. Le présent manuel technique s'adresse aux spécialistes qui souhaitent obtenir le maximum d'informations techniques sur le micro-ordinateur GOUPIL 2 tant au niveau du matériel que du logiciel de base pour la version de base ou les versions évoluées.

. A ce niveau, la politique de SMT est de pratiquer l'ouverture la plus large et de faire profiter son réseau de distributeurs et ses clients de toute l'information technique souhaitée.

Attention ! En dehors des listings, ce document a été rédigé selon la convention française pour les programmes informatiques :

Ø signifie la lettre O
O signifie le chiffre zéro

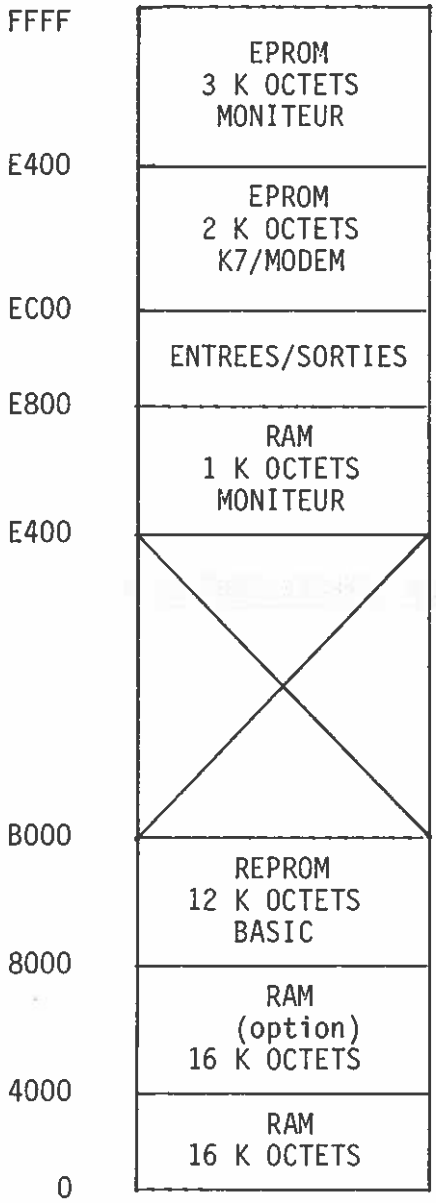
Cela peut être l'inverse dans d'autres documents, sur d'autres micro-ordinateurs ou sur l'écran de votre GOUPIL 2.

S O M M A I R E

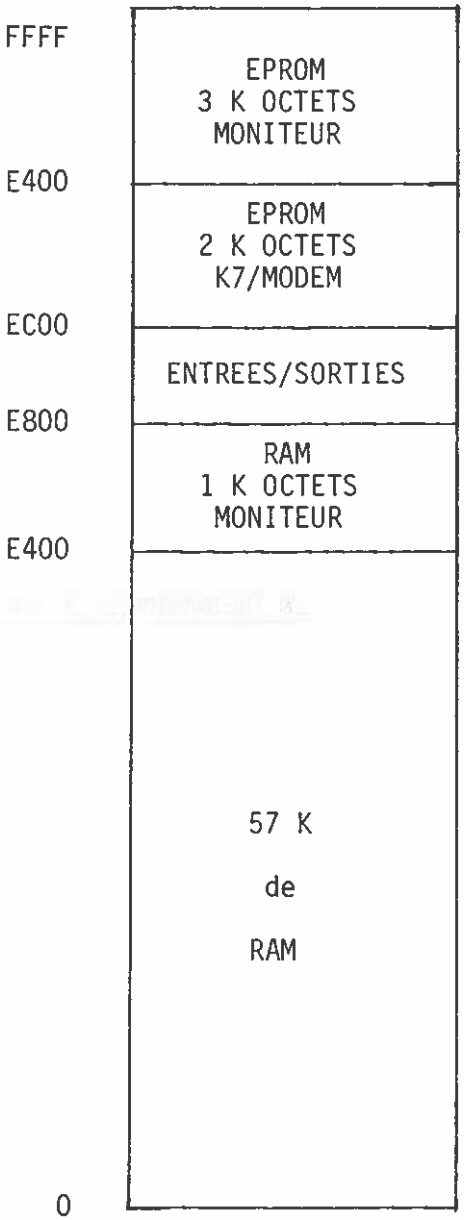
- I -

- LA GEOGRAPHIE DE LA MEMOIRE ET DES ENTREES-SORTIES -

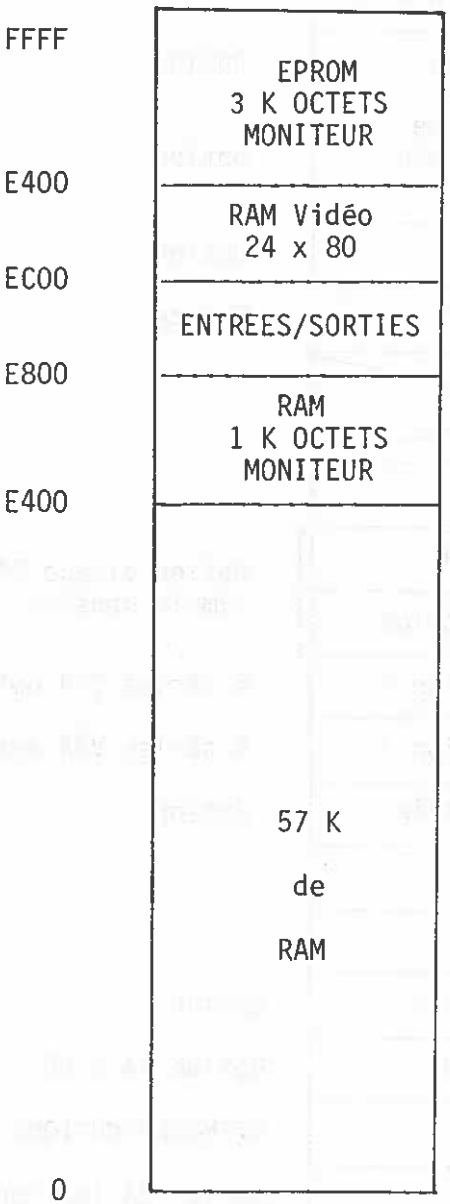
VERSION DE BASE

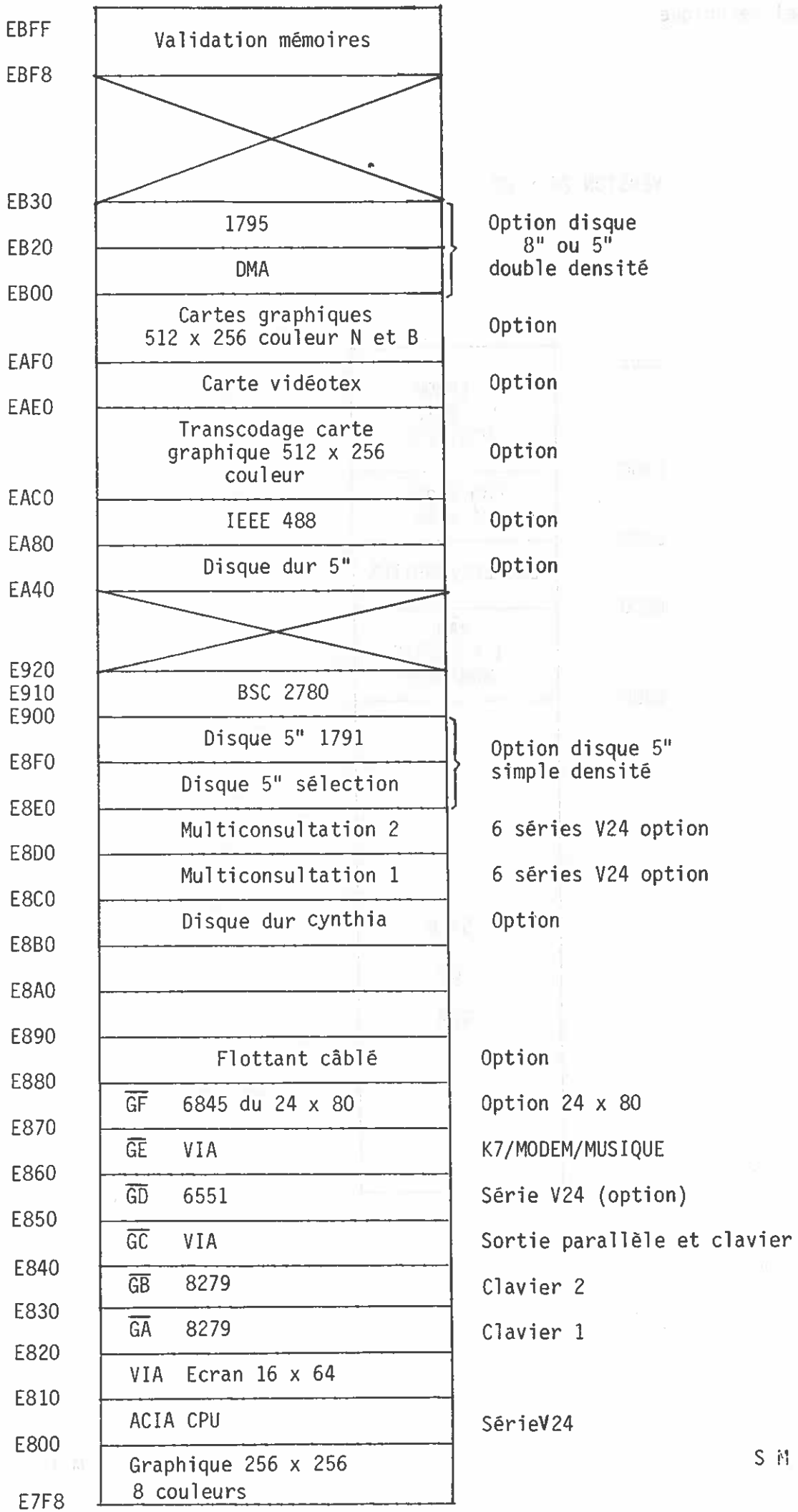


VERSION 16 x 64



VERSION 24 x 80




EBFF
Validation mémoires
EBF8
EB30
1795
EB20
DMA
EB00
Option disque 8" ou 5" double densité
Cartes graphiques 512 x 256 couleur N et B
Option
EAF0
Carte vidéotex
Option
EAE0
Transcodage carte graphique 512 x 256 couleur
Option
EAC0
IEEE 488
Option
EA80
Disque dur 5"
Option
EA40
E920
E910
BSC 2780
E900
Disque 5" 1791
E8F0
Disque 5" sélection
Option disque 5" simple densité
E8E0
Multiconsultation 2
6 séries V24 option
E8D0
Multiconsultation 1
6 séries V24 option
E8C0
Disque dur cynthia
Option
E8B0
E8A0
E890
Flottant câblé
Option
E880
GF 6845 du 24 x 80
Option 24 x 80
E870
GE VIA
K7/MODEM/MUSIQUE
E860
GD 6551
Série V24 (option)
E850
GC VIA
Sortie parallèle et clavier
E840
GB 8279
Clavier 2
E830
GA 8279
Clavier 1
E820
VIA Ecran 16 x 64
E810
ACIA CPU
SérieV24
E800
Graphique 256 x 256 8 couleurs
E7F8

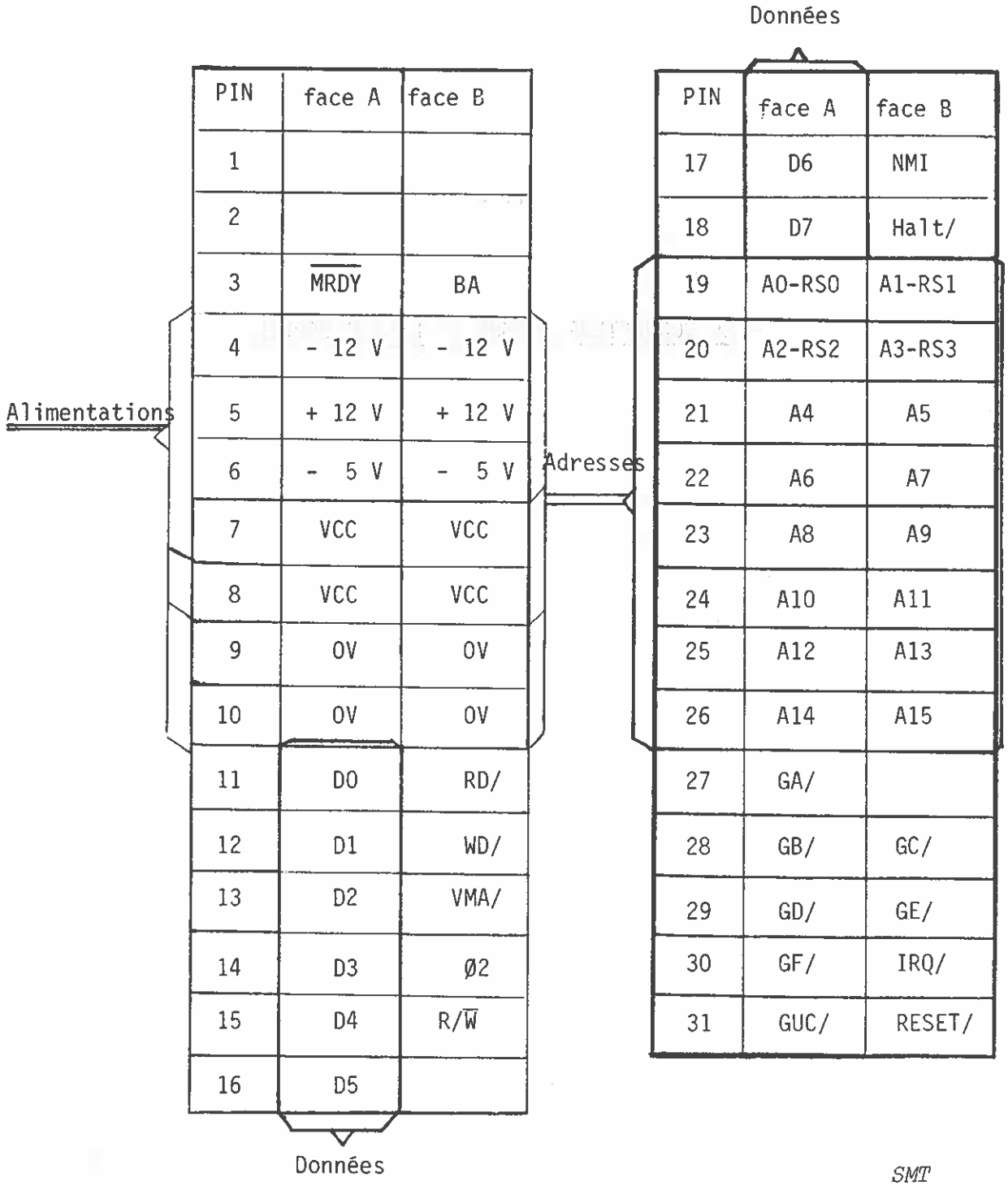
- II -

- LE CONNECTEUR COMMUN ET LE BUS GOUPIL -

Le fond de panier, dans la version de base, fournit 12 emplacements dont 9 sont équipés de connecteurs :

4 connecteurs sont utilisés en version de base pour les cartes suivantes :

- Carte CPU
- Carte E/S
- Carte Mémoire
- Carte Modem



PIN	face A	face B
1		
2		
3	$\overline{\text{MRDY}}$	BA
4	- 12 V	- 12 V
5	+ 12 V	+ 12 V
6	- 5 V	- 5 V
7	VCC	VCC
8	VCC	VCC
9	OV	OV
10	OV	OV
11	D0	RD/
12	D1	WD/
13	D2	VMA/
14	D3	Ø2
15	D4	R/$\overline{\text{W}}$
16	D5	

PIN	face A	face B
17	D6	NMI
18	D7	Halt/
19	A0-RS0	A1-RS1
20	A2-RS2	A3-RS3
21	A4	A5
22	A6	A7
23	A8	A9
24	A10	A11
25	A12	A13
26	A14	A15
27	GA/	
28	GB/	GC/
29	GD/	GE/
30	GF/	IRQ/
31	GUC/	RESET/

SCHEMA DE BROCHAGE D'UN CONNECTEUR

SMT

Signification des codes précédents :

VCC	: + 5V
OV	: masse
RD/	: lecture
WD/	: écriture
VMA/	: valid memory address
Φ2	: horloge système (1 MHz)
R/W/	: 1 : lecture ; 0 : écriture
BA	: bus available
NMI/	: non masquable interrupt
HALT/	: arrêt du microprocesseur
GA/ à GF/	: signaux de décodage d'adresse
GUC/	: décodage des adresses E000 à EFFF
RESET/	: initialisation du système
IRQ/	: interrupt request

* Les signaux dont le nom est suivi du / sont actifs à l'état zéro.

- III -

- LE CODE ASCII NORMALISE -

TABLE DE CODIFICATION ASCII

ASCII (HEX)	CAR ou CONTR.	ASCII (HEX)	CAR ou CONTR.	ASCII (HEX)	CAR ou CONTR.	ASCII (HEX)	CAR ou CONTR.
00	NUL	20	SP	40	à (@) **	60	`
01	SØH	21	!	41	A	61	a
02	STX	22	"	42	B	62	b
03	ETX	23	#	43	C	63	c
04	EØT	24	$	44	D	64	d
05	ENQ	25	%	45	E	65	e
06	ACK	26	&	46	F	66	f
07	BEL	27	'	47	G	67	g
08	BS	28	(	48	H	68	h
09	HT	29	)	49	I	69	i
0A	LF	2A	*	4A	J	6A	j
0B	VT	2B	+	4B	K	6B	k
0C	FF	2C	,	4C	L	6C	l
0D	CR	2D	-	4D	M	6D	m
0E	SØ	2E	.	4E	N	6E	n
0F	SI	2F	/	4F	Ø	6F	o
10	DLE	30	0	50	P	70	p
11	DC1(X.ØN)	31	1	51	Q	71	q
12	DC2(TAPE)	32	2	52	R	72	r
13	DC3(X.ØFF)	33	3	53	S	73	s
14	DC4	34	4	54	T	74	t
15	NAK	35	5	55	U	75	u
16	SYN	36	6	56	V	76	v
17	ETB	37	7	57	W	77	w
18	CAN	38	8	58	X	78	x
19	EM	39	9	59	Y	79	y
1A	SUB	3A	:	5A	Z	7A	z
1B	ESC	3B	;	5B	[	7B	é ({) **
1C	FS	3C	<	5C	\	7C	ù (!) **
1D	GS	3D	=	5D	]	7D	è (}) **
1E	RS	3E	>	5E	^ (↑) *	7E	(~) *
1F	US	3F	?	5F	_ (←) *	7F	DEL

* autres graphismes normalisés

** graphismes standards dans le cas des minuscules accentuées qui ne sont pas normalisées.

- SYNOPTIQUES DES CARTES ET COMMENTAIRES TECHNIQUES :

INDEX DES SYNOPTIQUES :

I) GENERALITES

1 - Architecture du système

L'architecture du système est liée à 2 impératifs techniques :

- Possibilité d'extension
- Maintenance

A - Structure interne

On trouve dans GOUPIL une "carte BUS" qui permet la connexion des cartes entre elles.

Sur cette carte sont regroupés, les tensions d'alimentation, les signaux du bus de données, les signaux du bus d'adresses, ainsi que les signaux du bus de commandes.

Les connecteurs disposés sur cette carte bus permettent la liaison avec les cartes du système. Douze positions existent équipées de 9 connecteurs dont quatre sont destinés à recevoir les cartes formant la version de base.

a) carte CPU G1 Version 7
b) carte mémoire dynamique G1 Version 3
c) carte E/S G1 Version 3
d) carte Modem G1 Version 3 (liaison 600 bauds entre GOUPIL et K7)
(liaison 300 bauds entre GOUPIL et modem)

Chaque carte utilise tout ou partie des signaux présents sur le bus. Ces signaux sont gérés par la carte CPU.

B - Possibilités d'extension

Les huit emplacements libres sont destinés à recevoir les cartes d'extension afin d'augmenter la puissance du système.

Ces cartes peuvent être par exemple :

- carte contrôleur de floppy-disque (5" et 8")
- carte entrée-sortie
- cartes graphiques (256 x 256 x 8 couleurs, 256 x 512 Noir et blanc)
- carte Modem Asynchrone (liaison à 1200 bauds vers les gros systèmes)
- carte interface disque dur
- carte 24 x 80
- carte liaison synchrone BSC
- carte commande magnétoscope
- carte vidéotex
- carte IEEE
- carte multiconsultation
- carte graphique avec incrustation et lightpen 256 x 512 x 8 couleurs

C - Maintenance

Le fait d'avoir des cartes ayant une fonction précise permet une maintenance plus efficace et moins coûteuse que pour un système mono-carte.

1) La conception modulaire permet la détection de pannes sur un sous-ensemble (carte) à l'aide des programmes de tests appropriés.

2) La carte défectueuse est rapidement détectée et son changement ne prend que quelques instants.

Ces éléments permettent la mise en oeuvre d'une maintenance sur le site qui évite l'arrêt prolongé du système. Le dépannage des cartes s'effectue au laboratoire après remise en service du système de l'utilisateur.

2) ENTREES SORTIES

La puissance du système GOUPIL est liée au fait que l'on peut lui connecter un grand nombre de périphériques.

Dans la version de base, l'utilisateur dispose des éléments suivants :

a) 1 écran-clavier (intégré au système)

b) 1 entrée-sortie pour magnétophone cassette (prise DIN)

c) 1 entrée-sortie modem acoustique (deux oreillettes)

d) 1 entrée-sortie V24 (gestion de périphériques en séries)

e) 1 entrée sortie 8 bits parallèle avec ses signaux de contrôle (option)

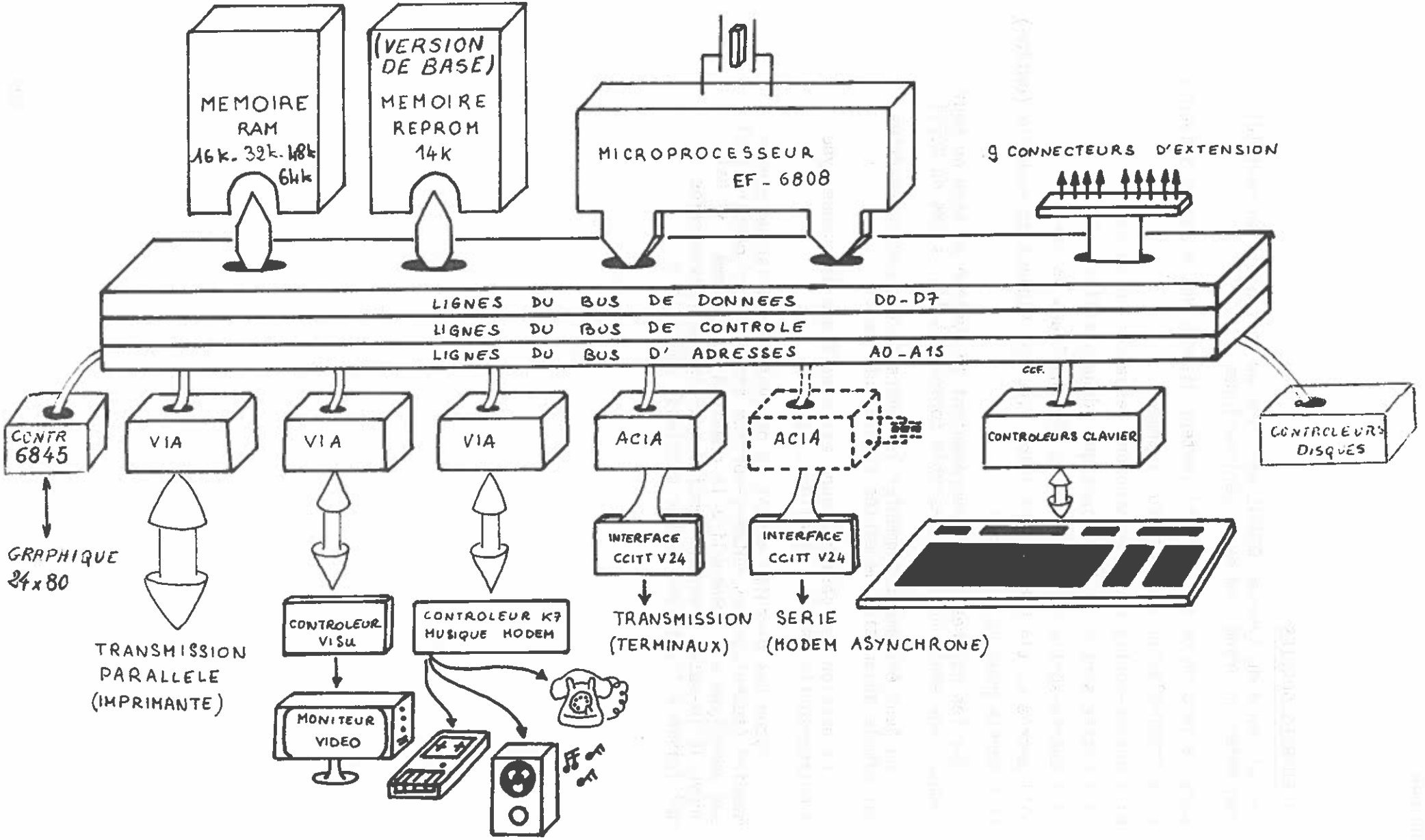
f) 1 sortie pour haut parleur.

Sur les quatres cartes représentant le système de base on peut ajouter également une entrée-sortie conforme à l'avis V24 du CCITT.

On peut également augmenter le nombre de ces entrées-sorties par simple adjonction de cartes supplémentaires.

La gestion des périphériques est spécifique de chaque type d'entrée-sortie (série, parallèle, etc...).

Chacun des circuits gérant la périphérie utilise un espace mémoire (technique des entrées-sorties projetées en mémoires) et son adressage est préétabli à la création du système ; il est prévu 1kilo-octet d'espace mémoire pour les entrées-sorties ce qui laisse à l'utilisateur une souplesse appréciable.



SYNOPTIQUE GENERAL DE GOUPIL

II - Présentation de la carte CPU

La carte d'unité centrale du GOUPIL regroupe en réalité deux sous-ensembles :

- La partie microprocesseur et les circuits qui lui sont nécessaires.

- La partie visualisation concentrée autour du boitier EF 9 364.

1) Partie microprocesseur :

Le coeur de cette partie est un microprocesseur fabriqué par MOTOROLA, référencé EF 6808.
Ce microprocesseur possède les caractéristiques suivantes :

- L'horloge est intégrée au boitier.

- Les instructions sont compatibles avec le MC 6800.

- Il peut adresser jusqu'à 64 K mots.

- Ses entrées-sorties sont compatibles avec la série TTL standard.

- Il travaille sur des mots de 8 bits.

- Il possède des broches d'interruptions.

Le EF 6808 (IC4C) permet de piloter, sur chacune de ses entrées-sorties, une charge TTL, soit 1. 6mA. De plus chaque ligne du bus de données et du bus d'adresse est amplifiée par un circuit tristate (Buffer).

Une autre particularité du EF 6808 est qu'il suffit de brancher aux bornes de l'unité centrale un quartz de 4 MHZ pour générer la fréquence de synchronisation du système $\Phi 2$, l'horloge et un diviseur par 4 étant intégrés dans le boitier.

Les lignes de contrôle du EF 6808, BA (Bus Available), Halt (Halte), IRQ (Interrupt Request), NMI (Non Masquable Interrupt), sont sorties sur le bus par l'intermédiaire du connecteur TB 1, avec une résistance du 4,7 K au +5 V sur chaque ligne (pull-up).

La sortie E (Enable) du EF 6808, qui sert à synchroniser les boitiers périphériques, est passée dans un amplificateur 3 états (buffer tri-state) avant d'être connectée au bus.

Sur certaines cartes, il existe des composants pilotés par des signaux de lecture et d'écriture séparés.
Cette séparation est réalisée par IC 6D synchronisé sur l'horloge Ø2 qui arrive de IC 6C.

De même, le signal VMA (Valid Memory Acc ess), protégé du Reset par IC 6D, est envoyé sur le bus.

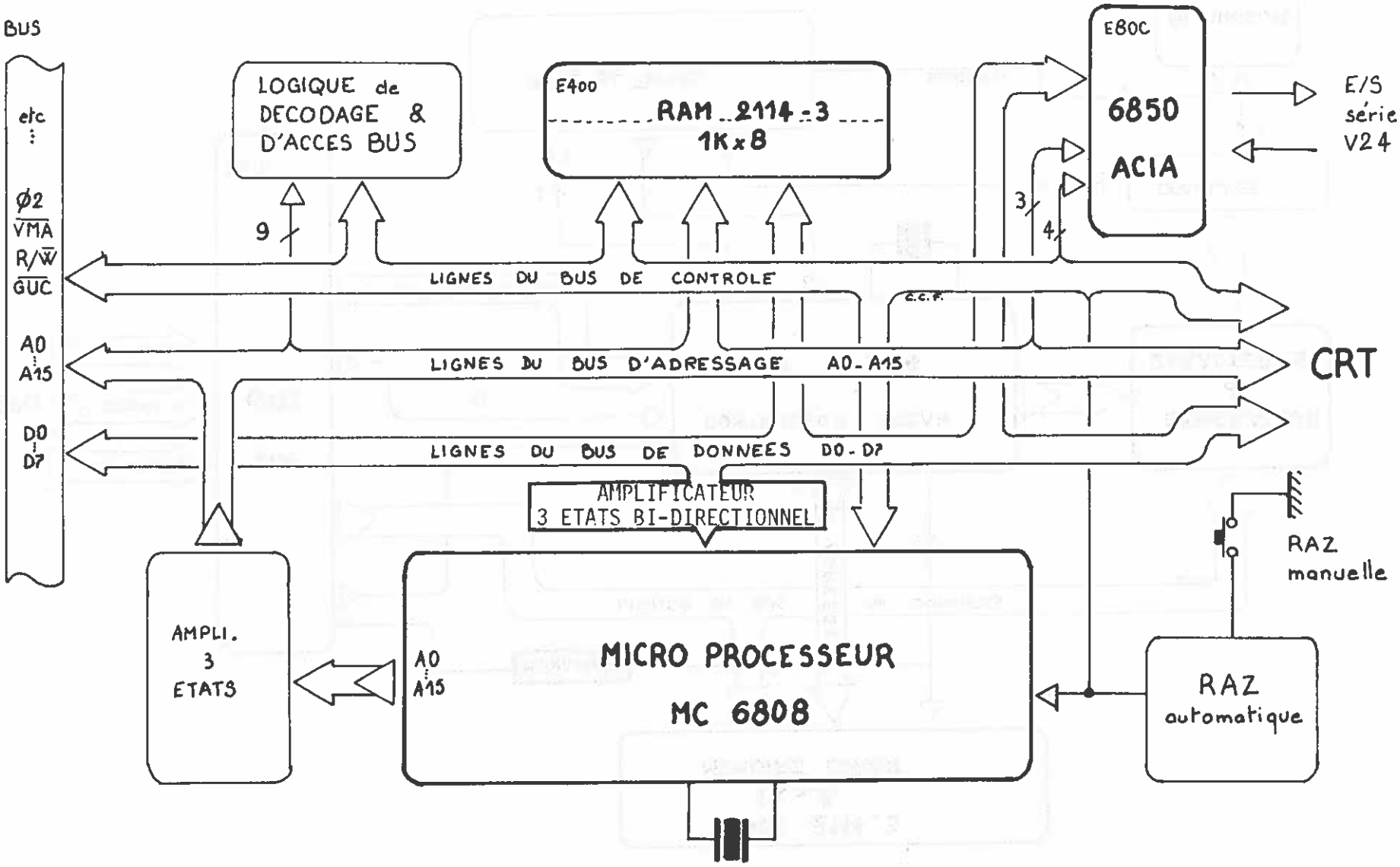
Il existe un dispositif de remise à zéro automatique, constitué par IC 6E et quelques composants passifs. Le transistor Q 1 permet de piloter l'ensemble des cartes connectées sur le bus.
Un bouton-poussoir placé sur le coté droit sous le GOUPIL autorise la réinitialisation manuelle du système (bouton RESET).

Un dispositif de décodage, (IC 6A, 6B) permet de sélectionner un ACIA (Asynchronous Communication Interface Adapter -IC 3A-) MC 6850, et une mémoire RAM (Random Access Memory -IC 4B, 4A-) d'une capacité de 1 K x 8 bits (2 x 2114 - 3).

Ces décodeurs trois parmi huit sont repérés dans l'espace mémoire par un signal GUC (Groupe Unité Centrale) généré sur la carte d'entrée-sortie.

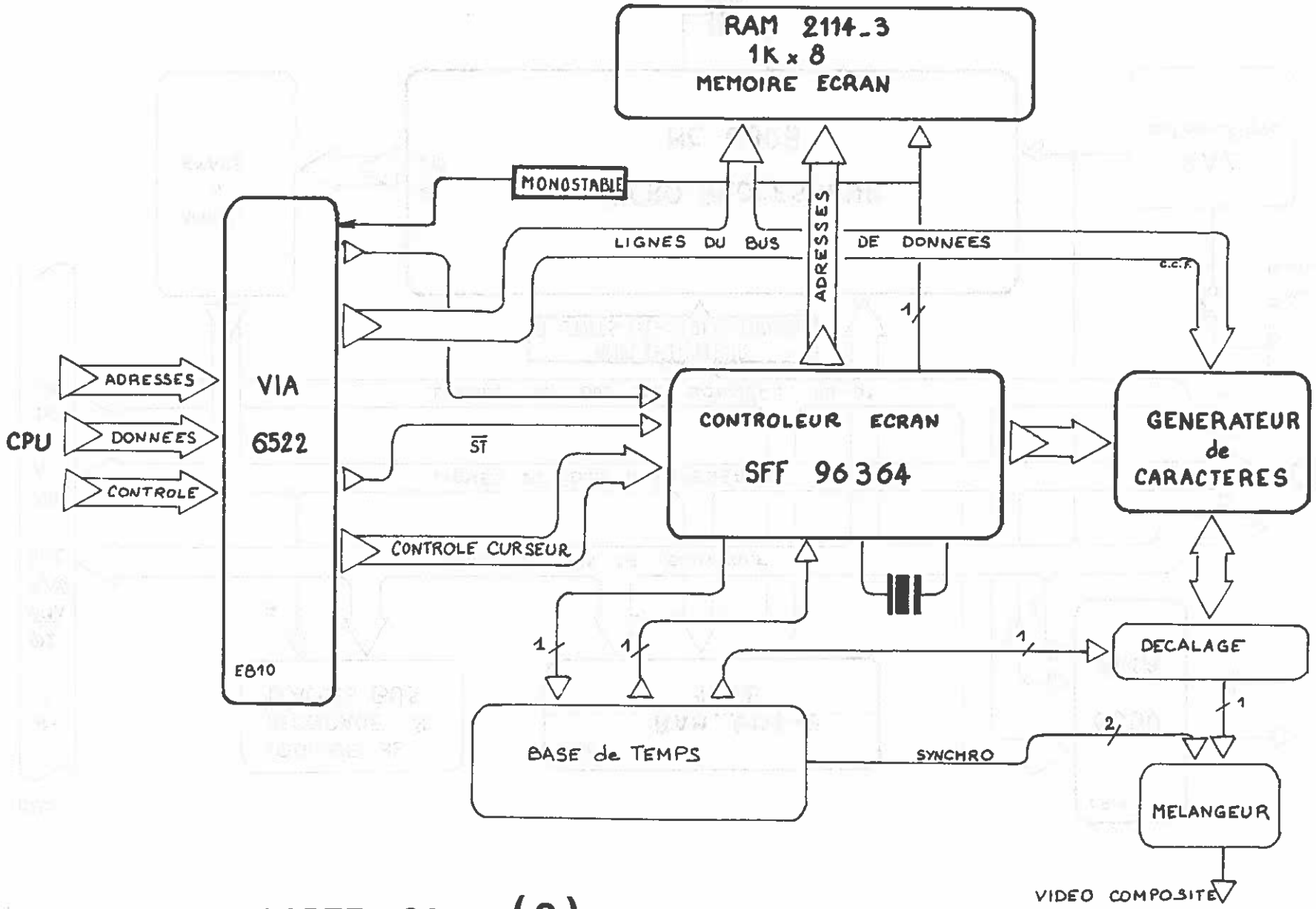
Retenons l'adresse de l'ACIA : E80C
et l'adresse de la RAM : E400.

Il faut remarquer que le boitier IC6A génère sur le bus six signaux de décodage appelés : GA, GB, GC, GD, GE, GF.
Ces signaux sont repris sur d'autres cartes pour économiser les circuits intégrés.
Le connecteur disponible pour un couplage suivant l'avis V 24 du CCITT, sur l'ACIA (MC 6850) est référencé P 1.



CARTE CPU (1)

smt



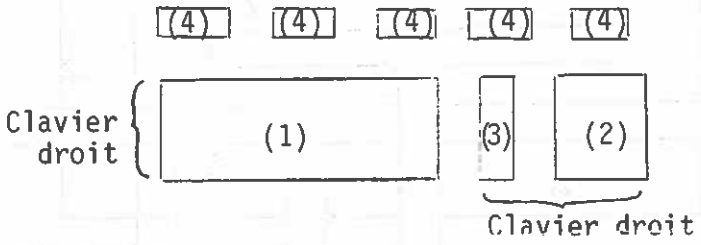
CARTE CPU (2)

smt

III - PRESENTATION DES CONTROLEURS CLAVIER + ENTREES-SORTIES + EPROMS

1) LE CLAVIER :

Le clavier est décodé par deux contrôleurs 8279 d'INTEL. Le premier contrôleur gère le clavier droit (clavier de machine à écrire) et le second, le clavier gauche (clavier hexadécimal, plus clavier de fonctions, plus clavier de gestion de la page).



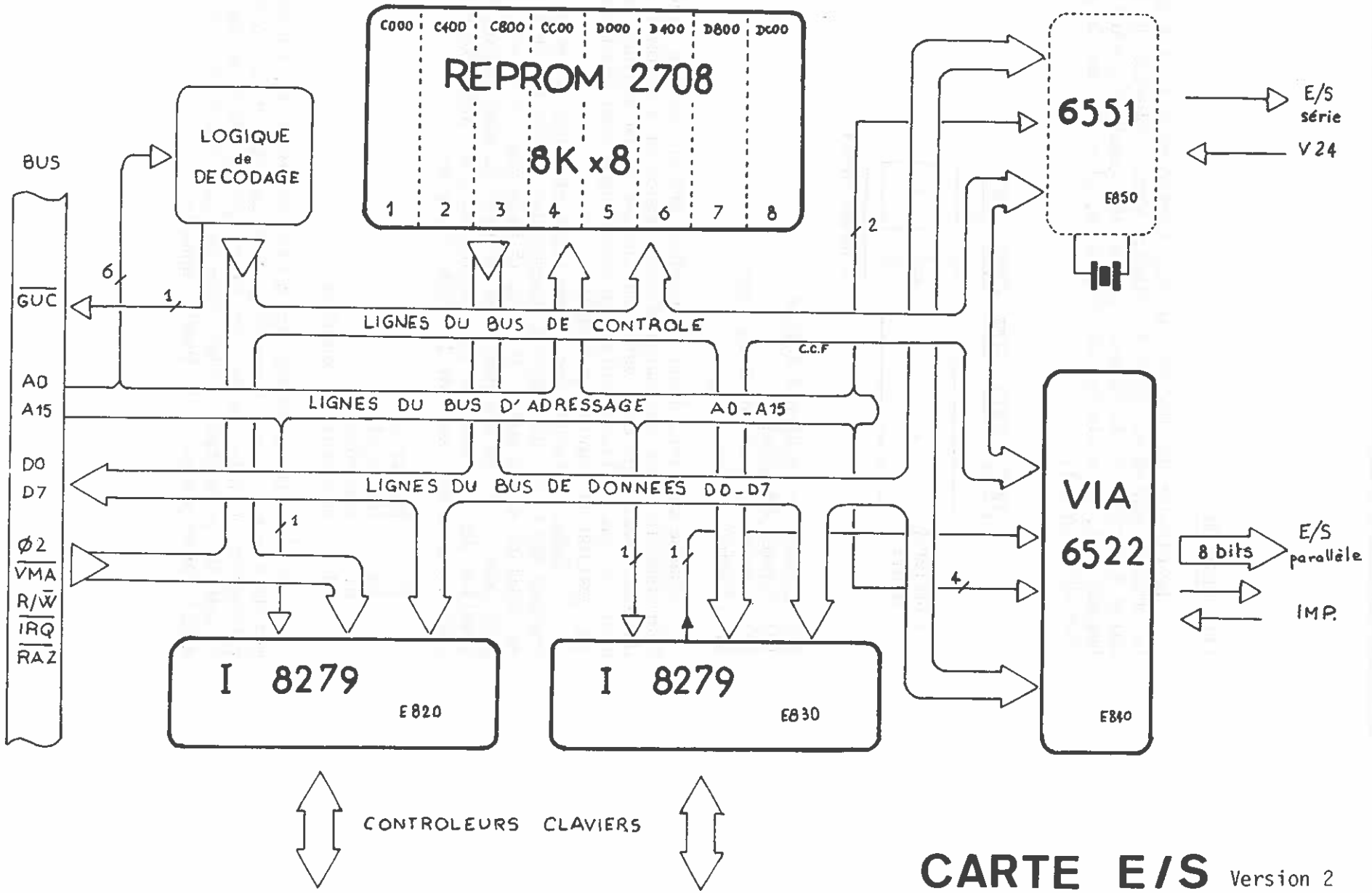
(1) clavier de machine à écrire
(2) clavier hexadécimal
(3) clavier de gestion de la page
(4) clavier de fonction

Chaque clavier, droit ou gauche, est vu par son contrôleur associé comme une matrice de 8 lignes et 8 colonnes. Un contrôleur peut donc gérer un clavier de 64 touches seulement, ce qui explique l'utilisation de deux contrôleurs pour la gestion du clavier du GOUPIL.

A chaque intersection d'une ligne et d'une colonne, correspond une touche du clavier. Lorsqu'une touche est appuyée, sa position codée sur 6 bits, (3 bits pour le n° de ligne + 3 bits pour le n° de colonne), est envoyée au contrôleur. L'utilisation des touches MAJ (Majuscule) et CTRL (Contrôle) sont codées sur 2 bits suivant la table ci-après :

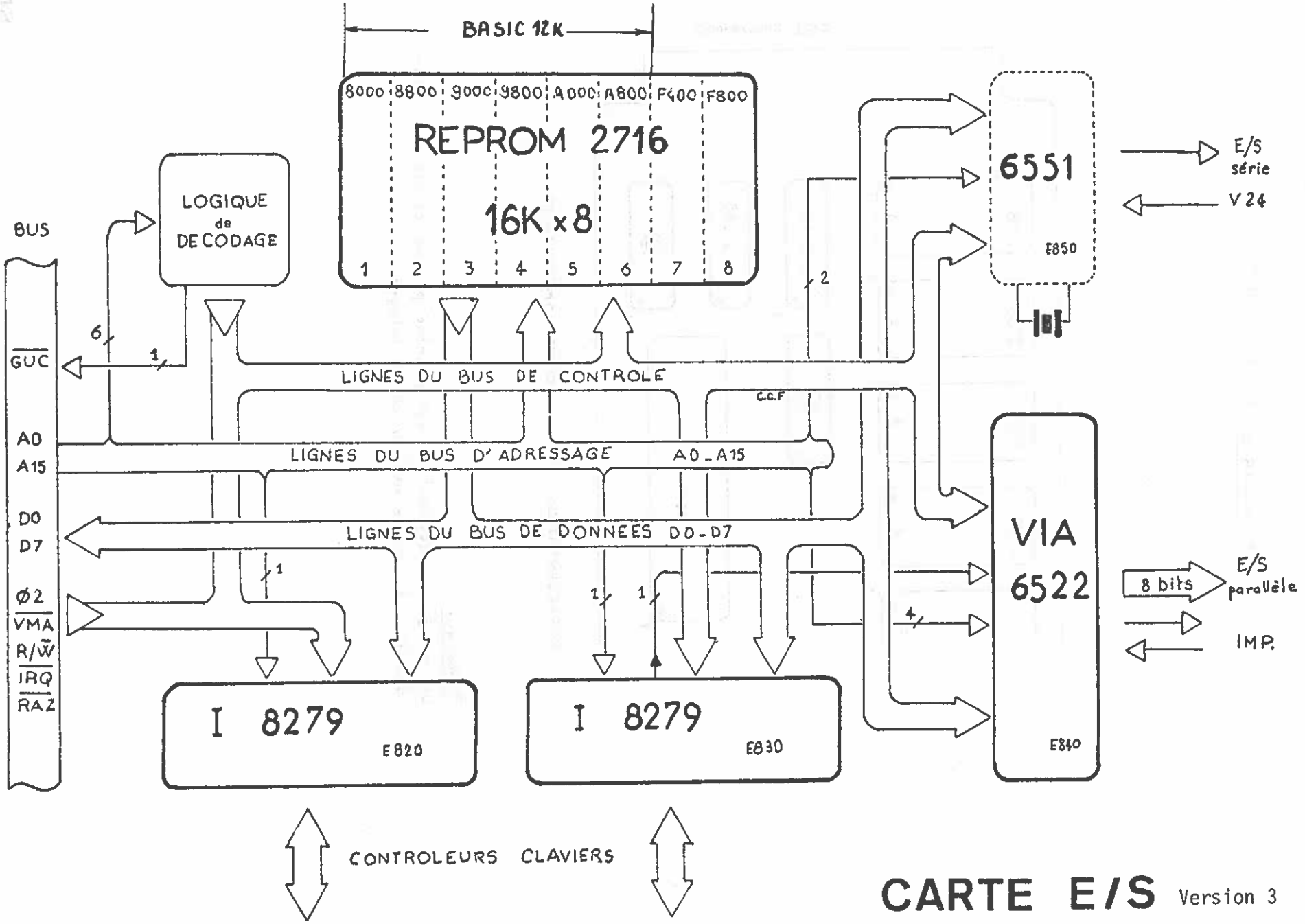
11	Majuscule
10	Majuscule
01	Contrôle
00	Majuscule + Contrôle

Ces 2 bits associés aux 6 bits de position, forment un mot de 8 bits qui est transmis par le contrôleur au microprocesseur. Celui-ci s'en sert alors pour construire une adresse qui lui permettra d'aller chercher dans une table le code ASCII correspondant à la touche appuyée.



CARTE E/S Version 2

smt

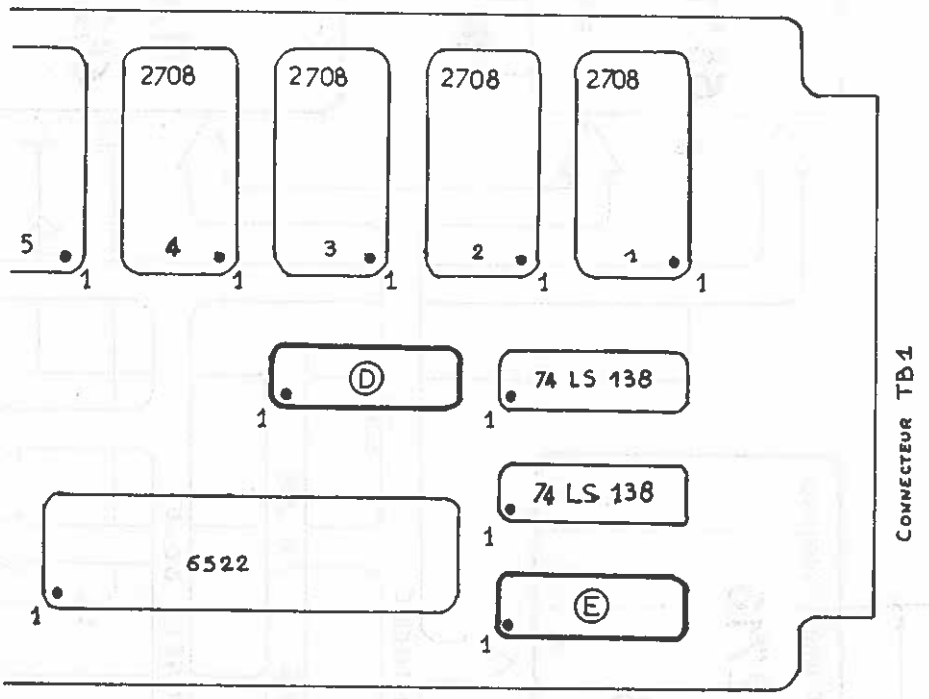


CARTE E/S Version 3

smt

UTILISATION DES PLOTS PROGRAMMABLES Ⓓ ET Ⓔ

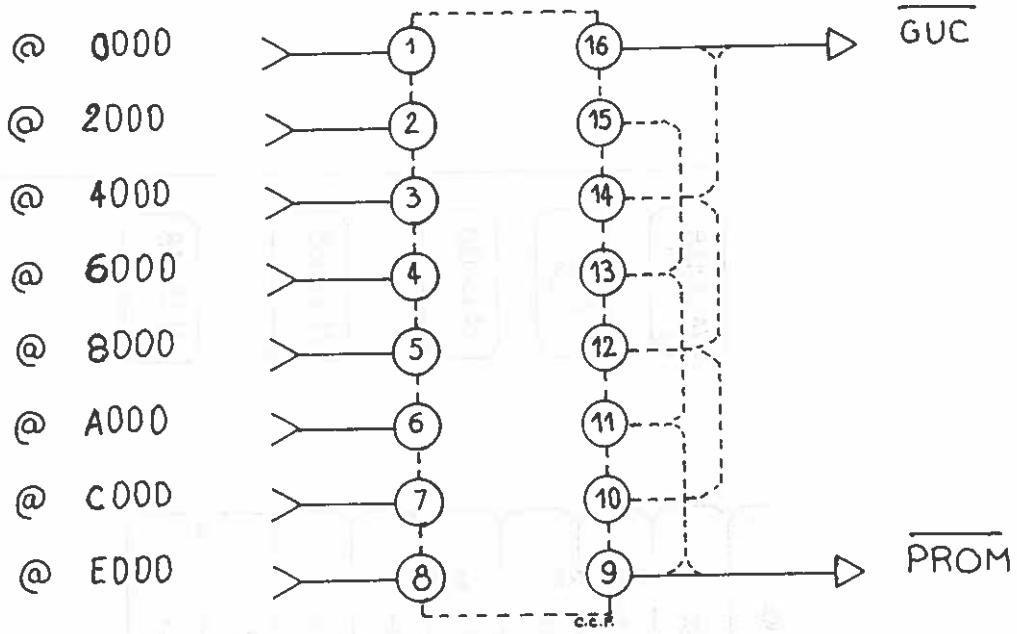
POSITION SUR LA CARTE E/S Version 2



REPRESENTATION VUE DU COTE COMPOSANTS

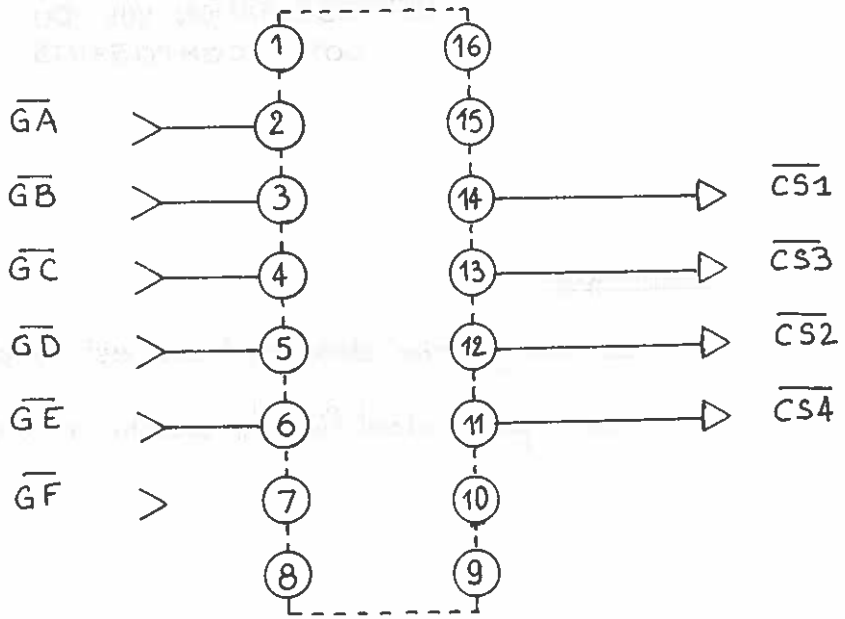
Remarque :
Le coin gauche des boitiers est accentué par un cercle noir pour identifier la broche n°1 du circuit intégré.

PLOT PROGRAMMABLE (E)
SUR CARTE E/S VERSION 2.



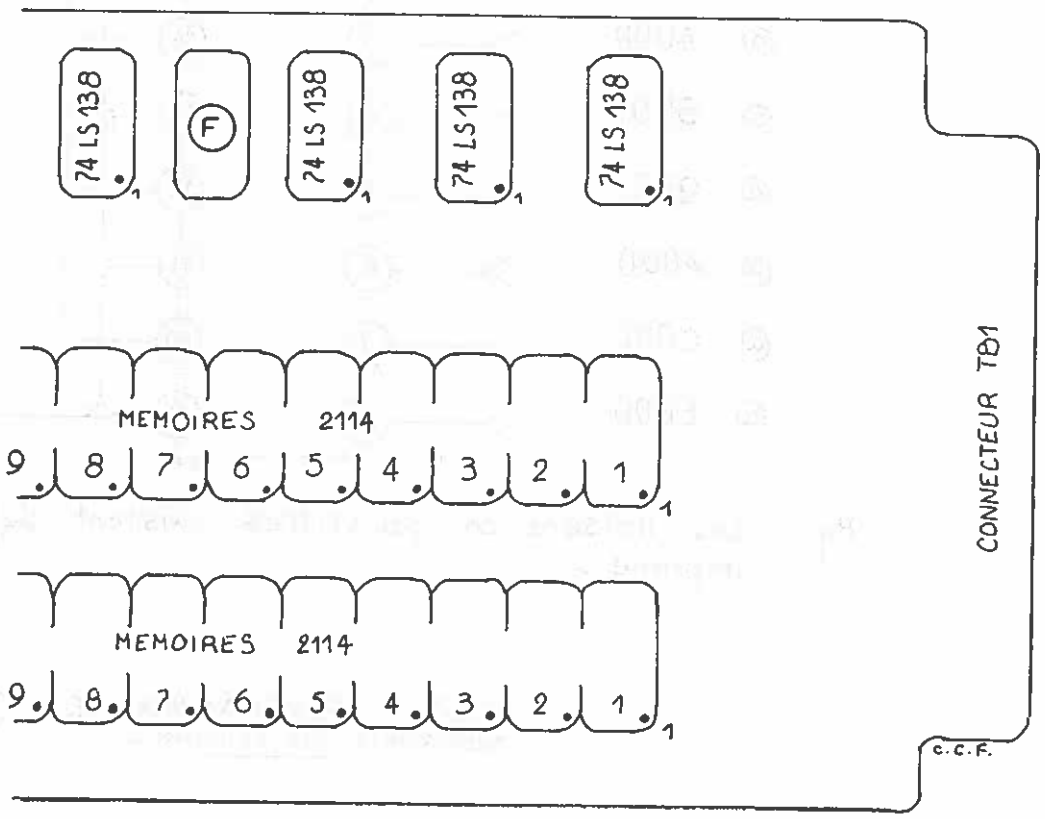
Rq: Les liaisons en pointillés existent déjà sur le circuit imprimé.

PLOT PROGRAMMABLE (D)
SUR CARTE E/S VERSION 2.



UTILISATION du PLOT PROGRAMMABLE (F)

POSITION SUR LA CARTE MEMOIRE STATIQUE



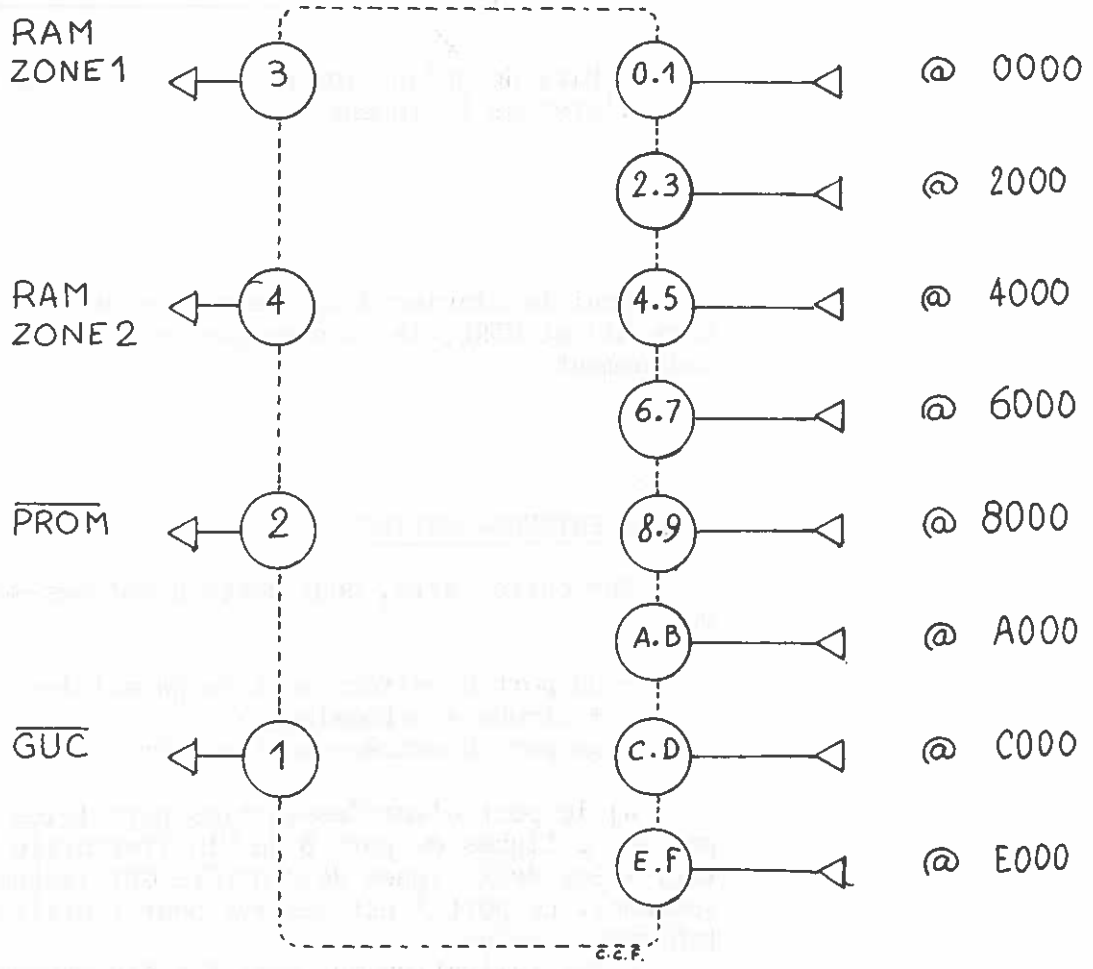
REPRESENTATION VUE DU
COTE COMPOSANTS

Remarque:

Le coin gauche des boitiers est accentué par un cercle noir pour identifier la broche n°1 des circuits intégrés

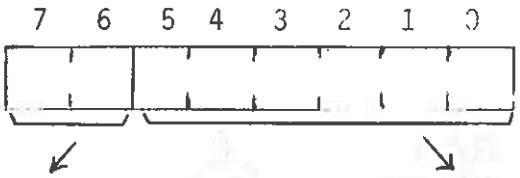
PLOT PROGRAMMABLE (F)

SUR CARTE MEMOIRE STATIQUE



(suite de la page IV-8)

Structure du mot transmis au microprocesseur :



2 bits de définition de l'état de la touche.

6 bits de position

Seul le clavier droit peut être utilisé avec les touches MAJ et CTRL, le clavier gauche est codé en majuscule uniquement.

2) LES ENTREES-SORTIES :

Sur cette carte, deux ports d'entrées-sorties sont prévus :

- un port d'entrées-sorties parallèles (8 bits + strobe + acknowledge)
- un port d'entrées-sorties séries optionnel.

a) le port d'entrées-sorties parallèles est réalisé par les 8 lignes du port B du VIA (Versatile Interface Adapter) + les deux lignes de contrôle CB1 (acknowledge) et CB2 (strobe). Le port A est réservé pour l'utilisation des contrôleurs clavier.

L'initialisation du port B a été prévue dans le moniteur SMTMON à l'adresse $ FAB2(INITVC). Dès la mise en marche ou après un RESET, les 8 lignes du port B sont programmées en sortie et CB2 enmode sortie manuelle. Le handler propre à l'imprimante connectée sur le GOUPIL, pourra être écrit par l'utilisateur, en tenant compte descaractéristiques du périphériques utilisé.

b) le port d'entrées-sorties séries est réalisé par un boitier SYNERTEC, le 6551 dont l'emplacement a été réservé sur la carte et qui sera monté à la demande de l'utilisateur.

3/ DECODAGE DES BOITIERS DES CARTES E/S version 3 et version 2

Le décodage des différents boitiers spécialisés et des EPROM se fait par l'intermédiaire de straps montés sur les connecteurs E et D. (uniquement sur version 2)

a) Décodage des boitiers spécialisés : Connecteur E

Les signaux $\overline{GA}$, $\overline{GB}$, $\overline{GC}$, $\overline{GD}$, $\overline{GE}$, et $\overline{GF}$ ont été générés sur la carte CRT + CPU et correspondent respectivement aux adresses $E820, $E830, $E840, $E850, $E860 et $E870. Les lignes $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$, et $\overline{CS4}$ sont utilisées pour sélectionner respectivement le contrôleur du clavier droit, le contrôleur du clavier gauche, le VIA pour le port d'Entrées-Sorties parallèles et le 6551 pour le port d'entrées-sorties séries. Les straps relieront donc ensemble :

$\overline{GA}$ et $\overline{CS1}$	Contrôleur clavier droit $E820
$\overline{GB}$ et $\overline{CS2}$	Contrôleur clavier gauche $E830
$\overline{GC}$ et $\overline{CS3}$	VIA du port d'E/S parallèle $E840

Le port d'entrées-sorties séries est programmé en $E850 ($\overline{GD}$ et $\overline{CS4}$) dans le cas de la version 3.

b) Décodage des EPROM : Connecteur D (version 2 uniquement)

Les EPROM sur la carte sont décodées de C000 à DFFF.

Le connecteur D permet de construire deux signaux formés à l'aide des adresses A13, A14 et A15 passées dans un 74 LS 138 ou décodeur.

Ces signaux sont d'une part le Groupe d'Unité Centrale ($\overline{GUC}$) qui est transmis à la carte unité centrale pour servir de racine au décodage des boitiers spécialisés de la carte CPU et des boitiers périphériques, et d'autre part un signal qui passé dans un 74 LS 138 avec les signaux du bus d'adresses A10, A11 et A12, permet le décodage 1K par 1K des EPROM 2708.

Les straps seront donc branchés de manière à relier la broche du signal $\overline{GUC}$ à la borne E-F (adresse E000-FFFF) et la broche constituant la racine du décodage des EPROM à la borne C-D (adresse C000-CFFF).

BROCHAGE DES CONNECTEURS DE GOUPIL (FICHES 25 POINTS)

a) Connecteur Série

Broche	Désignation
2	Réception des données
3	Emission des données
7	Masse

b) Connecteur parallèle

Broche	Fonction du 6522	Broche	Fonction du 6522/ Fonction imprimante//	Broche	Fonction du 6522/ Fonction imprimante//
1	PA0	10	CA_2 *	19	CB_1/Acknowlegde
2	PA_1	11	PB_0/Bit 1	20	CB_2/Strobe
3	PA_2	12	PB_1/Bit 2	21	Masse/GND
4	PA_3	13	PB_2/Bit 3	22	- 5 V
5	PA_4	14	PB_3/Bit 4	23	+ 12 V
6	PA_5	15	PB_4/Bit 5	24	———
7	PA_6	16	PB_5/Bit 6	25	+ 5 V
8	PA_7	17	PB_6/Bit 7		
9	CA_1*	18	PB_7/Bit 8		

* Ces lignes sont utilisées par la gestion du clavier

NOTE : Concernant la "fonction 6522", se référer à la fiche technique de ce circuit.

c) Réalisation d'un câble GOUPIL/Imprimante parallèle compatible Centronics

Broche connecteur 25 points GOUPIL	Broche connecteur 36 points imprimante
11	2
12	3
13	4
14	5
15	6
16	7
17	8
18	9
19	10
20	1
21	19 à 30

SCHEMA DE L'INTERFACE PARALLELLE POUR L'IMPRIMANTE

PINS	SIGNAL
1	$\overline{\text{Data}}$ $\overline{\text{Strobe}}$
2	Bit 1
3	Bit 2
4	Bit 3
5	Bit 4
6	Bit 5
7	Bit 6
8	Bit 7
9	Bit 8
10	$\overline{\text{AKNOWLEDGE}}$
11	BUSY
12	Paper End
13	Select
14	OV
15	N. C.
16	OV
17	Chassis Ground
18	+ 5 V

PINS	SIGNAL
19	G. N. D.
20	"
21	"
22	"
23	"
24	"
25	"
26	"
27	"
28	"
29	"
30	"
31	$\overline{\text{Input}}$ $\overline{\text{Prime}}$
32	Fault
33	OV
34	+ 10 V
35	N. C.
36	+ 23 V

IV - PRESENTATION DE LA CARTE MEMOIRE STATIQUE

Cette carte est équipée de mémoires d'une capacité de 1 k x 4 bits. (2114)

Chaque carte est extensible jusqu'à 16 K octets.

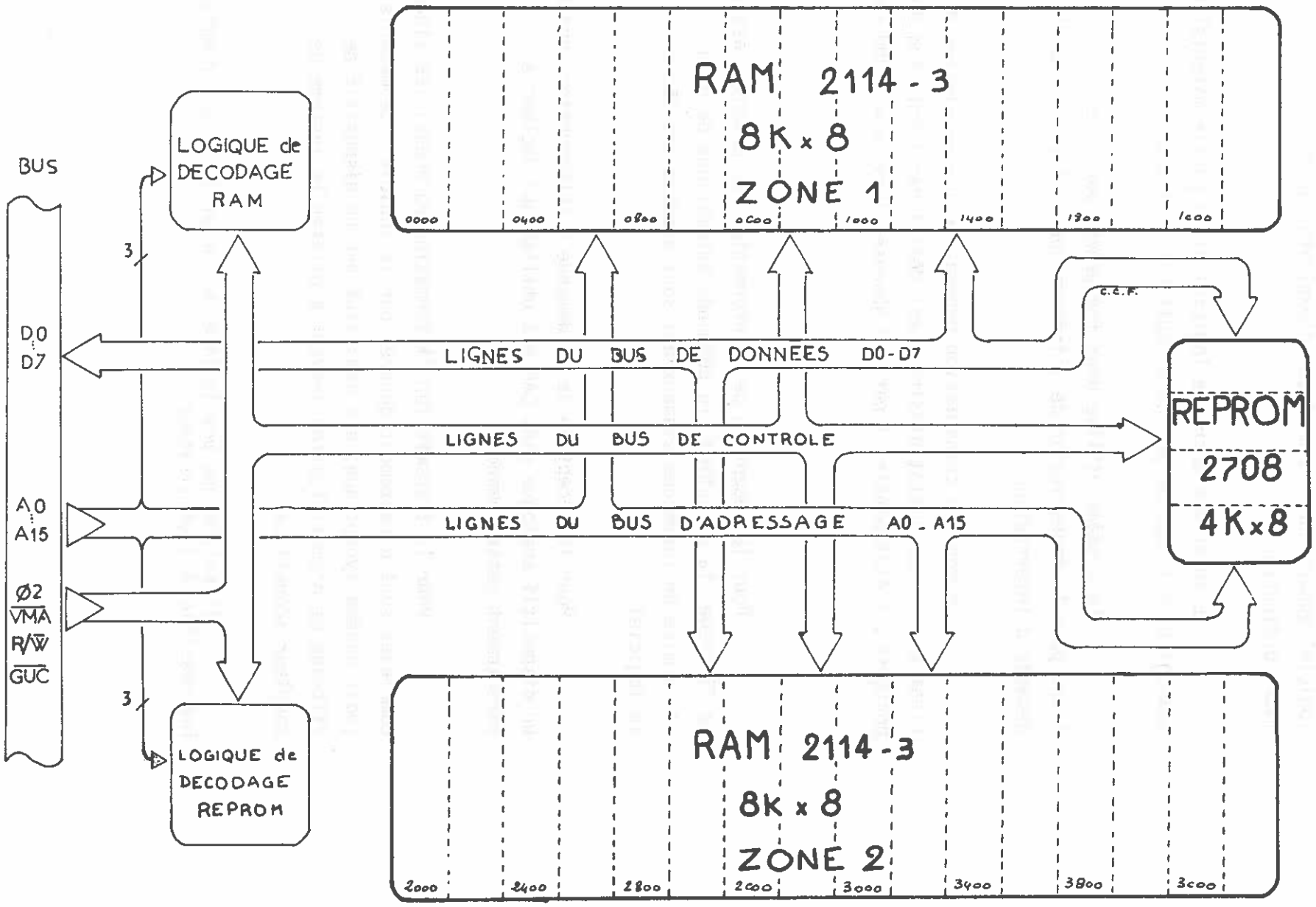
Cette carte contient aussi 4 supports recevant des mémoires EPROM d'une capacité de 1 k x 8 bits (2708)

Le décodage des mémoires RAM et REPROM est effectué par quatre 74LS138 qui sont des décodeurs 3 parmi 8 et qui permettent d'accéder aux 16 k RAM et aux 4 k REPROM n'importe où dans l'espace adressable du microprocesseur.

Grâce au plot programmable F , il est possible de réaliser les différentes combinaisons souhaitées.

Dans la configuration de base, le $\overline{GUC}$ décode les adresses E000 à FFFF.

CARTE MEMOIRE statique



V - PRESENTATION DE LA CARTE MODEM

Le coupleur acoustique de GOUPIL est géré par un logiciel appartenant au moniteur d'exploitation du micro-ordinateur.

L'interface entre ce logiciel et la partie matérielle analogique est réalisé par un circuit LSI (VIA 6522).

Le système utilise deux fréquences porteuses : l'une pour la transmission de l'information et l'autre pour la demande d'interruption.

Le mode de transmission repose sur une modulation d'amplitude c'est-à-dire que l'état binaire 0 est défini par l'émission de la porteuse , l'état binaire 1 par son absence (1 cycle = 1/300 seconde).

Pour la réception de l'information, ce matériel assure le filtrage, la détection, la commande automatique de gain et la mise en forme des signaux qui sont ensuite traités par le logiciel.

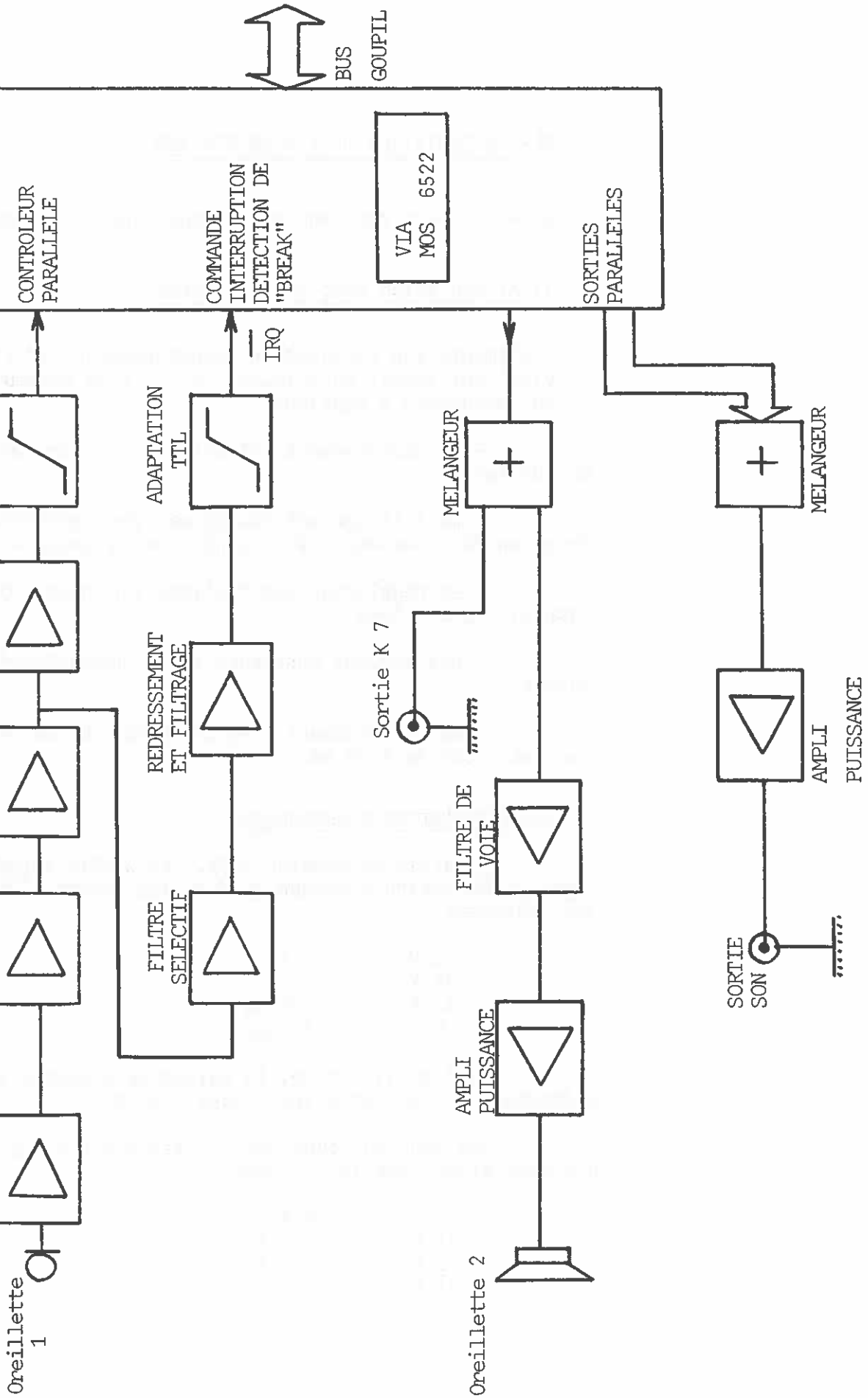
Pour la réception de la demande d'interruption (break), un dispositif analogue sans CAG est utilisé qui déclenche directement cette demande.

Pour la transmission (information ou break) les signaux complexes sont directement générés par le logiciel. Le matériel (voir schéma synoptique) est constitué par un dispositif de filtrage et d'amplification servant à piloter le système de coupleur acoustique.

Il existe une possibilité de réaler le niveau d'émission inaccessible à l'utilisateur.

SMT

SYNOPTIQUE DU COUPLEUR ACOUSTIQUE GOUPIL



VI - PRESENTATION DE L'ALIMENTATION

GOUPIL 2 peut être équipé de deux types d'alimentation :

1) Alimentation avec transformateur

L'alimentation contient un transformateur LTM (imprégné sous vide) qui reçoit au primaire le 220 V du secteur et distribue au secondaire 4 tensions.

- Le redressement est effectué par des ponts de diodes de puissance .

- Le filtrage est assuré par des capacités calculées en fonction des courants crêtes ainsi que du temps de maintien.

- La régulation est réalisée sur chaque tension par des dispositifs hybrides.

- Les sorties continues sont indépendantes les unes des autres.

- Le raccordement s'effectue sur un bornier de sortie par des clips de 6,35 mm.

2) Alimentation à découpage

A partir de Janvier 1982, les GOUPIL seront équipés d'une alimentation à découpage dont les caractéristiques sont les suivantes :

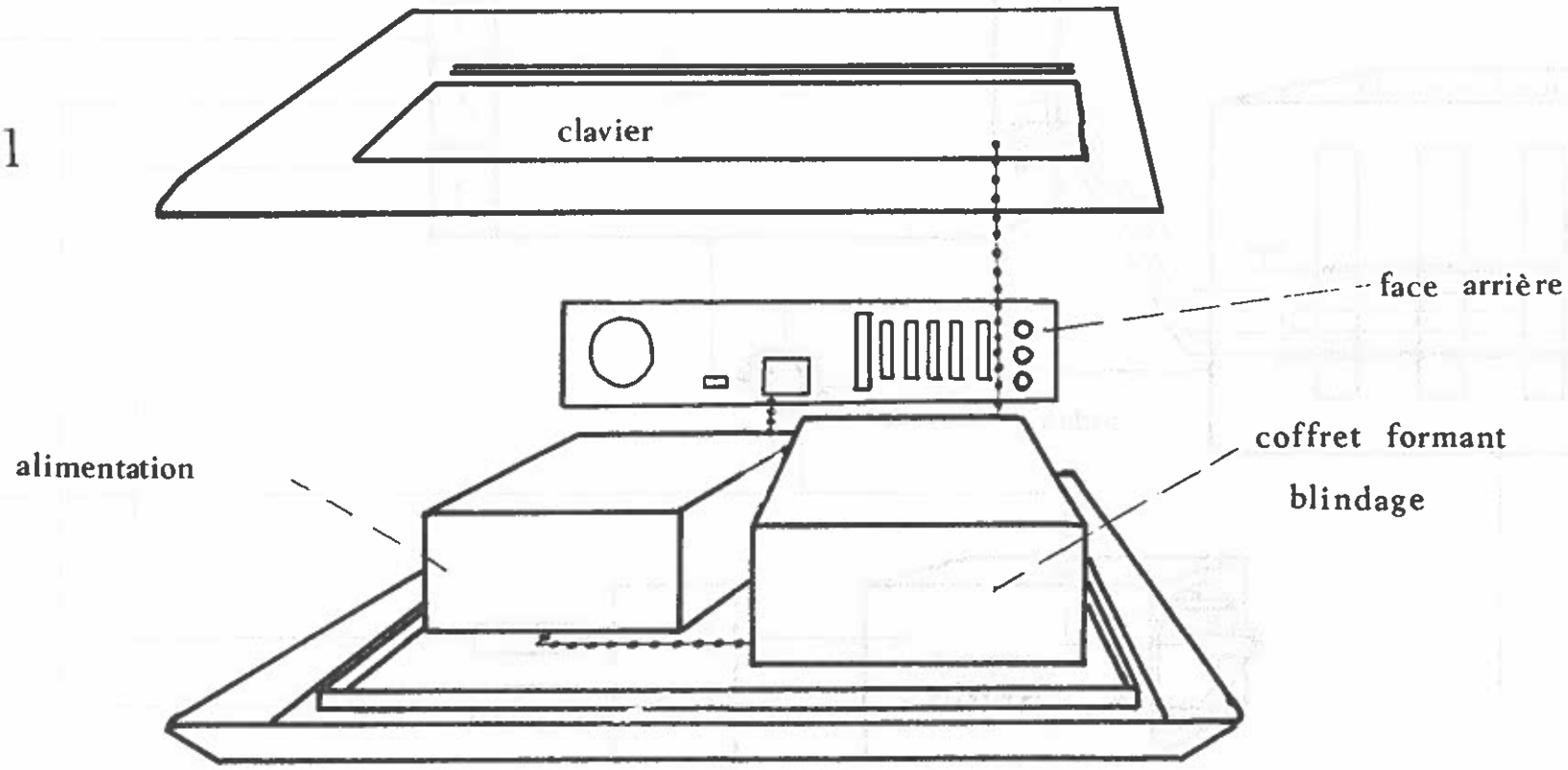
. 5 V	6 A
. 12 V	2 A
. -12 V	500 ma
. - 5 V	500 ma

Quoi qu'il arrive, la puissance produite par cette alimentation devra être inférieure à 50 Watt.

Les tensions continues restituées par ces deux types d'alimentation sont les suivantes :

. 5 V	6 A
. 12 V	2,5 A
. - 5 V	0,5 A
. -12 V	0,5 A

A1

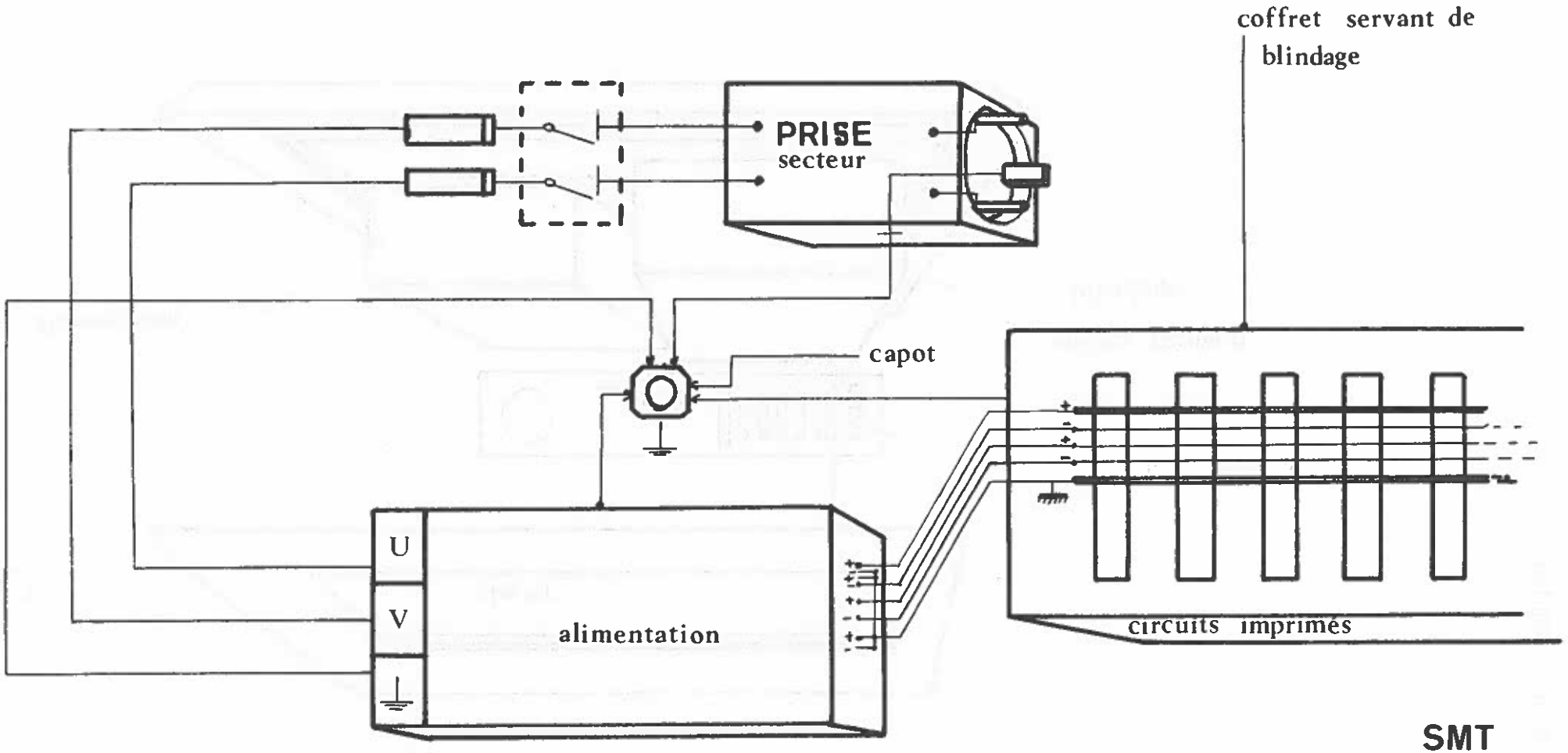


SCHEMA PHYSIQUE DES CIRCUITS
D'ALIMENTATION

LIAISONS EQUIPOTENTIELLES SIMPLES

SCHEMA FONCTIONNEL DES CIRCUITS D'ALIMENTATION

A2



SMT

VII - PRESENTATION DE LA CARTE GRAPHIQUE 256 x 256 POINTS x 8 COULEURS GOUPIL 2

Caractéristiques techniques :

- 256 x 256 points adressables individuellement par leurs coordonnées X.Y.
- 8 couleurs par point : noir, rouge, bleu, vert, jaune, magenta, cyan, blanc.
- possibilité d'utilisation en noir et blanc : l'image se compose alors de 3 plans 256 x 256 indépendants.
- commande de création d'un fond coloré sur l'écran pour l'une des huit couleurs.
- entrée (vidéo + synchro) auxiliaire pour affichage alphanumérique éventuel ou autre utilisation.
- 2 sorties vidéo indépendantes :

  couleur : R V B + synchro par prise péritélévision.

  Noir et blanc : vidéo composite.

MISE EN OEUVRE :

1/ initialisation : la carte graphique utilise le circuit Motorola MC 6845.

Avant toute opération celui-ci doit recevoir de la part de l'unité centrale un certain nombre de commandes et de données destinées à définir le format de l'image, les périodes des balayages, le cadrage, l'entrelaçage.

Le sous-programme d'initialisation est donné en annexe. Se reporter à la documentation du 6845 pour les différents paramètres. Ceux-ci doivent être adaptés à la fréquence du quartz utilisé pour l'horloge de la carte graphique.

Le sous-programme doit être présent soit comme initialisation lors du RESET, soit au début des programmes utilisant le graphique.

Il ne faut pas ré-initialiser le 6845 au milieu d'un programme graphique car on risque d'introduire des points parasites dans la mémoire d'écran.

Les adresses du 6845 sont :

- registre d'adresse $E7FE
- registre de données $E7FF

2/ Affichage

L'accès à la mémoire d'écran se fait au moyen de trois registres :

- registre X : détermine l'abscisse du point
- registre Y : détermine l'ordonnée du point
- registre C : détermine la couleur et permet les commandes spéciales

On peut écrire dans chaque registre au moyen de deux adresses consécutives

$ E7F8 - $ E7F9 : registre Y

$ E7FA $ E7FB : registre X

$ E7FC $ E7FD : registre C

Toute écriture à une adresse paire (E7F8, E7FA, E7FC) provoque une écriture en mémoire d'écran ainsi que la mise à jour du registre correspondant.

Les écritures à adresses impaires provoquent la mise à jour du registre sans affichage sur l'écran.

Exemple : affichage d'un point blanc en X = 0 Y = 0 c'est-à-dire en haut et à gauche de l'écran. On positionne les coordonnées sans affichage, puis la couleur avec affichage.

instructions		Contenu de	X	Y	C
LDAA	#0		**	**	**
pas d'écriture	STAA	SE7F9	**	00	**
sur écran	STAA	SE7FB	00	00	**
écriture	LDAA	BLANC			
	STAA	SE7FC	00	00	blanc

Il était également possible de positionner les registres dans un ordre différent C, X, Y par exemple :

```
LDAA    BLANC
STAA    $E7FD      <--|
                       |
LDAA    ## 0           |   positionnement sans écriture sur écran
STAA    $E7FB      <--|

STAA    $E7F8      <--     positionnement avec écriture
```

Cette méthode permet d'éviter l'existence d'une commande "affichage" et facilite l'écriture du logiciel.

Description du registre de commande C

Bit 7 : sélection vidéo.

à 1 : la sortie RVB reçoit la vidéo auxiliaire alphanumérique
la sortie N et B reçoit la vidéo graphique

à o la sortie RVB reçoit la vidéo graphique
la sortie N et B reçoit la vidéo auxiliaire.

Ceci permet de sélectionner à volonté sur le moniteur TV soit le signal graphique soit l'affichage alphanumérique du GOUPIL, au moyen d'une instruction programme.

Bits 0, 1, 2 : couleur

Ces bits commandent l'écriture des couleurs primaires :

Bit 0 : rouge
Bit 1 : vert
Bit 2 : bleu

Cela permet les combinaisons suivantes :

Bit 2	Bit 1	Bit 0	Couleur
0	0	0	Noir
0	0	1	Rouge
0	1	0	Vert
0	1	1	Jaune
1	0	0	Bleu
1	0	1	Magenta
1	1	0	Cyan
1	1	1	Blanc

Bits 3, 4, 5 bits de masque

Ces bits permettent d'autoriser ou d'inhiber l'écriture des primaires R V ou B en couleur

En noir et blanc, ils sélectionnent les plans d'images où l'on désire écrire :

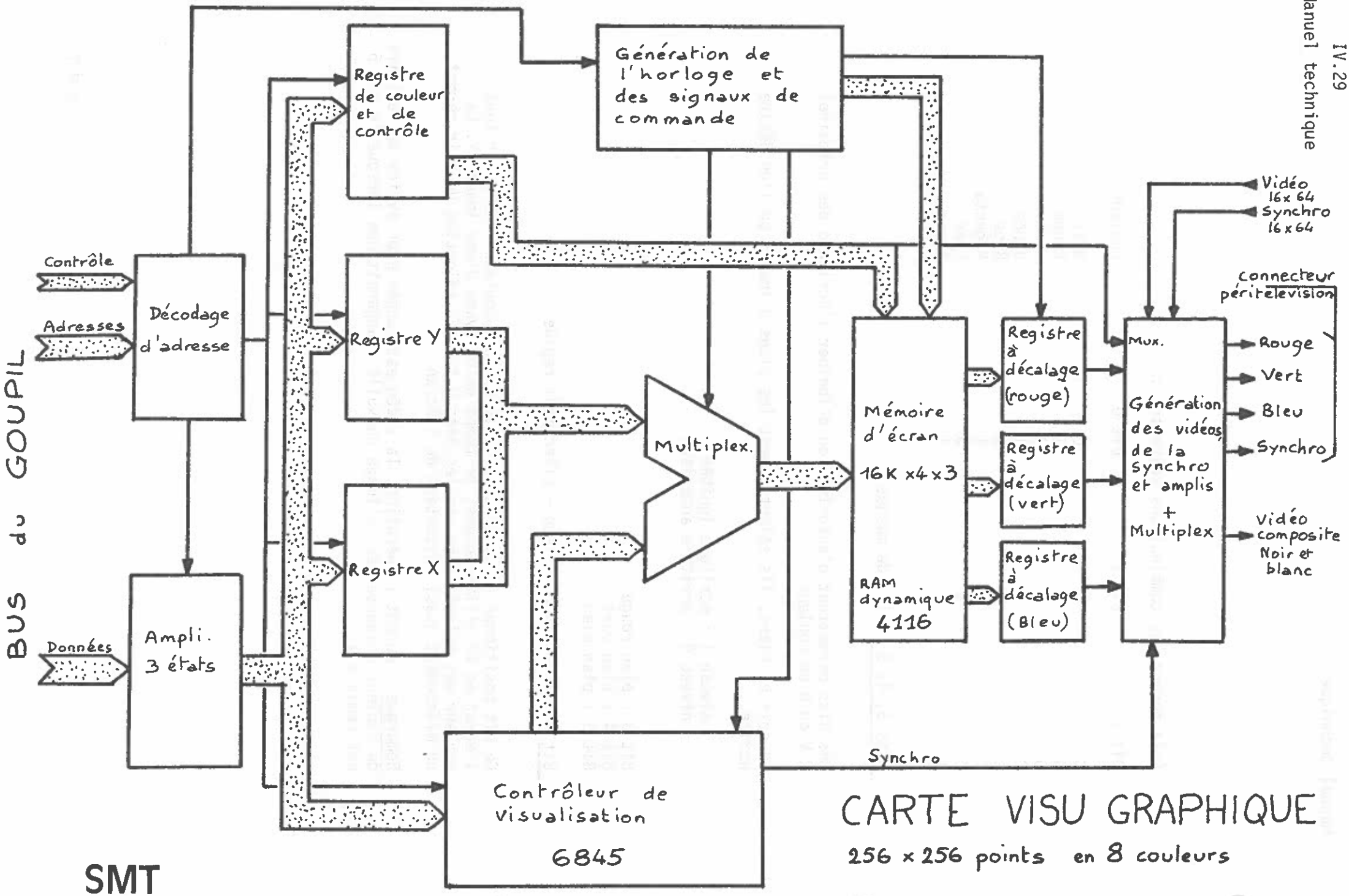
niveau 1 : écriture inhibée
niveau 0 : écriture autorisée

Bit 3 : plan rouge
Bit 4 : plan vert
Bit 5 : plan bleu

Bit 6 : création d'un fond - effacement rapide

Ce bit positionné à 0 permet l'écriture d'une couleur donnée sur tout l'écran en 20 millisecondes, le temps de balayage d'une image TV. La couleur est sélectionnée par les bits 0 à 5. La sélection du noir permet un effacement quasi instantané de l'écran.

Remarque : durant l'opération, la vidéo est coupée pour éviter des éclairs de couleur désagréables. L'image nouvelle n'apparait que lorsque le bit 6 est remis à 1



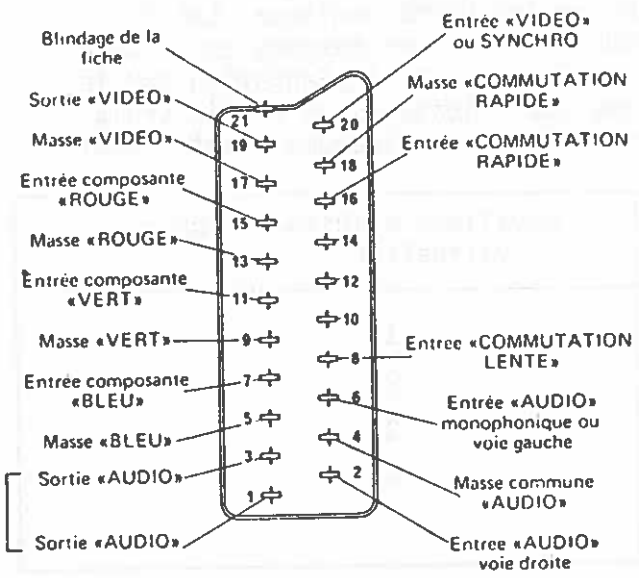
CARTE VISU GRAPHIQUE

256 x 256 points en 8 couleurs

SMT

CONNEXION DE LA VIDEO COULEUR

PRISE PERITELEVISION



BROCHAGE DU CONNECTEUR PERITELEVISION BX 210

Broche	Désignation	Valeurs d'adaptation
1	SORTIE « AUDIO » VD STEREO	100 mV eff 1 kΩ
2	ENTREE « AUDIO » VD STEREO	100 mV eff Ze ≥ 4,7 kΩ
3	SORTIE « AUDIO » VG STEREO	100 mV eff 1 kΩ
4	MASSE COMMUNE « AUDIO »	
5	MASSE « BLEU »	
6	ENTREE « AUDIO » (mono) ou VG STEREO	100 mV eff Ze ≥ 4,7 kΩ
7	ENTREE COMPOSANTE « BLEU »	1 Vcc Ze = 75 Ω
8	ENTREE COMMUT. « LENTE »	0 à 1 V TV ou 10 à 12 V PERIT Ze ≥ 4,7 kΩ
9	MASSE « VERT »	
10	HORLOGE	→ branchée sur futurs modèles
11	ENTREE COMPOSANTE « VERT »	1 Vcc Ze = 75 Ω
12	TELECOMMANDE	→ branchée sur futurs modèles
13	MASSE « ROUGE »	
14	MASSE TELECOMMANDE	→ branchée sur futurs modèles
15	ENTREE COMPOSANTE « ROUGE »	1 Vcc Ze = 75 Ω
16	ENTREE COMMUT. « RAPIDE »	0 à 0,4 V TV ou 1 à 3 V PERIT Ze = 75 Ω
17	MASSE VIDEO	
18	MASSE COMMUT. « RAPIDE »	
19	SORTIE VIDEO	1 Vcc 75 Ω
20	ENTREE VIDEO (ou synchro)	1 Vcc Ze = 75 Ω
21	BLINDAGE DE LA FICHE	

VIII - PRESENTATION DE LA CARTE MEMOIRE DYNAMIQUE

Cette carte est équipée de mémoires dynamiques 16 K x 1 bit (4116), et peut recevoir jusqu'à 64 K octets de mémoire, par incrément de 16 K octets (16K, 32K, 48K, 64K). Toutefois, lorsque 64K sont installés, seuls 56K sont effectivement accessibles par l'utilisateur, afin de laisser libre la place pour les entrées-sorties et les PROMS moniteur. Les 8 derniers K octets de la mémoire (de E000 à FFFF) sont masqués en lecture automatiquement lors du RESET *. Des cavaliers (voir implantation sur le synoptique) permettent la validation des amplificateurs de sortie trois états de la carte lorsque les boitiers mémoire correspondant sont installés.

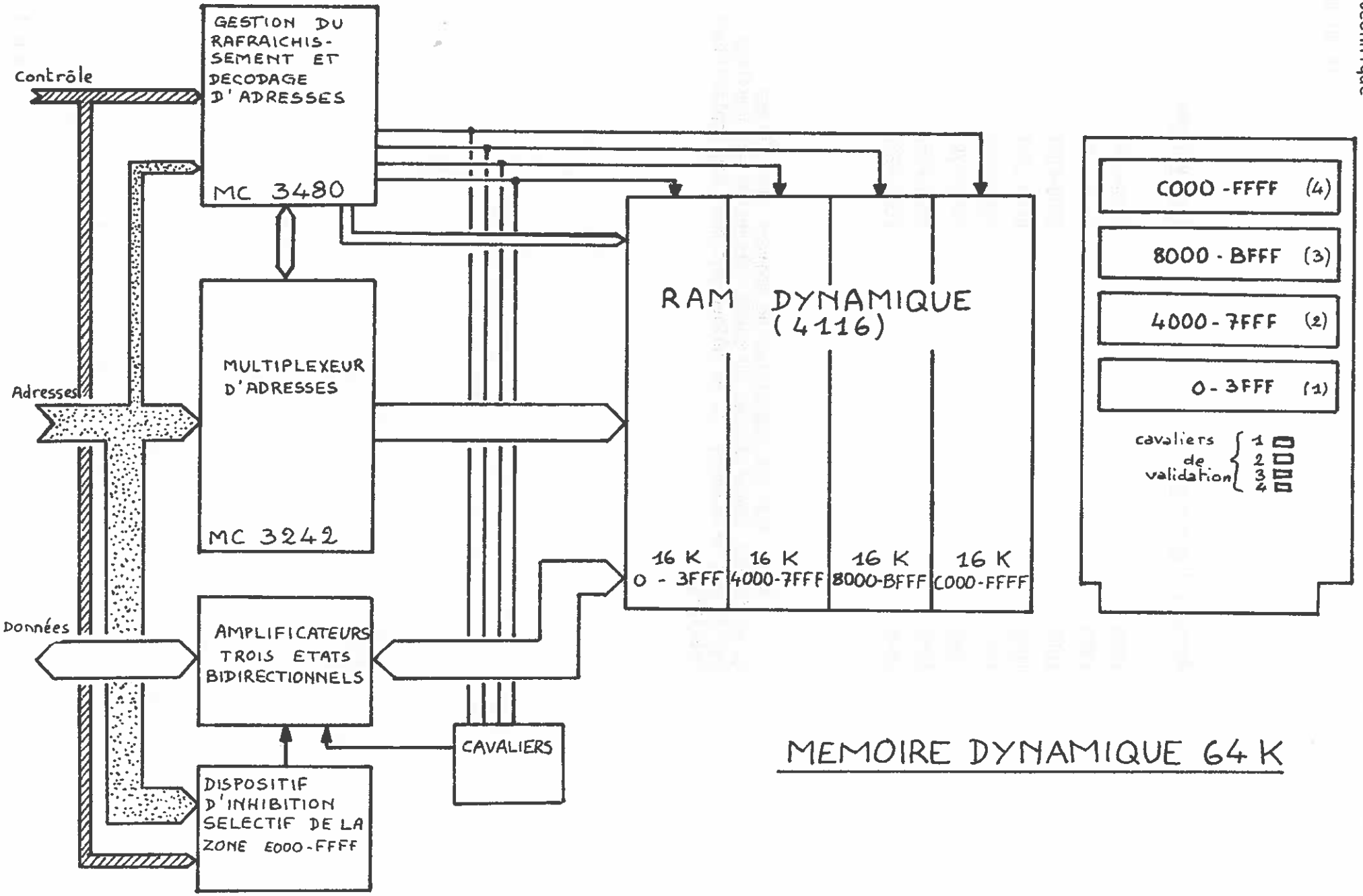
Zone équipée de boitiers	Cavaliers à installer pour validation
0 - 3FFF	1
4000 - 7FFF	2
8000 - BFFF	3
C000 - FFFF	4

Les mémoires dynamiques présentent deux avantages importants sur les mémoires statiques, d'une part une consommation électrique beaucoup plus faible et donc une température à l'intérieur de GOUPIL plus basse, d'autre part une plus grande densité permettant de réunir 48 K octets sur une carte au lieu de 3. Enfin, le rafraichissement des mémoires dynamiques est effectué en mode transparent, c'est-à-dire qu'il n'entraîne aucune diminution de vitesse du processeur.

* Note : Pour des applications particulières nécessitant l'accès à ces 8 derniers K octets, un dispositif permet de lever l'interdiction de lecture de la RAM, par blocs de 1K octets. La zone EBF8-EBFF contrôle cette validation ; ainsi une écriture de $ 01 en EBF8 valide la zone E000-E3FF, l'écriture de $00 masque cette zone.

Adresse d'écriture	Zone validée
EBF8	E000-E3FF
EBF9	E400-E7FF
EBFA	E800-EBFF
EBFB	EC00-EFFF
EBFC	F000-F3FF
EBFD	F400-F7FF
EBFE	F800-FBFF
EBFF	FC00-FFFF

Noter que ces opérations ne doivent être faites qu'en toutes connaissances de cause (attention aux conflits sur le bus de données !), et uniquement pour des applications spéciales.



MEMOIRE DYNAMIQUE 64 K

IX - PRESENTATION DE LA CARTE 24 x 80

Caractéristiques techniques :

a) Niveau matériel

- 1920 caractères organisés en 24 lignes de 80 caractères
- Jeu de 128 caractères définis dans une matrice 8 x 13
- Possibilité d'inversion vidéo (caractère noir sur fond blanc) au niveau de chaque caractère
- Sortie vidéo composite noir et blanc
- Mémoire d'écran directement accessible par le processeur en lecture et en écriture

b) Niveau logiciel : voir le tableau des fonctions disponibles sous moniteur GPMØN version 1.3.

Mise en oeuvre

Cette carte est organisée autour du contrôleur de visualisation MOTOROLA MC 6845 qui sert à générer le balayage de la mémoire d'écran, la synchronisation du moniteur, etc... Ce circuit est implanté aux adresses suivantes :

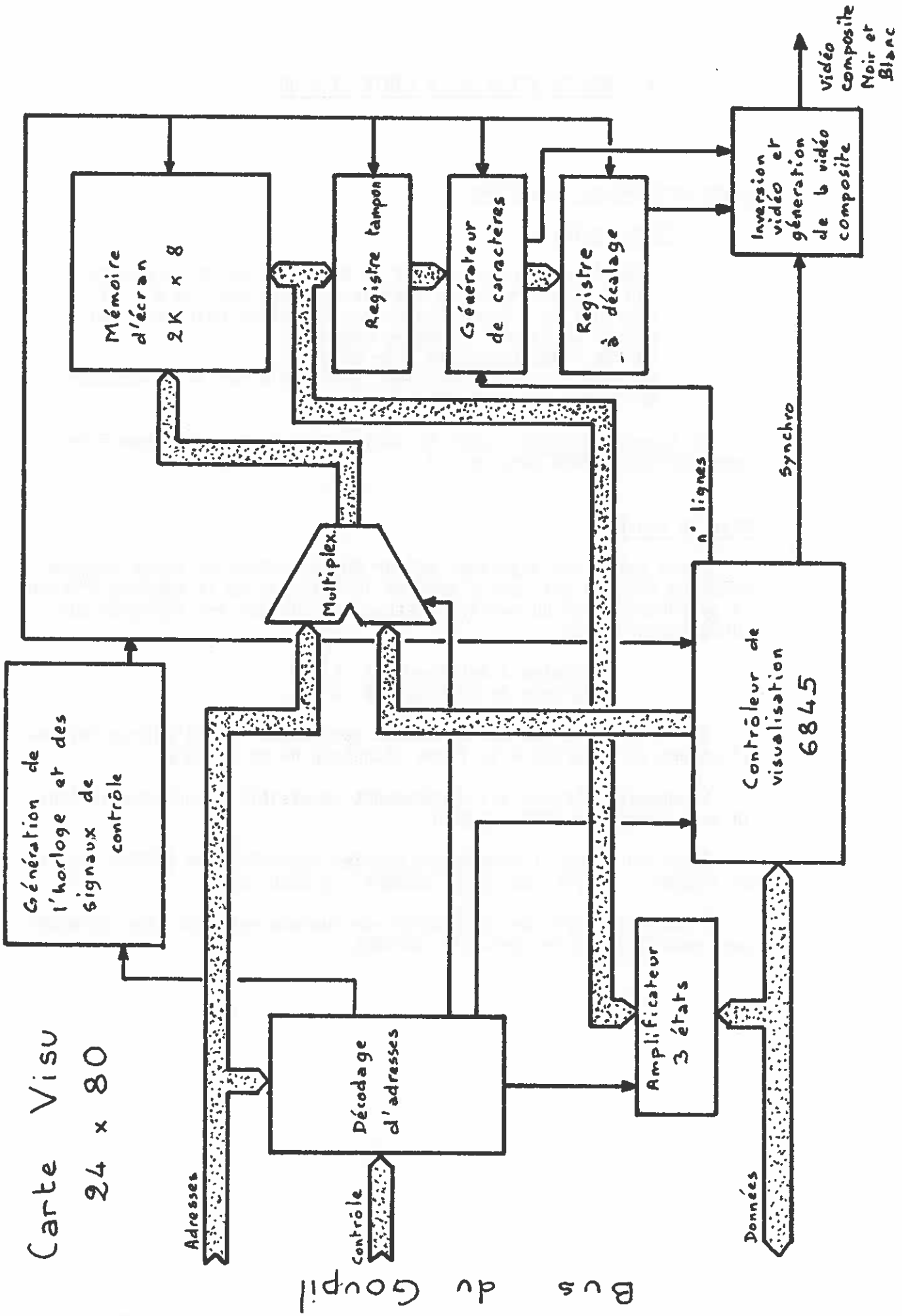
Registre d'adresses : $ E 870
Registre de données : $ E 871

Son initialisation est effectuée par GPMØN. Pour d'autres initialisations, se reporter à la fiche technique de ce circuit.

La mémoire d'écran est directement accessible au processeur dans la zone mémoire $ EC00 - $ F3FF.

Après un Reset, l'adresse du premier caractère est $ F000, celle du 1023ème : $ F3FF, celle du 1024ème : $ EC00, etc.

A noter qu'après des opérations de rouleau vertical, ces adresses sont modifiées et ne sont plus valides.


Carte Visu
24 x 80
Bus du Goupil
Mémoire d'écran 2K x 8
Registre tampon
Générateur de caractères
Registre à décalage
Inversion vidéo et génération de la vidéo composite
Vidéo composite Noir et Blanc
Synchro
n° lignes
Multiplex.
Contrôleur de visualisation 6845
Génération de l'horloge et des signaux de contrôle
Amplificateur 3 états
Décodage d'adresses
Adresses
Contrôle
Données

- V -

DESCRIPTION DE GPMØN

PRESENTATION DU MONITEUR

V - PRESENTATION DU MONITEUR

GPMØN

DESCRIPTION DE GPMØN

1 INTRODUCTION

GPMØN est un moniteur contenu en mémoire morte (REPROM) qui permet à l'utilisateur de communiquer avec son GOUPIL.

Il gère le "dialogue" homme/machine par l'intermédiaire d'un certain nombre de commandes. Il offre un ensemble de sous-programmes utilitaires (en particulier d'entrées/sorties) appelables par les programmes utilisateurs.

C'est aussi essentiellement un outil de développement logiciel et matériel, offrant les fonctions suivantes :

- gestion cassette et modem (voir notices correspondantes).
- affichage du contenu de la mémoire.
- affichage et modifications des registres.
- affichage et modification du contenu d'une cellule mémoire.
- insertion/effacement/affichage de 8 points d'arrêt.
- lancement d'un programme.
- trace d'un programme.
- exécution pas à pas.

GPMØN communique par l'intermédiaire du clavier et de l'écran du GOUPIL

2 IMPLANTATION MEMOIRE

ROM F800 - FFFF (2K)
RAM E400 - E4C5 (198 octets pour le noyau)

VECTEURS D'INTERRUPTION : FFF8 - FFFF

ADRESSES RAM UTILES :

E400 - E401	pointeur vers sous programme de gestion d'IRQ
E402 - E403	pointeur vers sous programme de gestion de NMI
E408	drapeau d'écho sur entrée au clavier
E409	drapeau de sortie simultanée sur imprimante

E434 - E435 pointeur pile utilisateurs (initialisé à E4BF pour 128 octets de pile)

E436 - E437 pointeur SWI

3. SOMMAIRE DES COMMANDES

Il y a deux types de commandes : celles qui ne requièrent pas d'arguments, celles qui ont besoin d'un ou deux arguments constitués de 4 caractères hexadécimaux.

Après l'entrée du dernier caractère hexadécimal du dernier argument ou de la commande s'il n'y a pas d'argument, celle-ci est exécutée immédiatement (un retour à la ligne n'est pas nécessaire). Un espace est automatiquement affiché après le caractère de commande.

Exemple : (les caractères soulignés sont affichés par le moniteur)

Type 1	+ W Ø	effacement des points d'arrêt
Type 2	GØ0100	exécuter un programme à l'adresse 0100

Commandes :

M AAAA	affichage/modifications de la cellule mémoire AAAA
D AAAA BBBB	affichage du contenu de la mémoire de l'adresse AAAA à l'adresse BBBB
R	affichage des registres
B	affichage des adresses des points d'arrêt
C	reprise de l'exécution à partir de l'adresse courante
N	exécution de l'instruction suivante
T NNNN	trace de NNNN instructions
G AAAA	exécution d'un programme à l'adresse AAAA
W	effacement des points d'arrêt
U AAAA	suppression du point d'arrêt à l'adresse AAAA
V AAAA	pose d'un point d'arrêt à l'adresse AAAA
K	exécution du programme de gestion cassette/modem
Ø	commande d'impression simultanée à l'affichage sur l'écran

4. ENTREES DU MONITEUR

Lors de l'exécution d'un programme utilisateur, il y a plusieurs façons de redonner le contrôle au moniteur.

- Couper l'alimentation et remettre le système sous tension. Le contenu de la RAM est perdu (dont le programme) et le système est réinitialisé.
- Actionner le bouton RESET de remise à zéro, le moniteur est réinitialisé, mais le contenu de la RAM est conservé.
- Mettre dans le programme une instruction de branchement au point d'entrée à chaud du moniteur CONTRL.

5. AFFICHAGE ET MODIFICATION MEMOIRE

Syntaxe : ± MØ AAAA

AAAAØNNØXXZ

Fonction :

AAAA	adresse hexadécimale de la cellule mémoire à visualiser.
NN	contenu actuel de la cellule mémoire
XX	deux caractères hexadécimaux pour modifier le contenu de la cellule mémoire. Si XX n'est pas tapé par l'utilisateur, le contenu de la mémoire est inchangé.
Z	un point à la position Z permet d'afficher la cellule mémoire suivante AAAA+1 et son contenu et le moniteur attend une nouvelle modification;un signe - permet de revenir à l'adresse précédente AAAA - 1 Tout autre caractère non hexadécimal termine la commande.

Exemple :

```
+M 0100
0100 86.        contenu de la cellule 0100
0101 01 10.     modification de la cellule suivante 0101
0102 C6.        affichage de la cellule suivante 0102
0103 A0 03.     modification de la cellule 0103
0104 7E-        affichage de la cellule suivante 0104
0103 03-        retour à la cellule 0103
0102 C6-        retour à la cellule 0102
0101 10 02↵     nouvelle modification de la cellule 0101
+               fin de la commande
```

6. AFFICHAGE DU CONTENU D'UNE ZONE MEMOIRE

Syntaxe : ± DØAAAAØBBBB

Fonction :

AAAA adresse hexadécimale du début de la zone mémoire à afficher

BBBB adresse hexadécimale de la fin de la zone mémoire à afficher

Exemple :

```
+ D0100 0112
0100 BE E4 32 BD F9 A0 7F E4 08 BD F9 54 16 BD F9 7F
0110 CE F8 00
```

L'affichage est fait à raison de 16 octets par ligne avec en début de ligne, l'adresse du premier octet.

7. AFFICHAGE DES REGISTRES

Syntaxe : ± R

Fonction :

La commande R permet d'afficher le contenu des registres courants du programme utilisateur, selon le format :

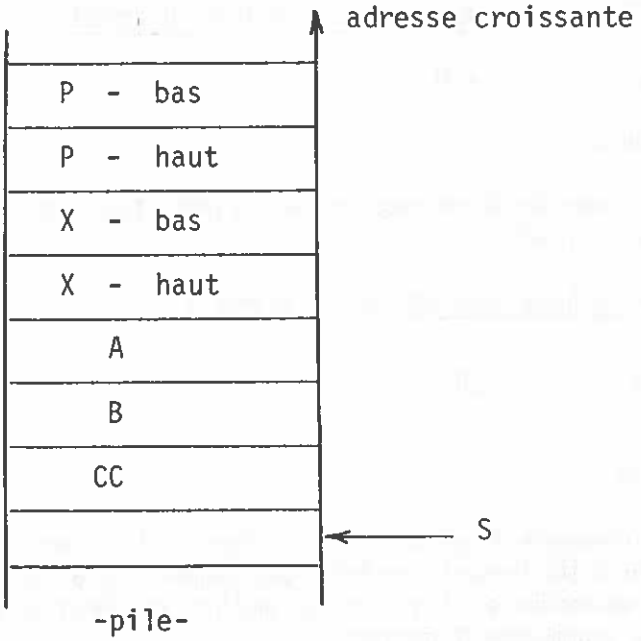
CCØBBØAAØXXXXØPPPPØSSSS

CC	2 chiffres hexadécimaux représentant le registre Code de condition	
BB	2 chiffres hexadécimaux représentant le registre	B
AA	2 " " " " "	A
XXXX	4 " " " " "	X
PPPP	4 " " " " "	P
SSSS	4 " " " " "	S

Le contenu affiché du registre de pile S est inférieur de 7 au registre S du programme utilisateur puisque le registre est affiché après la dernière trace ou le dernier point d'arrêt (SWI). C'est aussi vrai après l'initialisation du moniteur (mise sous tension ou RESET).

Modification du contenu des registres.

Le registre S est sauvegardé en mémoire RAM à l'adresse E434-E435 et peut être modifié par la commande M. Il pointe une cellule mémoire avantle contexte sauvegardé dans la pile lors de la dernière interruption logicielle (SWI), selon le schéma suivant :



La fonction M peut être utilisée pour modifier les registres directement dans la pile.

8. POINTS D'ARRET

8.1. Insertion d'un point d'arrêt :

Syntaxe : ± VØAAAA

Fonction :

La commande V permet de placer un point d'arrêt à l'adresse AAAA dans un programme utilisateur. Un maximum de 8 points d'arrêts peuvent être définis. AAAA est une adresse de 4 chiffres hexadécimaux.
Les points d'arrêt sont insérés au moyen de l'instruction SWI. Ils ne sont pas visibles par l'utilisateur parce qu'ils sont enlevés après chaque trace ou arrêt et restaurés dans le programme seulement quand une des commandes G ou C est exécutée.
Il faut noter qu'un point d'arrêt ne peut pas être inséré à l'adresse d'une instruction en mémoire ROM. L'adresse 0000 ne peut pas contenir un point d'arrêt.

8.2. Suppression d'un point d'arrêt :

Syntaxe : ± UØAAAA

Fonction :

La commande U permet de supprimer un point d'arrêt précédemment inséré à l'adresse AAAA.

8.3. Effacement de tous les points d'arrêt :

Syntaxe : ± W

Fonctions :

La commande W permet de supprimer tous les points d'arrêt préalablement insérés.

8.4. Affichage des points d'arrêt :

Syntaxe : ±B

Fonction :

La commande B permet d'afficher à l'écran les adresses où ont été préalablement insérés des points d'arrêt.
Par exemple s'il y a deux points d'arrêt aux adresses 1000 et 1100, la commande B donnera :

± BØ1000Ø1100

9. COMMANDES D'EXECUTION

9.1. Exécution d'un programme

Syntaxe : ±GØAAAA

Fonction :

La commande G permet d'exécuter un programme à l'adresse AAAA (4 chiffres hexadécimaux). Les registres X,A,B,CC (affichés par la commande R) sont chargés par le contenu de la pile courante. La pile peut être modifiée par la commande M. Le registre de pile est chargé par le contenu de SP qui contient le pointeur de pile courant lors de la dernière entrée dans le moniteur, soit sur un point d'arrêt, soit par le point d'entrée à chaud (S : = pointeur de pile utilisateur -7).

Le pointeur de pile peut ne pas être géré par le programme utilisateur. Si la pile moniteur est utilisée (E440-E4BF), il faut s'assurer que les besoins ne dépassent pas les 128 octets disponibles. Le pointeur de pile peut être modifié par la commande M à l'adresse E434-E435

9.2. Reprise de l'exécution

Syntaxe : ± C

Fonction :

La commande C permet de reprendre l'exécution d'un programme interrompu par un point d'arrêt, ou à la fin d'une trace. La commande charge le registre de pile S par le contenu de SP (E434-E435) et exécute une instruction RTI. La pile doit donc contenir des informations valides en particulier une adresse valide pour la restauration du compteur ordinal (P).

Une commande C entrée immédiatement après la mise sous tension provoque un changement de contexte et un branchement aléatoire, le contenu de la pile n'étant pas défini.

Pour la mise au point d'un programme, l'utilisateur doit poser au moins un point d'arrêt dans son programme, lancer l'exécution par la commande GØAAAA et utiliser la commande C pour reprendre l'exécution après la rencontre d'un point d'arrêt.

10. TRACE ET EXECUTION PAS A PAS

10.1 Exécution pas à pas

Syntaxe : ± N

Fonction :

L'instruction suivante est exécutée et les registres sont affichés. Les conditions nécessaires pour la commande C doivent être satisfaites pour la commande N (i.e. validité du contenu de la pile).

La commande détermine la prochaine instruction à exécuter après l'instruction courante (qui peut être un branchement) y place un point d'arrêt (SWI) et exécute l'instruction courante.

Il faut noter que la commande ne peut pas être utilisée sur un programme en ROM, ou sur une instruction de branchement vers un programme en ROM.

Il faut aussi noter que l'utilisation du registre S autrement qu'en pointeur de pile peut provoquer des problèmes (par ex : S utilisé comme index d'un tampon de données, l'exécution d'un SWI provoquera l'écrasement de 7 données dans le tampon)

Pour des raisons de sécurité, la commande N ne fonctionne pas sur une instruction SWI de l'utilisateur.

10.2 TRACE

Syntaxe : ±TØNNNN

Fonction :

La commande T permet de "tracer" le nombre d'instructions données par NNNN (4 chiffres hexadécimaux) à partir de l'instruction courante en affichant les registres à l'exécution de chaque instruction. Les commentaires donnés pour N concernent aussi la commande T.

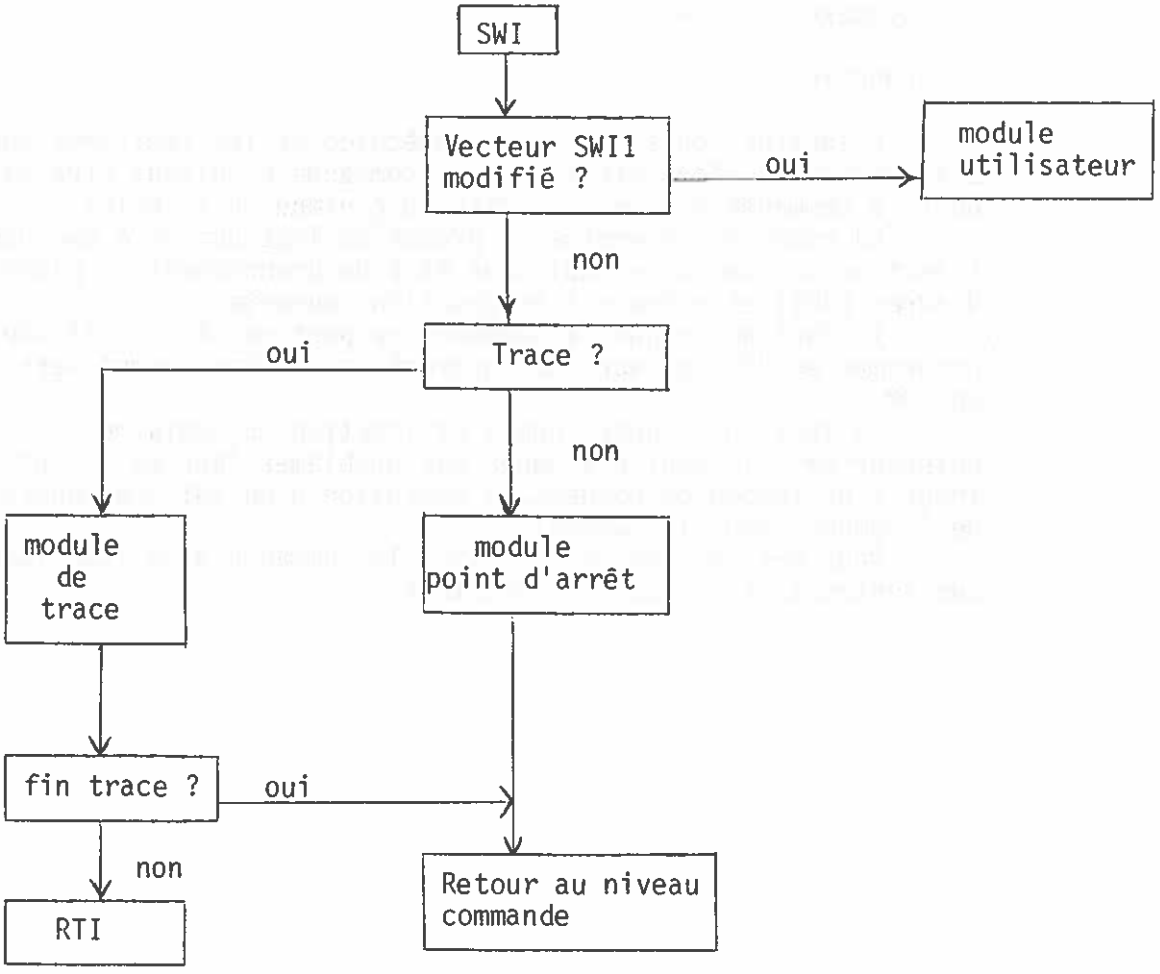
Une instruction SWI dans le programme de l'utilisateur arrête l'exécution de la trace.

La commande TØ0000 est équivalente à TØ10000 et trace 65536 instructions.

11. TRAITEMENT DES SWI

Le vecteur SWI1 à l'adresse E436-E437 pointe à l'initialisation vers le module de traitement des SWI du moniteur GPMØN. Si l'utilisateur modifie ce vecteur, les possibilités de points d'arrêt et de trace sont supprimées.

L'organigramme de traitement des SWI est le suivant :



12. INTERRUPTIONS IRQ - NMI

L'utilisateur a la possibilité d'utiliser les interruptions masquables IRQ ou non masquables NMI. Le moniteur GPMØN peut donner le contrôle à des modules de traitement externes par l'intermédiaire de deux vecteurs :

IØV	E400-E401	vecteur IRQ
NIØ	E402-E403	vecteur NMI

qui doivent être initialisés dynamiquement par le programme d'application ou manuellement au moyen de la commande M.

13. IMPRESSION SIMULTANEE

Syntaxe : ± Ø

Fonction :

La commande Ø permet d'imprimer tous les caractères affichés à l'écran sur une imprimante connectée à l'interface parallèle de type "Centronics".

Cette commande fonctionne en "bascule", la première frappe autorise l'impression, la suivante la supprime.

14. SOUS-PROGRAMMES UTILES ET DRAPEAUX

START (FCF1)	point d'entrée à froid (RESET)
CØNTRL (FFE6)	point d'entrée à chaud, module de reconnaissance des commandes (FD12)
INCH (FFEF)	entrée d'un caractère du clavier dans le registre A (8 bits) et affichage du caractère à l'écran si le drapeau d'écho (ØUTSW E408) est à zéro. A, CC, modifiés. X, B, préservés.
BYTE (FD69)	Entrée de deux chiffres hexadécimaux et conversion en un nombre binaire d'un octet dans A. A, B, CC modifiés. X préservé.

INHEX (FD42) — entrée d'un chiffre hexadécimal au clavier et conversion binaire dans le registre A.
A,CC, modifiés
B,X préservés
En cas d'entrée du caractère non hexadécimal retour au niveau commande du moniteur (CØNTRL)

BADDR (FD57) — entrée au clavier de 4 chiffres hexadécimaux et conversion binaire en un nombre de 16 bits dans le registre X.
A, B, CC, X modifiés
le nombre est sauvegardé en mémoire E4ØA-E4ØB (XHI, XLØW).

CHKCHR (FFE9) — test du clavier : si un caractère a été frappé
Z = 0, sinon Z = 1
A,CC, modifiés
B,X, préservés.

ØUTCH (FFEC) — sortie d'un caractère contenu dans le registre A vers l'écran et vers l'imprimante si le drapeau d'impression PRTFLG est égal à FF
(si PRTFLG = 0 pas d'impression)
CC, modifié
A,B,X préservés

PDATA1 (FFF2) — sortie vers l'écran d'une chaine de caractères pointée par le registre X avec possibilité d'impression simultanée.
La chaine doit être terminée par le caractère EØT(04)
A,CC,X modifiés
B préservé

ØUTHL (FD74) — sortie vers l'écran des 4 bits de gauche du registre A, convertis en un caractère hexadécimal.
A,CC modifiés
B,X préservés.

ØUTHR (FD78) — sortie vers l'écran des 4 bits de droite du registre A comme pour ØUTHL.

ØUT2H (FDA6)	sortie vers l'écran du contenu de la cellule mémoire pointée par le registre X sous forme de deux chiffres hexadécimaux. A,CC,X modifiés. B préservé.
ØUT2HA (FDA9)	sortie vers l'écran du contenu de A sous forme de deux chiffres hexadécimaux. A,CC, modifiés. B, X préservés.
ØUT2HS (FD81)	même fonction que ØUT2H, plus la sortie d'un espace suivant les deux chiffres. A,CC,X modifiés.
ØUT4HS (FDAF)	sortie vers l'écran, du contenu de deux cellules mémoires consécutives pointées par le registre X, sous la forme de 4 chiffres hexadécimaux suivis d'un espace A,CC,X modifiés. B préservé.
ØUTS (FDB3)	sortie d'un espace vers l'écran CC modifié. A,B,X préservés
PCRLF (FFF5)	sortie vers l'écran des deux caractères ASCII CR et LF, retour chariot et saut de ligne. A,CC, modifiés. B,X préservés.
ØUTSW (E408)	drapeau d'écho Si l'utilisateur positionne ce drapeau à une valeur non nulle, le sous programme d'entrée INCH ne renvoie pas le caractère vers l'écran (pas d'écho). ØUTSW est initialisé à 0 par le moniteur dans la boucle de contrôle (point d'entrée à chaud).
PRTFLG (E409)	drapeau d'impression simultanée. S'il est à zéro, il n'y a pas d'impression, s'il est à FF il y a impression (initialisé 0), à chaque appel du sous-programme de sortie ØUTCH.

PØUT (FC8F)

sortie d'un caractère contenu dans le registre A vers l'imprimante parallèle.
CC modifié
A,B,X préservés.

INICIA (FC25)

initialisation de l'ACIA-CPU ;
à l'entrée :
A : = mot de commande de l'ACIA
(parité, bits de stop, division de l'horloge,...)
X : = initialisation du timer 1 du VIA-CPU
pour obtenir l'horloge sur PB7 (conférer notice VIA).
Par exemple :

Pour 300 Bauds X : = 6400
Pour 1200 " X : = 1800

avec l'horloge divisée par 16
A,B,CC modifiés
X préservés

TOACIA (FC36)

sortie d'un caractère contenu dans le registre A vers l'ACIA (avec attente de l'état prêt)
CC modifiés
A,B,X préservés

INACIA (FC43)

entrée dans le registre A d'un caractère reçu par l'ACIA (avec attente de l'état prêt) ;
le bit de parité est mis à zéro.
A,CC modifiés
B,X préservés.

CHKACI (FC4D)

test de l'ACIA : Si un caractère a été reçu
Z = 0, sinon Z = 1
A,CC modifiés
B,X préservés.

```
                    NAM    GPMON V1.3
                    OPT    PAG

                ************************************
                *                                  *
                *    GPMON                         *
                *    VERSION 1.3  AUGUST 81        *
                *    16*64   5" FLOPPY DISK        *
                *    COPYRIGHT 1980 BY SMT         *
                *                                  *
                ************************************


                *
                *  COMMAND SET
                *  -----------
                *
                *     M  MEMORY CHANGE
                *     D  MEMORY DUMP
                *     R  DISPLAY CONTENTS OF TARGET STACK
                *         CC  B  A  X  P  S
                *     B  PRINT OUT ALL BREAKPOINTS
                *     C  CONTINUE EXECUTION FROM CURRENT LOCATION
                *     N  NEXT INSTRUCTION
                *     T  TRACE N INSTRUCTIONS
                *     G  GO TO LOCATION N
                *     W  DELETE ALL BREAKPOINTS
                *     U  RESET BREAKPOINT AT ADDRESS N
                *     V  SET BREAKPOINT AT ADDRESS N
                *     K  K7/MODEM HANDLER
                *     O  SWITCH OUTPUT TO PRINTER
                *
                *
                *  KEYBOARD FUNCTIONS
                *  ------------------
                *
                *     BASIC  EXECUTE ROM BASIC
                *     COM    SET TERMINAL MODE
                *     BOOT   EXECUTE DISK BOOTSTRAP
                *     GR     GRAPHIC MODE
                *
                *
                *  ENTRY AND I/O ROUTINES
                *  ----------------------
                *
                *     CONTRL $FFE6  WARM START
                *     CHKCHR $FFE9  CHECK FOR TYPED CHARACTER
                *     OUTCH  $FFEC  OUTPUT ONE CHARACTER
                *     INCH   $FFEF  INPUT ONE CHARACTER
                *     PDATA1 $FFF2  PRINT CHARACTERS STRING
                *     PCRLF  $FFF5  PRINT CR/LF
                *

                *
                * PSEUDO MACROS DEFINITIONS
                *
0085            SKIP1   EQU   $85       SKIP ONE BYTE
```

```
GPMON V1.3                          22-10-81  TSC ASSEMBLER  PAGE    1

   5B  008C          SKIP2   EQU    $8C       SKIP TWO BYTES
```

```
GPMON V1.3                          22-10-81  TSC ASSEMBLER  PAGE    2

   60  F8B0                  ORG    $F8B0
   62                        NAM    SCREEN HANDLER
   63                        OPT    PAG
   64
   65                  ***********************************
   66                  *                                 *
   67                  *        SCREEN HANDLER           *
   68                  *                                 *
   69                  ***********************************
   70
   71
   72                  *
   73                  *     I/O ROUTINES
   74                  *     ------------
   75                  *
   76                  *        INITVS    INITIALIZATIONS
   77                  *        OUTPUT    OUPUT ONE CHARACTER
   78                  *        BIPBIP    OUTPUT BELL
   79                  *
   80                  *
   81                  * EQUATES
   82                  *
   83  E810            VIAVIS  EQU    $E810
```

```
SCREEN HANDLER                    22-10-81  TSC ASSEMBLER  PAGE    3

                        *
                        *  SCREEN TABLE DEFINITION
                        *
F8B0                    TABVIS  EQU    *
F8B0 0F                         FCB    $F,$F,$F,$F,$F,$F,$F,$F
F8B8 04                         FCB    $4,$F,$A,$6,$8,$9,$F,$F
F8C0 0F                         FCB    $F,$F,$F,$F,$F,$F,$F,$F
F8C8 0F                         FCB    $F,$F,$F,$F,$0,$7,$2,$D


                        * INITIALIZATIONS
                        ************
                        *
F8D0 86 7F              INITVS  LDA A  £$7F
F8D2 B7 E8 1E                   STA A  VIAVIS+IER
F8D5 86 FF                      LDA A  £$FF
F8D7 B7 E8 12                   STA A  VIAVIS+DDRB
F8DA 86 7F                      LDA A  £$7F
F8DC B7 E8 13                   STA A  VIAVIS+DDRA
F8DF 86 FE                      LDA A  £$FE
F8E1 B7 E8 1C                   STA A  VIAVIS+PCR
F8E4 CE 01 00                   LDX    £$100
F8E7 FF E4 C5                   STX    VTEMP
F8EA 86 0C                      LDA A  £$C
F8EC 8D 01                      BSR    OUTPUT
F8EE 39                         RTS
```

```
SCREEN HANDLER                    22-10-81  TSC ASSEMBLER  PAGE    4

 112                    *
 113                    * OUTPUT ONE CHARACTER IN A REGISTER
 114                    *
 115  F8EF FF E4 C3  OUTPUT  STX    SAVEX
 116  F8F2 36                PSH A
 117  F8F3 37                PSH B
 118  F8F4 4D                TST A
 119  F8F5 27 59             BEQ    OUTPT5
 120  F8F7 81 9F             CMP A  £$9F      FUNCTIONS ?
 121  F8F9 22 55             BHI    OUTPT5
 122  F8FB 81 1F             CMP A  £$1F      CONTROL ?
 123  F8FD 22 35             BHI    OUTPT3
 124  F8FF 81 07             CMP A  £7
 125  F901 26 04             BNE    OUTPT1
 126  F903 8D 5C             BSR    BIPBIP
 127  F905 20 49             BRA    OUTPT5
 128                    * CONTROL CHARACTERS
 129  F907 CE F8 B0  OUTPT1  LDX    £TABVIS
 130  F90A FF E4 C0          STX    PTTAB
 131  F90D BB E4 C1          ADD A  PTTAB+1
 132  F910 24 03             BCC    OUTPT5
 133  F912 7C E4 C0          INC    PTTAB
 134  F915 B7 E4 C1  OUTPT5  STA A  PTTAB+1
 135  F918 FE E4 C0          LDX    PTTAB
 136  F91B A6 00             LDA A  0,X
 137  F91D 81 0F             CMP A  £$F
 138  F91F 27 2F             BEQ    OUTPT5
 139  F921 B7 E8 10  OUTPT2  STA A  VIAVIS+ORB
 140  F924 8D 30             BSR    STROBE
 141  F926 84 07             AND A  £7
 142  F928 26 05             BNE    OUTPT6
 143                    * CLEAR SCREEN
 144  F92A CE 40 73          LDX    £$4073
 145  F92D 20 1E             BRA    OUTPT4
 146  F92F CE 04 0C  OUTPT6  LDX    £$40C
 147  F932 20 19             BRA    OUTPT4
 148                    * STANDARD CHARACTERS
 149  F934 C6 0F     OUTPT3  LDA B  £$F
 150  F936 F7 E8 10          STA B  VIAVIS+ORB
 151  F939 B7 E8 11          STA A  VIAVIS+ORA
 152  F93C 8D 18             BSR    STROBE
 153  F93E F6 E8 11  OUTPT8  LDA B  VIAVIS+ORA
 154  F941 59                ROL B
 155  F942 24 FA             BCC    OUTPT8
 156  F944 F6 E8 11  OUTPT7  LDA B  VIAVIS+ORA
 157  F947 59                ROL B
 158  F948 25 FA             BCS    OUTPT7
 159  F94A FE E4 C5          LDX    VTEMP
 160  F94D 09        OUTPT4  DEX
 161  F94E 26 FD             BNE    OUTPT4
 162  F950 33        OUTPT5  PUL B
 163  F951 32                PUL A
 164  F952 FE E4 C3          LDX    SAVEX
```

```
SCREEN HANDLER                    22-10-81  TSC ASSEMBLER  PAGE    5

 165  F955 39                RTS
 166                         *
 167                         *  OUTPUT STROBE
 168                         *
 169  F956 C6 FC     STROBE  LDA B  £$FC
 170  F958 F7 E8 1C          STA B  VIAVIS+PCR
 171  F95B C6 FE             LDA B  £$FE
 172  F95D F7 E8 1C          STA B  VIAVIS+PCR
 173  F960 39                RTS
 174                         *
 175                         * OUTPUT BELL
 176                         *
 177  F961 86 7F     BIPBIP  LDA A  £$7F
 178  F963 8D 0D     BIPSN1  BSR    BIPDEL
 179  F965 53                COM B
 180  F966 F7 E8 61          STA B  VIAMUS+ORA
 181  F969 8D 07             BSR    BIPDEL
 182  F96B F7 E8 61          STA B  VIAMUS+ORA
 183  F96E 4A                DEC A
 184  F96F 26 F2             BNE    BIPSN1
 185  F971 39                RTS
 186  F972 C6 60     BIPDEL  LDA B  £$60
 187  F974 5A        BIPSN2  DEC B
 188  F975 26 FD             BNE    BIPSN2
 189  F977 39                RTS
 190                         END
```

```
SCREEN HANDLER                  22-10-81  TSC ASSEMBLER  PAGE    6


 192  F980                ORG    $F980
 194                      NAM    KEYBOARD HANDLER
 195                      OPT    PAG
 196
 197                ***********************************
 198                *                                 *
 199                *         KEYBOARD HANDLER        *
 200                *                                 *
 201                ***********************************
 202
 203
 204                *
 205                *     I/O ROUTINES
 206                *     -----------
 207                *
 208                *        INITCV    INITIALIZATIONS
 209                *        TINPUT    CHECK FOR TYPED CHARACTER
 210                *        INPUT     INPUT ONE CHARACTER
 211                *
 212                *
 213                * EQUATES
 214                *
 215  E840          VIACL   EQU    $E840
 216  E820          DATA1   EQU    $E820
 217  E830          DATA2   EQU    $E830
 218  E821          CMDE1   EQU    $E821
 219  E831          CMDE2   EQU    $E831
 220  E860          VIAMUS  EQU    $E860
 221  F403          GTEXT   EQU    $F403
 222  EBF8          SETMEM  EQU    $EBF8
 223  F400          GINIT   EQU    $F400
 224  0000          ORB     EQU    0
 225  0001          ORA     EQU    1
 226  0002          DDRB    EQU    2
 227  0003          DDRA    EQU    3
 228  0004          TILL    EQU    4
 229  000C          PCR     EQU    $C
 230  000D          IFR     EQU    $D
 231  000E          IER     EQU    $E
 232  00A1          SHIFT   EQU    $A1       SHIFT CODE
 233  00A0          REPT    EQU    $A0       REPEAT CODE
```

```
235                     *
236                     * KEYBOARD 1 TABLE DEFINITION
237                     *
238  F980               ADTAB1  EQU     *
239  F980 1A                    FCB     $1A,$04,$05,$13,$86,$83,$85,$18
240  F988 12                    FCB     $12,$07,$14,$06,$88,$02,$87,$16
241  F990 19                    FCB     $19,$0A,$15,$08,$8A,$9D,$89,$0E
242  F998 09                    FCB     $09,$11,$01,$00,$84,$17,$F1,$A1
243  F9A0 09                    FCB     $09,$0C,$0F,$0B,$8C,$9B,$8B,$9E
244  F9A8 0A                    FCB     $0A,$0D,$00,$7F,$08,$00,$1F,$20
245  F9B0 10                    FCB     $10,$1C,$1D,$0D,$9C,$00,$00,$1E
246  F9B8 00                    FCB     $00,$A0,$00,$00,$00,$00,$00,$00
247
248  F9C0 1A                    FCB     $1A,$04,$05,$13,$86,$83,$85,$18
249  F9C8 12                    FCB     $12,$07,$14,$06,$88,$02,$87,$16
250  F9D0 19                    FCB     $19,$0A,$15,$08,$8A,$9D,$89,$0E
251  F9D8 09                    FCB     $09,$11,$01,$00,$84,$17,$F1,$A1
252  F9E0 09                    FCB     $09,$0C,$0F,$0B,$8C,$9B,$8B,$9E
253  F9E8 0A                    FCB     $0A,$0D,$00,$7F,$08,$00,$1F,$20
254  F9F0 10                    FCB     $10,$1C,$1D,$0D,$9C,$00,$00,$1E
255  F9F8 00                    FCB     $00,$A0,$00,$00,$00,$00,$00,$00
256
257  FA00 5A                    FCB     $5A,$44,$45,$53,$33,$43,$32,$58
258  FA08 52                    FCB     $52,$47,$54,$46,$35,$42,$34,$56
259  FA10 59                    FCB     $59,$4A,$55,$48,$37,$3F,$36,$4E
260  FA18 09                    FCB     $09,$51,$41,$00,$31,$57,$F1,$A1
261  FA20 49                    FCB     $49,$4C,$4F,$4B,$39,$2F,$38,$2E
262  FA28 0A                    FCB     $0A,$0D,$00,$7F,$08,$00,$5F,$20
263  FA30 50                    FCB     $50,$25,$7E,$4D,$81,$00,$30,$2B
264  FA38 00                    FCB     $00,$A0,$00,$00,$00,$00,$00,$00
265
266  FA40 7A                    FCB     $7A,$64,$65,$73,$22,$63,$7B,$78
267  FA48 72                    FCB     $72,$67,$74,$66,$28,$62,$27,$76
268  FA50 79                    FCB     $79,$6A,$75,$68,$7D,$2C,$83,$6E
269  FA58 09                    FCB     $09,$71,$61,$00,$26,$77,$F1,$A1
270  FA60 69                    FCB     $69,$6C,$6F,$6B,$82,$3A,$21,$3B
271  FA68 0A                    FCB     $0A,$0D,$00,$7F,$08,$00,$2D,$20
272  FA70 70                    FCB     $70,$7C,$5E,$6D,$29,$00,$40,$3D
273  FA78 00                    FCB     $00,$A0,$00,$00,$00,$00,$00,$00
274                     *
275                     * KEYBOARD 2 TABLE DEFINITION
276                     *
277  FA80               ADTAB2  EQU     *
278  FA80 24                    FCB     $24,$F2,$0B,$F7,$0C,$00,$0A,$FE
279  FA88 00                    FCB     $00,$00,$00,$F5,$08,$00,$00,$00
280  FA90 5C                    FCB     $5C,$80,$00,$00,$00,$F4,$3D,$00
281  FA98 3C                    FCB     $3C,$00,$31,$34,$30,$F3,$2E,$37
282  FAA0 25                    FCB     $25,$3E,$2D,$2A,$2B,$00,$00,$2F
283  FAA8 5B                    FCB     $5B,$2F,$00,$F9,$1D,$00,$00,$F8
284  FAB0 2A                    FCB     $2A,$5D,$33,$36,$00,$00,$00,$39
285  FAB8 23                    FCB     $23,$1B,$32,$35,$00,$00,$2C,$38
```

```
KEYBOARD HANDLER                22-10-81  TSC ASSEMBLER  PAGE    8

                        * I/O INITIALIZATIONS
                        ************
                        *
FAC0 BD F4 03           TEXT    JSR    GTEXT     ALPHA OUTPUT ON GRAPHIC
FAC3 7E FD 12                   JMP    CONTRL
                        *
                        *
FAC6 BD F8 D0           INIT    JSR    INITVS    INITIALIZE SCREEN
FAC9 BD F4 00                   JSR    GINIT     INITIALIZE GRAPHIC
FACC 86 01                      LDA A  £1
FACE B7 EB F8                   STA A  SETMEM
                        *
                        * VIA FOR BELL
FAD1 86 7F              INITCV  LDA A  £$7F
FAD3 B7 E8 6E                   STA A  VIAMUS+IER
FAD6 86 FF                      LDA A  £$FF
FAD8 B7 E8 6C                   STA A  VIAMUS+PCR
FADB B7 E8 63                   STA A  VIAMUS+DDRA
                        *
                        * 8279
                        *
                        * ENCODED KEYBOARD N-KEY ROLLOVER
                        *
FADE 86 02              INIT82  LDA A  £2
FAE0 B7 E8 21                   STA A  CMDE1
FAE3 B7 E8 31                   STA A  CMDE2
FAE6 86 C1                      LDA A  £$C1
FAE8 B7 E8 21                   STA A  CMDE1
FAEB B7 E8 31                   STA A  CMDE2
FAEE 86 2F                      LDA A  £$2F
FAF0 B7 E8 21                   STA A  CMDE1
FAF3 B7 E8 31                   STA A  CMDE2
FAF6 86 A0                      LDA A  £$A0
FAF8 B7 E8 21                   STA A  CMDE1
                        *
                        * PARALLEL I/O PORT
FAFB 4F                 INITVC  CLR A
FAFC B7 E8 43                   STA A  VIACL+DDRA
FAFF 43                         COM A
FB00 B7 E8 42                   STA A  VIACL+DDRB
FB03 86 C3                      LDA A  £$C3
FB05 B7 E8 4C                   STA A  VIACL+PCR
FB08 86 E3                      LDA A  £$E3
FB0A B7 E8 4C                   STA A  VIACL+PCR
                        *
                        * WAIT FOR KEYBOARD
FB0D C6 0D                      LDA B  £13
FB0F 86 C2              STABI   LDA A  £$C2
FB11 B7 E8 21                   STA A  CMDE1
FB14 B7 E8 31                   STA A  CMDE2
FB17 CE 10 00                   LDX    £$1000
FB1A 09                 STAB1   DEX
FB1B 26 FD                      BNE    STAB1
```

```
340  FB1D 5A              DEC B
341  FB1E 26 EF           BNE    STABI
342  FB20 39              RTS
```



```
344                  *
345                  * GET THE CHARACTER TYPED ON KEYBOARD
346                  *
347  FB21 B6 E4 C2   INPUT  LDA A  SAVEA    GET CHARACTER
348  FB24 7F E4 13          CLR    FLGIN    CLEAR FLAG
349  FB27 39                RTS             RETURN
350                  *
351                  * REPEAT PROCESSING
352                  *
353  FB28 C6 47      REPTA  LDA B  £$47     READ FIFO SENSOR RAM AT ROW 7
354  FB2A F7 E8 21          STA B  CMDE1
355  FB2D CE 03 00          LDX    £$300    DELAY AFTER COMMAND
356  FB30 09         REPTA4 DEX
357  FB31 26 FD             BNE    REPTA4
358  FB33 F6 E8 20          LDA B  DATA1    GET DATA
359  FB36 C4 02             AND B  £2       CHECK FOR REPEAT CODE
360  FB38 27 0B             BEQ    REPTA2   NO MORE REPEAT
361  FB3A CE 20 00          LDX    £$2000   DELAY
362  FB3D 09         REPTA1 DEX
363  FB3E 26 FD             BNE    REPTA1
364  FB40 73 E4 13          COM    FLGIN    SET INPUT FLAG
365  FB43 20 08             BRA    REPTA3
366  FB45 C6 02      REPTA2 LDA B  £2       SET ENCODED SCAN KEYBOARD
367  FB47 F7 E8 21          STA B  CMDE1
368  FB4A 7F E4 12          CLR    FLGRPT   CLEAR REPEAT FLAG
369  FB4D C6 C1      REPTA3 LDA B  £$C1     CLEAR INTEL 8279
370  FB4F F7 E8 21          STA B  CMDE1
371  FB52 20 11             BRA    TINPT1   RETURN
372                  *
373                  * CHECK IF A CHARACTER HAS BEEN TYPED
374                  *
375  FB54 7D E4 13   TINPUT TST    FLGIN    CHARACTER ALREADY PRESENT
376  FB57 26 14             BNE    TINPT2   YES RETURN
377  FB59 FF E4 C3          STX    SAVEX    SAVE X
378  FB5C 36                PSH A           SAVE A
379  FB5D 37                PSH B           SAVE B
380  FB5E 7D E4 12          TST    FLGRPT   REPEAT ?
381  FB61 26 C5             BNE    REPTA    YES GO PROCESS REPEAT
382  FB63 8D 09             BSR    KEYB     CHECK AND READ KEYBOARD
383  FB65 FE E4 C3   TINPT1 LDX    SAVEX    RESTORE X
384  FB68 7D E4 13          TST    FLGIN    SET Z
385  FB6B 33                PUL B           RESTORE B
386  FB6C 32                PUL A           RESTORE A
387  FB6D 39         TINPT2 RTS             RETURN
388                  *
389                  * CHECK AND READ KEYBOARD
390                  *
391  FB6E B6 E8 21   KEYB   LDA A  CMDE1     CHECK KEYBOARD 1
392  FB71 84 0F             AND A  £$F
393  FB73 26 11             BNE    KEYB1    WE HAVE CHAR.
394  FB75 B6 E8 31          LDA A  CMDE2     CHECK KEYBOARD 2
395  FB78 84 0F             AND A  £$F
396  FB7A 27 38             BEQ    KEYB4    NOTHING, RETURN
```

```
KEYBOARD HANDLER                    22-10-81  TSC ASSEMBLER  PAGE   11

 397  FB7C B6 E8 30          LDA A  DATA2     GET CHAR.
 398  FB7F CE FA 80          LDX    £ADTAB2   ADDRESS OF TABLE 2
 399  FB82 84 3F             AND A  £$3F      ONLY 6 BITS
 400  FB84 20 0D             BRA    KEYB2
 401  FB86 B6 E8 20  KEYB1   LDA A  DATA1     GET CHAR.
 402  FB89 CE F9 80          LDX    £ADTAB1   ADDRESS OF TABLE 1
 403  FB8C 7D E4 11          TST    FLGSHF    CHECK SHIFT LOCK
 404  FB8F 26 02             BNE    KEYB2
 405  FB91 84 BF             AND A  £$BF      IF LOCKED,MASK THE BIT
 406  FB93 FF E4 C0  KEYB2   STX    PTTAB
 407  FB96 BB E4 C1          ADD A  PTTAB+1   ADDRESS OF CHAR. IN THE TABLE
 408  FB99 24 03             BCC    KEYB3
 409  FB9B 7C E4 C0          INC    PTTAB     ADD CARRY IF ANY
 410  FB9E B7 E4 C1  KEYB3   STA A  PTTAB+1
 411  FBA1 FE E4 C0          LDX    PTTAB     GET ADDRESS OF CHAR.
 412  FBA4 A6 00             LDA A  0,X       GET CODE
 413  FBA6 B7 E4 C2          STA A  SAVEA     SAVE IT
 414  FBA9 81 A1             CMP A  £SHIFT    SHIFT KEY ?
 415  FBAB 27 08             BEQ    GSHIFT    YES GO PROCESS IT
 416  FBAD 81 A0             CMP A  £REPT     REPEAT KEY ?
 417  FBAF 27 15             BEQ    GREPT     IF YES GO PROCESS IT
 418  FBB1 73 E4 13          COM    FLGIN     SET INPUT FLAG
 419  FBB4 39        KEYB4   RTS              RETURN
 420                 *
 421                 * SHIFT LOCK PROCESSING
 422                 *
 423  FBB5 7D E4 11  GSHIFT  TST    FLGSHF    CHECK FLAG
 424  FBB8 27 03             BEQ    GSHFT1    SKIP IF LOCKED
 425  FBBA 86 A0             LDA A  £$A0      LIGHT ON
 426  FBBC 8C                FCB    SKIP2
 427  FBBD 86 A3     GSHFT1  LDA A  £$A3      LIGHT OFF
 428  FBBF B7 E8 21          STA A  CMDE1
 429  FBC2 73 E4 11          COM    FLGSHF    SWITCH THE FLAG
 430  FBC5 39                RTS              RETURN
 431                 *
 432                 * FIRST TIME REPEAT HAS BEEN TYPED
 433                 *
 434  FBC6 8D A6     GREPT   BSR    KEYB      CHECK AND READ KEYBOARD
 435  FBC8 7D E4 13          TST    FLGIN     CHAR. TYPED ?
 436  FBCB 27 F9             BEQ    GREPT     NO,WAIT
 437  FBCD C6 04             LDA B  £4        YES SET ENCODED SCAN SENSOR MATRIX
 438  FBCF F7 E8 21          STA B  CMDE1
 439  FBD2 86 FF             LDA A  £$FF      SET REPEAT FLAG
 440  FBD4 B7 E4 12          STA A  FLGRPT
 441  FBD7 39                RTS              RETURN
 442                 *
 443                 * RAM RESERVED TO VARIABLE INFORMATIONS
 444                 *
 445  E411                   ORG    $E411     INITIALIZED TO ZERO AT START
 446  E411           FLGSHF  RMB    1
 447  E412           FLGRPT  RMB    1
 448  E413           FLGIN   RMB    1
 449  E4C0                   ORG    $E4C0
 450  E4C0           PTTAB   RMB    2

KEYBOARD HANDLER                    22-10-81  TSC ASSEMBLER  PAGE   12

 451  E4C2           SAVEA   RMB    1
 452  E4C3           SAVEX   RMB    2
 453  E4C5           VTEMP   RMB    2
 454
 455                        END
```

```
 457  FBD8                    ORG    $FBD8
 459                          NAM    DISK BOOTSTRAP
 460                          OPT    PAG
 461
 462                  ********************************
 463                  *                              *
 464                  * -S- COMMAND                  *
 465                  *   DISK BOOTSTRAP (5")        *
 466                  *                              *
 467                  ********************************
 468
 469                  * EQUATES FOR WD 1791
 470
 471  E8E0            DRVREG  EQU    $E8E0     DRIVE REGISTER
 472  E8F0            COMREG  EQU    $E8F0     COMMAND REGISTER
 473  E8F2            SECREG  EQU    $E8F2     SECTOR REGISTER
 474  E8F3            DATREG  EQU    $E8F3     DATA REGISTER
 475  A100            LOADER  EQU    $A100
 476
 477                  * PROGRAM STARTS HERE
 478
 479  FBD8 B6 E8 F0   DKBOOT  LDA A  COMREG    TURN MOTOR ON
 480  FBDB 7F E8 E0           CLR    DRVREG    SELECT DRIVE £0
 481  FBDE CE 00 00           LDX    £0
 482  FBE1 08         OVR     INX              DELAY FOR MOTOR SPEEDUP
 483  FBE2 09                 DEX
 484  FBE3 09                 DEX
 485  FBE4 26 FB              BNE    OVR
 486  FBE6 C6 F4              LDA B  £$F4      DO RESTORE COMMAND
 487  FBE8 F7 E8 F0           STA B  COMREG
 488  FBEB 8D 2F              BSR    DELAY
 489  FBED F6 E8 F0   LOOP1   LDA B  COMREG    CHECK WD STATUS
 490  FBF0 53                 COM B
 491  FBF1 C5 01              BIT B  £1        WAIT TILL NOT BUSY
 492  FBF3 26 F8              BNE    LOOP1
 493  FBF5 86 FE              LDA A  £$FE      SETUP FOR SECTOR £1
 494  FBF7 B7 E8 F2           STA A  SECREG
 495  FBFA 8D 20              BSR    DELAY
 496  FBFC 86 73              LDA A  £$73      SETUP READ COMMAND
 497  FBFE B7 E8 F0           STA A  COMREG
 498  FC01 8D 19              BSR    DELAY
 499  FC03 CE A1 00           LDX    £LOADER   ADDRESS OF LOADER
 500  FC06 C5 02      LOOP2   BIT B  £2        DTA PRESENT ?
 501  FC08 27 07              BEQ    LOOP3     SKIP IF NOT
 502  FC0A B6 E8 F3           LDA A  DATREG    GET A BYTE
 503  FC0D 43                 COM A
 504  FC0E A7 00              STA A  0,X       PUT IN MEMORY
 505  FC10 08                 INX              BUMP POINTER
 506  FC11 F6 E8 F0   LOOP3   LDA B  COMREG    CHECK WD STATUS
 507  FC14 53                 COM B
 508  FC15 C5 01              BIT B  £1        IS WD BUSY ?
 509  FC17 26 ED              BNE    LOOP2     LOOP IF SO
 510  FC19 7E A1 00           JMP    LOADER    JUMP TO FLEX LOADER
```



```
 511
 512  FC1C 8D 00      DELAY   BSR    DEL1
 513  FC1E 8D 00      DEL1    BSR    RTN
 514  FC20 39         RTN     RTS
 515
 516                          END
```

```
DISK BOOTSTRAP                    22-10-81  TSC ASSEMBLER  PAGE   15


                        NAM    GPMON V1.3
                        OPT    PAG
FC25                    ORG    $FC25

                ************************
                *                      *
                *    TERMINAL MODE     *
                *                      *
                ************************

                *
                * EQUATES
                *
E80C            ACIAC   EQU    $E80C     ACIA COMMAND REGISTER
E80D            ACIAD   EQU    ACIAC+1   ACIA DATA REGISTER
E810            VIAC    EQU    $E810     CPU VIA FOR ACIA EXTERNAL CLOCK
000B            ACR     EQU    $B        VIA AUXILIARY CONTROL REGISTER
0004            T1LL    EQU    4         VIA T1 LOW ORDER LATCH
                *
                * INITIALIZE ACIA (CPU)
                *    A:=ACIA COMMAND
                *    X:=CLOCK VIA
                *
FC25 C6 03      INICIA  LDA B  £3        MASTER RESET CODE
FC27 F7 E8 0C           STA B  ACIAC     RESET ACIA
FC2A B7 E8 0C           STA A  ACIAC     INITIALIZE ACIA
FC2D 86 C0              LDA A  £$C0      INZ CLOCK FOR ACIA
FC2F B7 E8 1B           STA A  VIAC+ACR
FC32 FF E8 14           STX    VIAC+T1LL INITIALIZE CLOCK
FC35 39                 RTS              RETURN
                *
                * OUTPUT ONE CHARACTER TO ACIA
                *
FC36 37         TOACIA  PSH B            SAVE B-REG
FC37 F6 E8 0C   OUTC1   LDA B  ACIAC     TEST READY
FC3A 57                 ASR B
FC3B 57                 ASR B
FC3C 24 F9              BCC    OUTC1     XMIT NOT READY
FC3E B7 E8 0D           STA A  ACIAD     OUTPUT CHAR.
FC41 33                 PUL B            RESTORE B-REG
FC42 39                 RTS              RETURN
                *
                * INPUT ONE CHARACTER FROM ACIA
                *
FC43 8D 00      INACIA  BSR    CHKACI    CHECK FOR RECEIVED CHAR.
FC45 27 FC              BEQ    INACIA    NOT READY
FC47 B6 E8 0D           LDA A  ACIAD     INPUT CHAR.
FC4A 84 7F              AND A  £$7F      RESET PARITY BIT
FC4C 39                 RTS              RETURN
                *
                * CHECK FOR TYPED CHARACTER
                *
FC4D B6 E8 0C   CHKACI  LDA A  ACIAC     READ STATUS
```

```
GPMON V1.3                          22-10-81  TSC ASSEMBLER  PAGE   16

 572  FC50 85 01           BIT A  £1        SHOW READY OR NOT READY
 573  FC52 39              RTS              RETURN
 574                  *
 575                  * PROGRAM STARTS HERE
 576                  *
 577  FC53 BD FD CC  COM     JSR    PCRLF
 578  FC56 CE FC 8C          LDX    £QUEST    ASK FOR SPEED
 579  FC59 BD FD 91          JSR    PDATA1
 580  FC5C BD FD 98          JSR    INCH      INPUT SPEED
 581  FC5F 16                TAB              SAVE IT IN B
 582  FC60 BD FD CC          JSR    PCRLF     OUTPUT CR/LF
 583  FC63 C1 31             CMP B  £$31
 584  FC65 27 09             BEQ    COM1      300BDS
 585  FC67 C1 32             CMP B  £$32
 586  FC69 26 E8             BNE    COM       NOT CORRECT
 587  FC6B CE 18 00          LDX    £$1800    1200 BDS
 588  FC6E 20 03             BRA    COM2
 589  FC70 CE 64 00  COM1    LDX    £$6400    300 BDS
 590  FC73 86 01     COM2    LDA A  £1        EVEN PARITY,2 STOP BITS,CLOCK/16
 591  FC75 8D AE             BSR    INICIA    INITIALIZE ACIA
 592                  *
 593  FC77 BD FB 54  TERM    JSR    CHKCHR    CHECK FOR TYPED CHAR.
 594  FC7A 27 05             BEQ    TERM1     NO CHAR.
 595  FC7C BD FB 21          JSR    INPUT     INPUT CHAR.
 596  FC7F 8D 85             BSR    TOACIA    XMIT CHAR.
 597                  *
 598  FC81 8D CA     TERM1   BSR    CHKACI    CHECK FOR RECEIVED CHAR.
 599  FC83 27 F2             BEQ    TERM      NO CHAR.
 600  FC85 8D BC             BSR    INACIA    GET RECEIVED CHAR.
 601  FC87 BD FD 82          JSR    OUTCH     OUTPUT IT
 602  FC8A 20 EB             BRA    TERM      CONTINUE
 603                  *
 604  FC8C 56        QUEST   FCC    'V='
 605  FC8E 04                FCB    4
```

```
GPMON V1.3                          22-10-81  TSC ASSEMBLER  PAGE   17

 607                 ********************************
 608                 *                              *
 609                 *    PARALLEL PRINTER HANDLER  *
 610                 *                              *
 611                 ********************************
 612
 613
 614                 *
 615                 * OUTPUT ONE CHARACTER TO PRINTER
 616                 *
 617  FC8F 37        POUT    PSH B           SAVE B
 618  FC90 F6 E8 4D  CPOUT   LDA B  VIACL+IFR WAIT FOR READY
 619  FC93 C5 10             BIT B  £$10
 620  FC95 27 F9             BEQ    CPOUT
 621  FC97 B7 E8 40          STA A  VIACL+ORB OUTPUT CHAR.
 622  FC9A C6 C3             LDA B  £$C3
 623  FC9C F7 E8 4C          STA B  VIACL+PCR HANDSHAKE
 624  FC9F C6 E3             LDA B  £$E3
 625  FCA1 F7 E8 4C          STA B  VIACL+PCR
 626  FCA4 33                PUL B           RESTORE B
 627  FCA5 39                RTS             RETURN


GPMON V1.3                          22-10-81  TSC ASSEMBLER  PAGE   18

 629                 ************************
 630                 *                      *
 631                 *      MONITOR         *
 632                 *                      *
 633                 ************************
 634
 635
 636                 *
 637                 * EQUATES
 638                 *
 639  FCA6           ROM     EQU    $FCA6
 640  E400           RAM     EQU    $E400
 641  003F           SWI     EQU    $3F       SWI OP CODE
 642  0030           SZRCL   EQU    SP-NIO-2  SIZE OF RAM TO CLEAR
 643  000D           CR      EQU    $D        CARRIAGE RETURN
 644  000A           LF      EQU    $A        LINE FEED
 645  002B           PRPT    EQU    '+        PROMPT CHARACTER
 646  002E           DOWN    EQU    '.        MEMORY INCREMENT CHAR.
 647  002D           UP      EQU    '-        MEMORY DECREMENT CHAR.
 648                 *
 649  EC00           K7MOD   EQU    $EC00     K7/MODEM HANDLER ADDRESS
 650  8000           BASIC   EQU    $8000     ROM BASIC EXECUTION ADDRESS
 651  FB54           CHKCHR  EQU    TINPUT
```

```
GPMON V1.3                          22-10-81  TSC ASSEMBLER  PAGE   19


 653  FCA6                 ORG    ROM
 654                  *
 655                  * JUMP TABLE TO ROUTINES PERFORMING GPMON FUNCTIONS
 656                  *
 657  FCA6            FCTABL EQU    *
 658  FCA6 4D                FCC    /M/ -M- MEMORY CHANGE
 659  FCA7 FD ED             FDB    CHANGE
 660  FCA9 47                FCC    /G/ -G- GOTO ENTERED ADDRESS
 661  FCAA FE 37             FDB    GOTO
 662  FCAC 52                FCC    /R/ -R- PRINT STACK
 663  FCAD FF 1E             FDB    PSTAK
 664  FCAF 56                FCC    /V/ -V- SET A BREAKPOINT
 665  FCB0 FE 2A             FDB    SETBRK
 666  FCB2 55                FCC    /U/ -U- RESET A BREAKPOINT
 667  FCB3 FE 1E             FDB    RSTBRK
 668  FCB5 57                FCC    /W/ -W- DELETE ALL BREAKPOINTS
 669  FCB6 FE 26             FDB    DELBRK
 670  FCB8 43                FCC    /C/ -C- CONTINUE
 671  FCB9 FE 43             FDB    CONT
 672  FCBB 4E                FCC    /N/ -N- NEXT (TRACE 1 INSTRUCTION)
 673  FCBC FE 4B             FDB    NEXT
 674  FCBE 54                FCC    /T/ -T- TRACE N INSTRUCTIONS
 675  FCBF FE 4F             FDB    TRACE
 676  FCC1 42                FCC    /B/ -B- PRINT ALL BREAKS
 677  FCC2 FE 23             FDB    PNTBRK
 678  FCC4 44                FCC    /D/ -D- MEMORY DUMP
 679  FCC5 FE 65             FDB    DUMP
 680  FCC7 F3                FCB    $F3       -BOOT- EXECUTE DISK BOOTSTRAP
 681  FCC8 FB D8             FDB    DKBOOT
 682  FCCA 4B                FCC    /K/ -K- LINK TO K7/MODEM HANDLER
 683  FCCB EC 00             FDB    K7MOD
 684  FCCD 4F                FCC    /O/ -O- SWITCH OUTPUT TO PRINTER
 685  FCCE FE 8C             FDB    SWITCH
 686  FCD0 F1                FCB    $F1       -BASIC- EXECUTE ROM BASIC
 687  FCD1 80 00             FDB    BASIC
 688  FCD3 F2                FCB    $F2       -COM- TERMINAL MODE
 689  FCD4 FC 53             FDB    COM
 690  FCD6 F4                FCB    $F4       -GR- GRAPHIC MODE
 691  FCD7 FA C0             FDB    TEXT
 692  FCD9            FCTBEN EQU    *
```

```
GPMON V1.3                    22-10-81  TSC ASSEMBLER  PAGE   20

                        *
                        * I/O INTERRUPT SEQUENCE
                        *
FCD9 FE E4 00  IO       LDX    IOV       GET IRQ VECTOR
FCDC 6E 00             JMP    X
                        *
                        * NMI SEQUENCE
                        *
FCDE FE E4 02  POWDWN   LDX    NIO       GET NMI VECTOR
FCE1 6E 00             JMP    X
                        *
                        * SWI INTERRUPT SEQUENCE
                        *
FCE3 FE E4 36  SFEI     LDX    SWI1      GET SWI VECTOR
FCE6 6E 00             JMP    X
                        *
                        * INITIALIZATION/RESET CODE
                        *
FCE8           ADRSTR   EQU    *
FCE8 E4 BE             FDB    STACK-7   INIT FOR "SP"
FCEA FF 06             FDB    SWI1S     INIT FOR "SWI1"
                        *
FCEC 20 03             BRA    BRG       "BRA" INST IS REPLACED BY COND BRA
FCEE 7E FF 5A          JMP    BRNOGO    INST IN ROUTINE WHICH DETERMINES
FCF1 7E FF 81  BRG      JMP    BRGO      IF BRA IS GO/NOGO
FCF3           ADREND   EQU    *-1
                        *
                        * CONSTANT INITIALIZATION
                        *
FCF4           START    EQU    *
FCF4 8E E4 3F          LDS    £STKEND-1 S:POINTER TO RAM
FCF7 CE FC F3          LDX    £ADREND   X:POINTER TO ROM
FCFA A6 00     START1  LDA A  X         GET NEXT CONSTANT BYTE
FCFC 36                PSH A           INIT NEXT RAM BYTE
FCFD 09                DEX
FCFE 8C FC E7          CPX    £ADRSTR-1 END OF CONSTANT ROM AREA
FD01 26 F7             BNE    START1    NO CONTINUE
                        *
                        * INITIALIZATION TO 0
                        *
FD03 85                FCB    SKIP1,$13 VERSION NUMBER AUGU,17 1981
FD05 C6 30             LDA B  £SZRCL    SIZE OF RAM TO CLEAR
FD07 4F                CLR A
FD08 36        START2  PSH A           CLEAR NEXT RAM LOCATION
FD09 5A                DEC B           ANYMORE BYTES TO INIT ?
FD0A 26 FC             BNE    START2    YES CONTINUE
                        *
                        * INITIALIZE I/O
                        *
FD0C BE E4 34          LDS    SP
FD0F BD FA C6          JSR    INIT      INITIALIZE I/O
```

```
GPMON V1.3                    22-10-81  TSC ASSEMBLER  PAGE   21


 746                    *
 747                    * MAIN COMMAND/CONTROL LOOP
 748                    *
 749  FD12              CONTRL  EQU   *
 750  FD12 BE E4 34             LDS   SP        RESTORE STACK POINTER
 751  FD15 CE FF 06             LDX   £SWI1S    RESTORE SWI VECTOR
 752  FD18 FF E4 36             STX   SWI1
 753  FD1B BD FD D4             JSR   PROMPT    PRINT PROMPT
 754  FD1E 7F E4 08             CLR   OUTSW     MAKE SURE INPUT IS ECHOED
 755  FD21 BD FD 98             JSR   INCH      READ COMMAND CHARACTER
 756  FD24 16                   TAB             SAVE IT IN B
 757  FD25 BD FD B3             JSR   OUTS      PRINT SPACE AFTER COMMAND
 758                    *
 759                    * B REGISTER HOLDS CHARACTER INPUT BY USER
 760                    * USE JUMP TABLE TO GO TO APPROPRIATE ROUTINE
 761                    *
 762  FD28 CE FC A6             LDX   £FCTABL   X:= ADDRESS OF JUMP TABLE
 763  FD2B E1 00        NXTCHR  CMP B 0,X       DOES INPUT CHAR. MATCH ?
 764  FD2D 27 0F                BEQ   GOODCH    YES,GO TO APPROPRIATE ROUTINE
 765  FD2F 08                   INX             NO UPDATE INDEX
 766  FD30 08                   INX
 767  FD31 08                   INX
 768  FD32 8C FC D9             CPX   £FCTBEN   END OF TABLE REACHED ?
 769  FD35 26 F4                BNE   NXTCHR    NO,TRY NEXT COMMAND
 770  FD37 B6 3F        WHAT    LDA A £'?       NO MATCH PRINT "?"
 771  FD39 BD FD 82             JSR   OUTCH
 772  FD3C 20 D4                BRA   CONTRL    REPROMPT USER
 773                    *
 774  FD3E EE 01        GOODCH  LDX   1,X       GET ADDRESS FROM J.T.
 775  FD40 6E 00                JMP   0,X       GO TO APPROPRIATE ROUTINE
```

```
777  FD42                ORG    $FD42
778                  *
779                  * BASIC ROUTINES
780                  *
781
782                  *
783                  * INPUT HEX CHARACTER
784                  *
785  FD42 8D 54      INHEX  BSR    INCH
786  FD44 80 30      INHEX2 SUB A  £$30
787  FD46 2B CA             BMI    CONTRL    NOT HEX
788  FD48 81 09             CMP A  £$9
789  FD4A 2F 0A             BLE    IN1HG
790  FD4C 81 11             CMP A  £$11
791  FD4E 2B C2             BMI    CONTRL    NOT HEX
792  FD50 81 16             CMP A  £$16
793  FD52 2E BE             BGT    CONTRL    NOT HEX
794  FD54 80 07             SUB A  £7
795  FD56 39         IN1HG  RTS
796                  *
797                  * BUILD ADDRESS
798                  *
799  FD57 8D 10      BADDR  BSR    BYTE      READ 2 CHAR.
800  FD59 B7 E4 0A          STA A  XHI
801  FD5C 8D 0B             BSR    BYTE      READ 2 CHAR.
802  FD5E B7 E4 0B          STA A  XLOW
803  FD61 FE E4 0A          LDX    XHI       X:= ADDRESS WE BUILT
804  FD64 39                RTS
805  FD65 8D F0      BADDRS BSR    BADDR     BUILD ADDRESS
806  FD67 20 4A             BRA    OUTS      PRINT SPACE AND RETURN
807                  *
808                  * INPUT BYTE
809                  *
810  FD69 8D D7      BYTE   BSR    INHEX     GET HEX CHAR.
811  FD6B 48         BYTE2  ASL A
812  FD6C 48                ASL A
813  FD6D 48                ASL A
814  FD6E 48                ASL A
815  FD6F 16                TAB
816  FD70 8D D0             BSR    INHEX     GET HEX CHAR.
817  FD72 1B                ABA
818  FD73 39                RTS
819                  *
820                  * OUT HEXA BCD DIGIT
821                  *
822  FD74 44         OUTHL  LSR A             OUT HEX LEFT BCD DIGIT
823  FD75 44                LSR A
824  FD76 44                LSR A
825  FD77 44                LSR A
826  FD78 84 0F      OUTHR  AND A  £$F        OUT HEX RIGHT BCD DIGIT
827  FD7A 8B 30             ADD A  £$30
828  FD7C 81 39             CMP A  £$39
829  FD7E 23 02             BLS    OUTCH
```

```
GPMON V1.3                      22-10-81  TSC ASSEMBLER  PAGE   23

 830  FD80 8B 07           ADD A  £7
 831                  *
 832                  * OUTPUT ONE CHARACTER
 833                  *
 834  FD82 BD F8 EF  OUTCH  JSR    OUTPUT   OUTPUT CHAR. TO SCREEN
 835  FD85 7D E4 09         TST    PRTFLG   TO PRINTER ?
 836  FD88 26 01            BNE    POUTS    IF YES PRINT CHAR.
 837  FD8A 39               RTS             RETURN
 838  FD8B 7E FC 8F  POUTS  JMP    POUT
 839                  *
 840                  * PRINT DATA POINTED BY X-REG
 841                  *
 842  FD8E BD F2     PDATA2 BSR    OUTCH    OUTPUT DATA
 843  FD90 08        PDATA3 INX
 844  FD91 A6 00     PDATA1 LDA A  X
 845  FD93 81 04            CMP A  £4       STOP IF EOT
 846  FD95 26 F7            BNE    PDATA2
 847  FD97 39               RTS             RETURN
 848                  *
 849                  * INPUT ONE CHARACTER
 850                  *
 851  FD98 BD FB 54  INCH   JSR    TINPUT   CHECK FOR TYPED CHAR
 852  FD9B 27 FB            BEQ    INCH     NOT READY
 853  FD9D BD FB 21         JSR    INPUT    INPUT CHAR.
 854  FDA0 7D E4 08         TST    OUTSW    SHOULD INPUT BE ECHOED?
 855  FDA3 27 DD            BEQ    OUTCH    IF SO ,OUTPUT CHAR.
 856  FDA5 39               RTS
 857                  *
 858                  * OUTPUT BYTE
 859                  *
 860  FDA6 A6 00     OUT2H  LDA A  0,X       OUTPUT 2 HEX CHAR.
 861  FDA8 08               INX
 862  FDA9 36        OUT2HA PSH A
 863  FDAA 8D C8            BSR    OUTHL    OUT LEFT HEX CHAR.
 864  FDAC 32               PUL A
 865  FDAD 20 C9            BRA    OUTHR    OUTPUT RIGHT HEX CHAR AND RTS
 866                  *
 867                  * OUTPUT ADDRESS
 868                  *
 869  FDAF 8D F5     OUT4HS BSR    OUT2H    OUTPUT 4 HEX CHAR.+SPACE
 870  FDB1 8D F3     OUT2HS BSR    OUT2H    OUTPUT 2 HEX CHAR.+SPACE
 871  FDB3 86 20     OUTS   LDA A  £$20      OUTPUT SPACE
 872  FDB5 20 CB            BRA    OUTCH    (BSR+RTS)
 873                  *
 874                  * PRINT STACK CONTENTS
 875                  *
 876  FDB7 8D 13     PRINT  BSR    PCRLF    OUTPUT CR/LF
 877  FDB9 FE E4 34         LDX    SP       PRINT OUT STACK
 878  FDBC 08               INX
 879  FDBD 8D F2            BSR    OUT2HS   CC-REG
 880  FDBF 8D F0            BSR    OUT2HS   ACC-B
 881  FDC1 8D EE            BSR    OUT2HS   ACC-A
 882  FDC3 8D EA            BSR    OUT4HS   X-REG
 883  FDC5 8D E8            BSR    OUT4HS   P-COUNTER
```

```
GPMON V1.3                          22-10-81  TSC ASSEMBLER  PAGE   24

 884  FDC7 CE E4 34          LDX    £SP
 885  FDCA 20 E3             BRA    OUT4HS    PRINT SP AND RETURN
 886                    *
 887                    * OUTPUT CR/LF
 888                    *
 889  FDCC 86 0D        PCRLF LDA A  £CR       OUTPUT CR
 890  FDCE 8D B2             BSR    OUTCH
 891  FDD0 86 0A             LDA A  £LF        OUTPUT LF
 892  FDD2 20 AE             BRA    OUTCH     AND RETURN
 893                    *
 894                    * OUTPUT PROMPT
 895                    *
 896  FDD4 8D F6        PROMPT BSR    PCRLF     DO CR/LF
 897  FDD6 86 2B             LDA A  £PRPT
 898  FDD8 20 A8             BRA    OUTCH     OUTPUT PROMPT AND RETURN
 899                    *
 900                    * ERROR : VALUE IN RAM DO NOT MATCH
 901                    *
 902  FDDA 16           ERROR  TAB               SAVE A
 903  FDDB 8D EF             BSR    PCRLF
 904  FDDD CE E4 0A          LDX    £XHI
 905  FDE0 8D CD             BSR    OUT4HS    PRINT ADDRESS
 906  FDE2 FE E4 0A          LDX    XHI
 907  FDE5 8D CA             BSR    OUT2HS    PRINT CONTENT
 908  FDE7 17                TBA               RESTORE A
 909  FDE8 8D BF             BSR    OUT2HA    PRINT VALUE WE WANT TO STORE
 910  FDEA 7E FD 37          JMP    WHAT      PRINT "?"
```

```
                        *
                        * -M- COMMAND
                        *   MEMORY EXAMINE AND CHANGE
                        *
FDED 8D 73      CHANGE  BSR     BADDRJ    BUILD ADDRESS
FDEF 8D DB              BSR     PCRLF     DO CR/LF
FDF1 CE E4 0A   CHANG   LDX     #XHI
FDF4 8D B9              BSR     OUT4HS    PRINT ADDRESS
FDF6 FE E4 0A           LDX     XHI
FDF9 8D B6              BSR     OUT2HS    PRINT OLD DATA
FDFB 09                 DEX
FDFC 8D 9A      CHA1    BSR     INCH      INPUT CHAR.
FDFE 81 2E              CMP A   #DOWN     DOWN ?
FE00 27 14              BEQ     BLF       YES
FE02 81 2D              CMP A   #UP       CHECK FOR UP
FE04 27 0E              BEQ     UA
FE06 BD FD 44           JSR     INHEX2    CHECK FIRST BYTE
FE09 BD FD 6B           JSR     BYTE2     GET NEW BYTE
FE0C A7 00              STA A   X         WRITE BYTE IN MEMORY
FE0E A1 00              CMP A   X         CHECK IF RAM
FE10 27 EA              BEQ     CHA1      GO INPUT NEXT CHAR.
FE12 20 C6              BRA     ERROR     NOT RAM
FE14 09         UA      DEX               DEC ADDR
FE15 85                 FCB     SKIP1     SKIP 1 BYTE
FE16 08         BLF     INX               INC ADDR
FE17 FF E4 0A   LF1     STX     XHI       SAVE ADDRESS
FE1A 8D B0              BSR     PCRLF     OUTPUT CR/LF
FE1C 20 D3              BRA     CHANG     CONTINUE
                        *
                        * -U- COMMAND
                        *   RESET ONE BREAKPOINT
                        *
FE1E 8D 42      RSTBRK  BSR     BADDRJ    PUT ADDRESS IN XHI,XLOW
FE20 86 09              LDA A   #BKFPG1   RESET 1 BREAK FLAG
FE22 8C                 FCB     SKIP2     SKIP 2 BYTES
                        *
                        * -B- COMMAND
                        *   PRINT OUT ALL NON-ZERO BREAK ADDRESSES
                        *
FE23 86 21      PNTBRK  LDA A   #BKFPRT   PRINT BREAK ADDR FLAG
FE25 8C                 FCB     SKIP2     SKIP 2 BYTES
                        *
                        * -W- COMMAND
                        *   RESET ALL BREAKPOINTS
                        *
FE26 86 11      DELBRK  LDA A   #BKFPGA   RESET BREAKS FLAG
FE28 20 08              BRA     SETB2
                        *
                        * -V- COMMAND
                        *   SET ONE BREAKPOINT
                        *
FE2A 8D 36      SETBRK  BSR     BADDRJ    PUT ADDRESS INTO XHI,XLOW
FE2C 86 09              LDA A   #BKFPG1
```

```
GPMON V1.3                      22-10-81  TSC ASSEMBLER  PAGE   26

  965  FE2E 8D 63           BSR    BRKSUB
  966  FE30 B6 05           LDA A  #BKFRCD   SET ONE BREAKPOINT FLAG
  967  FE32 8D 5F    SET82  BSR    BRKSUB    BREAK HANDLING SUBROUTINE
  968  FE34 7E FD 12 CTRLJ  JMP    CONTRL    RETURN TO COMMAND LEVEL
  969                *
  970                * -G- COMMAND
  971                *   GO TO ENTERED ADDRESS
  972                *
  973  FE37 8D 29    GOTO   BSR    BADDRJ    GET ADDR. FROM USER IN XHI,XLOW
  974  FE39 FE E4 34        LDX    SP        GET VALUE SAVED WHEN ENTERING GPMON
  975  FE3C A7 07           STA A  7,X       A-REG CONTAINS XLOW
  976  FE3E B6 E4 0A        LDA A  XHI
  977  FE41 A7 06           STA A  6,X       PLACE ADDRESS ON STACK
  978                *
  979                * -C- COMMAND
  980                *   CONTINUE EXECUTION
  981                *
  982  FE43 7C E4 10 CONT   INC    BRKTRC    TRACE 1 TO RESTORE SWI'S
  983  FE46 CE 00 01        LDX    #1
  984  FE49 20 09           BRA    TRACE3
  985                *
  986                * -N- COMMAND
  987                *   SINGLE INSTRUCTION TRACE REQUESTED
  988                *
  989  FE4B CE 00 01 NEXT   LDX    #1        #INSTRUCTION TO TRACE
  990  FE4E 8C              FCB    SKIP2     SKIP 2 BYTES
  991                *
  992                * -T- COMMAND
  993                *   MULTIPLE INSTRUCTION TRACE
  994                *
  995  FE4F 8D 11    TRACE  BSR    BADDRJ    GET N. OF INSTRUCTION TO TRACE
  996  FE51 7F E4 10 TRACE1 CLR    BRKTRC    CLEAR FLAG INDICATING TRACE IS DUE TO BREP
  997  FE54 FF E4 06 TRACE3 STX    NTRACE    SAVE # INST'S TO TRACE
  998  FE57 FE E4 34        LDX    SP        X:=STACK POINTER
  999  FE5A EE 06           LDX    6,X       X:=ADDR OF INSTR TO BE EXECUTED
 1000  FE5C BD FF 9A        JSR    SAVTRC    SAVE ADDR/OPCODE FOR TRACE
 1001  FE5F 7E FF 47        JMP    CONTRC    GOTO CONTINUE TRACE
 1002  FE62 7E FD 57 BADDRJ JMP    BADDR
 1003                *
 1004                * -D- COMMAND
 1005                *   MEMORY DUMP
 1006                *
 1007  FE65 BD FD 65 DUMP   JSR    BADDRS    GET START ADDRESS
 1008  FE68 FF E4 0C        STX    XSAVE     SAVE IT
 1009  FE6B 8D F5           BSR    BADDRJ    GET END ADDRESS IN XHI,XLOW
 1010  FE6D BD FD CC DUMP1  JSR    PCRLF     OUTPUT CR,LF
 1011  FE70 C6 10           LDA B  #16       16 BYTES PER LINE
 1012  FE72 CE E4 0C        LDX    #XSAVE    PRINT ADDRESS
 1013  FE75 BD FD AF        JSR    OUT4HS
 1014  FE78 FE E4 0C DUMP2  LDX    XSAVE     PRINT BYTE
 1015  FE7B BD FD B1        JSR    OUT2HS
 1016  FE7E FF E4 0C        STX    XSAVE     SAVE NEW ADDRESS
 1017  FE81 09              DEX
 1018  FE82 BC E4 0A        CPX    XHI       FINISHED ?
```

```
GPMON V1.3                    22-10-81  TSC ASSEMBLER  PAGE   27

1019  FE85 27 AD          BEQ    CTRLJ     YES,EXIT
1020  FE87 5A             DEC B            DEC THE COUNT
1021  FE88 26 EE          BNE    DUMP2     SAME LINE IF NOT 0
1022  FE8A 20 E1          BRA    DUMP1     NEW LINE
1023                *
1024                * -O- COMMAND
1025                *   SWITCH OUTPUT TO PRINTER
1026                *
1027  FE8C 73 E4 09 SWITCH  COM    PRTFLG    SWITCH FLAG
1028  FE8F 20 A3          BRA    CTRLJ     RETURN TO COMMAND LEVEL

GPMON V1.3                    22-10-81  TSC ASSEMBLER  PAGE   28

1030               *
1031               * SUBROUTINE TO INSERT BREAK IN USER CODE
1032               *
1033  FE91 86 82   INSBRK  LDA A  £BKFINS
1034               *
1035               * BRKSUB
1036               *
1037               * THIS ROUTINE HANDLES BREAKPOINT OPERATIONS
1038               * THE CONTENT OF A-REG DETERMINES THE FUNCTION PERFORMED:
1039               *
1040               *   A=BKFINS : PUT BREAKS INTO USER'S CODE
1041               *   A=BKFPG1 : PURGE BREAKPOINT WHOSE ADDRESS IS IN
1042               *              XHI,XLOW.ALL BREAKPOINTS ARE TEMPORARY REMOVED
1043               *   A=BKFPGA : PURGE ALL BREAKPOINTS
1044               *   A=BKFPRT : PRINT ALL BREAKS,TEMP. REMOVED
1045               *   A=BKFRMV : REMOVE TEMPORARY ALL BREAKPOINTS
1046               *   A=BKFRCD : PUT BREAK ADDRESS IN XHI,XLOW INTO THE
1047               *              FIRST ZERO BREAKPOINT POSITION;TEMP. REMOVED
1048               *
1049  0001         BKFRMV  EQU    %00000001
1050  0082         BKFINS  EQU    %10000010
1051  0005         BKFRCD  EQU    %00000101
1052  0009         BKFPG1  EQU    %00001001
1053  0011         BKFPGA  EQU    %00010001
1054  0021         BKFPRT  EQU    %00100001
1055               *
1056  FE93         BRKSUB  EQU    *
1057  FE93 36              PSH A            SAVE A-REG
1058  FE94 CE E4 1C        LDX    £BRKADR   X POINTS BREAK TABLE
1059  FE97 FF E4 0C BRKP0  STX    XSAVE     SAVE X
1060  FE9A 33              PUL B            GET FUNCTION NUMBER
1061  FE9B 37              PSH B            UPDATE STACK POINTER
1062  FE9C A6 02           LDA A  2,X       LOAD USER CODE SAVED IN BREAK TABLE
1063  FE9E EE 00           LDX    X         LOAD THE BREAKPOINTED USER ADDRESS
1064  FEA0 27 03           BEQ    BRKP01    IF ADDR=0 NOT A BREAKPOINT
1065  FEA2 C4 FB           AND B  £$FB      RECORD NOT ALLOWED ,MASK FLAG
1066  FEA4 8C              FCB    SKIP2     SKIP 2 BYTES
1067  FEA5 C4 C4     BRKP01  AND B  £$C4     RECORD ONLY ALLOWED
1068  FEA7 7D E4 0F  BRKP03  TST    BRKSIN    ARE BREAKS IN USER'CODE ?
1069  FEAA 27 03           BEQ    BRKP02    NO
1070  FEAC C4 FD           AND B  £$FD      YES,DO NOT WRITE SWI IN USER CODE
1071  FEAE 8C              FCB    SKIP2     SKIP 2 BYTES
1072  FEAF C4 FE     BRKP02  AND B  £$FE     AVOID THAT BREAKS BE TEMPORARY REMOVED
1073               *
1074               * REWRITE CODE IN USER PROGRAM FROM BREAKPOINT TABLE
1075               *
1076  FEB1 54        BRKP04  LSR B            PLACE FUNCTION FLAG IN CARRY
1077  FEB2 24 02           BCC    BRKP2     FUNCTION NOT REQUESTED
1078  FEB4 A7 00           STA A  X         REWRITE CODE IN USER PGM
1079               *
1080               * WRITE SWI'S IN USER CODE
1081               *
1082  FEB6 54        BRKP2   LSR B
```

```
1083  FEB7 24 0D          BCC    BRKP3
1084  FEB9 A6 00          LDA A  X          SAVE USER CODE
1085  FEBB 36             PSH A
1086  FEBC 86 3F          LDA A  £SWI
1087  FEBE A7 00          STA A  X          WRITE SWI
1088  FEC0 32             PUL A
1089  FEC1 FE E4 0C       LDX    XSAVE      ADDR OF BREAK IN BREAK TABLE
1090  FEC4 A7 02          STA A  2,X        SAVE USER CODE IN BREAK TABLE
1091                *
1092                * RECORD BREAKPOINT IN TABLE
1093                *
1094  FEC6 FE E4 0C BRKP3 LDX    XSAVE      GET ADDRESS OF BREAK IN TABLE
1095  FEC9 54             LSR B             NEXT FUNCTION
1096  FECA 24 0E          BCC    BRKP4
1097  FECC B6 E4 0A       LDA A  XHI        GET HI-BYTE OF NEW ADDRESS
1098  FECF A7 00          STA A  0,X        WRITE IN TABLE
1099  FED1 B6 E4 0B       LDA A  XLOW       GET LOW BYTE
1100  FED4 A7 01          STA A  1,X
1101  FED6 32             PUL A             GET FUNCTION NUMBER FROM STACK
1102  FED7 84 FB          AND A  £%11111011 DO NOT PLACE ADDRESS MORE THAN ONCE
1103  FED9 36             PSH A             CONTINUE TO TAKE OUT BREAKPOINTS
1104                *
1105                * PURGE BREAKPOINT WHOSE ADDRESS IS IN XHI,XLOW
1106                *
1107  FEDA 54           BRKP4  LSR B
1108  FEDB 24 0E          BCC    BRKP5
1109  FEDD A6 00          LDA A  X          GET HI-BYTE FROM TABLE
1110  FEDF B1 E4 0A       CMP A  XHI        COMPARE
1111  FEE2 26 07          BNE    BRKP5      NOT THIS BREAK TO BE PURGED
1112  FEE4 A6 01          LDA A  1,X        GET LOW BYTE
1113  FEE6 B1 E4 0B       CMP A  XLOW
1114  FEE9 27 03          BEQ    BRKP52     THIS BREAK IS TO BE PURGED
1115                *
1116                * PURGE ALL BREAKPOINTS
1117                *
1118  FEEB 54           BRKP5  LSR B
1119  FEEC 24 04          BCC    BRKP6
1120  FEEE 6F 00        BRKP52 CLR    0,X      CLEAR BREAKPOINT HIGH
1121  FEF0 6F 01          CLR    1,X        CLEAR LOW BYTE
1122                *
1123                * PRINT BREAKPOINT
1124                *
1125  FEF2 54           BRKP6  LSR B
1126  FEF3 24 04          BCC    BRKPE
1127  FEF5 BD FD AF       JSR    OUT4HS     PRINT BREAKPOINT ADDRESS
1128  FEF8 8C             FCB    SKIP2      SKIP 2 BYTES
1129                *
1130                * UPDATE LOOP INDEX AND LOOP IF APPROPRIATE
1131                *
1132  FEF9 08           BRKPE  INX
1133  FEFA 08             INX
1134  FEFB 08           BRKPE1 INX            X POINTS TO NEXT ADDR IN TABLE
1135  FEFC 8C E4 34       CPX    £BRKINS
1136  FEFF 26 96          BNE    BRKP0
```



```
1137                *
1138                * WRAP-UP PROCESSING AND EXIT
1139                *
1140  FF01 F7 E4 0F       STA B  BRKSIN     STORE APPROPRIATE FLAG
1141  FF04 32             PUL A             UPDATE STACK POINTER
1142  FF05 39             RTS               RETURN
```

```
GPMON V1.3                          22-10-81  TSC ASSEMBLER  PAGE   31


1144                    *
1145                    * SWI SOFTWARE INTERRUPT PROCESSING
1146                    *
1147  FF06              SWI1S   EQU     *
1148  FF06 BF E4 34             STS     SP          SAVE USER'S SP
1149  FF09 86 01                LDA A   #BKFRMV
1150  FF0B BD FE 93             JSR     BRKSUB      GO TAKE OUT ALL BREAKS
1151                    * DECREMENT P-COUNTER
1152  FF0E 30                   TSX                 X:=STACK POINTER-1
1153  FF0F 6D 06                TST     6,X         IF LOWER BYTE=0 : BORROW
1154  FF11 26 02                BNE     SWI1S1      BRANCH IF BORROW NOT REQUIRED
1155  FF13 6A 05                DEC     5,X         DECREMENT UPPER BYTE
1156  FF15 6A 06        SWI1S1  DEC     6,X         DECREMENT LOWER BYTE
1157                    * TEST FOR ADDRESS TRACE OR BREAK
1158  FF17 EE 05                LDX     5,X         X:=P-COUNTER
1159  FF19 BC E4 04             CPX     TRCADR      IS SWI FOR TRACE ?
1160  FF1C 27 06                BEQ     TRCINH      YES,GO TO TRACE INT HANDLER
1161                    *
1162                    * BREAK INTERRUPT HANDLER
1163                    *
1164                    *
1165                    * -R- COMMAND
1166                    *
1167  FF1E              PSTAK   EQU     *
1168  FF1E BD FD B7             JSR     PRINT       STOP AND SHOW REGISTER TO USER
1169  FF21 7E FD 12     CNTRL3  JMP     CONTRL      RETURN
1170                    *
1171                    * TRACE INTERRUPT HANDLER
1172                    * P-COUNTER HAS BEEN DECREMENTED TO POINT AT SWI
1173                    * TRCINS HOLDS OP CODE REPLACED BY SWI
1174                    * X HOLDS ADDRESS OF WHERE TRACE SWI IS
1175                    *
1176  FF24 B6 E4 0E     TRCINH  LDA A   TRCINS       GET OPCODE OF TRACED INSTR.
1177  FF27 A7 00                STA A   0,X         RESTORE TO USER'CODE
1178  FF29 7D E4 10             TST     BRKTRC      IS PROCESSING TO BE IMMEDIATELY CONTINUED ?
1179  FF2C 27 0D                BEQ     NBKTRC      BRANCH IF NOT
1180                    *
1181                    * PROCESSING IS TO "CONTINUE"
1182                    *
1183  FF2E 7F E4 10             CLR     BRKTRC      RESET CONTINUE FLAG
1184  FF31 BD FE 91             JSR     INSBRK      INSERT BREAKS IN USER CODE
1185  FF34 7F E4 04             CLR     TRCADR      NO MORE TRACE,SO CLEAR ADDRESS
1186  FF37 7F E4 05             CLR     TRCADR+1
1187  FF3A 3B                   RTI                 CONTINUE
1188                    *
1189                    * TRACE IS DUE TO N OR T TRACE COMMAND
1190                    *
1191  FF3B BD FD B7     NBKTRC  JSR     PRINT       PRINT STACK
1192  FF3E FE E4 06             LDX     NTRACE      GET # OF INSTRUCTIONS TO TRACE
1193  FF41 09                   DEX                 DECREMENT COUNT
1194  FF42 FF E4 06             STX     NTRACE      AND RESTORE
1195  FF45 27 DA                BEQ     CNTRL3      BRANCH IF ALL TRACES DONE
1196                    *
```

```
GPMON V1.3                          22-10-81  TSC ASSEMBLER  PAGE   32

1197                 * TRACE NOT DONE - TRACE NEXT INSTRUCTION
1198                 *
1199  FF47 B6 E4 0E  CONTRC  LDA A  TRCINS    GET CURRENT INSTRUCTION
1200  FF4A B7 E4 38          STA A  BRINS     SAVE IN CASE IT'S A BRANCH
1201  FF4D 8D 6F             BSR    OPCBYT   GO GET £ OF BYTES/TYPE
1202  FF4F 4D                TST A           CHECK FOR BRANCH
1203  FF50 2A 0A             BPL    CKOBRA   CHECK FOR OTHER THAN BRANCH
1204                 *
1205                 * RELATIVE BRANCH TYPE INSTRUCTION
1206                 * DETERMINE WHERE TO PUT SWI
1207                 * S HOLDS POINTER TO USER STACK AFTER SWI
1208                 *
1209  FF52 32                PUL A           GET CONDITION CODE
1210  FF53 34                DES             UPDATE STACK PTR AFTER PULL
1211  FF54 9A 10             ORA A  %00010000 MAKE INT'S INHIBITED
1212  FF56 06                TAP             RESTORE USER'S C CODE REG
1213  FF57 7E E4 38          JMP    BRINS    GO SEE HOW RELATIVE BRANCH FARES
1214                 *
1215                 * BRANCH WAS NOGO - PUT SWI AT NEXT INSTRUCTION
1216                 *
1217  FF5A B6 02     BRNOGO  LDA A  £2        A:=£ OF BYTES AFTER CURRENT INSTR.
1218                 *
1219                 * INSTRUCTION TO BE TRACED IS NOT A BRANCH
1220                 *
1221  FF5C FE E4 04  CKOBRA  LDX    TRCADR    X:=TRACE ADDRESS
1222  FF5F E6 00             LDA B  0,X       GET INSTR. TO BE TRACED
1223  FF61 C1 6E             CMP B  £$6E      IS IT A JUMP,INDEXED
1224  FF63 27 44             BEQ    JMPIDX    YES GO SIMULATE JUMP IDXED
1225  FF65 C1 7E             CMP B  £$7E      JUMP EXTENDED ?
1226  FF67 27 51             BEQ    JMPEXT
1227  FF69 C1 AD             CMP B  £$AD      JSR,INDEXED ?
1228  FF6B 27 3C             BEQ    JMPIDX    JUMP IDXED IS SAME AS TRANSFERT OF CONTROL
1229  FF6D C1 BD             CMP B  £$BD      JSR,EXTENDED ?
1230  FF6F 27 49             BEQ    JMPEXT
1231  FF71 C1 3B             CMP B  £$3B      RTI ?
1232  FF73 27 3B             BEQ    RTISIM
1233  FF75 C1 39             CMP B  £$39      RTS ?
1234  FF77 27 3C             BEQ    RTSSIM
1235  FF79 C1 3F             CMP B  £$3F      SWI ? IF SWI USER DON'T INCREMENT ADDR.
1236  FF7B 27 13             BEQ    BRG2
1237  FF7D C1 8D             CMP B  £$8D      BSR ? IF YES BRANCH PROCESSING
1238                 *
1239                 * NOT A BRANCH,JUMP,RTI,RTS
1240                 * A REGISTER HOLDS £ OF BYTES IN INSTRUCTION
1241                 *
1242  FF7F 26 0D             BNE    BRG1      PUT IN NEW SWI AND TRACE NEXT INSTR.
1243                 *
1244                 * BRANCH WAS GO,PUT SWI AT ADDRESS BEING JUMP TO
1245                 *
1246  FF81 FE E4 04  BRGO    LDX    TRCADR    X:=TRACE ADDRESS
1247  FF84 A6 01             LDA A  1,X       GET BRANCH OFFSET
1248  FF86 08                INX             OFFSET IS RELATIVE TO
1249  FF87 08                INX             INSTR FOLLOWING BRANCH
1250  FF88 2A 04             BPL    BRG1      BRANCH IS OFFSET POSITIVE
```

```
GPMON V1.3                      22-10-81  TSC ASSEMBLER  PAGE   33

 1251                 * X NEEDS TO BE DECREMENTED (OFFSET NEGATIVE)
 1252  FF8A 09        BRGODC  DEX             DECREMENT ADDRESS
 1253  FF8B 4C                INC A           INCREMENT COUNTER
 1254  FF8C 26 FC             BNE    BRGODC
 1255                 * CONTENT OF A IS 0
 1256  FF8E 8D 15     BRG1    BSR    INCX     INCREMENT X BY AMOUNT IN A-REG
 1257  FF90 8D 08     BRG2    BSR    SAVTRC   SAVE ADDR/OPC OF NEXT INSTR TO STOP ON
 1258  FF92 86 3F             LDA A  £SWI     REPLACE INSTR. WITH SWI
 1259  FF94 A7 00             STA A  X
 1260  FF96 BE E4 34  RTIPC   LDS    SP       GET ORIGINAL STACK POINTER
 1261  FF99 3B                RTI             TRACE ANOTHER INSTR.
 1262                 *
 1263                 * SUBROUTINE TO SAVE ADDR IN X INTO TRCADR AND OPC INTO TRCINS
 1264                 *
 1265  FF9A           SAVTRC  EQU    *
 1266  FF9A FF E4 04          STX    TRCADR
 1267  FF9D A6 00             LDA A  X
 1268  FF9F B7 E4 0E          STA A  TRCINS
 1269  FFA2 39                RTS
 1270                 *
 1271                 * SUBROUTINE TO INCREMENT X BY CONTENT OF A
 1272                 *
 1273  FFA3 08        INXLP   INX             INCREMENT X
 1274  FFA4 4A                DEC A           DECREMENT COUNT
 1275  FFA5 4D        INCX    TST A
 1276  FFA6 26 FB             BNE    INXLP    IF COUNT NOT YET 0 LOOP
 1277  FFA8 39                RTS             RETURN
 1278                 *
 1279                 * JUMP,JSR INDEXED SIMULATION
 1280                 *
 1281  FFA9 A6 01     JMPIDX  LDA A  1,X      A:=ADDR OFFSET
 1282  FFAB 30                TSX
 1283  FFAC EE 03             LDX    3,X      GET TARGET'S X REG
 1284  FFAE 20 DE             BRA    BRG1     UPDATEX TRACE NEXT INSTR
 1285                 *
 1286                 * RTI ENCOUNTERED
 1287                 *
 1288  FFB0 30        RTISIM  TSX
 1289  FFB1 EE 0C             LDX    12,X     GET P-COUNTER FROM STACK
 1290  FFB3 20 DB             BRA    BRG2     GO TRACE NEXT INSTR.
 1291                 *
 1292                 * RTS ENCOUNTERED
 1293                 *
 1294  FFB5 30        RTSSIM  TSX
 1295  FFB6 EE 07             LDX    7,X      GET RETURN P-REG FROM STACK
 1296  FFB8 20 D6             BRA    BRG2
 1297                 *
 1298                 * JUMP,JSR EXTENDED
 1299                 *
 1300  FFBA EE 01     JMPEXT  LDX    1,X
 1301  FFBC 20 D2             BRA    BRG2
```

```
1303                    *
1304                    * OPBCYT
1305                    *
1306                    * THIS ROUTINE DETERMINES THE # OF BYTES IN AN INSTRUCTION
1307                    * GIVEN ITS OP CODE
1308                    *
1309                    * INPUT : A HOLDS THE OP CODE
1310                    *
1311                    * OUTPUT : X HOLDS INDEX OF TABLE ELEMENT
1312                    * B NOT RESTORED
1313                    * A HOLDS # OF BYTES IN INSTRUCTION
1314                    * EXCEPT FOR BRANCHES IN WHICH CASE A IS NEGATIVE
1315                    *
1316 FFBE               OPCBYT  EQU     *
1317 FFBE 16                    TAB             B:=OP CODE
1318 FFBF 44                    LSR A
1319 FFC0 44                    LSR A
1320 FFC1 44                    LSR A           PUT 4 UPPER BITS OF OP CODE INTO
1321 FFC2 44                    LSR A           LOWER 4 BITS OF A
1322                    *
1323 FFC3 CE FF D6              LDX     #OPBTTB X:=ADDR OF TABLE
1324 FFC6 8D DD                BSR     INCX    INC X TO POINT TO CORRECT ENTRY
1325                    *
1326 FFC8 A6 00                LDA A   0,X     GET TABLE ENTRY
1327 FFCA 26 09                BNE     OPBTRT  IF NOT 0 THEN NO FURTHER PROCESSING NEEDED
1328                    *
1329                    * IF TOP 4 BITS=8 OR C ,THEN THERE ARE TWO CLASSE
1330                    * OF INSTRUCTIONS : 2 BYTE INSTRUCTIONS AND
1331                    * CE,8C AND EE WHICH ARE 3 BYTE INSTRUCTIONS
1332                    *
1333 FFCC 86 02                LDA A   #2      # OF BYTES IN MOST OF 8# INSTRUCTIONS
1334 FFCE C4 0D                AND B   #%00001101
1335 FFD0 C1 0C                CMP B   #%00001100
1336 FFD2 26 01                BNE     OPBTRT  NO ,RETURN
1337 FFD4 4C                   INC A           # OF BYTES IN INSTRUCTION :=3
1338 FFD5 39               OPBTRT  RTS             RETURN TO CALLER
1339                    *
1340                    * OP CODE TO NUMBER OF BYTES CONVERSION TABLE
1341                    *
1342                    *     # BYTES TOP 4 BITS OF OPCODE
1343                    *
1344 FFD6               OPBTTB  EQU     *
1345 FFD6 01                    FCB     1       0
1346 FFD7 01                    FCB     1       1
1347 FFD8 82                    FCB     2+%10000000 2  MINUS = BRANCHES
1348 FFD9 01                    FCB     1       3
1349 FFDA 01                    FCB     1       4
1350 FFDB 01                    FCB     1       5
1351 FFDC 02                    FCB     2       6
1352 FFDD 03                    FCB     3       7
1353 FFDE 00                    FCB     0       8 # BYTES=2 EXCEPT 8C,8E
1354 FFDF 02                    FCB     2       9
1355 FFE0 02                    FCB     2       A
```



```
1356 FFE1 03                    FCB     3       B
1357 FFE2 00                    FCB     0       C # BYTES=2 EXCEPT CE
1358 FFE3 02                    FCB     2       D
1359 FFE4 02                    FCB     2       E
1360 FFE5 03                    FCB     3       F
```

```
GPMON V1.3                          22-10-81  TSC ASSEMBLER  PAGE   36


1362                    *
1363                    * JUMP TABLE TO BASIC ROUTINES OF GPMON
1364                    *
1365  FFE6                      ORG    $FFE6
1366  FFE6 7E FD 12             JMP    CONTRL    WARM START ENTRY POINT
1367  FFE9 7E FB 54             JMP    CHKCHR    CHECK FOR TYPED CHARACTER
1368  FFEC 7E FD 82             JMP    OUTCH     OUTPUT ONE CHARACTER
1369  FFEF 7E FD 98             JMP    INCH      INPUT ONE CHARACTER
1370  FFF2 7E FD 91             JMP    PDATA1    PRINT STRING POINTED BY X-REG
1371  FFF5 7E FD CC             JMP    PCRLF     PRINT CR/LF
1372                    *
1373                    * INTERRUPTS VECTORS
1374                    *
1375  FFF8                      ORG    $FFF8
1376  FFF8 FC D9                FDB    IO        REGULAR INTERRUPT
1377  FFFA FC E3                FDB    SFEI      SOFTWARE INTERRUPT
1378  FFFC FC DE                FDB    POWDWN    NON MASKABLE INTERRUPT
1379  FFFE FC F4                FDB    START     RESET INTERRUPT
```

```
GPMON V1.3                      22-10-81  TSC ASSEMBLER  PAGE   37


1381                    *
1382                    * RAM - LOCATIONS DEVOTED TO VARIABLE INFORMATION
1383                    *
1384  E400                      ORG    RAM       START OF RAM
1385  0008              NBRBPT  EQU    8         £ OF BREAKPOINTS SUPPORTED
1386                    *
1387  E400              IOV     RMB    2         I/O INTERRUPT POINTER
1388  E402              NIO     RMB    2         NMI INTERRUPT POINTER
1389                    *
1390                    * THE FOLLOWING ARE INITIALIZED TO ZERO AT START
1391                    *
1392  E404              TRCADR  RMB    2         TRACE ADDRESS
1393  E406              NTRACE  RMB    2         NO. OF INSTRUCTIONS TO TRACE
1394  E408              OUTSW   RMB    1         OUTPUT SWITCH (0=ECHO INPUT)
1395  E409              PRTFLG  RMB    1         PRINT FLAG
1396  E40A              XHI     RMB    1         X REG HIGH (TEMP)
1397  E40B              XLOW    RMB    1         X REG LOW
1398                    *
1399  E40C              XSAVE   EQU    *
1400  E40C                      RMB    2
1401                    *
1402  E40E              TRCINS  RMB    1         OPCODE REPLACED BY SWI IN TRACE MODE
1403  E40F              BRKSIN  RMB    1         1=BREAKS ARE IN USER PGM
1404  E410              BRKTRC  RMB    1         1=P-COUNTER IS AT BREAKPOINT AND
1405                    *               USER WANTS TO CONTINUE - ONE TRACE WILL
1406                    *               BE DONE AND BREAKPOINTS RESTORED
1407  E411                      RMB    3         RESERVED FOR IO HANDLER
1408  E414                      RMB    8         FREE SPACE
1409                    *
1410  E41C              BRKADR  RMB    NBRBPT*3  BREAKPOINT ADDRESS TABLE
1411  E434              BRKINS  EQU    *
1412                    *
1413                    * THE FOLLOWING ARE INITIALIZED AT START
1414                    *
1415  E434              SP      RMB    2         USER STACK POINTER
1416  E436              SWI1    RMB    2         SWI VECTOR
1417  E438              BRINS   RMB    8         STORAGE FOR CONDITIONNAL BRANCH ROUTINE
1418  E440              BRANEN  EQU    *         END OF BRANCH ROUTINE+1
1419                    *
1420                    * STACK
1421                    *
1422  E440              STKEND  RMB    128       RAM USED FOR STACK
1423  E4BF              STACK   EQU    *-1       START OF STACK AREA
1424                    *
1425                    *
1426                    *
1427                            END    START
1428                            END

NO ERROR(S) DETECTED
```

GPMON V1.3 22-10-81 TSC ASSEMBLER PAGE 38

SYMBOL TABLE:

ACIAC	E80C	ACIAD	E80D	ACR	000B	ADREND	FCF3	ADRSTR	FCE8
ADTAB1	F980	ADTAB2	FA80	BADDR	FD57	BADDRJ	FE62	BADDRS	FD65
BASIC	8000	BIPBIP	F961	BIPDEL	F972	BIPSN1	F963	BIPSN2	F974
BKFINS	0082	BKFPG1	0009	BKFPGA	0011	BKFPRT	0021	BKFRCD	0005
BKFRMV	0001	BLF	FE16	BRANEN	E440	BRG	FCF1	BRG1	FF8E
BRG2	FF90	BRGD	FF81	BRGODC	FF8A	BRINS	E438	BRKADR	E41C
BRKINS	E434	BRKP0	FE97	BRKP01	FEA5	BRKP02	FEAF	BRKP03	FEA7
BRKP04	FEB1	BRKP2	FEB6	BRKP3	FEC6	BRKP4	FEDA	BRKP5	FEEB
BRKP52	FEEE	BRKP6	FEF2	BRKPE	FEF9	BRKPE1	FEFB	BRKSIN	E40F
BRKSUB	FE93	BRKTRC	E410	BRNOGO	FF5A	BYTE	FD69	BYTE2	FD6B
CHA1	FDFC	CHANG	FDF1	CHANGE	FDED	CHKACI	FC4D	CHKCHR	FB54
CKOBRA	FF5C	CMDE1	E821	CMDE2	E831	CNTRL3	FF21	COM	FC53
COM1	FC70	COM2	FC73	COMREG	E8F0	CONT	FE43	CONTRC	FF47
CONTRL	FD12	CPOUT	FC90	CR	000D	CTRLJ	FE34	DATA1	E820
DATA2	E830	DATREG	E8F3	DDRA	0003	DDRB	0002	DEL1	FC1E
DELAY	FC1C	DELBRK	FE26	DKBOOT	FBD8	DOWN	002E	DRVREG	E8E0
DUMP	FE65	DUMP1	FE6D	DUMP2	FE78	ERROR	FDDA	FCTABL	FCA6
FCTBEN	FCD9	FLGIN	E413	FLGRPT	E412	FLGSHF	E411	GINIT	F400
GOODCH	FD3E	GOTO	FE37	GREPT	FBC6	GSHFT1	FB8D	GSHIFT	FB85
GTEXT	F403	IER	000E	IFR	000D	IN1HG	FD56	INACIA	FC43
INCH	FD98	INCX	FFA5	INHEX	FD42	INHEX2	FD44	INICIA	FC25
INIT	FAC6	INIT82	FADE	INITCV	FAD1	INITVC	FAFB	INITVS	F8D0
INPUT	FB21	INSBRK	FE91	INXLP	FFA3	IO	FCD9	IOV	E400
JMPEXT	FFBA	JMPIDX	FFA9	K7MOD	EC00	KEYB	FB6E	KEYB1	FB86
KEYB2	FB93	KEYB3	FB9E	KEYB4	FBB4	LF	000A	LF1	FE17
LOADER	A100	LOOP1	FBED	LOOP2	FC06	LOOP3	FC11	NBKTRC	FF3B
NBRBPT	0008	NEXT	FE4B	NIO	E402	NTRACE	E406	NXTCHR	FD2B
OPBTRT	FFD5	OPBTTB	FFD6	OPCBYT	FFBE	ORA	0001	ORB	0000
OUT2H	FDA6	OUT2HA	FDA9	OUT2HS	FDB1	OUT4HS	FDAF	OUTC1	FC37
OUTCH	FD82	OUTHL	FD74	OUTHR	FD78	OUTPT1	F907	OUTPT2	F921
OUTPT3	F934	OUTPT4	F94D	OUTPT5	F950	OUTPT6	F92F	OUTPT7	F944
OUTPT8	F93E	OUTPTS	F915	OUTPUT	FBEF	OUTS	FDB3	OUTSW	E408
OVR	FBE1	PCR	000C	PCRLF	FDCC	PDATA1	FD91	PDATA2	FD8E
PDATA3	FD90	PNTBRK	FE23	POUT	FC8F	POUTS	FD8B	POWDWN	FCDE
PRINT	FDB7	PROMPT	FDD4	PRPT	0028	PRTFLG	E409	PSTAK	FF1E
PTTAB	E4C0	QUEST	FC8C	RAM	E400	REPT	00A0	REPTA	FB28
REPTA1	FB3D	REPTA2	FB45	REPTA3	FB4D	REPTA4	FB30	ROM	FCA6
RSTBRK	FE1E	RTIPC	FF96	RTISIM	FFB0	RTN	FC20	RTSSIM	FFB5
SAVEA	E4C2	SAVEX	E4C3	SAVTRC	FF9A	SECREG	E8F2	SETB2	FE32
SETBRK	FE2A	SETMEM	E8F8	SFEI	FCE3	SHIFT	00A1	SKIP1	0085
SKIP2	008C	SP	E434	STAB1	FB1A	STABI	FB0F	STACK	E4BF
START	FCF4	START1	FCFA	START2	FD08	STKEND	E440	STROBE	F956
SWI	003F	SWI1	E436	SWI1S	FF06	SWI1S1	FF15	SWITCH	FE8C
SZRCL	0030	T1LL	0004	TABVIS	F8B0	TERM	FC77	TERM1	FC81
TEXT	FAC0	TILL	0004	TINPT1	FB65	TINPT2	FB6D	TINPUT	FB54
TOACIA	FC36	TRACE	FE4F	TRACE1	FE51	TRACE3	FE54	TRCADR	E404
TRCINH	FF24	TRCINS	E40E	UA	FE14	UP	002D	VIAC	E810
VIACL	E840	VIAMUS	E860	VIAVIS	E810	VTEMP	E4C5	WHAT	FD37
XHI	E40A	XLOW	E40B	XSAVE	E40C				

```
1                       NAM   HANDLERS GRAPHIQUE/MUSIQUE
2                       OPT   PAG
3                 *
4                 * TABLE DE BRANCHEMENTS GRAPHIQUE/MUSIQUE
5                 *
6  F400                 ORG   $F400
7  F400 7E F4 20        JMP   GINIT     GR
8  F403 7E F4 15        JMP   GTEXT     TEXT
9  F406 7E F4 6A        JMP   GCOLOR    COLOR
10 F409 7E F4 75        JMP   GPLOT     PLOT
11 F40C 7E F7 56        JMP   SPTEMP    TEMPO
12 F40F 7E F7 27        JMP   SPNOTE    NOTE
13 F412 7E F4 51        JMP   GRAPH     INITIALIZE GRAPHIC
14                *
```

```
HANDLERS GRAPHIQUE/MUSIQUE          22-10-81  TSC ASSEMBLER  PAGE    1


                            NAM    GRAPHIQUE
                            OPT    PAG
                     *
                     * SOUS PROGRAMMES GRAPHIQUES
                     * POUR LA CARTE GRAPHIQUE COULEUR
                     * GOUPIL
                     *
                     * COPYRIGHT SMT 81
                     *
                     *
                     * POINTS D'ENTREE:
                     * -GINIT = INITIALISATION DU 6845
                     *          ET CREATION DU FOND COLORE
                     * -GCOLOR= DEFINITION DE LA COULEUR
                     *          DU FOND ET DU TRACE
                     * -GPLOT= TRACE DE POINTS ET DE LIGNES
                     * -GTEXT=COMMUTATION MODE TEXT
                     * -GRAPH=COMMUTATION MODE GRAPHIQUE
                     *          -0-0-0-0-0-0-
                     *


                     *
                     * ADRESSES DE LA CARTE GRAPHIQUE
                     *
E7F8                 ADVISU  EQU    $E7F8
E7FB                 GCURX   EQU    ADVISU+3
E7F9                 GCURY   EQU    ADVISU+1
E7FC                 GCONTR  EQU    ADVISU+4
E7FE                 GCRTC   EQU    ADVISU+6
                     *
                     * RESERVATIONS MEMOIRE RAM
                     *
E4C7                        ORG    $E4C7
E4C7                 GCUR    RMB    3        POINTEUR A INCREMENT NON ENTIER
E4CA                 GDELTA  RMB    3        INCREMENT NON ENTIER
E4CD                 GSD     RMB    1        SIGNE INCREMENT
E4CE                 GDX     RMB    1        DEPLACEMENT EN X
E4CF                 GDY     RMB    1        DEPLACEMENT EN Y
E4D0                 GSX     RMB    1        SIGNE DEPLACEMENT EN X
E4D1                 GSY     RMB    1        SIGNE DEPLACEMENT EN Y
E4D2                 GCMPT   RMB    1        NOMBRE DE PAS
E4D3                 DIRECT  RMB    1        COMMUTATION ECRAN
```

```
GRAPHIQUE                          22-10-81  TSC ASSEMBLER  PAGE    2

  60  F415                 ORG    $F415
  61                  *
  62                  *
  63                  * GTEXT
  64                  * ARGUMENTS D APPEL = AUCUN
  65                  * REGISTRE MODIFIE :A
  66                  *
  67  F415 86 80      GTEXT   LDA A  £$80
  68  F417 B7 E4 D3           STA A  DIRECT
  69  F41A 86 C0              LDA A  £$C0
  70  F41C B7 E7 FC           STA A  GCONTR
  71  F41F 39                 RTS
  72                  *
  73                  * GINIT
  74                  * ARGUMENTS D'APPEL=AUCUN
  75                  * REGISTRES MODIFIES=A,B
  76                  *
  77  F420 FF E4 CA   GINIT   STX    GDELTA
  78  F423 CE F4 5A           LDX    £GCRTTB   INITIALISATION 6845
  79  F426 5F                 CLR B
  80  F427 F7 E7 FE   GBCRT   STA B  GCRTC
  81  F42A A6 00              LDA A  0,X
  82  F42C B7 E7 FF           STA A  GCRTC+1
  83  F42F 08                 INX
  84  F430 5C                 INC B
  85  F431 C1 10              CMP B  £16
  86  F433 26 F2              BNE    GBCRT
  87  F435 B6 E7 FD           LDA A  GCONTR+1  DEFINITION DU FOND COLORE
  88  F438 88 40              EOR A  £$40
  89  F43A B7 E7 FD           STA A  GCONTR+1
  90  F43D 86 14              LDA A  £20       ATTENTE 20MS
  91  F43F 5F                 CLR B
  92  F440 5A         GBWAIT  DEC B
  93  F441 26 FD              BNE    GBWAIT
  94  F443 4A                 DEC A
  95  F444 26 FA              BNE    GBWAIT
  96  F446 B6 E7 FD           LDA A  GCONTR+1
  97  F449 88 40              EOR A  £$40
  98  F44B B7 E7 FD           STA A  GCONTR+1
  99  F44E FE E4 CA           LDX    GDELTA
 100                  *
 101                  * GRAPH
 102                  * ARGUMENT D APPEL = AUCUN
 103                  * REGISTRE MODIFIE : A
 104                  *
 105  F451 7F E4 D3   GRAPH   CLR    DIRECT
 106  F454 86 40              LDA A  £$40
 107  F456 B7 E7 FC           STA A  GCONTR
 108  F459 39                 RTS
 109                  *TABLE D'INITIALISATION DU 6845
 110  F45A 55         GCRTTB  FCB    $55,64,69,5,$4D,0,$40,$44
 111  F462 00                 FCB    0,3,0,0,0,0,0,0
 112                  *
```

```
GRAPHIQUE                          22-10-81  TSC ASSEMBLER  PAGE    3

  113                 * GCOLOR
  114                 *
  115                 * ARGUMENTS D'APPEL=(A)CODE DE COULEUR
  116                 * REGISTRES MODIFIES=A
  117                 *
  118  F46A 84 07     GCOLOR  AND A  £$07
  119  F46C 8A 40             ORA A  £$40
  120  F46E BA E4 D3          ORA A  DIRECT
  121  F471 B7 E7 FD          STA A  GCONTR+1
  122  F474 39                RTS
```

```
GRAPHIQUE                          22-10-81  TSC ASSEMBLER  PAGE    4

  124                 *
  125                 * GPLOT
  126                 *
  127                 * ARGUMENTS D'APPEL=X-)ZONE DE PARAMETRES
  128                 * SOUS-PROGRAMMES APPELES=GMOVE,GTRACE
  129                 * REGISTRES MODIFIES=A,B
  130                 *
  131  F475 A6 00     GPLOT   LDA A  0,X
  132  F477 81 00             CMP A  £0
  133  F479 26 06             BNE    GCMD1
  134  F47B A6 01     GCMD0   LDA A  1,X
  135  F47D E6 02             LDA B  2,X
  136  F47F 20 1B             BRA    GMOVE
  137  F481 81 01     GCMD1   CMP A  £1
  138  F483 26 06             BNE    GCMD2
  139  F485 A6 01             LDA A  1,X
  140  F487 E6 02             LDA B  2,X
  141  F489 20 1E             BRA    GTRACE
  142  F48B 81 02     GCMD2   CMP A  £2
  143  F48D 26 0C             BNE    GCMDE
  144  F48F A6 01             LDA A  1,X
  145  F491 E6 02             LDA B  2,X
  146  F493 8D 07             BSR    GMOVE
  147  F495 A6 03             LDA A  3,X
  148  F497 E6 04             LDA B  4,X
  149  F499 20 0E             BRA    GTRACE
  150  F49B 39        GCMDE   RTS
```

GRAPHIQUE 22-10-81 TSC ASSEMBLER PAGE 5

```
152                 *
153                 * GMOVE   POSITIONNEMENT SUR UN POINT
154                 *         ET ECRITURE
155                 *
156                 * ARGUMENTS D'APPEL=(A)ABCISSE,(B)ORDONNEE
157                 * REGISTRES MODIFIES=A
158                 *
159  F49C B7 E7 FB  GMOVE   STA A  GCURX
160  F49F F7 E7 F9          STA B  GCURY
161  F4A2 B6 E7 FD          LDA A  GCONTR+1
162  F4A5 B7 E7 FC          STA A  GCONTR
163  F4A8 39                RTS
164                 *
165                 * GTRACE   TRACE D'UNE LIGNE DROITE
166                 *
167                 * ARGUMENTS D'APPEL=(A)ABSCISSE,(B)ORDONNEE
168                 * SOUS-PROGRAMMES APPELES=GCINC,GAINC
169                 * REGISTRES MODIFIES=A,B
170                 *
171  F4A9 B0 E7 FB  GTRACE  SUB A  GCURX     CALCUL DU DEPLACEMENT
172  F4AC 22 07             BHI    GSUPX     SUIVANT X ET SON SIGNE
173  F4AE 26 0C             BNE    GINFX
174  F4B0 B7 E4 CE          STA A  GDX
175  F4B3 20 0D             BRA    GMVY
176  F4B5 B7 E4 CE  GSUPX   STA A  GDX
177  F4B8 86 01             LDA A  #1
178  F4BA 20 06             BRA    GMVY
179  F4BC 40        GINFX   NEG A
180  F4BD B7 E4 CE          STA A  GDX
181  F4C0 86 FF             LDA A  #-1
182  F4C2 B7 E4 D0  GMVY    STA A  GSX
183  F4C5 F0 E7 F9          SUB B  GCURY     CALCUL DU DEPLACEMENT
184  F4C8 22 07             BHI    GSUPY     SUIVANT Y ET SON SIGNE
185  F4CA 26 0C             BNE    GINFY
186  F4CC F7 E4 CF          STA B  GDY
187  F4CF 20 0D             BRA    GCMPD
188  F4D1 F7 E4 CF  GSUPY   STA B  GDY
189  F4D4 C6 01             LDA B  #1
190  F4D6 20 06             BRA    GCMPD
191  F4D8 50        GINFY   NEG B
192  F4D9 F7 E4 CF          STA B  GDY
193  F4DC C6 FF             LDA B  #-1
194  F4DE F7 E4 D1  GCMPD   STA B  GSY
195  F4E1 B6 E4 CE          LDA A  GDX       COMPARAISON DES DEPLACEMENTS
196  F4E4 B1 E4 CF          CMP A  GDY
197  F4E7 22 21             BHI    GPASX
198  F4E9 26 54             BNE    GPASY
199  F4EB 4D                TST A
200  F4EC 27 1B             BEQ    GPRET     ILS SONT NULS:RIEN A FAIRE
201  F4EE F6 E7 FB  GBPLT1  LDA B  GCURX     ILS SONT EGAUX:INCREMENTS
202  F4F1 FB E4 D0          ADD B  GSX       ENTIER GSX ET GSY
203  F4F4 F7 E7 FB          STA B  GCURX
204  F4F7 F6 E7 F9          LDA B  GCURY
```

```
GRAPHIQUE                              22-10-81  TSC ASSEMBLER  PAGE    6

 205  F4FA FB E4 D1          ADD B  GSY
 206  F4FD F7 E7 F9          STA B  GCURY
 207  F500 F6 E7 FD          LDA B  GCONTR+1
 208  F503 F7 E7 FC          STA B  GCONTR
 209  F506 4A                DEC A
 210  F507 26 E5             BNE    GBPLT1
 211  F509 39         GPRET  RTS
 212  F50A B6 E4 CE  GPASX   LDA A  GDX        DEPLACEMENT PLUS GRAND
 213  F50D F6 E4 CF          LDA B  GDY        SUIVANT X:INCREMENT ENTIER POUR X
 214  F510 27 DC             BEQ    GBPLT1     CALCUL DE L'INCREMENT POUR Y
 215  F512 F7 E4 CA          STA B  GDELTA
 216  F515 F6 E7 F9          LDA B  GCURY
 217  F518 F7 E4 C7          STA B  GCUR
 218  F51B F6 E4 D1          LDA B  GSY
 219  F51E F7 E4 CD          STA B  GSD
 220  F521 BD 50             BSR    GCINC
 221  F523 BD F5 B6  GBPLT2  JSR    GAINC      TRACE
 222  F526 F6 E7 FB          LDA B  GCURX
 223  F529 FB E4 D0          ADD B  GSX
 224  F52C F7 E7 FB          STA B  GCURX
 225  F52F F6 E4 C7          LDA B  GCUR
 226  F532 F7 E7 F9          STA B  GCURY
 227  F535 F6 E7 FD          LDA B  GCONTR+1
 228  F538 F7 E7 FC          STA B  GCONTR
 229  F53B 4A                DEC A
 230  F53C 26 E5             BNE    GBPLT2
 231  F53E 39                RTS
 232  F53F B6 E4 CF  GPASY   LDA A  GDY        DEPLACEMENT PLUS GRAND
 233  F542 F6 E4 CE          LDA B  GDX        SUIVANT Y:INCREMENT ENTIER POUR Y
 234  F545 27 A7             BEQ    GBPLT1     CALCUL DE L'INCREMENT POUR X
 235  F547 F7 E4 CA          STA B  GDELTA
 236  F54A F6 E7 FB          LDA B  GCURX
 237  F54D F7 E4 C7          STA B  GCUR
 238  F550 F6 E4 D0          LDA B  GSX
 239  F553 F7 E4 CD          STA B  GSD
 240  F556 BD 1B             BSR    GCINC
 241  F558 BD 5C     GBPLT3  BSR    GAINC      TRACE
 242  F55A F6 E4 C7          LDA B  GCUR
 243  F55D F7 E7 FB          STA B  GCURX
 244  F560 F6 E7 F9          LDA B  GCURY
 245  F563 FB E4 D1          ADD B  GSY
 246  F566 F7 E7 F9          STA B  GCURY
 247  F569 F6 E7 FD          LDA B  GCONTR+1
 248  F56C F7 E7 FC          STA B  GCONTR
 249  F56F 4A                DEC A
 250  F570 26 E6             BNE    GBPLT3
 251  F572 39                RTS
```

```
                        *
                        * GCINC   CALCUL DES INCREMENTS FRACTIONNAIRES
                        *
                        * ARGUMENTS D'APPEL=(A)NOMBRE DE PAS
                        *                   (GDELTA)DEPLACEMENT
                        *                  (GSD)SIGNE
                        * REGISTRES MODIFIES=9
                        *
F573 36         GCINC   PSH A
F574 C6 80              LDA B  £$80
F576 F7 E4 C8           STA B  GCUR+1
F579 7F E4 C9           CLR    GCUR+2
F57C C6 10              LDA B  £16
F57E F7 E4 D2           STA B  GCMPT
F581 16                 TAB
F582 B6 E4 CA           LDA A  GDELTA
F585 78 E4 CC   GBDIV   ASL    GDELTA+2  DIVISION DE GDELTA
F588 79 E4 CB           ROL    GDELTA+1
F58B 48                 ASL A             PAR LE NOMBRE DE PAS
F58C 25 03              BCS    GDSUB
F58E 11                 CBA
F58F 25 04              BCS    GFBDIV
F591 10         GDSUB   SBA
F592 7C E4 CC           INC    GDELTA+2
F595 7A E4 D2   GFBDIV  DEC    GCMPT
F598 26 EB              BNE    GBDIV
F59A 7F E4 CA           CLR    GDELTA    MULTIPLICATION PAR SON SIGNE
F59D 7D E4 CD           TST    GSD
F5A0 2A 12              BPL    GRCINC
F5A2 4F                 CLR A
F5A3 B0 E4 CC           SUB A  GDELTA+2
F5A6 B7 E4 CC           STA A  GDELTA+2
F5A9 86 00              LDA A  £0
F5AB B2 E4 CB           SBC A  GDELTA+1
F5AE B7 E4 CB           STA A  GDELTA+1
F5B1 7A E4 CA           DEC    GDELTA
F5B4 32         GRCINC  PUL A
F5B5 39                 RTS
                        *
                        * GAINC    ADDITION DE L'INCREMENT
                        *
                        * PARAMETRES D'APPEL=AUCUN
                        * REGISTRES MODIFIES=B
                        *
F5B6 F6 E4 C9   GAINC   LDA B  GCUR+2
F5B9 FB E4 CC           ADD B  GDELTA+2
F5BC F7 E4 C9           STA B  GCUR+2
F5BF F6 E4 C8           LDA B  GCUR+1
F5C2 F9 E4 CB           ADC B  GDELTA+1
F5C5 F7 E4 CB           STA B  GCUR+1
F5C8 F6 E4 C7           LDA B  GCUR
F5CB F9 E4 CA           ADC B  GDELTA
F5CE F7 E4 C7           STA B  GCUR
```



```
F5D1 39                 RTS
                        END
```

```
GRAPHIQUE                          22-10-81  TSC ASSEMBLER  PAGE    9


 310                       NAM    MUSIBA4 * 12.FEV.81
 311                       OPT    PAG
 312  E4D4                 ORG    $E4D4
 313                   *
 314                   * RESERVATIONS MEMOIRE RAM
 315                   *
 316  E4D4             POKTIM  RMB    1
 317  E4D5             POKCOR  RMB    1
 318  E4D6             TEMPO   RMB    1
 319  E4D7             COMMUN  RMB    2
 320  E4D9             EXPLOR  RMB    2
 321  E4DB             DELPUL  RMB    1
 322  E4DC             DELREP  RMB    1
 323  E4DD             DEL2    RMB    2
 324  E4DF             DEL8L   RMB    2
 325  E4E1             DEREPB  RMB    1
 326  E4E2             NBPER   RMB    1
 327  E4E3             NBPERB  RMB    1
 328  E4E4             NPH     RMB    1
 329  E4E5             NPB     RMB    1
 330  E4E6             FRAC    RMB    1
 331  E4E7             STROBE  RMB    1
 332  E4E8             NIVEAU  RMB    1
 333  E4E9             NIVB    RMB    1
 334  E4EA             DECLOG  RMB    1
 335  E4EB             DECPAS  RMB    1
 336  E4EC             SILENC  RMB    1
 337  E4ED             TIMJEU  RMB    1
 338  E4EE             GARX    RMB    2
 339  E4F0             BY1     RMB    1
 340  E4F1             BY2     RMB    1
 341  E4F2             BY3     RMB    1
 342  E4F3             BY4     RMB    1
 343  E4F4             BY5     RMB    1
 344                   ***********
 345  E861             LESON   EQU    $E861
 346                   ********************
 347  F5D3                     ORG    $F5D3
 348  F5D3 46          TABNOT  FCB    70,67,63,60,57,54,51,48,45,43,41,39,37
 349  F5E0 23                  FCB    35,33,31,29,27,26,24,23,21,20,19,18
 350  F5EC 11                  FCB    17,16,15,14,13,12,11,10,9,9,8,7
 351  F5F8 07                  FCB    7,6,5,5,4,4,3,3,3,2,2,1,0
 352  F605 12                  FCB    18,16,15,14,13,12,12,11,10,9,9,9,7
 353  F612 07                  FCB    7,6,6,5,5,5,4,4,4,3,3,3
 354  F61E 02                  FCB    2,2,2,2,1,1,1,1,1,1,1,1
 355  F62A 00                  FCB    0,0,0,0,0,0,0,0,0,0,0,0,0
 356  F637 3B                  FCB    59,239,238,213,213,235,23
 357  F63E 56                  FCB    86,121,221,34,119,218
 358  F644 4C                  FCB    76,204,88,240,147,15
 359  F649 C8                  FCB    200,89,35,198,112,34
 360  F650 DB                  FCB    219,155,97,45,254,213
 361  F656 B1                  FCB    177,146,119,48,29,14
 362  F65C D3                  FCB    211,202,197,147,148,104
```

```
MUSIBA4 * 12.FEV.81                    22-10-81  TSC ASSEMBLER  PAGE   10

 363  F662 6E                  FCB    110,70,32,45,12,28,46
 364  F669 10                  FCB    16,16,17,18,19,20,21
 365  F670 17                  FCB    23,24,25,27,28,30
 366  F676 20                  FCB    32,34,36,38,40,43
 367  F67C 2D                  FCB    45,48,51,54,57,60
 368  F682 3F                  FCB    63,66,72,76,80,85
 369  F688 5A                  FCB    90,96,101,107,114,120
 370  F68E 7F                  FCB    127,135,143,152,161,170
 371  F694 B4                  FCB    180,191,202,214,227,241,255
 372  F69B 44          DOASI   FCC    /DO33RE33MIFA33SO33LA33SI/
 373  F6B3 51          TRADUR  FCC    /QTDCNBR/
 374                   ********************************
 375  F6BA FF          TABTEM  FCB    255,220,180,136,105,86,73,63
 376  F6C2 38                  FCB    56,50,45,41,38,35,33,31
 377                   ************************
 378  F6CA F7 E4 E6    MUL2P1  STA B  FRAC
 379  F6CD FE E4 E2            LDX    NBPER
 380  F6D0 FF E4 E4            STX    NPH
 381  F6D3 4F                  CLR A
 382  F6D4 B7 E4 E2            STA A  NBPER
 383  F6D7 B7 E4 E3            STA A  NBPERB
 384  F6DA 86 80               LDA A  £$80
 385  F6DC B7 E4 E7            STA A  STROBE
 386  F6DF B6 E4 E7    MUL1    LDA A  STROBE
 387  F6E2 B4 E4 E6            AND A  FRAC
 388  F6E5 26 04               BNE    MUL2
 389  F6E7 8D 16               BSR    DEC1M
 390  F6E9 20 0E               BRA    MUL3
 391  F6EB 9D 12       MUL2    BSR    DEC1M
 392  F6ED FB E4 E3            ADD B  NBPERB
 393  F6F0 B9 E4 E2            ADC A  NBPER
 394  F6F3 B7 E4 E2            STA A  NBPER
 395  F6F6 F7 E4 E3            STA B  NBPERB
 396  F6F9 74 E4 E7    MUL3    LSR    STROBE
 397  F6FC 26 E1               BNE    MUL1
 398  F6FE 39                  RTS
 399  F6FF B6 E4 E4    DEC1M   LDA A  NPH
 400  F702 F6 E4 E5            LDA B  NPB
 401  F705 54          DECUN   LSR B
 402  F706 44                  LSR A
 403  F707 24 02               BCC    FDUN
 404  F709 CB 80               ADD B  £$80
 405  F70B B7 E4 E4    FDUN    STA A  NPH
 406  F70E F7 E4 E5            STA B  NPB
 407  F711 39                  RTS
 408  F712 FF E4 D7    PLUSA   STX    COMMUN
 409  F715 FE E4 D7            LDA B  COMMUN
 410  F718 B8 E4 D8            ADD A  COMMUN+1
 411  F71B C9 00               ADC B  £0
 412  F71D F7 E4 D7            STA B  COMMUN
 413  F720 B7 E4 D8            STA A  COMMUN+1
 414  F723 FE E4 D7            LDX    COMMUN
 415  F726 39                  RTS
 416                   ***********
```

```
MUSIBA4 * 12.FEV.81              22-10-81  TSC ASSEMBLER  PAGE   11

 417  F727 A6 00     SPNOTE  LDA A  X
 418  F729 84 DF             AND A  £$DF
 419  F72B B7 E4 F0          STA A  BY1
 420  F72E A6 01             LDA A  1,X
 421  F730 84 DF             AND A  £$DF
 422  F732 B7 E4 F1          STA A  BY2
 423  F735 A6 02             LDA A  2,X
 424  F737 B7 E4 F2          STA A  BY3
 425  F73A A6 03             LDA A  3,X
 426  F73C 84 DF             AND A  £$DF
 427  F73E B7 E4 F3          STA A  BY4
 428  F741 A6 04             LDA A  4,X
 429  F743 B7 E4 F4          STA A  BY5
 430  F746 B6 E4 D6          LDA A  TEMPO
 431  F749 26 05             BNE    OKTEM
 432  F74B 86 0A             LDA A  £13B
 433  F74D B7 E4 D6          STA A  TEMPO
 434  F750 8D 13     OKTEM   BSR    PREPAR
 435  F752 BD F8 34          JSR    LANOTE
 436  F755 39                RTS
 437                 *************
 438  F756 44        SPTEMP  LSR A
 439  F757 44                LSR A
 440  F758 44                LSR A
 441  F759 44                LSR A
 442  F75A CE F6 BA          LDX    £TABTEM
 443  F75D 8D B3             BSR    PLUSA
 444  F75F A6 00             LDA A  X
 445  F761 B7 E4 D6          STA A  TEMPO
 446  F764 39                RTS
 447                 **************
 448  F765 B6 E4 D4  PREPAR  LDA A  POKTIM
 449  F768 26 03             BNE    OKTIM
 450  F76A 73 E4 D4          COM    POKTIM
 451  F76D 86 02     OKTIM   LDA A  £2
 452  F76F CE F6 9B          LDX    £DDASI
 453  F772 C6 FF             LDA B  £$FF
 454  F774 F7 E4 EC          STA B  SILENC
 455  F777 FF E4 EE  RECNOT  STX    GARX
 456  F77A EE 20             LDX    X
 457  F77C BC E4 F0          CPX    BY1
 458  F77F 27 11             BEQ    OKMOT
 459  F781 4C                INC A
 460  F782 FE E4 EE          LDX    GARX
 461  F785 08                INX
 462  F786 08                INX
 463  F787 BC F6 B3          CPX    £DDASI+24
 464  F78A 26 EB             BNE    RECNOT
 465  F78C 4F                CLR A
 466  F78D 7F E4 EC          CLR    SILENC
 467  F790 20 11             BRA    PAUSE
 468  F792 F6 E4 F2  OKMOT   LDA B  BY3
 469  F795 C0 02             SUB B  £2
 470  F797 2E 02             BGT    EVIT1
```

```
MUSIBA4 * 12.FEV.81                    22-10-81  TSC ASSEMBLER  PAGE   12

 471  F799 C6 01           LDA B  £1
 472  F79B 58       EVIT1  ASL B
 473  F79C 58              ASL B
 474  F79D 1B              ABA
 475  F79E 58              ASL B
 476  F79F 1B              ABA
 477  F7A0 BB E4 F4        ADD A  BY5
 478  F7A3 CE F5 D3 PAUSE  LDX    £TABNOT
 479  F7A6 BD F7 12        JSR    PLUSA
 480  F7A9 A6 00           LDA A  X
 481  F7AB B7 E4 DB        STA A  DELPUL
 482  F7AE A6 32           LDA A  50,X
 483  F7B0 B7 E4 DC        STA A  DELREP
 484  F7B3 A6 64           LDA A  100,X
 485  F7B5 B7 E4 E1        STA A  DEREPB
 486  F7B8 A6 96           LDA A  150,X
 487  F7BA B7 E4 E2        STA A  NBPER
 488  F7BD 7F E4 E3        CLR    NBPER+1
 489  F7C0 44              LSR A
 490  F7C1 44              LSR A
 491  F7C2 44              LSR A
 492  F7C3 44              LSR A
 493  F7C4 B7 E4 EA        STA A  DECLOG
 494  F7C7 B7 E4 E9        STA A  DECPAS
 495  F7CA B6 E4 E1        LDA A  DEREPB
 496  F7CD 84 06           AND A  £6
 497  F7CF CE F8 2D        LDX    £DELAI8
 498  F7D2 4D       TESSAU TST A
 499  F7D3 27 05           BEQ    FINSAU
 500  F7D5 09              DEX
 501  F7D6 4A              DEC A
 502  F7D7 4A              DEC A
 503  F7D8 20 F8           BRA    TESSAU
 504  F7DA FF E4 BD FINSAU STX    DEL2
 505  F7DD B6 E4 DC        LDA A  DELREP
 506  F7E0 F6 E4 E1        LDA B  DEREPB
 507  F7E3 BD F7 05        JSR    DECUN
 508  F7E6 BD F7 05        JSR    DECUN
 509  F7E9 BD F7 05        JSR    DECUN
 510  F7EC B7 E4 DF        STA A  DEL8L
 511  F7EF F7 E4 E0        STA B  DEL8L+1
 512  F7F2 B6 E4 E1        LDA A  DEREPB
 513  F7F5 4F              CLR A
 514  F7F6 CE F6 B3        LDX    £TRADUR
 515  F7F9 E6 00    RECDUR LDA B  X
 516  F7FB F1 E4 F3        CMP B  BY4
 517  F7FE 27 07           BEQ    OKDUR
 518  F800 4C              INC A
 519  F801 08              INX
 520  F802 8C F6 BA        CPX    £TRADUR+7
 521  F805 26 F2           BNE    RECDUR
 522  F807 C6 01    OKDUR  LDA B  £1
 523  F809 4D       CALDUR TST A
 524  F80A 27 04           BEQ    CALPER
```

```
MUSIBA4 * 12.FEV.81              22-10-81  TSC ASSEMBLER  PAGE   13

  525  F80C 58              ASL B
  526  F80D 4A              DEC A
  527  F80E 20 F9           BRA    CALDUR
  528  F810 BD F6 CA  CALPER  JSR    MUL2P1
  529  F813 F6 E4 D6        LDA B  TEMPO
  530  F816 BD F6 CA        JSR    MUL2P1
  531  F819 86 3F           LDA A  £63
  532  F81B B4 E4 EC        AND A  SILENC
  533  F81E B7 E4 E8        STA A  NIVEAU
  534  F821 7F E4 E9        CLR    NIVB
  535  F824 39              RTS
  536                  **********
  537  F825 FE E4 DD  DELMUL  LDX    DEL2
  538  F828 6E 00           JMP    X
  539  F82A 01              NOP
  540  F82B 01              NOP
  541  F82C 01              NOP
  542  F82D FE E4 DF  DELAIB  LDX    DELBL
  543  F830 09        PLUS8   DEX
  544  F831 26 FD           BNE    PLUS8
  545  F833 39              RTS
  546                  **********
  547  F834 C6 0B     LANOTE  LDA B  £8
  548  F836 B6 E4 D4        LDA A  POKTIM
  549  F839 B7 E4 ED        STA A  TIMJEU
  550  F83C 86 40     LESBIM  LDA A  £64
  551  F83E 78 E4 ED        ASL    TIMJEU
  552  F841 24 05           BCC    BITABS
  553  F843 BB E4 E8        ADD A  NIVEAU
  554  F846 20 05           BRA    OUTSON
  555  F848 B0 E4 E8  BITABS  SUB A  NIVEAU
  556  F84B 20 00           BRA    OUTSON
  557  F84D B7 E8 61  OUTSON  STA A  LESON
  558  F850 B6 E4 DB        LDA A  DELPUL
  559  F853 27 03           BEQ    PASDE8
  560  F855 4A        ET8     DEC A
  561  F856 26 FD           BNE    ET8
  562  F858 5A        PASDE8  DEC B
  563  F859 26 E1           BNE    LESBIM
  564  F85B C6 40           LDA B  £64
  565  F85D F7 E8 61        STA B  LESON
  566  F860 8D C3           BSR    DELMUL
  567  F862 B6 E4 D5        LDA A  POKCOR
  568  F865 27 2D           BEQ    RAF1
  569  F867 7A E4 EB        DEC    DECPAS
  570  F86A 26 2C           BNE    RAF2
  571  F86C B6 E4 EA        LDA A  DECLOG
  572  F86F B7 E4 EB        STA A  DECPAS
  573  F872 B6 E4 E8        LDA A  NIVEAU
  574  F875 F6 E4 E9        LDA B  NIVB
  575  F878 F0 E4 E8        SUB B  NIVEAU
  576  F87B 82 00           SBC A  £0
  577  F87D F0 E4 E8        SUB B  NIVEAU
  578  F880 82 00           SBC A  £0
```

```
 579  F882 F0 E4 E8          SUB B  NIVEAU
 580  F885 82 00             SBC A  £0
 581  F887 F0 E4 E8          SUB B  NIVEAU
 582  F88A 82 00             SBC A  £0
 583  F88C B7 E4 E8          STA A  NIVEAU
 584  F88F F7 E4 E9          STA B  NIVB
 585  F892 20 09             BRA    TESTAM
 586  F894 86 09     RAF1    LDA A  £9
 587  F896 20 02             BRA    COMPEN
 588  F898 86 07     RAF2    LDA A  £7
 589  F89A 4A        COMPEN  DEC A
 590  F89B 26 FD             BNE    COMPEN
 591  F89D FE E4 E2  TESTAM  LDX    NBPER
 592  F8A0 09                DEX
 593  F8A1 FF E4 E2          STX    NBPER
 594  F8A4 26 8E             BNE    LANOTE
 595  F8A6 39                RTS
 596                 **********************
 597                        END
 598                        END
```

NO ERROR(S) DETECTED

SYMBOL TABLE:

```
                    NAM    VIMON V1.4
                    OPT    PAG

                ***********************************
                *                                 *
                *    VIMON                        *
                *    VERSION 1.4  AUGUST 81       *
                *    24*80   8 OR 5" FLOPPY DISK  *
                *    COPYRIGHT 1980 BY SMT        *
                *                                 *
                ***********************************


                *
                *   COMMAND SET
                *   -----------
                *
                *      M  MEMORY CHANGE
                *      D  MEMORY DUMP
                *      R  DISPLAY CONTENTS OF TARGET STACK
                *           CC  B  A  X  P  S
                *      B  PRINT OUT ALL BREAKPOINTS
                *      C  CONTINUE EXECUTION FROM CURRENT LOCATION
                *      N  NEXT INSTRUCTION
                *      T  TRACE N INSTRUCTIONS
                *      G  GO TO LOCATION N
                *      W  DELETE ALL BREAKPOINTS
                *      U  RESET BREAKPOINT AT ADDRESS N
                *      V  SET BREAKPOINT AT ADDRESS N
                *      O  SWITCH OUTPUT TO PRINTER
                *
                *
                *   KEYBOARD FUNCTIONS
                *   ------------------
                *
                *      COM    SET TERMINAL MODE
                *      BOOT   EXECUTE DISK BOOTSTRAP
                *
                *
                *   ENTRY AND I/O ROUTINES
                *   ----------------------
                *
                *      CONTRL $FFE6  WARM START
                *      CHKCHR $FFE9  CHECK FOR TYPED CHARACTER
                *      OUTCH  $FFEC  OUTPUT ONE CHARACTER
                *      INCH   $FFEF  INPUT ONE CHARACTER
                *      PDATA1 $FFF2  PRINT CHARACTERS STRING
                *      PCRLF  $FFF5  PRINT CR/LF
                *
                *
                * PSEUDO MACROS DEFINITIONS
                *
0085            SKIP1   EQU    $85       SKIP ONE BYTE
008C            SKIP2   EQU    $8C       SKIP TWO BYTES
```

```
V1MON V1.4                      9-12-81  ASSEMBLEUR TSC 6800   PAGE    1

                                ORG     $F400
                                NAM     HANDLER GRAPHIQUE
                        *
                        * SOUS PROGRAMMES GRAPHIQUES
                        * POUR LA CARTE GRAPHIQUE COULEUR
                        * GOUPIL
                        *
                        * COPYRIGHT SMT 81
                        *
                        *
                        * POINTS D'ENTREE:
                        * -GINIT = INITIALISATION DU 6845
                        *          ET CREATION DU FOND COLORE
                        * -GCOLOR= DEFINITION DE LA COULEUR
                        *          DU FOND ET DU TRACE
                        * -GPLOT= TRACE DE POINTS ET DE LIGNES
                        * -GTEXT=COMMUTATION MODE TEXT
                        * -GRAPH=COMMUTATION MODE GRAPHIQUE
                        *          -0-0-0-0-0-0-
                        *


                        *
                        * ADRESSES DE LA CARTE GRAPHIQUE
                        *
E7F8            ADVISU  EQU     $E7F8
E7FB            GCURX   EQU     ADVISU+3
E7F9            GCURY   EQU     ADVISU+1
E7FC            GCONTR  EQU     ADVISU+4
E7FE            GCRTC   EQU     ADVISU+6
```

```
HANDLER GRAPHIQUE                    9-12-81  ASSEMBLEUR TSC 6800   PAGE    2

  88                *
  89                * TABLE DE BRANCHEMENTS
  90                *
  91  F400 7E F4 1A          JMP    GINIT     GR
  92  F403 7E F4 0F          JMP    GTEXT     TEXT
  93  F406 7E F4 64          JMP    GCOLOR    COLOR
  94  F409 7E F4 6F          JMP    GPLOT     PLOT
  95  F40C 7E F4 4B          JMP    GRAPH     GRAF
  96                *
  97                *
  98                * GTEXT
  99                * ARGUMENTS D APPEL = AUCUN
 100                * REGISTRE MODIFIE :A
 101                *
 102  F40F 86 80     GTEXT   LDA A  £$80
 103  F411 B7 E4 D3          STA A  DIRECT
 104  F414 86 C0             LDA A  £$C0
 105  F416 B7 E7 FC          STA A  GCONTR
 106  F419 39                RTS
 107                *
 108                * GINIT
 109                * ARGUMENTS D'APPEL=AUCUN
 110                * REGISTRES MODIFIES=A,B
 111                *
 112  F41A FF E4 CA  GINIT   STX    GDELTA
 113  F41D CE F4 54          LDX    £GCRTTB   INITIALISATION 6845
 114  F420 5F                CLR B
 115  F421 F7 E7 FE  GBCRT   STA B  GCRTC
 116  F424 A6 00             LDA A  0,X
 117  F426 B7 E7 FF          STA A  GCRTC+1
 118  F429 08                INX
 119  F42A 5C                INC B
 120  F42B C1 10             CMP B  £16
 121  F42D 26 F2             BNE    GBCRT
 122  F42F B6 E7 FD          LDA A  GCONTR+1  DEFINITION DU FOND COLORE
 123  F432 88 40             EOR A  £$40
 124  F434 B7 E7 FD          STA A  GCONTR+1
 125  F437 86 14             LDA A  £20       ATTENTE 20MS
 126  F439 5F                CLR B
 127  F43A 5A        GBWAIT  DEC B
 128  F43B 26 FD             BNE    GBWAIT
 129  F43D 4A                DEC A
 130  F43E 26 FA             BNE    GBWAIT
 131  F440 B6 E7 FD          LDA A  GCONTR+1
 132  F443 88 40             EOR A  £$40
 133  F445 B7 E7 FD          STA A  GCONTR+1
 134  F448 FE E4 CA          LDX    GDELTA
 135                *
 136                * GRAPH
 137                * ARGUMENT D APPEL = AUCUN
 138                * REGISTRE MODIFIE : A
 139                *
 140  F44B 7F E4 D3  GRAPH   CLR    DIRECT
 141  F44E 86 40             LDA A  £$40
 142  F450 B7 E7 FC          STA A  GCONTR
```

```
HANDLER GRAPHIQUE                 9-12-81  ASSEMBLEUR TSC 6800   PAGE    3

  143  F453 39              RTS
  144                  *TABLE D'INITIALISATION DU 6845
  145  F454 55         SCRTTB  FCB    $55,64,69,5,$4D,0,$40,$44
  146  F45C 00                 FCB    0,3,2,0,0,0,0,0
  147                  *
  148                  * GCOLOR
  149                  *
  150                  * ARGUMENTS D'APPEL=(A)CODE DE COULEUR
  151                  * REGISTRES MODIFIES=A
  152                  *
  153  F454 84 07      GCOLOR  AND A  #$07
  154  F456 8A 40              ORA A  #$40
  155  F458 BA E4 D3           ORA A  DIRECT
  156  F46B B7 E7 FD           STA A  GCONTR+1
  157  F46E 3S                 RTS
```

```
HANDLER GRAPHIQUE                 9-12-81  ASSEMBLEUR TSC 6800   PAGE    4

  159                  *
  160                  * GPLOT
  161                  *
  162                  * ARGUMENTS D'APPEL=X->ZONE DE PARAMETRES
  163                  * SOUS-PROGRAMMES APPELES=GMOVE,GTRACE
  164                  * REGISTRES MODIFIES=A,B
  165                  *
  166  F45F A6 00      GPLOT   LDA A  0,X
  167  F471 81 00              CMP A  #0
  168  F473 25 06              BNE    GCMD1
  169  F475 A6 01      GCMD0   LDA A  1,X
  170  F477 E6 02              LDA B  2,X
  171  F479 20 1B              BRA    GMOVE
  172  F478 81 01      GCMD1   CMP A  #1
  173  F47D 25 06              BNE    GCMD2
  174  F47F A6 01              LDA A  1,X
  175  F481 E6 02              LDA B  2,X
  176  F483 20 1E              BRA    GTRACE
  177  F485 81 02      GCMD2   CMP A  #2
  178  F487 26 0C              BNE    GCMDE
  179  F489 A6 01              LDA A  1,X
  180  F48B E6 02              LDA B  2,X
  181  F48D 8D 07              BSR    GMOVE
  182  F48F A6 03              LDA A  3,X
  183  F491 E6 04              LDA B  4,X
  184  F493 20 0E              BRA    GTRACE
  185  F495 39         GCMDE   RTS
```

```
HANDLER GRAPHIQUE                    9-12-81  ASSEMBLEUR TSC 6800   PAGE    5

                        *
                        * GMOVE   POSITIONNEMENT SUR UN POINT
                        *         ET ECRITURE
                        *
                        * ARGUMENTS D'APPEL=(A)ABCISSE,(B)ORDONNEE
                        * REGISTRES MODIFIES=A
                        *
F496 B7 E7 FB   GMOVE   STA A  GCURX
F499 F7 E7 F9           STA B  GCURY
F49C 96 E7 FD           LDA A  GCONTR+1
F49F B7 E7 FC           STA A  GCONTR
F4A2 39                 RTS
                        *
                        * GTRACE   TRACE D'UNE LIGNE DROITE
                        *
                        * ARGUMENTS D'APPEL=(A)ABSCISSE,(B)ORDONNEE
                        * SOUS-PROGRAMMES APPELES=GCINC,GAINC
                        * REGISTRES MODIFIES=A,B
                        *
F4A3 B0 E7 FB   GTRACE  SUB A  GCURX     CALCUL DU DEPLACEMENT
F4A6 22 07              BHI    GSUPX     SUIVANT X ET SON SIGNE
F4A8 26 0C              BNE    GINFX
F4AA B7 E4 CE           STA A  GDX
F4AD 20 0D              BRA    GMVY
F4AF B7 E4 CE   GSUPX   STA A  GDX
F4B2 86 01              LDA A  #1
F4B4 20 06              BRA    GMVY
F4B6 40         GINFX   NEG A
F4B7 B7 E4 CE           STA A  GDX
F4BA 86 FF              LDA A  #-1
F4BC B7 E4 D0   GMVY    STA A  GSX
F4BF F0 E7 F9           SUB B  GCURY     CALCUL DU DEPLACEMENT
F4C2 22 07              BHI    GSUPY     SUIVANT Y ET SON SIGNE
F4C4 26 0C              BNE    GINFY
F4C6 F7 E4 CF           STA B  GDY
F4C9 20 0D              BRA    GCMPD
F4CB F7 E4 CF   GSUPY   STA B  GDY
F4CE C6 01              LDA B  #1
F4D0 20 06              BRA    GCMPD
F4D2 50         GINFY   NEG B
F4D3 F7 E4 CF           STA B  GDY
F4D6 C6 FF              LDA B  #-1
F4D8 F7 E4 D1   GCMPD   STA B  GSY
F4DB B6 E4 CE           LDA A  GDX       COMPARAISON DES DEPLACEMENTS
F4DE B1 E4 CF           CMP A  GDY
F4E1 22 21              BHI    GPASX
F4E3 26 54              BNE    GPASY
F4E5 4D                 TST A
F4E6 27 1B              BEQ    GPRET     ILS SONT NULS:RIEN A FAIRE
F4E8 F6 E7 FB   GBPLT1  LDA B  GCURX     ILS SONT EGAUX:INCREMENTS
F4EB FB E4 D0           ADD B  GSX       ENTIER GSX ET GSY
F4EE F7 E7 FB           STA B  GCURX
F4F1 F6 E7 F9           LDA B  GCURY
F4F4 FB E4 D1           ADD B  GSY
F4F7 F7 E7 F9           STA B  GCURY
```

```
HANDLER GRAPHIQUE                    9-12-81  ASSEMBLEUR TSC 6800   PAGE    6

  242  F4FA F6 E7 FD          LDA B  GCONTR+1
  243  F4FD F7 E7 FC          STA B  GCONTR
  244  F500 4A                DEC A
  245  F501 26 E5             BNE    GBPLT1
  246  F503 39         GPRET  RTS
  247  F504 B6 E4 CE   GPASX  LDA A  GDX       DEPLACEMENT PLUS GRAND
  248  F507 F6 E4 CF          LDA B  GDY       SUIVANT X:INCREMENT ENTIER POUR X
  249  F50A 27 DC             BEQ    GBPLT1    CALCUL DE L'INCREMENT POUR Y
  250  F50C F7 E4 CA          STA B  GDELTA
  251  F50F F6 E7 F9          LDA B  GCURY
  252  F512 F7 E4 C7          STA B  GCUR
  253  F515 F6 E4 D1          LDA B  GSY
  254  F518 F7 E4 CD          STA B  GSD
  255  F51B 8D 50             BSR    GCINC
  256  F51D BD F5 B0   GBPLT2 JSR    GAINC     TRACE
  257  F520 F6 E7 FB          LDA B  GCURX
  258  F523 FB E4 D0          ADD B  GSX
  259  F526 F7 E7 FB          STA B  GCURX
  260  F529 F6 E4 C7          LDA B  GCUR
  261  F52C F7 E7 F9          STA B  GCURY
  262  F52F F6 E7 FD          LDA B  GCONTR+1
  263  F532 F7 E7 FC          STA B  GCONTR
  264  F535 4A                DEC A
  265  F536 26 E5             BNE    GBPLT2
  266  F538 39                RTS
  267  F539 B6 E4 CF   GPASY  LDA A  GDY       DEPLACEMENT PLUS GRAND
  268  F53C F6 E4 CE          LDA B  GDX       SUIVANT Y:INCREMENT ENTIER POUR Y
  269  F53F 27 A7             BEQ    GBPLT1    CALCUL DE L'INCREMENT POUR X
  270  F541 F7 E4 CA          STA B  GDELTA
  271  F544 F6 E7 FB          LDA B  GCURX
  272  F547 F7 E4 C7          STA B  GCUR
  273  F54A F6 E4 D0          LDA B  GSX
  274  F54D F7 E4 CD          STA B  GSD
  275  F550 8D 1B             BSR    GCINC
  276  F552 8D 5C     GBPLT3  BSR    GAINC     TRACE
  277  F554 F6 E4 C7          LDA B  GCUR
  278  F557 F7 E7 FB          STA B  GCURX
  279  F55A F6 E7 F9          LDA B  GCURY
  280  F55D FB E4 D1          ADD B  GSY
  281  F560 F7 E7 F9          STA B  GCURY
  282  F563 F6 E7 FD          LDA B  GCONTR+1
  283  F566 F7 E7 FC          STA B  GCONTR
  284  F569 4A                DEC A
  285  F56A 26 E6             BNE    GBPLT3
  286  F56C 39                RTS
```

```
HANDLER GRAPHIQUE                9-12-81  ASSEMBLEUR TSC 6800   PAGE    7

                        *
                        * GCINC   CALCUL DES INCREMENTS FRACTIONNAIRES
                        *
                        * ARGUMENTS D'APPEL=(A)NOMBRE DE PAS
                        *                  (GDELTA)DEPLACEMENT
                        *                 (GSD)SIGNE
                        * REGISTRES MODIFIES=B
                        *
F56D 36                 GCINC   PSH A
F56E C6 80                      LDA B  £$80
F570 F7 E4 C8                   STA B  GCUR+1
F573 7F E4 C9                   CLR    GCUR+2
F576 C6 10                      LDA B  £16
F578 F7 E4 D2                   STA B  GCMPT
F57B 16                         TAB
F57C B6 E4 CA                   LDA A  GDELTA
F57F 78 E4 CC           GBDIV   ASL    GDELTA+2  DIVISION DE GDELTA
F582 79 E4 CB                   ROL    GDELTA+1
F585 48                         ASL A            PAR LE NOMBRE DE PAS
F586 25 03                      BCS    GDSUB
F588 11                         CBA
F589 25 04                      BCS    GFBDIV
F58B 10                 GDSUB   SBA
F58C 7C E4 CC                   INC    GDELTA+2
F58F 7A E4 D2           GFBDIV  DEC    GCMPT
F592 26 EB                      BNE    GBDIV
F594 7F E4 CA                   CLR    GDELTA    MULTIPLICATION PAR SON SIGNE
F597 7D E4 CD                   TST    GSD
F59A 2A 12                      BPL    GRCINC
F59C 4F                         CLR A
F59D B0 E4 CC                   SUB A  GDELTA+2
F5A0 B7 E4 CC                   STA A  GDELTA+2
F5A3 86 00                      LDA A  £0
F5A5 B2 E4 CB                   SBC A  GDELTA+1
F5A8 B7 E4 CB                   STA A  GDELTA+1
F5AB 7A E4 CA                   DEC    GDELTA
F5AE 32                 GRCINC  PUL A
F5AF 39                         RTS
                        *
                        * GAINC    ADDITION DE L'INCREMENT
                        *
                        * PARAMETRES D'APPEL=AUCUN
                        * REGISTRES MODIFIES=B
                        *
F5B0 F6 E4 C9           GAINC   LDA B  GCUR+2
F5B3 FB E4 CC                   ADD B  GDELTA+2
F5B6 F7 E4 C9                   STA B  GCUR+2
F5B9 F6 E4 C8                   LDA B  GCUR+1
F5BC F9 E4 CB                   ADC B  GDELTA+1
F5BF F7 E4 C8                   STA B  GCUR+1
F5C2 F6 E4 C7                   LDA B  GCUR
F5C5 F9 E4 CA                   ADC B  GDELTA
F5C8 F7 E4 C7                   STA B  GCUR
F5CB 39                         RTS
                        *
```

```
HANDLER GRAPHIQUE                    9-12-81  ASSEMBLEUR TSC 6800   PAGE    8

  343                 * RESERVATIONS MEMOIRE RAM
  344                 *
  345  E4C7                    ORG    $E4C7
  346  E4C7           GCUR     RMB    3         POINTEUR A INCREMENT NON ENTIER
  347  E4CA           GDELTA   RMB    3         INCREMENT NON ENTIER
  348  E4CD           GSD      RMB    1         SIGNE INCREMENT
  349  E4CE           GDX      RMB    1         DEPLACEMENT EN X
  350  E4CF           GDY      RMB    1         DEPLACEMENT EN Y
  351  E4D0           GSX      RMB    1         SIGNE DEPLACEMENT EN X
  352  E4D1           GSY      RMB    1         SIGNE DEPLACEMENT EN Y
  353  E4D2           GCMPT    RMB    1         NOMBRE DE PAS
  354  E4D3           DIRECT   RMB    1         COMMUTATION ECRAN
  355                          END
```

```
 357  F5CC                ORG    $F5CC
 359                      NAM    8 OR 5" DISK BOOTSTRAP
 360
 361
 362                 ********************************
 363                 *                              *
 364                 * -S- COMMAND                  *
 365                 *   DISK BOOTSTRAP             *
 366                 *   5 OR 8" DMAF2 - 5" MFB     *
 367                 *                              *
 368                 ********************************
 369
 370                 * EQUATES FOR WD 1795 AND DMA
 371
 372  EB20           STAREG  EQU    $EB20
 373  EB21           TRKREG  EQU    $EB21
 374  EB22           SECREG  EQU    $EB22
 375  EB23           DATREG  EQU    $EB23
 376  EB24           DRVREG  EQU    $EB24
 377  EB00           DMAADD  EQU    $EB00
 378  EB02           DMACON  EQU    $EB02
 379  EB10           DMACOM  EQU    $EB10
 380  EB14           DMAPRI  EQU    $EB14
 381  A100           LOADER  EQU    $A100
 382
 383                 * EQUATES FOR WD 1791
 384
 385  E8E0           DRVR91  EQU    $E8E0
 386  E8F0           COMR91  EQU    $E8F0
 387  E8F2           SECR91  EQU    $E8F2
 388  E8F3           DATR91  EQU    $E8F3
 389
 390                 * PROGRAM STARTS HERE
 391
 392  F5CC 86 01     DKBOOT  LDA A  £$01
 393  F5CE B7 EB 22          STA A  SECREG
 394  F5D1 8D 69             BSR    DEL1
 395  F5D3 81 EB 22          CMP A  SECREG   TEST IF DMAF2 PRESENT
 396  F5D6 26 67             BNE    MFBOOT   IF NOT BOOT ON MFB
 397
 398  F5D8 C6 BF     DMABOT  LDA B  £$BF
 399  F5DA F7 EB 24          STA B  DRVREG   SELECT DRIVE
 400  F5DD 8D 4C             BSR    READY    TEST READY
 401  F5DF 27 0E             BEQ    SBOOT    IF READY CONTINUE
 402  F5E1 C6 7F             LDA B  £$7F     SELECT 5"
 403  F5E3 F7 EB 24          STA B  DRVREG
 404  F5E6 8D 49             BSR    ONESEC   ONE SECOND DELAY
 405  F5E8 8D 41             BSR    READY    TEST READY
 406  F5EA 27 03             BEQ    SBOOT    IF READY CONTINUE
 407  F5EC 7E FD 12          JMP    CONTRL
 408  F5EF 86 09     SBOOT   LDA A  £9
 409  F5F1 B7 EB 20          STA A  STAREG   SEND RESTOR COMMAND
 410  F5F4 8D 44             BSR    DELAY
 411  F5F6 8D 2D             BSR    TEST     WAIT TILL NOT BUSY
 412  F5F8 CE FE FF          LDX    £$FEFF
```

```
8 OR 5" DISK BOOTSTRAP              9-12-81  ASSEMBLEUR TSC 6800   PAGE   10

  413  F5FB FF EB 02          STX    DMACON    SET BUFFER LENGTH
  414  F5FE CE 5E FF          LDX    £$5EFF
  415  F601 FF EB 00          STX    DMAADD    SET BUFFER ADDRESS
  416  F604 86 FD             LDA A  £$FD
  417  F606 B7 EB 10          STA A  DMACOM
  418  F609 86 FE             LDA A  £$FE
  419  F60B B7 EB 14          STA A  DMAPRI
  420  F60E 86 8C             LDA A  £$8C
  421  F610 B7 EB 20          STA A  STAREG    SEND READ COMMAND
  422  F613 FE EB 02  WAIT    LDX    DMACON
  423  F616 8C FE FF          CPX    £$FEFF
  424  F619 27 F8             BEQ    WAIT      WAIT TIL DMA OP. FINISHED
  425  F61B 86 FF             LDA A  £$FF      RESET DMA
  426  F61D B7 EB 14          STA A  DMAPRI
  427  F620 8D 03             BSR    TEST
  428  F622 7E A1 00  GOLOAD  JMP    LOADER    GOTO LOADER
  429  F625 8D 04     TEST    BSR    READY     GET STATUS
  430  F627 47                ASR A
  431  F628 25 FB             BCS    TEST      WAIT TIL NOT BUSY
  432  F62A 39                RTS
  433  F62B B6 EB 20  READY   LDA A  STAREG    GET WD STATUS
  434  F62E 85 80             BIT A  £$80
  435  F630 39                RTS
  436
  437  F631 CE 00 00  ONESEC  LDX    £0        ONE SECOND DELAY
  438  F634 08        D15     INX
  439  F635 09                DEX
  440  F636 09                DEX
  441  F637 26 FB             BNE    D15
  442  F639 39                RTS
  443
  444  F63A 8D 00     DELAY   BSR    DEL1
  445  F63C 8D 00     DEL1    BSR    RTN
  446  F63E 39        RTN     RTS
  447
  448                * BOOTSTRAP FOR MFB
  449
  450  F63F 43        MFBOOT  COM A            SET SECTOR ONE
  451  F640 B7 E8 F2          STA A  SECR91     AND START MOTOR
  452  F643 7F E8 E0          CLR    DRVR91     SELECT DRIVE 0
  453  F646 8D E9             BSR    ONESEC     DELAY ONE SECOND
  454  F648 C6 F4             LDA B  £$F4       SEND RESTOR COMMAND
  455  F64A F7 E8 F0          STA B  COMR91
  456  F64D 8D EB             BSR    DELAY
  457  F64F F6 E8 F0  LOOP1   LDA B  COMR91
  458  F652 57                ASR B
  459  F653 24 FA             BCC    LOOP1     WAIT TILL NOT BUSY
  460  F655 86 73             LDA A  £$73       SEND READ COMMAND
  461  F657 B7 E8 F0          STA A  COMR91
  462  F65A 8D DE             BSR    DELAY
  463  F65C CE A1 00          LDX    £LOADER   ADDRESS OF LOADER
  464  F65F F6 E8 F0  LOOP2   LDA B  COMR91     GET WD STATUS
  465  F662 57                ASR B             CHECK BUSY
  466  F663 25 BD             BCS    GOLOAD    IF NOT BUSY GO TO LOADER
  467  F665 57                ASR B             CHECK DATA PRESENT
  468  F666 25 F7             BCS    LOOP2     LOOP TO WAIT DATA
```

```
8 OR 5" DISK BOOTSTRAP              9-12-81  ASSEMBLEUR TSC 6800   PAGE   11

  469  F668 B6 EB F3         LDA A  DATRS1    GET DATA
  470  F66B 43               COM A
  471  F66C A7 00            STA A  0,X       STORE IT IN MEMORY
  472  F66E 08               INX              BUMP POINTER
  473  F66F 20 EE            BRA    LOOP2     CONTINUE
  474
  475                        END
```

```
8 OR 5" DISK BOOTSTRAP              9-12-81  ASSEMBLEUR TSC 6800  PAGE  12

 477  F671                ORG    $F671
 479                      NAM    2480 SCREEN HANDLER
 480
 481              ********************************
 482              *                              *
 483              *    24*80 SCREEN HANDLER      *
 484              *                              *
 485              ********************************
 486
 487
 488              *
 489              *    I/O ROUTINES
 490              *    ------------
 491              *
 492              *        CRTINT     INITIALIZATIONS
 493              *        OUTPUT     OUTPUT ONE CHARACTER
 494              *        BELL       OUTPUT BELL
 495              *
```

```
2400 SCREEN HANDLER              9-12-81  ASSEMBLEUR TSC 6800   PAGE   13

                        *
                        * 6845 INITIALIZATION
                        *
00EC                    HADRF   EQU     $EC
E870                    CRTC    EQU     $E870
000D                    VCR     EQU     13
000A                    VLF     EQU     10
F671                    INITVS  EQU     *
F671 5F                 CRTINT  CLR B   -
F672 CE F6 AE                   LDX     #CRTTAB
F675 8D 16                      BSR     CRTI1
F677 C6 80                      LDA B   #$80
F679 F7 E5 70                   STA B   VIDEO
F67C CE EC 00           CLEAR2  LDX     #$EC00
F67F 86 20                      LDA A   #$20
F681 BA E5 70                   ORA A   VIDEO
F684 A7 00              CLEAR1  STA A   0,X
F686 08                         INX
F687 8C F4 00                   CPX     #$F400
F68A 26 F8                      BNE     CLEAR1
F68C 39                         RTS
F68D F7 E8 70           CRTI1   STA B   CRTC
F690 A6 00                      LDA A   0,X
F692 B7 E8 71                   STA A   CRTC+1
F695 08                         INX
F696 5C                         INC B
F697 C1 10                      CMP B   #$10
F699 26 F2                      BNE     CRTI1
F69B CE 00 00                   LDX     #0
F69E FF E5 73                   STX     CURAD
F6A1 FF E5 71                   STX     LINE
F6A4 FF E5 75                   STX     DPAGE
F6A7 CE 07 7F                   LDX     #1919
F6AA FF E5 77                   STX     EPAGE
F6AD 39                         RTS

F6AE 61                 CRTTAB  FCB     97,80,83,5
F6B2 18                         FCB     24,9,24,24
F6B6 00                         FCB     0,12,$51,11
F6BA 00 00                      FDB     0,0

                        *
                        * OUTPUT BELL
                        *
F6BE 86 7F              BELL    LDA A   #$7F
F6C0 8D 0D              BELL1   BSR     BELDEL
F6C2 53                         COM B
F6C3 F7 E8 61                   STA B   VIAMUS+ORA
F6C6 8D 07                      BSR     BELDEL
F6C8 F7 E8 61                   STA B   VIAMUS+ORA
F6CB 4A                         DEC A
F6CC 26 F2                      BNE     BELL1
F6CE 39                         RTS
F6CF C6 60              BELDEL  LDA B   #$60
F6D1 5A                 BELDE1  DEC B
```

```
2400 SCREEN HANDLER                9-12-81  ASSEMBLEUR TSC 6800   PAGE   14

  552  F6D2 26 FD            BNE    BELDE1
  553  F6D4 39               RTS
  554
  555                 *
  556                 * CLEAR SCREEN
  557                 *
  558  F6D5 8D A5     CLEAR  BSR    CLEAR2    CLEAR REFRESH MEMORY
  559                 *
  560                 * CURSOR HOME
  561                 *
  562  F6D7 7F E5 71  HOME   CLR    LINE      RESET LINE NUMBER
  563                 *
  564                 * LINE RETURN
  565                 *
  566  F6DA 7F E5 72  LINRET CLR    COLUMN    RESET COLUMN NUMBER
  567  F6DD 20 58            BRA    MOVEA     GOTO MOVE CURSOR
  568
  569                 *
  570                 * MOVE CURSOR
  571                 *
  572  F6DF 7C E5 72  INCCL  INC    COLUMN    UPDATE COLUMN NUMBER
  573  F6E2 B6 E5 72         LDA A  COLUMN
  574  F6E5 81 4F            CMP A  #79       END OF LINE ?
  575  F6E7 22 14            BHI    NEWLIN    IF YES ,NEW LINE
  576  F6E9 FE E5 73         LDX    CURAD     UPDATE CURSOR ADDRESS
  577  F6EC 08               INX
  578  F6ED FF E5 73  DECCL1 STX    CURAD
  579  F6F0 20 E6            BRA    OUTCUR    GOTO OUTPUT CURSOR
  580
  581  F6F2 7A E5 72  DECCL  DEC    COLUMN    DEC COLUMN NUMBER
  582  F6F5 2B 24            BMI    UPLINE    BRANCH IF PREVIOUS LINE
  583  F6F7 FE E5 73         LDX    CURAD     UPDATE CURSOR ADDRESS
  584  F6FA 09               DEX
  585  F6FB 20 F0            BRA    DECCL1    GOTO OUTPUT NEW CURSOR
  586
  587  F6FD 7F E5 72  NEWLIN CLR    COLUMN    CLEAR COLUMN NUMBER
  588  F700 7C E5 71  INCLN  INC    LINE
  589  F703 B6 E5 71         LDA A  LINE
  590  F706 81 17            CMP A  #23
  591  F708 23 2D            BLS    MOVEA
  592  F70A 7D E4 18         TST    PAGFG
  593  F70D 26 07            BNE    INCLN1
  594  F70F 7A E5 71         DEC    LINE
  595  F712 8D 5A            BSR    SCROLL
  596  F714 20 21            BRA    MOVEA
  597  F716 7F E5 71  INCLN1 CLR    LINE
  598  F719 20 1C            BRA    MOVEA
  599                 *
  600  F71B B6 4F     UPLINE LDA A  #79       CURSOR ON END OF LINE
  601  F71D B7 E5 72         STA A  COLUMN
  602  F720 7A E5 71  DECLN  DEC    LINE
  603  F723 2A 12            BPL    MOVEA
  604  F725 7D E4 18         TST    PAGFG
  605  F728 26 08            BNE    DECLN1
  606  F72A 7C E5 71         INC    LINE
  607  F72D BD F7 CB         JSR    SCROLD
```

```
2480 SCREEN HANDLER                9-12-81  ASSEMBLEUR TSC 6800   PAGE   15

  608  F730 20 05            BRA    MOVEA
  609  F732 86 17     DECLN1  LDA A  £23
  610  F734 B7 E5 71          STA A  LINE
  611
  612                 * CURSOR RELATIVE ADDRESS
  613  F737 F6 E5 72  MOVEA   LDA B  COLUMN    GET COLUMN NUMBER
  614  F73A B6 E5 71          LDA A  LINE      GET LINE NUMBER
  615  F73D 27 0D             BEQ    MOVE2     BRANCH IF ZERO
  616  F73F B7 E5 79          STA A  TEMP1
  617  F742 4F                CLR A
  618  F743 CB 50     MOVE1   ADD B  £80
  619  F745 89 00             ADC A  £0
  620  F747 7A E5 79          DEC    TEMP1     DEC COUNT
  621  F74A 26 F7             BNE    MOVE1     LOOP IF NOT ZERO
  622
  623                 * CURSOR ABSOLUTE ADDRESS
  624  F74C FB E5 76  MOVE2   ADD B  DPAGE+1   ADD PAGE ADDRESS
  625  F74F B9 E5 75          ADC A  DPAGE
  626  F752 B7 E5 73          STA A  CURAD     SAVE CURRENT CURSOR ADDRESS
  627  F755 F7 E5 74          STA B  CURAD+1
  628
  629                 * OUTPUT CURSOR TO CRTC
  630  F758 C6 0E     OUTCUR  LDA B  £14
  631  F75A F7 E8 70          STA B  CRTC
  632  F75D B6 E5 73          LDA A  CURAD
  633  F760 B7 E8 71          STA A  CRTC+1
  634  F763 5C                INC B
  635  F764 F7 E8 70          STA B  CRTC
  636  F767 B6 E5 74          LDA A  CURAD+1
  637  F76A B7 E8 71          STA A  CRTC+1
  638
  639  F76D 39                RTS
  640
  641                 *
  642                 * SCROLL UP
  643                 *
  644
  645                 * UPDATE END ADDRESS AND FILL WITH SPACE
  646  F76E FE E5 77  SCROLL  LDX    EPAGE     X:= END OF PAGE
  647  F771 8D 32             BSR    DELLIN    ERASE NEW LINE
  648  F773 FF E5 77          STX    EPAGE     NEW END OF PAGE
  649  F776 B6 E5 77          LDA A  EPAGE
  650  F779 84 07             AND A  £7
  651  F77B B7 E5 77          STA A  EPAGE
  652
  653                 * UPDATE START ADDRESS
  654  F77E 4F                CLR A
  655  F77F F6 E5 76          LDA B  DPAGE+1
  656  F782 CB 50             ADD B  £80
  657  F784 B9 E5 75          ADC A  DPAGE
  658  F787 84 07             AND A  £7
  659  F789 B7 E5 75          STA A  DPAGE
  660  F78C F7 E5 76          STA B  DPAGE+1
  661
  662                 * UPDATE CRTC START ADDRESS
  663  F78F C6 0C     UCRTC   LDA B  £12
```

```
2400 SCREEN HANDLER                9-12-81  ASSEMBLEUR TSC 6800   PAGE   15

 664  F791 F7 E8 70          STA B  CRTC
 665  F794 B6 E5 75          LDA A  DPAGE
 666  F797 B7 E8 71          STA A  CRTC+1
 667  F79A 5C                INC B
 668  F79B F7 E8 70          STA B  CRTC
 669  F79E B6 E5 76          LDA A  DPAGE+1
 670  F7A1 B7 E8 71          STA A  CRTC+1
 671
 672  F7A4 39                RTS
 673
 674                  * DELETE LINE ROUTINE
 675  F7A5 C6 50     DELLIN  LDA B  #80       80 CHAR. PER LINE
 676  F7A7 08        DELL11  INX
 677  F7A8 FF E5 79          STX    TEMP1     SAVE X
 678  F7AB FF E5 7B          STX    TEMP2
 679  F7AE B6 E5 7B          LDA A  TEMP2
 680  F7B1 84 07             AND A  #7        MODULO 2K
 681  F7B3 8B EC             ADD A  #HADRF    PHYSICAL ADDRESS OF REFRESH MEMORY
 682  F7B5 88 1C             EOR A  #$1C
 683  F7B7 B7 E5 7B          STA A  TEMP2
 684  F7BA FE E5 7B          LDX    TEMP2     ABSOLUTE ADDRESS
 685  F7BD B6 20             LDA A  #$20
 686  F7BF BA E5 70          ORA A  VIDEO
 687  F7C2 A7 00             STA A  0,X
 688  F7C4 FE E5 79          LDX    TEMP1     RESTORE X
 689  F7C7 5A                DEC B
 690  F7C8 26 DD             BNE    DELL11    LOOP IF B NOT ZERO
 691  F7CA 39                RTS
 692
 693                  *
 694                  * SCROLL DOWN
 695                  *
 696  F7CB B6 E5 75  SCROLD  LDA A  DPAGE
 697  F7CE F6 E5 76          LDA B  DPAGE+1
 698  F7D1 C0 50             SUB B  #80
 699  F7D3 82 00             SBC A  #0
 700  F7D5 84 07             AND A  #7
 701  F7D7 B7 E5 75          STA A  DPAGE
 702  F7DA F7 E5 76          STA B  DPAGE+1
 703  F7DD FE E5 75          LDX    DPAGE
 704  F7E0 09                DEX
 705  F7E1 8D C2             BSR    DELLIN    ERASE NEW LINE
 706
 707                  * UPDATE END ADDRESS
 708  F7E3 B6 E5 77          LDA A  EPAGE
 709  F7E6 F6 E5 78          LDA B  EPAGE+1
 710  F7E9 C0 50             SUB B  #80
 711  F7EB 82 00             SBC A  #0
 712  F7ED 84 07             AND A  #7
 713  F7EF B7 E5 77          STA A  EPAGE
 714  F7F2 F7 E5 78          STA B  EPAGE+1
 715  F7F5 20 98             BRA    UCRTC
 716
 717                  *
 718                  * OUTPUT CHARACTER CONTAINED IN REG-A
 719                  *
```

```
2480 SCREEN HANDLER                    9-12-81  ASSEMBLEUR TSC 6800   PAGE   17

 720  F7F7 FF E5 7F  OUTPUT  STX    XREG
 721  F7FA 36                PSH A
 722  F7FB 37                PSH B
 723  F7FC 7D E4 16          TST    ESCFG
 724  F7FF 26 50             BNE    ESCS
 725  F801 4D        OUTPT1  TST A
 726  F802 27 27             BEQ    CRTRET
 727  F804 81 9F             CMP A  #$9F
 728  F806 22 23             BHI    CRTRET
 729  F808 81 1F             CMP A  #$1F
 730  F80A 23 25             BLS    FUNCT
 731  F80C F6 E5 73          LDA B  CURAD
 732  F80F C4 07             AND B  #7
 733  F811 CB EC             ADD B  #HADRF
 734  F813 C8 1C             EOR B  #$1C
 735  F815 F7 E5 79          STA B  TEMP1
 736  F818 F6 E5 74          LDA B  CURAD+1
 737  F81B F7 E5 7A          STA B  TEMP1+1
 738  F81E FE E5 79          LDX    TEMP1
 739  F821 84 7F             AND A  #$7F
 740  F823 BA E5 70          ORA A  VIDEO
 741  F826 A7 00             STA A  0,X
 742  F828 BD F6 DF          JSR    INCCL
 743  F82B FE E5 7F  CRTRET  LDX    XREG
 744  F82E 33                PUL B
 745  F82F 32                PUL A
 746  F830 39                RTS
 747
 748                    *
 749                    * FUNCTIONS
 750                    *
 751  F831 CE F9 91  FUNCT   LDX    #FCTCRT
 752  F834 A1 00     FUNCT2  CMP A  0,X
 753  F836 27 0A             BEQ    FUNCT1
 754  F838 08                INX
 755  F839 08                INX
 756  F83A 08                INX
 757  F83B 8C F9 AC          CPX    #ENFCT
 758  F83E 26 F4             BNE    FUNCT2
 759  F840 20 E9             BRA    CRTRET
 760
 761  F842 EE 01     FUNCT1  LDX    1,X
 762  F844 AD 00             JSR    0,X
 763  F846 20 E3             BRA    CRTRET
 764
 765                    *
 766                    * ESCAPE SEQUENCE
 767                    *
 768  F848 7C E4 17  CPOSS   INC    CPOSFG
 769  F84B 86 FF     ESCAPE  LDA A  #$FF
 770  F84D B7 E4 16          STA A  ESCFG
 771  F850 39                RTS
 772  F851 7D E4 17  ESCS    TST    CPOSFG
 773  F854 2B 2D             BNE    CPOS
 774  F856 7F E4 16          CLR    ESCFG
 775  F859 CE F9 AC          LDX    #ESCTBL
```

```
2480 SCREEN HANDLER                    9-12-81  ASSEMBLEUR TSC 6800   PAGE   18

 776  F85C A1 00     ESCSQ   CMP A  0,X
 777  F85E 27 E2             BEQ    FUNCT1
 778  F860 08                INX
 779  F861 08                INX
 780  F862 08                INX
 781  F863 8C F9 D6          CPX    #ENTBL
 782  F866 26 F4             BNE    ESCSQ
 783  F868 20 97             BRA    OUTPT1
 784
 785                 * CURSOR OFF
 786  F86A 86 20     CUROFF  LDA A  #$20
 787  F86C C6 0A     CDISP   LDA B  #10
 788  F86E F7 E8 70          STA B  CRTC
 789  F871 B7 E8 71          STA A  CRTC+1
 790  F874 39                RTS
 791
 792                 * CURSOR ON
 793  F875 86 61     CURON   LDA A  #$61
 794  F877 20 F3             BRA    CDISP
 795
 796                 * PAGE MODE
 797  F879 86 FF     PAGMOD  LDA A  #$FF
 798  F87B B7 E4 18          STA A  PAGFG
 799  F87E 39                RTS
 800
 801                 * ROLL MODE
 802  F87F 7F E4 18  ROLMOD  CLR    PAGFG
 803  F882 39                RTS
 804
 805                 * PUT CURSOR AT GIVEN LOCATION
 806  F883 81 60     CPOS    CMP A  #96       OFFSET ?
 807  F885 23 02             BLS    CPOS6     BRANCH IF NOT
 808  F887 80 60             SUB A  #96
 809  F889 4A        CPOS6   DEC A
 810  F88A F6 E4 17          LDA B  CPOSFG
 811  F88D C1 02             CMP B  #2
 812  F88F 27 0E             BEQ    CPOS1
 813  F891 81 17             CMP A  #23
 814  F893 23 02             BLS    CPOS3
 815  F895 86 17             LDA A  #23
 816  F897 B7 E5 71  CPOS3   STA A  LINE
 817  F89A 7C E4 17          INC    CPOSFG
 818  F89D 20 12             BRA    CRTRT1
 819  F89F 81 4F     CPOS1   CMP A  #79
 820  F8A1 23 02             BLS    CPOS4
 821  F8A3 86 4F             LDA A  #79
 822  F8A5 B7 E5 72  CPOS4   STA A  COLUMN
 823  F8A8 7F E4 16          CLR    ESCFG
 824  F8AB 7F E4 17          CLR    CPOSFG
 825  F8AE BD F7 37          JSR    MOVEA
 826  F8B1 7E F8 28  CRTRT1  JMP    CRTRET
 827
 828                 * INVERSE VIDEO
 829  F8B4 7F E5 70  INVERS  CLR    VIDEO
 830  F8B7 39                RTS
 831  F8B8 86 80     SINVER  LDA A  #$80
```

```
2480 SCREEN HANDLER                9-12-81  ASSEMBLEUR TSC 6800   PAGE   19

 832  F8BA B7 E5 70          STA A  VIDEO
 833  F8BD 39                RTS
 834
 835                 * ECHO OFF
 836  F8BE 86 FF     ECHOFF  LDA A  #$FF      SET NO ECHO FLAG
 837  F8C0 B7 E4 00          STA A  OUTSW
 838  F8C3 39                RTS
 839
 840                 * ECHO ON
 841  F8C4 7F E4 00  ECHON   CLR    OUTSW     SET ECHO FLAG
 842  F8C7 39                RTS
 843
 844                 * ERASE LINE
 845  F8C8 B6 E5 73  ERLIN   LDA A  CURAD     BEGINNING ADDRESS OF LINE
 846  F8CB F6 E5 74          LDA B  CURAD+1
 847  F8CE F0 E5 72          SUB B  COLUMN
 848  F8D1 82 00             SBC A  #0
 849  F8D3 B7 E5 79          STA A  TEMP1
 850  F8D6 F7 E5 7A          STA B  TEMP1+1
 851  F8D9 FE E5 79          LDX    TEMP1
 852  F8DC 09                DEX
 853  F8DD 7E F7 A5          JMP    DELLIN    GOTO ERASE LINE AND RETURN
 854
 855                 * ERASE TO END OF LINE
 856  F8E0 C6 50     EREOL   LDA B  #80
 857  F8E2 F0 E5 72          SUB B  COLUMN    NUMBER OF CHAR. TO ERASE
 858  F8E5 FE E5 73          LDX    CURAD     CURSOR ADDRESS
 859  F8E8 09                DEX
 860  F8E9 7E F7 A7          JMP    DELLI1    GOTO ERASE TO END OF LINE
 861
 862                 * ERASE TO END OF SCREEN
 863  F8EC 8D F2     EREOS   BSR    EREOL     ERASE TO END OF LINE
 864  F8EE 86 17             LDA A  #23
 865  F8F0 B0 E5 71          SUB A  LINE      NUMBER OF LINE TO ERASE
 866  F8F3 27 0B             BEQ    EREOS2
 867  F8F5 36        EREOS1  PSH A
 868  F8F6 FE E5 79          LDX    TEMP1
 869  F8F9 8D F7 95          JSR    DELLIN    ERASE LINE
 870  F8FC 32                PUL A
 871  F8FD 4A                DEC A
 872  F8FE 25 F5             BNE    EREOS1    CONTINUE TILL NOT ZERO
 873  F900 39        EREOS2  RTS               RETURN
 874
 875                 *
 876                 * HARD COPY ON PARALLEL PRINTER
 877                 *
 878  F901 FE E5 75  HCPY    LDX    DPAGE     BEGINNING OF PAGE
 879  F904 FF E5 79          STX    TEMP1
 880  F907 86 18             LDA A  #24       NUMBER OF LINE
 881  F909 B7 E5 7D          STA A  LINCNT
 882  F90C 86 50     HCPY1   LDA A  #80       NUMBER OF COLUMN
 883  F90E B7 E5 7E          STA A  COLCNT
 884  F911 FF E5 7B  HCPY2   STX    TEMP2     LOGICAL ADDRESS
 885  F914 B6 E5 7B          LDA A  TEMP2
 886  F917 84 07             AND A  #7        MODULO 2K
 887  F919 8B EC             ADD A  #HADRF    ADDRESS OF REFRESH MEMORY
```

```
2400 SCREEN HANDLER                    9-12-81  ASSEMBLEUR TSC 6800   PAGE   20

  888  F91B 88 1C           EOR A  £$1C
  889  F91D B7 E5 7B        STA A  TEMP2     ADDRESS OF CHAR.
  890  F920 FE E5 7B        LDX    TEMP2
  891  F923 A6 00           LDA A  0,X       GET CHAR.
  892  F925 84 7F           AND A  £$7F      MASK VIDEO
  893  F927 81 1F           CMP A  £$1F      ASCII ?
  894  F929 22 02           BHI    HCPY3     BRANCH IF YES
  895  F92B 86 2D           LDA A  £'-       NOT ASCII
  896  F92D BD FC 8F  HCPY3  JSR    POUT      OUTPUT CHAR.
  897  F930 FE E5 79        LDX    TEMP1     INC LOGICAL ADDRESS
  898  F933 08              INX
  899  F934 FF E5 79        STX    TEMP1
  900  F937 7A E5 7E        DEC    COLCNT    DEC COLUMN COUNT
  901  F93A 26 D5           BNE    HCPY2     BRANCH IF NOT ZERO
  902  F93C 86 0D           LDA A  £VCR      NEW LINE
  903  F93E BD FC 8F        JSR    POUT      OUTPUT CR LF
  904  F941 86 0A           LDA A  £VLF
  905  F943 BD FC 8F        JSR    POUT
  906  F946 7A E5 7D        DEC    LINCNT    DEC LINE COUNT
  907  F949 26 C1           BNE    HCPY1     CONTINUE IF NOT ZERO
  908  F94B 86 0C           LDA A  £12       OUTPUT FORM FEED
  909  F94D BD FC 8F        JSR    POUT
  910  F950 39              RTS              RETURN
  911
  912                 *
  913                 * VERSION
  914                 *
  915  F951 CE F9 81  RESERV LDX    £VERSIO
  916  F954 FF E5 7B  RESV1  STX    TEMP2
  917  F957 A6 00           LDA A  0,X
  918  F959 81 04           CMP A  £4
  919  F95B 27 23           BEQ    RESV2
  920  F95D F6 E5 73        LDA B  CURAD
  921  F960 C4 07           AND B  £7
  922  F962 CB EC           ADD B  £RADRF
  923  F964 C8 1C           EOR B  £$1C
  924  F966 F7 E5 79        STA B  TEMP1
  925  F969 F6 E5 74        LDA B  CURAD+1
  926  F96C F7 E5 7A        STA B  TEMP1+1
  927  F96F FE E5 73        LDX    TEMP1
  928  F972 8A E5 70        ORA A  VIDEO
  929  F975 A7 00           STA A  0,X
  930  F977 BD F6 DF        JSR    INDCL
  931  F97A FE E5 7B        LDX    TEMP2
  932  F97D 08              INX
  933  F97E 20 D4           BRA    RESV1
  934  F980 39        RESV2 RTS
  935  F981 47        VERSIO FCC    'BOUPIL 1.4 8/21'
  936  F990 04              FCB    4
  937
  938                 *
  939                 * FUNCTIONS TABLE
  940                 *
  941  F991 27        FCTCRT FCB    $7        BELL
  942  F992 F6 BE           FDB    BELL
  943  F994 08              FCB    $8        CURSOR LEFT
```

```
2400 SCREEN HANDLER                    9-12-81  ASSEMBLEUR TSC 6800   PAGE   21

  944  F993 F6 F2           FDB    DECCL
  945  F997 1D              FCB    $1D       CURSOR RIGHT
  946  F998 F6 DF           FDB    INCCL
  947  F99A 0A              FCB    $A        CURSOR DOWN
  948  F99B F7 00           FDB    INCLN
  949  F99D 0B              FCB    $B        CURSOR UP
  950  F99E F7 22           FDB    DECLN
  951  F9A0 0C              FCB    $C        CLEAR
  952  F9A1 F6 D5           FDB    CLEAR
  953  F9A3 0D              FCB    $D        LINE RETURN
  954  F9A4 F6 DA           FDB    LINRET
  955  F9A6 1B              FCB    $1B       ESCAPE
  956  F9A7 F8 4B           FDB    ESCAPE
  957  F9A9 1C              FCB    $1C       HOME
  958  F9AA F6 D7           FDB    HOME
  959  F9AC          ENFCT  EQU    *
  960
  961                *
  962                * ESCAPE SEQUENCE TABLE
  963                *
  964  F9AC 49       ESCTBL FCC    'I' INVERSE VIDEO
  965  F9AD F8 B4           FDB    INVERS
  966  F9AF 4A              FCC    'J' NORMAL VIDEO
  967  F9B0 F8 B8           FDB    SINVER
  968  F9B2 50              FCC    'P' PAGE MODE
  969  F9B3 F8 79           FDB    PAGMOD
  970  F9B5 52              FCC    'R' ROLL MODE
  971  F9B6 F8 7F           FDB    ROLMOD
  972  F9B8 45              FCC    'E' CURSOR OFF
  973  F9B9 F8 6A           FDB    CUROFF
  974  F9BB 43              FCC    'C' CURSOR ON
  975  F9BC F8 75           FDB    CURON
  976  F9BE 3D              FCC    '=' MOVE CURSOR
  977  F9BF F8 48           FDB    CPOSS
  978  F9C1 48              FCC    'H' ECHO ON
  979  F9C2 F8 C4           FDB    ECHON
  980  F9C4 47              FCC    'G' ECHO OFF
  981  F9C5 F8 BE           FDB    ECHOFF
  982  F9C7 4C              FCC    'L' ERASE LINE
  983  F9C8 F8 C8           FDB    ERLIN
  984  F9CA 58              FCC    'X' ERASE TO END OF LINE
  985  F9CB F8 E0           FDB    EREOL
  986  F9CD 59              FCC    'Y' ERASE TO END OF PAGE
  987  F9CE F8 EC           FDB    EREOS
  988  F9D0 44              FCC    'D' HARD COPY
  989  F9D1 F9 21           FDB    HCPY
  990  F9D3 5A              FCC    'Z' RESERVED
  991  F9D4 F9 51           FDB    RESERV
  992  F9D6          ENTBL  EQU    *
  993
  994                *
  995                * RESERVED  RAM
  996                *
  997  E416                 ORG    $E416
  998  E416          ESCFG  RMB    1
  999  E417          CPOSFG RMB    1
```

```
2480 SCREEN HANDLER                9-12-81  ASSEMBLEUR TSC 6800   PAGE   22

 1000  E41E           PAGFG   RMB    1
 1001  E570                   ORG    $E570
 1002  E570           VIDEO   RMB    1
 1003  E571           LINE    RMB    1
 1004  E572           COLUMN  RMB    1
 1005  E573           CURAD   RMB    2
 1006  E575           DPAGE   RMB    2
 1007  E577           EPAGE   RMB    2
 1008  E579           TEMP1   RMB    2
 1009  E57B           TEMP2   RMB    2
 1010  E57D           LINCNT  RMB    1
 1011  E57E           COLCNT  RMB    1
 1012  E57F           XREG    RMB    2
```

```
1014  F9D6                ORG    $F9D6
1015                      NAM    KEYBOARD HANDLER
1017                      OPT    PAG
1018
1019                ********************************
1020                *                              *
1021                *        KEYBOARD HANDLER      *
1022                *                              *
1023                ********************************
1024
1025
1026                *
1027                *     I/O ROUTINES
1028                *     -----------
1029                *
1030                *        INITCV    INITIALIZATIONS
1031                *        TINPUT    CHECK FOR TYPED CHARACTER
1032                *        INPUT     INPUT ONE CHARACTER
1033                *
1034                *
1035                * EQUATES
1036                *
1037  E840          VIACL   EQU    $E840
1038  E820          DATA1   EQU    $E820
1039  E830          DATA2   EQU    $EE30
1040  E821          CMDE1   EQU    $E821
1041  E831          CMDE2   EQU    $E831
1042  E860          VIAMUS  EQU    $E860
1043  EBF8          SETMEM  EQU    $EBF8
1044  0000          ORB     EQU    0
1045  0001          ORA     EQU    1
1046  0002          DDRB    EQU    2
1047  0003          DDRA    EQU    3
1048  0004          TILL    EQU    4
1049  000C          PCR     EQU    $C
1050  000D          IFR     EQU    $D
1051  000E          IER     EQU    $E
1052  00A1          SHIFT   EQU    $A1       SHIFT CODE
1053  00A0          REPT    EQU    $A0       REPEAT CODE
```

```
1055                    *
1056                    * KEYBOARD 1 TABLE DEFINITION
1057                    *
1058  F9D6              ADTAB1  EQU   *
1059  F9D6 1A                   FCB   $1A,$04,$05,$13,$8E,$03,$95,$18
1060  F9DE 12                   FCB   $12,$07,$14,$06,$88,$02,$87,$16
1061  F9E6 19                   FCB   $19,$0A,$15,$08,$8A,$9D,$89,$0E
1062  F9EE 09                   FCB   $09,$11,$01,$00,$84,$17,$F1,$A1
1063  F9F6 03                   FCB   $09,$0C,$0F,$0B,$8C,$9B,$8B,$9E
1064  F9FE 0A                   FCB   $0A,$0D,$00,$7F,$08,$00,$1F,$20
1065  FA06 10                   FCB   $10,$1C,$1D,$0D,$9C,$00,$00,$1E
1066  FA0E 00                   FCB   $00,$A0,$00,$00,$00,$00,$00,$00
1067
1068  FA16 1A                   FCB   $1A,$04,$05,$13,$86,$03,$85,$18
1069  FA1E 12                   FCB   $12,$07,$14,$06,$88,$02,$87,$16
1070  FA26 19                   FCB   $19,$0A,$15,$08,$8A,$9D,$89,$0E
1071  FA2E 09                   FCB   $09,$11,$01,$00,$84,$17,$F1,$A1
1072  FA36 09                   FCB   $09,$0C,$0F,$0B,$8C,$9B,$8B,$9E
1073  FA3E 0A                   FCB   $0A,$0D,$00,$7F,$08,$00,$1F,$20
1074  FA46 10                   FCB   $10,$1C,$1D,$0D,$9C,$00,$00,$1E
1075  FA4E 00                   FCB   $00,$A0,$00,$00,$00,$00,$00,$00
1076
1077  FA56 5A                   FCB   $5A,$44,$45,$53,$33,$43,$32,$58
1078  FA5E 52                   FCB   $52,$47,$54,$46,$35,$42,$34,$56
1079  FA66 59                   FCB   $59,$4A,$55,$48,$37,$3F,$36,$4E
1080  FA6E 09                   FCB   $09,$51,$41,$00,$31,$57,$F1,$A1
1081  FA76 49                   FCB   $49,$4C,$4F,$4B,$39,$2F,$38,$2E
1082  FA7E 0A                   FCB   $0A,$0D,$00,$7F,$08,$00,$3F,$20
1083  FA86 30                   FCB   $30,$25,$7E,$4D,$81,$00,$30,$2B
1084  FA8E 00                   FCB   $00,$A0,$00,$20,$00,$20,$00,$00
1085
1086  FA96 7A                   FCB   $7A,$64,$65,$73,$22,$63,$7B,$78
1087  FA9E 72                   FCB   $72,$67,$74,$66,$28,$62,$27,$76
1088  FAA6 79                   FCB   $79,$6A,$75,$68,$7D,$2C,$83,$6E
1089  FAAE 05                   FCB   $05,$71,$61,$00,$26,$77,$F1,$A1
1090  FAB6 55                   FCB   $55,$6C,$6F,$69,$82,$3A,$21,$3B
1091  FABE 0A                   FCB   $0A,$0D,$00,$7F,$08,$00,$20,$20
1092  FAC6 70                   FCB   $70,$7C,$5E,$60,$19,$00,$40,$3D
1093  FACE 00                   FCB   $00,$A0,$00,$00,$20,$20,$00,$00
1094                  *
1095                  * KEYBOARD 2 TABLE DEFINITION
1096                  *
1097  FAD6              ADTAB2  EQU   *
1098  FAD6 24                   FCB   $24,$F2,$0B,$F7,$0C,$00,$0A,$F6
1099  FADE 22                   FCB   $00,$20,$00,$F5,$23,$00,$00,$00
1100  FAE6 5C                   FCB   $5C,$00,$20,$20,$02,$F4,$30,$00
1101  FAEE 0D                   FCB   $0C,$20,$31,$34,$38,$F3,$25,$37
1102  FAF6 25                   FCB   $25,$23,$2D,$2A,$18,$00,$00,$2F
1103  FAFE 56                   FCB   $56,$2F,$00,$F9,$1D,$00,$00,$F3
1104  FB06 2A                   FCB   $2A,$2D,$33,$33,$00,$00,$00,$39
1105  FB0E 23                   FCB   $23,$18,$32,$35,$00,$20,$2D,$30
```

```
KEYBOARD HANDLER                    9-12-81  ASSEMBLER TSC 6800   PAGE   25

                        * I/O INITIALIZATIONS
                        ********************
                        *
                        *
FB16 8D F6 71  INIT     JSR    INITVS    INITIALIZE SCREEN
FB19 86 01             LDA A  #1
FB1B B7 EB F3          STA A  SETMEM
                        *
                        * VIA FOR BELL
FB1E 86 7F     INITCV  LDA A  #$7F
FB20 B7 E9 6E          STA A  VIAMUS+IER
FB23 86 FF             LDA A  #$FF
FB25 B7 E9 6C          STA A  VIAMUS+PCR
FB28 B7 E9 E3          STA A  VIAMUS+DDRA
                        *
                        * 8279
                        *
                        * ENCODED KEYBOARD N-KEY ROLLOVER
                        *
FB2B 86 02     INIT82  LDA A  #2
FB2D B7 EB 21          STA A  CMDE1
FB30 B7 EB 31          STA A  CMDE2
FB33 86 01             LDA A  #$01
FB35 B7 E8 21          STA A  CMDE1
FB38 B7 E8 31          STA A  CMDE2
FB3B 86 2F             LDA A  #$2F
FB3D B7 E8 21          STA A  CMDE1
FB40 B7 EB 31          STA A  CMDE2
FB43 8E A0             LDA A  #$A0
FB45 B7 E8 21          STA A  CMDE1
                        *
                        * PARALLEL I/O PORT
FB48 4F        INITVC  CLR A
FB49 B7 E9 43          STA A  VIACL+DDRA
FB4C 43                COM A
FB4D B7 E9 42          STA A  VIACL+DDRB
FB50 86 C3             LDA A  #$C3
FB52 B7 E9 4C          STA A  VIACL+PCR
FB55 86 E3             LDA A  #$E3
FB57 B7 E9 4C          STA A  VIACL+PCR
                        *
                        * WAIT FOR KEYBOARD
FB5A C6 2D             LDA B  #2D
FB5C 86 02     STAB1   LDA A  #$02
FB5E B7 E8 21          STA A  CMDE1
FB61 B7 E8 31          STA A  CMDE2
FB64 CE 10 00          LDX    #$1000
FB67 09        STAB1   DEX
FB68 26 FD             BNE    STAB1
FB6A 5A                DEC B
FB6B 26 EF             BNE    STAB1
FB6D 39                RTS
```

```
KEYBOARD HANDLER                    9-12-81  ASSEMBLEUR TSC 6800   PAGE   26


1160                    *
1161                    * GET THE CHARACTER TYPED ON KEYBOARD
1162                    *
1163  FB6E B6 E4 C2     INPUT   LDA A  SAVEA     GET CHARACTER
1164  FB71 7F E4 13             CLR    FLGIN     CLEAR FLAG
1165  FB74 39                   RTS              RETURN
1166                    *
1167                    * REPEAT PROCESSING
1168                    *
1169  FB75 C6 47        REPTA   LDA B  #$47      READ FIFO SENSOR RAM AT ROW 7
1170  FB77 F7 E8 21             STA B  CMDE1
1171  FB7A CE 03 00             LDX    #$300     DELAY AFTER COMMAND
1172  FB7D 09           REPTA4  DEX
1173  FB7E 26 FD                BNE    REPTA4
1174  FB80 F6 E8 20             LDA B  DATA1     GET DATA
1175  FB83 C4 02                AND B  #2        CHECK FOR REPEAT CODE
1176  FB85 27 0B                BEQ    REPTA2    NO MORE REPEAT
1177  FB87 CE 20 00             LDX    #$2000    DELAY
1178  FB8A 09           REPTA1  DEX
1179  FB8B 26 FD                BNE    REPTA1
1180  FB8D 73 E4 13             COM    FLGIN     SET INPUT FLAG
1181  FB90 20 08                BRA    REPTA3
1182  FB92 C6 02        REPTA2  LDA B  #2        SET ENCODED SCAN KEYBOARD
1183  FB94 F7 E8 21             STA B  CMDE1
1184  FB97 7F E4 12             CLR    FLGRPT    CLEAR REPEAT FLAG
1185  FB9A C6 C1        REPTA3  LDA B  #$C1      CLEAR INTEL 8279
1186  FB9C F7 E8 21             STA B  CMDE1
1187  FB9F 20 11                BRA    TINPT1    RETURN
1188                    *
1189                    * CHECK IF A CHARACTER HAS BEEN TYPED
1190                    *
1191  FBA1 7D E4 13     TINPUT  TST    FLGIN     CHARACTER ALREADY PRESENT
1192  FBA4 26 14                BNE    TINPT2    YES RETURN
1193  FBA6 FF E4 C3             STX    SAVEX     SAVE X
1194  FBA9 36                   PSH A            SAVE A
1195  FBAA 37                   PSH B            SAVE B
1196  FBAB 7D E4 12             TST    FLGRPT    REPEAT ?
1197  FBAE 26 C5                BNE    REPTA     YES GO PROCESS REPEAT
1198  FBB0 8D 09                BSR    KEYB      CHECK AND READ KEYBOARD
1199  FBB2 FE E4 C3     TINPT1  LDX    SAVEX     RESTORE X
1200  FBB5 7D E4 13             TST    FLGIN     SET Z
1201  FBB8 33                   PUL B            RESTORE B
1202  FBB9 32                   PUL A            RESTORE A
1203  FBBA 39           TINPT2  RTS              RETURN
1204                    *
1205                    * CHECK AND READ KEYBOARD
1206                    *
1207  FBBB B6 E8 21     KEYB    LDA A  CMDE1     CHECK KEYBOARD 1
1208  FBBE 84 0F                AND A  #$F
1209  FBC0 26 11                BNE    KEYB1     WE HAVE CHAR.
1210  FBC2 B6 E8 31             LDA A  CMDE2     CHECK KEYBOARD 2
1211  FBC5 84 0F                AND A  #$F
1212  FBC7 27 3B                BEQ    KEYB4     NOTHING, RETURN
1213  FBC9 B6 E8 30             LDA A  DATA2     GET CHAR.
1214  FBCC CE FA D6             LDX    #ADTAB2   ADDRESS OF TABLE 2
```

```
1215  FBCF 84 3F          AND A  #$3F     ONLY 6 BITS
1216  FBD1 20 0D          BRA    KEYB2
1217  FBD3 B6 E8 20 KEYB1 LDA A  DATA1    GET CHAR.
1218  FBD6 CE F9 D6       LDX    #ADTAB1  ADDRESS OF TABLE 1
1219  FBD9 7D E4 11       TST    FLGSHF   CHECK SHIFT LOCK
1220  FBDC 26 02          BNE    KEYB2
1221  FBDE 84 BF          AND A  #$BF     IF LOCKED,MASK THE BIT
1222  FBE0 FF E4 C0 KEYB2 STX    PTTAB
1223  FBE3 BB E4 C1       ADD A  PTTAB+1  ADDRESS OF CHAR. IN THE TABLE
1224  FBE6 24 03          BCC    KEYB3
1225  FBE8 7C E4 C0       INC    PTTAB    ADD CARRY IF ANY
1226  FBEB B7 E4 C1 KEYB3 STA A  PTTAB+1
1227  FBEE FE E4 C0       LDX    PTTAB    GET ADDRESS OF CHAR.
1228  FBF1 A6 00          LDA A  0,X      GET CODE
1229  FBF3 B7 E4 C2       STA A  SAVEA    SAVE IT
1230  FBF5 81 A1          CMP A  #SHIFT   SHIFT KEY ?
1231  FBF8 27 0B          BEQ    GSHIFT   YES GO PROCESS IT
1232  FBFA 81 A0          CMP A  #REPT    REPEAT KEY ?
1233  FBFC 27 15          BEQ    GREPT    IF YES GO PROCESS IT
1234  FBFE 73 E4 13       COM    FLGIN    SET INPUT FLAG
1235  FC01 39       KEYB4 RTS              RETURN
1236                *
1237                * SHIFT LOCK PROCESSING
1238                *
1239  FC02 7D E4 11 GSHIFT TST    FLGSHF   CHECK FLAG
1240  FC05 27 03          BEQ    GSHFT1   SKIP IF LOCKED
1241  FC07 86 00          LDA A  #$00     LIGHT ON
1242  FC09 0C             CLC    SKIP2
1243  FC0A 86 A3    GSHFT1 LDA A  #$A3     LIGHT OFF
1244  FC0C B7 E8 21       STA A  CMDE1
1245  FC0F 73 E4 11       COM    FLGSHF   SWITCH THE FLAG
1246  FC12 39             RTS              RETURN
1247                *
1248                * FIRST TIME REPEAT HAS BEEN TYPED
1249                *
1250  FC13 8D A5    GREPT BSR    KEYB     CHECK AND READ KEYBOARD
1251  FC15 7D E4 13       TST    FLGIN    CHAR. TYPED ?
1252  FC18 27 F9          BEQ    GREPT    NO,WAIT
1253  FC1A C6 04          LDA B  #4       YES SET ENCODED SCAN SENSOR MATRIX
1254  FC1C F7 E8 21       STA B  CMDE1
1255  FC1F 86 FF          LDA A  #$FF     SET REPEAT FLAG
1256  FC21 B7 E4 12       STA A  FLGRPT
1257  FC24 39             RTS              RETURN
1258                *
1259                * RAM RESERVED TO VARIABLE INFORMATIONS
1260                *
1261  E411                ORG    $E411    INITIALIZED TO ZERO AT START
1262  E411          FLGSHF RMB    1
1263  E412          FLGRPT RMB    1
1264  E413          FLGIN  RMB    1
1265  E4C0                ORG    $E4C0
1266  E4C0          PTTAB  RMB    2
1267  E4C2          SAVEA  RMB    1
1268  E4C3          SAVEX  RMB    2
1269  E4C5          VTEMP  RMB    2
1270
```



```
1271                      END
```

```
KEYBOARD HANDLER                    9-12-81  ASSEMBLEUR TSC 5800   PAGE   29


1274                        NAM    GPMON V1.4
1275                        OPT    PAG
1276  FC25                  ORG    $FC25
1277
1278                  ************************
1279                  *                      *
1280                  *    TERMINAL MODE     *
1281                  *                      *
1282                  ************************
1283
1284                  *
1285                  * EQUATES
1286                  *
1287  E80C            ACIAC  EQU    $E80C     ACIA COMMAND REGISTER
1288  E80D            ACIAD  EQU    ACIAC+1   ACIA DATA REGISTER
1289  EB10            VIAC   EQU    $EB10     CPU VIA FOR ACIA EXTERNAL CLOCK
1290  0008            ACR    EQU    $8        VIA AUXILIARY CONTROL REGISTER
1291  0004            TILL   EQU    4         VIA T1 LOW ORDER LATCH
1292                  *
1293                  * INITIALIZE ACIA (CPU)
1294                  *    S:=ACIA COMMAND
1295                  *    X:=CLOCK VIA
1296                  *
1297  FC25 C6 03      INICIA LDA B  #3        MASTER RESET CODE
1298  FC27 F7 E8 0C          STA B  ACIAC     RESET ACIA
1299  FC2A 97 E8 0C          STA A  ACIAC     INITIALIZE ACIA
1300  FC2D 86 C0             LDA A  #$C0      INT CLOCK FOR ACIA
1301  FC2F B7 E8 1B          STA A  VIAC+ACR
1302  FC32 FF E8 14          STX    VIAC+TILL INITIALIZE CLOCK
1303  FC35 39                RTS              RETURN
1304                  *
1305                  * OUTPUT ONE CHARACTER TO ACIA
1306                  *
1307  FC36 37         TOACIA PSH B            SAVE B-REG
1308  FC37 F6 E8 0C   OUTCI  LDA B  ACIAC     TEST READY
1309  FC3A 57                ASR B
1310  FC3B 57                ASR B
1311  FC3C 24 F9             BCC    OUTCI    XMIT NOT READY
1312  FC3E B7 E8 0D          STA A  ACIAD     OUTPUT CHAR.
1313  FC41 33                PUL B           RESTORE B-REG
1314  FC42 39                RTS              RETURN
1315                  *
1316                  * INPUT ONE CHARACTER FROM ACIA
1317                  *
1318  FC43 8D 03      INACIA BSR    CHKACI   CHECK FOR RECEIVED CHAR.
1319  FC45 27 FC             BEQ    INACIA   NOT READY
1320  FC47 B6 E8 0D          LDA A  ACIAD    INPUT CHAR.
1321  FC4A 84 7F             AND A  #$7F     RESET PARITY BIT
1322  FC4C 39                RTS              RETURN
1323                  *
1324                  * CHECK FOR TYPED CHARACTER
1325                  *
1326  FC4D B6 E8 0C   CHKACI LDA A  ACIAC     READ STATUS
1327  FC50 85 01             BIT A  #1        SHOW READY OR NOT READY
1328  FC52 39                RTS              RETURN
```

```
GPMON V1.4                          9-12-81  ASSEMBLEUR TSC 6800   PAGE   30

1329                    *
1330                    * PROGRAM STARTS HERE
1331                    *
1332  FC53 8D FD CC     COM     JSR    PCRLF
1333  FC56 CE FC 8C             LDX    #QUEST    ASK FOR SPEED
1334  FC59 8D FD 91             JSR    PDATA1
1335  FC5C 8D FD 9B             JSR    INCH      INPUT SPEED
1336  FC5F 16                   TAB              SAVE IT IN B
1337  FC60 8D FD CC             JSR    PCRLF     OUTPUT CR/LF
1338  FC63 C1 31                CMP B  #$31
1339  FC65 27 09                BEQ    COM1      300BDS
1340  FC67 C1 32                CMP B  #$32
1341  FC69 26 E8                BNE    COM       NOT CORRECT
1342  FC6B CE 18 00             LDX    #$1800    1200 BDS
1343  FC6E 20 03                BRA    COM2
1344  FC70 CE 64 00     COM1    LDX    #$6400    300 BDS
1345  FC73 86 01        COM2    LDA A  #1        EVEN PARITY,2 STOP BITS,CLOCK/16
1346  FC75 8D AE                BSR    INICIA    INITIALIZE ACIA
1347                    *
1348  FC77 8D FB A1     TERM    JSR    CHKCHR    CHECK FOR TYPED CHAR.
1349  FC7A 27 05                BEQ    TERM1     NO CHAR.
1350  FC7C 8D FB EE             JSR    INPUT     INPUT CHAR.
1351  FC7F 8D B5                BSR    TOACIA    XMIT CHAR.
1352                    *
1353  FC81 8D C9        TERM1   BSR    CHKACI    CHECK FOR RECEIVED CHAR.
1354  FC83 27 F2                BEQ    TERM      NO CHAR.
1355  FC85 8D BC                BSR    INACIA    GET RECEIVED CHAR.
1356  FC87 8D FD B2             JSR    OUTCH     OUTPUT IT
1357  FC8A 20 EB                BRA    TERM      CONTINUE
1358                    *
1359  FC8C 56           QUEST   FCC    'V='
1360  FC8E 04                   FCB    4
```

```
GPMON V1.4                        9-12-81  ASSEMBLEUR TSC 6800   PAGE   31

1362                ********************************
1363                *                              *
1364                *    PARALLEL PRINTER HANDLER  *
1365                *                              *
1366                ********************************
1367
1368
1369                *
1370                * OUTPUT ONE CHARACTER TO PRINTER
1371                *
1372  FC8F 37       POUT    PSH B           SAVE B
1373  FC90 F6 E8 4D CPOUT   LDA B  VIACL+IFR WAIT FOR READY
1374  FC93 C5 10            BIT B  #$10
1375  FC95 27 F9            BEQ    CPOUT
1376  FC97 B7 E8 40         STA A  VIACL+ORB OUTPUT CHAR.
1377  FC9A C6 C3            LDA B  #$C3
1378  FC9C F7 E8 4C         STA B  VIACL+PCR HANDSHAKE
1379  FC9F C6 E3            LDA B  #$E3
1380  FCA1 F7 E8 4C         STA B  VIACL+PCR
1381  FCA4 33               PUL B           RESTORE B
1382  FCA5 39               RTS             RETURN
```

```
GPMON V1.4                        9-12-81  ASSEMBLEUR TSC 6800   PAGE   32

1384                ************************
1385                *                      *
1386                *      MONITOR         *
1387                *                      *
1388                ************************
1389
1390
1391                *
1392                * EQUATES
1393                *
1394  FCA6          ROM     EQU    $FCA6
1395  E400          RAM     EQU    $E400
1396  003F          SWI     EQU    $3F       SWI OP CODE
1397  0030          SZRCL   EQU    SP-NIO-2  SIZE OF RAM TO CLEAR
1398  000D          CR      EQU    $D        CARRIAGE RETURN
1399  000A          LF      EQU    $A        LINE FEED
1400  002B          PRPT    EQU    '+        PROMPT CHARACTER
1401  002E          DOWN    EQU    '.        MEMORY INCREMENT CHAR.
1402  002D          UP      EQU    '-        MEMORY DECREMENT CHAR.
1403                *
1404  EC00          K7MOD   EQU    $EC00     K7/MODEM HANDLER ADDRESS
1405  8000          BASIC   EQU    $8000     ROM BASIC EXECUTION ADDRESS
1406  FBA1          CHKCHR  EQU    TINPUT
```

```
GPMON V1.4                          9-12-81  ASSEMBLEUR TSC 6800   PAGE   33

 1408  FCA6                 ORG    ROM
 1409                *
 1410                * JUMP TABLE TO ROUTINES PERFORMING GPMON FUNCTIONS
 1411                *
 1412  FCA6          FCTABL EQU    *
 1413  FCA6 4D              FCC    /M/ -M- MEMORY CHANGE
 1414  FCA7 FD ED           FDB    CHANGE
 1415  FCA9 47              FCC    /G/ -G- GOTO ENTERED ADDRESS
 1416  FCAA FE 37           FDB    GOTO
 1417  FCAC 52              FCC    /R/ -R- PRINT STACK
 1418  FCAD FF 1E           FDB    PSTAK
 1419  FCAF 56              FCC    /V/ -V- SET A BREAKPOINT
 1420  FCB0 FE 2A           FDB    SETBRK
 1421  FCB2 55              FCC    /U/ -U- RESET A BREAKPOINT
 1422  FCB3 FE 1E           FDB    RSTBRK
 1423  FCB5 57              FCC    /W/ -W- DELETE ALL BREAKPOINTS
 1424  FCB6 FE 26           FDB    DELBRK
 1425  FCB8 43              FCC    /C/ -C- CONTINUE
 1426  FCB9 FE 43           FDB    CONT
 1427  FCBB 4E              FCC    /N/ -N- NEXT (TRACE 1 INSTRUCTION)
 1428  FCBC FE 4B           FDB    NEXT
 1429  FCBE 54              FCC    /T/ -T- TRACE N INSTRUCTIONS
 1430  FCBF FE 4F           FDB    TRACE
 1431  FCC1 42              FCC    /B/ -B- PRINT ALL BREAKS
 1432  FCC2 FE 23           FDB    PNTBRK
 1433  FCC4 44              FCC    /D/ -D- MEMORY DUMP
 1434  FCC5 FE 65           FDB    DUMP
 1435  FCC7 F3              FCB    $F3       -BOOT- EXECUTE DISK BOOTSTRAP
 1436  FCC8 F5 CC           FDB    DKBOOT
 1437  FCCA 4B              FCC    /K/ -K- LINK TO K7/MODEM HANDLER
 1438  FCCB FD 37           FDB    WHAT
 1439  FCCD 4F              FCC    /O/ -O- SWITCH OUTPUT TO PRINTER
 1440  FCCE FE 8C           FDB    SWITCH
 1441  FCD0 F1              FCB    $F1       -BASIC- EXECUTE ROM BASIC
 1442  FCD1 FD 37           FDB    WHAT
 1443  FCD3 F2              FCB    $F2       -COM- TERMINAL MODE
 1444  FCD4 FC 53           FDB    COM
 1445  FCD6 F4              FCB    $F4       -GR- GRAPHIC MODE
 1446  FCD7 FD 37           FDB    WHAT
 1447  FCD9          FCTBEN EQU    *
```

```
GPMON V1.4                    9-12-81  ASSEMBLEUR T6C 6800   PAGE   34

                        *
                        * I/O INTERRUPT SEQUENCE
                        *
FDD9 FE E4 00   IO      LDX    IOV      GET IRQ VECTOR
FDDC 6E 00              JMP    X
                        *
                        * NMI SEQUENCE
                        *
FDDE FE E4 02   PGWDWN  LDX    NIO      GET NMI VECTOR
FDE1 6E 00              JMP    X
                        *
                        * SWI INTERRUPT SEQUENCE
                        *
FDE3 FE E4 36   SFEI    LDX    SWII     GET SWI VECTOR
FDE6 6E 00              JMP    X
                        *
                        * INITIALIZATION/RESET CODE
                        *
FDE8            ADRSTR  EQU    *
FDE8 E4 B9              FDB    STACK-7  INIT FOR "SP"
FDEA FF 05              FDB    SWI1S    INIT FOR "SWI1"
                        *
FDEC 20 03              BRA    BRG      "BRA" INST IS REPLACED BY COND BRA
FDEE 7E FF 5A           JMP    BRNOGO   INST IN ROUTINE WHICH DETERMINES
FDF1 7E FF 91   BRG     JMP    BRGO     IF BRA IS GO/NOGO
FDF3            ADREND  EQU    *-1
                        *
                        * CONSTANT INITIALIZATION
                        *
FDF4            START   EQU    *
FDF4 6E E4 3F           LDS    #STKEND-1 S:POINTER TO RAM
FDF7 CE FD F3           LDX    #ADREND  X:POINTER TO ROM
FDFA A6 00      START1  LDA A  X        GET NEXT CONSTANT BYTE
FDFC 36                 PSH A          INIT NEXT RAM BYTE
FDFD 09                 DEX
FDFE 8C FD E7           CPX    #ADRSTR-1 END OF CONSTANT ROM AREA
FE01 26 F7              BNE    START1   NO CONTINUE
                        *
                        * INITIALIZATION TO 0
                        *
FE03 B5                 FCB    SKIP1:$14 VERSION NUMBER AUGU.17 1981
FE05 C6 30              LDA B  #SIZRCL  SIZE OF RAM TO CLEAR
FE07 4F                 CLR A
FE08 36         START2  PSH A          CLEAR NEXT RAM LOCATION
FE09 5A                 DEC B          ANYMORE BYTES TO INIT ?
FE0A 26 FC              BNE    START2   YES CONTINUE
                        *
                        * INITIALIZE I/O
                        *
FE0C 8E E4 34           LDS    #P
FE0F BD FB 16           JSR    INIT     INITIALIZE I/O
```

```
GPMON V1.4                          9-12-81  ASSEMBLEUR T5C 6800   PAGE   35


1501                    *
1502                    * MAIN COMMAND/CONTROL LOOP
1503                    *
1504  FD12              CONTRL  EQU     *
1505  FD12 BE E4 34             LDS     SP          RESTORE STACK POINTER
1506  FD15 CE FF 06             LDX     #SWI1S      RESTORE SWI VECTOR
1507  FD18 FF E4 35             STX     SWI1
1508  FD1B BD FD D4             JSR     PROMPT      PRINT PROMPT
1509  FD1E 7F E4 08             CLR     OUTSW       MAKE SURE INPUT IS ECHOED
1510  FD21 BD FD 98             JSR     INCH        READ COMMAND CHARACTER
1511  FD24 16                   TAB                 SAVE IT IN B
1512  FD25 BD FD B3             JSR     OUTS        PRINT SPACE AFTER COMMAND
1513                    *
1514                    * B REGISTER HOLDS CHARACTER INPUT BY USER
1515                    * USE JUMP TABLE TO GO TO APPROPRIATE ROUTINE
1516                    *
1517  FD28 CE FC A6             LDX     #FCTABL     X:= ADDRESS OF JUMP TABLE
1518  FD2B E1 00        NXTCHR  CMP B   0,X         DOES INPUT CHAR. MATCH ?
1519  FD2D 27 0F                BEQ     GOODCH      YES,GO TO APPROPRIATE ROUTINE
1520  FD2F 08                   INX                 NO UPDATE INDEX
1521  FD30 08                   INX
1522  FD31 08                   INX
1523  FD32 8C FC D9             CPX     #FCTBEN     END OF TABLE REACHED ?
1524  FD35 26 F4                BNE     NXTCHR      NO,TRY NEXT COMMAND
1525  FD37 86 3F        WHAT    LDA A   #'?         NO MATCH PRINT "?"
1526  FD39 BD FD 82             JSR     OUTCH
1527  FD3C 20 D4                BRA     CONTRL      REPROMPT USER
1528                    *
1529  FD3E EE 01        GOODCH  LDX     1,X         GET ADDRESS FROM J.T.
1530  FD40 6E 00                JMP     0,X         GO TO APPROPRIATE ROUTINE
```

```
1532  FD42                ORG    $FD42
1533                  *
1534                  * BASIC ROUTINES
1535                  *
1536
1537                  *
1538                  * INPUT HEX CHARACTER
1539                  *
1540  FD42 8D 54      INHEX  BSR    INCH
1541  FD44 80 30      INHEX2 SUB A  #$30
1542  FD46 2B CA             BMI    CONTRL    NOT HEX
1543  FD48 81 09             CMP A  #$9
1544  FD4A 2F 0A             BLE    INHG
1545  FD4C 81 11             CMP A  #$11
1546  FD4E 2B C2             BMI    CONTRL    NOT HEX
1547  FD50 81 16             CMP A  #$16
1548  FD52 2E BE             BGT    CONTRL    NOT HEX
1549  FD54 80 07             SUB A  #7
1550  FD56 39         INHG   RTS
1551                  *
1552                  * BUILD ADDRESS
1553                  *
1554  FD57 8D 10      BADDR  BSR    BYTE      READ 2 CHAR.
1555  FD59 B7 E4 0A          STA A  XHI
1556  FD5C 8D 0B             BSR    BYTE      READ 2 CHAR.
1557  FD5E B7 E4 0B          STA A  XLOW
1558  FD61 FE E4 0A          LDX    XHI       X:= ADDRESS WE BUILT
1559  FD64 39                RTS
1560  FD65 8D F0      BADDRS BSR    BADDR     BUILD ADDRESS
1561  FD67 20 4A             BRA    OUTS      PRINT SPACE AND RETURN
1562                  *
1563                  * INPUT BYTE
1564                  *
1565  FD69 8D D7      BYTE   BSR    INHEX     GET HEX CHAR.
1566  FD6B 48         BYTE2  ASL A
1567  FD6C 48                ASL A
1568  FD6D 48                ASL A
1569  FD6E 48                ASL A
1570  FD6F 16                TAB
1571  FD70 8D D0             BSR    INHEX     GET HEX CHAR.
1572  FD72 1B                ABA
1573  FD73 39                RTS
1574                  *
1575                  * OUT HEXA BCD DIGIT
1576                  *
1577  FD74 44         OUTHL  LSR A             OUT HEX LEFT BCD DIGIT
1578  FD75 44                LSR A
1579  FD76 44                LSR A
1580  FD77 44                LSR A
1581  FD78 84 0F      OUTHR  AND A  #$F        OUT HEX RIGHT BCD DIGIT
1582  FD7A 8B 30             ADD A  #$30
1583  FD7C 81 39             CMP A  #$39
1584  FD7E 23 02             BLS    OUTCH
1585  FD80 8B 07             ADD A  #7
1586                  *
```

```
1587                    * OUTPUT ONE CHARACTER
1588                    *
1589  FD82 BD F7 F7  OUTCH   JSR    OUTPUT    OUTPUT CHAR. TO SCREEN
1590  FD85 7D E4 09          TST    PRTFLG    TO PRINTER ?
1591  FD88 26 01             BNE    POUTS     IF YES PRINT CHAR.
1592  FD8A 39                RTS              RETURN
1593  FD8B 7E FC 8F  POUTS   JMP    POUT
1594                    *
1595                    * PRINT DATA POINTED BY X-REG
1596                    *
1597  FD8E BD F2     PDATA2  BSR    OUTCH     OUTPUT DATA
1598  FD90 08        PDATA3  INX
1599  FD91 A6 00     PDATA1  LDA A  X
1600  FD93 81 04             CMP A  £4        STOP IF EOT
1601  FD95 26 F7             BNE    PDATA2
1602  FD97 39                RTS              RETURN
1603                    *
1604                    * INPUT ONE CHARACTER
1605                    *
1606  FD98 BD FB A1  INCH    JSR    TINPUT    CHECK FOR TYPED CHAR
1607  FD9B 27 FB             BEQ    INCH      NOT READY
1608  FD9D BD FB 6E          JSR    INPUT     INPUT CHAR.
1609  FDA0 7D E4 02          TST    OUTSW     SHOULD INPUT BE ECHOED?
1610  FDA3 27 DD             BEQ    OUTCH     IF SO ,OUTPUT CHAR.
1611  FDA5 39                RTS
1612                    *
1613                    * OUTPUT BYTE
1614                    *
1615  FDA6 A6 00     OUT2H   LDA A  0,X       OUTPUT 2 HEX CHAR.
1616  FDA8 08                INX
1617  FDA9 36        OUT2HA  PSH A
1618  FDAA BD C8             BSR    OUTHL     OUT LEFT HEX CHAR.
1619  FDAC 32                PUL A
1620  FDAD 20 C9             BRA    OUTHR     OUTPUT RIGHT HEX CHAR AND RTS
1621                    *
1622                    * OUTPUT ADDRESS
1623                    *
1624  FDAF BD F5     OUT4HS  BSR    OUT2H     OUTPUT 4 HEX CHAR.+SPACE
1625  FDB1 BD F3     OUT2HS  BSR    OUT2H     OUTPUT 2 HEX CHAR.+SPACE
1626  FDB3 86 20     OUTS    LDA A  £$20      OUTPUT SPACE
1627  FDB5 20 CB             BRA    OUTCH     (BSR+RTS)
1628                    *
1629                    * PRINT STACK CONTENTS
1630                    *
1631  FDB7 8D 13     PRINT   BSR    PCRLF     OUTPUT CR/LF
1632  FDB9 FE E4 34          LDX    SP        PRINT OUT STACK
1633  FDBC 08                INX
1634  FDBD BD F2             BSR    OUT2HS    CC-REG
1635  FDBF BD F0             BSR    OUT2HS    ACC-B
1636  FDC1 BD EE             BSR    OUT2HS    ACC-A
1637  FDC3 BD EA             BSR    OUT4HS    X-REG
1638  FDC5 BD E8             BSR    OUT4HS    P-COUNTER
1639  FDC7 CE E4 34          LDX    £SP
1640  FDCA 20 E3             BRA    OUT4HS    PRINT SP AND RETURN
1641                    *
1642                    * OUTPUT CR/LF
```

```
GPMON V1.4                          9-12-81  ASSEMBLEUR TSC 6800   PAGE   38

1643                    *
1644  FDCC 86 0D        PCRLF   LDA A  #CR      OUTPUT CR
1645  FDCE BD B2              BSR    OUTCH
1646  FDD0 86 0A              LDA A  #LF      OUTPUT LF
1647  FDD2 20 AE              BRA    OUTCH    AND RETURN
1648                    *
1649                    * OUTPUT PROMPT
1650                    *
1651  FDD4 8D F6        PROMPT  BSR    PCRLF    DO CR/LF
1652  FDD6 86 2B              LDA A  #PRPT
1653  FDD8 20 A8              BRA    OUTCH    OUTPUT PROMPT AND RETURN
1654                    *
1655                    * ERROR : VALUE IN RAM DO NOT MATCH
1656                    *
1657  FDDA 16           ERROR   TAB            SAVE A
1658  FDDB 8D EF              BSR    PCRLF
1659  FDDD CE E4 0A           LDX    #XHI
1660  FDE0 8D CD              BSR    OUT4HS   PRINT ADDRESS
1661  FDE2 FE E4 0A           LDX    XHI
1662  FDE5 8D CA              BSR    OUT2HS   PRINT CONTENT
1663  FDE7 17                 TBA            RESTORE A
1664  FDE8 8D BF              BSR    OUT2HA   PRINT VALUE WE WANT TO STORE
1665  FDEA 7E FD 37           JMP    WHAT     PRINT "?"
```

```
1667                    *
1668                    * -M- COMMAND
1669                    *   MEMORY EXAMINE AND CHANGE
1670                    *
1671  FDED 8D 73        CHANGE  BSR     BADDRJ    BUILD ADDRESS
1672  FDEF 8D 0B              BSR     PCRLF     DO CR/LF
1673  FDF1 CE E4 0A     CHANG   LDX     #XHI
1674  FDF4 8D B9              BSR     OUT4HS    PRINT ADDRESS
1675  FDF6 FE E4 0A           LDX     XHI
1676  FDF9 8D B6              BSR     OUT2HS    PRINT OLD DATA
1677  FDFB 09                 DEX
1678  FDFC 8D 9A        CHA1    BSR     INCH      INPUT CHAR.
1679  FDFE 81 2E              CMP A   #DOWN     DOWN ?
1680  FE00 27 14              BEQ     BLF       YES
1681  FE02 81 20              CMP A   #UP       CHECK FOR UP
1682  FE04 27 0E              BEQ     UA
1683  FE06 BD FD 44           JSR     INHEX2    CHECK FIRST BYTE
1684  FE09 BD FD 6B           JSR     BYTE2     GET NEW BYTE
1685  FE0C A7 00              STA A   X         WRITE BYTE IN MEMORY
1686  FE0E A1 00              CMP A   X         CHECK IF RAM
1687  FE10 27 EA              BEQ     CHA1      GO INPUT NEXT CHAR.
1688  FE12 20 C6              BRA     ERROR     NOT RAM
1689  FE14 09           UA      DEX             DEC ADDR
1690  FE15 85                 FCB     SKIP1     SKIP 1 BYTE
1691  FE16 08           BLF     INX             INC ADDR
1692  FE17 FF E4 0A     LF1     STX     XHI       SAVE ADDRESS
1693  FE1A 8D B0              BSR     PCRLF     OUTPUT CR/LF
1694  FE1C 20 D3              BRA     CHANG     CONTINUE
1695                  *
1696                  * -U- COMMAND
1697                  *   RESET ONE BREAKPOINT
1698                  *
1699  FE1E 8D 42        RSTBRK  BSR     BADDRJ    PUT ADDRESS IN XHI,XLOW
1700  FE20 86 09              LDA A   #BKFPG1   RESET 1 BREAK FLAG
1701  FE22 8C                 FCB     SKIP2     SKIP 2 BYTES
1702                  *
1703                  * -B- COMMAND
1704                  *   PRINT OUT ALL NON-ZERO BREAK ADDRESSES
1705                  *
1706  FE23 86 21        PNTBRK  LDA A   #BKFPRT   PRINT BREAK ADDR FLAG
1707  FE25 8C                 FCB     SKIP2     SKIP 2 BYTES
1708                  *
1709                  * -W- COMMAND
1710                  *   RESET ALL BREAKPOINTS
1711                  *
1712  FE26 86 11        DELBRK  LDA A   #BKFPGA   RESET BREAKS FLAG
1713  FE28 20 08              BRA     SETB2
1714                  *
1715                  * -V- COMMAND
1716                  *   SET ONE BREAKPOINT
1717                  *
1718  FE2A 8D 36        SETBRK  BSR     BADDRJ    PUT ADDRESS INTO XHI,XLOW
1719  FE2C 86 09              LDA A   #BKFPG1
1720  FE2E 8D 63              BSR     BRKSUB
1721  FE30 86 05              LDA A   #BKFRCD   SET ONE BREAKPOINT FLAG
```

```
1722  FE32 8D 5F     SETB2  BSR    BRKSUB   BREAK HANDLING SUBROUTINE
1723  FE34 7E FD 12  CTRLJ  JMP    CONTRL   RETURN TO COMMAND LEVEL
1724                 *
1725                 * -G- COMMAND
1726                 *   GO TO ENTERED ADDRESS
1727                 *
1728  FE37 8D 29     GOTO   BSR    BADDRJ   GET ADDR. FROM USER IN XHI,XLOW
1729  FE39 FE E4 34         LDX    SP       GET VALUE SAVED WHEN ENTERING GPMON
1730  FE3C A7 07            STA A  7,X      A-REG CONTAINS XLOW
1731  FE3E B6 E4 0A         LDA A  XHI
1732  FE41 A7 06            STA A  6,X      PLACE ADDRESS ON STACK
1733                 *
1734                 * -C- COMMAND
1735                 *   CONTINUE EXECUTION
1736                 *
1737  FE43 7C E4 10  CONT   INC    BRKTRC   TRACE 1 TO RESTORE SWI'S
1738  FE46 CE 00 01         LDX    #1
1739  FE49 20 09            BRA    TRACE3
1740                 *
1741                 * -N- COMMAND
1742                 *   SINGLE INSTRUCTION TRACE REQUESTED
1743                 *
1744  FE4B CE 00 01  NEXT   LDX    #1       #INSTRUCTION TO TRACE
1745  FE4E 8C               FCB    SKIP2    SKIP 2 BYTES
1746                 *
1747                 * -T- COMMAND
1748                 *   MULTIPLE INSTRUCTION TRACE
1749                 *
1750  FE4F 8D 11     TRACE  BSR    BADDRJ   GET N. OF INSTRUCTION TO TRACE
1751  FE51 7F E4 10  TRACE1 CLR    BRKTRC   CLEAR FLAG INDICATING TRACE IS DUE TO BREAK
1752  FE54 FF E4 06  TRACE3 STX    NTRACE   SAVE # INST'S TO TRACE
1753  FE57 FE E4 34         LDX    SP       X:=STACK POINTER
1754  FE5A EE 06            LDX    6,X      X:=ADDR OF INSTR TO BE EXECUTED
1755  FE5C BD FF 9A         JSR    SAVTRC   SAVE ADDR/OPCODE FOR TRACE
1756  FE5F 7E FF 47         JMP    CONTRC   GOTO CONTINUE TRACE
1757  FE62 7E FD 57  BADDRJ JMP    BADDR
1758                 *
1759                 * -D- COMMAND
1760                 *   MEMORY DUMP
1761                 *
1762  FE65 BD FD 65  DUMP   JSR    BADDR8   GET START ADDRESS
1763  FE68 FF E4 0C         STX    XSAVE    SAVE IT
1764  FE6B 8D F5            BSR    BADDRJ   GET END ADDRESS IN XHI,XLOW
1765  FE6D BD FD CC  DUMP1  JSR    PCRLF    OUTPUT CR,LF
1766  FE70 C6 10            LDA B  #16      16 BYTES PER LINE
1767  FE72 CE E4 0C         LDX    #XSAVE   PRINT ADDRESS
1768  FE75 BD FD AF         JSR    OUT4HS
1769  FE78 FE E4 0C  DUMP2  LDX    XSAVE    PRINT BYTE
1770  FE7B BD FD B1         JSR    OUT2HS
1771  FE7E FF E4 0C         STX    XSAVE    SAVE NEW ADDRESS
1772  FE81 09               DEX
1773  FE82 BC E4 0A         CPX    XHI      FINISHED ?
1774  FE85 27 AD            BEQ    CTRLJ    YES,EXIT
1775  FE87 5A               DEC B          DEC THE COUNT
1776  FE88 26 EE            BNE    DUMP2    SAME LINE IF NOT 0
1777  FE8A 20 E1            BRA    DUMP1    NEW LINE
```



```
1778                 *
1779                 * -O- COMMAND
1780                 *   SWITCH OUTPUT TO PRINTER
1781                 *
1782  FE8C 73 E4 09  SWITCH COM    PRTFLG   SWITCH FLAG
1783  FE8F 20 A3            BRA    CTRLJ    RETURN TO COMMAND LEVEL
```

```
                        *
                        * SUBROUTINE TO INSERT BREAK IN USER CODE
                        *
FE91 86 82              INSBRK  LDA A  #BKFINS
                        *
                        * BRKSUB
                        *
                        * THIS ROUTINE HANDLES BREAKPOINT OPERATIONS
                        * THE CONTENT OF A-REG DETERMINES THE FUNCTION PERFORMED:
                        *
                        *   A=BKFINS : PUT BREAKS INTO USER'S CODE
                        *   A=BKFPG1 : PURGE BREAKPOINT WHOSE ADDRESS IS IN
                        *              XHI,XLOW.ALL BREAKPOINTS ARE TEMPORARY REMOVED
                        *   A=BKFPGA : PURGE ALL BREAKPOINTS
                        *   A=BKFPRT : PRINT ALL BREAKS,TEMP. REMOVED
                        *   A=BKFRMV : REMOVE TEMPORARY ALL BREAKPOINTS
                        *   A=BKFRCD : PUT BREAK ADDRESS IN XHI,XLOW INTO THE
                        *              FIRST ZERO BREAKPOINT POSITION;TEMP. REMOVED
                        *
0001                    BKFRMV  EQU    %00000001
0002                    BKFINS  EQU    %10000010
0005                    BKFRCD  EQU    %00000101
0009                    BKFPG1  EQU    %00001001
0011                    BKFPGA  EQU    %00010001
0021                    BKFPRT  EQU    %00100001
                        *
FE93                    BRKSUB  EQU    *
FE93 36                         PSH A          SAVE A-REG
FE94 CE E4 1C                   LDX    #BRKADR  X POINTS BREAK TABLE
FE97 FF E4 0C           BRKP0   STX    XSAVE    SAVE X
FE9A 33                         PUL B          GET FUNCTION NUMBER
FE9B 37                         PSH B          UPDATE STACK POINTER
FE9C A6 02                      LDA A  2,X     LOAD USER CODE SAVED IN BREAK TABLE
FE9E EE 00                      LDX    X       LOAD THE BREAKPOINTED USER ADDRESS
FEA0 27 03                      BEQ    BRKP01  IF ADDR=0 NOT A BREAKPOINT
FEA2 C4 FB                      AND B  #$FB    RECORD NOT ALLOWED ,MASK FLAG
FEA4 8C                         FCB    SKIP2   SKIP 2 BYTES
FEA5 C4 C4              BRKP01  AND B  #$C4    RECORD ONLY ALLOWED
FEA7 7D E4 0F           BRKP03  TST    BRKSIN   ARE BREAKS IN USER'CODE ?
FEAA 27 03                      BEQ    BRKP02  NO
FEAC C4 FD                      AND B  #$FD    YES,DO NOT WRITE SWI IN USER CODE
FEAE 8C                         FCB    SKIP2   SKIP 2 BYTES
FEAF C4 FE              BRKP02  AND B  #$FE    AVOID THAT BREAKS BE TEMPORARY REMOVED
                        *
                        * REWRITE CODE IN USER PROGRAM FROM BREAKPOINT TABLE
                        *
FEB1 54                 BRKP04  LSR B          PLACE FUNCTION FLAG IN CARRY
FEB2 24 0C                      BCC    BRKP2   FUNCTION NOT REQUESTED
FEB4 A7 00                      STA A  X       REWRITE CODE IN USER PGM
                        *
                        * WRITE SWI'S IN USER CODE
                        *
FEB6 54                 BRKP2   LSR B
FEB7 24 0D                      BCC    BRKP3
FEB9 A6 00                      LDA A  X       SAVE USER CODE
```

```
GPMON V1.4                          9-12-81  ASSEMBLEUR TSC 6800   PAGE   43

 1840  FEBB 35              PSH A
 1841  FEBC 86 3F           LDA A  #SWI
 1842  FEBE A7 00           STA A  X         WRITE SWI
 1843  FEC0 32              PUL A
 1844  FEC1 FE E4 0C        LDX    XSAVE     ADDR OF BREAK IN BREAK TABLE
 1845  FEC4 A7 02           STA A  2,X       SAVE USER CODE IN BREAK TABLE
 1846                 *
 1847                 * RECORD BREAKPOINT IN TABLE
 1848                 *
 1849  FEC6 FE E4 0C  BRKP3  LDX    XSAVE     GET ADDRESS OF BREAK IN TABLE
 1850  FEC9 54              LSR B            NEXT FUNCTION
 1851  FECA 24 0E           BCC    BRKP4
 1852  FECC B6 E4 0A        LDA A  XHI       GET HI-BYTE OF NEW ADDRESS
 1853  FECF A7 00           STA A  0,X       WRITE IN TABLE
 1854  FED1 B6 E4 0B        LDA A  XLOW      GET LOW BYTE
 1855  FED4 A7 01           STA A  1,X
 1856  FED6 32              PUL A            GET FUNCTION NUMBER FROM STACK
 1857  FED7 84 FB           AND A  #%11111011 DO NOT PLACE ADDRESS MORE THAN ONCE
 1858  FED9 36              PSH A            CONTINUE TO TAKE OUT BREAKPOINTS
 1859                 *
 1860                 * PURGE BREAKPOINT WHOSE ADDRESS IS IN XHI,XLOW
 1861                 *
 1862  FEDA 54        BRKP4  LSR B
 1863  FEDB 24 0E           BCC    BRKP5
 1864  FEDD A6 00           LDA A  X         GET HI-BYTE FROM TABLE
 1865  FEDF B1 E4 0A        CMP A  XHI       COMPARE
 1866  FEE2 26 07           BNE    BRKP5     NOT THIS BREAK TO BE PURGED
 1867  FEE4 A6 01           LDA A  1,X       GET LOW BYTE
 1868  FEE6 B1 E4 0B        CMP A  XLOW
 1869  FEE9 27 03           BEQ    BRKP52    THIS BREAK IS TO BE PURGED
 1870                 *
 1871                 * PURGE ALL BREAKPOINTS
 1872                 *
 1873  FEEB 54        BRKP5  LSR B
 1874  FEEC 24 04           BCC    BRKP6
 1875  FEEE 6F 00     BRKP52 CLR    0,X       CLEAR BREAKPOINT HIGH
 1876  FEF0 6F 01           CLR    1,X       CLEAR LOW BYTE
 1877                 *
 1878                 * PRINT BREAKPOINT
 1879                 *
 1880  FEF2 54        BRKP6  LSR B
 1881  FEF3 24 04           BCC    BRKPE
 1882  FEF5 BD FD BF        JSR    OUT4HS    PRINT BREAKPOINT ADDRESS
 1883  FEF8 8C              FCB    $8C       SKIP 2 BYTES
 1884                 *
 1885                 * UPDATE LOOP INDEX AND LOOP IF APPROPRIATE
 1886                 *
 1887  FEF9 08        BRKPE  INX
 1888  FEFA 08              INX
 1889  FEFB 08        BRKPE1 INX              X POINTS TO NEXT ADDR IN TABLE
 1890  FEFC 8C E4 04        CPX    #BRKINS
 1891  FEFF 26 95           BNE    BRKP0
 1892                 *
 1893                 * WRAP-UP PROCESSING AND EXIT
 1894                 *
 1895  FF01 F7 E4 0F          STA B  BRKSIN    STORE APPROPRIATE FLAG

GPMON V1.4                          9-12-81  ASSEMBLEUR TSC 6800   PAGE   44

 1896  FF04 32              PUL A            UPDATE STACK POINTER
 1897  FF05 39              RTS              RETURN
```

```
1899                    *
1900                    * SWI SOFTWARE INTERRUPT PROCESSING
1901                    *
1902  FF06              SWI1S   EQU     *
1903  FF06 BF E4 34             STS     SP          SAVE USER'S SP
1904  FF09 86 01                LDA A   #BKFRMV
1905  FF0B BD FE 93             JSR     BRKSUB      GO TAKE OUT ALL BREAKS
1906                    * DECREMENT P-COUNTER
1907  FF0E 30                   TSX                 X:=STACK POINTER-1
1908  FF0F 60 06                TST     6,X         IF LOWER BYTE=0 : BORROW
1909  FF11 26 02                BNE     SWI1S1      BRANCH IF BORROW NOT REQUIRED
1910  FF13 6A 05                DEC     5,X         DECREMENT UPPER BYTE
1911  FF15 6A 06        SWI1S1  DEC     6,X         DECREMENT LOWER BYTE
1912                    * TEST FOR ADDRESS TRACE OR BREAK
1913  FF17 EE 05                LDX     5,X         X:=P-COUNTER
1914  FF19 BC E4 04             CPX     TRCADR      IS SWI FOR TRACE ?
1915  FF1C 27 06                BEQ     TRCINH      YES,GO TO TRACE INT HANDLER
1916                    *
1917                    * BREAK INTERRUPT HANDLER
1918                    *
1919                    *
1920                    * -R- COMMAND
1921                    *
1922  FF1E              PSTAK   EQU     *
1923  FF1E BD FD B7             JSR     PRINT       STOP AND SHOW REGISTER TO USER
1924  FF21 7E FD 12     CNTRL3  JMP     CONTRL      RETURN
1925                    *
1926                    * TRACE INTERRUPT HANDLER
1927                    * P-COUNTER HAS BEEN DECREMENTED TO POINT AT SWI
1928                    * TRCINS HOLDS OP CODE REPLACED BY SWI
1929                    * X HOLDS ADDRESS OF WHERE TRACE SWI IS
1930                    *
1931  FF24 B6 E4 0E     TRCINH  LDA A   TRCINS       GET OPCODE OF TRACED INSTR.
1932  FF27 A7 00                STA A   0,X         RESTORE TO USER'CODE
1933  FF29 7D E4 10             TST     BRKTRC      IS PROCESSING TO BE IMMEDIATELY CONTINUED ?
1934  FF2C 27 0D                BEQ     NBKTRC      BRANCH IF NOT
1935                    *
1936                    * PROCESSING IS TO "CONTINUE"
1937                    *
1938  FF2E 7F E4 10             CLR     BRKTRC      RESET CONTINUE FLAG
1939  FF31 BD FE 91             JSR     INSBRK      INSERT BREAKS IN USER CODE
1940  FF34 7F E4 04             CLR     TRCADR      NO MORE TRACE,SO CLEAR ADDRESS
1941  FF37 7F E4 05             CLR     TRCADR+1
1942  FF3A 3B                   RTI                 CONTINUE
1943                    *
1944                    * TRACE IS DUE TO N OR T TRACE COMMAND
1945                    *
1946  FF3B BD FD B7     NBKTRC  JSR     PRINT       PRINT STACK
1947  FF3E FE E4 06             LDX     NTRACE      GET # OF INSTRUCTIONS TO TRACE
1948  FF41 09                   DEX                 DECREMENT COUNT
1949  FF42 FF E4 06             STX     NTRACE      AND RESTORE
1950  FF45 27 DA                BEQ     CNTRL3      BRANCH IF ALL TRACES DONE
1951                    *
1952                    * TRACE NOT DONE - TRACE NEXT INSTRUCTION
1953                    *
```

```
GPMON V1.4                          9-12-81  ASSEMBLEUR TSC 6800   PAGE   46

 1954  FF47 B6 E4 0E  CONTRC  LDA A  TRCINS    GET CURRENT INSTRUCTION
 1955  FF4A B7 E4 38          STA A  BRINS     SAVE IN CASE IT'S A BRANCH
 1956  FF4D 8D 6F             BSR    OPCBYT    GO GET £ OF BYTES/TYPE
 1957  FF4F 4D                TST A            CHECK FOR BRANCH
 1958  FF50 2A 0A             BPL    CKOBRA    CHECK FOR OTHER THAN BRANCH
 1959                  *
 1960                  * RELATIVE BRANCH TYPE INSTRUCTION
 1961                  * DETERMINE WHERE TO PUT SWI
 1962                  * S HOLDS POINTER TO USER STACK AFTER SWI
 1963                  *
 1964  FF52 32                PUL A            GET CONDITION CODE
 1965  FF53 34                DES              UPDATE STACK PTR AFTER PULL
 1966  FF54 9A 10             ORA A  %00010000 MAKE INT'S INHIBITED
 1967  FF56 06                TAP              RESTORE USER'S C CODE REG
 1968  FF57 7E E4 38          JMP    BRINS     GO SEE HOW RELATIVE BRANCH FARES
 1969                  *
 1970                  * BRANCH WAS NOGO - PUT SWI AT NEXT INSTRUCTION
 1971                  *
 1972  FF5A 86 02     BRNOGO  LDA A  £2        A:=£ OF BYTES AFTER CURRENT INSTR.
 1973                  *
 1974                  * INSTRUCTION TO BE TRACED IS NOT A BRANCH
 1975                  *
 1976  FF5C FE E4 04  CKOBRA  LDX    TRCADR    X:=TRACE ADDRESS
 1977  FF5F E6 00             LDA B  0,X       GET INSTR. TO BE TRACED
 1978  FF61 C1 6E             CMP B  £$6E      IS IT A JUMP,INDEXED
 1979  FF63 27 44             BEQ    JMPIDX    YES GO SIMULATE JUMP IDXED
 1980  FF65 C1 7E             CMP B  £$7E      JUMP EXTENDED ?
 1981  FF67 27 51             BEQ    JMPEXT
 1982  FF69 C1 AD             CMP B  £$AD      JSR,INDEXED ?
 1983  FF6B 27 3C             BEQ    JMPIDX    JUMP IDXED IS SAME AS TRANSFERT OF CONTROL
 1984  FF6D C1 BD             CMP B  £$BD      JSR,EXTENDED ?
 1985  FF6F 27 49             BEQ    JMPEXT
 1986  FF71 C1 3B             CMP B  £$3B      RTI ?
 1987  FF73 27 3B             BEQ    RTISIM
 1988  FF75 C1 39             CMP B  £$39      RTS ?
 1989  FF77 27 3C             BEQ    RTSSIM
 1990  FF79 C1 3F             CMP B  £$3F      SWI ? IF SWI USER DON'T INCREMENT ADDR.
 1991  FF7B 27 13             BEQ    BRG2
 1992  FF7D C1 8D             CMP B  £$8D      BSR ? IF YES BRANCH PROCESSING
 1993                  *
 1994                  * NOT A BRANCH,JUMP,RTI,RTS
 1995                  * A REGISTER HOLDS £ OF BYTES IN INSTRUCTION
 1996                  *
 1997  FF7F 26 0D             BNE    BRG1      PUT IN NEW SWI AND TRACE NEXT INSTR.
 1998                  *
 1999                  * BRANCH WAS GO,PUT SWI AT ADDRESS BEING JUMP TO
 2000                  *
 2001  FF81 FE E4 04  BRGO    LDX    TRCADR    X:=TRACE ADDRESS
 2002  FF84 A6 01             LDA A  1,X       GET BRANCH OFFSET
 2003  FF86 08                INX              OFFSET IS RELATIVE TO
 2004  FF87 08                INX              INSTR FOLLOWING BRANCH
 2005  FF88 2A 04             BPL    BRG1      BRANCH IS OFFSET POSITIVE
 2006                  * X NEEDS TO BE DECREMENTED (OFFSET NEGATIVE)
 2007  FF8A 09        BRGODC  DEX              DECREMENT ADDRESS
 2008  FF8B 4C                INC A            INCREMENT COUNTER
 2009  FF8C 26 FC             BNE    BRGODC
```

```
GPMON V1.4                          9-12-81  ASSEMBLEUR TSC 6800   PAGE   47

2010                    * CONTENT OF A IS 0
2011  FF8E 8D 15        BRG1    BSR     INCX        INCREMENT X BY AMOUNT IN A-REG
2012  FF90 8D 08        BRG2    BSR     SAVTRC      SAVE ADDR/OPC OF NEXT INSTR TO STOP ON
2013  FF92 86 3F              LDA A   #SWI        REPLACE INSTR. WITH SWI
2014  FF94 A7 00              STA A   X
2015  FF96 8E E4 34     RTIPC   LDS     SP          GET ORIGINAL STACK POINTER
2016  FF99 3B                 RTI                 TRACE ANOTHER INSTR.
2017                    *
2018                    * SUBROUTINE TO SAVE ADDR IN X INTO TRCADR AND OPC INTO TRCINS
2019                    *
2020  FF9A              SAVTRC  EQU     *
2021  FF9A FF E4 04           STX     TRCADR
2022  FF9D A6 00              LDA A   X
2023  FF9F B7 E4 0E           STA A   TRCINS
2024  FFA2 39                 RTS
2025                    *
2026                    * SUBROUTINE TO INCREMENT X BY CONTENT OF A
2027                    *
2028  FFA3 08           INXLP   INX                 INCREMENT X
2029  FFA4 4A                 DEC A               DECREMENT COUNT
2030  FFA5 4D           INCX    TST A
2031  FFA6 26 FB              BNE     INXLP       IF COUNT NOT YET 0 LOOP
2032  FFA8 39                 RTS                 RETURN
2033                    *
2034                    * JUMP,JSR INDEXED SIMULATION
2035                    *
2036  FFA9 A6 01        JMPIDX  LDA A   1,X         A:=ADDR OFFSET
2037  FFAB 30                 TSX
2038  FFAC EE 03              LDX     3,X         GET TARGET'S X REG
2039  FFAE 20 DE              BRA     BRG1        UPDATEX TRACE NEXT INSTR
2040                    *
2041                    * RTI ENCOUNTERED
2042                    *
2043  FFB0 30           RTISIM  TSX
2044  FFB1 EE 0C              LDX     12,X        GET P-COUNTER FROM STACK
2045  FFB3 20 DB              BRA     BRG2        GO TRACE NEXT INSTR.
2046                    *
2047                    * RTS ENCOUNTERED
2048                    *
2049  FFB5 30           RTSSIM  TSX
2050  FFB6 EE 07              LDX     7,X         GET RETURN P-REG FROM STACK
2051  FFB8 20 D6              BRA     BRG2
2052                    *
2053                    * JUMP,JSR EXTENDED
2054                    *
2055  FFBA EE 01        JMPEXT  LDX     1,X
2056  FFBC 20 D2              BRA     BRG2
```

```
GPMON V1.4                        9-12-81  ASSEMBLEUR TSC 6800   PAGE   48

2058                    *
2059                    * OPBCYT
2060                    *
2061                    * THIS ROUTINE DETERMINES THE # OF BYTES IN AN INSTRUCTION
2062                    * GIVEN ITS OP CODE
2063                    *
2064                    * INPUT : A HOLDS THE OP CODE
2065                    *
2066                    * OUTPUT : X HOLDS INDEX OF TABLE ELEMENT
2067                    * B NOT RESTORED
2068                    * A HOLDS # OF BYTES IN INSTRUCTION
2069                    * EXCEPT FOR BRANCHES IN WHICH CASE A IS NEGATIVE
2070                    *
2071  FFBE              OPCBYT  EQU   *
2072  FFBE 16                   TAB             B:=OP CODE
2073  FFBF 44                   LSR A
2074  FFC0 44                   LSR A
2075  FFC1 44                   LSR A           PUT 4 UPPER BITS OF OP CODE INTO
2076  FFC2 44                   LSR A           LOWER 4 BITS OF A
2077                    *
2078  FFC3 CE FF D6             LDX   #OPBTTB   X:=ADDR OF TABLE
2079  FFC6 8D DD                BSR   INCX      INC X TO POINT TO CORRECT ENTRY
2080                    *
2081  FFC8 A6 00                LDA A 0,X       GET TABLE ENTRY
2082  FFCA 26 09                BNE   OPBTRT    IF NOT 0 THEN NO FURTHER PROCESSING NEEDED
2083                    *
2084                    * IF TOP 4 BITS=8 OR C ,THEN THERE ARE TWO CLASSE
2085                    * OF INSTRUCTIONS : 2 BYTE INSTRUCTIONS AND
2086                    * CE,8C AND 8E WHICH ARE 3 BYTE INSTRUCTIONS
2087                    *
2088  FFCC 86 02                LDA A #2        # OF BYTES IN MOST OF 8# INSTRUCTIONS
2089  FFCE C4 0D                AND B #%00001101
2090  FFD0 C1 0C                CMP B #%00001100
2091  FFD2 26 01                BNE   OPBTRT    NO ,RETURN
2092  FFD4 4C                   INC A           # OF BYTES IN INSTRUCTION :=3
2093  FFD5 39           OPBTRT  RTS             RETURN TO CALLER
2094                    *
2095                    * OP CODE TO NUMBER OF BYTES CONVERSION TABLE
2096                    *
2097                    *     # BYTES TOP 4 BITS OF OPCODE
2098                    *
2099  FFD6              OPBTTB  EQU   *
2100  FFD6 01                   FCB   1         0
2101  FFD7 01                   FCB   1         1
2102  FFD8 82                   FCB   2+%10000000 2  MINUS = BRANCHES
2103  FFD9 01                   FCB   1         3
2104  FFDA 01                   FCB   1         4
2105  FFDB 01                   FCB   1         5
2106  FFDC 02                   FCB   2         6
2107  FFDD 03                   FCB   3         7
2108  FFDE 00                   FCB   0         8 # BYTES=2 EXCEPT 8C,8E
2109  FFDF 02                   FCB   2         9
2110  FFE0 02                   FCB   2         A
2111  FFE1 03                   FCB   3         B
2112  FFE2 00                   FCB   0         C # BYTES=2 EXCEPT CE
```

```
GPMON V1.4                          9-12-81  ASSEMBLEUR TSC 6800   PAGE   49

 2113  FFE3 02              FCB   2         D
 2114  FFE4 02              FCB   2         E
 2115  FFE5 03              FCB   3         F
```

```
GPMON V1.4                          9-12-81  ASSEMBLEUR TSC 6800   PAGE   50

 2117                  *
 2118                  * JUMP TABLE TO BASIC ROUTINES OF GPMON
 2119                  *
 2120  FFE6                    ORG   $FFE6
 2121  FFE6 7E FD 12           JMP   CONTRL    WARM START ENTRY POINT
 2122  FFE9 7E FB A1           JMP   CHKCHR    CHECK FOR TYPED CHARACTER
 2123  FFEC 7E FD B2           JMP   OUTCH     OUTPUT ONE CHARACTER
 2124  FFEF 7E FD 9B           JMP   INCH      INPUT ONE CHARACTER
 2125  FFF2 7E FD 91           JMP   PDATA1    PRINT STRING POINTED BY X-REG
 2126  FFF5 7E FD CC           JMP   PCRLF     PRINT CR/LF
 2127                  *
 2128                  * INTERRUPTS VECTORS
 2129                  *
 2130  FFF8                    ORG   $FFF8
 2131  FFF8 FC D9              FDB   IG        REGULAR INTERRUPT
 2132  FFFA FC E3              FDB   SFEI      SOFTWARE INTERRUPT
 2133  FFFC FC DE              FDB   POWDWN    NON MASKABLE INTERRUPT
 2134  FFFE FC F4              FDB   START     RESET INTERRUPT
```

```
GPMON V1.4                    9-12-81  ASSEMBLEUR TSC 6800   PAGE   51

2136                *
2137                * RAM - LOCATIONS DEVOTED TO VARIABLE INFORMATION
2138                *
2139  E400                  ORG    RAM        START OF RAM
2140  0008          NBRBPT  EQU    8          # OF BREAKPOINTS SUPPORTED
2141                *
2142  E400          IOV     RMB    2          I/O INTERRUPT POINTER
2143  E402          NIO     RMB    2          NMI INTERRUPT POINTER
2144                *
2145                * THE FOLLOWING ARE INITIALIZED TO ZERO AT START
2146                *
2147  E404          TRCADR  RMB    2          TRACE ADDRESS
2148  E406          NTRACE  RMB    2          NO. OF INSTRUCTIONS TO TRACE
2149  E408          OUTSW   RMB    1          OUTPUT SWITCH (0=ECHO INPUT)
2150  E409          PRTFLG  RMB    1          PRINT FLAG
2151  E40A          XHI     RMB    1          X REG HIGH (TEMP)
2152  E40B          XLOW    RMB    1          X REG LOW
2153                *
2154  E40C          XSAVE   EQU    *
2155  E40C                  RMB    2
2156                *
2157  E40E          TRCINS  RMB    1          OPCODE REPLACED BY SWI IN TRACE MODE
2158  E40F          BRKSIN  RMB    1          1=BREAKS ARE IN USER PGM
2159  E410          BRKTRC  RMB    1          1=P-COUNTER IS AT BREAKPOINT AND
2160                *                    USER WANTS TO CONTINUE - ONE TRACE WILL
2161                *                    BE DONE AND BREAKPOINTS RESTORED
2162  E411                  RMB    3          RESERVED FOR IO HANDLER
2163  E414                  RMB    8          FREE SPACE
2164                *
2165  E41C          BRKADR  RMB    NBRBPT*3   BREAKPOINT ADDRESS TABLE
2166  E434          BRKINS  EQU    *
2167                *
2168                * THE FOLLOWING ARE INITIALIZED AT START
2169                *
2170  E434          SP      RMB    2          USER STACK POINTER
2171  E436          SWI1    RMB    2          SWI VECTOR
2172  E438          BRINS   RMB    8          STORAGE FOR CONDITIONNAL BRANCH ROUTINE
2173  E440          BRANEN  EQU    *          END OF BRANCH ROUTINE+1
2174                *
2175                * STACK
2176                *
2177  E440          STKEND  RMB    128        RAM USED FOR STACK
2178  E49F          STACK   EQU    *-1        START OF STACK AREA
2179                *
2180                *
2181                *
2182                        END    START
2183                        END

AUCUNE ERREUR DETECTEE
```

GPMON V1.4 9-12-81 ASSEMBLEUR TSC 6800 PAGE 52

TABLE SYMBOLES

ACIAC	E80C	ACIAD	E80D	ACR	002B	ADREND	FCF3	ADRSTR	FCE8
ADTAB1	F9D6	ADTAB2	FAD6	ADVISU	E7F8	BADDR	FD57	BADDRJ	FE52
BADDRS	FD65	BASIC	8000	BELDE1	F6D1	BELDEL	F6CF	BELL	F6BE
BELL1	F6C0	BKFINS	0082	BKFPG1	0009	BKFPGR	0011	BKFPRT	0021
BKFRCD	0005	BKFRMV	0001	BLF	FE16	BRANEN	E440	BRG	FCF1
BRG1	FF8E	BRG2	FF90	BRG8	FF21	BRGODC	FF8A	BRINS	E435
BRKADR	E41C	BRKINS	E434	BRKP0	FE97	BRKP01	FEA5	BRKP02	FEAF
BRKP03	FEA7	BRKP04	FEB1	BRKP2	FEB6	BRKP3	FEC5	BRKP4	FED9
BRKP5	FEEB	BRKP52	FEEE	BRKP6	FEF2	BRKPE	FEF9	BRKPE1	FEFB
BRKSIN	E40F	BRKSUB	FE93	BRKTRC	E410	BRNOGO	FF5A	BYTE	FD69
BYTE2	FD6B	CDISP	F86C	CHA1	FDFC	CHANG	FDF1	CHANGE	FDED
CHKACI	FC4D	CHKCHR	FBA1	CKOBRA	FF5C	CLEAR	F6D5	CLEAR1	F684
CLEAR2	F67C	CMDE1	E821	CMDE2	E831	CNTRL3	FF21	COLCNT	E57E
COLUMN	E572	COM	FC53	COM1	FC70	COM2	FC73	COMRS1	E8F0
CONT	FE43	CONTRC	FF47	CONTRL	FD12	CPOS	F983	CPOS1	F93F
CPOS3	F897	CPOS4	F8A5	CPOS5	F848	CPOSE	F829	CPOSFG	E417
CPOUT	FC90	CR	0020	CRTC	E870	CRTI1	F680	CRTINT	F671
CRTRET	F82B	CRTRT1	F881	CRTTAB	F8AE	CTRLJ	FE34	CURAD	E57C
CUROFF	F8EA	CURON	F875	D15	F634	DATA1	E820	DATA2	E830
DATR91	E8F3	DATRED	E823	DBRA	2003	DDR9	0002	DECCL	F6F2
DECCL1	F6ED	DECLN	F720	DECLN1	F732	DEL1	F63C	DELAY	F63A
DELBRK	FE26	DELLI1	F7A7	DELLIN	F7A5	DIRECT	E4D3	DKBOOT	F5CD
DMAADD	EB00	DMABOT	F508	DMACOM	EB10	DMACON	EB02	DMAPRI	EE14
DOWN	002E	DPAGE	E575	DRVRB1	E8E0	DRVRED	E824	DUMP	FE55
DUMP1	FE6D	DUMP2	FE78	ECHOFF	F6BE	ECHON	F6C4	ENFDT	F3AC
ENTBL	F9DE	EPAGE	E577	EREOL	F8E0	EREOS	F8EC	EREOS1	F8F5
EREOS2	F900	ERLIN	F8C8	ERROR	FDDA	ESCAPE	F848	ESCFG	E416
ESCS	F851	ESCSQ	F85C	ESCTBL	F9AC	FCTABL	FCAB	FCTBEN	FCD9
FCTCRT	F991	FLGIN	E413	FLGRPT	E412	FLGSHF	E411	FUNCT	F831
FUNCT1	F842	FUNCT2	F834	GAIND	F5B2	GCORT	F421	GDDIV	F67F
GGPLT1	F4EB	GGPLT2	F51D	GGPLT3	F552	GKWAIT	F43A	GOINC	F560
GOMOD	F475	GOMO1	F478	GOMO2	F485	GOMOE	F495	GOMPC	F4D6
GOMPT	E402	GOOLOR	F4E4	GOONTR	E7FC	GORTC	E7FE	GORTTB	F454
GOUR	E4C7	GOURX	E7FE	GOURY	E7F9	GOWLTB	E4CA	GOSUB	F58B
GOX	E4CE	GOY	E4CF	GPGDIV	F58F	GINFX	F485	GINFY	F4D2
GINIT	F41A	GMOVE	F496	GMVY	F49C	GOLOAD	F622	GOODOH	FD3E
GOTO	FE37	GPAGX	F504	GPAGY	F539	GPLOT	F46F	GPRET	F503
GRAPH	F449	GRGINC	F5AE	GREPT	FC13	GSO	E4C0	GSHFT1	FC0A
GSHIFT	FC02	GSUPX	F4AF	GSUPY	F4CB	GSX	E4D0	GSY	E4D1
GTEXT	F40F	GTRACE	F4A3	HADRF	00EC	HCPY	F901	HCPY1	F90C
HCPY2	F911	HCPY3	F92D	HOME	F6D7	IER	000E	IFR	000D
INIHG	F956	INACIA	FC43	INCCL	F6DF	INCH	FD98	INCLN	F700
INCLN1	F716	INCX	FFA5	INHEX	FD42	INHEX2	FD44	INICIA	FC25
INIT	F816	INIT82	FB2B	INITCV	FB1E	INITVC	FB48	INITVS	F671
INPUT	F63E	INSBRK	FE91	INVERS	F834	INXLP	FFA7	IO	FC09
IOV	E400	JMPEXT	FFB0	JMPIDX	FFA5	K7MOD	EC00	KEYB	FBBE
KEYB1	FBD3	KEYB2	FBE0	KEYBJ	FBEE	KEYB4	FC21	LF	000A
LF1	FE17	LINCNT	E57D	LINE	E571	LINRET	F6DA	LOADER	4100
LOOP1	F64F	LOOP2	F65F	MFBOOT	F63F	MOVE1	F74C	MOVE1	F74C
MOVEA	F737	NBRKTRC	FF3B	NBRBPT	000E	NEWLIN	F6FC	NEXT	FE4E
NIO	E402	NTRACE	E40E	NXTCHR	FD2E	ONESEC	F831	OPBTRT	FFDE
OPBTTB	FFD8	OPCSVT	FFBE	ORA	2001	ORB	0000	OUT2H	FDA5
OUT2HA	FDA9	OUT2HS	FD81	OUT4HB	FDAF	OUTCH	FD37	OUTCH	FC81

SPMON V1.4 5-12-81 ASSEMBLEUR TSC S800 PAGE 53

OUTOUR	F7S3	OUTHL	FD74	OUTHR	FD78	OUTPT1	FB01	OUTPUT	F7F7
OUTS	FD33	OUTSW	E40B	PADFS	E41B	PAGMOD	F873	PCR	0020
PCRLF	FDCC	PDATA1	FD31	PDATA2	FD3E	PDATA3	FD30	PNTBRK	FE23
POUT	FD3F	POUTS	FD3B	POWDWN	FCDE	PRINT	FDB7	PROMPT	FD34
PSPT	0020	PRTFLG	E403	PSTAK	FF1E	PTTAB	E4C0	QUEST	FC8C
RAM	E400	READY	F823	REPT	0003	REPTA	F975	REPTA1	F88A
REPTA2	F892	REPTA3	F83A	REPTA4	F87D	RESERV	F851	RESV1	F954
RESV2	F980	ROLMOD	F87F	ROM	FCFE	RSTBRK	FE1E	RTIPC	FF96
RTISIM	FFB0	RTN	FE3E	RTSSIM	FF85	SAVEA	E4C2	SAVEX	E4C3
SAVTRC	FF9A	SBOOT	F5EF	SCROLD	F7C8	SCROLL	F7EE	SECR91	E8F2
SECREG	E822	SETB2	FE22	SETBRK	FE2A	SETMEM	E8F8	SFE1	FCE3
SHIFT	0001	SINVER	F8B8	SKIP1	0005	SKIP2	0008	SP	E434
STAB1	FB67	STAB1	FB3C	STACK	E43F	STAREG	E820	START	FCF4
START1	FCFA	START2	FD08	STKEND	E440	SWI	003F	SWI1	E436
SXIOS	FF05	SWI181	FF15	SWITCH	FE0C	SZRCL	0030	TILL	0004
TEMP1	E579	TEXP2	E578	TERM	FC77	TERM1	FC81	TEST	F625
TILL	0004	TINPT1	F8B2	TINPT2	F88A	TINPUT	F8A1	TOACIA	FC36
TRACE	FE4F	TRACE1	FE51	TRACE3	FE54	TRCADR	E404	TRCINH	FF24
TRCINS	E40E	TRKREG	E821	UA	FE14	UCRTC	F73F	UP	0020
UPLINE	F71B	VCR	0020	VERSIO	F981	VIAC	E810	VIACL	E840
VIAMUS	E850	VIDEO	E570	VLF	0009	VTEMP	E4C5	WAIT	FE13
WHAT	FD37	XHI	E40A	XLOW	E40B	XREG	E57F	XSAVE	E40C

```
                        NAM   RBASK7
                        OPT   PAG
                *************************************
                *
                *    Gestion modem 300 bauds ( coupleur acoustique )
                *
                *          cassettes 600 bauds
                *
                *      A. Benit  - janvier 81 -
                *
                *************************************

EC00            ORGROM  EQU   $EC00
E4F5            ORGRAM  EQU   $E4F5     *  manoeuvre RAM
0000            D       EQU   0         *  deplacement pour implantation REPROM

                *------------------------------------------------- DEFS
                *
                *  adresses des peripheriques  VIA 6522
                *
E860            VIAT    EQU   $E850
                *
0000            IRB     EQU   0
0000            ORB     EQU   0
0002            DDRB    EQU   2
000C            PCR     EQU   12
                *
                * equivalences moniteur
                *
FFEF            INCH    EQU   $FFEF
FFEC            OUTCH   EQU   $FFEC
FFE6            MONIT   EQU   $FFE6
FFF2            PDATA   EQU   $FFF2
FFF5            CRLF    EQU   $FFF5
FD57            INAD    EQU   $FD57
                *
FFF2            CHAIN   EQU   PDATA
FFF5            RCLF    EQU   CRLF
                *
                *   zones diverses RAM
                *
E414                    ORG   $E414     zone RAM initialisee a 0 au RESET
E414            BFLAG   RMB   1         drapeau pour tampon
E415            EFLAG   RMB   1         drapeau BASIC
E419                    ORG   $E419
E419            MFLG    RMB   1
E4F5                    ORG   ORGRAM
E4F5            SVX1    RMB   2
E4F7            SVX2    RMB   2
E4F9            SVX3    RMB   2
E4FB            SVX4    RMB   2
E4FD            SVX5    RMB   2
E4FF            STACK   RMB   2
E501            NESSAIS RMB   1         * nombre de tentatives
E502            LRCC    RMB   1         *  LRC Calcule
E503            LPC     RMB   1         * Longueur Partielle calculee
E504            CRECU   RMB   1         * caractere recu
                *
                *  buffers pour echange de messages
                *
E505            B1      RMB   5
E505            SYN     EQU   B1
E506            L       EQU   B1+1      * et + 2
E508            S81     EQU   B1+3
E509            ENQ     EQU   B1+4
                *
E50A            B2      RMB   4
E50A            SOH     EQU   B2
E50B            LPB2    EQU   B2+1
```

```
RBASK7                          12-9-81  TSC ASSEMBLER  PAGE    2

   71  E50C           SQ2     EQU    B2+2
   72  E50D           STX     EQU    B2+3
   73                 *
   74  E50E           B3      RMB    1
   75  E50E           EOT     EQU    B3
   76                 *
   77  E50F           BAK     RMB    2
   78  E50F           ACKNAK  EQU    BAK
   79  E510           SQ3     EQU    BAK+1
   80                 *
   81                 * RAM pour modem BASIC
   82                 *
   83  E511           ERRB    RMB    1         code d erreur
   84  E512           PTBUF   RMB    2         pointeur tampon
   85  E514           BUF     RMB    80        tampon 80 octets
   86  E564           ENDBF   EQU    *         fin tampon
   87                 *
   88                 * RAM manoeuvre pour cassette
   89                 *
   90  E564           ETIQ    RMB    2         * etiquette demandee
   91  E566           ADRESD  RMB    2
   92  E568           ADRESF  RMB    2
   93  E56A           CSUM    RMB    1
   94  E56B           COMPTA  RMB    2
   95  E56D           ETIQLU  RMB    2
   96  E566           AD      EQU    ADRESD    ADRESSE DEBUT
   97  E568           AF      EQU    ADRESF    ADRESSE FIN
```

```
RBASK7                          12-9-81  TSC ASSEMBLER  PAGE    3

   99  EC00                    ORG    ORGROM
  101                 *------------------------------------------------------ VECTS
  102                 *
  103                 *  Vecteurs de branchement pour MON et BASIC
  104                 *
  105  EC00 7E EC 0F          JMP    KMON+0    COMMANDE K MONITEUR
  106  EC03 7E EC 88          JMP    SPMOD     ENVOI PROGRAMME BASIC
  107  EC06 7E EC AC          JMP    RPMOD     RECEPTION PROGRAMME BASIC
  108  EC09 7E EC DC          JMP    SDMOD     ENVOI DONNEES
  109  EC0C 7E ED 18          JMP    RDMOD     RECEPTION DONNEES
  110                 *
```

```
RBASK7                          12-9-81  TSC ASSEMBLER  PAGE    4

                        *-------------------------------------------- KMON
                        *
                        *  Gestion de la commande K moniteur
                        *
                        *--------------------------------------------

EC0F BF E4 FF  KMON     STS    STACK
EC12 CE F2 2C           LDX    £MLERT+D
EC15 BD FF F2           JSR    CHAIN
EC18 BD FF EF           JSR    INCH
EC1B 81 54              CMP A  £'T
EC1D 26 09              BNE    VOIRR
                        *               TRANSMISSION MONITEUR
EC1F 8D 2D              BSR    DAD
EC21 8D 38              BSR    DAF
EC23 BD ED 4D           JSR    EZ+D
EC26 20 23              BRA    MON
EC28 81 52     VOIRR    CMP A  £'R
EC2A 26 07              BNE    VOIRE
                        *              RECEPTION MONITEUR
EC2C 8D 20              BSR    DAD
EC2E BD EE F6           JSR    RZ+D
EC31 20 18              BRA    MON
EC33 81 45     VOIRE    CMP A  £'E
EC35 26 0B              BNE    VOIRL
                        *            ECRITURE CASSETTE MONITEUR
EC37 8D 15              BSR    DAD
EC39 8D 20              BSR    DAF
EC3B 8D 2B              BSR    DETIQ
EC3D BD F1 14           JSR    EZK+D
EC40 20 09              BRA    MON
EC42 81 4C     VOIRL    CMP A  £'L
EC44 26 05              BNE    MON
                        *              LECTURE CASSETTE MONITEUR
EC46 8D 20              BSR    DETIQ
EC48 BD F1 7E           JSR    LZK+D
EC4B 7E FF E6  MON      JMP    MONIT
                        *
                        * demande adresse etiquette
                        *
EC4E CE F2 3E  DAD      LDX    £MAD+D
EC51 BD FF F2           JSR    CHAIN
EC54 BD FD 57           JSR    INAD
EC57 FF E5 66           STX    AD
EC5A 39                 RTS
                        *
EC5B CE F2 3F  DAF      LDX    £MAF+D
EC5E BD FF F2           JSR    CHAIN
EC61 BD FD 57           JSR    INAD
EC64 FF E5 68           STX    AF
EC67 39                 RTS
                        *
EC68 CE 30 30  DETIQ    LDX    £$3030
```

```
RBASK7                              12-9-81  TSC ASSEMBLER  PAGE   5

  166  EC6B FF E5 64          STX    ETIQ
  167  EC6E CE F2 48          LDX    £METIQ+D
  168  EC71 BD FF F2          JSR    CHAIN
  169  EC74 BD FF EF          JSR    INCH
  170  EC77 81 0D             CMP A  £$0D
  171  EC79 27 0D             BEQ    DETIQX
  172  EC7B B7 E5 64          STA A  ETIQ
  173  EC7E BD FF EF          JSR    INCH
  174  EC81 81 2D             CMP A  £$0D
  175  EC83 27 03             BEQ    DETIQX
  176  EC85 B7 E5 65          STA A  ETIQ+1
  177  EC88 7E FF F5  DETIQX  JMP    RCLF     * qui fera le RTS ...
```

```
RBASK7                          12-9-81  TSC ASSEMBLER  PAGE    5

                        *-------------------------------------------------- IBAS
                        *
                        *  interface BASIC --) logiciel modem
                        *
                        *
                        * EMISSION PROGRAMME BASIC
                        *
EC8B BF E4 FF   SPMOD   STS     STACK
EC8E 8D 45              BSR     INBAS
EC90 A6 00              LDA A   0,X
EC92 B7 E5 66           STA A   AD
EC95 A6 01              LDA A   1,X
EC97 B7 E5 67           STA A   AD+1        ADRESSE DEBUT
EC9A A6 02              LDA A   2,X
EC9C B7 E5 68           STA A   AF
EC9F A6 03              LDA A   3,X
ECA1 B7 E5 69           STA A   AF+1        ADRESSE FIN
ECA4 FF E4 FD           STX     SVX5
ECA7 BD ED 40           JSR     EZ          EMISSION
ECAA 20 22              BRA     RETB        RETOUR BASIC
                *
                * RECEPTION PROGRAMME BASIC
                *
ECAC BF E4 FF   RPMOD   STS     STACK
ECAF 8D 24              BSR     INBAS
ECB1 A6 00              LDA A   0,X
ECB3 B7 E5 66           STA A   AD
ECB6 A6 01              LDA A   1,X
ECB8 B7 E5 67           STA A   AD+1        ADRESSE DEBUT
ECBB FF E4 FD           STX     SVX5
ECBE BD EE F6           JSR     RZ          RECEPTION
ECC1 FE E4 FD           LDX     SVX5
ECC4 B6 E5 68           LDA A   AF
ECC7 A7 02              STA A   2,X
ECC9 B6 E5 69           LDA A   AF+1
ECCC A7 03              STA A   3,X         ADRESSE FIN
ECCE 7F E4 15   RETB    CLR     EFLAG
ECD1 FE E4 FD           LDX     SVX5
ECD4 39                 RTS                 RETOUR BASIC
ECD5 73 E4 15   INBAS   COM     EFLAG
ECD8 7F E5 11           CLR     ERRB
ECDB 39                 RTS
                *
                * EMISSION DONNEES BASIC
                *
ECDC BF E4 FF   SDMOD   STS     STACK
ECDF FF E4 FD           STX     SVX5
ECE2 8D F1              BSR     INBAS
ECE4 7D E4 14           TST     BFLAG
ECE7 26 09              BNE     SDM1
ECE9 CE E5 14           LDX     #BUF
ECEC FF E5 12           STX     PTBUF
ECEF 73 E4 14           COM     BFLAG
```

```
RBASK7                          12-9-81  TSC ASSEMBLER  PAGE    7

 233  ECF2 FE E5 12  SDM1    LDX    PTBUF
 234  ECF5 A7 00             STA A  0,X
 235  ECF7 08                INX
 236  ECF8 8C E5 64          CPX    #ENDBF
 237  ECFB 27 09             BEQ    SDM2
 238  ECFD 81 0D             CMP A  #$D
 239  ECFF 27 05             BEQ    SDM2
 240  ED01 FF E5 12          STX    PTBUF
 241  ED04 20 C8     RETB1   BRA    RETB
 242  ED06 09        SDM2    DEX
 243  ED07 FF E5 68          STX    AF
 244  ED0A CE E5 14          LDX    #BUF
 245  ED0D FF E5 66          STX    AD
 246  ED10 BD ED 4D          JSR    EZ
 247  ED13 7F E4 14          CLR    BFLAG
 248  ED16 20 BE             BRA    RETB
 249                 *
 250                 * RECEPTION DONNEES BASIC
 251                 *
 252  ED18 BF E4 FF  RDMOD   STS    STACK
 253  ED1B FF E4 FD          STX    SVXS
 254  ED1E 8D B5             BSR    INBAS
 255  ED20 7D E4 14          TST    BFLAG
 256  ED23 26 0F             BNE    RDM1
 257  ED25 CE E5 14          LDX    #BUF
 258  ED28 FF E5 66          STX    AD
 259  ED2B FF E5 12          STX    PTBUF
 260  ED2E BD EE F6          JSR    RZ
 261  ED31 73 E4 14          COM    BFLAG
 262  ED34 FE E5 12  RDM1    LDX    PTBUF
 263  ED37 A6 00             LDA A  0,X
 264  ED39 08                INX
 265  ED3A 8C E5 64          CPX    #ENDBF
 266  ED3D 27 09             BEQ    RDM2
 267  ED3F 81 0D             CMP A  #$D
 268  ED41 27 05             BEQ    RDM2
 269  ED43 FF E5 12          STX    PTBUF
 270  ED46 20 BC             BRA    RETB1
 271  ED48 7F E4 14  RDM2    CLR    BFLAG
 272  ED4B 20 B7             BRA    RETB1
```

```
RBASK7                          12-9-81  TSC ASSEMBLER  PAGE    8


 275                    *-------------------------------------------------- EZB
 276                    *
 277                    *  Emission Zone  de (AD) a (AF)
 278                    *
 279  ED4D B6 E5 69     EZ      LDA A  AF+1      * calcul L := AF - AD + 1
 280  ED50 B0 E5 67             SUB A  AD+1
 281  ED53 16                   TAB
 282  ED54 B6 E5 68             LDA A  AF
 283  ED57 B2 E5 66             SBC A  AD
 284  ED5A 24 05                BCC    AOK
 285  ED5C 86 01                LDA A  £1
 286  ED5E 7E F2 7B             JMP    ERREUR+D
 287  ED61 CB 01        AOK     ADD B  £1
 288  ED63 F7 E5 07             STA B  L+1
 289  ED66 89 00                ADC A  £0
 290  ED68 B7 E5 06             STA A  L
 291  ED6B 86 16                LDA A  £$16      * preparation B1
 292  ED6D B7 E5 05             STA A  SYN
 293  ED70 4F                   CLR A
 294  ED71 B7 E5 01             STA A  NESSAIS
 295  ED74 B7 E5 08             STA A  SQ1
 296  ED77 86 05                LDA A  £5
 297  ED79 B7 E5 09             STA A  ENQ
 298  ED7C CE E5 05     EB1     LDX    £B1
 299  ED7F C6 05                LDA B  £5
 300  ED81 BD EE 16             JSR    EB+D      * emission B1
 301  ED84 CE ED 7C             LDX    £EB1+D
 302  ED87 BD ED E5             JSR    ATTAQR+D
 303  ED8A CE ED D1     EBSUIV  LDX    £EB3+D
 304  ED8D BD EE B2             JSR    CALCLP+D
 305  ED90 B7 E5 0B             STA A  LPB2
 306  ED93 86 01                LDA A  £1        * preparation B2
 307  ED95 B7 E5 0A             STA A  SQH
 308  ED98 B6 E5 08             LDA A  SQ1
 309  ED9B 4C                   INC A            *      +++   +1 sequence emission  +++
 310  ED9C B7 E5 08             STA A  SQ1
 311  ED9F B7 E5 0C             STA A  SQ2
 312  EDA2 86 02                LDA A  £2
 313  EDA4 B7 E5 00             STA A  STX
 314                    *
 315  EDA7 CE E5 0A     EB2     LDX    £B2
 316  EDAA C6 04                LDA B  £4
 317  EDAC BD EE 16             JSR    EB+D      * emission B2 *
 318  EDAF 7F E5 02             CLR    LRCC
 319  EDB2 FE E5 6E             LDX    AD
 320  EDB5 F5 E5 03             LDA B  LPC
 321  EDB8 BD EE 35             JSR    EMDTS+D   *  emission DONNEES
 322  EDBB B6 E5 02             LDA A  LRCC
 323  EDBE BD EE 47             JSR    EIMOT+D   * emission LRC
 324  EDC1 86 03                LDA A  £3
 325  EDC3 BD EE 47             JSR    EIMOT+D   * emission ETX
 326  EDC6 CE ED 97             LDX    £EB2+D
 327  EDC9 BD ED E5             JSR    ATTAQR+D
```

```
RBASK7                          12-9-81  TSC ASSEMBLER  PAGE    9

  328  EDCC BD EE D3          JSR    ADJADL+D
  329  EDCF 20 B9             BRA    EBSUIV
  330                  *
  331  EDD1 86 04     EB3     LDA A  £4
  332  EDD3 B7 E5 0E          STA A  EOT
  333  EDD6 CE E5 0E          LDX    £EB3
  334  EDD9 C6 01             LDA B  £1
  335  EDDB BD EE 16          JSR    EB+D      * emission B3 ( EOT )
  336  EDDE CE ED D1          LDX    £EB3+D
  337  EDE1 BD ED E5          JSR    ATTAQR+D
  338  EDE4 39                RTS             *       // fin normale EZ //
  339                  **********************
  340                  *
  341                  *   ATTente Accuse de Reception
  342                  *
  343
  344  EDE5 FF E4 F5  ATTAQR  STX    SVX1
  345  EDE8 CE E5 0F          LDX    £BAK
  346  EDEB C6 02             LDA B  £2
  347  EDED BD EF DA          JSR    RB+D
  348  EDF0 BE E5 0F          LDA A  ACKNAK
  349  EDF3 81 86             CMP A  £6
  350  EDF5 26 0C             BNE    RNAK
  351  EDF7 96 E5 10          LDA A  SQ3
  352  EDFA B1 E5 0E          CMP A  SQ1
  353  EDFD 26 04             BNE    RNAK
  354  EDFF 7F E5 01          CLR    NESSAIS   * on a recu ACK et un no seq correct
  355  EE02 39                RTS
  356  EE03 F6 E5 01  RNAK    LDA B  NESSAIS
  357  EE06 5C                INC B
  358  EE07 C1 04             CMP B  £4
  359  EE09 25 05             BNE    INSAKR
  360  EE2B 86 C2             LDA A  £2
  361  EE20 7E F2 7B          JMP    ERREUR+D
  362  EE10 F7 E5 01  INSAKR  STA B  NESSAIS
  363  EE13 7E EF DE          JMP    INSAK+D   ** retour a l adresse de reprise
  364                  **********************
  365                  *
  366                  *  Emission Bloc d adresse ( X ) long. ( B )
  367                  *
  368
  369  EE16 FF E4 F8  EB      STX    SVX4
  370  EE19 86 7E             LDA A  £%01111110
  371  EE1B B7 E6 62          STA A  VIA1+DORB
  372  EE1E CE 03 E4          LDX    £100      *  zero pendant 300 ms.
  373  EE21 FF E4 F7  EBL     STX    SVX2
  374  EE24 BD 5C             BSR    EMEZRO
  375  EE26 FE E4 F7          LDX    SVX2
  376  EE29 09                DEX
  377  EE2A 26 F5             BNE    EBL
  378  EE2C CE 04 E2          LDX    £1250     * on pendant 10 ms
  379  EE2F BD EE 9E          JSR    DELAI+D
  380  EE32 FE E4 F8          LDX    SVX4
  381  EE35 A6 00     EXCTB   LDA A  0,X
```

```
RBASK7                          12-9-81  TSC ASSEMBLER  PAGE  10

  382  EE37 36              PSH A
  383  EE38 5B E5 02        ADD A  LRCC
  384  EE3B B7 E5 02        STA A  LRCC
  385  EE3E 32              PUL A
  386  EE3F 9D EE 47        JSR    E1MOT+D
  387  EE42 08              INX
  388  EE43 5A              DEC B
  389  EE44 26 EF           BNE    EMOTS
  390  EE46 39              RTS
```

```
RBASK7                          12-9-81  TSC ASSEMBLER  PAGE   11

                        *-------------------------------------------------------EMOT
                        *
                        *
                        *     Emission 300 bauds
                        *
                        
                        ***
                        *     delais pour transmission
                        ***
001C                    X0H     EQU   28
001B                    X0B     EQU   27
0006                    NPER    EQU   6
0171                    X1E     EQU   369
                        *
EE47 37                 EMOT    PSH B
EE48 36                         PSH A
EE49 FF E4 F9                   STX   SVX3
EE4C 86 60                      LDA A £%01100000 * P 5 ET 6 EN SORTIE
EE4E B7 E8 62                   STA A VIAT+DDRB
EE51 B6 E8 6C           DEBC    LDA A VIAT+PCR
EE54 84 F1                      AND A £%11110001
EE56 8A 0C                      ORA A £%00001100 *  CA2 = 0  sync osc.
EE58 B7 E8 6C                   STA A VIAT+PCR
                        *
EE5B 86 20                      LDA A £%00100000 * PB 6-5 = 1/2
EE5D CE 00 03                   LDX   £3
EE60 BD EE AE                   JSR   DELAI     *   pulse sync £ 55 us
EE63 B6 E8 6C                   LDA A VIAT+PCR
EE66 8A 0E                      ORA A £%00001110 * CA2 = 1  sync. osc
EE68 B7 E8 6C                   STA A VIAT+PCR
EE6B 32                         PUL A
EE6C C6 0B                      LDA B £11       *   start + 8 bits + 2 stops
EE6E 0C                         CLC             *   pour start initial
EE6F 36                 EMBIT   PSH A           * sauver caractere decale
EE70 25 04                      BCS   EMIS1
EE72 8D 0E                      BSR   EMEZRO
EE74 20 20                      BRA   SHIFTC
                        ***
EE76 86 20              EMIS1   LDA A £%00100000 *  PB6-5 = 1/2
EE78 B7 E8 60                   STA A VIAT+ORB
EE7B CE 01 71                   LDX   £X1E
EE7E 8D 2E                      BSR   DELAI
EE80 20 21                      BRA   SHIFTC
                        ****
EE82 37                 EMEZRO  PSH B
EE83 C6 06                      LDA B £NPER
EE85 86 60              EMIP    LDA A £%01100000 * PB6-5 = 1
EE87 B7 E8 60                   STA A VIAT+ORB
EE8A CE 00 1C                   LDX   £X0H
EE8D 8D 1F                      BSR   DELAI
EE8F 86 00                      LDA A £%00000000 * PB5-5 = 0
EE91 B7 E8 60                   STA A VIAT+ORB
EE94 CE 00 1B                   LDX   £X0B
```

```
RBASK7                                  12-9-81  TSC ASSEMBLER  PAGE   12

  446  EE97 8D 15           BSR    DELAI
  447  EE99 5A              DEC B
  448  EE9A 26 E9           BNE    EM1P
  449  EE9C B6 20           LDA A  £%00100000 * PB6-5 = 1/2
  450  EE9E B7 E8 E0        STA A  VIAT+ORB
  451  EEA1 33              PUL B
  452  EEA2 39              RTS
  453                  ***
  454  EEA3 32        SHIFTC  PUL A
  455  EEA4 0D              SEC
  456  EEA5 46              ROR A
  457  EEA6 5A              DEC B
  458  EEA7 26 C6           BNE    EMBIT
  459                  ***
  460  EEA9 FE E4 F9        LDX    SVX3
  461  EEAC 33              PUL B
  462  EEAD 39              RTS
  463                  **
  464  EEAE 09        DELAI   DEX
  465  EEAF 26 FD           BNE    DELAI
  466  EEB1 39              RTS
  467                  ***
```

```
RBASK7                          12-9-81  TSC ASSEMBLER  PAGE   13


                        *-------------------------------------------------- SPER
                        *
                        * Sous Programmes communs Emission Reception
                        *

                        ****************
                        *
                        *   CALCul de Longueur Partielle ( taille d un bloc )
                        *
                        *    en fonction de la longueur de la zone
                        *
                        *     restatant a emettre ou a recevoir
                        *

EEB2 FF E4 F5   CALCLP  STX     SVX1
EEB5 FE E5 06           LDX     L
EEB8 26 07              BNE     CDX
EEBA 31                 INS             * L = 0 retour a l adresse d exception ( X )
EEBB 31                 INS             * reajustement du pointeur de pile
EEBC FE E4 F5           LDX     SVX1
EEBF 6E 00              JMP     0,X

EEC1 B6 E5 06   CDX     LDA A   L
EEC4 26 07              BNE     MAX1    *  car L ) 256
EEC6 B6 E5 07           LDA A   L+1
EEC9 81 50              CMP A   #80
EECB 2D 02              BLT     STLPC
EECD 86 50      MAX1    LDA A   #80
EECF B7 E5 03   STLPC   STA A   LPC
EED2 39                 RTS

                        ********************=*********************
                        *
                        *  Ajustement Adresse Debut et  L
                        *

EED3 B6 E5 67   ADJADL  LDA A   AD+1    *  AD := AD + LPC
EED6 BB E5 03           ADD A   LPC
EED9 B7 E5 67           STA A   AD+1
EEDC B6 E5 66           LDA A   AD
EEDF 89 00              ADC A   #0
EEE1 B7 E5 66           STA A   AD
                        *
EEE4 B6 E5 07           LDA A   L+1     *  L:= L - LPC
EEE7 B0 E5 03           SUB A   LPC
EEEA B7 E5 07           STA A   L+1
EEED B6 E5 06           LDA A   L
EEF0 82 00              SBC A   #0
EEF2 B7 E5 06           STA A   L
EEF5 39                 RTS
```

```
 522                    *---------------------------------------------------- RZB
 523                    *
 524                    *  Reception Zone de ( AD) a ( AD ) + ( L )
 525                    *
 526                    *
 527
 528  EEF6 4F           RZ      CLR A
 529  EEF7 B7 E5 01             STA A  NESSAIS
 530  EEFA CE E5 05     RB1     LDX    £B1
 531  EEFD C6 05                LDA B  £5
 532  EEFF BD EF DA             JSR    RB+D      * reception B1
 533                    *
 534  EF02 CE EE FA             LDX    £RB1+D
 535  EF05 B6 E5 05             LDA A  SYN
 536  EF08 81 16                CMP A  £$16
 537  EF0A 26 0A                BNE    RI1
 538  EF0C B6 E5 08             LDA A  SQ1
 539  EF0F 26 05                BNE    RI1
 540  EF11 B6 E5 09             LDA A  ENQ
 541  EF14 81 05                CMP A  £5
 542  EF16 BD EF 9C     RI1     JSR    ENVAQR+D
 543  EF19 CE EF 89     RBSUIV  LDX    £RB3+D
 544  EF1C BD EE B2             JSR    CALCLP+D
 545  EF1F 7C E5 08             INC    SQ1       *      ///  + 1  sequence  ///
 546  EF22 CE E5 0A     RB2     LDX    £B2
 547  EF25 C6 04                LDA B  £4
 548  EF27 BD EF DA             JSR    RB+D      * reception debut B2
 549  EF2A 7F E5 02             CLR    LRCC
 550  EF2D B6 E5 0A             LDA A  SOH
 551  EF30 81 01                CMP A  £1
 552  EF32 26 17                BNE    RI2
 553  EF34 B6 E5 0B             LDA A  LPB2
 554  EF37 B1 E5 03             CMP A  LPC
 555  EF3A 26 0F                BNE    RI2
 556  EF3C B6 E5 0C             LDA A  SQ2
 557  EF3F B1 E5 08             CMP A  SQ1
 558  EF42 26 07                BNE    RI2
 559  EF44 B6 E5 0D             LDA A  STX
 560  EF47 81 02                CMP A  £2
 561  EF49 27 08                BEQ    RSUIB2
 562  EF4B CE FF FF     RI2     LDX    £$FFFF    *  VERIFIER SI SUFFISANT ////
 563  EF4E BD EE 9E     DELSAUT JSR    DELAI+D   *  laisser passer les donnees ....
 564  EF51 20 26                BRA    ENVNAK    *   avant d envoyer NAK
 565  EF53 FE E5 66     RSUIB2  LDX    AD
 566  EF56 F6 E5 03             LDA B  LPC
 567  EF59 BD F0 01             JSR    RMOTS+D   * reception DONNEES
 568  EF5C 09                   DEX
 569  EF5D FF E5 68             STX    AF
 570  EF60 08                   INX
 571                    *
 572  EF61 BD F0 10             JSR    RIMOT+D   * reception LRC
 573  EF64 B1 E5 02             CMP A  LRCC
 574  EF67 27 05                BEQ    RETX
```

```
RBASK7                          12-9-81  TSC ASSEMBLER  PAGE   15

  575  EF69 CE 11 11          LDX    £$1111
  576  EF6C 20 E0             BRA    DELSAUT   *  laisser passer ETX
  577  EF6E BD F0 10  RETX    JSR    RIMOT+D   * reception ETX
  578  EF71 81 03             CMP A  £3
  579  EF73 20 06             BRA    ENVV2
  580  EF75 86 00     ENVACK  LDA A  £0
  581  EF77 20 02             BRA    ENVV2
  582  EF79 86 FF     ENVNAK  LDA A  £$FF
  583  EF7B 07        ENVV2   TPA
  584  EF7C 36                PSH A
  585  EF7D CE EF 22          LDX    £RB2+D
  586  EF80 32                PUL A
  587  EF81 06                TAP
  588  EF82 8D 18             BSR    ENVAQR
  589  EF84 BD EE D3          JSR    ADJADL+D
  590  EF87 20 90             BRA    RBSUIV
  591                 *
  592  EF89 CE E5 0E  RB3     LDX    £B3
  593  EF8C C6 01             LDA B  £1
  594  EF8E BD EF DA          JSR    RB+D      *  reception B3 ( EOT )
  595  EF91 CE EF 89          LDX    £RB3+D
  596  EF94 B6 E5 0E          LDA A  EOT
  597  EF97 81 04             CMP A  £4
  598  EF99 8D 01             BSR    ENVAQR
  599  EF9B 39                RTS
  600
  601                 ****************************
  602                 *
  603                 *   ENVoi Accuse de Reception
  604                 *
  605                 *     Z=1  =) ACK et   0-)NESSAIS
  606                 *
  607                 *     Z=0  =) NAK et controle NESSAIS :: 4
  608
  609  EF9C 07        ENVAQR  TPA
  610  EF9D 36                PSH A
  611  EF9E FF E4 F5          STX    SVX1
  612  EFA1 06                TAP
  613  EFA2 27 07             BEQ    EAK
  614  EFA4 86 0F             LDA A  £15       * NAK Ascii
  615  EFA6 7C E5 01          INC    NESSAIS
  616  EFA9 20 03             BRA    STABQ
  617                 *
  618  EFAB 86 06     EAK     LDA A  £06       *  ACK Ascii
  619  EFAD 5F                CLR B
  620  EFAE B7 E5 0F  STABQ   STA A  ACKNAK
  621  EFB1 F7 E5 01          STA B  NESSAIS
  622  EFB4 B6 E5 08          LDA A  SQ1
  623  EFB7 B7 E5 10          STA A  SQ3
  624  EFBA CE E5 0F          LDX    £BAK
  625  EFBD C6 02             LDA B  £2
  626  EFBF BD EE 16          JSR    EB+D      *  emission buffer ACK ou NAK  + seq
  627  EFC2 B6 E5 01          LDA A  NESSAIS   *  controler si moins de 4 essais
  628  EFC5 81 04             CMP A  £4
```

```
RBASK7                          12-9-81  TSC ASSEMBLER  PAGE   15

 629  EFC7 27 0C           BEQ    ERR3
 630  EFC9 32              PUL A
 631  EFCA 06              TAP
 632  EFCB 26 01           BNE    INSAK
 633  EFCD 39              RTS             *      === retour normal  ===
 634  EFCE 31       INSAK  INS
 635  EFCF 31              INS             *  ajuster pointeur de pile *
 636  EFD0 FE E4 F5        LDX    SVX1
 637  EFD3 6E 00           JMP    0,X      * retour d exception
 638  EFD5 86 03    ERR3   LDA A  £3       *  erreur ---) retour
 639  EFD7 7E F2 7B        JMP    ERREUR
 640
 641                 ********************************
 642                 *
 643                 *     Reception de Bloc  adresse ( X ) long. ( B )
 644                 *
 645
 646  EFDA FF E4 F7  RB    STX    SVX2
 647  EFDD 37              PSH B
 648  EFDE 86 7E           LDA A  £%01111110
 649  EFE0 B7 E8 62        STA A  VIAT+DDRB
 650  EFE3 5F       ATTZRO CLR B
 651  EFE4 B6 E8 60 TB00   LDA A  VIAT+IRB
 652  EFE7 84 01           AND A  £1
 653  EFE9 26 F8           BNE    ATTZRO
 654  EFEB CE 01 3E        LDX    £310
 655  EFEE BD EE AE        JSR    DELAI+D  *  environ 2.5 ms
 656  EFF1 5C              INC B
 657  EFF2 C1 14           CMP B  £20
 658  EFF4 25 EE           BNE    TB00
 659                 *  on est sur du zero initial sur 50 ms
 660  EFF6 B6 E8 60 TB01   LDA A  VIAT+IRB
 661  EFF9 84 01           AND A  £1
 662  EFFB 27 F9           BEQ    TB01
 663                 *  on a detecte le pseudo bit STOP
 664  EFFD FE E4 F7        LDX    SVX2
 665  F000 33              PUL B
 666
 667  F001 BD 0D    RMOTS  BSR    RIMOT
 668  F003 A7 00           STA A  0,X
 669  F005 BB E5 02        ADD A  LRCC
 670  F008 B7 E5 02        STA A  LRCC
 671  F00B 08              INX
 672  F00C 5A              DEC B
 673  F00D 26 F2           BNE    RMOTS
 674  F00F 39              RTS
```

```
ABASK7                          12-9-81  TSC ASSEMBLER  PAGE   17


677                 *---------------------------------------------------------- RMOT
678                 *
679                 *
680                 *   Reception 300 bauds
681                 *
682
683                 **   definition des constantes de delai
684
685  0014           XSTOP   EQU   20
686  0032           X01     EQU   53-3
687  001A           X02     EQU   26
688  008D           X1AR    EQU   141
689  00E2           X3AR    EQU   225
690  00EA           X3      EQU   240-6
691
692
693                 ***
694
695  F010 37        R1MOT   PSH B
696  F011 FF E4 F9          STX   SVX3
697
698  F014 B6 E8 60  RCC     LDA A VIAT+IRB
699  F017 84 01             AND A £1
700  F019 27 F9             BEQ   RCC       * recherche du stop
701  F01B CE 00 14          LDX   £XSTOP
702  F01E BD 4A             BSR   DELAIR
703  F020 B6 E8 60          LDA A VIAT+IRB
704  F023 84 01             AND A £1
705  F025 27 ED             BEQ   RCC       * PB0=1 instable re-chercher
706  F027 86 21             LDA A £1        * PB0=1 STABLE
707  F029 B5 E8 60  ATTSTR  BIT A VIAT+IRB  * boucle de
708  F02C 26 FB             BNE   ATTSTR    * detection du start
709  F02E 01        CLRCA2  NOP             **
710  F02F B6 EB 6C          LDA A VIAT+PCR  **
711  F032 84 11             AND A £%00010001
712  F034 8A CC             ORA A £%11001100 ** CA2 - CB2 niveau bas
713  F036 B7 EB 6C          STA A VIAT+PCR  **
714  F039 CE 00 32          LDX   £X01
715  F03C BD 25             BSR   DELAIR    * 01 = 24+9*X01
716  F03E B6 EB 6C          LDA A VIAT+PCR
717  F041 84 11             AND A £%00010001
718  F043 8A EE             ORA A £%11101110 ** CA2 + CB2 niveau haut
719  F045 B7 EB 6C          STA A VIAT+PCR
720  F048 B6 E8 60          LDA A VIAT+IRB
721  F04B 84 01             AND A £1
722  F04D 26 C5             BNE   RCC       * START INSTABLE RE_SYNCHRO
723  F04F CE 00 1A          LDX   £X02
724  F052 BD 16             BSR   DELAIR    * T02 = 36 + 9*X02
725  F054 BD 13             BSR   R1BIT     * pour confirmer le start
726  F056 25 BC             BCS   RCC       * perte du start RE-SYNCHRO
727  F058 C6 08             LDA B £8
728  F05A BD 12     BITSUI  BSR   R1BIT
729  F05C 76 E5 04          ROR   CRECU
```

```
RBASK7                              12-9-81  TSC ASSEMBLER  PAGE   18

  730  F05F 5A              DEC B
  731  F060 26 F8           BNE    BITSUI
  732
  733  F062 FE E4 F9        LDX    SVX3
  734  F065 B6 E5 04        LDA A  CRECU
  735  F068 33              PUL B
  736  F069 39              RTS              * T4 = 20
  737
  738  F06A 09       DELAIR DEX
  739  F06B 26 FD           BNE    DELAIR
  740  F06D 39              RTS
  741
  742                *
  743                **
  744                ***  RECEPTION 1 BIT
  745                **
  746                *
  747  F06E CE 00 8D R1BIT  LDX    £X1AR
  748  F071 8D F7           BSR    DELAIR
  749  F073 B6 E8 6C        LDA A  VIAT+PCR  *    INVERSION
  750  F076 88 20           EOR A  £X00100000 *      DE
  751  F078 B7 E8 6C        STA A  VIAT+PCR  *       C B 2
  752  F07B B6 E8 60        LDA A  VIAT+IRB
  753  F07E 46              ROR A
  754  F07F CE 00 E2        LDX    £X3AR
  755  F082 8D E6           BSR    DELAIR
  756  F084 39              RTS
```

```
                *-------------------------------------------------- GK7
                *
                *  Gestion cassette
                *
                * S/P DE DELAI SUIVANT (B)
F085 5A         DELAY   DEC B
F086 01                 NOP
F087 26 FC              BNE     DELAY     * DELAI=8(B)+16 MIC.SEC
F089 39                 RTS
                * S/P D'ECRITURE 1 BIT=0 OU 1 SUIVANT (A)
F08A 37         ECRBIT  PSH B
F08B 5F         ECRBI1  CLR B
F08C F7 E8 60           STA B  VIAT+ORB  *     NIVEAU "0"
F08F C6 16              LDA B  £22
F091 8D F2              BSR     DELAY     * DELAI 192 MIC.SEC
F093 C6 60              LDA B  £$60
F095 F7 E8 60           STA B  VIAT+ORB  *    NIVEAU "1"
F098 C6 16              LDA B  £22
F09A 8D E9              BSR     DELAY     * DELAI 192 MIC.SEC
F09C C6 20              LDA B  £$20
F09E F7 E8 60           STA B  VIAT+ORB  *    NIVEAU "1/2"
F0A1 C6 20              LDA B  £32
F0A3 8D E0              BSR     DELAY     * DELAI 270 MIC.SEC
F0A5 4D                 TST A
F0A6 26 06              BNE     PAZERO    *  CAS DU BIT "1"
F0A8 C6 4D              LDA B  £$4D
F0AA 8D D9              BSR     DELAY
F0AC 33         SORBIT  PUL B
F0AD 39                 RTS
F0AE 2B FC      PAZERO  BMI     SORBIT    * FIN DU BIT "1"
F0B0 86 FF              LDA A  £$FF       *  COMPLETER LE BIT "1"
F0B2 20 D7              BRA     ECRBI1
                * S/P D'ECRITURE DE 1 MOT  =(A)
F0B4 C6 08      ECRMOT  LDA B  £8
F0B6 36         ENCORE  PSH A
F0B7 84 01              AND A  £1         *  ISOLER LE BIT DU BAS
F0B9 8D CF              BSR     ECRBIT    * L'ECRIRE SUR LA CASSETTE
F0BB 32                 PUL A
F0BC 44                 LSR A
F0BD 5A                 DEC B
F0BE 26 F6              BNE     ENCORE
F0C0 39                 RTS
                * S/P DE LECTURE 1 BIT,0 OU 1 VERS (C)
F0C1 36         LIRBIT  PSH A
F0C2 37                 PSH B
F0C3 C6 80              LDA B  £$80
F0C5 5A         OUTTIM  DEC B
F0C6 2B 29              BMI     LBX
```

```
RBASK7                              12-9-81  TSC ASSEMBLER  PAGE   20

 812  F0C8 B6 E8 60  TIM2    LDA A  VIAT+IRB
 813  F0CB BA E8 60          ORA A  VIAT+IRB
 814  F0CE BA E8 60          ORA A  VIAT+IRB
 815  F0D1 84 80             AND A  £$80
 816  F0D3 26 F0             BNE    OUTTIM
 817  F0D5 C6 50             LDA B  £$50      PARTIE DU DELAI 780 MIC.SEC
 818  F0D7 8D AC             BSR    DELAY
 819  F0D9 C6 10             LDA B  £$10      * FENETRE DE 300 MIC SEC
 820  F0DB B6 E8 60  TEST1   LDA A  VIAT+IRB
 821  F0DE 84 80             AND A  £$80
 822  F0E0 27 0A             BEQ    LULEUN
 823  F0E2 5A                DEC B
 824  F0E3 26 F6             BNE    TEST1
 825  F0E5 C6 20             LDA B  £$20
 826  F0E7 8D 9C             BSR    DELAY
 827  F0E9 0C                CLC              * SORTIE POUR LE ZERO
 828  F0EA 20 05             BRA    LBX
 829  F0EC C6 30     LULEUN  LDA B  £$30      * DELAI APRES LE UN
 830  F0EE 8D 95             BSR    DELAY     * 210 MIC SEC
 831  F0F0 0D                SEC              * SORTIE POUR LE UN
 832  F0F1 33        LBX     PUL B
 833  F0F2 32                PUL A
 834  F0F3 39                RTS
 835
 836                 * S/P DE LECTURE 1 MOT =(A) = MOT1
 837  F0F4 C6 08     LIRMOT  LDA B  £8
 838  F0F6 4F                CLR A
 839  F0F7 8D C8     TOUJOU  BSR    LIRBIT    *
 840  F0F9 2A 05             BPL    TIM3
 841  F0FB 31                INS
 842  F0FC 31                INS
 843  F0FD 7E F1 81          JMP    COMLR+D
 844  F100 46        TIM3    ROR A
 845  F101 5A                DEC B
 846  F102 26 F3             BNE    TOUJOU    * COMPTAGE 1 A 8
 847  F104 4D                TST A
 848  F105 39                RTS
 849
 850                 * S/P DE CHEK-SUM
 851  F106 16        CSOMME  TAB
 852  F107 FB E5 6A          ADD B  CSUM      * COMPLETER
 853  F10A F7 E5 6A          STA B  CSUM      * CSUM PAR (A)
 854  F10D 39                RTS
 855
 856  F10E 86 7F     INIPR   LDA A  £$7F      INIT DES BITS DE GESTION K7
 857  F110 B7 E8 62          STA A  VIAT+DDRB
 858  F113 39                RTS
 859
 860                 ********************************************
 861                 *
 862                 * MODULE D'ECRITURE K7
 863                 *
 864                 ********************************************
 865  F114 8D F8     EZK     BSR    INIPR
```

```
RBASK7                          12-9-81  TSC ASSEMBLER  PAGE   21

  866  F116 CE 13 88  COMETR  LDX    £5000
  867  F119 FF E5 6B  BAL3    STX    COMPTA    * 3" DE BALISES (DES ZEROS)
  868  F11C 4F                CLR A
  869  F11D BD F0 8A          JSR    ECRBIT+D
  870  F120 FE E5 6B          LDX    COMPTA
  871  F123 09                DEX
  872  F124 26 F3             BNE    BAL3
  873  F126 86 01             LDA A  £1        * LE "1" TOUT SEUL
  874  F128 BD F0 8A          JSR    ECRBIT+D
  875  F12B 7F E5 6A          CLR    CSUM
  876  F12E CE E5 64          LDX    £ETIQ
  877  F131 FF E4 F5  ECEK7   STX    SVX1
  878  F134 A6 00             LDA A  0,X
  879  F136 BD F1 06          JSR    CSOMME+D
  880  F139 BD F0 B4          JSR    ECRMOT+D
  881  F13C FE E4 F5          LDX    SVX1
  882  F13F 08                INX
  883  F140 8C E5 6A          CPX    £ETIQ+6
  884  F143 26 EC             BNE    ECEK7
  885  F145 FE E5 66          LDX    ADRESD    * EXTRAIRE 1 MOT EN RAM
  886  F148 A6 00     UNMOT   LDA A  0,X
  887  F14A BD F1 06          JSR    CSOMME+D  * COMPLETER "CS"
  888  F14D BD F0 B4          JSR    ECRMOT+D  * ECRIRE LE MOT SUR LA K7
  889  F150 BC E5 68          CPX    ADRESF    * TESTER LA FIN
  890  F153 27 03             BEQ    ECRICS
  891  F155 08                INX              * PROGRESSER L'ADRESSE MOT
  892  F156 20 F0             BRA    UNMOT
  893  F158 B6 E5 6A  ECRICS  LDA A  CSUM      * ECRIRE LE CS EN FIN
  894  F15B BD F0 B4          JSR    ECRMOT+D
  895  F15E 7D E4 19          TST    MFLG
  896  F161 26 14             BNE    EZKRT
  897  F163 8D 13             BSR    SIGNAL
  898  F165 CE F2 6F          LDX    £MECR+D
  899  F168 BD FF F2          JSR    CHAIN     * ECRITURE TERMINE
  900  F16B B6 E5 64          LDA A  ETIQ
  901  F16E BD FF EC          JSR    OUTCH     * IMP.L'ETIQUETTE
  902  F171 B6 E5 65          LDA A  ETIQ+1
  903  F174 BD FF EC          JSR    OUTCH
  904  F177 39        EZKRT   RTS
  905
  906                 * envoi du message fin
  907                 *
  908  F178 CE F2 67  SIGNAL  LDX    £MFIN+D
  909  F178 7E FF F2          JMP    CHAIN     * qui fait RTS
  910
  911                 ************************
  912                 *
  913                 * MODULE DE LECTURE K7
  914                 *
  915                 *
  916  F17E BD F1 0E  LZK     JSR    INIPR+D
  917  F181 86 32     COMLR   LDA A  £50
  918  F183 37 E5 6B          STA A  COMPTA
  919  F186 BD F0 F4  ZERMOT  JSR    LIRMOT+D  * ATTENTE DE 50 MOTS
```

```
RBASK7                                  12-9-81  TSC ASSEMBLER  PAGE   22

 920  F189 26 F6           BNE    COMLR     * SUPPOSES DANS LES BALISES
 921  F18B 7A E5 6B        DEC    COMPTA
 922  F18E 26 F5           BNE    ZERMOT
 923  F190 86 07           LDA A  £7
 924  F192 BD FF EC        JSR    OUTCH
 925  F195 86 32    COMLR1 LDA A  £50
 926  F197 B7 E5 6B        STA A  COMPTA
 927  F19A BD F0 F4 ZERMO1 JSR    LIRMOT+D
 928  F19D 26 F6           BNE    COMLR1
 929  F19F 7A E5 6B        DEC    COMPTA
 930  F1A2 26 F6           BNE    ZERMO1
 931  F1A4 BD F0 C1 ATTEN1 JSR    LIRBIT+D  * ATTENTE DU "1" TOUT SEUL
 932  F1A7 2B D8           BMI    COMLR
 933  F1A9 24 F9           BCC    ATTEN1
 934  F1AB BD F0 F4        JSR    LIRMOT+D  * LIRE ETIQUETTE
 935  F1AE 26 05           BNE    ETQL1
 936  F1B0 86 05           LDA A  £5
 937  F1B2 7E F2 7B        JMP    ERREUR+D
 938  F1B5 B7 E5 6D ETQL1  STA A  ETIQLU
 939  F1B8 BD F0 F4        JSR    LIRMOT+D
 940  F1BB B7 E5 6E        STA A  ETIQLU+1
 941  F1BE BB E5 6D        ADD A  ETIQLU
 942  F1C1 B7 E5 6A        STA A  CSUM
 943  F1C4 FE E5 64        LDX    ETIQ      * TEST DE CONFORMITE
 944  F1C7 8C 30 30        CPX    £$3030
 945  F1CA 27 05           BEQ    INFO
 946  F1CC BC E5 6D        CPX    ETIQLU    * ETIQUETTES
 947  F1CF 26 B0           BNE    COMLR
 948  F1D1 CE E5 66 INFO   LDX    £ADRESD
 949  F1D4 FF E4 F5 INFOSX STX    SVX1
 950  F1D7 BD F0 F4        JSR    LIRMOT+D
 951  F1DA FE E4 F5        LDX    SVX1
 952  F1DD A7 00           STA A  0,X
 953  F1DF BD F1 06        JSR    CSOMME+D
 954  F1E2 08              INX
 955  F1E3 BC E5 6A        CPX    £ADRESD+4
 956  F1E6 26 EC           BNE    INFOSX
 957  F1E8 FE E5 66        LDX    ADRESD
 958  F1EB BD F0 F4 LIR1MO JSR    LIRMOT+D  * LIRE UN MOT
 959  F1EE A7 00           STA A  X         * LE GARER
 960  F1F0 BD F1 06        JSR    CSOMME+D  * COMPLETER LE "CS"
 961  F1F3 BC E5 68        CPX    ADRESF    * TEST TERMINE
 962  F1F6 27 03           BEQ    LIRCS
 963  F1F8 08              INX
 964  F1F9 20 F0           BRA    LIR1MO
 965  F1FB BD F0 F4 LIRCS  JSR    LIRMOT+D  * LIRE "CS"
 966  F1FE B1 E5 6A        CMP A  CSUM      * comparer au checksum recalcule
 967  F201 27 0E           BEQ    BONNLEC
 968  F203 CE F2 75        LDX    £MLEC
 969  F206 BD FF F2        JSR    CHAIN
 970  F209 CE F2 5B        LDX    £MANOR+D
 971  F20C BD FF F2        JSR    CHAIN
 972  F20F 20 0E           BRA    LZKS
 973  F211 7D E4 19 BONNLEC TST   MFLG
```

```
RBASK7                          12-9-81  TSC ASSEMBLER  PAGE   23

  974  F214 26 15            BNE    LZKRT
  975  F216 BD F1 78         JSR    SIGNAL+D  * envoi BIP + mess fin
  976  F219 CE F2 75         LDX    #MLEC+D
  977  F21C BD FF F2         JSR    CHAIN     * IMP."TERMINE"
  978  F21F B6 E5 6D  LZKS   LDA A  ETIQLU
  979  F222 BD FF EC         JSR    OUTCH     * IMP.L'ETIQUETTE
  980  F225 B6 E5 6E         LDA A  ETIQLU+1
  981  F228 7E FF EC         JMP    OUTCH     * QUI FAIT LE RTS
  982  F22B 39        LZKRT  RTS
```

```
RBASK7                          12-9-81  TSC ASSEMBLER  PAGE   24

  985                   *---------------------------------------------------- MSGS
  986                   *
  987                   *  MESSAGES
  988                   *
  989  F22C 0A 0D       MLERT   FDB    $0A0D
  990  F22E 4C                  FCC    /LERT ? /
  991  F235 04                  FCB    4
  992  F236 0A 0D       MAD     FDB    $0A0D
  993  F238 44                  FCC    /DEB ?↙/
  994  F23E 04                  FCB    4
  995  F23F 0A 0D       MAF     FDB    $0A0D
  996  F241 46                  FCC    /FIN ? /
  997  F247 04                  FCB    4
  998  F248 0A 0D       METIQ   FDB    $0A0D
  999  F24A 45                  FCC    /ETI ? /
 1000  F250 04                  FCB    4
 1001  F251 0A 0D       MERRU   FDB    $0A0D
 1002  F253 45                  FCC    /Erreur /
 1003  F25A 04                  FCB    4
 1004  F25B 07          MANOR   FCB    7
 1005  F25C 41                  FCC    /ANORMALE /
 1006  F265 07                  FCB    7
 1007  F266 04                  FCB    4
 1008  F267 0A 0D       MFIN    FDB    $0A0D
 1009  F269 07                  FCB    7
 1010  F26A 46                  FCC    /Fin /
 1011  F26E 04                  FCB    4
 1012  F26F 65          MECR    FCC    /ecr. /
 1013  F274 04                  FCB    4
 1014  F275 6C          MLEC    FCC    /lec. /
 1015  F27A 04                  FCB    4
 1016                   *
 1017                   * envoi du message d erreur
 1018                   *
 1019  F27B 36          ERREUR  PSH A
 1020  F27C CE F2 51            LDX    £MERRU+D
 1021  F27F BD FF F2            JSR    CHAIN
 1022  F282 32                  PUL A
 1023  F283 B7 E5 11            STA A  ERRB
 1024  F286 8B 30               ADD A  £'0
 1025  F288 BD FF EC            JSR    OUTCH
 1026  F28B BE E4 FF            LDS    STACK
 1027  F28E 7D E4 15            TST    EFLAG
 1028  F291 26 03               BNE    ERRS
 1029  F293 7E FF E5            JMP    MONIT
 1030  F296 7F E4 15  ERRS      CLR    EFLAG
 1031  F299 7F E4 14            CLR    BFLAG
 1032  F29C FE E4 FD            LDX    SVX5
 1033  F29F 86 0D               LDA A  £$D
 1034  F2A1 39                  RTS              RETOUR BASIC
```

RBASK7 12-9-81 TSC ASSEMBLER PAGE 25

```
1037                    *---------------------------------------------- KBAS
1038                    *
1039                    *  interface BASIC --) logiciel cassette
1040                    *
1041  F300              ORG   $F300     POUR BASIC REPROM
1042  F300 7E F3 51     JMP   PDTAPE
1043  F303 7E F3 A0     JMP   RDTAPE
1044  F306 7E F3 2E     JMP   LPTAPE
1045                    *
1046                    * SAUVEGARDE PROGRAMME BASIC'
1047                    *
1048  F309 BF E4 FF SPTAPE  STS   STACK
1049  F30C BD EC D5     JSR   INBAS
1050  F30F A6 01        LDA A 1,X
1051  F311 B7 E5 66     STA A AD
1052  F314 A6 02        LDA A 2,X
1053  F316 B7 E5 67     STA A AD+1      ADRESSE DEBUT
1054  F319 A6 03        LDA A 3,X
1055  F31B B7 E5 68     STA A AF
1056  F31E A6 04        LDA A 4,X
1057  F320 B7 E5 69     STA A AF+1      ADRESSE FIN
1058  F323 FF E4 FD     STX   SVX5
1059  F326 BD EC 68     JSR   DETIQ     ETIQUETTE
1060  F329 BD F1 14     JSR   EZK       ECRIRE PROGRAMME
1061  F32C 20 1C        BRA   RETB2     RETOUR BASIC
1062                    *
1063                    * CHARGEMENT PROGRAMME BASIC
1064                    *
1065  F32E BF E4 FF LPTAPE  STS   STACK
1066  F331 BD EC D5     JSR   INBAS
1067  F334 FF E4 FD     STX   SVX5
1068  F337 BD EC 68     JSR   DETIQ     ETIQUETTE
1069  F33A BD F1 7E     JSR   LZK       LIRE PROGRAMME
1070  F33D FE E4 FD     LDX   SVX5
1071  F340 B6 E5 68     LDA A AF
1072  F343 A7 03        STA A 3,X
1073  F345 B6 E5 69     LDA A AF+1
1074  F348 A7 04        STA A 4,X       ADRESSE FIN
1075  F34A 7F E4 15 RETB2   CLR   EFLAG
1076  F34D FE E4 FD     LDX   SVX5
1077  F350 39           RTS             RETOUR BASIC
1078                    *
1079                    * ECRITURE DONNEES BASIC
1080                    *
1081  F351 BF E4 FF PDTAPE  STS   STACK
1082  F354 FF E4 FD     STX   SVX5
1083  F357 BD EC D5     JSR   INBAS
1084  F35A 81 0A        CMP A £$A
1085  F35C 27 EC        BEQ   RETB2
1086  F35E 7D E4 14     TST   BFLAG
1087  F361 26 09        BNE   PDT1
1088  F363 CE E5 14     LDX   £BUF
1089  F366 FF E5 12     STX   PTBUF
```

```
RBASK7                                  12-9-81  TSC ASSEMBLER  PAGE   25

 1090  F369 73 E4 14          COM    BFLAG
 1091  F36C FE E5 12  PDT1    LDX    PTBUF
 1092  F36F A7 00             STA A  0,X
 1093  F371 08                INX
 1094  F372 8C E5 64          CPX    £ENDBF
 1095  F375 27 09             BEQ    PDT2
 1096  F377 81 0D             CMP A  £$D
 1097  F379 27 05             BEQ    PDT2
 1098  F37B FF E5 12          STX    PTBUF
 1099  F37E 20 CA     RETB3   BRA    RETB2
 1100  F380 09        PDT2    DEX
 1101  F381 FF E5 6B          STX    AF
 1102  F384 CE 30 30          LDX    £$3030
 1103  F387 FF E5 64          STX    ETIQ
 1104  F38A CE E5 14          LDX    £BUF
 1105  F38D FF E5 66          STX    AD
 1106  F390 73 E4 19          COM    MFLG
 1107  F393 36                PSH A
 1108  F394 BD F1 14          JSR    EZK
 1109  F397 32                PUL A
 1110  F398 7F E4 19          CLR    MFLG
 1111  F39B 7F E4 14          CLR    BFLAG
 1112  F39E 20 DE             BRA    RETB3
 1113                  *
 1114                  * LECTURE DONNEES BASIC
 1115                  *
 1116  F3A0 BF E4 FF  RDTAPE  STS    STACK
 1117  F3A3 FF E4 FD          STX    SVX5
 1118  F3A6 BD EC D5          JSR    INBAS
 1119  F3A9 7D E4 14          TST    BFLAG
 1120  F3AC 26 18             BNE    RDT1
 1121  F3AE CE 30 30          LDX    £$3030
 1122  F3B1 FF E5 64          STX    ETIQ
 1123  F3B4 CE E5 14          LDX    £BUF
 1124  F3B7 FF E5 12          STX    PTBUF
 1125  F3BA 73 E4 19          COM    MFLG
 1126  F3BD BD F1 7E          JSR    LZK
 1127  F3C0 7F E4 19          CLR    MFLG
 1128  F3C3 73 E4 14          COM    BFLAG
 1129  F3C6 FE E5 12  RDT1    LDX    PTBUF
 1130  F3C9 A6 00             LDA A  0,X
 1131  F3CB 08                INX
 1132  F3CC 8C E5 64          CPX    £ENDBF
 1133  F3CF 27 09             BEQ    RDT2
 1134  F3D1 81 0D             CMP A  £$D
 1135  F3D3 27 05             BEQ    RDT2
 1136  F3D5 FF E5 12          STX    PTBUF
 1137  F3D8 20 A4             BRA    RETB3
 1138  F3DA 7F E4 14  RDT2    CLR    BFLAG
 1139  F3DD 20 9F             BRA    RETB3
 1140                         END

NO ERROR(S) DETECTED
```

RBASK7 12-9-81 TSC ASSEMBLER PAGE 27

SYMBOL TABLE:

ACKNAK	E50F	AD	E566	ADJADL	EED3	ADRESD	E566	ADRESF	E568
AF	E568	AOK	ED61	ATTAQR	EDE5	ATTEN1	F1A4	ATTSTR	F029
ATTZRO	EFE3	B1	E505	B2	E50A	B3	E50E	BAK	E50F
BAL	EE21	BAL3	F119	BFLAG	E414	BITSUI	F05A	BONNLE	F211
BUF	E514	CALCLP	EEB2	CHAIN	FFF2	CLRCA2	F02E	COMETR	F116
COMLR	F181	COMLR1	F195	COMPTA	E56B	COX	EEC1	CRECU	E504
CRLF	FFF5	CSOMME	F106	CSUM	E56A	D	0000	DAD	EC4E
DAF	EC5B	DDRB	0002	DEBC	EE51	DELAI	EEAE	DELAIR	F06A
DELAY	F085	DELSAU	EF4E	DETIQ	EC68	DETIQX	EC88	E1MOT	EE47
EAK	EFAB	EB	EE16	EB1	ED7C	EB2	EDA7	EB3	EDD1
EBSUIV	ED8A	ECEK7	F131	ECRBI1	F08B	ECRBIT	F08A	ECRICS	F158
ECRMOT	F0B4	EFLAG	E415	EM1P	EE85	EMBIT	EE6F	EMEZRO	EE82
EMIS1	EE76	EMOTS	EE35	ENCORE	F086	ENDBF	E564	ENQ	E509
ENVACK	EF75	ENVAQR	EF9C	ENVNAK	EF79	ENVV2	EF7B	EOT	E50E
ERR3	EFD5	ERRB	E511	ERREUR	F27B	ERRS	F296	ETIQ	E564
ETIQLU	E56D	ETQL1	F1B5	EZ	ED4D	EZK	F114	EZKRT	F177
INAD	FD57	INBAS	ECD5	INCH	FFEF	INFO	F1D1	INFOSX	F1D4
INIPR	F10E	INSAK	EFCE	INSAKR	EE10	IRB	0000	KMON	EC0F
L	E506	LBX	F0F1	LIR1MO	F1EB	LIRBIT	F0C1	LIRCS	F1FB
LIRMOT	F0F4	LPB2	E50B	LPC	E503	LPTAPE	F32E	LRCC	E502
LULEUN	F0EC	LZK	F17E	LZKRT	F22B	LZKS	F21F	MAD	F236
MAF	F23F	MANOR	F25B	MAX1	EECD	MECR	F26F	MERRU	F251
METIQ	F248	MFIN	F267	MFLG	E419	MLEC	F275	MLERT	F22C
MON	EC4B	MONIT	FFE6	NESSAI	E501	NPER	0006	ORB	0000
ORGRAM	E4F5	ORGROM	EC00	OUTCH	FFEC	OUTTIM	F0C5	PAZERO	F0AE
PCR	000C	PDATA	FFF2	PDT1	F36C	PDT2	F380	PDTAPE	F351
PTBUF	E512	R1BIT	F06E	R1MOT	F010	RB	EFDA	RB1	EEFA
RB2	EF22	RB3	EF09	RBSUIV	EF19	RCC	F014	RCLF	FFF5
RDM1	ED34	RDM2	ED48	RDMOD	ED18	RDT1	F3C6	RDT2	F3DA
RDTAPE	F3A0	RETB	ECCE	RETB1	ED04	RETB2	F34A	RETB3	F37E
RETX	EF6E	RI1	EF16	RI2	EF48	RMOTS	F001	RNAK	EE03
RPMOD	ECAC	RSUIB2	EF53	RZ	EEF6	SDM1	ECF2	SDM2	ED06
SDMOD	ECDC	SHIFTC	EEA3	SIGNAL	F178	SDH	E50A	SORBIT	F0AC
SPMOD	EC8B	SPTAPE	F309	SQ1	E508	SQ2	E50C	SQ3	E510
STABQ	EFAE	STACK	E4FF	STLPC	EECF	STX	E50D	SVX1	E4F5
SVX2	E4F7	SVX3	E4F9	SVX4	E4FB	SVX5	E4FD	SYN	E505
TB00	EFE4	TB01	EFF6	TEST1	F0DB	TIM2	F0C8	TIM3	F100
TOUJOU	F0F7	UNMOT	F148	VIAT	E860	VOIRE	EC33	VOIRL	EC42
VOIRR	EC28	X01	0032	X02	001A	X0B	001B	X0H	001C
X1AR	008D	X1E	0171	X3	00EA	X3AR	00E2	XSTOP	0014
ZERMO1	F19A	ZERMOT	F186						

- LOGICIEL DE COMMUNICATION DU MICRO-ORDINATEUR GOUPIL -

I - INTRODUCTION :

Le micro-ordinateur GOUPIL permet de transmettre de l'information par réseau commuté.

Deux utilisateurs (GA et GB) peuvent dialoguer en posant leurs combinés dans les oreillettes de leurs appareils.

Pour permettre la communication, un ensemble d'outils logiciels est mis à la disposition de l'utilisateur. Ces outils sont axés autour d'une procédure de communication classique : le BSC (Binary Synchronous Communication) défini par IBM.

La suite de ce document présente les principes du logiciel de communication et de la méthode d'utilisation.

II - LE COUPLEUR ACOUSTIQUE

Le principe du coupleur acoustique du GOUPIL repose sur une modulation "temporelle" (qui n'est ni de la modulation de fréquence, ni de la modulation d'amplitude auxquelles font référence les AVIS CCITT).

Ce principe utilise une seule fréquence (2000 Hz) :

- La fréquence présente représente un 0.
- La fréquence absente représente un 1.

Par ailleurs, un signal d'interruption "BREAK" a été prévu en voie de retour. Il utilise une fréquence différente (1180 hz) et permet d'arrêter l'émission ou la réception du correspondant.

Le découpage dans le temps et la reconnaissance de ces trois signaux (0, 1 et BREAK) sont réalisés par programme. Ce programme étant inclus dans le logiciel de communication.

Ces émissions étant dans la bande passante de la parole, aucune perturbation n'intervient sur le réseau.

III - LE LOGICIEL DE TRANSMISSION

Le protocole de transmission développé sur le micro-ordinateur GOUPIL a pour but d'assurer des échanges corrects entre un émetteur et un récepteur.

Il doit résoudre toutes les situations anormales afin de rendre compréhensible au récepteur l'information binaire transmise par l'émetteur.

Une liaison établie à l'aide de deux GOUPILS au travers du réseau commuté est une liaison POINT A POINT.

Les fonctions d'un protocole pour une liaison POINT A POINT sont :

1) d'effectuer le transfert des informations utiles de E vers R
2) de protéger contre les erreurs de transmission
3) d'assurer les reprises en cas d'anomalies, en ajoutant des informations de service.

III-1) Transfert de l'information utile

Pour permettre la transmission et assurer au récepteur la compréhension de l'information utile, la fonction de transfert de l'information utile nécessite :

- la structuration de cette information utile.
- l'encadrement par des délimiteurs pour marquer le début et la fin d'un bloc et pour indiquer la nature de l'information.
- l'identification des blocs.

III-1/1) Structuration de l'information

L'unité de transmission est l'octetet c'est une suite de bits qui est transmise. Le coupleur acoustique utilise une transmission en série asynchrone où chaque octet possède son start (signal permettant au récepteur de se synchroniser) afin d'être restitué correctement par le récepteur.

III-1/2) Encadrement d'un bloc

Pour reconnaitre le début et la fin de chaque bloc on utilise des séquences spécifiques.

Le début de chaque bloc est précédé par des bits de synchronisation.

Viennent ensuite les caractères de commandes

SOH - Start of heading : début d'en-tête

Le numéro de chronologie : cette numérotation permet de contrôler la chronologie des échanges et de détecter les erreurs ou les pertes de messages.

La longueur du texte : longueur du texte transmis (nombre d'octets)

STX -(Start of text): début du texte

Puis les caractères définissant la fin du bloc :

ETX -(End of text): fin du texte

LRC Longitudinal Redundancy CHECK : Il s'agit d'information permettant de détecter les erreurs de transmission. On a opté dans le protocole pour un contrôle de parité.

SYN	SOH	NUMERO DE CHRONOLOGIE	LONGUEUR DU TEXTE	STX	TEXTE	ETX	LRC

III-1/3) Identification des blocs

Cette identification est réalisée de deux manières différentes :

1) Dans les blocs d'informations utiles par le numéro de chronologie.
2) Dans les blocs de commande par le contenu du bloc.

III-2) Protection contre les erreurs de transmission

Pour chaque bloc de données on ajoute à chaque colonne d'éléments binaires, un élément binaire supplémentaire appelé LRC (longitudinal redundancy check). Cet élément prend la valeur binaire 0 ou 1 pour que le nombre total des 1 de chaque colonne soit impair.

Exemple :

Caractère 1	0	1	1	0	0	1	1	1
Caractère 2	1	1	1	0	0	0	0	0
Caractère 3	1	0	0	1	0	1	1	1
Caractère 4	1	0	0	0	0	0	0	0
Caractère 5	0	1	1	0	0	1	0	0
	0	0	0	0	1	0	1	1

LRC

III-2/1) Détection des erreurs

a) A l'émission le protocole génère le LRC à chaque caractère A la fin de chaque bloc transmis, le LRC total est envoyé vers le correspondant.

b) A la réception, le protocole génère également le LRC. A la fin de chaque bloc il compare le LRC généré en réception et celui généré à l'émission et transmis en fin de bloc d'informations utiles :

- s'il y a égalité un ACK (transmission correcte) est envoyé
- s'il n'y a pas égalité il y a eu une erreur dans la transmission et un NAK (transmission incorrecte) est envoyé vers l'émetteur.

III-3) Reprises en cas d'anomalies

A la réception d'un NAK le protocole reprend la transmission du bloc défectueux ou perdu (voir schéma Annexe 1).

La retransmission d'un bloc peut être reprise 4 fois au maximum. Si la réception du bloc est toujours incorrecte, un message est envoyé à l'utilisateur et un "break" est généré sur la ligne.

IV - DIFFERENTES PHASES DU LOGICIEL DE TRANSMISSION (voir schéma annexe 1)

La mise en oeuvre de la procédure de communication se fait en quatre phases :

a) Etablissement de la liaison
b) Transfert des données
c) Reprises éventuelles en cas d'anomalies
e) Libération de la communication

IV-1) Phase 1 : établissement de la liaison

Après l'établissement de la liaison téléphonique par les deux usagers et la fixation verbale du statut de chacun des GOUPILS (MAITRE - ESCLAVE) l'initiative de la transmission est attribuée au MAITRE.

Le maitre émet le message d'établissement de communication, en fait sous forme du bloc suivant :

SYN	CØDE FØNCTIØN	LØNG	ENQ

Ce bloc est appelé bloc de commande et est généré par le protocole sans intervention de l'utilisateur ; il a pour but d'établir la communication et de transmettre des informations de service.

code fonction : 1 envoi d'un programme BASIC
2 envoi d'une zone mémoire.

long. : Nombre d'octets à transmettre.

ENQ : demande de communication.

Une fois ce message transmis, le GOUPIL Maitre attend une réponse l'autorisant ou non à continuer le traitement (Voir structure message d'accusé de réception-Annexe 2).

IV-2) Phase 2 : transmission des données utiles

L'accusé de réception positif autorisant le GOUPIL Maitre à continuer la transmission étant arrivé, les blocs d'informations utiles sont transmis séquentiellement.

A chaque bloc transmis le GOUPIL Maitre attend un accusé de réception positif (ACK) pour continuer sa transmission.

Si un accusé de réception négatif (NA K) arrive pour le bloc précédemment transmis on réemet ce bloc.

IV-3) Phase 3 : Libération de la communication

Elle est à l'initiative du GOUPIL Maitre qui envoi le bloc de commande suivant :

SYN	EØT

EØT : end of transmission
fin de transmission

IV-4) Phase 4 : Reprise en cas d'anomalie

Elle gère : - la perte de bloc
- incohérence entre le LRC et le texte transmis.

V - COMMANDES POUR L'UTILISATEUR AU NIVEAU MONITEUR

Pour transférer des informations sous moniteur par coupleur acoustique il faut utiliser la commande :

K

Après avoir échangé avec votre correspondant au téléphone :

1) le rôle de chacun des GOUPILS (Emetteur ou Récepteur)

2) le type d'informations (programme ou données) et l'emplacement de la mémoire où il faut le stocker

chacun place son combiné dans les oreillettes et répond aux questions suivantes :

LERT ? X { X = T pour transmettre
          { X = R pour recevoir

2 cas se présentent :

1) la réponse est T :

Il faut répondre aux deux questions

DEBUT ? XXXX

FIN ? XXXX

2) la réponse est R :

Il faut répondre à la question

DEBUT ? XXXX

Après avoir répondu à ces questions, le protocole gère les échanges.

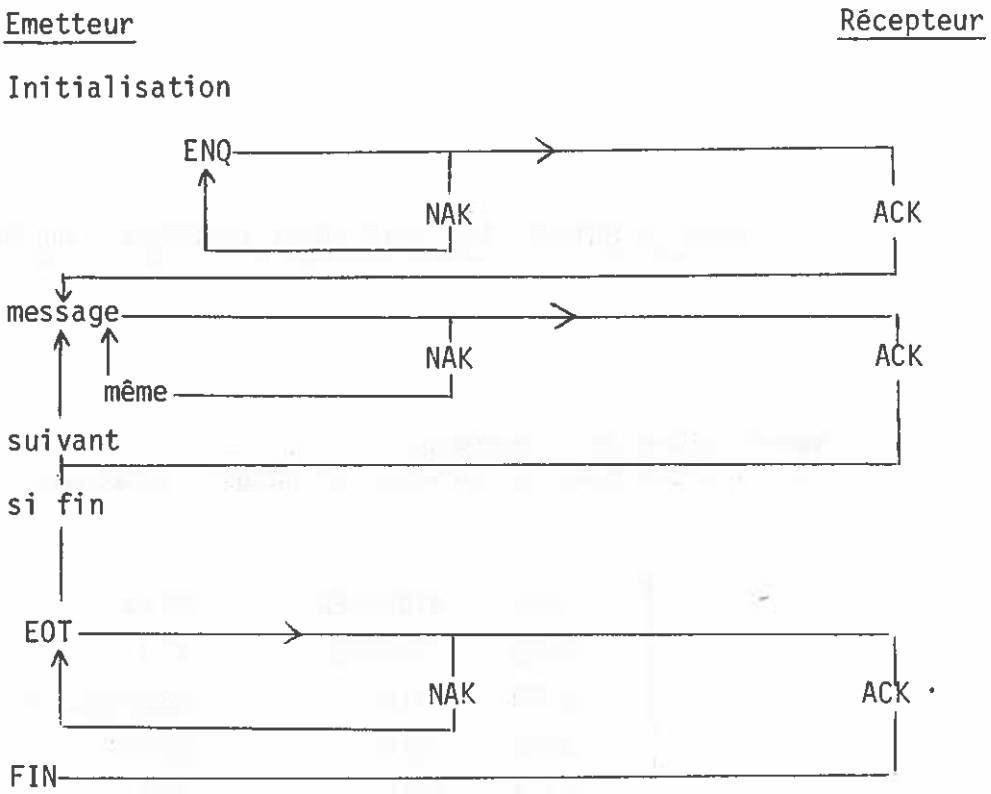
Si la transmission totale est correcte le message suivant apparait sur les deux écrans :

TRANSMISSION CORRECTE

Si la transmission totale est incorrecte le message suivant apparait sur les deux écrans :

TRANSMISSIØN INCØRRECTE
(Il faut reprendre la procédure)

Annexe 1 : Diagramme d'une liaison



Annexe 2 : Message d'accusé de réception

ACK accusé de réception positif (ACKNOWLEDGE)

Ce message indique que le bloc précédemment reçu était correct

SYN	N° CHRONOLOGIE	ACK

NACK accusé de réception négatif (NEGATIVE ACKNOWLEDGE)

Ce message indique que le bloc précédemment reçu était incorrect

SYN	N° CHRONOLOGIE	NAK

- VII -

- FICHES TECHNIQUES DES PRINCIPAUX COMPOSANTS DE GOUPIL -

L S I	- 6551	SYNERTEC	ACIA
	- 6522	SYNERTEC	VIA
	- 8279	INTEL	encodeur de clavier
	- 2708	INTEL	EPROM
	- 2114	INTEL	RAM
	- 4116	MOSTEK	RAM
	- 2716	NS	EPROM

TTL Standards	- 74 LS 138
	- 74 LS 139
	- 74 LS 367
	- 74 LS 244
	- 74 LS 273

L S I	- 6845	Contrôleur de CRT
	- 96364	Contrôleur de CRT
	- 6850	ACIA
	- 6808	Microprocesseur

Pour les autres composants, se reporter aux TTL data books.

MOSTEK

16,384 X 1-BIT DYNAMIC RAM

MK4116(P/N)-2/3

FEATURES

- Recognized industry standard 16-pin configuration from MOSTEK
- 150ns access time, 375ns cycle (MK 4116-2) 200ns access time, 375ns cycle (MK 4116-3)
- ± 10% tolerance on all power supplies (+12V, ±5V)
- Low power: 462mW active, 20mW standby (max)
- Output data controlled by $\overline{CAS}$ and unlatched at end of cycle to allow two dimensional chip selection and extended page boundary
- Common I/O capability using "early write" operation
- Read-Modify-Write, $\overline{RAS}$-only refresh, and Page-mode capability
- All inputs TTL compatible,low capacitance, and protected against static charge
- 128 refresh cycles
- ECL compatible on VBB power supply (-5.7V)

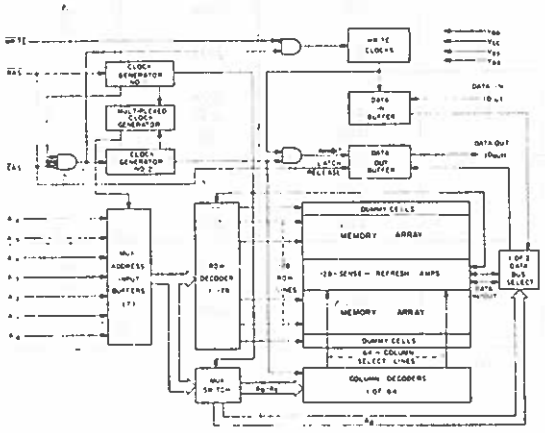
DESCRIPTION

The MK 4116 is a new generation MOS dynamic random access memory circuit organized as 16,384 words by 1 bit. As a state-of-the-art MOS memory device, the MK 4116 (16K RAM) incorporates advanced circuit techniques designed to provide wide operating margins, both internally and to the system user, while achieving performance levels in speed and power previously seen only in MOSTEK's high performance MK 4027 (4K RAM).

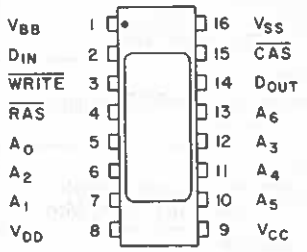
The technology used to fabricate the MK 4116 is MOSTEK's double-poly, N-channel silicon gate, POLY II™ process. This process, coupled with the use of a single transistor dynamic storage cell, provides the maximum possible circuit density and reliability, while maintaining high performance capability. The use of dynamic circuitry throughout, including sense amplifiers, assures that power dissipation is minimized without any sacrifice in speed or operating margin. These factors combine to make the MK 4116 a truly superior RAM product.

Multiplexed address inputs (a feature pioneered by MOSTEK for its 4K RAMS) permits the MK 4116 to be packaged in a standard 16-pin DIP. This recognized industry standard package configuration, while compatible with widely available automated testing and insertion equipment, provides highest possible system bit densities and simplifies system upgrade from 4K to 16K RAMs for new generation applications. Non-critical clock timing requirements allow use of the multiplexing technique while maintaining high performance.

FUNCTIONAL DIAGRAM



PIN CONNECTIONS



PIN NAMES

A_0 A_6	ADDRESS INPUTS
$\overline{CAS}$	COLUMN ADDRESS STROBE
D_{IN}	DATA IN
D_{OUT}	DATA OUT
$\overline{RAS}$	ROW ADDRESS STROBE
$\overline{WRITE}$	READ/WRITE INPUT
V_{BB}	POWER (-5V)
V_{CC}	POWER (+5V)
V_{DD}	POWER (+12V)
V_{SS}	GROUND

ABSOLUTE MAXIMUM RATINGS*

Voltage on any pin relative to V_{BB} ..... −0.5V to +20V
Voltage on V_{DD}, V_{CC} supplies relative to V_{SS} ..... −1.0V to +15.0V
V_{BB}−V_{SS} (V_{DD}−V_{SS}>0V) 0V
Operating temperature, T_A (Ambient) ..... 0°C to + 70°C
Storage temperature (Ambient) Ceramic ..... −55°C to + 150°C
Storage temperature, (Ambient) Plastic ..... -55°C to +125°C
Short circuit output current 50mA
Power dissipation ..... 1 Watt

*Stresses greater than those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.

RECOMMENDED DC OPERATING CONDITIONS[6]

(0°C ≤ T_A ≤ 70°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V_{DD} V_{CC} V_{SS} V_{BB}	10.8 4.5 0 −4.5	12.0 5.0 0 −5.0	13.2 5.5 0 −5.7	Volts Volts Volts Volts	2 2,3 2 2
Input High (Logic 1) Voltage, $\overline{RAS}$, $\overline{CAS}$, $\overline{WRITE}$	V_{IHC}	2.4	–	7.0	Volts	2
Input High (Logic 1) Voltage, all inputs except $\overline{RAS}$, $\overline{CAS}$ $\overline{WRITE}$	V_{IH}	2.2	–	7.0	Volts	2
Input Low (Logic 0) Voltage, all inputs	V_{IL}	−1.0	–	.8	Volts	2

DC ELECTRICAL CHARACTERISTICS

(0°C ≤ TA ≤ 70°C) (VDD = 12.0V ± 10%; VCC = 5.0V ±10%; -5.7V ≤ VBB ≤ -4.5V; VSS = 0V)

PARAMETER	SYMBOL	MIN	MAX	UNITS	NOTES
OPERATING CURRENT Average power supply operating current ($\overline{RAS}$, $\overline{CAS}$ cycling; t_{RC} = t_{RC} Min	I_{DD1} I_{CC1} I_{BB1}		35 200	mA µA	4 5
STANDBY CURRENT Power supply standby current ($\overline{RAS}$ = V_{IHC}, D_{OUT} = High Impedance)	I_{DD2} I_{CC2} I_{BB2}	 −10 	1.5 10 100	mA µA µA	
REFRESH CURRENT Average power supply current, refresh mode ($\overline{RAS}$ cycling, $\overline{CAS}$ = V_{IHC}; t_{RC} = t_{RC} Min	I_{DD3} I_{CC3} I_{BB3}	 −10 	25 10 200	mA µA µA	4
PAGE MODE CURRENT Average power supply current, page-mode operation ($\overline{RAS}$ = V_{IL}, $\overline{CAS}$ cycling; t_{PC} = t_{PC} Min	I_{DD4} I_{CC4} I_{BB4}		27 200	mA µA	4 5
INPUT LEAKAGE Input leakage current, any input (V_{BB} = −5V, 0V ≤ V_{IN} ≤ +7.0V, all other pins not under test = 0 volts)	$I_{I(L)}$	−10	10	µA	
OUTPUT LEAKAGE Output leakage current (D_{OUT} is disabled, 0V ≤ V_{OUT} ≤ +5.5V)	$I_{O(L)}$	−10	10	µA	
OUTPUT LEVELS Output high (Logic 1) voltage (I_{OUT} = −5mA) Output low (Logic 0) voltage (I_{OUT} = 4.2 mA)	V_{OH} V_{OL}	2.4 	 0.4	Volts Volts	3

NOTES:

1. T_A is specified here for operation at frequencies to $t_{RC} \geq t_{RC}$ (min). Operation at higher cycle rates with reduced ambient temperatures and higher power dissipation is permissible, however, provided AC operating parameters are met. See figure 1 for derating curve.
2. All voltages referenced to V_{SS}.
3. Output voltage will swing from V_{SS} to V_{CC} when activated with no current loading. For purposes of maintaining data in standby mode, V_{CC} may be reduced to V_{SS} without affecting refresh operations or data retention. However, the V_{OH} (min) specification is not guaranteed in this mode.
4. I_{DD1}, I_{DD3}, and I_{DD4} depend on cycle rate. See figures 2,3, and 4 for I_{DD} limits at other cycle rates.
5. I_{CC1} and I_{CC4} depend upon output loading. During readout of high level data V_{CC} is connected through a low impedance (135Ω typ) to data out. At all other times I_{CC} consists of leakage currents only.

ELECTRICAL CHARACTERISTICS AND RECOMMENDED AC OPERATING CONDITIONS (6,7,8)

($0°C \leq T_A \leq 70°C$)[1] (V_{DD} = 12.0V ± 10%; V_{CC} = 5.0V ±10%, V_{SS} = 0V, V_{BB} = **-5.7V ≤ VBB ≤ -4.5V**)

PARAMETER	SYMBOL	MK 4116-2		MK 4116-3		UNITS	NOTES
		MIN	MAX	MIN	MAX		
Random read or write cycle time	t_{RC}	375		375		ns	9
Read-write cycle time	t_{RWC}	375		375		ns	9
Read modify write cycle time	t_{RMW}	320		405		ns	9
Page mode cycle time	t_{PC}	170		225		ns	9
Access time from $\overline{RAS}$	t_{RAC}		150		200	ns	10,12
Access time from $\overline{CAS}$	t_{CAC}		100		135	ns	11,12
Output buffer turn-off delay	t_{OFF}	0	40	0	50	ns	13
Transition time (rise and fall)	t_T	3	35	3	50	ns	8
$\overline{RAS}$ precharge time	t_{RP}	100		120		ns	
$\overline{RAS}$ pulse width	t_{RAS}	150	10,000	200	10,000	ns	
$\overline{RAS}$ hold time	t_{RSH}	100		135		ns	
$\overline{CAS}$ hold time	t_{CSH}	150		200		ns	
$\overline{CAS}$ pulse width	t_{CAS}	100	10,000	135	10,000	ns	
$\overline{RAS}$ to $\overline{CAS}$ delay time	t_{RCD}	20	50	25	65	ns	14
$\overline{CAS}$ to $\overline{RAS}$ precharge time	t_{CRP}	−20		−20		ns	
Row Address set-up time	t_{ASR}	0		0		ns	
Row Address hold time	t_{RAH}	20		25		ns	
Column Address set-up time	t_{ASC}	−10		−10		ns	
Column Address hold time	t_{CAH}	45		55		ns	
Column Address hold time referenced to $\overline{RAS}$	t_{AR}	95		120		ns	
Read command set-up time	t_{RCS}	0		0		ns	
Read command hold time	t_{RCH}	0		0		ns	
Write command hold time	t_{WCH}	45		55		ns	
Write command hold time referenced to $\overline{RAS}$	t_{WCR}	95		120		ns	
Write command pulse width	t_{WP}	45		55		ns	
Write command to $\overline{RAS}$ lead time	t_{RWL}	50		70		ns	
Write command to $\overline{CAS}$ lead time	t_{CWL}	50		70		ns	
Data-in set-up time	t_{DS}	0		0		ns	15
Data-in hold time	t_{DH}	45		55		ns	15
Data-in hold time referenced to $\overline{RAS}$	t_{DHR}	95		120		ns	
$\overline{CAS}$ precharge time (for page-mode cycle only)	t_{CP}	60		80		ns	
Refresh period	t_{REF}		2		2	ms	
$\overline{WRITE}$ command set-up time	t_{WCS}	-20		-20		ns	16
$\overline{CAS}$ to $\overline{WRITE}$ delay	t_{CWD}	60		80		ns	16
$\overline{RAS}$ to $\overline{WRITE}$ delay	t_{RWD}	110		145		ns	16

NOTES (Continued)

6. Several cycles are required after power-up before proper device operation is achieved. Any 8 cycles which perform refresh are adequate for this purpose
7. AC measurements assume tT = 5ns.
8. VIHC (min) or VIH (min) and VIL (max) are reference levels for measuring timing of input signals. Also transition times are measured between VIHC or VIH and VIL.
9. The specifications for tRC (min) tRMW (min) and tRWC (min) are used only to indicate cycle time at which proper operation over the full temperature range ($0°C \leq T_A \leq 70°C$) is assured
10. Assumes that $t_{RCD} \leq t_{RCD}$ (Max). If tRCD is greater than the maximum recommended value shown in this table, tRAC will increase by the amount that tRCD exceeds the value shown
11. Assumes that tRCD (max)
12. Measured with a load equivalent to 2 TTL loads and 100pF.
13. tOFF (max) defines the time at which the output achieves the open circuit condition and is not referenced to output voltage levels
14. Operation within the tRCD (max) limit insures that tRAC (max) can be met. tRCD (max) is specified as a reference point only if tRCD is greater than the specified tRCD (max) limit, then access time is controlled exclusively by tCAC
15. These parameters are referenced to CAS leading edge in early write cycles and to WRITE leading edge in delayed write or read-modify-write cycles.
16. tWCS, tCWD and tRWD are restrictive operating parameters in read write and read modify write cycles only. If $t_{WCS} \geq t_{WCS}$ (min), the cycle is an early write cycle and the data out pin will remain open circuit (high impedance) throughout the entire cycle. If $t_{CWD} \geq t_{CWD}$ (min) and $t_{RWD} \geq t_{RWD}$ (min), the cycle is a read-write cycle and the data out will contain data read from the selected cell. If neither of the above sets of conditions is satisfied the condition of the data out (at access time) is indeterminate
17. Effective capacitance calculated from the equation $C = \frac{\Delta t}{\Delta V} \Delta V$ with ΔV = 3 volts and power supplies at nominal levels
18. CAS = VIHC to disable DOUT

AC ELECTRICAL CHARACTERISTICS

(0°C ≤ T_A ≤ 70°C) (V_{DD} = 12.0V ± 10%; V_{SS} = 0V; V_{BB} = **-5.7V ≤ VBB ≤ -4.5V**)

PARAMETER	SYMBOL	TYP	MAX	UNITS	NOTES
Input Capacitance (A_0–A_6), D_{IN}	C_{I1}	4	5	pF	17
Input Capacitance $\overline{RAS}$, $\overline{CAS}$, $\overline{WRITE}$	C_{I2}	8	10	pF	17
Output Capacitance (D_{OUT})	C_O	5	7	pF	17,18

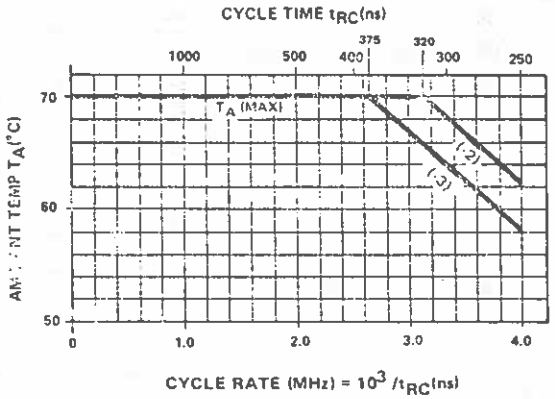


Fig. 1 Maximum ambient temperature versus cycle rate for extended frequency operation. T_A (max) for operation at cycling rates greater than 2.66 MHz (t_{CYC} ≤ 375ns) is determined by T_A (max)° C = 70–9.0 x (cycle rate MHz –2.66) for -3. T_A (max) °C = 70 – 9.0 x cycle rate MHz –3.125MHz) for -2 only.

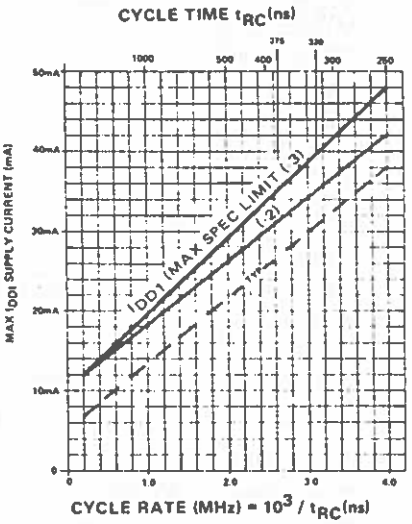


Fig. 2 Maximum I_{DD1} versus cycle rate for device operation at extended frequencies. I_{DD1} (max) curve is defined by the equation:

I_{DD1} (max) mA = 10 + 9.4 x cycle rate [MHz] for -3
I_{DD1} (max) mA = 10 + 8.0 x cycle rate [MHz] for -2

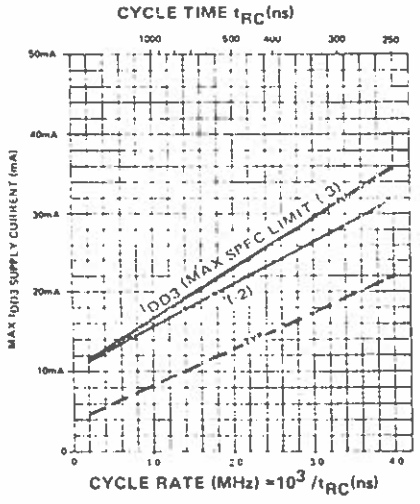


Fig. 3 Maximum I_{DD3} versus cycle rate for device operation at extended frequencies. I_{DD3} (max) curve is defined by the equation:

I_{DD3}(max) mA = 10 + 6.5 x cycle rate [MHz] for -3
I_{DD3}(max) mA = 10 + 5.5 x cycle rate [MHz] for -2

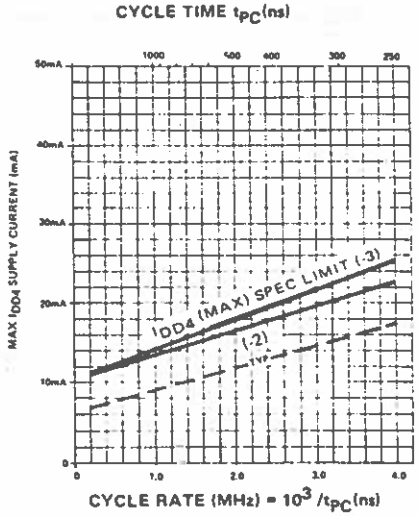
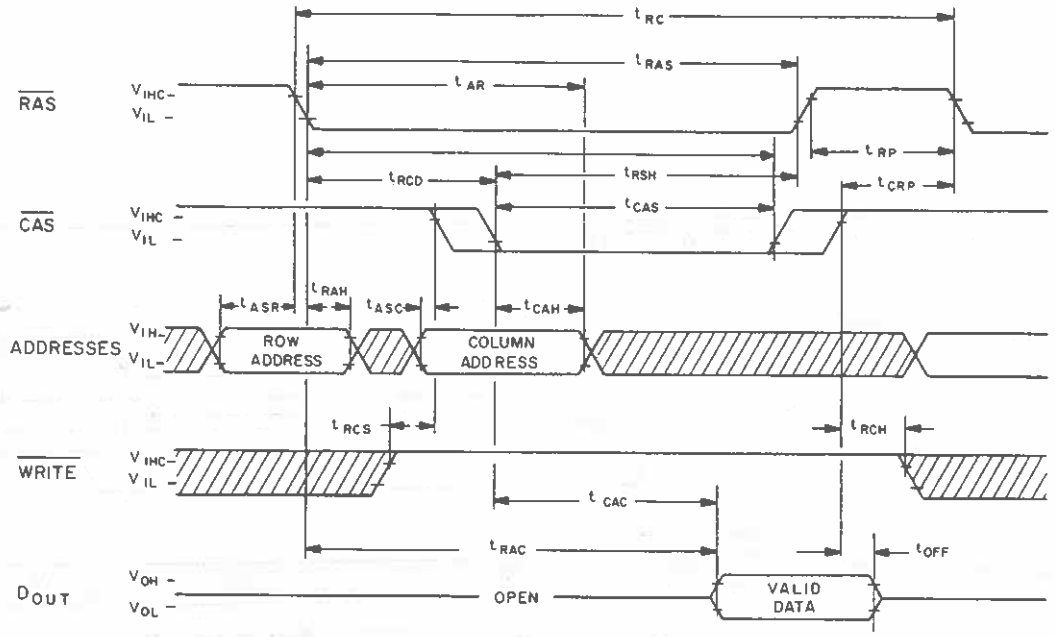


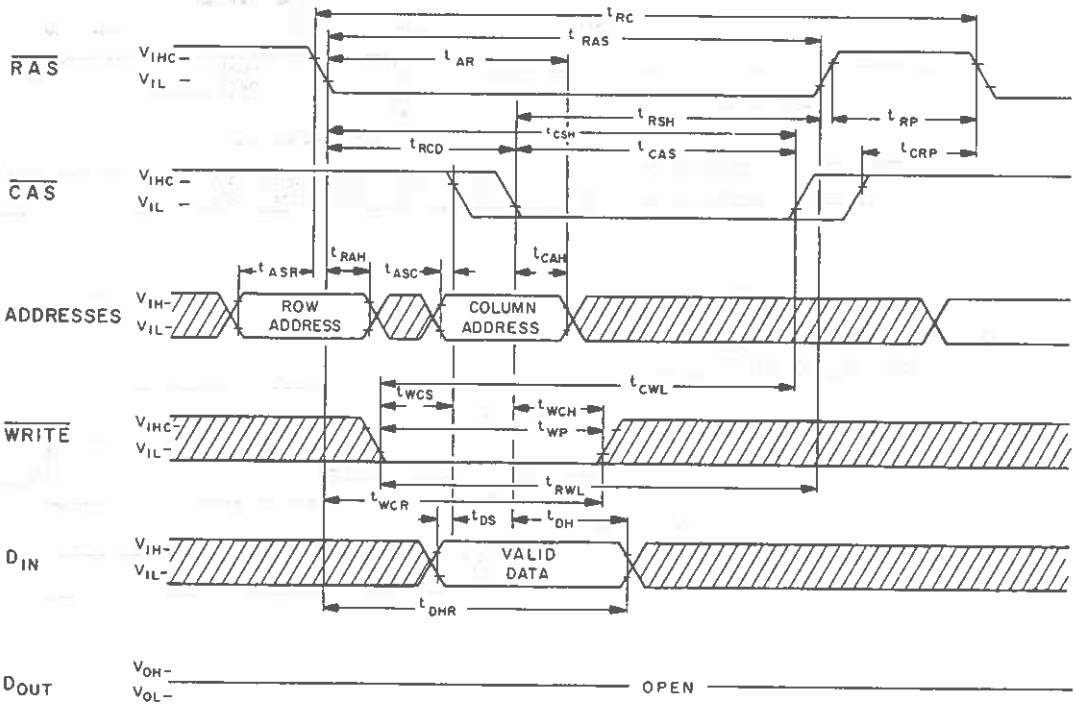
Fig. 4 Maximum I_{DD4} versus cycle rate for device operation in page mode. I_{DD4} (max) curve is defined by the equation:

I_{DD4} (max) mA = 10 + 3.75 x cycle rate [MHz] for -3
I_{DD4} (max) mA = 10 + 3.2 x cycle rate [MHz] for -2

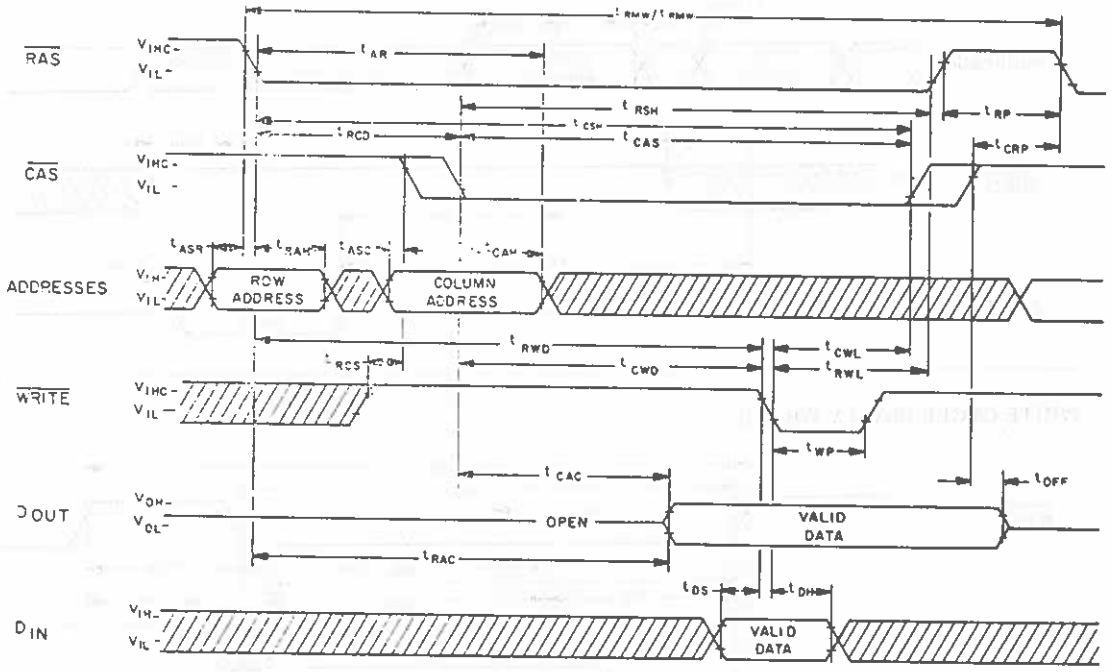
READ CYCLE



WRITE CYCLE (EARLY WRITE)

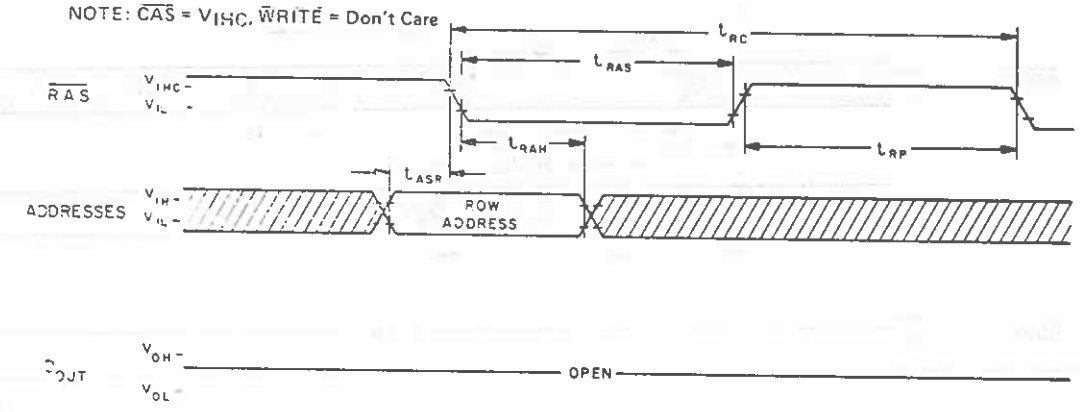


READ–WRITE/READ–MODIFY–WRITE CYCLE

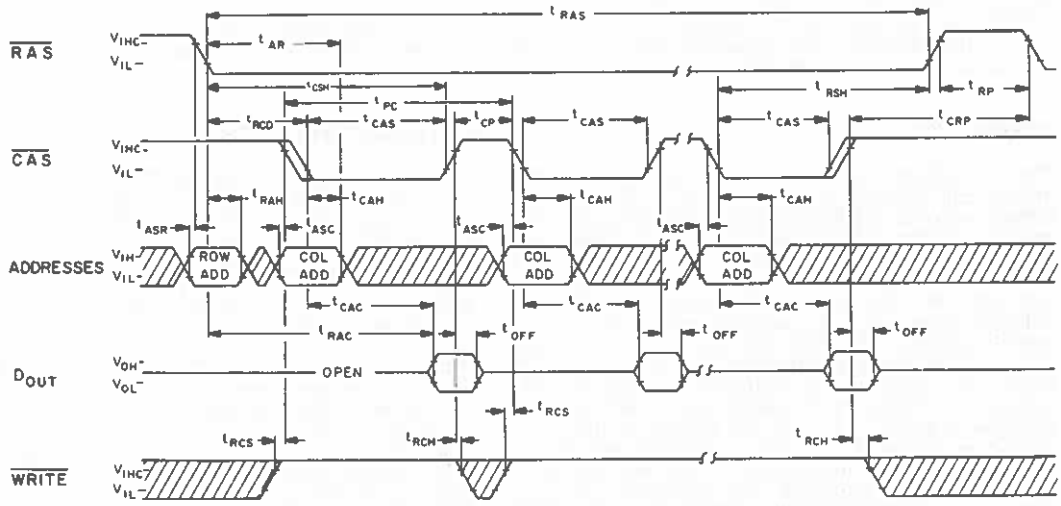


"RAS–ONLY" REFRESH CYCLE

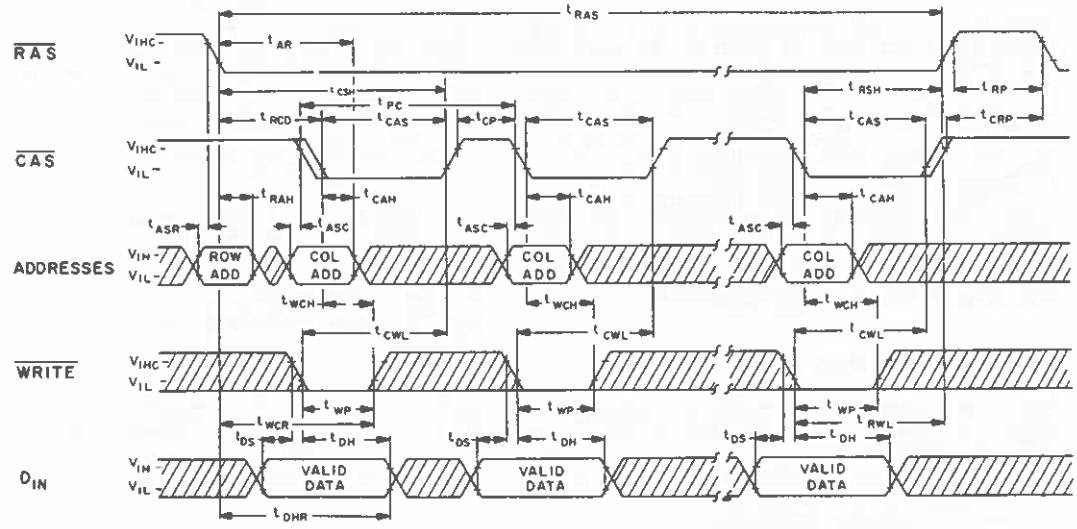
NOTE: $\overline{CAS}$ = V_{IHC}, $\overline{WRITE}$ = Don't Care



PAGE MODE READ CYCLE



PAGE MODE WRITE CYCLE



DESCRIPTION (continued)

System oriented features include ± 10% tolerance on all power supplies, direct interfacing capability with high performance logic families such as Schottky TTL, maximum input noise immunity to minimize "false triggering" of the inputs (a common cause of soft errors), on-chip address and data registers which eliminate the need for interface registers, and two chip select methods to allow the user to determine the appropriate speed/power characteristics of his memory system. The MK 4116 also incorporates several flexible timing/operating modes. In addition to the usual read, write, and read-modify-write cycles, the MK 4116 is capable of delayed write cycles, page-mode operation and RAS-only refresh. Proper control of the clock inputs(RAS, CAS and WRITE) allows common I/O capability, two dimensional chip selection, and extended page boundaries (when operating in page mode).

ADDRESSING

The 14 address bits required to decode 1 of the 16,384 cell locations within the MK 4116 are multiplexed onto the 7 address inputs and latched into the on-chip address latches by externally applying two negative going TTL-level clocks. The first clock, the Row Address Strobe (RAS), latches the 7 row address bits into the chip. The second clock, the Column Address Strobe (CAS), subsequently latches the 7 column address bits into the chip. Each of these signals, RAS and CAS, triggers a sequence of events which are controlled by different delayed internal clocks. The two clock chains are linked together logically in such a way that the address multiplexing operation is done outside of the critical path timing sequence for read data access. The later events in the CAS clock sequence are inhibited until the occurence of a delayed signal derived from the RAS clock chain. This "gated CAS" feature allows the CAS clock to be externally activated as soon as the Row Address Hold Time specification (t_{RAH}) has been satisfied and the address inputs have been changed from Row address to Column address information.

Note that CAS can be activated at any time after t_{RAH} and it will have no effect on the worst case data access time (t_{RAC}) up to the point in time when the delayed row clock no longer inhibits the remaining sequence of column clocks. Two timing endpoints result from the internal gating of CAS which are called t_{RCD} (min) and t_{RCD} (max). No data storage or reading errors will result if CAS is applied to the MK 4116 at a point in time beyond the t_{RCD} (max) limit. However, access time will then be determined exclusively by the access time from CAS (t_{CAC}) rather than from RAS (t_{RAC}), and access time from RAS will be lengthened by the amount that t_{RCD} exceeds the t_{RCD} (max) limit.

DATA INPUT/OUTPUT

Data to be written into a selected cell is latched into an on-chip register by a combination of WRITE and CAS while RAS is active. The later of the signals (WRITE or CAS) to make its negative transition is the strobe for the Data In (D_{IN}) register. This permits several options in the write cycle timing. In a write cycle, if the WRITE input is brought low (active) prior to CAS, the D_{IN} is strobed by CAS, and the set-up and hold times are referenced to CAS. If the input data is not available at CAS time or if it is desired that the cycle be a read-write cycle, the WRITE signal will be delayed until after CAS has made its negative transition. In this "delayed write cycle" the data input set up and hold times are referenced to the negative edge of WRITE rather than CAS. (To illustrate this feature, D_{IN} is referenced to WRITE in the timing diagrams depicting the read-write and page-mode write cycles while the "early write" cycle diagram shows D_{IN} referenced to CAS).

Data is retrieved from the memory in a read cycle by maintaining WRITE in the inactive or high state throughout the portion of the memory cycle in which CAS is active (low). Data read from the selected cell will be available at the output within the specified access time.

DATA OUTPUT CONTROL

The normal condition of the Data Output (D_{OUT}) of the MK 4116 is the high impedance (open-circuit) state. That is to say, anytime CAS is at a high level, the D_{OUT} pin will be floating. The only time the output will turn on and contain either a logic 0 or logic 1 is at access time during a read cycle. D_{OUT} will remain valid from access time until CAS is taken back to the inactive (high level) condition.

If the memory cycle in progress is a read, read-modify write, or a delayed write cycle, then the data output will go from the high impedance state to the active condition, and at access time will contain the data read from the selected cell. This output data is the same polarity (not inverted) as the input data. Once having gone active, the output will remain valid until CAS is taken to the precharge (logic 1) state, whether or not RAS goes into precharge.

If the cycle in progress is an "early-write" cycle (WRITE active before CAS goes active), then the output pin will maintain the high impedance state throughout the entire cycle. Note that with this type of output configuration, the user is given full control of the D_{OUT} pin simply by controlling the placement of WRITE command during a write cycle, and the pulse width of the Column Address Strobe during read operations. Note also that even though data is not latched at the output, data can remain valid from access time until the beginning of a subsequent cycle without paying any penalty in overall memory cycle time (stretching the cycle).

This type of output operation results in some very significant system implications.

Common I/O Operation — If all write operations are handled in the "early write" mode, then D_{IN} can be connected directly to D_{OUT} for a common I/O data bus.

Data Output Control — D_{OUT} will remain valid during a read cycle from t_{CAC} until CAS goes back to a high level (precharge), allowing data to be valid from one cycle up until a new memory cycle begins with no penalty in cycle time. This also makes the RAS/CAS clock timing relationship very flexible.

Two Methods of Chip Selection — Since D_{OUT}

is not latched, $\overline{CAS}$ is not required to turn off the outputs of unselected memory devices in a matrix. This means that both $\overline{CAS}$ and/or $\overline{RAS}$ can be decoded for chip selection. If both $\overline{RAS}$ and $\overline{CAS}$ are decoded, then a two dimensional (X,Y) chip select array can be realized.

Extended Page Boundary – Page-mode operation allows for successive memory cycles at multiple column locations of the same row address. By decoding $\overline{CAS}$ as a page cycle select signal, the page boundary can be extended beyond the 128 column locations in a single chip. (See page-mode operation).

OUTPUT INTERFACE CHARACTERISTICS

The three state data output buffer presents the data output pin with a low impedance to V_{CC} for a logic 1 and a low impedance to V_{SS} for a logic 0. The effective resistance to V_{CC} (logic 1 state) is 420 Ω maximum and 135Ω typically. The resistance to V_{SS} (logic 0 state) is 95 Ω maximum and 35 Ω typically. The separate V_{CC} pin allows the output buffer to be powered from the supply voltage of the logic to which the chip is interfaced. During battery standby operation, the V_{CC} pin may have power removed without affecting the MK 4116 refresh operation. This allows all system logic except the $\overline{RAS}$ timing circuitry and the refresh address logic to be turned off during battery standby to conserve power.

PAGE MODE OPERATION

The "Page Mode" feature of the MK 4116 allows for successive memory operations at multiple column locations of the same row address with increased speed without an increase in power. This is done by strobing the row address into the chip and maintaining the $\overline{RAS}$ signal at a logic 0 throughout all successive memory cycles in which the row address is common. This "page-mode" of operation will not dissipate the power associated with the negative going edge of $\overline{RAS}$. Also, the time required for strobing in a new row address is eliminated, thereby decreasing the access and cycle times.

The page boundary of a single MK 4116 is limited to the 128 column locations determined by all combinations of the 7 column address bits. However, in system applications which utilize more than 16,384 data words, (more than one 16K memory block), the page boundary can be extended by using $\overline{CAS}$ rather than $\overline{RAS}$ as the chip select signal. $\overline{RAS}$ is applied to all devices to latch the row address into each device and then $\overline{CAS}$ is decoded and serves as a page cycle select signal. Only those devices which receive both $\overline{RAS}$ and $\overline{CAS}$ signals will execute a read or write cycle.

REFRESH

Refresh of the dynamic cell matrix is accomplished by performing a memory cycle at each of the 128 row addresses within each 2 millisecond time interval. Although any normal memory cycle will perform the refresh operation, this function is most easily accomplished with "$\overline{RAS}$-only" cycles. $\overline{RAS}$-only refresh results in a substantial reduction in operating power. This reduction in power is reflected in the I_{DD3} specification.

POWER CONSIDERATIONS

Most of the circuitry used in the MK 4116 is dynamic and most of the power drawn is the result of an address strobe edge. Consequently, the dynamic power is primarily a function of operating frequency rather than active duty cycle (refer to the MK 4116 current waveforms in figure 5). This current characteristic of the MK 4116 precludes inadvertent burn out of the device in the event that the clock inputs become shorted to ground due to system malfunction.

Although no particular power supply noise restriction exists other than the supply voltages remain within the specified tolerance limits, adequate decoupling should be provided to suppress high frequency noise resulting from the transient current of the device. This insures optimum system performance and reliability. Bulk capacitance requirements are minimal since the MK 4116 draws very little steady state (DC) current.

In system applications requiring lower power dissipation, the operating frequency (cycle rate) of the MK 4116 can be reduced and the (guaranteed maximum) average power dissipation of the device will be lowered in accordance with the I_{DD1} (max) spec limit curve illustrated in figure 2. NOTE: The MK 4116 family is guaranteed to have a maximum IDD1 requirement of 35mA @ 375ns cycle (320ns cycle for the -2) with an ambient temperature range from 0° to 70°C. A lower operating frequency, for example 1 microsecond cycle, results in a reduced maximum Idd1 requirement of under 20mA with an ambient temperature range from 0° to 70°C.

It is possible the MK4116 family (-2 and 3 speed selections for 'example) at frequencies higher than specified, provided all AC operating parameters are met. Operation at shorter cycle times (<tRC min) results in higher power dissipation and, therefore, a reduction in ambient temperature is required. Refer to Figure 1 for derating curve.

NOTE Additional power supply tolerance has been included on the VBB supply to allow direct interface capability with both 5V systems 5 2V ECL systems

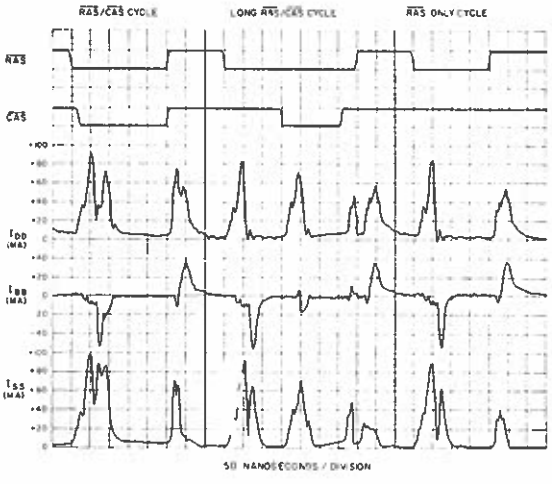


Fig. 5 Typical Current Waveforms

Although $\overline{RAS}$ and/or $\overline{CAS}$ can be decoded and used as a chip select signal for the MK 4116, overall system power is minimized if the Row Address Strobe ($\overline{RAS}$) is used for this purpose. All unselected devices (those which do not receive a $\overline{RAS}$) will remain in a low power (standby) mode regardless of the state of $\overline{CAS}$.

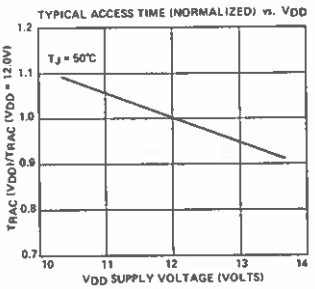
POWER UP

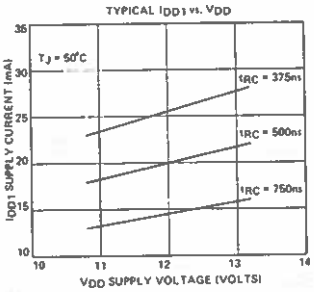
The MK 4116 requires no particular power supply sequencing so long as the Absolute Maximum Rating Conditions are observed. However, in order to insure compliance with the Absolute Maximum Ratings, MOSTEK recommends sequencing of power supplies such that V_{BB} is applied first and removed last. V_{BB} should never be more positive than V_{SS} when power is applied to V_{DD}.

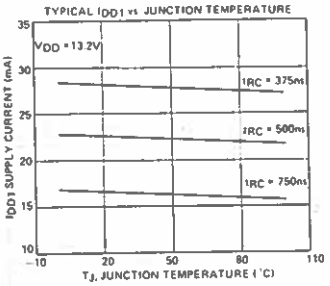
Under system failure conditions in which one or more supplies exceed the specified limits significant additional margin against catastrophic device failure may be achieved by forcing $\overline{RAS}$ and $\overline{CAS}$ to the inactive state (high level).

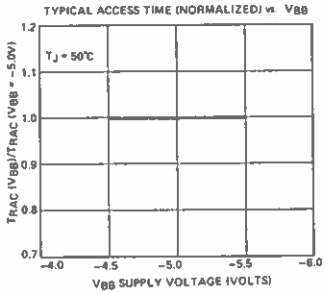
After power is applied to the device, the MK 4116 requires several cycles before proper device operation is achieved. Any 8 cycles which perform refresh are adequate for this purpose.

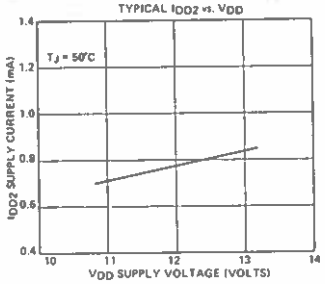
TYPICAL CHARACTERISTICS

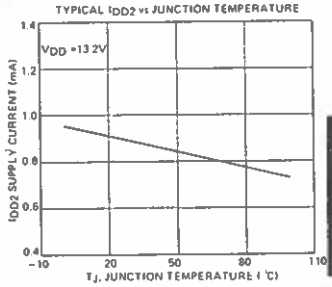


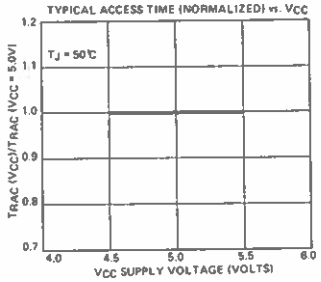


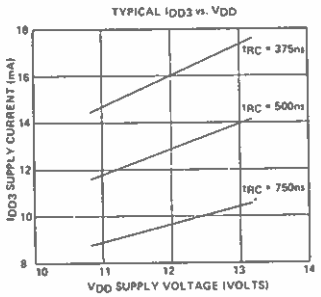


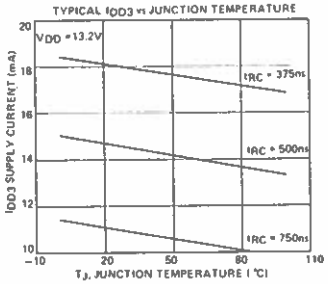


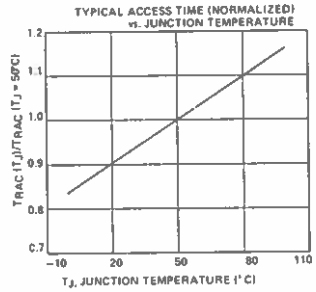


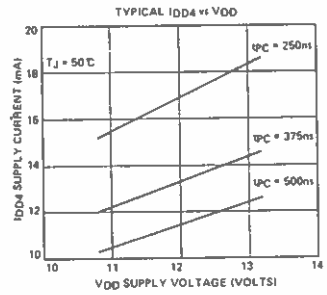


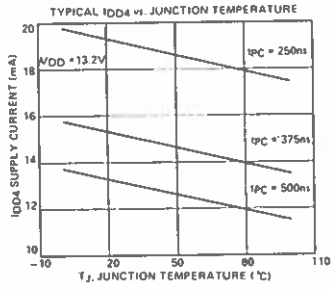


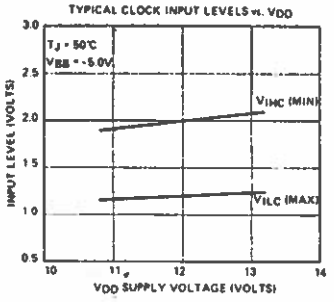


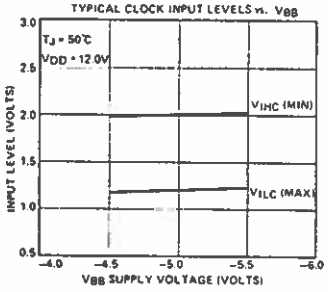


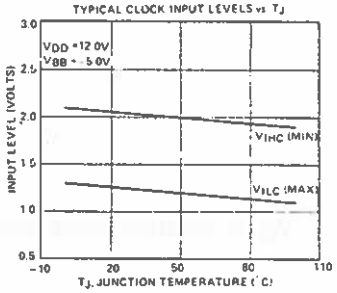


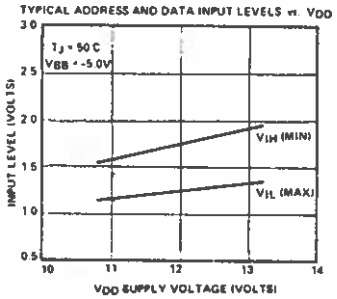


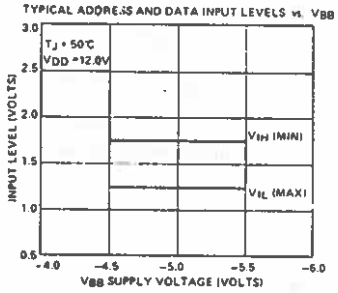


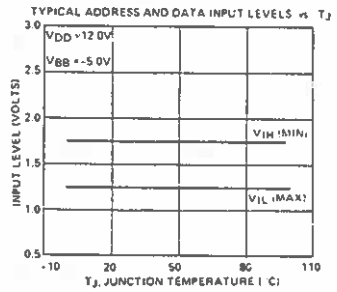

TYPICAL CLOCK INPUT LEVELS vs. VDD
TJ = 50°C
VBB = -5.0V
VIHC (MIN)
VILC (MAX)
INPUT LEVEL (VOLTS)
VDD SUPPLY VOLTAGE (VOLTS)


TYPICAL CLOCK INPUT LEVELS vs. VBB
TJ = 50°C
VDD = 12.0V
VIHC (MIN)
VILC (MAX)
INPUT LEVEL (VOLTS)
VBB SUPPLY VOLTAGE (VOLTS)


TYPICAL CLOCK INPUT LEVELS vs TJ
VDD = 12.0V
VBB = -5.0V
VIHC (MIN)
VILC (MAX)
INPUT LEVEL (VOLTS)
TJ, JUNCTION TEMPERATURE (°C)


TYPICAL ADDRESS AND DATA INPUT LEVELS vs. VDD
TJ = 50°C
VBB = -5.0V
VIH (MIN)
VIL (MAX)
INPUT LEVEL (VOLTS)
VDD SUPPLY VOLTAGE (VOLTS)


TYPICAL ADDRESS AND DATA INPUT LEVELS vs. VBB
TJ = 50°C
VDD = 12.0V
VIH (MIN)
VIL (MAX)
INPUT LEVEL (VOLTS)
VBB SUPPLY VOLTAGE (VOLTS)


TYPICAL ADDRESS AND DATA INPUT LEVELS vs TJ
VDD = 12.0V
VBB = -5.0V
VIH (MIN)
VIL (MAX)
INPUT LEVEL (VOLTS)
TJ, JUNCTION TEMPERATURE (°C)

Fiche technique :

EPROM 2716

MM2716

National Semiconductor

MOS EPROMs

MM2716 16,384-Bit (2048 x 8) UV Erasable PROM

General Description

The MM2716 is a high speed 16k UV erasable and electrically reprogrammable EPROM ideally suited for applications where fast turn-around and pattern experimentation are important requirements.

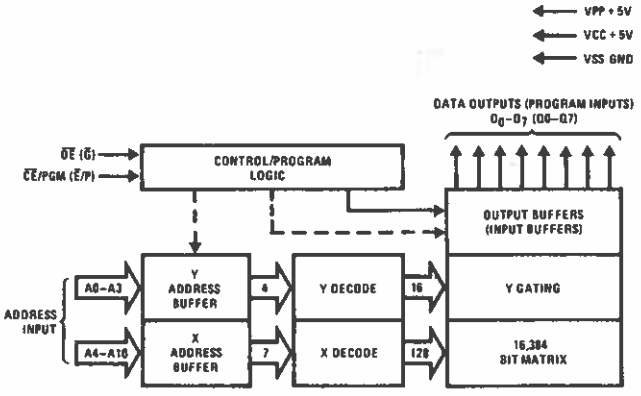
The MM2716 is packaged in a 24-pin dual-in-line package with transparent lid. The transparent lid allows the user to expose the chip to ultraviolet light to erase the bit pattern. A new pattern can then be written into the device by following the programming procedure.

This EPROM is fabricated with the reliable, high volume, time proven, N-channel silicon gate technology.

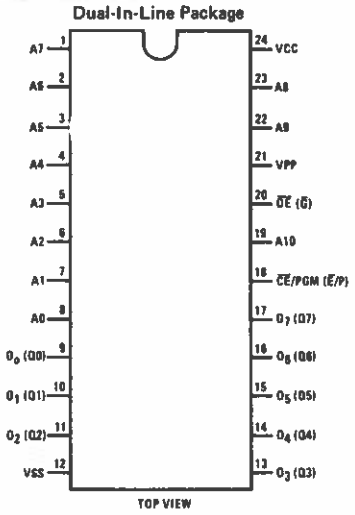
Features

- 2048 x 8 organization
- 525 mW max active power, 132 mW max standby power
- Low power during programming
- Access time—MM2716, 450 ns; MM2716-1, 350 ns; MM2716-2, 390 ns
- Single 5V power supply
- Static—no clocks required
- Inputs and outputs TTL compatible during both read and program modes
- TRI-STATE® output

Block and Connection Diagrams *



Dual-In-Line Package



Order Number MM2716Q, MM2716Q-1 or MM2716Q-2
See NS Package J24CQ

Pin Connection During Read or Program

MODE	PIN NAME/NUMBER				
	CE/PGM (E/P) 18	OE (G) 20	VPP 21	VCC 24	OUTPUTS 9–11, 13–17
Read	VIL	VIL	5	5	DOUT
Program	Pulsed VIL to VIH	VIH	25	5	DIN

Pin Names

A0–A10	Address Inputs
O0–O7 (Q0–Q7)	Data Outputs
CE/PGM (E/P)	Chip Enable/Program
OE (G)	Output Enable
VPP	Read 5V, Program 25V
VCC	Power (5V)
VSS	Ground

*Symbols in parentheses are proposed industry standard

Absolute Maximum Ratings (Note 1)

Temperature Under Bias	−25°C to +85°C	All Input or Output Voltages with Respect to VSS (except VPP)	6V to −0.3V
Storage Temperature	−65°C to +125°C	Power Dissipation	1.5 W
VPP Supply Voltage with Respect to VSS	26.5V to −0.3V	Lead Temperature (Soldering, 10 seconds)	300°C

READ OPERATION (Note 2)

DC Operating Characteristics

T_A = 0°C to +70°C, VCC = 5V ±5%, (VCC = 5V ±10% for MM2716-1), VPP = VCC ±0.6V (Note 3), VSS = 0V, unless otherwise noted.

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
ILI	Input Current	VIN = 5.25V or VIN = VIL			10	μA
ILO	Output Leakage Current	VOUT = 5.25V, $\overline{CE}$/PGM = 5V			10	μA
IPP1	VPP Supply Current	VPP = 5.85V			5	mA
ICC1	VCC Supply Current (Standby)	$\overline{CE}$/PGM = VIH, $\overline{OE}$ = VIL		10	25	mA
ICC2	VCC Supply Current (Active)	$\overline{CE}$/PGM = $\overline{OE}$ = VIL		57	100	mA
VIL	Input Low Voltage		0.1		0.8	V
VIH	Input High Voltage		2.0		V_{CC} + 1	V
VOH	Output High Voltage	IOH = −400 μA	2.4			V
VOL	Output Low Voltage	IOL = 2.1 mA			0.45	V

AC Characteristics (Note 4)

T_A = 0°C to +70°C, VCC = 5V ±5%, (VCC = 5V ±10% for MM2716-1), VPP = VCC ±0.6V (Note 3), VSS = 0V, unless otherwise noted.

SYMBOL		PARAMETER	CONDITIONS	MM2716		MM2716-1		MM2716-2		UNITS
ALTERNATE	STANDARD			MIN	MAX	MIN	MAX	MIN	MAX	
t_{ACC}	TAVQV	Address to Output Delay	$\overline{CE}$/PGM = $\overline{OE}$ = VIL		450		350		390	ns
t_{CE}	TELQV	$\overline{CE}$ to Output Delay	$\overline{OE}$ = VIL		450		350		390	ns
t_{OE}	TGLQV	Output Enable to Output Delay	$\overline{CE}$/PGM = VIL		120		120		120	ns
t_{DF}	TGHQZ	Output Enable High to Output Hi-Z	$\overline{CE}$/PGM = VIL	0	100	0	100	0	100	ns
t_{OH}	TAXQX	Address to Output Hold	$\overline{CE}$/PGM = $\overline{OE}$ = VIL	0		0		0		ns
t_{OD}	TEHQZ	$\overline{CE}$ to Output Hi-Z	$\overline{OE}$ = VIL	0	100	0	100	0	100	ns

Capacitance (Note 5)

T_A = 25°C, f = 1 MHz

SYMBOL	PARAMETER	CONDITIONS	TYP	MAX	UNITS
CI	Input Capacitance	VIN = 0V	4	6	pF
CO	Output Capacitance	VOUT = 0V	8	12	pF

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. Except for "Operating Temperature Range" they are not meant to imply that the devices should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: Typical conditions are for operation at: T_A = 25°C, VCC = 5V, VPP = VCC, and VSS = 0V.

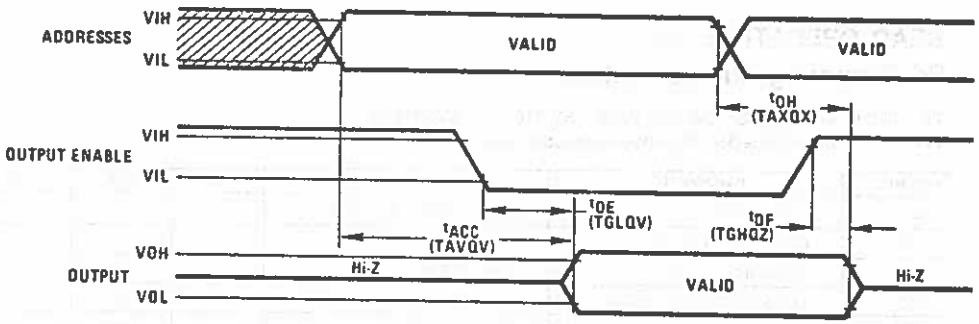
Note 3: VPP may be connected to VCC except during program. The ±0.6V tolerance allows a circuit to switch VPP between the read voltage and the program voltage.

Note 4: Output load: 1 TTL gate and CL = 100 pF. Input rise and fall times ≤ 20 ns.

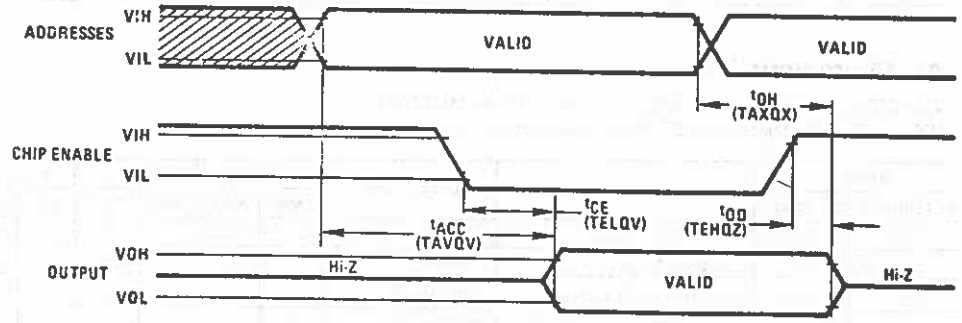
Note 5: Capacitance is guaranteed by periodic testing.

Switching Time Waveforms *

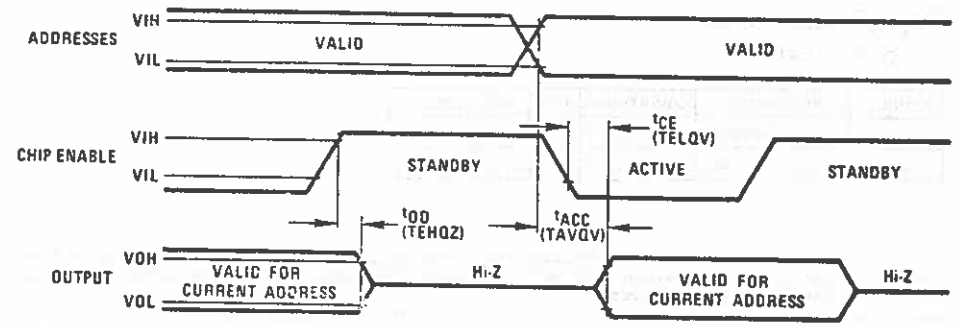
Read Cycle ($\overline{CE}$/PGM = VIL)



Read Cycle ($\overline{OE}$ = VIL)



Standby Power Down Mode ($\overline{OE}$ = VIL)



*Symbols in parentheses are proposed industry standard

PROGRAM OPERATION

DC Electrical Characteristics and Operating Conditions (Notes 1 and 2)

(T_A = 25°C ±5°C) (VCC = 5V ±5%, VPP = 25V ±1V)

SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS
ILI	Input Leakage Current (Note 3)			10	μA
VIL	Input Low Level	−0.1		0.8	V
VIH	Input High Level	2.0		VCC + 1	V
ICC	VCC Power Supply Current			100	mA
IPP1	VPP Supply Current (Note 4)			5	mA
IPP2	VPP Supply Current During Programming Pulse (Note 5)			30	mA

AC Characteristics and Operating Conditions (Notes 1, 2, and 6)

(T_A = 25°C ±5°C) (VCC = 5V ±5%, VPP = 25V ±1V)

SYMBOL		PARAMETER	MIN	TYP	MAX	UNITS
ALTERNATE	STANDARD					
tAS	TAVPH	Address Setup Time	2			μs
tOS	TGHPH	$\overline{OE}$ Setup Time	2			μs
tDS	TDVPH	Data Setup Time	2			μs
tAH	TPLAX	Address Hold Time	2			μs
tOH	TPLGX	$\overline{OE}$ Hold Time	2			μs
tDH	TPLDX	Data Hold Time	2			μs
tDF	TGHQZ	Chip Disable to Output Float Delay (Note 4)	0		100	ns
tCE	TGLQV	Chip Enable to Output Delay (Note 4)			120	ns
tPW	TPHPL	Program Pulse Width	45	50	55	ms
tPR	TPH1PH2	Program Pulse Rise Time	5			ns
tPF	TPL2PL1	Program Pulse Fall Time	5			ns

Note 1: VCC must be applied at the same time or before VPP and removed after or at the same time as VPP. To prevent damage to the device it must not be inserted into a board with power applied.

Note 2: Care must be taken to prevent overshoot of the VPP supply when switching to +25V.

Note 3: $0.45V \leq V_{IN} \leq 5.25V$.

Note 4: $\overline{CE}/PGM = V_{IL}$, VPP = VCC + 0.6V.

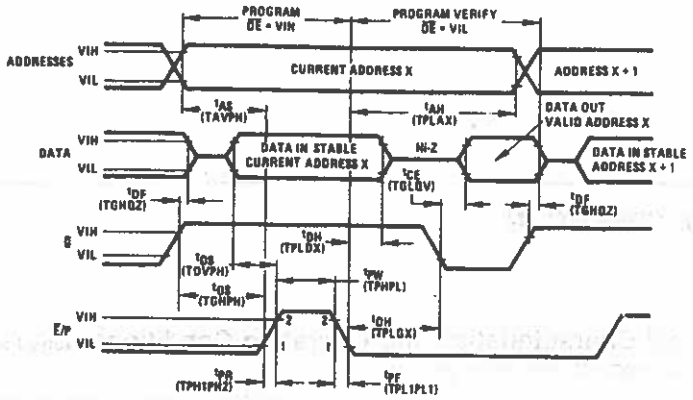
Note 5: VPP = 26V.

Note 6: Transition times ≤ 20 ns unless noted otherwise.

5

Timing Diagram *

Program Mode



Functional Description

DEVICE OPERATION

The MM2716 has 3 modes of operation in the normal system environment. These are shown in Table I.

Read Mode

The MM2716 read operation requires that $\overline{OE}$ = VIL, $\overline{CE}$/PGM = VIL and that addresses A0–A10 have been stabilized. Valid data will appear on the output pins after t_{ACC}, t_{OE} or t_{CE} times (see Switching Time Waveforms) depending on which is limiting.

Deselect Mode

The MM2716 is deselected by making $\overline{OE}$ = VIH. This mode is independent of $\overline{CE}$/PGM and the condition of the addresses. The outputs are Hi-Z when $\overline{OE}$ = VIH. This allows OR-tying 2 or more MM2716's for memory expansion.

Standby Mode (Power Down)

The MM2716 may be powered down to the standby mode by making $\overline{CE}$/PGM = VIH. This is independent of $\overline{OE}$ and automatically puts the outputs in their Hi-Z state. The power is reduced to 25% (132 mW max) of the normal operating power. VCC and VPP must be maintained at 5V. Access time at power up remains either t_{ACC} or t_{CE} (see Switching Time Waveforms).

PROGRAMMING

The MM2716 is shipped from National completely erased. All bits will be at a "1" level (output high) in this initial state and after any full erasure. Table II shows the 3 programming modes.

TABLE I. OPERATING MODES (VCC = VPP = 5V)

MODE	PIN NAME/NUMBER		
	$\overline{CE}$/PGM ($\overline{E}$/P) 18	$\overline{OE}$ ($\overline{G}$) 20	OUTPUTS 9–11, 13–17
Read	VIL	VIL	DOUT
Deselect	Don't Care	VIH	Hi-Z
Standby	VIH	Don't Care	Hi-Z

TABLE II. PROGRAMMING MODES (VCC = 5V)

MODE	PIN NAME/NUMBER			
	$\overline{CE}$/PGM ($\overline{E}$/P) 18	$\overline{OE}$ ($\overline{G}$) 20	VPP 21	OUTPUTS O 9–11, 13–17
Program	Pulsed VIL to VIH	VIH	25	DIN
Program Verify	VIL	VIL	25(5)	DOUT
Program Inhibit	VIL	VIH	25	Hi-Z

*Symbols in parentheses are proposed industry standard

Functional Description (Continued)

Program Mode

The MM2716 is programmed by introducing "0"s into the desired locations. This is done 8 bits (a byte) at a time. Any individual address, a sequence of addresses, or addresses chosen at random may be programmed. Any or all of the 8 bits associated with an address location may be programmed with a single program pulse applied to the chip enable pin. All input voltage levels, including the program pulse on chip enable are TTL compatible. The programming sequence is:

With VPP = 25V, VCC = 5V, $\overline{OE}$ = VIH and $\overline{CE}$/PGM = VIL, an address is selected and the desired data word is applied to the output pins. (VIL = "0" and VIL = "1" for both address and data.) After the address and data signals are stable the program pin is pulsed from VIL to VIH with a pulse width between 45 ms and 55 ms.

Multiple pulses are not needed but will not cause device damage. No pins should be left open. A high level (VIH or higher) *must not* be maintained longer than $t_{PW(MAX)}$ on the program pin during programming. MM2716's may be programmed in parallel with the same data in this mode.

Program Verify Mode

The programming of the MM2716 may be verified either 1 word at a time during the programming (as shown in the timing diagram) or by reading all of the words out at the end of the programming sequence. This can be done with VPP = 25V (or 5V) in either case.

Program Inhibit Mode

The program inhibit mode allows programming several MM2716's simultaneously with different data for each one by controlling which ones receive the program pulse. All similar inputs of the MM2716 may be paralleled. Pulsing the program pin (from VIL to VIH) will program a unit while inhibiting the program pulse to a unit will keep it from being programmed and keeping $\overline{OE}$ = VIH will put its outputs in the Hi-Z state.

ERASING

The MM2716 is erased by exposure to high intensity ultraviolet light through the transparent window. This exposure discharges the floating gate to its initial state through induced photo current. It is recommended that the MM2716 be kept out of direct sunlight. The UV content of sunlight may cause a partial erasure of some bits in a relatively short period of time. Direct sunlight can also cause temporary functional failure. Extended exposure to room level fluorescent lighting will also cause erasure. An opaque coating (paint, tape, label, etc.) should be placed over the package window if this product is to be operated under these lighting conditions.

An ultraviolet source of 2537 Å yielding a total integrated dosage of 15 watt-seconds/cm^2 is required. This will erase the part in approximately 15 to 20 minutes if a UV lamp with a 12,000 $\mu W/cm^2$ power rating is used. The MM2716 to be erased should be placed 1 inch away from the lamp and no filters should be used.

An erasure system should be calibrated periodically. The distance from lamp to unit should be maintained at 1 inch. The erasure time is increased by the square of the distance (if the distance is doubled the erasure time goes up by a factor of 4). Lamps lose intensity as they age. When a lamp is changed, the distance is changed, or the lamp is aged, the system should be checked to make certain full erasure is occurring. Incomplete erasure will cause symptoms that can be misleading. Programmers, components, and system designs have been erroneously suspected when incomplete erasure was the basic problem.

Fiches techniques :

74 LS 138 et 74 LS 139

TTL MSI

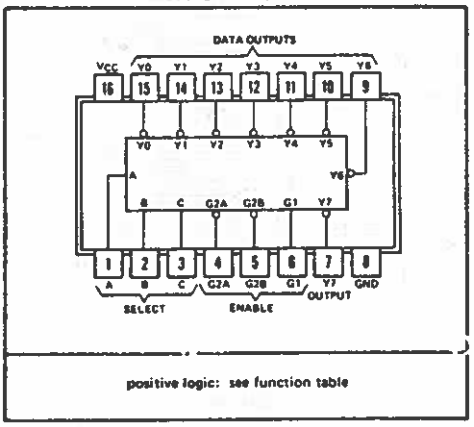
TYPES SN54LS138, SN54LS139, SN54S138, SN54S139, SN74LS138, SN74LS139, SN74S138, SN74S139 DECODERS/DEMULTIPLEXERS

BULLETIN NO. DL-S 7611804, DECEMBER 1972—REVISED OCTOBER 1976

- Designed Specifically for High-Speed:
  Memory Decoders
  Data Transmission Systems
- 'S138 and 'LS138 3-to-8-Line Decoders Incorporate 3 Enable Inputs to Simplify Cascading and/or Data Reception
- 'S139 and 'LS139 Contain Two Fully Independent 2-to-4-Line Decoders/ Demultiplexers
- Schottky Clamped for High Performance

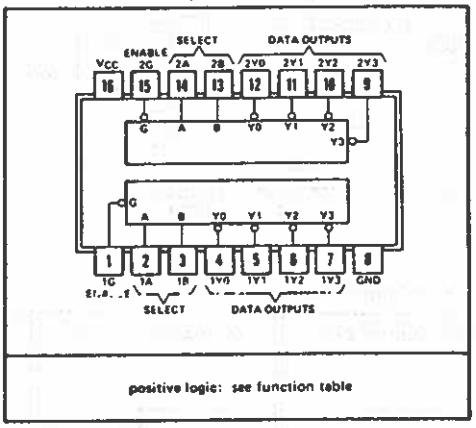
TYPE	TYPICAL PROPAGATION DELAY (3 LEVELS OF LOGIC)	TYPICAL POWER DISSIPATION
'LS138	22 ns	32 mW
'S138	8 ns	245 mW
'LS139	22 ns	34 mW
'S139	7.5 ns	300 mW

SN54LS138, SN54S138 . . . J OR W PACKAGE
SN74LS138, SN74S138 . . . J OR N PACKAGE
(TOP VIEW)



positive logic: see function table

SN54LS139, SN54S139 . . . J OR W PACKAGE
SN74LS139, SN74S139 . . . J OR N PACKAGE
(TOP VIEW)



positive logic: see function table

description

These Schottky-clamped TTL MSI circuits are designed to be used in high-performance memory-decoding or data-routing applications requiring very short propagation delay times. In high-performance memory systems these decoders can be used to minimize the effects of system decoding. When employed with high-speed memories utilizing a fast-enable circuit the delay times of these decoders and the enable time of the memory are usually less than the typical access time of the memory. This means that the effective system delay introduced by the Schottky-clamped system decoder is negligible.

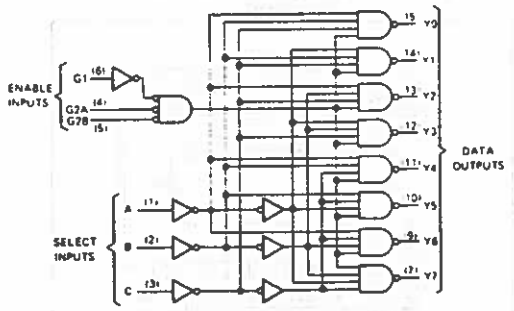
The 'LS138 and 'S138 decode one-of-eight lines dependent on the conditions at the three binary select inputs and the three enable inputs. Two active-low and one active-high enable inputs reduce the need for external gates or inverters when expanding. A 24-line decoder can be implemented without external inverters and a 32-line decoder requires only one inverter. An enable input can be used as a data input for demultiplexing applications.

The 'LS139 and 'S139 comprise two individual two-line-to-four-line decoders in a single package. The active-low enable input can be used as a data line in demultiplexing applications.

All of these decoders/demultiplexers feature fully buffered inputs each of which represents only one normalized Series 54LS/74LS load ('LS138, 'LS139) or one normalized Series 54S/74S load ('S138, 'S139) to its driving circuit. All inputs are clamped with high-performance Schottky diodes to suppress line-ringing and simplify system design. Series 54LS and 54S devices are characterized for operation over the full military temperature range of −55°C to 125°C; Series 74LS and 74S devices are characterized for 0°C to 70°C industrial systems.

functional block diagrams and logic

'LS138, 'S138

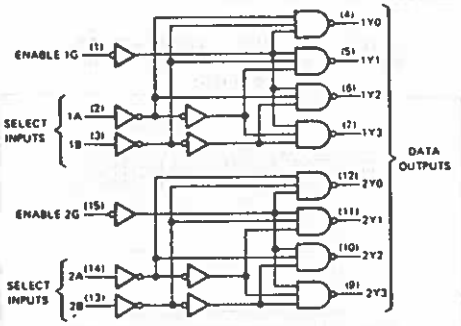


'LS138, 'S138
FUNCTION TABLE

INPUTS					OUTPUTS							
ENABLE		SELECT										
G1	G2*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

*G2 = G2A + G2B
H = high level, L = low level, X = irrelevant

'LS139, 'S139

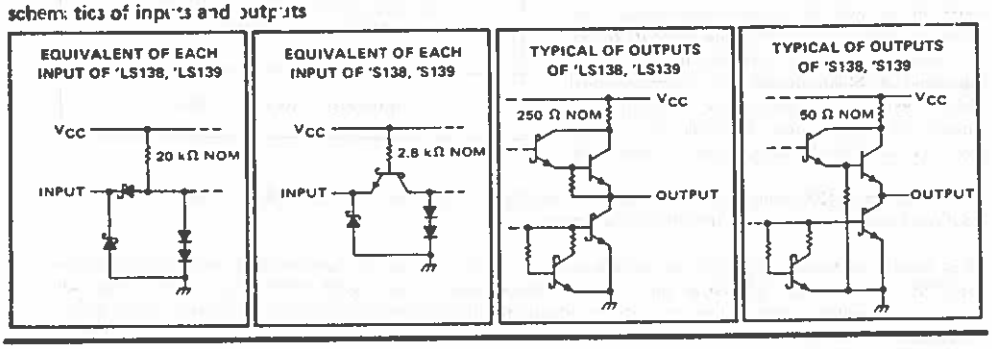


'LS139, 'S139
(EACH DECODER/DEMULTIPLEXER)
FUNCTION TABLE

INPUTS			OUTPUTS			
ENABLE	SELECT					
G	B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

H = high level, L = low level, X = irrelevant

schematics of inputs and outputs



TYPES SN54LS138, SN54LS139, SN74LS138, SN74LS139, DECODERS/DEMULTIPLEXERS

REVISED OCTOBER 1976

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1) . . . . . 7 V
Input voltage . . . . . 7 V
Operating free-air temperature range: SN54LS138, SN54LS139 Circuits . . . . . −55°C to 125°C
SN74LS138, SN74LS139 Circuits . . . . . 0°C to 70°C
Storage temperature range . . . . . −65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

recommended operating conditions

	SN54LS138 SN54LS139			SN74LS138 SN74LS139			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			−400			−400	μA
Low-level output current, I_{OL}			4			8	mA
Operating free-air temperature, T_A	−55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS†		SN54LS138 SN54LS139			SN74LS138 SN74LS139			UNIT
				MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IH}	High-level input voltage			2			2			V
V_{IL}	Low-level input voltage					0.7			0.8	V
V_{IK}	Input clamp voltage	V_{CC} = MIN, I_I = −18 mA				−1.5			−1.5	V
V_{OH}	High-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = $V_{IL\ max}$, I_{OH} = −400 μA		2.5	3.4		2.7	3.4		V
V_{OL}	Low-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = $V_{IL\ max}$	I_{OL} = 4 mA		0.25	0.4		0.25	0.4	V
			I_{OL} = 8 mA					0.35	0.5	
I_I	Input current at maximum input voltage	V_{CC} = MAX, V_I = 7 V				0.1			0.1	mA
I_{IH}	High-level input current	V_{CC} = MAX, V_I = 2.7 V				20			20	μA
I_{IL}	Low-level input current	V_{CC} = MAX, V_I = 0.4 V				−0.4			−0.4	mA
I_{OS}	Short-circuit output current§	V_{CC} = MAX		−6		−40	−5		−42	mA
I_{CC}	Supply current	V_{CC} = MAX, Outputs enabled and open	'LS138		6.3	10		6.3	10	mA
			'LS139		6.8	11		6.8	11	

†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable device type.
‡All typical values are at V_{CC} = 5 V, T_A = 25°C.
§Not more than one output should be shorted at a time.

switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER¶	FROM (INPUT)	TO (OUTPUT)	LEVELS OF DELAY	TEST CONDITIONS	SN54LS138 SN74LS138			SN54LS139 SN74LS139			UNIT
					MIN	TYP	MAX	MIN	TYP	MAX	
t_{PLH}	Binary Select	Any	2	C_L = 15 pF, R_L = 2 kΩ, See Note 2		13	20		13	20	ns
t_{PHL}						27	41		22	33	ns
t_{PLH}			3			18	27		18	29	ns
t_{PHL}						26	39		25	38	ns
t_{PLH}	Enable	Any	2			12	18		16	24	ns
t_{PHL}						21	32		21	32	ns
t_{PLH}			3			17	26				ns
t_{PHL}						25	38				ns

¶t_{PLH} = propagation delay time, low-to-high-level output; t_{PHL} = propagation delay time, high-to-low-level output.
NOTE 2: Load circuits and waveforms are shown on page 3-11.

7

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1) . . . . . 7 V
Input voltage . . . . . 5.5 V
Operating free-air temperature range: SN54S138, SN54S139 Circuits . . . . . −55°C to 125°C
SN74S138, SN74S139 Circuits . . . . . 0°C to 70°C
Storage temperature range . . . . . −65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

recommended operating conditions

	SN54S138 SN74S139			SN74S138 SN74S139			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			−1			−1	mA
Low-level output current, I_{OL}			20			20	mA
Operating free-air temperature, T_A	−55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS†		SN54S138 SN74S138			SN54S139 SN74S139			UNIT
			MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IH} High-level input voltage			2			2			V
V_{IL} Low-level input voltage					0.8			0.8	V
V_{IK} Input clamp voltage	V_{CC} = MIN, I_I = −18 mA				−1.2			−1.2	V
V_{OH} High-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V, I_{OH} = −1 mA	SN54S'	2.5	3.4		2.5	3.4		V
		SN74S'	2.7	3.4		2.7	3.4		
V_{OL} Low-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V, I_{OL} = 20 mA				0.5			0.5	V
I_I Input current at maximum input voltage	V_{CC} = MAX, V_I = 5.5 V				1			1	mA
I_{IH} High-level input current	V_{CC} = MAX, V_I = 2.7 V				50			50	µA
I_{IL} Low-level input current	V_{CC} = MAX, V_I = 0.5 V				−2			−2	mA
I_{OS} Short-circuit output current§	V_{CC} = MAX		−40		−100	−40		−100	mA
I_{CC} Supply current	V_{CC} = MAX, Outputs enabled and open			49	74		60	90	mA

†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable device type.
‡All typical values are at V_{CC} = 5 V, T_A = 25°C.
§Not more than one output should be shorted at a time, and duration of the short-circuit test should not exceed one second.

switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER¶	FROM (INPUT)	TO (OUTPUT)	LEVELS OF DELAY	TEST CONDITIONS	SN54S138, SN74S138			SN54S139 SN74S139			UNIT
					MIN	TYP	MAX	MIN	TYP	MAX	
t_{PLH}	Binary select	Any	2	C_L = 15 pF, R_L = 280 Ω, See Note 3		4.5	7		5	7.5	ns
t_{PHL}						7	10.5		6.5	10	
t_{PLH}			3			7.5	12		7	12	ns
t_{PHL}						8	12		8	12	
t_{PLH}	Enable	Any	2			5	8		5	8	ns
t_{PHL}						7	11		6.5	10	
t_{PLH}			3			7	11				ns
t_{PHL}						7	11				

¶t_{PLH} ≡ propagation delay time, low-to-high-level output
t_{PHL} ≡ propagation delay time, high-to-low-level output
NOTE 3: Load circuits and waveforms are shown on page 3-10.

Fiche technique :

74 LS 367-A

recommended operating conditions

		54 FAMILY 74 FAMILY	SERIES 54 SERIES 74 '365A, '366A '367A, '368A			SERIES 54LS SERIES 74LS 'LS366, 'LS368 'LS367, 'LS369			UNIT
			MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage	54 Family	4.5	5	5.5	4.5	5	5.5	V
		74 Family	4.75	5	5.25	4.75	5	5.25	
I_{OH}	High-level output current	54 Family			−2			−1	mA
		74 Family			−5.2			−2.6	
I_{OL}	Low-level output current	54 Family			32			12	mA
		74 Family			32			24	
T_A	Operating free-air temperature	54 Family	−55		125	−55		125	°C
		74 Family	0		70	0		70	

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

	PARAMETER		TEST FIGURE	TEST CONDITIONS†		SERIES 54 SERIES 74 '365A, '366A '367A, '368A			SERIES 54LS SERIES 74LS 'LS366, 'LS368 'LS367, 'LS369			UNIT
						MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IH}	High-level input voltage		1, 2			2			2			V
V_{IL}	Low-level input voltage		1, 2		54 Family			0.8			0.7	V
					74 Family			0.8			0.8	
V_{IK}	Input clamp voltage		3	V_{CC} = MIN, I_I = §,				−1.5			−1.5	V
V_{OH}	High-level output voltage		1	V_{CC} = MIN, V_{IH} = 2 V,	54 Family	2.4	3.3		2.4	3.3		V
				V_{IL} = V_{IL} max, I_{OH} = MAX	74 Family	2.4	3.1		2.4	3.1		
V_{OL}	Low-level output voltage		2	V_{CC} = MIN, I_{OL} = MAX,	54 Family			0.4			0.25	0.4	V
				V_{IH} = 2 V, I_{OL} = 12 mA	74 Family			0.4			0.35	0.5	
				V_{IL} = V_{IL} max	Series 74LS						0.25	0.4	
I_{OZ}	Off-state (high-impedance state) output current		19	V_{CC} = MAX, V_{IH} = 2 V, V_O = 2.4 V				40			20	μA
				V_{IL} = V_{IL} max	V_O = 0.4 V			−40			−20	
I_I ¶	Input current at maximum input voltage		4	V_{CC} = MAX	V_I = 5.5 V			1			0.1	mA
					V_I = 7 V							
I_{IH}	High-level input current		4	V_{CC} = MAX	V_{IH} = 2.4 V			40			20	μA
					V_{IH} = 2.7 V							
I_{IL}	Low-level input current	A inputs		V_{CC} = MAX, V_I = 0.5 V, Either $\overline{G}$ input at 2 V				−40			−20	μA
				V_{CC} = MAX, V_I = 0.4 V, Both $\overline{G}$ inputs at 0.4 V				−1.6			−0.4	mA
		$\overline{G}$ inputs		V_{CC} = MAX, V_I = 0.4 V,				−1.6			−0.4	
I_{OS}	Short-circuit output current#		6	V_{CC} = MAX		−40		−130	−40		−225	mA
I_{CC}	Supply current		7	V_{CC} = MAX				See table on next page				mA

†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡All typical values are at V_{CC} = 5 V, T_A = 25°C.
§I_I = −12 mA for SN54'/SN74' and −18 mA for SN54LS'/SN74LS', SN54S/SN74S'.
¶Not more than one output should be shorted at a time, and for SN54LS'/SN74LS' and SN54S'/SN74S', duration of output short-circuit should not exceed one second.

supply current[1]

TYPE	DATA INPUTS	OUTPUT CONTROLS	I_{CC} (mA) TYP	I_{CC} (mA) MAX
'365A, '367A	0 V	4.5 V	65	85
'366A, '368A	0 V	4.5 V	50	77
'LS365, 'LS367	0 V	4.5 V	14	24
'LS366, 'LS368	0 V	4.5 V	12	21

[1] Maximum values of I_{CC} are over the recommended operating ranges of V_{CC} and T_A; typical values are at V_{CC} = 5 V, T_A = 25°C.

switching characteristics, V_{CC} = 5 V, T_A = 25°C, see note 1

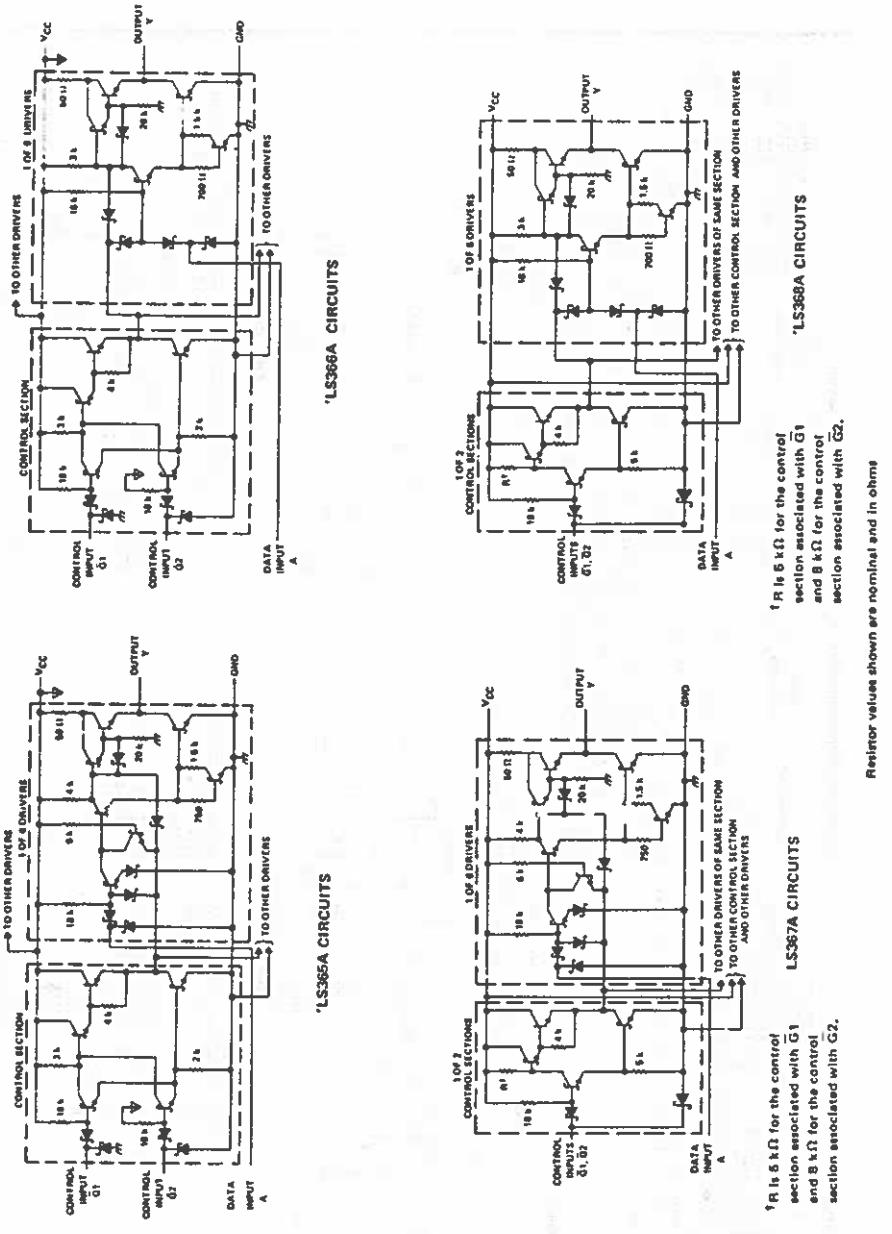
PARAMETER*	TEST CONDITIONS	SERIES 54/74 '365A, '367A TYP	SERIES 54/74 '365A, '367A MAX	SERIES 54/74 '366A, '368A TYP	SERIES 54/74 '366A, '368A MAX	TEST CONDITIONS	SERIES 54LS/74LS 'LS365A, 'LS367A TYP	SERIES 54LS/74LS 'LS365A, 'LS367A MAX	SERIES 54LS/74LS 'LS366A, 'LS368A TYP	SERIES 54LS/74LS 'LS366A, 'LS368A MAX
t_{PLH}	C_L = 45 pF, R_L = 667 Ω		16		17	C_L = 30 pF, R_L = 1 kΩ	10	16	7	15
t_{PHL}			22		16		9	22	12	18
t_{PZH}			35		35		18	35	18	35
t_{PZL}			37		37		24	40	28	45
t_{PHZ}	C_L = 5 pF, R_L = 667 Ω		11		11	C_L = 5 pF, R_L = 1 kΩ		30		32
t_{PLZ}			27		27			35		35

*t_{PLH} = Propagation delay time, low-to-high-level output
t_{PHL} = Propagation delay time, high-to-low-level output
t_{PZH} = Output enable time to high level
t_{PZL} = Output enable time to low level
t_{PHZ} = Output disable time from high level
t_{PLZ} = Output disable time from low level

NOTE 1: Load circuits and voltage waveforms are shown on pages 3-10 and 3-11.

schematics

HEX BUS DRIVERS WITH 3-STATE OUTPUTS



Fiche technique :

74 LS 244

TYPES SN54LS240,SN54LS241,SN54LS244,SN54S240,SN54S241, SN74LS240,SN74LS241,SN74LS244,SN74S240,SN74S241 OCTAL BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS

	Typical I_{OL} (Sink Current)	Typical I_{OH} (Source Current)	Typical Propagation Delay Times		Typical Enable/ Disable Times	Typical Power Dissipation (Enabled)	
			Inverting	Noninverting		Inverting	Noninverting
SN54LS'	12 mA	−12 mA	10.5 ns	12 ns	18 ns	130 mW	135 mW
SN74LS'	24 mA	−15 mA	10.5 ns	12 ns	18 ns	130 mW	135 mW
SN54S'	48 mA	−12 mA	4.5 ns	6 ns	9 ns	450 mW	538 mW
SN74S'	64 mA	−15 mA	4.5 ns	6 ns	9 ns	450 mW	538 mW

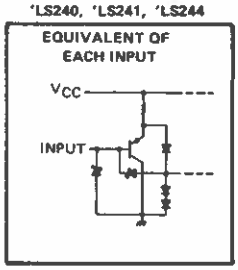
- 3-State Outputs Drive Bus Lines or Buffer Memory Address Registers
- P-N-P Inputs Reduce D-C Loading
- Hysteresis at Inputs Improves Noise Margins

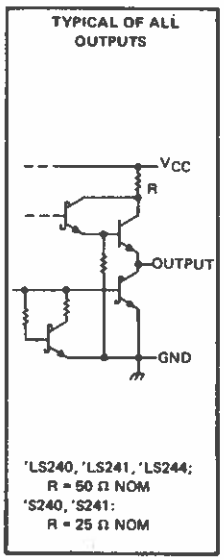
description

These octal buffers and line drivers are designed specifically to improve both the performance and density of three-state memory address drivers, clock drivers, and bus-oriented receivers and transmitters. The designer has a choice of selected combinations of inverting and noninverting outputs, symmetrical $\overline{G}$ (active-low output control) inputs, and complementary G and $\overline{G}$ inputs. These devices feature high fan-out, improved fan-in, and 400-mV noise-margin. The SN74LS' and SN74S' can be used to drive terminated lines down to 133 ohms.

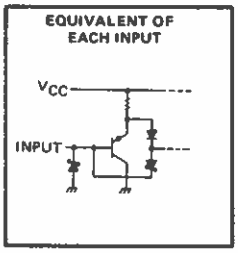
schematics of inputs and outputs

'LS240, 'LS241, 'LS244

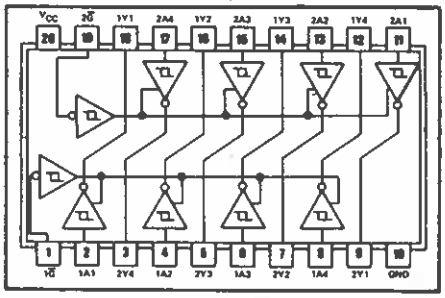




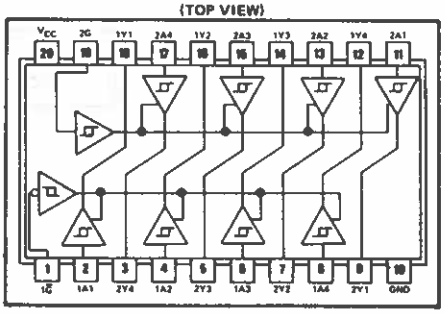
'S240 'S241



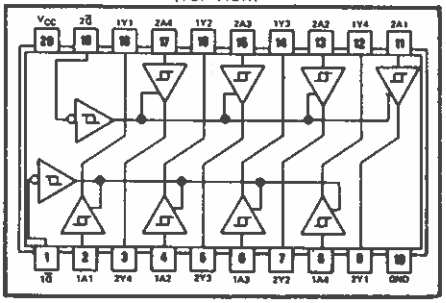
SN54LS240, SN54S240 . . . J
SN74LS240, SN74S240 . . . J OR N
(TOP VIEW)



SN54LS241, SN54S241 . . . J
SN74LS241, SN74S241 . . . J OR N
(TOP VIEW)



SN54LS244 . . . J
SN74LS244 . . . J OR N
(TOP VIEW)



TYPES SN54LS240,SN54LS241,SN54LS244, SN74LS240,SN74LS241,SN74LS244 BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS

recommended operating conditions

PARAMETER	SN54LS'			SN74LS'			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC} (see Note 1)	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			−12			−15	mA
Low-level output current, I_{OL}			12			24	mA
Operating free-air temperature, T_A	−55		125	0		70	°C

NOTE 1: Voltage values are with respect to network ground terminal.

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER			TEST CONDITIONS†		SN54LS'			SN74LS'			UNIT
					MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IH}	High-level input voltage				2			2			V
V_{IL}	Low-level input voltage						0.7			0.8	V
V_{IK}	Input clamp voltage		V_{CC} = MIN,	I_I = −18 mA			−1.5			−1.5	V
	Hysteresis ($V_{T+} - V_{T-}$)		V_{CC} = MIN		0.2	0.4		0.2	0.4		V
V_{OH}	High-level output voltage		V_{CC} = MIN, V_{IL} = V_{IL}max	V_{IH} = 2 V, I_{OH} = −3 mA	2.4	3.4		2.4	3.4		V
			V_{CC} = MIN, V_{IL} = 0.5 V,	V_{IH} = 2 V, I_{OH} = MAX	2			2			
V_{OL}	Low-level output voltage		V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = V_{IL}max	I_{OL} = 12 mA			0.4			0.4	V
				I_{OL} = 24 mA						0.5	
I_{OZH}	Off-state output current, high-level voltage applied		V_{CC} = MAX, V_{IH} = 2 V, V_{IL} = V_{IL}max	V_O = 2.7 V			20			20	μA
I_{OZL}	Off-state output current, low-level voltage applied			V_O = 0.4 V			−20			−20	
I_I	Input current at maximum input voltage		V_{CC} = MAX,	V_I = 7 V			0.1			0.1	mA
I_{IH}	High-level input current, any input		V_{CC} = MAX,	V_I = 2.7 V			20			20	μA
I_{IL}	Low-level input current		V_{CC} = MAX,	V_{IL} = 0.4 V			−0.2			−0.2	mA
I_{OS}	Short-circuit output current§		V_{CC} = MAX		−40		−225	−50		−225	mA
I_{CC}	Supply current	Outputs high	V_{CC} = MAX	All		13	23		13	23	mA
		Outputs low		'LS240		26	44		26	44	
			Outputs open	'LS241, 'LS244		27	46		27	46	
		All outputs disabled		'LS240		29	50		29	50	
				'LS241, 'LS244		32	54		32	54	

†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡All typical values are at V_{CC} = 5 V, T_A = 25°C.

§Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second.

switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER		TEST CONDITIONS		'LS240			'LS241, 'LS244			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	
t_{PLH}	Propagation delay time, low-to-high-level output	C_L = 45 pF, See Note 2	R_L = 667 Ω,		9	14		12	18	ns
t_{PHL}	Propagation delay time, high-to-low-level output				12	18		12	18	ns
t_{PZL}	Output enable time to low level				20	30		20	30	ns
t_{PZH}	Output enable time to high level				15	23		15	23	ns
t_{PLZ}	Output disable time from low level	C_L = 5 pF, See Note 2	R_L = 667 Ω,		15	25		15	25	ns
t_{PHZ}	Output disable time from high level				10	18		10	18	ns

NOTE 2: Load circuit and voltage waveforms are shown on page 3-11.

Fiche technique :

74 LS 273

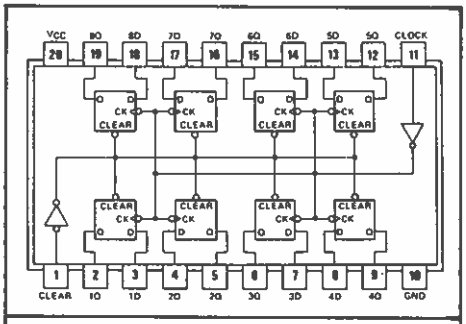
TTL MSI

TYPES SN54273, SN54LS273, SN74273, SN74LS273
OCTAL D-TYPE FLIP-FLOP WITH CLEAR

BULLETIN NO. DL-S 7612091, OCTOBER 1976

- Contains Eight Flip-Flops with Single-Rail Outputs
- Buffered Clock and Direct Clear Inputs
- Individual Data Input to Each Flip-Flop
- Applications Include:
  - Buffer/Storage Registers
  - Shift Registers
  - Pattern Generators

SN54273, SN54LS273 . . . J PACKAGE
SN74273, SN74LS273 . . . J OR N PACKAGE



positive logic: see function table

description

These monolithic, positive-edge-triggered flip-flops utilize TTL circuitry to implement D-type flip-flop logic with a direct clear input.

Information at the D inputs meeting the setup time requirements is transferred to the Q outputs on the positive-going edge of the clock pulse. Clock triggering occurs at a particular voltage level and is not directly related to the transition time of the positive-going pulse. When the clock input is at either the high or low level, the D input signal has no effect at the output.

These flip-flops are guaranteed to respond to clock frequencies ranging from 0 to 30 megahertz while maximum clock frequency is typically 40 megahertz. Typical power dissipation is 39 milliwatts per flip-flop for the '273 and 10 milliwatts for the 'LS273.

FUNCTION TABLE
(EACH FLIP-FLOP)

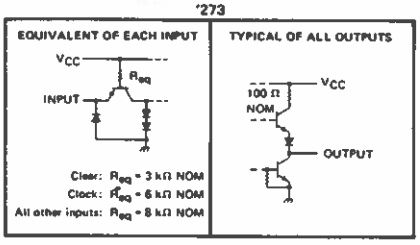
INPUTS			OUTPUT
CLEAR	CLOCK	D	Q
L	X	X	L
H	↑	H	H
H	↑	L	L
H	L	X	Q_0

See explanation of function tables on page 3-8.

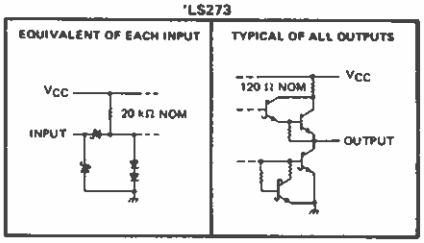
7

schematics of inputs and output

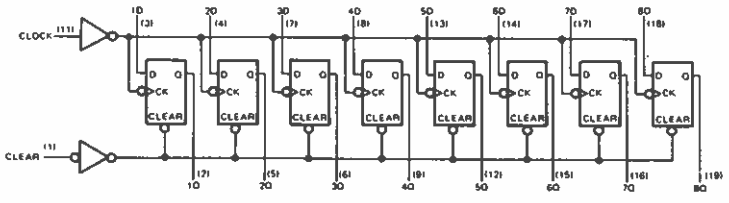
'273



'LS273



functional block diagram



Fiche technique :

6845

MOTOROLA

MC6845

Advance Information

MOS

(N-Channel, Silicon-Gate)

CRT CONTROLLER (CRTC)

CRT CONTROLLER (CRTC)

The MC6845 CRT Controller performs the interface to raster scan CRT displays. It is intended for use in processor-based controllers for CRT terminals in stand-alone or cluster configurations.

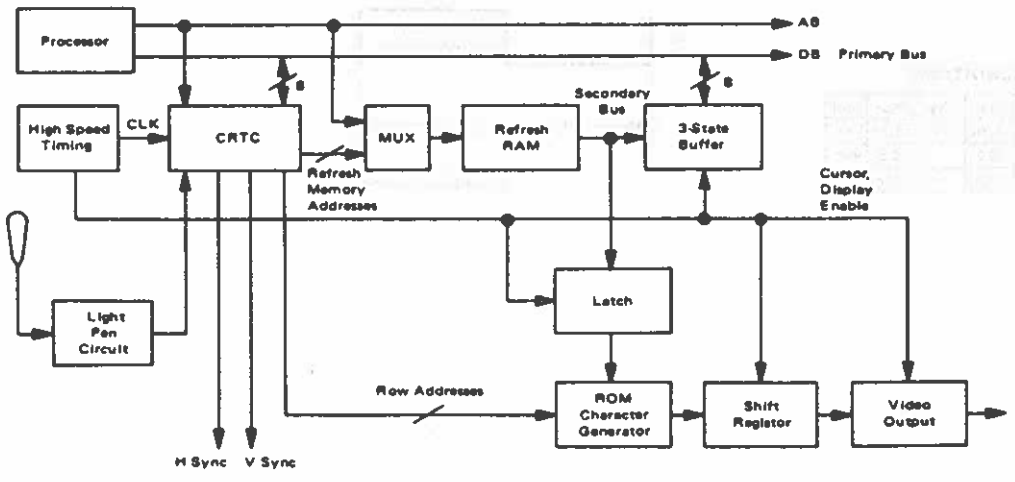
The CRTC is optimized for hardware/software balance in order to achieve integration of all key functions and maintain flexibility. For instance, all keyboard functions, R/W, cursor movements, and editing are under processor control; whereas the CRTC provides video timing and Refresh Memory Addressing.

- Applications include "glass-teletype," smart, programmable, intelligent CRT terminals; video games; information display.
- Alphanumeric, semi-graphic, and full graphic capability.
- Fully programmable via processor data bus. Can generate timing for almost any alphanumeric screen density, e.g. 80 x 24, 72 x 64, 132 x 20, etc.
- Single +5 volt supply. TTL/6800 compatible I/O.
- Hardware scroll (paging or by line or by character)
- Compatible with CPU's and MPU's which provide a means for synchronizing external devices.
- Cursor register and compare circuitry.
- Cursor format and blink are programmable.
- Light pen register.
- Line buffer-less operation. No external DMA required. Refresh Memory is multiplexed between CRTC and MPU.
- Programmable interlace or non-interlace scan.
- 14-bit wide refresh address.

L SUFFIX
CERAMIC PACKAGE
CASE 716

NOT SHOWN:
P SUFFIX
PLASTIC PACKAGE
CASE 711

FIGURE 1 – TYPICAL CRT CONTROLLER APPLICATION



This is advance information and specifications are subject to change without notice.

SYSTEM BLOCK DIAGRAM DESCRIPTION

As shown in Figure 1, the primary function of the CRTC is to generate refresh addresses (MA0-MA13), row selects (RA0-RA4), and video monitor timing (HSYNC, VSYNC) and Display Enable. Other functions include an internal cursor register which generates a Cursor output when its contents compare to the current Refresh Address. A light-pen strobe input signal allows capture of Refresh Address in an internal light pen register.

All timing in the CRTC is derived from the Clk input. In alphanumeric terminals, this signal is the character rate. Character rate is divided down from video rate by external High Speed Timing when the video frequency is greater than 3 MHz. Shift Register, Latch, and MUX Control signals are also provided by external High Speed Timing.

The processor communicates with the CRTC through a buffered 8-bit Data Bus by reading/writing into the 18-register file of the CRTC.

The Refresh Memory address is multiplexed between the Processor and CRTC. Data appears on a Secondary Bus which is buffered from the processor Primary Bus. A number of approaches are possible for solving contentions for the Refresh Memory.

1. Processor always gets priority.
2. Processor gets priority access anytime, but can be synchronized by an interrupt to perform accesses only during horizontal and vertical retrace times.
3. Synchronize processor by memory wait cycles.
4. Synchronize processor to character rate (See Figure 2). The 6800 MPU family lends itself to this configuration because it has constant cycle lengths. This method provides zero burden on the processor because there is never a contention for memory. All accesses are *"transparent."*

The secondary data bus concept in no way precludes using the Refresh RAM for other purposes. It looks like any other RAM to the Processor. For example, using Approach 4, a 64K byte RAM Refresh Memory could perform refresh and program storage functions transparently.

MAXIMUM RATINGS

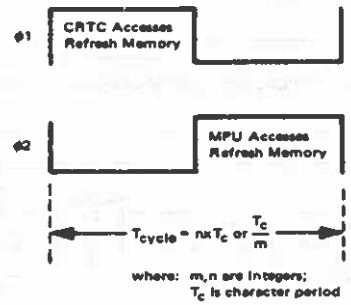
Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}*	-0.3 to +7.0	Vdc
Input Voltage	V_{in}*	-0.3 to +7.0	Vdc
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	-55 to +150	°C

*With respect to V_{SS} (Gnd).

RECOMMENDED OPERATING CONDITIONS

Characteristics	Symbol	Min	Typ	Max	Unit
Supply Voltage	V_{CC}	4.75	5.0	5.25	Vdc
Input Low Voltage	V_{IL}	-0.3	–	0.8	Vdc
Input High Voltage	V_{IH}	2.0	–	V_{CC}	Vdc

FIGURE 2 – TRANSPARENT REFRESH MEMORY CONFIGURATION TIMING USING 6800 MPU FAMILY



MC6845

ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 V ±5%, V_{SS} = 0, T_A = 0 to 70°C unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	2.0	–	V_{CC}	Vdc
Input Low Voltage		V_{IL}	-0.3	–	0.8	Vdc
Input Leakage Current		I_{in}	–	1.0	2.5	µAdc
Three-State (V_{CC} = 5.25 V) (V_{in} = 0.4 to 2.4 V)		I_{TSI}	-10	2.0	10	µAdc
Output High Voltage (I_{load} = -205 µA) (I_{load} = -100 µA)	 D0-D7 Other Outputs	V_{OH}	 2.4 2.4	 – –	 – –	Vdc
Output Low Voltage (I_{load} = 1.6 mA)		V_{OL}	–	–	0.4	Vdc
Power Dissipation		P_D	–	600	–	mW
Input Capacitance	 D0-D7 All others	C_{in}	 – –	 – –	 12.5 10	pF
Output Capacitance	All Outputs	C_{out}	–	–	10	pF
Minimum Clock Pulse Width, Low		P_{WCL}	160	–	–	ns
Minimum Clock Pulse Width, High		P_{WCH}	200	–	–	ns
Clock Frequency		f_c	–	–	2.5	MHz
Rise and Fall Time for Clock Input		t_{cr}, t_{cf}	–	–	20	ns
Memory Address Delay Time		t_{MAD}	–	–	160	ns
Raster Address Delay Time		t_{RAD}	–	–	160	ns
Display Timing Delay Time		t_{DTD}	–	–	300	ns
Horizontal Sync Delay Time		t_{HSD}	–	–	300	ns
Vertical Sync Delay Time		t_{VSD}	–	–	300	ns
Cursor Display Timing Delay Time		t_{CDD}	–	–	300	ns
Light Pen Strobe Minimum Pulse Width		PW_{LPH}	100	–	–	ns
Light Pen Strobe Disable Time		t_{LPD1}	–	–	120	ns
		t_{LPD2}	–	–	0	ns

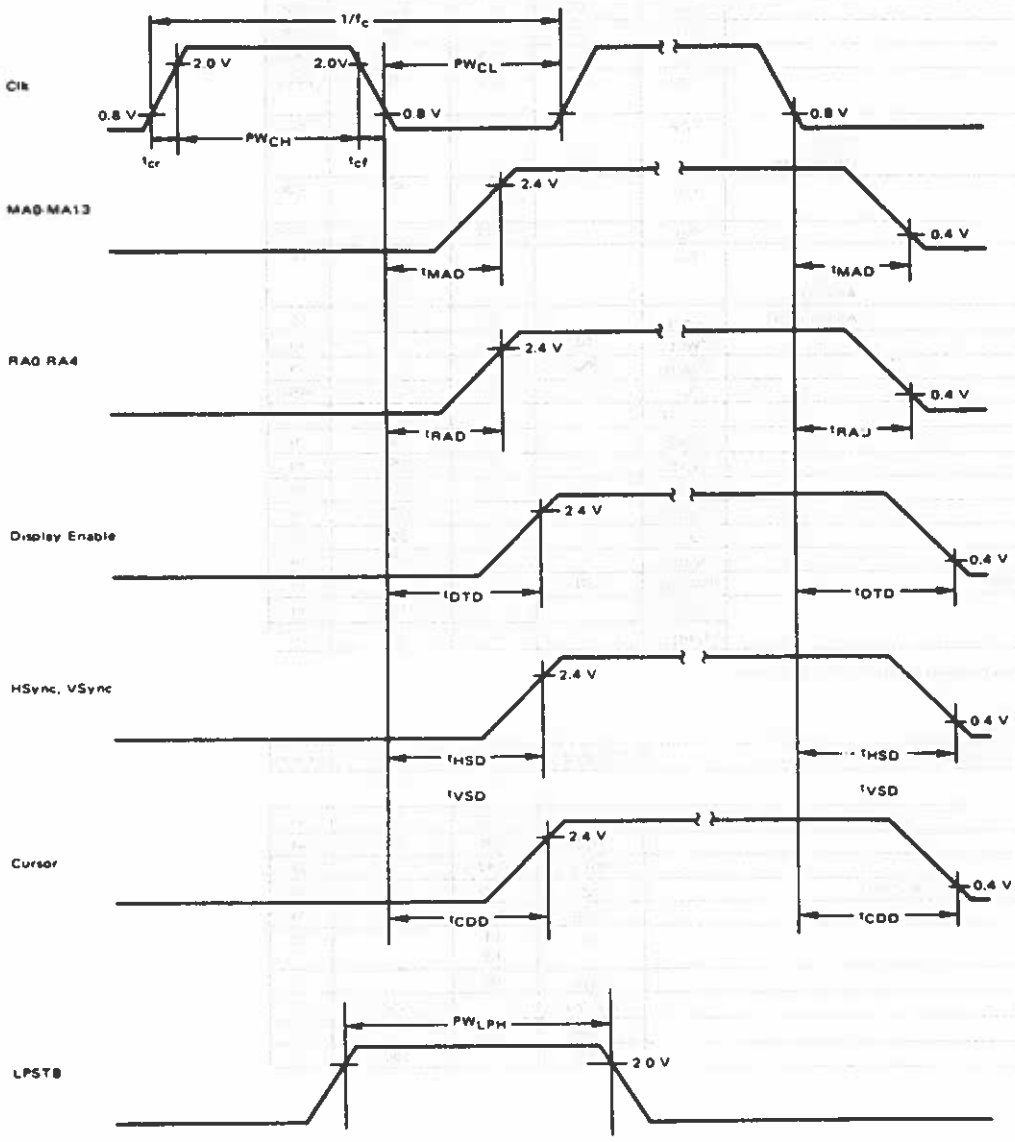
Note The light pen strobe must fall to low level before VSYNC pulse rises.

BUS TIMING CHARACTERISTICS

Characteristic	Symbol	Min	Max	Unit
READ/WRITE				
Enable Cycle Time	t_{cycE}	1.0	–	µs
Enable Pulse Width, High	PW_{EH}	0.45	25	µs
Enable Pulse Width, Low	PW_{EL}	0.43	–	µs
Setup Time, CS and RS valid to enable positive transition	t_{AS}	160		ns
Data Delay Time	t_{DDR}		320	ns
Data Hold Time (Read) (write)	t_H	10 10	– –	ns
Address Hold Time	t_{AH}	10	–	ns
Rise and Fall Time for Enable Input	t_{Er}, t_{Ef}		25	ns
Data Setup Time	t_{DSW}	195	–	ns
Data Access Time	t_{ACC}		480	ns

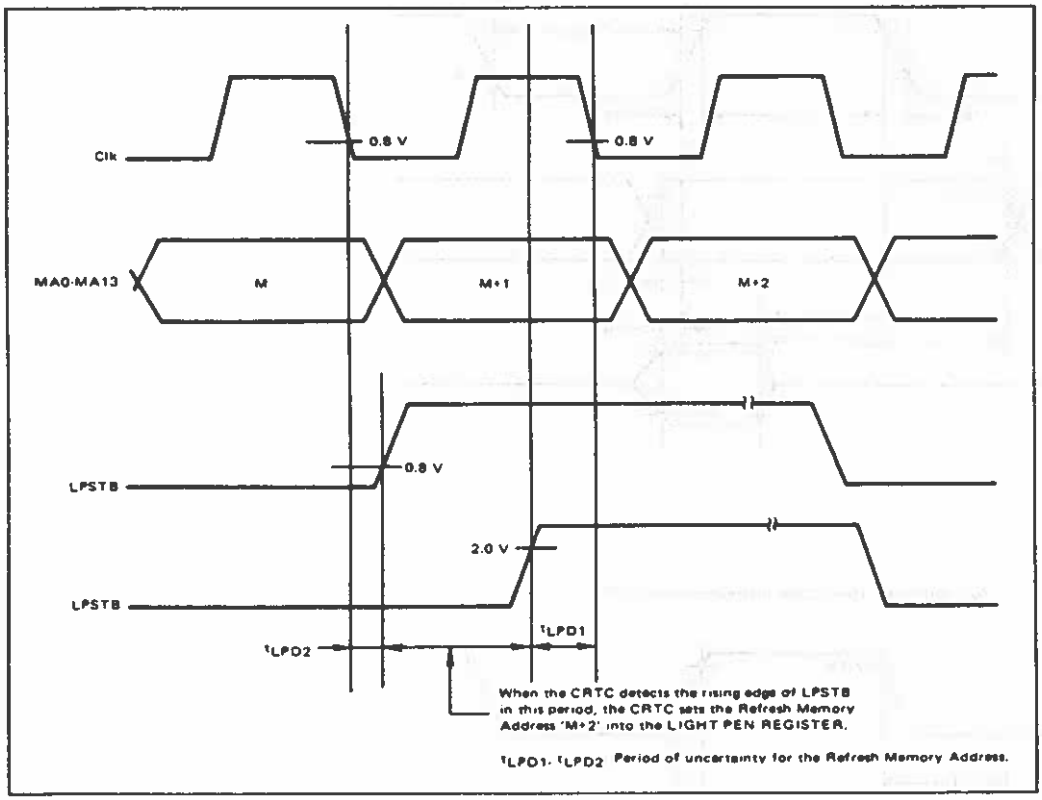
MC6845

FIGURE 3 – CRTC TIMING CHART



MC6845

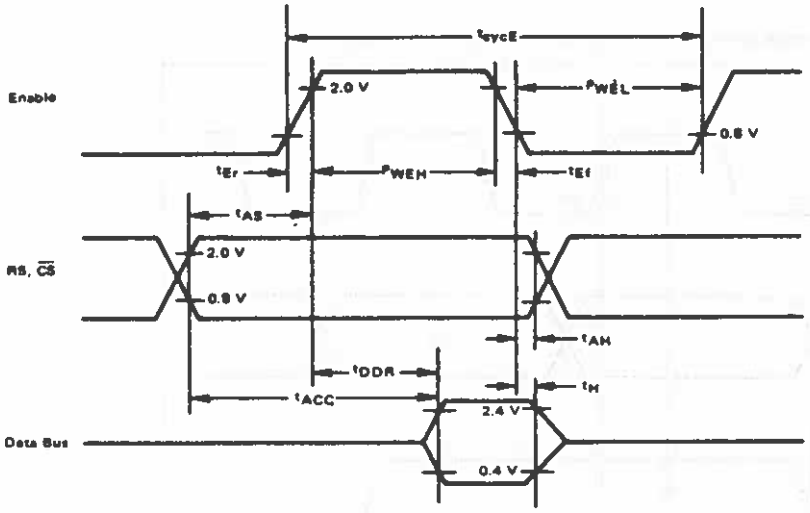
FIGURE 4 – RELATION BETWEEN LPSTB AND REFRESH MEMORY ADDRESS



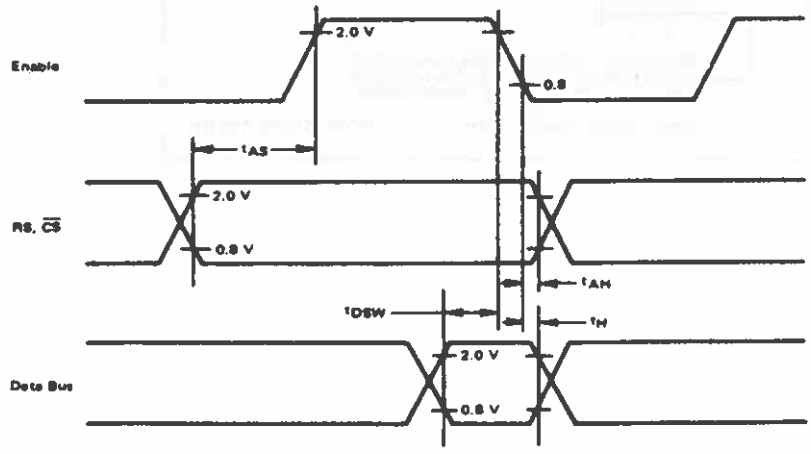
MC6845

FIGURE 5 – BUS TIMING CHART

5a – Bus Read Timing (Read Information From CRTC)



5b – Bus Write Timing (Write Information Into CRTC)



MC6845

FIGURE 6 – BUS TIMING TEST LOAD

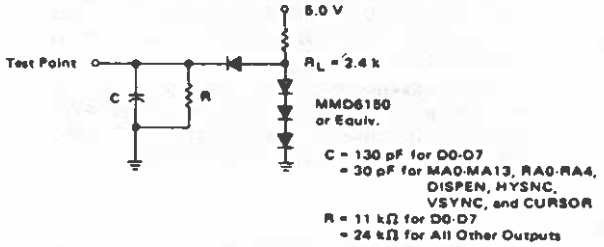
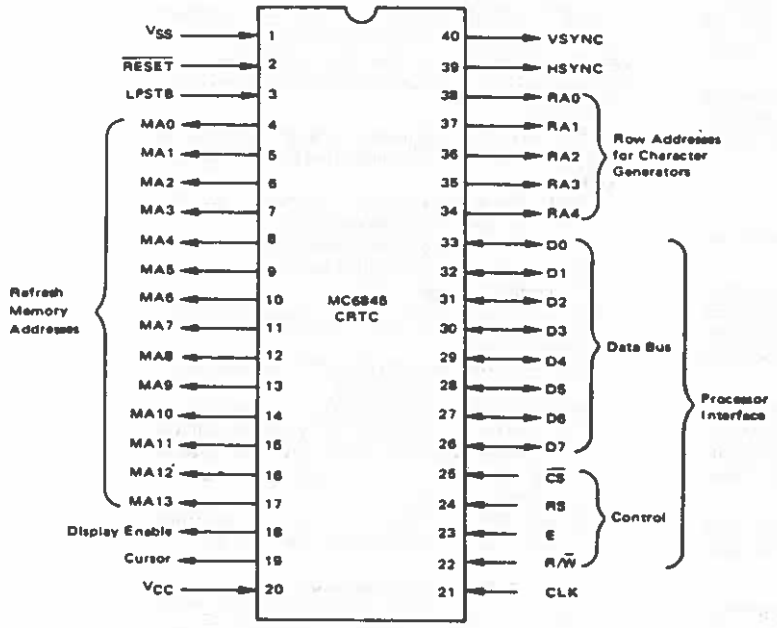


FIGURE 7 – PIN ASSIGNMENT



MC6845

PIN DESCRIPTION

PROCESSOR INTERFACE

The CRTC interfaces to a processor bus on the bidirectional data bus (D0-D7) using $\overline{CS}$, RS, E, and R/$\overline{W}$ for control signals.

Data Bus (D0-D7) – The bidirectional data lines (D0-D7) allow data transfers between the CRTC internal Register File and the processor. Data bus output drivers are 3-state buffers which remain in the high impedance state except when the processor performs a CRTC read operation. A high level on a data pin is a logical "1."

Enable (E) – The Enable signal is a high impedance TTL/MOS compatible input which enables the data bus input/output buffers and clocks data to and from the CRTC. This signal is usually derived from the processor clock, and the high to low transition is the active edge.

Chip Select ($\overline{CS}$) – The $\overline{CS}$ line is a high impedance TTL/MOS compatible input which selects the CRTC when low to read or write the internal Register File. This signal should only be active when there is a valid stable address being decoded from the processor.

Register Select (RS) – The RS line is a high impedance TTL/MOS compatible input which selects either the Address Register (RS = "0") or one of the Data Registers (RS = "1") of the internal Register File.

Read/Write (R/$\overline{W}$) – The R/$\overline{W}$ line is a high impedance TTL/MOS compatible input which determines whether the internal Register File gets written or read. A write is active low ("0").

CRT CONTROL

The CRTC provides horizontal sync (HS), vertical sync (VS), and Display Enable signals.

Vertical Sync (V SYNC) – This TTL compatible output is an active high signal which drives the monitor directly or is fed to Video Processing Logic for composite generation. This signal determines the vertical position of the displayed text.

Horizontal Sync (H SYNC) – This TTL compatible output is an active high signal which drives the monitor directly or is fed to Video Processing Logic for composite generation. This signal determines the horizontal position of the displayed text.

Display Enable – This TTL compatible output is an active high signal which indicates the CRTC is providing addressing in the active Display Area.

REFRESH MEMORY/CHARACTER GENERATOR ADDRESSING

The CRTC provides Memory Addresses (MA0-MA13) to scan the Refresh RAM. Also provided are Raster Addresses (RA0-RA4) for the character ROM.

Refresh Memory Addresses (MA0-MA13) – These 14 outputs are used to refresh the CRT screen with pages of data located within a 16K block of refresh memory. These outputs drive a TTL load and 30pF. A high level on MA0-MA13 is a logical "1."

Raster Addresses (RA0-RA4) – These 5 outputs from the internal Raster Counter address the Character ROM for the row of a character. These outputs drive a TTL load and 30pF. A high level (on RA0-RA4) is a logical "1."

OTHER PINS

Cursor – This TTL compatible output indicates Cursor Display to external Video Processing Logic. Active high signal.

Clock (CLK) – The CLK TTL/MOS compatible input is used to synchronize all CRT control signals. An external dot counter is used to derive this signal which is usually the character rate in an alphanumeric CRT. The active transition is high to low.

Light Pen Strobe (LPSTR) – This high impedance TTL/MOS compatible input latches the current Refresh Addresses in the Register File. Latching is on the low to high edge and is synchronized internally to character clock.

V_{CC}, Gnd

$\overline{RES}$ – The $\overline{RES}$ input is used to Reset the CRTC. An input low level on $\overline{RES}$ forces CRTC into following status:

(A) All the counters in CRTC are cleared and the device stops the display operation.
(B) All the outputs go down to low level.
(C) Control registers in CRTC are not affected and remain unchanged.

This signal is different from other M6800 family in the following functions:

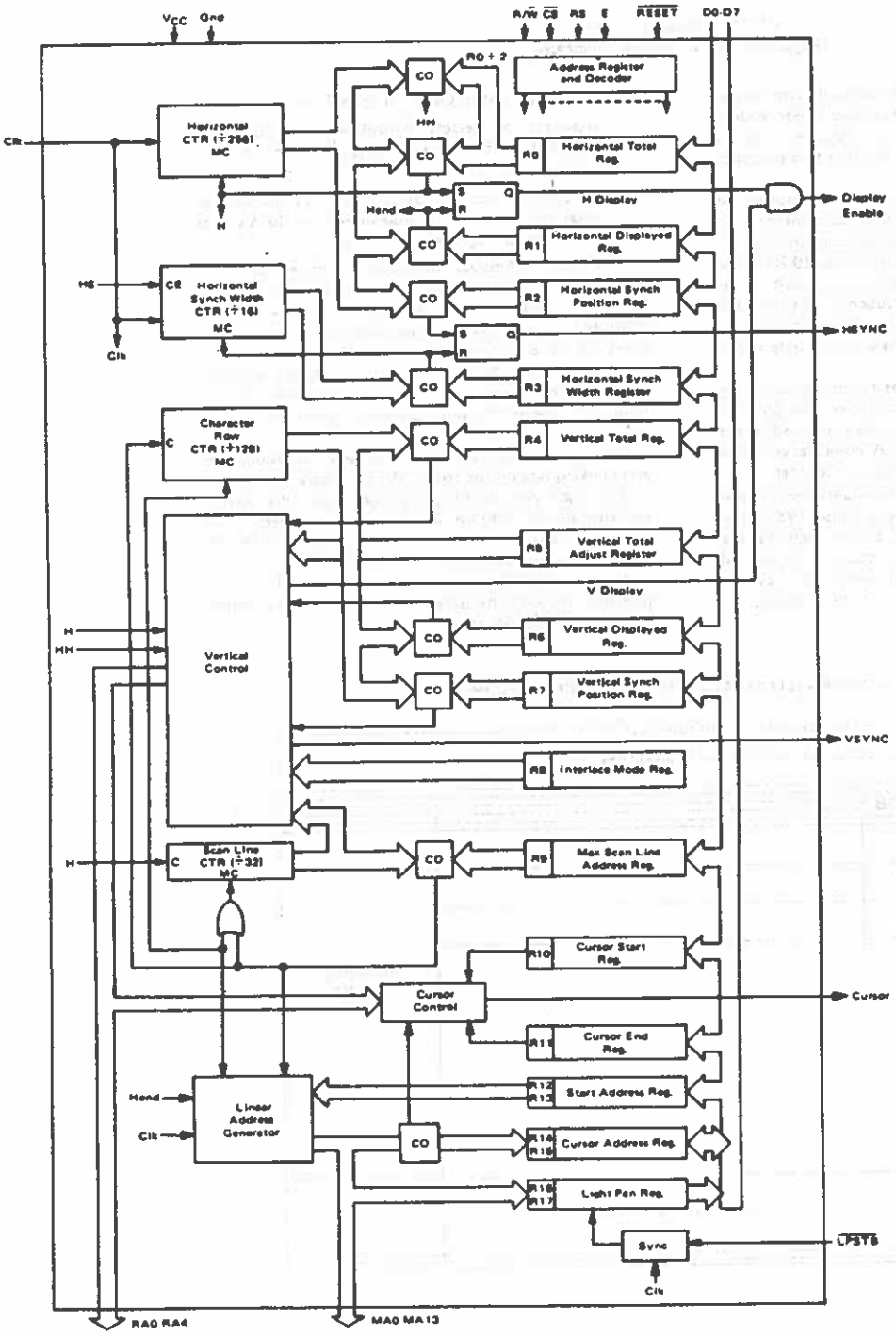
(A) $\overline{RES}$ signal has capability of reset function only when LPSTB is at low level.
(B) After $\overline{RES}$ has gone down to low level, output signals of MA0-MA13 and RA0-RA4, synchronizing with CLK low level, goes down to low level. (At least 1 cycle CLK signal is necessary for reset.)
(C) The CRTC starts the Display operation immediately after the release of $\overline{RES}$ signal.

TABLE 1 – CRTC Operating Mode

$\overline{RES}$	LPSTB	OPERATING MODE
0	0	Reset
0	1	Test Mode
1	0	Normal Mode
1	1	Normal Mode

MC6845

FIGURE 8 – CRTC FUNCTIONAL BLOCK DIAGRAM



CRTC DESCRIPTION
(Figure 8: CRTC Block Diagram)

The CRTC consists of programmable horizontal and vertical timing generators, programmable linear address register, programmable cursor logic, light pen capture register, and control circuitry for interface to a processor bus.

All CRTC timing is derived from CLK, usually the output of an external dot rate counter. Coincidence (CO) circuits continuously compare counter contents to the contents of the programmable register file, R0-R17. For horizontal timing generation, comparisons result in: 1) Horizontal sync pulse (HS) of a frequency, position, and width determined by the registers, 2) Horizontal Display Signal of a frequency, position, and duration determined by the registers.

The Horizontal counter produces H clock which drives the Scan Line Counter and Vertical Control. The contents of the Raster Counter are continuously compared to the Max Scan Line Address Register. A coincidence resets the Raster Counter and clocks the Vertical Counter.

Comparisons of Vertical Counter contents and Vertical Registers result in: 1) Vertical sync pulse (VS) of a frequency and positon determined by the registers—the width is fixed at 16 raster lines in the vertical control section and is not programmable, 2) Vertical Display of a frequency and position determined by the registers.

The Vertical Control Logic has other functions.

1. Generate row selects, RA0-RA4, from the Raster Count for the corresponding interlace or non-interlace modes.
2. Extend the number of scan lines in the vertical total by the amount programmed in the Vertical Total Adjust Register.

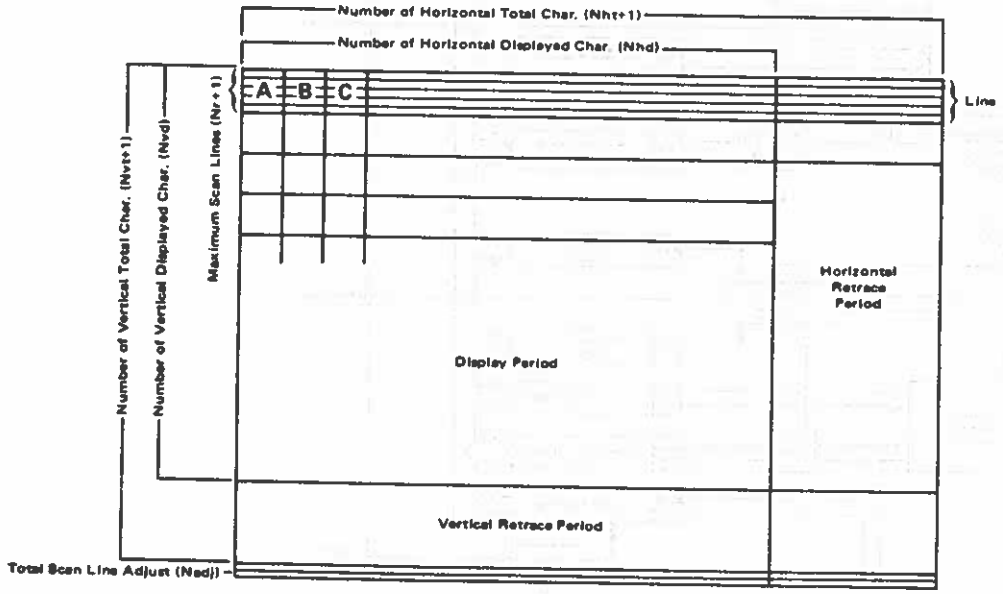
The Linear Address Generator is driven by CLK and locates the relative positions of characters in memory with their positions on the screen. Fourteen lines, MA0-MA13, are available for addressing up to four pages of 4K characters, 8 pages of 2K characters, etc. Using the Start Address Register, hardware scrolling through 16K characters is possible. The Linear Address Generator repeats the same sequence of addresses for each scan line of a character row.

The cursor logic determines the cursor location, size, and blinking rate on the screen. All are programmable.

The light pen strobe going high causes the current contents of the Address Counter to be latched in the Light Pen Register. The contents of the Light Pen Register are subsequently read by the Processor.

Internal CRTC registers are programmed by the processor through the data bus, D0-D7, and the control signals—R/$\overline{W}$, $\overline{CS}$, RS and E.

FIGURE 9 – ILLUSTRATION OF THE CRT SCREEN FORMAT



MC6845

REGISTER FILE DESCRIPTION
(See Table 2)

Nineteen registers in the CRTC can be accessed by means of the data bus. Register addressing and lengths are shown in Table 2.

ADDRESS REGISTER

The Address Register is a 5 bit write-only register used as an "indirect" or "pointer" register. Its contents are the address of one of the other 18 registers in the file. When RS and $\overline{CS}$ are low, the Address Register itself is addressed. When RS is high, the Register File is accessed.

HORIZONTAL TIMING REGISTERS R0, R1, R2, and R3

Figure 9 shows the visible display area of a typical CRT monitor giving the point of reference for horizontal registers as the left most displayed character position. Horizontal registers are programmed in "character time" units with respect to the reference.

Horizontal Total Register (R0) – This 8 bit write-only register determines the horizontal frequency of HS. It is the total of displayed plus non-displayed character time units minus one.

Horizontal Displayed Register (R1) – This 8 bit write-only register determines the number of displayed characters per horizontal line.

Horizontal Sync Position Register (R2) – This 8 bit write-only register determines the horizontal sync position on the horizontal line.

Horizontal Sync Width Register (R3) – This 4 bit write-only register determines the width of the HS pulse. It may not be apparent why this width needs to be programmed. However, consider that all timing widths must be programmed as multiples of the character clock period which varies. If HS width were fixed as an integral number of character times, it would vary with character rate and be out of tolerance for certain monitors. The rate programmable feature allows compensating HS width.

VERTICAL TIMING REGISTERS R4, R5, R6, R7, R8, and R9

The point of reference for vertical registers is the top character position displayed. Vertical registers are programmed in character row times or scan line times.

Vertical Total Register (R4) and Vertical Total Adjust Register (R5) – The vertical frequency of VS is determined by both R4 and R5. The calculated number of character line times is usually an integer plus a fraction to get exactly a 50 or 60 Hz vertical refresh rate. The integer number of character line times minus one is programmed in the 7 bit write-only Vertical Total Register; the fraction is programmed in the 5 bit write-only Vertical Scan Adjust Register as a number of scan line times.

Vertical Displayed Register (R6) – This 7 bit write-only register determines the number of displayed character rows on the CRT screen, and is programmed in character row times.

Vertical Sync Position (R7) – This 7 bit write-only register determines the vertical sync position with respect

TABLE 2 – CRTC INTERNAL REGISTER ASSIGNMENT

$\overline{CS}$	RS	Address Register 4	3	2	1	0	Register #	Register File	Program Unit	Read	Write	Number of Bits 7	6	5	4	3	2	1	0
1	X	X	X	X	X	X	X	-		-	-								
0	0	X	X	X	X	X	X	Address Register	-	No	Yes								
0	1	0	0	0	0	0	R0	Horizontal Total	Char	No	Yes								
0	1	0	0	0	0	1	R1	Horizontal Displayed	Char	No	Yes								
0	1	0	0	0	1	0	R2	H Sync Position	Char	No	Yes								
0	1	0	0	0	1	1	R3	H Sync Width	Char	No	Yes								
0	1	0	0	1	0	0	R4	Vertical Total	Char Row	No	Yes								
0	1	0	0	1	0	1	R5	V Total Adjust	Scan Line	No	Yes								
0	1	0	0	1	1	0	R6	Vertical Displayed	Char Row	No	Yes								
0	1	0	0	1	1	1	R7	V Sync Position	Char Row	No	Yes								
0	1	0	1	0	0	0	R8	Interlace Mode	-	No	Yes								
0	1	0	1	0	0	1	R9	Max Scan Line Address	Scan Line	No	Yes								
0	1	0	1	0	1	0	R10	Cursor Start	Scan Line	No	Yes		B	P				(Note 1)	
0	1	0	1	0	1	1	R11	Cursor End	Scan Line	No	Yes								
0	1	0	1	1	0	0	R12	Start Address (H)	-	No	Yes								
0	1	0	1	1	0	1	R13	Start Address (L)	-	No	Yes								
0	1	0	1	1	1	0	R14	Cursor (H)	-	Yes	Yes								
0	1	0	1	1	1	1	R15	Cursor (L)		Yes	Yes								
0	1	1	0	0	0	0	R16	Light Pen (H)		Yes	No								
0	1	1	0	0	0	1	R17	Light Pen (L)		Yes	No								

Note (1) Bit 5 of the Cursor Start Raster Register is used for blink period control, and Bit 6 is used to select blink or non blink

to the reference. It is programmed in character row times.

Interlace Mode Register (R8) – This 2 bit write-only register controls the raster scan mode (see Figure 11). When bit 0 and bit 1 are reset, or bit 0 is reset and bit 1 set, the non-interlace raster scan mode is selected. Two interlace modes are available. Both are interlaced 2 fields per frame. When bit 0 is set and bit 1 is reset, the interlace sync raster scan mode is selected. Also when bit 0 and bit 1 are set, the interlace sync and video raster scan mode is selected.

Maximum Scan Line Address Register (R9) – This 5 bit write-only register determines the number of scan lines per character row including spacing. The programmed value is a max address and is one less than the number of scan lines.

OTHER REGISTERS

Cursor Start Register (R10) – This 7 bit write-only register controls the cursor format (see Figure 10). Bit 5 is the blink timing control. When bit 5 is low, the blink frequency is 1/16 of the vertical field rate, and when bit 5 is high, the blink frequency is 1/32 of the vertical field rate. Bit 6 is used to enable a blink. The cursor start scan line is set by the lower 5 bits.

Cursor End Register (R11) – This 5 bit write-only register sets the cursor end scan line.

Start Address Register (H & L) (R12, R13) – Start Address Register is a 14 bit write-only register which determines the first address put out as a refresh address after vertical blanking. It consists of an 8 bit lower register, and a 6 bit higher register.

Light Pen Register (H & L) (R16, R17) – This 14 bit read-only register is used to store the contents of the Address Register (H & L) when the LPSTB input pulses high. This register consists of an 8 bit lower and 6 bit higher register.

Cursor Register (H & L) (R14, R15) – This 14 bit read/write register stores the cursor location. This register consists of an 8 bit lower and 6 bit higher register.

CURSOR

The Cursor Start and End Registers allow a cursor of up to 32 scan lines in height to be placed on any scan lines of the character block as shown in Figure 10. Using Bits 5 & 6 of the Cursor Start Register, the cursor is programmed with blink periods of 16 or 32 times the field period. Optional non-blink and non-display modes can also be selected. When an external 2X blink on characters is required, it may be necessary to perform cursor blink externally as well so that both blink rates are synchronized. Note that an invert/non-invert cursor is easily implemented by programming the CRTC for blinking cursor and externally inverting the video signal with an exclusive-OR.

The cursor is positioned by changing the contents of registers R14 and R15. The cursor can be placed at any of 16K character positions, thus facilitating hardware paging and scrolling through memory without loss of the cursor's original position.

INTERLACE/NON-INTERLACE DISPLAY MODES

An illustration of the 3 raster scan modes of operation is shown in Figure 11. Normal sync mode is non-interlace. In this mode, each scan line is refreshed at the vertical field rate (e.g., 50 or 60 Hz). Frame time is divided into even and odd alternating fields. The horizontal and vertical timing relationship results in the displacement of scan lines in the odd field with respect to the even field. When the same information is painted in both fields, the mode is called "Interlace Sync;" this is a useful mode for enhancing readability by filling in a character. When the even lines of a character are displayed in the even field and the odd lines in the odd field, the mode is called "Interlace Sync and Video." This last mode effectively doubles the character density on a monitor of a given bandwidth. The disadvantage of both interlace modes is an apparent flicker effect, which can be reduced by careful monitor design.

There are restrictions on the programming of CRTC registers for interlace operation:

1) Horizontal total character count, N_{ht} must be odd (i.e., an even number of character times)
2) For Interlace Sync and Video mode only, the max scan line address, N_{sl}, must be odd (i.e., an even number of scan lines)
3) For Interlace Sync and Video mode only, the Vertical Displayed Total characters must be even. The programmed number, N_{vd}, must be *one-half* the actual number required.
4) For Interlace Sync & Video mode only, the Cursor START and Cursor End Registers must both be even or both odd.

LIGHT PEN

The contents of the CRTC Address Counter are strobed into R16/R17 Light Pen Registers on the next high to low CLK transition after LPSTB goes high. In most systems, the light pen signal would also cause a processor interrupt routine to read R16/R17. Slow light pen response requires the processor software to modify the captured address read from R16/R17 by a calibration factor.

PROGRAMMING CONSIDERATIONS

Initialization – Registers R0-R15 must be initialized after power is turned on. The processor normally loads the CRTC registers sequentially from a firmware table. Henceforth, R0-R11 are not changed in most systems. The 6800 program in Table 3 and Figure 12 shows a typical CRTC initialization.

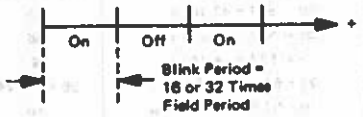
Hardware Scrolling – Registers R12/R13 contents determine which memory location is the first displayed character on the screen. Since the CRTC Linear Address Generator counts from this beginning count, the displayed portion of the screen may be a window on any continuous string of characters within a 16K block or refresh memory. By centering the R12/R13 pointer in the middle of the available memory space, scrolling up or down is possible . . . by line, page, or character.

MC6845

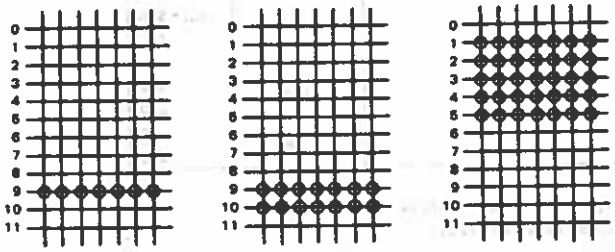
FIGURE 10 – CURSOR CONTROL

Cursor Start Register

B Bit 6	P Bit 5	Cursor Display Mode
0	0	Non-Blink
0	1	Cursor Non-Display
1	0	Blink, 1/16 Field Rate
1	1	Blink, 1/32 Field Rate



Example of Cursor Display Mode



Cursor Start Adr. = 9
Cursor End Adr. = 9

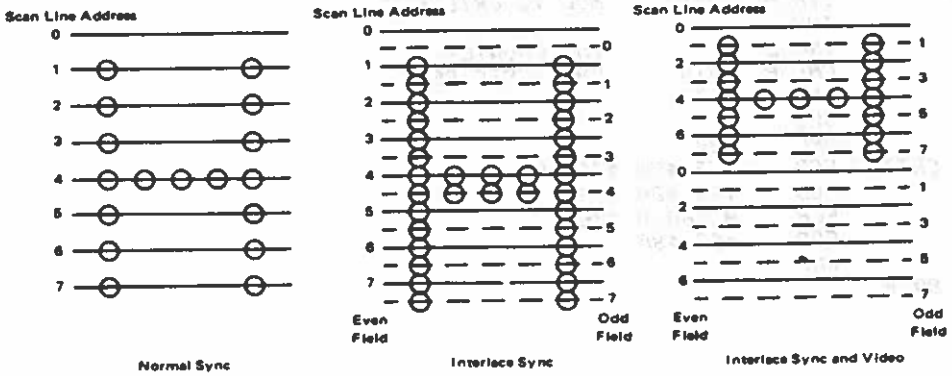
Cursor Start Adr. = 9
Cursor End Adr. = 10

Cursor Start Adr. = 1
Cursor End Adr. = 5

FIGURE 11 – INTERFACE CONTROL

Interlace Mode Register

Bit 1	Bit 0	Mode
0 1	0 0	Normal Sync Mode (Non-Interlace)
0	1	Interlace Sync Mode
1	1	Interlace Sync & Video Mode



MC6845

TABLE 3 – Typical 80 x 24 Screen Format Initialization of CRTC

Reg. #	Register File	Program Unit	Calculation*	Programmed Value	
				Decimal	Hex
R0	H Total	T_c	102 x .527 = 53.76 µs	102 - 1 = 101	N_{ht} = $65
R1	H Displayed	T_c	80 x .527 = 42.16 µs	80	N_{hd} = $50
R2	H Sync Position	T_c	86 x .527 = 45.32 µs	86	N_{hsp} = $56
R3	H Sync Width	T_c	9 x .527 = 4.74 µs	9	N_{hsw} = $09
R4	V Total	T_{cr}	25 x 645.12 = 16.13 ms	25 - 1 = 24	N_{vt} = $18
R5	V Total Adjust	T_{sl}	10 x 53.76 = .54 ms	10	N_{adj} = $0A
R6	V Displayed	T_{cr}	24 x 645.12 = 15.48 ms	24	N_{vd} = $18
R7	V Sync Position	T_{cr}	24 x 645.12 = 15.48 ms	24	N_{vsp} = $18
R8	Interlace Mode	–		–	$00
R9	Max Scan Line Address	T_{sl}		11	N_{sl} = $0B
R10	Cursor Start	T_{sl}		0	$00
R11	Cursor End	T_{sl}		11	$0B
R12	Start Address (H)	–		128	$00
R13	Start Address (L)	–			$80
R14	Cursor (H)	–		128	$00
R15	Cursor (L)	–			$80

Clock Period = T_c = .527 µs
Scan Line Period = T_{sl} = $(N_{ht} + 1) \times T_c$ = 102 x .527 µs = 53.76 µs
Character Row Period = $T_{cr} = N_{sl} \times T_{sl}$ = 12 x 53.76 µs = 645.12 µs

*These are typical values for the Motorola M3000 Monitor; values may vary for other monitors.

FIGURE 12 – INITIALIZATION OF CRTC FOR 80x24 SCREEN FORMAT IN TABLE 3

```
PAGE  001   CRTINT

00001                          NAM    CRTINT
00002 0000                     ORG    $0
00003 0000 5F                  CLR B                CLEAR COUNTER
00004 0001 CE 0020             LDX    #$20
00005 0004 F7 9000 CRTII       STA B  $9000        CRTC ADDR REG
00006 0007 A6 00               LDA A  0,X
00007 0009 B7 9001             STA A  $9001        ACC TO CRTC REG
00008 000C 08                  INX
00009 000D 5C                  INC B                INC COUNTER
00010 000E C1 10               CMP B  #$10         LAST CRTC REG?
00011 0010 26 F2               BNE    CRTII
00012 0012 3F                  SWI
00013 0020                     ORG    $20
00014 0020 65          CRTTAB  FCB    $65,$50,$56,$9
00015 0024 18                  FCB    $18,$0A,$18,$18
00016 0028 00                  FCB    0,$0B,0,$0B
00017 002C 0080                FDB    $80,$80
00018      0000                END
CRTII  0004 CRTTAB 0020

TOTAL ERRORS 00000
```

OPERATION OF THE CRTC

Timing Chart of the CRT Interface Signals — Timing charts of CRT interface signals are illustrated in this section with the aid of programmed example of the CRTC. When values listed in Table 4 are programmed into CRTC control registers, the device provides the outputs as shown in the Timing Diagrams (Figures 13 through 15). The screen format of this example is shown in Figure 9. Figure 16 is an illustration of the relation between Refresh Memory Address (MA0-MA13), Raster Address (RA0-RA4) and the position on the screen. In this example, the start address is assumed to be "0".

TABLE 4 — Values Programmed Into CRTC Registers

Reg. #	Register Name	Value	Programmed Value
R0	H. Total	$N_{ht}+1$	N_{ht}
R1	H. Displayed	N_{hd}	N_{hd}
R2	H. Sync Position	N_{hsp}	N_{hsp}
R3	H. Sync Width	N_{hsw}	N_{hsw}
R4	V. Total	$N_{vt}+1$	N_{vt}
R5	V. Scan Line Adjust	N_{adj}	N_{adj}
R6	V. Displayed	N_{vd}	N_{vd}
R7	V. Sync Position	N_{vsp}	N_{vsp}
R8	Interlace Mode		
R9	Max. Scan Line Address	N_{sl}	N_{sl}
R10	Cursor Start		
R11	Cursor End		
R12	Start Address (H)	0	
R13	Start Address (L)	0	
R14	Cursor (H)		
R15	Cursor (L)		
R16	Light Pen (H)		
R17	Light Pen (L)		

MC6845

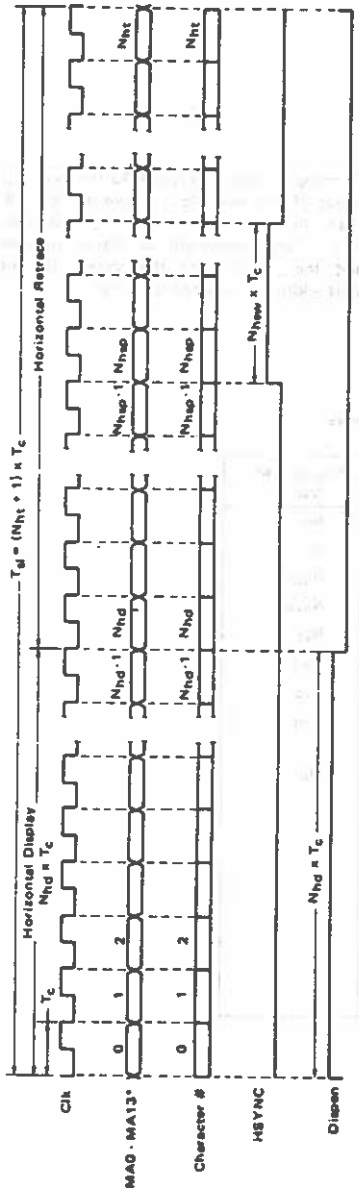


FIGURE 13 – CRTC HORIZONTAL TIMING

*Timing is shown for first displayed scan row only. See Chart in Figure 16 for other rows. The initial MA is determined by the contents of Start Address Register. R12/R13. Timing is shown for R12/R13 = 0.

MC6845

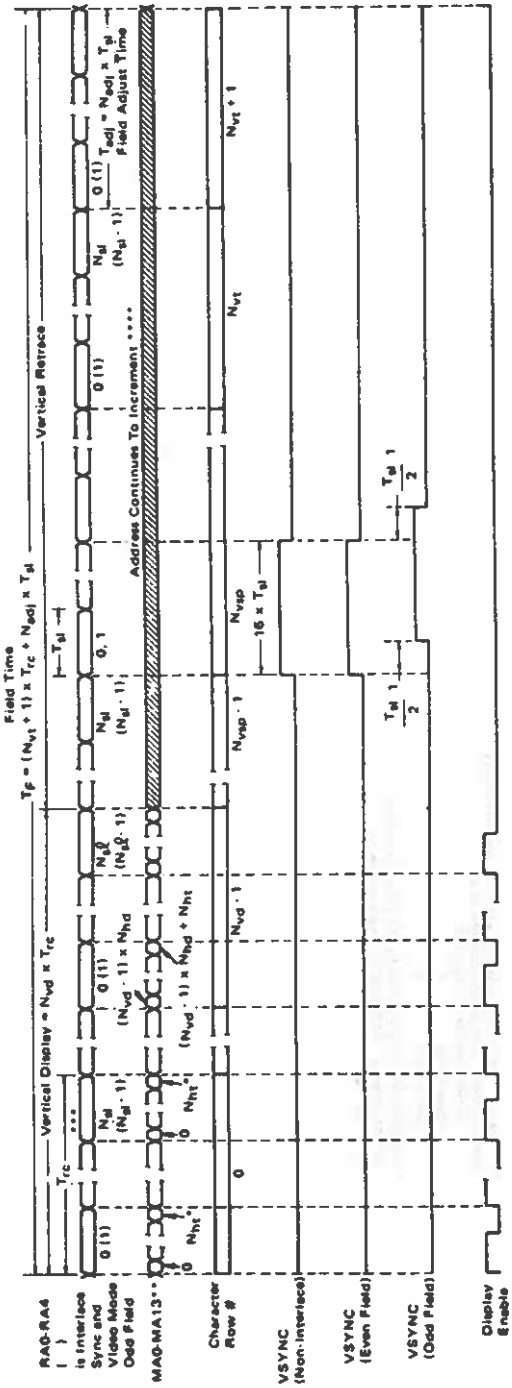


FIGURE 14 – CRTC VERTICAL TIMING

*N_{ht} must be an odd number for both interlace modes.
**Initial MA is determined by R12/R13 (Start Address Register), which is zero in this timing example.
***N_{sl} must be an odd number for Interlace Sync and Video Mode.
****The present CRTC freezes MA addresses at $N_{vd} \times N_{hd}$ during vertical retrace. A design change is pending to allow MA to free run during vertical retrace time.

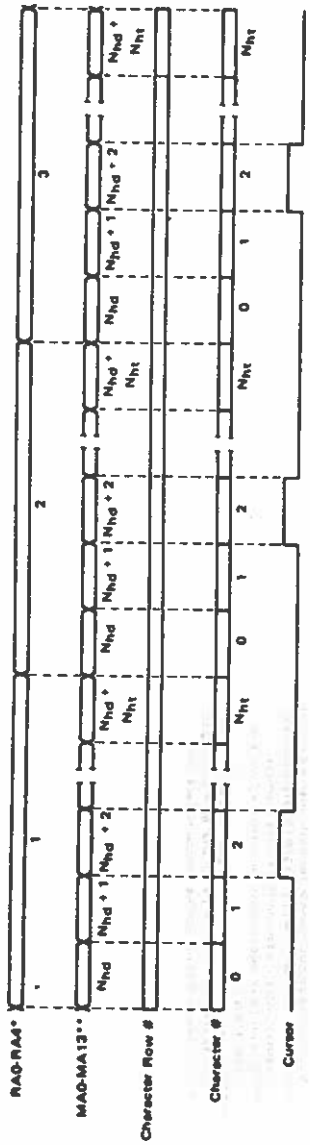


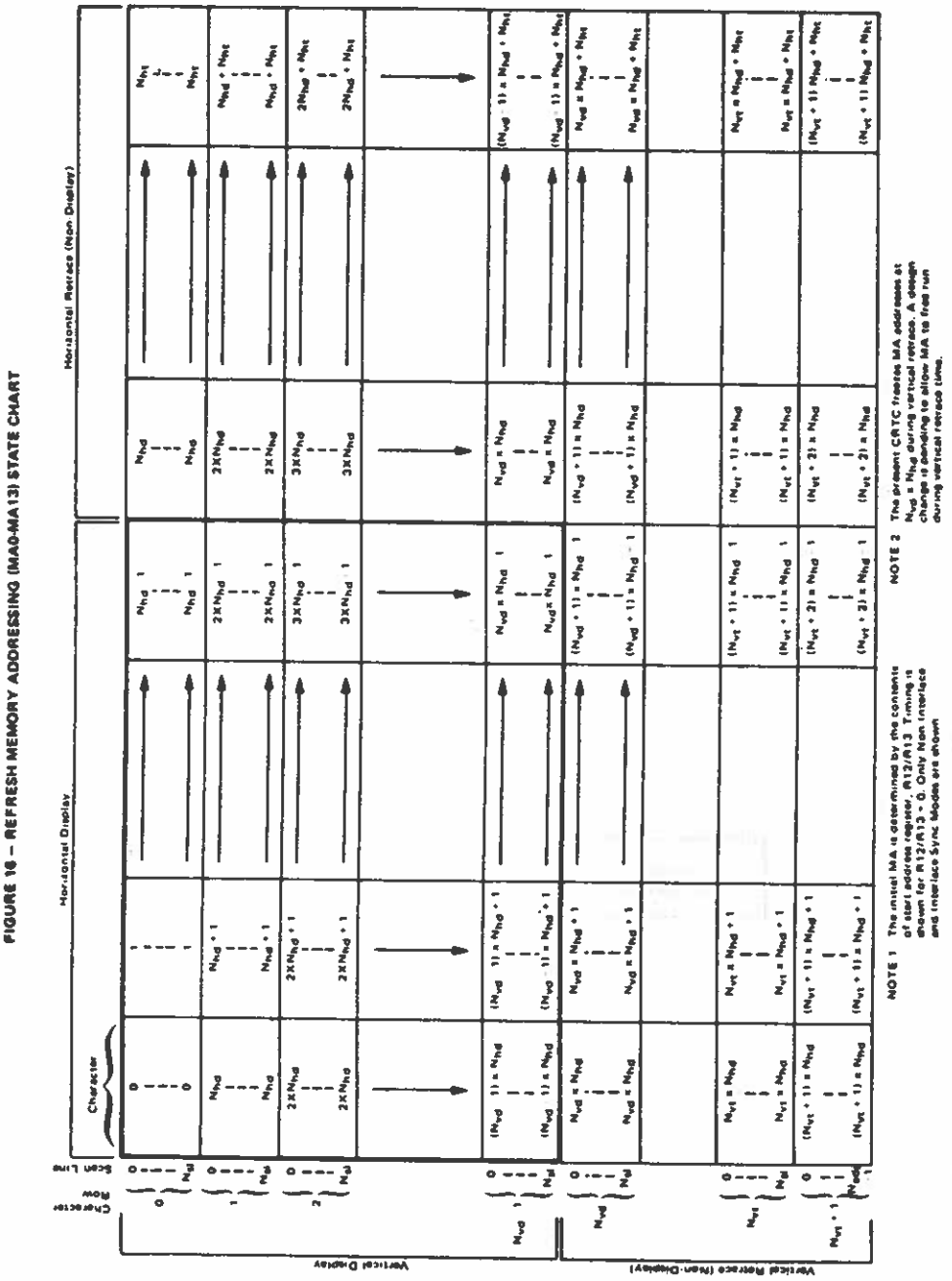
FIGURE 15 – CURSOR TIMING

*Timing is shown for non-interlace and interlace sync modes.
Example shown has cursor programmed as:
Cursor Register = $N_{hd}+2$
Cursor Start = 1
Cursor End = 3

**The initial MA is determined by the contents of Start Address Register, R12/R13. Timing is shown for R12/R13 = 0.

MC6845

FIGURE 16 – REFRESH MEMORY ADDRESSING (MA0-MA13) STATE CHART



NOTE 1 The initial MA is determined by the contents of start address register, R12/R13. Timing is shown for R12/R13 = 0. Only Non Interlace and Interlace Sync Modes are shown.

NOTE 2 The present CRTC freezes MA addresses at $N_{vd} \times N_{hd}$ during vertical retrace. A design change is pending to allow MA to free run during vertical retrace time.

Fiche technique :

96364

CRT 96364 A*
CRT 96364 B*

μPC FAMILY

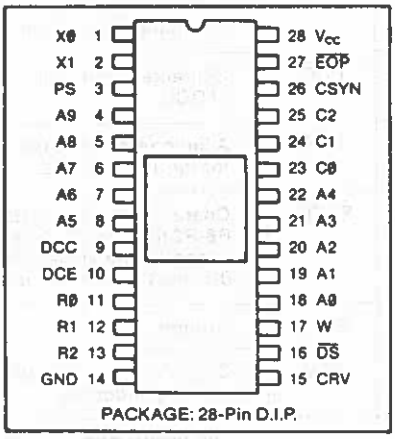
Preliminary Specifications

CRT Controller

FEATURES

- Single +5v power supply
- 16 line x 64 character display
- On chip sync oscillator
- Complete cursor control
- Automatic scrolling
- Erase functions built in
- Performs character entry during horizontal sync
- Internal blinking cursor
- Page linking logic built in
- LS-TTL compatible
- Compatible with CRT 8002, CRT 7004

PIN CONFIGURATION



PACKAGE: 28-Pin D.I.P.

GENERAL DESCRIPTION

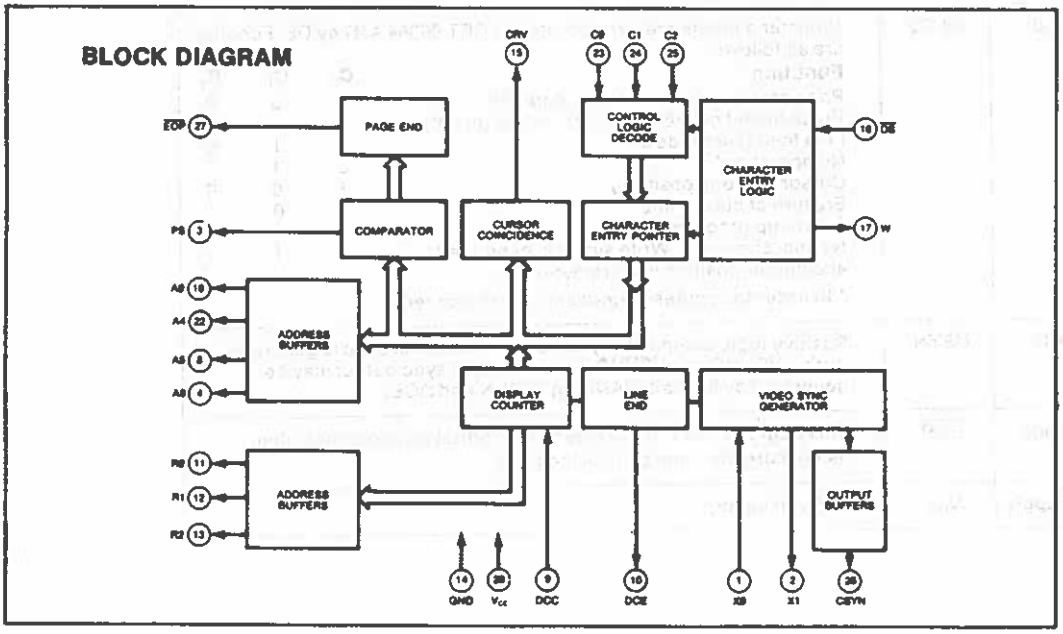
The CRT 96364 A/B is a CRT Controller which controls all of the functions associated with a 16 line x 64 character video display. Functions include CRT refresh, character entry, and cursor management.

The CRT 96364 A/B contains an internal oscillator which produces the composite sync output. The CRT 96364 B generates a 60 Hz vertical sync while the CRT 96364 A generates a 50 Hz vertical sync.

Standard functions such as ERASE PAGE, ERASE LINE, and ERASE TO END OF LINE make the CRT 96364 A/B easy to interface to any computer or microprocessor, or to use as a stand-alone video processor.

The CRT 96364 A/B requires only +5v power at less than 100 mA. It is manufactured in COPLAMOS® N channel silicon gate technology.

BLOCK DIAGRAM



*FOR FUTURE RELEASE

DESCRIPTION OF PIN FUNCTIONS

PIN NO.	NAME	SYMBOL	FUNCTION
1 2	Crystal in Crystal out	XØ X1	Pin one is the sync clock input. It may be driven directly from a TTL gate or from a parallel mode crystal connected between pins one and two. When a crystal is used, a 10 MΩ resistor should be connected in parallel. For standard 60 Hz line operation, a 1.018 MHz frequency source or crystal is required (with the CRT 96364B). For 50 Hz line operation, the CRT 96364A requires a 1.008 MHz crystal.
3	Page Select	PS	PS provides automatic page selection when two pages of memory are used. A "zero" output indicates selection of page 1; a logic "one" indicates page 2.
4-8	Memory Address	A9-A5	Upper order memory address lines; A6-A9 determine which lines of text are being refreshed or written. A5 along with AØ-A4 determine the character position.
9	Character Clock	DCC	Character clock input. Addresses are changed on the trailing edge of DCC.
10	Dot Clock Enable	DCE	A logic zero from DCE is used to inhibit oscillation of the dot clock for retrace blanking.
11-13	Row Address	RØ-R2	Character Generator row addresses. Blanks are generated by forcing RØ-R2 to "000". During character entry, R2 gates data into memory to control the erase function. Row addressing follows the sequence 0-1-2-3-4-5-6-7-0-0-0-0-increment text line-0-1-2-etc.
14	Ground	GND	Ground
15	Cursor	CRV	Cursor video output. Indicates cursor location by a 2 Hz blinking underline.
16	Data Strobe	$\overline{DS}$	The rising edge of $\overline{DS}$ strobes the appropriate CØ-C2 control word into the CRT 96364A/B.
17	Write	W	A positive going signal which indicates that the CRT 96364A/B is allowing a memory write. W is approximately 4 µs, and occurs during H sync. Memory address lines are latched at the cursor address during W.
18-22	Memory Address	AØ-A4	Lower order memory addresses. AØ-A4 plus A5 (pin 8) determine the character position.
23-25	Command Inputs	CØ-C2	Command inputs are strobed into the CRT 96364A/B by $\overline{DS}$. Functions are as follows: **Function** — C_2 C_1 C_0 Page erase and cursor home (top-left) — 0 0 0 Erase to end of line and return cursor (to left) — 0 0 1 Line feed (cursor down) — 0 1 0 No operation* — 0 1 1 Cursor left (one position) — 1 0 0 Erasure of cursor-line — 1 0 1 Cursor up (one position) — 1 1 0 Normal character. Write signal is generated and cursor position is incremented — 1 1 1 * In order to suppress non-displayed characters
26	Composite Sync	CSYN	Positive logic composite sync output. Horizontal sync is generated during VSYNC and $\overline{VSYNC}$ time. A vertical sync output may be generated by logically "ANDing" CSYN and DCE.
27	End of Page	$\overline{EOP}$	This output is used to increment an external page counter when using more than one page of memory.
28	Power Supply	V_{CC}	+ 5 volt supply.

OPERATION

The CRT 96364A/B provides all of the control functions required by a CRT display with a minimum of external circuitry.

The cursor and erase commands may be decoded from the data bus by a low cost 256 x 4 PROM. The CRT 96364A/B then provides the necessary cursor movement and gates the memory for writing or erasing. Erase is controlled by providing a write signal to RAM, and gating "zeros" to the RAM input bus. Use of an external PROM allows user selection of control words.

The RAM write command, "W", is generated during horizontal retrace. At this time, the RAM address is set to the cursor address. Immediately following the write command, the RAM addresses revert to refresh addressing and the cursor is shifted one character.

CURSOR

The cursor location is indicated by an alternating high on pin 15 (CRV) at row 7, and a low on pin 15 with RØ-R2 forced low at rows 0-6. These alternate at a 2 Hz rate. If CRV is used to force the display on, the result will be a blink of the cursor character position alternating with an underline at a 2 Hz rate.

CHARACTER ENTRY

When a Normal Character code (C2, C1, CØ = 1, 1, 1) and a Data Strobe are received, the write command will be generated during horizontal retrace. If, at the end of the horizontal retrace, the cursor is at the last position on a line, a carriage return and line feed will automatically occur. When the cursor is at the last position of the last line, a carriage return and up-scroll will automatically occur.

EXTRA FUNCTIONS

By using the fourth bit of the decoder PROM as a write enable signal, and properly programming the PROM, the additional commands of Home Cursor, Return Cursor, and Roll Screen may be generated. This is done by inhibiting the W signal to the page memory and inputting the control codes, respectively, of Page Erase and Home Cursor, Erase to end of line and Return Cursor, and Line Feed.

SCROLLING

Scrolling of the screen text will occur under any of the following characteristics:

1. Inputting a line feed command when the cursor is at the bottom line of the screen.
2. Inputting a character when the cursor is at the bottom right hand side of the screen.

Scrolling will result in the entire top line of the screen being erased and all of the remaining lines shifting up. Alternatively, a Roll (defined as all of the lines shifting up with the previous top line reappearing at the bottom of the screen) may be performed by inhibiting the write signal to the page memory as described in "Extra Functions."

MAXIMUM GUARANTEED RATINGS

Operating Temperature Range ..... 0°C to + 70°C
Storage Temperature Range ..... −55°C to + 150°C
Lead Temperature (soldering, 10 sec.) ..... +325°C
Positive Voltage on any Pin, with respect to ground ..... + 7.0V
Negative Voltage on any Pin, with respect to ground ..... −0.3V

Stresses above those listed may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied.

ELECTRICAL CHARACTERISTICS (T_A = 0 C to 70 C, Vcc = + 5V ± 5%, unless otherwise noted)

PRELIMINARY

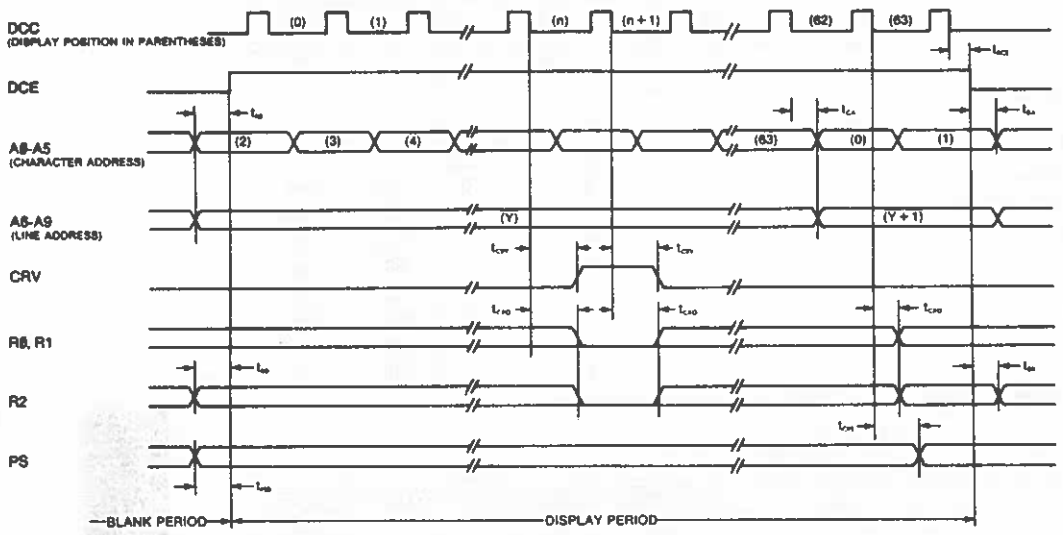
Parameter	Min.	Typ.	Max.	Unit	Comments
D.C. CHARACTERISTICS					
INPUT VOLTAGE LEVELS (except DCC)					
Low-level, V_{IL}			0.65	V	excluding DCC
High-level, V_{IH}	2.2			V	excluding DCC
INPUT VOLTAGE LEVELS—DCC					
Low-level, V_{IL}			0.65	V	
High-level, V_{IH}	3.5			V	
OUTPUT VOLTAGE LEVELS (DCE Only)					
Low-level, V_{OL}			0.4	V	I_{OL} = 1.9 mA
High-level, V_{OH}	2.2			V	I_{OH} = −100 µA
OUTPUT VOLTAGE LEVELS (except DCE)					
Low-level, V_{OL}			0.4	V	I_{OL} = 0.36 mA
High-level, V_{OH}	2.2			V	I_{OH} = −100 µA
INPUT CURRENT					
Low-level, I_{IL}			10	µA	$0 \leq V_{IN} \leq +5V$
INPUT CAPACITANCE					
All inputs, C_{IN} (except DCE)		5		pF	V_{IN} = GND
C_{IN} (DCC Only)		25		pF	V_{IN} = GND
POWER SUPPLY CURRENT					
I_{CC}		100	120	mA	

SECTION IV

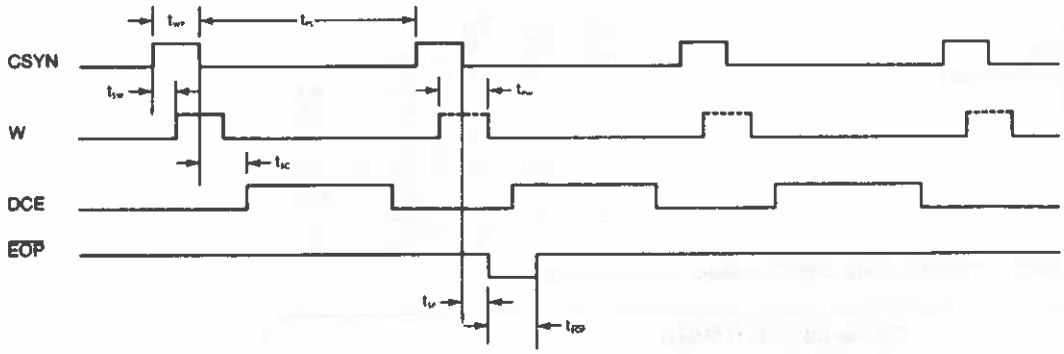
AC CHARACTERISTICS

PARAMETERS	SYMBOL	VALUES			UNIT
		MIN.	TYP.	MAX.	
Frequency of control clock DCC	f_{DCC}		1.6		MHz
Crystal Frequency CRT 96364 A	f_X		1.008		MHz
CRT 96364 B	f_X		1.018		MHz
DCC pulse width	t_{DCC}	200			ns
Rise and fall times	t_r t_f		20	40	ns
Refresh memory address access time	t_{CA}		200	250	ns
Character memory address access time	t_{CRO}		200	250	ns
PS access time (read)	t_{CPS}		300	1000	ns
CRV access time	t_{CRV}		200	250	ns
DCE access time (high to low)	t_{DCE}		100		ns
SYNC period	t_{PS}		64		µs
SYNC pulse width	t_{WP}		4		µs
DCE access time (low to high level)	t_{SC}		11		µs
$\overline{EOP}$ access time (high to low level)	t_{SP}		1	1.5	µs
W access time (low to high)	t_{SW}		500	1000	ns
W pulse width	t_{PW}		4		µs
$\overline{EOP}$ pulse width	$t_{\overline{EOP}}$		10		µs
Address to rising edge of DCE delay	t_{AD}	0		2.1	µs
Falling edge of DCE to Address delay	t_{DA}	0		1	µs
Row to rising edge of DCE delay	t_{RD}	0		2.1	µs
Falling edge of DCE to row delay	t_{DR}	0		1	µs
PS to rising edge of DCE delay	t_{PSD}	0			µs

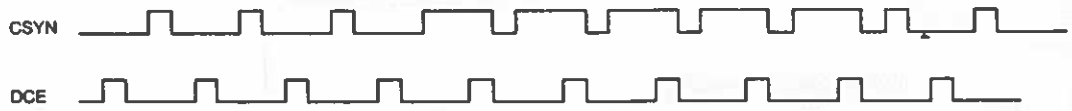
LINE TIMING



SYNC TIMING



FRAME TIMING



DATA INPUT TIMING

Asynchronous Operation

PARAMETER	SYMBOL	Value MIN	TYP	MAX	UNIT
DS Pulse Width	t_{PW}	0.5			µs
CØ-C2 Set Up Time	t_{COS}	1			µs
CØ-C2 Hold Time	t_{OSC}	90			µs
Minimum Strobe Period (Operation Execution Time)	t_{OS}				

FUNCTION	C2	C1	CØ		
Page Erase & Cursor Home	0	0	0	132	ms
Erase to End of Line & Return Cursor	0	0	1	4.2	ms
Line Feed (Cursor Down)	0	1	0	130*	µs
No Operation	0	1	1	80	µs
Cursor Left	1	0	0	80	µs
Erasure of Cursor Line	1	0	1	8.3	ms
Cursor Up	1	1	0	80	µs
Normal Character	1	1	1	130*	µs

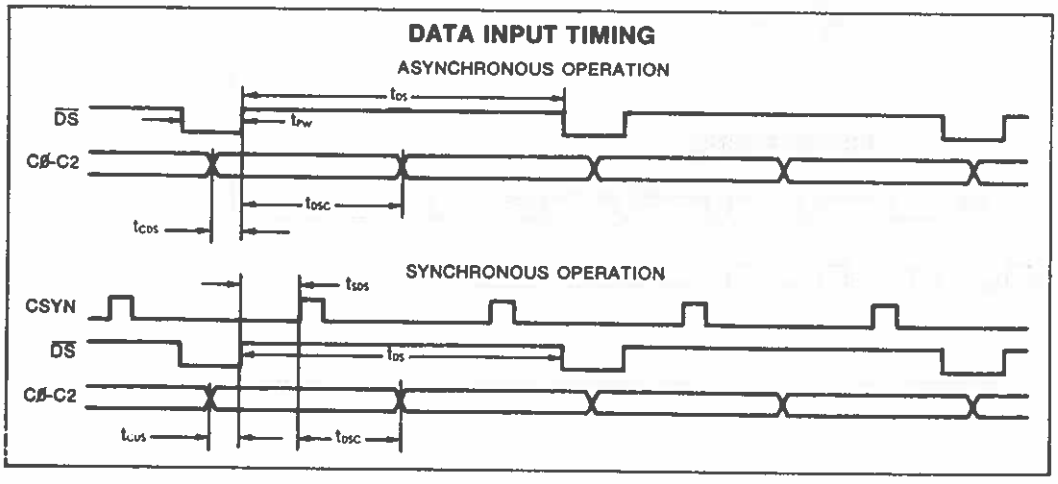
*Will increase to 8.3 ms when text scroll occurs. See "Scrolling" for conditions.

Synchronous Operation

PARAMETER	SYMBOL	Value MIN	TYP	MAX	UNIT
DS Pulse Width	t_{PW}	0.5			µs
C0-C2 Set-Up Time	t_{COS}	1			µs
C0-C2 Hold Time	t_{OSC}	16			µs
DS Set Up Time	t_{SOS}	1			µs
Minimum Strobe Period (Operation Execution Time)	t_{OS}				

FUNCTION	C2	C1	CØ		
Page Erase & Cursor Home	0	0	0	132	ms
Erase to End of Line & Return Cursor	0	0	1	4.2	ms
Line Feed (Cursor Down)	0	1	0	64*	µs
No Operation	0	1	1	64	µs
Cursor Left	1	0	0	64	µs
Erasure of Cursor Line	1	0	1	8.3	ms
Cursor Up	1	1	0	64	µs
Normal Character	1	1	1	64*	µs

*Will increase to 8.3 ms when text scroll occurs. See "Scrolling" for conditions.

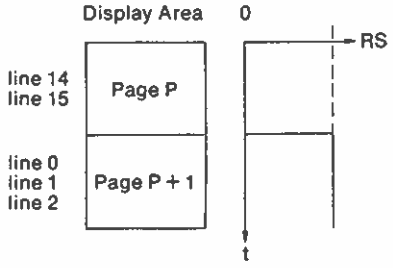


SECTION IV

MULTIPLE PAGE DISPLAY

When linking two or more pages, the $\overline{EOP}$ and RS signals may be used to allow a "moving window" text display.

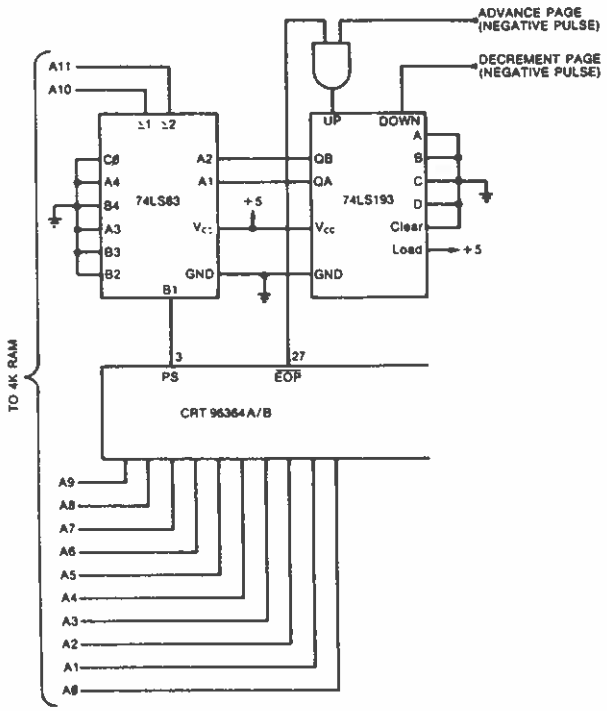
PS (Page Select) indicates the end of page location. If a scroll has occurred, PS will show the transition from the end of line 15 of page P and the beginning of line 0 of page P + 1.



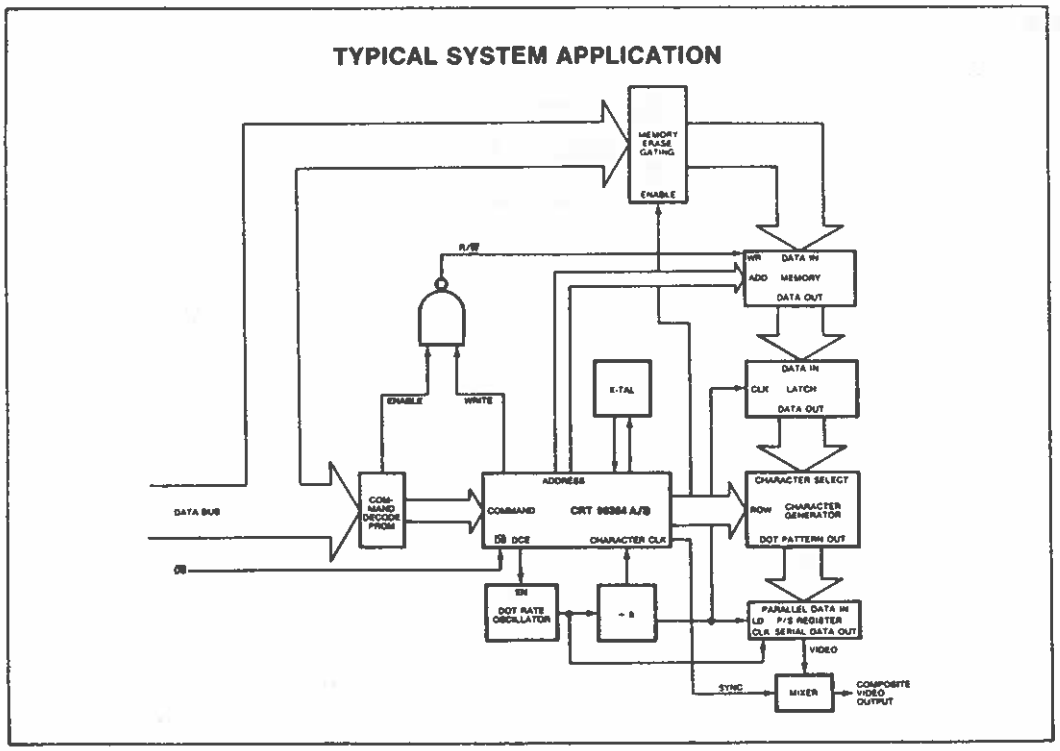
To properly maintain the memory address when displaying more than two pages, $\overline{EOP}$ pulses low at the point in time when page P is scrolled completely off the screen. At this time, RS will remain low for the entire frame since page P + 1 is now the only displayed page.

The circuit at the right will allow scrolling through 4 pages of memory.

4 PAGE DISPLAY



TYPICAL SYSTEM APPLICATION



Fiche technique :

6850

MOTOROLA

MC6850
1.0 MHz
MC68A50
1.5 MHz
MC68B50
2.0 MHz

ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (ACIA)

The MC6850 Asynchronous Communications Interface Adapter provides the data formatting and control to interface serial asynchronous data communications information to bus organized systems such as the MC6800 Microprocessing Unit.

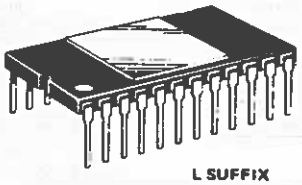
The bus interface of the MC6850 includes select, enable, read/write, interrupt and bus interface logic to allow data transfer over an 8-bit bi-directional data bus. The parallel data of the bus system is serially transmitted and received by the asynchronous data interface, with proper formatting and error checking. The functional configuration of the ACIA is programmed via the data bus during system initialization. A programmable Control Register provides variable word lengths, clock division ratios, transmit control, receive control, and interrupt control. For peripheral or modem operation three control lines are provided. These lines allow the ACIA to interface directly with the MC6860L 0-600 bps digital modem.

- Eight and Nine-Bit Transmission
- Optional Even and Odd Parity
- Parity, Overrun and Framing Error Checking
- Programmable Control Register
- Optional ÷1, ÷16, and ÷64 Clock Modes
- Up to 500 kbps Transmission
- False Start Bit Deletion
- Peripheral/Modem Control Functions
- Double Buffered
- One or Two Stop Bit Operation

MOS

(N-CHANNEL, SILICON-GATE)

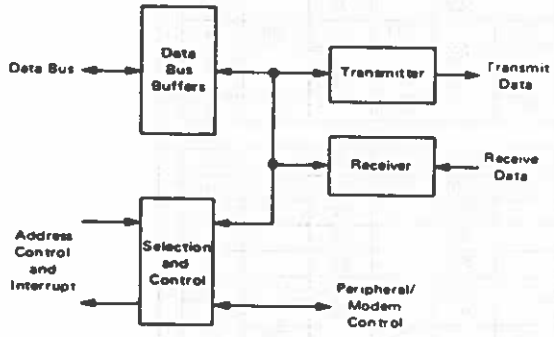
ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER



L SUFFIX
CERAMIC PACKAGE
CASE 716

NOT SHOWN: P SUFFIX
PLASTIC PACKAGE
CASE 709

MC6850 ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER BLOCK DIAGRAM



ORDERING INFORMATION

Speed	Device	Temperature Range
1.0 MHz	MC6850P, L	0 to +70°C
	MC6850CP, CL	-40 to +85°C
MIL-STD-883B MIL-STD-883C	MC6850BJCS MC6850CJCS	-55 to +125°C
1.5 MHz	MC68A50P, L	0 to +70°C
	MC68A50CP, CL	-40 to +85°C
2.0 MHz	MC68B50P, L	0 to +70°C

DS 9493 R1

MAXIMUM RATINGS

Rating		Symbol	Value	Unit
Supply Voltage		V_{CC}	-0.3 to +7.0	Vdc
Input Voltage		V_{in}	-0.3 to +7.0	Vdc
Operating Temperature Range		T_A	0 to +70	°C
Storage Temperature Range		T_{stg}	-55 to +150	°C
Thermal Resistance	Plastic	θ_{JA}	120	°C/W
	Ceramic		60	

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit.

ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 V ±5%, V_{SS} = 0, T_A = 0 to 70°C unless otherwise noted.)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	V_{SS} + 2.0	–	V_{CC}	Vdc
Input Low Voltage		V_{IL}	V_{SS} -0.3	–	V_{SS} + 0.8	Vdc
Input Leakage Current (V_{in} = 0 to 5.25 Vdc)	R/W, CS0, CS1, $\overline{CS2}$, Enable, RS, Rx D, Rx C, $\overline{CTS}$, $\overline{DCD}$	I_{in}	–	1.0	2.5	μAdc
Three-State (Off State) Input Current (V_{in} = 0.4 to 2.4 Vdc)	D0-D7	I_{TSI}	–	2.0	10	μAdc
Output High Voltage (I_{Load} = -205 μAdc, Enable Pulse Width ≤ 25 μs)	D0-D7	V_{OH}	V_{SS} + 2.4	–	–	Vdc
(I_{Load} = -100 μAdc, Enable Pulse Width ≤ 25 μs)	Tx Data, $\overline{RTS}$		V_{SS} + 2.4	–	–	
Output Low Voltage (I_{Load} = 1.6 mAdc, Enable Pulse Width ≤ 25 μs)		V_{OL}	–	–	V_{SS} + 0.4	Vdc
Output Leakage Current (Off State) (V_{OH} = 2.4 Vdc)	$\overline{IRQ}$	I_{LOH}	–	1.0	10	μAdc
Power Dissipation		P_D	–	300	525	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7	C_{in}	–	10	12.5	pF
	E, Tx Clk, Rx Clk, R/W, RS, Rx Data, CS0, CS1, $\overline{CS2}$, $\overline{CTS}$, $\overline{DCD}$		–	7.0	7.5	
Output Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	$\overline{RTS}$, Tx Data	C_{out}	–	–	10	pF
	$\overline{IRQ}$		–	–	5.0	
Minimum Clock Pulse Width, Low (Figure 1)	16, 64 Modes	PW_{CL}	600	–	–	ns
Minimum Clock Pulse Width, High (Figure 2)	16, 64 Modes	PW_{CH}	600	–	–	ns
Clock Frequency	1 Mode	f_C	–	–	500	kHz
	16, 64 Modes		–	–	800	
Clock-to-Data Delay for Transmitter (Figure 3)		t_{TDD}	–	–	1.0	μs
Receive Data Setup Time (Figure 4)	1 Mode	t_{RDSU}	500	–	–	ns
Receive Data Hold Time (Figure 5)	1 Mode	t_{RDH}	500	–	–	ns
Interrupt Request Release Time (Figure 6)		t_{IR}	–	–	1.2	μs
Request-to-Send Delay Time (Figure 6)		t_{RTS}	–	–	1.0	μs
Input Transition Times (Except Enable)		t_r, t_f	–	–	1.0*	μs

*1.0 μs or 10% of the pulse width, whichever is smaller.

BUS TIMING CHARACTERISTICS

Characteristic	Symbol	MC6850 Min	MC6850 Max	MC68A50 Min	MC68A50 Max	MC68B50 Min	MC68B50 Max	Unit
READ (Figures 7 and 9)								
Enable Cycle Time	t_{cycE}	1.0	–	0.666	–	0.500	–	μs
Enable Pulse Width, High	PW_{EH}	0.45	25	0.28	25	0.22	25	μs
Enable Pulse Width, Low	PW_{EL}	0.43	–	0.28	–	0.21	–	μs
Setup Time, Address and R/W valid to Enable positive transition	t_{AS}	160	–	140	–	70	–	ns
Data Delay Time	t_{DDR}	–	320	–	220	–	180	ns
Data Hold Time	t_H	10	–	10	–	10	–	ns
Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
Rise and Fall Time for Enable input	t_{Er}, t_{Ef}	–	25	–	25	–	25	ns
WRITE (Figures 8 and 9)								
Enable Cycle Time	t_{cycE}	1.0	–	0.666	–	500	–	μs
Enable Pulse Width, High	PW_{EH}	0.45	25	0.28	25	0.22	25	μs
Enable Pulse Width, Low	PW_{EL}	0.43	–	0.28	–	0.21	–	μs
Setup Time, Address and R/W valid to Enable positive transition	t_{AS}	160	–	140	–	70	–	ns
Data Setup Time	t_{DSW}	195	–	80	–	60	–	ns
Data Hold Time	t_H	10	–	10	–	10	–	ns
Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
Rise and Fall Time for Enable input	t_{Er}, t_{Ef}	–	25	–	25	–	25	ns

FIGURE 1 – CLOCK PULSE WIDTH, LOW-STATE

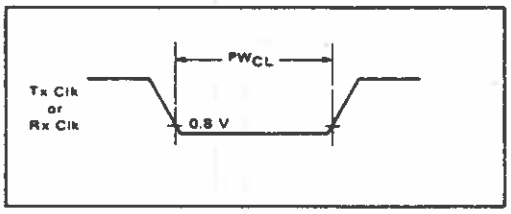


FIGURE 2 – CLOCK PULSE WIDTH, HIGH-STATE

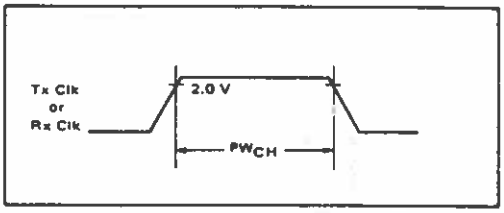


FIGURE 3 – TRANSMIT DATA OUTPUT DELAY

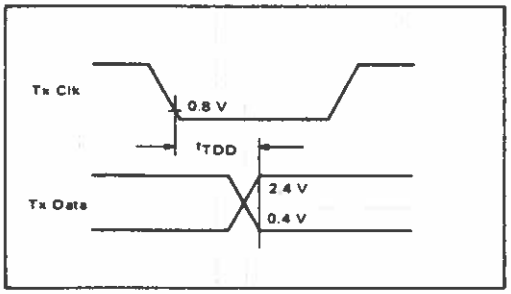


FIGURE 4 – RECEIVE DATA SETUP TIME
(÷1 Mode)

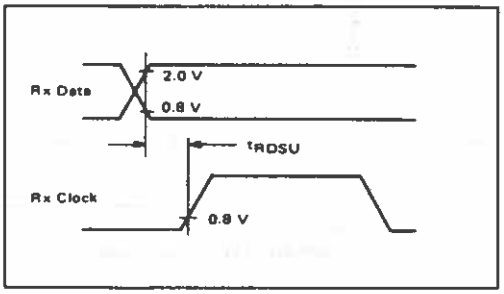


FIGURE 5 – RECEIVE DATA HOLD TIME
(÷1 Mode)

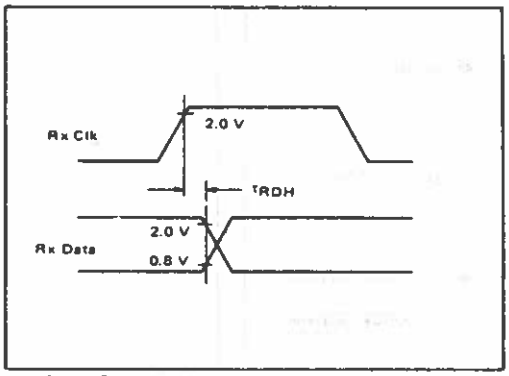


FIGURE 6 – REQUEST-TO-SEND DELAY AND INTERRUPT-REQUEST RELEASE TIMES

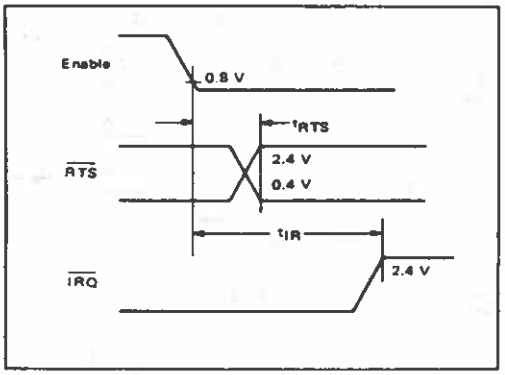


FIGURE 7 – BUS READ TIMING CHARACTERISTICS
(Read information from ACIA)

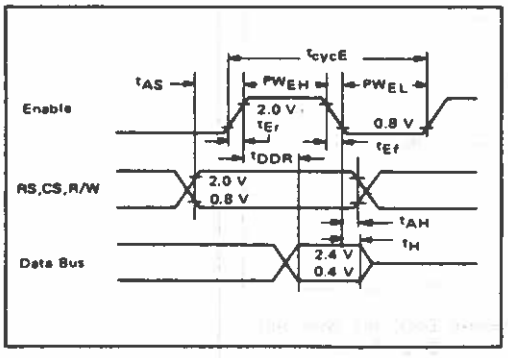
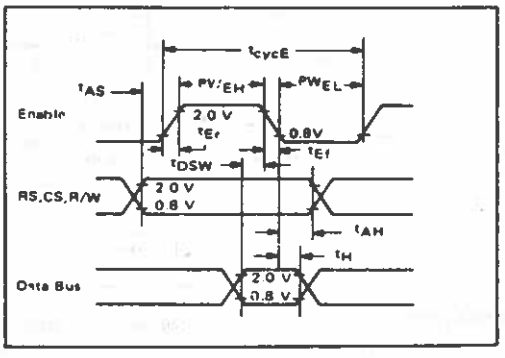
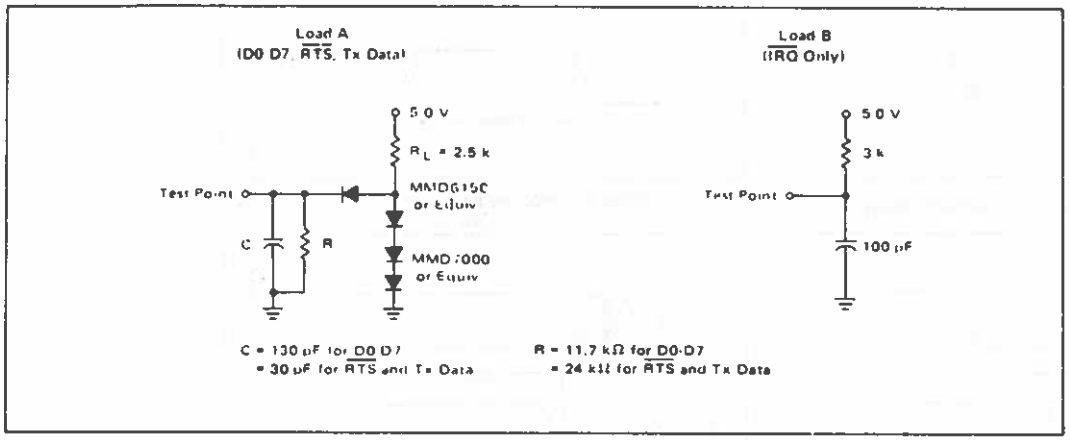


FIGURE 8 – BUS WRITE TIMING CHARACTERISTICS
(Write information into ACIA)

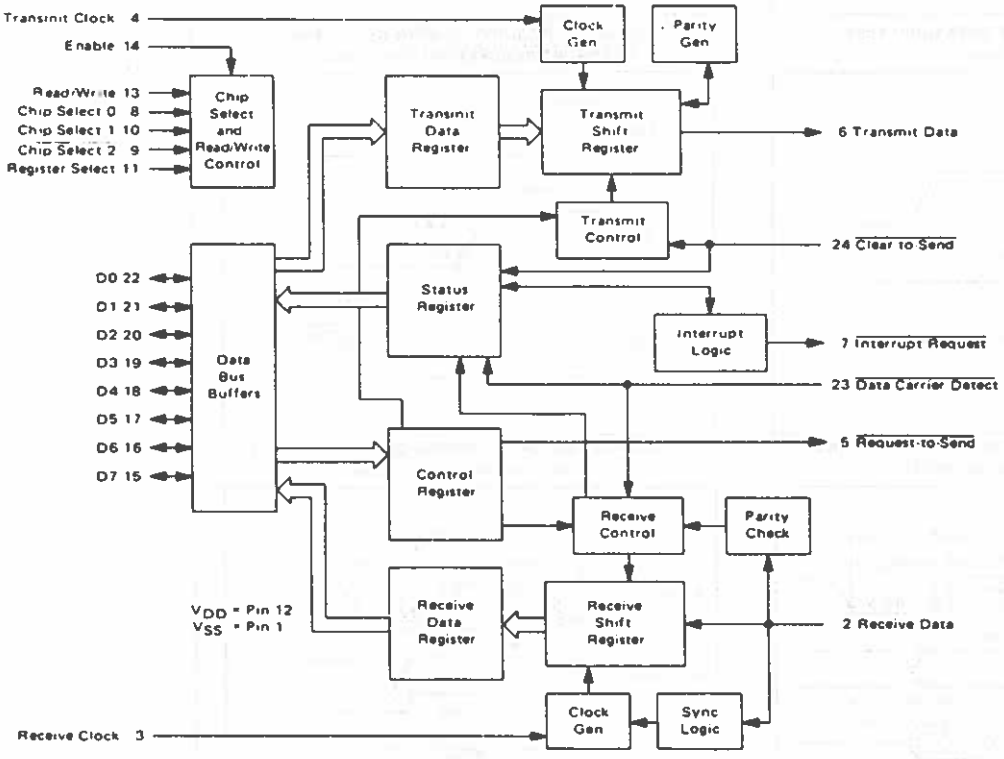


MOTOROLA Semiconductor Products Inc.

FIGURE 9 – BUS TIMING TEST LOADS



EXPANDED BLOCK DIAGRAM



DEVICE OPERATION

At the bus interface, the ACIA appears as two addressable memory locations. Internally, there are four registers: two read-only and two write-only registers. The read-only registers are Status and Receive Data; the write-only registers are Control and Transmit Data. The serial interface consists of serial input and output lines with independent clocks, and three peripheral/modem control lines.

POWER ON/MASTER RESET

The master reset (CR0, CR1) should be set during system initialization to insure the reset condition and prepare for programming the ACIA functional configuration when the communications channel is required. Control bits CR5 and CR6 should also be programmed to define the state of $\overline{RTS}$ whenever master reset is utilized. The ACIA also contains internal power-on reset logic to detect the power line turn-on transition and hold the chip in a reset state to prevent erroneous output transitions prior to initialization. This circuitry depends on clean power turn-on transitions. The power-on reset is released by means of the bus-programmed master reset which must be applied prior to operating the ACIA. After master resetting the ACIA, the programmable Control Register can be set for a number of options such as variable clock divider ratios, variable word length, one or two stop bits, parity (even, odd, or none), etc.

TRANSMIT

A typical transmitting sequence consists of reading the ACIA Status Register either as a result of an interrupt or in the ACIA's turn in a polling sequence. A character may be written into the Transmit Data Register if the status read operation has indicated that the Transmit Data Register is empty. This character is transferred to a Shift Register where it is serialized and transmitted from the Transmit Data output preceded by a start bit and followed by one or two stop bits. Internal parity (odd or even) can be optionally added to the character and will occur between the last data bit and the first stop bit. After the first character is written in the Data Register, the Status Register can be read again to check for a Transmit Data Register Empty condition and current peripheral status. If the register is empty, another character can be loaded for transmission even though the first character is in the process of being transmitted (because of double buffering). The second character will be automatically transferred into the Shift Register when the first character transmission is completed. This sequence continues until all the characters have been transmitted.

RECEIVE

Data is received from a peripheral by means of the Receive Data input. A divide-by-one clock ratio is provided for an externally synchronized clock (to its data) while the divide-by-16 and 64 ratios are provided for internal synchronization. Bit synchronization in the divide-by-16 and 64 modes is initiated by the detection of the leading mark-to-space transition of the start bit. False start bit deletion capability insures that a full half bit of a start bit has been received before the internal clock is synchronized to the bit time. As a character is being received, parity (odd or even) will be checked and the error indication will be available in the Status Register along with framing error, overrun error, and Receive Data Register full. In a typical receiving sequence, the Status Register is read to determine if a character has been received from a peripheral. If the Receiver Data Register is full, the character is placed on the 8-bit ACIA bus when a Read Data command is received from the MPU. When parity has been selected for an 8-bit word (7 bits plus parity), the receiver strips the parity bit (D7 = 0) so that data alone is transferred to the MPU. This feature reduces MPU programming. The Status Register can continue to be read again to determine when another character is available in the Receive Data Register. The receiver is also double buffered so that a character can be read from the data register as another character is being received in the shift register. The above sequence continues until all characters have been received.

INPUT/OUTPUT FUNCTIONS

ACIA INTERFACE SIGNALS FOR MPU

The ACIA interfaces to the MC6800 MPU with an 8-bit bi-directional data bus, three chip select lines, a register select line, an interrupt request line, read/write line, and enable line. These signals, in conjunction with the MC6800 VMA output, permit the MPU to have complete control over the ACIA.

ACIA Bi-Directional Data (D0-D7) – The bi-directional data lines (D0-D7) allow for data transfer between the ACIA and the MPU. The data bus output drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs an ACIA read operation.

ACIA Enable (E) – The Enable signal, E, is a high impedance TTL compatible input that enables the bus input/output data buffers and clocks data to and from the ACIA. This signal will normally be a derivative of the MC6800 ϕ2 Clock.

Read/Write (R/W) – The Read/Write line is a high impedance input that is TTL compatible and is used to control the direction of data flow through the ACIA's input/output data bus interface. When Read/Write is high (MPU Read cycle), ACIA output drivers are turned on and a selected register is read. When it is low, the ACIA output drivers are turned off and the MPU writes into a selected register. Therefore, the Read/Write signal is used to select read-only or write-only registers within the ACIA.

Chip Select (CS0, CS1, $\overline{CS2}$) – These three high impedance TTL compatible input lines are used to address the ACIA. The ACIA is selected when CS0 and CS1 are high and $\overline{CS2}$ is low. Transfers of data to and from the ACIA are then performed under the control of the Enable signal, Read/Write, and Register Select.

Register Select (RS) – The Register Select line is a high impedance input that is TTL compatible. A high level is used to select the Transmit/Receive Data Registers and a low level the Control/Status Registers. The Read/Write signal line is used in conjunction with Register Select to select the read-only or write-only register in each register pair.

$\overline{\text{Interrupt Request}}$ ($\overline{IRQ}$) – $\overline{\text{Interrupt Request}}$ is a TTL compatible, open-drain (no internal pullup), active low

output that is used to interrupt the MPU. The $\overline{IRQ}$ output remains low as long as the cause of the interrupt is present and the appropriate interrupt enable within the ACIA is set. The IRQ status bit, when high, indicates the $\overline{IRQ}$ output is in the active state.

Interrupts result from conditions in both the transmitter and receiver sections of the ACIA. The transmitter section causes an interrupt when the Transmitter Interrupt Enabled condition is selected (CR5 · $\overline{CR6}$), and the Transmit Data Register Empty (TDRE) status bit is high. The TDRE status bit indicates the current status of the Transmitter Data Register except when inhibited by Clear-to-Send ($\overline{CTS}$) being high or the ACIA being maintained in the Reset condition. The interrupt is cleared by writing data into the Transmit Data Register. The interrupt is masked by disabling the Transmitter Interrupt via CR5 or CR6 or by the loss of $\overline{CTS}$ which inhibits the TDRE status bit. The Receiver section causes an interrupt when the Receiver Interrupt Enable is set and the Receive Data Register Full (RDRF) status bit is high, an Overrun has occurred, or Data Carrier Detect ($\overline{DCD}$) has gone high. An interrupt resulting from the RDRF status bit can be cleared by reading data or resetting the ACIA. Interrupts caused by Overrun or loss of $\overline{DCD}$ are cleared by reading the status register after the error condition has occurred and then reading the Receive Data Register or resetting the ACIA. The receiver interrupt is masked by resetting the Receiver Interrupt Enable.

CLOCK INPUTS

Separate high impedance TTL compatible inputs are provided for clocking of transmitted and received data. Clock frequencies of 1, 16 or 64 times the data rate may be selected.

Transmit Clock (Tx Clk) – The Transmit Clock input is used for the clocking of transmitted data. The transmitter initiates data on the negative transition of the clock.

Receive Clock (Rx Clk) – The Receive Clock input is used for synchronization of received data. (In the ÷ 1 mode, the clock and data must be synchronized externally.) The receiver samples the data on the positive transiton of the clock.

SERIAL INPUT/OUTPUT LINES

Receive Data (Rx Data) – The Receive Data line is a high impedance TTL compatible input through which data is received in a serial format. Synchronization with a clock for detection of data is accomplished internally when clock rates of 16 or 64 times the bit rate are used. Data rates are in the range of 0 to 500 kbps when external synchronization is utilized.

Transmit Data (Tx Data) – The Transmit Data output line transfers serial data to a modem or other peripheral. Data rates are in the range of 0 to 500 kbps when external synchronization is utilized.

PERIPHERAL/MODEM CONTROL

The ACIA includes several functions that permit limited control of a peripheral or modem. The functions included are Clear-to-Send, Request-to-Send and Data Carrier Detect.

Clear-to-Send ($\overline{CTS}$) – This high impedance TTL compatible input provides automatic control of the transmitting end of a communications link via the modem Clear-to-Send active low output by inhibiting the Transmit Data Register Empty (TDRE) status bit.

Request-to-Send ($\overline{RTS}$) – The Request-to-Send output enables the MPU to control a peripheral or modem via the data bus. The $\overline{RTS}$ output corresponds to the state of the Control Register bits CR5 and CR6. When CR6 = 0 or both CR5 and CR6 = 1, the $\overline{RTS}$ output is low (the active state). This output can also be used for Data Terminal Ready ($\overline{DTR}$).

Data Carrier Detect ($\overline{DCD}$) – This high impedance TTL compatible input provides automatic control, such as in the receiving end of a communications link by means of a modem Data Carrier Detect output. The $\overline{DCD}$ input inhibits and initializes the receiver section of the ACIA when high. A low to high transition of the Data Carrier Detect initiates an interrupt to the MPU to indicate the occurrence of a loss of carrier when the Receive Interrupt Enable bit is set.

ACIA REGISTERS

The expanded block diagram for the ACIA indicates the internal registers on the chip that are used for the status, control, receiving, and transmitting of data. The content of each of the registers is summarized in Table 1.

TRANSMIT DATA REGISTER (TDR)

Data is written in the Transmit Data Register during the negative transition of the enable (E) when the ACIA has been addressed and RS · $\overline{R/W}$ is selected. Writing data into the register causes the Transmit Data Register Empty bit in the Status Register to go low. Data can then be transmitted. If the transmitter is idling and no character is being transmitted, then the transfer will take place within one bit time of the trailing edge of the Write command. If a character is being transmitted, the new data character will commence as soon as the previous character is complete. The transfer of data causes the Transmit Data Register Empty (TDRE) bit to indicate empty.

RECEIVE DATA REGISTER (RDR)

Data is automatically transferred to the empty Receive Data Register (RDR) from the receiver deserializer (a shift register) upon receiving a complete character. This event causes the Receive Data Register Full bit (RDRF) in the status buffer to go high (full). Data may then be read through the bus by addressing the ACIA and selecting the Receive Data Register with RS and R/W high when the ACIA is enabled. The non-destructive read cycle causes the RDRF bit to be cleared to empty although

TABLE 1 – DEFINITION OF ACIA REGISTER CONTENTS

Data Bus Line Number	Buffer Address			
	RS • R/W Transmit Data Register (Write Only)	RS • R/W Receive Data Register (Read Only)	RS • R/W Control Register (Write Only)	RS • R/W Status Register (Read Only)
0	Data Bit 0*	Data Bit 0	Counter Divide Select 1 (CR0)	Receive Data Register Full (RDRF)
1	Data Bit 1	Data Bit 1	Counter Divide Select 2 (CR1)	Transmit Data Register Empty (TDRE)
2	Data Bit 2	Data Bit 2	Word Select 1 (CR2)	Data Carrier Detect (DCD)
3	Data Bit 3	Data Bit 3	Word Select 2 (CR3)	Clear to Send (CTS)
4	Data Bit 4	Data Bit 4	Word Select 3 (CR4)	Framing Error (FE)
5	Data Bit 5	Data Bit 5	Transmit Control 1 (CR5)	Receiver Overrun (OVRN)
6	Data Bit 6	Data Bit 6	Transmit Control 2 (CR6)	Parity Error (PE)
7	Data Bit 7***	Data Bit 7**	Receive Interrupt Enable (CR7)	Interrupt Request (IRQ)

* Leading bit = LSB = Bit 0
** Data bit will be zero in 7 bit plus parity modes.
*** Data bit is "don't care" in 7 bit plus parity modes

the data is retained in the RDR. The status is maintained by RDRF as to whether or not the data is current. When the Receive Data Register is full, the automatic transfer of data from the Receiver Shift Register to the Data Register is inhibited and the RDR contents remain valid with its current status stored in the Status Register.

CONTROL REGISTER

The ACIA Control Register consists of eight bits of write-only buffer that are selected when RS and R/W are low. This register controls the function of the receiver, transmitter, interrupt enables, and the Request-to-Send peripheral/modem control output.

Counter Divide Select Bits (CR0 and CR1) – The Counter Divide Select Bits (CR0 and CR1) determine the divide ratios utilized in both the transmitter and receiver sections of the ACIA. Additionally, these bits are used to provide a master reset for the ACIA which clears the Status Register (except for external conditions on CTS and DCD) and initializes both the receiver and transmitter. Master reset does not affect other Control Register bits. Note that after power-on or a power fail/restart, these bits must be set high to reset the ACIA. After reseting, the clock divide ratio may be selected. These counter select bits provide for the following clock divide ratios:

CR1	CR0	Function
0	0	÷ 1
0	1	÷ 16
1	0	÷ 64
1	1	Master Reset

Word Select Bits (CR2, CR3, and CR4) – The Word Select bits are used to select word length, parity, and the number of stop bits. The encoding format is as follows:

CR4	CR3	CR2	Function
0	0	0	7 Bits + Even Parity + 2 Stop Bits
0	0	1	7 Bits + Odd Parity + 2 Stop Bits
0	1	0	7 Bits + Even Parity + 1 Stop Bit
0	1	1	7 Bits + Odd Parity + 1 Stop Bit
1	0	0	8 Bits + 2 Stop Bits
1	0	1	8 Bits + 1 Stop Bit
1	1	0	8 Bits + Even Parity + 1 Stop Bit
1	1	1	8 Bits + Odd Parity + 1 Stop Bit

Word length, Parity Select, and Stop Bit changes are not buffered and therefore become effective immediately.

Transmitter Control Bits (CR5 and CR6) – Two Transmitter Control bits provide for the control of the interrupt from the Transmit Data Register Empty condition, the Request-to-Send (RTS) output, and the transmission of a Break level (space). The following encoding format is used:

CR6	CR5	Function
0	0	RTS = low, Transmitting Interrupt Disabled.
0	1	RTS = low, Transmitting Interrupt Enabled.
1	0	RTS = high, Transmitting Interrupt Disabled.
1	1	RTS = low, Transmits a Break level on the Transmit Data Output. Transmitting Interrupt Disabled.

Receive Interrupt Enable Bit (CR7) – The following interrupts will be enabled by a high level in bit position 7 of the Control Register (CR7): Receive Data Register Full, Overrun, or a low to high transistion on the Data Carrier Detect (DCD) signal line.

STATUS REGISTER

Information on the status of the ACIA is available to the MPU by reading the ACIA Status Register. This read-only register is selected when RS is low and R/W is high. Information stored in this register indicates the status of the Transmit Data Register, the Receive Data Register and error logic, and the peripheral/modem status inputs of the ACIA.

Receive Data Register Full (RDRF), Bit 0 – Receive Data Register Full indicates that received data has been transferred to the Receive Data Register. RDRF is cleared after an MPU read of the Receive Data Register or by a master reset. The cleared or empty state indicates that the contents of the Receive Data Register are not current. $\overline{\text{Data Carrier Detect}}$ being high also causes RDRF to indicate empty.

Transmit Data Register Empty (TDRE), Bit 1 – The Transmit Data Register Empty bit being set high indicates that the Transmit Data Register contents have been transferred and that new data may be entered. The low state indicates that the register is full and that transmission of a new character has not begun since the last write data command.

$\overline{\text{Data Carrier Detect}}$ ($\overline{\text{DCD}}$), Bit 2 – The $\overline{\text{Data Carrier Detect}}$ bit will be high when the $\overline{\text{DCD}}$ input from a modem has gone high to indicate that a carrier is not present. This bit going high causes an Interrupt Request to be generated when the Receive Interrupt Enable is set. It remains high after the $\overline{\text{DCD}}$ input is returned low until cleared by first reading the Status Register and then the Data Register or until a master reset occurs. If the $\overline{\text{DCD}}$ input remains high after read status and read data or master reset has occurred, the interrupt is cleared, the $\overline{\text{DCD}}$ status bit remains high and will follow the $\overline{\text{DCD}}$ input.

$\overline{\text{Clear-to-Send}}$ ($\overline{\text{CTS}}$), Bit 3 – The $\overline{\text{Clear-to-Send}}$ bit indicates the state of the $\overline{\text{Clear-to-Send}}$ input from a modem. A low $\overline{\text{CTS}}$ indicates that there is a $\overline{\text{Clear-to-Send}}$ from the modem. In the high state, the Transmit Data Register Empty bit is inhibited and the $\overline{\text{Clear-to-Send}}$ status bit will be high. Master reset does not affect the $\overline{\text{Clear-to-Send}}$ Status bit.

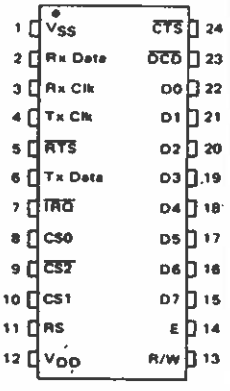
Framing Error (FE), Bit 4 – Framing error indicates that the received character is improperly framed by a start and a stop bit and is detected by the absence of the 1st stop bit. This error indicates a synchronization error, faulty transmission, or a break condition. The framing error flag is set or reset during the receive data transfer time. Therefore, this error indicator is present throughout the time that the associated character is available.

Receiver Overrun (OVRN), Bit 5 – Overrun is an error flag that indicates that one or more characters in the data stream were lost. That is, a character or a number of characters were received but not read from the Receive Data Register (RDR) prior to subsequent characters being received. The overrun condition begins at the midpoint of the last bit of the second character received in succession without a read of the RDR having occurred. The Overrun does not occur in the Status Register until the valid character prior to Overrun has been read. The RDRF bit remains set until the Overrun is reset. Character synchronization is maintained during the Overrun condition. The Overrun indication is reset after the reading of data from the Receive Data Register or by a Master Reset.

Parity Error (PE), Bit 6 – The parity error flag indicates that the number of highs (ones) in the character does not agree with the preselected odd or even parity. Odd parity is defined to be when the total number of ones is odd. The parity error indication will be present as long as the data character is in the RDR. If no parity is selected, then both the transmitter parity generator output and the receiver parity check results are inhibited.

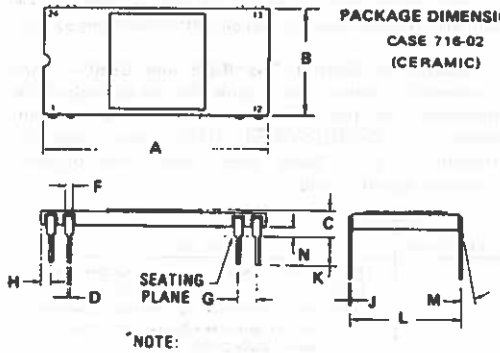
Interrupt Request (IRQ), Bit 7 – The IRQ bit indicates the state of the $\overline{\text{IRQ}}$ output. Any interrupt condition with its applicable enable will be indicated in this status bit. Anytime the $\overline{\text{IRQ}}$ output is low the IRQ bit will be high to indicate the interrupt or service request status. IRQ is cleared by a read operation to the Receive Data Register or a write operation to the Transmit Data Register.

PIN ASSIGNMENT



PACKAGE DIMENSIONS

CASE 716-02
(CERAMIC)



DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	29.97	30.99	1.180	1.220
B	14.88	15.62	0.585	0.615
C	3.05	4.19	0.120	0.165
D	0.38	0.53	0.015	0.021
F	0.76	1.40	0.030	0.055
G	2.54 BSC		0.100 BSC	
H	0.76	1.78	0.030	0.070
J	0.20	0.30	0.008	0.012
K	2.54	4.19	0.100	0.165
L	14.88	15.37	0.585	0.605
M	–	10°	–	10°
N	0.51	1.52	0.020	0.060

*NOTE:
1. LEADS TRUE POSITIONED WITHIN 0.25mm (0.010) DIA (AT SEATING PLANE) AT MAXIMUM MATERIAL CONDITION.

MOTOROLA Semiconductor Products Inc.

3B-A78/8.0 Printed in Switzerland

Fiche technique :

6808

MOTOROLA

MC6808

Advance Information

MICROPROCESSOR WITH CLOCK

The MC6808 is a monolithic 8-bit microprocessor that contains all the registers and accumulators of the present MC6800 plus an internal clock oscillator and driver on the same chip.

The MC6808 is completely software-compatible with the MC6800 as well as the entire M6800 family of parts. Hence the MC6808 is expandable to 65K words.

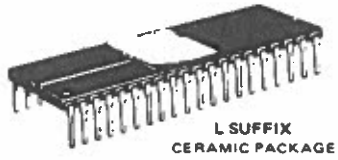
This very cost-effective MPU allows the designer to use the MC6808 in consumer as well as industrial applications without sacrificing industrial specifications.

- On-Chip Clock Circuit
- Software-Compatible with the MC6800
- Expandable to 65K words
- Standard TTL-Compatible Inputs and Outputs
- 8-Bit Word Size
- 16-Bit Memory Addressing
- Interrupt Capability

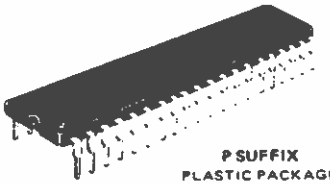
MOS

(N-CHANNEL, SILICON-GATE, DEPLETION LOAD)

MICROPROCESSOR WITH CLOCK

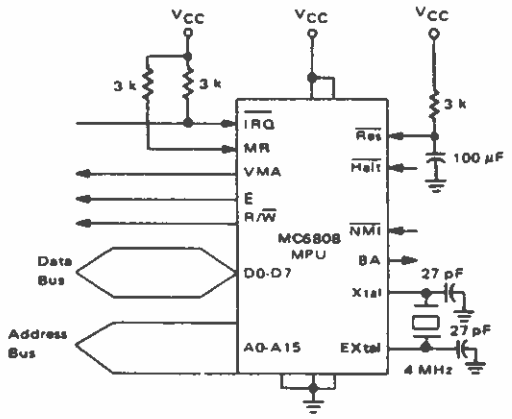


L SUFFIX
CERAMIC PACKAGE
CASE 715

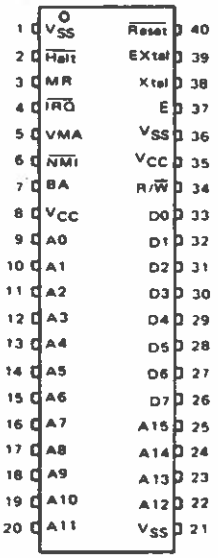


P SUFFIX
PLASTIC PACKAGE
CASE 711

FIGURE 1 – TYPICAL MICROPROCESSOR INTERFACE



PIN ASSIGNMENT



This is advance information and specifications are subject to change without notice.

ADI-805

MC6808

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.3 to +7.0	Vdc
Input Voltage	V_{in}	-0.3 to +7.0	Vdc
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	-55 to +150	°C
Thermal Resistance Plastic Ceramic	θ_{JA}	 100 50	°C/W

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit.

ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 V ± 5%, V_{SS} = 0, T_A = 0 to 70°C unless otherwise noted.)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	Logic, EXtal RESET	V_{IH}	V_{SS} + 2.0 V_{SS} + 4.0		V_{CC} V_{CC}	Vdc
Input Low Voltage	Logic, EXtal RESET	V_{IL}	V_{SS} - 0.3 V_{SS} - 0.3		V_{SS} + 0.8 V_{SS} + 2.3	Vdc
Input Leakage Current (V_{in} = 0 to 5.25 V, V_{CC} = max)	Logic*	I_{in}		1.0	2.5	μAdc
Output High Voltage (I_{Load} = -205 μAdc, V_{CC} = min) (I_{Load} = -145 μAdc, V_{CC} = min) (I_{Load} = -100 μAdc, V_{CC} = min)	 D0-D7 A0-A15, R/W, VMA, E, BA	V_{OH}	 V_{SS} + 2.4 V_{SS} + 2.4 V_{SS} + 2.4			Vdc
Output Low Voltage (I_{Load} = 1.6 mAdc, V_{CC} = min)		V_{OL}			V_{SS} + 0.4	Vdc
Power Dissipation		P_D**		0.600	1.2	W
Capacitance = (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7 Logic Inputs, EXtal A0-A15, R/W, VMA	C_{in} C_{out}		10 6.5	12.5 10 12	pF pF
Frequency of Operation (Input Clock 4) (Crystal Frequency)		f f_{Xtal}	0.1 1.0		1.0 4.0	MHz
Clock Timing Cycle Time Clock Pulse Width (measured at 2.4V) (measured at 0.4V) Fall Time (Measured between V_{SS} + 0.4 V and V_{SS} + 2.4 V)		 t_{cyc} $PW_{\phi H}$ $PW_{\phi L}$ t_f	 1.0 450 450		 10 4500 4500 25	 μs ns ns

*Except $\overline{IRQ}$ and $\overline{NMI}$, which require 3 kΩ pullup load resistors for wire OR capability at optimum operation. Does not include EXtal and Xtal, which are crystal inputs.

**Capacitances are periodically sampled rather than 100% tested.

READ/WRITE TIMING (Figures 2 through 6, Load Circuit of Figure 4.)

Characteristic	Symbol	Min	Typ	Max	Unit
Address Delay	t_{AD}	-		270	ns
Peripheral Read Access Time $t_{acc} = t_{ut} - (t_{AD} + t_{DSR})$; $t_{ut} = t_{cyc} - t_f$	t_{acc}	-	-	530	ns
Data Setup Time (Read)	t_{DSR}	100	-	-	ns
Input Data Hold Time	t_H	10	-	-	ns
Output Data Hold Time	t_H	30	-	-	ns
Address Hold Time (Address, R/W, VMA)	t_{AH}	20	-	-	ns
Data Delay Time (Write)	t_{DDW}	-	165	225	ns
Processor Controls Processor Control Setup Time Processor Control Rise and Fall Time (Measured between 0.8 V and 2.0 V)	 t_{PCS} t_{PCr}, t_{PCf}	 200 -	 - -	 - 100	 ns ns
Bus Available Delay Time	t_{BA}	-	-	250	ns

FIGURE 2 – READ DATA FROM MEMORY OR PERIPHERALS

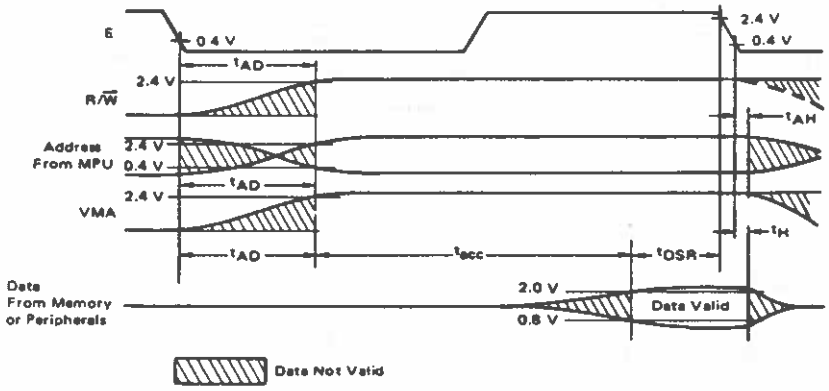


FIGURE 3 – WRITE DATA IN MEMORY OR PERIPHERALS

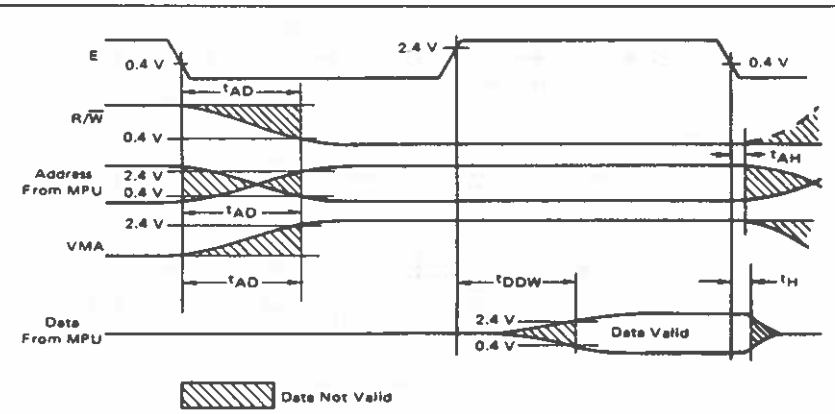


FIGURE 4 – BUS TIMING TEST LOAD

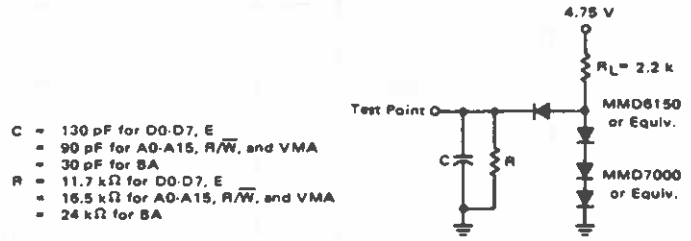


FIGURE 5 – TYPICAL DATA BUS OUTPUT DELAY versus CAPACITIVE LOADING

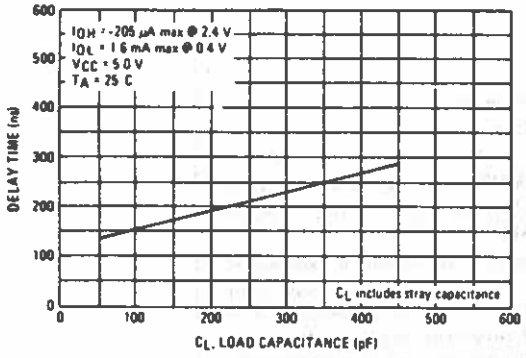


FIGURE 6 – TYPICAL READ/WRITE, VMA, AND ADDRESS OUTPUT DELAY versus CAPACITIVE LOADING

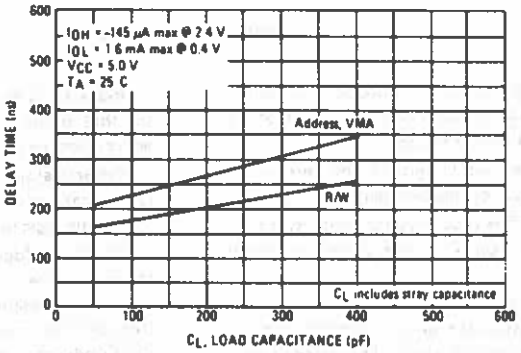
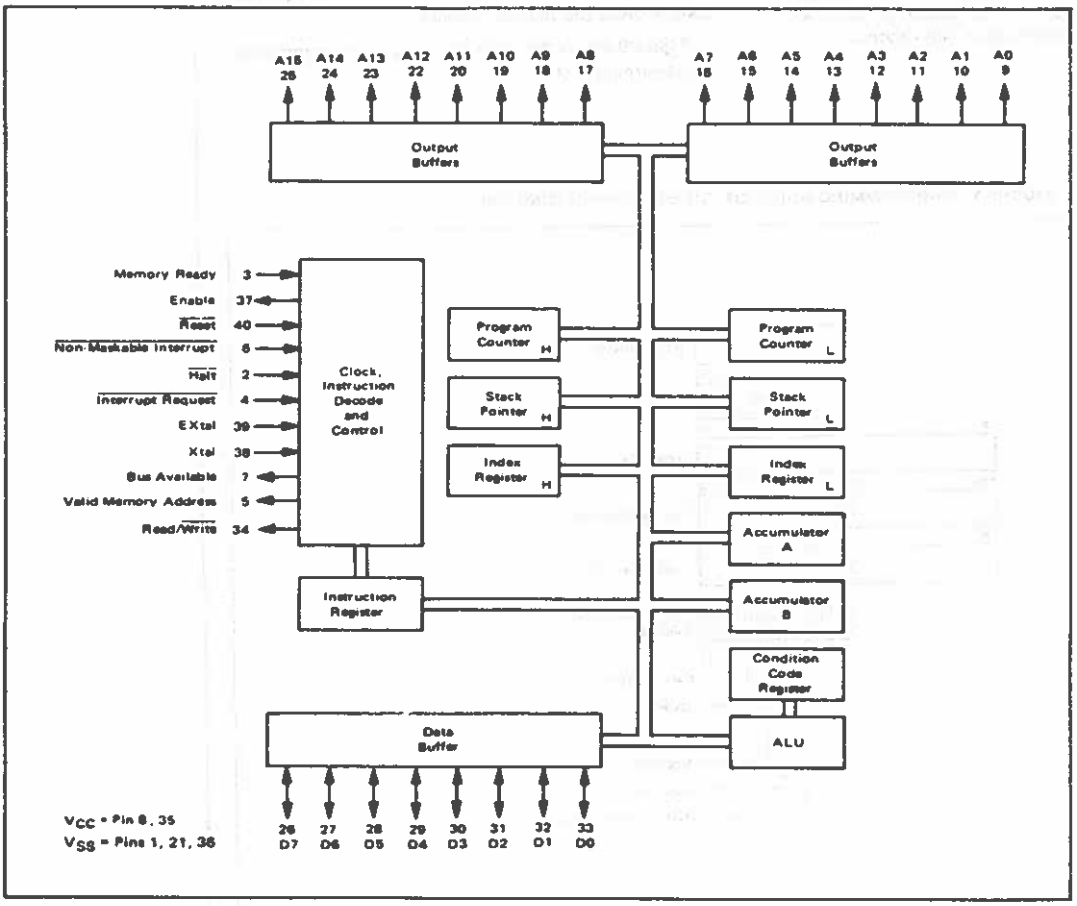


FIGURE 7 – MC6808 EXPANDED BLOCK DIAGRAM



MPU REGISTERS

A general block diagram of the MC6808 is shown in Figure 7. As shown, the number and configuration of the registers are the same as for the MC6800.

The MPU has three 16-bit registers and three 8-bit registers available for use by the programmer (Figure 8).

Program Counter – The program counter is a two byte (16-bits) register that points to the current program address.

Stack Pointer – The stack pointer is a two byte register that contains the address of the next available location in an external push-down/pop-up stack. This stack is normally a random access Read/Write memory that may have any location (address) that is convenient. In those applications that require storage of information in the stack when power is lost, the stack must be non-volatile.

Index Register – The index register is a two byte register that is used to store data or a sixteen bit memory address for the Indexed mode of memory addressing.

Accumulators – The MPU contains two 8-bit accumulators that are used to hold operands and results from an arithmetic logic unit (ALU).

Condition Code Register – The condition code register indicates the results of an Arithmetic Logic Unit operation: Negative (N), Zero (Z), Overflow (V), Carry from bit 7 (C), and half carry from bit 3 (H). These bits of the Condition Code Register are used as testable conditions for the conditional branch instructions. Bit 4 is the interrupt mask bit (I). The used bits of the Condition Code Register (b6 and b7) are ones.

Figure 9 shows the order of saving the microprocessor status within the stack.

FIGURE 8 – PROGRAMMING MODEL OF THE MICROPROCESSING UNIT

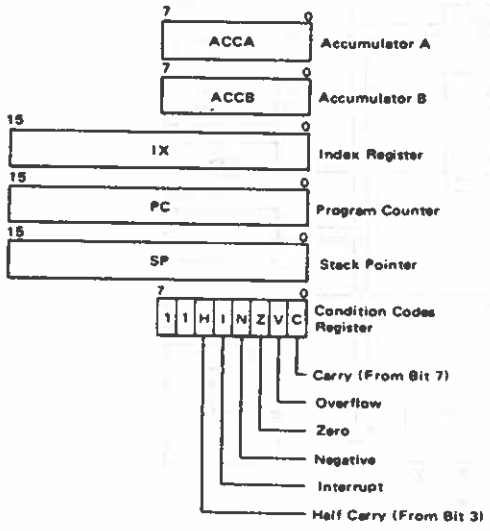
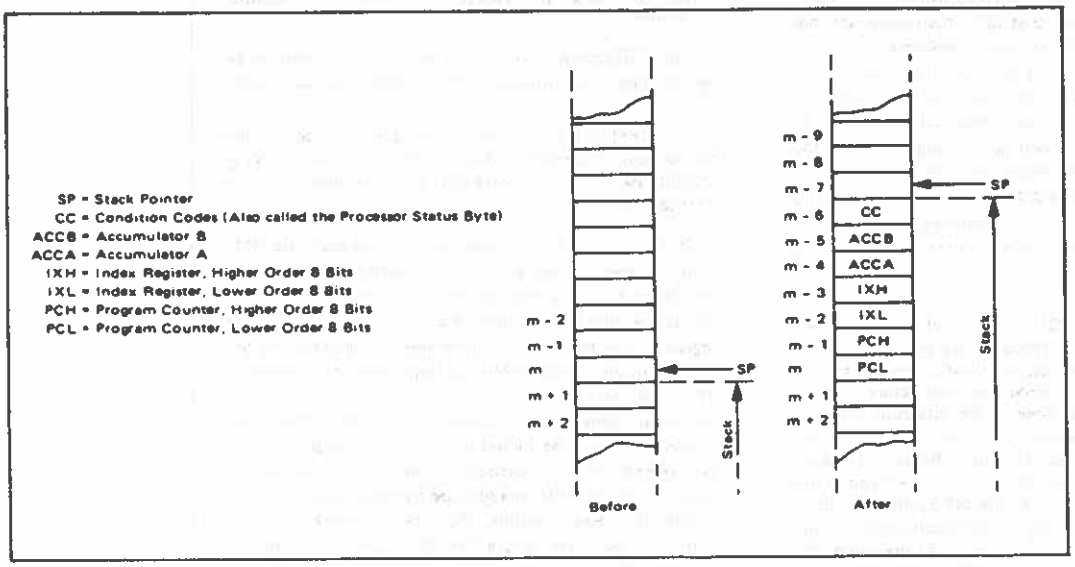


FIGURE 9 – SAVING THE STATUS OF THE MICROPROCESSOR IN THE STACK



MC6808 MPU SIGNAL DESCRIPTION

Proper operation of the MPU requires that certain control and timing signals be provided to accomplish specific functions and that other signal lines be monitored to determine the state of the processor. These control and timing signals for the MC6808 are similar to those of the MC6800 except that TSC, DBE, $\phi 1$, $\phi 2$ input, and two unused pins have been eliminated, and the following signal and timing lines have been added:

Crystal Connections EXtal and Xtal
Memory Ready (MR)
Enable $\phi 2$ Output (E)

The following is a summary of the MC6808 MPU signals:

Address Bus (A0-A15) – Sixteen pins are used for the address bus. The outputs are capable of driving one standard TTL load and 90 pF.

Data Bus (D0-D7) – Eight pins are used for the data bus. It is bidirectional, transferring data to and from the memory and peripheral devices. It also has three-state output buffers capable of driving one standard TTL load and 130 pF.

$\overline{\text{Halt}}$ – When this input is in the low state, all activity in the machine will be halted. This input is level sensitive. In the halt mode, the machine will stop at the end of an instruction, Bus Available will be at a high state, Valid Memory Address will be at a low state, and all other three-state lines will be in the three-state mode. The address bus will display the address of the next instruction.

To insure single instruction operation, transition of the $\overline{\text{Halt}}$ line must not occur during the last 200 ns of E and the $\overline{\text{Halt}}$ line must go high for one Clock cycle.

Read/$\overline{\text{Write}}$ (R/$\overline{\text{W}}$) – This TTL compatible output signals the peripherals and memory devices whether the MPU is in a Read (high) or Write (low) state. The normal standby state of this signal is Read (high). When the processor is halted, it will be in the logical one state. This output is capable of driving one standard TTL load and 90 pF.

Valid Memory Address (VMA) – This output indicates to peripheral devices that there is a valid address on the address bus. In normal operation, this signal should be utilized for enabling peripheral interfaces such as the PIA and ACIA. This signal is not three-state. One standard TTL load and 90 pF may be directly driven by this active high signal.

Bus Available (BA) — The Bus Available signal will normally be in the low state; when activated, it will go to the high state indicating that the microprocessor has stopped and that the address bus is available but not in three-state. This will occur if the $\overline{Halt}$ line is in the low state or the processor is in the WAIT state as a result of the execution of a WAIT instruction. At such time, all three-state output drivers will go to their off state and other outputs to their normally inactive level. The processor is removed from the WAIT state by the occurrence of a maskable (mask bit I = 0) or nonmaskable interrupt. This output is capable of driving one standard TTL load and 30 pF.

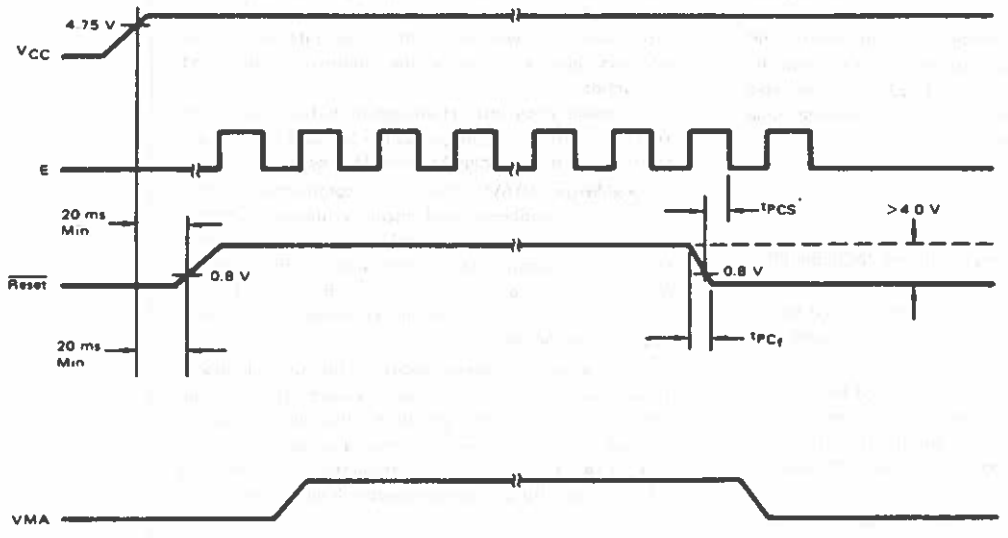
$\overline{Interrupt\ Request}$ ($\overline{IRQ}$) — This level sensitive input requests that an interrupt sequence be generated within the machine. The processor will wait until it completes the current instruction that is being executed before it recognizes the request. At that time, if the interrupt mask bit in the Condition Code Register is not set, the machine will begin an interrupt sequence. The Index Register, Program Counter, Accumulators, and Condition Code Register are stored away on the stack. Next the MPU will respond to the interrupt request by setting the interrupt mask bit high so that no further interrupts may occur. At the end of the cycle, a 16-bit address will be loaded that points to a vectoring address which is located in memory locations FFF8 and FFF9. An address loaded at these locations causes the MPU to branch to an interrupt routine in memory.

The $\overline{Halt}$ line must be in the high state for interrupts to be serviced. Interrupts will be latched internally while $\overline{Halt}$ is low.

The $\overline{IRQ}$ has a high impedance pullup device internal to the chip; however a 3 kΩ external resistor to V_{CC} should be used for wire-OR and optimum control of interrupts.

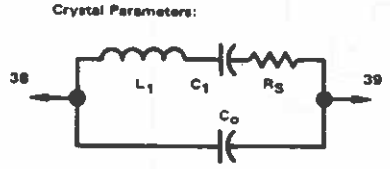
$\overline{Reset}$ — This input is used to reset and start the MPU from a power down condition, resulting from a power failure or an initial start-up of the processor. When this line is low, the MPU is inactive and the information in the registers will be lost. If a high level is detected on the input, this will signal the MPU to begin the restart sequence. This will start execution of a routine to initialize the processor from its reset condition. All the higher order address lines will be forced high. For the restart, the last two (FFFE, FFFF) locations in memory will be used to load the program that is addressed by the program counter. During the restart routine, the interrupt mask bit is set and must be reset before the MPU can be interrupted by $\overline{IRQ}$. Power-up and reset timing sequences are shown in Figure 10.

FIGURE 10 – POWER-UP AND RESET TIMING



EXtal and Xtal – The MC 6808 has an internal oscillator that may be crystal controlled. These connections are for a parallel resonant fundamental crystal. (AT cut.) A divide-by-four circuit has been added to the MC6808 so that a 4 MHz crystal may be used in lieu of a 1 MHz crystal for a more cost-effective system. Pin 39 of the MC6808 may be driven externally by a TTL input signal if a separate clock is required. Pin 38 is to be left open in this mode. Crystal parameters to be specified are in Figure 11.

FIGURE 11–CRYSTAL PARAMETERS



AT – Cut Parallel Resonance Crystal
C_O = 7 pF Max.
FREQ = 4.0 MHz @ C_L = 24pF
R_S = 60 ohms Max.
Frequency Tolerance – ± 5% to ± 0.02%
The best E output "Worst Case Design" tolerance is ± 0.05% (500 ppM) using A ± 0.02% crystal.

Non-Maskable Interrupt (NMI) – A low-going edge on this input requests that a non-mask-interrupt sequence be generated within the processor. As with the $\overline{\text{Interrupt Request}}$ signal, the processor will complete the current instruction that is being executed before it recognizes the $\overline{\text{NMI}}$ signal. The interrupt mask bit in the Condition Code Register has no effect on $\overline{\text{NMI}}$.

The index Register, Program Counter, Accumulators, and Condition Code Register are stored away on the stack. At the end of the cycle, a 16-bit address will be loaded that points to a vectoring address which is located in memory locations FFFC and FFFD. An address loaded at these locations caused the MPU to branch to a non-maskable interrupt routine in memory.

$\overline{\text{NMI}}$ has a high impedance pullup resistor internal to the chip; however a 3 kΩ external resistor to V_{CC} should be used for wire-OR and optimum control of interrupts.

Inputs $\overline{\text{IRQ}}$ and $\overline{\text{NMI}}$ are hardware interrupt lines that are sampled when E is high and will start the interrupt routine on a low E following the completion of an instruction.

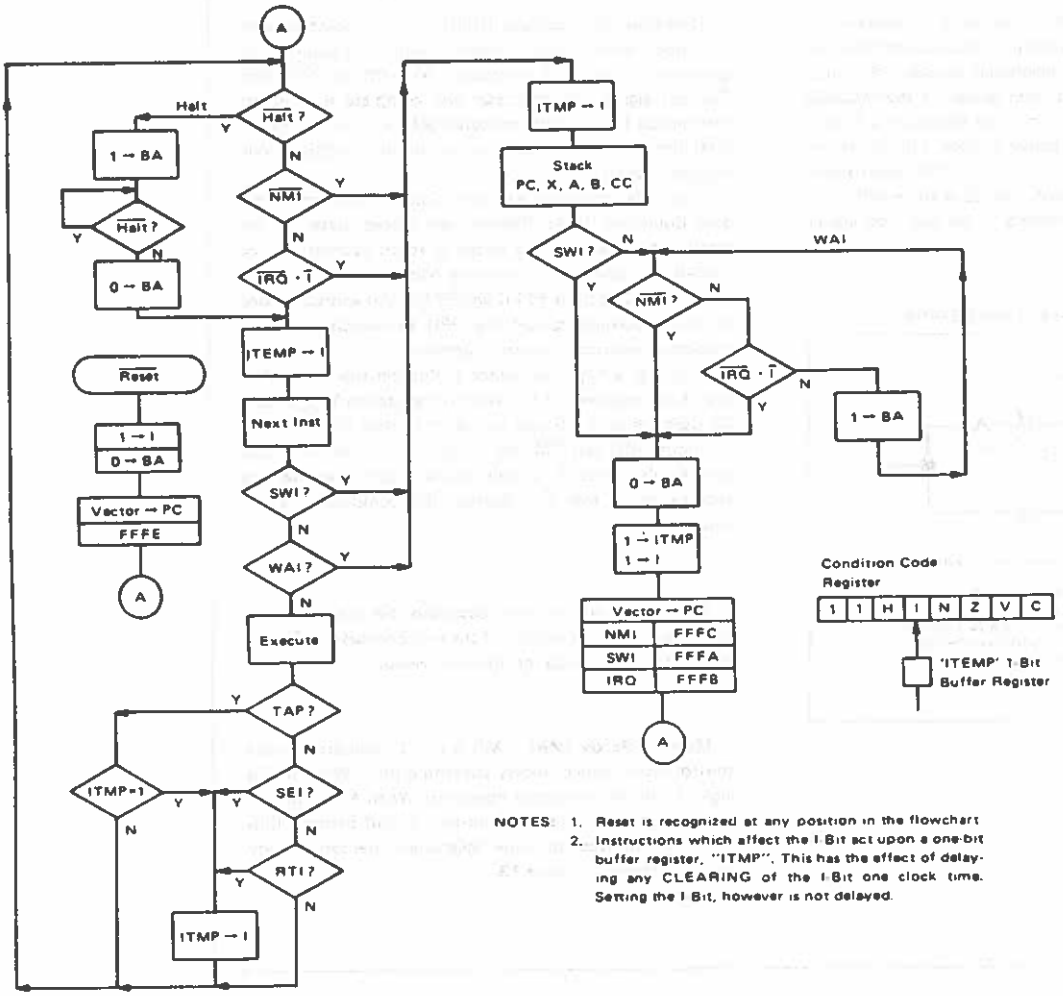
Figure 12 is a flow chart describing the major decision paths and interrupt vectors of the microprocessor. Table 1 gives the memory map for interrupt vectors.

Memory Ready (MR) – MR is a TTL compatible input control signal which allows stretching of E. When MR is high, E will be in normal operation. When MR is low, E may be stretched integral multiples of half periods, thus allowing interface to slow memories. Memory Ready timing is shown in Figure 13.

TABLE 1 – MEMORY MAP FOR INTERRUPT VECTORS

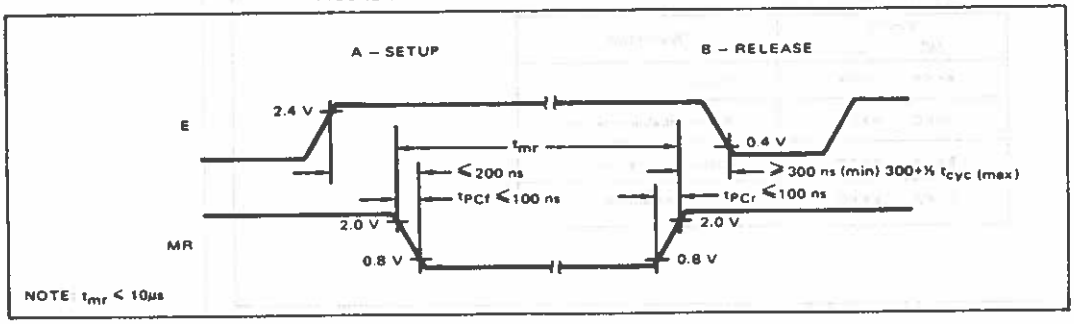
Vector MS	Vector LS	Description
FFFE	FFFF	Restart
FFFC	FFFD	Non-maskable Interrupt
FFFA	FFFB	Software Interrupt
FFF8	FFF9	Interrupt Request

FIGURE 12 – MPU FLOW CHART



NOTES:
1. Reset is recognized at any position in the flowchart
2. Instructions which affect the I Bit act upon a one-bit buffer register, "ITMP". This has the effect of delaying any CLEARING of the I-Bit one clock time. Setting the I Bit, however is not delayed.

FIGURE 13 – MEMORY READY CONTROL FUNCTION



MPU INSTRUCTION SET

The MC6808 has a set of 72 different instructions. Included are binary and decimal arithmetic, logical, shift, rotate, load, store, conditional or unconditional branch, interrupt and stack manipulation instructions (Tables 2 thru 6). This instruction set is the same as that for the MC6800.

MPU ADDRESSING MODES

The MC6808 eight-bit microprocessing unit has seven address modes that can be used by a programmer, with the addressing mode a function of both the type of instruction and the coding within the instruction. A summary of the addressing modes for a particular instruction can be found in Table 7 along with the associated instruction execution time that is given in machine cycles. With a clock frequency of 1 MHz, these times would be microseconds.

Accumulator (ACCX) Addressing — In accumulator only addressing, either accumulator A or accumulator B is specified. These are one-byte instructions.

Immediate Addressing — In immediate addressing, the operand is contained in the second byte of the instruction except LDS and LDX which have the operand in the second and third bytes of the instruction. The MPU addresses this location when it fetches the immediate instruction for execution. These are two or three-byte instructions.

Direct Addressing — In direct addressing, the address of the operand is contained in the second byte of the instruction. Direct addressing allows the user to directly address the lowest 256 bytes in the machine i.e., locations zero through 255. Enhanced execution times are achieved by storing data in these locations. In most configurations, it should be a random access memory. These are two-byte instructions.

Extended Addressing — In extended addressing, the address contained in the second byte of the instruction is used as the higher eight bits of the address of the operand. The third byte of the instruction is used as the lower eight bits of the address for the operand. This is an absolute address in memory. These are three-byte instructions.

Indexed Addressing — In indexed addressing, the address contained in the second byte of the instruction is added to the index register's lowest eight bits in the MPU. The carry is then added to the higher order eight bits of the index register. This result is then used to address memory. The modified address is held in a temporary address register so there is no change to the index register. These are two-byte instructions.

Implied Addressing — In the implied addressing mode the instruction gives the address (i.e., stack pointer, index register, etc.). These are one-byte instructions.

Relative Addressing — In relative addressing, the address contained in the second byte of the instruction is added to the program counter's lowest eight bits plus two. The carry or borrow is then added to the high eight bits. This allows the user to address data within a range of -125 to +129 bytes of the present instruction. These are two-byte instructions.

TABLE 2 — MICROPROCESSOR INSTRUCTION SET — ALPHABETIC SEQUENCE

ABA	Add Accumulators	CLR	Clear	PUL	Pull Data
ADC	Add with Carry	CLV	Clear Overflow	ROL	Rotate Left
ADD	Add	CMP	Compare	ROR	Rotate Right
AND	Logical And	COM	Complement	RTI	Return from Interrupt
ASL	Arithmetic Shift Left	CPX	Compare Index Register	RTS	Return from Subroutine
ASR	Arithmetic Shift Right	DAA	Decimal Adjust	SBA	Subtract Accumulators
BCC	Branch if Carry Clear	DEC	Decrement	SBC	Subtract with Carry
BCS	Branch if Carry Set	DES	Decrement Stack Pointer	SEC	Set Carry
BEQ	Branch if Equal to Zero	DEX	Decrement Index Register	SEI	Set Interrupt Mask
BGE	Branch if Greater or Equal Zero	EOR	Exclusive OR	SEV	Set Overflow
BGT	Branch if Greater than Zero	INC	Increment	STA	Store Accumulator
BHI	Branch if Higher	INS	Increment Stack Pointer	STS	Store Stack Register
BIT	Bit Test	INX	Increment Index Register	STX	Store Index Register
BLE	Branch if Less or Equal	JMP	Jump	SUB	Subtract
BLS	Branch if Lower or Same	JSR	Jump to Subroutine	SWI	Software Interrupt
BLT	Branch if Less than Zero	LDA	Load Accumulator	TAB	Transfer Accumulators
BMI	Branch if Minus	LDS	Load Stack Pointer	TAP	Transfer Accumulators to Condition Code Reg.
BNE	Branch if Not Equal to Zero	LDX	Load Index Register	TBA	Transfer Accumulators
BPL	Branch if Plus	LSR	Logical Shift Right	TPA	Transfer Condition Code Reg. to Accumulator
BRA	Branch Always	NEG	Negate	TST	Test
BSR	Branch to Subroutine	NOP	No Operation	TSX	Transfer Stack Pointer to Index Register
BVC	Branch if Overflow Clear	ORA	Inclusive OR Accumulator	TXS	Transfer Index Register to Stack Pointer
BVS	Branch if Overflow Set	PSH	Push Data	WAI	Wait for Interrupt
CBA	Compare Accumulators				
CLC	Clear Carry				
CLI	Clear Interrupt Mask				

TABLE 3 – ACCUMULATOR AND MEMORY INSTRUCTIONS

OPERATIONS	MNEMONIC	IMMED OP	IMMED ~	IMMED #	DIRECT OP	DIRECT ~	DIRECT #	INDEX OP	INDEX ~	INDEX #	EXTND OP	EXTND ~	EXTND #	IMPLIED OP	IMPLIED ~	IMPLIED #	BOOLEAN/ARITHMETIC OPERATION (All register labels refer to contents)	5 H	4 I	3 N	2 Z	1 V	0 C
Add	ADDA	3B	2	2	9B	3	2	AB	5	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	5	2	FB	4	3				B + M → B	↕	•	↕	↕	↕	↕
Add Accums	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add with Carry	ADCA	89	2	2	99	3	2	A9	5	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	5	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	5	2	B4	4	3				A · M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	5	2	F4	4	3				B · M → B	•	•	↕	↕	R	•
Bit Test	BITA	85	2	2	95	3	2	A5	5	2	B5	4	3				A · M	•	•	↕	↕	R	•
	BITB	C5	2	2	D5	3	2	E5	5	2	F5	4	3				B · M	•	•	↕	↕	R	•
Clear	CLR							6F	7	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	5	2	B1	4	3				A − M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	5	2	F1	4	3				B − M	•	•	↕	↕	↕	↕
Compare Accums	CBA													11	2	1	A − B	•	•	↕	↕	↕	↕
Complement 1's	COM							63	7	2	73	6	3				M̄ → M	•	•	↕	↕	R	S
	COMA													43	2	1	Ā → A	•	•	↕	↕	R	S
	COMB													53	2	1	B̄ → B	•	•	↕	↕	R	S
Complement 2's (Negate)	NEG							60	7	2	70	6	3				00 − M → M	•	•	↕	↕	②	②
	NEGA													40	2	1	00 − A → A	•	•	↕	↕	②	②
	NEGB													50	2	1	00 − B → B	•	•	↕	↕	②	②
Decimal Adjust A	DAA													19	2	1	Converts Binary Add of BCD Characters into BCD Format	•	•	↕	↕	↕	③
Decrement	DEC							6A	7	2	7A	6	3				M − 1 → M	•	•	↕	↕	④	•
	DECA													4A	2	1	A − 1 → A	•	•	↕	↕	④	•
	DECB													5A	2	1	B − 1 → B	•	•	↕	↕	④	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	5	2	B8	4	3				A⊕M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	5	2	F8	4	3				B⊕M → B	•	•	↕	↕	R	•
Increment	INC							6C	7	2	7C	6	3				M + 1 → M	•	•	↕	↕	⑤	•
	INCA													4C	2	1	A + 1 → A	•	•	↕	↕	⑤	•
	INCB													5C	2	1	B + 1 → B	•	•	↕	↕	⑤	•
Load Accum	LDAA	86	2	2	96	3	2	A6	5	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	5	2	F6	4	3				M → B	•	•	↕	↕	R	•
Or Inclusive	ORAA	8A	2	2	9A	3	2	AA	5	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	5	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	4	1	A → M_{SP}, SP − 1 → SP	•	•	•	•	•	•
	PSHB													37	4	1	B → M_{SP}, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULA													32	4	1	SP + 1 → SP, M_{SP} → A	•	•	•	•	•	•
	PULB													33	4	1	SP + 1 → SP, M_{SP} → B	•	•	•	•	•	•
Rotate Left	ROL							69	7	2	79	6	3				M	•	•	↕	↕	⑥	↕
	ROLA													49	2	1	A	•	•	↕	↕	⑥	↕
	ROLB													59	2	1	B	•	•	↕	↕	⑥	↕
Rotate Right	ROR							66	7	2	76	6	3				M	•	•	↕	↕	⑥	↕
	RORA													46	2	1	A	•	•	↕	↕	⑥	↕
	RORB													56	2	1	B	•	•	↕	↕	⑥	↕
Shift Left Arithmetic	ASL							68	7	2	78	6	3				M	•	•	↕	↕	⑥	↕
	ASLA													48	2	1	A	•	•	↕	↕	⑥	↕
	ASLB													58	2	1	B	•	•	↕	↕	⑥	↕
Shift Right Arithmetic	ASR							67	7	2	77	6	3				M	•	•	↕	↕	⑥	↕
	ASRA													47	2	1	A	•	•	↕	↕	⑥	↕
	ASRB													57	2	1	B	•	•	↕	↕	⑥	↕
Shift Right Logic	LSR							64	7	2	74	6	3				M	•	•	R	↕	⑥	↕
	LSRA													44	2	1	A	•	•	R	↕	⑥	↕
	LSRB													54	2	1	B	•	•	R	↕	⑥	↕
Store Accum	STAA				97	4	2	A7	6	2	B7	5	3				A → M	•	•	↕	↕	R	•
	STAB				D7	4	2	E7	6	2	F7	5	3				B → M	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	5	2	B0	4	3				A − M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	5	2	F0	4	3				B − M → B	•	•	↕	↕	↕	↕
Subtract Accums	SBA													10	2	1	A − B → A	•	•	↕	↕	↕	↕
Subtr with Carry	SBCA	82	2	2	92	3	2	A2	5	2	B2	4	3				A − M − C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	5	2	F2	4	3				B − M − C → B	•	•	↕	↕	↕	↕
Transfer Accums	TAB													16	2	1	A → B	•	•	↕	↕	R	•
	TBA													17	2	1	B → A	•	•	↕	↕	R	•
Test Zero or Minus	TST							6D	7	2	7D	6	3				M − 00	•	•	↕	↕	R	R
	TSTA													4D	2	1	A − 00	•	•	↕	↕	R	R
	TSTB													5D	2	1	B − 00	•	•	↕	↕	R	R
																		H	I	N	Z	V	C

ADDRESSING MODES — BOOLEAN/ARITHMETIC OPERATION — CONDITION CODE REGISTER

LEGEND

OP Operation Code (Hexadecimal)
~ Number of MPU Cycles
# Number of Program Bytes
+ Arithmetic Plus
− Arithmetic Minus
· Boolean AND
+ Boolean Inclusive OR
⊕ Boolean Exclusive OR
M̄ Complement of M
→ Transfer Into
0 Bit Zero
00 Byte Zero
M_{SP} Contents of memory location pointed to be Stack Pointer

Note: Accumulator addressing mode instructions are included in the column for IMPLIED addressing

CONDITION CODE SYMBOLS

H Half-carry from bit 3
I Interrupt mask
N Negative (sign bit)
Z Zero (byte)
V Overflow, 2's complement
C Carry from bit 7
R Reset Always
S Set Always
↕ Test and set if true, cleared otherwise
• Not Affected

TABLE 4 – INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

POINTER OPERATIONS	MNEMONIC	IMMED OP	IMMED ~	IMMED =	DIRECT OP	DIRECT ~	DIRECT =	INDEX OP	INDEX ~	INDEX =	EXTND OP	EXTND ~	EXTND =	IMPLIED OP	IMPLIED ~	IMPLIED =	BOOLEAN/ARITHMETIC OPERATION	COND. CODE REG. 5 H	4 I	3 N	2 Z	1 V	0 C
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	6	2	BC	5	3				$X_H - M, X_L - (M+1)$	•	•	⑦	↕	⑧	•
Decrement Index Reg	DEX													09	4	1	$X - 1 \rightarrow X$	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	4	1	$SP - 1 \rightarrow SP$	•	•	•	•	•	•
Increment Index Reg	INX													08	4	1	$X + 1 \rightarrow X$	•	•	•	↕	•	•
Increment Stack Pntr	INS													31	4	1	$SP + 1 \rightarrow SP$	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	6	2	FE	5	3				$M \rightarrow X_H, (M+1) \rightarrow X_L$	•	•	⑨	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	6	2	BE	5	3				$M \rightarrow SP_H, (M+1) \rightarrow SP_L$	•	•	⑨	↕	R	•
Store Index Reg	STX				DF	5	2	EF	7	2	FF	6	3				$X_H \rightarrow M, X_L \rightarrow (M+1)$	•	•	⑨	↕	R	•
Store Stack Pntr	STS				9F	5	2	AF	7	2	BF	6	3				$SP_H \rightarrow M, SP_L \rightarrow (M+1)$	•	•	⑨	↕	R	•
Indx Reg → Stack Pntr	TXS													35	4	1	$X - 1 \rightarrow SP$	•	•	•	•	•	•
Stack Pntr → Indx Reg	TSX													30	4	1	$SP + 1 \rightarrow X$	•	•	•	•	•	•

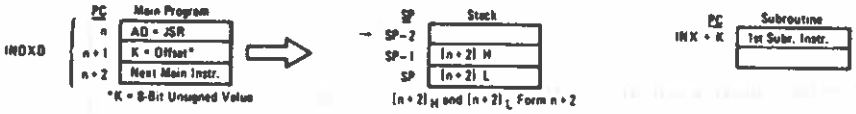
TABLE 5 – JUMP AND BRANCH INSTRUCTIONS

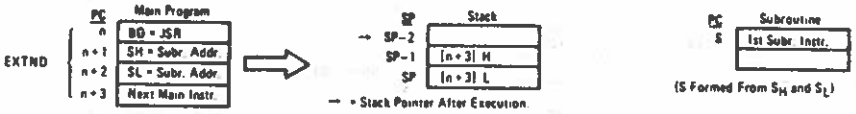
OPERATIONS	MNEMONIC	RELATIVE OP	RELATIVE ~	RELATIVE =	INDEX OP	INDEX ~	INDEX =	EXTND OP	EXTND ~	EXTND =	IMPLIED OP	IMPLIED ~	IMPLIED =	BRANCH TEST	COND. CODE REG. 5 N	4 I	3 N	2 Z	1 V	0 C
Branch Always	BRA	20	4	2										None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	4	2										C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	4	2										C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	4	2										Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE	2C	4	2										N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	4	2										Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	4	2										C + Z = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	4	2										Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	4	2										C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	4	2										N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI	2B	4	2										N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	4	2										Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	4	2										V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	4	2										V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	4	2										N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	8	2											•	•	•	•	•	•
Jump	JMP				6E	4	2	7E	3	3				See Special Operations	•	•	•	•	•	•
Jump To Subroutine	JSR				AD	8	2	BD	9	3					•	•	•	•	•	•
No Operation	NOP										01	2	1	Advances Prog. Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI										3B	10	1		⑩					
Return From Subroutine	RTS										39	5	1		•	•	•	•	•	•
Software Interrupt	SWI										3F	12	1	See Special Operations	•	•	•	•	•	•
Wait for Interrupt*	WAI										3E	9	1		•	⑪	•	•	•	•

*WAI puts Address Bus, R/W, and Data Bus in the three-state mode while VMA is held low.

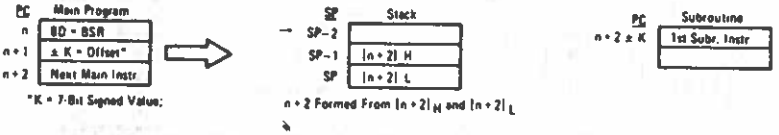
SPECIAL OPERATIONS

JSR, JUMP TO SUBROUTINE:





BSR, BRANCH TO SUBROUTINE:



JMP, JUMP:

RTS, RETURN FROM SUBROUTINE:

RTI, RETURN FROM INTERRUPT:

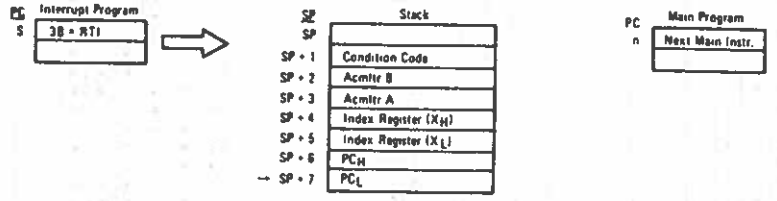


TABLE 6 – CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

OPERATIONS	MNEMONIC	IMPLIED OP	~	=	BOOLEAN OPERATION	COND. CODE REG. 5 H	4 I	3 N	2 Z	1 V	0 C
Clear Carry	CLC	0C	2	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	2	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	2	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	2	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	2	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	2	1	1 → V	•	•	•	•	S	•
Acmltr A → CCR	TAP	06	2	1	A → CCR	——— (12) ———					
CCR → Acmltr A	TPA	07	2	1	CCR → A	•	•	•	•	•	•

CONDITION CODE REGISTER NOTES: (Bit set if test is true and cleared otherwise)

1 (Bit V) Test: Result = 10000000?
2 (Bit C) Test: Result = 00000000?
3 (Bit C) Test: Decimal value of most significant BCD Character greater than nine? (Not cleared if previously set)
4 (Bit V) Test: Operand = 10000000 prior to execution?
5 (Bit V) Test: Operand = 01111111 prior to execution?
6 (Bit V) Test: Set equal to result of N⊕C after shift has occurred
7 (Bit N) Test: Sign bit of most significant (MS) byte = 1?
8 (Bit V) Test: 2's complement overflow from subtraction of MS bytes?
9 (Bit N) Test: Result less than zero? (Bit 15 = 1)
10 (All) Load Condition Code Register from Stack. (See Special Operations)
11 (Bit I) Set when interrupt occurs. If previously set, a Non Maskable Interrupt is required to exit the wait state
12 (All) Set according to the contents of Accumulator A

TABLE 7 – INSTRUCTION ADDRESSING MODES AND ASSOCIATED EXECUTION TIMES
(Times in Machine Cycles)

	(Dual Operand)	ACCX	Immediate	Direct	Extended	Indexed	Implied	Relative
ABA		•	•	•	•	•	2	•
ADC	x	•	2	3	4	5	•	•
ADD	x	•	2	3	4	5	•	•
AND	x	•	2	3	4	5	•	•
ASL		2	•	•	6	7	•	•
ASR		2	•	•	6	7	•	•
BCC		•	•	•	•	•	•	4
BCS		•	•	•	•	•	•	4
BEA		•	•	•	•	•	•	4
BGE		•	•	•	•	•	•	4
BGT		•	•	•	•	•	•	4
BHI		•	•	•	•	•	•	4
BIT	x	•	2	3	4	5	•	•
BLE		•	•	•	•	•	•	4
BLS		•	•	•	•	•	•	4
BLT		•	•	•	•	•	•	4
BMI		•	•	•	•	•	•	4
BNE		•	•	•	•	•	•	4
BPL		•	•	•	•	•	•	4
BRA		•	•	•	•	•	•	4
BSR		•	•	•	•	•	•	8
BVC		•	•	•	•	•	•	4
BVS		•	•	•	•	•	•	4
CBA		•	•	•	•	•	2	•
CLC		•	•	•	•	•	2	•
CLI		•	•	•	•	•	2	•
CLR		2	•	•	6	7	•	•
CLV		•	•	•	•	•	2	•
CMP	x	•	2	3	4	5	•	•
COM		2	•	•	6	7	•	•
CPX		•	3	4	5	6	•	•
DAA		•	•	•	•	•	2	•
DEC		2	•	•	6	7	•	•
DES		•	•	•	•	•	4	•
DEX		•	•	•	•	•	4	•
EOR	x	•	2	3	4	5	•	•

	(Dual Operand)	ACCX	Immediate	Direct	Extended	Indexed	Implied
INC		2	•	•	6	7	•
INS		•	•	•	•	•	4
INX		•	•	•	•	•	4
JMP		•	•	•	3	4	•
JSR		•	•	•	9	8	•
LDA	x	•	2	3	4	5	•
LDS		•	3	4	5	6	•
LDX		•	3	4	5	6	•
LSR		2	•	•	6	7	•
NEG		2	•	•	6	7	•
NOP		•	•	•	•	•	2
ORA	x	•	2	3	4	5	•
PSH		•	•	•	•	•	4
PUL		•	•	•	•	•	4
ROL		2	•	•	6	7	•
ROR		2	•	•	6	7	•
RTI		•	•	•	•	•	10
RTS		•	•	•	•	•	5
SBA		•	•	•	•	•	2
SBC	x	•	2	3	4	5	•
SEC		•	•	•	•	•	2
SEI		•	•	•	•	•	2
SEV		•	•	•	•	•	2
STA	x	•	•	4	5	6	•
STS		•	•	5	6	7	•
STX		•	•	5	6	7	•
SUB	x	•	2	3	4	5	•
SWI		•	•	•	•	•	12
TAB		•	•	•	•	•	2
TAP		•	•	•	•	•	2
TBA		•	•	•	•	•	2
TPA		•	•	•	•	•	2
TST		2	•	•	6	7	•
TSX		•	•	•	•	•	4
TSX		•	•	•	•	•	4
WAI		•	•	•	•	•	9

NOTE Interrupt time is 12 cycles from the end of the instruction being executed, except following a WAI instruction. Then it is 4 cycles.

SUMMARY OF CYCLE BY CYCLE OPERATION

Table 8 provides a detailed description of the information present on the Address Bus, Data Bus, Valid Memory Address line (VMA), and the Read/Write line (R/W) during each cycle for each instruction.

This information is useful in comparing actual with expected results during debug of both software and hardware as the control program is executed. The information is categorized in groups according to Addressing Mode and Number of Cycles per instruction. (In general, instructions with the same Addressing Mode and Number of Cycles execute in the same manner; exceptions are indicated in the table.)

TABLE 8 – OPERATION SUMMARY

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/W Line	Data Bus
IMMEDIATE						
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	2	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Operand Data
CPX LDS LDX	3	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Operand Data (High Order Byte)
		3	1	Op Code Address + 2	1	Operand Data (Low Order Byte)
DIRECT						
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	3	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand
		3	1	Address of Operand	1	Operand Data
CPX LDS LDX	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand
		3	1	Address of Operand	1	Operand Data (High Order Byte)
		4	1	Operand Address + 1	1	Operand Data (Low Order Byte)
STA	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Destination Address
		3	0	Destination Address	1	Irrelevant Data (Note 1)
		4	1	Destination Address	0	Data from Accumulator
STS STX	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand
		3	0	Address of Operand	1	Irrelevant Data (Note 1)
		4	1	Address of Operand	0	Register Data (High Order Byte)
		5	1	Address of Operand + 1	0	Register Data (Low Order Byte)
INDEXED						
JMP	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	1	Index Register Plus Offset	1	Operand Data
CPX LDS LDX	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	1	Index Register Plus Offset	1	Operand Data (High Order Byte)
		6	1	Index Register Plus Offset + 1	1	Operand Data (Low Order Byte)

TABLE 8 – OPERATION SUMMARY (Continued)

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/W Line	Data Bus
INDEXED (Continued)						
STA	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	0	Index Register Plus Offset	1	Irrelevant Data (Note 1)
		6	1	Index Register Plus Offset	0	Operand Data
ASL LSR ASR NEG CLR ROL COM ROR DEC TST INC	7	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	1	Index Register Plus Offset	1	Current Operand Data
		6	0	Index Register Plus Offset	1	Irrelevant Data (Note 1)
		7	1/0 (Note 3)	Index Register Plus Offset	0	New Operand Data (Note 3)
STS STX	7	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	0	Index Register Plus Offset	1	Irrelevant Data (Note 1)
		6	1	Index Register Plus Offset	0	Operand Data (High Order Byte)
		7	1	Index Register Plus Offset + 1	0	Operand Data (Low Order Byte)
JSR	8	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer	0	Return Address (Low Order Byte)
		5	1	Stack Pointer – 1	0	Return Address (High Order Byte)
		6	0	Stack Pointer – 2	1	Irrelevant Data (Note 1)
		7	0	Index Register	1	Irrelevant Data (Note 1)
		8	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
EXTENDED						
JMP	3	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Jump Address (High Order Byte)
		3	1	Op Code Address + 2	1	Jump Address (Low Order Byte)
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	1	Address of Operand	1	Operand Data
CPX LDS LDX	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	1	Address of Operand	1	Operand Data (High Order Byte)
		5	1	Address of Operand + 1	1	Operand Data (Low Order Byte)
STA A STA B	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Destination Address (High Order Byte)
		3	1	Op Code Address + 2	1	Destination Address (Low Order Byte)
		4	0	Operand Destination Address	1	Irrelevant Data (Note 1)
		5	1	Operand Destination Address	0	Data from Accumulator
ASL LSR ASR NEG CLR ROL COM ROR DEC TST INC	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	1	Address of Operand	1	Current Operand Data
		5	0	Address of Operand	1	Irrelevant Data (Note 1)
		6	1/0 (Note 3)	Address of Operand	0	New Operand Data (Note 3)

16

TABLE 8 – OPERATION SUMMARY (Continued)

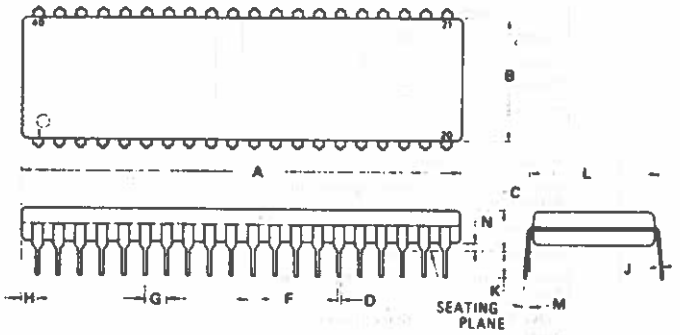
Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/W Line	Data Bus
EXTENDED (Continued)						
STS STX	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	0	Address of Operand	1	Irrelevant Data (Note 1)
		5	1	Address of Operand	0	Operand Data (High Order Byte)
		6	1	Address of Operand + 1	0	Operand Data (Low Order Byte)
JSR	9	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Subroutine (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Subroutine (Low Order Byte)
		4	1	Subroutine Starting Address	1	Op Code of Next Instruction
		5	1	Stack Pointer	0	Return Address (Low Order Byte)
		6	1	Stack Pointer – 1	0	Return Address (High Order Byte)
		7	0	Stack Pointer – 2	1	Irrelevant Data (Note 1)
		8	0	Op Code Address + 2	1	Irrelevant Data (Note 1)
		9	1	Op Code Address + 2	1	Address of Subroutine (Low Order Byte)
INHERENT						
ABA DAA SEC ASL DEC SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
DES DEX INS INX	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	0	Previous Register Contents	1	Irrelevant Data (Note 1)
		4	0	New Register Contents	1	Irrelevant Data (Note 1)
PSH	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	1	Stack Pointer	0	Accumulator Data
		4	0	Stack Pointer – 1	1	Accumulator Data
PUL	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	0	Stack Pointer	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer + 1	1	Operand Data from Stack
TSX	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	0	Stack Pointer	1	Irrelevant Data (Note 1)
		4	0	New Index Register	1	Irrelevant Data (Note 1)
TXS	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	0	Index Register	1	Irrelevant Data
		4	0	New Stack Pointer	1	Irrelevant Data
RTS	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Irrelevant Data (Note 2)
		3	0	Stack Pointer	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer + 1	1	Address of Next Instruction (High Order Byte)
		5	1	Stack Pointer + 2	1	Address of Next Instruction (Low Order Byte)

TABLE 8 – OPERATION SUMMARY (Continued)

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/W Line	Data Bus
INHERENT (Continued)						
WAI	9	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	1	Stack Pointer	0	Return Address (Low Order Byte)
		4	1	Stack Pointer – 1	0	Return Address (High Order Byte)
		5	1	Stack Pointer – 2	0	Index Register (Low Order Byte)
		6	1	Stack Pointer – 3	0	Index Register (High Order Byte)
		7	1	Stack Pointer – 4	0	Contents of Accumulator A
		8	1	Stack Pointer – 5	0	Contents of Accumulator B
		9	1	Stack Pointer – 6 (Note 4)	1	Contents of Cond. Code Register
RTI	10	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Irrelevant Data (Note 2)
		3	0	Stack Pointer	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer + 1	1	Contents of Cond. Code Register from Stack
		5	1	Stack Pointer + 2	1	Contents of Accumulator B from Stack
		6	1	Stack Pointer + 3	1	Contents of Accumulator A from Stack
		7	1	Stack Pointer + 4	1	Index Register from Stack (High Order Byte)
		8	1	Stack Pointer + 5	1	Index Register from Stack (Low Order Byte)
		9	1	Stack Pointer + 6	1	Next Instruction Address from Stack (High Order Byte)
		10	1	Stack Pointer + 7	1	Next Instruction Address from Stack (Low Order Byte)
SWI	12	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Irrelevant Data (Note 1)
		3	1	Stack Pointer	0	Return Address (Low Order Byte)
		4	1	Stack Pointer – 1	0	Return Address (High Order Byte)
		5	1	Stack Pointer – 2	0	Index Register (Low Order Byte)
		6	1	Stack Pointer – 3	0	Index Register (High Order Byte)
		7	1	Stack Pointer – 4	0	Contents of Accumulator A
		8	1	Stack Pointer – 5	0	Contents of Accumulator B
		9	1	Stack Pointer – 6	0	Contents of Cond. Code Register
		10	0	Stack Pointer – 7	1	Irrelevant Data (Note 1)
		11	1	Vector Address FFFA (Hex)	1	Address of Subroutine (High Order Byte)
		12	1	Vector Address FFF8 (Hex)	1	Address of Subroutine (Low Order Byte)
RELATIVE						
BCC BHI BNE BCS BLE BPL BEQ BLS BRA BGE BLT BVC BGT BMI BVS	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Branch Offset
		3	0	Op Code Address + 2	1	Irrelevant Data (Note 1)
		4	0	Branch Address	1	Irrelevant Data (Note 1)
BSR	8	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Branch Offset
		3	0	Return Address of Main Program	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer	0	Return Address (Low Order Byte)
		5	1	Stack Pointer – 1	0	Return Address (High Order Byte)
		6	0	Stack Pointer – 2	1	Irrelevant Data (Note 1)
		7	0	Return Address of Main Program	1	Irrelevant Data (Note 1)
		8	0	Subroutine Address	1	Irrelevant Data (Note 1, Note 5)

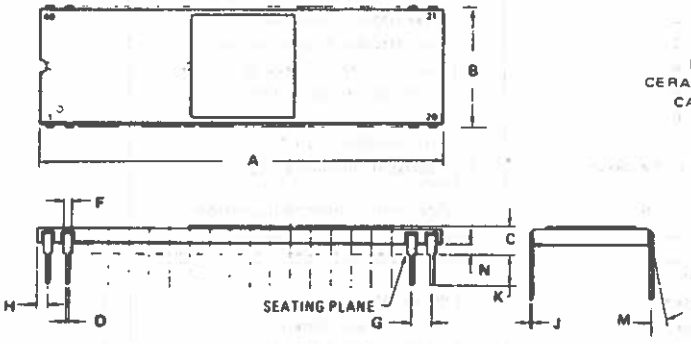
Note 1. If device which is addressed during this cycle uses VMA, then the Data Bus will go to the high impedance three-state condition. Depending on bus capacitance, data from the previous cycle may be retained on the Data Bus.
Note 2. Data is ignored by the MPU.
Note 3. For TST, VMA = 0 and Operand data does not change.
Note 4. While the MPU is waiting for the interrupt, Bus Available will go high, VMA is low.
Note 5. MS Byte = MS Byte of BSR instruction address, LS Byte = LS Byte of subroutine address.

P SUFFIX
PLASTIC PACKAGE
CASE 711-03



NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25 mm (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.
4. 711-02 OBSOLETE, NEW STANDARD 711-03.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	51.69	52.45	2.035	2.065
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.02	0.020	0.040



L SUFFIX
CERAMIC PACKAGE
CASE 715-02

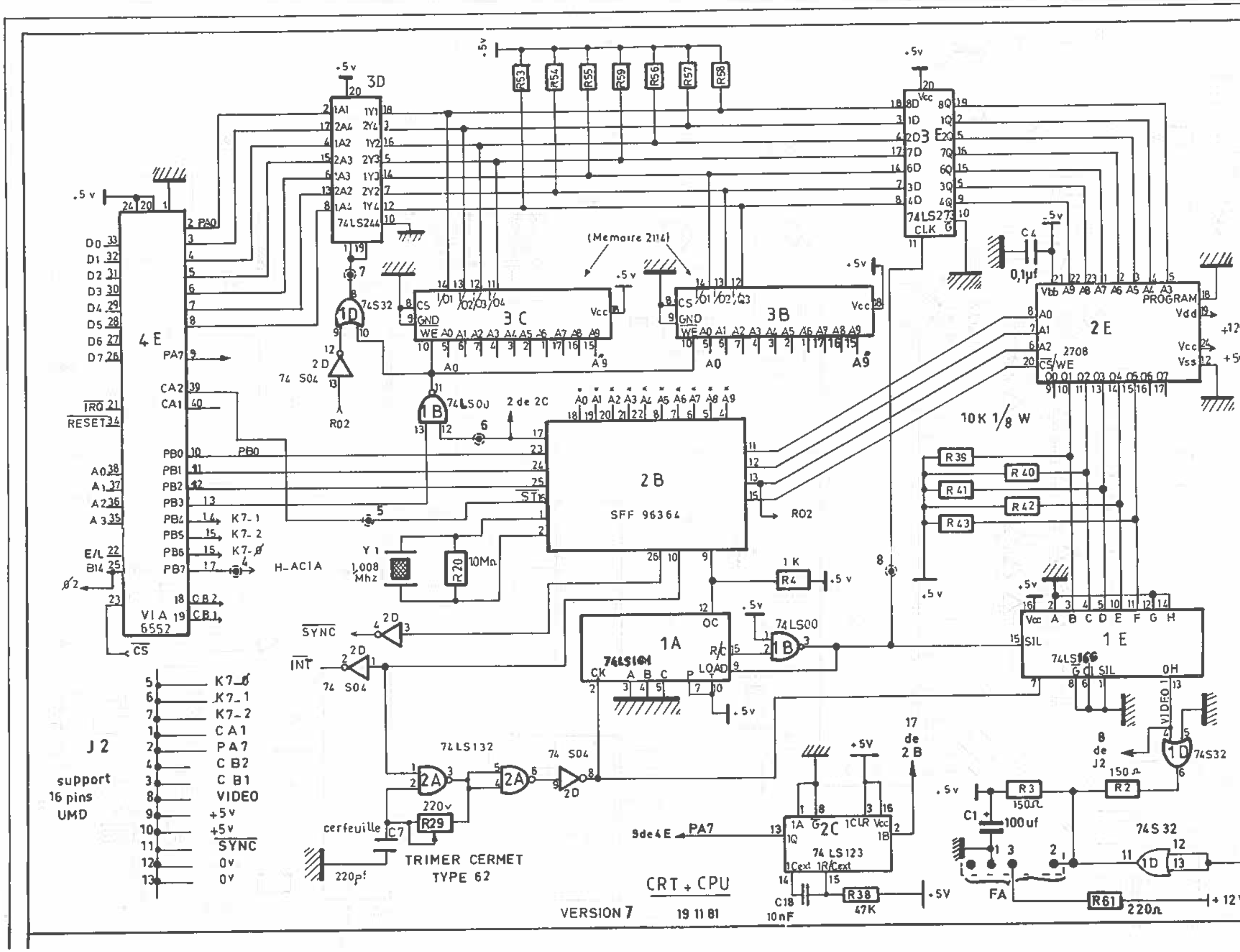
	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	50.29	51.31	1.980	2.020
B	14.86	15.62	0.585	0.615
C	2.54	4.19	0.100	0.165
D	0.38	0.53	0.015	0.021
F	0.76	1.40	0.030	0.055
G	2.54 BSC		0.100 BSC	
H	0.76	1.78	0.030	0.070
J	0.20	0.33	0.008	0.013
K	2.54	4.19	0.100	0.165
M	0°	10°	0°	10°
N	0.51	1.52	0.020	0.060

NOTE
1. LEADS, TRUE POSITIONED WITHIN 0.25 mm (0.010) DIA (AT SEATING PLANE), AT MAX. MAT'L CONDITION.

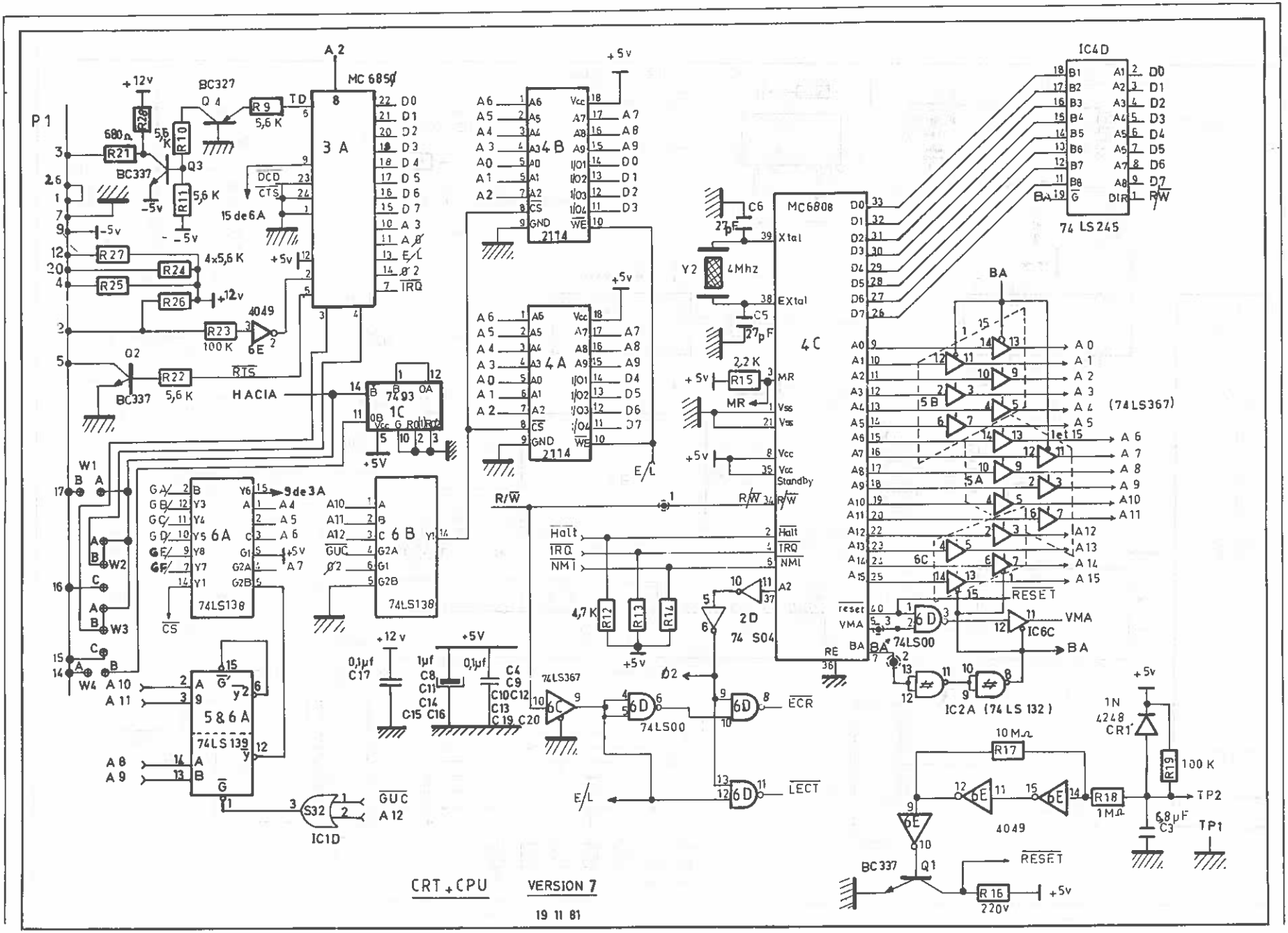
- VIII -

- SCHEMAS LOGIQUES DE FONCTIONNEMENT DES CARTES DE GOUPIL 2 -

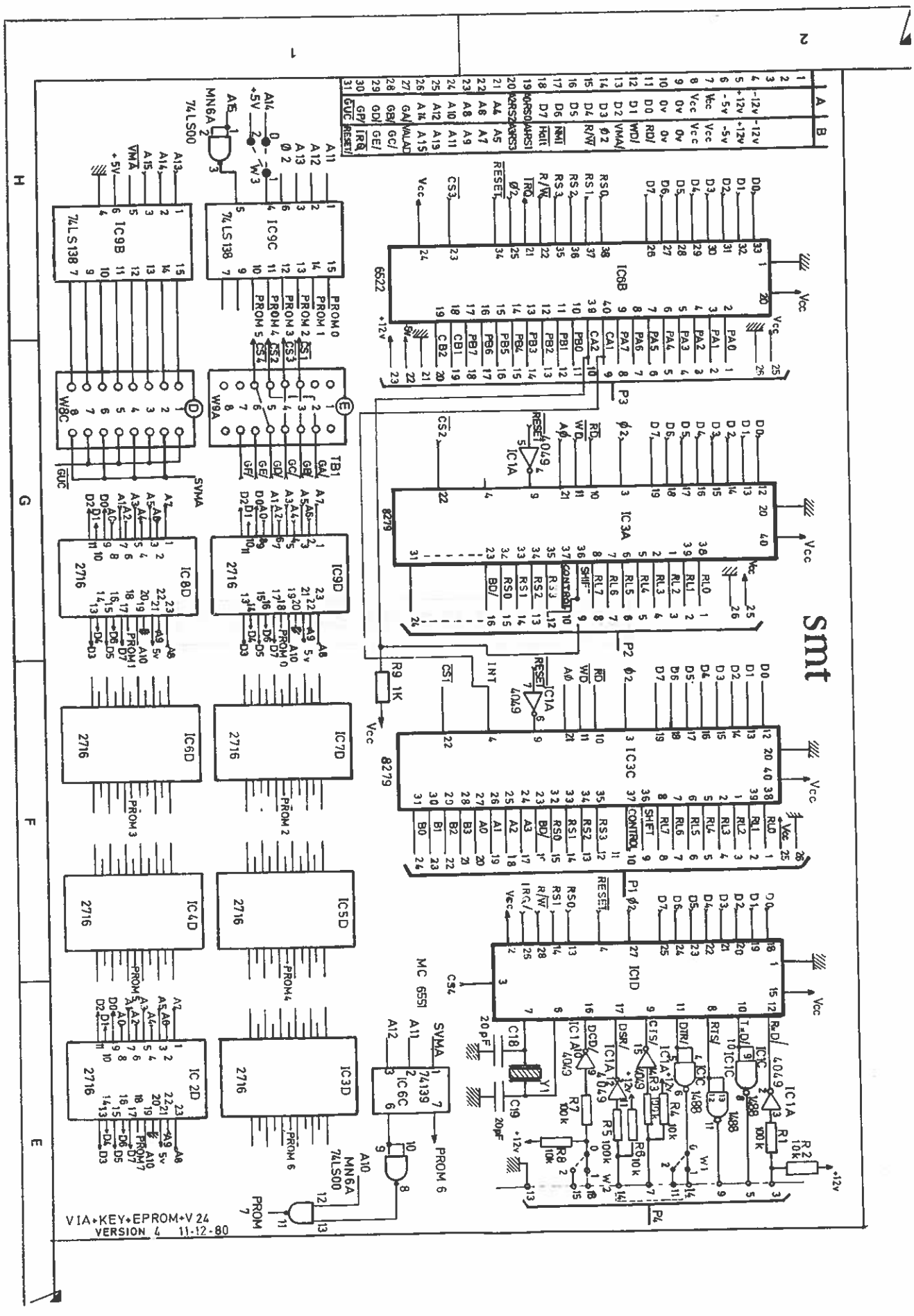
SCHEMAS DE LA CARTE CRT-CPU



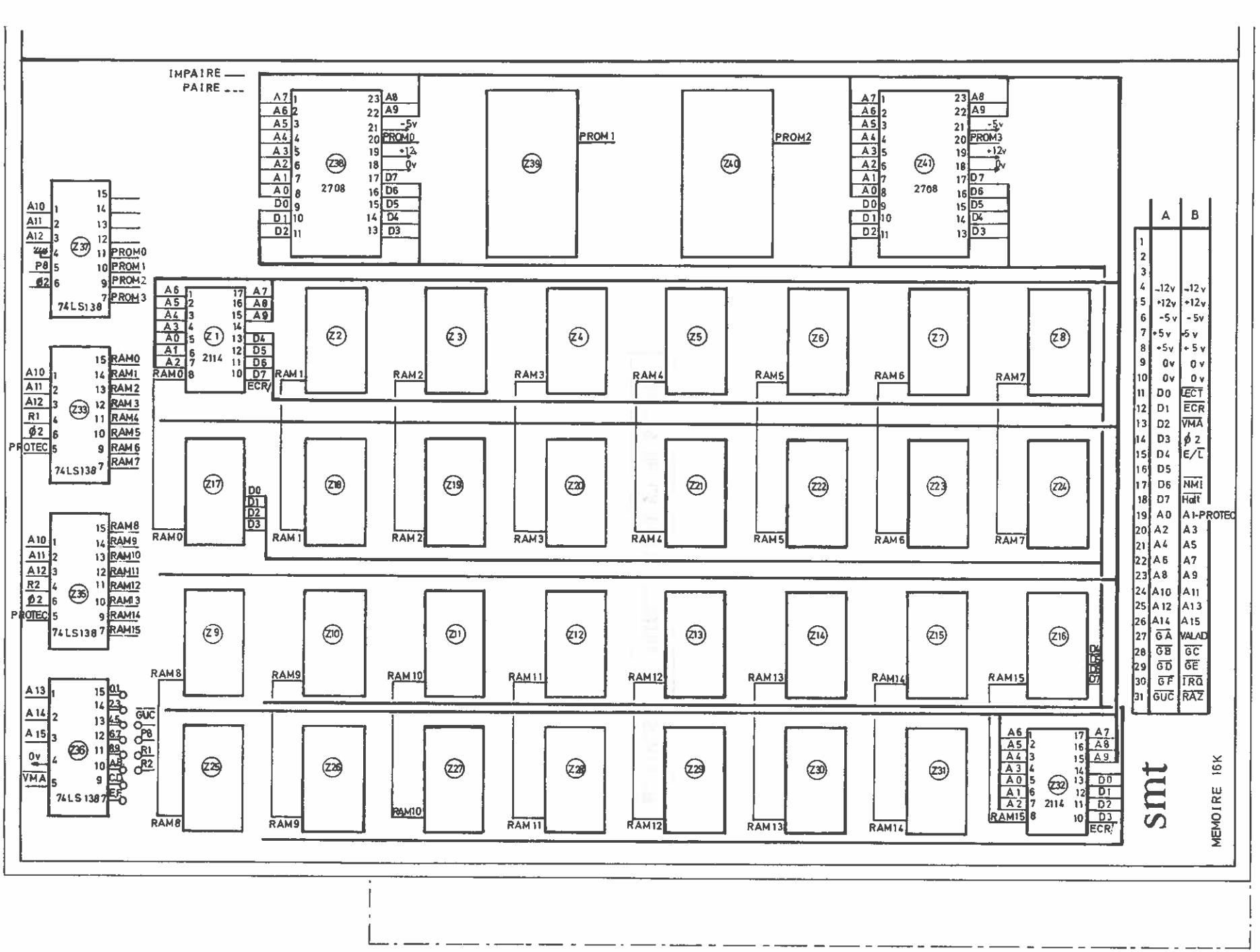
	A	B
1		
2		
3	MR	BA
4	-12v	-12v
5	+12v	+12v
6	-5v	-5v
7	+5v	+5v
8	+5v	+5v
9	0v	0v
10	0v	0v
11	D0	Lect/
12	D1	Ecr/
13	D2	VMA/
14	D3	Ø2
15	D4	E/L
16	D5	Hext
17	D6	NMI
18	D7	Halt
19	A0	A1
20	A2	A3
21	A4	A5
22	A6	A7
23	A8	A9
24	A10	A11
25	A12	A13
26	A14	A15
27	GA/	VALAD
28	GB/	GC/
29	GD/	GE/
30	GF/	IRQ
31	GUC	RESET


MC 6809
3 A
4 B
2114
4 A
2114
4 C
MC6808
IC4D
74 LS245
74LS138
74LS139
74LS367
74LS00
74 S04
IC2A (74 LS 132)
IC1D
4049
74LS367
(74LS367)
CRT + CPU
VERSION 7
19 11 81

SCHEMAS DE LA CARTE VIA ENTREES-SORTIES

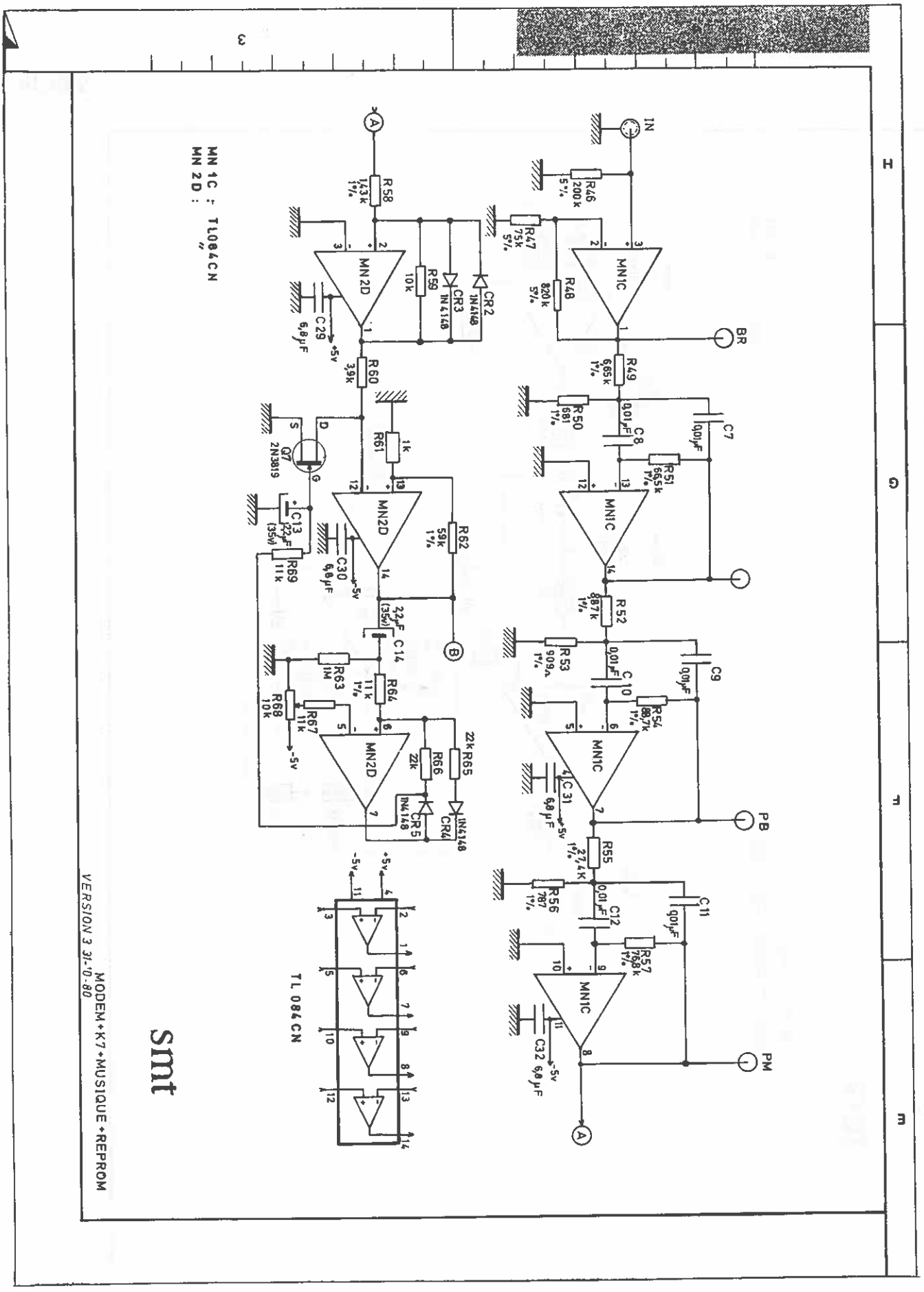

Smt
VIA+KEY+EPROM+V 24
VERSION 4 11-12-80

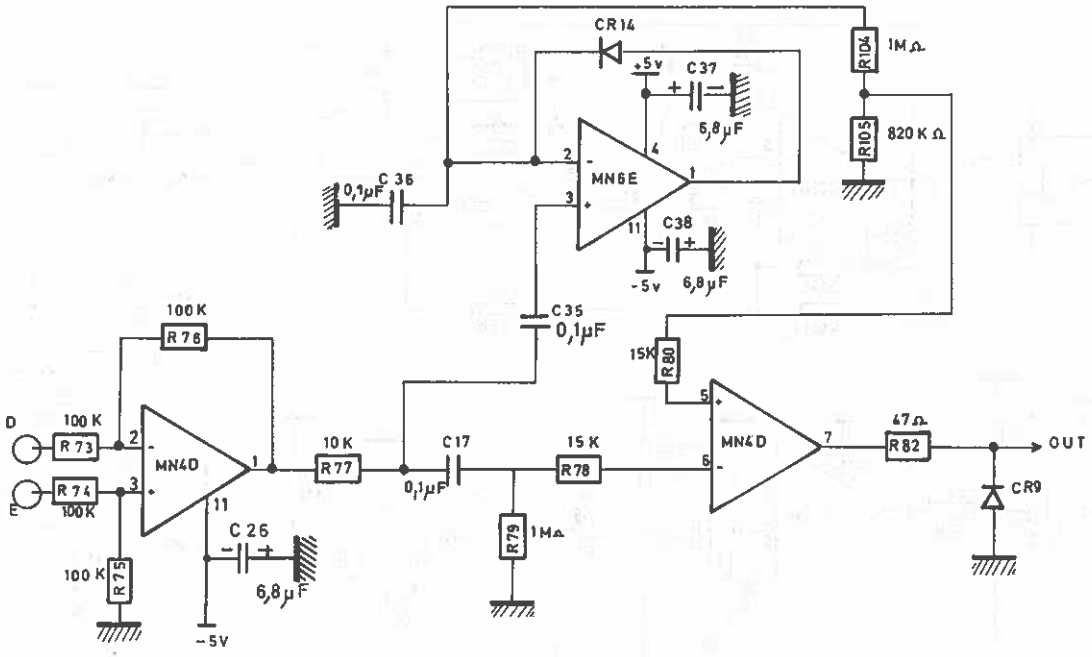
SCHEMAS DE LA CARTE MEMOIRE 16 K STATIQUE


IMPAIRE
PAIRE
2708
2114
74LS138
PROM0
PROM1
PROM2
PROM3
RAM0
RAM1
RAM2
RAM3
RAM4
RAM5
RAM6
RAM7
RAM8
RAM9
RAM10
RAM11
RAM12
RAM13
RAM14
RAM15
PROTEC
GUC
VMA
ECR/
A
B
-12v
+12v
-5v
+5v
0v
D0
D1
D2
D3
D4
D5
D6
D7
NMI
Halt
A1-PROTEC
VALAD
IRQ
RAZ
smt
MEMOIRE 16K

SCHEMAS DE LA CARTE MODEM + K7 + MEMOIRE DYNAMIQUE

- Version 3 sans paddles
- Version 6 avec paddles


MN 1C : TL084CN
MN 2D : "
IN
BR
PB
PM
MN1C
MN2D
TL 084 CN
Q7 2N3819
CR2 1N4148
CR3 1N4148
CR4 1N4148
CR5 1N4148
-5v
+5v
smt
MODEM+K7+MUSIQUE+REPROM
VERSION 3 31-10-80

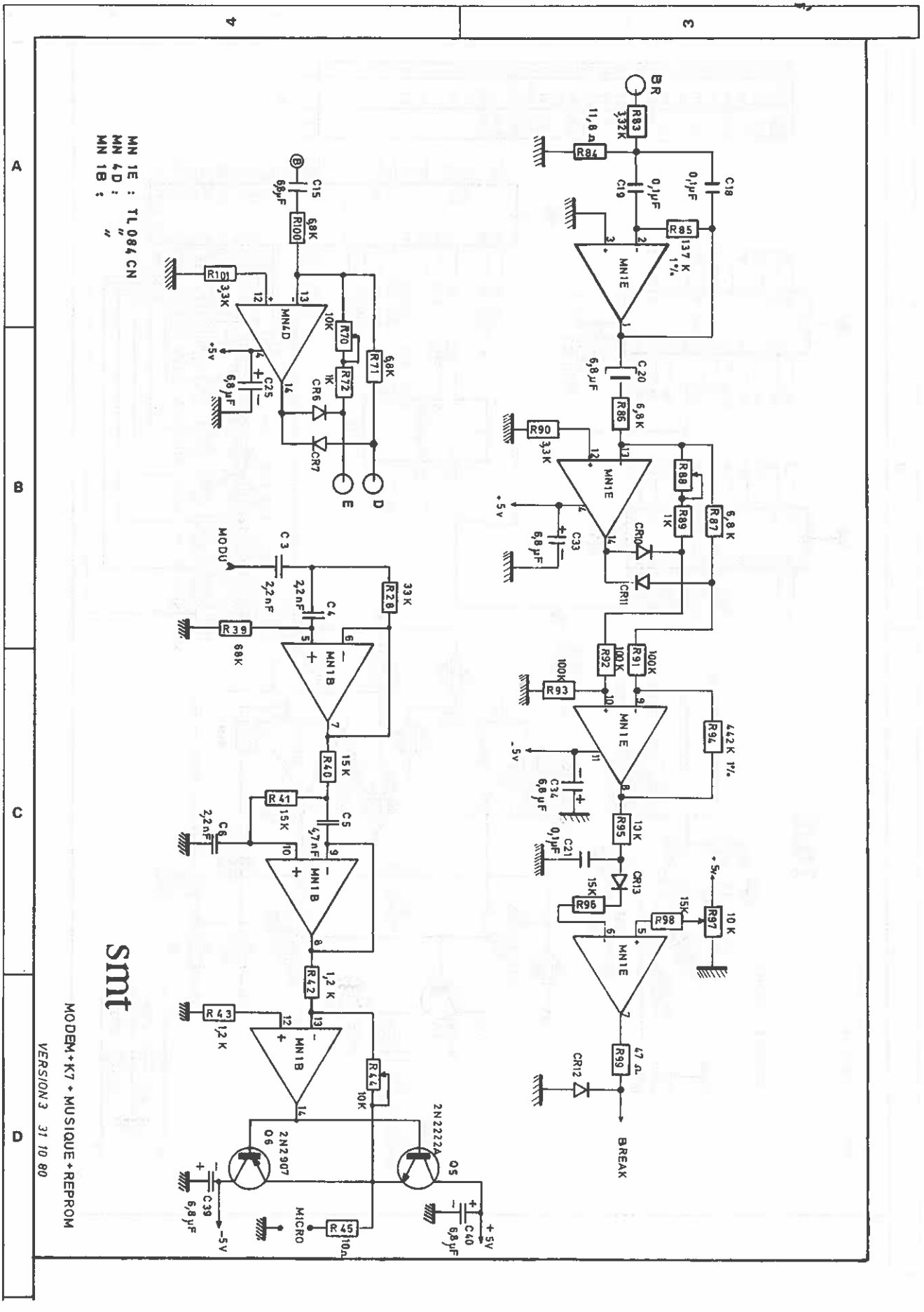


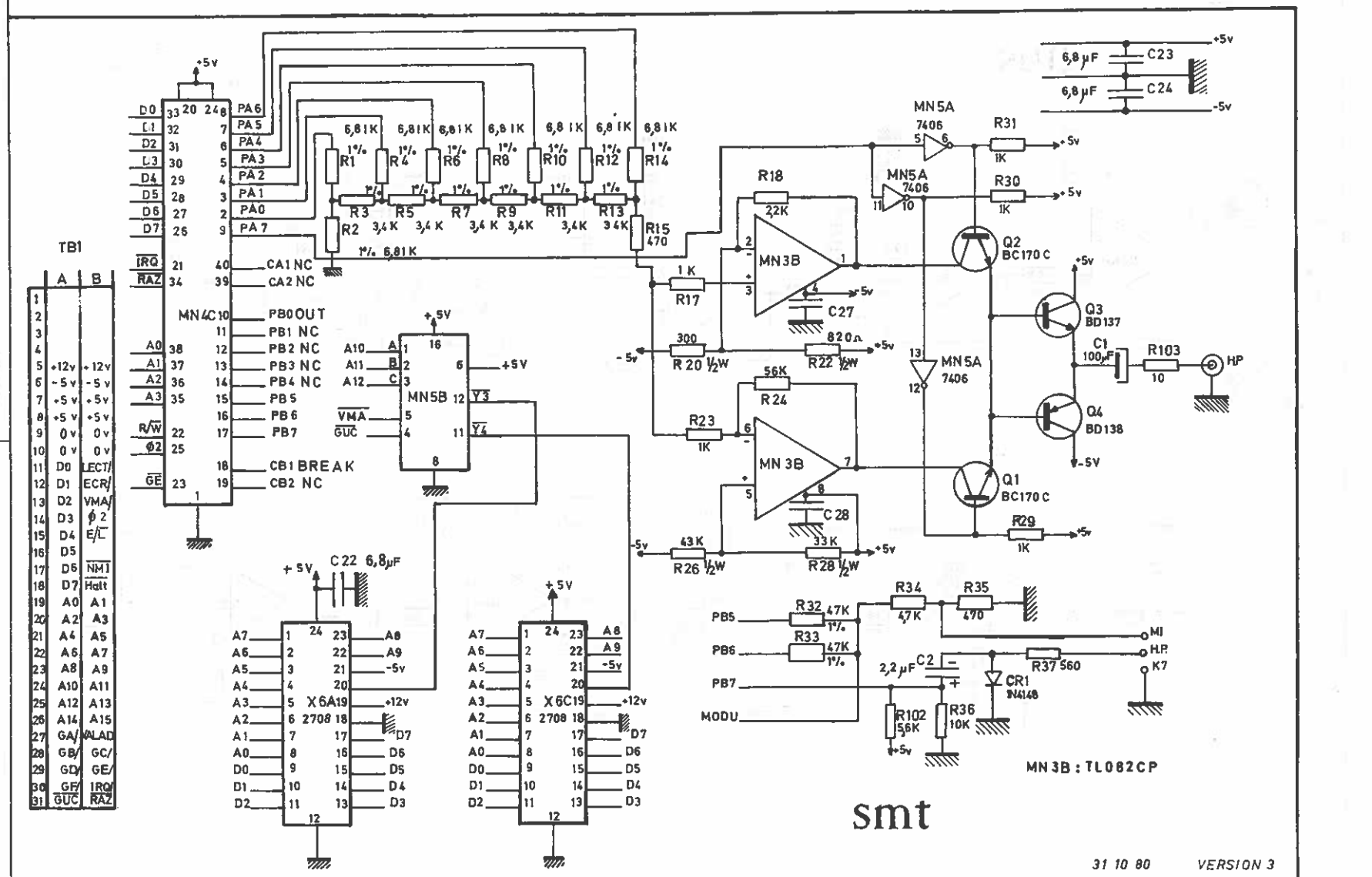
MN4D : TL084CN
MN6E : "

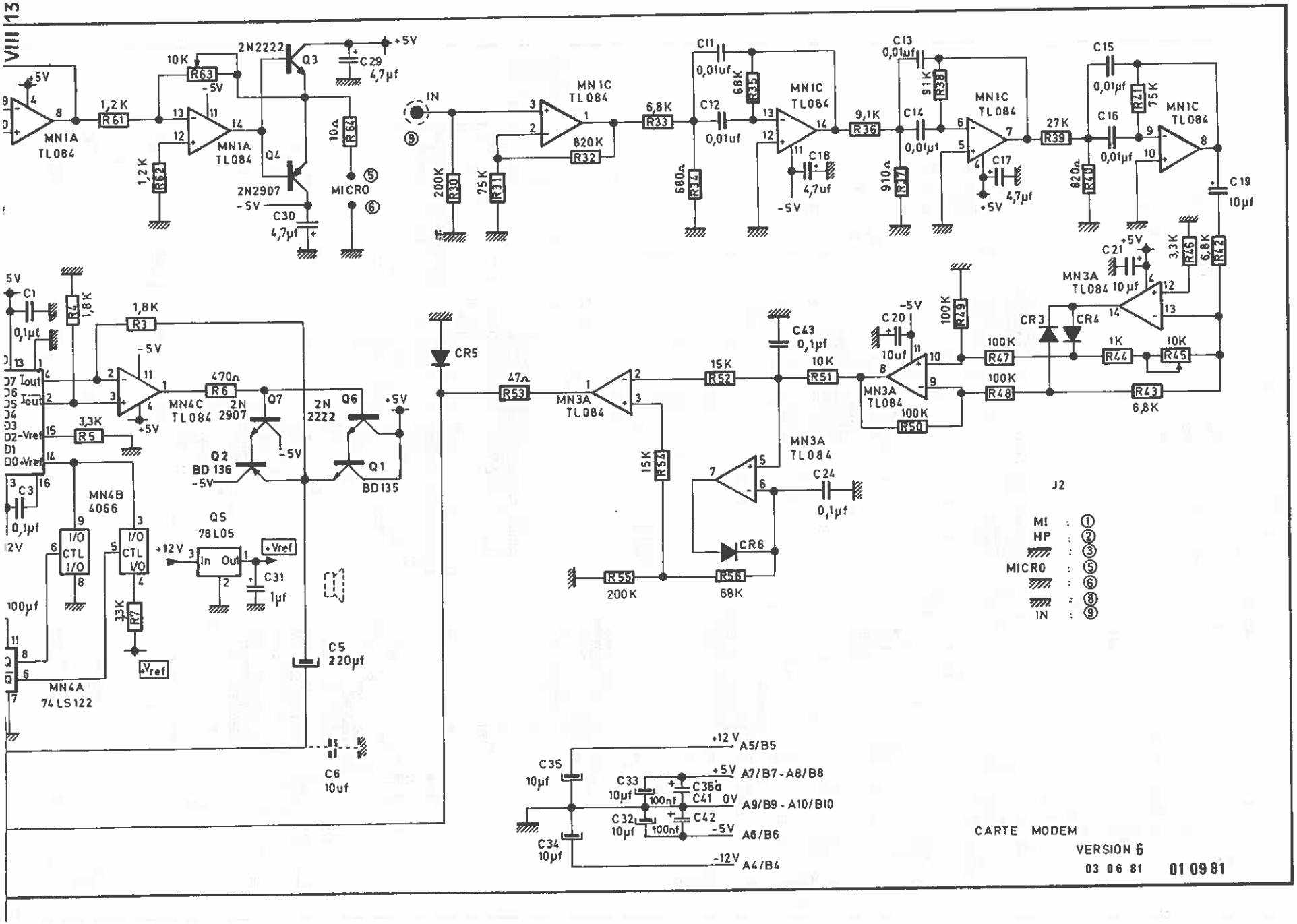
MODEM + K7 + MUSIQUE + REPROM

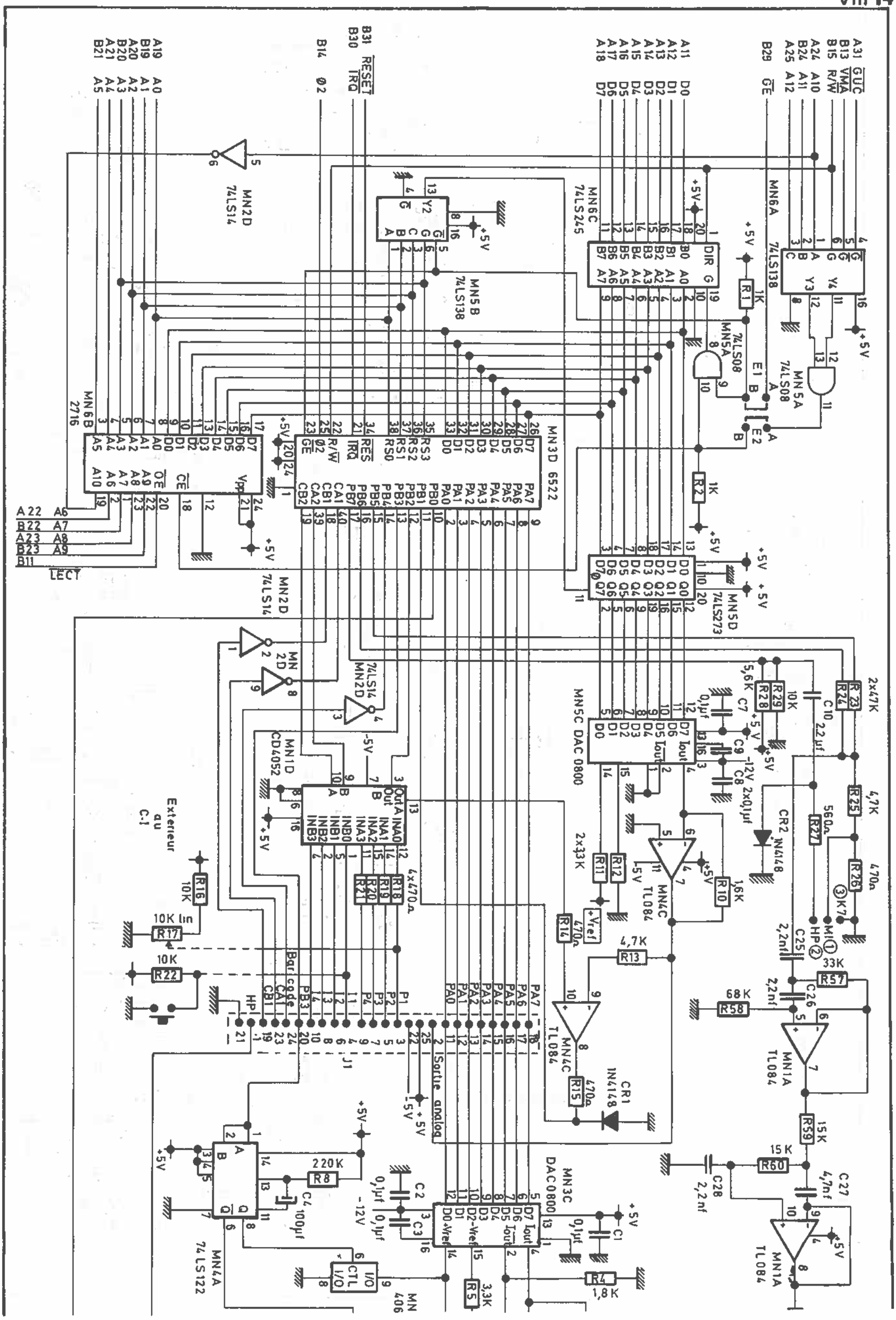
VERSION 3 31 10 80

smt


MN 1E : TL 084 CN
MN 4D : "
MN 1B : "
smt
MODEM+K7+MUSIQUE+REPROM
VERSION 3 31 10 80
BR
BREAK
MICRO
MODU
MN1E
MN4D
MN1B
2N2907
2N2222A
+5v
-5v

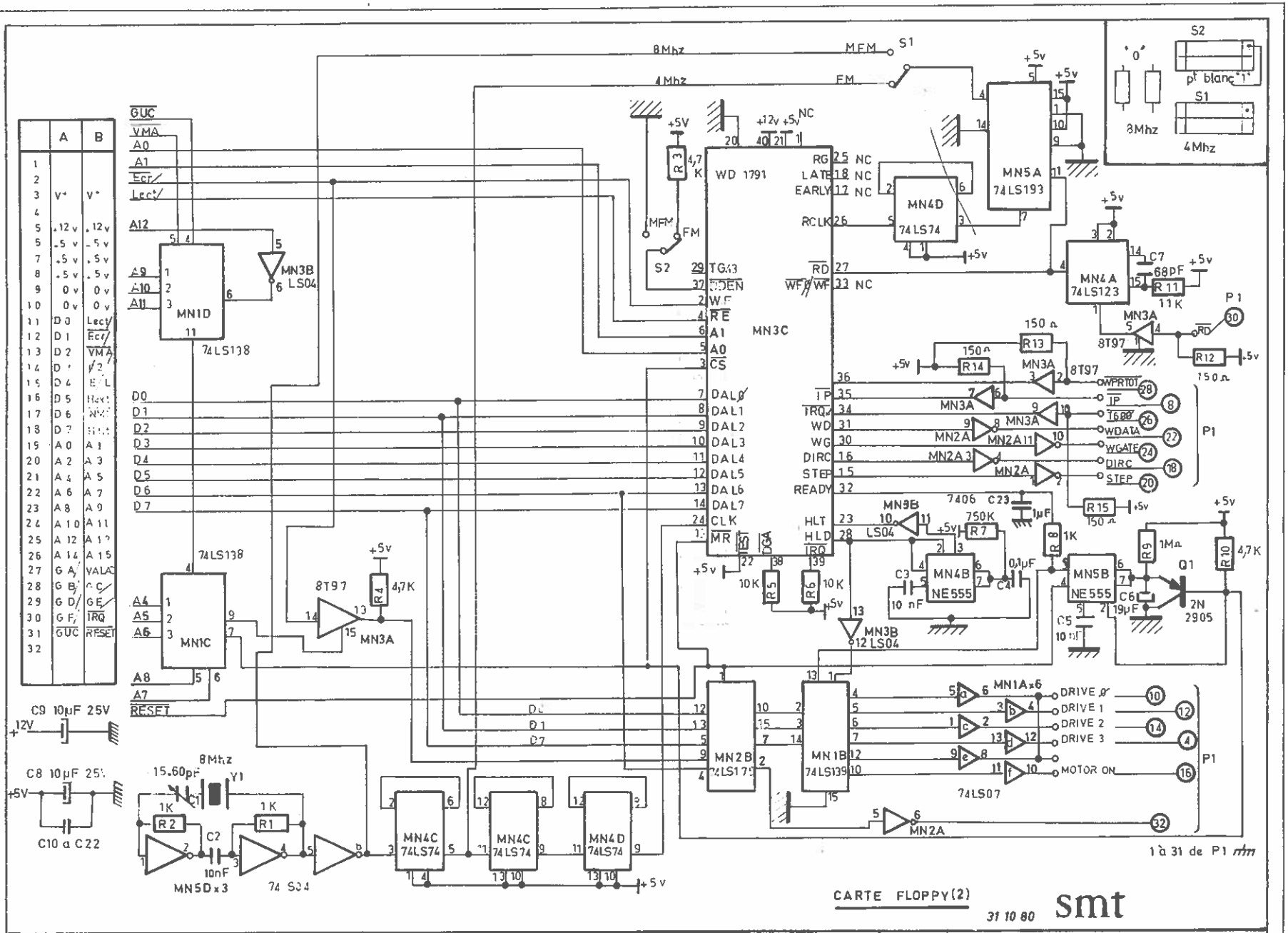

TB1
MN4C
MN5B
X6A19
X6C19
MN5A 7406
MN3B
Q1 BC170C
Q2 BC170C
Q3 BD137
Q4 BD138
CR1 1N4148
HP
MI
K7
MODU
PB5
PB6
PB7
MN3B : TL082CP
smt
31 10 80 VERSION 3
MODEM + K7 + MUSIQUE + REPROM


MN1A TL084
MN1C TL084
MN3A TL084
MN4C TL084
MN4B 4066
MN4A 74LS122
2N2222
2N2907
2N 2907
2N 2222
BD 136
BD135
78L05
MICRO
IN
J2
MI : ①
HP : ②
MICRO : ⑤
IN : ⑨
CARTE MODEM
VERSION 6
03 06 81
01 09 81
+12 V A5/B5
+5V A7/B7 - A8/B8
0V A9/B9 - A10/B10
-5V A6/B6
-12V A4/B4

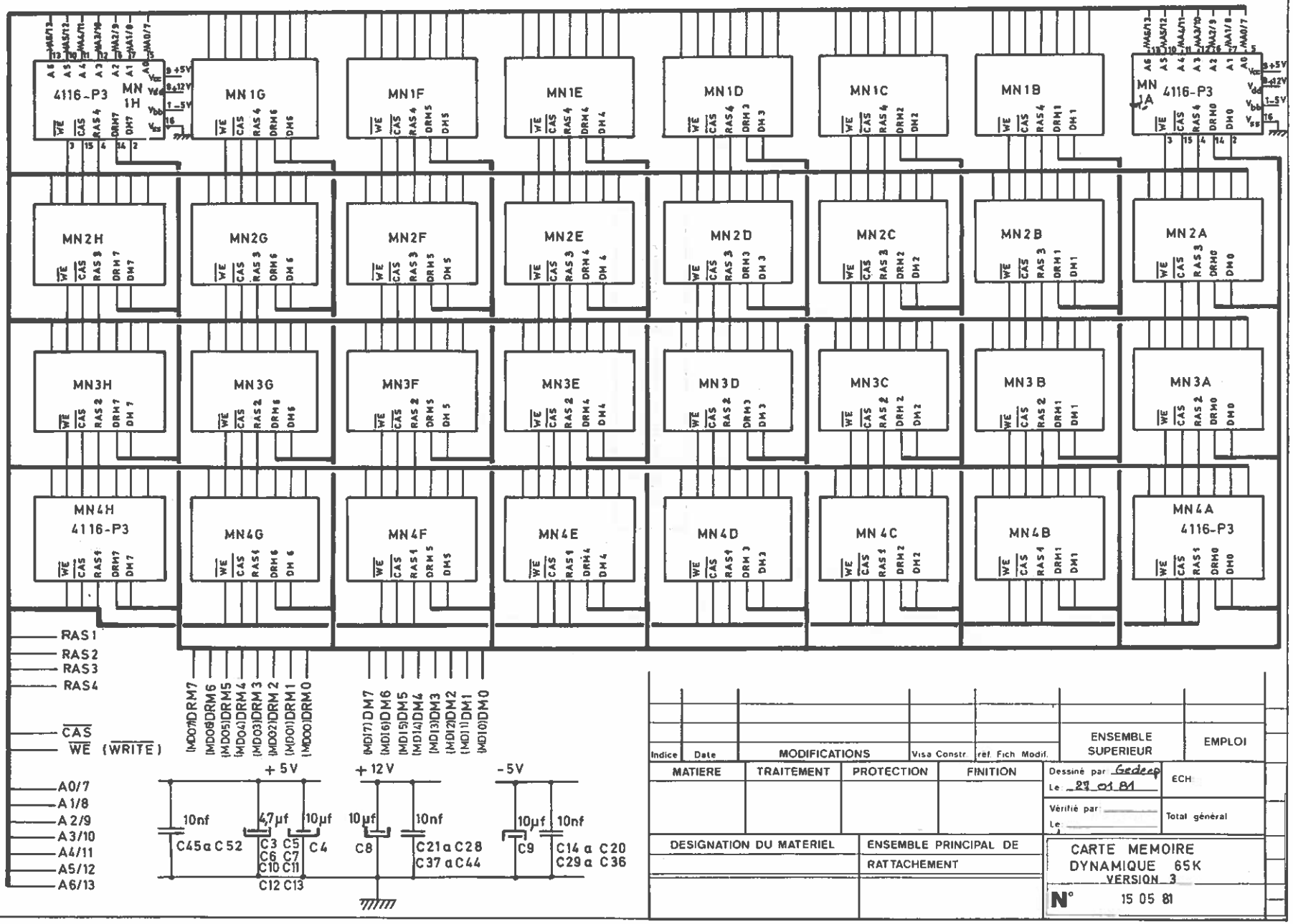


MN3D 6522
MN6B 2716
MN5D 74LS273
MN5C DAC 0800
MN3C DAC0800
MN4A 74LS122
MN1D CD4052
MN6C 74LS245
MN6A 74LS138
MN5B 74LS138
MN2D 74LS14
MN5A 74LS08
MN4C TL084
MN1A TL084
Sortie analog
Exterieur au C.I
Bar code
LECT

SCHEMAS DE LA CARTE FLOPPY


CARTE FLOPPY(2)
31 10 80
smt
WD 1791
74LS138
74LS74
74LS123
74LS193
NE555
74LS139
74LS07
74LS04
8T97
2N 2905
8Mhz
4Mhz
MFM
FM
1 à 31 de P1

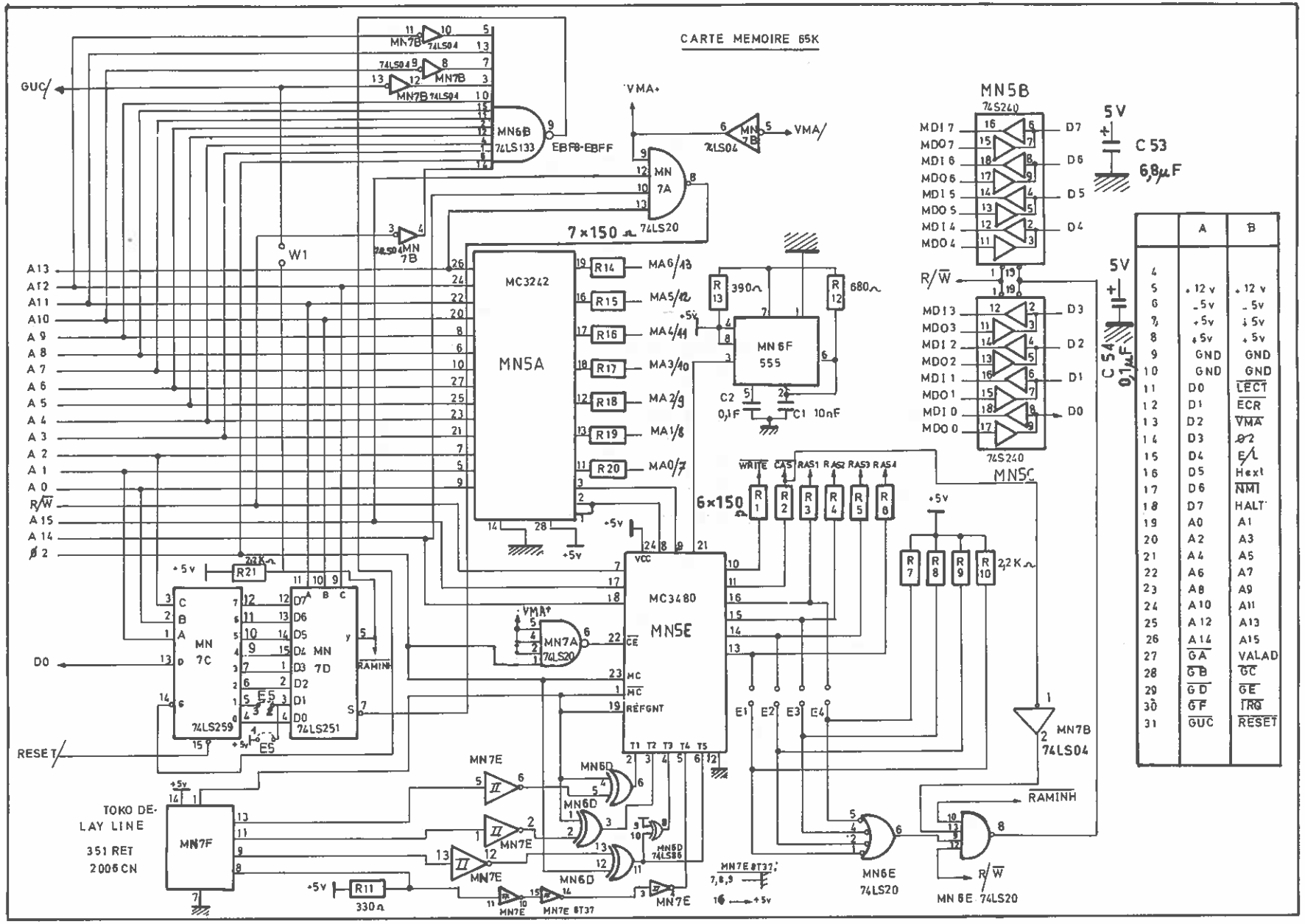
CARTE MEMOIRE DYNAMIQUE



Indice	Date	MODIFICATIONS	Visa Constr.	ref. Fich. Modif.	ENSEMBLE SUPERIEUR	EMPLOI

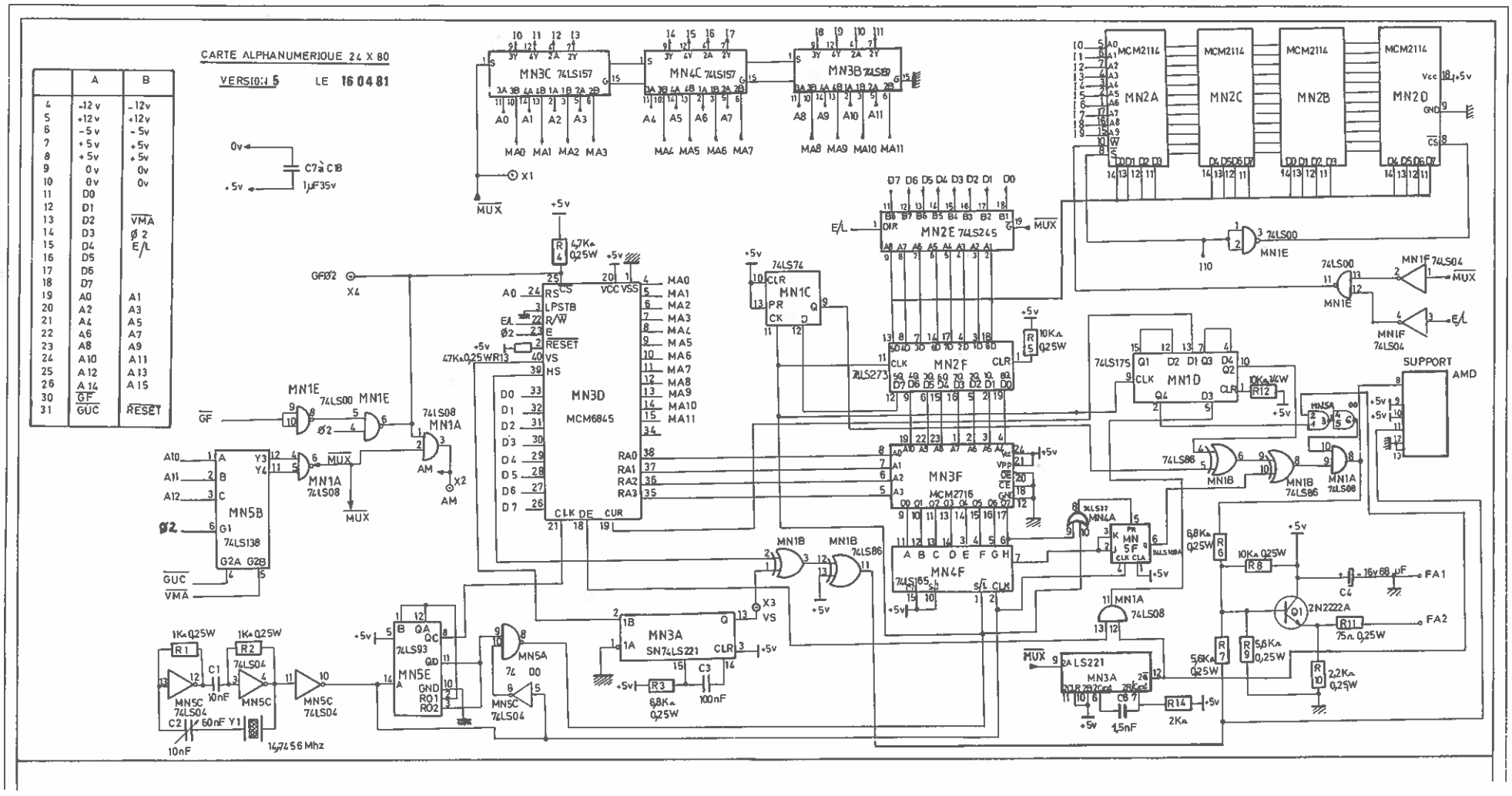
MATIERE	TRAITEMENT	PROTECTION	FINITION	Dessiné par Gedeon Le 23 01 81	ECH
				Vérifié par Le	Total général

DESIGNATION DU MATERIEL	ENSEMBLE PRINCIPAL DE RATTACHEMENT	CARTE MEMOIRE DYNAMIQUE 65K VERSION 3
		N° 15 05 81



	A	B
4		
5	+ 12 v	+ 12 v
6	- 5v	- 5v
7	+ 5v	+ 5v
8	+ 5v	+ 5v
9	GND	GND
10	GND	GND
11	D0	LECT
12	D1	ECR
13	D2	VMA
14	D3	Ø2
15	D4	E/L
16	D5	Hext
17	D6	NMI
18	D7	HALT
19	A0	A1
20	A2	A3
21	A4	A5
22	A6	A7
23	A8	A9
24	A10	A11
25	A12	A13
26	A14	A15
27	GA	VALAD
28	GB	GC
29	GD	GE
30	GF	IRQ
31	GUC	RESET

CARTE 24 x 80


CARTE ALPHANUMERIQUE 24 X 80
VERSION 5
LE 16 04 81
MN3C 74LS157
MN4C 74LS157
MN3B 74LS157
MCM2114
MN2A
MN2C
MN2B
MN2D
MN2E 74LS245
MN1C 74LS74
MN2F 74LS273
MN1D 74LS175
MN3D MCM6845
MN3F MCM2716
MN4F 74LS165
MN5B 74LS138
MN3A SN74LS221
MN5E 74LS93
MN5C 74LS04
MN1A 74LS08
MN1B 74LS86
MN1E 74LS00
MN1F 74LS04
MN4A 74LS37
SUPPORT AMD
2N2222A
14,7456 Mhz
MUX
GF
GUC
VMA
RESET
AM

Imprimé en France. — Imprimerie JOUVE, 18, rue Saint-Denis, 75001 PARIS