1	1	0	66

- PORT B -

On envoie successivement la donnée F6 sur le port A, puis 99 sur le port B, on lie ensuite la réponse du C.I et on la compare de nouveau à F6 et 99 ... etc.

Les mots seront de cette façon envoyés deux à deux, ce qui nous donne les "double mots" suivants:

F6
99
F9
66

Pour notre programme, on aura besoin de la longueur de la table de ce double-mot, qui est égale à 2 pour l'exemple du "C.I 7404". Cette table sera implantée à partir d'une certaine adresse dans la mémoire du programme.

1.4 - L'organigramme de Test :

Le programme établi est le même pour la majorité des circuits intégrés qu'on doit tester. Il suffit de préciser pour chaque circuit les paramètres suivants :

- le sens de transfert de données (progr E/S du PIA).
- la longueur de la table des double-mots de test (table de vérité).
- l'adresse du début de cette table dans la mémoire du programme.
- "Ecriture" plus "lecture" sur le user PIA.

- Remarque: La terminologie "Ecriture" plus "Lecture" sur le user PIA veut dire: Voir de quelle façon sera adopté l'Ecriture/Lecture parmi les cas suivants:

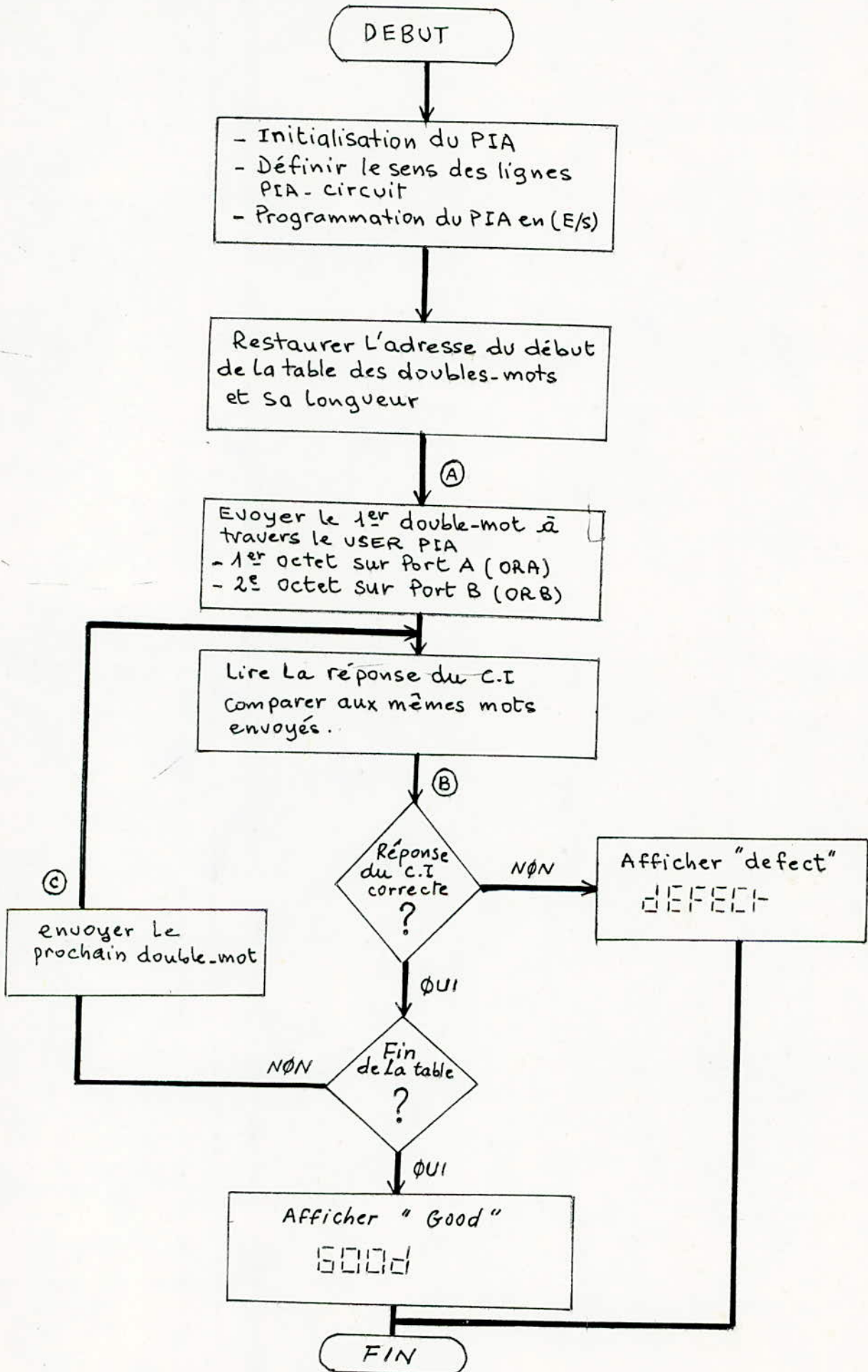
- Ecriture sur port A et Lecture sur port B, ou bien l'inverse.
- E/L sur port A et E/L sur port B

II - L'organigramme général :

Cet organigramme est dit général parce qu'il est commun à tous les circuits intégrés à tester, en ajoutant une précision sur la façon adoptée à l'Ecriture/lecture sur le user PIA.

De ce fait, on a remarqué que la façon attribuée à cette Ecriture/Lecture sur PIA figure parmi les différents cas suivants:

- Pour les C.I à 14 pattes, les cas rencontrés dans notre travail sont :



```

CLRA
STAA M-
JSR SUB 1
LDAA #$ XX
STAA PIA DDRA
LDAA #$ XX
STAA PIA DDRB
LDAA #$ 04

```

} Effacement de la position mémoire où se trouve l'instruction RTS.

- Initialisation du PIA
- Définition du sens des lignes du PIA - CI
- Programmation du PIA en E/S.

```

JSR SUB 1
LDAB #$ XX

```

} chargement de la longueur de la table de vérité du CI.

```

RTS
LDX #$ M

```

} Restaurer l'adresse du début de la table des double-mots

Suite 1

```

LDAA 0,X
STAA PIA ORA
LDAA 1,X
STAA PIA ORB
LDAA PIA ORA
CMPA 0,X
BNE Suite 2

```

} Lecture et Ecriture des données à travers le PIA

```

LDAA PIA ORB
CMPA 1,X
BNE suite 2

```

```

JSR SUB 2
BNE suite 1
JMP SUB 3

```

} Appel de la subroutine de visualisation du mot "Good"

Suite 2

```

JMP SUB 4

```

} Appel de la subroutine de visualisation du mot "DEFECT"

- Ecriture sur ORA et Ecriture/Lecture sur ORB.
- Lecture sur ORA et Ecriture/Lecture sur ORB.
- Lecture sur ORA et Ecriture sur ORB
- Ecriture / Lecture sur ORA et Ecriture/Lecture sur ORB.

- Pour les C.I à 16 pattes, on a rencontré les cas suivants:

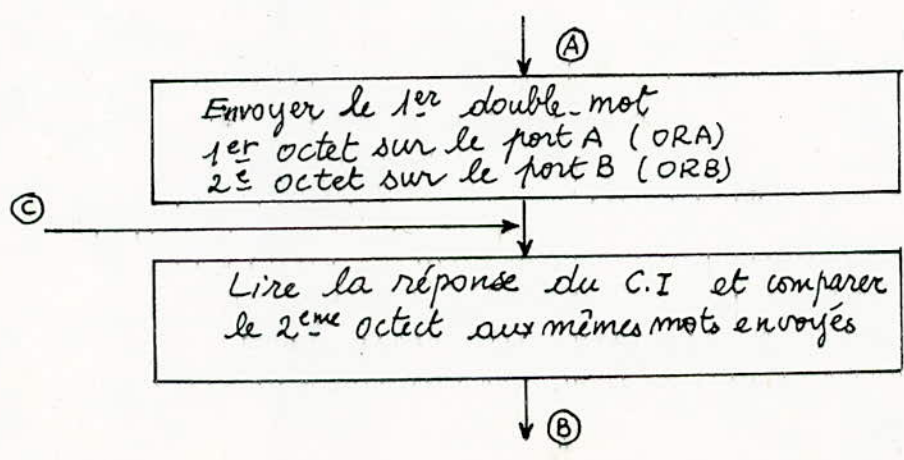
- Ecriture/Lecture sur ORA et Ecriture/Lecture sur ORB.
- Lecture sur ORA et Ecriture/Lecture sur ORB
- Lecture sur ORA et Ecriture sur ORB.

Pour ces 2 derniers cas, le premier mot du "double-mot" ne sera pas envoyé mais simplement affecté d'une comparaison après l'envoi sur le port B. (le 1^{er} octet est pour la lecture simplement, donc une légère modification au niveau du programme.

2.1 - Les différents organigrammes et leurs programmes:

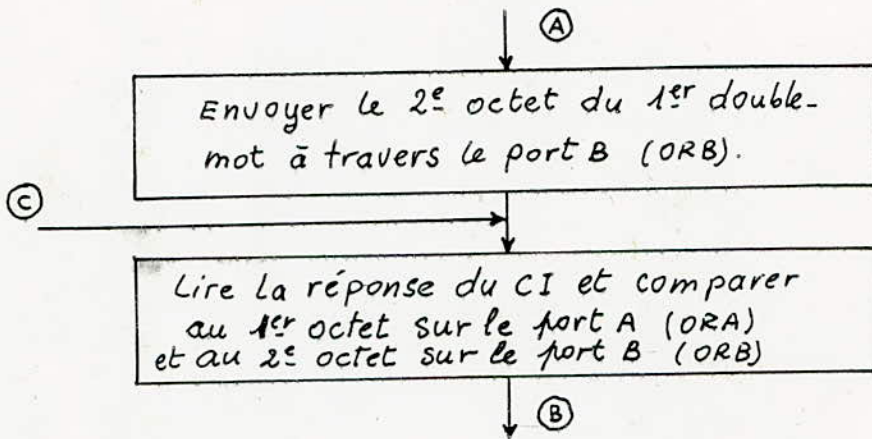
La seule différence qui existe entre les organigrammes des différentes familles de C.I, se trouve au niveau des parties (A) et (B) de notre organigramme général. Toutes les autres parties sont identiques.

- 1^{er} cas : Ecriture sur ORA et Ecriture/Lecture sur ORB:



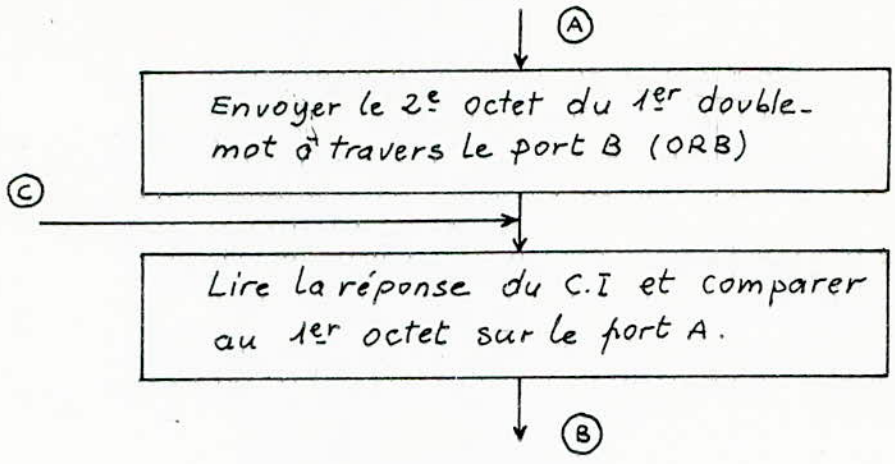
LDAA 0,X	}	Envoi du 1 ^{er} octet sur ORA
STAA PIA ORA		
LDAA 1,X	}	Ecriture/Lecture du 2 ^e octet sur ORB
STAA PIA ORB		
LDAA PIA ORB		
CMPA 1,X	}	Comparaison et test
BNE suite 2		

- 2^e cas: Lecture sur ORA et Ecriture/Lecture ORB:



LDAA 1,X	}	- Ecriture/Lecture du 2 ^e octet - comparaison et test pour le 2 ^{eme} octet.
STAA PIA ORB		
LDAA PIA ORB		
CMPA 1,X		
BNE suite 2	}	- Lecture du 1 ^{er} octet - comparaison et test pour le 1 ^{er} octet
LDAA PIA ORA		
CMPA 0,X		
BNE suite 2		

- 3^e cas: Lecture sur ORA et Ecriture sur ORB:

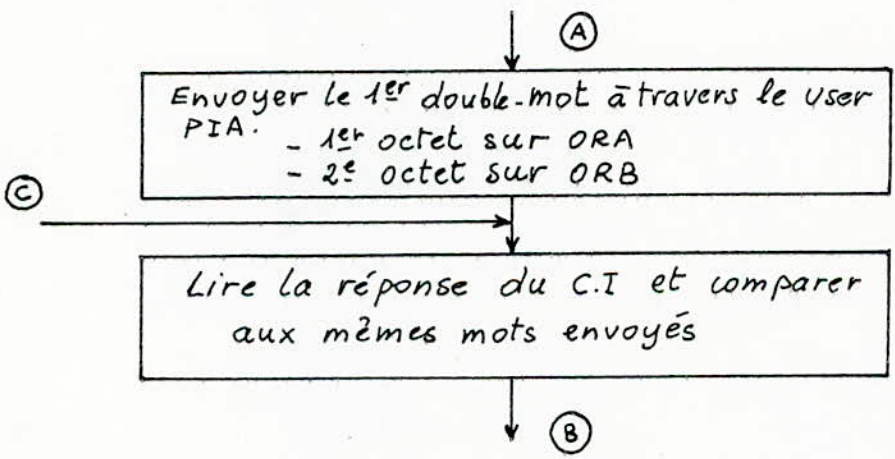


```

LDA 1,X
STAA PIA ORB
LDA PIA ORA
CMPA 0,X
BNE suite 2
  
```

} Ecriture du 2^e octet
 } Lecture plus comparaison et test du 2^e octet

— 4^e cas: Ecriture/Lecture sur ORA et Ecriture/Lecture sur ORB:



```

LDA 0,X
STAA PIA ORA
CMPA 0,X
BNE suite 2
LDA 1,X
STAA PIA ORB
CMPB 1,X
BNE suite 2
  
```

} - Ecriture / Lecture du 1^{er} octet
 } - comparaison et test du 1^{er} octet
 } - Ecriture / Lecture du 2^e octet
 } - comparaison et test du 2^e octet

III - la procédure du test d'un C.I :

Une fois les programmes des circuits intégrés sont logés dans la RAM, le test d'un CI peut s'effectuer selon la procédure suivante :

- Introduire d'abord l'adresse du début du programme
- Appuyer sur la touche G (les afficheurs sont mis à blanc).

- Introduire le numéro du CI, en chargeant son numéro correspondant en Hexadécimal dans l'accumulateur B.

(Comme on teste que des C.I de la famille 74, il n'est pas nécessaire d'introduire le numéro 74 à chaque fois. Ainsi si le numéro du C.I est du type :

- * 74xy → Introduire la valeur en Hexadécimale correspondante à xy.
- * 74xyz → Introduire la valeur en Hexadécimale correspondante à xyz.

- une fois le numéro est introduit, le kit nous informera si notre CI peut être testé ou non par le testeur en affichant : "out" dans le cas où le numéro introduit n'appartient pas à l'ensemble des C.I inclus dans notre programme de test.

- Placer le C.I sur le support correspondant (on ajoute des résistances aux sorties du CI dans le cas d'un circuit à collecteur ouvert)

- Déclenchement du programme de recherche du code ; une fois le code retrouvé le micro se branche directement au programme de test.

- le kit nous répondra alors par affichage de :

- * DEFECT si le CI est défectueux
- * GOOD si le CI est bon (c.à.d fonctionne).

Notre programme de test a été porté sur 42 circuits intégrés. on a donc 42 codes correspondant à une table de : $3 \times 42 + 1$ octets = 127 octets parce que chaque circuit intégré prend 3 octets, et un octet pour le compteur.

La table.1. est logée dans une zone mémoire qui commence par l'adresse E000 jusqu'à E07F. Le compteur sera chargé par le nombre d'octets que forme la table.1., c'est à dire le chargement de la position mémoire E000 (compteur) par la valeur 126 oct correspondant à "7E" en Hexadécimal.

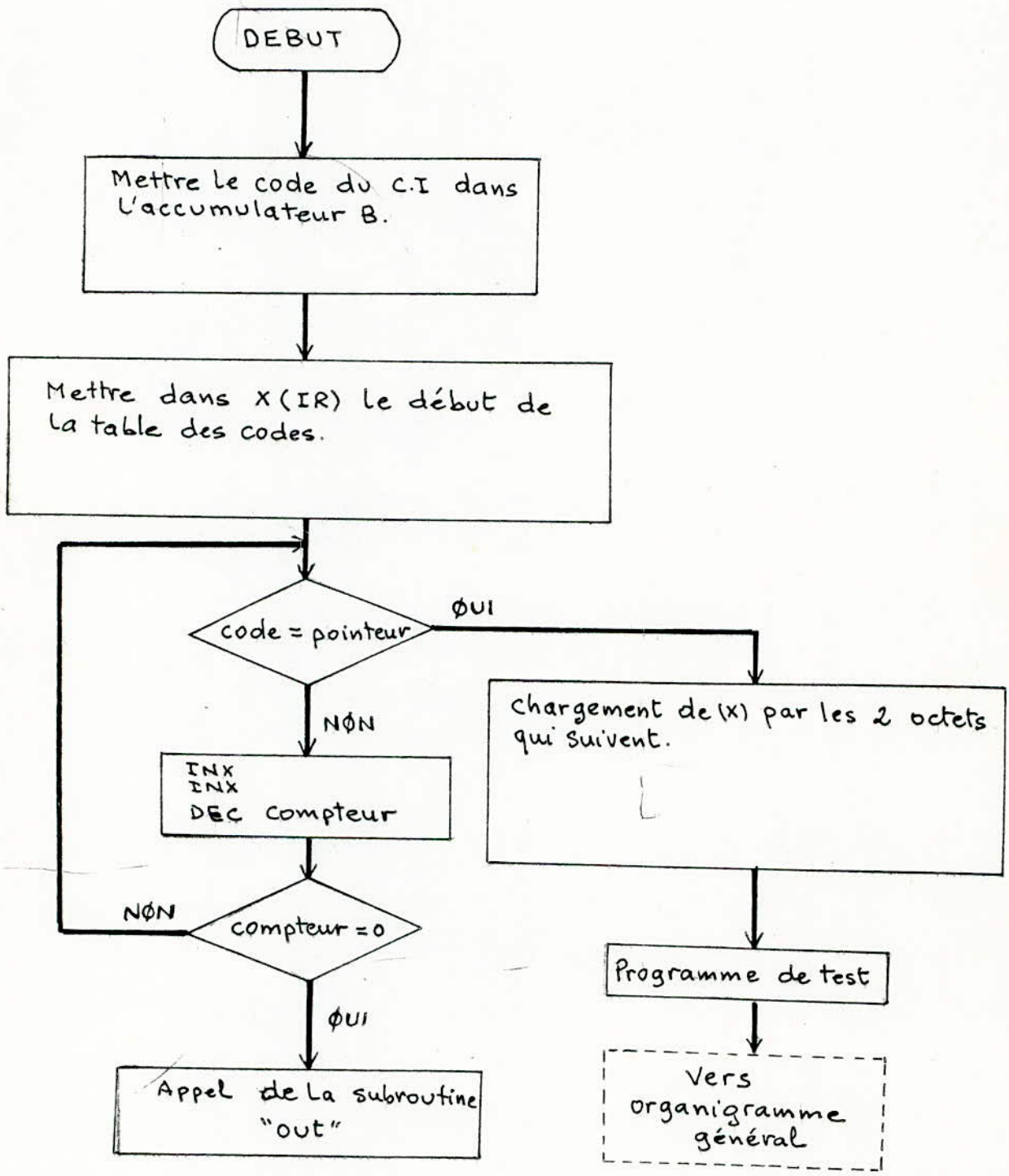
On voit bien sur la table.1. que chaque code du C.I à tester est suivi par l'adresse du programme de test sur les deux octets suivants.

Compteur
code 1
adresse du programme de test
code 2
adresse du programme de test
code 3
adresse du programme de test
code 42
adresse du programme de test

"table.1."

Une fois le numéro du C.I est introduit, il faut alors comparer les chiffres qui le constituent successivement aux différents nombres de la table des circuits testables par notre testeur qui est stockée en mémoire. Une méthode générale consisterait à convertir le code en Hexadécimal et le comparer à la table, ce qui nécessite l'emploi d'une routine de conversion Décimal - Hexadécimal. Et comme, on est limité par le champ mémoire, on a opté pour une autre méthode qui consiste à introduire le code en Hexadécimal directement, dans l'accumulateur B.

3.1 - Organigramme de test du C.I : Recherche du code et de l'adresse du programme de test correspondant :



Une initialisation consiste à définir le sens du dialogue entre le testeur et le C.I à tester.

On restaure l'adresse du début de la table des doubles mots de test ainsi que sa longueur. On les charge dans des positions mémoires propres au programme de test.

Il suffit de comparer le mot lu à celui envoyé, s'ils diffèrent, le C.I est alors défectueux, on arrête le déroulement du test et on affiche le mot "DEFFECT".

S'il n'y a aucune erreur (différence) en parcourant toute la table, le C.I est bon, on arrête le test en affichant le mot "E000".

Voici, le programme de test : (recherche du code et de l'adresse du programme de test du C.I) :

```

LDAA # $7E
STAA M
LDX # M
suite 1 LDAA 0, X
CBA
BNE suite 2
LDX 1, X
JMP 0, X
INX
INX
INX
DEC M
BNE suite 1
JMP Adr

```

Programme de recherche du code ainsi que l'adresse du programme de test du C.I



Appel de la subroutine de visualisation du mot "OUT"

Ce programme est logé dans la zone mémoire qui s'étend de l'adresse E07F jusqu'à E098.

3.2 - Pagination :

La zone mémoire dans le logiciel est insérée puis représentée de la manière suivante :

E000	Les différents codes des C.I ainsi que les adresses des différents programmes de test
E07E	Programme de recherche du code plus l'adresse du programme de test du C.I
E07F	
E09A	Sub ① et sub ②
E09B	
E0A5	sub ③, sub ④ et subroutine du mot "out"
E0A6	
E0E5	Tables de vérité et programmes de test des C.I.
E0E6	
E3FF	

Remarques :

- sub ① : Permettre l'accès aux 2 registres DDRA et DDRB
- sub ② : Permettre le passage d'une donnée à l'autre sur la table de vérité, sachant qu'une donnée est représentée sur 2 octets.
- sub ③ : Visualisation du mot "GOOD"
- sub ④ : Visualisation du mot "DEFECT"
- sub "out" : Visualisation du mot "OUT"

IV - Les différents circuits intégrés à tester :

4.1 - Brochage et paramètres du test du C.I :

Nous avons précisé auparavant que pour pouvoir tester un C.I, on avait besoin de connaître un certain nombre de paramètres correspondant à ce circuit :

Parmi ces paramètres, on a :

- la définition du sens des lignes user PIA - C.I. :
 - 1^{er} mot (à 8 bits) : à charger dans DDRA
 - 2^e mot (à 8 bits) : à charger dans DDRB
- la table de test du C.I. (cette table est constituée par des doubles-mots.)
- la longueur de cette table.

Pour donner ici, l'ensemble de ces paramètres à tous les C.I. liés à notre programme de test sous forme de tableau :

SN 7404 - 7405 - 7406 - 7414 - 7416	Sens des Lignes	Table de Verité	Ecriture/Lecture sur PIA	
Brochage : Positive Logic $y = \bar{A}$ 	Mot 1 : F9	F6	ORA	E/L
	Mot 2 : 66	99	ORB	E/L
	Longueur de la table : 02	F9	ORA	E/L
		66	ORB	E/L

SN 7407 - 7417	Sens des Lignes	Table de Verité	Ecriture/Lecture sur PIA	
Brochage : Positive Logic $y = A$ 	Mot 1 : F9	F0	ORA	E/L
	Mot 2 : 66	00	ORB	E/L
	Longueur de la table : 02	FF	ORA	E/L
		FF	ORB	E/L

SN: 7400-7403-7426-7437-7438

Brochage: Positive Logic: $y = \overline{AB}$

Sens des Lignes	table de vérité	Ecriture - Lecture sur PIA	
Mot 1: FF	F0	ORA	E
Mot 2: A5	5A	ORB	E/L
Longueur de la table 04	FC	ORA	E
	5F	ORB	E/L
	F3	ORA	E
	FA	ORB	E/L
	FF	ORA	
	A5	ORB	

SN: 7402-7428-7433-74128

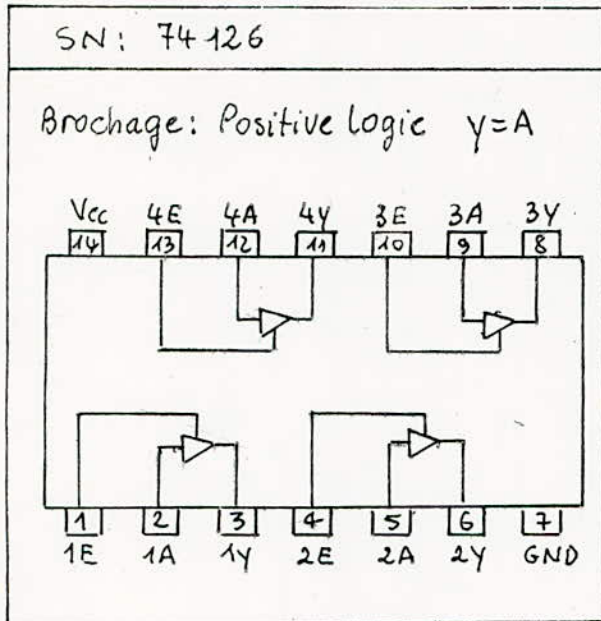
Brochage: Positive Logic $y = \overline{A+B}$

Sens des Lignes	table de vérité	Ecriture - Lecture sur PIA	
Mot 1: F0	0F	ORA	L
Mot 2: FF	00	ORB	E
Longueur de la table 04	00	ORA	L
	AA	ORB	E
	00	ORA	L
	55	ORB	E
	00	ORA	L
	FF	ORB	E

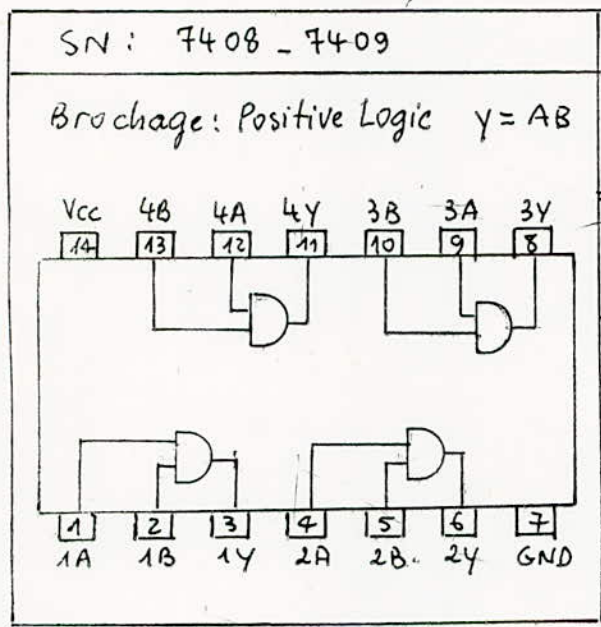
SN: 74125

Brochage: Positive Logic $y = A$

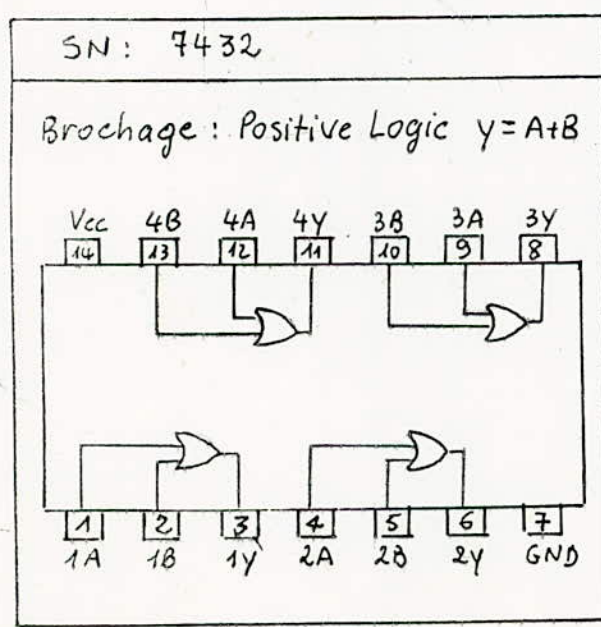
Sens des Lignes	table	E/L sur PIA	
Mot 1: FF	F0	ORA	E
Mot 2: A5	00	ORB	E/L
Longueur de la table 04	FC	ORA	E
	5F	ORB	E/L
	F3	ORA	E
	FA	ORB	E/L
	FF	ORA	E
	FF	ORB	E/L



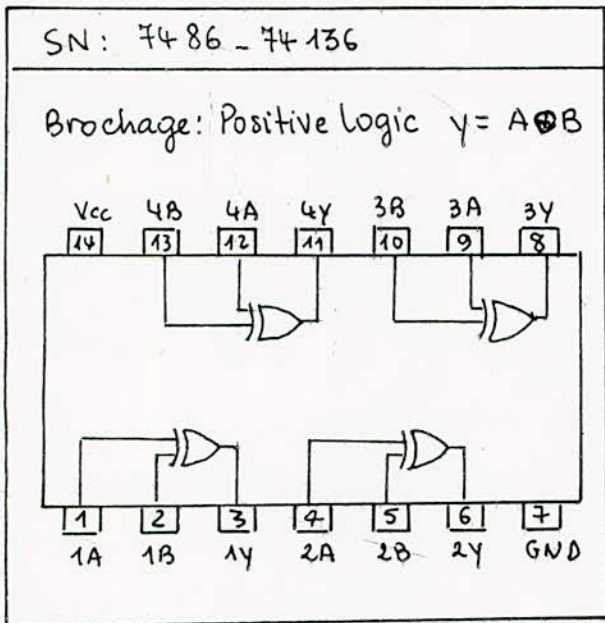
Sens des Lignes	Table	E/L sur PIA	
Mot 1: FF	FC	ORA	E
Mot 2: A5	0A	ORB	E/L
Longueur de la table 04	FC	ORA	E
	AF	ORB	E/L
	F3	ORA	E
	50	ORB	E/L
	FF	ORA	E
	FF	ORB	E/L



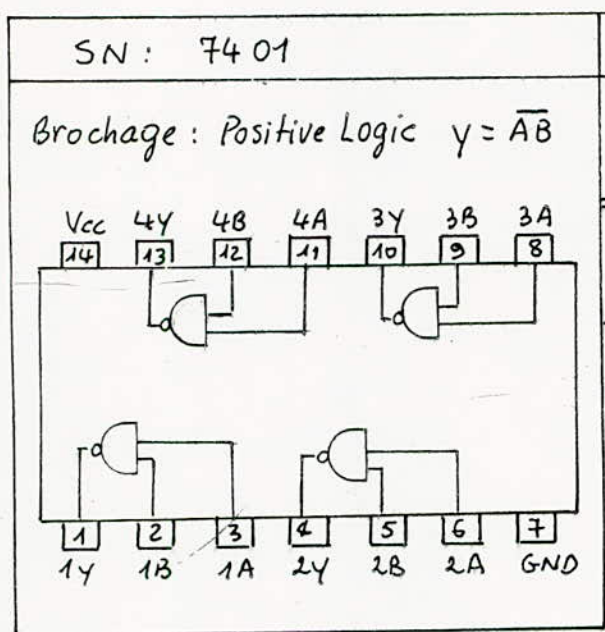
Sens des Lignes	table	E/L sur PIA	
Mot 1: FF	F0	ORA	E
Mot 2: A5	00	ORB	E/L
longueur de la table 04	FC	ORA	E
	05	ORB	E/L
	F3	ORA	E
	A0	ORB	E/L
	FF	ORA	E
	FF	ORB	E/L



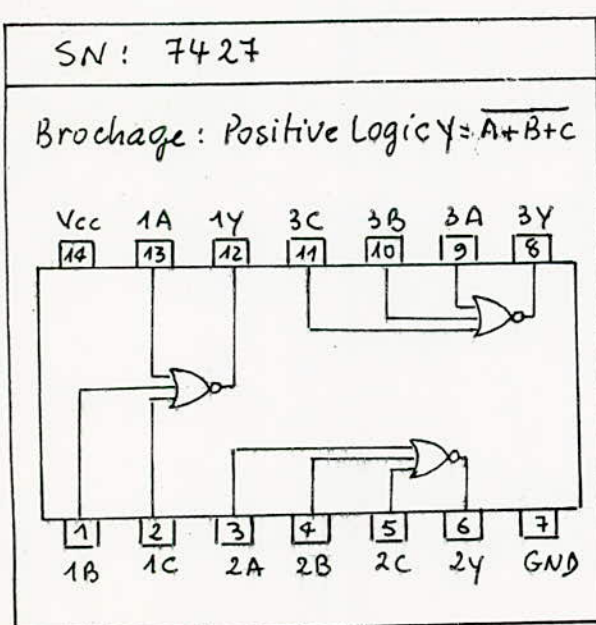
Sens des Lignes	table	E/L sur PIA	
Mot 1: FF	F0	ORA	E
Mot 2: A5	00	ORB	E/L
Longueur de la table 04	FC	ORA	E
	5F	ORB	E/L
	F3	ORA	E
	FA	ORB	E/L
	FF	ORA	E
	FF	ORB	E/L



Sens des Lignes	Table	E/L sur PIA	
Mot1: FF	F0	ORA	E
Mot2: A5	00	ORB	E/L
longueur de la table 04	FC	ORA	E
	5F	ORB	E/L
	F3	ORA	E
	FA	ORB	E/L
	FF	ORA	E
	A5	ORB	E/L



Sens des Lignes	Table	E/L sur PIA	
Mot1: F0	00	ORA	L
Mot2: FF	FF	ORB	E
longueur de la table 04	AA	ORA	L
	FF	ORB	E
	55	ORA	L
	FF	ORB	E
	FF	ORA	L
	F0	ORB	E



Sens des lignes	Table	E/L sur PIA	
Mot1: FF	F0	ORA	E
Mot2: 67	98	ORB	E/L
longueur de la table 04	F8	ORA	E
	44	ORB	E/L
	F7	ORA	E
	23	ORB	E/L
	FF	ORA	E
	67	ORB	E/L

SN: 7411 - 7415

Brochage: Positive Logic $y = ABC$

Sens des lignes	Table	E/L sur PIA	
Mot1: FF	F0	ORA	E
Mot2: 67	00	ORB	E/L
longueur de la table 04	F8	ORA	E
	44	ORB	E/L
	F7	ORA	E
	23	ORB	E/L
	FF	ORA	E
	FF	ORB	E/L

SN: 7410 - 7412

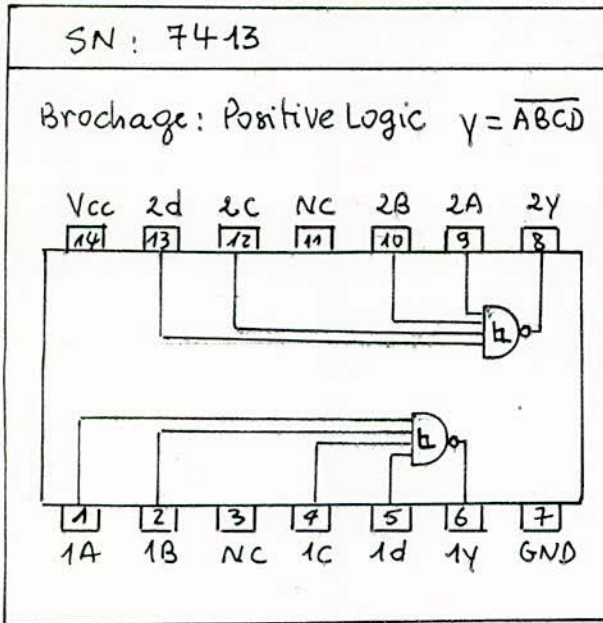
Brochage: Positive Logic $y = \overline{ABC}$

Sens des lignes	Table	E/L sur PIA	
Mot1: FF	F0	ORA	E
Mot2: 67	98	ORB	E/L
longueur de la table 04	F8	ORA	E
	dc	ORB	E/L
	F7	ORA	E
	BB	ORB	E/L
	FF	ORA	E
	67	ORB	E/L

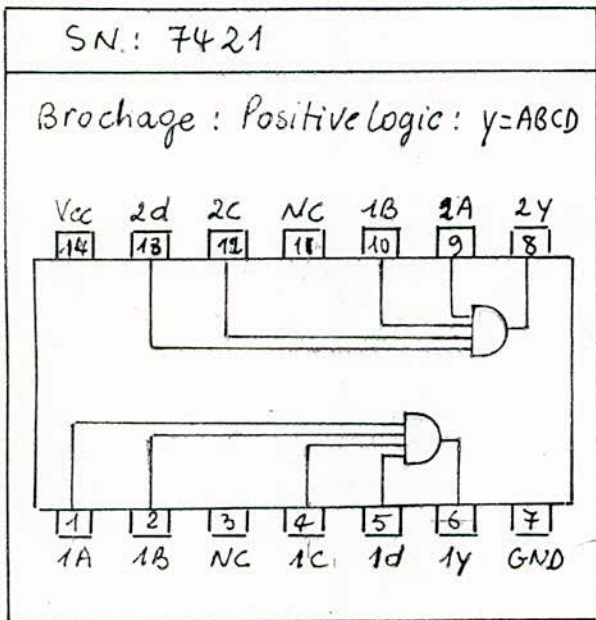
SN: 7420 - 7422

Brochage: Positive Logic $y = \overline{ABCD}$

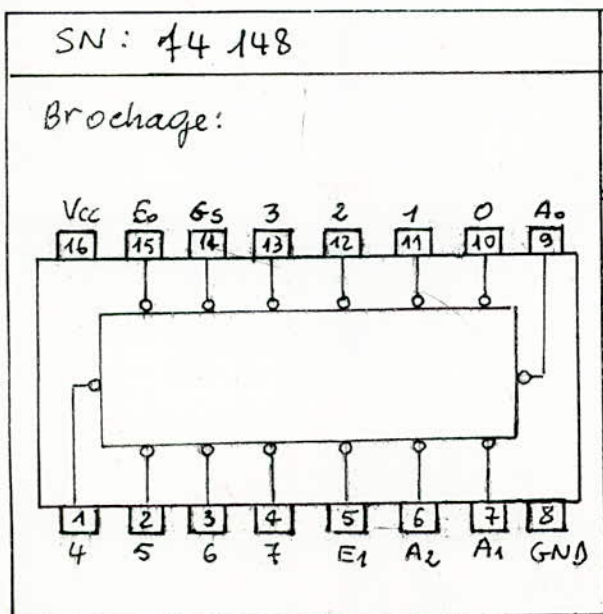
Sens des lignes	Table	E/L sur PIA	
Mot1: FF	F6	ORA	E
Mot2: E7	dB	ORB	E/L
longueur de la table 04	FA	ORA	E
	dE	ORB	E/L
	FF	ORA	E
	E7	ORB	E/L
	F0	ORA	E
	5A	ORB	E/L



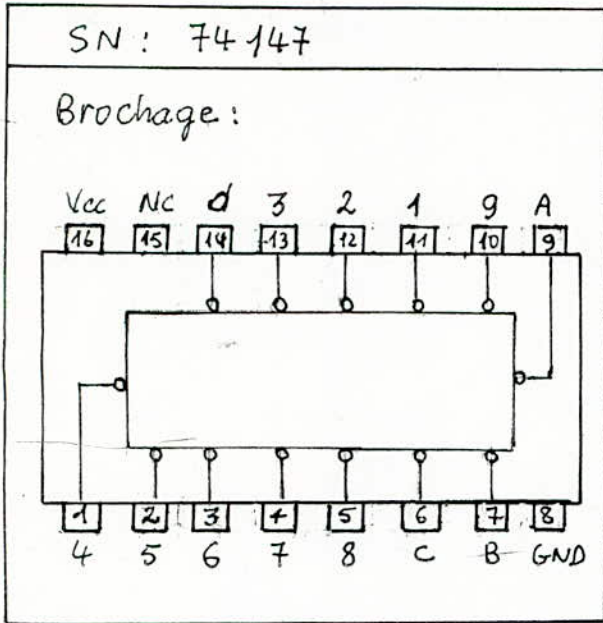
Sens des lignes	Table	E/L sur PIA	
Mot1: FF	F6	ORA	E
Mot2: E7	d8	ORB	E/L
Longueur de la table 04	FA	ORA	E
	dE	ORB	E/L
	FF	ORA	E
	E7	ORB	E/L
	F0	ORA	E
	5A	ORB	E/L



Sens des lignes	Table	E/L sur PIA	
Mot1: FF	F6	ORA	E
Mot2: E7	C3	ORB	E/L
Longueur de la table 04	FA	ORA	E
	C6	ORB	E/L
	FF	ORA	E
	FF	ORB	E/L
	F0	ORA	E
	42	ORB	E/L



Sens des lignes	Table		E/L	
	(1)	(2)		
Mot1: dE	E1	C1	ORA	E/L
Mot2: 1F	F0	CE	ORB	E/L
Longueur de la table 0A	FF	C0	ORA	E/L
	6F	AF	ORB	E/L
	C0	d1	ORA	E/L
	80	AF	ORB	E/L
	C1	d8	ORA	E/L
	88	FF	ORB	E/L
	C0	dd	ORA	E/L
	CC	EF	ORB	E/L



Sens des lignes	← Table →		E/L	
Mot1: dE	FF	E2	ORA	E/L
Mot2: 9F	FF	dC	ORB	E/L
longueur de la Table 0A	C0	E3	ORA	E/L
	E0	dE	ORB	E/L
L	C3	E2	ORA	E/L
	E0	BF	ORB	E/L
	E2	F3	ORA	E/L
	90	BF	ORB	E/L
	E3	F2	ORA	E/L
	98	FF	ORB	E/L

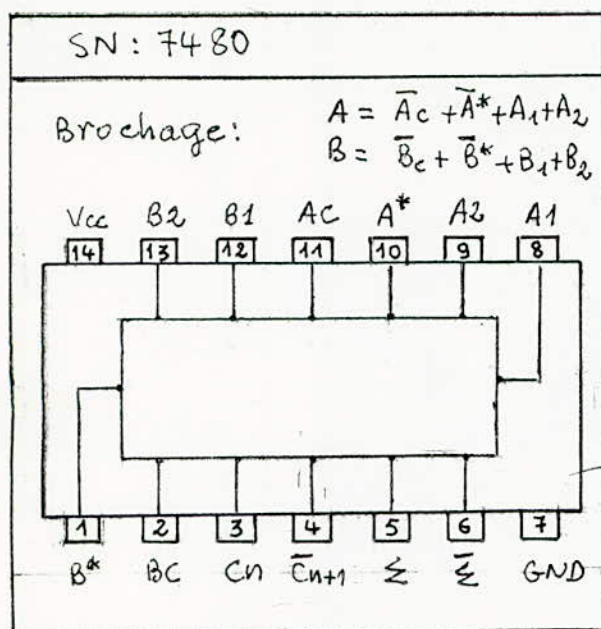
SN: 74147
- Table de Vérité -

Entrées									Sorties			
1	2	3	4	5	6	7	8	9	d	C	B	A
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	L	H	H	H	L	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	L	H	L
X	X	L	H	H	H	H	H	H	H	H	L	H
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

SN: 74148
Table de Vérité

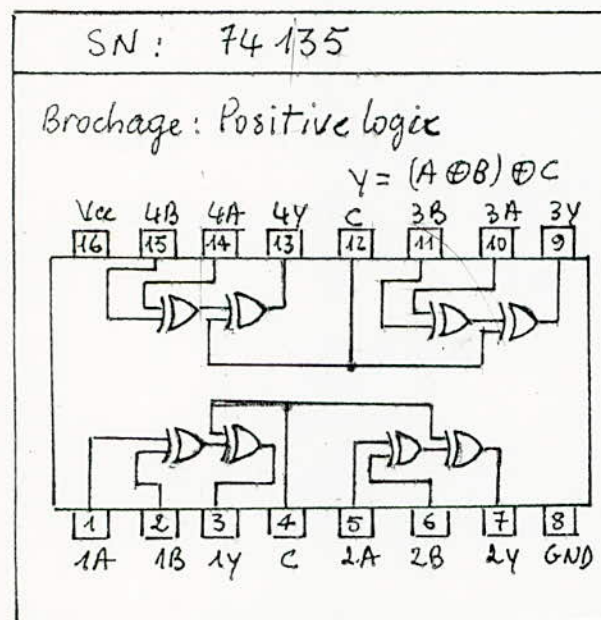
Entrées								Sorties					
E ₁	0	1	2	3	4	5	6	7	A ₂	A ₁	A ₀	G ₀	E ₀
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	L	H	H	H	L	H	L	L	H
L	X	X	X	L	H	H	H	H	L	H	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

H: niveau Haut : "1"
 L: niveau Bas : "0"
 X: quelconque (dans notre Application X=0)



Sens des lignes

	← Table →		E/L	
Mot1: Fd	F7	F7	ORA	E/L
Mot2: F3	49	47	ORB	E/L
Longueur de la table 08	F2	F0	ORA	E/L
	45	48	ORB	E/L
	F7	F5	ORA	E/L
	45	48	ORB	E/L
	F0	F0	ORA	E/L
	49	47	ORB	E/L



Sens des lignes

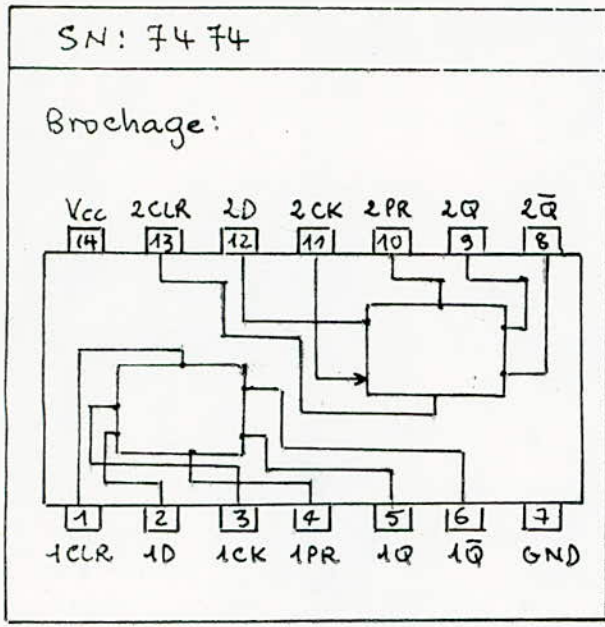
	← Table →		E/L	
Mot1: EE	C0	d9	ORA	E/L
Mot2: BB	00	4C	ORB	E/L
Longueur de la table 08	d5	CC	ORA	E/L
	E6	AA	ORB	E/L
	F3	EA	ORA	E/L
	55	19	ORB	E/L
	E6	FF	ORA	E/L
	B3	FF	ORB	E/L

Entrées			Sorties		
Cn	B	A	\bar{C}_{n+1}	\bar{Z}	Z
L	L	L	H	H	L
L	L	H	H	L	H
L	H	L	H	L	H
L	H	H	L	H	L
H	L	L	H	L	H
H	L	H	L	H	L
H	H	L	L	H	L
H	H	H	L	L	H

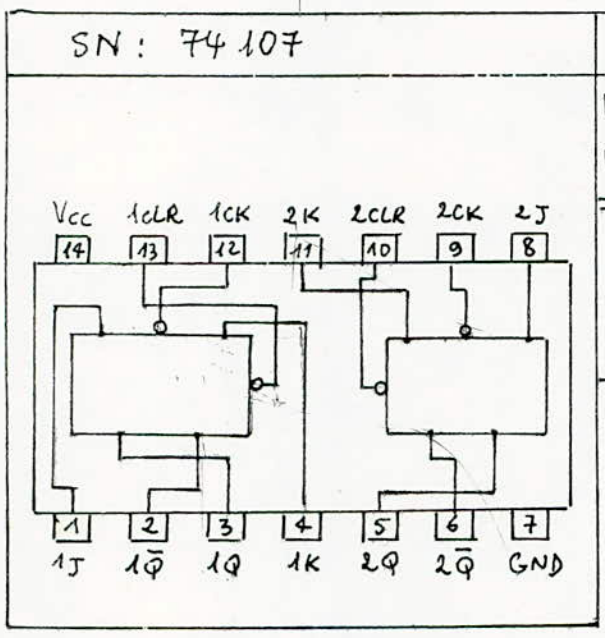
Entrées			S
A	B	C	Y
L	L	L	L
L	H	L	H
H	L	L	H
H	H	L	L
L	L	H	H
L	H	H	L
H	L	H	L
H	H	H	H

SN 7480
 - table modifiée -

SN 74135
 - table modifiée -



Sens des lignes	Table		E/L	
	(1)	(2)		
Mot1: FF	Fd	FF	ORA	E
Mot2: C3	24	24	ORB	E/L
longueur de la table 08	F6	FF	ORA	E
	18	24	ORB	E/L
	FF	FF	ORA	E
	99	5A	ORB	E/L
	FF	FF	ORA	E
	E7	18	ORB	E/L



S.L	Table			E/L	
Mot1: FF	Autres Etats		TOGGLE	Sur PIA	
Mot2: F0	F0	FC	FF	ORA	E
longueur de la table 23	09	A6	59	ORB	E/L
	FC	FC	FF	ORA	E
	09	06	59	ORB	E/L
	FC	FE	FF	ORA	E
	A9	46	F9	ORB	E/L
	FC	FE	FF	ORA	E
	09	E6	56	ORB	E/L
	Fd	FE	FF	ORA	E
	19	49	56	ORB	E/L
	Fd	FF	FF	ORA	E
	B9	59	59	ORB	E/L
	Fd	FF	FF	ORA	E
	16	F9	F9	ORB	E/L
	FC	ORA ←	FF	ORA	E
	06	ORB ←	59	ORB	E/L

Entrées				S	
clear	clock	J	K	Q	Q̄
L	0	0	0	L	H
H	∩	L	L	Q ₀	Q̄ ₀
H	∩	H	L	H	L
H	∩	L	L	Q ₀	Q̄ ₀
H	∩	L	H	L	H
H	∩	H	H	TOGGLE	
H	∩	H	H	TOGGLE	

Entrées				S	
Préset	clear	clock	D	Q	Q̄
L	H	0	0	H	L
H	L	0	0	L	H
H	H	↑	H	H	L
H	H	L	0	Q	Q̄
H	H	↑	L	L	H
H	H	L	0	Q ₀	Q̄ ₀

SN: 74.74
- Table modifiée -

SN: 74107
- Table modifiée -

Entrées				Sorties	
Preset	clear	clock	D	Q	\bar{Q}
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H*	H*
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q ₀	\bar{Q}_0

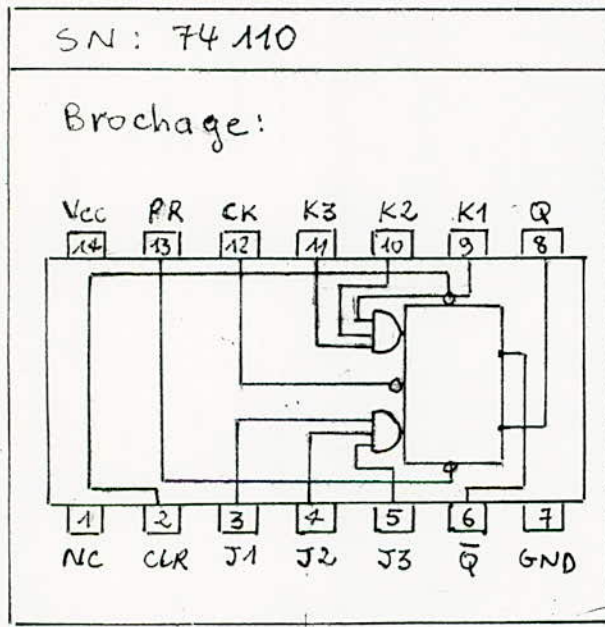
- Table de vérité = SN:74.74

Entrées				Sorties	
clear	clock	J	K	Q	\bar{Q}
L	X	X	X	L	H
H	↯	L	L	Q ₀	\bar{Q}_0
H	↯	H	L	H	L
H	↯	L	H	L	H
H	↯	H	H	TOGGLE	

- Table de vérité = SN:74107

H* : Etat indéterminé

Q₀, \bar{Q}_0 : conservation d'état précédent



sens des lignes	← Table →			EIL sur PIA	
	Autres Etats	TOGGLE			
Mot1: FF	F1	Fd	FF	ORA	E
Mot2: EF	11	71	6F	ORB	\bar{E}
longueur de la table 24	F9	Fd	FF	ORA	E
	08	F1	6F	ORB	\bar{E}
	FB	Fd	FF	ORA	E
	0F	69	EF	ORB	\bar{E}
	FB	F9	FF	ORA	E
	8F	09	77	ORB	\bar{E}
	FB	F9	FF	ORA	E
	17	89	77	ORB	\bar{E}
	F9	F9	FF	ORA	E
	11	09	6F	ORB	\bar{E}
	F9	FF	FF	ORA	E
	91	6F	EF	ORB	\bar{E}
F9	FF	FF	ORA	E	
11	EF	6F	ORB	\bar{E}	

Entrées				Sorties		
Preset	clear	clock	J	K	Q	\bar{Q}
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H*	H*
H	H	↯	L	L	Q ₀	\bar{Q}_0
H	H	↯	H	L	H	L
H	H	↯	L	H	L	H
H	H	↯	H	H	TOGGLE	

- Table de vérité: SN74110

Entrées				Sorties		
Preset	clear	clock	J	K	Q	\bar{Q}
L	H	0	0	0	H	L
H	L	0	0	0	L	H
H	H	↯	H	L	H	L
H	H	↯	L	L	Q ₀	\bar{Q}_0
H	H	↯	L	H	L	H
H	H	↯	L	L	Q ₀	\bar{Q}_0
H	H	↯	H	H	TOGGLE	
H	H	↯	H	H	TOGGLE	

- Table Modifié: SN:74110

SN : 74 109

Brochage:

Vcc 2CLR 2J 2K 2CK 2PR 2Q 2Q̄
 16 15 14 13 12 11 10 9

1 2 3 4 5 6 7 8
 1CLR 1J 1K 1CK 1PR 1Q 1Q̄ GND

Sens des lignes ← Table →

Autres états			E/L sur PIA	
C2	dd	EE	ORA	E/L
A1	dd	BB	ORB	E/L
C5	F5	E6	ORA	E/L
50	d7	B3	ORB	E/L
C5	FE	Ed	ORA	E/L
d1	BF	dB	ORB	E/L
cd	CE	Ed	ORA	E/L
d9	B1	dB	ORB	E/L
d5	E6	E6	ORA	E/L
d5	B3	B3	ORB	E/L
		EE	ORA	E/L
		BB	ORB	E/L

Mot1: FC
Mot2: 9F

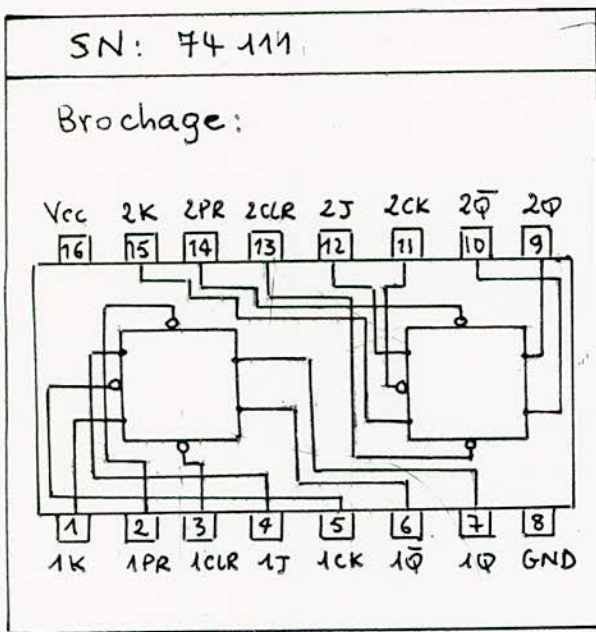
Longueur de la table
16

Entrées					Sorties	
Préset	clear	clock	J	K	Q	Q̄
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H*	H*
H	H	↑	L	L	L	H
H	H	↑	H	L	TOGGLE	
H	H	↑	L	H	Q	Q̄
H	H	↑	H	H	H	L
H	H	L	X	X	Q ₀	Q̄ ₀

- Table de Vérité : SN 74 109

Entrées					Sorties	
Préset	clear	clock	J	K	Q	Q̄
L	H	0	0	0	H	L
H	L	0	0	0	L	H
H	H	↑	L	L	L	H
H	H	↑	L	H	Q ₀	Q̄ ₀
H	H	↑	H	H	H	L
H	H	L	0	0	Q	Q̄
H	H	↑	H	L	TOGGLE (H)	
H	H	↑	H	L	TOGGLE (L)	

- Table modifiée : SN 74 109



sens des lignes	← Table →			E/L	
	Autres Etats	TOGGLE	sur Pi A		
Mot1: FC	d1	F1	FA	ORA	ELL
Mot2: 9F	44	C7	AF	ORB	EIL
longueur de la table 24	E2	F5	FA	ORA	EIL
	22	d7	AF	ORB	EIL
	FA	F2	FE	ORA	EIL
	2E	A7	BF	ORB	EIL
	FE	F2	F9	ORA	EIL
	3E	26	CF	ORB	EIL
	F9	F6	F9	ORA	EIL
	4E	36	CF	ORB	EIL
	F1	F2	FA	ORA	EIL
	46	26	AF	ORB	EIL
	F5	FA	FE	ORA	EIL
	56	AF	BF	ORB	EIL
	F1	FE	FA	ORA	EIL
	46	BF	CF	ORB	EIL

Entrées					Sorties	
Pre-set	clear	clock	J	K	Q	Q-bar
L	H	0	0	0	H	L
H	L	0	0	0	L	H
H	H	∩	H	L	H	L
H	H	∩	L	L	Q ₀	Q ₀ -
H	H	∩	L	H	L	H
H	H	∩	L	L	Q ₀	Q ₀ -
H	H	∩	H	H	TOGGLE (1)	
H	H	∩	H	H	TOGGLE (2)	

Entrées					Sorties	
Pre-set	clear	clock	J	K	Q	Q-bar
L	H	X	X	X	H	L
H	L	X	X	X	L	H
H	H	∩	L	L	Q ₀	Q ₀ -
H	H	∩	H	L	H	L
H	H	∩	L	H	L	H
H	H	∩	H	H	TOGGLE	

4.2 - Représentation de la table de vérité du C.I séquentiel :

Le logiciel adopté pour les circuits intégrés séquentiels a été fait de la manière suivante :

On élargit la table de vérité -1- en une table plus complète, qui prend en considération les différents états et en particulier les états (Q_0, \bar{Q}_0 et TOGGLE) qui présentent quelques particularités.

- l'état (Q_0, \bar{Q}_0) veut dire conservation de l'état précédent. Pour bien montrer cette conservation, on se misera cet état devant deux autres états différents, comme l'indique la table n°2.

- l'état (TOGGLE) est un état qui assure un changement d'état à chaque impulsion d'horloge. On a alors inséré deux (TOGGLE) successivement dans la nouvelle table pour bien prouver ce changement d'état.

La table de vérité a été conçue suivant ce mode :

- fixer le X à zéro
- gérer l'impulsion \square , donc assurer le passage successif des états suivants (0 - 1 - 0).
- gérer l'état H (haut niveau) : mise à 1
- gérer l'état L (bas niveau) : mise à 0

La table n°3 a été déduite d'après la table n°2 et à partir de celle-ci découle notre table de vérité qui prend en considération les données précédentes. Cette est composée de deux parties : la première représente l'ensemble des états, la seconde représente l'état TOGGLE au niveau du software. On a donc adopté un logiciel pour l'état TOGGLE qui consiste à faire une recherche d'état, c'est à dire voir quel est l'état actuel du 1^{er} TOGGLE et assurer le changement d'état pour le 2^{em} TOGGLE.

- Considérons le C.I 74107 :

- table fonction -

Inputs				outputs	
clear	clock	J	K	Q	\bar{Q}
L	X	X	X	L	H
H		L	L	Q ₀	\bar{Q}_0
H		H	L	H	L
H		L	H	L	H
H		H	H	TOGGLE	
*	*	*	*	H*	H*

- table n°1 -

- Table Modifiée -

Inputs				outputs	
clear	clock	J	K	Q	\bar{Q}
L	X	X	X	L	H
H		L	L	Q ₀	\bar{Q}_0
H		H	L	H	L
H		L	L	Q ₀	\bar{Q}_0
H		L	H	L	H
H		H	H	TOGGLE(1)	
H		H	H	TOGGLE(2)	

- Table n°2 -

- Remplissage de table Modifiée -

Inputs				outputs		
clear	clock	J	K	Q	\bar{Q}	
0	0	0	0	0	1	
1		0	0	0	1	} Q ₀ , \bar{Q}_0
1		1	0	0	1	
1		0	0	0	1	
1		0	1	0	1	} Q ₀ , \bar{Q}_0
1		1	0	0	1	
1		0	1	0	1	
1		0	0	0	1	} TOGGLE (1)
1		1	0	0	1	
1		0	0	0	1	
1		0	1	1	0	} TOGGLE (2)
1		1	1	0	1	
1		0	1	1	0	

- table n°3 -

- Table de vérité du C.I

F 0
0 9
F C
0 9
F C
A 9
F C
0 9
F d
1 9
F d
B 9
F d
1 6
F C
0 6
F C
A 6
F C
0 6
F E
4 6
F E
E 6
F E
4 9
F F
5 9
F F
F 9
F F
5 9/6
F F
5 9
F F
F 9
F F
5 6/9

(1)

F 0
0 9
F C
0 9
F C
A 9
F C
0 9
F d
1 9
F d
B 9
F d
1 6
F C
0 6
F C
A 6
F C
0 6
F E
4 6
F E
E 6
F E
4 9
F F
5 9
F F
F 9
F F
5 9
F F
F 9
F F
5 6
F F
5 6
F F
5 9
F F
F 9
F F
5 9

(2)

Dans ce chapitre, on a donné les tables représentant l'ensemble des C.I à tester. Chaque table récapitule les données suivantes: (Brochage, fonction, sens des lignes, longueur de la table, Ecriture/Lecture sur PIA et la table de vérité); qui nous renseignent sur toutes les caractéristiques de ces C.I, facilitant ainsi notre travail.

Pour les C.I séquentiels, on a misé la fonction table élargie ou modifiée qui découle de la table de vérité du C.I.

Finalement, dans la partie logiciel, voici un tableau récapitulatif, sur lequel on trouve: le début et la fin de la table de vérité et du programme de test, ainsi qu'une conversion décimal - Hexadécimal du code de chaque C.I de notre travail:

4.3 - Tableau récapitulatif:

code du C.I en décimal	code du C.I en Hexadécimal	Début de La table de Vérité	Fin de la table de vérité	Début du programme de test	Fin du programme de test
74.00	4A.00	E1.46	E1 47	E1.47	E1 7E
74.01	4A.01	E2.13	E2 14	E2 14	E2 25
74.02	4A.02	E1 86	E1 87	E1 87	E1 B9
74.03	4A.03	E1 46	E1 47	E1 47	E1 7E
74.04	4A.04	E0 E9	E0 EA	E0 EA	E1 28
74.05	4A.05	E0 E9	E0 EA	E0 EA	E1 28
74.06	4A.06	E0 E9	E0 EA	E0 EA	E1 28
74.07	4A.07	E1 2C	E1 2d	E1 2d	E1 3E
74.08	4A.08	E1 E9	E1 EA	E1 EA	E1 E9
74.09	4A.09	E1 E9	E1 EA	E1 EA	E1 E9
74.10	4A.0A	E2 6A	E2 6B	E2 6B	E2 70
74.11	4A.0B	E2 54	E2 55	E2 55	E2 62
74.12	4A.0C	E2 6A	E2 6B	E2 6B	E2 70
74.13	4A.0d	E2 78	E2 79	E2 79	E2 8F

74.14	4A 0E	E0 E6	E0 E9	E0 EA	E1 28
74 15	4A 0F	E2 4d	E2 54	E2 55	E2 62
74 16	4A 10	E0 E6	E0 E9	E0 EA	E1 28
74 17	4A 11	E1 29	E1 2C	E1 2d	E1 3E
74 20	4A 14	E2 71	E2 78	E2 79	E2 8F
74 21	4A 15	E2 90	E2 97	E2 98	E2 9d
74 22	4A 16	E2 71	E2 78	E2 79	E2 8F
74 26	4A 1A	E1 3F	E1 46	E1 47	E1 7E
74 27	4A 1B	E2 26	E2 2d	E2 2E	E2 4C
74 28	4A 1C	E1 7F	E1 86	E1 87	E1 89
74 32	4A 20	E1 F0	E1 F7	E1 F8	E1 Fd
74 33	4A 21	E1 7F	E1 86	E1 87	E1 89
74 37	4A 25	E1 3F	E1 46	E1 47	E1 7E
74 38	4A 26	E1 3F	E1 46	E1 47	E1 7E
74 74	4A 4A	E3 3C	E3 4B	E3 4C	E3 6C
74 80	4A 50	E2 9E	E2 Ad	E2 AE	E2 CE
74 86	4A 56	E1 FE	E2 05	E2 06	E2 0B
74.107	4A 6B	E3 6d	E3 9A	E3 9B	E3 FC
74 109	4A 6d	00 00	00 2F	00 30	00 4C
74.110	4A 6E	-	-	-	-
74.111	4A 6F	-	-	-	-
74.125	4A 7d	E1 BA	E1 C1	E1 C2	E1 d3
74 126	4A 7E	E1 d4	E1 dB	E1 dC	E1 E1
74 128	4A 80	E1 7F	E1 86	E1 87	E1 89
74 135	4A 87	E2 CF	E2 dE	E2 dF	E2 EE
74 136	4A 88	E1 FE	E2 05	E2 06	E2 0B
74 147	4A 93	E3 18	E3 2B	E3 2C	E3 3B
74 148	4A 94	E2 EF	E3 02	E3 03	E3 17

V - listing du programme :

E000
 E001
 E002
 E003
 E004
 E005
 E006
 E007
 E008
 E009
 E00A
 E00B
 E00C
 E00D
 E00E
 E00F
 E010
 E011
 E012
 E013
 E014
 E015
 E016
 E017
 E018
 E019
 E01A
 E01B
 E01C
 E01D
 E01E
 E01F
 E020
 E021
 E022
 E023
 E024
 E025
 E026
 E027
 E028
 E029
 E02A
 E02B
 E02C
 E02D
 E02E
 E02F
 E030
 E031
 E032
 E033

•••••
 00 Code 1
 E1 }
 47 } adresse du Programme
 } de test
 01 Code 2
 E2 }
 14 } adresse du Programme
 } de test
 02 Code 3
 E1 }
 87 } adresse du Programme
 } de test
 03 Code 4
 E1 }
 47 } adresse du Programme
 } de test
 04 Code 5
 E0 }
 EA } adresse du Programme
 } de test
 05 Code 6
 E0 }
 EA } adresse du Programme
 } de test
 06 Code 7
 E0 }
 EA } adresse du Programme
 } de test
 07 Code 8
 E1 }
 2D } adresse du Programme
 } de test
 08 Code 9
 E1 }
 EA } adresse du Programme
 } de test
 09 Code 10
 E1 }
 EA } adresse du Programme
 } de test
 0A Code 11
 E2 }
 6B } adresse du Programme
 } de test
 0B Code 12
 E2 }
 55 } adresse du Programme
 } de test
 0C Code 13
 E2 }
 6B } adresse du Programme
 } de test
 0D Code 14
 E2 }
 79 } adresse du Programme
 } de test
 0E Code 15
 E0 }
 EA } adresse du Programme
 } de test
 0F Code 16
 E2 }
 55 } adresse du Programme
 } de test
 10 Code 17
 E0 }
 EA } adresse du Programme
 } de test

E034	1A	Code 18
E035	E1	} adresse du Programme
E036	2D	} de test
E037	14	Code 19
E038	E2	} adresse du Programme
E039	79	} de test
E03A	15	Code 20
E03B	E2	} adresse du Programme
E03C	98	} de test
E03D	16	Code 21
E03E	E2	} adresse du Programme
E03F	79	} de test
E040	1A	Code 22
E041	E1	} adresse du Programme
E042	47	} de test
E043	1B	Code 23
E044	E2	} adresse du Programme
E045	2E	} de test
E046	1C	Code 24
E047	E1	} adresse du Programme
E048	87	} de test
E049	20	Code 25
E04A	E1	} adresse du Programme
E04B	F8	} de test
E04C	21	Code 26
E04D	E1	} adresse du Programme
E04E	87	} de test
E04F	25	Code 27
E050	E1	} adresse du Programme
E051	47	} de test
E052	26	Code 28
E053	E1	} adresse du Programme
E054	47	} de test
E055	4A	Code 29
E056	E3	} adresse du Programme
E057	4C	} de test
E058	50	Code 30
E059	E2	} adresse du Programme
E05A	AE	} de test
E05B	56	Code 31
E05C	E2	} adresse du Programme
E05D	06	} de test
E05E	6B	Code 32
E05F	E3	} adresse du Programme
E060	9B	} de test
E061	6D	Code 33
E062	--	} adresse du Programme
E063	--	} de test
E064	6E	Code 34
E065	00	} adresse du Programme
E066	30	} de test
E067	6F	Code 35

E068	--	} adresse du programme
E069	--	} de test
E06A	7D	Code 36
E06B	E1	} adresse du programme
E06C	C2	} de test
E06D	7E	Code 37
E06E	E1	} adresse du programme
E06F	DC	} de test
E070	80	Code 38
E071	E1	} adresse du programme
E072	87	} de test
E073	87	Code 39
E074	E2	} adresse du programme
E075	DF	} de test
E076	88	Code 40
E077	E2	} adresse du programme
E078	06	} de test
E079	93	Code 41
E07A	E3	} adresse du programme
E07B	2C	} de test
E07C	94	Code 42
E07D	E3	} adresse du programme
E07E	03	} de test

E07F	86 7E	LDA A	#7E	} Programme de recherche du code ainsi que l'adresse du programme de test du CI	
E080	B7 E0 00	STAA	M		
E081	CE E0 01	LDA X	#M		
E082	A6 00	ST1	LDA A		0,X
E083	11		CBA		
E084	26 04		BNE		ST2
E085	EE 01		LDA X		1,X
E086	6E 00		JMP		0,X
E087	08	ST2	INX		
E088	08		INX		
E089	08		INX		
E090	7A E0 00		DEC		M
E091	26 EF		BNE		ST1
E092	7E E0 C7		JMP		ADR
E093					
E094	B7 E4 81	STAA	PIA.CRA	} Subroutine 1 Validation des registres de direction des données	
E095	B7 E4 83	STAA	PIA.CRB		
E096	39	RTS			
E097				} Subroutine 2 realisant le passage au double-Mot qui suit	
E098	08	INX			
E099	08	INX			
E100	5A	DEC B			
E101	39	RTS		} Subroutine 3 Visualisation du mot "Good"	
E102	CE 6F 5C	LDA X	#6F5C		
E103	FF E4 1D	STX	DISBUF		
E104	CE 5C 5E	LDA X	#5C5E		
E105	7E E0 D3	JMP	M		

E0B2	CE 5E 79	LDX	#5E 79	} subroutines 4 VISUALISATION du mot "DEFECT"
E0B5	FF E4 1D	STX	DISBUF	
E0B8	CE 71 79	LDX	# 71 79	
E0BB	FF E4 1F	STX	DISBUF+2	
E0BE	CE 39 70	LDX	# 39 70	
E0C1	FF E4 21	STX	DISBUF+4	} subroutines de VISUALISATION du mot "OUT"
E0C4	7E E0 DC	JMP	M	
E0C7	CE 3F 3E	LDX	#3F3E	
E0CA	FF E4 1D	STX	DISBUF	
E0CD	CE 70 40	LDX	#7040	
E0D0	7E E0 D3	JMP	M	
E0D3	FF E4 1F	STX	DISBUF+2	} appel de la subroutines de visualisation
E0D6	CE 40 40	LDX	# 40 40	
E0D9	FF E4 21	STX	DISBUF+4	
E0DC	CE F0 A2	LDX	#DIDDLE	
E0DF	FF E4 19	STX	MNPTR	
E0E2	7E F0 BB	JMP	PUT	
E0E5	3F	SWI		
E0E6			F6	
E0E7			99	} table de verite
E0E8			F9	
E0E9			66	
E0EA	4F	CLRA		
E0EB	B7 E1 02	STAA	M	} effacement de la position memoire ou se trouve l'instruction RTS Acces aux registres DDRA(B)
E0EE	BDE0 9B	JSR	SUB 1	
E0F1	86 F9	LDAA	#F9	} programmation du PIA en (E/S)
E0F3	B7 E4 80	STAA	PIADDR	
E0F6	86 66	LDAA	#66	} Acces aux registres ORA(B)
E0F8	B7 E4 82	STAA	PIADDRB	
E0FB	86 04	LDAA	#04	} longueur de la table de verite Retour
E0FD	BDE0 9B	JSR	SUB 1	
E100	C6 02	LDAB	#02	} Pointage au debut de (T.V)
E102	39	RTS		
E103	CEE0 E6	LDX	#M	} Ecriture + lecture donnees representees Par la table de verite verite (T.V)
E106	A6 00	LDAA	0,X	
E108	B7 E4 80	STAA	PIAORA	
E10B	A6 01	LDAA	1,X	
E10D	B7 E4 82	STAA	PIAORB	
E110	B6 E4 80	LDAA	PIAORA	
E113	A1 00	CMPA	0,X	
E115	26 0F	BNE	ST 2	
E117	B6 E4 82	LDAA	PIAORB	
E11A	A1 01	CMPA	1,X	
E11C	26 08	BNE	ST 2	
E11E	BDE0 A2	JSR	SUB 2	} Passage au double mot suivant Poursuite de l'écriture-lecture Appel de la subroutines "Good" Appel de la subroutines "DEFECT"
E121	26 E3	BNE	ST 1	
E123	7E E0 A6	JMP	SUB 3	
E126	7E E0 B2	JMP	SUB 4	

ST 1

ST 2

E129			F0	} table de Verite (T.V)
E12A			00	
E12B			FF	
E12C			FF	
E12D	CEE129	LDX	#M	} Pointage au debut de la(T.V)
E130	86 39	LDAA	#39	
E132	B7E102	STAA	M	
E135	4F	CLRA		} Mettre en evidence l'insto RTS dans le Prog- Principal Acces aux registres DDRA(B)
E136	BDE09B	JSR	SUB1	
E139	BDE0F1	JSR	ADR0	
E13C	7E E1 06	JMP	ADR1	

E13F			F0	} table de Verite
E140			5A	
E141			FC	
E142			5F	
E143			F3	
E144			FA	
E145			FF	
E146			A5	

E147	4F	CLRA		} effacement de la position memoire Correspondante à RTS
E148	B7E15F	STAA	M	
E14B	BDE09B	JSR	SUB1	} Acces aux registres DDRA(B)
E14E	86 FF	LDAA	#FF	
E150	B7E480	STAA	PIADDR	} Programmation du PIA en (E/S)
E153	86 A5	LDAA	#A5	
E155	B7E482	STAA	PIADRB	} Acces aux registres ORA(B)
E158	86 04	LDAA	#04	
E15A	BDE09B	JSR	SUB1	} Longueur de la (T.V)
E15D	C6 04	LDAB	#04	
E15F	39	RTS		} Pointage au debut de la(T.V)
E160	CEE13F	LDX	#M	
E163	A6 00	LDAA	0,X	} Ecriture + lecture ainsi que les comparaisons et les tests des donnees
E165	B7E480	STAA	PIADDR	
E168	A6 01	LDAA	1,X	
E16A	B7E482	STAA	PIADRB	
E16D	B6E482	LDAA	PIADRB	
E170	A1 01	CMPA	1,X	
E172	26 08	BNE	ST2	} Passage au double-Mot suivant Poursuite de "E/L" subroutine "Good" subroutine "DEFECT"
E174	BDE0A2	JSR	SUB2	
E177	26 EA	BNE	ST1	
E179	7E E0 A6	JMP	SUB3	
E17C	7E E0 B2	JMP	SUB4	

E17F			0F	} table de
E180			00	
E181			00	

E182			AA	} verité
E183			00	
E184			55	
E185			00	
E186			FF	
E187	4F		CLRA	} effacement de la Position memoire correspondante RTS
E188	B7 E19F		STAA M	
E18B	BD E09B		JSR SUB 1	} Acces aux registres DDRA(B)
E18E	86 F0		LDAA #F0	
E190	B7 E480		STAA PIADDR	} Programmation du PIA en E/S
E193	86 FF		LDAA #FF	
E195	B7 E482		STAA PIADDRB	} Acces aux registres ORA(B)
E198	86 04		LDAA #04	
E19A	BD E09B		JSR SUB 1	} Longueur de La(T.V)
E19D	C6 04		LDAB #04	
E19F	39		RTS	} Pointage au debut de La(T.V)
E1A0	CE E17F		LDX #M	
E1A3	A6 01	ST1	LDAA 1,X	} Ecriture + lecture ainsi que les Comparaisons et les tests des donnees
E1A5	B7 E482		STAA PIAORB	
E1A8	B6 E480		LDAA PIAORA	
E1AB	A1 00		CMPA 0,X	
E1AD	26 08		BNE ST2	} Passage au double-Mot suivant
E1AF	BDE0A2		JSR SUB 2	
E1B2	26 EF		BNE ST1	} Poursuite de E/L Subroutine "Good" Subroutine "DEFECT"
E1B4	7E E0A6		JMP SUB 3	
E1B7	7E E0B2	ST2	JMP SUB 4	

E1BA			F0	} table de verite
E1BB			00	
E1BC			FC	
E1BD			5F	
E1BE			F3	
E1BF			FA	
E1C0			FF	
E1C1			FF	

E1C2	CE E1 BA		LDX #M	} Pointage au debut de La(T.V)
E1C5	86 39		LDAA #39	
E1C7	B7 E1 5F		STAA M	} Mettre en evidence RTS dans le Programme Principal
E1CA	4F		CLRA	
E1CB	BD E0 9B		JSR SUB 1	} Acces aux registres DDRA(B) realiser la Prog du PIA en E/S realiser l'écriture + lecture ainsi que les Visualisations
E1CE	BD E1 4E		JSR ADRO	
E1D1	7E E1 63		JMP ADR1	

E1D4			FC	} table de verite
E1D5			0A	
E1D6			FC	
E1D7			AF	
E1D8			F3	

E1D9			50	}	
E1DA			FF		
E1DB			FF		
E1D				}	
E1DC	CE E1D4	LDX	#M		Pointage au debut de la T.V Realiser la programmation du PIA en (E/S) ainsi que l'écriture + lecture
E1DF	7E E1C5	JMP	ADR		

E1E2			F0	}	
E1E3			00		
E1E4			FC		table de verite
E1E5			05		
E1E6			F3		
E1E7			A0		
E1E8			FF		
E1E9			FF		

E1EA	CE E1E2	LDX	#M	}	Pointage au debut de la T.V Realiser la Prog- du PIA en (E/S) ainsi que (E/L)
E1ED	7E E1C5	JMP	ADR		

E1F0			F0	}	
E1F1			00		
E1F2			FC		table de verite
E1F3			5F		
E1F4			F3		
E1F5			FA		
E1F6			FF		
E1F7			FF		

E1F8	CE E1F0	LDX	#M	}	Pointage au debut de la T.V Realiser la Prog- du PIA en (E/S) ainsi que (E/L)
E1FB	7E E1C5	JMP	ADR		

E1FE
E1FF
E200
E201
E202
E203
E204
E205

E206	CE E1FE	LDX	#M	}	Pointage au debut de la T.V Realiser la Prog- du PIA en (E/S) ainsi que (E/L)
E209	7E E1C5	JMP	ADR		

E20C			00	}
E20D			FF	

E20E		AA	} table de verite
E20F		FF	
E210		55	
E211		FF	
E212		FF	
E213		F0	

E214	CE E20C	LDX	#M	} Pointage au debut de Pa.T.V
E217	86 39	LDAA	#39	
E219	B7 E1 9F	STAA	M	} Acces aux registres DDRA(B)
E21C	4F	CLRA	A	
E21D	BD E0 9B	JSR	SUB 1	
E220	BD E1 8E	JSR	ADR0	
E223	7E E1 A3	JMP	ADR1	

E226		F0	} table de verite
E227		98	
E228		F8	
E229		44	
E22A		F7	
E22B		23	
E22C		FF	
E22D		67	

E22E	4F	CLRA		} effacement de la position correspondante a l'ins-RTS
E22F	B7 E2 46	STAA	M	
E232	BD E0 9B	JSR	SUB 1	} Programmation du PIA en (E/S)
E235	86 FF	LDAA	#FF	
E237	B7 E4 80	STAA	PIADDRB	} Longueur de Pa (T.V)
E23A	86 67	LDAA	#67	
E23C	B7 E4 82	STAA	PIADDRB	} realiser (E/S) sur PIA
E23F	86 04	LDAA	#04	
E241	BD E0 9B	JSR	SUB 1	} realiser (E/S) sur PIA
E244	C6 04	LDAB	#04	
E246	39	RTS		
E247	CE E2 26	LDX	#M	
E24A	7E E1 67	JMP	M	

E24D		F0	} table de verite
E24E		00	
E24F		F8	
E250		44	
E251		F7	
E252		23	
E253		FF	
E254		FF	

E255	CE E2 4D	LDX	#M	} Pointage au debut de Pa(T.V)	
E258	86 39	LDAA	#39		} Mettre en evidence l'ins-RTS dans le Prog-Principal
E25A	B7 E2 46	STAA	M		

E25D	BD E2 35	JSR	ADR0	Programmation du PIA en E/S
E260	7E E1 63	JMP	ADR1	(E/L) sur PIA

E263			F0	} table de verite
E264			98	
E265			F8	
E266			DC	
E267			F7	
E268			BB	
E269			FF	
E26A			67	

E26B	CE E2 63	LDX	#M	Pointage
E26E	7E E2 58	JMP	M	realiser le test

E271			F6	} table de verite
E272			DB	
E273			FA	
E274			DE	
E275			FF	
E276			E7	
E277			F0	
E278			5A	

E279	CE E2 71	LDX	#M	Pointage
E27C	86 39	LDAA	#39	} RTS dans le programme principal
E27E	B7 E2 46	STAA	M	
E281	86 E7	LDAA	#E7	} donnee qui va etre envoyee sur DDRB
E283	B7 E2 3B	STAA	M	
E286	4F	CLRA		} Acces aux registres DDRA(B) Realisant le test
E287	BD E0 9B	JSR	SUB 1	
E28A	BD E2 35	JSR	M	
E28D	7E E1 63	JMP	M	

E290			F6	} table de verite
E291			C3	
E292			FA	
E293			C6	
E294			FF	
E295			FF	
E296			F0	
E297			42	

E298	CE E2 90	LDX	#M	Pointage
E29B	7E E2 7C	JMP	M	Realiser la programmation du PIA en (E/S) ainsi que (E/L) et chargement par la longueur de la (T.V).

E29E		F7
E29F		49
E2A0		F2
E2A1		45
E2A2		F7
E2A3		45
E2A4		F0
E2A5		49
E2A6		F7
E2A7		47
E2A8		F0
E2A9		4B
E2AA		F5
E2AB		4B
E2AC		F0
E2AD		47

table de verite

E2AE	CE E29E	LDX	#M
E2B1	86 FD	LDAA	#FD
E2B3	B7 E0 F2	STAA	M
E2B6	86 F3	LDAA	F3
E2B8	B7 E0 F7	STAA	M
E2BB	86 08	LDAA	#08
E2BD	B7 E1 01	STAA	M
E2C0	86 39	LDAA	#39
E2C2	B7 E1 02	STAA	M
E2C5	4F	CLRA	4F
E2C6	BDE0 9B	JSR	SUB 1
E2C9	BDE0 F1	JSR	M
E2CC	7EE1 06	JMP	M

Pointage
 Donnee qui va être envoyee sur DDRA.
 Donnee qui va etre envoyee sur DDRB
 Longueur de la (T.V)
 Mettre en evidence l'ins-RTS dans le prog- Principal
 Acces aux registres DDRA(B)
 prog- du PIA en (E/S)
 realiser (E/L) sur PIA

E2CF		C0
E2D0		00
E2D1		D5
E2D2		E6
E2D3		F3
E2D4		55
E2D5		E6
E2D6		B3
E2D7		D9
E2D8		4C
E2D9		CC
E2DA		AA
E2DB		EA
E2DC		19
E2DD		FF
E2DE		FF

table de verite

E2DF	CE E2CF	LDX	#M
E2E2	86 EE	LDAA	#EE
E2E4	B7 E0 F2	STAA	M

Pointage
 Donnee qui va être envoyee sur DDRA

E2E7 86 BB
 E2E9 B7 E0 F7
 E2EC 7E E2 BB

LDAA #BB
 STAA M
 JMP M

} Donnee qui va être
 envoyee sur DDRB
 Branchement au
 Programme Principal
 de test

E2EF
 E2F0
 E2F1
 E2F2
 E2F3
 E2F4
 E2F5
 E2F6
 E2F7
 E2F8
 E2F9
 E2FA
 E2FB
 E2FC
 E2FD
 E2FE
 E2FF
 E300
 E301
 E302

E1
 F0
 FF
 6F
 C0
 80
 C1
 88
 C0
 CC
 C1
 CE
 C0
 AF
 D1
 AF
 D8
 EF
 DD
 EF

table de verite

E303 CE E2 EF
 E306 86 DE
 E308 B7 E0 F2
 E30B 86 1F
 E30D B7 E0 F7
 E310 86 0A
 E312 B7 E1 01
 E315 7E E2 C0

LDX #M
 LDAA #DE
 STAA M
 LDAA #1F
 STAA M
 LDAA #0A
 STAA M
 JMP M

} Pointage
 Donnee qui va être envoyee
 sur DDRA
 } Donnee qui va être envoyee
 sur DDRB
 } longueur de la (T.V)
 stockee en M
 Branchement au Progra-
 -mme Principal de test

E318
 E319
 E31A
 E31B
 E31C
 E31D
 E31E
 E31F
 E320
 E321
 E322
 E323
 E324
 E325

FF
 FF
 C0
 E0
 C3
 E0
 E2
 90
 E3
 98
 E2
 DC
 E3
 DE

table de verite

E326
E327
E328
E329
E32A
E32B

E2
BF
F3
BF
F2
FF

E32C CE E3 18
E32F 86 DE
E331 B7 E0 F2
E334 86 9F
E336 B7 E0 F7
E339 7E E3 10

LDX #M
LDAA #DE
STAA M
LDAA #9F
STAA M
JMP M

Pointage
Donnee qui va être
envoyee sur DDRA
Donnee qui va être
envoyee sur DDRB
Branchement au Prog-
Principal de test

E33C
E33D
E33E
E33F
E340
E341
E342
E343
E344
E345
E346
E347
E348
E349
E34A
E34B

FD
24
F6
18
FF
99
FF
E7
FF
24
FF
24
FF
5A
FF
18

table de verite

E34C CEE33C
E34F 86 FF
E351 B7 E14F
E354 86 C3
E356 B7 E154
E359 86 39
E35B B7 E15F
E35E 86 08
E360 B7 E15E
E363 4F
E364 BDE09B
E367 BD E14E
E36A 7E E163

LDX #M
LDAA #FF
STAA M
LDAA #C3
STAA M
LDAA #39
STAA M
LDAA #08
STAA M
CLRA
JSR SUB1
JSR M
JMP M

Pointage
Donnee qui va être
envoyee sur DDRA
Donnee qui va être
envoyee sur DDRB
Mettre en evidence
l'ins- RTS au Prog-
Principal
Longueur de la (T.V)
Acces aux registres
DORA(B)
Programmation en (E/S) dupIA
realiser (E/L) sur PIA

E36D
E36E
E36F
E370

F0
09
FC
09

E371
 E372
 E373
 E374
 E375
 E376
 E377
 E378
 E379
 E37A
 E37B
 E37C
 E37D
 E37E
 E37F
 E380
 E381
 E382
 E383
 E384
 E385
 E386
 E387
 E388
 E389
 E38A
 E38B
 E38C
 E38D
 E38E
 E38F
 E390
 E391
 E392
 E393
 E394
 E395
 E396
 E397
 E398
 E399
 E39A

FC
 A9
 FC
 09
 FD
 19
 FD
 B9
 FD
 16
 FC
 06
 FC
 A6
 FC
 06
 FE
 46
 FE
 E6
 FE
 49
 FF
 59
 FF
 F9
 FF
 59
 FF
 59
 FF
 F9
 FF
 56
 FF
 56
 FF
 59
 FF
 F9
 FF
 59

T.v sans l'etat
 "TOGGLE"

T.v pour l'etat
 "TOGGLE"

E39B
 E39C
 E39F
 E3A1
 E3A4
 E3A6
 E3A9
 E3AB
 E3AE

4F
 BDE09B
 86 FF
 B7 E4 80
 86 F0
 B7 E4 82
 86 04
 BDE09B
 C6 0F

CLRA
 JSR SUB 1
 LDAA #FF
 STAA PIADDR
 LDAA #F0
 STAA PIADDRB
 LDAA #04
 JSR SUB 1
 LDAB #0F

Acces aux registres
 DDRA(B)
 Programmation du
 PIA en (E/S)
 Acces aux registres
 ORA(B)
 Longueur de la(T.v)

```

E3B0 CE E3 6D LDX # M Pointage au debut de la (TV)
                                sans "TOGGLE"
E3B0 BDE3 C6 ST1 JSR SUB 5 Realiser (E/L) sur PIA
E3B6 26 08 BNE ST3
E3B8 BDE0 A2 JSR SUB 2 Passage au double-Mot suivant
E3BB 26 F6 BNE ST1 Branchement pour l'execution
                                de ce mot
E3BD CE E3 8B LDX # M' Pointage au debut de la
                                (T.V) pour l'etat "TOGGLE"
E3C0 7E E3 D6 JMP M Vers Programme de test
                                de l'etat "TOGGLE"
E3C3 7E E0 B2 ST3 JMP SUB 4 vers subroutine
                                "DEFECT"
    
```

```

E3C6 A6 00 LDAA 0,X
E3C8 B7 E4 80 STAA PIAORA
E3CB A6 01 LDAA 1,X
E3CD B7 E4 82 STAA PIAORB
E3D0 B6 E4 82 LDAA PIAORB
E3D3 A1 01 CMPA 1,X
E3D5 39 RTS
    }
    Soubroutine 5
    realisant
    l'écriture + lecture
    des données.
    
```

Programme de test
de l'etat TOGGLE

```

E3D6 C6 04 LDAB #04 taille de la T.V
E3D8 BDE3 C6 ST1 JSR SUB 5 realiser (E/L) sur PIA
E3DB 26 08 BNE ST 2
E3DD BDE0 A2 JSR SUB 2 Passage au double-Mot
                                suivant
E3E0 26 F6 BNE ST1 Branchement pour realiser
                                l'execution de ce Mot
E3E2 7E E0 A6 ST2 JMP SUB 3 Appel de la subroutine "Good"
E3E5 08 INX
E3E6 08 INX realisation du passage a
E3E7 08 INX l'autre etat; cad a
E3E8 08 INX l'etat "TOGGLE"
E3E9 08 INX
E3EA 08 INX
E3EA 08 INX
E3EA 08 INX
E3ED BDE3 C6 ST4 JSR SUB 5 realiser (E/L) sur PIA
E3F0 26 08 BNE ST3
E3F2 BDE0 A2 JSR SUB 2 Passage au double-Mot suivant
E3F5 26 F6 BNE ST4 branchement pour realiser
                                l'execution de ce Mot
E3F7 7E E0 A6 JMP SUB 3 vers la subroutine "Good"
E3FA 7E E0 B2 JMP SUB 4 vers la subroutine
                                "DEFECT"
    
```

0000
 0001
 0002
 0003
 0004
 0005
 0006
 0007
 0008
 0009
 000A
 000B
 000C
 000D
 000E
 000F
 0010
 0011
 0012
 0013
 0014
 0015
 0016
 0017
 0018
 0019
 001A
 001B
 001C
 001D
 001E
 001F
 0020
 0021
 0022
 0023
 0024
 0025
 0026
 0027
 0028
 0029
 002A
 002B
 002C
 002D
 002E
 002F

F1
 11
 F9
 08
 FB
 0F
 FB
 8F
 FB
 17
 F9
 11
 F9
 91
 F9
 11
 FD
 71
 FD
 F1
 FD
 69
 F9
 09
 F9
 89
 F9
 09
 FF
 6F
 FF
 EF
 FF
 6F
 FF
 6F
 FF
 EF
 FF
 77
 FF
 FF
 77
 FF
 FF
 6F
 FF
 EF
 FF
 6F

table de vérité sans
 l'état "TOGGLE"

table de vérité pour
 l'état "TOGGLE"

0030
 0032

86 E7
 B7 E3 A5

LDAA #E7
 STAA M

Donnée qui va être
 envoyée sur DDRB

0035	86 10	LDAA	#10	} longueur de la (T.V) sans l'etat "TOGGLE"
0037	B7 E3 AF	STAA	M	
003A	86 00	LDAA	#00	
003C	B7 E3 B1	STAA	M	
003F	B7 E3 B2	STAA	M	} Realiser une modification au Niveau du Prog- Principal qui est le pointage a «0000» Pour le cas sans "TOGGLE"
0042	B7 E3 BE	STAA	M	} Realiser une modification au Niveau du Prog- Principal qui est le pointage a «0020» Pour le cas de "TOGGLE"
0045	86 20	LDAA	#20	
0047	B7 E3 BF	STAA	M	
004A	7E E3 9B	JMP	M	} branchement au Programme Principal apres les Modifications necessaires Pour realiser le test

CHAPITRE V

**INTERPRETATION DES
RESULTATS**

INTERPRETATION DES RESULTATS

- Résultats du Logiciel :

La RAM dans laquelle on a inséré la partie logiciel n'a pu prendre que les programmes de 39 circuits intégrés. Ce qui nous a obligé d'utiliser l'autre zone mémoire celle qui va de l'adresse 0000 à 007F, dans laquelle on a pu insérer le programme d'un seul circuit logique. Ce qui augmente le nombre des circuits testables par notre testeur à 40.

Les programmes des CI : 74110 et 74111 n'ont pu être logés, mais on peut utiliser une extension mémoire si on veut éventuellement allonger la liste des circuits intégrés à tester.

On remarque que les 42 C.I. qu'on a utilisé, se divisent en deux catégories : combinatoires et séquentiels. Ces derniers malgré que leur nombre est limité à 9 C.I., leur logiciel a occupé une grande partie au niveau de zone mémoire.

Le logiciel qu'on a élaboré, se limite aux résultats suivants :

- la table de vérité du C.I. a été minimisée au maximum (l'écriture / lecture utilise un même octet pour une donnée ou lieu d'utiliser un octet pour l'écriture et un autre octet pour la lecture.).

- l'initialisation ainsi que la programmation en entrée / sortie a été faite suivant la configuration du C.I. assurant le dialogue KIT 05 - utilisateur.

- le regroupement des c.I testables en plusieurs familles possédant chacune une même programmation du PIA en E/S, a pu assurer un gain en position mémoire.

- le gain en location mémoires est assuré par la méthode adoptée dans notre travail, en évitant au maximum les répétitions de certaines instructions.

Enfin, on a essayé d'adopter une autre méthode qui consiste à générer les données (Table de vérité propre au c.I) par un compteur, puis trouver un algorithme qui permet de convertir cette donnée en une autre, suivant le code de branchement, représentant la même donnée aux entrées (c'est à dire accès sur les instructions pour réaliser le programme de test, qui dans ce cas ne fait pas appel à une table de vérité).

Mais le problème qui se pose dans cette méthode, est que les c.I ne suivent pas le même algorithme, ce qui nous oblige à utiliser une grande capacité mémoire et cela représente un inconvénient pour la programmation.

On a pu faire un essai sur le c.I 74.04, en dégageant les programmes propres aux deux méthodes, puis on a fait une comparaison sur les zones mémoires correspondantes à ces deux méthodes. finalement on a remarqué que la méthode qu'on a adoptée dans notre travail était plus efficace.

Pour terminer, on dira que notre logiciel peut se développer pour être plus performant, car la méthode qu'on a adoptée n'est pas unique, mais elle est bénéfique en pratique.

_ CONCLUSION _

Cette étude aidera à comprendre l'importance de la programmation et l'utilité de tout système à base de microprocesseurs.

Le nombre de circuits à tester peut toute fois être élargi autant qu'on veut; il suffit d'ajouter les paramètres relatifs du nouveau circuit aux différentes tables du programme de contrôle à l'aide d'une extension mémoire.

Néanmoins nous pouvons préciser qu'une étude plus approfondie pourrait être poursuivie dans le but d'améliorer les performances de ce testeur; en construisant un système autonome avec son propre moniteur, clavier et une visualisation permettant un meilleur dialogue avec l'utilisateur.

- Bibliographie -

I - livres :

- Microprocesseurs et Mémoires
- Thomson EFCIS -
Catalogue 1980.
- Microprocesseurs du 6800 au 6809
mode d'interfaçage
- G. Revelin -

II - Manuel :

- MEK 6802 DSE
Microcomputer Evaluation Board user's manual
MOTOROLA INC 1980

III - Thèses :

- Etude d'un micro-ordinateur basé autour du
6802 : le Kit D5 de Motorola
F. Akabi et Y. Kassab
Jan 1982.
- Mise en oeuvre de travaux pratiques sur le
microordinateur : le Kit D5 de Motorola.
- A. Assi -
Juin 1982.