

وزارة التعليم و البحث العلمي  
MINISTÈRE DE L'ENSEIGNEMENT ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT D'ELECTRONIQUE

ECOLE NATIONALE POLYTECHNIQUE  
BIBLIOTHEQUE

## PROJET DE FIN D'ETUDES

### SUJET

ETUDE ET MISE EN  
OEUVRE DE  
L'ANALYSEUR LOGIQUE  
7D02 TEKTRONIX

Proposé par :  
MM: M. MEHENNI  
et: R. SAADOUN

Etudié par :  
MM: Rachid OUSSAID  
et: Mohamed HADBOUNE

Dirigé par :  
MM: M. MEHENNI  
et: R. SAADOUN

DEDICACES

- A mes parents.
- A toute ma famille.
- A tous ceux qui me sont chers

Rachid .

- A mes parents.
- A ma famille.
- A tous mes amis.

M.HADBOUNE

7-EMERCIMENTS

Nous tenons à remercier nos promoteurs MM:MEHENNI et SAADOUN pour l'aide et l'effort qu'ils nous ont fournis tout au long de ce projet.Et que tous ceux qui nous ont porté aide et soutien trouvent ici l'expression de nos sincères remerciements.

# TABLE DES MATIERES

INTRODUCTION .....	1
CHAPITRE I :GENERALITES	
I-INTRODUCTION.....	3
II-OUTILS DE DEVELOPPEMENT,DE MISE AU POINT ET DE TEST DES SYSTEMES LOGIQUES.	3
II-1 L'EMULATEUR.....	3
II-2 L'ANALYSEUR DE SIGNATURE.....	4
II-3 L'ANALYSEUR LOGIQUE.....	4
II-3-1 CONTEXTE D'UTILISATION.....	4
II-3-2 UTILISATION ET MODES DE FONCTIONNEMENT .....	5
II-3-2-1 GENERALITES .....	5
II-3-2-2 MODES DE DECLENCHEMENT ET DE MEMORISATION .....	6
a-DECLENCHEMENT .....	6
b-MEMORISATION .....	6
II-3-2-3 MODES DE VISUALISATION DES INFORMATIONS .....	7
CHAITRE II: PRESENTATION ET ETUDE DU 7DO2	
I-PRESENTATION .....	10
I-1 INTRODUCTION .....	10
I-2 VUE GENERALE DU FONCTIONNEMENT .....	12
II-ETUDE DESCRIPTIVE DU 7DO2 .....	14
CHAPITRE III:MISE EN OEUVRE	
I-CONCEPT DE PROGRAMMATION DU 7DO2.....	24
II-INTRODUCTION.....	24
II-SYNTAXE .....	24
III-TESTS	
II-L'OPTION TIMING .....	36
I-INTRODUCTION .....	36
II-CONCEPT DE PROGRAMMATION .....	39
III-DEFINITION DES DONNEES ET PARASITES	42
CHAPITRE IV:APPLICATIONS	
I-INTRODUCTION .....	48
II- APPLICATIONS SUR LE KIT D5 .....	48
III-APPLICATIONS SUR LA CARTE TM 990/189 DE TEXAS INSTRUMENT .....	69

## INTRODUCTION

L'avènement des systèmes numériques a posé des difficultés lors des développement et test de ces derniers.

Tous ceux qui sont amenés à les manipuler ressentent le besoin de disposer d'un outil d'investigation visuel, permettant la mise au point et le test de ces systèmes.

L'analyseur logique 7D02 est conçu spécialement pour répondre à ces objectifs.

Le but de notre projet est de donner sa procédure d'utilisation et d'étudier son fonctionnement interne.

Dans le premier chapitre, on expose des généralités sur les appareils utilisés pour le développement et la mise au point des systèmes numériques. Le second chapitre présente le 7D02, ainsi qu'une description de son fonctionnement interne. Son concept de programmation est exposé dans le chapitre III. Le dernier chapitre est réservé aux applications développées.

CHAPITRE I

G E N E R A L I T E S

## I-INTRODUCTION

L'évolution des systèmes électroniques a donné naissance à des systèmes logiques très complexes. Avant l'apparition d'équipements spécialisés, le développement, la mise au point et le dépannage de ces systèmes se sont avérés très laborieux, par conséquent des instruments capables de palier à cette difficulté sont devenus nécessaires. Parmi ces instruments, nous citerons: Les analyseurs logiques, les émulateurs et les analyseurs de signature.

Nous allons présenter sommairement les principes de chaque appareil, nous insisterons, en particulier, sur l'analyseur logique qui est l'objet de notre étude.

## II-OUTILS DE DEVELOPPEMENT, DE MISE AU POINT, ET DE TEST DES SYSTEMES LOGIQUES.

Ils sont au nombre de trois:

- L'émulateur.
- L'analyseur de signature.
- L'analyseur logique.

### II-1 L'émulateur

C'est un outil de développement et de mise au point des systèmes à base de microprocesseurs. La conception de tels systèmes nécessite une réalisation matérielle et l'élaboration d'un logiciel gérant le fonctionnement de ceux-ci.

Lors du développement d'un système, l'émulateur permet de suivre les phases suivantes:

#### -Emulation "0"

Elle consiste en un développement de logiciel pour une application donnée. Dans ce mode le programme est développé dans la mémoire programme de l'émulateur, sans interaction avec le prototype.

#### -Emulation "1"

C'est la phase d'exécution du programme dans le prototype (sous contrôle de l'émulateur) avec possibilité de modification, de correction dans le cas d'un mauvais fonctionnement.

## Emulation "2"

ce mode consiste en l'intégration du logiciel dans le prototype et l'analyse de son fonctionnement.

### Remarque

Dans les modes "1" et "2", l'horloge est fournie par le système en développement, contrairement au mode "0"

## II-2 L'analyseur de signature

C'est un équipement de test des systèmes programmables. La signature d'un noeud est définie comme étant la valeur numérique affichée après décodage d'un mot de 16 bits prélevé au niveau du noeud et dépend des conditions particulières suivantes:

- Du programme bouclé choisi
- de la transition de départ choisie
- de la transition d'arrêt choisie
- du signal d'horloge choisi

A partir de ce principe, le dépannage d'un système consiste en un prélèvement de signatures de noeuds

Lorsqu'un système tombe en panne, en se plaçant dans les mêmes conditions, on relève les signatures des différents noeuds qui, après avoir été comparées aux signatures exactes, on détermine la vraisemblance de celles-ci.

## II-3 L'analyseur logique

C'est un outil d'investigation visuel capable d'assurer le test fonctionnel et organique des circuits intégrés complexes et des divers systèmes logiques.

Il permet par exemple, la vérification d'une séquence d'un compteur ou l'essai d'un programme de microprocesseur.

### II-3-1 Contexte d'utilisation

L'utilisation des analyseurs logiques est plus orientée vers les systèmes à base de microprocesseurs que ceux à logiques câblées.

L'opérateur doit prévoir la succession des états logiques d'un ou plusieurs points particuliers de son système et leurs incidences sur d'autres parties de ce système.



Une simple comparaison des signaux acquis et ceux prévus donne la possibilité de détecter une défaillance et localiser sa source.

En ce qui concerne les systèmes à microprocesseurs, les analyseurs logiques peuvent être adaptés à n'importe quel microprocesseur 8 ou 16 bits, à condition que le micro soit en bon état de fonctionnement. Pour remédier à cette contrainte, on peut lui associer un émulateur qui remplacera le microprocesseur du système.

## II-3-2 UTILISATION ET MODES DE FONCTIONNEMENT

### II-3-2-1 Généralités

Dans un système logique, on s'intéresse à prélever plusieurs signaux à la fois, soit sous forme de tables d'états soit sous forme de chronogrammes. Les informations prélevées nous informent sur l'état du système. L'acquisition de celles-ci est conditionnée par un ou plusieurs événements particuliers, ce qui donne la possibilité d'avoir des informations bien définies.

Les analyseurs logiques permettent la mémorisation des états logiques disponibles sur les entrées et l'affichage de ces données sur l'écran.

Ils doivent donc pouvoir acquérir des données, les enregistrer puis les restituer sous une forme exploitable.

L'enregistrement est en principe fonction d'un déclenchement, sur une combinaison binaire ou sur un mot présélectionné, qui indiquera à quel moment les données doivent être collectées.

Le début de mémorisation et l'arrêt peuvent être retardés d'un nombre programmable de tops d'horloge.

Les données ainsi mémorisées peuvent être restituées à des moments différents. Il peut être, en effet nécessaire de faire apparaître des événements qui ont précédé le déclenchement, qui l'ont suivi ou qui l'entourent, ce qui facilite la recherche des causes et des conséquences d'un défaut quelconque.

Quant à l'affichage, il n'est pas toujours possible de visualiser sur un écran l'ensemble des états mémorisés, ce qui oblige à définir des fenêtres de visualisation qu'il est loisible de déplacer à l'intérieur du contenu de la mémoire.

Les analyseurs logiques étant pilotés par une horloge de synchronisation, ils permettent généralement d'introduire des conditions sur cette horloge. Cela signifie que lors d'une séquence de prise de données, une impulsion de l'horloge de synchronisation ne déclenchera la mémorisation que si certaines conditions sont réalisées. Les signaux représentant ces conditions sont appelés QUALIFICATEURS.

L'analyseur peut fonctionner dans deux modes:

-Mode synchrone: l'horloge est fournie par le système sous test.

-Mode asynchrone: l'horloge est fournie par l'analyseur lui-même. Ceci pourrait être nécessaire pour une observation plus fine des signaux.

## II-3-2-2 Modes de déclenchements et mémorisation

### a déclenchement

Lors d'un test, le déclenchement peut être provoqué par un ou une combinaison d'événements suivants:

-données

-adresse

-un ou plusieurs signaux de contrôle y compris le signal de déclenchement externe

-parasites

### Remarque

Il est à signaler qu'un déclenchement peut s'opérer sur une transition ou un état logique.

### b-mémorisation

Les analyseurs logiques offrent de nombreuses possibilités pour la prise de données. Al'instar des oscilloscopes, ils permettent de visualiser des états apparaissant juste après l'événement choisi pour prélever les informations, mais contrairement aux oscilloscopes, ils peuvent aussi visualiser des états apparaissant avant cet événement.

1-On veut connaître les états qui ont précédé l'apparition d'un signal:

L'analyseur logique enregistre en permanence les états logiques et les informations sont rentrées au fur et à mesure dans la

dans la mémoire. Lorsque celle-ci est totalement remplie, la prochaine information rentre et chasse la première information du cycle passé. A l'apparition du signal de déclenchement, la mémorisation est stoppée et l'on pourra alors observer les états qui ont précédé le signal.

2-on veut connaître les états qui ont suivi l'apparition du signal de déclenchement:

A l'apparition du signal de déclenchement, la mémoire jusqu'alors vège se remplit au maximum de sa capacité et s'arrête ensuite.

3-On veut connaître les états qui ont précédé et ceux qui ont suivi l'apparition du signal de déclenchement:

Seule une partie de la mémoire est utilisée dans un premier temps. Cette fraction de mémoire fonctionne en permanence comme dans le premier cas. A l'apparition du signal de déclenchement, elle est bloquée dans sa dernière position, tandis que l'autre portion mémoire, jusqu'ici dépourvue de toute information, se remplit jusqu'à sa capacité maximale.

4-On désire mémoriser les événements seulement après un certain temps:

Ce retard est le plus souvent obtenu en utilisant un compteur à présélection sur lequel on affiche le nombre de tops d'horloge représentant le retard désiré. A l'apparition du signal de déclenchement, le compteur commence à décompter et une fois arrivé à zéro, il commande la mémorisation ou au contraire la stoppe.

### II-3-2-3 Modes de visualisation des informations.

Les informations mémorisées peuvent être visualisées sur l'écran selon des possibilités appelées modes.

1-Mode états:

Dans ce mode, les contenus des différentes voies apparaissent à raison d'une ligne de visualisation par pas, en notation hexadécimale, octale ou binaire.

2-Mode signaux:

Dans ce mode, la forme des signaux de chaque voie d'entrée apparaît sur une ligne. L'analyseur logique se comporte alors comme un oscilloscope multicanaux. Cette représentation temporelle des signaux permet de détecter des erreurs de matériel.

3-MODE cartographique;

Ce mode permet de représenter de façon globale le déroulement d'une séquence ou d'un programme en visualisant chaque mot sous forme d'un point unique sur l'écran dont les coordonnées ne correspondent qu'à ce mot. Pour une adresse de 16 bits, les 8 bits de poids faible sont en abscisse et les 8 bits de poids fort en ordonnée. Dans ce mode un programme se traduit par un graphe constitué de nombreux segments de droite résultant du passage d'une adresse à une autre.

4-Mode "GLITCH" (détection de parasites);

Le terme anglo-saxon "glitch" désigne un signal parasite qui apparaît et disparaît entre deux impulsions adjacentes de l'horloge de synchronisation, ou un front qui n'est pas net.

## CHAPITRE II

PRESENTATION ET ETUDE DE  
L'ANALYSEUR LOGIQUE 7D02

## I/PRESENTATION

### I-1 INTRODUCTION

Proposé par TECTRONIX, l'analyseur logique 7D02 est un outil de mise au point, de dépannage et d'aide pour l'utilisation dans le développement des systèmes digitaux, spécialement pour les systèmes à base de microprocesseurs.

C'est un appareil programmable et interactif. Il permet de guider l'utilisateur à l'aide des menus en indiquant des erreurs de manipulation.

Il offre une mémoire d'acquisition de 256 positions définies sur 28 bits (sans option), quatre reconnaisseurs de mot, deux compteurs d'usage général, une horloge, un circuit de qualification de données, un organe qui constitue le noyau du système appelé "STATE MACHINE" (voir étude) et trois mémoires.

Il est adaptable à une large variété de microprocesseurs par l'intermédiaire de modules de personnalisation qui seront décrits ultérieurement. (Voir annexe)

L'acquisition de données à partir du système sous test se fait sur 28 voies organisées comme suit:

- Un bus de 16 lignes d'adresses
- Un bus de 8 lignes de données
- Quatre lignes de contrôle

Les reconnaisseurs de mot accomplissent leur fonction sur 28 voies plus 3 voies supplémentaires qui sont:

- Une <sup>ligne</sup> de déclenchement extérieur
- Deux lignes de contrôle supplémentaires.

L'utilisateur peut donc définir son modèle sur 31 voies.

Les reconnaisseurs de mot peuvent être programmés soit pour reconnaître la présence du modèle soit pour reconnaître son absence.

Les compteurs peuvent aller jusqu'à 65534 événements, **comme ils peuvent** compter jusqu'à 65534 microsecondes ou millisecondes.

L'exécution des commandes comme celles de déclenchement (TRIGGER) ou de qualification (QUALIFY) peuvent être conditionnées par des compteurs dont on spécifie la valeur qui doit être atteinte.

L'analyseur de base contient trois mémoires à savoir:

-Une mémoire de programme:qui contient le programme courant introduit par l'utilisateur.

-Une mémoire d'acquisition(ou mémoire principale d'acquisition): Celle-ci contient les données acquises après le déroulement du programme.

-Une mémoire de stockage; qui peut être utilisée pour sauvegarder le programme ou le contenu de la mémoire d' acquisition.

Pour élargir le champ d'utilisation du 7D02,on a la possibilité de recourir à deux options supplémentaires à savoir:

-L'option d' extension.

-L'option timing.

-L'option d'extention(Expansion option)rend le 7D02 capable de supporter les microprocesseurs 16 bits.

Cette option augmente la largeur de la mémoire d'acquisition et de la reconnaissance de mot de 16 voies,ce qui fait 44voies partagées comme suit:

-UN bus d'adresse de 24 lignes

-Un bus de données de 16 lignes

-Quatre lignes de controle.

#### REMARQUE

Comme dans l'analyseur de base deux lignes de contrôle supplémentaires et la ligne de déclenchement extérieur sont aussi disponibles pour la reconnaissance de mot.

-L'option timing transforme le 7D02 en deux analyseurs logiques. Cette option a son propre déclencheur, sa propre mémoire d'acquisition et son propre reconnaisseur de mot.

En utilisation synchrone,l'option timing fonctionne comme une extension de 8 voies de la section principale.

En utilisation asynchrone(c-à-d l'analyseur fonctionne avec sa propre base de temps),il reconnaît et mémorise des données et des parasites éventuels sur 8 voies.

Un parasite est reconnu lorsque plus d'une transition se présente entre deux impulsions d'échantillonnage successives(voir "option TIMING").

L'option timing peut être utilisée pour déclencher la section principale et vice-versa.

Elle autorise sur 8 voies:

- la représentation temporelle.
- la représentation en table d'états (HEXA, BINAIRE, OCTAL, ASCII)
- le déclenchement sur les parasites.
- la capture ou la détection de parasites du premier et du second ordre (voir page 43)

Une mémoire spécifique de 255 mots de 8 bits est réservée à cet effet.

L'option timing permet le test aisé de l'environnement des microprocesseurs (entrées/sorties, interruption, etc...) que la section principale ne permet de réaliser.

#### I-2 VUE GENERALE DU FONCTIONNEMENT:

A l'aide du clavier, l'utilisateur introduit un programme contenant les conditions de déclenchement sur un ou une séquence d'événements supposés apparaître sur le système sous test "SUT" (system under test).

Lors de la présence de l'événement, le matériel d'acquisition hardware (compteurs, reconnaisseurs de mot, "STATE MACHINE", qualificateurs d'horloge et de données, etc...) est activé et l'échantillonnage de données commence.

L'échantillonnage se fait au rythme et sur un front de l'horloge propre au "SUT"

Un circuit de qualification d'horloge sélectionne les fronts sur lesquels la donnée serait validée. Cette horloge validée du "SUT" est utilisée comme horloge d'échantillonnage du 7D02 appelée "State clock" (Voir Etude).

Sur chaque cycle, la donnée est échantillonnée, les conditions définies par le programme sont testées et les commandes associées sont exécutées.

Bien que la donnée soit échantillonnée sur chaque cycle, elle n'est pas nécessairement mémorisée. En effet le circuit de qualification de données utilise le critère défini par le programme;



pour déterminer si chaque mot de données échantillonnées doit être mémorisée ou non; c'est la qualification de données.

L'exécution de la commande de déclenchement a pour effet d'activer un compteur à retard associé à la mémoire d'acquisition. Une fois activé, le compteur s'incrémente chaque fois qu'un mot de données est mémorisé.

Lorsque le compteur aura atteint la valeur spécifiée par programme, le 7D02 stoppe l'acquisition ainsi que l'exécution du programme. A cet instant, les données mémorisées sont affichées. L'utilisateur sera libre de transférer le programme ou les données acquises de ou vers la mémoire de stockage, d'examiner ces données dans différents formats, d'introduire un nouveau programme, de redérouler le programme, etc...

## II-ETUDE DESCRIPTIVE DU 7DO2.

### -INTRODUCTION

Après avoir décrit les principales fonctions du 7DO2, nous allons présenter dans ce qui suit sa structure et son fonctionnement interne.

Le 7DO2 est basé sur une structure modulaire de 11 cartes fonctionnelles. Le schéma synoptique (Fig 1) illustre les différents blocs.

Les liaisons internes sont assurées par différents bus qui peuvent être communs ou propre à chaque bloc.

Ce schéma peut être divisé en deux parties correspondant respectivement aux sections principale et "timing".

- La section principale est composée des blocs suivants:

- 1-Le clavier (KEYBOARD)
- 2-Le "FRONT PANEL"
- 3-Le reconnaiseur de mot (WORD RECOGNIZER)
- 4-Le "FRONT END"
- 5-La mémoire d'acquisition (Acquisition memory)
- 6-"STATE MACHINE"
- 7-Le CPU
- 8-L'affichage (Display)

L'interfaçage entre cette section et le "SUT" est assuré par un module personnalisé.

-La section "TIMING" englobe les blocs suivants:

- 1-Trigger et base de temps
- 2-"IC acquisition" (c'est la carte d'acquisition et de reconnaissance)

La sonde P6451 permet l'interfaçage entre cette section et le "SUT".

L'alimentation de l'ensemble est assurée par le bloc d'alimentation indiqué sur le synoptique.

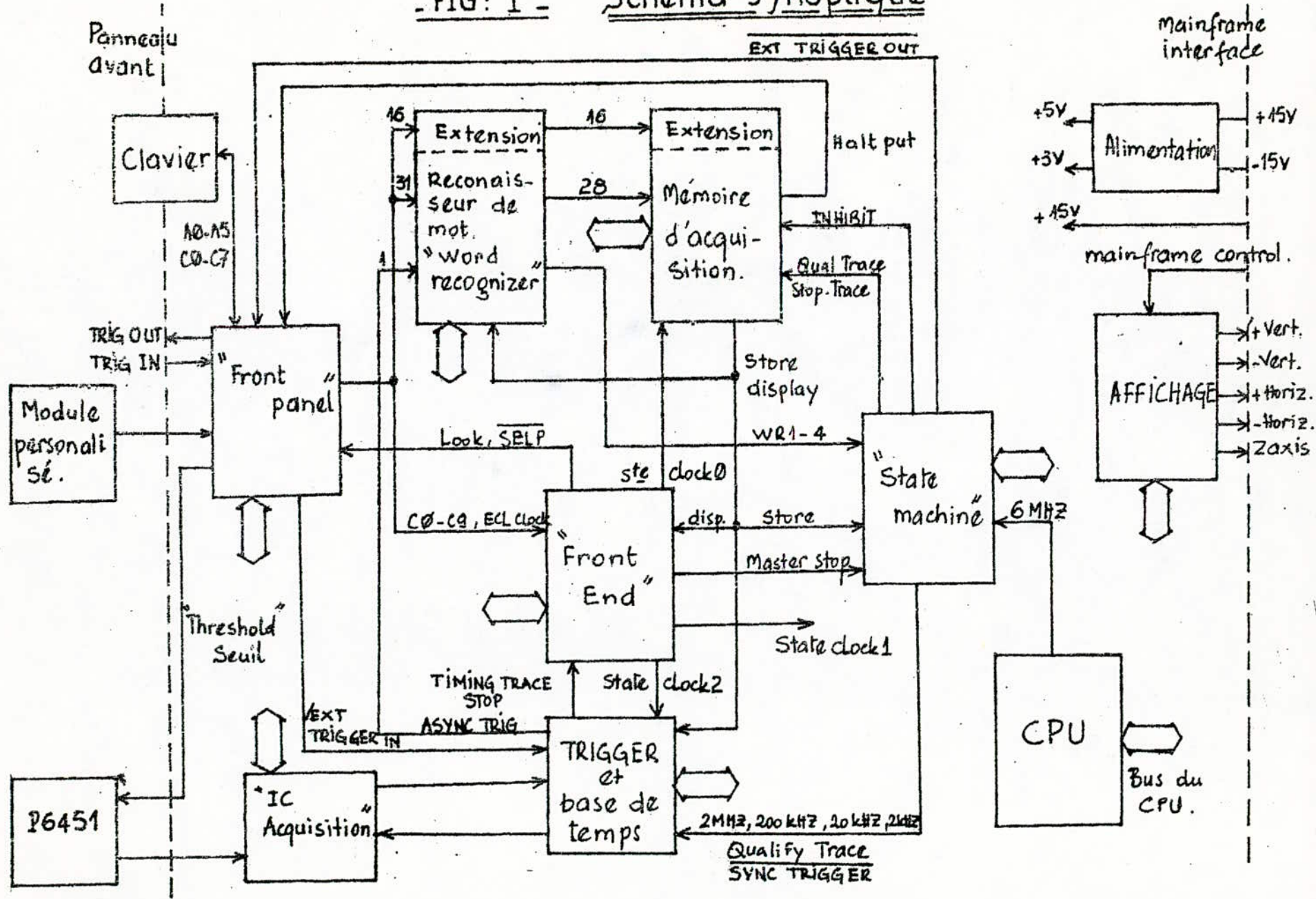
### -Description des fonctions principales de chaque bloc

La section principale:

#### 1-Le clavier

C'est l'organe d'entrée à partir duquel l'opérateur

- FIG: 1 -      Schéma synoptique



communiqué avec le 7D02, il est organisé en matrice (6 rangées x 8 colonnes).

Le CPU utilise les six lignes d'adresse de poids faible (A0-A5) pour scruter le clavier. Celles-ci constituent les lignes de la matrice. Les 8 colonnes (C0-C7) sont reliées au bus de données (D0-D7) à travers le "FRONT PANEL".

Le principe consiste en l'envoi d'une adresse spécifique à une ligne, après la lecture du bus de données, le CPU détermine la touche appuyée.

## 2-Le "FRONT PANEL"

Outre l'amplification de bus (C0-C7, A0-A5) et l'inversion du signal SYNC TRIG, apparaissant sur "TRIG OUT", il assure:

-La conversion digitale analogique de la donnée issue du CPU, permettant ainsi le choix du seuil des niveaux logiques des signaux appliqués à la sonde P6451.

-La comparaison du niveau de tension du signal de déclenchement externe "TRIG IN" avec le seuil de tension choisi par l'opérateur, et émet les signaux "SE TRIG" et "EXT TRIG" qui sont envoyés respectivement au bloc "reconnaisseur de mot", mémoire d'acquisition et trigger.

## 3-Reconnaisseur de mot:

Sa fonction principale est la détection d'un ou plusieurs événements, ceci se réalise par l'apport de quatre mémoires contenant chacune l'événement imposé par l'opérateur et propre à chaque test. (Il existe quatre reconnaisseurs de mot au maximum).

Un signal actif à l'état bas WRX (x=1,2,3,4) sera émis pour indiquer une comparaison vraie. Ce signal est envoyé au bloc "STATE MACHINE", qui exécute une action appropriée décrite par le programme à savoir:

- Changement de test
- Démarrage ou rechargement d'un compteur
- déclenchement du processus d'acquisition
- Qualification de données
- Etc...

#### 4-Le "FRONT-END"

Les principales fonctions de ce bloc sont:

a-La transformation ECL-TTL du signal d'horloge issu du module personnalisé (RAW-CLOCK).

b-La génération de différents signaux dérivés du signal "RAW CLOCK" qui sont:

- "STATE CLOCK  $\emptyset$ "

- "STATE CLOCK 1"

- "STATE CLOCK 2"

"STATE CLOCK  $\emptyset$ ": C'est un signal inverse de "RAW-CLOCK, c-à-d (RAW-CLOCK). Il est appliqué au bloc "mémoire d'acquisition", qui l'utilise comme entrée d'horloge du contrôleur des modes; affichage (display) ou mémorisation (store). A cet effet deux signaux "DISPLAY" et "STORE" sont générés (voir paragraphe 5).

"STATE CLOCK 1": Issu d'un des deux générateurs d'impulsion disponibles sur la carte, il joue le rôle d'horloge pour les blocs suivants: "State machine", reconnaisseur de mot; option d'extension" et mémoire d'acquisition". CE signal est actif en mode mémorisation et inactif en mode d'affichage.

"State clock 2": Issu du deuxième générateur d'impulsion, il est appliqué aux blocs de l'option timing ainsi qu'aux circuits du "FRONT end". Ce signal est actif continuellement.

#### 5-MEMOIRE D'ACQUISITION:

Ce bloc réalise les fonctions principales suivantes:

a-La mémorisation de 28 voies de données.

b-La production du retard requis pour la génération de déclencheurs et de qualificateurs de données.

c-La génération de la commande d'arrêt du "SUT" une fois que la mémoire est remplie.

d-La génération des signaux de contrôle "STORE" et "DISPLAY" qui permettent aux différents blocs du système d'être en mode d'affichage ou stockage.

e-Le positionnement du mot déclencheur (événement) par rapport aux données mémorisées.

6-"STATE MACHINE": FIG 2

Elle est constituée principalement de:

- Sequenseur
- Bloc compteur
- Générateur de fréquence.

Sa fonction principale est de générer toutes les commandes associées à un événement donné.

a-Sequenseur:

C'est la partie maitresse de ce bloc. Elle est constituée:

-D'une RAM(256x9bits), qui reçoit initialement toutes les instructions relatives à un ou plusieurs événements. Celles-ci proviennent, sous un certain format approprié, de la carte CPU.

L'exécution d'une instruction consiste à mettre sur le bus de contrôle de cette carte, les commandes associées à chaque fonction (déclenchement, qualification, mode de comptage, etc...), - -

L'adressage des instructions se faisait à l'aide des lignes:

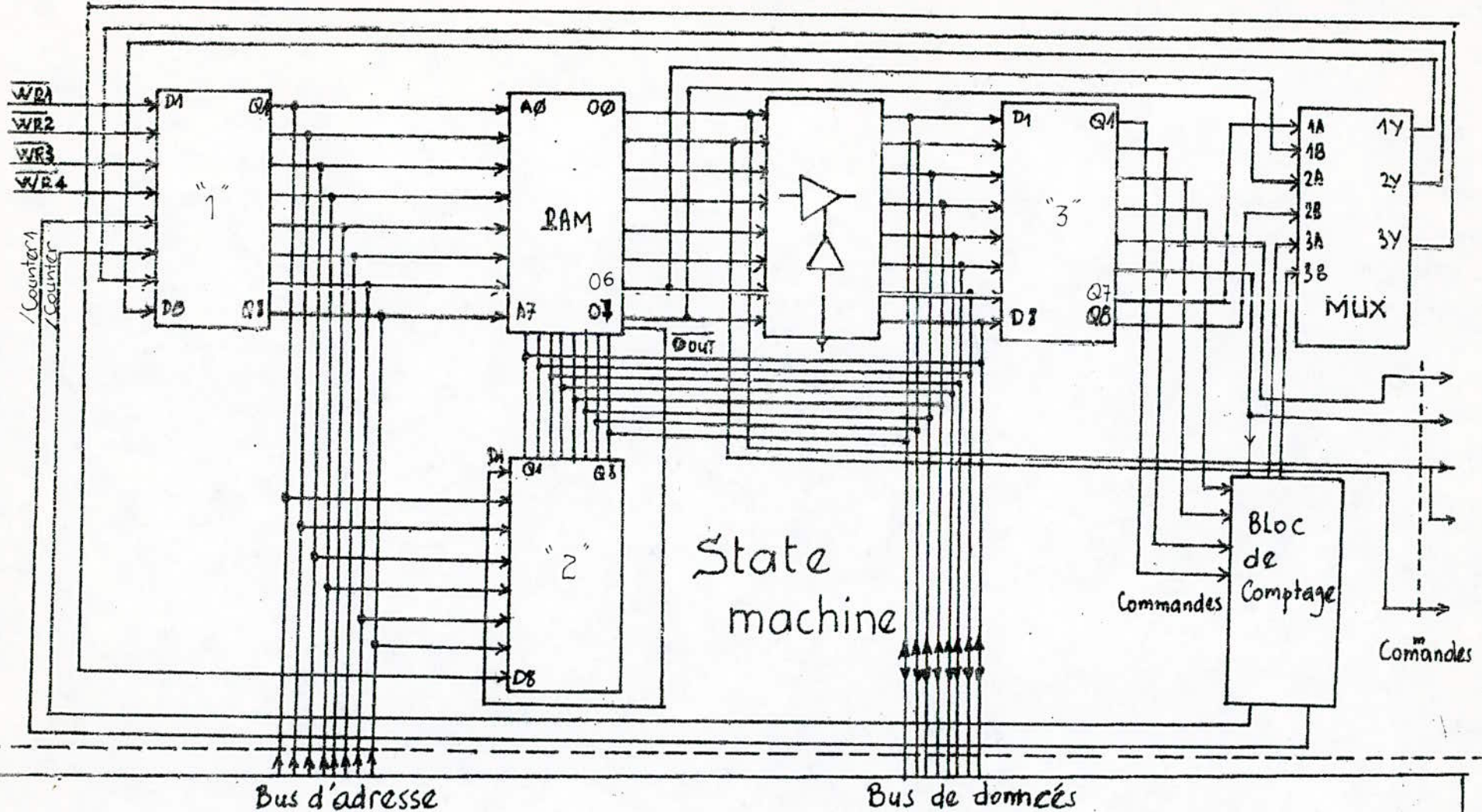
- WR1-WR4 : Issues du bloc reconnaisseur de mot.
- COUNTER1 et COUNTER2: Issues du bloc de comptage
- (Q7 et Q8) ou (Q6 et Q8): issues du multiplexeur et qui sont utilisées comme un Feedback.

-D'un latch( $\bar{Z}$ ): Celui-ci envoie au CPU des informations sur l'état de "State machine".

b-Le bloc compteur : Il est constitué de:

Deux compteurs programmables(COUNTER 1 et 2) utilisés pour le comptage d'événements ou de temps. Ces deux compteurs sont chargés par le CPU. Comme les reconnaisseurs de mot, ces deux compteurs détectent la présence de l'événement (valeur programmée atteinte) Par exemple: IF COUNTER = 100 ev<sup>t</sup>.

Lors d'une reconnaissance, ils envoient les signaux COUNTER 1 (COUNTER 2), comme WRX, qui déterminent la commande appropriée, à prendre de la RAM.



Carte : CPU

- FIG:2 -

C-Générateur de fréquence :

Quatre fréquences(2MHZ,200KHZ,20KHZ et 2KHZ)sont générées pour

-Déterminer la période de comptage à l'aide des fréquences:

2MHZ---->1 $\mu$ s et 2KHZ---->1ms.

-Pour être envoyées au bloc Trigger et base de temps de l'option timing,utiles à la sélection de la fréquence d'échantillonnage (1 $\mu$ s,10 $\mu$ s,100 $\mu$ s et 1ms).

7-La carte CPU:

Elle s'articule essentiellement sur le microprocesseur 8085 de INTEL.Elle assure le contrôle du 7D02 et contient essentiellement :

-Un moniteur s'étendant sur 48 k octets,qui assure la gestion du système.

-Une mémoire RAM de 8 k octets utilisée pour le stockage du programme utilisateur.

- Un circuit d'horloge délivrant un signal de 6MHZ utilisé simultanément par "STATE MACHINE" et le microprocesseur. Ce dernier génère un signal de 3Mhz utile au fonctionnement de cette carte et aux blocs"DISPLAY" et "FRONT-panel".

8-Les circuits d'affichage:

Ces circuits contiennent essentiellement une RAM qui reçoit une copie codée de ce qui doit être affiché.Un caractère est codé sur 8 bits,7 bits représentant le code ASCII du caractère, le bit le plus significatif(MSB)sélectionne le mode de représentation(video inversé ou normal)

Les caractères sont interprétés par une ROM(appelée ROM de caractères).A l'aide de circuits associés,elle génère un signal vidéo représentant la donnée à visualiser.

Le balayage consiste en un arrangement de points 256 P<sup>ts</sup>/ligne et 216 lignes(de points).Un caractère est représenté à l'aide d'une matrice de points,9 lignes x 8 colonnes,ce qui donne 24 rangées de 32 caractères(24 x 9 =216 lignes,32 x 8 =256 p<sup>ts</sup>/ligne



Le balayage du spot est assuré par deux générateurs, l'un délivrant une rampe pour le balayage horizontal, l'autre un signal en escalier pour le balayage vertical.

### LA SECTION "TIMING"

Cette section peut opérer seule ou en liaison avec la section principale.

Les données à l'entrée, sur 8 voies, sont appliquées au bloc "IC ACQUISITION" à travers la sonde P6451. Ce bloc contient la mémoire d'acquisition, le reconnaiseur de mot et les circuits associés aux entrées/sorties et au contrôle. Ces circuits opèrent d'une façon similaire à ceux de la section principale.

En mode synchrone, elle fonctionne comme une extension de la section principale, ajoutant ainsi 8 voies de données et un reconnaiseur de mot.

Les circuits relatifs au bloc "TRIGGER et base de temps" délivrent les signaux de contrôle permettant le choix du mode de fonctionnement. Ils contiennent une base de temps, un filtre programmable (actif en mode asynchrone seulement), un compteur à retard, un déclencheur et un séquenceur.

La base de temps est générée soit par un oscillateur interne de 100 mhz. soit par des différentes entrées basses fréquences (2 MHz, 200 KHZ, 20 khz et 2 khz) issues de "State machine"

Si l'option timing fonctionne en mode synchrone, le déclenchement est causé par le signal "SYN TRIG" issu de "State machine".

En mode asynchrone, l'option timing est préparée (armed) par le signal "SYNC TRIG" puis déclenchée par une coïncidence sur son propre reconnaiseur de mot.

SEQUENCE D'OPERATIONS:

Après avoir décrit le fonctionnement et le rôle de chaque bloc, l'exemple suivant nous permet de suivre la séquence d'opération du point de vue hardware.

A titre d'exemple, supposant le programme suivant:

- Condition de déclenchement WR:00 (sur le bus de données)
- Fonctionnement synchrone
- Prédéclenchement (16 mots, après l'événement doivent être mémorisés).
- Le "SUT" ne doit pas être arrêté.
- Pas de qualification de données.
- Pas de qualification d'horloge.

Les séquences d'opérations sont les suivantes:

Le CPU charge les RAM et les Latches dans tous les blocs appropriés au programme, active les reconnaisseurs de mot, "State machine", les circuits du "front-end", les déclencheurs et les circuits intégrés d'acquisition.

Lorsque la touche START/STOP est appuyée, pour dérouler le programme, le signal "STORE", issu du bloc mémoire d'acquisition, fait démarrer le processus d'acquisition. La mémorisation se fait au rythme de l'horloge du "SUT".

Les données collectées sont appliquées aux reconnaisseurs de mots et à la mémoire d'acquisition. Tandis que les lignes de contrôle et d'horloge sont envoyées aux circuits du "FRONT-END". Cependant puisqu'il n'y a pas de qualification d'horloge. Les lignes de contrôle du "SUT" ne sont pas utilisées.

Les données sont chargées dans les RAM de la mémoire d'acquisition d'une manière continue. Après 256 cycles, le chargement de la mémoire reprend dès le début. Après la détection de l'événement, le reconnaisseur de mot envoie un signal à "STATE MACHINE", qui à son tour envoie un signal "STOP TRACE TRIG" à la mémoire d'acquisition. Ce signal initialise le compteur à retard qui compte 16 impulsions d'horloge et stoppe la section principale. La section timing reçoit le signal "SYNC TRIG" pour initialiser un compteur puis le stoppe après 16 impulsions d'horloge. Le CPU active les circuits d'affichage pour visualiser les données mémorisées.

CHAPITRE III

M I S E E N O E U V R E

## I-CONCEPT DE PROGRAMMATION DU 7D02

### I- INTRODUCTION

L'exécution d'un test ou d'une fonction se fait à partir d'une structure de programmation du type:

IF-THEN-ELSE.

Les séquences du programme sont entrées à l'aide du clavier. Chaque séquence se programme à l'aide d'un menu dont la conception dérive des langages évolués.

### II- CONTENUS DE LA SYNTAXE "IF-THEN-ELSE"

-Les événements (simples ou composés) sont introduits sous "IF" ou "OR IF"

-Les commandes (simples ou composées) sont introduites sous "THEN DO" ou "ELSE DO". Selon la structure suivante:

-IF

    Événement

-THEN DO

    Commande

-OR IF

    Événement

-THEN DO

    Commande

-OR IF

    Événement

-THEN DO

    Commande

    ⋮

-ELSE DO

    Commande

"IF"; "THEN DO", "OR IF" sont fournies automatiquement par le 7D02, cependant les événements ou les commandes sont introduits par le programmeur suivant la syntaxe ci-dessus.

Si "ELSE DO" est envisagé dans le programme, il peut être introduit sous "ELSE" (se trouvant sur le clavier), entraînant ainsi l'apparition de "ELSE DO" à la place de "IF" ou de "OR IF".

Lorsque l'événement associé à un "IF" ou un "OR IF" devient vrai, la commande sous "THEN DO" qui suit est exécutée. Si plusieurs événements deviennent vrais en même temps, toutes les commandes associées seront exécutées simultanément. Si aucun événement n'est vrai, la commande sous "ELSE DO" est exécutée (Si elle existe).

Remarque

L'utilisateur peut avoir autant de paires ("OR IF"-"THENDO" que la mémoire de programme peut contenir. Par contre, un seul "ELSE DO" peut exister dans un test.

III-TESTS

Un test est une structure qui inclue la syntaxe décrite précédemment.

- Un programme peut contenir au maximum 4 tests. La fin du test est indiquée par un "END".
- Seul un test est activé à un moment donné. C'est à dire que si le test 1 est activé et un événement du test 2 devient vrai, la commande associée ne serait pas exécutée.
- Un test pourrait être activé par la commande "GOTO n" ou "n" est le numéro du test.

IV- EVENEMENTS et COMMANDES

1-Sous "IF" ou "OR IF" on peut introduire les événements suivants:

-WORD RECOGNIZER

--COUNTER

Les paramètres associés au reconnaiseur de mot sont:

WORD RECOGNIZER  $\neq$  "1"

DATA=XX

ADDRESS=XXXX

/NMI=X /IRQ=X FETCH=X R/W=X

BA=X INVAL OP=X EXT TRIG IN=X

TIMING WR=X

Un premier appui sur la touche "WORD RECOGNIZER" nous affiche le menu précédent.

Il est à signaler que les 6 lignes de contrôle dépendent du module personnalisé utilisé (ici c'est relatif au PM Io3 du 6802)

Dans le cas du module personnalisé d'usage général (PM 101) on aura le menu suivant:

```
WORD RECOGNIZER = 1
DATA=XX
ADDRESS=XXXX
Co=X C1=X C2=X C3=X
C4=X C5=X EXT TRIG IN=X
TIMING WR=X
```

- WORD RECOGNIZER = "1"

Ce champ est utilisé pour sélectionner lequel des quatre reconnaisseurs de mot sera utilisé. Il peut prendre les valeurs 1,2,3,4. La valeur par défaut est "1".

-DATA=XX  
-ADDRESS=XXXX } Ces champs spécifient le modèle cherché dans les bus donnée et adresse du "SUT"

Ces champs dépendent du format choisi, ils peuvent être écrits en HEXA, BINAIRE, OCTAL ou ASCII. (Voir format en Annexe)

Ici les 2 digits du champ de donnée et les 4 digits du champ d'adresse représentent l'écriture en HEXA.

-Les champs spécifiques aux six lignes de contrôle représentent le modèle cherché sur les bus de contrôle.

-EXT TRIG IN=X: Ce champ peut comprendre "0" ou "1".

Un "0" ("1") indique que le signal "EXT TRIG IN" doit être actif à l'état bas (haut)

-TIMING WR=X : Ce paramètre n'apparaît que si l'option timing est installée. c'est à dire que si ce module d'option est enfilé dans le 7D02. Il permet la connexion logicielle du reconnaisseur de mot avec celui de la section principale. (Voir option .T)

#### -COUNTER

COUNTER peut être utilisée soit comme commande soit comme événement.

Il est défini comme événement lorsqu'il est introduit dans le programme après "IF", "OR IF" ou "STORE ON".

Il sera considéré comme commande lorsqu'il est introduit dans le programme après "THEN DO" ou "ELSE DO"

Les paramètres associés à "counter(événement)

-COUNTER # 1 = 0002 0-EVENTS  
0 events  
1 uS  
2 MS

- COUNTER # "1": Ce champ numérique est utilisé pour sélectionner l'un des deux compteurs . La valeur par défaut est "1"
- 0002: indique la valeur finale que doit atteindre le compteur pour que l'événement devient vrai. La valeur introduite doit être une valeur décimale entre 0002 et 65534 (incluse) 0002 est la valeur prise par défaut.

-0-EVENTS

0 EVENTS  
1 uS  
2 MS

Dans ce champ on spécifie le mode de comptage.  
Un "0" sélectionne le comptage d'événements  
Un "1" sélectionne le mode temps (en uS)  
Un "2" sélectionne le mode temps (en MS)

2-sous "THEN DO" ou "ELSE do" on peut introduire les commandes suivantes:

- TRIGGER
- COUNTER
- GOTO
- QUALIFY

-TRIGGER : Cette commande permet à l'utilisateur de déclencher la mémorisation à partir d'un point intéressant parmi les données échantillonnées, il définit une fenêtre dans laquelle les données issues du système (SUT) doivent être mémorisées dans la mémoire d'acquisition.

Par la spécification du point de déclenchement, l'utilisateur indique la valeur du compteur à retard associé à la mémoire d'acquisition.

L'exécution de la commande de déclenchement doit activer le compteur à retard. Celui-ci doit s'incrémenter chaque fois qu'un mot est mémorisé dans la mémoire d'acquisition.

L'acquisition est achevée une fois que le compteur aura atteint sa valeur programmée.

-LES PARAMETRES ASSOCIES

TRIGGER 0-MAIN

- 0 MAIN
- 1 TIMING
- 0-BEFORE DATA
  - 0 BEFORE DATA
  - 1 CENTRED
  - 2 AFTER DATA
  - 3 ZERO DELAY
- 0-SYSTEM UNDER TEST CONT
  - 0 SYSTEM UNDER TEST CONT
  - 1 SYSTEM UNDER TEST HALT
- 0-STANDARD CLOCK QUAL
  - 0 STANDARD CLOCK QUAL
  - 1 USER CLOCK QUAL.

0-MAIN Ce menu apparait seulement lorsque l'option timing est placée dans le 7D02. Il indique laquelle des mémoires (principale ou timing) doit être déclenchée avec cette commande.

- 0 MAIN
- 1 TIMING

Si l'option timing n'est pas installée ou si un "0" est introduit dans ce menu, le menu suivant apparaitra. Par contre si l'option timing est installée et un "1" est introduit un groupe de menus différents seront affichés (voir l'option timing page 36)

- 0-BEFORE DATA
  - 0 BEFORE DATA
  - 1 CENTERED
  - 2 AFTER DATA
  - 3 ZERO DELAY

Ce menu est utilisé pour programmer la valeur du compteur à retard associé à la mémoire d'acquisition.

Si "0" est introduit on aura 240 acquisitions après le déclenchement.

Si "1" est introduit on aura 128 acquisitions après le déclen<sup>t</sup>.

Si "2" est introduit on aura 16 acquisitions après le déclen<sup>t</sup>.

Si "3" est introduit on n'aura aucune acquisition après le mot déclencheur.



0-SYSTEM UNDER TEST CONT.  
0 SYSTEM UNDER TEST CONT  
1 SYSTEM UNDER TEST HALT

Ce menu nous indique l'état que doit prendre le "SUT" une fois l'acquisition terminée.

0-STANDARD CLOCK QUAL  
0 STANDARD CLOCK QUAL  
1 USER CLOCK QUAL

Ce menu apparait selon le module personnalisé utilisé. Il permet à l'utilisateur de programmer l'hor-

loge d'état(ou d'échantillonnage).

Un "0" indique que l'analyseur logique utilise directement comme horloge d'état, l'horloge du "SUT".

Un "1" permet la qualification de l'horloge qui, à partir de la sélection de ses fronts, génère une fréquence d'échantillonnage appropriée.

Le menu associé à "1" USER CLOCK QUAL, dépend du module personnalisé. Le menu le plus détaillé est donné en utilisant le PM(IOI) d'usage général. Les autres modules personnalisés n'affichent qu'une partie de ce menu.

1-USER CLOCK QUAL  
0 RISING EDGE OF CLOCK  
1 FALLING EDGE OF CLOCK  
C9-C4 (ANDED CLOCK)=XXXXXX  
1 USER CLOCK SYNTHESIS  
0 STANDARD CLOCK SYNTHESIS  
1 USER CLOCK SYNTHESIS  
1-DIVIDE CLOCK BY 1  
0 DELAY CLOCK BY  
1 DIVIDE CLOCK BY  
ESYNC: C6=X OR C8=X  
WAIT :C7=X OR C9=X

0-RISING EDGE OF CLOCK

1-FALLING EDGE OF CLOCK. Ce menu spécifie lequel des fronts de l'horloge du "SUT" serait actif. C'est à dire sur se fera l'échantillonnage des données.

C9-C4 (ANDED CLOCK)= Ce champ spécifie des conditions sur les lignes de contrôle du "SUT", une fois qu'elles sont vraies, l'horloge du "SUT" est qualifiée.

0-STANDARD CLOCK SYNTHESIS : Ce menu indique que l'horloge n'est pas manipulée par l'utilisateur.

1-USER CLOCK SYNTHESIS: Ceci indique que l'utilisateur a l'intention de manipuler l'horloge du "SUT". Cette sélection fait apparaître le menu suivant: (voir fig 3)

0-DELAY CLOCK BY

1-DIVIDE CLOCK BY

-ESYNC: C6=X OR C8=X

WAIT: C7=X OR C9=X

0-DELAY CLOCK BY: Ce champ sélectionne le nombre de fronts de l'horloge du "SUT" (spécifié par l'utilisateur), après que le signal ESYNC <sup>devient vrai</sup>, pour générer l'impulsion d'échantillonnage.

1-DIVIDE CLOCK BY: Ayant sélectionné ceci, le signal d'horloge sera divisé (de 1 à 4) pour générer l'impulsion d'échantillonnage. Le signal "ESYNC" synchronise la division quand son état devient vrai.

- "ESYNC" C6=X OR C8=X :

La présence d'un "0" ou d'un "1" sur l'une des deux lignes C6 ou C8 provoque la synchronisation de l'horloge d'échantillonnage.

"0"/"1" indique qu'il suffit que l'une des 2 lignes soit à "0"/"1" pour que le signal ESYNC soit vrai.

-WAIT C7=X OR C9=X :

Ces lignes sont utilisées pour produire le signal WAIT (signal d'attente). Chaque fois que ce signal est activé, il fait retarder le signal d'horloge d'un cycle. Un 0/1 veut dire qu'il suffit que l'une des deux lignes soit à 0/1 pour que le signal soit vrai.

Les lignes C4-C9 dépendent du module personnalisé utilisé.

NOTE

Lors de l'utilisation du PM IO: RELATIF au 6802, l'introduction d'un "1" dans le menu associé à "0-STANDARD CLOCK QUAL" c'est à dire la sélection de "1-USER CLOCK QUAL" fait apparaître le menu suivant ;

1-FALLING EDGE OF CLOCK

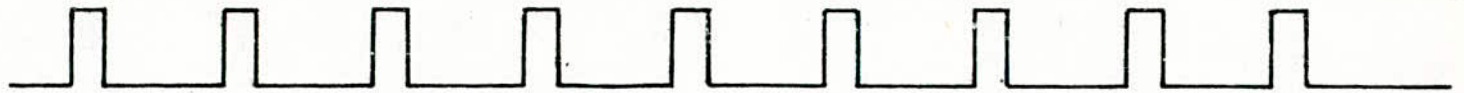
0-RISING EDGE OF CLOCK

1-FALLING EDGE OF CLOCK

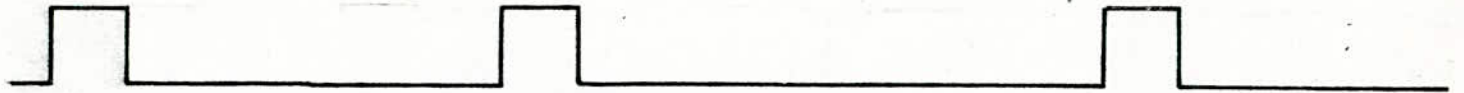
C9-C4 (anded clock) =XXXLXX

- Fig. 3 - Synthèse d'horloge "User clock synthesis" (PM 101)

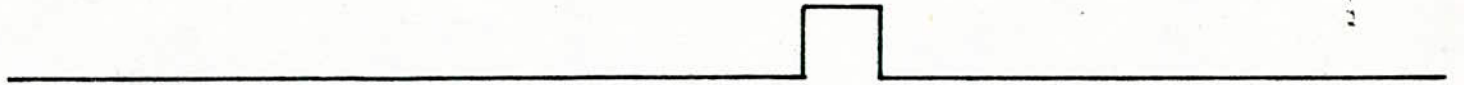
Impulsions d'horloge  
provenant du MP 101



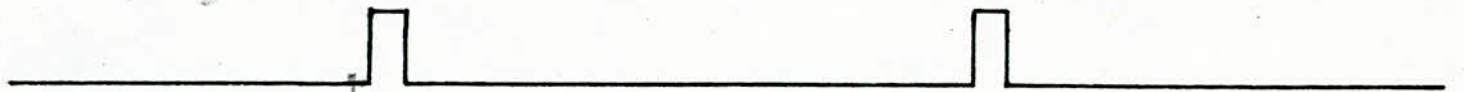
ESYNC



WAIT

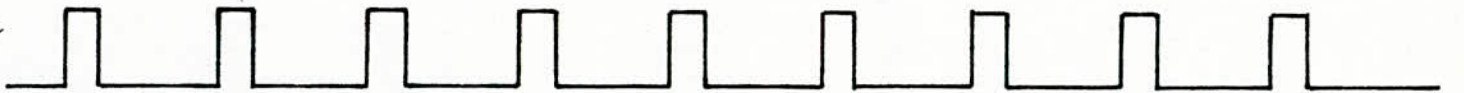


Horloge d'échantillon-  
nage "State clock".

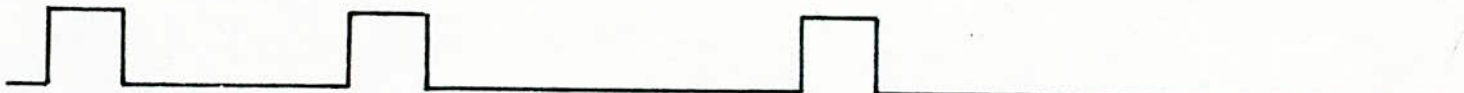


Synthèse d'horloge "User clock synthesis" - retard par 2 - "Delay by 2"

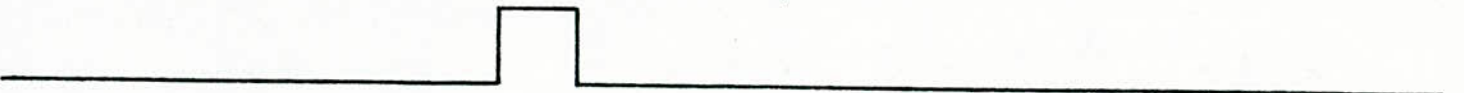
Impulsions d'horloge  
provenant du MP 101



ESYNC



WAIT



Horloge d'échantillon-  
nage "State clock"



Synthèse d'horloge - Division par 2 - "Divide by 2"

C9-C4 sont les lignes de contrôle relatives aux PM IO2/PM IO3 qui peuvent être utilisées pour redéfinir l'horloge d'échantillonnage du 7D02 .Celle-ci est générée lorsque chaque état des lignes de contrôle C9-C4 est vrai et qu'on est en présence du front sélectionné de l'horloge du "SUT"(pin 37 du 6802).

-C9 /RESET:C'est la ligne reset du uP.

Quand cette ligne est à l'état bas le Mpu est inactif et les informations contenues dans les registres seront perdues . Le passage à l'état haut de cette ligne conduit le MPU. à commencer la séquence de redémarrage;un programme d'initialisation s'exécute alors.

lorsque C9 est à 0,la qualification d'horloge ne se fait que sur la séquence de redémarrage.

-C8 :Cette ligne est connectée à la masse (ground),elle doit toujours mise à zéro "0" ou à "X"

- C7 /HALT :Cette ligne permet de communiquer l'état d'arrêt du système au 7D02 .La qualification d'horloge se fait sur l'état d'arrêt si C7 est mise à "0".

-C6=VMA:C'est la pin VMA du uProcesseur.Cette ligne indique au "SUT" qu'il y'a une adresse valide sur le bus adresse. La valeur par défaut est "1" ce qui signifie que l'horloge n'est qualifiée que lorsque le signal VMA est à l'état haut.

-C5:Cest une ligne qui permet de détecter les codes opérations illégaux,elle est issue du "PM" vers l'analyseur logique. sont utilisation comme qualificateur d'horloge n'est pas recommandée.

-C4:BA: C'est la ligne bus disponible du uP vers le "SUT". Elle passe à l'état haut indiquant que le processeur est à l'arrêt et que le bus adresse est disponible .Ceci se réalise quand la ligne "HALT"est à l'état bas ou que le processeur est en état "WAIT",résultat d'exécution d'une instruction "WAIT"(attente d'interruption).

-LA commande "COUNTER"

"COUNTER" est utilisé comme commande s'il est introduit dans le programme immédiatement sous "THEN DO" ou "ELSE DO", ou comme une partie d'une commande composée.

LES PARAMETRES ASSOCIES

COUNTER ≠ 0-EVENTS

0 EVENTS

1 uS

2 MS

0-INCREMENT

0 INCREMENT

1 RESET

ou

0-RUN

0 RUN

1 STOP

2 RESET AND RUN

-COUNTER ≠ "1"

Ce champ est utilisé pour sélectionner l'un des deux compteurs.

0-INCREMENT

0 INCREMENT

1 RESET

Ce menu apparait seulement si le mode comptage d'événement a été sélectionné.

Un "0" indique que le compteur doit s'incrémenter chaque fois que la commande est exécutée.

Un "1" sert à initialiser le compteur chaque fois que la commande est exécutée.

0-RUN

0 RUN

1 STOP

2 RESET AND RUN

Ce menu n'apparait que lorsque le mode temps a été sélectionné.

L'introduction d'un "0" causera le début d'un comptage. Le compteur continu à s'incrémenter jusqu'à ce que la commande soit exécutée. L'introduction d'un "1" stoppe le comptage et maintient la valeur atteinte.

L'introduction d'un "2" initialise et redémarre le comptage.

### La commande "GOTO"

Cette commande provoque le transfert d'un test à un autre au cours d'exécution d'un programme.

Elle peut aussi être utilisée comme commande dans le mode immédiat pour exécuter un test dans un programme (voir annexe)

### PARAMETRES

GOTO "1" : Le seul paramètre qui apparaît, quand la touche "GOTO" est appuyée, indique le numéro du test dans le programme courant auquel le contrôle sera transféré. Il peut prendre les valeurs 1, 2, 3, 4. Si "GOTO" est introduite dans le mode programme, la valeur par défaut est le numéro du test courant plus un (+1).

### - "QUALIFY"

La qualification de donnée est entièrement différente de la qualification d'horloge.

La qualification d'horloge détermine le moment où la donnée sera échantillonnée.

La qualification de donnée introduit un critère dans lequel la donnée échantillonnée doit être mémorisée. "QUALIFY" peut être utilisée comme commande ou comme structure. Elle est comme structure quand elle est introduite seule ou immédiatement après un "END TEST".

Elle est utilisée comme commande si elle est introduite dans un programme immédiatement après "THEN DO" ou "ELSE DO" ou comme une partie d'une commande composée.

### - La commande "QUALIFY"

Lorsque "QUALIFY" est utilisée comme commande, elle signifie "mémorisation" (STORE). Sur chaque cycle où la commande est exécutée, la donnée échantillonnée sera mémorisée.

### - Le bloc "QUALIFY" (QUALIFY BLOCK)

Lorsque QUALIFY est utilisée comme structure, elle crée le bloc Qualify. CE bloc est toujours actif lorsque le programme se déroule sans tenir compte de l'activation d'un autre test. Il est défini en appuyant sur la touche "QUALIFY" suivie d'un événement (ou un événement composé) et terminé par "END".

Les données ne sont mémorisées que sur les cycles sur lesquels l'événement dans le bloc QUALIFY est vrai.

-UN seul bloc de qualification peut être défini dans un programme.

-Ce bloc peut apparaitre avant ou après n'importe quel test.

---

### AUTRES INDICATIONS

- La touche **[ ]** EST utilisée pour combiner des événements (ou des commandes) pour introduire des événements (ou commandes) composés.

Entre les crochets, on peut introduire un ou plusieurs événements, comme on peut les complémentés en appuyant sur la touche "NOT"

-L'événement composé devient vrai, si au moins un des événements est vrai.

L'exécution de la commande composée implique l'exécution simultanée de toutes les commandes contenues entre crochets.

-La touche "DELETE" sert à éliminer ce qui n'est pas désiré dans le programme.

"DATA SCROLLING": Les deux touches (DATA SCROLLING) sont utilisées pour déplacer la fenêtre de visualisation durant le mode d'affichage.

"CURSOR" : Les quatre touches "CURSOR" déplacent le curseur clignottant dans la direction indiquée sur la touche.

## II-L'OPTION "TIMING"

### I-INTRODUCTION:

L'option "TIMING" permet d'élargir le champ d'utilisation du 7D02. Son usage primordial est d'acquérir les informations en mode asynchrone, cependant, elle peut être utilisée pour acquérir des informations en mode synchrone.

Dans les deux modes de fonctionnement, on distingue les trois fonctions suivantes:

- L'acquisition .
- La reconnaissance .
- Le déclenchement .

#### -Mode SYNCHRONE:

##### a-L'acquisition:

1-La mémoire d'acquisition s'étend sur 255 positions de 8 bits.

2-L'horloge d'échantillonnage est la même que celle de la section principale.

3-La qualification de données se fait de la même manière que dans la section principale.

##### b-La reconnaissance

1-Si le modèle spécifié dans le champ associé au reconnaisseur de mot apparaît sur les 8 voies de la sonde P6451, l'événement est reconnu comme vrai.

2-La sélection du reconnaisseur de mot est disponible seulement si TIMING WR=0 ou 1. Ainsi le champ associé à ce mot est affiché sur l'écran.

##### c-Le déclenchement:

Il se fait par l'exécution de la commande suivante:

"TRIGGER 1-TIMING".

#### -Mode ASYNCHRONE:

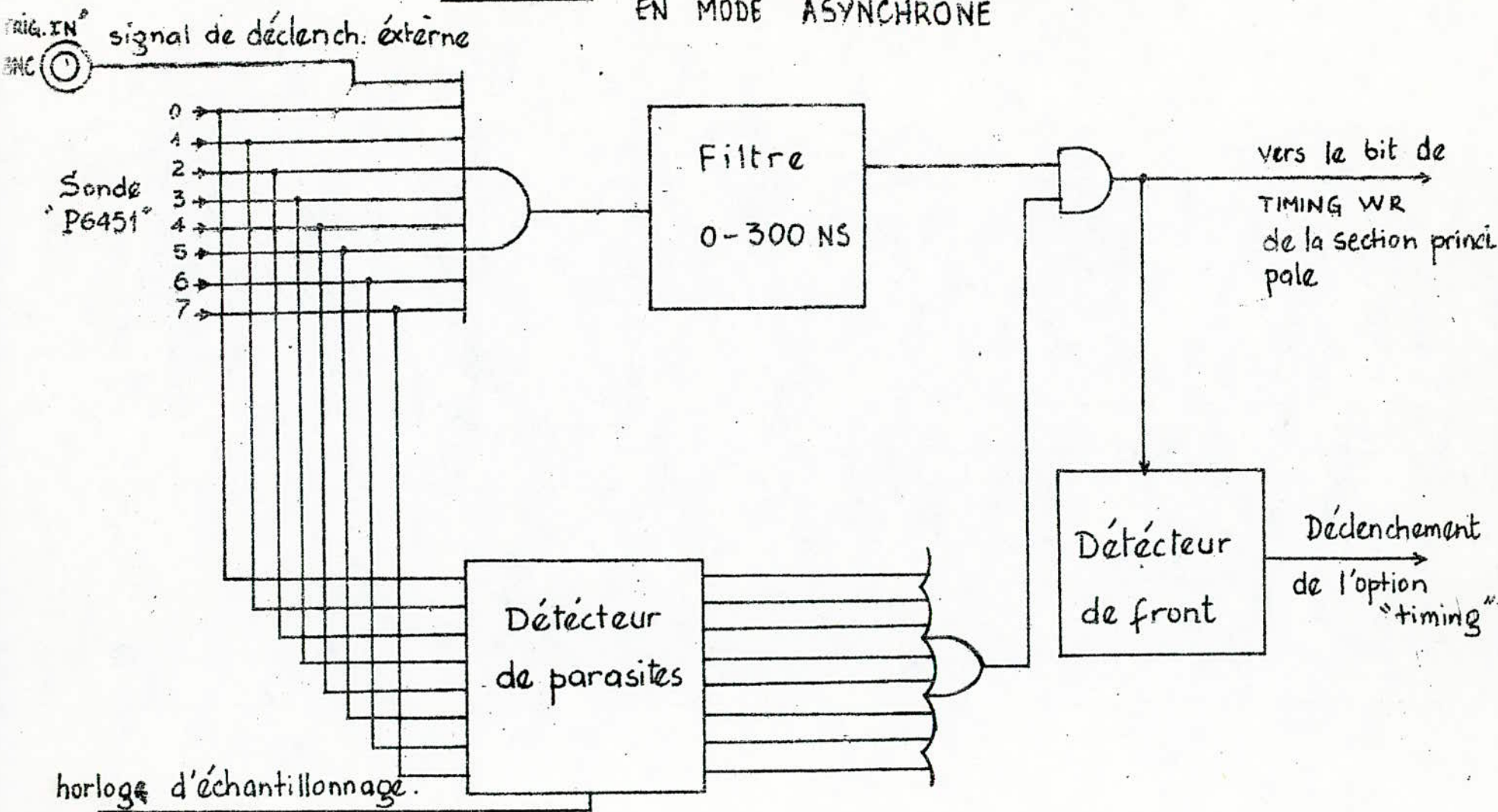
Dans ce mode l'option "TIMING" fonctionne comme un analyseur indépendant. Elle utilise sa propre base de temps pour générer l'horloge d'échantillonnage.

L'avantage qu'offre ce mode est la détection de parasites, ainsi que la mémorisation des signaux avec des périodes d'échantillonnage variables.

Les différentes fonctions de l'option "timing" sont illustrées par le schéma de la figure ( 4 ).



FIG: 4 RECONAISSANCE D'ÉVÉNEMENTS EN MODE ASYNCHRONE



a-L'acquisition:

1-La même mémoire d'acquisition du mode synchrone est utilisée dans ce cas.

2-La base de temps interne à l'option "TIMING",génére l'horloge d'échantillonnage dont la gamme s'étale de 20 ns à 5 Ms c'est à dire de 200 HZ à 50 MHZ.

3-255 positions mémoires sont réservées à la mémorisation de parasites.

Remarque: Dans ce mode on n'a pas de qualification de données, donc toutes les données sont échantillonnées et mémorisées.

b-LA reconnaissance d'événements:

Dans ce mode,l'événement est défini par un ~~recon~~reconnaisseur de mot de 8 bits,un signal de déclenchement externe et un reconnaisseur de parasites de 8 bits.

L'événement n'est vrai que si le mot de données,le mot de parasites et le déclencheur externe sont vrais simultanément.

b-1 Le reconnaisseur de mot

Il s'étend sur 8 bits,le mot est considéré comme vrai si le modèle spécifié par l'utilisateur coïncide avec la donnée acquise sur les huit voies de la sonde.

b-2 Le signal de déclenchement externe

Ce signal provient de la fiche BNC de la face avant du 7D02. Cette ligne peut être considérée comme la neuvième voie du reconnaisseur de mot.

Remarque: Le mot à analyser et le signal de déclenchement externe doivent avoir un temps de maintien de 10 ns au minimum(cela si la valeur du filtre est égale à zéro).

Le filtre programmable sert à éliminer les signaux dont le temps de maintien est <sup>(ns)</sup> inférieur à la valeur programmée  $t$  du filtre telle que  $t=20K\gamma$  avec  $K=0,1,2,\dots,15$ .

b-3 Le reconnaisseur de parasite:

Un parasite est défini comme un signal qui transite plus d'un fois entre deux impulsions d'échantillonnage successives(voir #35) Le mot de parasites est vrai si au moins un parmi les 8 bits de ce mot coïncide avec le modèle spécifié par l'utilisateur.

Ce champ peut prendre les valeurs "1" ou "X"

Le temps de maintien du parasite doit être au minimum de 5 ns pour être visualisé.

c-Le déclenchement:

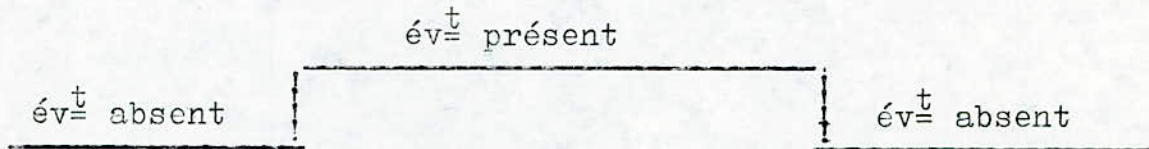
Le déclenchement dans ce mode est plus compliqué que dans le mode synchrone, vu que le déclenchement se fait sur des transitions et qu'il nécessite deux conditions:

1-Préparer l'option "timing" à accepter le déclenchement ceci à lieu en exécutant l'instruction "TRIGGER 1- TIMING".

2-Détéction de l'événement.

Dans ce mode, le déclenchement se fait suivant les transitions qui pourraient se produire c-à-d

- Passage d'événement absent à événement présent.
- " " " présent " absent.



II- CONCEPT DE PROGRAMMATION de l'option "timing"

VOIR TABLEAU 1

-définitions:

-TIMING WR= Le signal issu de l'option "timing"(fig.4 ) va informer la section principale de la présence ou l'absence de l'événement à travers "TIMING WR# "

UN "1" signifie que la reconnaissance se fait sur la présence de l'événement de "timing"

Un "0" signifie que la reconnaissance se fait sur l'absence de cet événement (sur tout le cycle d'horloge)

Un "X" signifie que l'événement de "timing" est ignoré par la section principale.

\* En asynchrone, le signal issu de l'option "timing" est le résultat d'un "ET" logique du:

- Reconnaisseur de mot
- Déclencheur externe
- Reconnaisseur de parasite

\* En synchrone, le signal est seulement un reconnaisseur de mot.

WORD RECOGNIZER # 1  
DATA=XX  
ADDRESS=XXXX  
/NMI=X /IRQ=X FETCH=X R/W=X  
BA=X INVAL OP=X EXT TRIG IN=X  
TIMING WR=0 OU 1



THRESHOLD V. = 0-PLUS 1.40  
0 PLUS  
1 MINUS

0-SYNC  
0 SYNC  
1 ASYNC

WORD RECOGNIZER=XXXXXXXX

SAMPLE PERIOD 1 \* 1-100NS  
0 10 NS  
1 100NS  
2 1 US  
3 10 US  
4 100US  
5 1 MS

WORD RECOGNIZER=XXXXXXXX  
EXT TRIG IN=X  
GLITCH RECOGNIZER=XXXXXXXX  
FILTER=000 NS



TRIGGER 1-TIMING

0 MAIN  
1 TIMING

DECLENCHEMENT DE LA  
SECTION PRINCIPALE

0-BEFORE DATA  
0 BEFORE DATA  
1 CENTERED  
2 AFTER DATA  
3 DELAYED

THRESHOLD V. = 0-PLUS 1.40  
0 PLUS  
1 MINUS

1-ARM ASYNC, TRIG ON WR  
0 SYNC, TRIGGER IMMEDIATE  
1 ARM ASYNC, TRIG ON WR   
2 ARM ASYNC, TRIG ON WR 

SAMPLE PERIOD 1 \* 1-100NS  
0 10 NS  
1 100NS  
2 1 US  
3 10 US  
4 100US  
5 1 MS

WORD RECOGNIZER=XXXXXXXX  
EXT TRIG IN=X  
GLITCH RECOGNIZER=XXXXXXXX  
FILTER=000 NS

-THRESHOLD V.= Ce champ détermine un seuil de basculement pour la sonde P6451. Un signal à l'entrée inférieur à ce seuil est considéré comme un état logique bas. Un niveau supérieur correspond à un niveau logique haut. La gamme s'étale de -6,35 à +6,35 avec un pas de 0,05 v (50 mv).

-"SYNC - ASYNC": Il spécifie le mode pour lequel le reconnaisseur d'événement de l'option "timing" est connecté au reconnaisseur de mot principal.

0="Synchronous"(synchrone) veut dire une extension du reconnaisseur de mot principal.

1="Asynchronous"(asynchrone) c-à-d indépendamment de la section principale, la reconnaissance se fait au rythme de l'horloge délivrée par la base de temps interne de l'option "timing".

"WORD RECOGNIZER": Ce champ est une spécification sur 8 voies du reconnaisseur d'événement de l'option "timing".

-Sample PERIOD:

Sous "TIMING WR#" , ce paramètre définit la période d'échantillonnage de parasites tandis que sous "TRIGGER 1-TIMING", il définit celle des entrées de la sonde.

Les multiplicateurs valides sont 1 , 2 et 5.

L'intervalle des valeurs possibles s'étale de 20 ns à 5 ms .

"EXT TRIG IN=" Ce champ n'apparaît qu'en mode asynchrone, il est considéré comme une neuvième voie du reconnaisseur d'événement de l'option "timing". Il peut prendre les valeurs suivantes:

X=non chargé

0=niveau bas

1=niveau haut.

"GLITCH RECOGNIZER=" n'apparaît qu'en mode asynchrone. Il spécifie lesquelles des voies sont utilisées pour contrôler la présence de parasites. La sortie du reconnaisseur de parasite est le résultat d'un "OR" logique de 8 Voies. Il peut prendre X ou 1.

1 =chercher le parasite sur la voie spécifiée.

"FILTER": A l'aide de ce champ, on impose la durée pendant laquelle les informations collectées par la sonde P6451 et la ligne du déclencheur externe doivent se maintenir avant d'être comparées avec les valeurs programmées.

La gamme s'étale de 0 à 300 ns avec des pas de 20 ns.

TRIGGER 1-TIMING : Cette commande est utilisée pour spécifier laquelle des sections sera utilisée pour la mémorisation de données; ici c'est l'option "timing" (0 c'est la section principale).

-POSITION DE Déclenchement :

Ce menu permet de choisir le nombre de positions mémoires qui doivent être mémorisées relativement au mot déclencheur.

0-BEFORE DATA: Signifie 240 positions mémoires seront mémorisées après le mot déclencheur.

1-CENTERED :128 POSITIONS après le déclenchement.

2-AFTER DATA :16 " " " "

3-DELAYED :Donne la possibilité de mémoriser les données les plus lointaines du mot déclencheur (de 00000 à 65534 positions ) si le retard est de 00000 seulement le mot déclencheur ainsi que les données qui le précèdent sont mémorisées. Si le retard est supérieur à 255 seulement les dernières 255 données qualifiées sont mémorisées et on n'aura pas la mémorisation du mot déclencheur.

-MODE DE DECLENCHEMENT: A l'aide du menu affiché, on sélectionne le mode de déclenchement de l'option "timing".

En mode synchrone, déclenchement immédiat.

En mode asynchrone ; le déclenchement est provoqué sur détection d'un front (montant ou descendant ).

### III- définition de données et parasites

Les données concernent les 8 canaux, l'affichage de chaque voie est représentée par une paire de lignes sur l'écran; celle du bas est relative aux informations chronologiques et celle du haut est relative aux parasites.

ON note que les parasites ne sont pas affichés en mode synchrone.

Un parasite est défini lorsque plus d'une transition est détectée entre deux impulsions d'échantillonnage successives. Les parasites sont affichés dans le mode signaux sous forme de flèches placées au dessus du canal. (Voir figure 5) .

Un parasite du premier ordre est défini comme un nombre pair de transition entre deux impulsions d'échantillonnage successives.

Un parasite du second ordre est défini comme un nombre impair (supérieur à 1 ) de transition entre deux impulsions d'échantillonnage successives.

Lors de l'affichage et pour distinguer les deux types de parasites, une flèche sur un niveau indique un parasite du premier ordre, tandis que sur un front, elle indique un parasite du second ordre. Dans le mode état, un parasite est indiqué par un astérisque(\*) sur la colonne appropriée.

#### -ORDRE D'AFFICHAGE DES VOIES :

Le positionnement de voie sur l'écran est assurée par le champ associé à "CHANNEL ORDER=". Le tracé du chronogramme en haut de l'écran correspond au canal le plus à droite.

Par exemple, si CHANNEL ORDER=53216704", le chronogramme du canal 4 doit se placer en haut de l'écran suivi par les chronogrammes des canaux 0,7,6,1,2,3 et enfin le canal 5 en bas de l'écran.

Un "X" sur un canal élimine l'affichage de la voie correspondante.

#### -LECTURE D'UNE INFORMATION PAR POSITIONNEMENT DU CURSEUR

La ligne "CURSOR" contient: (fig 6)

"DATA": valeur du mot de données spécifié par le marqueur. Cette valeur peut être affichée en binaire, en hexa ou en octal .

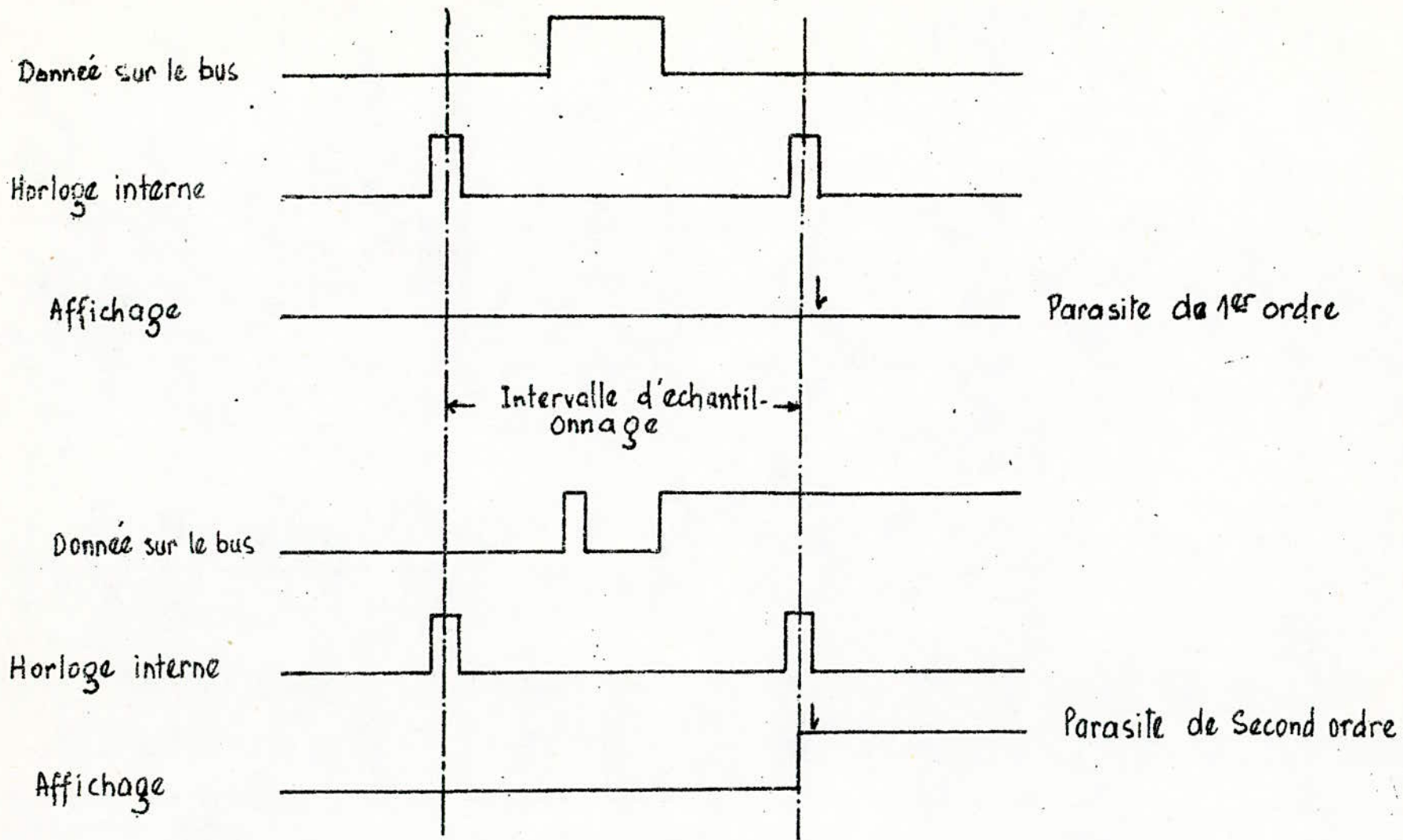
"LOC": Position mémoire relative au début des données acquises.

"DELTA": Différence de temps entre la position courante du curseur et la position de déclenchement. Si la position courante correspond à la position de déclenchement on aura l'affichage de "TRIGGER" sur la ligne "CURSOR".

-ECHELLE : Le chronogramme peut être affiché avec deux échelles différentes:

TEMPS 1 (0-x 1 : multiplication par 1 ),

TEMPS 4 (1-x 4 : multiplication par 4 ).



- FIG :5 -

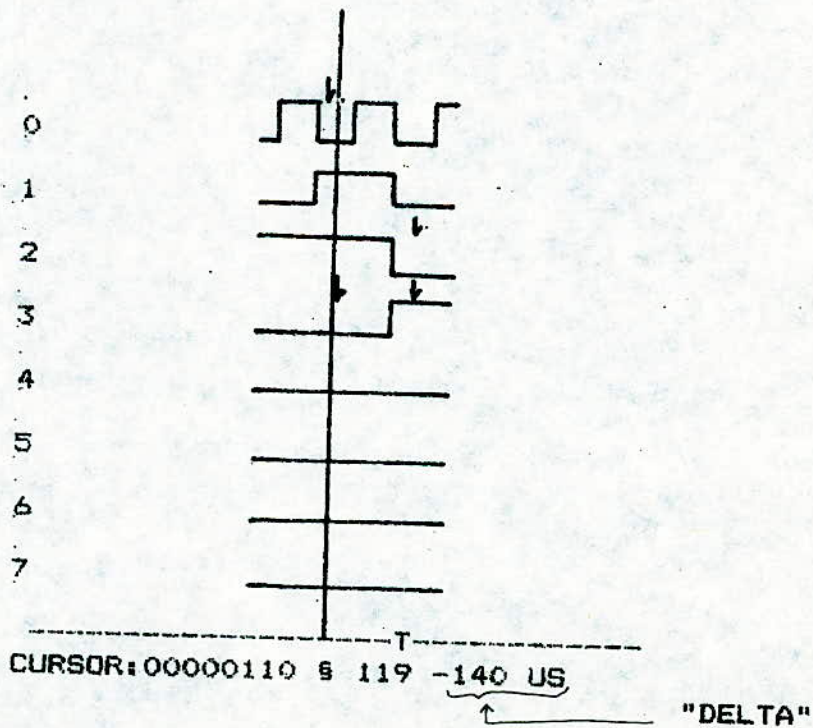
Definition et affichage de parasites



LOC	DATA	GLITCH	
118	00000110	-----*	
119	00000110	-----*	-140 US
...	.....	.....	↑ "DELTA"
...	.....	.....	

126	00001000	-----**	TRIGGER
...	.....	.....	

.....



- FIG:6 -

L'agrandissement est sélectionné lorsque le chronogramme est affiché sur l'écran. La sélection est faite sur le dernier menu en bas de l'écran. L'échelle sélectionnée persiste jusqu'à ce que l'utilisateur la modifie. La valeur par défaut est 1.

-AFFICHAGE EN TABLE D'ETAT : (Voir figure 7 )

LA table d'état contient trois champs (position mémoire, donnée parasite ). Les données et les parasites sont affichés suivant l'ordre des canaux spécifiés dans le champ associé à "CHANNEL ORDER ".

LOC= position mémoire de la donnée acquise.

TRIGGER=position du mot déclencheur.

DATA=mot de donnée en accord avec l'ordre des canaux.

GLITCH=Les informations parasites sont positionnées suivant l'ordre courant des canaux. Un "X" indique la présence du parasite sur le canal correspondant et un "-" indique qu'il n'y a pas de parasites.

Lors de l'acquisition , en mode synchrone, l'affichage relatif aux parasites reste vierge, de même, lorsque l'utilisateur élimine l'affichage de parasite (par un choix dans le menu du mode format).

-----  
CTR1=00000 EVT TRIG LOC = 126  
CTR2=00000 EVT 5 US/SAMPLE  
-----

LOC DATA      GLITCH  
112 00000001 -----  
... ..  
... ..

125 00000101 -----  
126 00000110 ----- TRIGGER  
... ..  
... ..

.....

-----  
DISPLAY <-ACQMEM 1-TIMING  
CHANNEL ORDER=76543210  
0-STATE DISPLAY  
-----

CHAPITRE IV

A P P L I C A T I O N S

## I-INTRODUCTION

Dans ce chapitre nous allons présenter quelques exemples d'application sur les KITS;D5 de **MOTOROLA** et TMS 990/189 de **TEXAS INSTRUMENTS**

## II-APPLICATIONS SUR LE KIT D5.

LES premiers exemples (1 à 4) sont relatifs à l'utilisation de la séction principale seulement. Le programme introduit dans le KIT consiste en l'incrémentation et le transfert de l'accumulateur A, initialement remis à zéro, vers la mémoire d'adresse 0020 jusqu'à la valeur FF. (voir fig 8).

### a-EXEMPLE 1 (fig9 )

Le programme, dans cet exemple, est une simple reconnaissance de mot. La détection de l'adresse E000 sur le bus adresse provoquera le déclenchement de la mémorisation des données. En appuyant sur la touche "WORD RECOGNIZER", l'analyseur affiche les premières lignes du programme.

Remarque: Les valeurs introduites sont en HEXA .

On peut écrire une valeur, lorsque le curseur clignotant est positionné sur le digit désiré (représenté en vidéo inversé). La ligne de donnée (DATA=) n'est pas spécifiée dans cet exemple ce qui veut dire qu'elle doit être ignorée par le reconnaiseur de mot.

La ligne suivante indique l'adresse qui doit être cherchée sur le bus d'adresse du système sous test.

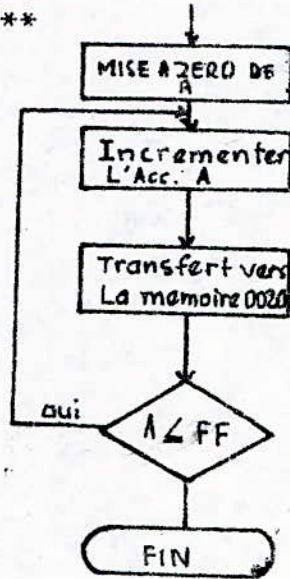
Les deux lignes suivantes contiennent six paramètres de contrôle (/NMI, /IRQ, FETCH, R/W, BA, INVAL OP) relatifs au uP 6802, et une ligne de déclenchement externe (EXT TRIG IN).

En introduisant un "0" ou un "1" dans les champs de ces paramètres, les lignes de contrôle seront une partie de l'événement. Dans cet exemple aucune ligne de contrôle n'est utilisée, ainsi que la ligne de déclenchement externe.

Le dernier paramètre n'est pas spécifié car on n'utilise pas l'événement de la séction "TIMING".

En appuyant sur la touche "TRIGGER" le 7D02 affiche les lignes suivantes du programme. Le choix disponible des paramètres est écrit sous 0-MAIN. On peut maintenant choisir le déclenchement de la mémoire principale, en laissant le "0", ou celui de la mémoire "TIMING".

\*\*\*\*\*  
 \* ORGANIGRAMME \*  
 \*\*\*\*\*



\*\*\*\*\*  
 \* PROGRAMME \*  
 \*\*\*\*\*

LP	CLR A	E000	4F
	INC A	E001	4C
	STA A \$20	E002	97
	CMP A #\$FF	E003	20
	BNE LP	E004	81
	SWI	E005	FF
		E006	26
		E007	F9
		E008	3F

- FIG : 8 -

PROGRAMME

=====

```

TEST 1
1IF
1 WORD RECOGNIZER # 1
1 DATA=XX
1 ADDRESS=E000
1 /NMI=X /IRQ=X FETCH=X R/W=X
1 BA=X INVAL OP=X EXT TRIG IN=X
1 TIMING WR=X
1THEN DO
1 TRIGGER O-MAIN
1 O-BEFORE DATA
1 O-SYSTEM UNDER TEST CONT.
1 O-STANDARD CLOCK QUAL.
END TEST 1

```

RESULTAT

=====

- FIG: 9 -

```

-----
CTR1 =00000 EVT TRIG LOC = 015
CTR2 =00000 EVT TRIG IN TEST 1
-----

```

LOC	ADDRESS	OPERATION	/IRQ/NMI
015T	E000	CLR	11
016	E001	4C READ	11
017	E001	INC A	11
018	E002	97 READ	11
019	E002	STA A	11
020	E003	20 READ	11
021	0020	01 WRITE	11
022	E004	CMP A #\$FF	11
023	E005	FF READ	11
024	E006	BNE \$E001	11
025	E007	F9 READ	11
026	E001	INC A	11
027	E002	97 READ	11
028	E002	STA A \$20	11
029	E003	20 READ	11
030	0020	02 WRITE	11
031	E004	CMP A #\$FF	11

```

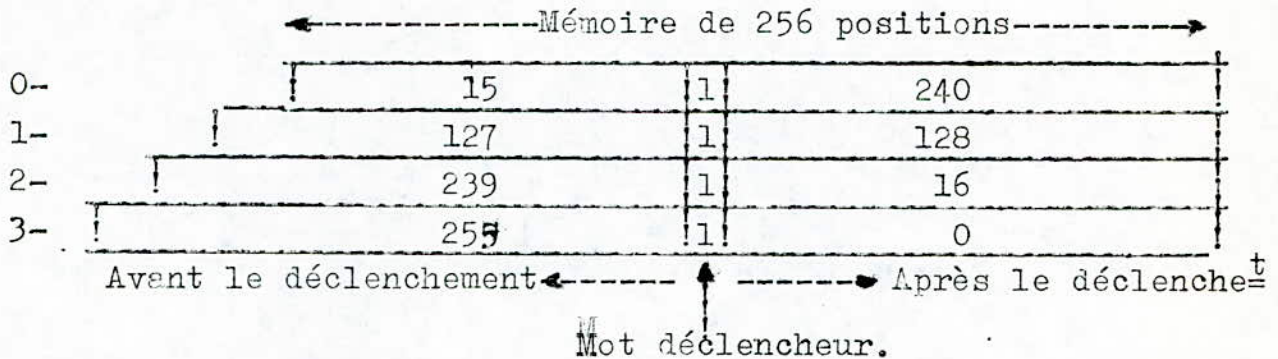
-----
DISPLAY <-ACQMEM O-MAIN
1-MNEMONIC
0 ABSOLUTE
1 MNEMONIC

```

- FIG: 10 -

Dans cet exemple on a laissé 0 c'est à dire déclenchement de la mémoire principale.

-Le curseur est placé maintenant sur la position suivante c-à-d 0-BEFORE DATA, on peut donc spécifier sur quelle position mémoire on veut acquérir les données. On a quatre choix illustrés par la figure suivante:



-LE curseur étant placé sur 0-SYSTEM UNDER TEST CONT; on peut spécifier au 7D02 ce qu'il doit faire lorsque; l'événement est trouvé, le déclenchement a eu lieu et l'acquisition est achevée. 0-SYSTEM UNDER TEST CONT : veut dire que le uP du système sous test continue son fonctionnement lorsque l'acquisition de données est terminée.

1-SYSTEM UNDER TEST HALT : indique que le uP stoppe son fonctionnement une fois que les données sont acquises.

-En plaçant le curseur sur la position suivante on peut déterminer l'horloge d'échantillonnage.

Ici on a laissé les valeurs affichées par défaut, c-à-d;

0-SYSTEM UNDER TEST CONT et 0- STANDARD CLOCK QUAL .

On note que le curseur peut être positionné en n'importe quel champ.

Pour terminer le programme il suffit d'appuyer sur la touche "END". Sur l'affichage  $\left. \begin{matrix} 1 \text{ OR IF} \\ 1 \end{matrix} \right\}$  est remplacé par END TEST 1.

Lorsque la touche "START" est appuyée, le 7D02 commence la recherche de E000.

REMARQUE: Sur l'écran il apparaît trois parties; (voir fig 10)

1-En haut de l'écran une partie en vidéo inversé dans laquelle est affiché l'état du 7D02.



2-Au milieu de l'écran il apparaît le programme ou le résultat de ce dernier (valeurs acquises).

3-En bas de l'écran une autre partie en vidéo inversé affiche les modes immédiats.

Lorsque l'événement s'est présenté, l'affichage prend la forme de la figure- 10 -.

On note que seulement deux lignes de contrôle /IRQ et /NMI sont affichées d'une manière explicite.

La lecture de toute la mémoire est obtenue par le déplacement de la fenêtre de visualisation, en utilisant les touches "DATA SCROLLING". Dans cet exemple le mot déclencheur est en position 015 et il est indiqué par un T.

L'affichage peut être en "mnémonic" ou en "absolue" suivant la sélection d'un "O" ou d'un "I" en bas de l'écran.

#### RETOUR AU MODE PROGRAMME

Une fois qu'on ait terminé avec l'acquisition de données, et on veut dérouler un autre programme, on a besoin de changer le mode d'affichage en mode programme, ceci se fait en appuyant la séquence de touches suivantes:

IMMEDIATE, DISPLAY, PROGRAM

Si on veut effacer le programme courant et d'introduire un nouveau programme il suffit juste de déplacer le curseur vers la fin du programme et d'appuyer la touche "DELETE". A chaque appui on peut effacer une ou plusieurs du programme courant en commandant par la dernière ligne.

b-Exemple 2

BUT: Utilisation de la commande "QUALIFY" ainsi que "COUNTER" comme commande et comme événement.

PROGRAMME:(fig11), on veut mémoriser dix états de la mémoire d'adresse 0020 puis déclenchement et mémorisation de tous les mots une fois que le compteur a compté dix événements.

RESULTAT:(fig12). Les états relatifs à la mémoire d'adresse 0020 occupent les dix premières positions(000-009) de la mémoire d'acquisition principale.

Les positions 010-255 sont occupées par les mots relatifs à l'exécution du programme.

c-Exemple 3

BUT: Utilisation simultanée de la commande QUALIFY et du bloc QUALIFY.

PROGRAMME:(fig13): Dans le test 1, la reconnaissance se fait sur les adresses 0020 ou E003. Si l'adresse 0020 est détectée, elle sera mémorisée, car elle est qualifiée dans le programme (grâce au bloc QUALIFY), et l'analyseur passe à l'exécution du test 2. Si l'adresse E000 est détectée avant l'adresse 0020, elle sera mémorisée (grâce à la commande QUALIFY).

Dans le test 2, l'analyseur se déclenche à la reconnaissance de l'adresse E007. Le mot correspondant sera le seul à être mémorisé et l'analyseur passe au test 1 (et ainsi de suite).

RESULTAT:

On constate (fig14) que seul les données qualifiées sont mémorisées.

Le mot déclencheur s'est positionné à la position 002. On aura donc 240 positions mémoires après le déclenchement, la dernière position sera 242.

d-Exemple 4

BUT: Relever le temps d'exécution d'un programme.

PROGRAMME:(fig15) : A la détection de l'adresse E000, le compteur 1 est activé. Celui-ci sera arrêté lors de la détection de l'adresse E008. Le temps écoulé entre E000 et E008 correspond au temps d'exécution.

Les adresses 0020 sont à mémoriser.

\*\*\*\*\*  
 \* PROGRAMME \*  
 \*\*\*\*\*

```

TEST 1
1 IF
1 WORD RECOGNIZER # 1
1 DATA=XX
1 ADDRESS=0020
1 /NMI=X /IRQ=X FETCH=X R/W=X
1 BA=X INVAL OP=X EXT TRIG IN=X
1 TIMING WR=X
1 THEN DO
1
1 QUALIFY
1 COUNTER # 1 0-EVENTS
1 O-INCREMENT
1
1 OR IF
  
```

```

1 COUNTER # 1 = 00010 0-EVENTS
1 THEN DO
1
1 QUALIFY
1 TRIGGER 0-MAIN
1 1-CENTERED
1 0-SYSTEM UNDER TEST CONT.
1 0-STANDARD CLOCK QUAL.
1 GOTO 2
1
1 END TEST 1
TEST 2
2 ELSE DO
2 QUALIFY
END TEST 2
  
```

- FIG: 11 -

CTR1=00010 EVT TRIG LOC = 010  
 CTR2=00000 EVT TRIG IN TEST 1

LOC	ADDRESS	OPERATION	/IRQ/NMI
000	0020	01 WRITE	11
001	0020	02 WRITE	11
002	0020	03 WRITE	11
003	0020	04 WRITE	11
004	0020	05 WRITE	11
005	0020	06 WRITE	11
006	0020	07 WRITE	11
007	0020	08 WRITE	11
008	0020	09 WRITE	11
009	0020	0A WRITE	11
010T	E004	CMF A #FF	11
011	E005	FF READ	11
012	E006	BNE \$E001	11
013	E007	F9 READ	11
014	E001	INC A	11
015	E002	97 READ	11
016	E002	STA A \$20	11

DISPLAY <-ACOMEM 0-MAIN  
 1-MNEMONIC  
 0 ABSOLUTE  
 1 MNEMONIC

- FIG: 12 -

```
*****  
* PROGRAMME *  
*****
```

```
TEST 1  
1IF  
1 WORD RECOGNIZER # 1  
1 DATA=XX  
1 ADDRESS=0020  
1 /NMI=X /IRQ=X FETCH=X R/W=X  
1 BA=X INVAL OP=X EXT TRIG IN=X  
1 TIMING WR=X  
1THEN DO  
1 GOTO 2  
1OR IF  
1 WORD RECOGNIZER # 2  
1 DATA=XX  
1 ADDRESS=E003  
1 /NMI=X /IRQ=X FETCH=X R/W=X  
1 BA=X INVAL OP=X EXT TRIG IN=X  
1 TIMING WR=X  
1THEN DO  
1 QUALIFY  
END TEST 1  
TEST 2  
2IF  
2 WORD RECOGNIZER # 3  
2 DATA=XX  
2 ADDRESS=E007
```

```
2 /NMI=X /IRQ=X FETCH=X R/W=X  
2 BA=X INVAL OP=X EXT TRIG IN=X  
2 TIMING WR=X  
2THEN DO  
2  
2 TRIGGER 0-MAIN  
2 0-BEFORE DATA  
2 0-SYSTEM UNDER TEST CONT.  
2 0-STANDARD CLOCK QUAL.  
2 QUALIFY  
2 GOTO 1  
2  
END TEST 2  
QUALIFY  
QSTORE ON  
Q WORD RECOGNIZER # 4  
Q DATA=XX  
Q ADDRESS=0020  
Q /NMI=X /IRQ=X FETCH=X R/W=X  
Q BA=X INVAL OP=X EXT TRIG IN=X  
Q TIMING WR=X  
END QUALIFY
```

- FIG : 13 -

**\*\* RESULTAT \*\***

=====

-----  
CTR1=00000 EVT TRIG LOC = 002  
CTR2=00000 EVT TRIG IN TEST 2  
-----

LOC	ADDRESS	OPERATION	/IRQ/NMI
000	E003	20 READ	11
001	0020	01 WRITE	11
002T	E007	F9 READ	11
003	E003	20 READ	11
004	0020	02 WRITE	11
005	E007	F9 READ	11
006	E003	20 READ	11
007	0020	03 WRITE	11
008	E007	F9 READ	11
009	E003	20 READ	11
010	0020	04 WRITE	11
011	E007	F9 READ	11
012	E003	20 READ	11
013	0020	05 WRITE	11
014	E007	F9 READ	11
015	E003	20 READ	11
016	0020	06 WRITE	11

-----  
DISPLAY <-ACMEM 0-MAIN

1-MNEMONIC

0 ABSOLUTE

1 MNEMONIC  
-----

- FIG : 14 -

\*\*PROGRAMME\*\*  
 \*\*\*\*\*

```

TEST 1
1 IF
1 WORD RECOGNIZER # 1
1 DATA=XX
1 ADDRESS=E000
1 /NMI=X /IRQ=X FETCH=X R/W=X
1 BA=X INVAL OP=X EXT TRIG IN=X
1 TIMING WR=X
1 THEN DO
1 COUNTER # 1 1-US
1 0-RUN
1 OR IF
1 WORD RECOGNIZER # 2
1 DATA=XX
1 ADDRESS=E008
1 /NMI=X /IRQ=X FETCH=X R/W=X
1 BA=X INVAL OP=X EXT TRIG IN=X
1 TIMING WR=X
1 THEN DO
1 COUNTER # 1 1-US
1 1-STOP
END TEST 1
QUALIFY
Q STORE ON
Q WORD RECOGNIZER # 3
Q DATA=XX
Q ADDRESS=0020
Q /NMI=X /IRQ=X FETCH=X R/W=X
Q BA=X INVAL OP=X EXT TRIG IN=X
Q TIMING WR=X
END QUALIFY
  
```

\_ FIG: 15 \_

---

CTR1=03422 US TRIG LOC = \*\*\*  
 CTR2=00000 EVT LAST TEST = 1

---

LOC	ADDRESS	OPERATION	/IRQ/NMI
000	0020	01 WRITE	11
001	0020	02 WRITE	11
002	0020	03 WRITE	11
003	0020	04 WRITE	11
004	0020	05 WRITE	11
005	0020	06 WRITE	11
006	0020	07 WRITE	11
007	0020	08 WRITE	11
008	0020	09 WRITE	11
009	0020	0A WRITE	11
010	0020	0B WRITE	11
011	0020	0C WRITE	11
012	0020	0D WRITE	11
013	0020	0E WRITE	11
014	0020	0F WRITE	11
015	0020	10 WRITE	11
016	0020	11 WRITE	11

---

DISPLAY <- ACDMEM 0-MAIN  
 1-MNEMONIC  
 0 ABSOLUTE  
 1 MNEMONIC

---

\_ FIG: 16 \_

Remarque: Dans le cas où on n'utilise pas la qualification de données, la mémoire d'acquisition reste vide, puisqu'il n'y a pas de déclenchement dans le programme.

RESULTAT: Le compteur 1 à compté un temps de 3422 uS.

Les valeurs qu'occupait la mémoire d'adresse 0020 sont mémorisées. (voir fig 16)

Les exemples ci-après correspondent à l'utilisation en modes synchrone et asynchrone de l'option timing.

Le programme introduit dans le KIT permet l'envoi des données sur le port d'entrée/sortie du KIT D5 (Port A ,PIA USER).

( Voir fig 17)

Exemple 5:

BUT: Déclenchement des deux séctions, principale et timing à partir de l'événement de la séction principale.

PROGRAMME: (fig18).

-Déclenchement des deux séctions.

-Fonctionnement synchrone.

RESULTAT:

-Séction principale:

-La figure(19) représente deux fenêtres de visualisation qui correspondent aux positions mémoires(000 - 018)et(164-182)

-La position 015 est occupée par le mot déclencheur .

-La position 171 est occupée par le mot E016 3F qui correspond à la dernière instruction du programme.

-Les positions 171 à 255 correspondent au programme moniteur du KIT.

-Séction TIMING:

a-Table d'état:

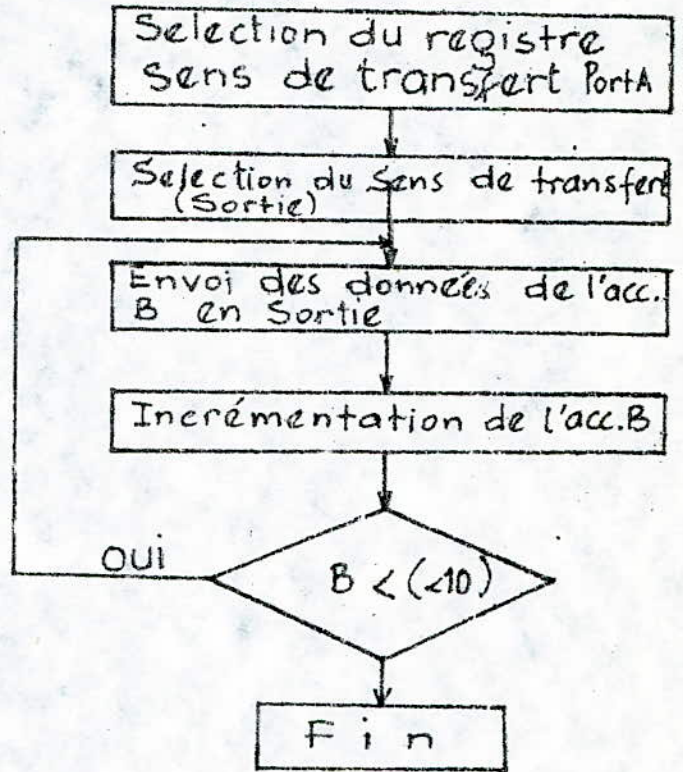
-Deux fenêtres de visualisation relatives aux positions mémoire(006-023) et (050-067) sont illustrées par la figure(20)

-Le déclenchement se fait à l'apparition du mot 00000000 qui correspond au passage à zéro de l'accumulateur B (CLR B )

b-DIAGRAMME

De même la figure(21) correspond à deux fenêtres de visualisation différentes.

\*\*\*\*\*  
 \* ORGANIGRAMME \*  
 \*\*\*\*\*



\*\*\*\*\*  
 \* PROGRAMME \*  
 \*\*\*\*\*

LDA A #00		E000	86
STA A \$E481	SELECTION DU REGISTRE SENS DE	E001	00
	TRANSFERT (PORT A)	E002	B7
LDA A #\$FF		E003	E4
STA A \$E480	SELECTION DU SENS DE TRANSFERT	E004	B1
	(SORTIE)	E005	86
STA A \$E481	SELECTION DU REGISTRE DE DONNEES	E006	FF
CLR B		E007	B7
LP STA B \$E480	ENVOI DES DONNEES DE L'ACC. B EN	E008	E4
	SORTIE	E009	80
INC B		E00A	B7
CMP B #10		E00B	E4
BNE LP		E00C	B1
SWI		E00D	5F
		E00E	F7
		E00F	E4
		E010	80
		E011	5C
		E012	C1
		E013	10
		E014	26
		E015	F8
		E016	3F



\*\*\*\*\*  
 \*\*\*\*\*PROGRAMME\*\*\*\*\*  
 \*\*\*\*\*

```

TEST 1
1IF
1 WORD RECOGNIZER # 1
1 DATA=5C
1 ADDRESS=E011
1 /NMI=X /IRQ=X FETCH=X R/W=X
1 BA=X INVAL OP=X EXT TRIG IN=X
1 TIMING WR=X
1THEN DO
1
1 TRIGGER 0-MAIN
1   0-BEFORE DATA
1   0-SYSTEM UNDER TEST CONT.
1   0-STANDARD CLOCK QUAL.
1 TRIGGER 1-TIMING
1   0-BEFORE DATA
1   THRESHOLD V. = 0-PLUS 1.40
1   0-SYNC, TRIGGER IMMEDIATE
1
END TEST 1
  
```

- FIG: 18 -

-----  
 CTR1=00000 EVT TRIG LOC = 015  
 CTR2=00000 EVT TRIG IN TEST 1  
 -----

LOC	ADDRESS	DATA	CNTL
000	E006	FF	R11
001	E007	B7	F11
002	E008	E4	R11
003	E009	80	R11
004	E480	FF	W11
005	E00A	B7	F11
006	E00B	E4	R11
007	E00C	81	R11
008	E481	FF	W11
009	E00D	5F	F11
010	E00E	F7	R11
011	E00E	F7	F11
012	E00F	E4	R11
013	E010	80	R11
014	E480	00	W11
015T	E011	5C	F11
016	E012	C1	R11
017	E012	C1	F11
018	E013	10	R11

-----  
 DISPLAY <-ACQMEM 0-MAIN  
 0-ABSOLUTE  
 -----

-----  
 CTR1=00000 EVT TRIG LOC = 015  
 CTR2=00000 EVT TRIG IN TEST 1  
 -----

LOC	ADDRESS	DATA	CNTL
164	E006	0F	W11
165	E011	5C	F11
166	E012	C1	R11
167	E012	C1	F11
168	E013	10	R11
169	E014	26	F11
170	E015	FB	R11
171	E016	3F	F11
172	E017	03	R11
173	E418	17	W11
174	E417	E0	W11
175	E416	00	W11
176	E415	00	W11
177	E414	FF	W11
178	E413	10	W11
179	E412	D4	W11
180	FFFA	F7	R11
181	FFFB	D2	R11
182	F7D2	01	F11

-----  
 DISPLAY <-ACQMEM 0-MAIN  
 0-ABSOLUTE  
 -----

- FIG: 19 -

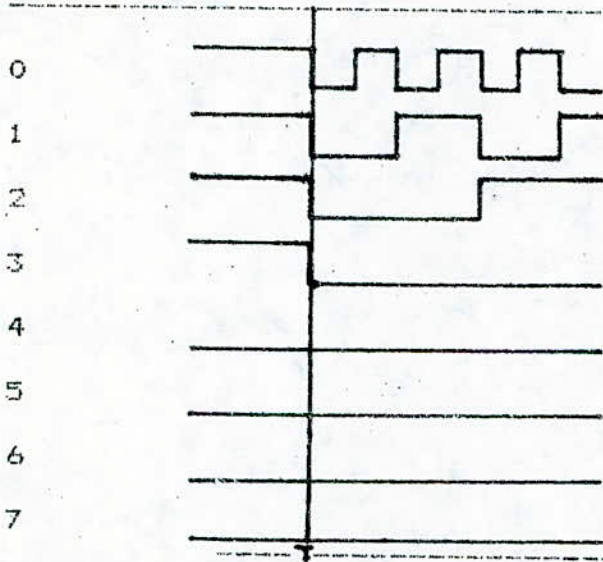
\*\*\* RESULTATS DE L'OPTION TIMING \*\*\*  
 \*\*\*\*\*

CTR1=00000 EVT TRIG LDC = 014 CTR2=00000 EVT SYNCHRONOUS	CTR1=00000 EVT TRIG LDC = 014 CTR2=00000 EVT SYNCHRONOUS
LOC DATA	LOC DATA
006 00001111	050 00000011
007 00001111	051 00000011
008 00001111	052 00000011
009 00001111	053 00000011
010 00001111	054 00000100
011 00001111	055 00000100
012 00001111	056 00000100
013 00001111	057 00000100
014 00000000 -----TRIGGER-----	058 00000100 -----+44-----
015 00000000	059 00000100
016 00000000	060 00000100
017 00000000	061 00000100
018 00000000	062 00000100
019 00000000	063 00000100
020 00000000	064 00000101
021 00000000	065 00000101
022 00000000	066 00000101
023 00000000	067 00000101
DISPLAY <-ACQMEM 1-TIMING CHANNEL ORDER=76543210 0-STATE DISPLAY	DISPLAY <-ACQMEM 1-TIMING CHANNEL ORDER=76543210 0-STATE DISPLAY

- FIG : 20 -

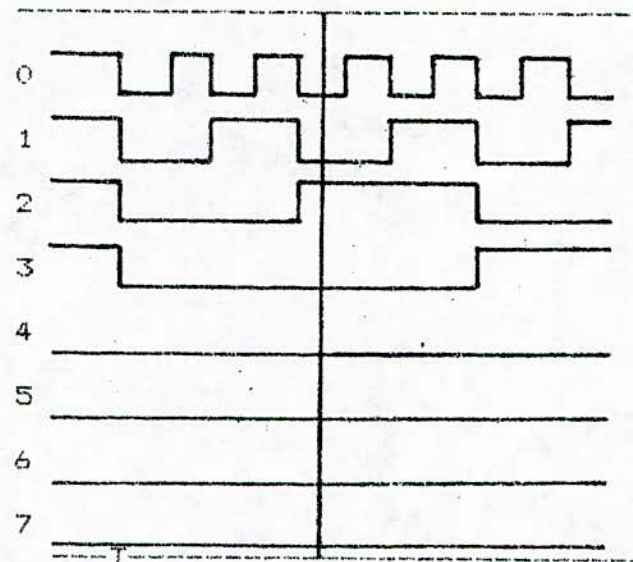
\*\*\* RESULTATS DE L'OPTION TIMING \*\*\*  
 \*\*\*\*\*

CTR1=00000 EVT TRIG LOC = 014  
 CTR2=00000 EVT SYNCHRONOUS



CURSOR:0000000 5 014 TRIGGER

CTR1=00000 EVT TRIG LOC = 014  
 CTR2=00000 EVT SYNCHRONOUS



CURSOR:0000100 5 058 +44

DISPLAY <-ACQMEM 1-TIMING  
 CHANNEL ORDER=76543210  
 1-TIMING DIAGRAM 0-\*1

DISPLAY <-ACQMEM 1-TIMING  
 CHANNEL ORDER=76543210  
 1-TIMING DIAGRAM 0-\*1

A gauche:

Le curseur est sur la position de déclenchement(00000000) occupant la position mémoire 014 .

A droite:

Le diagramme est décalé à gauche, le curseur est sur 00000100 correspondant à la position mémoire 058.

+44 représente le temps écoulé entre le déclenchement et la présence du mot 00000100.

EXEMPLE 6:

BUT: dans cet exemple, on veut déclencher, en mode synchrone, la mémorisation à partir d'un événement de la section timing

L'événement est défini par le mot 00000111. (Voir fig 22)

RESULTAT: A la détection du mot 00000111, le déclenchement a eu lieu. (fig 23)

EXEMPLE 7:

BUT: Déclenchement, en mode asynchrone de l'option timing à partir de son propre événement.

PROGRAMME: (fig 24)

L'événement est défini par WR=XXXXXX10.

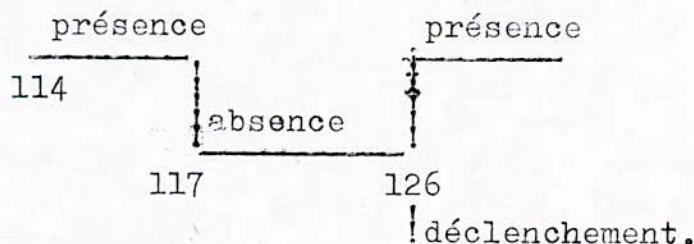
La première détection de ce mot prépare l'option (T) au déclenchement, la seconde la fait déclencher.

La fréquence d'échantillonnage est de  $1/5\mu\text{S} = 200 \text{ KHZ}$ .

RESULTAT: (fig 25)

ON remarque que le déclenchement n'a eu lieu que sur la seconde présence du mot XXXXXX10

En effet en position mémoire 114, le mot est détecté pour la première fois, celui-ci disparaît puis réapparaît une seconde fois (position 126) provoquant ainsi le déclenchement.



EXEMPLE 8 :

BUT: Déclenchement de la section timing en mode asynchrone par un événement de la section principale.

PROGRAMME

=====

```

TEST 1
1 IF
1 WORD RECOGNIZER # 1
1 DATA=XX
1 ADDRESS=XXXX
1 /NMI=X /IRQ=X FETCH=X R/W=X
1 BA=X INVAL OP=X EXT TRIG IN=X
1 TIMING WR=1
1 THRESHOLD V. = 0-PLUS 1.40
1 0-SYNC
1 WORD RECOGNIZER=00000111
1 THEN DO
  
```

```

1
1 TRIGGER 1-TIMING
1 1-CENTERED
1 THRESHOLD V. = 0-PLUS 1.40
1 0-SYNC, TRIGGER IMMEDIATE
1 TRIGGER 0-MAIN
1 2-AFTER DATA
1 0-SYSTEM UNDER TEST CONT.
1 0-STANDARD CLOCK QUAL.
1
END TEST 1
  
```

- FIG: 22 -

CTR1=00000 EVT TRIG LOC = 239				CTR1=00000 EVT TRIG LOC = 126			
CTR2=00000 EVT TRIG IN TEST 1				CTR2=00000 EVT SYNCHRONOUS			
LOC	ADDRESS	OPERATION	/IRQ/NMI	LOC	DATA		
238	E480	07 WRITE	11	118	00000110		
239T	E011	INC B	11	119	00000110		
240	E012	C1 READ	11	120	00000110		
241	E012	CMP B #10	11	121	00000110		
242	E013	10 READ	11	122	00000110		
243	E014	BNE \$E00E	11	123	00000110		
244	E015	FB READ	11	124	00000110		
245	E00E	STA B \$E480	11	125	00000110		
246	E00F	E4 READ	11	126	00000111	-----	TRIGGER-----
247	E010	B0 READ	11	127	00000111		
248	E480	08 WRITE	11	128	00000111		
249	E011	INC B	11	129	00000111		
250	E012	C1 READ	11	130	00000111		
251	E012	CMP B #10	11	131	00000111		
252	E013	10 READ	11	132	00000111		
253	E014	BNE \$E00E	11	133	00000111		
254	E015	FB READ	11	134	00000111		
255	E00E	STA B \$E480	11	135	00000111		

DISPLAY <-ACQMEM 0-MAIN  
1-MNEMONIC

DISPLAY <-ACQMEM 1-TIMING  
CHANNEL ORDER=76543210  
0-STATE DISPLAY

- FIG: 23 -

\*\* PROGRAMME \*\*

=====

```
TEST 1
1 IIF                                1 THEN DO
1 WORD RECOGNIZER #1                1 TRIGGER 1-TIMING
1 DATA=XX                          1 1-CENTERED
1 ADDRESS=XXXX                      1 THRESHOLD V. = 0-PLUS 1.40
1 /NMI=X /IRQ=X FETCH=X R/W=X      1 1-ARM ASYNC, TRIG ON WR┘
1 BA=X INVAL OP=X EXT TRIG IN=X    1 SAMPLE PERIOD 5 * 2-1 US
1 TIMING WR=1                       1 WORD RECOGNIZER=XXXXXX10
1 THRESHOLD V. = 0-PLUS 1.40       1 EXT TRIG IN=X
1 1-ASYNC                           1 GLITCH RECOGNIZER=XXXXXXXX
1 SAMPLE PERIOD 5 * 2-1 US         1 FILTER=000 NS
1 WORD RECOGNIZER=XXXXXX10        END TEST 1
1 EXT TRIG IN=X
1 GLITCH RECOGNIZER=XXXXXXXX
```

- FIG: 24 -

\*\* RESULTAT \*\*

=====

```
-----
CTR1=00000 EVT TRIG LOC = 126
CTR2=00000 EVT 5 US/SAMPLE
-----
LOC DATA      GLITCH
112 00000001  -----
113 00000001  -----
114 00000010  -----
115 00000010  -----
116 00000010  -----
117 00000011  -----
118 00000011  -----
119 00000011  -----
120 00000100  -----30US-----
121 00000100  -----
122 00000100  -----
123 00000101  -----
124 00000101  -----
125 00000101  -----
126 00000110  ----- TRIGGER
127 00000110  -----
128 00000110  -----
129 00000111  -----
-----
DISPLAY <-ACQMEM 1-TIMING
CHANNEL ORDER=76543210
0-STATE DISPLAY
-----
```

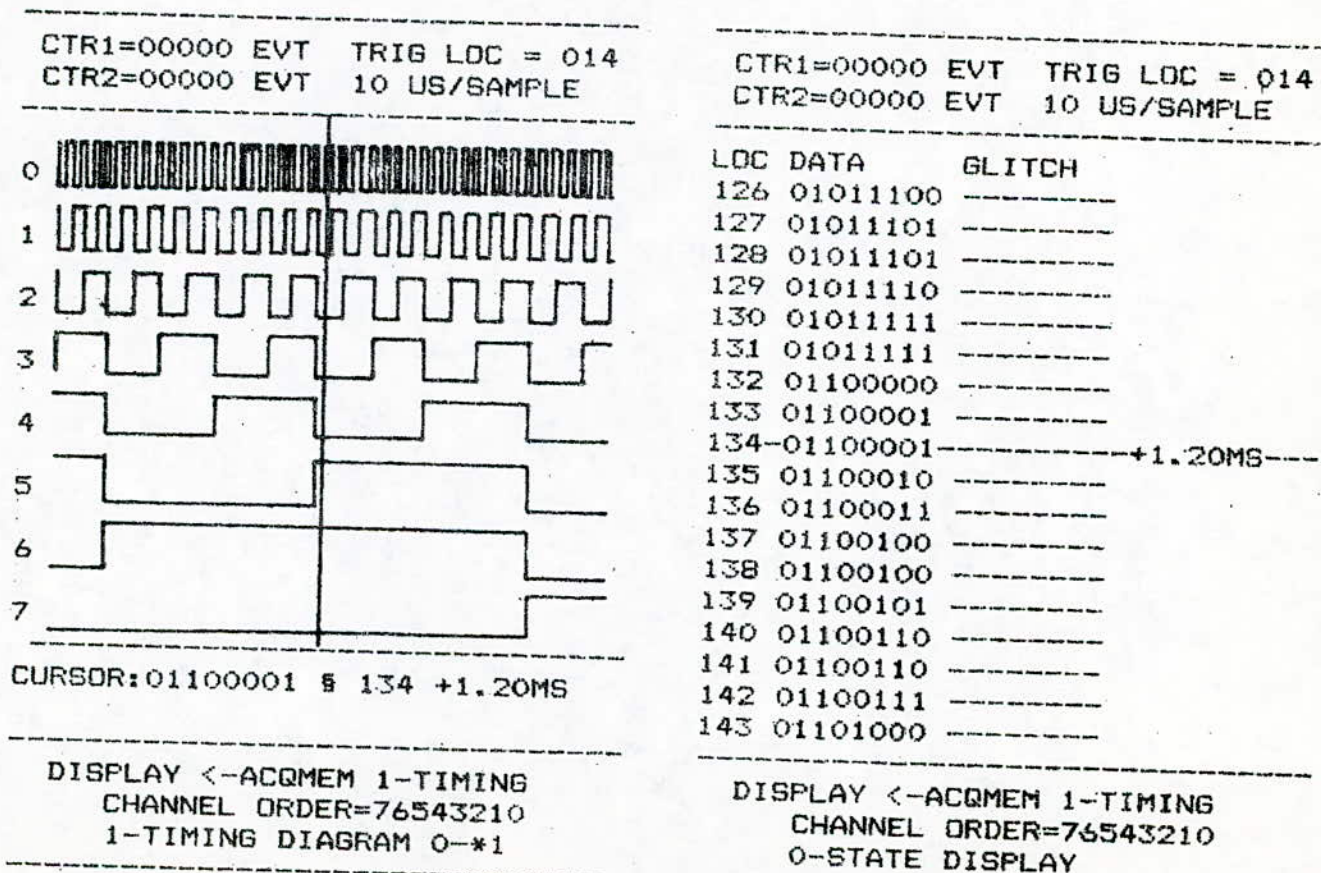
- FIG: 25 -

\*\* PROGRAMME \*\*

```
TEST 1
1IF
1 WORD RECOGNIZER # 1
1 DATA=XX
1 ADDRESS=E000
1 /NMI=X /IRQ=X FETCH=X R/W=X
1 BA=X INVAL OP=X EXT TRIG IN=X
1 TIMING WR=X
1THEN DO
1 TRIGGER 1-TIMING
1 O-BEFORE DATA
1 THRESHOLD V. = 0-PLUS 1.40
1 1-ARM ASYNC, TRIG ON WR
1 SAMPLE PERIOD 1 * 3-10 US
1 WORD RECOGNIZER=00001111
1 EXT TRIG IN=X
1 GLITCH RECOGNIZER=XXXXXXXXX
1 FILTER=000 NS
END TEST 1
```

-FIG:26

\*\* RESULTATS \*\*



-FIG:27

REMARQUE:La valeur maximale que doit prendre l'ACCUMULATEUR B est FO au lieu de 10 dans les exemples précédents.

Programme:(fig 26)

-L'adresse E000 prépare l'option timing à accepter le déclenchement.

-La présence du mot 00001111 provoque le déclenchement de la section timing.

Remarque:Avec la fréquence choisie(100KHZ),on a l'avantage de mémoriser plus d'informations vis à vis temps,mais on a moins de précision.

RESULTAT; VOIR fig 27.

EXEMPLE 9 (fig 28)

BUT:-visualisation des effets de rebondissement des touches du clavier du KIT.

-Déclenchement sur des parasites.

PROGRAMME:La section timing doit se déclencher immédiatement à la présence d'un parasite sur l'une des quatre voies(0,1,2,3) de la sonde. (se référer au schéma du KIT D5,manuel d'utilisation MEK 6802 D5E(D1)).

Les voies 0,1,2 et 3 sont connectées respectivement aux colonnes 1,2,3,4 du clavier.

La voie 5 est connectée à la sortie de la porte NAND du74LS20.

La voie 6 est connectée à la pin 37 du PIA pour prélever le signal de contrôle /NMI.

RESULTAT:Le déclenchement a eu lieu un cycle d'échantillonnage après l'apparition du parasite.Ce parasite est dû au rebondissement de la touche ,sur la colonne 2,de la matrice clavier. On remarque qu'il y a deux types de parasites:

-Parasite de premier ordre;position 157 (X)

-Parasite de second ordre;position 153 et 155.

(Voir fig 29 )

REMARQUE: Dans ce cas, aucun programme n'est introduit dans le KIT.



\*\* PROGRAMME \*\*  
 =====

```

TEST 1
1ELSE DO
1 TRIGGER 1-TIMING
1   3-DELAYED 00100
1   THRESHOLD V. = 0-PLUS 1.40
1   1-ARM ASYNC, TRIG ON WR
1   SAMPLE PERIOD 2 * 0-10 NS
1   WORD RECOGNIZER=XXXXXXXX
1   EXT TRIG IN=X
1   GLITCH RECOGNIZER=XXXX1111
1   FILTER=300 NS
END TEST 1
  
```

— FIG: 28 —

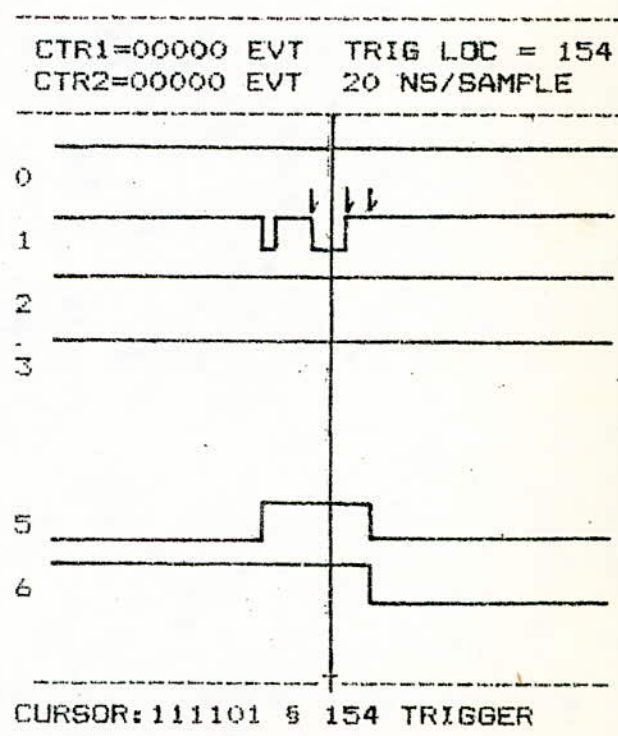
\*\* RESULTATS \*\*  
 =====

CTR1=00000 EVT TRIG LOC = 154  
 CTR2=00000 EVT 20 NS/SAMPLE

---

LOC DATA	GLITCH
146 101111	-----
147 101111	-----
148 101111	-----
149 101111	-----
150 101101	-----
151 111111	-----
152 111111	-----
153 111101	-----*
154-111101	-----TRIGGER-----
155 111111	-----*
156 111111	-----
157 001111	-----*
158 001111	-----
159 001111	-----
160 001111	-----
161 001111	-----
162 001111	-----
163 001111	-----

DISPLAY <-ACQMEM 1-TIMING  
 CHANNEL ORDER=X65X3210  
 0-STATE DISPLAY



DISPLAY <-ACQMEM 1-TIMING  
 CHANNEL ORDER=X65X3210  
 1-TIMING DIAGRAM

— FIG: 29 —

### III-APPLICATION SUR LA CARTE TM 990/I89 DE TEXAS INSTRUMENT.

Il existe deux méthodes essentielles permettant à un système de mesurer un intervalle de temps, ou de générer une temporisation; Les boucles de temporisation contrôlées par programme, et l'horloge temps réel.

Dans l'exemple suivant, on s'intéressera à la méthode utilisant l'horloge temps réel.

Vu que le TMS 9980 n'a pas de module personnalisé spécifique, on utilisera donc le PM IOI d'usage général.

Le programme introduit dans la carte permet de visualiser sur les quatre LEDs le contenu d'un registre (R1) qui s'incrémente à la fréquence programmable de l'horloge temps réel du CRU.

(Voir ORGANIGRAMME et PROG. figures (30) et (31)).

#### EXEMPLE 10:

BUT: -Visualiser sur l'écran:

- 1-Les valeurs des bits de sortie du CRU.
  - 2-L'interruption provoquée par l'horloge.
- Mesurer la fréquence de l'horloge temps réel pour différentes valeurs de temporisation.

PROGRAMME: (voir fig 32).

TEST 1: Reconnaissance de l'état haut de l'interruption attendue sur la ligne de contrôle Co, puis transfert vers le test 2.

TEST 2: Le passage du signal d'interruption de 1 à 0, provoque l'activation d'un compteur (avec remise à zéro préalable) et transfert vers le test 3.

TEST 3: Le passage au TEST 4 est conditionné par remise à l'état haut du signal d'interruption.

TEST 4: La présence de la seconde interruption provoque;

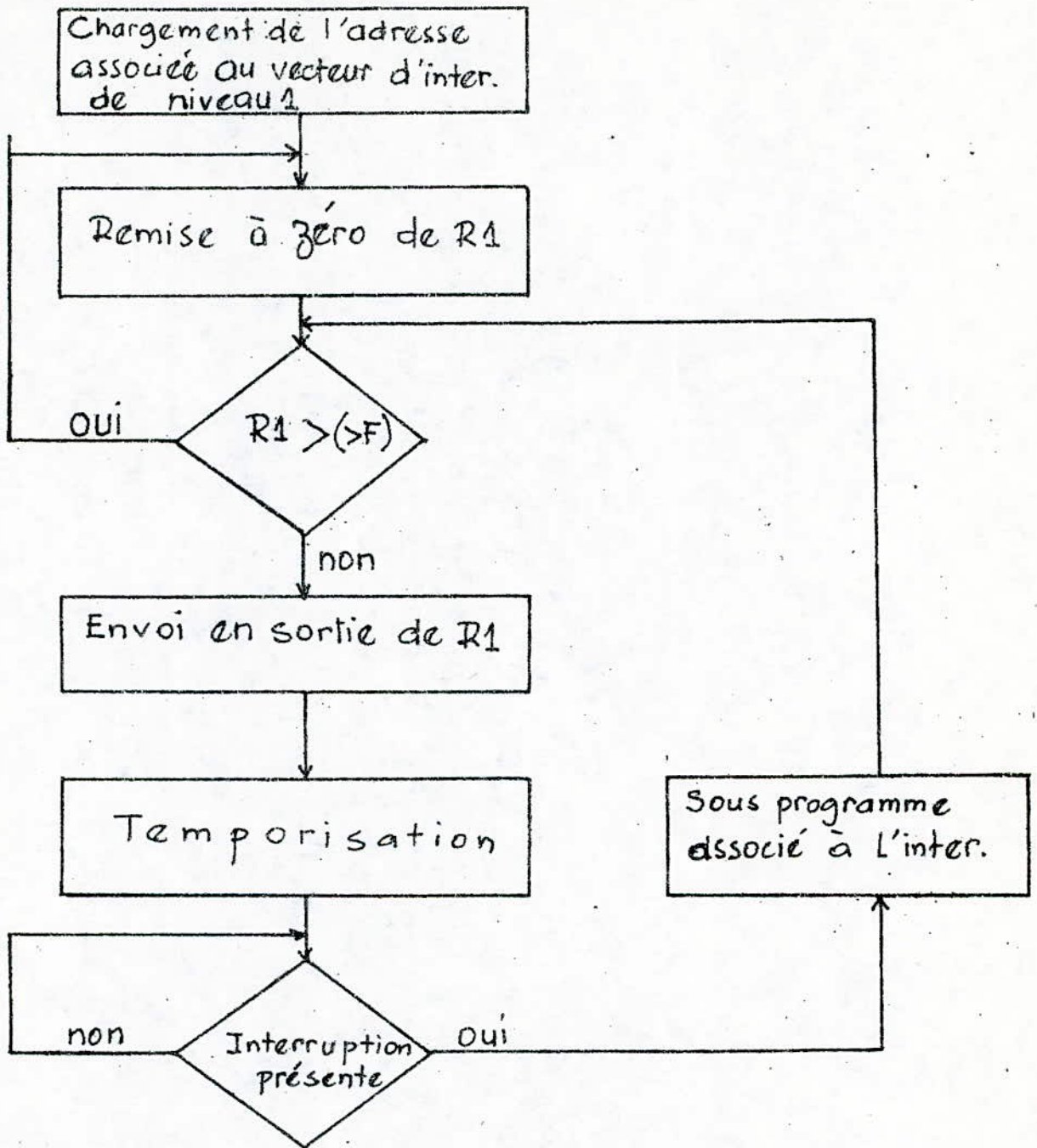
- L'arrêt du comptage.
- Le déclenchement des deux sections, principale et timing.

RESULTAT: (voir figure 33)

\* Sur l'affichage relatif à la section principale;

-On lit la valeur atteinte par le compteur. (387 uS)

ORGANIGRAMME



- FIG:30 -

PROGRAMME

=====

0300 LP	BSS 32	
	LWPI LP	
	LI 0, >300	CHARGEMENT DE L'ADRESSE ASSOCIE AU
	MOV 0, 54	VECTEUR D'INTERRUPTION DE NIVEAU 1
	LI 0, >200	
	MOV 0, 56	
LP2	CLR 1	
LP1	CI 1, >F	
	JGT LP2	
	LI 12, >20	INITIALISATION DE L'ADRESSE DE BASE DU CRU
	LDCR 1, 9	ENVOI EN SORTIE DE R1
	LIMI 1	VALIDATION DE L'INTERRUPTION DE NIVEAU 1
	LI 12, 0	INITIALISATION DE L'ADRESSE DE BASE DU CRU
		(BIT DE CONTROLE)
	LI 2, >F	CHARGEMENT DE R2 PAR LES VALEURS DES BITS
		DE CONTROLE ET D'HORLOGE
	LDCR 2, 15	TRANSFERT DE 15 BITS DE R2 DANS LE CRU
	SBZ 0	SELECTION DU MODE INTERRUPTIBLE
	SBO 3	VALIDATION DE L'INTERRUPTION D'HORLOGE INT.3
	IDLE	ATTENTE DE L'INTERRUPTION D'HORLOGE
	JMP LP1	
0358	END	

SOUS PROGRAMME ASSOCIE A L'INTERRUPTION

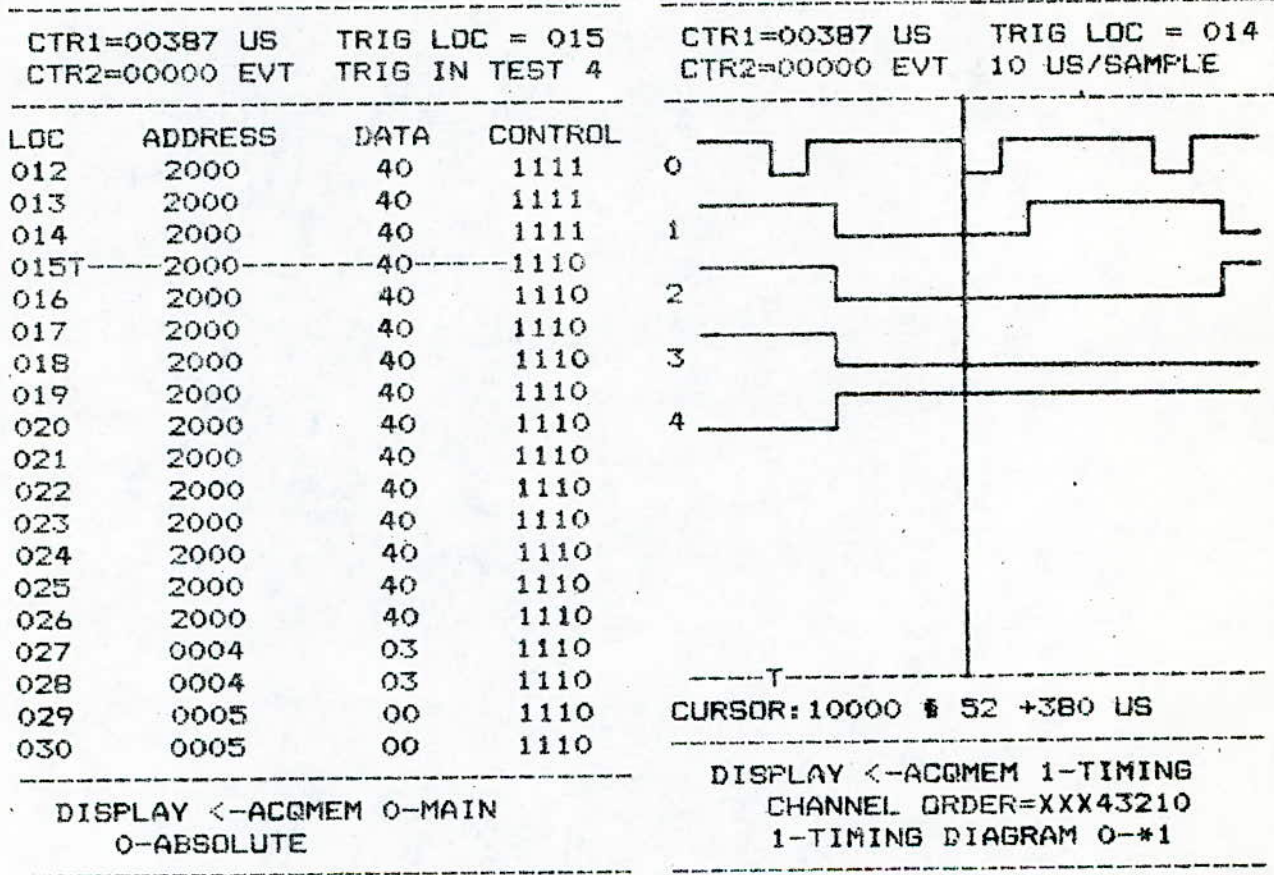
0200	LIMI 0	INHIBITION DES INTERRUPTIONS
	INC 1	
	SBO 3	
	RTWP	RETOUR AU PROGRAMME PRINCIPAL

```

TEST 1
1IF
1 WORD RECOGNIZER # 1
1 DATA=XX
1 ADDRESS=XXXX
1 C0=1    C1=X    C2=X    C3=X
1 C4=X    C5=X    EXT TRIG IN=X
1THEN DO
1 GOTO 2
    END TEST 1
TEST 2
2IF
2 WORD RECOGNIZER # 2
2 DATA=XX
2 ADDRESS=XXXX
2 C0=0    C1=X    C2=X    C3=X
2 C4=X    C5=X    EXT TRIG IN=X
2THEN DO
2
2 COUNTER # 1 1-US
2     2-RESET AND RUN
2 GOTO 3
2
    END TEST 2
TEST 3
3IF
3 WORD RECOGNIZER # 3
3 DATA=XX
3 ADDRESS=XXXX
3 C0=1    C1=X    C2=X    C3=X
3 C4=X    C5=X    EXT TRIG IN=X
3THEN DO
3 GOTO 4
    END TEST 3
TEST 4
4IF
4 WORD RECOGNIZER #4
4 DATA=XX
4 ADDRESS=XXXX
4 C0=0    C1=X    C2=X    C3=X
4 C4=X    C5=X    EXT TRIG IN=X
4THEN DO
4
4 COUNTER # 1 1-US
4     1-STOP
4 TRIGGER 0-MAIN
4     0-BEFORE DATA
4     0-SYSTEM UNDER TEST CONT.
4     0-STANDARD CLOCK QUAL.
4 TRIGGER 1-TIMING
4     0-BEFORE DATA
4     THRESHOLD V. = 0-PLUS 1.40
4     1-ARM ASYNC, TRIG ON WR
4     SAMPLE PERIOD 1 * 3-10 US
4     WORD RECOGNIZER=XXXXXXXX
4     EXT TRIG IN=X
4     GLITCH RECOGNIZER=XXXXXXXX
4     FILTER=000 NS
4
    END TEST 4

```

\*\* RESULTATS \*\*



- FIG:33 -

-On voit le passage de 1 à 0 de la ligne de contrôle Co.  
-On constate le passage au sous programme d'interruption dont les vecteurs associés sont 0004 et 0005.

\* Sur l'affichage de chronogrammes;

-Visualisation du signal d'interruption (sur le canal 0)

-Visualisation des quatre signaux de sortie du CRU.

-Le curseur est positionné sur la seconde interruption qui lui correspond un temps de **380**  $\mu$ S.

REMARQUE: Pour une temporisation maximale de l'horloge, le résultat relevé est, dans ce cas, est  $T=525$  MS .

## CONCLUSION

L'utilité d'un tel appareil, dans les laboratoires, réside dans le fait qu'il apporte des solutions aux problèmes posés par les systèmes digitaux.

Les applications réalisées à l'aide de cet appareil ont montrés sa souplesse d'utilisation et ses capacités à résoudre les problèmes rencontrés dans les systèmes logiques, spécialement, ceux à base de microprocesseurs.

La difficulté principale rencontrée est la sauvegarde d'informations collectées sur le système, ou des programmes développés vu sa faible capacité mémoire et l'absence de ports d'entrée/sortie pour sa connexion avec des périphériques (mémoires de masse, imprimante, ...).

Un développement d'interfaces dans ce but est envisageable.



## ANNEXE I

### LE MODULE PERSONNALISE

La fonction principale du module personnalisé est de collecter les données à partir du système sous test et de les transférer à l'analyseur logique dans un format que ce dernier peut interpréter. C'est donc un interface entre le 7D02 et le "SUT".

La série PM 100 des modules de personnalisation adapte le 7D02 à une large gamme de uP, par une simple connexion du module personnalisé sur le panneau avant du 7D02.

L'utilisateur peut choisir à l'heure actuelle les modules de personnalisation adaptables aux processeurs suivants:

8 bits: 6800, 6802, 8085, Z80

16 bits: 8086, Z8002, 68000

Le module personnalisé d'usage général est le PM 101. IL est disponible pour les systèmes n'utilisant pas de microprocesseurs ou pour les systèmes à base de uP pour lesquels le module personnalisé spécifique n'est pas disponible.

Le module personnalisé fait le désassemblage et génère en plus des informations dont l'analyseur en a besoin.

#### Description du PM 102/PM 103

Ils sont utilisés pour les systèmes à base de 6800/6802.

Du point de vu HARD, ils se présentent sous forme:

- d'un boîtier renfermant un circuit.
- D'un câble ruban terminé par un connecteur .
- D'une paire torsos sous forme de toile de câble terminée par un connecteur qui s'enfiche à la place du uP.

Le boîtier contient:

- Un interface qui personnalise l'analyseur logique pour qu'il travaille avec les uP 6800/6802.
- Un support (ZIF) "ZERO-INSERTION-FORCE" sur le boîtier pour les uP.
- Un microprogramme qui permet à l'analyseur de désassembler l'information qu'il reçoit en code mémorique du 6800/6802.
- Un circuit pour générer l'horloge d'état (state clock) et d'autres entrées vers l'analyseur logique.

## ANNEXE II

### LE MODE IMMEDIAT

L'appui sur la touche "IMMEDIATE" met le 7D02 dans le mode immédiat, ainsi il sera prêt à exécuter une commande immédiatement.

Les commandes utilisées dans ce mode sont partagées en deux catégories : Transfert de données et la commande GOTO

IMMEDIATE DISPLAY ACQMEM.

IMMEDIATE DISPLAY PROGRAM.

IMMEDIATE DISPLAY STOREMEM.

IMMEDIATE PROGRAM STOREMEM.

IMMEDIATE STOREMEM PROGRAM.

IMMEDIATE STOREMEM ACQMEM.

IMMEDIATE GOTO (n)

A l'exception de la commande "GOTO", chaque paire (ex: DISPLAY ACQMEM) indique la destination suivie de la source .

Les trois premières séquences si-dessus sont exécutées une fois que la source est définie.

Les trois séquences suivantes ne sont exécutées que si le curseur est placé sur le champ "EXECUTE", qui apparaît à la fin de la commande. IL est de même pour la dernière instruction, où "n" spécifie le numéro du test.

### LE MODE FORMAT

Le mode FORMAT est sélectionné en appuyant sur la touche "FORMAT", lorsque le 7D02 n'est pas entrainé d'exécuter un programme.

Ce mode affiche des menus permettant à l'opérateur de changer la forme des données qui seront introduites ou affichées.

Par exemple l'opérateur peut:

### ANNEXE III

-Changer les valeurs des champs relatifs au "WORD RECOGNIZER" l'affichage des données acquises etc...

-valider ou inhiber la brillance des différences entre la mémoire de stockage et la mémoire d'acquisition dans le mode d'affichage.

-valider ou inhiber l'affichage de l'information parasite acquise sur l'affichage des données de l'option timing.

-Inverser les voies de données de l'option timing et/ou les bus du "SUT"(si le PMIOI est utilisé).

L'inversion des voies signifie que les valeurs du reconnaiseur de mot introduites seront complémentées avant qu'elles soient utilisées et les données acquises par la voie seront inversées avant qu'elles soient affichées.

Les contenus du mode FORMAT dépendent du module personnalisé et de la présence ou l'absence de l'option timing.

... exemple d'affichage avec le PM IO3 relatif au 6802 avec la présence de l'option timing.

```
TIMING OPTION WORD RECOGNIZER
      0-BINARY
WORD RECOGNIZER ADDRESS FIELD
      2-HEX
WORD RECOGNIZER DATA FIELD
      2-HEX
TIMING OPTION DATA DISPLAY
      0-BINARY
ADDRESS FIELD DISPLAY
      2-HEX
DATA FIELD DISPLAY
      2-HEX
HIGHLIGHT MEMORY DIFFERENCES
      1-NO
DISPLAY GLITCHES
      0-YES
TIMING OPTION DATA INVERSION
      DATA=00000000
```

## BIBLIOGRAPHIE

- 7D02 LOGIC ANALYZER (SERVICE MANUAL) TEKTRONIX
- 7D02 LOGIC ANALYZER (OPERATORS) TEKTRONIX
- INTRODUCTION AUX MICROPROCESSEURS. AUMIAUX 1982
- MICROPROCESSEURS ET MEMOIRES. EFCIS 1980
- INTRODUCTION AUX MICROPROCESSEURS. TEXAS INSTRUMENTS 1980
- REVUE:  
ELECTRONIQUE ET MICROELECTRONIQUE INDUSTRIELLES.  
N° 217 , MARS 1975 , pages: 16 à 23 .