

2002

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT D'ELECTRONIQUE



PROJET DE FIN D'ETUDES

S U J E T

**REALISATION
D'UN TESTEUR DE PISTES
POUR
CARTES CIRCUIT IMPRIME**

Proposé par :

M. et M^{me} HAMAMI

Etudié par :

**SOUAM Morad
HAMMANI Rabah**

Dirigé par :

M. HAMAMI

PROMOTION : JUIN 1986

SOMMAIRE

Chap	TITRE	Page
-	BUT DU PROJET	1
-	INTRODUCTION	2
Chap 1	PARTIE HARDWARE	
I	Présentation du circuit imprimé	3
II	Méthodes de Tests	5
III	Réalisation du Testeur	13
IV	Présentation du Micro-Contrôleur	30
V	Carte UC	51
Chap 2	PARTIE SOFTWARE	
I	Organigramme général du moniteur du testeur	63
II	Organigrammes des différentes parties du moniteur	65
III	Organigrammes des vecteurs d'Interruptions et des sous programmes	86
-	CONCLUSION	131
-	REFERENCES BIBLIOGRAPHIQUES	132

BUT DU PROJET

--:--:--:--:--:--oOo--:--:--:--:--:--

Le but de notre travail est de réaliser un testeur automatique pour détecter les pannes de coupures et de courts-circuits des cartes de circuits imprimés.

Le testeur doit répondre aux exigences suivantes :

- Etre Universel :

Il doit être capable de tester n'importe quel circuit imprimé moyennant certains informations le concernant.

- Etre intelligent :

Il doit pouvoir reconnaître la nature de la panne et l'afficher sur écran. Il devra aussi être capable de faire une impression des résultats sur une imprimante. Les résultats devront être donnés d'une manière très exploitable pour l'opérateur.

Pour satisfaire ces contraintes, notre testeur sera bâti autour d'une carte unité centrale commandée par un micro-contrôleur, le 8031.

C'est pourquoi notre travail sera divisé en 2 parties : une partie HARDWARE et une partie SOFTWARE.

II INTRODUCTION

-----oOo-----

Les défauts de Pistes pouvant survenir sur des cartes à circuit imprimés sont soit des défauts de coupure de pistes soit des défauts de pistes court-circuitées soit les deux sortes de défauts réunis.

Ces défauts peuvent arriver soit pendant l'impression du circuit sur la carte soit lors de la soudure des supports sur la carte.



Ces défauts sont très difficiles et fastidieux à détecter surtout lorsqu'il s'agit de cartes contenant un nombre élevé de Chips donc de liaisons. Les pistes sont alors très rapprochées les unes des autres et tellement fines que s'il y a coupure ou court-circuit on ne pourrait les détecter à l'œil nu.

La détection des défauts avec un ohmètre est la méthode classique utilisée ; celle-ci demande énormément de temps et de patience.

Pour toutes ces raisons il est nécessaire de concevoir et réaliser un testeur de pistes qui puisse réaliser la fonction d'un ohmètre Automatique Autonome et Conversationnel.

De même, le testeur devra s'adapter à n'importe quel carte à tester donc être universel.

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

 A R T I E -HARDWARE
-----oO-----

1) PRESENTATION DU CIRCUIT IMPRIME :

1) Introduction :

Un circuit imprimé est constitué d'une multitude de pistes en cuivre, chaque piste pouvant joindre plusieurs composants implantés sur le circuit imprimé.

La figure 1. nous montre un exemple où 5 circuits ont des broches jointes, vice une seule piste.

Cette piste joint la broche 1 de U1 aux broches 3 et 6 de U2 à la broche 5 de U3, la broche 9 de U4 et la broche 12 de U5.

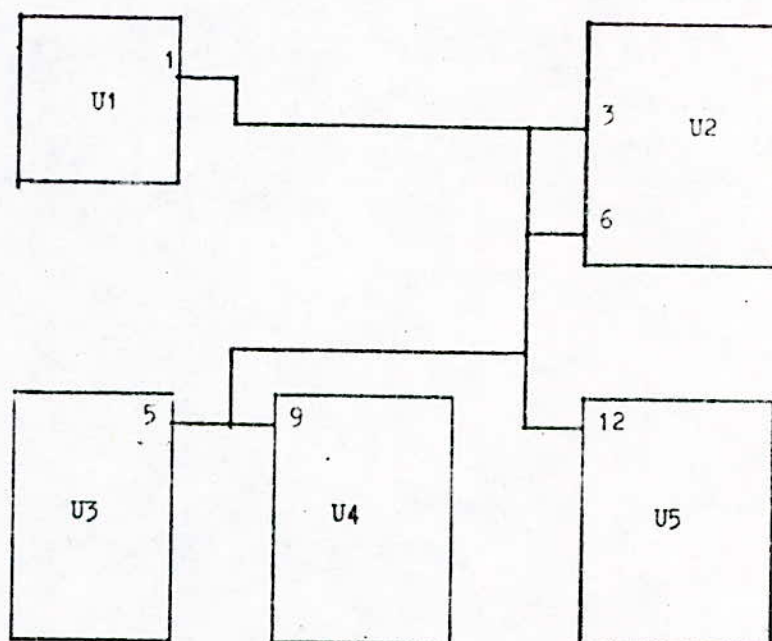


Figure -1-

2) Schématisation d'une piste de circuit imprimé :

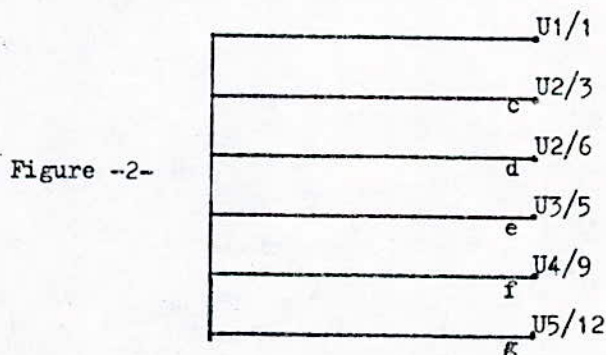


Figure -2-

La piste que l'on vient de voir peut être schématisé suivant la figure -2-. Elle est constituée de 6 pistes secondaires reliées entre elles et qui forment la piste principale.

3) Quelques définitions :

On appellera piste de départ une piste secondaire arbitrairement choisie, et qui débouche sur le reste des pistes secondaires.

On appellera pistes d'arrivées les pistes secondaires autres que la piste de départ et dont elles dérivent.

On appellera un bloc, l'ensemble de pistes d'arrivée issues d'un même piste de départ.

On appellera point de départ, le début d'une piste principale et point d'arrivée, le bout d'une piste secondaire.

Chaque point de départ ou d'arrivée étant relié à une broche d'un quelconque composant.

Ainsi, si on choisi la broche 1 du circuit U1 comme point de départ et les autres broches des autres circuit comme points d'arrivée, on aura la schématisation suivante (fig. 3).

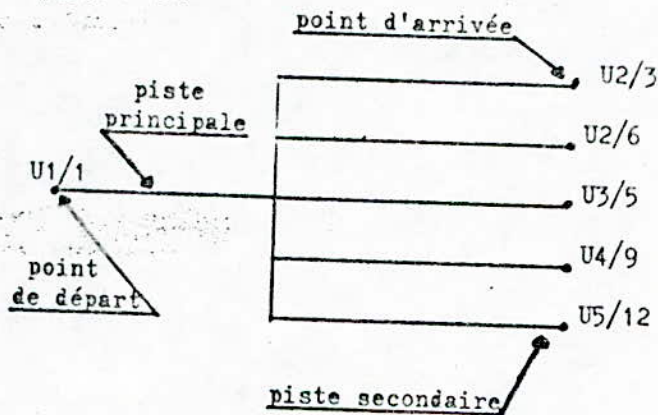


Figure -3-

Dans cette figure il y a :

un seul point de départ qui est la broche (1) du composant U1 et 5 points d'arrivées qui sont :

U2/3, U2/6, U3/5, U4/9 et U5/12.

4) Schématisation du circuit imprimé :

Dans les paragraphes précédents on a vu comment une seule piste peut être schématisée, maintenant on peut généraliser pour tout le circuit imprimé.

La figure 4 montre la schématisation complète du circuit imprimé.

Un circuit imprimé est composé de :

n pistes principales, chaque piste débouche sur un nombre supérieur ou égal à 1 de piste secondaires, le nombre de l'ensemble des pistes secondaires du circuit imprimé est m avec $m \geq n$, le nombre de blocs est bien sur égal au nombre de points de départs.

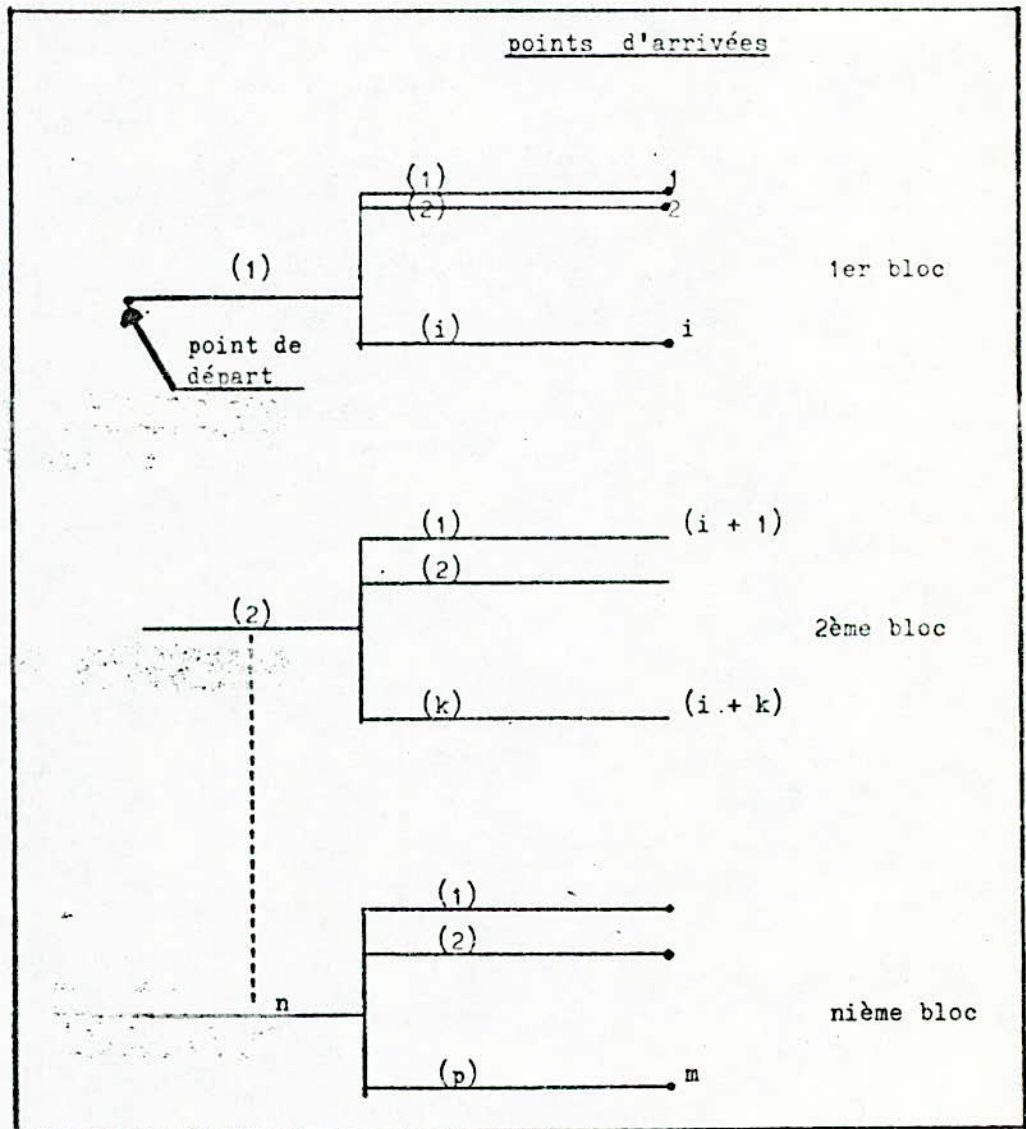


fig: 4

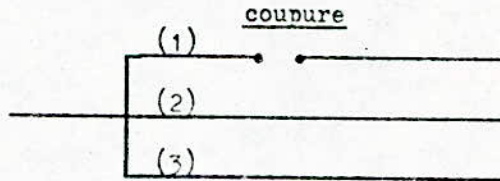
IV) METHODES DE TEST DE PISTES :

1) Présentation des défauts de pistes -

Les défauts éventuels que peut avoir une piste sont de 2 natures :

- Défaut de coupure :

Soit la figure 5 nous montrant un cas très pratique : une piste principale débouchant sur 3 pistes secondaires dont l'une d'elles -piste N°1- présente un défaut de coupure.



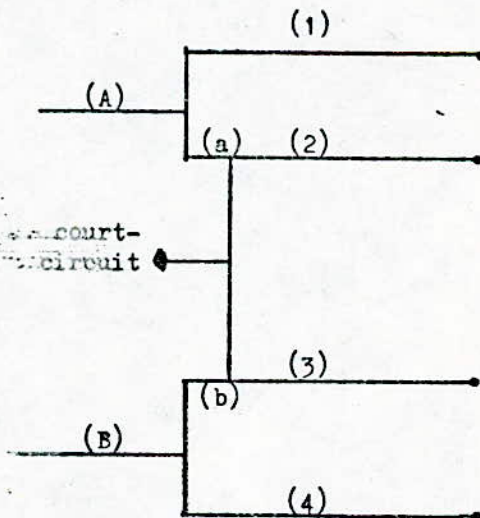
- figure: 5

On va collecter sur les points d'arrivées les informations suivantes :

- Défauts de court-circuit :

Il faut noter d'abord que le défaut de court-circuit ne peut survenir qu'entre deux ou plusieurs pistes de blocs différents, car en effet les pistes d'un même bloc sont déjà reliées entre elles, c'est-à-dire court-circuitées.

Soit donc le schéma sur la figure 6 où l'on a deux pistes principales et chacune avec deux pistes secondaires, la figure 6 démontre aussi l'existence d'un défaut de court-circuit.



- figure 6

2/ Idée de base :

L'idée de base est d'envoyer ^{sur} un seul point de départ l'information Niveau Logique **bas**, des Niveaux Logiques haut sur tous les autres points de départs.

Le testeur devra répondre aux exigences suivantes :

- Si sur une piste où le point de départ est au niveau bas il y a une coupure, on doit pouvoir lire sur le point d'arrivée coupée un Niveau Haut.
- Si un point au Niveau Bas est en court-circuit avec un point au Niveau Haut, le Niveau Bas devra l'emporter sur le Niveau Haut

De cette façon on conclue que :

- Si pour un point d'arrivée appartenant à un bloc du point de départ où a été envoyée l'information Niveau Bas, on trouve un Niveau Haut, on peut affirmer sans équivoque que la liaison entre ce point d'arrivée et son point de départ est coupé.

Donc pour détecter les coupures le testeur devra pouvoir envoyer à tour de rôle un niveau bas sur chaque point de départ et recueillir les signaux à ces points d'arrivées un par un.

- Si pour un point d'arrivée appartenant à un bloc différent du bloc d'arrivée du point de départ du niveau zéro, on a trouvé le Niveau Logique zéro on peut affirmer sans équivoque que ce point d'arrivée est en court-circuit avec le point de départ de l'information zéro donc normalement en court-circuit avec tous les autres points d'arrivées du bloc d'arrivées, du point de départ de l'information zéro.

3/ Développement de l'idée de base :

Question :

Lorsqu'on a trouvé qu'un point X d'un bloc A est en court-circuit avec un point de départ α (voir fig 4).

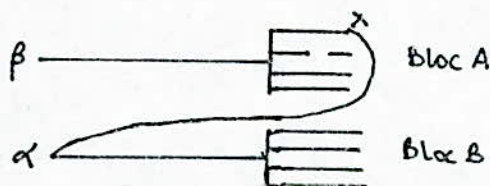


fig: 4

Faut-il tester les autres points du bloc A pour voir s'il sont en court-circuit avec le point α ?

Réponse :

Tester les autres points du bloc A autre que X ne nous donnerait pas d'informations en plus, même si un point du bloc A est coupé de son point de départ, car cette information aurait due être trouvée, si on a envoyé un niveau bas sur le point de départ β où reste à trouver si on n'a pas encore envoyé l'information zéro sur β .

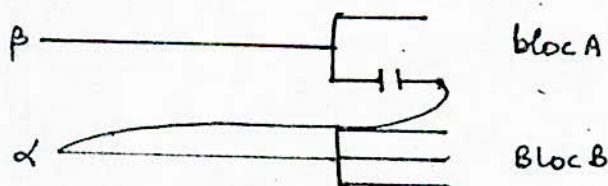
Donc le testeur devra, lorsqu'il a trouvé un défaut de court-circuit dans un bloc donné, ne pas continuer de tester les autres points du bloc mais passer à un autre bloc.

Question :

Faut-il tester ou non des points d'arrivé d'un bloc donné A lors de l'envoi d'un niveau bas sur un point de départ α . Si on a déjà envoyé un niveau bas sur le point de départ B correspondant du bloc A et qu'on ait testé les points du bloc B correspondant au point de départ α ?

Réponse :

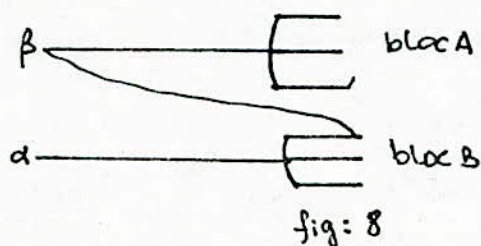
Imaginons le cas suivant :



Lorsqu'on envoi un niveau bas sur β et qu'on teste le bloc B on n'aura pas d'information de court-circuit alors que si on envoi un niveau bas sur α et on teste le bloc A on trouvera un court-circuit.

Donc il faut croiser les tests, c'est-à-dire envoyer toutes les combinaisons possibles à tour de rôle, et tester tous les blocs mais avec les réserves formulées auparavant à savoir que si on trouve un court-circuit dans un bloc donné passer au bloc suivant.

Mais imaginons que la coupure de la figure n'a pas lieu, en croisant les tests, on va comptabiliser 2 fois le même défaut car il sera détecté lors des 2 combinaisons.

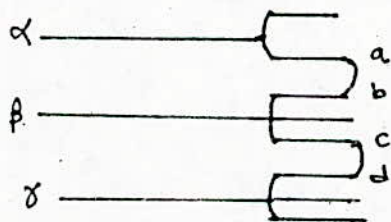


Donc, lorsqu'on revient tester un point d'un bloc A où on a déjà envoyé un niveau bas dans une combinaison précédente et qu'on trouve un court-circuit, il ne faut comptabiliser cette panne que si le point où on a trouvé un court circuit est coupé de son point de départ, coupure que si elle existe aurait été trouvée auparavant puisque le niveau bas sur le point de départ du bloc où on a trouvé un court circuit a déjà été envoyé de même que les tests ont déjà fait.

Donc on peut classer les points de départ et les blocs d'arrivée comme sur la figure ci-dessous et envoyer le niveau bas sur le 1er point de départ, faire les tests puis l'envoyer sur le second etc..., ce qui facilitera la tâche vu ce qui a été dit ^{en} quelques lignes plus haut.

Question :

Supposons qu'on ait les défauts suivants :



En utilisant ce qui a été dit jusqu'à présent on aura les informations suivantes :

- α en court circuit avec b
- α en court circuit avec d
- β en court circuit avec d

Le testeur donnera 3 défauts alors qu'en réalité il n'y en a que deux. Qu'en dire s'il y a plus de 2 défauts ?

Réponse :

La méthode à adopter pour éviter la répétition des informations de court-circuit est la suivante :

lorsqu'on trouve un défaut de court-circuit entre 1 point par exemple β avec d ;

- il faut voir si le point d n'est pas en court-circuit avec un autre point de départ ceci non pas en générant d'autres signaux mais en regardant dans une certaine zone où le testeur est supposé conserver les défauts trouvés comme on le verra plus loin. 2 casse présentent :

. 1er cas : on n'a pas trouvé un point de départ en court-circuit avec d, cela veut dire que valider ce court-circuit ne sera pas une répétition.

. 2ème cas : on trouve que le point d est en court-circuit avec un point de départ α ;

il faut maintenant voir si α est ou n'est pas en court-circuit avec 1 point d'arrivée du bloc d'arrivée du point β

- Si on trouve que α est en court-circuit avec 1 point d'arrivée du point de départ β , si on valide le court-circuit de β avec d ça sera une répétition d'informations, donc il ne faut pas la considérer dans ce cas.

- Si α n'est pas en court-circuit avec 1 point d'arrivée du point de départ β il faut valider le court-circuit de β avec d et il n'y a aucun risque de répétition de la même information.

Cette procédure est à adopter pour n'importe quel court-circuit trouvé, si on commence à faire ça dès le début des tests avec les procédures décrites avant cette dernière. On s'assure du non dédoublement des informations de court-circuit et que l'on détectera toutes les pannes que l'on peut détecter avec un Ohmètre.

Ainsi avec la figure précédente et en utilisant le principe formulé ci-dessus on aura les informations suivantes :

α en court-circuit avec b

α en court-circuit avec d

Donc deux pannes au lieu de trois.

R E S U M E :

Voyons en résumé la façon de procéder dans les tests :

1/ - Envoyer sur le 1er point de départ un niveau bas et sur tous les autres 1 niveau haut.

2/ - Commencer les tests à partir du 1er point d'arrivée. lorsque le point d'arrivée fait partie du bloc d'arrivée du point de départ de l'information zéro, on doit s'attendre à trouver un zéro si on trouve un niveau haut ; il s'agit d'une coupure qu'il faut sauvegarder et passer au point suivant.

Lorsque le point d'arrivée ne fait pas partie du bloc d'arrivée du point de départ de l'information zéro on doit s'attendre à trouver un niveau haut, si on trouve un niveau bas:

- si le point de départ du point d'arrivée lue est classé avant le point de départ où on a envoyé un zéro, il faut voir si la liaison entre le point de départ du point d'arrivée lue et ce dernier est coupée.

. Si c'est le cas faire les tests expliqués une page auparavant pour éviter la répétition et suivant son résultat validé ou non ce court-circuit et s'il est validé on passe au bloc suivant.

. Si ce n'est pas le cas, il ne faut pas valider ce court-circuit.

- Si le point de départ du point d'arrivée lue est classé après le point de départ de l'information zéro faire les tests dévitemment de la répétition, et, suivant les résultats valider ou non la panne de court-circuit, si on la valide on passe au bloc suivant.

3/ - Envoyer le Niveau Bas sur le point de départ suivant et des Niveaux Hauts partout ailleurs et sauter à 2/.

4/ DEFAUTS NECESSITANTS PLUS D'UN TEST

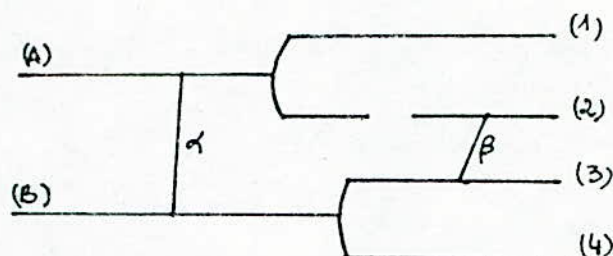


fig: 10

Soit le schéma de la figure 10 montrant un cas où plusieurs défauts s'accroissent, testons cette carte en opérant de la façon expliquée en 2/ on aura l'information suivante :

(A) en court-circuit avec (3) :

donc le testeur ne signale qu'un seul défaut alors qu'il y en a 3.

On supprime, donc, le défaut trouvé c'est-à-dire supprimer le court-circuit α .

Puis on fait repasser la 'carte' dans le testeur on aura l'information suivante :

la liaison entre (A) et (2) est coupée.

il reste encore un défaut que le testeur n'a pas détecté ; on repère puis on repasse la 'carte' dans le testeur on aura alors le dernier défaut soit :

(A) en court-circuit avec (3).

Cet exemple très imaginaire, car pratiquement il a une très faible probabilité d'existence, a nécessité 3 tests pour pouvoir détecter toutes les pannes et un quatrième test pour au moins vérifier que les retouches n'ont pas créés d'autres défauts.

Donc la démarche à adopter sera la suivante :

- faire passer la 'carte' dans le testeur
- repérer les défauts signalés
- refaire les deux étapes précédentes jusqu'à ce que le testeur ne signale aucun défaut.

En pratique, deux tests suffiront pour dépanner une carte complètement, vu les défauts existants en pratique.

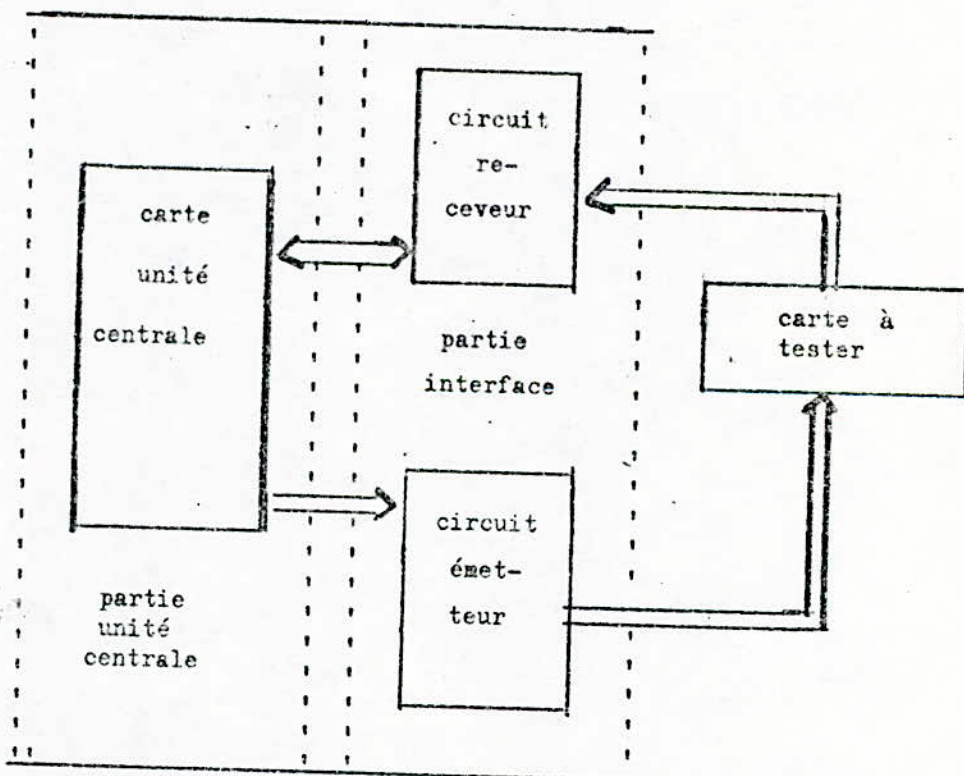
III/ REALISATION DU TESTEUR :

1) Introduction -

Le testeur que nous nous proposons de réaliser est constitué de deux grandes parties, une partie unité centrale et une partie interface.

La partie unité centrale sera réalisée autour d'un micro-contrôleur. Elle sera capable de gérer, l'envoi des informations vers la carte à tester et la réception de celles qui en sortent, de les comparer et de tirer les conclusions qu'il faut.

La partie interface est quant à elle se compose de deux circuits commandés par le micro-contrôleur, un circuit émetteur qui aura pour tâche l'émission des informations et un circuit receveur, C'est lui qui devra récupérer les informations de ladite carte et de les renvoyer vers le contrôleur pour l'analyse. Le schéma synoptique du testeur est le suivant :



2) Le circuit d'interface -

2.1 Circuit d'émission des informations :

Dans le chapitre (II) nous avons vu qu'il était nécessaire d'envoyer des informations sur toutes les pistes de départ du circuit et qu'en suite il fallait changer la combinaison de ces informations.

Si on choisit comme le montre la figure -11-

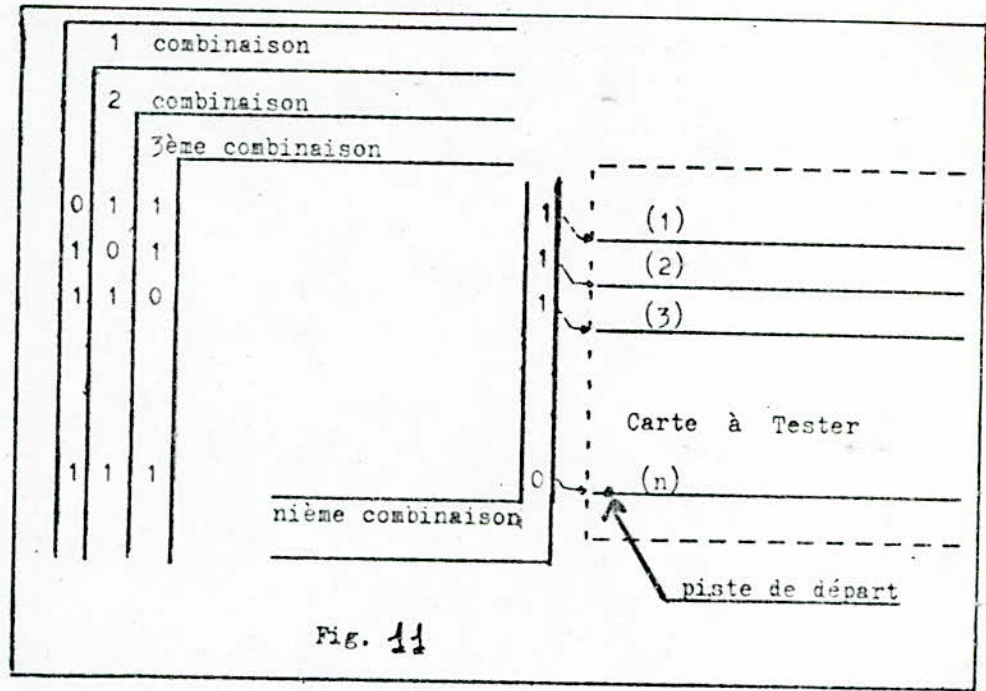


Fig. 11

Pour la première combinaison un "0" sur la première piste et des "1" sur toutes les autres, la deuxième combinaison serait un "0" sur la deuxième piste et des "1" sur toutes les autres et ainsi de suite jusqu'à la nième combinaison où il faudrait envoyer un "0" sur la dernière piste et des "1" sur toutes les autres.

La figure -11- montre aussi que l'opération de changement de combinaison revient à faire des décalages circulaires aux bits de la première combinaison. Sur la figure -12- on peut voir les chronogrammes d'une telle procédure.

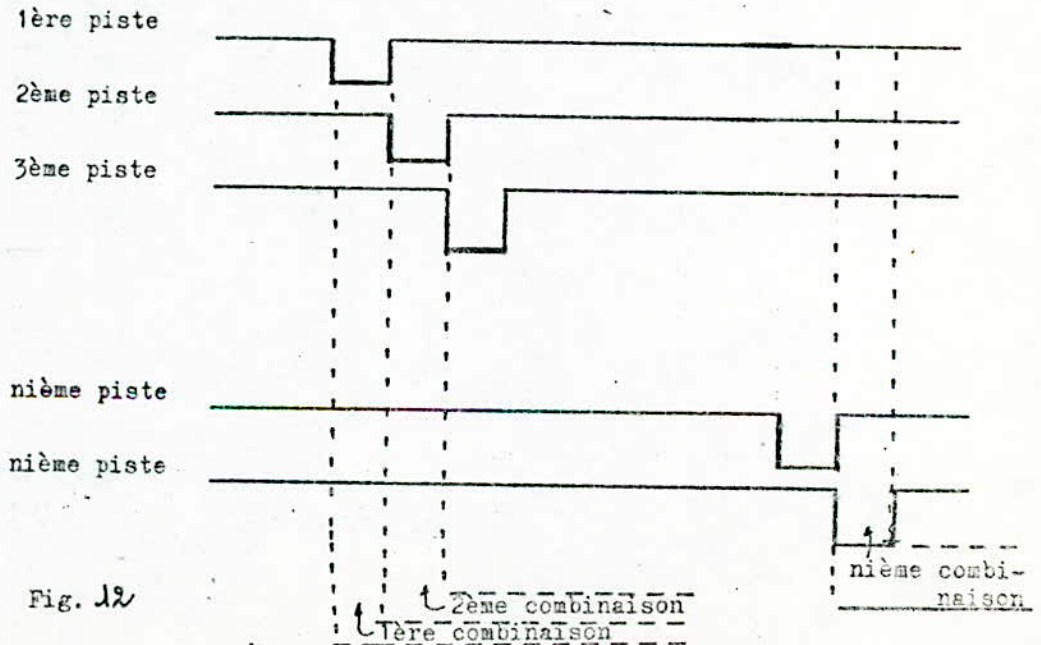


Fig. 12

Etant donné que notre méthode de test est basée sur l'émission et la réception de certaines informations, notre circuit d'émission doit être capable de répondre aux exigences suivantes :

- Envoyer les informations sur les points de départ du circuit imprimé.
- ~~Maintenir~~ ces informations omniprésentes jusqu'à ce que le ~~circuit~~ de réception les ait reçues et traitées.
- Décaler les informations de manière à changer de combinaison.

Le circuit le plus simple et le plus approprié répondant à ces critères est le registre à décalage.

On rappelle qu'il existe trois types de registres à décalage ;

- . registre à chargement sériel *et sorties parallèles*
- . registre à chargement parallèle *et sortie sérielle*
- . registre à chargement parallèle et sorties parallèles.

Le type que nous choisissons est celui qui correspond le plus à notre émetteur est le 1er c'est-à-dire le registre sériel/parallèle.

211) Fonctionnement du registre d'émission : soit le schéma de la figure-13

Un registre sériel/parallèle a ces n sorties connectées aux points de départ d'une carte à tester. La procédure de chargement et d'émission est la suivante:

- on commence par envoyer sur l'entrée du registre le n ème bit, c'est-à-dire celui qui doit être émis sur la n ème piste, à la première impulsion d'horloge le registre mettra ce bit sur Q_1 .
- On envoie le $(n-1)$ ème bit puis on actionne l'horloge, le n ème bit est décalé vers Q_2 et le $(n-1)$ ème est mis sur Q_1 , et on continue comme ça jusqu'à avoir toutes les données sur les sorties du registre, lesquelles données forment la 1ère combinaison d'informations à envoyer sur le circuit imprimé. Pour changer de combinaison il suffirait d'envoyer des impulsions d'horloge pour effectuer des décalages.

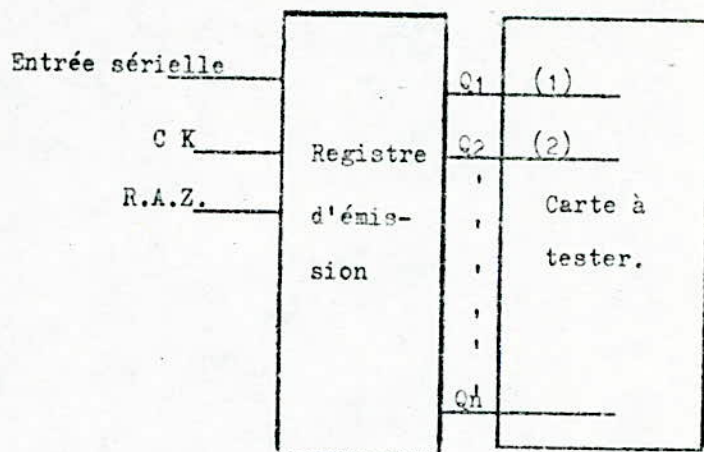


Figure - 13-

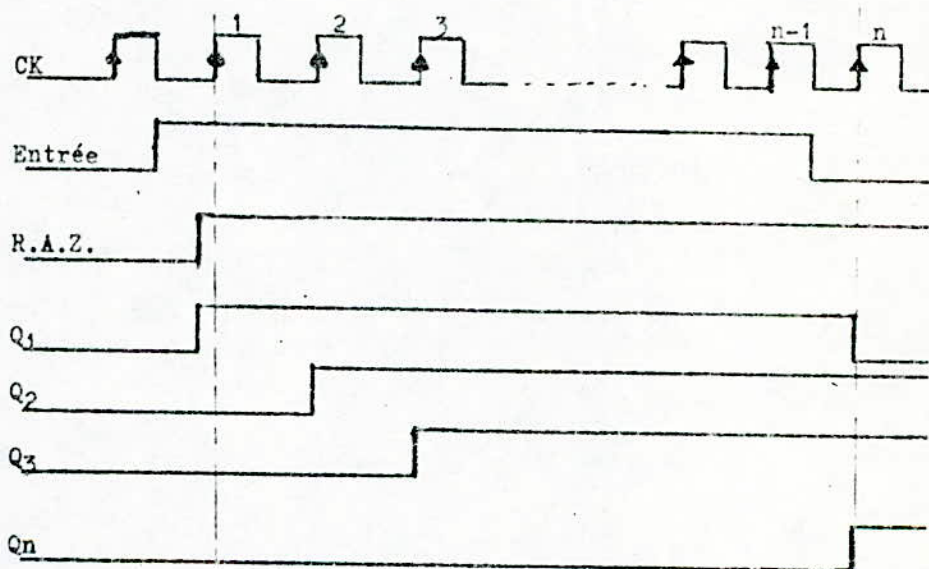


Figure - 14-

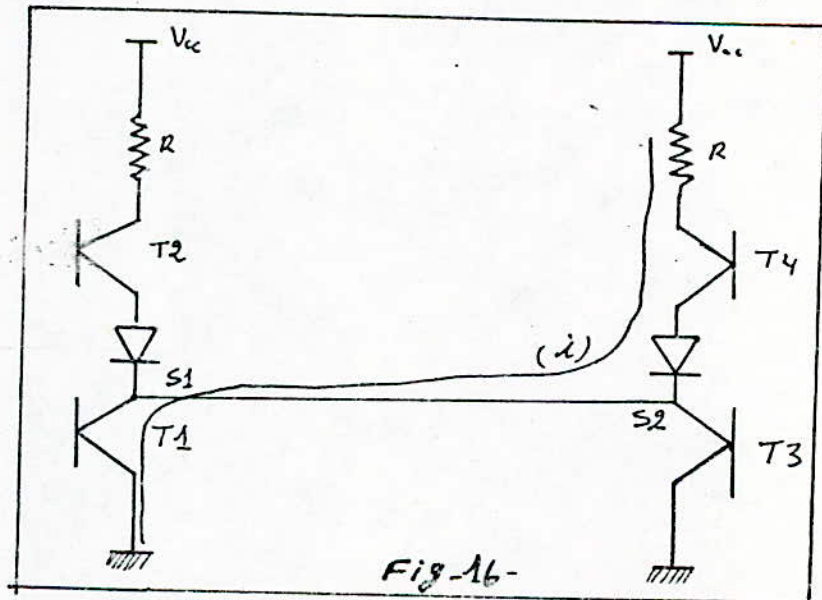
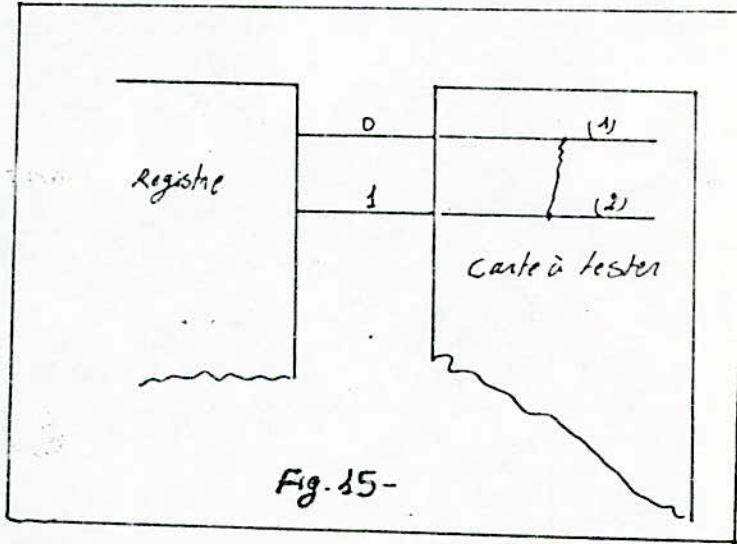
Supposons que l'on ait choisit comme bits de la première conjugaison un "0" sur la sortie "Q₁" et des "1" sur toutes les autres, comme le montre la figure - 14- après une R.A.Z. on maintient l'entrée du registre à l'état haut d'actionner l'horloge pendant (n-1) périodes, puis on met l'entrée à l'état bas et on envoie une dernière impulsion d'horloge -la nième- pour avoir le "0" sur Q₁.

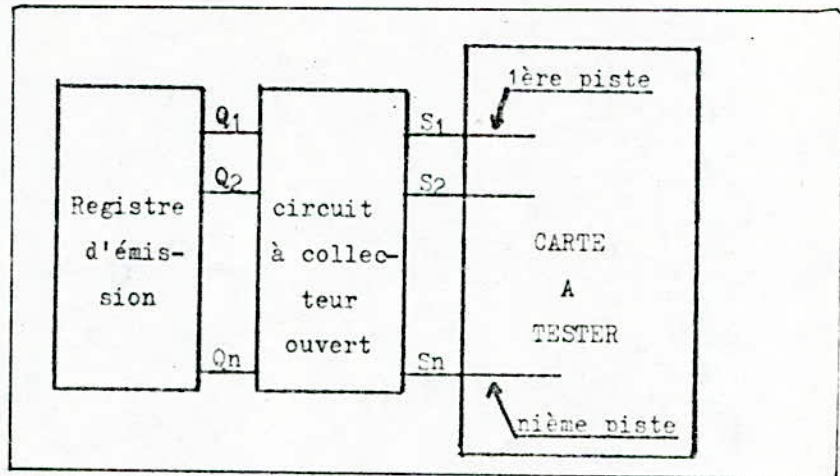
2.12) Nécessité d'un circuit à collecteurs ouverts :

Un registre à décalage étant composé de portes T.T.L., supposons que deux pistes soient en C.C. et que le registre envoie un "0" sur la première et un "1" sur la seconde (voir fig. 15), les deux sorties seront de ce fait reliées, chose qui va conduire à la destruction immédiate du registre.

Rappelons que l'étage de sortie d'une porte T.T.L. est constitué de deux transistors qui sont en permanence dans un état contraire, si l'un est à "1" l'autre est à "0" et vis-versa .

Si par exemple la sortie S_1 est à "0" et la sortie S_2 est à "1" et qu'on les relie ensemble on va réaliser un court circuit entre le V_{cc} et la masse à travers T_4 et T_1 (voir fig. 16). Il faut donc éliminer cet inconvénient en insérant entre le registre et la carte à tester une circuiterie appropriée.





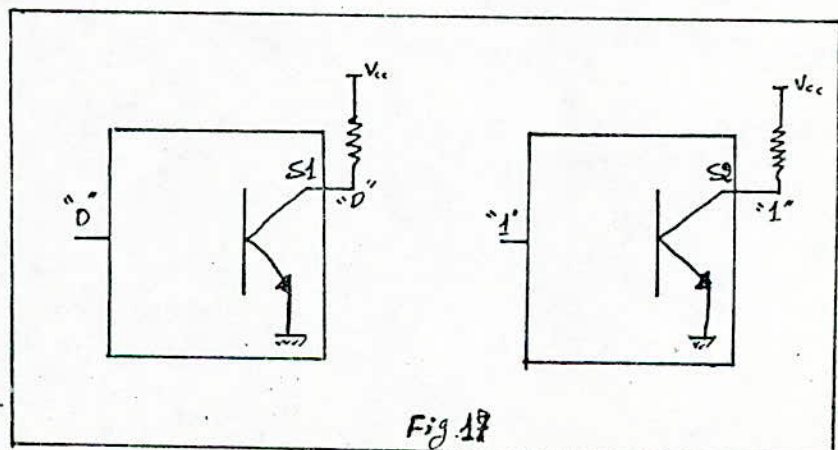
L'un des moyens les plus utilisés pour remédier à ce problème est un circuit ayant pour étage de sortie un transistor à collecteur ouvert.

Supposons que la sortie du circuit à collecteur ouvert suit fidèlement son entrée et ce à condition d'être polarisé par une charge extérieure (fig. 17).

La figure -18- montre comment ce circuit élimine le problème.

La sortie Q_1 étant égale à "0" le transistor T_1 est conducteur et $S_1 = "0"$. La deuxième sortie $Q_2 = "1"$, le transistor T_2 est bloqué et la sortie S_2 devrait être égale à "1"/

Si maintenant on court-circuite les 2 sorties S_2 passe bien sûr au niveau "0" bien que T_2 est toujours bloqué. Tout le courant passera par le collecteur du transistor T_1 qui est maintenant polarisé par une charge $R = R_1 // R_2$ et de ce fait aucun composant ne sera endommagé.



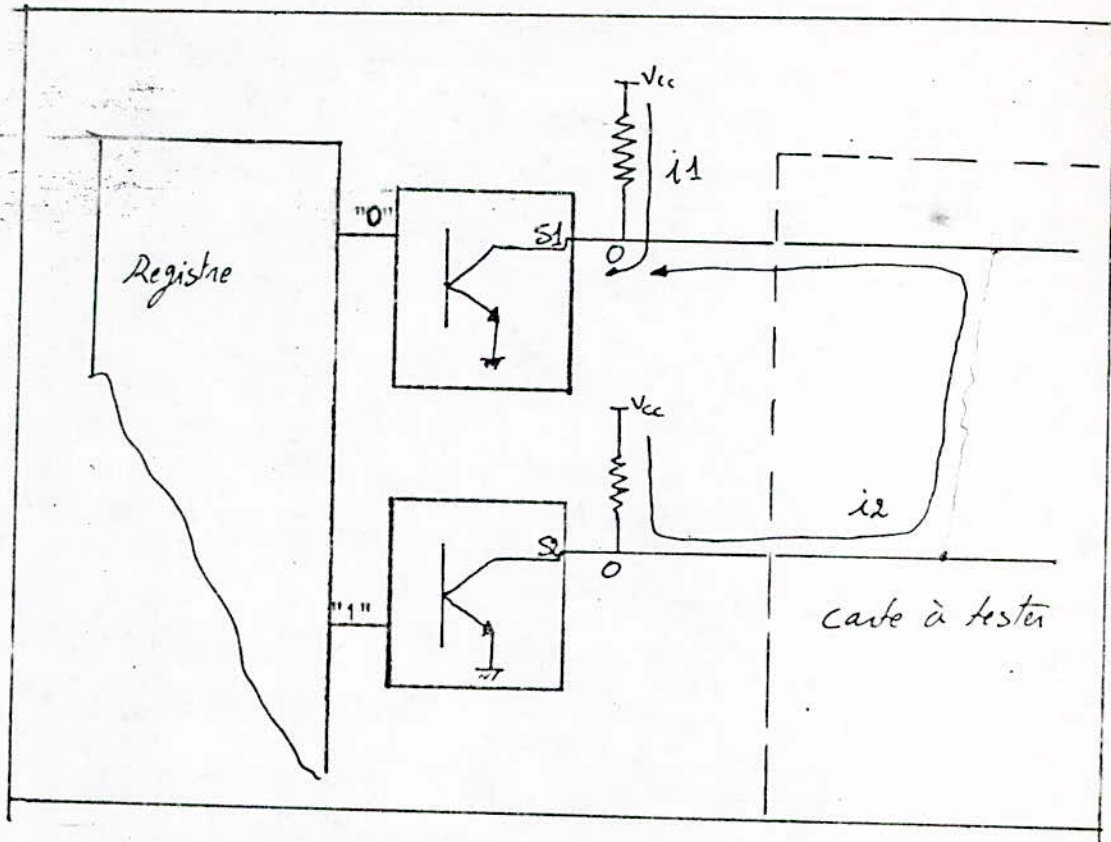


Figure 18

2.1.3) Composants utilisés :

A) - Le Registre à décollage -

On utilise comme registre à décollage la TTL 74 LS 164. C'est un registre à deux entrées sérielles (A et B) et 8 sorties parallèles (Q_A, \dots, Q_H) en plus des entrées d'horloge et de la R.A.Z. comme le montre le schéma du brochage (Fig. 19)

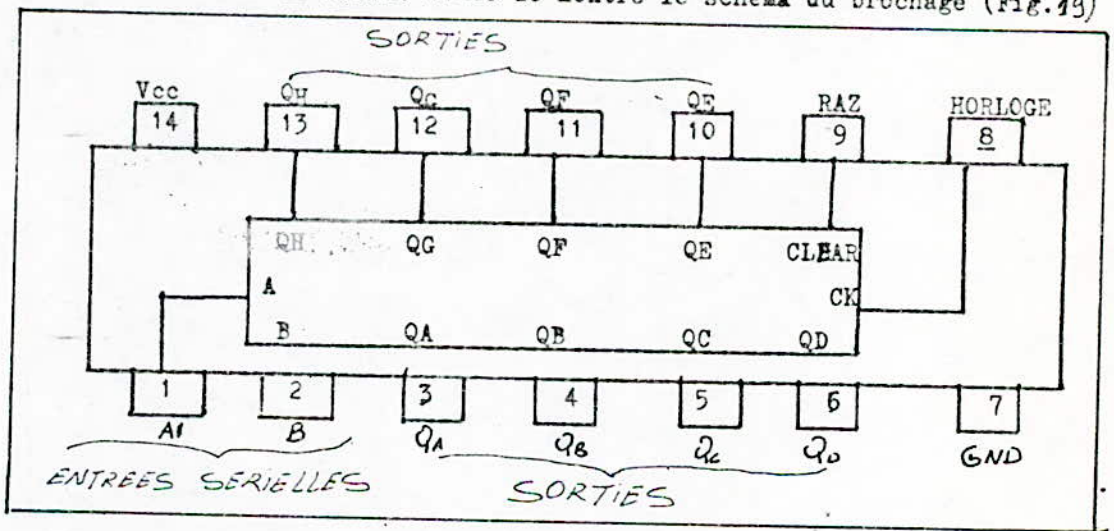


Figure 19

TABLE DE VERITE

ENTREES				SORTIES		
RAZ	HORLOGE	A	B	QA	QB.....	QH
L	X	X	X	L	L	L
H	L	X	X	QAO	QBO	QHO
H	↑	H	H	H	QAN	QEN
H	↑	L	X	L	QAN	QGN
H	↑	X	L	L	QAN	QGN

Figure -20-

Dans les paragraphes précédents on a parlé d'un registre à une seule entrée et à n sorties.

Pour avoir une entrée unique il suffit de relier l'entrée A à l'entrée B, et pour avoir n sorties ; si le nombre n est supérieur à 8 il suffit de relier le nombre qu'il faut de registres en cascade . On relie la dernière sortie du registre R_i c'est-à-dire $Q_H(i)$, à l'entrée du registre R_{i+1} .

La figure montre comment effectuer un tel branchement.

b) Le circuit à collecteur ouvert :

On utilise le T.T.L. SN 74 17, c'est un buffer/Driver comportant 6 portes dont les sorties sont à collecteurs ouverts

c) Buffer des signaux de commande :

Vu que les signaux de commande des registres viennent du contrôleur, ces signaux ne peuvent alimenter qu'une seule charge T.T.L. à la fois et vu notre disposition du circuit d'émission (voir fig.) un buffer est nécessaire pour amplifier ces signaux de commande.

Le buffer utilisé est la T.T.L 74 LS 126, il est constitué de 4 portes avec des sorties, *TRI STATE* .

D) Circuit à double inversion :

Lors du développement de notre hardware, nous avons rencontré un problème au niveau du signal d'horloge du registre car il était mélangé avec un signal parasite qui faussait complètement le bon fonctionnement du registre. On a pu l'éliminer en faisant passer le signal d'horloge par une double inversion. Le circuit utilisé est la T.T.L. 74 LS 04.

2.1.4 Schéma final du circuit d'émission :

Le schéma final du circuit d'émission auquel nous aboutissons est montré sur la figure 30. Sur ce schéma nous pouvons faire les remarques suivantes :

- Les registres ont 4 entrées et trois d'entre-elles reçoivent leurs signaux de commande à partir du Micro contrôleur.
- Les deux entrées du chargement sériel (A et B) de chaque registre sont reliées entre-elles pour former une seule entrée.
- Les registres ont une entrée d'horloge commune.
- La 4ème entrée, la RAZ ("CLEAR") est neutralisée et forcée à 5 volts.
- Les signaux de commande passe pour des amplificateurs (Buffer) car un tel signal ne peut être connecté à plus d'une charge T.T.L.
- Le nombre des buffers, des registres et des buffers à collecteur ouvert est déterminé par le nombre de points de départ de la carte à tester.

2.2) Circuit de réception :

Notre choix du circuit de réception est basé sur le fait qu'il doit lui aussi répondre à certaines contraintes à savoir :

- Recevoir les informations qui arrivent sur les points d'arrivée
- Les garder en mémoire pour les transmettre ensuite vers le contrôleur pour y être traitées.
- Vu le nombre important des informations que notre circuit va recevoir et de la difficulté de les transmettre toutes en même *temps* vers le contrôleur, il doit être capable de les émettre sériellement c'est-à-dire bit par bit.

Etant donné ces critères, notre choix est tombé une autre fois, sur le registre à décalage, mais cette fois à chargement parallèle et sortie sérielle. Le fonctionnement d'un tel registre est très simple. Dès qu'une combinaison est envoyée sur la carte à tester le registre récepteur reçoit celles qui arrivent sur le point d'arrivée, les charge dans ces mémoires internes pour les transmettre bit par bit.

221/ Fonctionnement du registre de réception :

Un registre à chargement parallèle est caractérisé en général par trois entrées de commande : la RAZ (CLEAR), le chargement (PRESET) et l'entrée d'horloge (voir fig. 20₁).

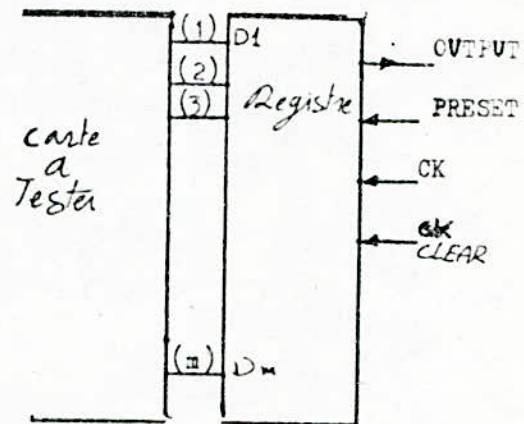


Figure . 20₁

Le fonctionnement du registre est le suivant :

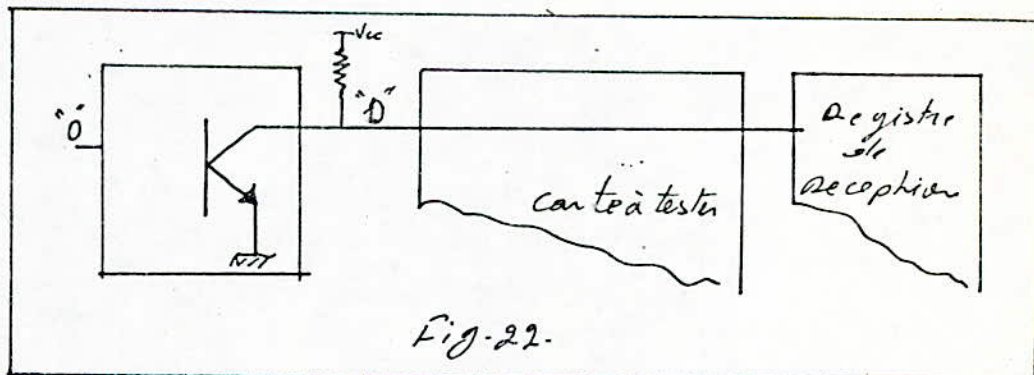
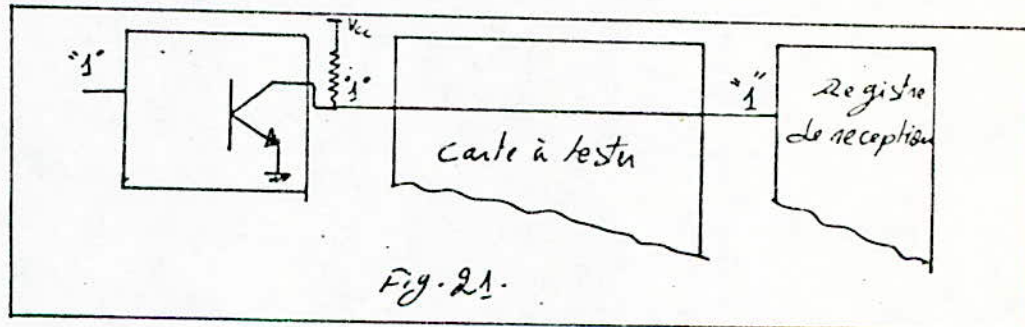
Dès qu'il reçoit une commande sur l'entrée du PRESET le registre charge dans ses mémoires les informations qui se trouvaient sur ses entrées parallèles et en même temps il fait sortir la donnée D1 sur la sortie output, à chaque fois qu'il reçoit une impulsion d'horloge il décale les informations stockées dans ces registres internes et fait sortir en même temps l'une d'entre elles, qui se présentera ainsi sur le output . Ainsi, si le registre peut charger m bits il faudra une impulsion sur le PRESET et $(m-1)$ autres sur l'horloge pour les faire sortir et envoyer vers le contrôleur.

222/ Les résistances de polarisation des collecteurs ouverts :

Nous avons vu lors de la présentation du circuit d'émission que c'était les buffers à collecteurs ouverts qui devaient être reliés directement à la carte à tester. C'est pourquoi ces collecteurs ouverts sont polarisés par des résistances extérieures.

Ainsi, si un "1" est envoyé par le registre vers l'entrée du buffer à collecteur ouvert, son transistor de sortie sera bloqué et la sortie sur son collecteur se mettra en haute impédance. Par contre si le collecteur est polarisé par une résistance il sera à un niveau "1" logique, c'est-à-dire que la sortie du buffer suit son entrée. Si maintenant un "0" est envoyé vers le buffer son transistor de l'étage de sortie devient conducteur et la chute de tension aux bornes de la résistance donne un niveau logique "0" ; donc dans les deux cas la sortie suit fidèlement l'entrée. (voir figs. 21, 22).

Seulement il faut brancher la résistance comme il est montré par la (figure 23), c'est-à-dire après la piste et non pas avant pour la simple raison que si cette piste était coupée et que l'on ai envoyé un "0" sur son point de départ le registre va avoir sur son entrée un "1" grâce à la résistance, par contre si cette résistance était branchée avant la piste à l'entrée du registre il y aurait une haute impédance. Quoique théoriquement un état haute impédance est vu par une entrée T.T.L. comme un état logique "1", la pratique a montré que ce n'était pas toujours vrai, voir (figures 24 et 25)



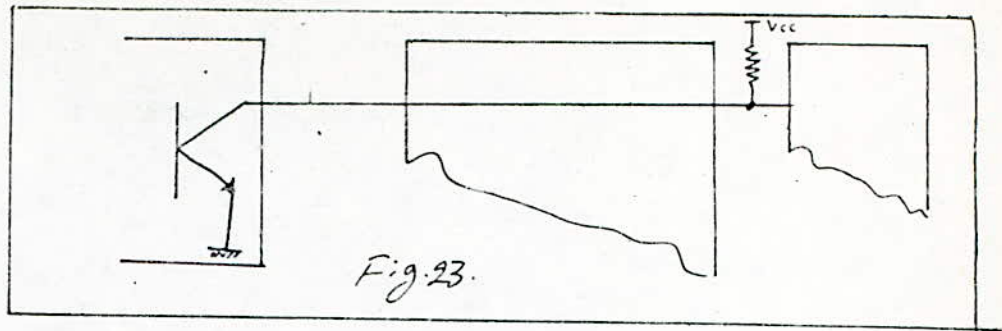


Fig. 23.

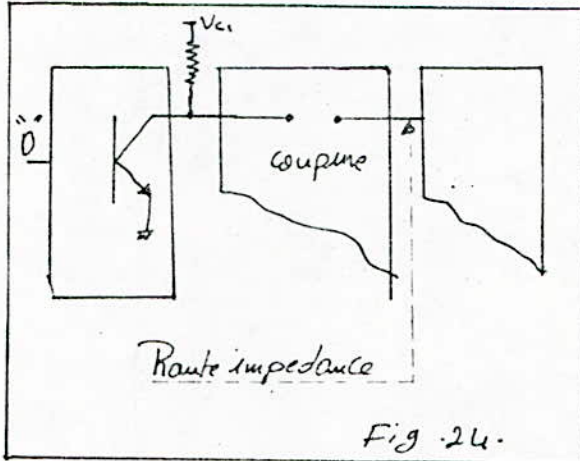


Fig. 24.

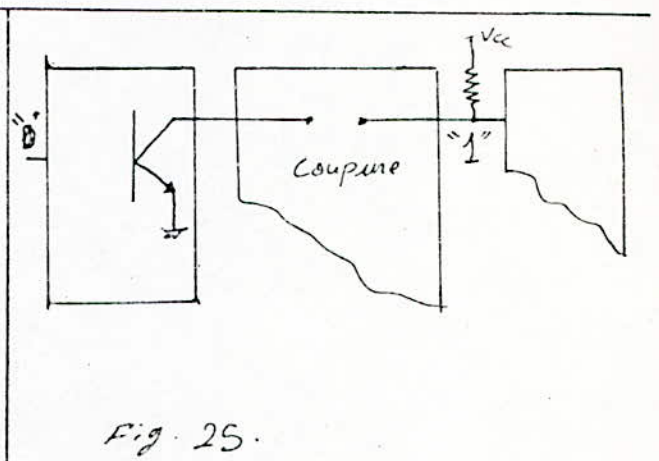


Fig. 25.

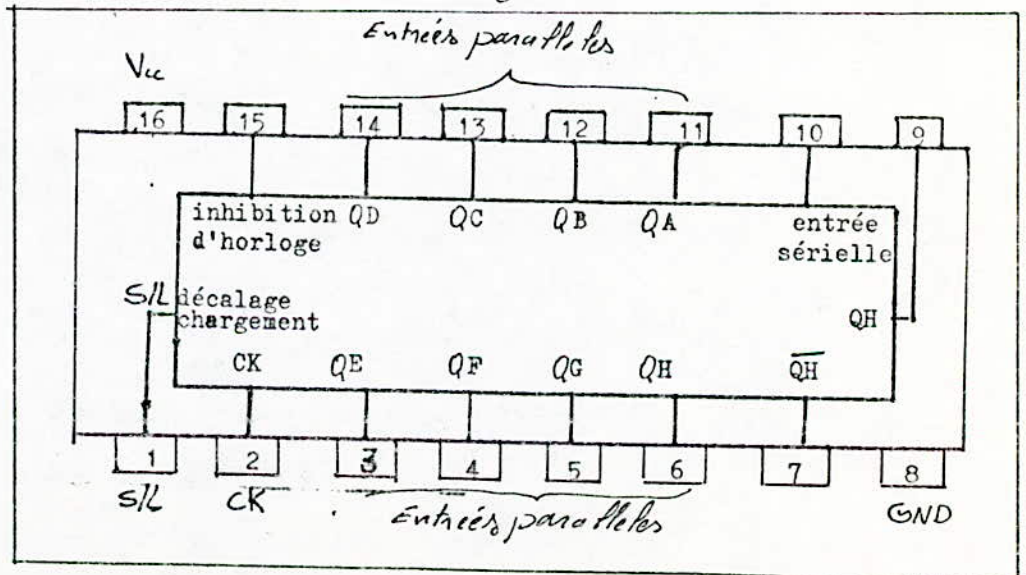
2.2.3 Composants utilisés :

A) - Registre de réception -

Le registre utilisé pour la réception est la T.T.L. SN 74 165.

C'est un registre à décalage 8 bits avec chargement parallèle. Ce registre possède aussi une entrée sérielle qui sera exploitée lors de la réalisation du circuit, en effet pour avoir un registre à n entrées parallèles il faut relier là aussi en cascade le nombre qu'il faut de registre. On reliera la sortie du registre R_i avec l'entrée sérielle du registre R_{i+1} (voir fig. 26)

Fig. 26.



ENTRÉES					Sorties internes	Sorties QH
S/L	CLOCK INHIBIT	CLOCK	SERIEL	PARALLELE		
				A. . . ; H	QA QB	
L	X	X	X	a . . . h	a . . . b	h
H	L	L	X	X	QAO QBO	QHO
H	L	↑	H	X	H QAN	QGN
H	L	↑	L	X	L QAN	QGN
H	H	X	X	X	QAC QBC	QHC

B) Buffer des signaux de commande :

Là aussi il faut amplifier les signaux de commande issus du contrôleur. On utilise le même buffer que pour les signaux de commande du registre d'émission, c'est-à-dire la T.T.L 74 LS 126.

c) Calcul de la résistance :

D'après les fiches signalétiques des composants (voir DATA-BOOK) on peut tirer le tableau suivant :

7417	$\bar{I}_{OH}(\max) [mA]$	$\bar{I}_{OL}(\max) [mA]$	$V_{OL}(\max) [V]$
	250	40	0,4
74LS165	$\bar{I}_{IH}(\max) [mA]$	$\bar{I}_{IL}(\max) [mA]$	$V_{IH}(\min) [V]$
	20	-0,4	2

1er cas : Un "0" est envoyé sur le bloc. soit I le courant (max) admis par le collecteur du buffer 74 17, I_i le courant à travers la résistance R_i et I'_i le courant max sortant du registre r_j .

Supposons que le maximum de point d'arrivée par bloc est n (voir fig. 27)

$$I \leq 40 \text{ mA}, \quad I = \sum_{i=1}^n \bar{I}_i + \sum_{i=1}^n \bar{I}'_i$$

$$I = (I_1 + I_2 + \dots + I_n) + (I'_1 + I'_2 + \dots + I'_n) =$$

$$= \frac{nV_{CC}}{R} + 0,4 n \leq 40 \text{ mA}$$

Si on prend R en k Ω et Vcc = 5V

$$R \geq \frac{5 \times n}{40 - 0,4n}$$

La variation de R en fonction de n est sur la (fig. 28) qui montre que le nombre de points par bloc n doit être tel que $n < 100$, c'est dire que $n \leq 99$.

Si maintenant il y a un court-circuit entre des blocs il faudra que la somme de leurs points d'arrivée ne dépasse pas 99.

2ème cas : Un "1" est envoyé sur le bloc, pour que le registre reconnait le niveau logique "1" il faut que sa tension d'entrée Vih soit telle que $V_{ih} \geq 2V$ c'est-à-dire que la chute de tension à travers la résistance ne doit pas dépasser 3 V/

Soit I le courant max admis par le buffer, Ii le courant passant à travers la résistance Ri et Ii' le courant max admis par le registre, voir (fig. 29)

$$V_{cc} = RI_i + V_r \quad / \quad V_r \geq 2V$$

$$I_i \text{ max} = \frac{I(\text{max})}{n} + I_i'(\text{max})$$

$$V_{cc} - RI_i = V_r \geq 2V \Rightarrow RI_i \leq V_{cc} - 2$$

$$\Rightarrow R \leq \frac{V_{cc} - 2}{\frac{I_i(\text{max})}{n} + I_i'(\text{max})} = \frac{V_{cc} - 2}{\frac{I(\text{max})}{n} + I_i'(\text{max})}$$

$$V_{cc} = 5V, I(\text{max}) = 0,25 \text{ mA}, I_i'(\text{max}) = 20 \mu A$$

$$R \leq \frac{3}{0,02 + \frac{0,25}{n}}$$

$$R \leq \frac{300 \text{ n}}{2n + 25} \quad [k\Omega]$$

Il faut tenir compte du cas le plus défavorable où le bloc présente des coupures sur toutes les pistes sauf une (1) ou bien qu'il ne contient qu'une seule, dans ce cas là $n = 1$.

$$\Rightarrow R \leq 11,11 \text{ k}\Omega$$

donc la valeur de la résistance devra se trouver dans une fourchette telle que :

$$\frac{5n}{40 - 0,4n} \leq R \leq 11 \text{ k}\Omega$$

Le choix de la résistance R est limité par le nombre de points d'arrivée par bloc.

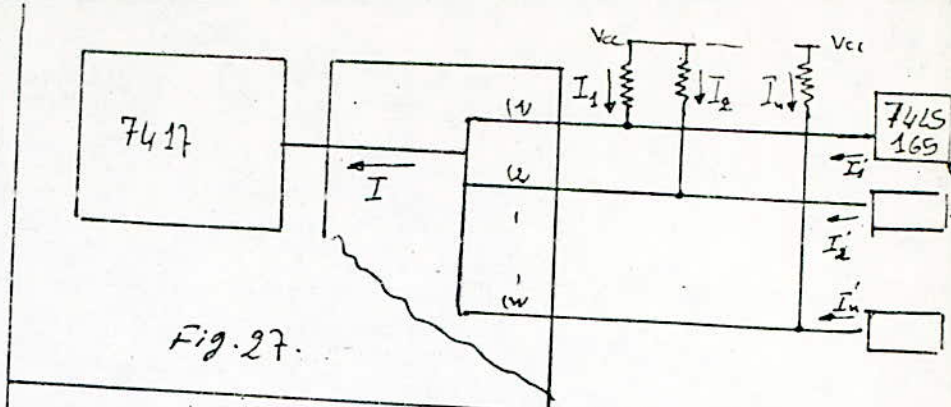


Fig. 27.

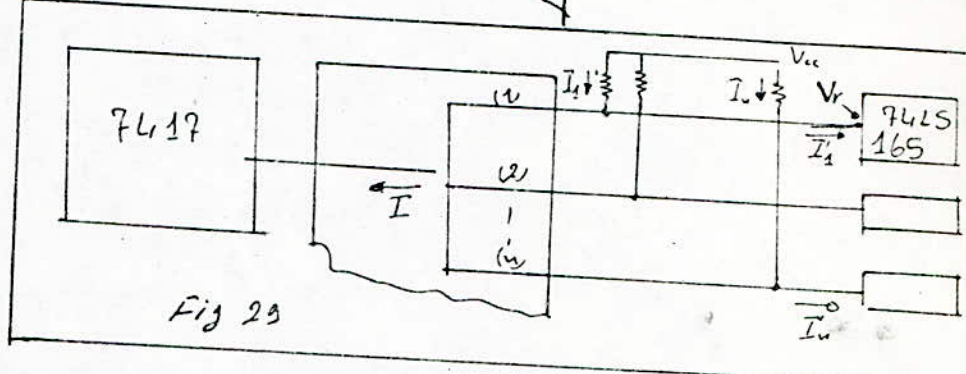


Fig 29

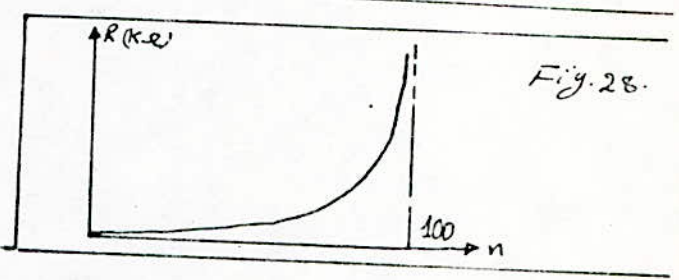


Fig. 28.

2-24) Schéma final du circuit de réception :

Sur ce schéma final on peut faire les remarques suivantes :

- Le registre à décalage possède 3 entrées de commande les, les entrées de chargement et d'horloge reçoivent leurs signaux du contrôleur et la troisième entrée en l'occurrence le ck inhibit est forcée à un état bus.
- Les entrées d'horloge et de chargement sont communes pour tous les registres.
- Les signaux de commande passent par des buffers dont le nombre dépend du nombre des registres à utiliser lesquels dépendent du nombre de points d'arrivée m sur la carte à tester. Fig. 30.

3) Procédure des tests :

Après avoir présenter les circuits d'émission et de réception, nous allons voir comment d'une manière générale les tests seront effectués.

- On charge tout d'abord les registres d'émission avec la première combinaison, à savoir, un "0" sur la sortie Q₁ et des "1" sur les autres sorties. Dès que le chargement est terminé les Datas de la combinaison arrivent sur les entrées des registres de réception.
- On charge le registre de réception en envoyant un signal sur l'entrée (S/L). Le chargement étant effectué, on fait sortir les données sériellement pour les envoyer vers le contrôleur.
- Une fois les données ont été toutes envoyées vers le contrôleur on change de combinaison en envoyant un signal d'horloge.
- On rejoint les mêmes opérations qu'en (2) et (3) jusqu'à avoir transmis toutes les combinaisons.
- Nous avons dit que le décalage devrait circuler, cette condition est obtenue en forçant dès l'envoi de la première combinaison l'entrée à "1".

4) Exemple d'application :

Notre exemple d'application a été réalié pour tester le circuit imprimé d'une carte mémoire, il a la capacité de tester un circuit imprimé ayant jusqu'à 96 points de départ et 640 points d'arrivée. Le schéma du testeur est sur la fig. 30.

La réalisation de cet exemple a nécessité l'utilisation de :

- 12 : registres 74 LS 164
- 80 : registres 74 LS 165
- 4 : buffers 74 LS 126
- 16 : buffers à collecteurs ouverts 74 17
- 1 : 74 LS 04 (inverseurs)
- 43 : réseaux de résistance chacun en contient 15.
la valeur de chaque résistance étant 4,7 K Ω

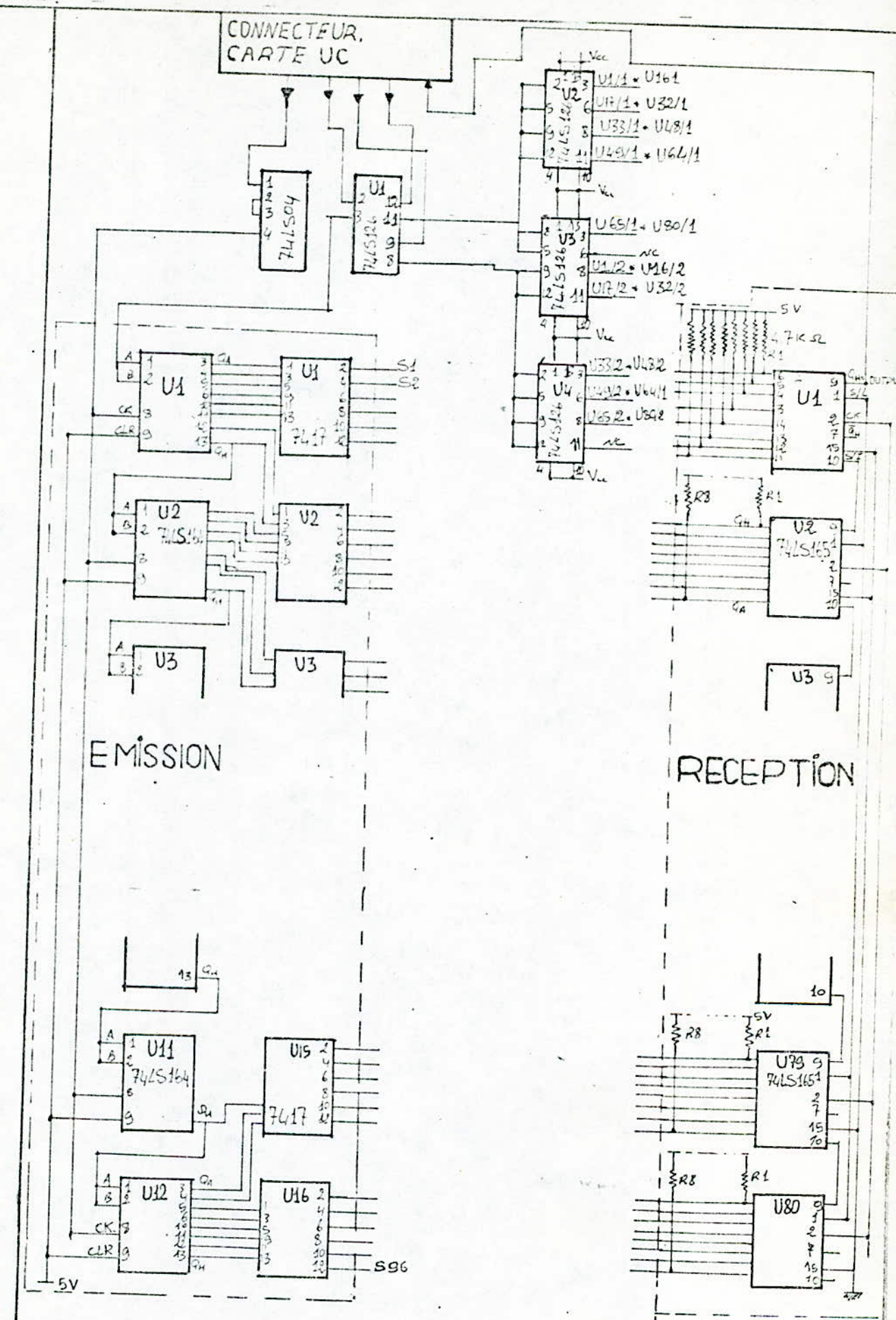


FIG. 30

IV/ PRESENTATION DU MICRO-CONTROLEUR 8031 :

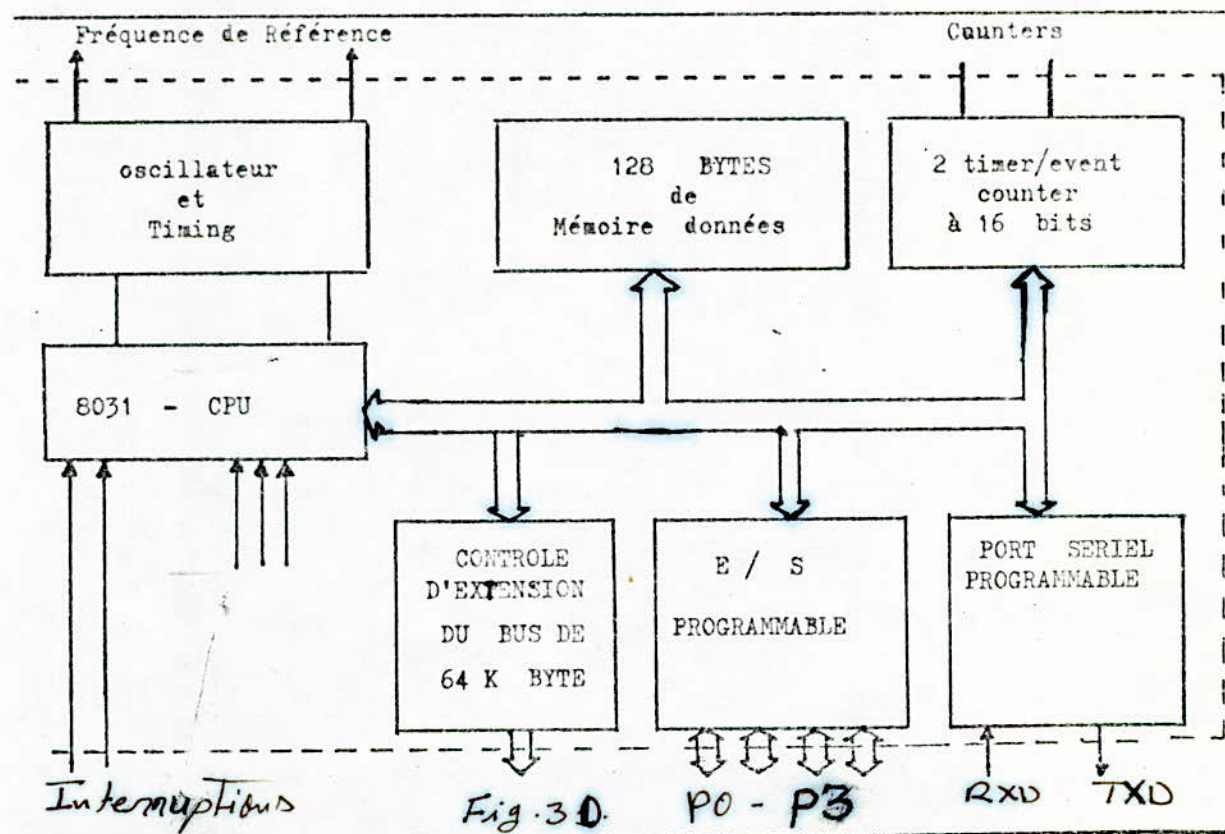
4.1/INTRODUCTION

En une seule puce le Micro-contrôleur 8031 combine :

- le CPU
- 128 bytes de Mémoire de données à écriture lecture volatile
- 32 lignes d'Entrée Sortie
- 2 timer/event conter à 16 bits chacun
- 5 sources d'interruption chacune avec 2 niveaux de priorité possibles.
- 1 port seriel d'Entrée Sortie pouvant servir soit :
 - pour la communication entre 8031
 - pour l'extension en E/S
 - en tant que UART (Universel Asynchrone Receiver Transmitter)
- Enfin 1 oscillateur et des circuits d'horloge intégrés. (Fig.30)

Le CPU du 8031 manipule les opérands dans 4 espaces mémoires à savoir :

- 65 K bytes de Mémoire programme
- 64 K bytes de Mémoire données
- 256 mots de Mémoire Data interne dont 128 sont des Registres Spécialisés, les 128 autres mots forment la RAM interne de données (voir fig.31).



Dans ce qui va suivre, nous allons donner des informations générale sur le 8031 en nous appuyant particulièrement sur les parties utilisées dont la compréhension est nécessaire quant à la compréhension du fonctionnement de la carte UC et du Software.

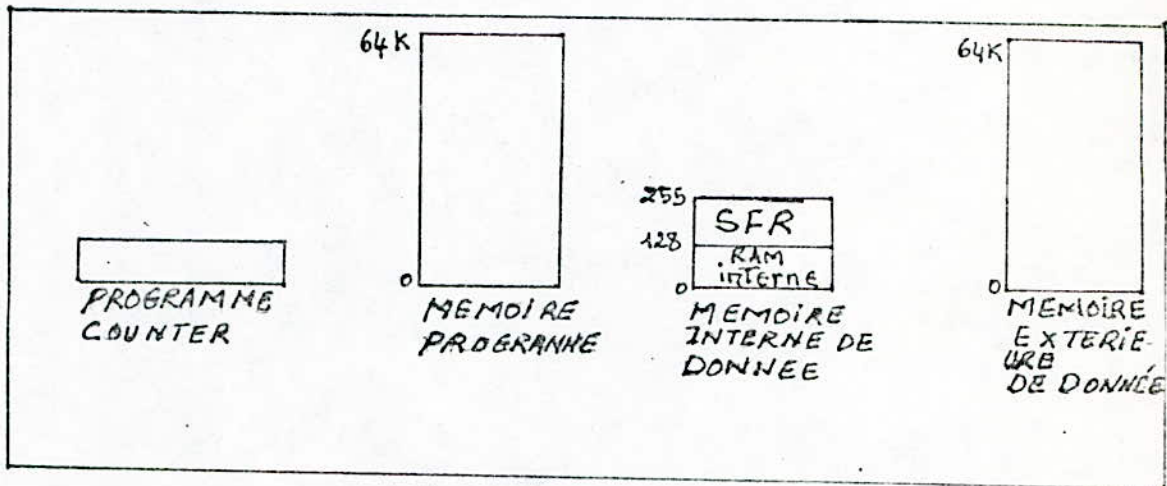


fig: 31

4-2/ ORGANISATION DE LA MÉMOIRE INTERNE :

4-2-1/ Compteur programme :

Le PC à 16 bits contrôle les séquences dans lesquelles les instructions stockées en Mémoire programme sont exécutées.

4-2-2/ Mémoire interne de données :

Celle-ci inclut 128 mots de RAM interne de données et 20 registres spécialisés.

A/ RAM interne de données :

constituée par 4 banks de registres contenant chacun 8 registres plus 128 bits adressables comme bits.

B/ Banks de registres :

Il y a 4 banks de registres chacun contenant 8 registres R7 - R0

b/ 128 bits adressables :

Il y a 128 bits adressables par soft, ils occupent la zone mémoire commençant à l'adresse 32 et se terminant à l'adresse 47 soit 16 mots de l'espace d'adressage de la RAM interne.

C/ File :

Elle peut être n'importe où dans l'espace d'adressage de la RAM interne de données. Elle peut être aussi large que 128 mots soit toute la RAM interne. (fig. 32)

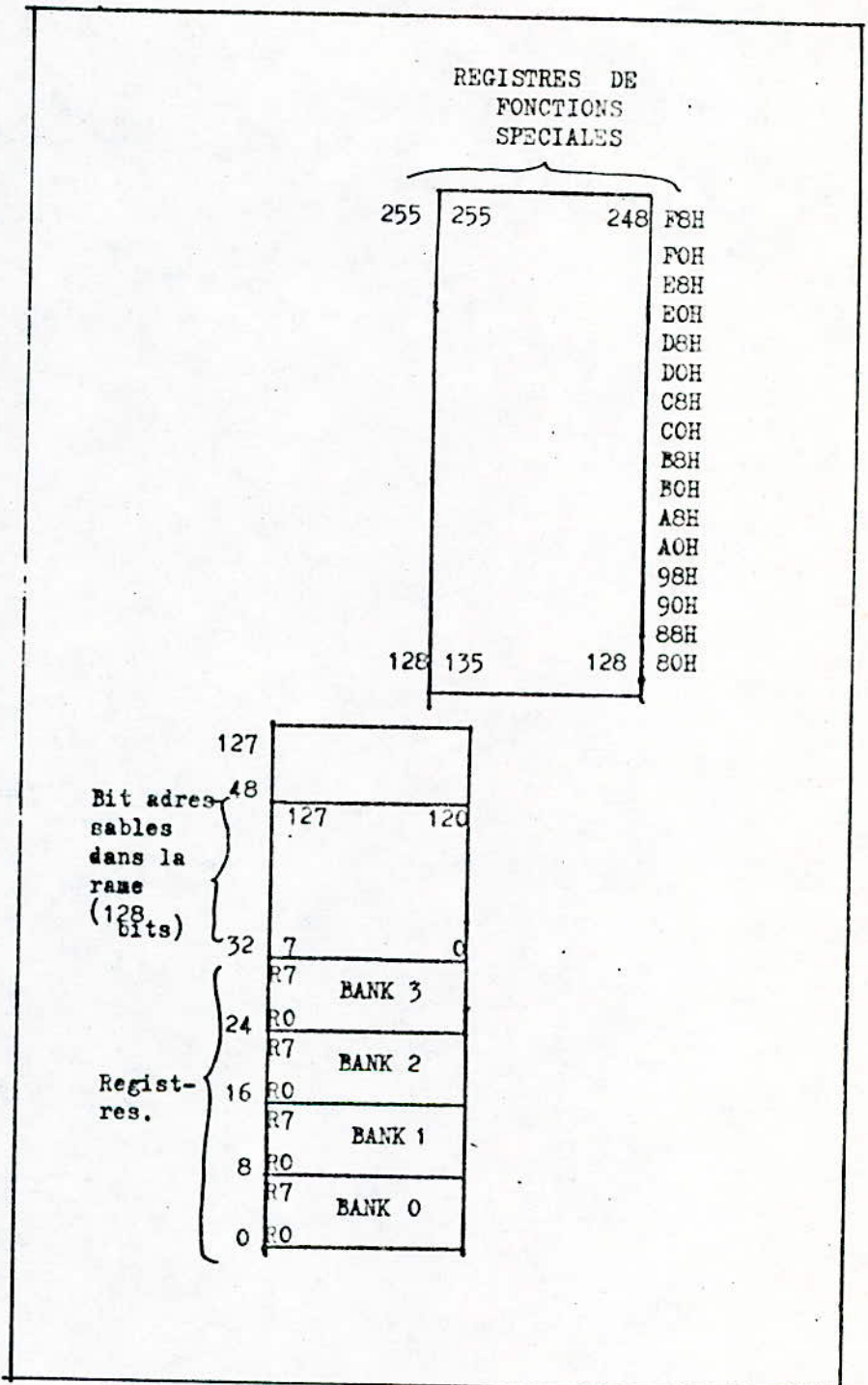


fig: 32

B/ Registres spécialisés :

Ils incluent les registres arithmétiques (A, B, PSW), les pointeurs (SP, DPH, DPL) et les registres qui servent d'interface entre le CPU et les fonctions périphériques du micro-contrôleur même. Il y a aussi 128 bits adressables dans les registres.

— Registre A : c'est l'Accumulateur
— Registre B : Il est dédié à la multiplication et la division et peut aussi bien servir en tant que source ou destinataire pendant toutes les autres opérations il est tout simplement une autre location de l'espace des registres spécialisés.

— Registre d'Etat (PSW : 8 bits) contient :

- . 1 bit de retenue : CY
- . 1 bit de demi-retenu : AC
- . 1 bit drapeau laissé à l'utilisateur : FO
- . 2 bits pour la sélection de bank : RSO et RSI
- . 1 bit de dépassement : OV
- . 1 bit de parité : P

les bits CY, AC, et BV reflètent généralement la position de la dernière opération arithmétique.

Le bit P reflète la parité de l'accumulateur.

— Pointeur de Pile (SP : 8 bits) :

Celui-ci contient l'adresse à laquelle le dernier mot a été poussé dans la pile, c'est aussi l'adresse du prochain mot qui sera pompé,

— Pointeur de données (mot de poids fort) et pointeur de données (mot de poids faible). (DPH et DPL : 8 bits) .

Le registre pointeur de données (DPTR) à 16 bits est l'enchaînement des registres DPH et DPL ; il est utilisé pour lire des données de la Mémoire programme et/ou écrire des données dans la Mémoire Extérieure de données.

— Port 3, Port 2, Port 1, Port 0 (P3, P2, P1, P0 : 8 bits).

Les 4 ports sont munis des 32 lignes d'Entrées/sorties pour l'inter-face avec le monde extérieur. Ils sont tous adressables en tant que mot ou en tant que bits. Les ports 0 et 2 peuvent être utilisés pour l'extension Mémoire ; alors que le port 3 contient des signaux de contrôle spéciaux tel que le Read, Write..., le port 1 ne sert exclusivement d'Entrées/Sorties.

- Registre de priorité des Interruptions - (IPC : 8 bits)
Contient les bits de contrôle pour mettre une interruption au niveau désiré. Un bit mis à 1 donne à l'interruption correspondante un haut niveau de priorité.
- Registre de validation des Interruptions (IEC : 8 bits) ;
Contient le bit de validation d'interruption pour chacune des cinq sources d'interruption. Est aussi inclu, de validation ou d'invalidation globale du système d'interruption
- Registre Mode des Timer/Counter (TMOD : 8 bits)
Contient les bits qui selectent où le timer ou le compteur et l'opération que fera celui qui a été selecté ; ceci pour les 2 timers/compteurs indépendamment l'un de l'autre.
- Registre de contrôle des Timers/Counter (TCON : 8 bits)
Contient les bits de démarrage et d'arrêt des Timers/Counters, les bits de demande d'interruption de ceux-ci.
De même il contient les bits de demande d'interruption des 2 sources extérieures, et les bits de choix du front descendant ou du niveau bas de déclenchement des demandes d'interruption des sources extérieures.
- Timer/Counter 1 (poids fort) Timer/Counter 1 (poids faible) (TH1 et TL1)
Timer/Counter 0 (poids fort) Timer/Counter 0 (poids faible) (TH0 et TL0)
Ce sont les 4 mots alloués pour les 2 Timers/Counters 16 bits. Ceux-ci peuvent être lues ou écrits.
- Registre de Contrôle du port sériel (SCON : 8 bits)
Celui-ci contient des bits qui valident la réception de ce port, selectent le mode d'opération.
- Buffer Sériel de données (SBUF : 8 bits)
Il est utilisé pour tenir le mot entrée ou sortie par le port sériel suivant que le port est entrain de recevoir ou d'émettre des données.

4-3/ PERIPHERIQUES DU 8031 :

4-3-1/ A/ Vue générale du système d'interruption du 8031 -

Le 8031 reconnaît des demandes d'interruption de la part de 5 sources :

- deux demandes de la part de 2 sources extérieures par les entrées INT0 et INT1.

- Une demande de la part de chacun des 2 Timers/Counters
- Une demande de la part du port Sérieel.

Ces interruptions sont toutes de nature matérielle. Chaque interruption se branche à son programme de service. Chacune des cinq interruptions peut être assignée à un des 2 niveaux de priorité et peut être indépendamment validée ou invalidée. De plus toutes les sources peuvent être validées ou invalidées globalement. Chacune des 2 sources extérieures peut être programmée pour le déclenchement soit au front descendant soit au niveau bas ;

La figure suivante donne la signification des bits du registre de validation des INTERRUPTIONS.

(MSB)								(LSB)	
EA	—	—	ES	ET1	EX1	ETO	EXO		
FONCTION								ENABLE	BIT LOCATION
Valide tous les bits de contrôle. MAZ par Soft pour invalider toutes les interruptions.								EA	IE.7
RESERVE								—	IE.6
RESERVE								—	IE.5
VALIDE le bit de contrôle du port sérieel. MA1/MAZ par soft pour valider / invalider les interruptions venant des Flags de TI ou RI.								ES	IE.4
Valide le bit de contrôle TIMER 1. MA1/MAZ par soft pour valider / invalider les interruptions venant du TIMER/COUNTER 1								ET1	IE.3
Valide l'interruption extérieure -1-. MA1/MAZ par Soft pour valider/invalider les interruptions venant de INT1.								EX1	IE.2
Valide le bit de contrôle du TIMER 0 . MA1/MAZ par soft pour valider / invalider les interruptions venant du TIMER/COUNTER 0								ETO	IE.1
Valide l'interruption extérieure -0- MA1/MAZ par soft pour valider / invalider les interruptions venant de INTO.								EXO	IE.0

Le Niveau de priorité d'une source d'interruption est établi comme haut ou bas par la priorité d'un bit dans le registre de priorité des Interruptions (voir fig. ci-dessous).

La mise à 1 du bit associé à l'interruption, programme celle-ci au haut niveau. La priorité de sources multiples arrivant simultanément et assignées au même niveau de priorité est signalisée.

Sur la fig. 34 .si une source d'interruption de bas niveau de priorité est entrain d'être servi, une interruption de haut niveau pourra l'interrompre. Cependant, une source d'interruption ne peut pas interrompre un programme de service d'une interruption de même ou plus haut niveau de priorité. La plupart des bits indicateurs de demande d'interruption sont mis à zéro quand le processus transfère le contrôle à la 1ère instruction du programme de service de l'interruption. C'est le cas des bits correspondants aux 2 sources extérieures et aux 2 Timers. Ce qui n'est pas le cas pour les 2 bits correspondant au Sérieel port, ceux-ci doivent être mis à zéro par le soft dans le programme de service.

(MSB)							(LSB)		
			PS	PT1	PX1	PT0	PX0		
FONCTION								Priorité	Bit Location
Réservé								—	IP.7
Réservé								—	IP.6
Réservé								—	IP. 5
Bit de contrôle de priorité du port sériel . MA1/MAZ par SOFT pour spécifier le niveau haut/bas de l'interruption								PS	IP.4
Bit de contrôle de priorité du Timer 1. MA1/MAZ par SOFT pour spécifier le niveau haut/bas de l'interruption								PT1	IP.3
Bit de contrôle de priorité de INT1. MA1/MAZ par SOFT pour spécifier le niveau haut/bas de l'interruption								PX1	IP.2
Bit de contrôle de priorité du Timer. MA1/MAZ par SOFT pour spécifier le niveau haut/bas de l'interruption.								PT2	IP.1
Bit de contrôle de priorité de INTO. MA1/MAZ par SOFT pour spécifier le niveau haut/bas de l'interruption								PX0	IP.0

Fig:33

SOURCE	D ' INTERRUPTION	Flag de priorité	Niveau de priorité	bit location
Réservé		/		IP.7
Réservé		/		IP.6
Réservé		/		IP.5
PORT SERIEL INTERNE		PS	4 (le plus bas)	IP.4
TIMER/COUNTER INTERNE 1		PT.1	3	IP.3
INTI		PX.1	2	IP.2
TIMER/COUNTER INTERNE 0		PT.0	1	IP.1
I N T O		PX.0	0 (le plus haut)	IP.0

Figure 34

FLAGS DE PRIORITE D'INTERRUPTION

B/ Système d'interruption :

Les adresses de début des programmes de service des interruptions sont données ci-dessous, les 4 premiers ne doivent pas dépasser 8 mots sous peine de se chevaucher, si le programme est plus long que 8 mots il peut être terminé plus loin dans la Mémoire Programme en insérant à la fin un RETI. Démarche de la demande à la validation d'une interruption:

- Une source demande une interruption en positionnant à 1 son bit de demande d'interruption associé, contenu soit dans le TCON soit dans SCON (fg. 35).
- La demande d'interruption sera reconnue si son bit de validation dans le registre de validation d'interruption est à 1 de même que le bit de validation globale.
- Si les conditions, priorité d'interruption, sont respectées à savoir qu'il n'y est pas de programme de service d'une source de même ou de plus haut niveau de priorité. Le contrôleur va établir un appel Hardware au programme de service en poussant le PC et seulement le PC dans la pile à la fin de l'instrument en progrès.

- Quand l'instruction RETI est exécutée le processeur pompe le PC et retourne à l'endroit où il a été interrompu par l'interruption.

SOURCE D'INTERRUPTION	ADRESSE DE DEBUT
Demande EXTERIEURE I N T 0	3 (0003 H)
TIMER/COUNTER Interne 0	11 (000B H)
Demande EXTERIEURE I N T 1	19 (0013 H)
TIMER/COUNTER INTERNE 1	27 (001B H)
PORT SERIEL INTERNE	35 (0023 H)

- LOCATION DES VECTEURS D'INTERRUPTION -

SOURCE D'INTERRUPTION	FLAG	BIT LOCATION
Demande EXTERIEURE INTO	IE.0	TCON.1
TIMER/COUNTER Interne 0	TR.0	TCON.2
Demande EXTERIEURE INT1	TE.1	TCON.3
TIMER/COUNTER INTERNE 1	TR.1	TCON.4
PORT SERIEL INTERNE	TI	TCON.5

Fig: 35 - FLAGS DE DEMANDE D'INTERRUPTION -

Remarque :

Pour ce qui est de notre cas, nous n'utilisons qu'une seule interruption à la fois, soit celle du Sériel Port (en réception) soit celle de la Source Extérieure INTO donc le registre de priorité n'a pas d'intérêt pour nous, de même que les programmes de Service des 3 sources restantes ne contiennent rien puisque elles sont invalidées dans le registre de valisation d'interruption.

4-3-2/4 Vue Générale du Système d'E/S du 8031 :

Le 8031 possède des instructions qui traitent ses 32 lignes d'E/S soit comme 32 lignes de bits adressables, soit comme 4 ports de 8 bits parallèles adressables.

Les ports 0, 2 et 3 assument d'autres fonctions :

- Le Porto est un ^{bus} ~~bit~~ multiplexé entre l'adresse mot de poids faible et la donnée lors de l'extension du 8031 avec les Mémoires et Périphériques.
- Le Port 2 contient l'adresse de poids fort lorsque le 8031 est utilisé avec une mémoire programme extérieure ce qui est toujours le cas ou avec plus de 256 bytes de mémoire extérieure de données.
- Port 3 : peut être configuré individuellement pour :
 - . les 2 Entrées, de demande d'interruption extérieure (INTI, INT1)
 - . les 2 Entrées des Timers/Counters (T0 et T1)
 - . d'Entrée de réception et la Sortie d'émission du Port Série (RXD et TXD)
 - . Générer les signaux de contrôle pour la lecture ou l'écriture dans une Mémoire Extérieure de données (RD, WR)
 - . Générer le signal de lecture dans la ou les mémoire(s) programme extérieure(s) (PSEN)
- La génération ou l'utilisation alternative d'une fonction sur le Port 3 est faite automatiquement par le 8031 aussi longtemps que la pin utilisée est configurée en Entrée.

B/ Bus Du Microprocesseur :

Le bus multiplexé adresse donnée est un interface compatible avec les Mémoires Standard MCS 80 tq les EPROMS 27161, 2732A, les RAMS 2114A, 21283. Lors d'un accès à une Mémoire Externe l'adresse de poids fort est émise à travers le port 2 et l'adresse de poids faible par le Porto. Le signal ALE (Adress latch Enable) sert à échantillonner l'adresse dans un latch extérieur, le signal PSEN (Program Store Enable) sert à valider la sortie d'une mémoire extérieure au port 0 pendant une lecture dans l'espace d'adressage de la Mémoire Programme.

Quand une instruction MOVX est exécutée le port 3 génère automatiquement les signaux RD si c'est une lecture WR si c'est une écriture dans l'espace d'adressage de la Mémoire Extérieure de données.

Le Port 0 émet l'adresse et la donnée à l'espace Mémoire Extérieure à travers un étage pu~~C~~ pull driver qui peut piloter au plus 2 charges TTL.

A la fin du cycle de lecture/écriture du bus le porto est automatiquement reprogrammé à l'état haute impédance et le port 2 est retourné à l'état dans lequel il était avant le cycle du bus.

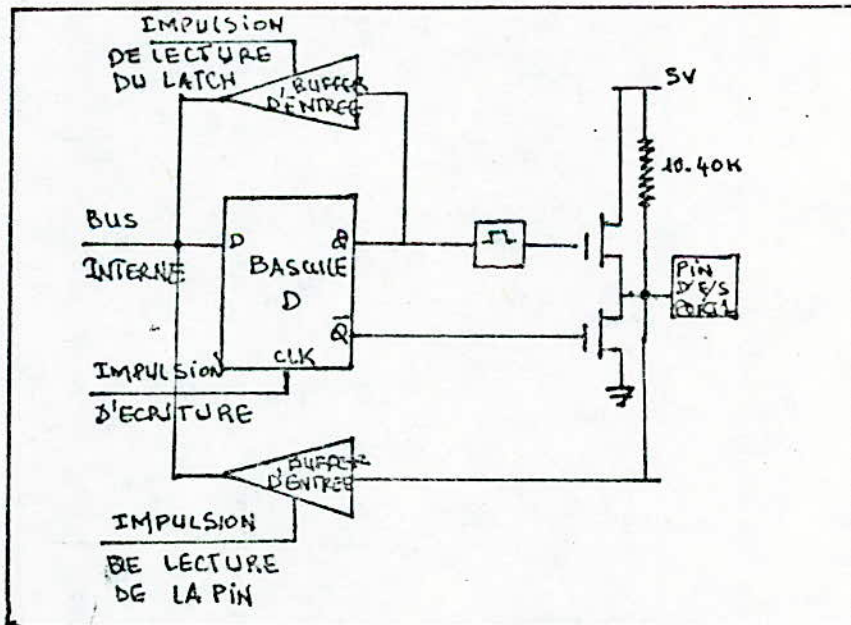
C/ Le Port 1 et ses Pins :

Le Port 1 est un Port Quasi-bidirectionnel, chaque bit du Port est un buffer quasi-bidirectionnel qui a une résistance interne de 10 à 40 K Ω connecté entre la Pin et le Vcc.

Après une Reset le Micro computer programme chaque Pin comme une Entrée en écrivant un (1) au latch ; si un zéro est écrit plus tard au latch (Entrée D) le "Bit" devient configuré en sortie et il débitera continuellement du courant. Chaque Pin configurée en sortie sera configurée en entrée si on écrit un (1) à l'Entrée D, ainsi le circuit se trouvant entre la sortie A et le FET driver sera "validé" et fournira une impulsion pendant 2 périodes de l'oscillateur ce qui déblocuera le driver qui va fournir du courant pendant 2 périodes de l'oscillateur, ce qui empêche le Driver d'un bit programmé en entrée de débiter du courant dans le circuit qui émet des niveaux en entrée ainsi la résistance interne de 10 K à 40 K sert à maintenir le niveau sur la pin à un niveau haut TTL.

De même, il existe 2 sortes d'instruction de lecture de port :

- des instructions qui lisent le niveau sur la Pin
- des instructions qui lisent le niveau sur le latch puis écrivent de nouveau sur le latch (Read-Modify-Write).



Les premières serviront donc après avoir programmé la Fin et Entrée, grâce aux secondes instructions à lire le niveau en Entrée.

Les deuxièmes serviront soit à changer de mode entrée ou sortie soit d'envoyer en sortie des niveaux haut ou bas.

D/ Accès à la Mémoire Extérieure :

- Chronogramme des cycles du bus -

4/ Séquence de lecture dans la mémoire programme

Chaque cycle de lecture dans la mémoire programme consiste en 6 périodes de l'oscillateur. Elles sont référées sur la figure N° 36 par T1; T2, T3, T4, T5 et T6.

L'adresse est émise par le processeur pendant T3. Le transfert de données se produit sur le bus pendant T5 et T6 et la période T1 du cycle suivant.

Le cycle de lecture débute pendant T2 avec le front montant du signal ALE 1. Le front descendant de l'ALE 2 est utilisé pour échantillonner l'adresse qui est présente sur le bus à ce moment là 3 dans un circuit extérieur ayant besoin de l'adresse poids faible. A T5 les port 0 passent à l'état haute impédance 4. Le signal de contrôle de lecture mémoire PSEN 5 est activé pendant T5.

Le PSEN fait que le circuit adressé valide ses sorties pour sortir une donnée en effet un peu plus tard le bus du micro-processeur valide la donnée vers le micro-processeur 6.

Quand le PSEN retourne au niveau haut 7, le circuit adressé met ses sorties de données à l'état haute impédance abordant le bus du contrôleur 8.

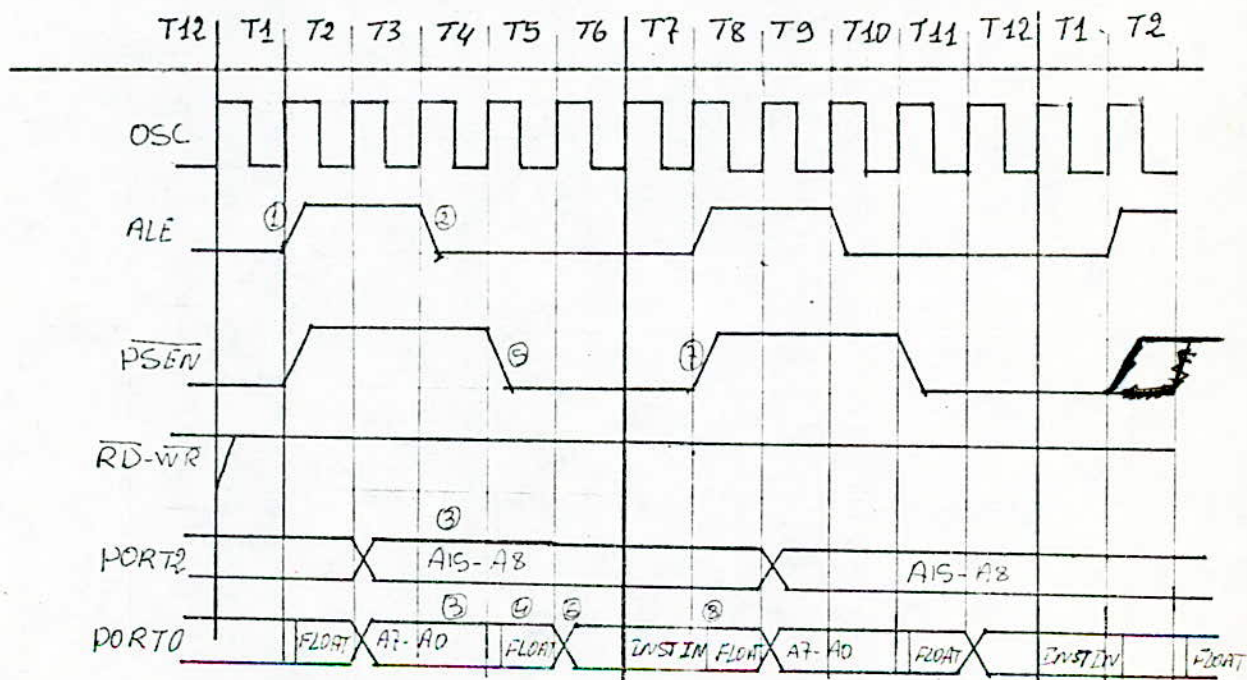


FIG. 36

2/ Séquence de lecture dans une Mémoire de Données :

Chaque cycle de bus de Mémoire Externe de données consiste en 12 périodes de l'oscillateur de T1 à T12 sur la Fig. 37 les adresses sont émises di processeur pendant T3, le transfert de données sur le bus a lieu pendant T7 à T12. T5 à T6 est la période pendant laquelle la direction du bus est changée pour une opération de lecture. Le cycle de lecture commence à partir de T2 avec le front montant de l'ALE 1. Le front descendant de l'ALE 2 est utilisé pour échantillonner l'adresse qui est présente sur le bus à ce moment 3 dans un circuit extérieur dans notre carte micro, le circuit en question sera un ensemble de 8 bascules. A T5 l'adresse disparaît du bus et ce dernier prend l'état de haute impédance 4. Le signal RD est activé pendant T7.5. RD va valider donc les sorties du ou des circuits adressés. Un moment après le bus du processeur se mettra en Etat de recevoir des données 6. Quand le signal RD retourne au niveau haut, le circuit adressé va donc abandonner le bus de niveau 8. (fig.).

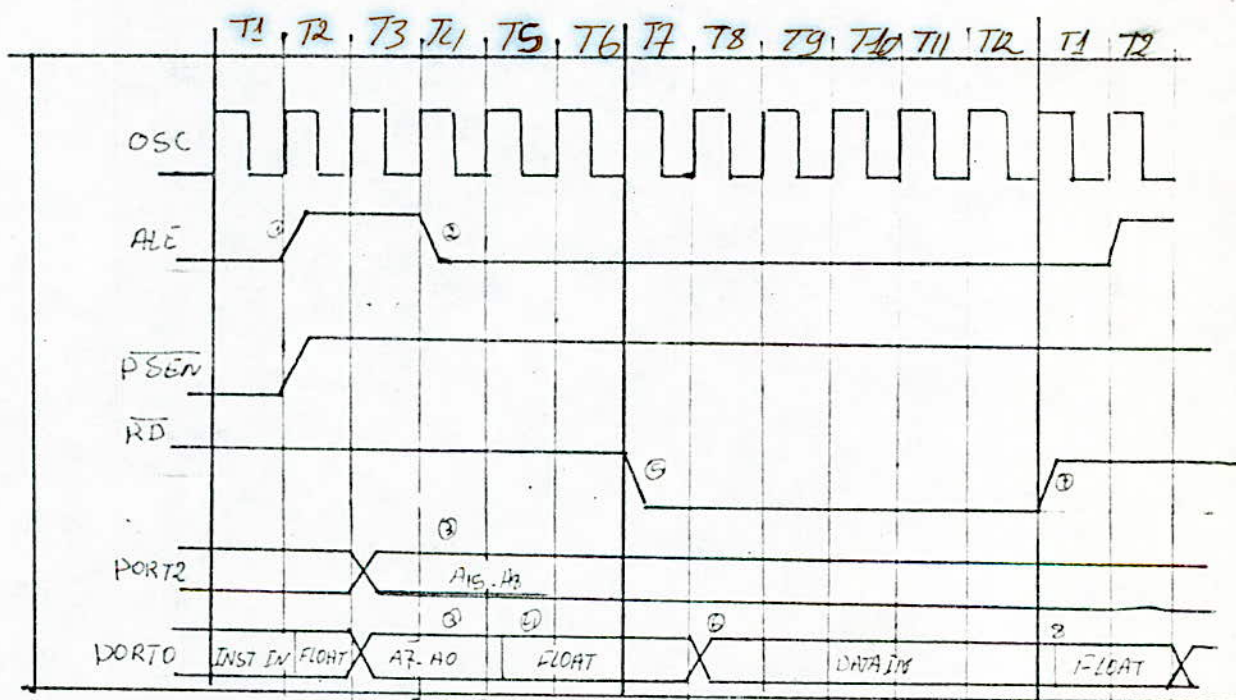


Fig: 37

3/ Séquence d'écriture dans une Mémoire de Données :

Le cycle d'écriture (fig. 38) tout comme le cycle de lecture commence avec l'assertion de l'ALE 1 et l'émission de l'adresse 2. En T6 le processeur émet la donnée pour être écrite dans la mémoire adressée 3. Cette donnée va rester sur le bus jusqu'à la fin de la période T2 du cycle suivant 4, le signal d'écriture WE est activé en T6 5 et le reste jusqu'à T12.

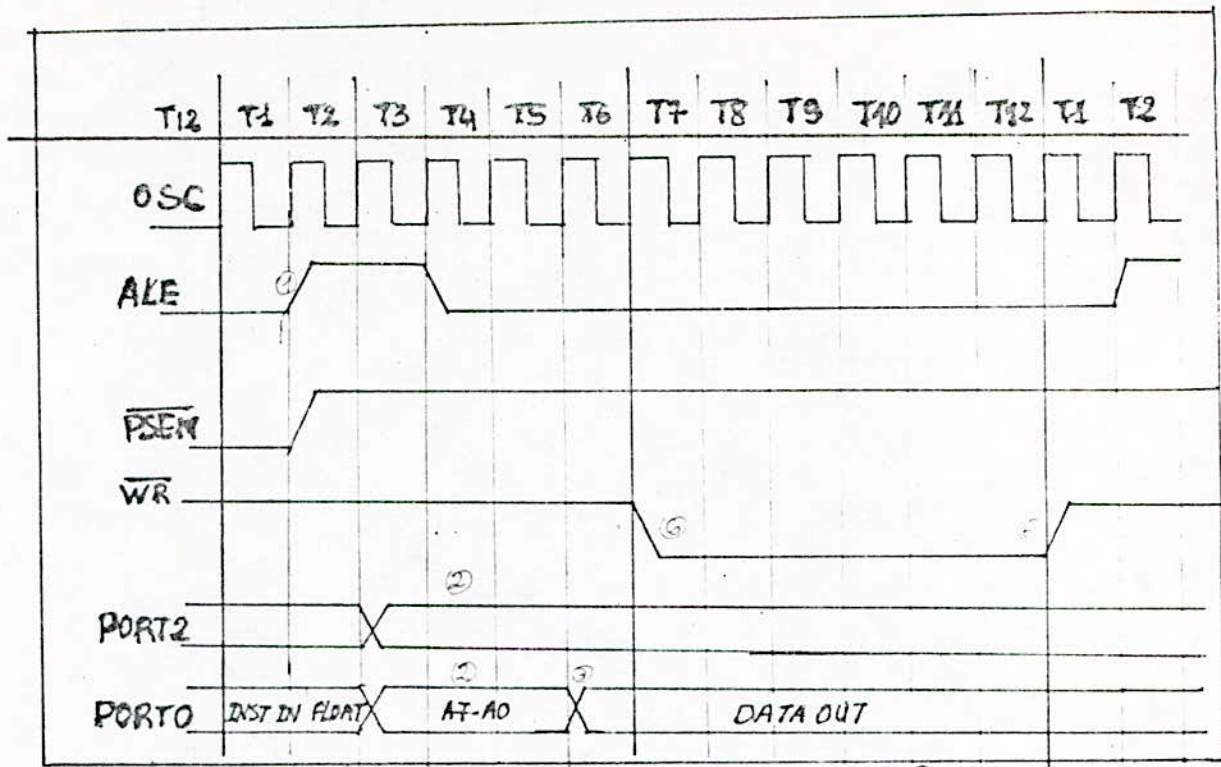


Fig: 38

4-3-31 VUE GENERALE DES TIMERS/COUNTERS DU 8031 :

Le 8031 contient 2 Timers/counters à 16 bits pour mesurer des intervalles de temps, des largeurs d'impulsion, compter des événements et générer des demandes d'interruption périodiques et des impulsions périodiques au port sériel. Cette dernière possibilité est propre au Timer 1 seulement et c'est précisément cette propriété du Timer 1 qui nous intéresse dans le cadre de notre travail (voir Fig. 39).

En mode Timer le Timer/Counter reçoit son Horloge de l'horloge du Quartz du micro contrôleur divisé en fréquence par 12, alors qu'en mode Counter le Timer/Counter reçoit sa fréquence d'horloge de l'extérieur, celle-ci ne pouvant dépasser 0,5 MHz.

Les Timers sont démarrable et arrêtable par contrôle Software, comme nous allons le voir en decrivant le registre de contrôle des Timers et les différents modes des Timers.

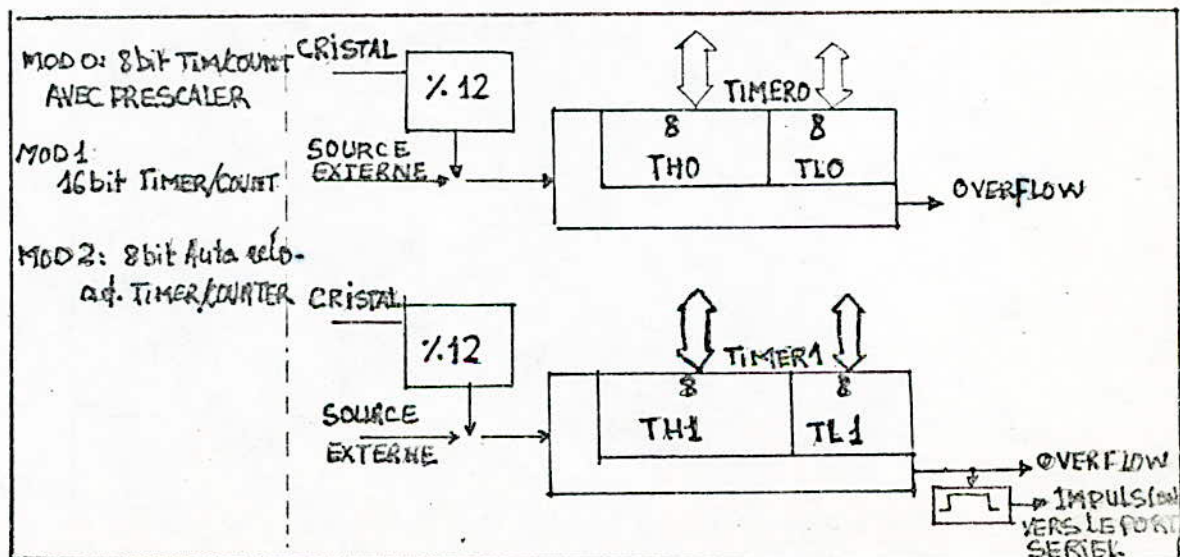


Fig: 39

B/ MODE DE SELECTION DU TIMER/COUNTER 1:

Le Timer/Counter 1 peut être programmé en 4 modes :

Mode 0 : c'est un compteur à 8 bits avec division de la fréquence de la source extérieure par 32.

Mode 1 : configure de Timer/Counter 1 comme un 16 bits.

Mode 2 : configure le Timer/Counter 1 comme un Timer/Counter à 8 bits avec chargement automatique. Ou TH1 contient la valeur à recharger dans TL1. TL1 s'incrémente à la fréquence d'horloge à son Entrée (soit 1 fréquence extérieure soit la fréquence du Quartz divisé par 12), lorsque TL1 arrive à la valeur FFH il envoie une impulsion vers le Sérieel Port et reprend la valeur contenue dans TH1 et ainsi de suite.

Mode 3 : C'est une autre alternative pour arrêter le Timer 1 au lieu de la mise à zéro d'un bit de contrôle, de mise en marche ou d'arrêt (TRI) contenu dans un registre de contrôle TCON (voir plus loin).

C/ CONFIGURATION DU TIMER/COUNTER :

L'utilisation des Timers/Counters est déterminée par deux registres à 8 bits TMOD (Timer mode) et TCON (Timer contrôle).

Pour l'utilisation comme un compteur, l'entrée à la de la part d'une référence extérieure, ou de la part de l'oscillateur pour l'utilisation comme timer. Ceci dépendant du bit C/T du registre TMOD (voir fig. 40) suivant qu'il est à 1 ou à zéro. Quand le bit Gate de TMOD est à 1 l'entrée de référence pour le Counter n'est validé que si les entrées d'interruption extérieures sont hautes (INT0, INT1), quand le bit Gate est à zéro:

Les entrées TO et TI de référence sont validées quelquessoit les niveaux aux entrées INTO et INTI. Donc lors de l'utilisation du Timer plutôt que du Counter, ce bit Gate n'a pas d'intérêt.

Les bits MI et MO servent à choisir le mode du Timer/Counter :

MI	MO	Mode Timer
0	0	0
0	1	1
1	0	2
1	1	3

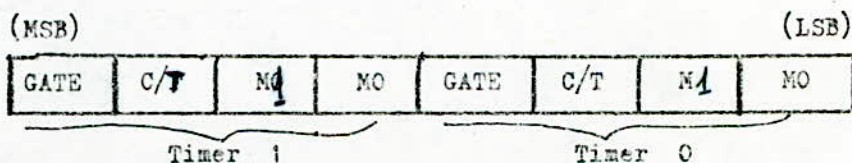


Fig: 40

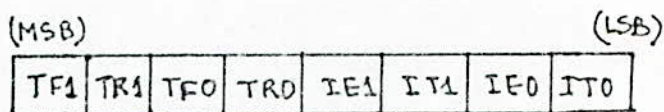
Dans notre cas comme on le verra plus loin avec le Port Sériel, on travaillera avec le Timer 1 en mode 2 (chargement automatique) les valeurs M1 et MO du Timer 0 importent peu puisqu'on ne l'utilisera pas, ni les Entrées des Counters T1 et T0 non plus.

D/ OPERATION DU TIMER :

Pour la mise en marche du Timer/Counter il faut mettre à 1 le bit TR1 dans le registre TCON (voir fig. 41).

Les autres bits du registre n'ont pas d'importance dans notre cas vu que les interruptions des Timers sont invalidées.

Il en est de la même pour l'interruption INT1 qui n'est pas utilisée, l'interruption INTO est par contre utilisée alternativement avec l'interruption du Port Sériel mais que son déclenchement soit au front descendant ou au niveau bas, importe peu, donc pas de Nécessité de Programmer les autres bits, il suffit de mettre à 1 le Bit TR1 qui est un bit adressable en tant que bit pour mettre le Timer en marche.



FONCTION	FLAG	BIT LOCATION	FONCTION	FLAG	BIT LOCATION
Bit de demande d'Interruption Par Le Timer 1	TF1	TCON.7	Bit de demande d'Interruption extérieure 1	IE1	TCON.3
Bit de mise en marche ou d'arrêt du Timer 1	TR1	TCON.6	Bit de choix du front ou du Niveau Pour L'INT1	IT1	TCON.2
Bit de demande d'Interruption par le Timer 0	TFO	TCON.5	Bit de demande d'Interruption extérieure 0	IE0	TCON.1
Bit de mise en marche ou d'arrêt du Timer 0	TR0	TCON.4	Bit de Choix du front ou du Niveau Pour INTO	IT0	TCON.0

FIG: 41

43.4/A/VUE GENERALE DU PORT SERIEL DE COMMUNICATION DU 8031 :

Le 8031 est munit d'un port sériel utilisé pour des liaisons sérielles avec des périphériques ou avec de multiples contrôleurs 8031 à travers des protocoles standard asynchrone. Il est aussi possible de faire des extensions en lignes d'E/S en utilisant ce port avec des registres à décalage ITL ou CMOS.

Le port sériel est munit de modes asynchrone pour faciliter les communications avec des périphériques ayant des UART standard, tel que les imprimantes ou les TRC. Quand on l'interface avec des périphériques UART standard le canal sériel peut être programmé en 4 modes :

- mode 0 : Extension d'E/S synchrone utilisant des registres à décalage, la chaîne consiste en 1 bit départ 7 ou 8 bits de données, 1 ou 2 bits stop.
- mode 1 : qui transmet / reçoit une chaîne de 10 bits par caractère
- mode 2 ou 3: qui transmet / reçoit une chaîne de 11 bits par caractère.

Dans les modes 1 et 3 la circuiterie de la vitesse de transmission reçoit une impulsion du port timer 1 chaque fois que celui-ci déborde. Dans le mode 2 la circuiterie reçoit des impulsions de l'oscillateur même et la vitesse de transmission dans ce mode là est figée. (voir Fig. 42).

8/ REGISTRE DE CONTROLE ET REGISTRE BUFFER DU PORT SERIEL (SCON et SBUF : 8 bits)

La donnée pour la transmission et de la réception réside dans le buffer du port sériel. Une écriture dans le SBUF met à jour le registre transmetteur, une lecture dans le SBUF lit le buffer mis à jour par le registre receveur.

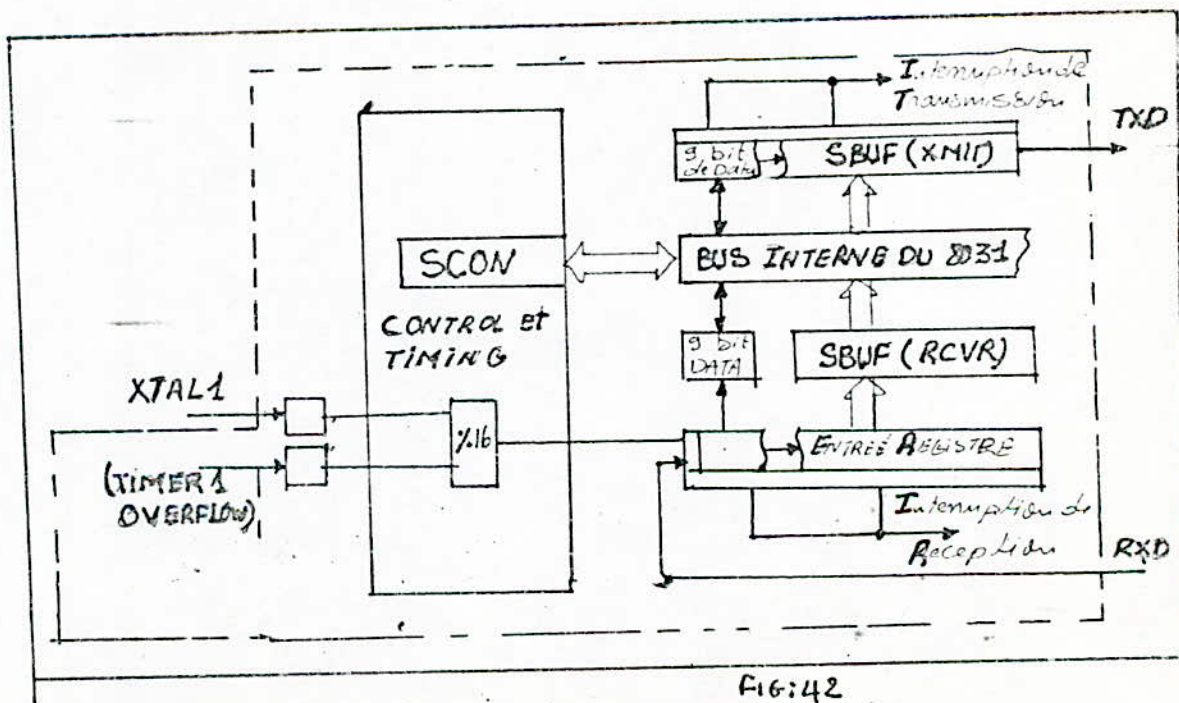


Fig:42

Le port sériel est contrôlé par un registre le SCON dont les fonctions
 des bits sont sur le tableau fig. 43

(MSB)				(LSB)			
SM0	SM1	SM2	REN	TB8	RB8	TI	RI
FONCTION						FLAG	BIT LOCATION
Bit de contrôle du MOD -0- du port sériel. MA1/MAZ par soft.						SM0	SCON.7
Bit de contrôle du MOD -1- du port sériel. MA1/MAZ par soft.						SM1	SCON.6
Bit de contrôle du MOD -2- du port sériel. MA1/MAZ par soft.						SM2	SCON.5
Bit de contrôle de validation de réception.						REN	SCON.4
8ème bit de transmission. MA1/MAZ par soft pour déterminer l'état du 9ème bit transmi dans le mode UART à 9 bits.						TB8	SCON.3
8ème bit de réception.						RB8	SCON.2
Flag d'interruption de transmission.						TI	SCON.1
Flag d'interruption de <i>reception</i> transmission .						RI.	SCON.0

Fig: 43

Les bits SMO et SMI de contrôle du MODE programment le Port Sérieel en un des 4 modes possibles. Le bit REN bit de validation de la réception sert à valider ou non la réception de données, le CPU est informé que le SBUF pour la transmission est vide ou que le SBUF pour la réception est plein par 2 bits TI et RI respectivement positionnés à 1, le TI doit être mis à zéro après avoir servi l'interruption correspondante ou le test correspondant.

Le bit TBS est utilisé dans le mode 9 bits UART pour déterminer l'état du 9ème bit envoyé ; exemple : parité, stop/.

Le bit RBS est aussi utilisé en mode 9 bits UART pour indiquer l'état du 9ème bit reçu. Le bit SM2 sert à invalider les chaines pour lesquelles le bit 8 est un zéro.

C/ LA GENERATION DE LA VITESSE DE TRANSMISSION :

Dans notre travail nous avons utilisé les modes 1 et 3 suivant que l'on ait besoin du mode à 10 bits la chaine ou du mode à 11 bits la chaine avec une vitesse variable de 150 B à 9600 bands par multiple de 2.

Dans le mode 1 ou 3 le port sériel reçoit ces impulsions du timer 1, ce dernier est comme on l'a expliqué auparavant attaqué par la fréquence de l'oscillateur divisé par 12. Les impulsions reçues par le port sériel sont divisées par 2 puis par 16 pour générer la vitesse voulue.

CHOIX DE LA FREQUENCE DU QUARTZ :

Soit N : le nombre d'incrémentations requises par le timer 1 pour qu'il déborde lorsqu'il est en mode Auto-rebord, on aura :

- la fréquence des impulsions reçues par le port sériel f' est donnée par :

$$f' = \frac{f_{osci}}{N \times 12}$$

- la vitesse de transmission en Baud est donnée par :

$$B = \frac{f'}{16 \times 2} \quad \text{soit donc} \quad B = \frac{f_{osci}}{N \times 12 \times 16 \times 2}$$

N devant être entier et compris entre 1 et FFH pour toutes les vitesses de transmission envisagées.

Vu toutes ces contraintes et les contraintes d'existence de Quartz on est arrivé au point suivant :

Pour un quartz de 11.05920 MHz les valeurs N seront :

pour 9600 Band	N = 3
4800 Band	N = 6
2400 Band	N = 12
1200 Band	N = 24
600 Band	N = 48
300 Band	N = 96
150 Band	N = 192

Quartz qui bien sûr existe sur le marché.

4-3.5/A/ INTERFACAGE EXTERIEUR :

A.a/ Réset du processeur et initialisation -

L'initialisation du processeur est accompli par activation de la pin RST. Pour initialiser le processeur cette pin doit rester haute pour au moins 1 ms après que l'alimentation se soit stabilisée. Le front descendant sur le RST initialise une séquence qui après environ 12 périodes de l'oscillateur va générer le signal ALE et une opération normale commence avec l'instruction à la location 0000H.

L'entrée Réset est suivie d'une faible résistance puis d'un trigger intégrés dans la puce en ajoutant une faible capacité connectée entre la pin RST et le Vcc (voir fig. 44) un réset est alors généré automatiquement dès que l'on met l'alimentation en marche.

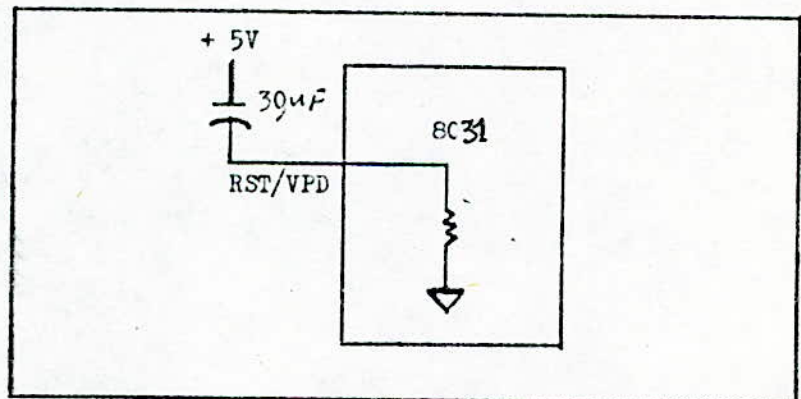


Fig:44

Le tableau suivant contient les valeurs que prennent les registres spécialisés et la RAM interne après un Réset.

REGISTRE	CONTENU
PC	0000 H
SP	07 H
PSW, DPH, DPL, A, B,	00 H
IP	E0H or 00H
IE	60 H or 00H
SCON, TCON, TMOD,	00H
TH1, TH0	00H
TL1, TL0	00H
SBUF	Indéterminante
Port 3 - Port 0	FFH (configure toutes les pins d'E/S comme entrées.)

b/ Oscillateur et circuit d'horloge (timing)

La génération de timing pour le 8031 est contenue complètement dans la puce ; à l'exception près pour la fréquence de référence qui peut être soit : un cristal ou une horloge extérieure.

L'oscillateur intégré est un circuit anti-résonnant parallèle pouvant fournir des signaux de fréquence variant de 1,2 M à 12 MHz.

V/CARTE UNITE CENTRALE

-----oOo-----

Réalisée autour du Micro-contrôleur 8031 d'INTEL cette carte devra contenir :

- Une Mémoire EPROM ou PROM de 2 K bytes qui contiendra le programme Moniteur du testeur complet
- Une mémoire EPROM de 2 K bytes qui contiendra la table des nombres de points de départs total en 2 bytes et les 2046 bytes restant contiendront le nombre de points d'arrivées par bloc (un nombre par byte donc le nombre de points d'arrivées ne doit pas dépasser 256 points par bloc vû la disposition de la table mais il ne faut pas perdre de vue les conditions sur le nombre de points d'arrivées imposées par l'interface de réception des données en provenance de la carte à tester (voir partie Interface de réception)).
Puisqu'il y a 2046 blocs possibles il y a donc 2046 points de départs possibles.

- 3 Mémoires EPROM de 4K Bytes chacune vq :

Pour la 1ère Mémoire :

- . les 2 premières K bytes contiendront les Numéros de Chip de point de départ
- . les 2 dernières K bytes contiendront les numéros de Pin de point d'arrivée.

La deuxième Mémoire :

- . contiendra les numéros de Chip de point d'arrivée

La troisième Mémoire :

- . contiendra les numéros de Pin de point d'arrivée.

Donc nous avons 4096 points d'arrivées possibles et 2046 points de départs possibles. Ce qui est largement suffisant pour n'importe quel carte à tester ; il suffirait d'adapter le hard Inter/Face émission et Inter/Face réception (c'est-à-dire point de vue, nombre de registre etc...) à la carte à tester.

- Les 3 dernières mémoires servent à afficher le défaut en Numéro de Chip/Pin départ et Numéro de Chip/Pin arrivée pour que l'information de défaut soit compréhensible par le dépanneur.

On a choisi des mémoires mortes de façon à ce que l'utilisateur n'ait à charger les informations qu'une seule fois c'est lors de leur programmation.

De plus on les a prises reprogrammable de façon à ce que la carte UC soit universelle c'est-à-dire qu'il suffirait de les recharger avec d'autres tables pour pouvoir tester une autre carte correspondant à ces tables.

- Une mémoire RAM de 2K octets servant à sauvegarder les adresses de départ et d'arrivée des défauts de façon à ce que le processeur puisse les relire pour pouvoir imprimer les défauts à la fin du test. Puisque les adresses sont en 4 bytes il y a donc environ 512 défauts possibles à sauvegarder ce qui est largement suffisant pour n'importe quelle carte à tester.

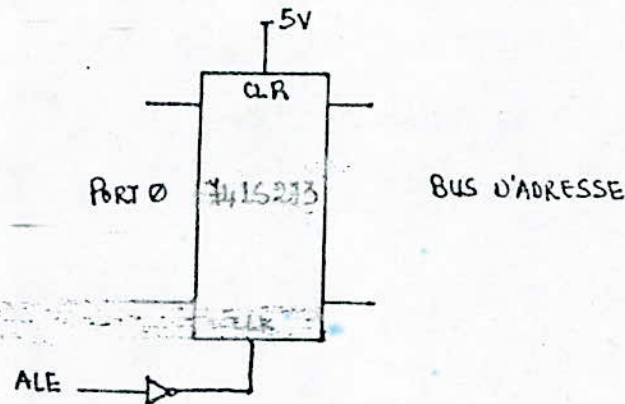
- Deux Switchs de 8 bits à positions chacun servant pour le choix
- du protocole (DTR ou XON - XOFF) DB0
- de la vitesse DB1 - DB3
- de la validation de la parité DB4
- du choix du mode de parité DB5
- du choix du nombre de bit par chaine (10 ou 11 bits) DB6
- du choix du nombre de bit stop (1 ou 2) dans le mode 11 bits DB7

Pour l'interface sériel avec le terminal ou l'imprimante.

- Des buffers à 3 états qui serviront lorsqu'ils sont validés à lire les switchs sinon leur sorties seront dans l'état haute impédance.
- Des circuits TTL pour le démultiplexage adresse donnée du Port 0, amplification et mise en forme des signaux d'adresse et de data démultiplexés de l'adresse du port 2, des signaux de contrôle (ALE, PSEN, RD, WR) de 256 mots de 8 bits.
- Une PROM bipolaire pour le décodage d'adresse
- Une circuiterie logique pour valider le port sériel et l'interruption DTR (INT0) s'il y a lieu soit vers le terminal soit vers la vue
- Des circuits pour interfacage avec le terminal ou l'imprimante dans le standard RS 232.C.

Démultiplexage du Port 0 en 2 bus adresse et data séparés:

- Echantillonnage de l'adresse poids faible: comme expliqué dans le chapitre réservé au microcontrôleur lors du front descendant du signal de contrôle ALE, le processeur envoie à travers le port 0 l'adresse de poids faible, c'est donc à ce moment là qu'il faut échantillonner l'adresse, grâce au signal ALE. et cette adresse doit rester jusqu'à la fin du cycle de l'instruction donc jusqu'au prochain front descendant de l'ALE. On aboutit ainsi au circuit suivant :



00, le latch 74 LS 273 est constitué de 8 bascules D dont toutes les entrées CLOCK sont reliées entre-elles, celles-ci étant déclenchables au front montant de l'horloge il faut donc inverser le signal ALE pour déclencher les bascules au front descendant du signal horloge.

L'entrée CLR commune à toutes les bascules est invalidée en la forçant au niveau logique haut.

De plus ce circuit agit comme un buffer pouvant alimenter plusieurs charges TTL :

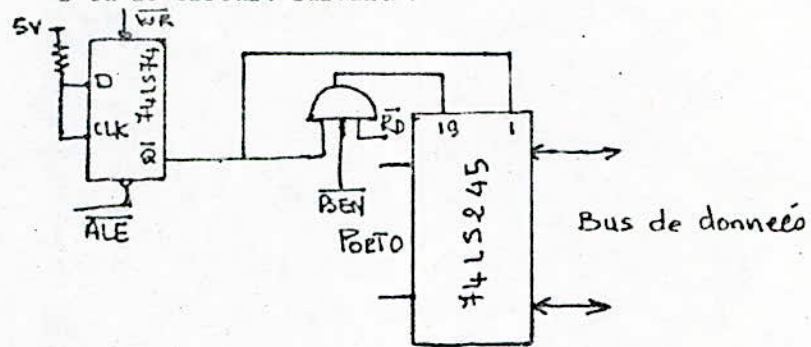
— Obtention du bus de données bidirectionnel.

Lors de la lecture dans une mémoire programme le port 0 se met en état de recevoir une data peu après que le $\overline{\text{PSEN}}$ soit descendu au niveau bas. De même lors de la lecture ou de l'écriture dans une mémoire data extérieure le port 0 se met en état de recevoir ou d'émettre une data respectivement peu de temps après que le signal $\overline{\text{RD}}$ soit descendu au niveau bas ou peu de temps avant que le signal $\overline{\text{WR}}$ ne descende au niveau bas.

C'est donc avec le début du niveau bas d'un des signaux \overline{PSEN} , \overline{RD} , \overline{WR} qu'il faut 'piquer' le port 0.

Cependant le bus de données doit être bidirectionnel, émettre une data quand il s'agit d'un des signaux \overline{PSEN} , \overline{RD} ou recevoir une data quand il s'agit du signal \overline{WR} .

D'où le circuit suivant :



ou le composant 74 LS 245 est un buffer bidirectionnel à 3 état ou l'entrée 19 est une entrée de validation du chip active niveau bas, le Chip est donc validé avec un des niveaux bas \overline{PSEN} ou \overline{RD} ou \overline{WR} et l'entrée 1 est une entrée de commande de direction, si elle est haute le buffer se met dans le sens entrée, c'est-à-dire émission de data sinon il est en mode lecture.

Le circuit 74 LS 74 qui est une simple bascule D, horloge et entrée forcées au niveau haut, sert à valider et maintenir le sens de direction du buffer ceci en mode écriture jusqu'à la prochaine instruction; la data subsiste encore sur le port 0 bien après que le signal \overline{WR} ne remonte au niveau haut à cause du fait que beaucoup de mémoire RAM en mode écriture surtout les mémoires sont déclenchables pour écriture au front montant du \overline{WR} .

Donc si le \overline{WR} est activé le bus gardera son mode jusqu'au prochain front descendant de l' \overline{ALE} donc front montant de l' \overline{ALE} .

De plus le buffer triggéré donc remet en forme les signaux plus ou moins distordu.

Amplification et mise en forme des signaux du Port 2 et des signaux de contrôle :

Comme expliqué dans la partie réservée au micro contrôleur, les signaux d'adresse poids fort du port 2 et les signaux de contrôle \overline{RD} , \overline{WR} , \overline{PSEN} , ALE sont passés à travers des buffers 74 LS 244 qui les amplifient pour pouvoir attaquer plusieurs charges TTL.

De plus ces buffers sont triggérés ainsi remettent en forme les signaux qui les traversent.

Ces buffers à 3 états sont tout le temps validés.

Décodage d'adresse :

Pour le décodage d'adresse, nous avons utilisé une PROM bipolaire la TTL 74 S 471 avec sorties à 3 états. Cette PROM à 256 mots de 8 bits admet pour entrées le bus d'adresse poids fort en provenance du buffer 74 LS 244 et pour sorties des entrées de Chip Select des 5 EPROMS de la RAM et des 2 Switchs 8 bits adressés comme 2 zones mémoire.

Le décodage d'adresse est fait selon le tableau (fig. 45).

Sachant que les Chips Selects des circuits à sélectionner sont actifs niveau bas, il suffit de mettre dans une ou des case(s) mémoires(s) de la PROMS 471 un bit à 0 et les autres à 1.

Ce bit mis à 0 servira à sélectionner un circuit donné lorsque l'adresse poids fort correspond à l'adresse de la zone mémoire considérée de la PROM sera envoyée par le processeur.

Avec ce décodage nous sélectionons 8 chips différents et un seul à la fois. Enfin, cette FROM retarde le Chip Select par rapport à l'adresse de 50 NS.

Les Switchs et leur Buffers :

Pour chaque bit d'un Switch on a :

- une entrée reliée à la masse, l'autre entrée reliée auVcc à travers une résistance de 4,7 K Ω de façon à avoir un niveau haut sur le bout de la résistance relié au switch lorsque celui-ci est ouvert. Ce dernier bout sert ensuite d'entrée d'un buffer à 3 états avec commande active niveau bas.

Adresse Hexa dans la PROM	CONTENUE BINAIRE								Zone d'adresse Hexa à 16 bits y correspondants	Chip Selecté
	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
00	0	1	1	1	1	1	1	1	0000	1ere
07	0	1	1	1	1	1	1	1	07FF	2416
08	1	0	1	1	1	1	1	1	0800	2eme
0F	1	0	1	1	1	1	1	1	0FFF	2416
10	1	1	0	1	1	1	1	1	1000	1ere
1F	1	1	0	1	1	1	1	1	1FFF	2432
20	1	1	1	0	1	1	1	1	2000	2eme
2F	1	1	1	0	1	1	1	1	2FFF	2432
30	1	1	1	1	0	1	1	1	3000	3eme
3F	1	1	1	1	0	1	1	1	3FFF	2432
40	1	1	1	1	1	0	1	1	4000	RAM
47	1	1	1	1	1	0	1	1	47FF	2128
48	1	1	1	1	1	1	1	1	4800	Aucun
FD	1	1	1	1	1	1	1	1	FDFF	Aucun
FE	1	1	1	1	1	1	0	1	FE00	Switchs
	1	1	1	1	1	1	0	1	FEFF	Visu
FF	1	1	1	1	1	1	1	0	FF00	Switchs
	1	1	1	1	1	1	1	0	FFFF	Imprim- -aute

fig 45. Tableau du decodage d'adresse.

La sortie du buffer est reliée a un fil donné du bus de données et les lignes de commande des 8 bits de chacun des deux Switchs (visu ou imprimante) sont reliées entre elles et à un fil de sortie de la PROM ~~des~~ deux d'adresse.

Ainsi les Switchs sont vu par le processeur comme deux zones mémoires adressables de la mémoire programme ou de la mémoire data dans lesquelles il ne peut que lire :

les composants sont des 74 LS 125 qui contiennent 4 buffers chacun.

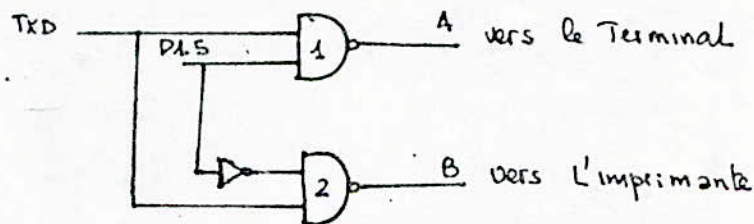
Logique de sélection du terminal ou de l'imprimante et interfaces

Line Driver et Line Receiver :

La sélection se fait par Soft avec un fil du Port i, soit P1.5.

Si le niveau de P1.5 est à 1 c'est le Terminal sinon c'est la visue, qui est selecté .

Circuiterie de sélection du TXD.



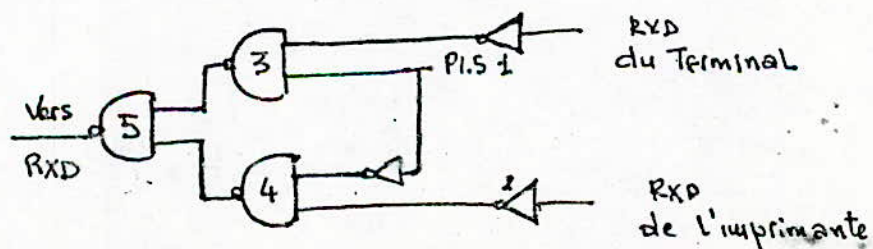
Le rôle des deux driver de lignes est de transformer les signaux du TXD de la logique TTL en logique : + 12 v, - 12 v du standard RS 232.

Si P1.5 = 1, sortie (A) = TDX mais en + 12, - 12 v, (B) = + 12 v

P2.5 = 2, sortie (A) = 12v (B) = TDX en + 12v, - 12 v.

Ces deux drivers sont contenues dans le circuit interface , Line driver MC 1488 de motorola.

Circuiterie de selection du RXD :



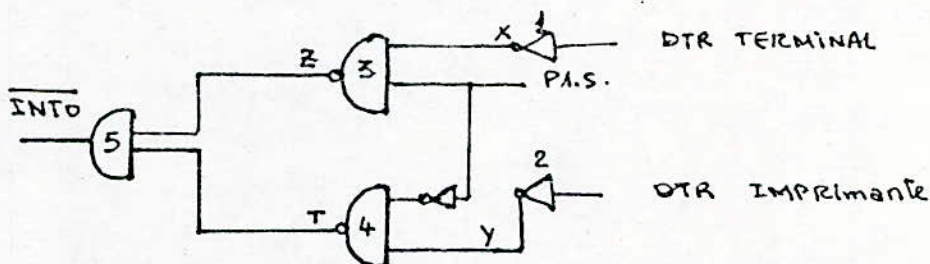
Les circuits 1 et 2 sont des lignes receivers, font la fonction inverse des line driver en l'occurrence transforment des signaux + 12v, - 12v en logique TTL. (3), (4), (5) sont des circuits NANDTTL.

Si P1.5 = 1 = RXD = RXD du terminal

P1.5 = 0 = RXD = RXD de l'imprimante.

Les lignes receivers sont des PC 1489 de Motorola.

Circuiterie de sélection du DTR visu ou imprimante :



- (1) (2) sont des lignes drivers
- (3) (4) sont des portes NAND TTL
- (5) est une porte AND TTL

Si P1.5 = 1 $X = \overline{\overline{DTR}}$ mais en TTL $Z = \overline{\overline{DTR}} = DTR$

$Y = \overline{\overline{DTR}}$ $T = 1$ logique TTL

$\overline{INT0} = DTR$ TTL c'est-à-dire 5v si DTR = 12v
0v si DTR = 0v

Si P1.5 = 0 c'est DTR imprimante qui est sélectionné.

Choix des Mémoires EPROM 2K et 4K et de la RAM 2K:

- Choix des Mémoires -

Mode lecture dans la Mémoire programme :

soit T_1 le temps d'accès requis par le \overline{PSEN}

T_2 le temps d'accès requis par l'adresse.

Le constructeur donne que pour 1 fréquence d'horloge f

$$f = \frac{1}{TCLCL} \quad f \text{ entre } 1,2 \text{ et } 12 \text{ MHz}$$

$$T_1 = 3 \text{ TCLCL} - 100 \text{ (ns)}$$

$$T_2 = 5 \text{ TCLCL} - 100 \text{ (ns) (voir 8051 user's MANUEL).}$$

Dans notre cas $f = 11.059 \text{ 20 MHz}$

$$T_1 = 171 \text{ ns}$$

$$T_2 = 352 \text{ ns}$$

- Le temps d'accès requis par le \overline{CE} -

Le \overline{CE} est obtenu à partir de la PROM 471 qui a pour entrée, l'adresse de poids fort, la PROM 471 introduit un retard de près de 50 ns sur le \overline{CE} par rapport à l'adresse.

Le CE requiert un temps d'accès $T_3 = T_2 - 50\text{ns} \approx 300\text{ns}$.

Pour les EPROMS à 2K Bytes notre choix s'est porté sur les EPROMS 2716 d'Intel.

Elles ont les caractéristiques suivantes :

- temps d'accès requis par l'adresse 350ns

- temps d'accès requis par le \overline{CE} 350ns

- temps d'accès requis par le \overline{OE} 120ns

(OE = Output Enable)

Pour les EPROMS à 4 Bytes notre choix s'est porté sur les EPROMS 2732 A d'Intel ;

Elles possèdent les caractéristiques suivantes :

- temps d'accès requis par l'adresse 250ns

- temps d'accès requis par le \overline{CE} 250ns

- temps d'accès requis par le \overline{OE} 100ns

-Mode lecture dans la Mémoire de données :

T3 = 4 TCLCL - 130

T4 = 5 TCLCL - 170

T3 = le temps d'accès requis par l'adresse

T4 = le temps d'accès requis par le \overline{RD}

T5 = le temps d'accès requis par le \overline{CE} = $T_3 - 50$ ns

Pour $f = 11.05920$ MHz

$T_3 = 231$ ns

$T_4 = 282$ ns

$T_5 = T_3 - 50$ ns = $231 - 50 = 181$ ns.

Notre choix s'est porté sur la Mémoire statique 2128 d'Intel, possédant les caractéristiques suivantes :

- . temps d'accès requis par l'adresse = 250 ns
- . temps d'accès requis par le \overline{CE} = 250 ns
- . temps d'accès requis par le \overline{OE} = 200 ns.

N.B. : De plus ces trois sortes de mémoires que l'on a choisies font partie de la famille MCS 80/85, compatible avec le micro-contrôleur 8031.

Composants utilisés et Schémas global :

- Composants utilisés -

- Micro-contrôleur 8031 (U1)
- 2 Mémoires EPROM 2716.1 U2 et (U4)
- 3 Mémoires EPROM 2732.A U5, U6 et (U7)
- 1 Buffer bidirectionnel de bus à 3 états 74 LS 245 (U8)
- 1 latch 74 LS 273 (U9)
- 2 buffers de bus à 3 états 74 LS 244 U 10 et (U 11)
- 1 bascule D 74 LS 74 (U 12)
- 3 portes AND 3 Entrées 74 11 (U13)
- 6 inverseurs 74 LS 04 (U 14)
- 3 portes NAND à 2 entrées 7400 (U 15)
- 2 Line Driver MC 1488 (U 16)
- 4 Line Receiver MC 1489 (U 17)
- 1 Quadruple bus buffer avec sortie 3 Etats 74 LS 125 U 19, U 20, U 21 et U 22.
- 1 PROM bipolaire 25 mots de 8 bits 74 S 471 (U18)
- 2 réseaux de résistances (15 résistances de 4,7K chacun)
- 2 Switchs à 8 bit
- 1 Quartz de 11.059 20 MHz
- 1 capacité de 30 μ F.

*P*ARTIE *//* OPTWARE

-----oO-----

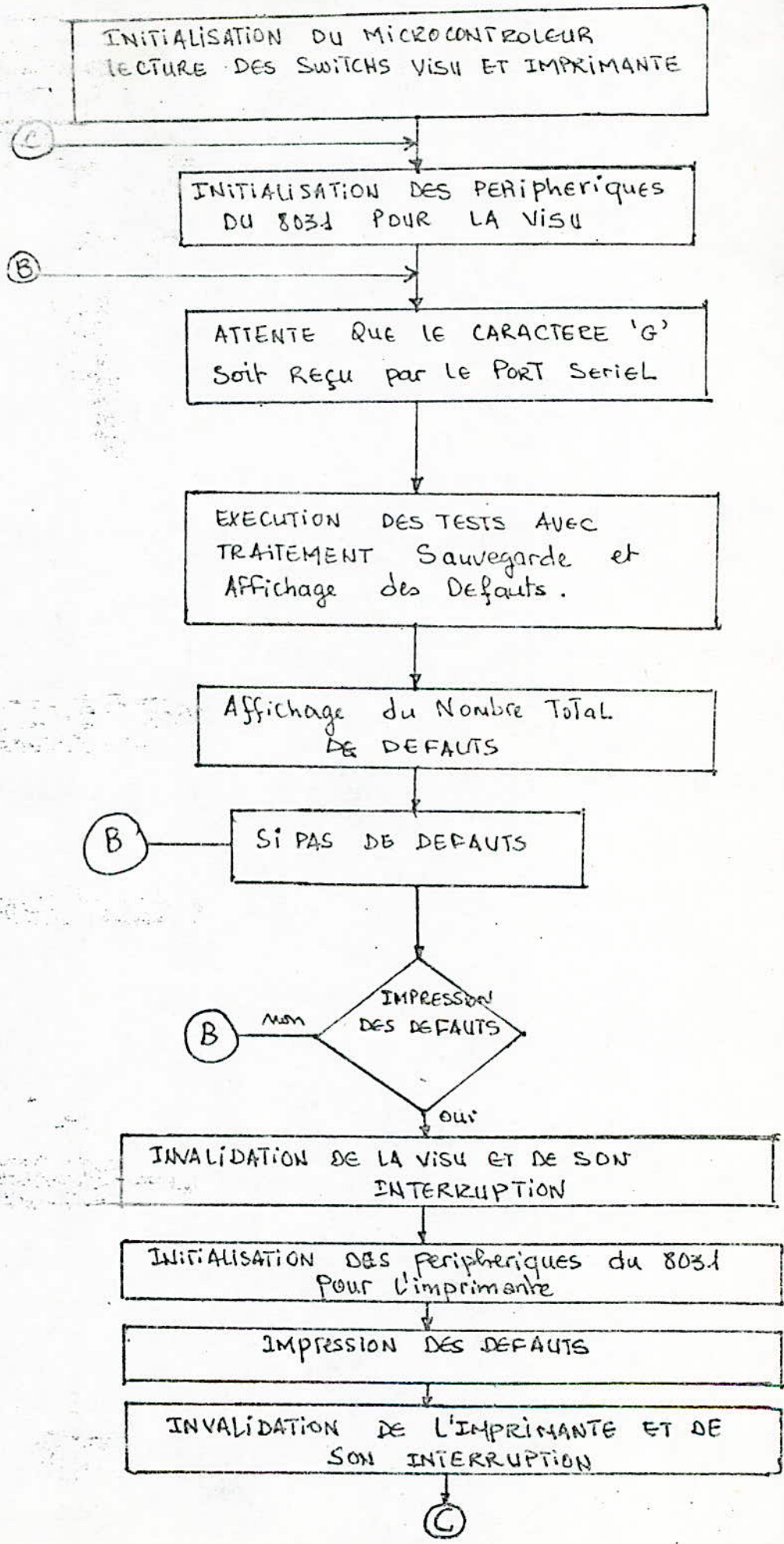
PROGRAMMATION

-----oOo-----

Ce chapitre est divisé en 3 parties :

- 1ère partie : Organigramme général du Moniteur du testeur
- 2ème partie : Organigramme des différentes parties du programme Moniteur
- 3ème partie : Organigramme des vecteurs d'interruptions et sous programme.

1ere PARTIE



2eme PARTIE

PROGRAMME D'INITIALISATION DU MICRO CONTROLLEUR
ET LECTURE DES SWITCHS VISU ET IMPRIMANTE • COM-
MENCE A INITIA: JUSQU'A BOUCLE

-----oOo-----

A/ FONCTION :

Initialise le pointeur de pile à l'adresse 22H correspondant au 3^{me} byte des bytes adressables bit par bit dans la RAM interne du processeur, initialise le Timer 1 au mode Auto-reload, initialise le mode du port sériel à 10 bits la chaîne, positionne le bit de validation globale des interruptions, initialise le port 1 tq : de P1.0 à P1.3 en sortie et les autres restent en entrée ~~lit~~, le switch de la visu charge sa valeur dans la première case mémoire adressable bit par bit de la RAM interne (adresse 20H) ~~lit~~ les switchs vitesses retrouve la valeur qu'il faut charger dans TH1 pour générer la vitesse demandée mais la charge dans une case mémoire au lieu de TH1, puis fait la même chose pour l'imprimante mais avec les adresses suivantes 21H et 1FH.

B/ PARAMETRES D'ENTREES :

- Aucun.

C/ ZONE DE TRAVAIL :

- Routine : RESWBR
- Le pointeur de pile
- Le bit validation globale des interruptions EA
- Le registre de contrôle du port sériel SCON
- Le registre mode des Timers
- Le port 1
- Les Switchs
- Le registre pointeur DPTR
- RAM interne du processeur
- L'Acc
- Le registre B.

D/ PARAMETRES DE SORTIE :

- Citées dans la fonction.

INITIALISATION DE DEBUT DE LA PILE
INITIALISATION DU Mode du TIMER A
INITIALISATION DU Mode du port SERIEL
VALIDATION GLOBALE DES INTERRUPTIONS
INITIALISATION DU PORT A

Lecture des switchs visu
et sauvegarde dans la case
Memoire 20H.

LECTURE DE LA VITESSE
VISU et sauvegarde dans
la case Memoire d'adresse 1EH

LECTURE DES Switchs Imprimante
et sauvegarde dans la case
Memoire d'adresse 20H

LECTURE DE LA VITESSE
Imprimante et sauvegarde
dans la case Memoire 1FH

PROGRAMME D'INITIALISATION DE LA VITESSE
DE TRANSMISSION DU PORT SERIEL DE SON MODE ET DE L'INTERRUPTION
CORRESPONDANTE AU PROTOCOLE D'ECHANGES AVEC LA VISU NOTE
INTISV

-----oOo-----

A/ FONCTION :

Le role de Ce programme est d'initialiser les périphériques du 8031 et l'interruption correspondante au protocole pour la visu en utilisant des routines séparées.

B/ PARAMETRES D'ENTREES :

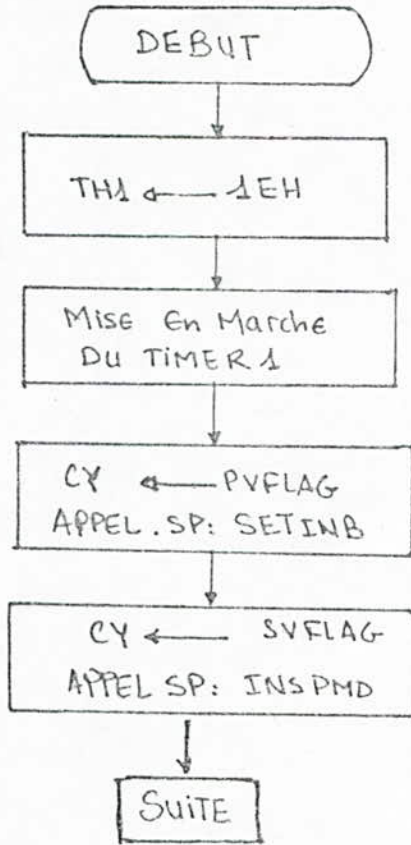
- Contenu de la zone mémoire de la RAM interne d'adresse 1EH
- Le bit PVFLAG : bit protocole d'échange avec la visu (DTR ou XON - XOFF)
DTR = 0 = Protocole DTR
DTR = 1 = Protocole XON - XOFF
- Le bit SVFLAG : bit indiquant le mode 10 ou 11 bits d'échanges avec la visu :
SVFLAG à 1 - mode 10 bits
SVFLAG à 0 - mode 11 bits.

C/ ZONE DE TRAVAIL :

- Routine : INSPMD
- Routine : SETINB
- Registre TH1 du Timer 1
- Zone mémoire 1EH

D/ PARAMETRES DE SORTIES :

- Chargement de TM1 à la vitesse désirée et déclenchement du Timer 1
- Ceux de INSPMD et de SETINB.



PROGRAMME D'ATTENTE QUE LE
CARACTERE G SOIT ENVOYE DE
LA VISU

-----oOo-----

A/ FONCTION :

Affiche un message demandant d'appuyer sur 'G' pour lancer les tests, attend que le code ASCII de 'G' soit envoyé de la visu, le reconnait et l'affiche.

B/ PARAMETRES D'ENTREES :

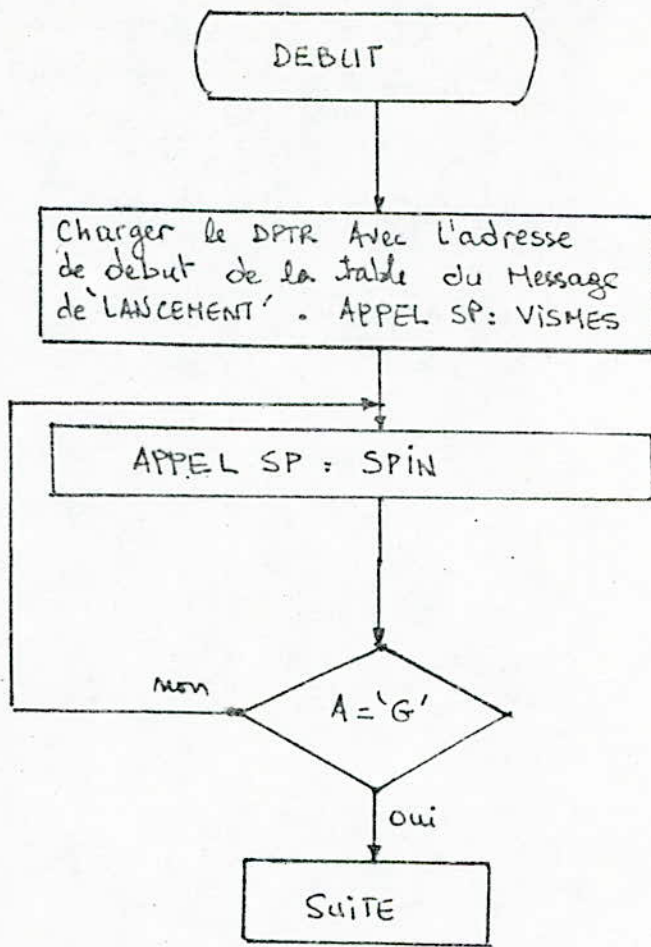
- Aucun.

C/ ZONE DE TRAVAIL :

- Routines, VIGNES, SPIN, SPOUT
- Accumulateur
- Tube de caractère débutant à l'adresse ADRIAN.

D/ PARAMETRES DE SORTIES :

- Aucun.



PROGRAMME PRINCIPAL D'EXECUTION
DES TESTS DE PISTES

-----oOo-----

A/ FONCTION :

La fonction de ce programme est d'envoyer les signaux : DATA, HORLOGE 1 Shift/Load sur les registres à décalage 74 LS 165 et Horloge 2 sur les 74 LS 166 de même il a pour tâche de tester la sortie série du 1er registre 74 LS 166. Pour cela il faut noter les notations suivantes :

- soit CP1H et CP1L 2 registres à 8 bits contenant l'adresse du point de départ où l'information niveau zéro (0) logique a été envoyée. Soit CP1 : l'enchaînement fictif de ces 2 registres 8 bits , soit N le nombre de points de départ total ; celui-ci étant aussi un enchaînement fictif de 2 registres NH et NL, ces deux derniers sont chargés à partir des 2 premiers bytes de la table des points de départ et nombre de points d'arrivées par bloc. CP1 peut prendre des valeurs de 1 à N.
- Soit CP3H et CP3L 2 registres à 8 bits représentant l'adresse du bloc d'arrivée dans lequel se trouve le point d'arrivée que l'on est entrain de tester. CP3 s'incrémente lors du passage d'un bloc à un autre. Comme il y a autant de bloc que de point de départ CP3 varie de 1 à N.
Ainsi pour une valeur de CP1 donnée CP3 va varier de 1 à N.
- Soit un registre à 8 bits : CP2 représentant l'adresse du point d'arrivée testé dans le bloc le contenant.
Pour un bloc donné par exemple de 10 points, CP2 pourra prendre les valeurs de 1 à 10. CP2 est initialisé à 1 lorsqu'on rentre dans 1 nouveau bloc cad où lorsqu'on incrémente CP3.
- Soit LP2H et LP2L l'adresse d'un point d'arrivé que l'on est entrain de tester. LP2 prend les valeurs à partir de 1 jusqu'au nombre total de points d'arrivées existants.
- Soit le registre SAUV contient le nombre de points d'arrivées du bloc considéré.

- Lorsque CP1 est égal à 1 c'est-à-dire lorsqu'on envoie la 1ère configuration en entrée, on teste le 1er point d'arrivée c'est-à-dire le point pour lequel LP2 = 1 (LP2H = 0 LP2L = 1) CP2 = 1 et CP3 = 1 (CP3H = 0 - CP3L = 1).
- Une configuration donnée est le fait que le niveau logique zéro (0) soit à un point de départ précis ou donné.

B/ PARAMETRES D'ENTREES :

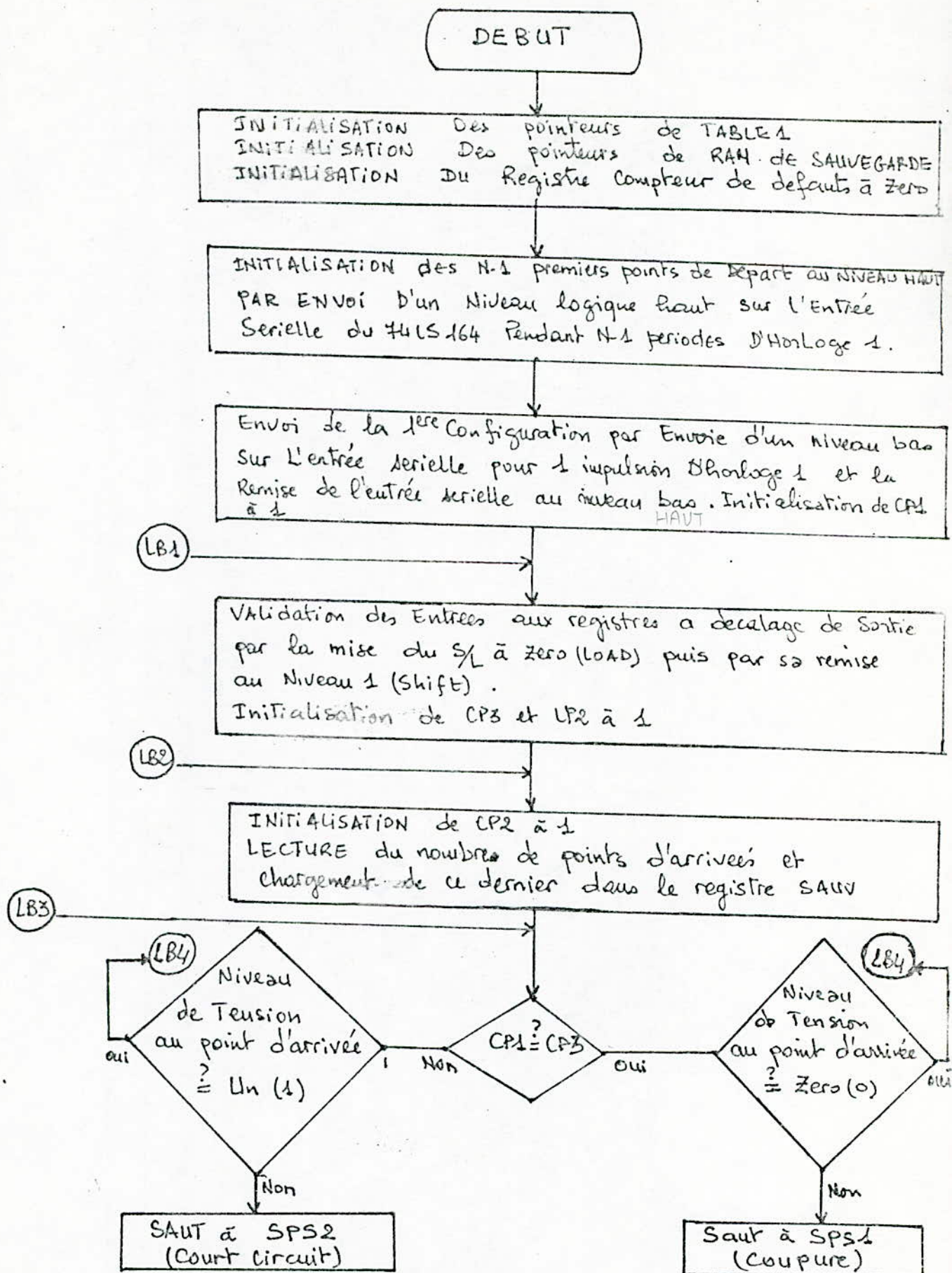
- Table 1 : table nombre de points de départ total en 2 bytes et nombre de points d'arrivées par bloc (contenue dans EPROM 2716 (2K bytes)).

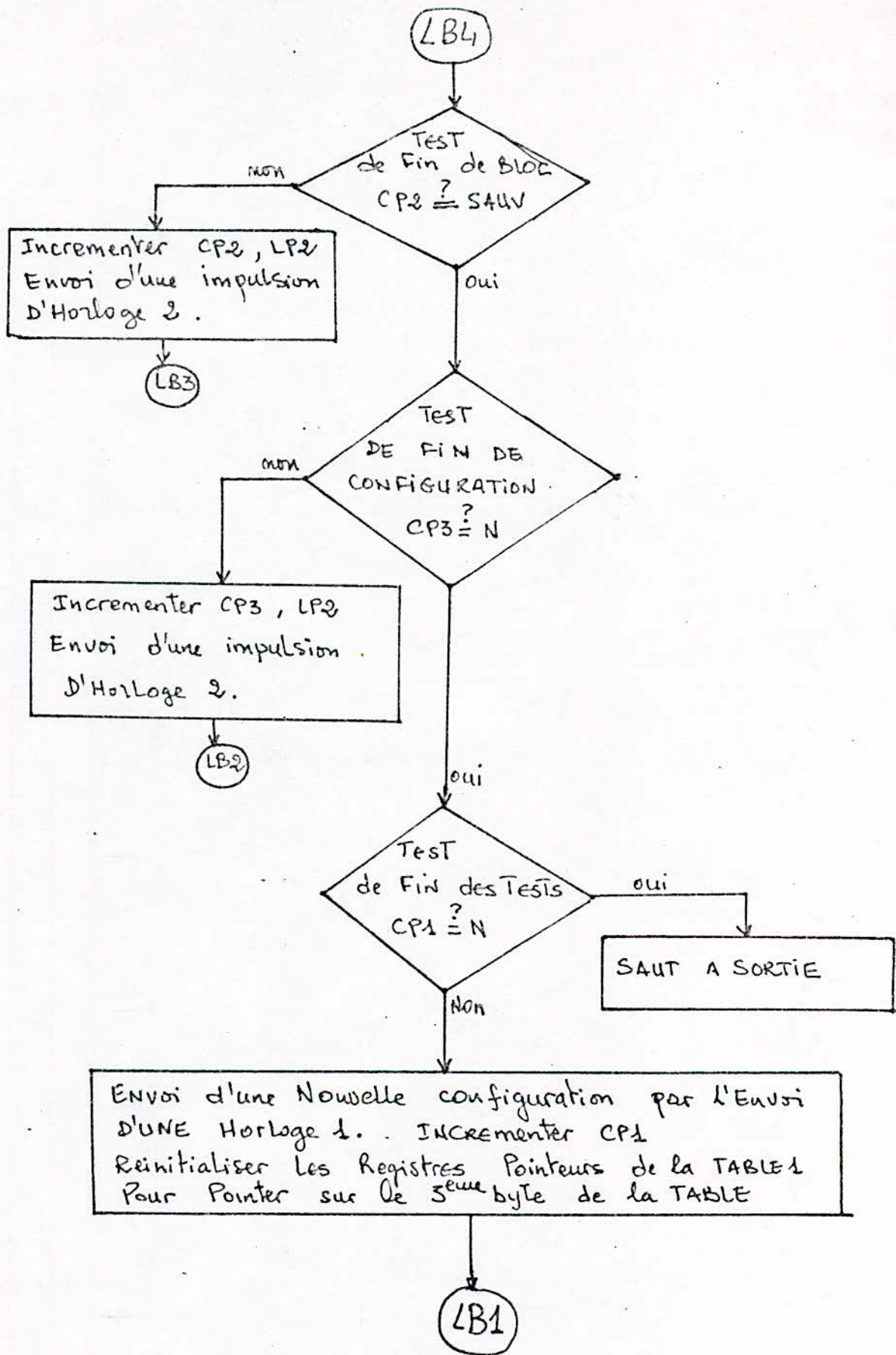
C/ ZONE DE TRAVAIL :

- Acc
- Registres de la RAM interne : CP1H, CP1L, LP2H, LP2L, CP3H, CP3L, CP2, SAUV.
- Registres pointeurs RAM de sauvegarde REGSAH, REGSAL
- Registres pointeurs de table 1 RGNPDH, RGNPEDL
- Routines : LECT, SPH1, SPH2
- Bits adressables port 1 P1.0, P1.1, P1.2, P1.3, P1.4 respectivement entrée sérielle 74 LS 165, Horloge1, Horloge.2, Shift .Load, point de Test.

D/ PARAMETRES DE SORTIES :

- Se branche en cas de coupure sur la routine SPS1 et en cas de court-circuit sur la routine SPS2.





PROGRAMME D'AFFICHAGE DU NOMBRE TOTAL
DE DEFAUTS ET DE CHOIX DE L'IMPRESSION

-----oOo-----

A/ FONCTION :

Donner le nombre total de défauts grâce notamment au registre RGCPDE (ou un message) indiquant que la carte n'a pas de défaut et boucler pour un nouveau test), puis afficher un message de choix d'impression et attendre un caractère O ou N pour imprimer ou boucler pour un nouveau test.

B/ PARAMETRES D'ENTRES :

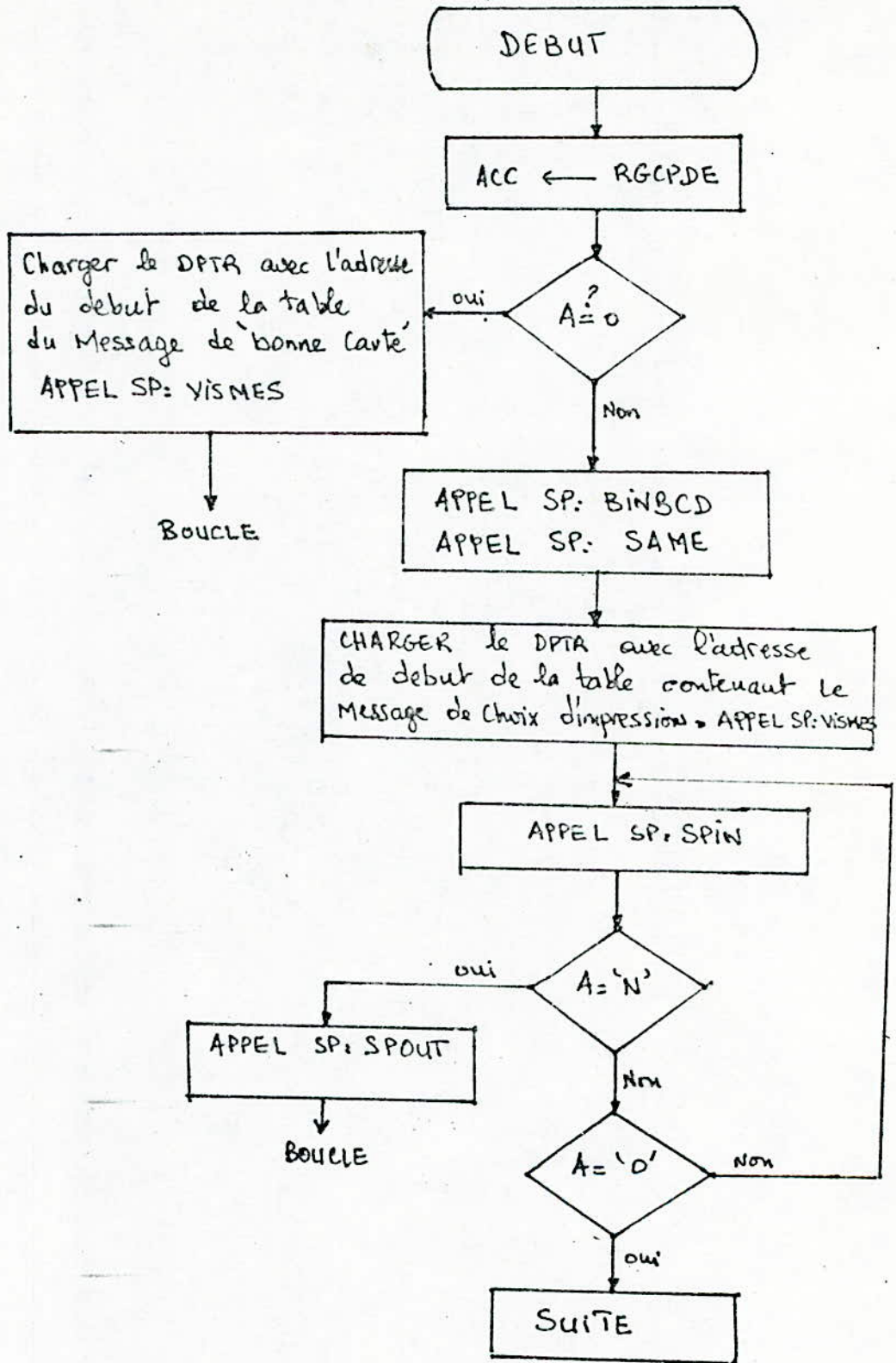
- Registre RGCPDE
- Caractère envoyé de la vision

C/ ZONE DE TRAVAIL :

- Accumulateur
- Registre RGCPDE
- Registre pointeur DPTR
- Routine VISMES, BINBCD, SAME, SPIN, SPOUT.

D/ PARAMETRES DE SORTIES :

- Se branche sur l'étiquette ou continue.



PROGRAMME D'INITIALISATION DE LA VITESSE
DE TRANSMISSION DU PORT SERIEL DE SON MODE ET DE L'INTER-
RUPTION CORRESPONDANTE AU PROTOCOLE DE L'IMPRIMANTE NOTE
(INTISP)

-----oOo-----

A/ FONCTION :

Tout comme la routine INTISV cette routine a pour fonction d'initialiser les périphériques du 8031 et l'interruption correspondante au protocole à la différence près qu'il s'agit de le faire pour l'imprimante.

B/ PARAMETRES D'ENTREES :

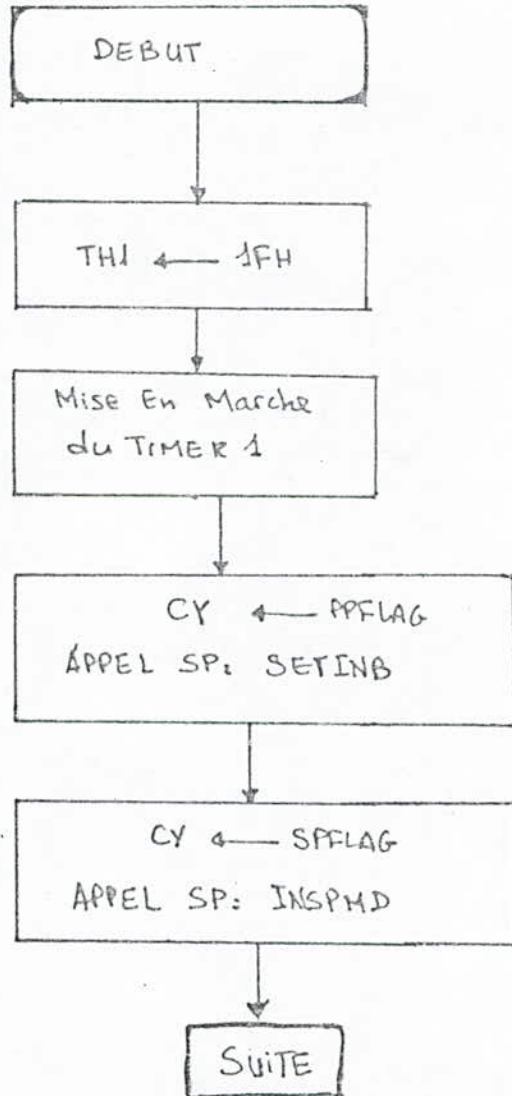
- Contenu de la zone mémoire 1FH
- bit PPFLAG : bit protocole d'échange avec l'imprimante
- bit SVFLAG : bit indiquant le mode (10 ou 11 bits) d'échange avec l'imprimante.

C/ ZONE DE TRAVAIL :

- Routine : INSPMD
- Routine : SETINB
- Registre THI du Timer 1
- Zone mémoire 1FH

D/ PARAMETRE DE SORTIE :

- Chargement de THI à la vitesse désirée et déclenchement du Timer 1
- Ceux de INSPMD et de SETINB



PROGRAMME
IMPRESSION DES DEFAUTS

-----000-----

A/ FONCTION :

Sa fonction est de relire la partie de la RAM de sauvegarde où sont placées les adresses de défauts trouvés lors du dernier test (il s'agit du test complet : cad les N configuration) de prendre 4 bytes par 4 (2adresses de départ 2 pour l'arrivée) de retrouver la nature de la panne et de l'imprimer.

B/ PARAMETRES D'ENTREES :

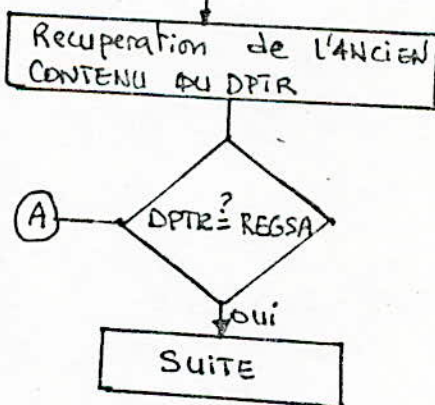
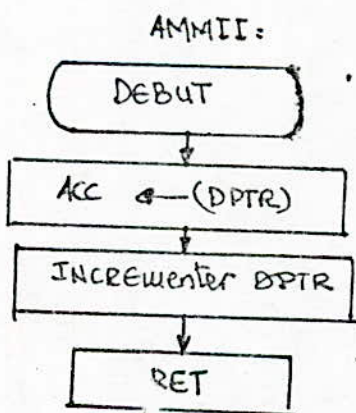
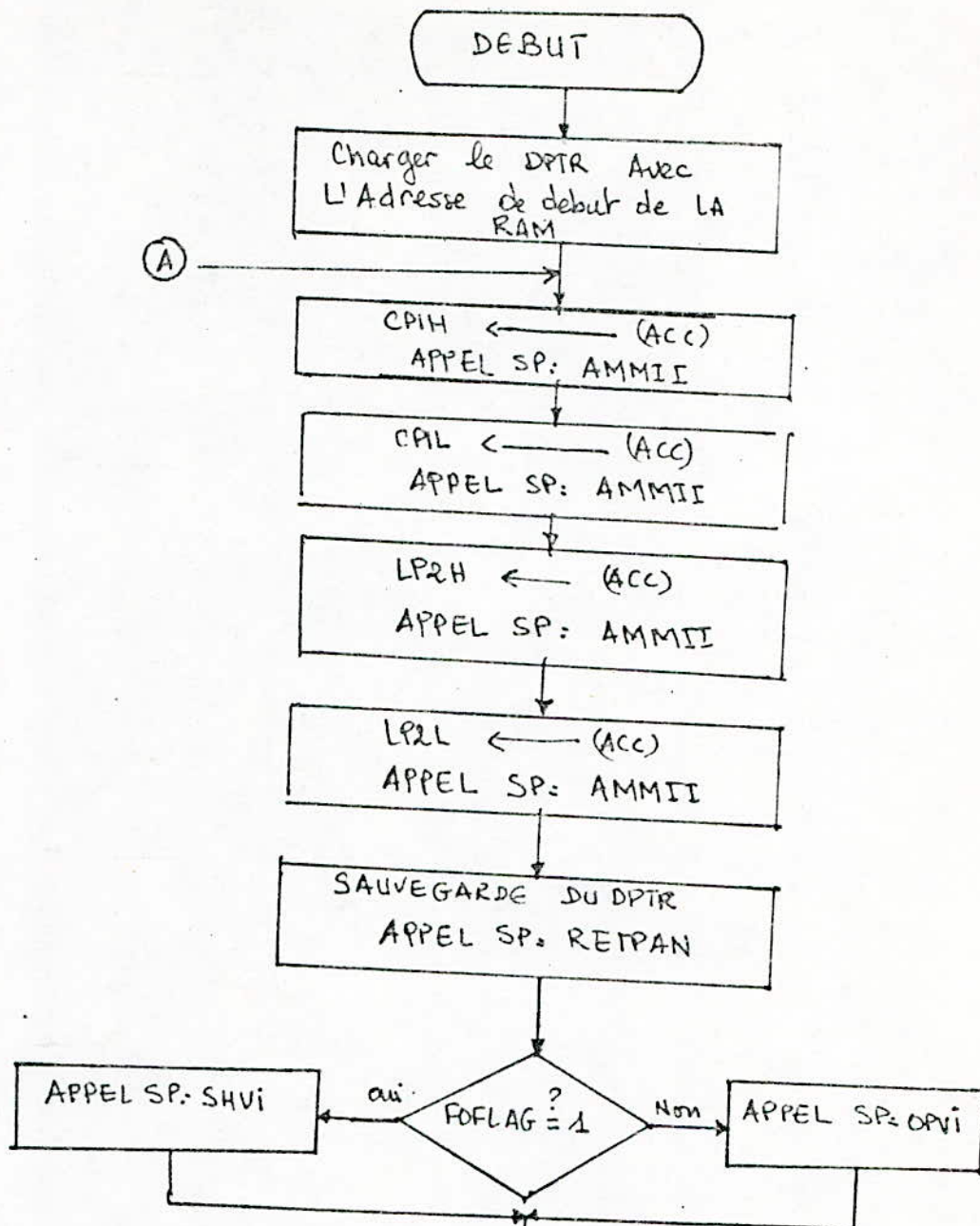
- RAM sauvegarde avec ces 2 pointeurs REGSAH et REGSAL

C/ ZONE DE TRAVAIL :

- Registre pointeur à 16 bits : DPTR
- Registre CP1H et CP1L , LP2H et LP2L
- Pile interne
- Bit FOFLAG
- Routines : RETPAN, OPVI, SHVI.

D/ PARAMETRES DE SORTIES :

- Aucun.



PROGRAMME DE SAUVEGARDE D'ADRESSE
D'AFFICHAGE SUR VISU DE L'INFORMATION DE
COUPURE : ETIQUETTE - SPS1

-----oOo-----

A/ FONCTION :

Lors de la détection d'une coupure, le processeur se dirige vers l'étiquette SPS1, il sera fait appel alors à la routine SPS0 de sauvegarder des adresses départ et arrivée dans la RAM, ensuite il sera fait appel à la routine OPVI qui affiche le N° Chip/Fin départ et N° Chip/Fin arrivée et le défaut soit OPEN (coupure), le registre RGCPDE compteur de défauts est alors incrémenté. Puis il y a retour du programme pour continuer les tests soit à l'étiquette LB4 où il y a le test de fin de bloc.

B/ PARAMETRES D'ENTREES :

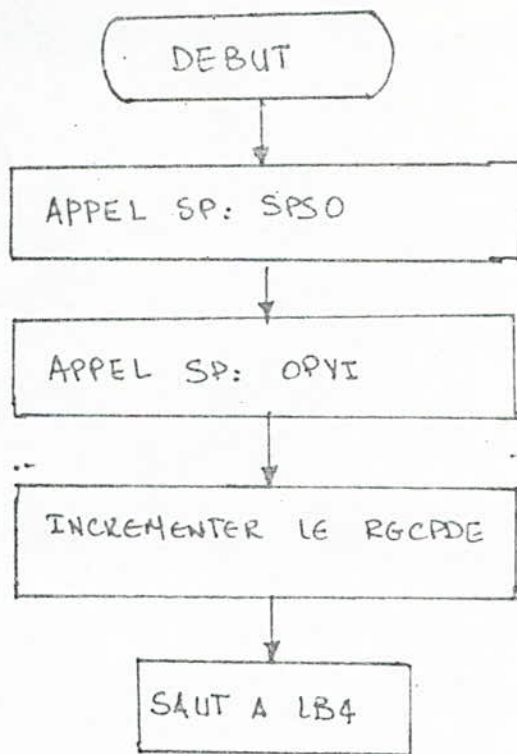
- Ceux des routines OPVI, SPS0.

C/ ZONE DE TRAVAIL :

- Routines OPVI, SPS0
- Registre RGCPDE.

D/ PARAMETRES DE SORTIES :

- Incrémentation de RGCPDE.



PROGRAMME DE TRAITEMENT DE
SAUVEGARDE ET D'AFFICHAGE DU DEFAUT DE CIRCUIT
ETIQUETTE SPS2.

-----oOo-----

A/ FONCTION :

Cette partie du programme a pour fonction de décider de la validation ou de la non validation d'un défaut de court-circuit.

D'abord celle-ci regarde si le point d'arrivée d'adresse LP2 en court-circuit avec CP1 se trouve dans un bloc CP3 se situant avant ou après le bloc d'arrivée de CP1.

Donc, si $CP1 < CP3$ pour éviter la répétition des informations de court-circuit, il faut voir si le point LP2 est en court-circuit avec un point de départ CP1' autre que CP1, si c'est le cas il faut voir si CP1' est en court-circuit avec un point du bloc d'arrivée de CP1. Si c'est le cas il ne faut pas valider ce court-circuit mais passer à un autre bloc, dans tous les cas, autre que ce dernier, il faut sauvegarder donc valider cette panne de court-circuit, l'afficher et passer au bloc suivant.

Si $CP1 > CP3$, il faut voir si la liaison point de départ correspondant à LP2 avec LP2 même est ouverte ou non.

Si la liaison est fermée il ne faut pas valider ce court-circuit.

Si la liaison est ouverte il faut refaire les mêmes opérations citées précédemment pour éviter la répétition des informations de court-circuit.

B/ PARAMETRES D'ENTREES :

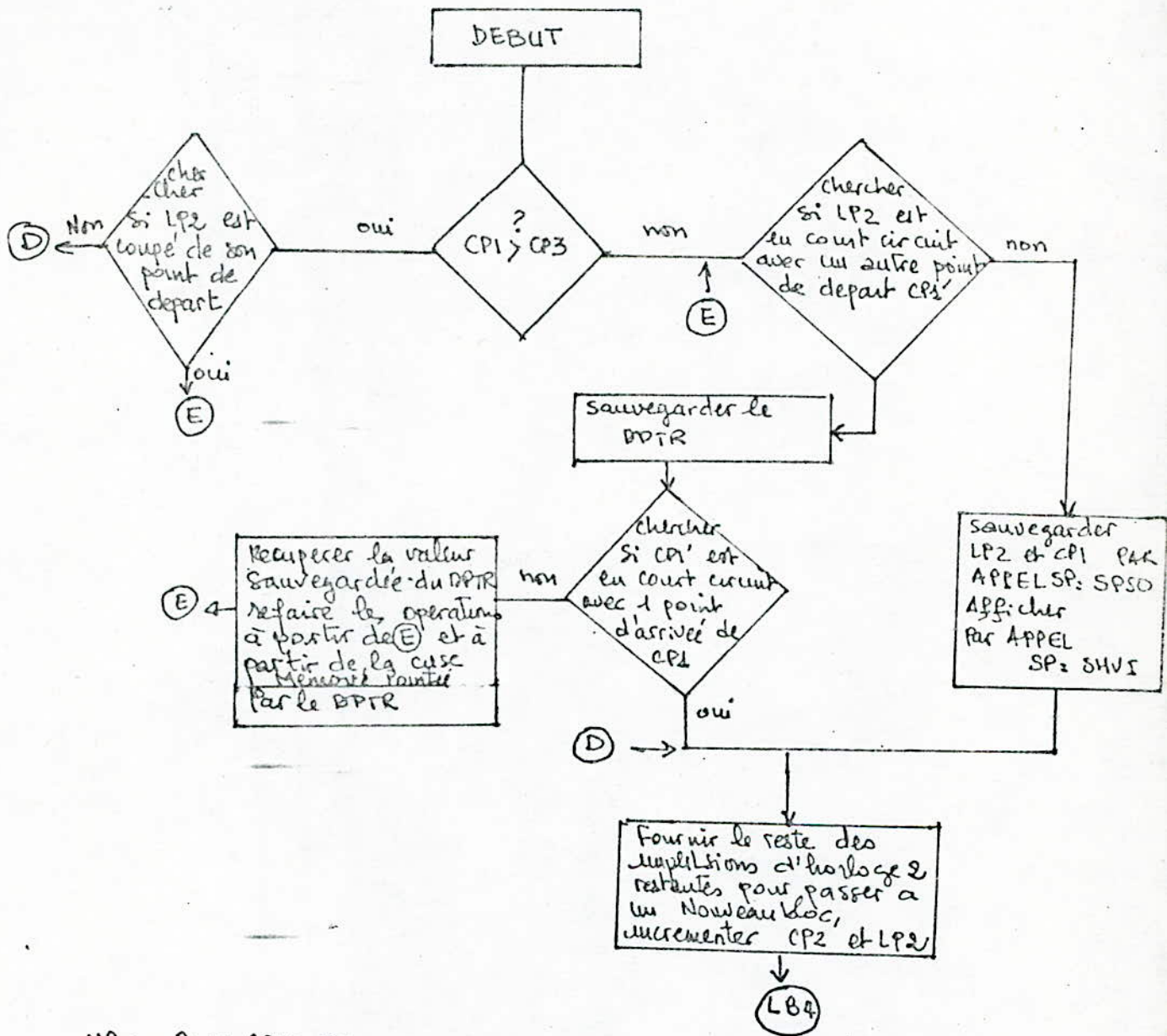
- Registres d'adresses CP1H, CP1L, LP2H, LP2L
- Partie de la zone mémoire RAM écrite depuis le début du test.

C/ ZONE DE TRAVAIL :

- Partie de la RAM écrite depuis le début du test
- Les registres CP1H, CP1L, LP2H, LP2L, RGNDPH, RGNPDL, DPTR, RGGPDE, REGSAH, REGSAL.
- Le carry, le bit FOFLAG
- Les routines RETPAN, SPS0, SMVI.

D/ PARAMETRES DE SORTIES :

- Si validation du court-circuit → incrémentation du registre RGSPEE, puis passage à un nouveau bloc, sinon passage à un nouveau bloc.



NB: Pour $CPI > CP3$:

Si dans un point quelconque de l'organigramme on a dépassé la zone Mémoire de la RAM écrite pendant le test en cours sans trouver ce qu'on cherche ou sauvegarde la adresse etc..

Pour $CP1 > CP3$

si après le TEST on est sorti à travers (E) ou retourne de la même façon que pour $CPI > CP3$

si après le TEST on est sorti par D, il n'y a pas de problème mais si pendant le Test on a terminé la partie de la RAM valide on ne sauvegarde Pas.

3eme PARTIE

ROUTINE DU VECTEUR D'INTERRUPTION DU
PORT SERIEL

-----oOo-----

A/ FONCTION :

Elle sert à arrêter le processeur si un code ASCII 13H est reçu par le RXD, et reste dans le Vecteur jusqu'à recevoir un ASCII ^{11H} par le RXD, si un autre code que le code 13H ou 11H est reçu le processeur sort de la routine avec le code dans le SBUF si le SBUF ne reçoit pas une autre caractère. Elle sert exclusivement pour la réception, si c'est l'émission qui a demandé l'interruption (cad Ti = 1) le processus ressort et continue son chemin.

B/ PARAMETRES D'ENTREES :

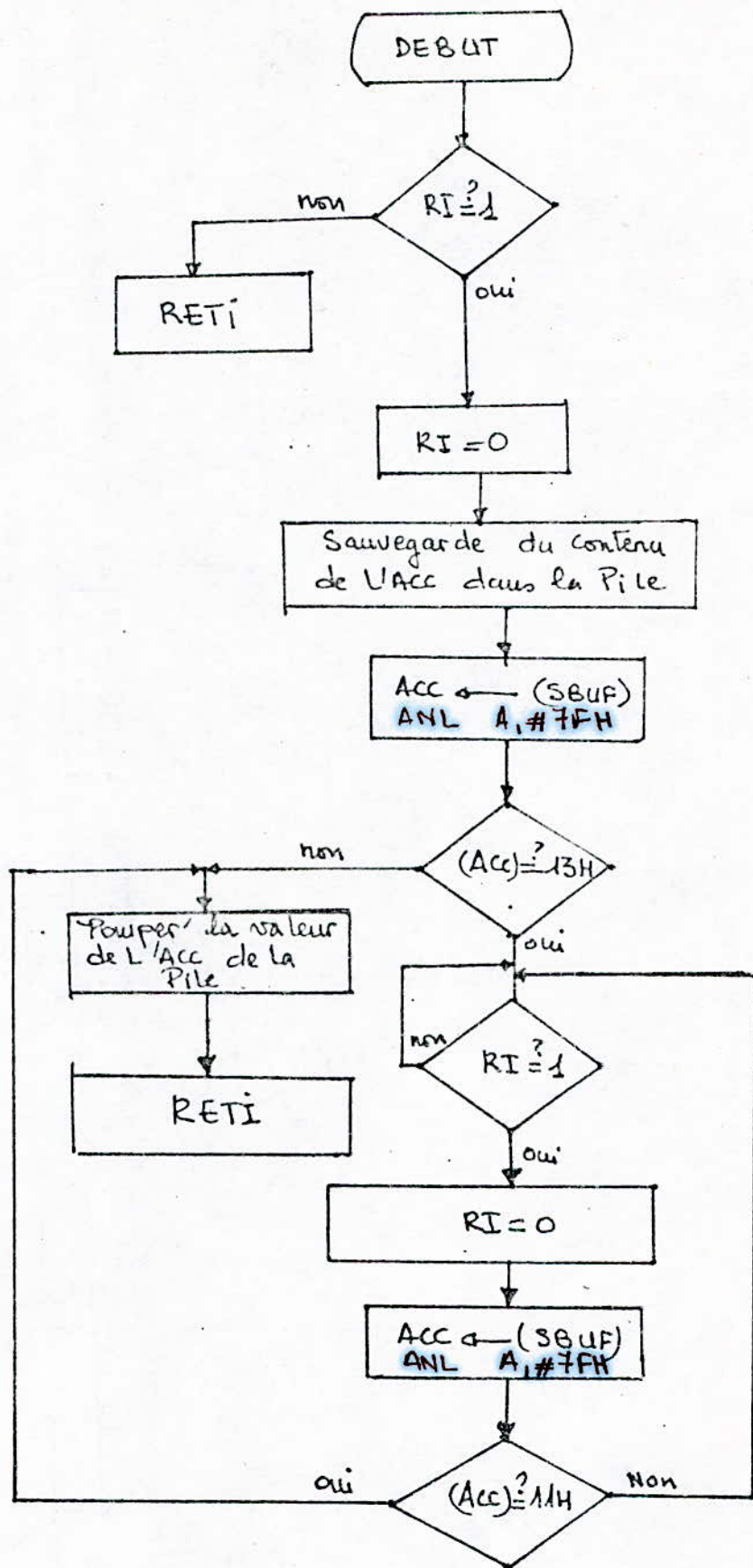
- Ti ou Ri positionnés à 1

C/ ZONE DE TRAVAIL :

- Adresse vecteur d'interruption 23H
- Bit Ri
- ACC
- SBUF

D/ PARAMETRES DE SORTIES :

- Ri à zéro
- Caractère reçu est dans le SBUF.



ROUTINE DU VECTEUR D'INTERRUPTION
EXTERIEURE ZERO (0)

-----oOo-----

A/ FONCTION :

Sert à arrêter le processeur dans le protocole DTR, si la ligne DTR qui normalement est à 12V passe à -12V. Le processeur entre dans le vecteur d'interruption de EX0 et n'en ressort que si le DTR remonte au niveau + 12V c'ad lorsque le niveau sur le Pin INT0 devient haut donc dans le vecteur d'interruption on attend jusqu'à ce que le niveau de la Pin INT0 soit haut.

B/ PARAMETRES D'ENTRES :

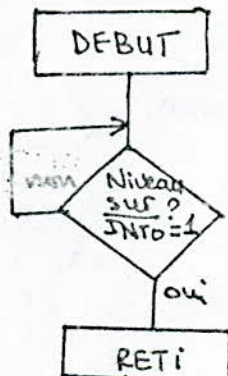
- Bit P3.2 soit INT0

C/ ZONE DE TRAVAIL :

- Adresse vecteur EX0 soit 003H
- P3.2

D/ PARAMETRES DE SORTIES :

- Aucun.



SOUS PROGRAMME DE LECTURE DE LA VITESSE
TRANSMISSION NOTE : RESWBR

-----oOo-----

A/ FONCTION :

Cette routine permet de lire les Switchs de la vitesse de transmission de la visu ou de l'imprimante contenues dans deux zones mémoires de la RAM interne du processus et suivant leurs valeurs changera en retour dans l'accumulateur, la valeur hexadécimal qu'il faudra charger dans le mot de poids fort du Timer 1 (TH1) pour générer la vitesse de transmission voulue.

B/ PARAMETRES D'ENTREES :

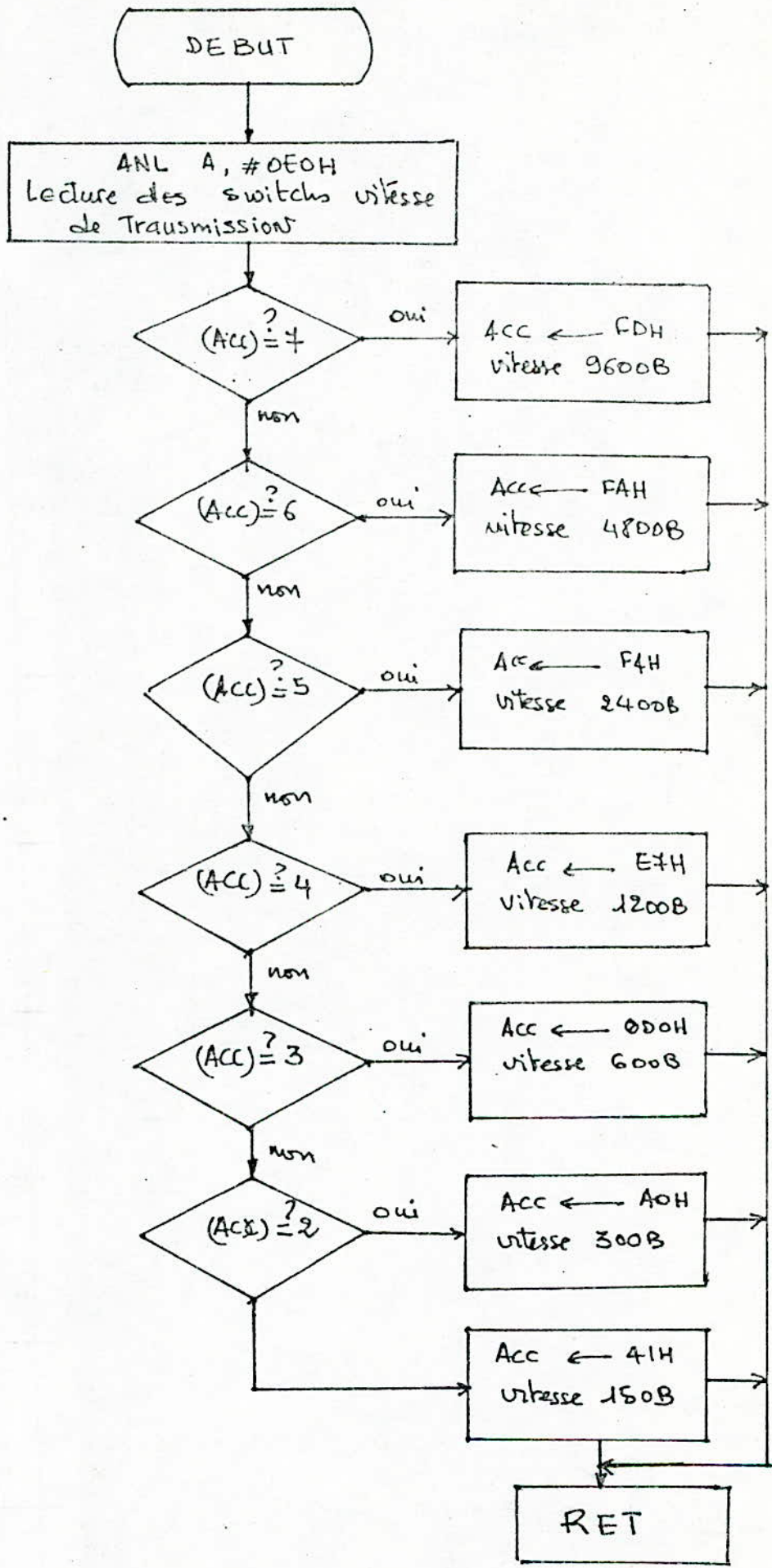
- Valeur du mot Switch visu ou imprimante à mettre dans l'ACC avant d'accéder à la routine.

C/ ZONE DE TRAVAIL :

- L'accumulateur.

D/ PARAMETRES DE SORTIES :

- La valeur hexadécimal qu'il faudra charger dans TH1 pour générer la vitesse en Baud désirée et elle sera contenue dans l'accumulateur.



SOUS PROGRAMME D'AFFICHAGE COMPLET
DU DEFAUT DE COURT-CIRCUIT NOTE : OPVI.

-----300-----

A/ FONCTION :

Cette routine permet par appel à la routine OPSHVI d'afficher les N°s de Chip/Pin départ et les N°s de Chip/Pin arrivée plus l'information de court circuit ('Short') en utilisant pour ce dernier affichage c'ad le 'Short' une table contenant ces caracteres plus la routine VISMES.

B/ PARAMETRES D'ENTREES :

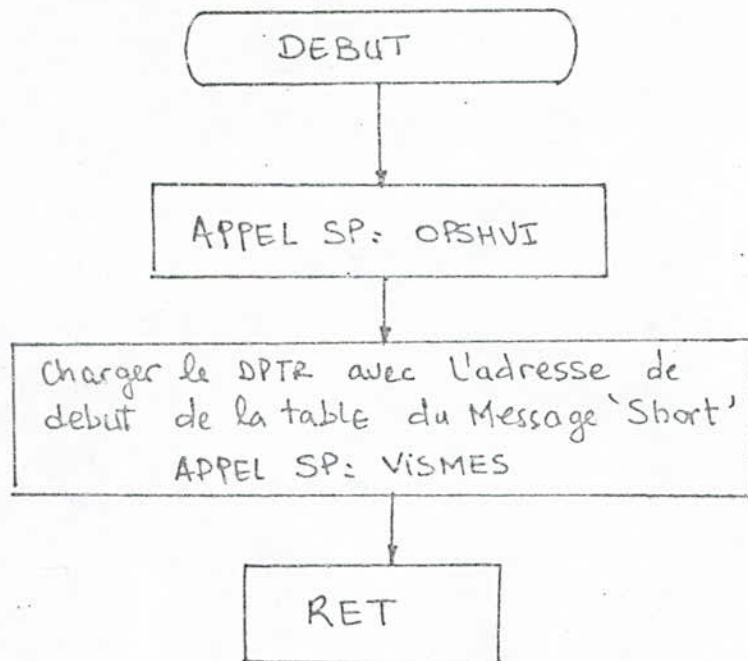
- Ceux de la routine OPSHVI.

C/ ZONE DE TRAVAIL :

- Zone de travail de la routine OPSHVI.
- La table contenant la chaine 'Short'.
- Routine VISMES, DPTR.

D/ PARAMETRES DE SORTIES :

- Aucun.



SOUS PROGRAMME D'AFFICHAGE COMPLET
DU DEFAUT DE COUPURE NOTÉ : SHVI

-----cCo-----

A/ FONCTION :

Cette routine permet par appel à la routine OPSHVI , d'afficher les N° de Chip/Pin départ et les N° de Chip/Pin arrivée plus l'information de coupure ('OPEN') en utilisant pour ce dernier affichage d'ad 'OPEN' une table contenant ces caractères plus la routine VISMES.

B/ PARAMETRES D'ENTREES :

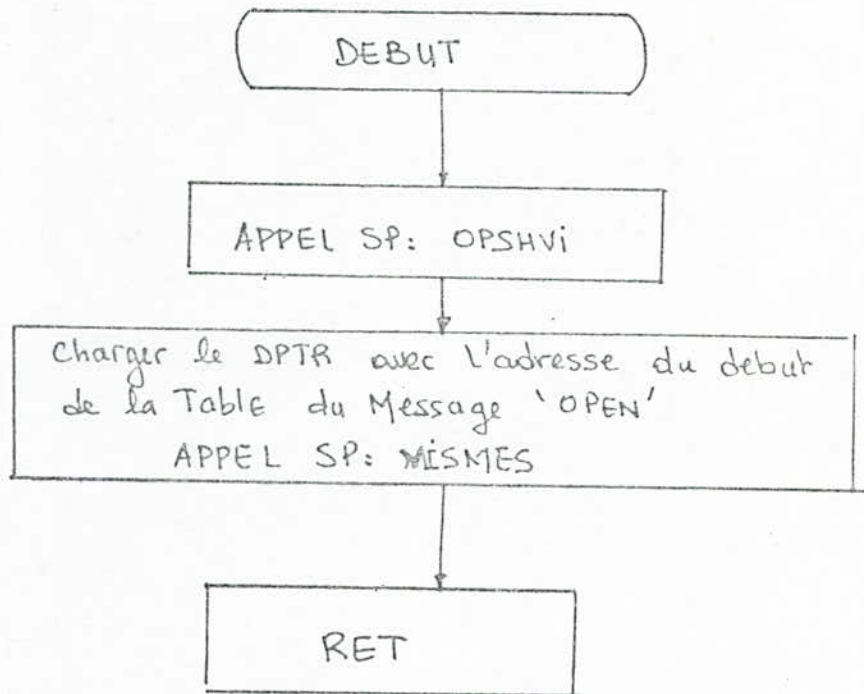
- Ceux de la routine OPSHVI.

C/ ZONE DE TRAVAIL :

- Zone de travail de la routine OPSHVI.
- La table du message 'OPEN'.
- Routine VISMES, DPTR.

D/ PARAMETRES DE SORTIES :

- Aucun.



SOUS PROGRAMME D'AFFICHAGE DES NUMEROS
DE CHIP et PIN ARRIVEE ET DEPART SANS LE DEFAUT
NOTE : OPSHVI.

-----oOo-----

A/ FONCTION :

La fonction de ce sous programme est d'afficher les Numéros de Chip/Pin départ et Chip/Pin d'arrivée quelque soit le défaut.

B/ PARAMETRES D'ENTRES :

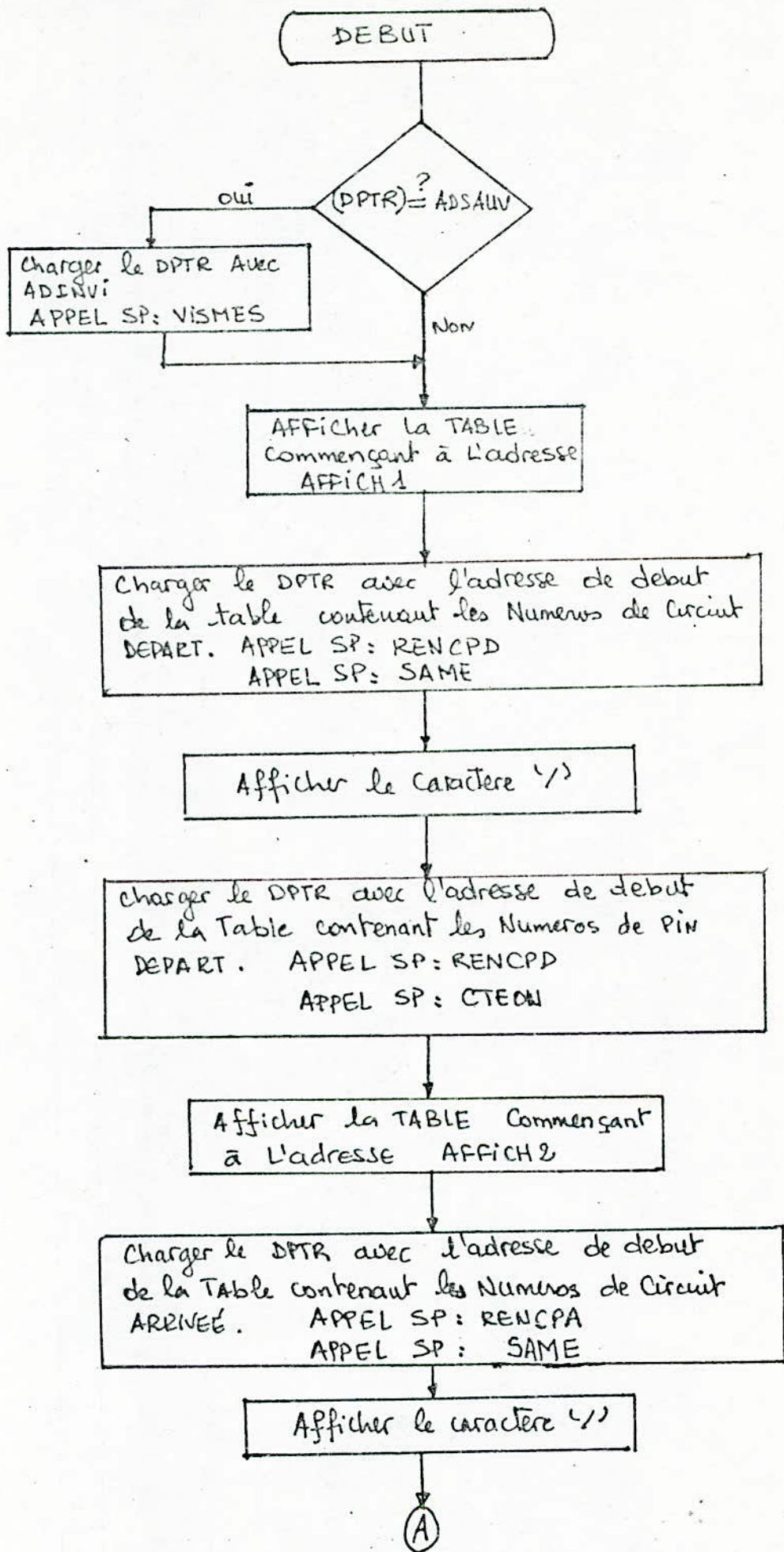
- Adresse de départ, CPIH, CPIL
- Adresse d'arrivée LP2H, LP2L
- Adresse où pointent les registres de gestion de la RAM soit : REGSAH, REGSAL mais l'adresse logée dans le DPTR.

C/ ZONE DE TRAVAIL :

- Accumulateur, DPTR
- Routines : VISMES, RENCPD, SAME, SPOUT, VTEON, RENCPA.
- Tables : tables utilisées par les routines (voir pour chaque routine) plus tables routines.
- ADINVI : adresse début de table contenant le message (Chip/pin Chip/Pin). Affichage lors de la 1ère panne seulement.
- AFICHI : adresse de début de table contenant le message (....U)
- AFICH2 : adresse de début de table contenant le message (....U)
- AFICH3 : adresse de début de table contenant le message ()
toutes terminées par un caractère '0', et,
le caractère ASCII 2FH représentant un slash '/'
à insérer entre le numéro de circuit et de Pin.

D/ PARAMETRE DE SORTIES :

- Aucun/



(A)

Charger le DPTR avec l'adresse de début
de la Table contenant les Numéros de Pins
ARRIVÉE. APPEL SP: RENCPA
APPEL SP: CTEON

Afficher la TABLE commençant
à l'adresse AFFICH3

RET

SOUS PROGRAMME DE LECTURE DU NUMERO
DE CHIP OU DE PIN D'UN POINT DE DEPART NOTE :
RENC PD.

-----oCCo-----

A/ FONCTION :

La fonction de ce sous programme est qu'à partir de l'adresse d'un point de départ, le sous programme se pointe soit sur la table binaire des numéros des circuits de départ soit des Numéros des Pins de départ lit le Numéro de circuit ou de Pin correspondant au point de départ considéré, le place dans l'accumulateur puis appelle la routine de conversion BINBCD ; de façon à pouvoir afficher l'information de défauts d'une façon compréhensible par le dépanneur.

B/ PARAMETRES D'ENTREES :

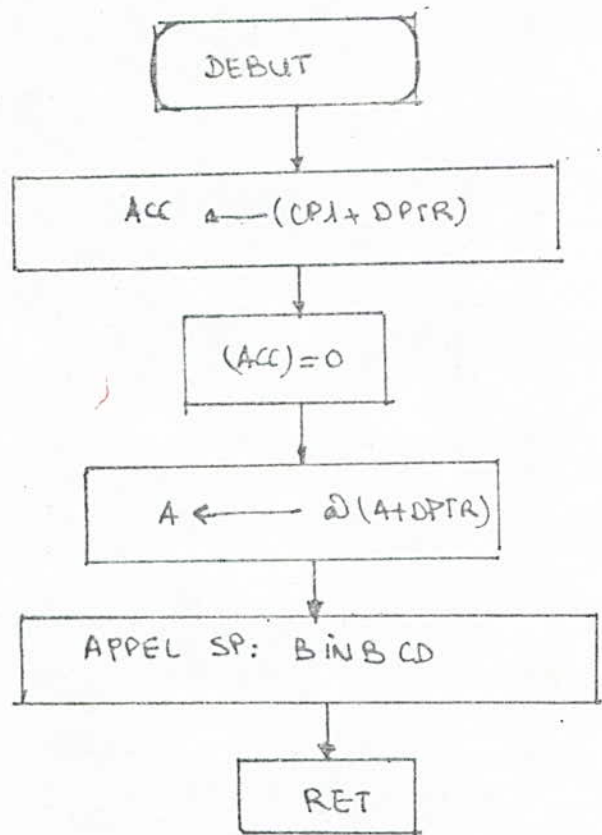
- L'adresse du début de la table de départ (soit la table des N°s de circuit, soit celle des N°s des Pins) qui doit être chargée dans le registre pointeur (DPTR) avant d'appeler la routine.
- L'adresse du point de départ où on a trouvé le défaut celle-ci est contenue dans 2 registres, adresse de départ en l'occurrence CP1H et CP1L.

C/ ZONE DE TRAVAIL :

- Le registre pointeur de données à 16 bits DPTR
- L'accumulateur
- Les 2 registres adresse de départ CP1 et CP1L
- Sous programme de conversion BINAIRE BCD

D/ PARAMETRES DE SORTIES :

- Le numéro de circuit ou de Pin du point de départ considéré en 3 digits BCD placés dans les registres de la routine BNBCD soient HUND et TENONE.



SOUS PROGRAMME DE LECTURE DU NUMERO
DE CHIP OU DE PIN D'UN POINT D'ARRIVEE
NOTE : RENCPA.

-----oO-----

A/ FONCTION :

Semblable à celle du sous programme RENCPD à la différence près que cette routine s'occupe des points d'arrivées.

En effet, le SP se pointe soit sur la table des Numéros de Circuits d'Arrivées soit sur la table des Numéros de Pins d'Arrivées, lit le mot linaire contenu dans la zone pointée et le change dans l'accumulateur pour ensuite appeler la routine BIN BCD.

B/ PARAMETRES D'ENTREES :

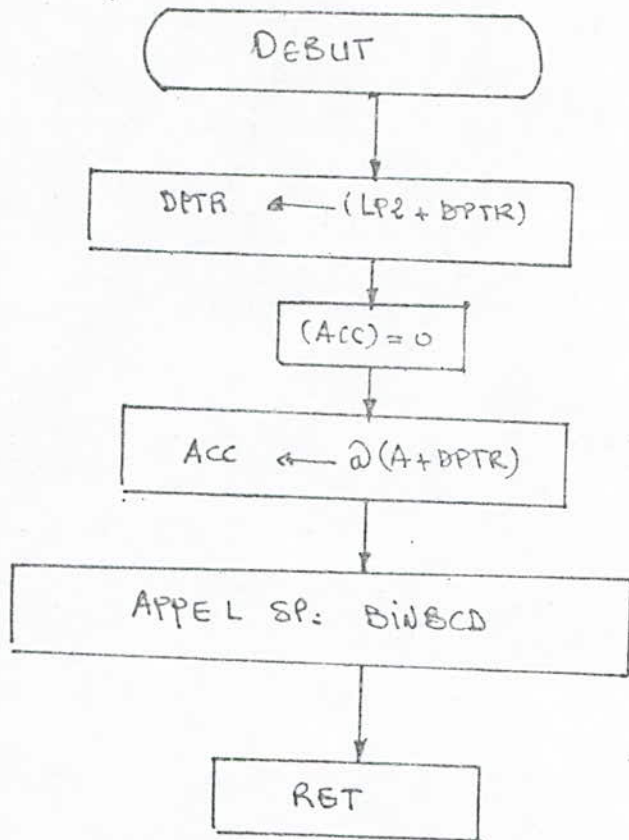
- L'adresse du début de la table d'arrivée considérée celle devant être chargé dans le registre pointeur DPTR avant d'appeler la routine.
- L'adresse du point d'arrivée où on a trouvé le défaut, celle-ci est contenue dans 2 registres adresse d'arrivée absolue, en l'occurrence LP2H et LP2L

C/ ZONE DE TRAVAIL :

- Registre pointeur de données DPTR
- L'accumulateur
- Les 2 registres adresse d'arrivée absolue LP2H et LP2L
- Sous programme de conversion binaire BCD.

D/ PARAMETRES DE SORTIE :

- Le Numéro de circuit ou de Pin du point d'arrivée considéré en 3 digits BCD placés dans les registres HUND et TENONE.



SOUS PROGRAMME D'AFFICHAGE DU NUMERO DE
CHIP OU DE PIN DEPART OU ARRIVEE - NOTE :
. SAME POUR L'AFFICHAGE DES N°S CIRCUITS
. CTEDN POUR L'AFFICHAGE DES N°S PINS.

-----oOo-----

A/ FONCTION :

Le rôle de cette routine est qu'après avoir lu le numéro de CHIP ou de Pin du point de départ ou d'arrivée où on a trouvé un défaut grâce notamment à la routine RENCPC ou RENCPC afficher le numéro de Chip ou de Pin d'un point de départ ou d'arrivée. Cette routine, entre autre, affiche le numéro de Pin en 2 digits BCD au lieu de 3 digits car il n'existe pas à notre connaissance des circuits ayant plus de 99 Pins. De même que pour le numéro de Chip (circuit) si le 1er digit est nul il sera affiché comme un blanc pour ne pas encombrer l'affichage.

Ainsi, pour afficher le N° CHIP appeler la routine SAME
pour afficher le N° de PIN appeler la routine CTEDN.

B/ PARAMETRES D'ENTREES :

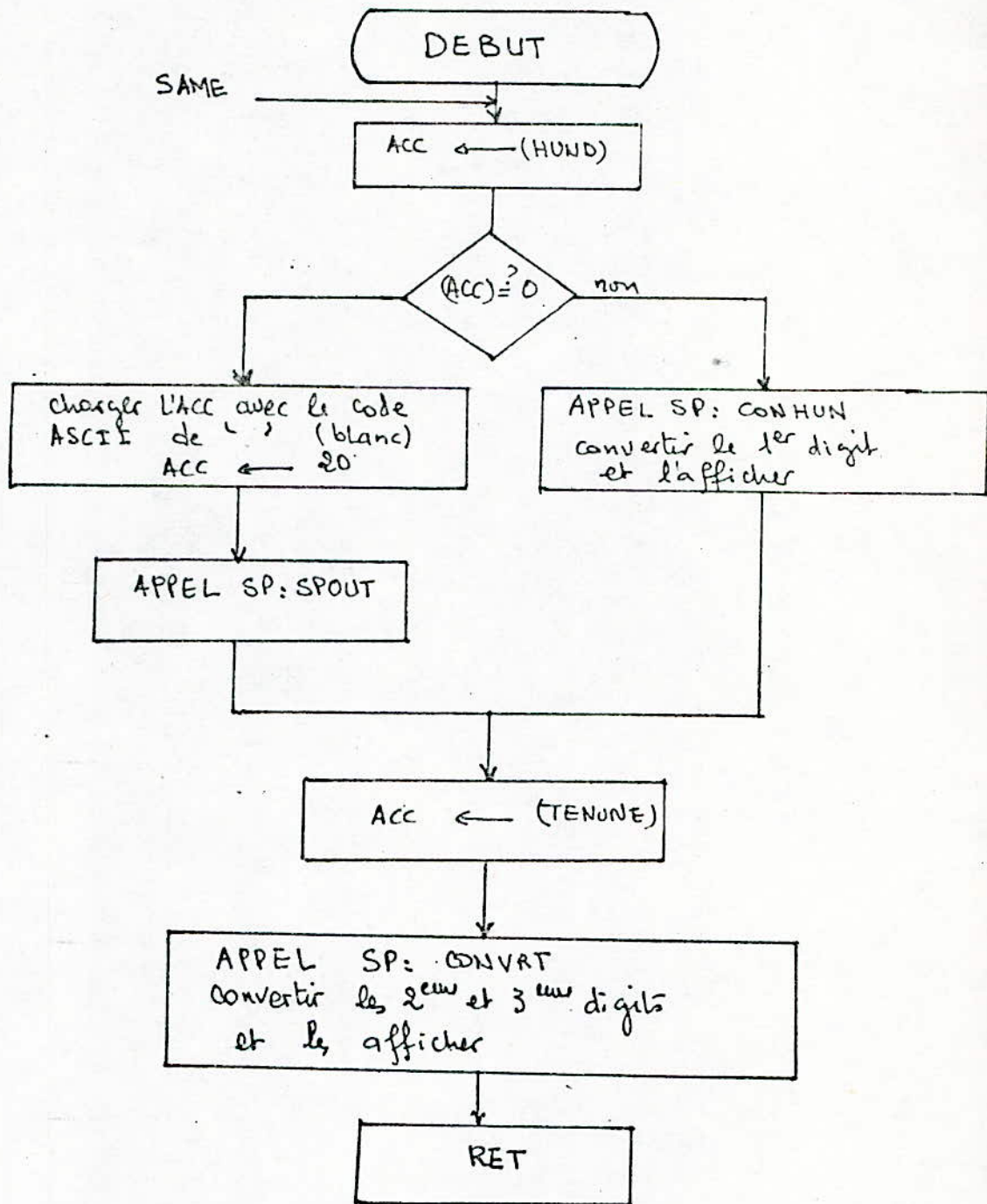
- Le numéro que l'on désire afficher, celui-ci devant être en B.C.D. à 3 digits dans deux registres HUND et TENONE.

C/ ZONE DE TRAVAIL :

- Registre HUND et TENONE
- Registre R23
- Accumulateur
- Sous programme CONHUN
- Sous programme CONVRT
- Sous programme SPOUT

D/ DEPARTEMENTS DE SORTIES :

- Aucun.



SOUS PROGRAMME DE CONVERSION BINAIRE - BCD

NOTE : BIN BCD

-----oOo-----

A/ FONCTION :

A pour rôle de convertir un nombre binaire placé dans l'ACC en un nombre BCD (binaire codée décimal) à 3 digits et le placer dans deux registres HUND (contenant les centaines) et TENONE (contenant les dizaines : 1er digit et les unités : 2ème digit).

B/ PARAMETRES D'ENTREE :

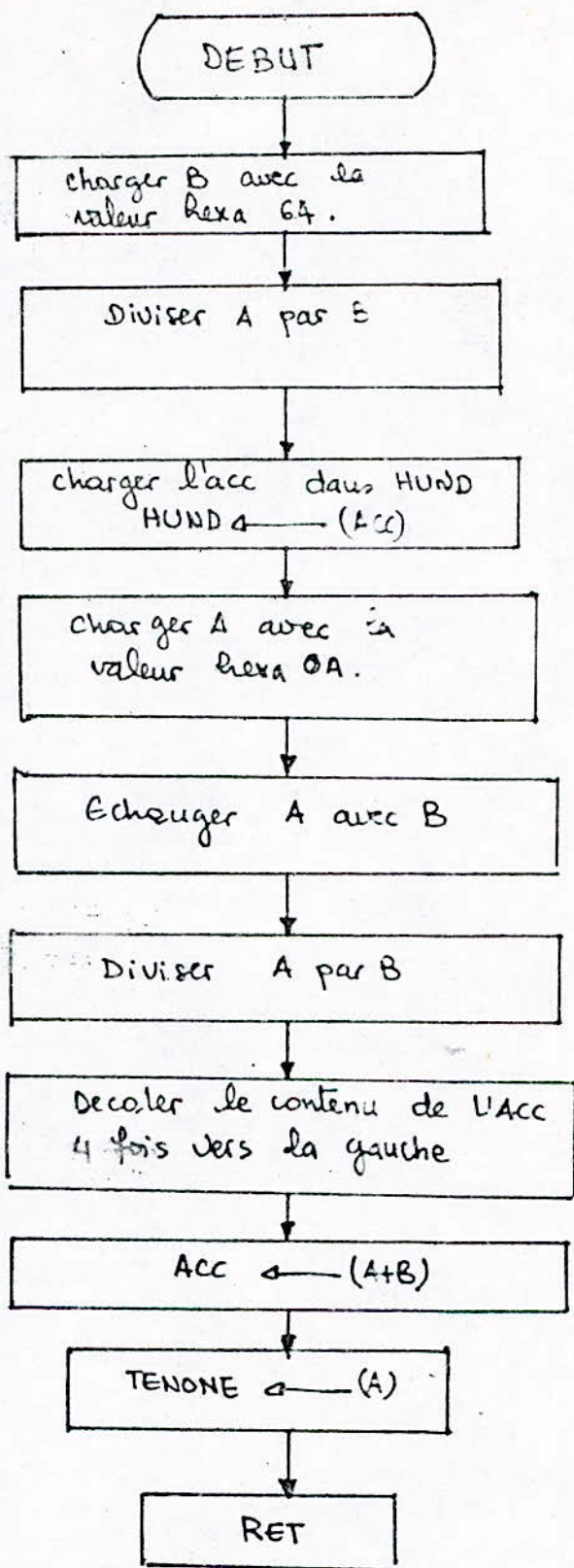
- Le nombre binaire à convertir (à placer dans l'ACC avant d'accéder à la routine).

C/ ZONE DE TRAVAIL :

- L'accumulateur
- Le registre B

D/ PARAMETRES DE SORTIES :

- Le digit des centaines sera contenu dans le registre HUND.
- Le digit des dizaines et celui des unités seront dans le registre TENONE.



SOUS PROGRAMME DE CONVERSION D'UN MOT
BCD EN ASCII ET SON AFFICHAGE
NOTE : CONVET (2 digit) ou CONHUN (1 digit)

-----oOo-----

A/ FONCTION :

A pour rôle de convertir 2 digits BCD placés dans un registre R23 en 2 codes ASCII et de les afficher sur visu ou de les imprimer sur imprimante. Elle peut aussi ne convertir qu'un seul digit placé dans le digit de poids fort de R23 et l'afficher, ceci est utilisé pour convertir et afficher les centaines.

B/ PARAMETRES D'ENTREES :

- Le mot à 2 digits BCD, à 1 digit BCD doit être contenu dans le registre R23.

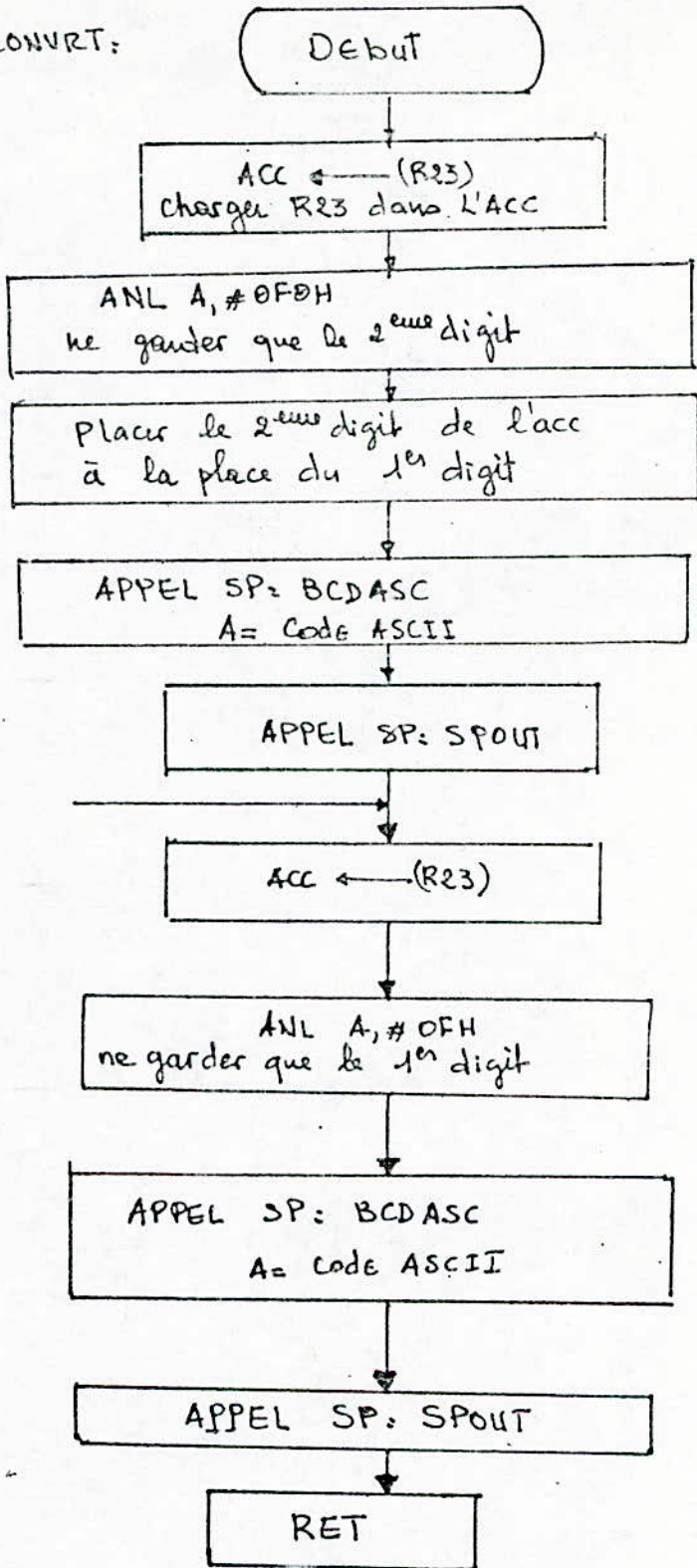
C/ ZONE DE TRAVAIL :

- Registre R23 du BANK 2 de la RAM interne au processeur.
- l'Accumulateur
- Sous programme BCD ASC
- Sous programme SPOUT.

D/ PARAMETRES DE SORTIES :

- Aucun.

CONVRT:



CONHUN:

SOUS PROGRAMME TABLE DE CONVERSION
D'UN DIGIT BCD EN ASCII. NOTE : BCDASC.

-----oOo-----

A/ FONCTION :

Convertit un digit BCD contenu dans l'accumulateur en code ASCII correspondant et le chargera dans l'ACC.

B/ PARAMETRES D'ENTREES:

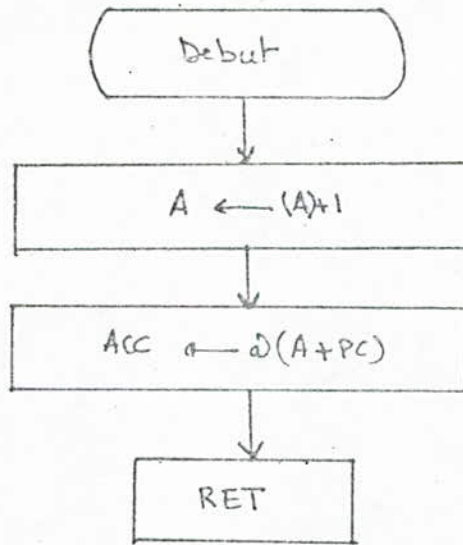
- Le digit BCD à convertir à placer dans le digit poids faible de l'accumulateur.

C/ ZONE DE TRAVAIL :

- Compteur programme (PC)
- L'accumulateur
- La table d'équivalence BCD - ASCII insérée juste après le RET (RETURN From Subroutine) de façon à utiliser le PC + l'accumulateur pour pointer une donnée de la table et la charger dans A.

D/ PARAMETRES DE SORTIES :

- Le code ASCII du digit à convertir : placé dans l'accumulateur.
- La table devant se trouver juste après le RET, le nombre à convertir est d'abord additionné avec un (1) et chargé dans A de façon à pointer sur la table la donnée correspondante ceci a cause du RET qui occupe 1 byte.



SOUS PROGRAMME DE RETROUVAILE
DE LA NATURE DE LA PANNE NOTE : RETPAN

-----oOo-----

A/ FONCTION :

A partir de la donnée d'une adresse de départ quelconque CP1H et CP1L et, d'une adresse d'arrivée quelconque LP2H et LP2L cette routine permet de dire si le point d'arrivée d'adresse LP2H et LP2L se trouve dans le bloc d'arrivée correspondant à CP1H et CP1L ceci en positionnant 1 bit drapeau FOFLAG :

- à 0 si le point est dans le bloc
- à 1 si le point est extérieur au bloc.

B/ PARAMETRES D'ENTREES :

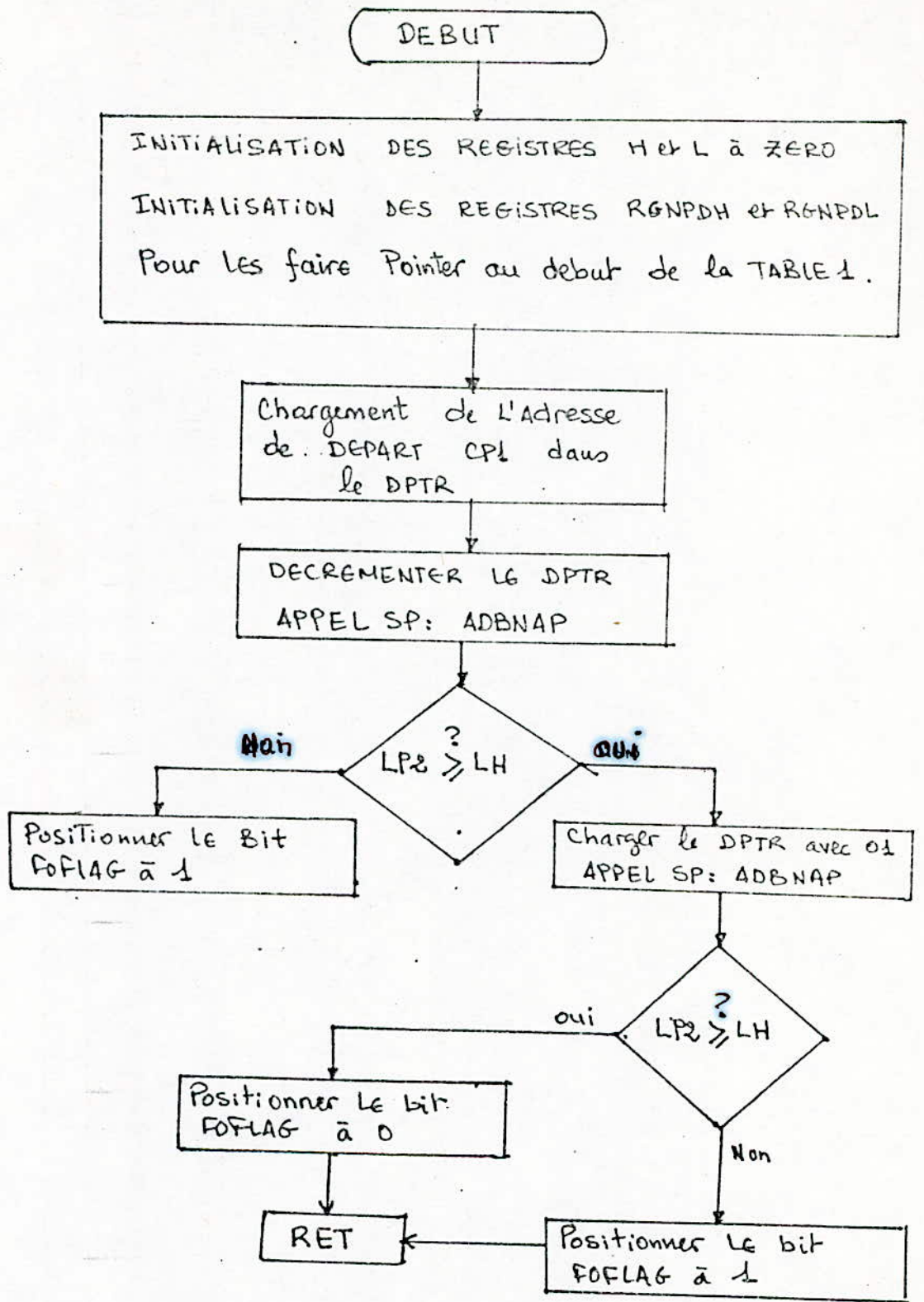
----- adresse de départ contenue dans CP1H et CP1L
----- adresse d'arrivée contenue dans LP2H et LP2L

C/ ZONE DE TRAVAIL :

- Routine ADBNAP
- Registre H et L
- Registre RGNPDH et RGNPDL
- Adresse départ CP1H CP1L
- Adresse arrivée LP2H LP2L
- DPTR
- Accumulateur
- Carry CY
- Bit FOFLAG contenue dans le PSW

D/ PARAMETRES DE SORTIES :

Positionnement du bit FOFLAG à 0 si le point d'arrivée ne fait pas partie du bloc du point de départ sinon il est positionné à 1.



NB: LH est un Enchaînement fictif des registres L et H

SOUS PROGRAMME D'ADDITION DE NOMBRE DE
POINTS D'ARRIVEES PAR BLOC SUCCESSIFS NOTE ADENAP.

-----oOo-----

A/ FONCTION :

Cette routine a pour rôle d'additionner des nombres de points d'arrivées par bloc se suivant sur la table des points d'arrivées par bloc : (EPROM : 2716 de 2K Bytes). En fait cette routine utilise la routine LECT pour lire le nombre de points d'arrivées/bloc sur lequel pointent les registres RGNPDL et RGNPDH puis l'additionne avec le couteau de 2 registres H et L, ceci autant de fois que la valeur contenue dans le registre DPTR (16 bits). Celui-ci étant décrémenté lors de chaque addition, les registres L et H vont donc contenir le résultat des additions.

B/ PARAMETRES D'ENTREES :

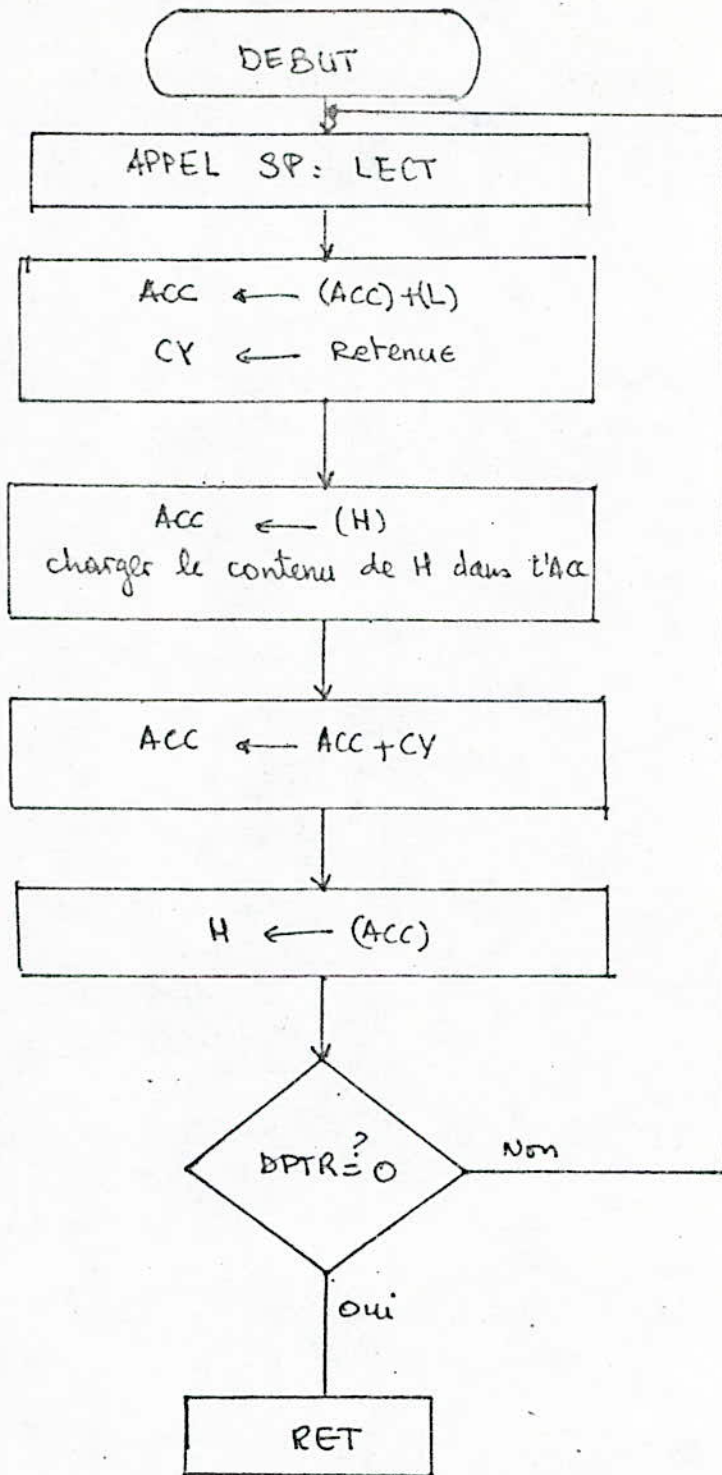
- Charger le registre DPTR avec le nombre d'additions à faire (en général on le charge avec CPIH et CPIL).
- Faire éventuellement pointer les registres RGNPDH et RGNPDL sur la 1ère donnée à additionner.
- Initialiser éventuellement les registres L et H à zéro.

C/ ZONE DE TRAVAIL :

- Routine LECT
- Accumulateur
- Le bit Carry CY
- Deux registres L et H
- Le registre 16 bits DPTR.

D/ PARAMETRES DE SORTIES :

- Les registres L et H contiennent le mot de poids faible (L) et le mot de poids fort (H), résultat des additions successives de nombre de points d'arrivées par bloc.



SOUS PROGRAMME DE LECTURE DU NOMBRE DE
POINTS DE DEPARTS OU DU NOMBRE DE POINTS D'ARRIVEES
PAR BLOC NOTE : LECT

-----000-----

A/ FONCTION :

Une table de 2K mots (EPROM 2716) est réservée pour contenir en 2 bytes le nombre de points de départs (poids fort puis poids faible) puis dans chaque byte le nombre de points d'arrivées par bloc par numéro de point de départ croissant.

Aussi, cette routine a pour fonction de lire la nombre de points de départs une seule fois après un RESET puis de lire successivement le nombre de points d'arrivées par bloc.

Ainsi, il existe deux registres RGNPDH - RGNPDL qui sont utilisés pour sauvegarder la valeur du registre pointeur DPTR avant de sortir de la routine de façon à pointer toujours au prochain appel à la routine sur le byte qui suit le byte tiré lors du dernier appel à la routine.

Ces deux registres sont initialisés avant d'accomplir le tout. 1er test, de façon à pointer le début de la table des 2K octets puis seront réinitialisés avant le prochain test de façon à pointer sur le 1er byte des points d'arrivées par bloc, c'est-à-dire avec un saut de 2 octets à partir du début de la table ; ceci car le nombre de points de départs est sauvegardé dans deux registres NH et NL, donc on a plus besoin de relire la table. Pour le nombre de points de depart . De plus comme le DPTR est utilisé par ailleurs ; il est poussé dans la pile dès le début de la routine et pompé avant la sortie.

B/ PARAMETRES D'ENTREES :

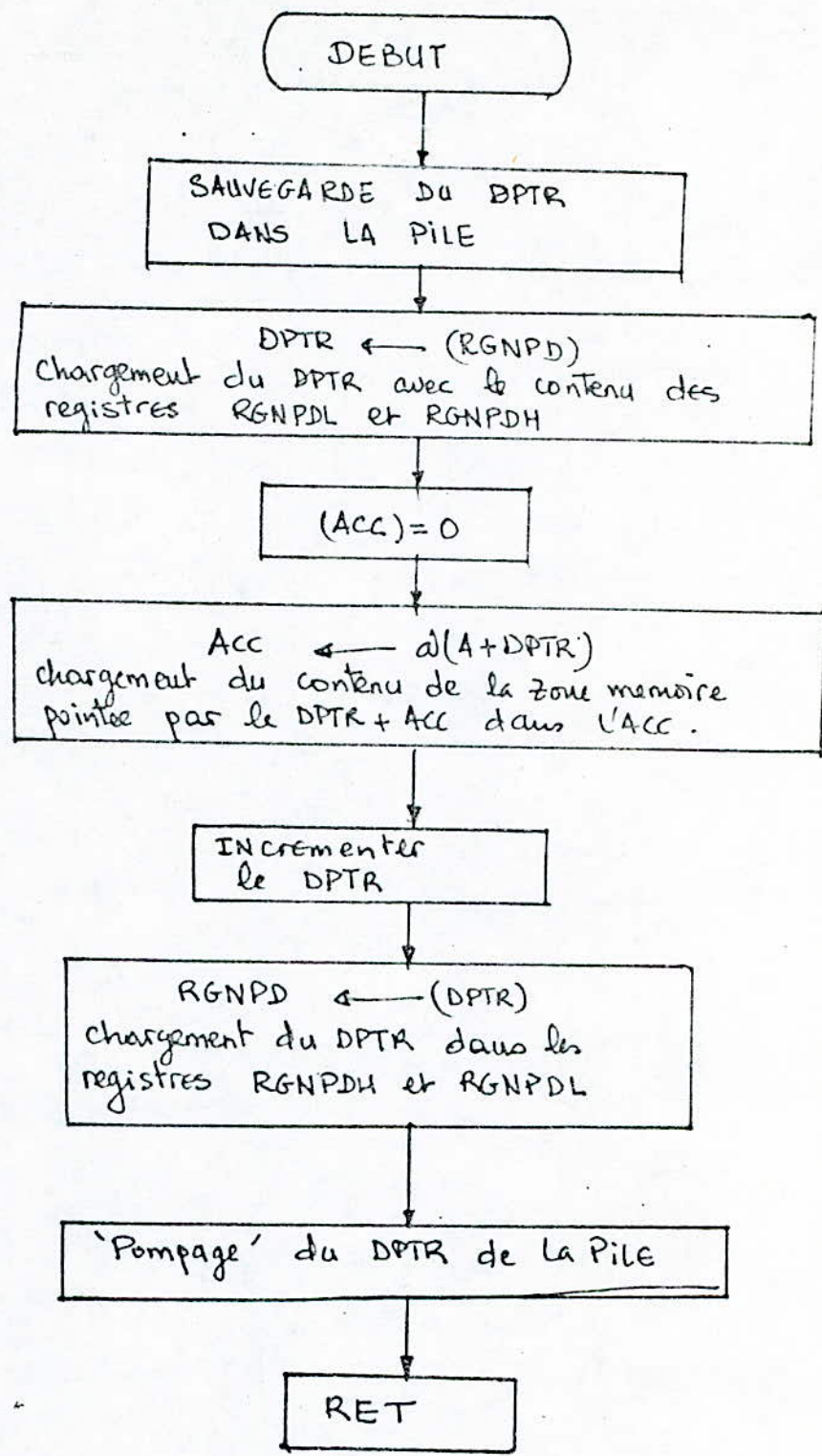
- Aucun.

C/ ZONE DE TRAVAIL :

- Registre pointeur 16 bits DPTR
- Accumulateur
- 2 registres RGNPDL et RGNPDH
- Pile servant à sauvegarder le DPTR.

D/ PARAMETRES DES SORTIES :

- Nombre de points de départs ou nombre de points d'arrivées par bloc : contenu dans l'accumulateur.



SOUS PROGRAMME DE SAUVEGARDE DANS LA
RAM 2128 DES ADRESSES DE DEPART ET D'ARRIVEE APRES
VALIDATION DE LA PANNE NOTE SPS 0

-----oOo-----

A/ FONCTION :

Cette routine a pour rôle de sauvegarder l'adresse de départ (en 2 bytes : CPIH et CPIL) et l'adresse d'arrivée (en 2 bytes : LP2H et LP2L) dans la 2128 après que le 'programme' ait décidé de tenir compte de ce défaut (en particulier les court-circuits).

Il existe dans ce sous programme 2 registres REGSAH et REGSAL chargés de sauvegarder le DPTR de façon à toujours écrire sans qu'il y ait de chevauchement c'est donc le DPTR qui a la tâche de pointer sur la case mémoire dans laquelle on va écrire. Ces registres servent ainsi à gérer l'espace RAM 2128. Après une sortie de la routine ils pointeront vers la future case mémoire. Ceux-ci sont initialisés à chaque nouveau test de façon à pointer sur le début de la RAM.

B/ PARAMETRES D'ENTREES :

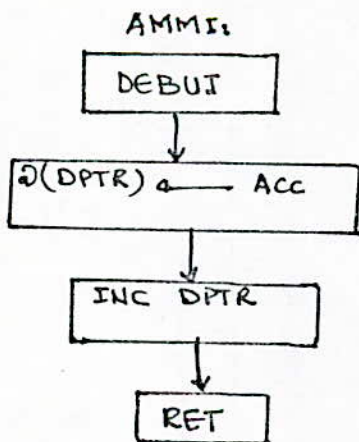
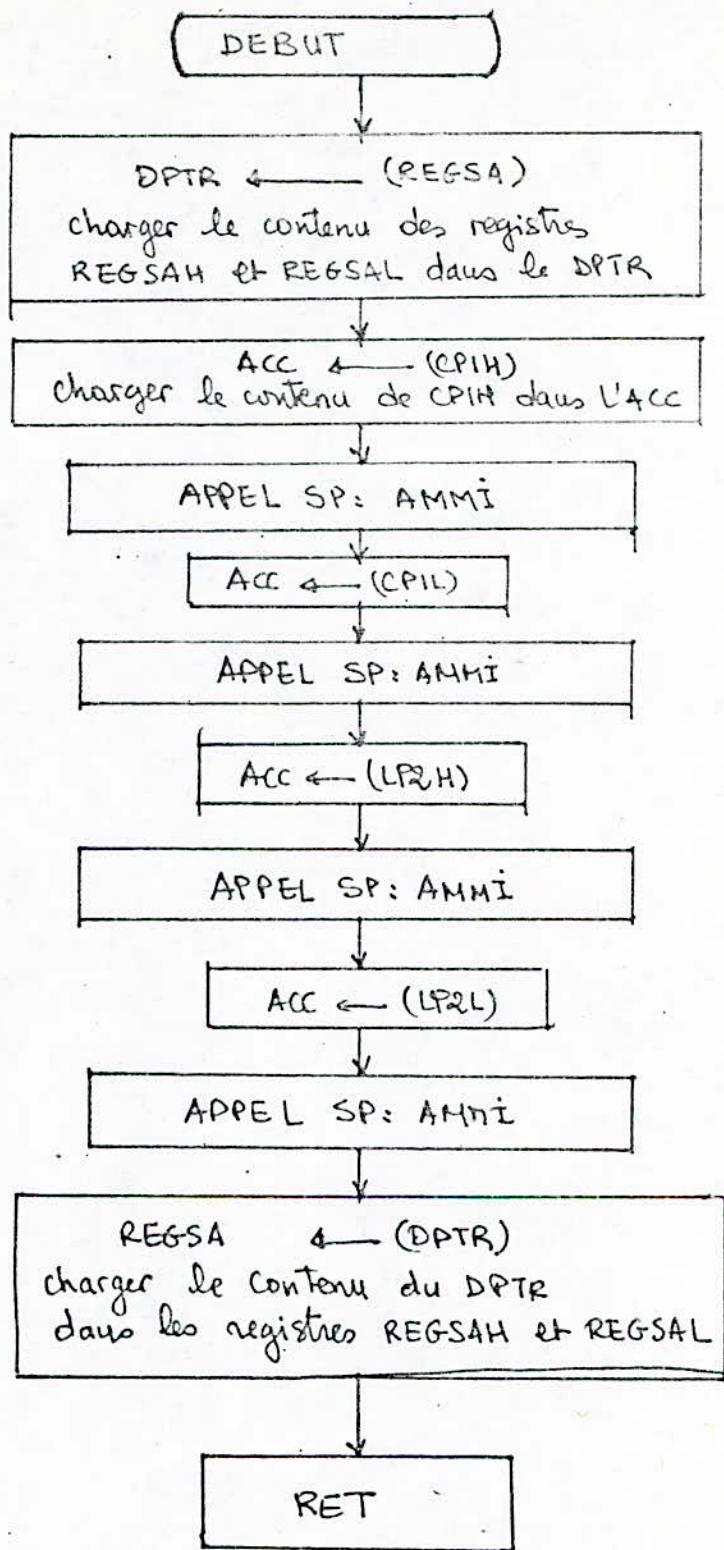
- Adresse de départ contenues dans CPIH et CPIL
- Adresse d'arrivée contenues dans LP2H et LP2L
- Eventuellement les registres REGSAH et REGSAL, si on fait un accès pour la première fois.

C/ ZONE DE TRAVAIL :

- RAM 2K (2128 intel)
- Les registres REGSAH et REGSAL
- Le pointeur de données DPTR
- Les registres CPIH et CPIL LP2H et LP2L
- L'accumulateur
- Routine AMMI : charge la zone pointée par le DPTR avec le contenu de l'ACC, incrémente le DPTR, puis RET.

D/ PARAMETRES DE SORTIES :

- Aucun.



NB: $\partial(x)$ = zone memoire pointée par le registre x.

SOUS PROGRAMME DE VISUALISATION OU
IMPRESSION D'UNE TABLE DE CARACTERES ASCII

NOTE : VISHES

-----oOo-----

A/ FONCTION :

Cette routine a pour fonction de visualiser ou d'imprimer une table de caractères ASCII, aux conditions suivantes :

- L'adresse du début de la table doit être connue
- la table ne doit pas contenir de caractère '⊘';
. insérer à la fin de la table le caractère '⊘', il ne sera pas affiché.

B/ PARAMETRES D'ENTREES :

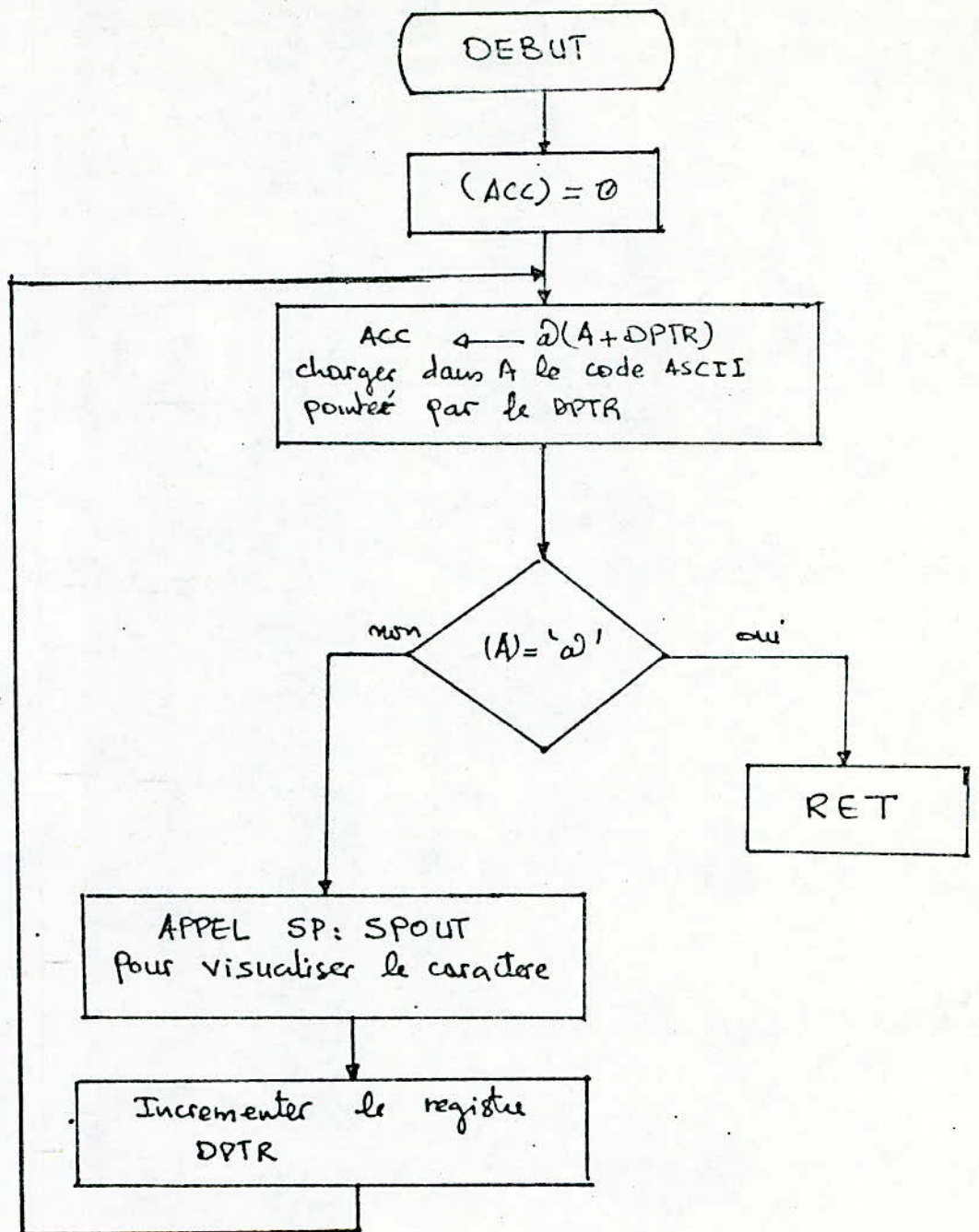
- La table ASCII
- L'adresse du début de la table.

C/ ZONE DE TRAVAIL :

- L'accumulateur
- Le registre à 16 bits pointeur de données (DPTR) dans lequel on chargera au préalable l'adresse du début de la table avant d'appeler la routine.
- La routine de sortie d'un caractère par le port sériel : SPOUT.

D/ PARAMETRES DE SORTIES :

- Aucun.



SOUS PROGRAMME D'ENTREE D'UN CARACTERE
PAR LE PORTE SERIEL NOTE : SPIN

-----oCo-----

A/ FONCTION :

Cette routine permet de faire entrer un caractère contenu dans le registre buffer sériel de réception (SBUF) dans l'accumulateur quelque soit le protocole dans lequel il a été envoyé en ignorant le 8ème bit reçu qui pourrait être soit une parité soit un bit de caractère, ceci vu que les caractères que nous nous attendons à recevoir sont des caractères ASCII ne dépassant pas la valeur Hexadécimale 7FH. Ignorer le 8ème bit et un éventuel 9ème bit (mode 11 bits) suffit amplement dans un système pré réglé tel que le notre.

B/ PARAMETRES D'ENTREES :

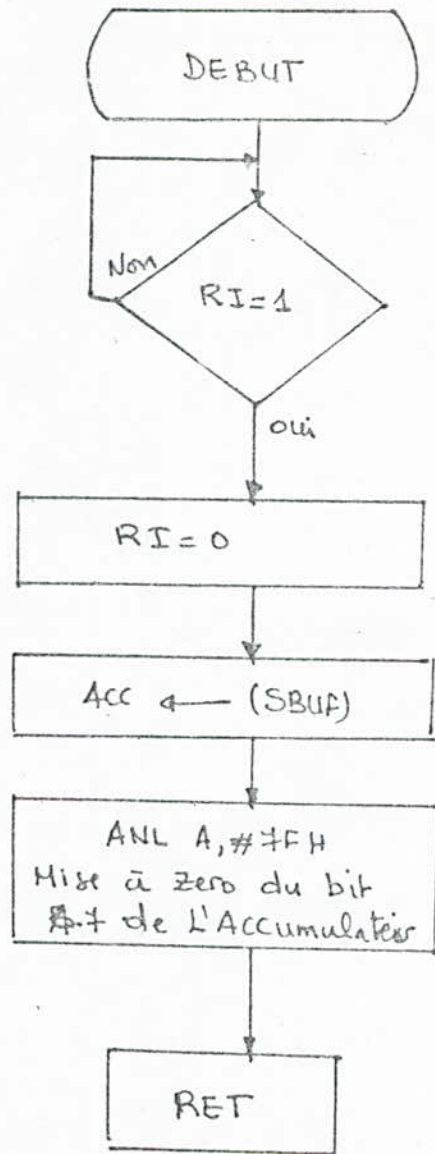
- Le bit Ri du registre de contrôle du port sériel (SCON) qui indique que le port a reçu ou non un caractère suivant qu'il est à un (1) ou à Zéro (0).

C/ ZONE DE TRAVAIL :

- Le registre buffer sériel de réception SBUF
- L'accumulateur

D/ PARAMETRES DE SORTIES :

- Le caractère ASCII reçu par le SBUF écrit en 7 bits (c'est-à-dire que le 8ème bit est annulé) et contenu dans l'accumulateur.



SOUS PROGRAMME DE SORTIE D'UN CARACTERE
PAR LE PORT SERIEL - NOTE : SPOUT.

-----oOo-----

A/ FONCTION :

Cette routine permet la sortie d'un caractère ASCII contenue dans l'accumulateur à travers le port sériel vers une visu ou une imprimante suivant un protocole d'échange donné requérant :

- 7 ou 8 bits pour le caractère ASCII
- Un bit parité ou non et s'il y a lieu parité paire ou parité impaire
- 1 ou 2 bits stop en mode 11 bits
- 10 ou 11 bits pour la chaîne complète.

B/ PARAMETRES D'ENTREES :

- Les valeurs des 2 Switchs réservés aux protocoles d'échanges avec la visu ou le terminal, celles-ci étant contenues dans 2 zones mémoires de la RAM interne du processeur.
- Le caractère ASCII contenu dans l'accumulateur et éventuellement le bit P (parité) du registre d'état
- Le bit P.5 du port 1 d'E/S réservé à la sélection de la visu ou de l'imprimante.
- Le bit Ti du registre de contrôle du port sériel (SCON) qui indique que le port est prêt ou non à recevoir le caractère à sortir suivant qu'il est à un (1) ou à zéro).

C/ ZONE DE TRAVAIL :

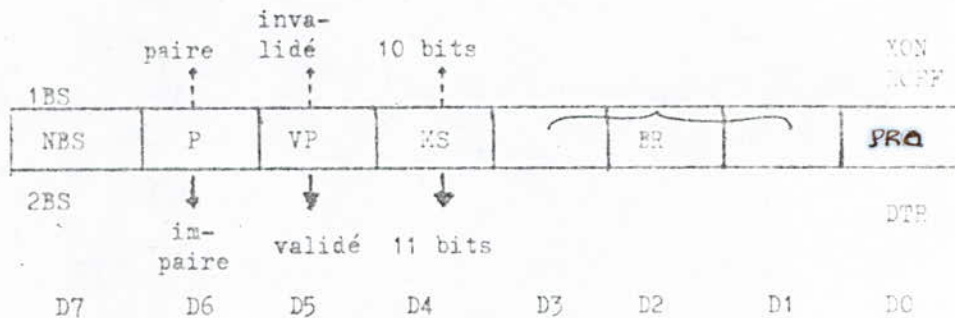
- Le registre B adressable bit par bit
- Le registre buffer sériel SBUF ~~et~~ éventuellement le bit TB8 du registre SCON représentant le 9ème bit à envoyer dans le mode 11 bits.
- Le bit Carry (CY) servant d'intermédiaire pour l'envoi d'un bit dans le bit TB8.

D/ PARAMETRES DE SORTIES :

- Aucun.

REMARQUE :

Signification des Switchs visu ou imprimante.



PRO = protocole (DTR ou XON - XOFF)

ER = Baud rate (vitesse de transmission)

MS = Mode du service port (10 ou 11 bits)

VP = Validation de la parité

P = Parité paire ou impaire

NBS = Nombre de bits stop

ER = 001 = 150B

010 = 300B

011 = 600B

100 = 1200B

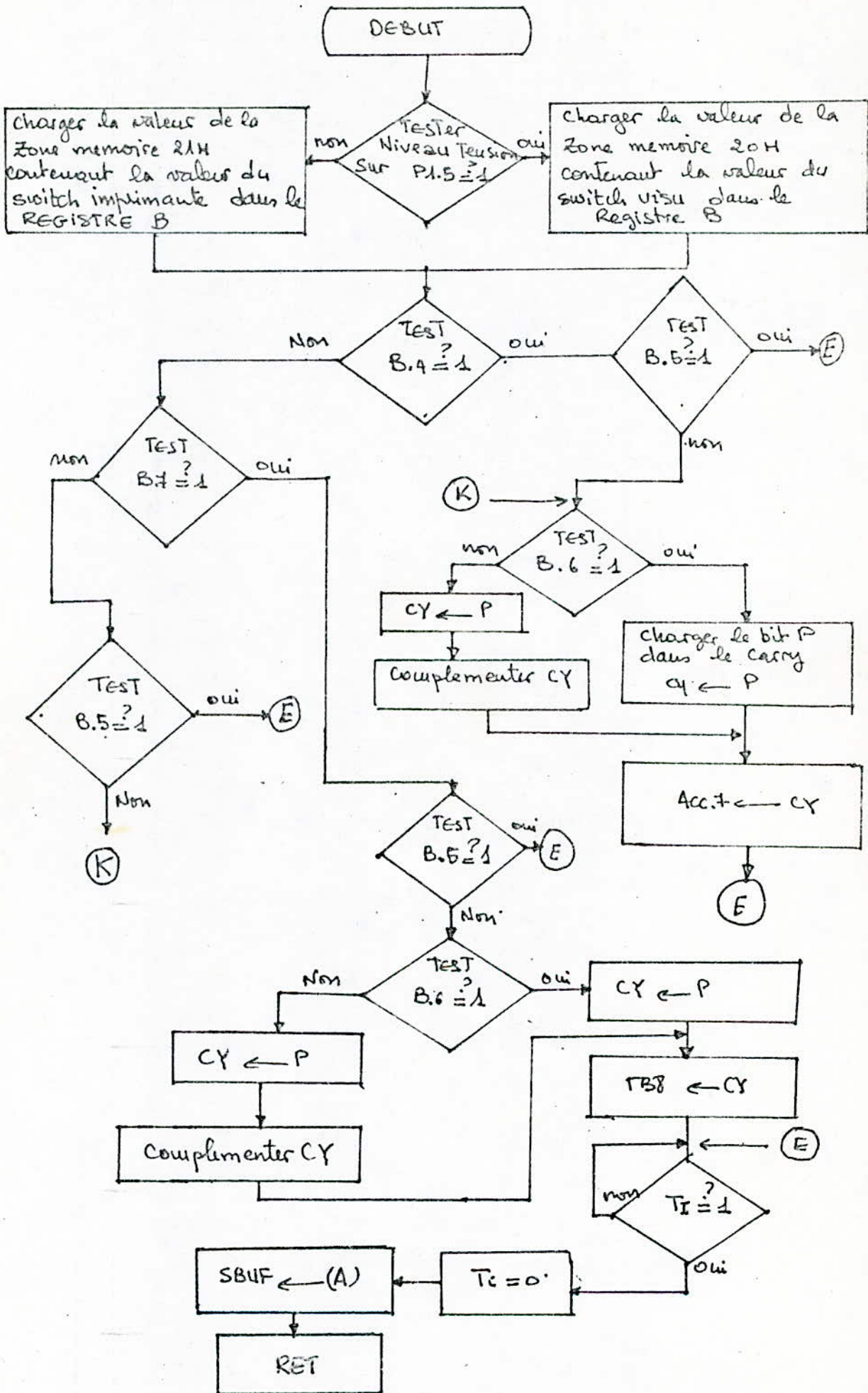
101 = 2400B

110 = 4800B

111 = 9600B

↑ = 1

↓ = 0



SOUS PROGRAMME DE CHOIX DU MODE
10 OU 11 BITS DU PORT SERIEL NOTE : INSPMD

'='='='='='='='eCe'='='='='=''

A/ FONCTION :

Au début du programme le port sériel est initialisé au mode 10 bits

Comme expliqué dans la partie réservée au Microcontrôleur, il existe 2 bits SMO et SM1 qu'il faut positionner à 01 pour le mode 10 bits ou à 11 pour le mode 11 bits, donc il suffit d'initialiser le mode à 10 bits puis suivant la valeur du bit SVFLAG ou SPFLAG des switches visus ou imprimante, on peut soit positionner SMO à 0 donc conserver le mode 10 bits soit positionner le bit SMO à 1 ce qui sélectionne le mode 11 bits. Cette routine sert justement à ça.

B/ PARAMETRES D'ENTREES :

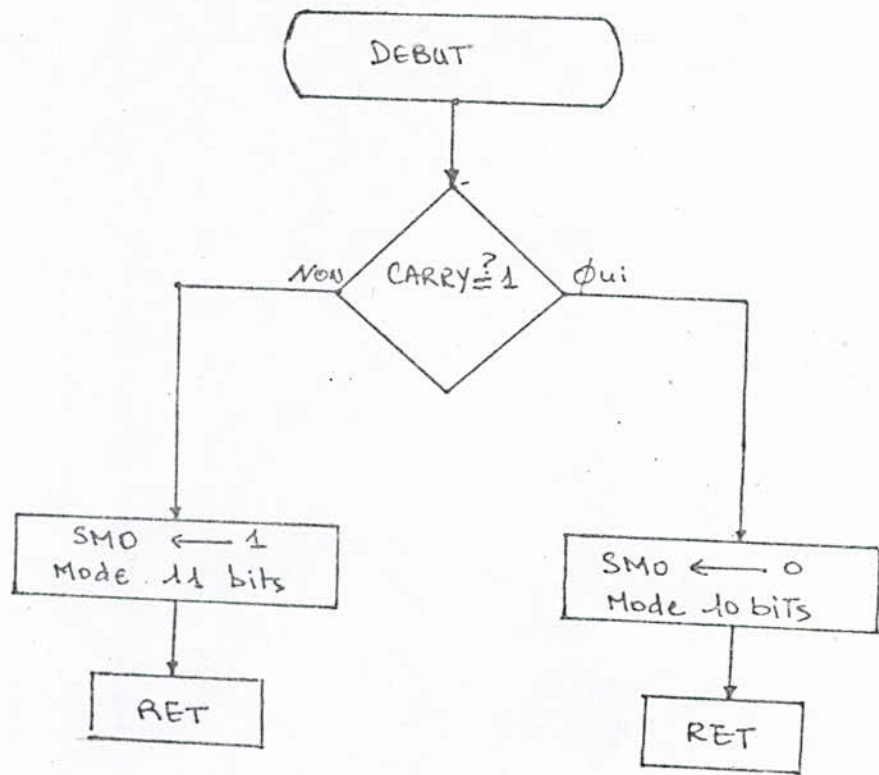
- Bit de sélection du mode du port sériel celui-ci est contenu dans une zone mémoire de la RAM interne adressable bit/bit, à charger dans le carry avant d'appeler la routine.

C/ ZONE DE TRAVAIL :

- Le carry CY
- Le bit SMO du registre de contrôle du port sériel : SCON

D/ PARAMETRES DE SORTIES :

- reinitialisation du mode du port sériel



SOUS PROGRAMME D'INVALIDATION DE
L'INTERRUPTION CORRESPONDANTE AU PROTOCOLE DTR
OU XON-XOFF NOTE CLRNB.

-----oOo-----

A/ FONCTION :

Cette routine a pour rôle d'invalider l'interruption correspondante au protocole (DTR ou XON-XOFF) qui est respectivement :

- soit l'interruption extérieure 0,EXO ,
- soit l'interruption en réception du port sériel ES.

B/ PARAMETRES D'ENTREES :

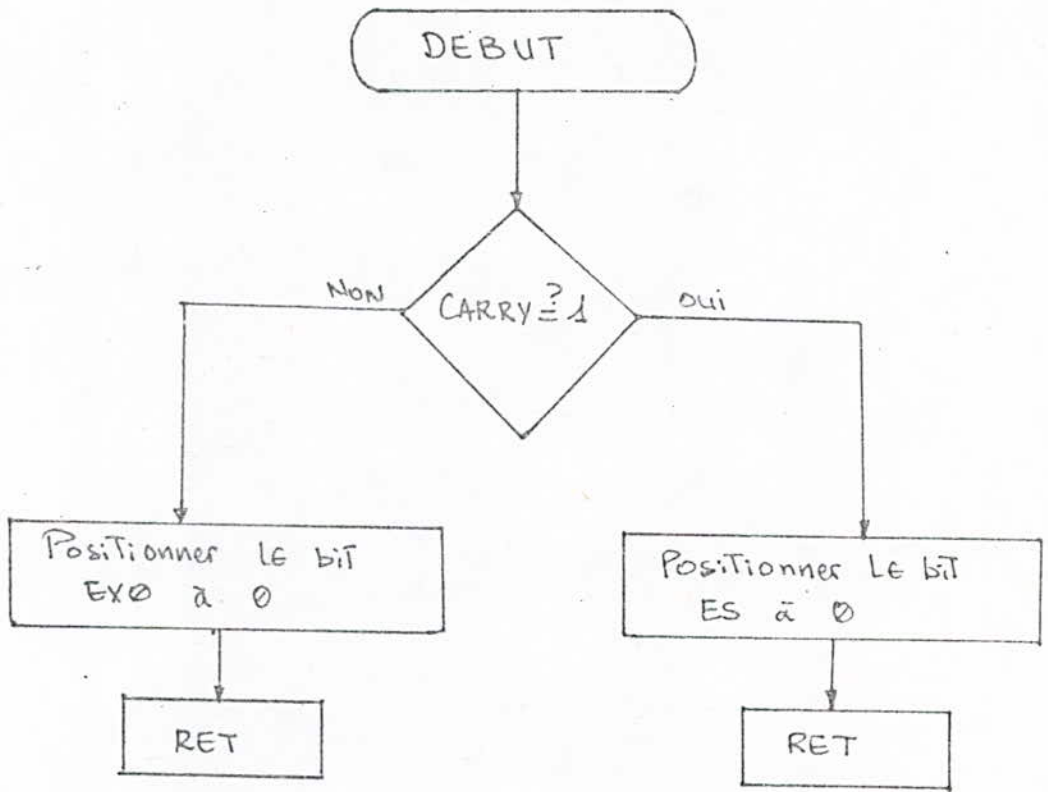
- Bit protocole placé dans une zone mémoire de la RAM interne adressable bit par bit, avant d'accéder à la routine, charger le bit protocole visu ou imprimante dans le carry CY.

C/ ZONE DE TRAVAIL :

- Le carry CY
- Le bit de validation d'interruption du port sériel
- Le bit de validation d'interruption extérieure zéro (EXO)

D/ PARAMETRES DE SORTIES :

- Invalidation de l'interruption correspondante au protocole visu ou imprimante.



SOUS PROGRAMME DE VALIDATION DE
L'INTERRUPTION CORRESPONDANTE AU PROTOCOLE DTR OU
XON-XOFF NOTE - SETINB

-----oOo-----

A/ FONCTION :

Cette routine a pour rôle de valider l'interruption correspondante au protocole (DTR ou XON-XOFF) qui est respectivement :

- . soit l'interruption extérieure O EXO ,
- . soit l'interruption en réception du port sériel ES.

B/ PARAMETRES D'ENTREES :

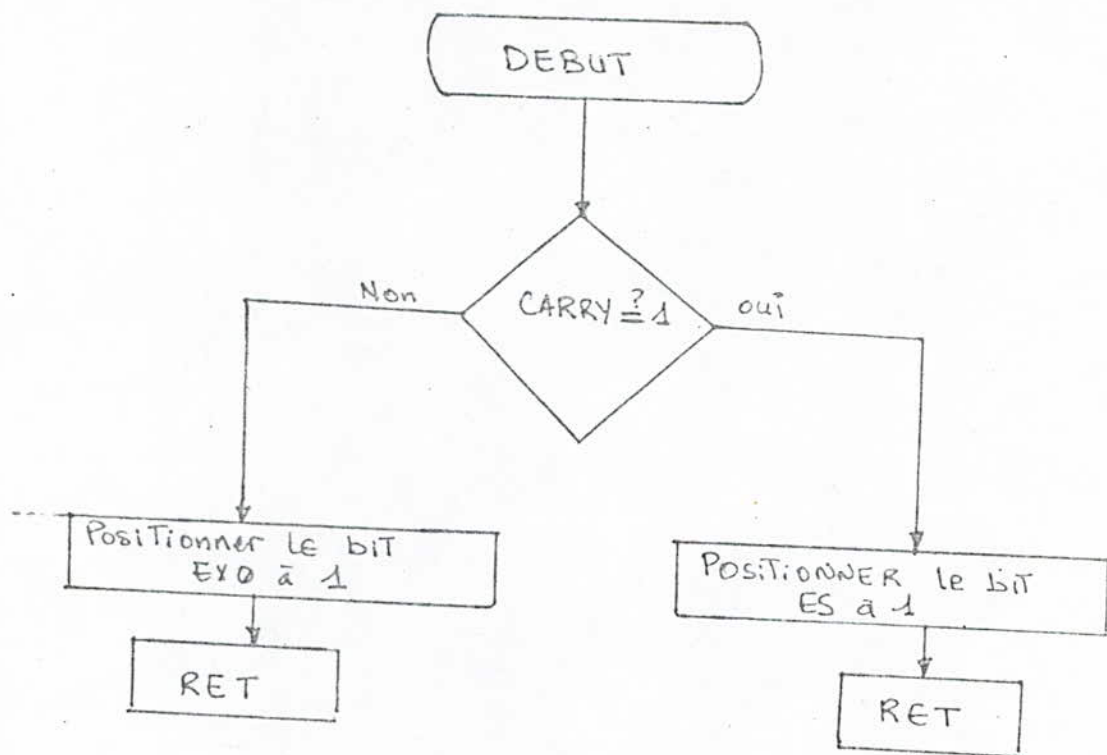
- Bit protocole placé dans une zone mémoire (20H ou 21H) adressable bit par bit avant d'accéder à la routine, charger le bit protocole visu ou imprimante dans le carry CY.

C/ ZONE DE TRAVAIL :

- Le carry CY
- Le bit de validation d'interruption du port sériel
- le bit de validation d'interruption extérieure zéro (0) EXO.

D/ PARAMETRES DE SORTIES :

- Validation de l'interruption correspondante au protocole visu ou imprimante.



CONCLUSION :

De par l'originalité du sujet, l'absence complète de documentation sur le sujet et la variété des combinaisons de défauts pouvant se produire théoriquement il nous a été difficile d'opter pour la façon matérielle d'exécution des tests et la façon logicielle de traiter les défauts trouvés en particulier les court-circuits. Les résultats obtenus avec le testeur réalisé sont très satisfaisants. Mais pour s'annoncer d'une façon définitive il faudra lancer le testeur dans la chaîne de tests de pistes de cartes circuit imprimé pendant un temps suffisant pour pouvoir avoir des probabilités sur la détection d'erreurs ; ceci par rapport au défauts qui arrivent pratiquement.

De plus pour utiliser le testeur d'une façon rationnelle il faudra que le département concerné par ce testeur conçoive un système mécanique fiable qui servira à poser (et à retirer) le gabarit sur la carte à tester et assurer un parfait contact des points de départ et d'arrivée du testeur avec ceux de la carte à tester y correspondants pour ne pas défausser les tests.

Le bénéfice que nous avons tiré de cette étude de projet de fin d'études est sans aucun doute très grand. En effet, cela nous a permis de travailler avec un système de développement bâti autour d'un PDP 11 d'un terminal et d'une carte Emulateur à base du 8031, de connaître un micro-contrôleur et son assembleur de se familiariser avec beaucoup de circuits TTL et compatibles TTL et enfin, c'est le plus important de mettre au point une méthode de test de piste à travers un hard et un soft et de réaliser une carte UC pour la gestion de ce testeur et son interfacage sériel avec une visu ou une imprimante.

Ce projet nous a aussi permis de nous confronter avec les problèmes pratiques de l'Electronique Hardware et Software.

Pour terminer, la question reste posée pour d'éventuels améliorations à apporter à cette méthode de tests à laquelle nous sommes arrivés.

 I B L I O G R A P H I E

-----c0o-----

- 1) THE T.T.L DATA BOOK
TEXAS INSTRUMENT - 1976 -
- 2) MEMORY COMPONENTS HAND BOOK (1983)
INTEL CORPORATION
- 3) R.L. TOKHEIM TECHNIQUES NUMERIQUES
Edition Mc GRAW-HILL
- 4) M C S - 51 USER'S MANUAL
INTEL - 1981 -
- 5) LE HAUT PARLEUR N° 1706 - Juillet 1984.
- 6) MICRO CONTROLER HAND BOOK
INTEL
- 7) THE INTERFACE CIRCUITS DATA BOOK
TEXAX INSTRUMENTS -1977 -