L'ORIC A NU

L'ORIC A NU

par Fabrice BROCHE

Editions SORACOM Informatique

Avant-propos du scribe

Ce travail de conservation numérique se veut un hommage à cet ouvrage fabuleux conçu, écrit, et mis en page par Fabrice Broche et édité en juillet 1986 par les éditions SORACOM Informatique.

L'Oric A Nu, a constitué et constitue encore un des meilleurs ouvrages de référence sur la conception matérielle et logicielle des ordinateurs Oric 1 et Oric Atmos. Il inclut de nombreuses informations et schémas sur le hardware, ainsi qu'un listing complet et commenté du désassemblage de la ROM, complété par des informations très détaillées concernant les principes ayant gouverné la conception et l'implémentation des diverses familles de routines, que ce soit celles liées à l'interpréteur BASIC, aux routines graphiques et sonores, à la gestion des entrées-sorties cassette ou aux routines système.

Il existe de nombreuses versions scannées de cet ouvrage, malheureusement il n'existait pas encore de version entièrement numérisée qui permette la recherche plein texte et c'est donc pour pallier ce manque que j'ai entrepris de recopier manuellement une à une toutes les pages de ce livre en essayant de reproduire également toutes les illustrations présentes, au lieu de céder à la facilité (toute relative!) d'emploi d'un logiciel d'OCR.

Ce livre est donc dédié avant tout à l'auteur ainsi qu'à l'éditeur, en espérant qu'ils apprécieront ce travail à sa juste valeur et qu'ils le recevront comme un hommage à cette œuvre qui a éduqué et fait rêver toute une génération d'oriciens dans les années 1980, et qui continue d'être un ouvrage de référence pour la communauté Oric du 21e siècle. C'est donc également un tribut à la communauté oricienne, et notamment pour les membres du CEO (Club Europe Oric).

Oricquement vôtre,

Laurent

*Notes :*

* *la mise en page de ce manuscrit a été conçue de manière à respecter scrupuleusement les numéros de page de l'ouvrage original, facilitant ainsi la recherche.*
* *ceci étant, je me suis permis ça et là quelques corrections de texte ou de style, qui sont évidemment quelque peu arbitraires, mais en prenant garde évidemment à respecter scrupuleusement le sens original du texte et le style de l'auteur.*
* *j'ai essayé de rassembler et d'appliquer l'ensemble des corrections connues concernant les erreurs ou fautes de frappe qui s'étaient glissées dans l'ouvrage original.*
* *enfin, j'ai ajouté quelques informations complémentaires (signalées comme telles par des encarts ou une typographie spécifique) là où j'ai jugé qu'il était opportun d'apporter des compléments susceptibles d'enrichir ou d'éclaircir un point particulier.*

*page blanche - la partie "INTRODUCTION" doit commencer page* ***5****.*

# INTRODUCTION

## INTRODUCTION

### But de l'ouvrage

Ce livre est unique en son genre : il se propose d'aller au plus profond de la structure logicielle d'un ordinateur, en commentant toute la mémoire morte, la ROM BASIC.

Ce commentaire se situe à deux niveaux :

* Commentaire du code machine, ligne par ligne, et ce pour les 16 Ko de la ROM. Chaque routine est expliquée, pour rendre sa compréhension et son utilisation accessibles à tous. Les routines les plus compliquées sont accompagnées d'organigrammes.
* Chapitres généraux, faisant le point sur un thème précis : la gestion des entrées/sorties, le fonctionnement d'un interpréteur, etc.

Et c'est là la nouveauté par rapport aux ouvrages du genre, quel que soit l'ordinateur considéré. Il existait jusqu'à présent deux types d'ouvrages : ceux qui donnaient un listing commenté de la ROM, sans autre explication, ce qui n'est utile que pour celui qui connaît déjà le principe des routines employées, ou ceux qui sont une sorte de recueil de connaissances sur une machine, qui ne sont dès lors pas exhaustifs, et décrivent plus l'utilisation que le fonctionnement de la machine.

Cet ouvrage fait donc la synthèse de tout ce que l'on peut savoir sur l'Oric.

Sa lecture ne requiert que peu de connaissances particulières : il suffit de connaître le jeu d'instructions du 6502, et d'avoir le goût de comprendre ce qu'il se passe.

A ce sujet, il n'est de meilleur apprentissage du langage assembleur que d'analyser des programmes écrits très correctement : l'effet des mnémoniques s'assimile tout seul au fur et à mesure de la lecture de la ROM.

Une dernière précision : ce livre s'adresse surtout aux possesseurs d'Oric 1 et d'Atmos, mais il intéressera également tous les utilisateurs de Basic Microsoft : les routines sont exactement les mêmes, à l'emplacement près. Les possesseurs de Commodore 64, Apple II, Vic 20, etc., sont donc concernés.

### Comment lire ce livre

La lecture peut se faire à deux niveaux :

- Niveau consultation : on veut juste savoir comment utiliser telle ou telle routine, ou savoir s'il existe une routine qui… Il suffit de se reporter au répertoire des routines, classées par thèmes, et de consulter l'en-tête des routines, qui indique les paramètres en entrée, et les résultats retournés.

- Niveau approfondissement des connaissances : il s'agit cette fois d'une lecture attentive des routines, du principe de fonctionnement, etc.

Il est dans tous les cas conseillé de parcourir l'ensemble du livre, pour savoir où chercher le renseignement souhaité.

L'ensemble des commentaires est conçu de telle sorte que le lecteur qui a lu attentivement tout l'ouvrage doit être capable d'écrire un BASIC.

A part les quelques notices techniques dont la liste est donnée en annexe, tous les renseignements et commentaires sont le fruit de recherches personnelles. Tous ce qui est avancé dans cet ouvrage a été scrupuleusement vérifié, mais des erreurs peuvent bien entendu subsister.

## CONVENTIONS ET NOTATIONS

Pour éviter d'alourdir le commentaire, un certain nombre de conventions ont été adoptées.

### Structure du commentaire

Le début d'une routine est repéré par un titre centré et en majuscules.

Des séparations logiques à l'intérieur d'une routine sont repérées par un titre non centré et en minuscules.

Le titre décrit le plus précisément possible l'action globale de la routine.

Un titre comprend parfois des mots entre parenthèses :

Par exemple, « (COMMANDE) » : il s'agit de la routine d'exécution d'une commande du BASIC.

Ex : 'RUN' (COMMANDE)

De même, (FONCTION) et (OPERATEUR) indiquent l'exécution d'une fonction ou d'un opérateur.

Les commentaires généraux sur la routine sont repérés à l'aide de noms génériques :

Entrée : désigne les paramètres particuliers qui doivent être positionnés correctement pour avoir les résultats escomptés.

Sortie : désigne les paramètres retournés par la routine, ainsi que quelques renseignements indirectement liés, mais utilisés dasn la ROM, ou utilisables du type registre inchangé, indicateur positionné, etc.

Programmation : des remarques sur les astuces employées, ou sur les manques d'optimisation. Ce genre de remarque est surtout fait dans le corps du commentaire.

Principe : explique le principe général de la routine, éventuellement aidé par des organigrammes ou divers schémas.

Bogue : signale les erreurs de programmation qui font que la routine ne donne jamais (ou pas toujours) le résultat escompté. Elles sont plus nombreuses qu'on ne le croit !

D'autres noms génériques sont parfois employés, ils parlent alors d'eux-mêmes et il est inutile de s'étendre sur leur signification.

La structure du listing est peu classique, mais très pratique : il présente le code deux ROMs côte-à-côte, sachant que bien souvent, les routines sont les mêmes, mais simplement décalées. Là où les ROMs diffèrent, des points de suspension, remplacent le listing (en aucun cas la linéarité du listing n'est rompue).

Le premier listing est toujours celui de la ROM V1.0 , et le deuxième celui de la ROM V1.1.

Les mnémoniques sont bien entendu standards. Toutefois, les modes d'adressage ont une syntaxe particulière, adoptée pour réduire la place du listing et donc privilégier le commentaire.

Voici les conventions utilisées pour les modes d'adressage :

|  |  |
| --- | --- |
| LSR A | Adressage accumulateur |
| LDA #12 | Immédiat : charge A avec la valeur hexadécimale 12 |
| LDA #'2' | Immédiat : charge A avec le code ASCII du caractère '2', soit #32 |
| LDA #&END | Immédiat : charge A avec le token du mot-clé indiqué, soit ici #80 |
| STX 0230 | Absolu (adresse en hexadécimal) |
| ADC 91 | Page 0 (adresse en hexadécimal) |
| TXA | Implicite |
| LDA (12, X) | Pré-indexé indirect (jamais utilisé dans la ROM) |
| LDA (E9), Y | Post-indexé indirect (argument en hexadécimal) |
| STA 40, X | Page 0 indexé |
| LDY 0235, X | Absolu indexé |
| BNE C570 | Relatif |
| JMP $C000 | Saut |
| JMP (0091) | Indirect |

Un mnémonique spécial est utilisé : BYT, qui indique l'entrée d'octets,

d'adresses (stockées poids faible d'abord) ou de chaînes de caractères.

Ex : C3AA C3A6 BYT ' ERROR'

Un commentaire revient souvent en regard des branchements relatifs : **Inconditionnel**. Il signifie que compte-tenu de la position des indicateurs concernés, le branchement sera toujours effectué.

Une dernière précision : toutes les tables, ainsi évidemment que le listing, ont été générés par programme et non recopiées, ce qui devrait assurer un listing exempt d'erreur.

### Abréviations

Tous les registres et indicateurs du 6502 sont reprérés directement par leur lettre en majuscule :

|  |  |
| --- | --- |
| **Registres du 6502 :**  A= Accumulateur  X = registre d'index X  Y = registre d'index Y  P = registre d'état  S = pointeur de pile | **Indicateurs du registre d'état P :**  Z = indicateur de zéro  C = indicateur de retenue  N = indicateur de signe  V = indicateur de dépassement (overflow)  I = drapeau d'autorisation d'interruption IRQ  B = indicateur de BRK  D = drapeau de mode décimal |

Un octet est représenté sous la forme : *b7 b6 b5 b4 b3 b2 b1 b0*. Ainsi, *b3* signifie que l'on fait référence au bit 3 de l'octet.

Les accumulateurs flottants sont désignés par ACC1, ACC2, ACC3, ACC4 et ACC5 :

ACC1 : #DO-#D5 (accumulateur principal)

ACC2 : #D8-#DD

ACC1 : #95-#9B

ACC1 : #C6-#CA

ACC1 : #CB-#CF

De plus, les accumulateurs, après arrondi, sont notés AACC1, AACC2, etc.

Les valeurs contenues par les registres sont notées avec le registre poids faible d'abord.

Ex : YA --> ACC1 veut dire *placer le nombre dont le poids faible est dans Y et le poids fort dans A dans l'accumulateur principal*.

Lorsque des pointeurs mémoire sont utilisés, les poids fort et faible sont séparés par un tiret ,

Ex : #9A-#9B, #200-#201

Comme d'habitude, les parenthèses désignent une indirection : *le contenu de l'adresse pointée par...*

Ex : si A vaut #04 et Y vaut #C0, AY représente #C004, YA désigne #04C0, et (AY) signifie le contenu de la case mémoire #C004.

Ce procédé a été étendu un peu abusivement pour les calculs en flottant : (AY) désign les 5 octets débutant à l'adresse AY.

Ex : (AY) --> ACC1. Si A vaut #12 et Y vaut #45, cela signifie : transférer le nombre codé en virgule flottante dans la zone mémoire #4512-{4516 dans l'accumulateur flottant principal.

Ne disposant pas des labels standards, il n'était pas concevable de se lancer dans l'invention, et de donner des noms aux routines. Lorsque dans le texte il est fait référence à des routines ou à des pointeurs, ils sont repérés par leurs adresses. Dans le cas d'une routine d'adresse différente pour les 2 ROMs, les 2 adresses sont précisées, séparées par un '/'.

Ex : #F276 / #F30A.

Une seule exception : le pointeur de texte BASIC est noté TXTPTR dans les commentaires (mais pas dans le listing). TXTPTR = #00E9.

***Page 10 blanche***

CHAPITRE II  
ORGANISATION MATERIELLE

## PRESENTATION GENERALE

### Généralités

L'ORIC est architecturé autour d'une unité centrale (CPU pour "Central Processing Unit") 6502 A tournant à 1 MHz.

La structure est classique: CPU, ROM, RAM, circuits d'E/S.

Un micro-ordinateur est un curieux mélange de composants très puissants qui comportent des dizaines de milliers de portes logiques (CPU 6502, RAM, etc.), et de composants très peu puissants comportant 4 ou 8 portes logiques seulement, sans compter les éléments discrets.

En effet, il faut relier tous ces composants principaux entre eux.

On peut alors distinguer (assez artificiellement) les Bus (Bus de données qui véhiculent les informations, et Bus d'adresse qui indiquent la destination ou la source de l'information), et les lignes de contrôle.

Les bus sont simples à mettre en œuvre. Toutefois, il est souvent nécessaire de les amplifier pour éviter une trop grande déperdition dans les boitiers. Les bus de l'Oric ne sont pas amplifiés, ce qui est gênant lorsqu'on branche des extensions.

Les lignes d'extension regroupent tout ce qui a trait à la validation des données présentes sur le Bus (c'est le cas de l'horloge) ou à la validation des différents boîtiers susceptibles d'être adressés.

Ainsi, la plupart des composants ont une ou plusieurs lignes 'CS' (abréviation de *Chip Select*, sélection du boîtier). Etant donné la manière dont le 6502 gère son espace adressable, chque boîtier doit être sélectionné pour une zone mémoire particulière, et il faut bien sûr éviter que deux boîtiers soient sélectionnés en même temps.

Le plus gros travail consiste à effectuer un décodage sur le bus d'adresse: sélectionner la ROM uniquement entre #C000 et #FFFF, ou le VIA entre #300 et #3FF. Un tel décodage est très facile à faire, mais nécessite un nombre important de portes logiques.

Il faut aussi s'occuper de générer une image vidéo, c'est-à-dire lire le contenu de l'écran en mémoire et transformer ces informations en un signal couleur. Il faut enfin 'rafraîchir' la RAM, qui est de type dynamique

(on peut schématiser une cellule RAM par une capacité chargée pour un 1, et déchargée pour un 0), pour éviter de perdre so contenu.

Toutes ces opérations sont relativement simples mais nécessitent un très grand nombre de composants.

C'est pourquoi les concepteurs ont choisi la technologie de l'ULA (*Uncommitted Logic Array*, réseau logique programmable), qui est un circuit comportant un grand nombre de portes logiques, que le concepteur organise selon ses besoins. Il y a quelques années, une centaine de circuits auraient été nécessaires pour réaliser le travail de l'ULA (ne pas confondre avec l'UAL, l'Unité Arithmétique et Logique, ALU en anglais).

Aucune documentation n'a filtré, hélas, sur l'ULA. On en est donc réduit à son brochage et à beaucoup de suppositions. On sait toutefois globalement son action:

* + Lire la mémoire vidéo et générer les signaux RVB et synchro.
  + Décoder le bus d'adresse pour générer les signaux de validation de la ROM, du VIA et de la RAM.
  + Faire une lecture "bidon" de la RAM pour la rafraîchir. Etant donné l'organisation de la RAM, il est possible que le rafraîchissement vidéo suffise à rafraîchir la RAM. Sous toutes réserves.

Le 6502 n'utilisant pas le bus tout le temps, l'ULA utilise les temps morts pour travailler, ce qui donne des timings très serrés (cf. le timing du signal MAP par exemple). Elle dispose pour cela d'une horloge à 12 MHz, à partir de laquelle elle génère un signal d'horloge à 1 MHz pour le 6502, le VIA, etc.

### Synoptique

Voici le schéma de principe de l'Oric:

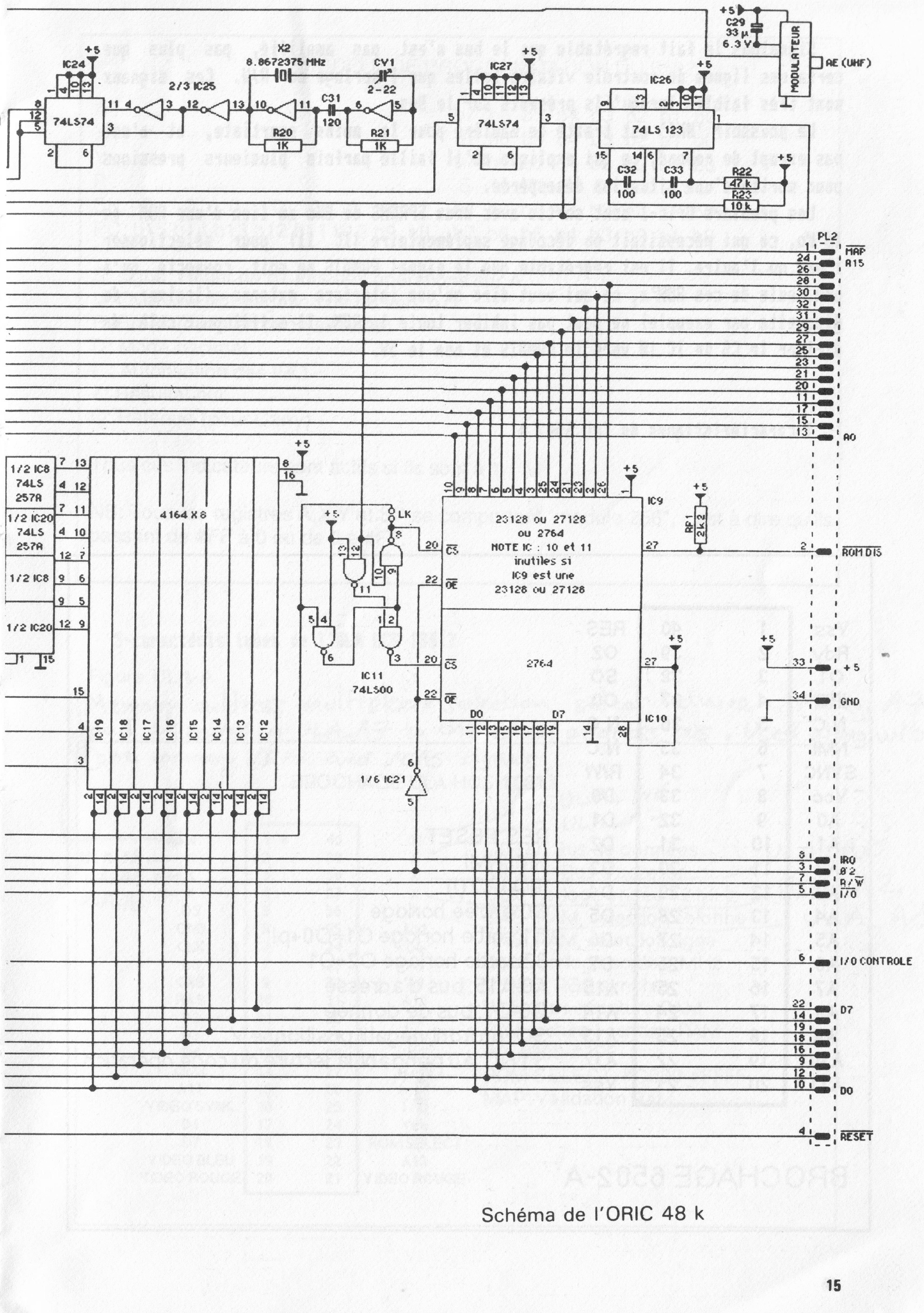
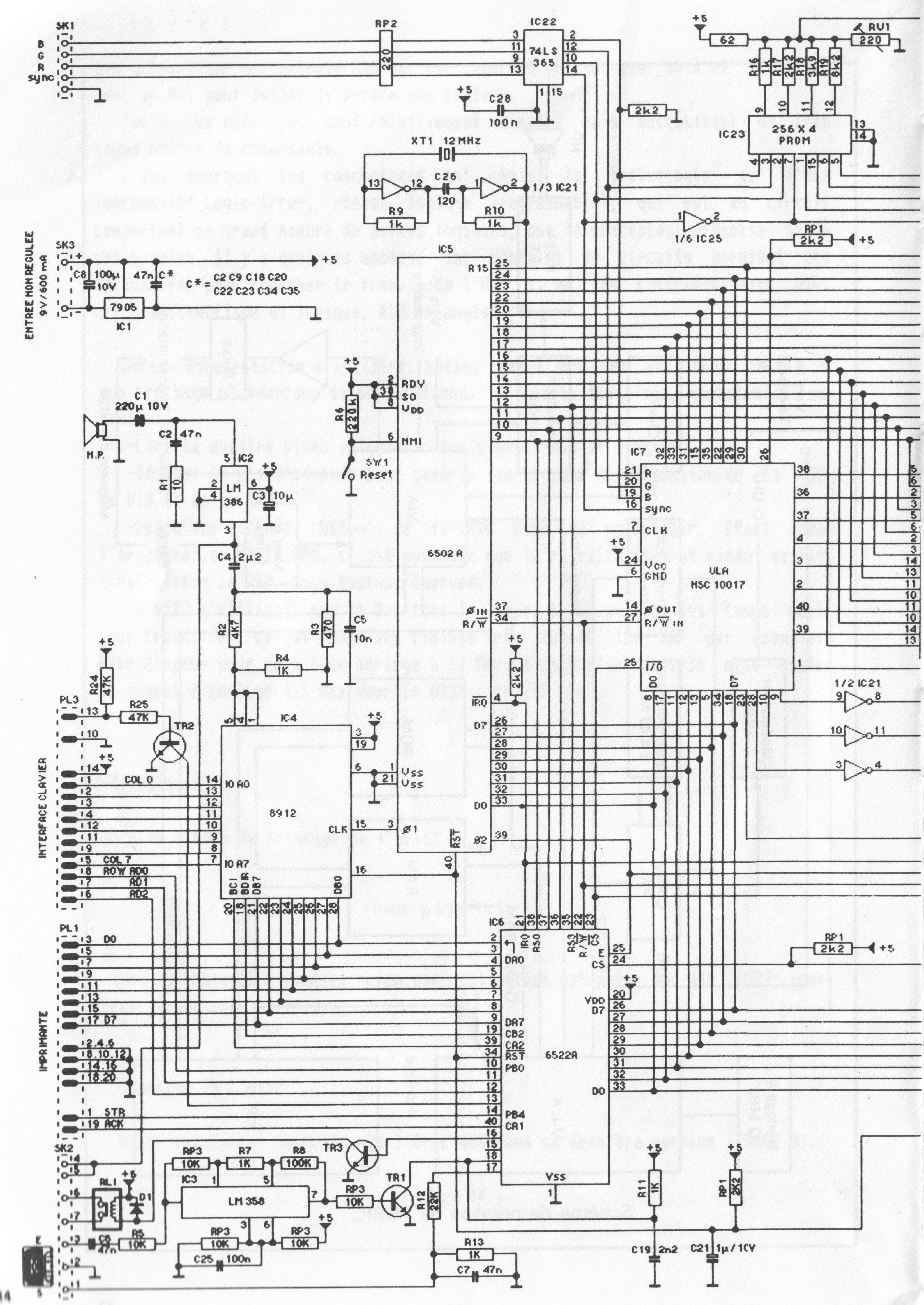
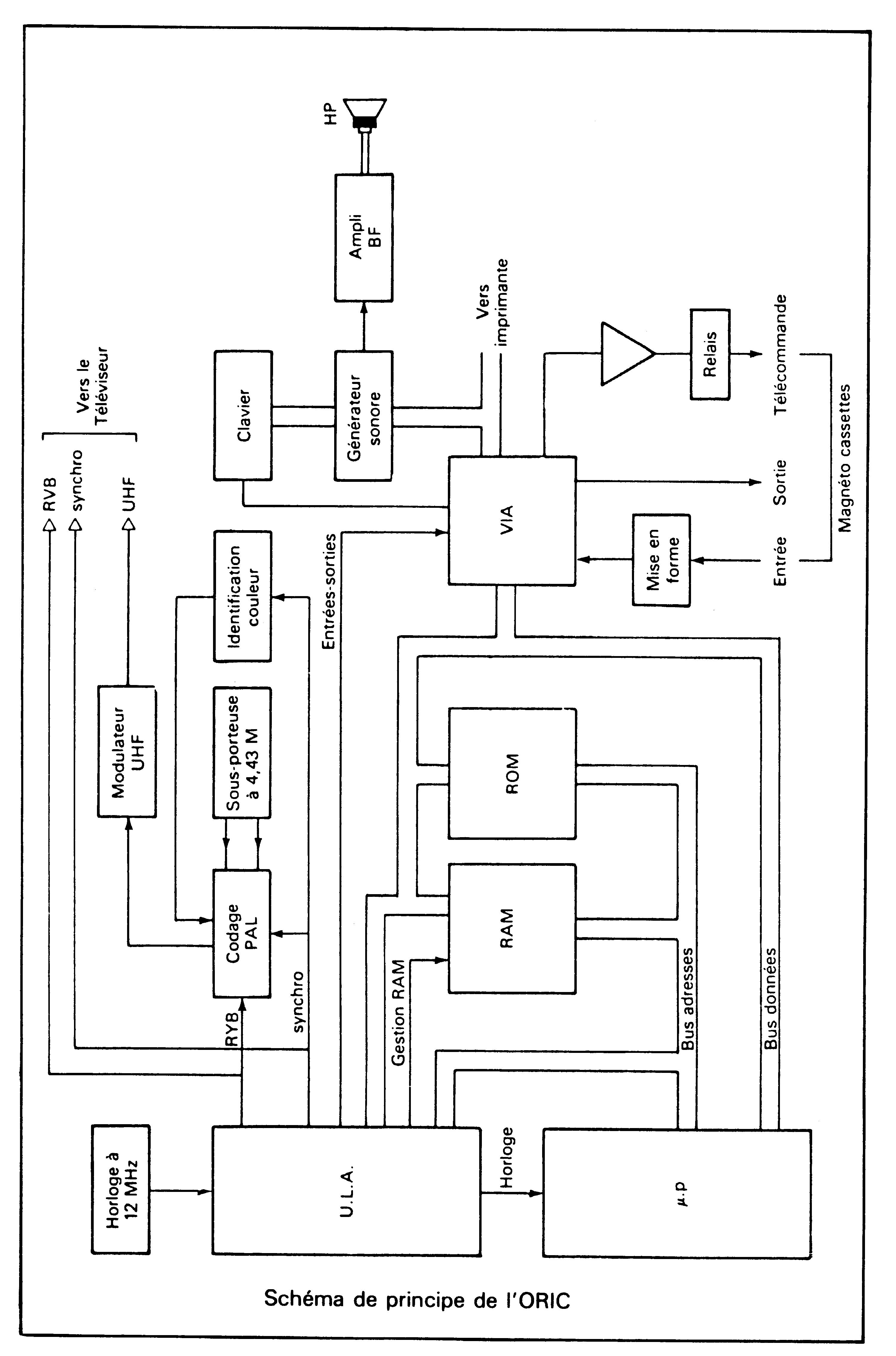
*(voir page 13 ci-contre)*

Peu de choses à signaler, sinon une utilisation optimale du VIA 6522 pour gérer les entrées/sorties.

### Schéma de l'Oric

Voici maintenant le schéma de l'Oric dans sa dernière version (ISSUE 4).

*(voir page 14)*



Signalons le fait regrettable que le bus n'est pas amplifié, pas plus que certaines lignes de contrôle vitales telles que l'horloge ou R/W. Ces signaux sont très faibles lorsqu'ils sont présents sur le bus.

Le poussoir "NMI" est traité de manière pour le moins spartiate, et n'est pas exempt de rebond, ce qui explique qu'il faille parfois plusieurs pressions pour sortir d'une situation désespérée.

Les premiers Oric-1 sont sortis avec deux EPROMs de 8 Ko au lieu d'une ROM de 16 Ko, ce qui nécessitait un décodage supplémentaire (IC 11) pour sélectionner l'une ou l'autre. Il est regrettable que le signal ROMDIS ne soit connecté qu'à une seule de ces ROMs, ce qui veut dire qu'une interface externe (lecteur de disquette par exemple) ne peut pas inhiber toute la ROM. Il suffit pour remédier à cela de diriger le CS de IC 10 vers le ROMDIS et non vers le 5V.

### Caractéristiques du CPU 6502 A